MULTI-OBJECTIVE SOFTWARE PROJECT COST ESTIMATION USING RECENT MACHINE LEARNING APPROACHES

DOĞAY DERYA

AUGUST 2023

ÇANKAYA UNIVERSITY

GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES

DEPARTMENT OF COMPUTER ENGINEERING

M.Sc. Thesis in

COMPUTER ENGINEERING

MULTI-OBJECTIVE SOFTWARE PROJECT COST ESTIMATION USING RECENT MACHINE LEARNING APPROACHES

DOĞAY DERYA

AĞUSTOS 2023

ABSTRACT

MULTI-OBJECTIVE SOFTWARE PROJECT COST ESTIMATION USING RECENT MACHINE LEARNING APPROACHES

DERYA, DOĞAY

M.Sc. in Computer Engineering

Supervisor: Assoc. Prof. Dr. Tansel DOKEROGLU

August 2023, 122 pages

Software projects are gaining strategic importance day by day, even in the daily operations of companies in various sectors. With the increasing need, many companies develop software by creating projects both within their own structure and for the needs of different sectors. Accurately estimating the workforce needed for software projects is crucial to accurately estimating project costs and ensuring timely completion.

Since the 1970s, the field of software effort estimation has been the subject of extensive research in the literature. While non-algorithmic methods such as expert opinion were used in the beginning, as the problems became more complex and technology and hardware features diversified, the need for different solution approaches emerged. To overcome these difficulties, algorithmic methods such as regression and model-based estimation have been developed. In recent years, however, with advances in technology, especially in the last decade, there has been an escalating interest in applying Machine Learning-based models and Artificial Intelligence to software cost estimation.

The focus of this study is to explore Machine Learning based prediction methods in the context of software projects. The aim is to analyze their effectiveness by investigating how these methods can improve software cost estimation.

Keywords: Software Cost Estimation, Software Effort Estimation, Artificial Intelligence, Machine Learning, Feature Selection

ÖZET

GÜNCEL MAKİNE ÖĞRENME YAKLAŞIMLARI İLE ÇOK AMAÇLI YAZILIM PROJESİ MALİYET TAHMİNLEMESİ

DERYA, DOĞAY

Bilgisayar Mühendisliği Yüksek Lisans

Danışman: Doç. Dr.Tansel DÖKEROĞLU

Ağustos 2023, 122 sayfa

Yazılım projeleri, çeşitli sektörlerdeki şirketlerin günlük operasyonlarında dahi günden güne stratejik önem kazanmaktadır. Artan ihtiyaçla birçok şirket gerek kendi bünyesinde, gerekse farklı sektörlerin ihtiyacına yönelik olarak projeler yaratarak yazılımlar geliştirmektedir. Yazılım projeleri için ihtiyaç duyulan işgücünü doğru tahmin etmek, proje maliyetlerini doğru tahmin etmek ve zamanında tamamlanmasını sağlamak için çok önemlidir.

1970'lerden bu yana, yazılım efor tahmini alanı, literatürde kapsamlı araştırmaların konusu olmuştur. Başlangıçta uzman görüşü gibi algoritmik olmayan yöntemler kullanılırken, sorunlar karmaşıklaştıkça, teknoloji ve donanım özellikleri çeşitlendikçe farklı çözüm yaklaşımlarına olan ihtiyaç da ortaya çıkmıştır. Bu zorlukların üstesinden gelmek için regresyon ve model tabanlı tahmin gibi algoritmik yöntemler geliştirilmiştir. Son yıllarda ise, özellikle son on yılda olmak üzere teknolojideki gelişmelerle birlikte, Makine Öğrenimi tabanlı modelleri ve Yapay Zekayı yazılım maliyet tahminine uygulamaya yönelik artan bir ilgi olmuştur.

Bu çalışmanın odak noktası, yazılım projeleri bağlamında Makine Öğrenimi tabanlı tahmin yöntemlerini keşfetmektir. Amaç, bu yöntemlerin yazılım maliyet tahminini nasıl iyileştirebileceğini araştırarak, etkinliklerini analiz etmektir.

Anahtar Kelimeler: Yazılım Maliyet Tahmini, Yazılım Efor Tahmini, Yapay Zeka, Makine Öğrenimi, Özellik Seçimi

ACKNOWLEDGEMENT

I would like to thank my supervisor, Associate Professor Tansel DOKEROGLU for guiding me in completing this thesis.

Furthermore, I would like to offer my special thanks to my dear husband, Osman Berkcan DERYA, whose unwavering support throughout my master's degree and in every aspect of life has been invaluable.

Also, I would like to express my heartfelt appreciation to my family who have always supported me throughout my life. I am grateful to each and every one of you.

TABLE OF CONTENTS

STATEMENT OF NONPLAGIARISM ..... **III**
ABSTRACT ..... **IV**
ÖZET ..... **VI**
ACKNOWLEDGEMENT ..... **VIII**
LIST OF TABLES ..... **XIII**
LIST OF FIGURES ..... **XVI**
LIST OF ABBREVIATIONS ..... **XVII**
CHAPTER I **1**
INTRODUCTION **1**
1.1 RESEARCH OBJECTIVES 2
1.2 CONTRIBUTIONS OF THE THESIS 2
CHAPTER II **3**
OVERVIEW **3**
2.1 THE IMPORTANCE OF COST ESTIMATION IN SOFTWARE PROJECTS 3
2.2 SOFTWARE EFFORT ESTIMATION METHODS 5
2.2.1 Algorithmic Model 7
2.2.1.1 COCOMO Model 7
2.2.1.2 Putnam's Model 9
2.2.1.3 Function-Point Based Model 9
2.2.2 Non-Algorithmic Model 10
2.2.2.1 Expert Judgment 10
2.2.2.2 Estimation by Analogy 10
2.2.2.3 Top-Down Estimating Method 11
2.2.2.4 Bottom-up Estimating Method 11
2.2.2.5 Parkinson's Law 11
2.2.2.6 Pricing to Win 11
2.2.3 Machine Learning Model 12
2.3 MACHINE LEARNING ALGORITHM SELECTION METHOD 13
2.4 FEATURE SELECTION 14
2.5 LITERATURE REVIEW 16

CHAPTER III 19
METHODOLOGY 19
3.1 DATASET 19
3.1.1 Finnish 19
3.1.2 China 20
3.1.3 Kemerer 20
3.1.4 Maxwell 21
3.2 APPLICATION PLATFORM 22
3.3 MACHINE LEARNING ALGORITHMS 27
3.3.1 Linear regression 28
3.3.2 Multilayer Perceptron 29
3.3.3 SMOreg (Sequential Minimal Optimization Regression) 30
3.3.4 IBk (Instance-Based learning with k parameter) 32
3.3.5 KStar (Instance-based classifier) 33
3.3.6 Bagging 34
3.3.7 M5P (M5 Model trees) 35
3.3.8 RandomForest 37
3.3.9 Random Tree 38
3.4 FEATURE SELECTION TECHNIQUES 39
3.4.1 Attribute Evaluators for Feature Selection 40
3.4.1.1 Correlation Based Feature Selection (CFS) 40
3.4.1.2 ClassifierAttEval 40
3.4.1.3 Corr. Att.Evaluation 41
3.4.1.4 Relief Att.Evaluation 41
3.4.2 Search Methods for Feature Selection 42
3.4.2.1 Random Search 42
3.4.2.2 Particle Swarm Optimization (PSO) 42
3.4.2.3 Genetic Algorithm (GA) 43
3.4.2.4 Ranker 44
3.5 PERFORMANCE MEASURES 45
3.5.1 Correlation Coefficient 45
3.5.2 Mean Absolute Error (MAE) 45
3.5.3 Relative Absolute Error (RAE) 46
3.5.4 Mean Relative Error (MRE) 47

3.5.5 Mean Magnitude of Relative Error (MMRE)..... 47
3.5.6 Percentage of Estimations (PRED (0.25))..... 47
CHAPTER IV..... 48
FINDINGS..... 48
4.1 MACHINE LEARNING MODELS EXPERIMENTS AND RESULTS 49
4.2 BY USING FEATUE SELECTION MACHINE LEARNING MODELS EXPERIMENTS AND RESULTS..... 52
4.2.1 Finnish Dataset Cost Estimation Results with Feature Selection Methods..... 53
4.2.1.1 CfsSubset Evaluation and Random Search Method Results 53
4.2.1.2 CfsSubset Evaluation and Particle Swarm Optimization (PSO) Method Results..... 54
4.2.1.3 CfsSubset Evaluation and Genetic Algorithm (GA) Method Results 55
4.2.1.4 ClassifierAtt. Evaluation and Ranker Search Method Results 56
4.2.1.5 Corr. Att. Evaluation and Ranker Search Method Results 58
4.2.1.6 Relief Att.Evaluation and Ranker Search Method Results..... 59
4.2.2 China Dataset Cost Estimation Results with Feature Selection Methods..... 60
4.2.2.1 CfsSubset Evaluation and Random Search Method Results 60
4.2.2.2 CfsSubset Evaluation and Particle Swarm Optimization (PSO) Method Results..... 61
4.2.2.3 CfsSubset Evaluation and Genetic Algorithm (GA) Method Results 63
4.2.2.4 ClassifierAtt. Evaluation and Ranker Search Method Results 64
4.2.2.5 Corr. Att. Evaluation and Ranker Search Search Method Results 65
4.2.2.6 Relief Att. Evaluation and Ranker Search Method Results..... 66
4.2.3 Maxwell Dataset Cost Estimation Results with Feature Selection Methods..... 68
4.2.3.1 CfsSubset Evaluation and Random Search Method Results 68
4.2.3.2 CfsSubset Evaluation and Particle Swarm Optimization (PSO) Method Results..... 69
4.2.3.3 CfsSubset Evaluation and Genetic Algorithm (GA) Method Results 70

4.2.3.4 ClassifierAttEval Evaluation and Ranker Search Method Results 72
4.2.3.5 Corr. Att. Evaluation and Ranker Search Method Results 73
4.2.3.6 Relief Att. Evaluation and Ranker Search Method Results 75
4.2.3.6.1 First Experiment-By Removing Last 3 Features 75
4.2.3.6.2 Second Experiment-By Removing All the Tagged Feature with Negative Coefficient 77
4.2.4 Kemerer Dataset Cost Estimation Results with Feature Selection Methods 78
4.2.4.1 CfsSubset Evaluation and Random Search Method Results 78
4.2.4.2 CfsSubset Evaluation and Particle Swarm Optimization (PSO) Method Results 79
4.2.4.3 CfsSubset Evaluation and Genetic Algorithm (GA) Method Results 81
4.2.4.4 ClassifierAtt. Evaluation and Ranker Search Method Results 82
4.2.4.5 Corr. Att. Evaluation and Ranker Search Method Results 83
4.2.4.6 Relief Att. Evaluation and Ranker Search Method Results 84
4.3 ANALYSIS OF FINDINGS 85
CHAPTER V 91
CONCLUSION 91
REFERENCES 100

LIST OF TABLES

Table 2.1: Comparison of Algorithmic, Non.Algorithmic and Machine Learning Approaches 7
Table 2.2: Machine Learning Techniques 13
Table 3.1: Information of Datasets 19
Table 3.2: Finnish Dataset Statistics 20
Table 3.3: China Dataset Statistics 20
Table 3.4: Kemerer Dataset Statistics 21
Table 3.5: Maxwell Dataset Statistics 22
Table 4.1: Performance Measures of Models Constructed with Finnish Original Feature 50
Table 4.2: Performance Measures of Models Constructed with China Original Feature 50
Table 4.3: Performance Measures of Models Constructed with Maxwell Original Feature 51
Table 4.4 Performance Measures of Models Constructed with Kemerer Original Feature 51
Table 4.5 Performance Measures of Models with Finnish Original Feature and Selected Feature Set by RandomSearch 53
Table 4.6: Performance Measures of Models with Finnish Original Feature and Selected Feature Set by PSO 54
Table 4.7: Performance Measures of Models with Finnish Original Feature and Selected Feature Set by GA 56
Table 4.8: Performance Measures of Models with Finnish Original Feature and Selected Feature Set by ClassifierAttEvaluation + Ranker 57
Table 4.9: Performance Measures of Models with Finnish Original Feature and Selected Feature Set by Corr. Att.Evaluation + Ranker 58
Table 4.10: Performance Measures of Models with Finnish Original Feature and Selected Feature Set by Relief Att.Evaluation + Ranker 59

Table 4.11: Performance Measures of Models with China Original Feature and Selected Feature Set by CFS+ RandomSearch 61
Table 4.12: Performance Measures of Models with China Original Feature and Selected Feature Set by CFS+ PSO 62
Table 4.13: Performance Measures of Models with China Original Feature and Selected Feature Set by CFS+ GA 63
Table 4.14: Performance Measures of Models with China Original Feature and Selected Feature Set by ClassifierAttEvaluation + Ranker 64
Table 4.15: Performance Measures of Models with China Original Feature and Selected Feature Set by Corr. Att.Evaluation + Ranker 66
Table 4.16: Performance Measures of Models with China Original Feature and Selected Feature Set by Relief Att.Evaluation + Ranker 67
Table 4.17: Performance Measures of Models with Maxwell Original Feature and Selected Feature Set by CFS+ RandomSearch 68
Table 4.18: Performance Measures of Models with Maxwell Original Feature and Selected Feature Set by CFS+ PSO 70
Table 4.19: Performance Measures of Models with Maxwell Original Feature and Selected Feature Set by CFS+ GA 71
Table 4.20: Performance Measures of Models with Maxwell Original Feature and Selected Feature Set by ClassifierAttEvaluation + Ranker 73
Table 4.21: Performance Measures of Models with Maxwell Original Feature and Selected Feature Set by Corr. Att.Evaluation + Ranker 74
Table 4.22: Performance Measures of Models with Maxwell Original Feature and 24 Selected Feature Set by Relief Att.Evaluation + Ranker 76
Table 4.23: Performance Measures of Models with Maxwell Original Feature and 14 Selected Feature Set by Relief Att.Evaluation + Ranker 77
Table 4.24: Performance Measures of Models with Kemerer Original Feature and 14 Selected Feature Set by RandomSearch 78
Table 4.25: Performance Measures of Models with Kemerer Original Feature and Selected Feature Set by CFS+ PSO 80
Table 4.26: Performance Measures of Models with Kemerer Original Feature and Selected Feature Set by CFS+ GA 81
Table 4.27: Performance Measures of Models with Kemerer Original Feature and Selected Feature Set by ClassifierAttEvaluation + Ranker 82

Table 4.28: Performance Measures of Models with Kemerer Original Feature and Selected Feature Set by Corr. Att.Evaluation + Ranker 83
Table 4.29: Performance Measures of Models with Kemerer Original Feature and Selected Feature Set by Relief Att.Evaluation + Ranker 85
Table 4.30: The Outcomes of The Models Constructed Using the Finnish Dataset. 87
Table 4.31: The Outcomes of The Models Constructed Using the China Dataset ... 88
Table 4.32: The Outcomes of The Models Constructed Using the Maxwell Dataset 89
Table 4.33: The Outcomes of The Models Constructed Using the Kemerer Dataset90
T**able 5.1:** Finnish Dataset 's Highest and Lowest Performance Measures Before and After Feature Selection 92
Table 5.2: China Dataset 's Highest and Lowest Performance Measures Before and After Feature Selection 93
Table 5.3: Maxwell Dataset 's Highest and Lowest Performance Measures Before and After Feature Selection 93
Table 5.4: Kemerer Dataset 's Highest and Lowest Performance Measures Before and After Feature Selection 93
Table 5.5: Model Performance Results with Feature Selection 96
Table 5.6: Comparative Analysis of Gained Results with Literature 98

LIST OF FIGURES

Figure 3.1: The opening screen of WEKA 23
Figure 3.2: Explorer Tab Content of WEKA..... 24
Figure 3.3: Classify Section of WEKA 26
Figure 3.4: Classifier Details in WEKA 26
Figure 5.1: Comparison Graph for Actual and Predicted Effort of Finnish Dataset 94
Figure 5.2: Comparison Graph for Actual and Predicted Effort of China Dataset... 94
Figure 5.3: Comparison Graph for Actual and Predicted Effort of Kemerer Dataset 94
Figure 5.4: Comparison Graph for Actual and Predicted Effort of Maxwell Dataset 95
Figure 5.5: Relation of Accuracy to Number of Features for Fininsh..... 96
Figure 5.6: Relation of Accuracy to Number of Features for China 97
Figure 5.7: Relation of Accuracy to Number of Features for Maxwell..... 97
Figure 5.8: Relation of Accuracy to Number of Feures for Kemerer 97

LIST OF ABBREVIATIONS

GA	: Genetic Algorithm
PSO	: Particle Swarm Optimization
SVM	: Support Vector Machine
CFS	: Correlation Based Feature Selection
WEKA	: Waikato Environment for Knowledge Analysis
COCOMO	: Constructive Cost Estimation Model
KDLOC	: Thousands of Delivered Lines of Source Code
PROMISE	: Predictor Models in Software Engineering
MAE	: Mean Absolute Error
RAE	: Relative Absolute Error
MMRE	: Mean Magnitude of Relative Error
PRED	: Percentage of Estimations
ML	: Machine Learning

CHAPTER I
INTRODUCTION

Effective project management becomes indispensable for software projects that increase in importance and scope in parallel with the increase in trust in electronic technologies.

Project predictability is a critical factor in software project management, as it makes possible to mitigate potential risks by enabling precise cost and workforce planning. Accurate software effort estimation is a crucial component of software development, providing essential inputs for feasibility analysis, planning, budgeting, bidding. Deviating significantly from the required effort causes losses in terms of cost and quality. Thus, it is particularly important to estimate development time accurately in the highly competitive software industry, where quality is highly valued.

Currently, the most prevalent methods for effort estimation rely on expert judgment. However, these methods may lack reliability as they can be influenced by various factors. Additionally, relying solely on human judgment can be burdensome and time-consuming when dealing with numerous estimation items.

In recent years, the dynamic nature of the market has led to a growing adoption of agile methods in software project management, replacing traditional approaches. Within the agile project management methodology, the most commonly used metric for effort prediction is story scores. Presently, these estimations are typically made intuitively by relevant individuals for each request, with subsequent review by unit managers. However, this process lacks consistency and continuity, despite consuming significant human resources.

The objective of this study is to propose a machine learning-based approach for effort estimation, aiming to accurately and swiftly predict effort. The study will handle machine learning approach that establish models by learning from past data to predict development efforts. Furthermore, innovative feature selection techniques will be employed to enhance the accuracy and effectiveness of the estimation process.

In the study, Linear Regression, Multilayer Perceptron, Bagging, SMOreg, IBk, KStar, RandomTree, Random Forest, M5P algorithms included in WEKA (Waikato Environment for Knowledge Analysis) tool, China, Finnish, Kemerer, Maxwell datasets were both trained and tested with the same datasets by choosing the 10-fold cross-validation technique. In the first part of the application, the results are obtained with the original feature set. In the second part, it is aimed to apply feature selection by analyzing low-impact features and by focusing on increasing the performance of model outputs and to obtain models that prevent overfitting and not to include unnecessary inputs. In feature selection, hybrid approaches of evaluation and search methods are used together in different configurations. Among the search methods, methods such as RandomSearch, PSO, GA, Ranker are selected and the capabilities of these methods in searching optimized subsets are utilized.

1.1 RESEARCH OBJECTIVES

In this thesis, investigated to deal with problems on software cost estimation subjects are described below.

a. How is it possible to gain high performance-low cost application of machine learning algorithm as a trendy on estimation techniques in last decades?

b. Is selection of features which are using as input of models effective on result and how optimized subset of features is essential on machine learning algorithms?

c. Which search techniques is shows success for finding optimized subset of features?

1.2 CONTRIBUTIONS OF THE THESIS

a. High-performance approaches were emphasized by training, testing and comparing 9 different machine learning algorithms with 6 different feature selection methods in four different datasets.

b. With the WEKA tool, which is easily accessible due to its open source nature, alternatives to low execution time, high predictive models have been presented.

c. When the estimation error rates obtained were compared with the results in the literature, it was observed that successful performances were achieved.

CHAPTER II
OVERVIEW

2.1 THE IMPORTANCE OF COST ESTIMATION IN SOFTWARE PROJECTS

The using area and the volume of needs it is meeting of software is constantly increasing nowadays. Due to the highly competitive environment, companies are being forced to generate software projects on budget and timely. The precision and reliability of the effort estimation of software projects is also gain importance for the competitiveness of software companies. This precise and reliable forecast, a solid foundation allows to laid for the production of quality and timely software that will enable software companies to compete.

Knowing the approximate cost of a project at the beginning of the project is important for the reasons for starting the project. The customers of the project or the top management decides whether or not to carry out the project according to the predictive values. Incorrect estimations make the institutions or organizations in the position of customers economically and strategically affects. For example, 60% of large projects exceeded their project budgets. It has been observed that some projects were never completed due to a 15% cost overrun [1].

When the failures of software projects around the world are examined, the reasons are that the constraints of the project cannot be determined exactly, the correct cost estimation cannot be made, the changing customer expectations cannot be met, the technical aspects of the employees are insufficient, and the customer's expectations are not fully reflected. However, most of the software projects is collapsing due to incorrect cost estimation and timeout. Software costs is increasing rapidly due to wrong estimates. Therefore, these important problems are raising both in the country and in the world.

The increasing need for project planning and management requires software project managers to conduct more careful analysis. Project planning, which is the initial phase of all project management processes, gains significance due to the requirement for all subsequent work to be executed in accordance with this plan.

Therefore, various techniques have been proposed to assist in the planning stages. When it comes to software project management, planning is based on the estimation of the required effort/time value to develop the program or service to be developed. Once this value is calculated and predicted, other elements of planning (such as budget, timeline, etc.) can be determined. The process of estimating the resources required for a software project is referred to as software effort estimation. Effort estimation becomes an input for calculating the resources and costs needed for the development of the system.

The accuracy of effort estimation for a project has a direct impact on project success. Plans are shaped according to the estimates made and other important elements, budget, calendar, procurement processes are determined accordingly. Therefore, one of the most important issues for a project manager or project team is to estimate effort with a high percentage of success. If the actual time-cost exceeds what was planned, the project may fail; in the opposite case, a problem such as improper use of resources may occur. Project managers are in search of helpful techniques and methods so that the effort estimation can be made accurately.

Software effort estimation is difficult, mainly for two reasons. The first reason is that software is intangible and is outside the definition of conventional physical product. The second reason is that the software development job is an intellectual rather than a physical job. Software startups are easy, but as the software size increases, the workforce estimation process becomes more difficult. It is possible to write a program that is close to a few thousand lines in a week. But then the speed slows down as the program grows. When this program reaches several tens of thousands of lines, adding a line is worth a few days' effort, maybe even months. Therefore, it has become difficult to follow the side effects of the addition [2]. The dynamically fluctuating technology environment in the software development industry also makes effort estimation confusing [3].

The workforce estimation depends on many parameters such as the technologies used, the experience of the software developers, the project history of the software team in the same work area, and the detailed features of the functions created. Software workforce estimation is a complex field because of the multiplicity of parameters and the fact that the relationships between these parameters cannot always be accurately predicted. As current challenges persists, techniques are evolving to

remove or minimize them. Many techniques and methods have been proposed that can increase the success rate of effort estimation values.

It has been stated that the history of effort estimation dates back to the 1960s [4], but studies mostly concentrate on 1990 and later. Each study is then divided into categorical areas, and in this way, a header is provided for the solution method needed. In the field of effort estimation, where different approaches and solutions are proposed, the studies are divided into certain groups and the effort estimation field is divided into certain main categories by classification.

2.2 SOFTWARE EFFORT ESTIMATION METHODS

The development cost is basically the project labor cost it includes, so the labor account is used in both cost and software project timing estimation. In the software effort estimation research literature, there is often no distinction between effort and cost. This is mainly due to the fact that in software development, almost all costs are personnel costs, which are directly tied to effort. However, in global software development, cost rates may differ in different areas. This means that effort in one region may result in higher costs than effort elsewhere [5]. However, in this research effort and costs will be used synonymously unless otherwise stated.

The effort and time required for a software to be realized can be affected by many factors. Various criteria are given in many sources for cost estimation. For example, the complexity of the software to be developed, the experience or expertise of the institutions and organizations participating in the development and the team members who made the development, the technology and hardware infrastructure used, quality requirements, customer participation, time may affect the cost of the software. However, since the collection of values for the proposed criteria is very difficult and demanding, it is recommended to use some of them or the most effective set. One strategy for a company could be to start data collection including the standard factors and in addition collect factors specific to the organization. Thus, the benefits of the global data could be utilized, as well as organizational characteristics could be considered. [6].

The history of the models developed to predict the software development effort dates back to the 1970s. In a study conducted in 2000, current estimation methods were classified under six headings [7]:

Model-Based Techniques: Techniques using mathematical equations, based on historical data and a theory. (Putnam SLIM, Function Point, Estimacs, COCOMO, Checkpoint, SEER)

Expert-Based Techniques: Techniques based on obtaining the opinions of experts with knowledge and experience in a field. (Delphi, Rule)

Learning-Based Techniques: Automated systems that learn by themselves using previous experience and data. (Neural, Genetic, Case-based)

Dynamic-Based Techniques: Techniques based on the prediction that effort factors can change during the software development process and adapting to this. (Abdel, Hamid, Madnick)

Regression-Based Techniques: Techniques that work together with model-based techniques and rely on statistical regression approaches. (OLS, Robust)

Mixed Techniques: Techniques that combine several of the above-mentioned techniques. (Bayesian, COCOMO II)

In a study conducted in 2002, software cost estimation methods were basically examined under two main groups [8]: “Algorithmic Models”, “Non-Algorithmic Models”. “Learning-Based Models” can be added as a third category to this classification with the studies carried out in the last 10 years. Approaches below provides an analysis of algorithmic, non-algorithmic, and machine learning methods for cost estimation, highlighting the strengths and weaknesses of different cost estimation.

Table 2.1: Comparison of Algorithmic, Non-Algorithmic and Machine Learning Approaches

Approach	Description	Strengths	Weaknesses
Algorithmic	Rely on predefined formulas or algorithms to estimate costs.	Simplicity and transparency	Reliance on assumptions
		Well-defined formulas/algorithms	Limited flexibility
		Quick estimation process	Lack of adaptability
Non-algorithmic	Rely on expert judgment and experience for cost estimation.	Utilizes expert knowledge and experience in the estimation process	Subjectivity and bias
		Incorporates human insights	Difficulty in quantification
		Can capture unique project factors	Time-consuming
			Lack of reproducibility
Machine Learning	Use data-driven models to estimate costs based on historical data.	Ability to handle complex data	Dependency on quality of data
		Can learn from historical data	Overfitting and generalization issues
		Adaptability to different projects	Need for substantial data

2.2.1 Algorithmic Model

Algorithmic cost model involves using formula to estimate cost of software referring to predictions of project size, the number of programmers, and other process and factors of production. A cost model based on algorithms can be created by examining the expenses and attributes of finished projects and identifying the equation that best matches real-life observations.

2.2.1.1 COCOMO Model

COCOMO was introduced in 1981 and is one of the most widely used software estimation models globally. COCOMO estimates the cost and schedule of a software product referring to the size of software product.

The procedure of this model comprises the following steps:

● Generate an initial estimate of the development effort by analyzing numerous delivered lines of source code (KDLOC).

● Identify a collection of 15 scaling factors based on various characteristics of the project.

● Compute the effort estimate by multiplying the initial estimate with all the scaling factors, i.e., by combining the values obtained in the first and second steps.

● The initial estimate (also known as the nominal estimate) is determined using a formulation resembling static single-variable models, where KDLOC is employed as the size metric. The following equation is utilized to determine the initial effort in person-months, represented as Ei.

$$Ei = a * (KDLOC)^b \quad (2.1)$$

The values of the constants a and b depend on the project type.

In COCOMO, projects are categorized into three types:

o Organic

o Semidetached

o Embedded

Organic:

A development project can be classified as organic if the project involves developing a well-understood application program, the size of the development team is reasonably small, and the team members are experienced in developing similar types of projects. Examples of this type of projects are simple business systems, straightforward inventory management systems, and data processing systems.

Semidetached:

A development project can be classified as semidetached if the development consists of a mix of experienced and inexperienced staff. Team members may have limited experience with similar systems but may be new to certain aspects of the project being developed. Examples of semidetached systems include developing a new operating system (OS), a Database Management System (DBMS), and complex inventory management systems.

Embedded:

A development project is classified as embedded if the software being developed is tightly coupled to complex hardware or if stringent rules on the operational procedure exist. Examples include ATMs and air traffic control systems. According to Boehm, it is possible to do software cost estimation with three phases:

- Basic Model
- Intermediate Model
- Detailed Model

2.2.1.2 Putnam's Model

Putnam was suggested and improved based on labor distribution and research of various software projects. Putnam's Model ‘s main equation is:

$$S = E * (Effort)^{(1/3)} * (td)^{(4/3)} \quad (2.2)$$

In Putnam's Model, the environment factor E represents the environment capability, while td denotes the delivery time. The measures of effort and S are expressed in person-years and lines of code, respectively. Additionally, Putnam's Model introduces an additional equation to calculate the effort involved.

$$Effort = D0 * (td)^3 \quad (2.3)$$

The manpower build-up factor, denoted as D0, ranges from 8 to 27 depending on whether the software is newly developed or rebuilt.

