Software Maintenance Effort Estimation

Serkan Yıldırım, Sedat Tuna Akın
Bilgisayar Mühendisliği Bölümü
Yıldız Teknik Üniversitesi, 34220 Istanbul, Türkiye
{l1117078, l1118058}@std.yildiz.edu.tr

***Özetçe* —Proje kapsamında mevcut yazılımlar için yapılan bakım ve geliştirme çalışmalarında emek ve zaman kestirimi yapılmaktadır. Veri setlerinin oluşturulması için özellik seçimi algoritması ve korelasyon matrisi yöntemi kullanılmaktadır. Algoritmanın eğitimi için KNN (K-En yakın komşu) ve Doğrusal Regresyon algoritmaları kullanılmaktadır. Oluşturulan farklı veri setleri üzerinde bu algoritmalar çalıştırılarak elde edilen sonuca göre isabetliliği en yüksek olan veri seti ve algoritma eğitim için kullanılmıştır. Sonuç olarak hedeflenen R2 skoruna ulaşılamadığı için farklı bir çözüm denenmiş ve çalışma kapsamında edinilen bilgiler eşliğinde mevcut yazılımlar için yapılan bakım ve geliştirme çalışmaları efor tahmini için formül oluşturulmuştur.**

Anahtar Kelimeler—Özellik Seçimi, Korelasyon Matrisi, KNN, Doğrusal Regresyon.

***Abstract*—Within the scope of the project, effort and time estimation is made in the maintenance and development studies for existing software. Feature selection algorithm and correlation matrix method are used to create data sets. KNN (K-Nearest Neighbors) and Linear Regression algorithms are used for training the algorithm. The data set and algorithm with the highest accuracy according to the result obtained by running these algorithms on the different data sets created were used for training of the algorithm. As a result, since the targeted R2 score could not be reached, a different solution was tried and a formula was created for the effort estimation of maintenance and development studies for existing software with the information obtained within the scope of the study.**

Keywords—Future Selection, Correlation Matrix, KNN, Linear Regression.

I. Introduction

When the effort/time calculation studies for software are examined, the applications developed for the calculations to be made before the software development occupy a very large place among these studies. In addition, applications developed for effort/time estimation in maintenance and development studies for existing software are insufficient for the developing and living software ecosystem. For this reason, we started to work on the project with the approval of our consultant, thinking that it would be right to work on this area.

As a result of our initial research, we observed that methods such as COCOMO, KLOC and FP are among the existing studies. In addition to these methods, we searched to meet our need for a database containing development records on existing software. After the examinations on the database, we concluded that the KLOC method would give more accurate results on the extracted data set, and we decided to compare it with the outputs of our algorithm.

Different data sets to be used for algorithms were created by performing operations on the database. Firstly, the data that was copied was cleaned and different commits for a job were combined. After the data set was simplified and made usable, different data sets were created by using the Correlation Matrix method and Feature Selection algorithm to determine the features to be used.

To decide which method and which data set to use for the training of the algorithm, we tested the KNN (K-Nearest Neighbors) algorithm and the Linear Regression algorithm on all data sets we extracted and determined the pair of algorithm and data set that gave the most accurate result as output.

After the studies, the desired high R2 score could not be reached due to the lack of data. For this reason, in the continuation of the study, we proceeded in a different direction and a formula was created by which we can make software maintenance effort/time estimation by using the ones found appropriate according to the scores of the features we have in the regression models and correlation matrix within the study.