2.2.1.3 Function-Point Based Model

In 1983, Albrecht presented the Function Point Metric, a measurement designed to assess project effectiveness. This model incorporates five variables, which are as follows:

- User Inputs,
- User Outputs,
- Logic Files,
- Inquiries, and
- Interfaces,

to assess the size of the project. The complexity of a function is determined based on its simplicity or complexity, measured on a scale of 1, 2, or 3. Each variable is assigned a weight ranging from 3 to 15.

2.2.2 Non-Algorithmic Model

The Non-Algorithmic Model, in contrast to Algorithmic techniques, relies on analytical examinations and inference [9]. To use Non-Algorithmic techniques, it is necessary to have information about previous projects that are similar to the project being estimated. Typically, the estimation process in these techniques involves analyzing past datasets.

2.2.2.1 Expert Judgment

"Master Judgment" is a technique where evaluations are based on a specific set of criteria and the expertise acquired in a particular field of knowledge, application domain, product area, specific discipline, or industry. So on such mastery might be furnished by any gathering or individual with particular schooling, information, expertise, experience, or training [10]. The knowledge base for expert judgment can come from members of the project team, multiple individuals within the team, team leaders, or project managers. However, expert judgment often requires expertise that is not available within the project team, so it is common to seek external individuals or groups with a specific relevant skill set or knowledge base for consultation. Any group or individual with specialized knowledge or training can offer such expertise, and it can be acquired from a variety of sources. Such as customers or sponsors, professional and technical associations, industry groups, subject matter experts (SMEs), project management offices (PMOs), and suppliers [11].

2.2.2.2 Estimation by Analogy

Estimation by Analogy involves determining the cost of a project by comparing it to a similar project in the same application domain. In order to make this estimation, specific conditions must be satisfied. These conditions encompass gathering data from previous and ongoing projects, such as weekly work hours per team member, project completion costs, the similarity between the current project and previous ones, and the existence of modules or activities in past projects resembling those in the current one. If the current project is novel and lacks prior similar projects, alternative methods may be necessary. The selected data from past projects are used in conjunction with the expertise of the project manager and the estimation team to ensure informed judgment in the estimations.

2.2.2.3 Top-Down Estimating Method

The Top-Down estimating method, commonly referred to as the Macro Model, entails obtaining a comprehensive cost estimation is determined by taking into account the software project's overarching characteristics. This estimation is then further broken down into various low-level mechanisms or components. The Putnam model is often employed as a means for implementing this approach. The Top-Down method is particularly suitable for early cost estimation when only high-level information about the project is available. In the early phase of the software cost estimation, top-down is very useful because there is no detailed information available [12].

2.2.2.4 Bottom-up Estimating Method

The Bottom-Up estimating method involves assessing the cost of each individual software component and then combining the results to arrive at an estimated cost for the entire project. The Bottom-Up method focuses on building the estimate of a system by considering the information gathered about the small software components and their relationships. The strategy utilizing this methodology is COCOMO's point by point model [12].

2.2.2.5 Parkinson's Law

Parkinson's Law states that the project costs will expand to consume the available resources. This means that the cost estimation is based solely on the resources at hand, such as hardware, software, power, space, etc. For example, if the customer requires the software to be completed within 10 months and only 4 people are available, the cost estimate would be calculated as 10 * 4, resulting in 40 person-months. In this strategy, consider just the client spending plan, and not the number of people is needed for building up the product or some other assets for estimation [13].

2.2.2.6 Pricing to Win

Pricing to Win is a method that relies solely on the customer's budget rather than the functionality of the software. In this approach, only the customer's budget is considered, without considering the number of people required for software development or any other resources. For instance, if the client can burn through 40 man/month, yet real exertion is 60 man/month. At that point assessor is approached to

alter in assessment and fit in 40 man/months. Again, this strategy isn't better for acceptable programming rehearses [13].

2.2.3 Machine Learning Model

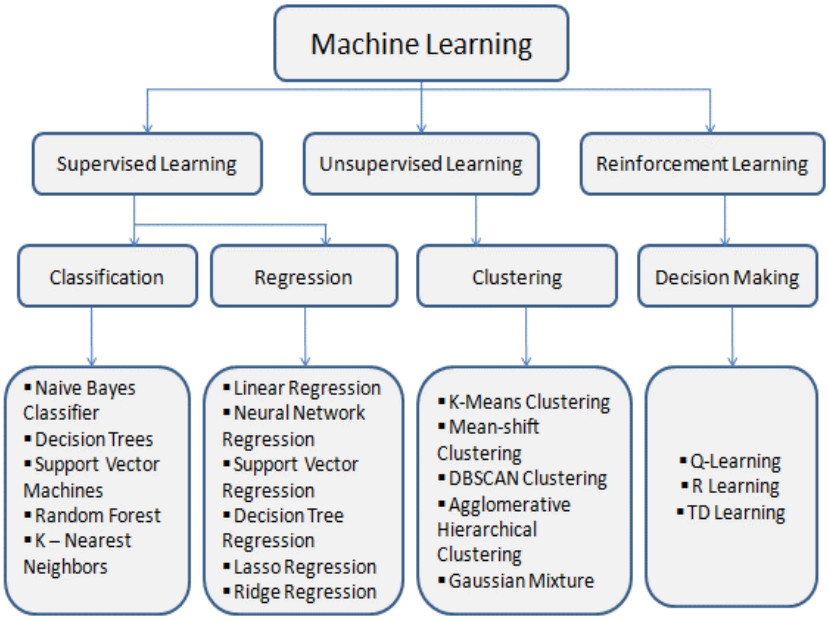
Most cost estimation procedures rely on statistical methods, which may lack explanation and reliable results. However, machine learning techniques can be suitable as they can enhance the accuracy of estimation by training assessment rules and iterating the process.

Machine learning is a scientific field that deals with ways in which machines can learn through experience. Learning is the fundamental characteristic of what is commonly referred to as intelligent entities. The goal of machine learning is to build computer systems that can learn. Machine learning can be seen as a collection of methods that acquire knowledge from existing data using various mathematical techniques and can make predictions based on this acquired knowledge. Machine learning algorithms extract more meaningful data representations that represent the raw data before using them. For example, during the training phase, the existing training data is processed using mathematical methods used in the algorithm, and a model is derived. With this model, predictions can be made about any test data. Machine learning algorithms can be broadly categorized into three main categories, namely supervised learning, unsupervised learning, and reinforcement learning.

- Supervised Learning
- Unsupervised Learning
- Reinforcement Learning

In supervised learning, there is a teacher during the learning process. In this process, input values and expected outputs are already presented to the network. Examples of these methods include regression and classification algorithms. In unsupervised learning, there is no teacher during the learning process. Expected outputs are not presented to the network. The system learns by discovering and adapting the structural features in the input data model. Examples of these methods include clustering algorithms. In reinforcement learning, there is a teacher, but the expected output is not available in the network. Only whether the output is correct or incorrect is indicated. In this method, a penalty is given for incorrect output and a reward is given for correct output. Commonly used machine learning techniques are shown in Table 2.2.

Table 2.2: Machine Learning Techniques



2.3 MACHINE LEARNING ALGORITHM SELECTION METHOD

Because there are dozens of supervised, unsupervised and reinforcement machine learning algorithms, and the accuracy of the data-specific algorithm results that can be provided will vary, choosing the algorithm that will give the best results for solving the problem under consideration is a grueling process.

It is important to note that there is no universal best method that applies to all scenarios. Determining the right algorithm often just possible with trial and error. Evaluating multiple variables simultaneously, such as data variability, quality, accuracy, and parameter values, can be challenging, making it difficult to determine if an algorithm will perform effectively on a given dataset without experimentation. Nevertheless, the choice of algorithm also depends on factors such as the size and nature of the data, the desired insights to be gained, and how those insights will be utilized.

It is difficult to determine which technique gives more accurate results on which dataset. In the literature, a lot of research has been conducted in the field of machine education techniques. These findings suggest that machine learning techniques have the potential to provide more adequate prediction models for software development projects [14]. These studies have shown that machine learning methods can provide more adequate predictions models, especially in software development projects compared to traditional models.

In machine learning application, it offers various techniques and models that can be selected according to the size of the data being handled and the type of problem to be solved. A successful deep learning application requires huge amounts of data (thousands) to train the model and GPUs or graphics processing units to process the data quickly.

When choosing between machine learning and deep learning, it is necessary to consider whether you have a high-performance GPU and lots of tagged data. While not having any of these, it's more feasible to use machine learning instead of deep learning. Deep learning is usually more complex, so thousands are needed to get reliable results.

When machine learning is chosen, it will be possible to have the option to train the model on many different classifiers, and also be available to try which features can be selected to achieve the best results. Additionally, having the flexibility to try a combination of approaches and use different classifiers and features to see which arrangement works best for the data is possible. As a result, machine learning methods that can be used in a wide area will be progressed, as they allow working with more limited data, allowing use on less qualified hardware, and allowing easy application of hybrid methods.

2.4 FEATURE SELECTION

Estimating the cost in software projects relies on various factors, including the technology employed, the expertise of developers, the team's past project experiences in a similar domain, and the specific characteristics of the functions being developed. Software workforce estimation is a challenging task due to the multitude of parameters involved, and accurately predicting the relationships between these parameters is not always feasible. To address these ongoing challenges, techniques are continuously evolving to mitigate their impact. Numerous approaches and methods have been suggested to enhance the accuracy and success rate of effort estimation values.

In general, useful features are unpredictable, and features with low correlation and missing data can affect classification performance. Including low-impact variables in model training reduces the model's ability to generalize and may also reduce the overall accuracy of a classifier. Also, adding more variables to a model increases the overall complexity of the model. Therefore, deciding on the optimum features to include in model training is critical in obtaining a generically high-performing model.

Various techniques are used in various fields to eliminate unnecessary features.

Generally, useful features are unpredictable, and features with low correlation and additional data can affect classification performance. Including low-impact variables in model training reduces the model's ability to generalize and may also reduce the overall accuracy of a classifier. Also, adding extra variables to a model increases the overall complexity of the model, add noise to your model and make model interpretation problematic. Therefore, deciding on the optimum features to be included in the model training is critical in obtaining a model with high performance as generically.

Various techniques are used in various fields to eliminate unnecessary features. The techniques for feature selection in machine learning can be broadly classified into the following categories:

- Feature selection based on combining the features for evaluation
- Feature selection based on the supervised learning algorithm used

Feature selection through the amalgamation of features for assessment is categorized into two types: feature subset-centered and feature ranking-centered techniques. Within the feature subset-centered approach, features are amalgamated in potential combinations forming feature subsets, employing any one of several search strategies. These feature subsets are subsequently assessed employing statistical metrics or supervised learning algorithms to gauge the importance of each subset. The most substantial subset is then chosen as the significant feature subset tailored to a specific dataset. If the subset is evaluated using the supervised learning algorithm, then this method is known as wrapper method [15] PSO, GA are heuristic searching strategies. One of the widely accepted fundamental benefits of metaheuristic algorithms is that they provide mechanisms to solve large or intractable problems in reasonable execution times while the exact algorithms fail to succeed due to time limitations [16]. Numerous research works on feature selection have utilized the genetic algorithm to create subsets of features for evaluation, with the supervised machine learning algorithm employed to assess these subsets. For instance, Erguzel et al. utilized the genetic algorithm and artificial neural network to classify electroencephalogram signals [17]. Oreski & Oreski proposed an approach for feature selection that combined GA with neural networks for credit risk assessment [18]. Additionally, Wang et al. applied the GA to generate subsets alongside SVM in the process of feature selection for data classification applications [19]. In their research,

Yang et al. created a feature selection method for land cover classification using PSO [20]. Feature ranking-based methods involve weighting each feature in a dataset based on statistical or information-theoretic measures and then ranking them according to their weights. The noteworthy attributes are picked utilizing a pre-established threshold that dictates the number of features to be selected from the dataset. As these techniques do not necessitate the use of a supervised learning algorithm for appraising feature importance, they adhere to a filter-based methodology. As a result, feature ranking-based methods are more versatile and computationally efficient, regardless of the specific supervised learning algorithm used. Hence, they are a viable choice for selecting important features from datasets with high dimensions. From a taxonomic point of view, these techniques are classified into filter, wrapper, embedded, and hybrid methods.

Hybrid methods are a fusion of filter and wrapper-based approaches. Dealing with high-dimensional data can be challenging when using the wrapper method. To address this, Bermejo et al. devised a hybrid feature selection method called the approach which used filter methods and wrapper methods together. In this method, they initially employ a statistical measure to rank the features based on their relevance. The features with superior rankings are subsequently forwarded to the wrapper technique, resulting in a substantial reduction in the number of necessary assessments, rendering it an efficient linear process. As a result, this hybrid approach reduces the computational complexity when applied to medical data classification tasks. . The hybrid algorithms are developed by combining the current metaheuristics or classical algorithms. The main purpose of hybrid algorithms is to combine the skills of diverse algorithms to obtain better results. Therefore, hybrid metaheuristic algorithms have significant improvements compared to single metaheuristic algorithms [21]The feature selection algorithm developed by Ruiz and colleagues, which employs a statistical ranking method to identify genes for medical diagnosis, was integrated into the wrapper approach. This combination of the filter and wrapper approach was used to distinguish the significant genes causing cancer disease in the diagnosis process [22].

2.5 LITERATURE REVIEW

In the literature, there are many studies conducted with machine learning algorithms for effort estimation in software projects, which differ according to the applied project management methodologies, the indicators that the effort is

represented, the data sets used, performance evaluation metrics, the application platform and the applied methods. Some examples of these studies will be described below.

In 2013 Nassif, Ho and Capretz studied on a comparison between the MLP and log-linear regression models was conducted based on the size of the projects. Results demonstrate that the MLP model can surpass the regression model when small projects are used, but the log-linear regression model gives better results when estimating larger projects [23].

Sharma and Singh concludes that significant amount of research has carried out in software effort estimation using machine learning approaches. The distribution of research over years is stable. The major machine learning approaches used are Artificial Neural Networks, Fuzzy Logics, Genetic Algorithms and Regression Trees for software effort estimation. Most of the studies recommended the use of Line of Code (LOC) and Function Point (FP) software metrics for effort estimations. The review further revealed the lack of real-life datasets which are in accordance to current software development methods and also need of other reliable metrics that can be used for estimation of effort. Diverse validation methods are available which could be considered in augmenting studies to validate the results of software effort estimations. The major validation methods are Cross Validations, Jacknife method and Iterative method [24].

In 2018, Pospieszny, Chrobot and Koylinski conducted a study by applying smart data preparation to dataset of ISBSG, three different ML algorithms as SVM, Neural Networks and Generalized Linear Models in predicting the effort required for software development [25].

In 2018 BaniMustafa suggests performing prediction using three machine learning techniques that were applied to a preprocessed COCOMO NASA benchmark data which covered 93 projects: Naïve Bayes, Logistic Regression and Random Forests. [26].

Asad and Carmine's analysis reveals that artificial neural network (ANN) as ML model, NASA as dataset, and mean magnitude of relative error (MMRE) as accuracy measure are widely used in the selected studies. ANN and support vector machine (SVM) are the two techniques which have outperformed other ML techniques in more studies. Regression techniques are the mostly used among the non-ML techniques, which outperformed other ML techniques in about 19 studies. Moreover,

SVM and regression techniques in combination are characterized by better predictions when compared with other ML and non-ML techniques [27].

In 2020, Singh and Kumar Linear Regression (LR), Multi-layer perceptron (MLP), Random Forest (RF) algorithms are implemented using WEKA toolkit, and results shows that Linear Regression shows better estimation accuracy than Multilayer Perceptron and Random Forest [28].

In 2021, Asad and Gravino the authors performed six bio-inspired feature selection algorithms (GA, PSO, ACO, TS, HS, and FA) and four traditional non-bio-inspired algorithms (Best-First Search, Greedy Stepwise, Subset Forward Selection, and Random Search), used in combination with five widely used estimation techniques (MultiLayerPerceptron, Support Vector Regression, Random Forest, Linear Regression, and M5P algorithm) and applied to eight publicly available datasets widely used in the SDEE community (Albrecht, China, COCOMO, Finnish, Kemerer, Maxwell, Miyazaki, and NASA) [29].

Similarly, Ritu and Gang' s paper suggests different machine learning techniques such as Naïve Bayes, Random Forests Logistic Regression, stochastic gradient boosting, decision tree, and story point for estimation to assess prediction more efficiently [30].

In 2022, Sharma and Chaudhary a comparative study has been done for agile development and traditional development using the neural network (NN) and genetic algorithm (GA). The minimum error and maximum accuracy for estimated values of effort achieved using the machine learning methods. The dataset with the story point give best results followed by projects with lines of code [31].

Also, in 2023, Jadhav and Shandilya applied eight different machine learning based regression algorithms namely; SVM, Random Forest (RF), Decision Tree (DT), Stochastic Gradient Boosting (SGB), Naïve Bayes (NB), MLP, LinearRegression (LR) and kNN over twelve different publicly available datasets; Albrecht, China, COCOMO81, Desharnais, Finnish, Kemerer, Kitchenham, Maxwell, Miyazaki, NASA18, NASA93 and Telecom. On considering RF outperforms other ML algorithms. Considering 36 cases, top three cases of each dataset; RF proves to be more accurate in terms of prediction accuracy. RF gives high prediction accuracy with 9 datasets, followed by kNN which gives higher accuracy with 6 datasets, NB with 5 datasets, DT and LR with 4 datasets, MLP and SGB with 3 datasets and finally SVM with only 2 datasets. [32].

CHAPTER III
METHODOLOGY

In this section, the datasets discussed in problem solving, the tool used, the machine learning algorithms to be applied, and then the feature selection techniques to obtain the optimized dataset subset, the performance measurement metrics used to measure the success of the models applied in the study are explained in detail.

3.1 DATASET

In this study, Finnish, Kemerer, Maxwell and China datasets were examined for software cost estimation from the Promise Data repository [33] The primary objective behind utilizing these datasets is their widespread recognition, simplicity, and accessibility to the public. This facilitates easy replication and verification of results, and potentially encourages further exploration and expansion. It is important to note that the approach is not limited to any specific dataset or model, but can be applied across various datasets and models. Related datasets ‘s information given in Table 3.1.

Table 3.1: Information of Datasets

Dataset	Project Number	Feature Number	Size (Measure Unit)	Cost (Measure Unit)
China	499	19	Function Point	Man-Hour
Finnish	38	9	Function Point	Man-Hour
Kemerer	15	8	KSLOC	Man-Month
Maxwell	62	27	Function Point	Man-Hour

3.1.1 Finnish

Finnish Dataset: The dataset from Finland includes 40 project records gathered by the TIEKE organization from nine companies in Finland. The size and complexity of the projects were assessed using the function point approach introduced by Kitchenham and Kansala in 1993. Detailed statistics of the Finnish dataset are presented in the Table 3.2.

Table 3.2: Finnish Dataset Statistics

No	Feature	Description	Min	Max	Mean
1	ID	Project no	1	38	1905
2	Dev.eff,hrs	Development effort hours	460	26670	767829
3	hw	Hardware type	1	3	126
4	at	Application type	1	5	224
5	FP	Function point data	65	1814	76354
6	co	Application area	2	10	626
7	prod	Project duration (calendar months)	147	2947	1007
8	lnsize	System requirements size in raw Albrecht function points	417	75	636
9	lneff	Effort provided by application user	613	1019	840

3.1.2 China

China Dataset: The China dataset is a more recent addition to the PROMISE repository, included in 2010. It consists of 499 records, as documented by Bosu and MacDonell in 2019. The China dataset comprises 19 features, with 18 being independent variables and 1 being the dependent variable. The Table 3.3 provides statistical information regarding the China dataset.

Table 3.3: China Dataset Statistics

No	Feature	Min	Max	Mean
1	ID	1	499	250
2	AFP	9	17518	487
3	Input	0	9404	167
4	Output	0	2455	114
5	Enquiry	0	952	62
6	File	0	2955	91
7	Interface	0	1572	24
8	Added	0	13580	360
9	Changed	0	5193	85
10	Deleted	0	2657	12
11	PDR_AFP	0.3	83.8	12
12	PDR_UFP	0.3	96.6	12
13	NPDR_AFP	0.4	101	13
14	NPDU_UFP	0.4	108	14
15	Resource	1	4	1
16	Dev.Type	0	0	0
17	Duration	1	84	9
18	N_effort	31	54620	4278
19	Effort	26	54620	3921

3.1.3 Kemerer

Kemerer Dataset: The Kemerer dataset, collected in 1987, originates from an American company engaged in the development of data processing software. It

comprises 15 projects, each having eight attributes. The dataset contains projects that began between 1981 and 1983, with data collected in 1985. Specific statistical details for the Kemerer dataset can be found in the Table 3.4.

Table 3.4: Kemerer Dataset Statistics

No	Feature	Description	Min	Max	Mean
1	ID	Project ID	1	15	8
2	Language	Software used	1	3	12
3	Hardware	Hardware used	1	6	233
4	Duration	Duration	5	31	1427
5	KSLOC	Number of source lines code in thousands	39	450	18657
6	AdjFP	Adjusted function points	999	23068	99914
7	RAWFP	Raw function points	97	2284	99387
8	EffortMM	Effort Man Months	232	110731	21925

3.1.4 Maxwell

Maxwell Dataset: Collected from a Finnish commercial bank, the Maxwell dataset encompasses 62 projects described by 27 attributes. Maxwell documented this dataset in 2002. The projects in the dataset initiated between 1985 and 1993. Comprehensive statistical information for the Maxwell dataset is provided in the Table 3.5.

Table 3.5: Maxwell Dataset Statistics

No	Feature	Description	Min	Max	Mean
1	Syear	Year	85	93	8958
2	App	Application type	1	5	235
3	Har	Hardware platform	1	5	261
4	Dba	Database	0	4	103
5	Ifc	User interface	1	2	193
6	Source	Where developed	1	2	187
7	Telonuse	Telon use	0	1	24
8	Nlan	# of development languages	1	4	255
9	T01	Customer participation	1	5	305
10	T02	Development Env, adequacy	1	5	305
11	T03	Staff availability	2	5	303
12	T04	Standards use	2	5	319
13	T05	Methods use	1	5	305
14	T06	Tools use	1	4	290
15	T07	Software logical complexity	1	5	324
16	T08	Requirements volatility	2	5	381
17	T09	Quality requirements	2	5	406
18	T10	Efficiency requirements	2	5	361
19	T11	Installation requirements	2	5	342
20	T12	Staff analysis skills	2	5	382
21	T13	Staff application	1	5	306
22	T14	Staff tool skills	1	5	326
23	T15	Staff team skills	1	5	334
24	Duration	Duration	4	54	1721
25	Size	Function points	48	3643	67330
26	Time	Time	1	9	558
27	Effort	Work hours Effort	583	63694	822321

3.2 APPLICATION PLATFORM

This study was conducted utilizing the WEKA platform, which is an open-source application written in Java. It was originally developed by a PhD student at the University of Waikato in New Zealand and is governed by the General Public License. WEKA offers a range of algorithms for performing machine learning and data engineering tasks, including classification, clustering, visualization, estimation, correlation analysis, feature selection, and data preprocessing for scientific research. The version utilized in this study was WEKA 3.8.6 which published in 2022.

While WEKA is installed, it presents the weka.jar file, which includes the necessary libraries. WEKA Jar allows for the development of projects by accessing WEKA classes from other platforms such as Java or C#. Within WEKA, datasets are typically in the arff (Attribute Relationship File Format) extension, although it also supports other formats such as textual csv, dat, libsvm, json, and xrff.

The opening screen of WEKA, as depicted in the provided Figure 3.1 The opening screen of WEKA, serves as the interface from which the platform is launched.

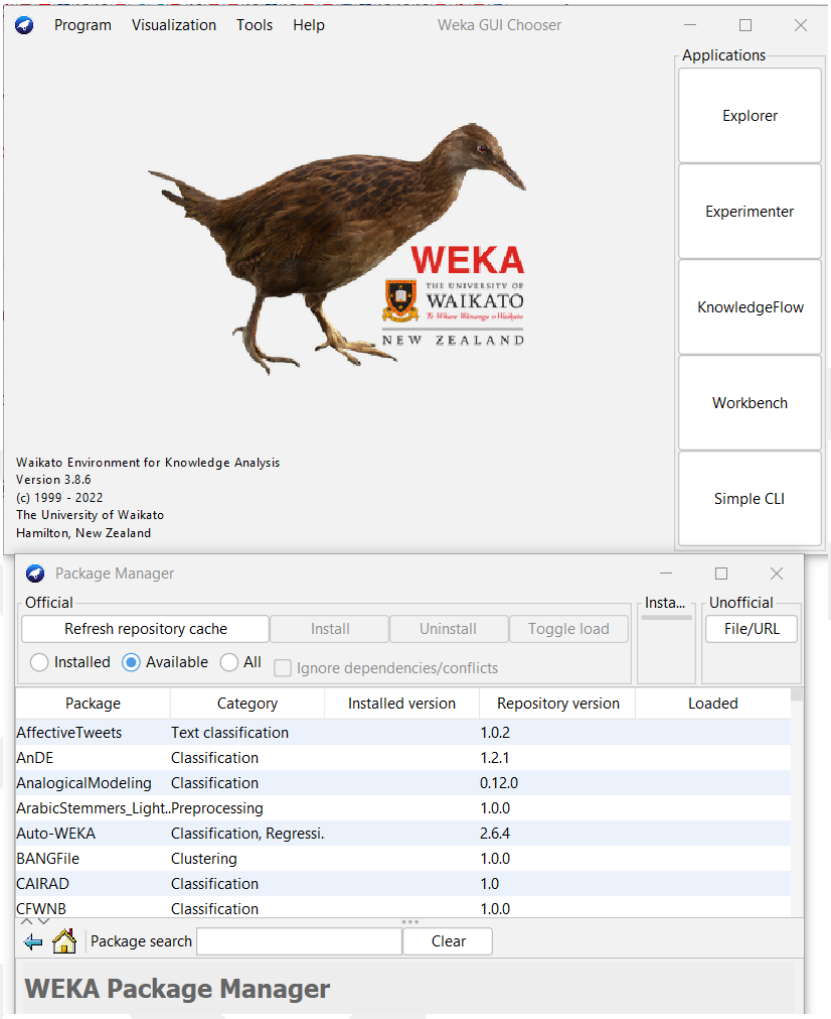


Figure 3.1: The opening screen of WEKA

- Explorer button enables users to perform tasks such as loading datasets, conducting classification, clustering, preprocessing, and feature selection operations.
- By utilizing the Experimenter button, users can identify the most suitable methods and parameter values for classification and regression techniques.
- Knowledge Flow button handles extensive data manipulations, allowing users to create a workflow by dragging and combining boxes representing learning algorithms and data sources.
- Workbench button provides a consolidated interface that incorporates the functionality of other buttons and allows for the inclusion of user-added plug-ins.

- On the other hand, Simple CLI button opens the console screen, enabling users to execute all WEKA operations using text commands.

In addition, some algorithms, attribute selection tools, etc. that are not already installed can be included in the relevant version by selecting Package Manager from the Tool tab in the top menu. In this study, it has been added to the version via PSO and GA tool.

After this stage, it was proceeded to load the dataset and develop the model directly with the explorer tab. Which is given Figure 3.2 Explorer Tab Content of WEKA.

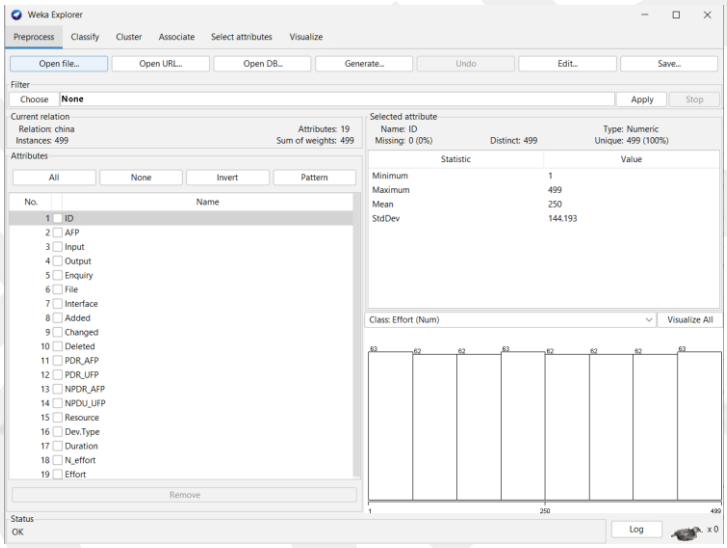


Figure 3.2: Explorer Tab Content of WEKA

Descriptions of some of the key features available in the Explorer menu are as follows:

- Open File/URL/DB: These options allows users to load a dataset from specified location. It supports various formats such as ARFF (Attribute-Relation File Format), CSV (Comma-Separated Values), and more.
- Save: Users can save the modified dataset or the results of their analysis using this option. They can choose the desired file format and specify the destination to save the file.
- Preprocess: The Preprocess option provides a range of data preprocessing techniques. It allows users to perform tasks such as attribute selection, attribute transformation, instance filtering, and missing value handling.

Users can apply these techniques to clean and prepare their data for further analysis.

- The "Choose" option in the WEKA Explorer menu allows users to select the target attribute or class for their machine learning tasks.

Additionally, after selecting the target attribute using the "Choose" option, users can transform unusual datas to valid forms, apply filters which enables attribute selection, classification implementation further configure and customize the machine learning algorithms, evaluation metrics, and visualization options to refine the analysis and could achieve the desired results.

In this section, the previously acquired China.arff data is imported into the system by using the open file option. Attributes of the data become visible in rows in the data window. In addition, 499 samples in the dataset are also depicted as visible. Analysis of the existing data can be performed on the Edit Data tab or on the graphic images created when each attribute is selected. As described, data types can be changed from nominal to numeric in the Choose/Filter tab, etc. preprocessing is allowed.

At this stage, firstly, model training was carried out by going to the Classify tab on the original dataset. Secondly, with the Attribute Selection function in the Preprocess/Choose/Filter tab, it was tried to optimize the number of features by applying hybrid feature selection techniques in different configurations before model training. An example is given in which the number of features is reduced from 19 to 10 by testing the Genetic Algorithm, which is the search method, with Cfs, which is the evaluator in WEKA.

Classify: This option enables users to build and evaluate classification models using various algorithms. Users can select the target class attribute and choose from a wide range of classifiers, including decision trees, support vector machines, naive Bayes, and more as given at with Figure 3.3. The Classify option also provides evaluation metrics and visualization tools to assess the performance of the models.

Firstly, the classify application, which was carried out on the original dataset, then reduced to 10 and the model training was carried out by selecting the SMOreg function containing the Support Vector Machine for Regression feature from the Choose button and selecting 10-fold cross validation from the left Test Options menu.

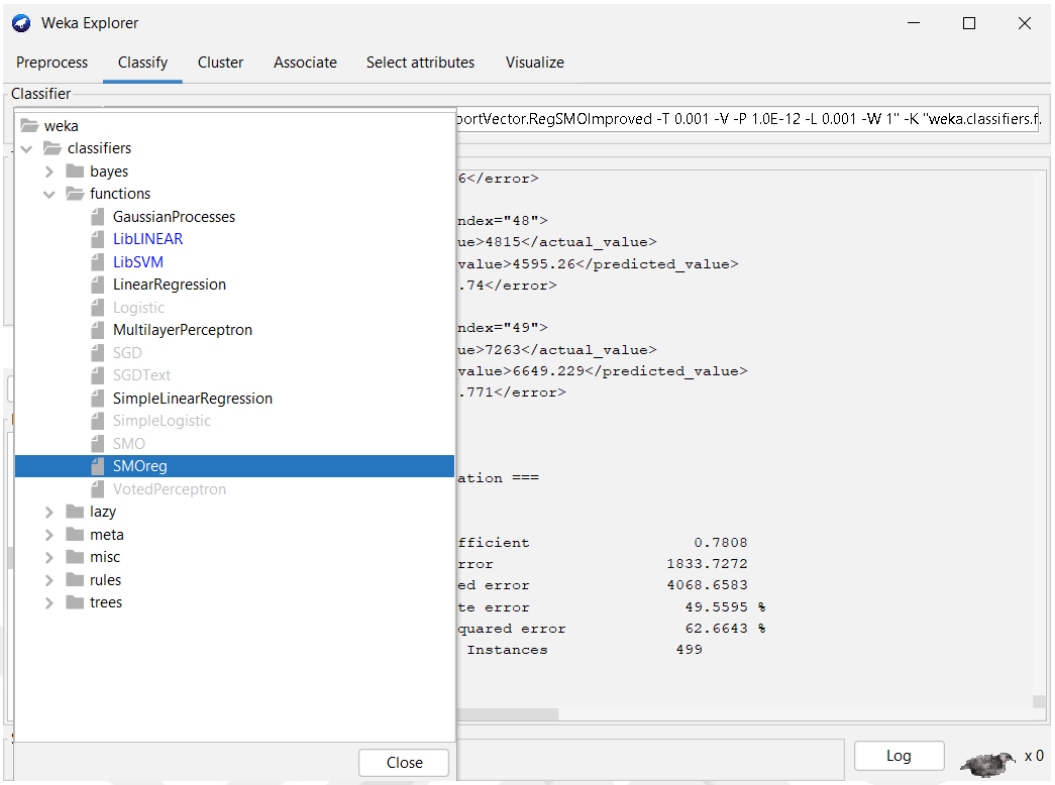


Figure 3.3: Classify Section of WEKA

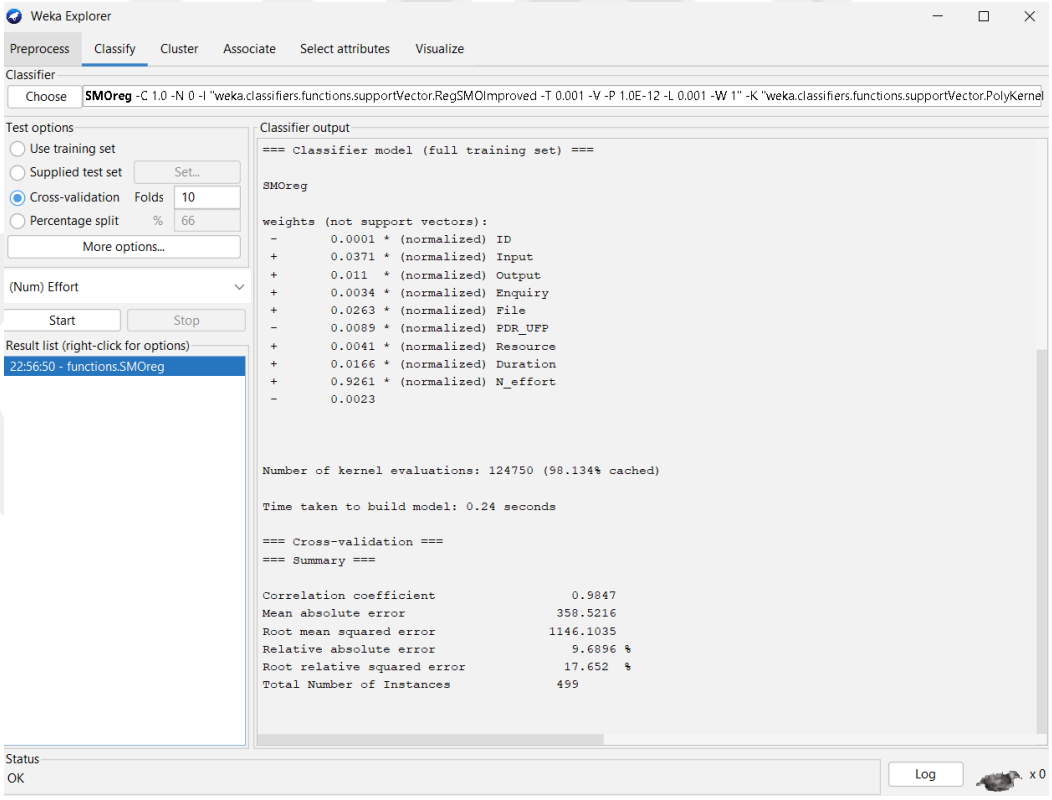


Figure 3.4: Classifier Details in WEKA

As seen in Figure 3.4, the right panel shows the model training time and results. Model performance parameters Correlation Coefficient, Mean Absolute Error, Root Mean Squared Error, Relative Absolute Error, Root Relative Squared Error are presented and Total Number of Instances which give the number of samples used in model training and testing is showed.

Cluster: The Cluster option allows users to perform clustering analysis on their dataset. Users can select from a variety of clustering algorithms, such as k-means, hierarchical clustering, and density-based clustering. They can explore the resulting clusters and analyze the patterns within their data.

Associate: The Associate option is used for association rule mining. It enables users to discover interesting associations and relationships between different attributes in their dataset. Users can set parameters, such as minimum support and confidence thresholds, and generate association rules based on their data.

Visualize: The Visualize option provides visual representations of the data, models, or evaluation results. Users can explore different visualization techniques, such as scatter plots, line charts, decision tree visualizations, and more, to gain insights and understand the patterns and relationships in their data.

Evaluate: This option allows users to evaluate the performance of their models using various evaluation metrics. Users can assess classification accuracy, precision, recall, F-measure, and other measures to gauge the effectiveness of their models.

These are some of the main features typically found in the Explorer menu of WEKA 3.8.6. They provide a comprehensive set of tools to load, preprocess, model, evaluate, and visualize data for machine learning tasks.

3.3 MACHINE LEARNING ALGORITHMS

In this section, the ML algorithms used in the thesis study and included in the classification area of the WEKA tool are presented.

ML algorithms in WEKA are listed under the following headings and the algorithms used in model training in the thesis study are listed under the relevant headings.

a. Functions
    - LinearRegression
    - Multilayer Perceptron
    - SMOreg (Sequential Minimal Optimization Regression)

b. Lazy Classifiers
    - IBk (K-nearest neighbors classifier)
    - KStar (Instance-based classifier)
c. Meta
    - Bagging
d. Tree
    - M5P (M5 Model trees)
    - RandomForest
    - RandomTree

3.3.1 Linear regression

Linear regression is a prediction algorithm that is frequently used in the field of machine learning. This algorithm is used to model the relationship of a dependent variable (target variable) with independent variables. Basically, a linear equation is created by multiplying the features in the dataset by a weight and adding them together.

The main purpose of the linear regression algorithm is to determine the relationship between the independent variables and the effect of these variables on the dependent variable. In this way, we can make predictions for new data points. The results show that linear-regression-based methods show some advantages than deep learning such as being applicable to a small number of training samples and complicated natural object image [34].

As an example, we can use linear regression to estimate the selling price of a house. In this case, the dependent variable is the selling price, and the independent variables are the size of the house, location, number of rooms, etc. it could be. The model can predict the selling price by determining the weights of these independent variables.

Linear regression has some important assumptions, such as the assumptions that the errors are normally distributed, that the errors are independent, and that there is no heteroscedasticity. These assumptions can affect the accuracy of the model and are points that need to be analyzed.

Weaknesses:

- Linear regression assumes that the relationships in the dataset are linear. It may have poor performance in nonlinear relationships.
- Outliers can affect the performance of the model.

- In the case of multiple collinearity (a high correlation between features), the performance of the model may decrease.

Strengths:

- It is a simple and fast algorithm.
- It is easy to interpret. The weights of the model show the effect of the features on the target variable.
- If the data is linearly correlated, it can give good results.

3.3.2 Multilayer Perceptron

Multilayer perceptron (MLP) is a multilayer artificial neural network-based algorithm. While creating the model, a multi-layered sensor network is created. This network includes the input layer, hidden layer(s), and output layer. The study investigated the application of Multilayer Perceptron (MLP) neural networks with back-propagation learning for churn prediction in a telecommunication company[35].

Nodes in each layer are associated with activation functions (usually sigmoid or ReLU). The weights of the connections in the model are initialized with random initial values. Using Forward Propagation, a sample in the dataset is passed in the forward direction of the network, starting from the input layer. By using activation functions in each layer, the outputs at the nodes are calculated and transmitted to the next layer. Also, using Backpropagation, the predictions in the output layer are compared with the target values to calculate the error. Then the error propagates backwards to the layers and the gradient descent method is used to update the weights. This step is aimed at reducing the model's errors and improving its performance. Updating the weights and training the model with repeated forward propagation and back propagation steps on the dataset. At the same time, a separate validation dataset is used so that the model does not overfit. The model trained for Evaluation of the Model is evaluated on the test dataset.

Weaknesses:

The training process can take time and require computational power, especially for large datasets and complex network structures.

- The model has the risk of overfitting, that is, it can overfit the training dataset and reduce the generalization performance.

- Setting up the structure and parameters of the network correctly may require experience and trial and error.

Strengths:

- Multilayer perceptrons are powerful when modeling complex relationships and non-linear problems.
- Since it is based on artificial neural networks, it has the ability to capture non-linear features.
- It performs well overall and is effective in a variety of data analysis and forecasting problems.

MLP is used in many fields, especially image processing, natural language processing, financial analysis and control systems. In WEKA, MLP is an algorithm used especially in datasets with complex structures and non-linear relationships.

3.3.3 SMOreg (Sequential Minimal Optimization Regression)

SMOreg (Sequential Minimal Optimization for Regression) is a machine learning algorithm used for support vector regression (SVR). The SMO algorithm addresses the dual optimization problem of SVMs using a coordinate descent approach. By selecting two variables at a time for optimization, the SMO algorithm efficiently converges to the global solution. Experimental results on various datasets demonstrate the effectiveness of SMO in training SVMs with competitive classification performance. The study provides insights into the underlying principles of SMO and its role in training SVM models [36]. SMOreg aims to best order data points around a regression line or plane.

- SMOreg creates a regression line or plane by projecting data points onto a high-dimensional feature space. This projection is performed using the kernel function. For example, the RBF (Gaussian) kernel is a frequently used option.
- SMOreg solves an optimization problem by splitting the dataset into support vectors. This problem aims to identify support vectors that line the regression line or plane.
- SMOreg selects support vectors to solve the optimization problem. These support vectors are the points that best describe the regression line or plane.

- SMOreg iteratively solves the optimization problem. In each iteration, a pair of support vectors is selected and optimization is performed on this pair. This helps to update the support vectors and the regression line/plane.
- SMOreg generates the regression line or plane using support vectors and kernel function. This model is used to make predictions of the regression problem.

Some features of the SMOreg algorithm are:

- SMOreg is resistant to outliers within the dataset and therefore ensures that the regression model is stable.
- Projecting into a high-dimensional feature space using a kernel function is suitable for solving non-linear regression problems.
- SMOreg performs well on low-size datasets, but there may be scalability issues for large datasets.
- The complexity of the model depends on the choice of kernel function and number of nodes. Therefore, the model may tend to overfit.

Strengths:

- SMOreg is an efficient algorithm for support vector regression (SVR). SVR can provide high performance in regression problems.
- Support vector regression is robust to outliers and can reduce the effect of outliers in the dataset on the model.
- It is capable of solving non-linear regression problems, by using kernel function to project onto high-dimensional feature space.
- SMOreg can perform well on low dimensional datasets and can run quickly in some cases.
- It may be capable of making good generalizations by ordering the support vectors around the regression line or plane in the best way possible.

Weaknesses:

- SMOreg can cause scalability issues for large datasets. It may take time to train the model and make predictions on large datasets.
- It may be necessary to set the parameters of the kernel function and model correctly. This may require experience and trial and error with hyperparameter optimization.

- SMOreg may be prone to overfitting in some cases. Overfitting of the model may reduce the generalization performance of the model.
- SMOreg may not be able to fully solve nonlinear regression problems in some cases. In this case, it may be necessary to use other kernel functions or different models.

SMOreg is an efficient algorithm for solving regression problems and is also available in machine learning tools such as WEKA.

3.3.4 IBk (Instance-Based learning with k parameter)

IBk (Instance-Based learning with k parameter) or k-nearest neighbors (k-NN) is a machine learning algorithm. IBk is an algorithm used to solve classification and regression problems. The study invesgtigated several methods have been studied in text categorization and mostly are inspired by the statistical distribution features in the texts, such as the implementation of Machine Learning (ML) methods. The SMO and IBk methods were the best, while AdaBoost was the worst. [37].

The IBk algorithm chooses a proximity metric to calculate similarity between samples. Euclidean distance is a commonly used measure of sample distance, but other metrics can be used.

Creates a model that represents the dataset in the feature space of the samples. This model includes all samples and their labels.

When a test sample arrives, the IBk algorithm chooses the k number of training samples closest to that sample. Estimates are made using the labels of the nearest neighbors. While choosing the most common class label for classification, the mean or weighted average of the target values of the nearest neighbors can be used for regression.

Weaknesses:

- Memory and computational requirements: The IBk algorithm stores all training samples and their labels in memory. Memory and computing power issues can arise when working with large datasets and a large number of features.
- Scaling issues: Proximity measurement is affected if features in the dataset have different scales. This is why it's important to pre-scale features.

- Sensitivity to outliers: The IBk algorithm makes a prediction based on close neighbors. Outliers may affect the proximity measurement and adversely affect the results.

Strengths:

- Simple and straightforward: The IBk algorithm uses a simple approach and is easy to understand.
- Flexibility: The algorithm can solve classification and regression problems. It can also be customized using different proximity measures.
- Effective results: The IBk algorithm gives successful results in many application areas. In particular, it can perform well where relationships in the dataset are complex.
- Adjusting parameters: k, a parameter of the algorithm, affects the performance of the model. Choosing a good k value can improve the accuracy of the model.

3.3.5 KStar (Instance-based classifier)

The KStar algorithm in WEKA is based on models in the literature. The KStar algorithm is an extension of the C4.5 decision tree algorithm proposed by Quinlan. The C4.5 algorithm is a method used to construct decision trees in classification problems, and the KStar algorithm is built on this basic idea. KStar uses the k-NN (k-nearest neighbor) method when classifying, and therefore KStar can be considered as an extension of the k-NN algorithm in the literature.

Unlike the C4.5 algorithm, the KStar algorithm is customized to work on nominal datasets. Therefore, numeric features may need to be converted to categorical format. Also, data preprocessing steps such as filling in or removing missing data can be applied. The Star algorithm is effectively used in classification problems. By learning the relationship of the features in the dataset with the class labels, they can classify the new samples.

The system takes into account various project attributes, such as lines of code, complexity, and team experience, to estimate effort required for future projects. The performance of the estimation system was evaluated using metrics such as mean

absolute error and coefficient of determination. The K-star algorithm uses similarity measurements to classify the data based on the classes' likelihood [38].

The KStar algorithm creates a model based on the dataset. The model is represented as a decision tree structure. The decision tree contains rules that are used to make class predictions based on the values of the properties. When a test sample arrives, the KStar algorithm makes a class prediction using the decision tree. For example, a class label is determined for the test sample by following the relevant path in the decision tree.

Strengths:

- Understandability: The KStar algorithm makes it easier to understand the results thanks to the decision tree structure it creates. This increases the interpretability of the model.
- Dealing with missing data: Instead of filling in missing data or subtracting it, KStar can make predictions by taking this data into account in the model building process.
- Feature selection: The KStar algorithm can perform feature selection to identify important features. This can help make the model more effective and less complex.

Weaknesses:

- Scaling issues: The KStar algorithm is numerically insensitive when operating on nominal datasets. Therefore, it may be necessary to pre-convert numeric features or use data preprocessing methods.
- Memory requirements: The KStar algorithm uses memory to store samples and relationships. Memory requirements may increase when working with large datasets.
- Calculation time: The KStar algorithm may require computation time when creating models and making predictions. Especially in case of large datasets or complex tree structures, the computation time can be longer.

3.3.6 Bagging

The Bagging algorithm is one of the ensemble learning methods and is used in classification or regression problems. Ensemble methods aim to create a stronger and

more generalizing model by combining multiple learning models. Bagging is short for Bootstrap Aggregating.

The Bagging algorithm creates multiple sub-datasets from the dataset with the bootstrap sampling method. Each sub-dataset is generated by randomly drawing samples from the original dataset. This sampling process increases the diversity in the dataset. Independent learning models are created using the same learning algorithm on each sub-dataset. These models usually represent a single algorithm, such as decision trees or support vector machines. Each sub model predicts a new sample. Bagging provides unification of estimates, often using the majority voting method in classification problems. In regression problems, the average of the estimates is taken.

Strengths:

- Variance Reduction: Bagging reduces variance by combining the estimates of multiple models. This results in more stable and reliable forecasts.
- Generalization Ability: Bagging reduces overfitting by increasing the diversity in the dataset and increases the generalizability of the model.
- Outlier Resistance: Bagging can reduce the effect of outliers as each sub model is trained on different samples.

Weaknesses:

- Computation Cost: Because bagging requires training and estimating multiple models, the computational cost can increase.
- High Diversity: Bagging sometimes fails to provide high diversity due to the random generation of sub-datasets. In this case, the performance of the model may decrease

Bagging is generally used in classification and regression problems. The bagging algorithm is available in machine learning tools such as WEKA and in many machine learning libraries.

3.3.7 M5P (M5 Model trees)

The M5P algorithm in WEKA is based on a regression model known in the literature as M5P (M5's Model Tree). M5P is an expansion of the M5 algorithm developed by Ross Quinlan.

The M5P model is a tree structure used as a regression model. The model consists of a parent tree and its subtrees. The M5P algorithm uses the tree structure to model complex relationships in the dataset. The tree structure draws attention with its ability to capture non-linear relationships and interactions between variables. The M5P algorithm was developed based on the M5P model in the literature, but the M5P application used in WEKA is a unique application and designed specifically for the WEKA library. Therefore, the M5P algorithm in WEKA may show some differences from the M5P model in the literature. However, the basic principles and model structure are the same.

In the main tree created with M5P, the most important bisectable node is selected and similar operations are applied to the subtrees coming out of this node. This tree structure is suitable for capturing complex relationships in the dataset. After the model is created, tree editing can be done as needed. The editing process involves removing or trimming unnecessary branches. This makes the model simpler and clearer. After the model is created and edited, the M5P model is used to predict a new sample. The model estimates the target value using the input properties, for example.

The M5P algorithm in WEKA is used in regression problems and is particularly effective when there are complex relationships in the dataset. Also, here are some strengths and weaknesses of the M5P algorithm:

Strengths:

- Modeling Complex Relationships: M5P is capable of modeling complex relationships in the dataset through tree structure and node splits.
- Understandability: The tree structure created ensures that the model is understandable and interpretable.

Weaknesses:

- Overfitting Tendency: M5P may tend to overfit the dataset. Therefore, over-learning of the model should be controlled considering the dataset size and complexity.
- Sensitivity: M5P can be sensitive to noises and outliers in the dataset. Therefore, cleaning and preprocessing of the dataset is important.

3.3.8 RandomForest

Random Forest is one of the ensemble learning methods and is used for classification, regression and feature importance ranking problems in machine learning algorithms. Random Forest is a model created by combining multiple decision trees.

Random Forest does random sampling for each decision tree with Bootstrap Sampling. This sampling process is performed by randomly selecting samples from the dataset. Although each sample is the same size, some data may be selected more than once, while others may not.

A decision tree is created on each sample. While constructing the decision tree, a subset of the dataset is used and branch splitting is performed on this subset. The splitting operation selects the combination of features and thresholds that will provide the best separation. This step is repeated for each decision tree.

Classification or regression estimation of a new sample is made using all the decision trees created. In the case of classification, the majority class is determined by voting. In the case of regression, the estimates obtained from the decision trees are averaged.

Strengths:

- High Performance: Random Forest generally provides high performance because it is built by combining multiple decision trees. May be more resistant to overfitting.
- Feature Significance Rating: Random Forest provides a severity rating to evaluate the importance of each feature in classification or regression.
- Resistance to Outlier and Missing Data: Random Forest can better deal with missing or abnormal values in the dataset.

Weaknesses:

- Model Explainability: The Random Forest model may be less explainable than a single tree because it consists of combining multiple decision trees.
- Computation Cost: Because Random Forest requires training and predictions of multiple decision trees, the computation cost may be higher.

It is available in Random Forest, WEKA and many other machine learning libraries and has a wide range of applications.

3.3.9 Random Tree

The RandomTree algorithm is often used to solve classification and regression problems. RandomTree algorithm is a classification and regression method based on decision trees. RandomTree creates decision trees using random feature selection and division operations. Each decision tree provides diversity through random feature selection and division operations and is associated with Random Forest, one of the ensemble learning methods.

Random feature selection randomly selects the features used in each node as a subset. The splitting operation selects the combination of features and thresholds that will provide the best separation. This step is repeated until the decision tree is complete.

By using the created decision tree, classification or regression estimation of a new sample is made. In the case of classification, the majority class estimate is taken at the leaf nodes. In the case of regression, the target values at the leaf nodes are averaged.

Strengths:

- Simplicity and Speed: Because RandomTree is a decision tree-based algorithm, it can work quickly and present a simple model.
- Good Generalization: RandomTree can be resistant to overfitting and show good generalization performance.

Weaknesses:

- Feature Severity Rating: RandomTree can sometimes fail to provide feature severity ratings.
- Less Flexibility: RandomTree may provide less flexibility than some other algorithms.

Random Tree offers classification problems such as medical diagnosis, spam filtering, customer segmentation, etc., and regression problems such as home price forecasting, income forecasting, energy consumption forecasting, etc. can be used in the fields.

3.4 FEATURE SELECTION TECHNIQUES

Attribute selection in WEKA is performed by the Attribute Evaluator and Search method working together. Attribute Evaluator evaluates the importance of the attributes and tries to find the best set of attributes, guided by the Search method. This approach is used to evaluate the quality of features and to eliminate unimportant features, so that a smaller and more meaningful set of features can be obtained. This can provide the model with a better generalization ability and a faster training time. Feature selection can reduce the dimensionality to enable many data mining algorithms to work effectively on data with large dimensionality [39].

Selecting Attribute Evaluator: The first step is to select the Attribute Evaluator method. The Attribute Evaluator measures the effect of each attribute on classification or regression. Weka has various Attribute Evaluator methods, such as Information Gain, Gain Ratio, ReliefF, Chi-Square, etc. Choosing one of these methods determines the evaluator who will rate the importance of the features.

Search Method Selection: The second step is the selection of the Search method to be used in the feature selection. Search methods try to find the best set of attributes based on the importance rating generated by the Attribute Evaluator. Various Search methods are available in Weka, for example GreedyStepwise, BestFirst, GeneticSearch, etc. Choosing one of these methods determines a search strategy to find the best feature set.

Attribute Selection: Attribute selection is performed using the selected Attribute Evaluator and Search method. In this step, the necessary parameters for feature selection are set and the selection process is started. Evaluation and selection of features are performed on a specific criterion or threshold value. As a result, the best feature set is determined.

In this section, the Attribute Evaluators and Search Methods used in the thesis study and included in the SelectAttributes area of the WEKA tool are presented.

Attribute Evaluators

- CfsSubsetEval
- ClassifierAttEval
- Corr. Att.Evaluation
- Relief Att.Evaluation

Search Methods

- Random Search
- Particle Swarm Optimization (PSO)
- Genetic Algorithm (GA)
- Ranker

In Title 4.2 results from hybrid techniques obtained using the given Evaluators and Search Methods in different configurations will be compared and analyzed.

3.4.1 Attribute Evaluators for Feature Selection

3.4.1.1 Correlation Based Feature Selection (CFS)

In Weka, Correlation Based Feature Selection (CFS) named as CFSubsetEval and it is a feature selection algorithm. The CFSubsetEval algorithm is derived from a filter known in the literature. CFSubsetEval is a filter evaluation method for feature selection. This algorithm is based on the k-NN (k-nearest neighbor) classifier and uses classification performance to evaluate feature subsets.

The sequence of the CFSubsetEval algorithm consists of the following steps:

- The feature subset size is determined.
- All possible subsets of features are generated.
- A sub model is created for each feature subset and its performance is evaluated.
- Subsets of features are ranked by performance criterion (for example, classification accuracy or mean square error).
- The feature subset with the best performance is selected and reported as a result.

That is, the algorithm considers all feature subsets, builds a model for each subset, and evaluates its performance. It then sorts by performance criteria and selects the subset of features with the best performance.

3.4.1.2 ClassifierAttEval

Classifier Attribute Evaluation is used to measure the contribution of attributes to classification performance and to identify important attributes. In this way, it is possible to identify unnecessary or low-impact features and to make the classification model more effective.

Classifier Attribute Evaluation performs the following operations in the process of evaluating attributes and calculating their effects on classification:

The effect of the features in the dataset on the classification is evaluated. Classifier Attribute Evaluation determines the importance of attributes using various statistical calculations or algorithms.

3.4.1.3 Corr. Att.Evaluation

Correlation Attribute Evaluation (Corr. Att. Evaluation) is an attribute selection method used in Weka. This method tries to select the most important features by evaluating the relationship between the features. corr. Att. The Evaluation method makes selection by evaluating the relationship of the attributes to the dataset. Attributes with a higher relevance are considered more important and selected. This class evaluates by calculating the correlation between attributes. The correlation value between the features is calculated using the Pearson correlation coefficient. The Pearson correlation coefficient measures the linear relationship between two continuous variables. The correlation coefficient between attributes shows the relationship of one attribute with other attributes. Builds the importance ranking of each attribute using correlation values. A higher correlation value indicates that the features have a stronger relationship and are considered more important.

It requires a search method to select the most important features above a certain threshold or a certain number of features based on the feature evaluation with the determined importance levels.

3.4.1.4 Relief Att.Evaluation

In this method, the RELIEF algorithm is used. The RELIEF (ReliefF) algorithm is a machine learning algorithm for determining the order of importance among attributes. This algorithm measures the relationship of attributes to class labels and tries to identify the most informative attributes.

The RELIEF algorithm considers each sample in the dataset and randomly selects a neighboring sample. If the selected instance is an instance of the same class, it simulates the attributes of this instance to those of the current instance. If it is an instance that does not belong to the same class, it makes the attributes of that instance oppositely similar to those of the current instance. This process is repeated to calculate the importance score given to the sample.

The RELIEF algorithm calculates the importance score for each attribute and evaluates the attributes' relationship to the class label. A higher importance score indicates that the attribute has a stronger association with the class label and is considered more important.

The Relief Attribute Evaluation method needs a search method to sort the attributes based on these importance scores. Thus, an effective selection method is obtained to evaluate the relationship of the attributes with the class label and to identify the most effective attributes in the dataset.

3.4.2 Search Methods for Feature Selection

3.4.2.1 Random Search

RandomSearch, on the other hand, is a search method used to optimize the operation of the feature selection algorithm.

RandomSearch works by generating random feature subsets for a given feature subset size. Initially, a random feature subset is chosen for a given feature subset size. This random subset of features is then evaluated against performance criteria.

Evaluation is performed based on a specific performance measure (for example, classification accuracy or mean square error). If the performance of the current feature subset exceeds the current best performance, the current feature subset is considered the best. Otherwise, a new random subset of features is selected and the evaluation is done again.

This process is repeated over a certain number of iterations (steps). Each step seeks to achieve the best available performance. RandomSearch helps optimize the feature selection process by randomly selecting a subset of features and trying to find the best feature subset with a performance-based evaluation.

3.4.2.2 Particle Swarm Optimization (PSO)

PSO is a naturally inspired optimization algorithm, where a set of particles tries to find the best solution by moving in a given search space. An optimization algorithm, PSO (Particle Swarm Optimization), becomes a feature selection method when a filters (CFS etc.) used for feature selection in the Weka library are used with various optimization algorithms to evaluate the feature subset.

PSO aims to move particles in the feature subset search space and find the best feature subset. Steps:

- The feature subset search space is defined and a random set of particles is initially generated.
- Each particle represents a subset of features. The velocity and position of the particles are then updated, and the performance of each particle is evaluated using the fitness function.
- The position of the best performing particles is recorded.
- These steps are repeated until a certain number of iterations or stopping criteria. The PSO algorithm handles the movements of particles by tracking the best position and doing a balance of search and exploration to collectively find the best solution.
- As a result, PSO aims to find the best feature subset for feature selection by working with feature evaluation methods.

Due to the nature of PSO, it can navigate a large search space effectively and find the subset of features with better classification performance.

3.4.2.3 Genetic Algorithm (GA)

Genetic Algorithm (GA) is an optimization algorithm inspired by natural selection and genetic processes. GA is a population-based approach in which one generation is passed on to another using genetic operators (selection, crossover, mutation).

GA (Genetic Algorithm), which is an optimization algorithm, becomes a feature selection method that aims to find the best feature subset for feature selection when used with feature evaluation function.

When working with GA to find the best feature subset:

- Initially, a population is created. Each individual represents a subset of traits.
- Then, the fitness values of the population are calculated and the performance of each individual is evaluated.
- Then, individuals with better performance are selected based on their fitness values using the selection operator, and new individuals are produced by applying the crossover and mutation operators.
- These steps are repeated until a certain number of iterations or stopping criteria.

- The GA algorithm allows the population to evolve to produce better-fit individuals.
- The selection operator transfers individuals with better performance to the next generation, the crossover operator creates new solutions by combining the genetic material, and the mutation operator provides variation by making random gene changes.

As a result, due to the nature of GA, it evolves the population using genetic operators and can find the subset of traits with better classification performance.

3.4.2.4 Ranker

The Ranker algorithm needs an evaluation function to evaluate the features. This function measures the importance of attributes. In Weka, the ClassifierAttributeEval function etc. is often used as an evaluation function. The Ranker ranks the features while selecting the features and selects the most important ones. The Ranker algorithm has a parameter that is used to determine the number of attributes to be selected. These parameters affect the feature selection process. For example, parameters such as the number of attributes to be selected or the cut-off point of the criterion should be set. That is, the Ranker algorithm is capable of specifying a certain number of features for feature selection. In this way, a choice can be made between assigning fewer attributes to the model or using all attributes. This parameter is called "-N" and can take a value greater than zero.

If the "-N" parameter is set to a certain number, The Ranker algorithm uses the entered parameter value or threshold value to select the most important ones from among the best ranked attributes. The selected attributes will be the ones at the top of the ranking.

If the "-N" parameter is not set or set to zero, the Ranker algorithm uses all attributes and returns all attributes to the model. In this case, the contribution of the features to the model varies depending on their order.

It can be observed that the selected features are important and can provide good performance on the dataset. In addition, the outputs of the evaluation function can be examined to understand the importance of the selected features and which features are less important.

As mentioned in Title 3.2, in order for the Ranker to select the N attributes that have priority in this ranking or which are above a certain threshold value, the value is

entered by double-clicking on the Ranker selection in WEKA. In the study, "Number of features - 3" was entered as N so that the lowest 3 features were not given as input to the model.

3.5 PERFORMANCE MEASURES

3.5.1 Correlation Coefficient

The Correlation Coefficient is a statistical value that measures the strength and direction of the relationship between two variables. It is often called the Pearson Correlation Coefficient and takes values between -1 and +1.

The formula for the Pearson Correlation Coefficient is expressed as:

$$r = (\Sigma((x_i - \bar{x}) * (y_i - \bar{y}))) / \sqrt{((\Sigma(x_i - \bar{x})^2) * (\Sigma(y_i - \bar{y})^2))} \quad (3.1)$$

Formula:

- r represents the Correlation Coefficient.
- xi and yi represent the values of the data points.
- $\bar{x}$ and $\bar{y}$ represent the mean values of xi and yi.

The correlation coefficient is used to evaluate the relationship between the variables in the dataset. If the relationship between two variables is positive, the coefficient approaches +1, and if it is negative, it approaches -1. A value of 0 indicates that there is no relationship between the variables.

The correlation coefficient evaluates whether the relationship between the variables is linear. If the relationship is linear, the coefficient can be used as a good estimator. However, the correlation coefficient does not indicate that the relationship is causal or the effect of other variables. Therefore, care should be taken for the interpretation of the correlation coefficient and should be used in conjunction with other analysis methods.

3.5.2 Mean Absolute Error (MAE)

Mean Absolute Error (MAE) is a method of evaluating the accuracy of a prediction model by calculating the mean of the absolute differences between the measured and predicted values. MAE measures how close a model's predictions are to the true values and represents the mean errors of the predictions.

The formula for MAE is expressed as follows:

$$MAE = (1/n) * \Sigma|yi - xi| \quad (3.2)$$

Formula:

- MAE stands for Mean Absolute Error.
- n stands for the total number of data points.
- yi represents the true value.
- xi represents the predicted value.

The MAE converts the differences between the measured and predicted values into their absolute values, so that negative and positive errors have the same significance. Then it adds up these differences and divides by the number of data points to get the mean error value.

The closer the value of the MAE is to zero, the closer the model's predictions are to the true values and perform better. However, one downside of MAE is that major errors are of the same importance as minor ones. That is, outliers can have a large impact and mislead the overall performance of the model. Therefore, when MAE is used alone, the distribution of estimation errors and other performance measures must be considered.

3.5.3 Relative Absolute Error (RAE)

Relative Absolute Error (RAE) calculates the accuracy of a predictive model. RAE can be used in machine learning. Furthermore, RAE is expressed as the ratio; it computes the mean error (residual) of errors produced by a trivial or naive model. The model is considered non-trivial if the result is less than 1. This is the model for a dataset (k):

$$R_k = \frac{\sum_{i=1}^{n} \left| E_{ki} - D_i \right|}{\sum_{i=1}^{n} \left| D_i - \bar{D} \right|} \quad (3.3)$$

where Ei's is prediction, Di's is actual values, and Rae is the measure of forecast accuracy. D is the mean of Di's; n is the size of the dataset (in data points)

3.5.4 Mean Relative Error (MRE)

Mean Relative Error (MRE) is a performance metric that measures the mean errors of a prediction model's predictions relative to actual values. The MRE calculates the ratio of the prediction errors to the true values for each data point and averages them. In this way, estimation errors are evaluated proportionally to the actual values. The lower the value, the more accurate the predictions are considered. However, one disadvantage of MRE is that true values of zero or very small greatly affect the MRE. Therefore, a model's performance should be more thoroughly evaluated using other performance metrics and evaluation metrics besides the MRE. The formula for MRE is expressed as follows:

$$MRE = \Sigma(|Ei - Ai| / Ai) / k \quad (3.4)$$

Formula:

- MRE stands for Mean Relative Error.
- k represents the total number of data points.
- Ei stands for the estimated value.
- Ai express true value

3.5.5 Mean Magnitude of Relative Error (MMRE)

It is a metric used to measure the accuracy of a predictive model, particularly in the context of regression analysis.

$$MMRE = \frac{1}{m} \times \sum_{i=1}^{m} MRE \quad (3.5)$$

3.5.6 Percentage of Estimations (PRED (0.25))

This measurement is called as PRED (0.25) with the definition of percentage of estimations which fall within 25 percent of the original values.

$$PRED = \frac{1}{m} \times \sum_{i=1}^{m} (MRE \leq 0.25) \quad (3.6)$$

CHAPTER IV
FINDINGS

At this stage, considering the Finnish, Kemerer, China, Maxwell datasets presented in 3.1 DATASET, using the algorithms given in Title 3.3 and choosing the 10-fold cross-validation technique as the validation technique:

- In the first part, with the original datasets,
- In the second part, by using the hybrid configurations of given below evaluation and search methods among feature selection methods given in Title 3.4, wit same datasets made up of optimized and formed most effective features subsets.

1. CFS+ RandomSearch
2. CFS+ PSO
3. CFS+ GA
4. ClassifierAttEval + Ranker
5. Corr. Att.Evaluation + Ranker
6. Relief Att.Evaluation + Ranker

The performance of the models created was evaluated according to the Correlation Coefficient and the error rate according to MAE, RMSE, RAE and RRSE which mentioned in Title 3.5.

Thus, each algorithm discussed in the Title 3.3 was first tested with the original data and then the most effective subsets of the features were created with the six different hybrid methods discussed eight times on different features and the results were evaluated. In the first phase, the results obtained with the original dataset will be examined, and in the second phase, the findings obtained as a result of the feature selection applied dataset will be presented. Finally, by examining the performance criteria reached with different subsets obtained without feature selection and with different feature selection methods,

- The highest performances that can be achieved with the original datasets,
- Dataset-specific and holistic analysis of algorithms that tend to show the highest performance in the model formed with the original data,
- Highest achievements after attribute selection
- Analysis of which feature selection is superior compared to the others, specific to the dataset and holistically,

Ultimately, the goal is to obtain a generic approach that is not reliant on the specific dataset by considering the overall evaluation of these results.

The computer hardware information used while performing the studies is as follows

Processor: Intel(R) Core(TM) i5-10210U CPU @ 1.60GHz 2.11GHz
Installed RAM: 16.0 GB (usable: 15.2 GB)
System type: 64-bit operating system, x64-based processor

4.1 MACHINE LEARNING MODELS EXPERIMENTS AND RESULTS

In this section of the study, Finnish, Kemerer, China and Maxwell datasets are used with their original feature sets, with the choice of 10-fold cross validation technique, with WEKA's LinearRegression, RandomForest, Bagging, MultilayerPerceptron, SMOreg, IBk, KStar, Random Tree, M5p algorithms which given in Title 3.3 one by one. Models were created and the results were evaluated according to the correlation coefficient, MAE and RAE. In WEKA version 3.8.6, the default parameter values of the algorithms are utilized. The specific parameter values employed in this study for the chosen algorithms are outlined below:

- In Linear Regression, the attributeSelectionMethod parameter is designated as the M5 Method.
- In the MLP algorithm, the "hiddenLayers" parameter is set as "a." This implies that the quantity of hidden layers and neurons is automatically determined based on the data. The LearningRate is 0.3, and the momentum is 0.2.
- The SMOReg complexity parameter c1 is opted for. The FilterType is Normalize training data, Kernel is PolyKernel, and regOptimizer is RegSMOImproved.
- In KNN (k-nearest neighbors), k is set to 1, and distanceWeighting is disabled in IBk.

- KStar is configured with a globalBlend of 20, and missingMode utilizes Average entropy curves.
- Bagging employs the REPTree classifier. The numExecutions for setting up the ensemble model is 1, and the number of iterations is 10.
- For M5P, the minimum instance count required for acceptance in a leaf node is set to 4.
- In RandomForest, the maxDepth is set to 0, indicating no depth limit. numIterations is 100, and numExecutions is 1 for ensemble model setup.
- RandomTree utilizes a minNum of 1 for the total instance weight in a leaf and sets maxDepth to 0 for no depth restriction.

The performance evaluation criteria obtained by applying machine learning algorithms to the original feature sets of Finnish, China, Maxwell and Kemerer datasets are given in Table 4.1, Table 4.2, Table 4.3, Table 4.4.

Table 4.1: Performance Measures of Models Constructed with Finnish Original Feature

Machine Learning Algorithms	**Finnish Dataset (Original Feature Set)**		
	Correlation Coefficient	**MAE**	**RAE (%)**
LinearRegression	0.9607	0.254	24.8268
RandomForest	0.9818	0.1649	16.1147
Bagging	0.9801	0.1747	17.074
MultilayerPerceptron	0.9575	0.2297	22.4464
SMOreg	0.962	0.2341	22.8759
IBk	0.7697	0.539	52.6711
KStar	**0.9889**	**0.1344**	**13.1344**
Random Tree	0.9029	0.3654	35.7136
M5p	0.9692	0.2146	20.9732

Table 4.2: Performance Measures of Models Constructed with China Original Feature

Machine Learning Algorithms	**China Dataset (Original Feature Set)**		
	Correlation Coefficient	**MAE**	**RAE (%)**
LinearRegression	0.9889	362.939	9.809
RandomForest	0.9591	557.7718	15.0747
Bagging	0.9605	511.9898	13.8374
MultilayerPerceptron	0.9733	461.3901	12.4698
SMOreg	**0.9897**	**270.4561**	**7.3095**
IBk	0.8918	1571.1824	42.4638
KStar	0.9646	628.608	16.9892
Random Tree	0.9283	943.0361	25.4871
M5p	0.9842	392.7912	10.6158

Table 4.3: Performance Measures of Models Constructed with Maxwell Original Feature

Machine Learning Algorithms	Maxwell Dataset (Original Feature Set)		
	Correlation Coefficient	MAE	RAE (%)
LinearRegression	0.8085	4157.5897	66.1746
RandomForest	0.7612	3998.2174	63.638
Bagging	0.7711	3949.1671	62.8573
MultilayerPerceptron	0.7641	4764.3788	75.8327
SMOreg	**0.8191**	**3812.9653**	**60.6894**
IBk	0.463	5517.129	87.8139
KStar	0.7336	4618.2302	73.5065
Random Tree	0.569	5686.9672	90.5171
M5p	0.8175	3718.2692	59.1822

Table 4.4: Performance Measures of Models Constructed with Kemerer Original Feature

Machine Learning Algorithms	Kemerer Dataset (Original Feature Set)		
	Correlation Coefficient	MAE	RAE (%)
LinearRegression	0.3692	173.2407	107.6474
RandomForest	0.3532	129.0567	80.1926
Bagging	0.1277	185.4463	115.2317
MultilayerPerceptron	0.3511	129.4589	80.4425
SMOreg	**0.5737**	**114.3301**	**71.0419**
IBk	0.4665	142.054	88.2688
KStar	0.5589	134.6747	83.6835
Random Tree	-0.0271	250.9131	155.9111
M5p	0.3291	176.3236	109.5631

When examining the obtained results in terms of performance metrics:

- While KStar algorithm found the best result with 0.9889 in the Finnish original set, RandomForest and Bagging gave 0.9818, 0.9801 results, respectively, and were above 0.98, and M5P, SMOreg, LinearRegression, MultilayerPerceptron and RandomTree algorithms, respectively, showed high success over 0.90, It was seen that the worst result was achieved with the IBk algorithm, while the lowest result was obtained with the IBk algorithm with 0.7697,
- While SMOreg algorithm found the best result with 0.9897 in the original set of China, LinearRegression and M5P gave 0.9889, 0.9842 results, respectively, watching over 090, the worst result was IBk algorithm, and the lowest result was IBk algorithm with 0.8918. It has been found that with
- While Maxwell found the best result with 0.8191 in the original set, SMOreg algorithm found the best result, LinearRegression and M5P gave 0.8175 and

0.8085 results, respectively, and were above 080, the worst result was IBk algorithm, the lowest result was IBk algorithm with 0.8918. It has been found that,

- In Kemerer 's original set, the SMOreg algorithm found the best result with 05737, while the KStar algorithm gave 05589 results and remained above 050, while the worst result was obtained with the RandomTree algorithm, with a high performance of -0.0271.

has been obtained.

Examining the first results obtained with the original datasets, it was observed that the SMOreg algorithm was successful in reaching the best result, while the LinearRegression, M5P and KStar algorithms converged to high performance, while the IBk algorithm showed low performance.

4.2 BY USING FEATUE SELECTION MACHINE LEARNING MODELS EXPERIMENTS AND RESULTS

At Title 4.1, LinearRegression, RandomForest, Bagging, Multilayer Perceptron, SMOreg, IBk, KStar, Random Tree, M5p algorithms were applied by using the original datasets of Finnish, Kemerer, China, Maxwell. On the other hand, in this section; among the feature selection methods given in Title 3.4, by using hybrid configurations of the evaluation and search methods given below were applied to the datasets one by one to find out the most effective features. Each algorithm was run and model established separately by taking as input the feature subset obtained by applying each hybrid feature selection method, all results were recorded and the method that gave the most optimized feature subset was analyzed. In addition, the improvement results were also recorded by looking at which feature selection method showed an increase in performance compared to the original datasets. Also, the techniques which gained to best performance by selecting the feature were recorded.

1. CfsSubsetEval + RandomSearch
2. CfsSubsetEval + PSO
3. CfsSubsetEval + GA
4. ClassifierAttEval + Ranker
5. Corr. Att.Evaluation + Ranker
6. Relief Att.Evaluation + Ranker

4.2.1 Finnish Dataset Cost Estimation Results with Feature Selection Methods

4.2.1.1 CfsSubset Evaluation and Random Search Method Results

For the Finnish dataset, attribute selection was applied by selecting the evaluation criterion CfsSubsetEval and the search criterion RandomSearch from the SelectionAttributes function of WEKA. The number of attributes, which was 9 before the process, was reduced to 5 by applying the method, by selecting the attributes named Development effort hours, Hardware type, Function point data, Project duration (calendar months), System requirements size in raw Albrecht function points. With the addition of the dependent variable, Effort provided by application Use, 6 attributes selected for the most effective subset of attributes were determined as the model input. The results of the algorithms obtained without feature selection and the performance outputs of the model created with feature subset after feature selection is given in Table 4.5.

Table 4.5: Performance Measures of Models with Finnish Original Feature and Selected Feature Set by RandomSearch

Machine Learning Algorithms	**Finnish Original Dataset (9 Feature)**			**Feature Selection with CFS+ RandomSearch (6 Feature)**		
	Correlation Coefficient	**MAE**	**RAE (%)**	**Correlation Coefficient**	**MAE**	**RAE (%)**
LinearRegression	0.9607	0.254	24.8268	0.9607	0.254	24.8268
RandomForest	0.9818	0.1649	16.1147	0.989	0.1344	13.1385
Bagging	0.9801	0.1747	17.074	0.9854	0.1548	15.1257
MultilayerPerceptron	0.9575	0.2297	22.4464	0.9724	0.1595	15.5839
SMOreg	0.962	0.2341	22.8759	0.9654	0.2311	22.5861
IBk	0.7697	0.539	52.6711	0.931	0.3319	32.4397
KStar	**0.9889**	**0.1344**	**13.1344**	**0.9916**	**0.1208**	**11.8098**
Random Tree	0.9029	0.3654	35.7136	0.9563	0.2403	23.4836
M5p	0.9692	0.2146	20.9732	0.9715	0.2158	21.0941

The best performance was obtained by KStar before and after the feature selection was applied. The performance measures, which were 0.9889 Correlation Coefficient, 0.1344 MAE, 13.1344 RAE (%) before the feature selection was applied, improved to 0.9916 Correlation Coefficient, 0.1208 MAE, 11.8098 RAE (%) after the feature selection was applied.

Similarly, the lowest performance was obtained with IBk before and after feature selection. The performance measures, which were 0.7697 Correlation Coefficient, 0.539 MAE, 52.6711 RAE (%) before the feature selection was applied, improved to 0.931 Correlation Coefficient, 0.3319 MAE, 32.4397 RAE (%) after the feature selection was applied, and a great improvement was achieved.

In addition, after applying this method, 8 out of 9 algorithms showed the improvement and the highest improvement of this method was recorded as 17% improvement in the correlation coefficient in the first and last results of the IBk algorithm.

4.2.1.2 CfsSubset Evaluation and Particle Swarm Optimization (PSO) Method Results

Attribute selection was applied by selecting the evaluation criterion CfsSubsetEval and the search criterion Particle Swarm Optimization (PSO) from the SelectionAttributes function of WEKA. The number of attributes, which was 9 before the process, was reduced to 4 by applying the method, by selecting the attributes named Development effort hours, Function point data, Project duration (calendar months), System requirements size in raw Albrecht function points. With the addition of the dependent variable, Effort provided by application Use, 5 attributes selected for the most effective subset of attributes were determined as the model input.

The results of the algorithms obtained without feature selection and the performance outputs of the model created with feature subset after feature selection is given in Table 4.6.

Table 4.6: Performance Measures of Models with Finnish Original Feature and Selected Feature Set by PSO

Machine Learning Algorithms	**Finnish Original Dataset (9 Feature)**			**Feature Selection with CFS+ PSO (5 Feature)**		
	Correlation Coefficient	**MAE**	**RAE (%)**	**Correlation Coefficient**	**MAE**	**RAE (%)**
LinearRegression	0.9607	0.254	24.8268	0.9607	0.254	24.8268
RandomForest	**0.9818**	**0.1649**	**16.1147**	**0.9942**	**0.0976**	**9.5354**
Bagging	0.9801	0.1747	17.074	0.9857	0.1532	14.967
MultilayerPerceptron	0.9575	0.2297	22.4464	0.9853	0.1376	13.4468
SMOreg	0.962	0.2341	22.8759	0.9702	0.2213	21.6292
IBk	0.7697	0.539	52.6711	0.934	0.3142	30.7074
KStar	0.9889	0.1344	13.1344	0.9923	0.1127	11.0156
Random Tree	0.9029	0.3654	35.7136	0.9843	0.1461	14.2734
M5p	0.9692	0.2146	20.9732	0.9715	0.2158	21.0941

The best performance was obtained by KStar before feature selection, and performance measurements were recorded as 0.9889 Correlation Coefficient, 0.1344 MAE, 13.1344 RAE (%). By applying feature selection in Finnish dataset, the improvement was gained in RandomForest algorithm and the best performance was

revised as RandomForest. Performance measures were observed as 0.9942 Correlation Coefficient, 0.0976 MAE, 9.5354 RAE (%). It has been observed that higher results have been achieved. Based on this improvement, it has succeeded in being the method that can achieve higher results compared to the original feature set.

The lowest performance was obtained with IBk before and after feature selection. The performance measures, which were 0.7697 Correlation Coefficient, 0.539 MAE, 52.6711 RAE (%) before the feature selection. By application of GA to select of features, performance measures improved to 0.934 Correlation Coefficient, 0.3142 MAE, 30.7074 RAE (%) after. And a great improvement was achieved.

In addition, application of CfsSubsetEval by selecting PSO Search, 8 out of 9 algorithms showed the improvement and the highest improvement of this method was recorded as 17% improvement in the correlation coefficient in the first and last results of the IBk algorithm.

4.2.1.3 CfsSubset Evaluation and Genetic Algorithm (GA) Method Results

Attribute selection was applied by selecting the evaluation criterion CfsSubsetEval and the search criterion RandomSearch from the SelectionAttributes function of WEKA. The number of attributes, which was 9 before the process, was reduced to 4 by applying the method, by selecting the attributes named Development effort hours, Function point data, Project duration (calendar months), System requirements size in raw Albrecht function points. With the addition of the dependent variable, Effort provided by application Use, 5 attributes selected for the most effective subset of attributes were determined as the model input.

The results of the algorithms obtained without feature selection and the performance outputs of the model created with feature subset after feature selection is given in Table 4.7.

Table 4.7: Performance Measures of Models with Finnish Original Feature and Selected Feature Set by GA

Machine Learning Algorithms	Finnish Original Dataset (9 Feature)			Feature Selection with CFS+ GA (5 Feature)		
	Correlation Coefficient	MAE	RAE (%)	Correlation Coefficient	MAE	RAE (%)
LinearRegression	0.9607	0.254	24.8268	0.9607	0.254	24.8268
RandomForest	**0.9818**	**0.1649**	**16.1147**	**0.9942**	**0.0976**	**9.5354**
Bagging	0.9801	0.1747	17.074	0.9857	0.1532	14.967
MultilayerPerceptron	0.9575	0.2297	22.4464	0.9853	0.1376	13.4468
SMOreg	0.962	0.2341	22.8759	0.9702	0.2213	21.6292
IBk	0.7697	0.539	52.6711	0.934	0.3142	30.7074
KStar	0.9889	0.1344	13.1344	0.9923	0.1127	11.0156
Random Tree	0.9029	0.3654	35.7136	0.9843	0.1461	14.2734
M5p	0.9692	0.2146	20.9732	0.9715	0.2158	21.0941

The best performance was obtained by KStar before feature selection, and performance measurements were recorded as 0.9889 Correlation Coefficient, 0.1344 MAE, 13.1344 RAE (%). By applying feature selection in Finnish dataset, the improvement was gained in RandomForest algorithm and the best performance was revised as RandomForest. Performance measures were observed as 0.9942 Correlation Coefficient, 0.0976 MAE, 9.5354 RAE (%).

The lowest performance was obtained with IBk before and after feature selection. The performance measures, which were 0.7697 Correlation Coefficient, 0.539 MAE, 52.6711 RAE (%) before the feature selection. By application of GA to select of features, performance measures improved to 0.934 Correlation Coefficient, 0.3142 MAE, 30.7074 RAE (%) after. And a great improvement was achieved.

In addition, application of CfsSubsetEval by selecting GA Search, 8 out of 9 algorithms showed the improvement and the highest improvement of this method was recorded as 17% improvement in the correlation coefficient in the first and last results of the IBk algorithm.

4.2.1.4 ClassifierAtt. Evaluation and Ranker Search Method Results

Attribute selection was applied by selecting the evaluation criterion ClassifierAttEval and the search criterion Ranker from the SelectionAttributes function of WEKA. The number of attributes was 9 before the process. The ranker search method, sorts the attributes in order of impact on the model when implemented with ClassifierAttEval the resulting sequence was obtained as: Insize, prod, dev.eff.hrs., hw, at, FP, co, ID. In order not to insert the most ineffective features into

the model as unnecessary inputs, the last 3 features FP, co, ID were removed from the set.

The results of the algorithms obtained without feature selection and the performance outputs of the model created with feature subset after feature selection is given Table 4.8

Table 4.8: Performance Measures of Models with Finnish Original Feature and Selected Feature Set by ClassifierAtt.Evaluation + Ranker

Machine Learning Algorithms	**Finnish Original Dataset (9 Feature)**			**Feature Selection with ClassifierAtt.Evaluation + Ranker (6 Feature)**		
	Correlation Coefficient	**MAE**	**RAE (%)**	**Correlation Coefficient**	**MAE**	**RAE (%)**
LinearRegression	0.9607	0.254	24.8268	0.9658	0.235	22.9611
RandomForest	0.9818	0.1649	16.1147	0.9892	0.1378	13.4643
Bagging	0.9801	0.1747	17.074	0.9807	0.1692	16.5387
MultilayerPerceptron	0.9575	0.2297	22.4464	0.9656	0.2006	19.6052
SMOreg	0.962	0.2341	22.8759	0.9629	0.2445	23.8935
IBk	0.7697	0.539	52.6711	0.7421	0.5756	56.2528
KStar	**0.9889**	**0.1344**	**13.1344**	**0.9948**	**0.0873**	**8.5274**
Random Tree	0.9029	0.3654	35.7136	0.9642	0.2274	22.2251
M5p	0.9692	0.2146	20.9732	0.9783	0.1868	18.2566

The best performance was obtained by KStar before and after the feature selection was applied. The performance measures, which were 0.9889 Correlation Coefficient, 0.1344 MAE, 13.1344 RAE (%) before the feature selection. By selecting the feature subset by ClassifierAttEval and Ranker technique, the results are improved to 0.9948 Correlation Coefficient, 0.0873 MAE, 8.53274 RAE (%).

The lowest performance was obtained with IBk before and after feature selection. The performance measures, which were 0.7697 Correlation Coefficient, 0.539 MAE, 52.6711 RAE (%) before the feature selection. When applied ClassifierAttEval and Ranker to select the feature subset, performance measures are noted as 0.7421 Correlation Coefficient, 0.0873 MAE, 8.5274 RAE (%) and decrease was observed.

In addition, application of ClassifierAttEval by selecting Ranker Search, 8 out of 9 algorithms showed the improvement and the highest improvement of this method was recorded as 6% improvement in the correlation coefficient on RandomTree algorithm.

4.2.1.5 Corr. Att. Evaluation and Ranker Search Method Results

Attribute selection was applied by selecting the evaluation criterion Corr. Att.Evaluation and the search criterion Ranker from the SelectionAttributes function of WEKA. The number of attributes was 9 before the process. The ranker search method, sorts the attributes in order of impact and gives coefficient to the feature in accordance with rank value to take with coefficient the impact of feature, when implemented with Corr. Att.Evaluation.

The final sequence was obtained as: dev.eff.hrs., Insize, FP, Prod, co, ID, at, hw. In order not to insert the most ineffective features into the model as unnecessary inputs, the last 3 features ID, at, hw were removed from the set.

The results of the algorithms obtained without feature selection and the performance outputs of the model created with feature subset after feature selection is given in Table 4.9.

Table 4.9: Performance Measures of Models with Finnish Original Feature and Selected Feature Set by Corr. Att.Evaluation + Ranker

Machine Learning Algorithms	**Finnish Original Dataset (9 Feature)**			**Feature Selection with Corr. Att.Evaluation + Ranker (6 Feature)**		
	Correlation Coefficient	**MAE**	**RAE (%)**	**Correlation Coefficient**	**MAE**	**RAE (%)**
LinearRegression	0.9607	0.254	24.8268	0.9607	0.254	24.8268
RandomForest	0.9818	0.1649	16.1147	0.9901	0.1298	12.6897
Bagging	0.9801	0.1747	17.074	0.9845	0.1587	15.5069
MultilayerPerceptron	0.9575	0.2297	22.4464	0.9833	0.1535	15.0001
SMOreg	0.962	0.2341	22.8759	0.9686	0.2221	21.7002
IBk	0.7697	0.539	52.6711	0.9158	0.3518	34.3834
KStar	**0.9889**	**0.1344**	**13.1344**	**0.9912**	**0.1284**	**12.546**
Random Tree	0.9029	0.3654	35.7136	0.9663	0.2324	22.7083
M5p	0.9692	0.2146	20.9732	0.9729	0.2006	19.6081

The best performance was obtained by KStar before and after the feature selection was applied. The performance measures, which were 0.9889 Correlation Coefficient, 0.1344 MAE, 13.1344 RAE (%) before the feature selection. By selecting the feature subset by Corr. Att.Evaluation and Ranker technique, the results are improved to 0.9912 Correlation Coefficient, 0.1284 MAE, 12.546 RAE (%).

The lowest performance was obtained with IBk before and after feature selection. The performance measures, which were 0.7697 Correlation Coefficient, 0.539 MAE, 52.6711 RAE (%) before the feature selection. When applied Corr.

Att.Evaluation and Ranker to select the feature subset, performance measures are noted as 0.9158 Correlation Coefficient, 0.3518 MAE, 34.3834 RAE (%) and high improvement was observed.

In addition, application of Corr. Att.Evaluation by selecting Ranker Search, 8 out of 9 algorithms showed the improvement and the highest improvement of this method was recorded as 15% improvement in the correlation coefficient on IBk algorithm.

4.2.1.6 Relief Att.Evaluation and Ranker Search Method Results

Attribute selection was applied by selecting the evaluation criterion Relief Att.Evaluation and the search criterion Ranker from the SelectionAttributes function of WEKA. The number of attributes was 9 before the process. The ranker search method, sorts the attributes in order of impact and gives coefficient to the feature in accordance with rank value to take with coefficient the impact of feature, when implemented with Relief Att.Evaluation.

The final sequence was obtained as: dev.eff.hrs., Insize, prod, FP, hw, co, ID, at. In order not to insert the most ineffective features into the model as unnecessary inputs, the last 3 features co, ID, at were removed from the set.

The results of the algorithms obtained without feature selection and the performance outputs of the model created with feature subset after feature selection is given in Table 4.10.

Table 4.10: Performance Measures of Models with Finnish Original Feature and Selected Feature Set by Relief Att.Evaluation + Ranker

Machine Learning Algorithms	**Finnish Original Dataset (9 Feature)**			**Feature Selection with Relief Att.Evaluation + Ranker (6 Feature)**		
	Correlation Coefficient	**MAE**	**RAE (%)**	**Correlation Coefficient**	**MAE**	**RAE (%)**
LinearRegression	0.9607	0.254	24.8268	0.9607	0.254	24.8268
RandomForest	0.9818	0.1649	16.1147	0.9907	0.118	11.5345
Bagging	0.9801	0.1747	17.074	0.9852	0.1544	15.0929
MultilayerPerceptron	0.9575	0.2297	22.4464	0.9647	0.1809	17.6834
SMOreg	0.962	0.2341	22.8759	0.9653	0.2314	22.6141
IBk	0.7697	0.539	52.6711	0.931	0.3319	32.4397
KStar	**0.9889**	**0.1344**	**13.1344**	**0.9916**	**0.1208**	**11.8098**
Random Tree	0.9029	0.3654	35.7136	0.9705	0.2041	19.9438
M5p	0.9692	0.2146	20.9732	0.9729	0.2006	19.6081

The best performance was obtained by KStar before and after the feature selection was applied. The performance measures, which were 0.9889 Correlation Coefficient, 0.1344 MAE, 13.1344 RAE (%) before the feature selection. By selecting the feature subset by Relief Att.Evaluation and Ranker technique, the results are improved to 0.9916 Correlation Coefficient, 0.1208 MAE, 11.8098 RAE (%).

The lowest performance was obtained with IBk before and after feature selection. The performance measures, which were 0.7697 Correlation Coefficient, 0.539 MAE, 52.6711 RAE (%) before the feature selection. When applied Relief Att.Evaluation and Ranker to select the feature subset, performance measures are noted as 0.931 Correlation Coefficient, 0.3319 MAE, 32.4397 RAE (%) and high improvement was observed.

In addition, application of Relief Att.Evaluation by selecting Ranker Search, 8 out of 9 algorithms showed the improvement and the highest improvement of this method was recorded as 17% improvement in the correlation coefficient on IBk algorithm.

4.2.2 China Dataset Cost Estimation Results with Feature Selection Methods

4.2.2.1 CfsSubset Evaluation and Random Search Method Results

For the China dataset, attribute selection was applied by selecting the evaluation criterion CfsSubsetEval and the search criterion RandomSearch from the SelectionAttributes function of WEKA. The number of attributes, which was 19 before the process, was reduced to 6 by applying the method, by selecting the attributes named ID, Output, Interface, Added, Duration, N_effort. With the addition of the dependent variable, Effort, provided by application Use, 7 attributes selected for the most effective subset of attributes were determined as the model input.

The results of the algorithms obtained without feature selection and the performance outputs of the model created with feature subset after feature selection is given in Table 4.11.

Table 4.11: Performance Measures of Models with China Original Feature and Selected Feature Set by CFS+ RandomSearch

Machine Learning Algorithms	China Original Dataset (19 Feature)			Feature Selection with CFS+ RandomSearch (7 Feature)		
	Correlation Coefficient	MAE	RAE (%)	Correlation Coefficient	MAE	RAE (%)
LinearRegression	0.9889	362.939	9.809	0.986	365.1502	9.8688
RandomForest	0.9591	557.7718	15.0747	0.9681	553.6869	14.9643
Bagging	0.9605	511.9898	13.8374	0.9625	508.1267	13.733
MultilayerPerceptron	0.9733	461.3901	12.4698	0.9776	548.369	14.8206
SMOreg	**0.9897**	**270.4561**	**7.3095**	**0.9866**	**351.7668**	**9.5071**
IBk	0.8918	1571.1824	42.4638	0.9362	1175.988	31.783
KStar	0.9646	628.608	16.9892	0.9648	496.0903	13.4077
Random Tree	0.9283	943.0361	25.4871	0.9244	850.8577	22.9958
M5p	0.9842	392.7912	10.6158	0.9832	372.001	10.0539

The best performance was obtained by SMOreg before and after the feature selection was applied. Before the selection of feature, performance measures was noted as 0.9897 Correlation Coefficient, 362.939 MAE, 9.809 RAE (%). After selection of features, performance measures observed as 0.9866 Correlation Coefficient, 365.1502 MAE, 9.8688 RAE (%) and the improvement could not be obtained.

The lowest performance was obtained with IBk before selection of feature, which saved performance measures as 0.8918 Correlation Coefficient, 1571.1824 MAE, 42.4638 RAE (%). After feature selection, while IBk performance measures gained improvement as 0.9362 Correlation Coefficient, 1175.988, MAE, 31.783 RAE (%), the lowest performing algorithm is updated as RandomTree with 0. 9244 Correlation Coefficient, 850.8577 MAE, 22.9958 RAE (%).

Although there was no improvement in the best result compared to the original dataset by applying feature selection in the China dataset, after applying this method, 5 out of 9 algorithms showed the improvement and the highest improvement of this method was recorded as 4% improvement on the correlation coefficient of IBk algorithm.

4.2.2.2 CfsSubset Evaluation and Particle Swarm Optimization (PSO) Method Results

Attribute selection was applied by selecting the evaluation criterion CfsSubsetEval and the search criterion RandomSearch from the SelectionAttributes function of WEKA. The number of attributes, which was 19 before the process, was

reduced to 8 by applying the method, by selecting the attributes named ID, Input, Output, Enquiry, File, Resource, Duration, N_effort. With the addition of the dependent variable, Effort provided by application Use, 9 attributes selected for the most effective subset of attributes were determined as the model input.

The results of the algorithms obtained without feature selection and the performance outputs of the model created with feature subset after feature selection is given in Table 4.12.

Table 4.12: Performance Measures of Models with China Original Feature and Selected Feature Set by CFS+ PSO

Machine Learning Algorithms	**China Original Dataset (19 Feature)**			**Feature Selection with CFS+ PSO (9 Feature)**		
	Correlation Coefficient	**MAE**	**RAE (%)**	**Correlation Coefficient**	**MAE**	**RAE (%)**
LinearRegression	0.9889	362.939	9.809	0.986	395.7794	10.6966
RandomForest	0.9591	557.7718	15.07	0.9602	583.8073	15.7784
Bagging	0.9605	511.9898	13.83	0.9597	519.3041	14.035
MultilayerPerceptron	0.9733	461.3901	12.46	0.9767	514.2803	13.8993
SMOreg	**0.9897**	**270.4561**	**7.30**	**0.9853**	**360.9192**	**9.7544**
IBk	0.8918	1571.182	42.46	0.9081	1500.1563	40.5442
KStar	0.9646	628.608	16.98	0.9717	528.6757	14.2883
Random Tree	0.9283	943.0361	25.48	0.9155	872.0081	23.5675
M5p	0.9842	392.7912	10.61	0.9832	391.7027	10.5864

Before feature selection, the best performance was obtained with LinearRegression in the China dataset, however, with the application of feature selection with PSO, the best result among the algorithms was SMOReg with 0.9853 Correlation Coefficient, 360.9192 MAE, 9.7544 RAE (%) performance measurements. However, it was observed that the correlation coefficient obtained in the original dataset could not be reached.

The lowest performance was obtained with IBk before and after feature selection. Before the feature selection, the performance measures were 0.8918 Correlation Coefficient, 1571.182 MAE, 42.46 RAE (%). The measures improved to 0.9081 Correlation Coefficient, 1500.1563 MAE, 40.5442 RAE (%) after the feature selection.

Although there was no improvement in the best result compared to the original dataset by applying feature selection in the China dataset, by applying this method, 4 out of 9 algorithms showed improvement, and the highest improvement of this method was recorded as 1.5% improvement on the correlation coefficient of IBk algorithm.

4.2.2.3 CfsSubset Evaluation and Genetic Algorithm (GA) Method Results

Attribute selection was applied by selecting the evaluation criterion CfsSubsetEval and the search criterion GA from the SelectionAttributes function of WEKA. The number of attributes, which was 19 before the process, was reduced to 9 by applying the method, by selecting the attributes named ID, Input, Output, Enquiry, File, PDR_UFP, Resource, Duration, N_effort. With the addition of the dependent variable, Effort provided by application Use, 10 attributes selected for the most effective subset of attributes were determined as the model input.

The results of the algorithms obtained without feature selection and the performance outputs of the model created with feature subset after feature selection is given in Table 4.13.

Table 4.13: Performance Measures of Models with China Original Feature and Selected Feature Set by CFS+ GA

Machine Learning Algorithms	**China Original Dataset (19 Feature)**			**Feature Selection with CFS+ GA (10 Feature)**		
	Correlation Coefficient	**MAE**	**RAE (%)**	**Correlation Coefficient**	**MAE**	**RAE (%)**
LinearRegression	**0.9889**	**362.939**	**9.809**	**0.9859**	**411.7442**	**11.1281**
RandomForest	0.9591	557.7718	15.0747	0.9584	578.3479	15.6308
Bagging	0.9605	511.9898	13.8374	0.9596	519.546	14.0416
MultilayerPerceptron	0.9733	461.3901	12.4698	0.9746	497.3849	13.4426
SMOreg	0.9897	270.4561	7.3095	0.9847	358.5216	9.6896
IBk	0.8918	1571.182	42.4638	0.9076	1444.875	39.0501
KStar	0.9646	628.608	16.9892	0.9726	543.4151	14.6867
Random Tree	0.9283	943.0361	25.4871	0.8726	1056.691	28.5588
M5p	0.9842	392.7912	10.6158	0.984	401.4127	10.8488

Before feature selection, the best performance was obtained with SMOreg in the China dataset, however, with the application of feature selection with GA, the best result among the algorithms is LinearRegression with 0.9859 Correlation Coefficient, 411.7442 MAE, 11.1281 RAE (%) performance measurements. However, it was observed that the correlation coefficient obtained as 0.9897 in the original dataset could not be reached.

The lowest performance was obtained with IBk before and after feature selection. Before the feature selection, the performance measures were 0.8918 Correlation Coefficient, 1571.1824 MAE, 42.4638 RAE (%). The measures improved to 0.9076 Correlation Coefficient, 1444.8758 MAE, 39.0501 RAE (%) after the feature selection.

Although there was no improvement in the best result compared to the original dataset by applying feature selection in the China dataset, by applying this method, 3 out of 9 algorithms showed improvement, and the highest improvement of this method was recorded as 1.5% improvement on the correlation coefficient of IBk algorithm.

4.2.2.4 ClassifierAtt. Evaluation and Ranker Search Method Results

Attribute selection was applied by selecting the evaluation criterion ClassifierAttEval and the search criterion Ranker from the SelectionAttributes function of WEKA. The number of attributes was 19 before the process. The ranker search method, sorts the attributes in order of impact on the model when implemented with ClassifierAttEval. The resulting sequence was obtained as: N_effort, Enquiry, Interface, File, Output, Duration, Input, AFP, Added, Changed, Deleted, PDR_AFP, Dev.Type, Resource, NPDU_UFP, NPDR_AFP, PDR_UFP, ID. In order not to insert the most ineffective features into the model as unnecessary inputs, the last 3 features NPDR_AFP, PDR_UFP, ID were removed from the set.

The results of the algorithms obtained without feature selection and the performance outputs of the model created with feature subset after feature selection is given in Table 4.14.

Table 4.14: Performance Measures of Models with China Original Feature and Selected Feature Set by ClassifierAttEvaluation + Ranker

Machine Learning Algorithms	**China Original Dataset (19 Feature)**			**Feature Selection with ClassifierAttEvaluation + Ranker (16 Feature)**		
	Correlation Coefficient	**MAE**	**RAE (%)**	**Correlation Coefficient**	**MAE**	**RAE (%)**
LinearRegression	0.9889	362.939	9.809	0.9872	413.9045	11.1865
RandomForest	0.9591	557.7718	15.0747	0.9583	586.4616	15.8501
Bagging	0.9605	511.9898	13.8374	0.9642	492.293	13.305
MultilayerPerceptron	0.9733	461.3901	12.4698	0.965	549.6564	14.8554
SMOreg	**0.9897**	**270.4561**	**7.3095**	**0.9887**	**304.4746**	**8.2289**
IBk	0.8918	1571.1824	42.4638	0.847	1343.4469	36.3089
KStar	0.9646	628.608	16.9892	0.9694	596.5132	16.1218
Random Tree	0.9283	943.0361	25.4871	0.8617	1073.5312	29.0139
M5p	0.9842	392.7912	10.6158	0.9835	390.3778	10.5506

Before feature selection, the best performance was obtained with SMOreg in the China dataset, similarly, with the application of feature selection with ClassifierAttEval and Ranker, the best result among the algorithms again SMOreg with

0.9887 Correlation Coefficient, 304.4746 MAE, 8.2289 RAE (%) performance measurements. However, it was observed that the correlation coefficient obtained in the original dataset could not be reached.

The lowest performance was obtained with IBk before and after feature selection. Before the feature selection, the performance measures were 0.847 Correlation Coefficient, 1343.4369 MAE, 42.4638 RAE (%). The measures improved to 0.9076 Correlation Coefficient, 1444.8758 MAE, 36.3089 RAE (%) after the feature selection. And decrease has saved for IBk.

In addition, application of ClassifierAttEval by selecting Ranker Search, 2 out of 9 algorithms showed the improvement and the highest improvement of this method was recorded as 0.5% improvement in the correlation coefficient on KStar algorithm.

4.2.2.5 Corr. Att. Evaluation and Ranker Search Search Method Results

Attribute selection was applied by selecting the evaluation criterion Corr. Att.Evaluation and the search criterion Ranker from the SelectionAttributes function of WEKA. The number of attributes was 9 before the process. The ranker search method, sorts the attributes in order of impact and gives coefficient to the feature in accordance with rank value to take with coefficient the impact of feature, when implemented with Corr. Att.Evaluation.

The final sequence was obtained as: N_effort, Added, AFP, File, Input, Output, Enquiry, Duration, Interface, PDR_UFP, NPDU_UFP, PDR_AFP, Resource, NPDR_AFP, Changed, ID, Deleted, Dev.Type. In order not to insert the most ineffective features into the model as unnecessary inputs, the last 3 features ID, Deleted, Dev.Type were removed from the set.

The results of the algorithms obtained without feature selection and the performance outputs of the model created with feature subset after feature selection is given in Table 4.15.

Table 4.15: Performance Measures of Models with China Original Feature and Selected Feature Set by Corr. Att.Evaluation + Ranker

Machine Learning Algorithms	China Original Dataset (19 Feature)			Feature Selection with Corr. Att.Evaluation + Ranker (16 Feature)		
	Correlation Coefficient	MAE	RAE (%)	Correlation Coefficient	MAE	RAE (%)
LinearRegression	0.9889	362.939	9.809	0.989	362.2063	9.7892
RandomForest	0.9591	557.7718	15.0747	0.9623	543.9122	14.7001
Bagging	0.9605	511.9898	13.8374	0.9618	503.6359	13.6116
MultilayerPerceptron	**0.9733**	**461.3901**	**12.4698**	**0.9912**	**406.1195**	**10.976**
SMOreg	0.9897	270.4561	7.3095	0.9898	270.2385	7.3036
IBk	0.8918	1571.182	42.4638	0.8396	1418.9499	38.3495
KStar	0.9646	628.608	16.9892	0.9638	619.7006	16.7484
Random Tree	0.9283	943.0361	25.4871	0.8409	1142.3578	30.8741
M5p	0.9842	392.7912	10.6158	0.9835	386.561	10.4474

Before feature selection, the best performance was obtained with SMOreg in the China dataset. However, with the application of feature selection with Corr. Att.Evaluation and Ranker, the best result among the algorithms is MultilayerPerceptron with 0. 9912 Correlation Coefficient, 406. 1195 MAE, 10.976 RAE (%) performance measurements. It has been observed that higher results have been achieved even if it is with low rate. Based on this low improvement, it has succeeded in being the method that can achieve higher results compared to the original feature set.

The lowest performance was obtained with IBk before and after feature selection. Before the feature selection, the performance measures were 0.8918 Correlation Coefficient, 1571.1824MAE, 42.4638 RAE (%). The measures noted as 0.8396 Correlation Coefficient, 1418.94998 MAE, 38.3495 RAE (%) after the feature selection. And decrease has saved for IBk.

In addition, application of Corr. Att.Evaluation by selecting Ranker Search, 5 out of 9 algorithms showed the improvement and the highest improvement of this method was recorded as 2% improvement in the correlation coefficient on MultilayerPerceptron algorithm.

4.2.2.6 Relief Att. Evaluation and Ranker Search Method Results

Attribute selection was applied by selecting the evaluation criterion Relief Att.Evaluation and the search criterion Ranker from the SelectionAttributes function of WEKA. The number of attributes was 9 before the process. The ranker search method, sorts the attributes in order of impact and gives coefficient to the feature in

accordance with rank value to take with coefficient the impact of feature, when implemented with Relief Att.Evaluation.

The final sequence was obtained as: N_effort, File, AFP, Added, Enquiry, Output, Input, Resource, Interface, Duration, ID, PDR_AFP, PDR_UFP, NPDU_UFP, NPDR_AFP, Deleted, Dev.Type. In order not to insert the most ineffective features into the model as unnecessary inputs, the last 3 features NPDR_AFP, Deleted, Dev.Type were removed from the set.

The results of the algorithms obtained without feature selection and the performance outputs of the model created with feature subset after feature selection is given in Table 4.16.

Table 4.16: Performance Measures of Models with China Original Feature and Selected Feature Set by Relief Att.Evaluation + Ranker

Machine Learning Algorithms	**China Original Dataset (19 Feature)**			**Feature Selection with Relief Att.Evaluation + Ranker (16 Feature)**		
	Correlation Coefficient	**MAE**	**RAE (%)**	**Correlation Coefficient**	**MAE**	**RAE (%)**
LinearRegression	0.9889	362.939	9.809	0.9886	366.1417	9.8956
RandomForest	0.9591	557.7718	15.0747	0.9627	546.8556	14.7797
Bagging	0.9605	511.9898	13.8374	0.9629	501.3717	13.5504
MultilayerPerceptron	**0.9733**	**461.3901**	**12.4698**	**0.9914**	**370.1846**	**10.0048**
SMOreg	0.9897	270.4561	7.3095	0.9898	269.3637	7.28
IBk	0.8918	1571.182	42.4638	0.887	1581.3467	1581.34
KStar	0.9646	628.608	16.9892	0.965	617.641	16.6928
Random Tree	0.9283	943.0361	25.4871	0.8098	1263.9482	34.1603
M5p	0.9842	392.7912	10.6158	0.9852	378.51	10.2299

Before feature selection, the best performance was obtained with SMOreg in the China dataset. However, with the application of feature selection with Corr. Att.Evaluation and Ranker, the best result among the algorithms is MultilayerPerceptron with 0. 9914 Correlation Coefficient, 370.1846 MAE, 10.0048 RAE (%) performance measurements. It has been observed that higher results have been achieved even if it is with low rate. Based on this low improvement, it has succeeded in being the method that can achieve higher results compared to the original feature set.

The lowest performance was obtained with IBk before feature selection. The performance measures were 0.8918 Correlation Coefficient, 1571.1824MAE, 42.4638 RAE (%). By applying feature selection technique, the lowest measured algorithm is

updated as RandomTree and measures noted as 0.8098 Correlation Coefficient, 1263.9482 MAE, 34.1603 RAE (%) after the feature selection.

In addition, application of Relief Att.Evaluation by selecting Ranker Search, 6 out of 9 algorithms showed the improvement and the highest improvement of this method was recorded as 2% improvement in the correlation coefficient on MultilayerPerceptron algorithm.

4.2.3 Maxwell Dataset Cost Estimation Results with Feature Selection Methods

4.2.3.1 CfsSubset Evaluation and Random Search Method Results

For the Maxwell dataset, attribute selection was applied by selecting the evaluation criterion CfsSubsetEval and the search criterion RandomSearch from the SelectionAttributes function of WEKA. The number of attributes, which was 27 before the process, was reduced to 15 by applying the method, by selecting the attributes named Year, Application type, Hardware platform, Development Env, adequacy, Staff availability, Software logical complexity, Requirements volatility, Quality requirements, Efficiency requirements, Installation requirements, Staff application knowledge, Staff tool skills, Duration, Function points, Time. With the addition of the dependent variable, Work hours Effort, 16 attributes selected for the most effective subset of attributes were determined as the model input.

The results of the algorithms obtained without feature selection and the performance outputs of the model created with feature subset after feature selection is given in Table 4.17.

Table 4.17: Performance Measures of Models with Maxwell Original Feature and Selected Feature Set by CFS+ RandomSearch

Machine Learning Algorithms	**Maxwell Original Dataset (27 Feature)**			**Feature Selection with CFS+ RandomSearch (16 Feature)**		
	Correlation Coefficient	**MAE**	**RAE (%)**	**Correlation Coefficient**	**MAE**	**RAE (%)**
LinearRegression	**0.8085**	**4157.5897**	**66.1746**	**0.8354**	**3610.4878**	**57.4666**
RandomForest	0.7612	3998.2174	63.638	0.7624	3768.4553	59.9809
Bagging	0.7711	3949.1671	62.8573	0.7753	3835.1733	61.0429
MultilayerPerceptron	0.7641	4764.3788	75.8327	0.7215	5389.5218	85.7828
SMOreg	0.8191	3812.9653	60.6894	0.8091	3520.7457	56.0383
IBk	0.463	5517.129	87.8139	0.7427	4725.7419	75.2177
KStar	0.7336	4618.2302	73.5065	0.8174	4154.1848	66.1204
Random Tree	0.569	5686.9672	90.5171	0.6147	5068.6745	80.676
M5p	0.8175	3718.2692	59.1822	0.8173	3677.0131	58.5255

The highest performance was obtained with SMOreg before feature selection, and performance measurements were recorded as 0.8191 Correlation Coefficient, 3812.9653 MAE, 60.6894 RAE. By applying feature selection in Maxwell dataset, 3% improvement was achieved in LinearRegression algorithm and the best performance was revised as LinearRegression. Performance measures were observed as 0.8354 Correlation Coefficient, 3610.4878, MAE, 57.4666 RAE (%).

The lowest performance was obtained with IBk before selection of feature, which saved performance measures as 0. 463 Correlation Coefficient, 5517.129 MAE, 87.8139 RAE (%). After feature selection, while IBk performance measures gained improvement as 0.7427 Correlation Coefficient, 4725.7419, MAE, 75.2177 RAE (%), and the lowest performing algorithm is updated as RandomTree with 0. 6147 Correlation Coefficient, 5068.6745 MAE, 80.676 RAE (%).

By applying CfsSubsetEval and Random Search before the model was set up, 6 out of 9 algorithms showed the improvement and the highest improvement of this method was recorded as 28% improvement in the correlation coefficient of IBk algorithm.

4.2.3.2 CfsSubset Evaluation and Particle Swarm Optimization (PSO) Method Results

Attribute selection was applied by selecting the evaluation criterion CfsSubsetEval and the search criterion PSO from the SelectionAttributes function of WEKA. The number of attributes, which was 27 before the process, was reduced to 15 with the implementation of the method, Year, Hardware platform, Database, Development Env, adequacy, Staff availability, Software logical complexity, Requirements volatility, Quality requirements, Efficiency requirements, Installation requirements, Staff application knowledge, Staff tool skills, Duration, Function points, Time. With the addition of the dependent variable, Effort provided by application Use, 16 attributes selected for the most effective subset of attributes were determined as the model input.

The results of the algorithms obtained without feature selection and the performance outputs of the model created with feature subset after feature selection is given in Table 4.18.

Table 4.18: Performance Measures of Models with Maxwell Original Feature and Selected Feature Set by CFS+ PSO

Machine Learning Algorithms	Maxwell Original Dataset (19 Feature)			Feature Selection with CFS+ PSO (9 Feature)		
	Correlation Coefficient	MAE	RAE (%)	Correlation Coefficient	MAE	RAE (%)
LinearRegression	0.8085	4157.5897	66.1746	0.835	3550.9565	56.5191
RandomForest	0.7612	3998.2174	63.638	0.7916	3655.784	58.1876
Bagging	0.7711	3949.1671	62.8573	0.7738	3840.6234	61.1296
MultilayerPerceptron	0.7641	4764.3788	75.8327	0.712	5175.8943	82.3826
SMOreg	0.8191	3812.9653	60.6894	0.8361	3188.8894	50.7562
IBk	0.463	5517.129	87.8139	0.7432	4848.2419	77.1675
KStar	0.7336	4618.2302	73.5065	**0.85**	**4040.6726**	**64.3137**
Random Tree	0.569	5686.9672	90.5171	0.613	5151.1169	81.9882
M5p	0.8175	3718.2692	59.1822	0.834	3654.1942	58.1623

The highest performance was obtained with SMOreg before feature selection, and performance measurements were recorded as 0.8191 Correlation Coefficient, 3812.9653 MAE, 60.6894 RAE. By applying feature selection in Maxwell dataset, 15.9% improvement was achieved in KStar algorithm and the best performance was revised as KStar. Performance measures were observed as 0.85 Correlation Coefficient, 4040.6726, MAE, 58.1623 RAE (%).

The lowest performance was obtained with IBk before selection of feature, which saved performance measures as 0.463 Correlation Coefficient, 5517.129 MAE, 87.8139 RAE (%). After feature selection, while IBk performance measures gained improvement as 0.7432 Correlation Coefficient, 4848.2419, MAE, 77.1675 RAE (%), and the lowest performing algorithm is updated as RandomTree with 0. 0.613 Correlation Coefficient, 5151.1169 MAE, 81.9882 RAE (%).

In addition, after applying this method, 8 out of 9 algorithms showed the improvement and the highest improvement of this method was recorded as 28% improvement in the correlation coefficient in the first and last results of the IBk algorithm.

4.2.3.3 CfsSubset Evaluation and Genetic Algorithm (GA) Method Results

Attribute selection was applied by selecting the evaluation criterion CfsSubsetEval and the search criterion RandomSearch from the SelectionAttributes function of WEKA. The number of attributes, which was 27 before the process, was reduced to 19 by applying the method, by selecting the attributes named Year, Application type, Hardware platform, Database, where developed, Customer

participation, Development Env, adequacy, Standards use, Tools use, Software logical complexity, Requirements volatility, Quality requirements, Efficiency requirements, Installation requirements, Staff application knowledge, Staff tool skills, Duration, Function points, Time. With the addition of the dependent variable, Effort provided by application Use, 20 attributes selected for the most effective subset of attributes were determined as the model input.

The results of the algorithms obtained without feature selection and the performance outputs of the model created with feature subset after feature selection is given in Table 4.19.

Table 4.19: Performance Measures of Models with Maxwell Original Feature and Selected Feature Set by CFS+ GA

Machine Learning Algorithms	**Maxwell Original Dataset (27 Feature)**			**Feature Selection with CFS+ GA (20 Feature)**		
	Correlation Coefficient	**MAE**	**RAE (%)**	**Correlation Coefficient**	**MAE**	**RAE (%)**
LinearRegression	0.8085	4157.5897	66.1746	0.8544	3395.0666	54.0379
RandomForest	0.7612	3998.2174	63.638	0.7621	3827.5684	60.9218
Bagging	0.7711	3949.1671	62.8573	0.7704	3898.5336	62.0514
MultilayerPerceptron	0.7641	4764.3788	75.8327	0.816	4146.3094	65.9951
SMOreg	0.8191	3812.9653	60.6894	0.818	3522.3771	56.0642
IBk	0.463	5517.129	87.8139	0.7593	4494.6774	71.5399
KStar	**0.7336**	**4618.2302**	**73.5065**	**0.8596**	**4078.3244**	**64.913**
Random Tree	0.569	5686.9672	90.5171	0.4398	5223.2222	83.1359
M5p	0.8175	3718.2692	59.1822	0.8092	3685.2814	58.6571

The best performance was obtained by SMOReg before the feature selection. The performance measures, which were 0.8191 Correlation Coefficient, 3812.9653 MAE, 60.6894 RAE (%). By application of feature selection-based GA, the best performing algorithm is updated as KStar with 0.8596 Correlation Coefficient, 4078.3244 MAE, 64.913 RAE (%). KStar has shown 17% improvement by GA Feature selection. It has been observed that higher results have been achieved. Based on this improvement, it has succeeded in being the method that can achieve higher results compared to the original feature set.

The lowest performance was obtained with IBk before selection of feature, which saved performance measures as 0.463 Correlation Coefficient, 5517.129 MAE, 87.8139 RAE (%). After feature selection, while IBk performance measures gained improvement as 0.7593 Correlation Coefficient, 4494.6774 MAE, 71.5399 RAE (%),

and the lowest performing algorithm is updated as RandomTree with 0.4398 Correlation Coefficient, 5223.2222 MAE, 64.913RAE (%).

In addition, after applying this method, 5 out of 9 algorithms showed improvement, also application of this method has achieved best among all selection methods, and the highest improvement of this method was recorded as 29% improvement in the correlation coefficient on IBk algorithm.

4.2.3.4 ClassifierAttEval Evaluation and Ranker Search Method Results

Attribute selection was applied by selecting the evaluation criterion ClassifierAttEval and the search criterion Ranker from the SelectionAttributes function of WEKA. The number of attributes was 27 before the process. The ranker search method, sorts the attributes in order of impact on the model when implemented with ClassifierAttEval. The resulting sequence was obtained as: Time, Telon use, Customer participation, Development Env, adequacy, Staff availability, # of development languages, Where developed, Function points, User interface, Application type, Hardware platform, Database, Standards use, Methods use, Tools use, Software logical complexity, Staff tool skills, Staff team skills, Duration, staff application, staff analysis skills, Installation requirements, Requirements volatility, Quality requirements, Efficiency requirements, Year. In order not to insert the most ineffective features into the model as unnecessary inputs, the last 3 features Quality requirements, Efficiency requirements, Year were removed from the set.

The results of the algorithms obtained without feature selection and the performance outputs of the model created with feature subset after feature selection is given in Table 4.20.

Table 4.20: Performance Measures of Models with Maxwell Original Feature and Selected Feature Set by ClassifierAttEvaluation + Ranker

Machine Learning Algorithms	Maxwell Original Dataset (27 Feature)			Feature Selection with ClassifierAttEvaluation + Ranker (24 Feature)		
	Correlation Coefficient	MAE	RAE (%)	Correlation Coefficient	MAE	RAE (%)
LinearRegression	0.8085	4157.5897	66.1746	0.8282	4164.5366	66.2852
RandomForest	0.7612	3998.2174	63.638	0.8015	3654.0125	58.1594
Bagging	0.7711	3949.1671	62.8573	0.7589	4041.8054	64.3317
MultilayerPerceptron	0.7641	4764.3788	75.8327	0.6961	5487.8816	87.3483
SMOreg	0.8191	3812.9653	60.6894	0.8281	3639.1298	57.9225
IBk	0.463	5517.129	87.8139	0.4904	5397.3871	85.908
KStar	0.7336	4618.2302	73.5065	0.7209	4539.8698	72.2592
Random Tree	0.569	5686.9672	90.5171	0.6184	6120.3704	97.4154
M5p	**0.8175**	**3718.2692**	**59.1822**	**0.8515**	**3447.3519**	**54.8701**

The best performance was obtained by SMOReg before the feature selection. The performance measures, which were 0.8191 Correlation Coefficient, 3812.9653 MAE, 60.6894 RAE (%). By application of feature selection-based ClassifierAttEval Evaluation and Ranker Search, the best performing algorithm is updated as M5P with 0.8515 Correlation Coefficient, 3447.3519 MAE, 54.8701 RAE (%). M5P has shown 4% improvement by ClassifierAttEval Evaluation and Ranker Search Feature selection.

The lowest performance was obtained with IBk before selection of feature, which saved performance measures as 0.463 Correlation Coefficient, 5517.129 MAE, 87.8139 RAE (%). After feature selection, while IBk performance measures gained improvement as 0.4904 Correlation Coefficient, 5397.3871 MAE, 85.908 RAE (%), and the lowest performing algorithm is kept as IBk.

In addition, after applying this method, 6 out of 9 algorithms showed improvement, also application of this method has achieved best among all selection methods, and the highest improvement of this method was recorded as 5% improvement in the correlation coefficient on RandomTree algorithm.

4.2.3.5 Corr. Att. Evaluation and Ranker Search Method Results

Attribute selection was applied by selecting the evaluation criterion Corr. Att.Evaluation and the search criterion Ranker from the SelectionAttributes function of WEKA. The number of attributes was 27 before the process. The ranker search method, sorts the attributes in order of impact and gives coefficient to the feature in

accordance with rank value to take with coefficient the impact of feature, when implemented with Corr. Att.Evaluation.

The final sequence was obtained as: Function points, Duration – Süre, Software logical complexity, Installation requirements, # of development languages, Staff team skills, Efficiency requirements, Requirements volatility, Quality requirements, Customer participation, Staff application, Staff analysis skills, Staff availability, User interface, Telon use, Application type, Where developed, Methods use, Tools use, Standards use, Database, Hardware platform, Development Env, adequacy, Time, Year, Staff tool skills. In order not to insert the most ineffective features into the model as unnecessary inputs, the last 3 features Time, Year, Staff tool skills were removed from the set.

The results of the algorithms obtained without feature selection and the performance outputs of the model created with feature subset after feature selection is given in Table 4.21.

Table 4-21: Performance Measures of Models with Maxwell Original Feature and Selected Feature Set by Corr. Att.Evaluation + Ranker

Machine Learning Algorithms	**Maxwell Original Dataset (27 Feature)**			**Feature Selection with Corr. Att.Evaluation + Ranker (24 Feature)**		
	Correlation Coefficient	**MAE**	**RAE (%)**	**Correlation Coefficient**	**MAE**	**RAE (%)**
LinearRegression	0.8085	4157.5897	66.1746	0.8163	3937.4256	62.6704
RandomForest	0.7612	3998.2174	63.638	0.7932	3718.2019	59.1811
Bagging	0.7711	3949.1671	62.8573	0.791	3847.9297	61.2459
MultilayerPerceptron	0.7641	4764.3788	75.8327	0.6801	6143.2445	97.7795
SMOreg	**0.8191**	**3812.9653**	**60.6894**	**0.8336**	**3728.4524**	**59.3442**
IBk	0.463	5517.129	87.8139	0.4487	5720.7419	91.0547
KStar	0.7336	4618.2302	73.5065	0.7315	4558.8152	72.5608
Random Tree	0.569	5686.9672	90.5171	0.6913	5393.8065	85.851
M5p	0.8175	3718.2692	59.1822	0.8247	3643.2708	57.9884

The best performance was obtained by SMOReg before the feature selection. The performance measures, which were 0.8191 Correlation Coefficient, 3812.9653 MAE, 60.6894 RAE (%). With the application of feature selection with Corr. Att.Evaluation and Ranker, the algorithm with best result is kept as SMOreg. Performance measures are saved as 0.8336 Correlation Coefficient, 3728.4524 MAE, 59.3442 RAE (%). It has been observed that higher results have been achieved even if it is with low rate.

The lowest performance was obtained with IBk before selection of feature, which saved performance measures as 0.463 Correlation Coefficient, 5517.129 MAE, 87.8139 RAE (%). After feature selection, while IBk performance measures observed as 0.4487 Correlation Coefficient, 5720.7419 MAE, 91.0547 RAE (%), the decrease is observed and the lowest performing algorithm is kept as IBk.

In addition, after applying this method, 6 out of 9 algorithms showed improvement, also application of this method has achieved best among all selection methods, and the highest improvement of this method was recorded as 13% improvement in the correlation coefficient on RandomTree algorithm.

4.2.3.6 Relief Att. Evaluation and Ranker Search Method Results

Attribute selection was applied by selecting the evaluation criterion Relief Att.Evaluation and the search criterion Ranker from the SelectionAttributes function of WEKA. The number of attributes was 27 before the process. The ranker search method, sorts the attributes in order of impact and gives coefficient to the feature in accordance with rank value to take with coefficient the impact of feature, when implemented with Relief Att.Evaluation.

When Relief Att. Evaluation and Ranker Search Method applied On Maxwell dataset, it has been seen that the data of 13 of the dataset is marked with a negative coefficient. Since it is mentioned in section c that tagged features with negative coefficient affect the model badly in theory or are considered as unnecessary inputs, this feature selection method will be tried twice for this dataset. firstly, by removing the last 3 features as was until this part of the study, and then as second experiment, algorithms will be handled with same feature selection method by removing 13 features to eliminate all the negative values.

4.2.3.6.1 First Experiment-By Removing Last 3 Features

The final sequence was obtained as: Function points, Duration, Staff tool skills, Development Env, adequacy, Telon use, Methods use, Staff team skills, Tools use, Software logical complexity, Staff availability, Time, Year, Installation requirements, Customer participation, Quality requirements, Where developed, Standards use, Database, User interface, Staff analysis skills, # of development languages, Efficiency requirements, Requirements volatility, Hardware platform, Application type, Staff application. In order not to insert the most ineffective features into the model as

unnecessary inputs, the last 3 features Hardware platform, Application type, Staff application were removed from the set.

The results of the algorithms obtained without feature selection and the performance outputs of the model created with feature subset after feature selection is given in Table 4.22

Table 4.22: Performance Measures of Models with Maxwell Original Feature and 24 Selected Feature Set by Relief Att.Evaluation + Ranker

Machine Learning Algorithms	**Maxwell Original Dataset (27 Feature)**			**Feature Selection with Relief Att.Evaluation + Ranker (24 Feature)**		
	Correlation Coefficient	**MAE**	**RAE (%)**	**Correlation Coefficient**	**MAE**	**RAE (%)**
LinearRegression	0.8085	4157.5897	66.1746	0.8208	4278.3998	68.0975
RandomForest	0.7612	3998.2174	63.638	0.7854	3700.0468	58.8921
Bagging	0.7711	3949.1671	62.8573	0.787	3873.8563	61.6586
MultilayerPerceptron	0.7641	4764.3788	75.8327	0.7762	4360.284	69.4008
SMOreg	0.8191	3812.9653	60.6894	0.8206	3695.2361	58.8155
IBk	0.463	5517.129	87.8139	0.5089	5206.0484	82.8625
KStar	0.7336	4618.2302	73.5065	0.6642	4841.8553	77.0658
Random Tree	0.569	5686.9672	90.5171	0.5882	4303.6581	68.4995
M5p	**0.8175**	**3718.2692**	**59.1822**	**0.8472**	**3443.7698**	**54.8131**

The best performance was obtained by SMOReg before the feature selection. The performance measures, which were 0.8191 Correlation Coefficient, 3812.9653 MAE, 60.6894 RAE (%). With the application of feature selection with Relief Att.Evaluation and Ranker, the algorithm with best result is revised as M5P. Performance measures are saved as 0.8472 Correlation Coefficient, 3443.7698 MAE, 54.8131 RAE (%). It has been observed that higher results have been achieved even if it is with low rate.

The lowest performance was obtained with IBk before selection of feature, which saved performance measures as 0.463 Correlation Coefficient, 5517.129 MAE, 87.8139 RAE (%). After feature selection, while IBk performance measures observed as 0.5089 Correlation Coefficient, 5206.0484 MAE, 82.8625 RAE (%), and the lowest performing algorithm is kept as IBk.

In addition, application of Relief Att.Evaluation by selecting Ranker Search, 8 out of 9 algorithms showed the improvement and the highest improvement of this method was recorded as 4% improvement in the correlation coefficient on IBk algorithm.

4.2.3.6.2 Second Experiment-By Removing All the Tagged Feature with Negative Coefficient

The final sequence was obtained as: Function points, Duration – Süre, Staff tool skills, Development Env, adequacy, Telon use, Methods use, Staff team skills, Tools use, Software logical complexity, Staff availability, Time, Year, Installation requirements, Customer participation, Quality requirements, Where developed, Standards use, Database, User interface, Staff analysis skills, # of development languages, Efficiency requirements, Requirements volatility, Hardware platform, Application type, Staff application. In order not to insert the ineffective features into the model as unnecessary inputs, the last 13 features Customer participation, Quality requirements, where developed, Standards use, Database, User interface, Staff analysis skills, # of development languages, Efficiency requirements, Requirements volatility, Hardware platform, Application type, Staff application were removed from the set.

The results of the algorithms obtained without feature selection and the performance outputs of the model created with feature subset after feature selection is given in Table 4.23.

Table 4.23: Performance Measures of Models with Maxwell Original Feature and 14 Selected Feature Set by Relief Att.Evaluation + Ranker

Machine Learning Algorithms	**Maxwell Original Dataset (27 Feature)**			**Feature Selection with Relief Att.Evaluation + Ranker (14 Feature)**		
	Correlation Coefficient	**MAE**	**RAE (%)**	**Correlation Coefficient**	**MAE**	**RAE (%)**
LinearRegression	0.8085	4157.5897	66.1746	0.8359	3515.087	55.9482
RandomForest	0.7612	3998.2174	63.638	0.8102	3479.8312	55.387
Bagging	0.7711	3949.1671	62.8573	0.795	3719.2772	59.1982
MultilayerPerceptron	0.7641	4764.3788	75.8327	0.7346	5356.9926	85.265
SMOreg	**0.8191**	**3812.9653**	**60.6894**	**0.838**	**3702.8557**	**58.9368**
IBk	0.463	5517.129	87.8139	0.5988	5502.4032	87.5795
KStar	0.7336	4618.2302	73.5065	0.7651	4005.7376	63.7577
Random Tree	0.569	5686.9672	90.5171	0.7425	5064.0022	80.6016
M5p	0.8175	3718.2692	59.1822	0.8322	3476.3101	55.331

The best performance was obtained by SMOReg before the feature selection. The performance measures, which were 0.8191 Correlation Coefficient, 3812.9653 MAE, 60.6894 RAE (%). With the application of feature selection with Corr. Att.Evaluation and Ranker, the algorithm with best result is kept as SMOreg. Performance measures are saved as 0.838 Correlation Coefficient, 3702.8557 MAE,

58.9368 RAE (%). It has been observed that higher results have been achieved even if it is with low rate.

The lowest performance was obtained with IBk before selection of feature, which saved performance measures as 0.463 Correlation Coefficient, 5517.129 MAE, 87.8139 RAE (%). After feature selection, while IBk performance measures observed as 0.5988 Correlation Coefficient, 5502.4032 MAE, 87.5795 RAE (%), and the lowest performing algorithm is kept as IBk.

In addition, application of Relief Att.Evaluation by selecting Ranker Search, 8 out of 9 algorithms showed the improvement and the highest improvement of this method was recorded as 18% improvement in the correlation coefficient on RandomTree algorithm.

4.2.4 Kemerer Dataset Cost Estimation Results with Feature Selection Methods

4.2.4.1 CfsSubset Evaluation and Random Search Method Results

For the Kemerer dataset, attribute selection was applied by selecting the evaluation criterion CfsSubsetEval and the search criterion RandomSearch from the SelectionAttributes function of WEKA. The number of attributes, which was 8 before the process, was reduced to 4 by applying the method, by selecting the attributes named ID, hardware, KSLOC, Adjusted Function Points. With the addition of the dependent variable, Effort Man Months, 5 attributes were determined as the model input as the most effective attributes.

The results of the algorithms obtained without feature selection and the performance outputs of the model created with feature subset after feature selection is given in Table 4.24.

Table 4.24: Performance Measures of Models with Kemerer Original Feature and 14 Selected Feature Set by RandomSearch

Machine Learning Algorithms	Kemerer Original Dataset (8 Feature)			Feature Selection with CFS+ RandomSearch (5 Feature)		
	Correlation Coefficient	MAE	RAE (%)	Correlation Coefficient	MAE	RAE (%)
LinearRegression	0.3692	173.2407	107.6474	0.3684	161.7829	100.5279
RandomForest	0.3532	129.0567	80.1926	0.2857	149.997	93.2044
Bagging	0.1277	185.4463	115.2317	0.1168	182.9247	113.6648
MultilayerPerceptron	0.3511	129.4589	80.4425	0.3134	140.4294	87.2593
SMOreg	**0.5737**	**114.3301**	**71.0419**	**0.6795**	**98.1538**	**60.9903**
IBk	0.4665	142.054	88.2688	0.2849	193.1747	120.0339
KStar	0.5589	134.6747	83.6835	0.6034	137.9592	85.7244
Random Tree	-0.0271	250.9131	155.9111	0.1189	194.6347	120.9411
M5p	0.3291	176.3236	109.5631	0.348	180.1463	111.9384

The best performance was obtained by SMOreg before and after the feature selection was applied. The performance measures, which were 0.5737 Correlation Coefficient, 114.3301 MAE, 71.0419 RAE (%) before the feature selection was applied, improved to 0.6795 Correlation Coefficient, 98.1538MAE, 60.9903 RAE (%) after the feature selection was applied. The improvement of correlation coefficient has been gained as 16% on Kemerer dataset for best performance measure.

The lowest performance was obtained with RandomTree before selection of feature, which saved performance measures as -0.0271 Correlation Coefficient, 250.9131 MAE, 155.9111RAE (%). After feature selection, while RandomTree performance measures is gaining improvement, the lowest performing algorithm is updated as Bagging with 0.1168 Correlation Coefficient, 182.9247 MAE, 113.6648 RAE (%).

By evaluation of the results were evaluated in the Maxwell dataset in general, although feature selection increased the highest performance obtained before the application with a high percentage, 4 of the 9 algorithms were improved and 5 of them fell backwards. The highest improvement of this method was recorded as 10% improvement in the correlation coefficient on SMOreg algorithm.

4.2.4.2 CfsSubset Evaluation and Particle Swarm Optimization (PSO) Method Results

Attribute selection was applied by selecting the evaluation criterion CfsSubsetEval and the search criterion RandomSearch from the SelectionAttributes function of WEKA. The number of attributes, which was 8 before the process, was

reduced to 4 by applying the method, by selecting the attributes named ID, Language, KSLOC, Adjusted Function Points. With the addition of the dependent variable, Effort provided by application Use, 5 attributes were determined as the model input as the most effective attributes.

The results of the algorithms obtained without feature selection and the performance outputs of the model created with feature subset after feature selection is given in Table 4.25.

Table 4.25: Performance Measures of Models with Kemerer Original Feature and Selected Feature Set by CFS+ PSO

Machine Learning Algorithms	**Kemerer Original Dataset (8 Feature)**			**Feature Selection with CFS+ PSO (5 Feature)**		
	Correlation Coefficient	**MAE**	**RAE (%)**	**Correlation Coefficient**	**MAE**	**RAE (%)**
LinearRegression	0.3692	173.2407	107.6474	0.3425	190.2161	118.1955
RandomForest	0.3532	129.0567	80.1926	0.2925	143.9352	89.4377
Bagging	0.1277	185.4463	115.2317	0.117	180.8072	112.3491
MultilayerPerceptron	0.3511	129.4589	80.4425	0.3277	150.4623	93.4935
SMOreg	**0.5737**	**114.3301**	**71.0419**	**0.6946**	**96.4073**	**59.9051**
IBk	0.4665	142.054	88.2688	0.336	160.028	99.4374
KStar	0.5589	134.6747	83.6835	0.6219	124.3199	77.2492
Random Tree	-0.0271	250.9131	155.9111	0.329	163.9313	101.8628
M5p	0.3291	176.3236	109.5631	0.3385	188.2263	116.9591

The best performance was obtained by SMOreg before and after the feature selection was applied. The performance measures, which were 0.5737 Correlation Coefficient, 114.3301 MAE, 71.0419 RAE (%) before the feature selection was applied, improved to 0.6946 Correlation Coefficient, 96.4073 MAE, 59.9051 RAE (%) after the feature selection was applied.

Similarly, the lowest performance was obtained with Bagging before and after feature selection

Although there was no improvement in the best result compared to the original dataset by applying feature selection in the Kemerer dataset, by applying this method, 4 out of 9 algorithms showed improvement, 5 of them fell backwards, and the highest improvement of this method was recorded as 34% improvement on the correlation coefficient of RandomTree algorithm.

4.2.4.3 CfsSubset Evaluation and Genetic Algorithm (GA) Method Results

Attribute selection was applied by selecting the evaluation criterion CfsSubsetEval and the search criterion RandomSearch from the SelectionAttributes function of WEKA. The number of attributes, which was 8 before the process, was reduced to 4 by applying the method, by selecting the attributes named ID, Development Language, KSLOC, Adjusted Function Points. With the addition of the dependent variable, Effort provided by application Use, 5 attributes were determined as the model input as the most effective attributes.

The results of the algorithms obtained without feature selection and the performance outputs of the model created with feature subset after feature selection is given in Table 4.26.

Table 4.26: Performance Measures of Models with Kemerer Original Feature and Selected Feature Set by CFS+ GA

Machine Learning Algorithms	**Kemerer Original Dataset (8 Feature)**			**Feature Selection with CFS+ GA (5 Feature)**		
	Correlation Coefficient	**MAE**	**RAE (%)**	**Correlation Coefficient**	**MAE**	**RAE (%)**
LinearRegression	0.3692	173.2407	107.6474	0.3425	190.2161	118.1955
RandomForest	0.3532	129.0567	80.1926	0.2925	143.9352	89.4377
Bagging	0.1277	185.4463	115.2317	0.117	180.8072	112.3491
MultilayerPerceptron	0.3511	129.4589	80.4425	0.3277	150.4623	93.4935
SMOreg	**0.5737**	**114.3301**	**71.0419**	**0.6946**	**96.4073**	**59.9051**
IBk	0.4665	142.054	88.2688	0.336	160.028	99.4374
KStar	0.5589	134.6747	83.6835	0.6219	124.3199	77.2492
Random Tree	-0.0271	250.9131	155.9111	0.329	163.9313	101.8628
M5p	0.3291	176.3236	109.5631	0.3385	188.2263	116.9591

On Kemerer dataset, the best performance was obtained by SMOreg before and after the feature selection was applied. The performance measures, which were 0.5737 Correlation Coefficient, 114.3301 MAE, 71.0419 RAE (%) before the feature selection was applied. The performance measures improved to 0.6946 Correlation Coefficient, 96.4073 MAE, 59.9051 RAE (%) after the feature selection was applied.

Similarly, the lowest performance was obtained with Bagging before and after feature selection.

Although there was no improvement in the best result compared to the original dataset by applying feature selection in the Kemerer dataset, by applying this method, 4 out of 9 algorithms showed improvement, 5 of them fell backwards, and the highest

improvement of this method was recorded as 34% improvement on the correlation coefficient of RandomTree algorithm.

4.2.4.4 ClassifierAtt. Evaluation and Ranker Search Method Results

Attribute selection was applied by selecting the evaluation criterion ClassifierAttEval and the search criterion Ranker from the SelectionAttributes function of WEKA. The number of attributes was 8 before the process. The ranker search method, sorts the attributes in order of impact on the model when implemented with ClassifierAttEval. The resulting sequence was obtained as: RAWFP, Hardware, Language, Duration, AdjFP, KSLOC, ID. In order not to insert the most ineffective features into the model as unnecessary inputs, the last 3 features AdjFP, KSLOC, ID were removed from the set.

The results of the algorithms obtained without feature selection and the performance outputs of the model created with feature subset after feature selection is given in Table 4.27.

Table 4.27: Performance Measures of Models with Kemerer Original Feature and Selected Feature Set by ClassifierAttEvaluation + Ranker

Machine Learning Algorithms	**Kemerer Original Dataset (8 Feature)**			**Feature Selection with ClassifierAttEvaluation + Ranker (5 Feature)**		
	Correlation Coefficient	**MAE**	**RAE (%)**	**Correlation Coefficient**	**MAE**	**RAE (%)**
LinearRegression	0.3692	173.2407	107.6474	0.4009	158.8259	98.6904
RandomForest	0.3532	129.0567	80.1926	0.4872	120.4307	74.8326
Bagging	0.1277	185.4463	115.2317	0.1766	140.8899	87.5454
MultilayerPerceptron	0.3511	129.4589	80.4425	0.3519	134.8493	83.792
SMOreg	**0.5737**	**114.3301**	**71.0419**	**0.5405**	**112.9512**	**70.1851**
IBk	0.4665	142.054	88.2688	0.4626	145.174	90.2075
KStar	0.5589	134.6747	83.6835	0.418	136.8207	85.017
Random Tree	-0.0271	250.9131	155.9111	0.3455	0.7869	85.9987
M5p	0.3291	176.3236	109.5631	0.4084	149.358	92.8073

Before feature selection, the best performance was obtained with SMOreg in the Kemerer dataset, similarly, with the application of feature selection with ClassifierAttEval and Ranker, the best result among the algorithms again SMOreg with 0.5405 Correlation Coefficient, 112.9512 MAE, 70.1851 RAE (%) performance measurements. However, it was observed that the correlation coefficient obtained in the original dataset could not be reached.

The lowest performance was obtained with RandomTree in the Kemerer dataset, however, with the application of feature selection with ClassifierAttEval and Ranker, the worst result among the algorithms is revised as Bagging.

In addition, with application of ClassifierAttEval evaluation and Ranker Search, 6 out of 9 algorithms showed the improvement and high improvements are obtained for RandomTree, RandomForest and Bagging, as 36%, 13%, 5% respectively.

4.2.4.5 Corr. Att. Evaluation and Ranker Search Method Results

Attribute selection was applied by selecting the evaluation criterion Corr.Att.Evaluation and the search criterion Ranker from the SelectionAttributes function of WEKA. The number of attributes was 8 before the process. The ranker search method, sorts the attributes in order of impact and gives coefficient to the feature in accordance with rank value to take with coefficient the impact of feature, when implemented with Corr.Att.Evaluation.

The final sequence was obtained as: AdjFP, RAWFP, KSLOC, Duration, Hardware, Language, ID. In order not to insert the most ineffective features into the model as unnecessary inputs, the last 3 features Hardware, Language, ID were removed from the set.

The results of the algorithms obtained without feature selection and the performance outputs of the model created with feature subset after feature selection is given in Table 4.28.

Table 4.28: Performance Measures of Models with Kemerer Original Feature and Selected Feature Set by Corr.Att.Evaluation + Ranker

Machine Learning Algorithms	**Kemerer Original Dataset (8 Feature)**			**Feature Selection with Corr.Att.Evaluation + Ranker (5 Feature)**		
	Correlation Coefficient	**MAE**	**RAE (%)**	**Correlation Coefficient**	**MAE**	**RAE (%)**
LinearRegression	0.3692	173.2407	107.6474	0.354	172.4716	107.1695
RandomForest	0.3532	129.0567	80.1926	0.4692	112.564	69.9445
Bagging	0.1277	185.4463	115.2317	0.1611	138.01	85.756
MultilayerPerceptron	0.3511	129.4589	80.4425	0.2937	155.0315	96.3327
SMOreg	**0.5737**	**114.3301**	**71.0419**	**0.7171**	**103.4371**	**64.2732**
IBk	0.4665	142.054	88.2688	0.5812	134.974	83.8695
KStar	0.5589	134.6747	83.6835	0.2984	128.1335	79.6189
Random Tree	-0.0271	250.9131	155.9111	0.6658	126.7154	78.7378
M5p	0.3291	176.3236	109.5631	0.3397	173.3476	107.7139

On Kemerer dataset, the best performance was obtained by SMOreg before and after the feature selection was applied. The performance measures, which were 0.5737 Correlation Coefficient, 114.3301 MAE, 71.0419 RAE (%) before the feature selection and the best result among the algorithms again SMOreg with 0.7171 Correlation Coefficient, 103.4371 MAE, 64.2732 RAE (%) performance measurements. It has been observed that higher results have been achieved. Based on this improvement, it has succeeded in being the method that can achieve higher results compared to the original feature set.

The lowest performance was obtained with RandomTree before feature selection with -0.0271 Correlation Coefficient, 250.9131 MAE, 155.911 RAE (%), by application of feature selection, performance measures gained high improvement and noted as 0. 6658 Correlation Coefficient, 126.7154 MAE, 78.7378 RAE (%). So new lowest measurement is saved as Bagging.

In addition, application of Corr.Att.Evaluation by selecting Ranker Search, 6 out of 9 algorithms showed the improvement and the highest improvement of this method was recorded as 68% for RandomTree.

4.2.4.6 Relief Att. Evaluation and Ranker Search Method Results

Attribute selection was applied by selecting the evaluation criterion Relief Att.Evaluation and the search criterion Ranker from the SelectionAttributes function of WEKA. The number of attributes was 8 before the process. The ranker search method, sorts the attributes in order of impact and gives coefficient to the feature in accordance with rank value to take with coefficient the impact of feature, when implemented with Relief Att.Evaluation.

The final sequence was obtained as: KSLOC, AdjFP, RAWFP, Hardware, ID, Duration, Language. In order not to insert the most ineffective features into the model as unnecessary inputs, the last 3 features ID, Duration, Language were removed from the set.

The results of the algorithms obtained without feature selection and the performance outputs of the model created with feature subset after feature selection is given in Table 4.29.

Table 4.29: Performance Measures of Models with Kemerer Original Feature and Selected Feature Set by Relief Att.Evaluation + Ranker

Machine Learning Algorithms	Kemerer Original Dataset (8 Feature)			Feature Selection with Relief Att.Evaluation + Ranker (5 Feature)		
	Correlation Coefficient	MAE	RAE (%)	Correlation Coefficient	MAE	RAE (%)
LinearRegression	0.3692	173.2407	107.6474	0.3803	173.5323	107.8286
RandomForest	0.3532	129.0567	80.1926	0.302	139.0277	86.3884
Bagging	0.1277	185.4463	115.2317	0.1408	178.4307	110.8724
MultilayerPerceptron	**0.3511**	**129.4589**	**80.4425**	**0.6295**	**119.1639**	**74.0455**
SMOreg	0.5737	114.3301	71.0419	0.627	126.2262	78.4338
IBk	0.4665	142.054	88.2688	0.4654	146.6073	91.0981
KStar	0.5589	134.6747	83.6835	0.6295	119.1639	74.0455
Random Tree	-0.0271	250.9131	155.9111	0.2338	179.156	111.323
M5p	0.3291	176.3236	109.5631	0.4148	148.5519	92.3064

Before feature selection, the best performance was obtained with SMOreg in the Kemerer dataset which the performance measures were 0.5737 Correlation Coefficient, 114.3301 MAE, 71.0419 RAE (%). However, with the application of feature selection with Relief Att.Evaluation and Ranker, the best result among the algorithms is MultilayerPerceptron and KStar at the same time with 0.6295 Correlation Coefficient, 119.1639 MAE, 74.0455 RAE (%) performance measurements

The lowest performance was obtained with RandomTree before feature selection with -0.0271 Correlation Coefficient, 250.9131 MAE, 155.911 RAE (%), by application of feature selection, performance measures gained high improvement and new lowest algorithm result is noted as Bagging with 0.1408 Correlation Coefficient, 178.4307 MAAE, 110.8724 RAE (%).

In addition, application of Relief Att.Evaluation by selecting Ranker Search, 7 out of 9 algorithms showed the improvement and high improvements are obtained for MultilayerPerceptron, RandomTree as 27%, 25% respectively.

4.3 ANALYSIS OF FINDINGS

In this section, an extensive examination is provided of developed models via machine learning algorithms by handling original and selected feature sets for Software Cost Estimations. The detailed analyses of the investigated studies are presented in tables, showcasing the results obtained. The comparison of existing studies is conducted based on several factors, including the software cost estimation method employed, the datasets utilized, feature selection technique, selected feature number, and the evaluation criteria employed.

Until this section, different feature selection techniques have been applied separately for each dataset, and it has been tried to converge better results than the original dataset. In this section, the results obtained are evaluated together:

- The highest achievable success rates for the dataset,
- Algorithms and techniques that converge to the highest performance rate,
- Techniques that tend to show higher performance compared to the original dataset,
- The lowest success rates for the dataset,

By emphasizing it, it is aimed to add a study to the literature that will provide input on which machine learning methods can provide high-performance accuracy estimation in software cost estimation. The performance evaluation criteria employed in these analyses and comparisons are the Correlation Coefficient, Mean Absolute Error (MAE), and Relative Absolute Error (RAE).

The outcomes of the models constructed using the Finnish dataset are showcased in Table 4.30. By considering the model outputs, it is seen that the highest performance measurement is obtained with KStar, and the result that is very close to the highest is obtained with RandomForest by using ClassifierAttEval + Ranker and PSO, GA method, respectively. In addition, the best result is achieved with ClassifierAttEval + Ranker feature selection method thanks to model development with 6 feature selection (differently hardware attribute is included) instead of 5. For that reason, PSO and GA models, which are so close to the best result with 5 features, should not be ignored as more robust against overfitting. On the other hand, IBk has the lowest performance compared to other algorithms.

Examining the model outputs, it is seen that the model outputs developed by feature selection in 8 out of 9 algorithms achieve better results than the model output created using the original dataset. In addition, it has been observed that PSO and GA feature selection techniques outperform other methods in capturing the highest value of the algorithm. The best results which could be achieved with 6 of 9 algorithms were obtained with PSO and GA.

Table 4.30: The Outcomes of The Models Constructed Using the Finnish Dataset

Finnish Dataset										
FEATURE SELECTION TECHNIQUES		ALGORITHMS								
		Linear Regression	Random Forest	Bagging	Multilayer Perceptron	SMOreg	IBk	Kstar	Random Tree	M5p
Original Feature Set (9 Feature)	Correlation Coefficient	**0.9607**	0.9818	0.9801	0.9575	0.962	0.7697	0.9889	0.9029	0.9692
	MAE	0.254	0.1649	0.1747	0.2297	0.2341	0.539	0.1344	0.3654	0.2146
	RAE (%)	24.8268	16.1147	17.074	22.4464	22.8759	52.6711	13.1344	35.7136	20.9732
CFS + RandomSearch (6 Feature)	Correlation Coefficient	**0.9607**	0.989	0.9854	0.9724	0.9654	0.931	0.9916	0.9563	0.9715
	MAE	0.254	0.1344	0.1548	0.1595	0.2311	0.3319	0.1208	0.2403	0.2158
	RAE (%)	24.8268	13.1385	15.1257	15.5839	22.5861	32.4397	11.8098	23.4836	21.0941
CFS+ PSO (5 Feature)	Correlation Coefficient	**0.9607**	**0.9942**	**0.9857**	**0.9853**	**0.9702**	**0.934**	**0.9923**	**0.9843**	0.9715
	MAE	0.254	0.0976	0.1532	0.1376	0.2213	0.3142	0.1127	0.1461	0.2158
	RAE (%)	24.8268	9.5354	14.967	13.4468	21.6292	30.7074	11.0156	14.2734	21.0941
CFS + GA (5 Feature)	Correlation Coefficient	**0.9607**	**0.9942**	**0.9857**	**0.9853**	**0.9702**	**0.934**	**0.9923**	**0.9843**	0.9715
	MAE	0.254	0.0976	0.1532	0.1376	0.2213	0.3142	0.1127	0.1461	0.2158
	RAE (%)	24.8268	9.5354	14.967	13.4468	21.6292	30.7074	11.0156	14.2734	21.0941
ClassifierAttEval + Ranker (6 Feature)	Correlation Coefficient	**0.9658**	0.9892	0.9807	0.9656	0.9629	0.7421	**0.9948**	0.9642	**0.9783**
	MAE	0.235	0.1378	0.1692	0.2006	0.2445	0.5756	0.0873	0.2274	0.1868
	RAE (%)	22.9611	13.4643	16.5387	19.6052	23.8935	56.2528	8.5274	22.2251	18.2566
Corr. Att.Evaluation + Ranker (6 Feature)	Correlation Coefficient	**0.9607**	0.9901	0.9845	0.9833	0.9686	0.9158	0.9912	0.9663	**0.9729**
	MAE	0.254	0.1298	0.1587	0.1535	0.2221	0.3518	0.1284	0.2324	0.2006
	RAE (%)	24.8268	12.6897	15.5069	15.0001	21.7002	34.3834	12.546	22.7083	19.6081
Relief. Att.Evaluation + Ranker (6 Feature)	Correlation Coefficient	**0.9607**	0.9907	0.9852	0.9647	0.9653	0.931	0.9916	0.9705	**0.9729**
	MAE	0.254	0.118	0.1544	0.1809	0.2314	0.3319	0.1208	0.2041	0.2006
	RAE (%)	24.8268	11.5345	15.0929	17.6834	22.6141	32.4397	11.8098	19.9438	19.6081

The outcomes of the models constructed using the China dataset are showcased in Table 4.31. By considering the model outputs, it is seen that the highest performance measurement is obtained with Multiexciton, and the very close result obtained with SMOReg by using Relief Att.Evaluation+Ranker and Corr.Att.Evaluation+Ranker attribute selection method, respectively, thanks to model development with 16 feature selection instead of 7/9/10 for both. However, considering that overfitting may occur in training with too many attributes, it is recommended to consider that the RandomSearch technique with 7 attributes also yields a very high result of 0.9866 Correlation Coefficient with SMOreg. On the other hand, RandomTree has the lowest performance and IBk has a very low result compared to other algorithms.

Examining the model outputs, it is seen that the model outputs developed by Relief Att.Evaluation+Ranker feature selection in 6 out of 9 algorithms achieve better results than the model output created using the original dataset. In addition, the RandomSearch applied model also obtained better results in 5 of the 9 algorithms compared to the model output with the original dataset. Also, the best results which

could be achieved with algorithms are captured 3 times by Relief Att.Evaluation + Ranker.

Table 4.31: The Outcomes of The Models Constructed Using the China Dataset

China Dataset									
FEATURE SELECTION TECHNIQUES		ALGORITHMS							
		Linear Regression	Random Forest	Bagging	Multilayer Perceptron	SMOreg	IBk	Kstar	Random Tree	M5p
Original Feature Set (19 Feature)	Correlation Coefficient	**0.9889**	0.9591	0.9605	0.9733	0.9897	0.8918	0.9646	**0.9283**	**0.9842**
	MAE	362.939	557.7718	511.9898	461.3901	270.4561	1571.1824	628.608	943.0361	392.7912
	RAE (%)	9.809	15.0747	13.8374	12.4698	7.3095	42.4638	16.9892	25.4871	10.6158
CFS + RandomSearch (7 Feature)	Correlation Coefficient	0.986	**0.9681**	0.9625	0.9776	0.9866	**0.9362**	0.9648	**0.9244**	0.9832
	MAE	365.1502	553.6869	508.1267	548.369	351.7668	1175.988	496.0903	850.8577	372.001
	RAE (%)	9.8688	14.9643	13.733	14.8206	9.5071	31.783	13.4077	22.9958	10.0539
CFS+ PSO (9 Feature)	Correlation Coefficient	0.986	0.9602	0.9597	0.9767	0.9853	**0.9081**	**0.9717**	0.9155	0.9832
	MAE	395.7794	583.8073	519.3041	514.2803	360.9192	1500.1563	528.6757	872.0081	391.7027
	RAE (%)	10.6966	15.7784	14.035	13.8993	9.7544	40.5442	14.2883	23.5675	10.5864
CFS + GA (10 Feature)	Correlation Coefficient	0.9859	0.9584	0.9596	0.9746	0.9847	0.9076	**0.9726**	0.8726	0.984
	MAE	411.7442	578.3479	519.546	497.3849	358.5216	1444.8758	543.4151	1056.6915	401.4127
	RAE (%)	11.1281	15.6308	14.0416	13.4426	9.6896	39.0501	14.6867	28.5588	10.8488
ClassifierAttEval + Ranker (15 Feature)	Correlation Coefficient	0.9872	0.9583	**0.9642**	0.965	0.9887	0.847	0.9694	0.8617	0.9835
	MAE	413.9045	586.4616	492.293	549.6564	304.4746	1343.4469	596.5132	1073.5312	390.3778
	RAE (%)	11.1865	15.8501	13.305	14.8554	8.2289	36.3089	16.1218	29.0139	10.5506
Corr. Att.Evaluation + Ranker (16 Feature)	Correlation Coefficient	**0.989**	0.9623	0.9618	**0.9912**	**0.9898**	0.8396	0.9638	0.8409	0.9835
	MAE	362.2063	543.9122	503.6359	406.1195	270.2385	1418.9499	619.7006	1142.3578	386.561
	RAE (%)	9.7892	14.7001	13.6116	10.976	7.3036	38.3495	16.7484	30.8741	10.4474
Relief. Att.Evaluation + Ranker (16 Feature)	Correlation Coefficient	0.9886	**0.9627**	**0.9629**	**0.9914**	**0.9898**	0.887	0.965	0.8098	**0.9852**
	MAE	366.1417	546.8556	501.3717	370.1846	269.3637	1581.3467	617.641	1263.9482	378.51
	RAE (%)	9.8956	14.7797	13.5504	10.0048	7.28	1581.3467	16.6928	34.1603	10.2299

The outcomes of the models constructed using the Maxwell dataset are showcased in Table 4.32. By considering the model outputs, it is seen that the highest performance measurement is obtained with KStar, and the very close result obtained with LinearRegression by using GA for both thanks to model development with 20 feature selection. On the other hand, However, considering that overfitting may occur in training with too many attributes, it is recommended to consider that the PSO technique with 9 attributes also yields a very high result of 0.85 Correlation Coefficient with KStar. RandomTree has the lowest performance and IBk has a very low result compared to other algorithms.

Examining the model outputs, it is seen that the model outputs developed by PSO and Relief Att.Evaluation+Ranker feature selection in 8 out of 9 algorithms achieve better results than the model output created using the original dataset. In addition to this, the best results which could be achieved with algorithms are captured 4 times by Relief Att.Evaluation+Ranker.

Table 4.32: The Outcomes of The Models Constructed Using the Maxwell Dataset

Maxwell Dataset										
ALGORITHMS		**Linear Regression**	**Random Forest**	**Bagging**	**Multilayer Perceptron**	**SMOreg**	**Ibk**	**Kstar**	**Random Tree**	**M5p**
Original Feature Set (27 Feature)	**Correlation Coefficient**	0.8085	0.7612	0.7711	**0.7641**	0.8191	0.463	0.7336	0.569	0.8175
	MAE	4157.5897	3998.2174	3949.1671	4764.3788	3812.9653	5517.129	4618.2302	5686.9672	3718.2692
	RAE (%)	66.1746	63.638	62.8573	75.8327	60.6894	87.8139	73.5065	90.5171	59.1822
CFS + RandomSearch (16 Feature)	**Correlation Coefficient**	**0.8354**	0.7624	**0.7753**	0.7215	0.8091	**0.7427**	**0.8174**	0.6147	0.8173
	MAE	3610.4878	3768.4553	3835.1733	5389.5218	3520.7457	4725.7419	4154.1848	5068.6745	3677.0131
	RAE (%)	57.4666	59.9809	61.0429	85.7828	56.0383	75.2177	66.1204	80.676	58.5255
CFS+ PSO (9 Feature)	**Correlation Coefficient**	0.835	0.7916	0.7738	0.712	**0.8361**	**0.7432**	**0.85**	0.613	**0.834**
	MAE	3550.9565	3655.784	3840.6234	5175.8943	3188.8894	4848.2419	4040.6726	5151.1169	3654.1942
	RAE (%)	56.5191	58.1876	61.1296	82.3826	50.7562	77.1675	64.3137	81.9882	58.1623
CFS +GA (20 Feature)	**Correlation Coefficient**	**0.8544**	0.7621	0.7704	**0.816**	0.818	**0.7593**	**0.8596**	0.4398	0.8092
	MAE	3395.0666	3827.5684	3898.5336	4146.3094	3522.3771	4494.6774	4078.3244	5223.2222	3685.2814
	RAE (%)	54.0379	60.9218	62.0514	65.9951	56.0642	71.5399	64.913	83.1359	58.6571
ClassifierAttEval + Ranker (24 Feature)	**Correlation Coefficient**	0.8282	**0.8015**	0.7589	0.6961	0.8281	0.4904	0.7209	**0.6184**	**0.8515**
	MAE	4164.5366	3654.0125	4041.8054	5487.8816	3639.1298	5397.3871	4539.8698	6120.3704	3447.3519
	RAE (%)	66.2852	58.1594	64.3317	87.3483	57.9225	85.908	72.2592	97.4154	54.8701
Corr. Att.Evaluation +Ranker (24 4Feature)	**Correlation Coefficient**	0.8163	**0.7932**	**0.791**	0.6801	**0.8336**	0.4487	0.7315	**0.6913**	0.8247
	MAE	3937.4256	3718.2019	3847.9297	6143.2445	3728.4524	5720.7419	4558.8152	5393.8065	3643.2708
	RAE (%)	62.6704	59.1811	61.2459	97.7795	59.3442	91.0547	72.5608	85.851	57.9884
Relief. Att.Evaluation + Ranker (14 Feature)	**Correlation Coefficient**	**0.8359**	**0.8102**	**0.795**	**0.7346**	**0.838**	0.5988	0.7651	**0.7425**	**0.8322**
	MAE	3515.087	3479.8312	3719.2772	5356.9926	3702.8557	5502.4032	4005.7376	5064.0022	3476.3101
	RAE (%)	55.9482	55.387	59.1982	85.265	58.9368	87.5795	63.7577	80.6016	55.331

The outcomes of the models constructed using the Kemerer dataset are showcased in Table 4.33. By considering the model outputs, it is seen that the highest performance measurement and the very close result observed are obtained with SMOreg by using Corr.Att.Evaluation+Ranker and PSO, GA, respectively. On the other hand, RandomTree has the lowest performance and Bagging has a very low result compared to other algorithms.

Examining the model outputs, it is seen that the model outputs developed by Relief Att.Evaluation+Ranker feature selection in 7 out of 9 algorithms achieve better results than the model output created using the original dataset. Despite this, the best results which could be achieved with algorithms are captured 3 times by Corr.AttEvaluation+Ranker.

Table 4.33: The Outcomes of The Models Constructed Using the Kemerer Dataset

Kemerer Dataset										
FEATURE SELECTION TECHNIQUES		ALGORITHMS								
		Linear Regression	Random Forest	Bagging	Multilayer Perceptron	SMOreg	IBk	Kstar	Random Tree	M5p
Original Feature Set (8 Feature)	Correlation Coefficient	0.3692	0.3532	0.1277	0.3511	0.5737	**0.4665**	0.5589	-0.0271	0.3291
	MAE	173.2407	129.0567	185.4463	129.4589	114.3301	142.054	134.6747	250.9131	176.3236
	RAE (%)	107.6474	80.1926	115.2317	80.4425	71.0419	88.2688	83.6835	155.9111	109.5631
CFS + RandomSearch (5 Feature)	Correlation Coefficient	0.3684	0.2857	0.1168	0.3134	0.6795	0.2849	0.6034	0.1189	0.348
	MAE	161.7829	149.997	182.9247	140.4294	98.1538	193.1747	137.9592	194.6347	180.1463
	RAE (%)	100.5279	93.2044	113.6648	87.2593	60.9903	120.0339	85.7244	120.9411	111.9384
CFS+ PSO (5 Feature)	Correlation Coefficient	0.3425	0.2925	0.117	0.3277	**0.6946**	0.336	**0.6219**	0.329	0.3385
	MAE	190.2161	143.9352	180.8072	150.4623	96.4073	160.028	124.3199	163.9313	188.2263
	RAE (%)	118.1955	89.4377	112.3491	93.4935	59.9051	99.4374	77.2492	101.8628	116.9591
CFS + GA (5 Feature) (5 Feature)	Correlation Coefficient	0.3425	0.2925	0.117	0.3277	**0.6946**	0.336	**0.6219**	0.329	0.3385
	MAE	190.2161	143.9352	180.8072	150.4623	96.4073	160.028	124.3199	163.9313	188.2263
	RAE (%)	118.1955	89.4377	112.3491	93.4935	59.9051	99.4374	77.2492	101.8628	116.9591
ClassifierAttEval + Ranker (5 Feature)	Correlation Coefficient	**0.4009**	**0.4872**	**0.1766**	**0.3519**	0.5405	0.4626	0.418	**0.3455**	**0.4084**
	MAE	158.8259	120.4307	140.8899	134.8493	112.9512	145.174	136.8207	0.7869	149.358
	RAE (%)	98.6904	74.8326	87.5454	83.792	70.1851	90.2075	85.017	85.9987	92.8073
Corr. Att.Evaluation + Ranker (5 Feature)	Correlation Coefficient	0.354	**0.4692**	**0.1611**	0.2937	**0.7171**	**0.5812**	0.2984	**0.6658**	0.3397
	MAE	172.4716	112.564	138.01	155.0315	103.4371	134.974	128.1335	126.7154	173.3476
	RAE (%)	107.1695	69.9445	85.756	96.3327	64.2732	83.8695	79.6189	78.7378	107.7139
Relief. Att.Evaluation + Ranker (5 Feature)	Correlation Coefficient	**0.3803**	0.302	0.1408	**0.6295**	0.627	0.4654	**0.6295**	0.2338	**0.4148**
	MAE	173.5323	139.0277	178.4307	119.1639	126.2262	146.6073	119.1639	179.156	148.5519
	RAE (%)	107.8286	86.3884	110.8724	74.0455	78.4338	91.0981	74.0455	111.323	92.3064

CHAPTER V
CONCLUSION

The main goal of a successful software project is to produce software that will meet the expectations of the customer with a predetermined budget at a predetermined time. The failure of many software projects is due to the fact that the estimates made at the initial planning stage were not correct. For this reason, it can be said that the most basic and first project management activity in the success of a software project is the appropriate and effective allocation of necessary resources. In other words, it is critical to determine the resources that will be needed in the realization of the relevant project by making the planning on the right basis. Cost is the crux of these resources and is highly dependent on the effort within the project. In this case, estimating the effort needed is important in determining the cost.

For the software cost estimation process, which is a very important step in software project management, traditionally and predominantly manual input and expert opinion are still used today. However, these techniques cannot handle to estimate the cost of large and complex software. Therefore, to improve the software cost estimation process has aimed in this thesis. For this purpose, a machine learning-based approach has been adopted to make the software cost estimation process faster, more consistent and repeatable accurately. By leveraging machine learning techniques, the goal is to automate and optimize the software cost estimation process, reducing the reliance on manual and subjective judgments.

During the development of a machine learning-driven approach, the Finnish, Kemerer, China, and Maxwell datasets provided in Title 3.1 were utilized for software cost estimation. Models were constructed using the algorithms outlined in Title 3.3, and the validation technique employed was 10-fold cross-validation.

In the first part of the study, titled 4.1, the models were constructed with original datasets and the performance of the models were showcased. Subsequently, in Title 4.2, the performances of the models constructed using optimized and efficient feature subsets obtained by hybrid feature selection approaches from the same datasets

were analyzed. These techniques were described in detail in Title 3.4, with their steps, areas of application, strengths, and weaknesses. In the development of the models with feature subsets, feature selection techniques such as CFS + RandomSearch, CFS + PSO, CFS + GA, ClassifierAttEval + Ranker, Corr.Att.Evaluation + Ranker, Relief Att.Evaluation + Ranker were utilized. As discussed in Title 3.5, the performance evaluation of the models was carried out by considering the Correlation Coefficient as well as several error metrics, including MAE, RMSE, RAE, and RRSE.

The most successful and unsuccessful results of the model outputs established with the original datasets obtained in the first and second stages of the study and the feature subsets obtained by the feature selection methods are presented in the tables. In order not to be affected by small deviations while examining the results, the values close to the best and the worst results with a small percentage difference were added to the table. In addition, due to its higher resistance to overfitting, models with less number of feature subsets and close to the best results are also included.

Finnish model performance measurements are presented in Table 5.1, China model performance measurements are presented in Table 5.2, Maxwell model performance measurements are presented in Table 5.3, Kemerer model performance measurements are presented in Table 5.4.

Table 5.1: Finnish Dataset 's Highest and Lowest Performance Measures Before and After Feature Selection

Finnish Dataset						
	Machine Learning Algortihm	**Number Of Selected Features**	**Feature Selection Technique**	**Correlation Coefficient**	**MAE**	**RAE (%)**
The Highest Result Without Feature Selection	Kstar	9	Original Feature Set	0.9889	0.1344	13.1344
The Lowest Result Without Feature Selection	IBk	9	Original Feature Set	0.7697	0.539	52.6711
The Hightest Results	Kstar	6	ClassifierAttEval+ Ranker	0.9948	0.0873	8.5274
	RandomForest	5	CFS + PSO	0.9942	0.0976	9.5354
	RandomForest	5	CFS+ GA	0.9942	0.0976	9.5354
The Lowest Results	Ibk	6	ClassifierAttEval+ Ranker	0.7421	0.5756	56.2528

Table 5.2: China Dataset 's Highest and Lowest Performance Measures Before and After Feature Selection

China Dataset						
	Machine Learning Algortihm	**Number Of Selected Features**	**Feature Selection Technique**	**Correlation Coefficient**	**MAE**	**RAE (%)**
The Highest Result Without Feature Selection	Linear Regression	19	Original Feature Set	0.9889	362.939	9.809
The Lowest Result Without Feature Selection	IBk	19	Original Feature Set	0.8918	1571.1824	42.4638
The Hightest Results	MultilayerPerceptron	16	Relief. Att.Evaluation + Ranker	0.9914	370.1846	10.0048
	MultilayerPerceptron	16	Corr. Att.Evaluation + Ranker	0.9912	406.1195	10.976
	SMOreg	16	Relief. Att.Evaluation + Ranker	0.9898	269.3637	7.28
	SMOreg	7	Random Search	0.9866	351.7668	9.5071
	LinearRegression	9	CFS + PSO	0.986	395.7794	10.6966
	LinearRegression	10	CFS+ GA	0.9859	411.7442	11.1281
The Lowest Results	IBk	16	Corr. Att.Evaluation + Ranker	0.8396	1418.9499	38.3495
	Random Tree	16	Relief. Att.Evaluation + Ranker	0.8098	1263.9482	34.1603

Table 5.3: Maxwell Dataset 's Highest and Lowest Performance Measures Before and After Feature Selection

Maxwell Dataset						
	Machine Learning Algortihm	**Number Of Selected Features**	**Feature Selection Technique**	**Correlation Coefficient**	**MAE**	**RAE (%)**
The Highest Result Without Feature Selection	SMOreg	27	Original Feature Set	0.8191	3812.9653	60.6894
The Lowest Result Without Feature Selection	Ibk	27	Original Feature Set	0.463	5517.129	87.8139
The Hightest Results	Kstar	20	CFS+ GA	0.8596	4078.3244	64.913
	LinearRegression	20	CFS+ GA	0.8544	3395.0666	54.0379
	Kstar	9	CFS + PSO	0.85	4040.6726	64.3137
The Lowest Results	Ibk	24	Corr. Att.Evaluation + Ranker	0.4487	5720.7419	91.0547
	Random Tree	20	CFS+ GA	0.4398	5223.2222	83.1359

Table 5.4: Kemerer Dataset 's Highest and Lowest Performance Measures Before and After Feature Selection

Kemerer Dataset						
	Machine Learning Algortihm	**Number Of Selected Features**	**Feature Selection Technique**	**Correlation Coefficient**	**MAE**	**RAE (%)**
The Highest Result Without Feature Selection	SMOreg	8	Original Feature Set	0.5737	114.3301	71.0419
The Lowest Result Without Feature Selection	Random Tree	8	Original Feature Set	-0.0271	250.9131	155.9111
The Hightest Results	SMOreg	5	Corr. Att.Evaluation + Ranker	0.7171	103.4371	64.2732
	SMOreg	5	CFS + PSO	0.6946	96.4073	59.9051
	SMOreg	5	CFS+ GA	0.6946	96.4073	59.9051
The Lowest Results	Bagging	5	CFS+ RandomSearch	0.1168	182.9247	113.6648
	Random Tree	5	CFS+ RandomSearch	0.1189	194.63	120.94

In the Finnish dataset, the KStar algorithm was found to be the most successful to achieve best estimation. The ClassifierAttEval and Ranker methods were utilized during the analysis. The efforts comparison for the actual and predicted by the model are depicted in Figure 5.1.

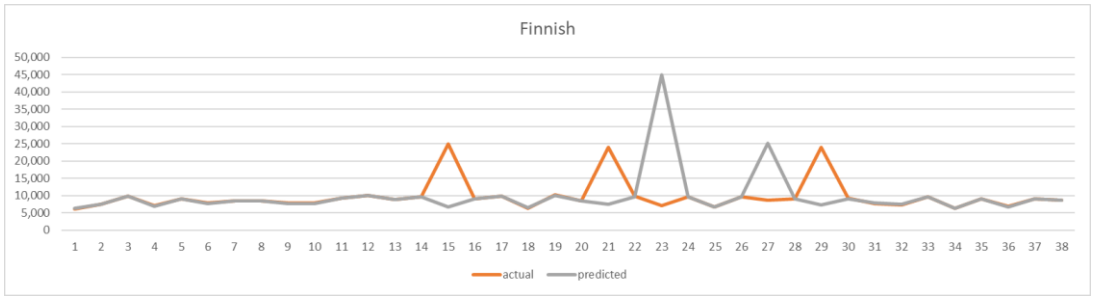


Figure 5.1: Comparison Graph for Actual and Predicted Effort of Finnish Dataset

In the China dataset, the Multilayer Perceptron algorithm was found to be the most successful to achieve best estimation. The Relief Att. Evaluation and Ranker methods were utilized during the analysis. The efforts comparison for the actual and predicted by the model are depicted in Figure 5.2.

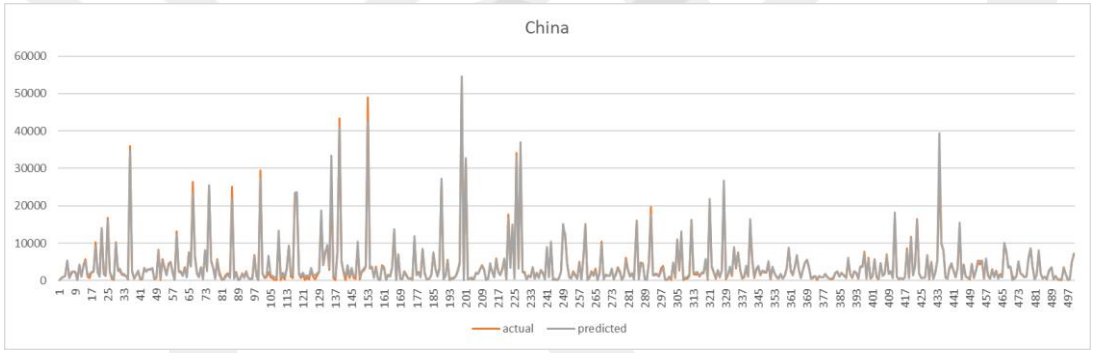


Figure 5.2: Comparison Graph for Actual and Predicted Effort of China Dataset

In the Kemerer dataset, the SMOreg algorithm was found to be the most successful to achieve best estimation. The Corr. Att. Evaluation and Ranker methods were utilized during the analysis. The efforts comparison for the actual and predicted by the model are depicted in Figure 5.3.

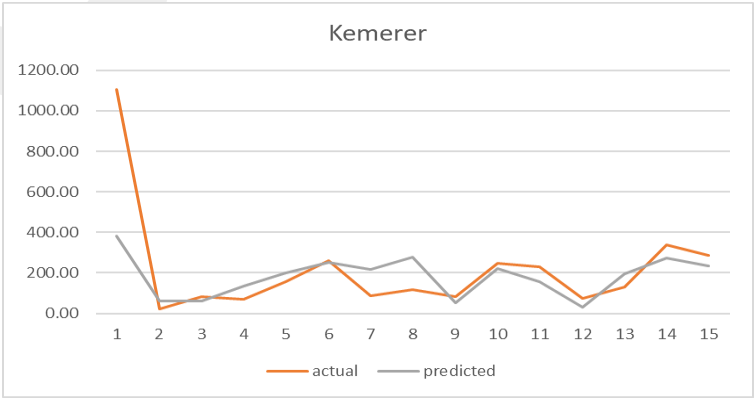


Figure 5.3: Comparison Graph for Actual and Predicted Effort of Kemerer Dataset

In the Maxwell dataset, the KStar algorithm was found to be the most successful to achieve best estimation. The CFS and Genetic Algorithm methods were utilized during the analysis. The efforts comparison for the actual and predicted by the model are depicted in Figure 5.4.

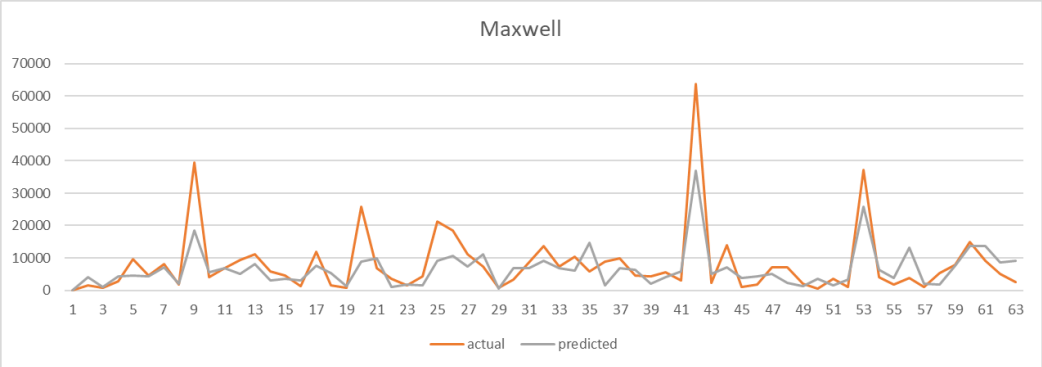


Figure 5.4: Comparison Graph for Actual and Predicted Effort of Maxwell Dataset

When the results are examined in detail, it is observed that the most successful and unsuccessful results of the models created with both the original data and a subset of the dataset are achieved with different algorithms and feature selection techniques. From this point of view, it should not be ignored that algorithms and techniques may lead to different results depending on the dataset characteristics.

It has been clearly seen that the models obtained by creating the most effective feature subsets in the models created with both the original dataset and different feature selection techniques of the most effective features achieve much more successful results than the models created with the original dataset. As a result, it can be said that advancing with feature selection in machine learning-based approaches in software cost estimation will be an effective method.

When the model results obtained by feature selection are examined, it is observed that Ranker-based, PSO and GA search methods generally achieve successful results in all four datasets. It has also been observed that in general, PSO and GA search methods yield good results, even if the feature subset is small. It is clearly seen in Table 5.5 that even if the best results were not achieved with the lowest feature set, close to the best results can also be obtained with a less numbered feature set. Figure 5.5, Figure 5.6, Figure 5.7, Figure 5.8 shows the noticeable effect of feature

selection methods on model accuracy and illustrates the increase in success of the model in accurately predicting values with the efficient selection of features.

Table 5.5: Model Performance Results with Feature Selection

Dataset	Orginal Feature Set	Model	FeatureSelection	Selected Feature Set	Correlation Coefficient
Finnish	9	KStar	CFS+ RandomSearch	5	0.9916
Finnish	9	RandomForest	CFS+ PSO	5	0.9942
Finnish	9	RandomForest	CFS+ GA	5	0.9942
Finnish	**9**	**KStar**	**ClassifierAttEval+ Ranker**	**6**	**0.9948**
Finnish	9	KStar	Corr. Att.Evaluation + Ranker	6	0.9912
Finnish	9	KStar	Relief. Att.Evaluation + Ranker	6	0.9916
China	19	SMOreg	CFS+ RandomSearch	7	0.9866
China	19	SMOreg	CFS+ PSO	9	0.9853
China	19	LinearRegression	CFS+ GA	10	0.9859
China	19	SMOreg	ClassifierAttEval+ Ranker	16	0.9887
China	19	MultilayerPerceptron	Corr. Att.Evaluation + Ranker	16	0.9912
China	**19**	**MultilayerPerceptron**	**Relief. Att.Evaluation + Ranker**	**16**	**0.9914**
Maxwell	27	LinearRegression	CFS+ RandomSearch	16	0.8354
Maxwell	27	K Star	CFS+ PSO	9	0.85
Maxwell	**27**	**K Star**	**CFS+ GA**	**20**	**0.8596**
Maxwell	27	M5p	ClassifierAttEval+ Ranker	24	0.8515
Maxwell	27	SMOreg	Corr. Att.Evaluation + Ranker	24	0.8336
Maxwell	27	M5p	Relief. Att.Evaluation + Ranker	24	0.8472
Maxwell	27	SMOreg	Relief. Att.Evaluation + Ranker	14	0.838
Kemerer	8	SMOreg	CFS+ RandomSearch	5	0.6795
Kemerer	8	SMOreg	CFS+ PSO	5	0.6946
Kemerer	8	SMOreg	CFS+ GA	5	0.6946
Kemerer	8	SMOreg	ClassifierAttEval+ Ranker	5	0.5405
Kemerer	**8**	**SMOreg**	**Corr. Att.Evaluation + Ranker**	**5**	**0.7171**
Kemerer	8	KStar	Relief. Att.Evaluation + Ranker	5	0.6295

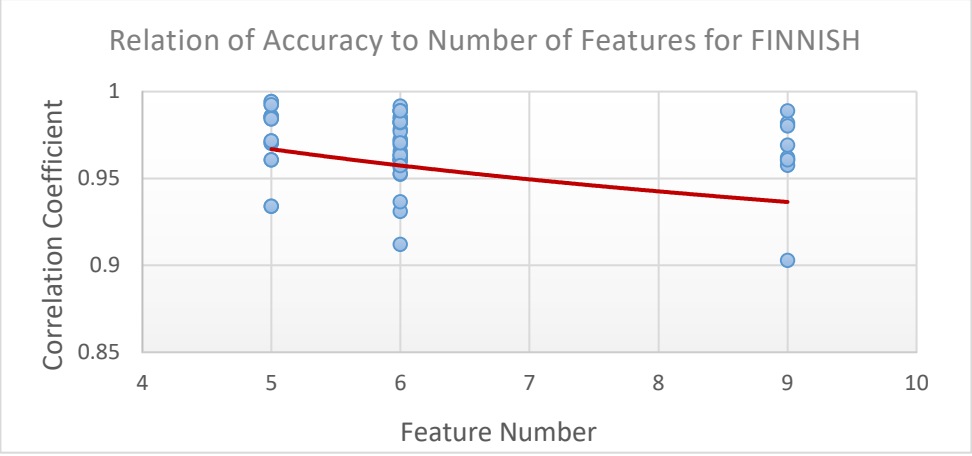


Figure 5.5: Relation of Accuracy to Number of Features for Finnish

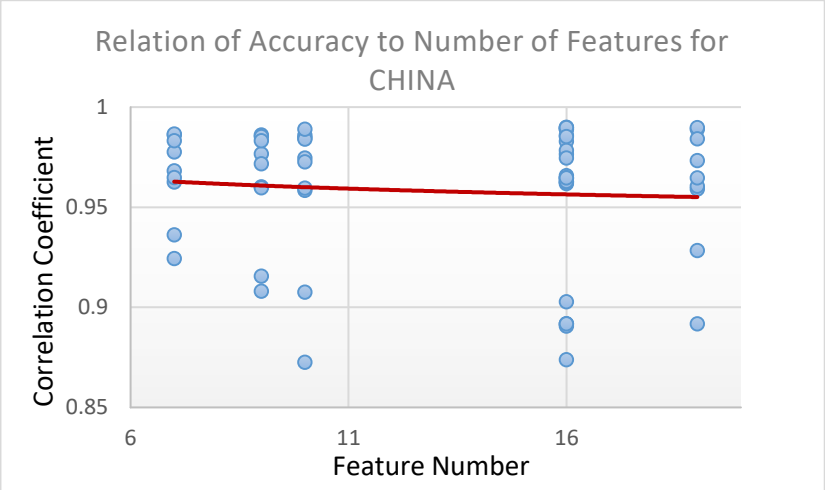


Figure 5.6 Relation of Accuracy to Number of Features for China

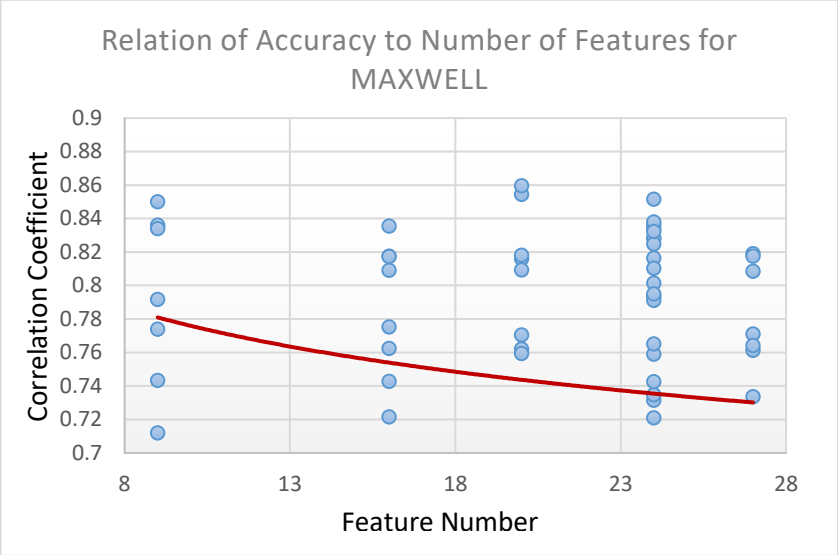


Figure 5.7 Relation of Accuracy to Number of Features for Maxwell

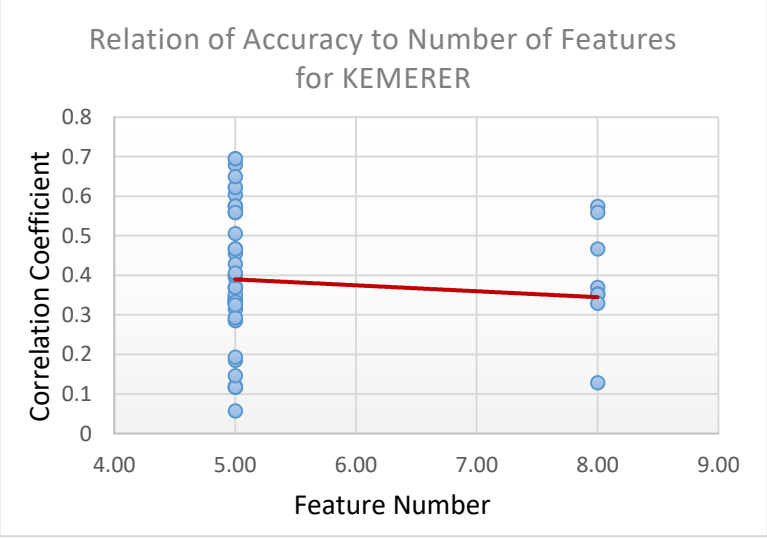


Figure 5.8 Relation of Accuracy to Number of Features for Kemerer

Table 5.6 Comparative Analysis of Gained Results with Literature

Dataset	Author(s)	Intelligent method	MMRE	PRED	Correlation Coefficient	MAE	RAE
China	(Rehal & Sharma, 2021)	SMOReg			0.9897	270.4561	7.3095
	(Kumar, Behera, Kumari, Nayak & Nail, 2020)	Spiking Neural Network	0.23				
		fuzzy c-means clustering-Functional Link Artificial Neural Networks	0.45				
		intuitionistic fuzzy c-means clustering-Functional Link Artificial Neural Networks	0.33				
		Long short-term memory	0.41				
		Output layer self-connection recurrent neural networks	0.32				
	Proposed Model	MLP & Relief Att.Eval. + Ranker	0.2655	0.0847	0.9914	370.1846	10.005
Finnish	(Benala & Bandarupalli, 2016)	AnalogyBased Estimation - Least Squares Support Vector Machin	1.7974	0.52			
		AnalogyBased Estimation - Extreme Learning Machines	2.3929	0.15			
		AnalogyBased Estimation - Artificial Neural Networks	2.124	0.32			
	Proposed Model	Kstar & ClassifierAttEval + Ranker	0.2521	0.0104	0.9948	0.0873	8.5274
Maxwell	(Benala & Bandarupalli, 2016)	AnalogyBased Estimation - Least Squares Support Vector Machin	1.1529	0.42			
		AnalogyBased Estimation - Extreme Learning Machines	4.2891	0.16			
		AnalogyBased Estimation - Artificial Neural Networks	4.4466	0.12			
	(Kumar, Behera, Kumari, Nayak & Nail, 2020)	Artificial Neural Network	1.32				
		Functional Link Artificial Neural Networks	0.42				
		Elman neural network	1.3748				
		Long short-term memory	0.37				
		Output layer self-connection recurrent neural networks	0.31				
	Proposed Model	Kstar & CFS + GA	0.7644	0.1274	0.8596	4078.324	64.913
Kemerer	(Benala & Bandarupalli, 2016)	AnalogyBased Estimation - Least Squares Support Vector Machin	0.66412	0.4			
		AnalogyBased Estimation - Extreme Learning Machines	1.8071	0.13			
		AnalogyBased Estimation - Artificial Neural Networks	2.0333	0.08			
	Proposed Model	SMOReg & Corr. Att.Evaluation + Ranker	0.5940	0.1289	0.7171	103.4371	64.273

In Table 5.6, the best results obtained and the literature studies found with Artificial Neural Network methods applied to the same datasets and Machine Learning methods without feature selection are given. It is clear that high performance can be achieved with machine learning models by applying the low-cost and sustainable model feature selection targeted in the study. In the model outputs created with the relevant datasets, it was determined that the highest performance measurements as algorithms were obtained when KStar, SMOreg, MultilayerPerceptron and LinearRegression were used. It has been noted that models created using IBk, RandomTree and Bagging algorithms tend to give low results.

As a result, it seems that Machine Learning Based Approaches can be used as a high-performance method for software cost estimation and it is an open area for

improvement. In future studies, similar methods can be studied with more and different datasets in order to generalize the obtained inferences and improve performance with different parameter values.

REFERENCES

[1] LIU Qin and MINTRAM Robert C. (2005), "Preliminary Data Analysis", *Software Quality Journal*, Volume 13, pp. 91-115.

[2] SONER Taner (2014), *Parametrik Tahmin Modellerinin Yazılım Projelerine Uygulanmasına Yönelik Bir Yazılım Paketinin Geliştirilmesi* (Master's Thesis), Ankara University Institute of Science, Ankara.

[3] RIJWANI Poonam and JAIN Sonal (2016), "Enhanced Software Effort Estimation using Multi Layered Feed Forward Artificial Neural Network Technique", *Procedia Computer Science*, Volume 89, pp. 307-312.

[4] WIKIPEDIA (2023), *Software Effort Estimation Methods*, http://en.wikipedia.org/wiki/Software_development_effort_estimation, DoA. 15.6.2023.

[5] RICHARDSON Ita, CASEY Valentine, BURTON John and MCCAFERY Fergal (2010), "Global Software Engineering: A Software Process Approach", *Collaborative Software Engineering*, Ed. Ivan Mistrik, John Grundy, Andre Hoek, Jim Whitehead, pp. 35-36, Springer, Berlin.

[6] WIECZOREK Isabella and RUHE Melanie (2002), "How valuable is company-specific data compared to multi-company data for software cost estimation?", *Proceedings Metrics'02,* pp. 237-246, Ottowa.

[7] BOEHM Barry, ABTS Chris and CHULANI Sunita (2000), "Software Development Cost Estimation Approaches – A Survey", *Annals of Software Engineering*, Volume 1, No 4, pp. 177-205.

[8] LEUNG Hareton and FAN Zhang (2002), "Software Cost Estimation", *In*, *Handbook of Software Engineering and Knowledge Engineering*, Ed. S K Chang, Volume 2, pp. 307-324, Emerging Technologies, World Scientific Publishing Co. Pte Ltd., Singapore.

[9] WIN Pa, MYINT War, H. MON Phyu and W THU Seint (2019), "Review on Algorithmic and Non-Algorithmic Software Cost Estimation Techniques", *International Journal of Trend in Scientific Research August,* Volume 3, No 5, pp. 890-895.

[10] RAJESWARI K. and BEENA R. (2018), "A Critique on Software Cost Estimation", *International Journal of Pure and Applied Mathematics*", Volume 118, No 20, pp. 3851-3862.

[11] Project Management Institute (2013), *A GUIDE TO THE PROJECT MANAGEMENT BODY OF KNOWLEDGE (PMBOK® guide),* 5th Edition, Project Management Institute, USA.

[12] TRIPATHI Rekha and DR. RAI P. K. (2016), "Comparative Study of Software Cost Estimation Techniques", *International Journal of Advanced Research in Computer Science and Software Engineering,* Volume 6, No 1, pp. 4771-4776.

[13] SHWETA K. R., DURAISMY S. and LathaMaheswari T. (2021), "Comparative Analysis Of Algorithmic, Non Algorithmic And Machine Learning Models For Software Cost Estimation: A Survey", *International Research Journal of Modernization in Engineering Technology and Science,* Volume 3, No 3, pp. 2119-2129.

[14] SMITH J. and JOHNSON A. (2022), "Comparing Machine Learning and Traditional Models for Software Development Project Prediction", *Journal of Software Engineering,* Volume 15, No 2, pp. 123-137.

[15] DANASINGH Asir Antony, BALAMURUGAN Suganya and EPIPHANY Jebamalar Leavline (2016), "Literature Review on Feature Selection Methods for High-Dimensional Data", *International Journal of Computer Applications,* Volume 136, No 10, pp. 9-17.

[16] DOKEROGLU Tansel, SEVINC Ender, KUCUKYILMAZ Tayfun and COSAR Ahmet (2019), "A survey on new generation metaheuristic algorithms", *Computers & Industrial Engineering,* Volume 137, pp. 106040, DOI: 10.1016/j.cie.2019.106040.

[17] ERGUZEL Turker, OZEKES Serhat, TAN Oguz and GULTEKIN Selahattin (2015), "Feature Selection and Classification of Electroencephalographic Signals an Artificial Neural Network and Genetic Algorithm Based Approach", *Clinical EEG and Neuroscience,* Volume 46, No 4, pp. 321-326.

[18] ORESKI Stjepan and ORESKI Goran (2014),"Genetic algorithm-based heuristic for feature selection in credit risk assessment", *Expert systems with applications,* Volume 41, No 4, pp. 2052-2064.

[19] WANG Yanqiu, CHEN Xiaowen, JIANG Wei, LI Li, LI Wei, YANG Lei, LIAO Mingzhi, LIAN Baofeng, LV Yingli, WANG Shiyuan, WANG Shuyuan and LI Xia (2011), "Predicting human microRNA precursors based on an optimized feature subset generated by GA–SVM", *Genomics,* Volume 98, No 2, pp. 73-78.

[20] YANG He, DU Qian and CHEN Genshe (2012), "Particle swarm optimization-based hyperspectral dimensionality reduction for urban land cover classification", *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing,* Volume 5, No 2, pp. 544-554.

[21] DOKEROGLU Tansel, DENIZ Ayca and KIZILOZ Hakan Ezgi (2022), "A comprehensive survey on recent metaheuristics for feature selection", *Neurocomputing,* Volume 494, No 14, pp. 269-296.

[22] RUIZ Roberto, RIQUELME Jose C. and AGUILAR-RUIZ Jesus S. (2006), "Incremental wrapper-based gene selection from microarray data for cancer classification", *Pattern Recognition,* Volume 39, No 12, pp. 2383-2392.

[23] NASSIF Ali, HO Danny, CAPRETZ Luiz (2013), "Towards an early software estimation using log-linear regression and a multilayer perceptron model", *Journal of Systems and Software,* Volume 86, pp. 244-160.

[24] SHARMA Pinkashia and SINGH Jaiteg (2017), "Systematic Literature Review on Software Effort Estimation Using Machine Learning Approaches", *International Conference on Next Generation Computing and Information Systems (ICNGCIS),* pp. 43-47, Jammu, India.

[25] POSPIESZNY Przemyslaw, CZARNACKA-CHROBOT Beata and KOBYLINSKI Andrzej (2018), "An effective approach for software project effort and duration estimation with machine learning algorithms", *Journal of Systems and Software*. Volume 137, pp. 184-196.

[26] BANIMUSTAFA Ahmed (2018), "Predicting Software Effort Estimation Using Machine Learning Techniques", *8th International Conference on Computer Science and Information Technology (CSIT)*, pp. 249-256, Amman, Jordan.

[27] ASAD Ali and CARMINE Gravino (2019), "A systematic literature review of software effort prediction using machine learning methods", *Journal of Software: Evolution and Process,* Volume 31, No 10, pp. e2211.

[28] SINGH A.J and KUMAR Mukesh. (2020), "Comparative Analysis on Prediction of Software Effort Estimation Using Machine Learning Techniques", *Proceedings of the International Conference on Innovative Computing & Communications (ICICC)*, Shimla, India, DOI: 10.2139/ssrn.3565813.

[29] ASAD Ali and CARMINE Gravino (2021), "Improving software effort estimation using bio-inspired algorithms to select relevant features: An empirical study ", *Science of Computer Programming*, Volume 205, pp. 102621, DOI: 10.1016/j.scico.2021.102621.

[30] RITU and GARG Yashika (2022), "Comparative Analysis of Machine Learning Techniques in Effort Estimation", *International Conference on Machine Learning, Big Data, Cloud and Parallel Computing (COM-IT-CON)*, pp. 401-405, Faridabad, India.

[31] J SHARMA Amrita and CHAUDHARY Neha (2022), "Analysis of Software Effort Estimation Based on Story Point and Lines of Code using Machine Learning", *International Journal of Computing and Digital Systems*, Volume 12, No 1, pp. 131-140.

[32] ADHAV Akshay and SHANDILYA Shishir Kumar (2023), "Reliable machine learning models for estimating effective software development efforts: A comparative analysis ", *Journal of Engineering Research*, DOI: 10.1016/j.jer.2023.100150.

[33] SAYYAD Shirabad and MENZIES Tim (2005), *The PROMISE Repository of Software Engineering Databases*, School of Information Technology and Engineering University of Ottawa, Canada, http://promise.site.uottawa.ca/SERepository.

[34] JIAO Shuming, GAO Yang, FENG Jun, LEI Ting, and YUAN Xiaocong (2020), "Does deep learning always outperform simple linear regression in optical imaging?", *Optics Express*, Volume 28, No 3, pp. 3717-3731.

[35] ADWAN Omar, FARIS Hossam, JARADAT Khalid, HARFOUSHI Osama and GHATASHEH Nazeeh (2014), "Predicting customer churn in telecom industry using multilayer preceptron neural networks: Modeling and analysis", *Life Science Journal*, Volume 11, No 3, pp. 75-81.

[36] PLATT John (1998), "Sequential Minimal Optimization: A Fast Algorithm for Training Support Vector Machines", *Advances in Kernel Methods - Support Vector Learning*, Volume 208, pp. 185-208.

[37] SHAREF Nurfadhlina Mohd, MARTIN Trevor, KASMIRAN Khairul Azhar, MUSTAPHA Aida, SULAIMAN Md. Nasir and AZMI-MURAD Masrah Azrifah (2015), " A comparative study of evolving fuzzy grammar and machine learning techniques for text categorization", *Soft Computing*, Volume 19, No 6, pp. 1701-1714.

[38] HAMMAD Mustafa and ALQADDOUMI Abdulla (2018), "Features-Level Software Effort Estimation Using Machine Learning Algorithms", *International Conference on Innovation and Intelligence for Informatics, Computing, and Technologies (3ICT)*, pp. 1-3, Sakhier, Bahrain.

[39] LIU Huan and YU Lei (2005), "Toward integrating feature selection algorithms for classification and clustering", *IEEE Transactions on Knowledge and Data Engineering*, Volume 17, No 4, pp. 491-502.