II. Related Work

There are many studies on effort/time estimation that can be used before the software project deployed, but there is not much work on software maintenance effort estimation. The most known of the studies that can be used before the software project deployed, is the COCOMO model. The COCOMO model calculates the software size based on the KLOC method and the coefficient in its formula varies for organic, semi-independent and embedded projects. The COCOMO model has been used as a template for many models. In fact, it is not a software maintenance effort/time estimation method, but it can be used for this process with appropriate parameters and formulas.[1]

It is seen that almost all the models produced are derived from the COCOMO model. The COCOMO 2.0 model that emerged in this way is more about project management than a effort/time estimation model. Therefore, it needs a lot of input and is not suitable for effort/time estimation.[1]

The ACT model makes a calculation that indicates how much a software has changed during the year. It is found by dividing the sum of the total number of new rows and changed rows by the number of original rows. Calculating the size of the project and multiplying it by this value is used to estimate the maintenance effort/time value.[2]

Studies have found that using FP for the size and complexity of their software is a much more consistent

method. For this reason, there are many studies in which FPs are used as a project scale.[3] Studies have shown with mathematical formulas that as FP increases, effort increases.

III. DATA SET

The data set we chose for our project is from projects older than 3 years, more than 500 commits and more than 100 classes, written in JAVA, bug records opened in JIRA.Data extracted with Pydriller, Ptidej, Refactoring Miner, SonarQube and SZZ Algorithm tools. It is a dataset that contains the properties of the code and files committed for The data set itself is used as a database. It consists of 9 tables. SQL was used for the information to be retrieved from the database. By matching the information about the project in the SONARMEASURES table with the closed records in the JIRAISSUES table with the commitHash values, the data is retrieved. It took a long time to pull the data due to both the size of the database and the large number of joins in the SQL query.

A. Data Cleaning

First of all, the data set to be used in our project was processed. While processing the data, firstly, duplicate rows are removed because it affects the regression model to be created badly. Then the added and removed rows obtained from each commit are collected and put into a new column.

B. Feature Selection

Correlation matrix and feature selection methods were used for feature selection, which is the next step after data processing. After the correlation matrix was created, five different levels were determined and five data sets were created.

IV. REGRESSION ALGORITHMS

Linear Regression and KNN Regression algorithms were tried on the data sets created with the correlation matrix and then normalized, and it was used to understand which one worked better in estimating the effort value. The analysis was further examined with OLS Regression.

V. CONCLUSION

In this project, effort/time estimation was made for maintenance of the software. The data set used for effort estimation is divided by the correlation matrix to include different features. Regression analysis was applied on the divided data sets and it was tried to observe which features were associated with effort.

Due to the small number of samples in our data set, regression analyzes did not work successfully, but it did give us an idea to generate a formula for effort.

In the project, the number of rows added and removed for each error record in the data set was divided by the total number of rows and normalized, and the first stage of the formula was formed by multiplying the cyclical complexity of that file. The effort formula was obtained by adding the newSqaleDebtRatio feature, which expresses the ratio

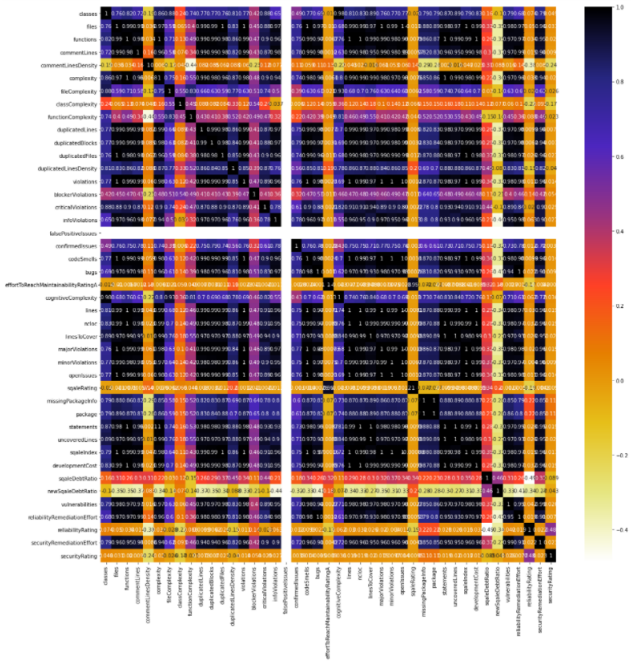
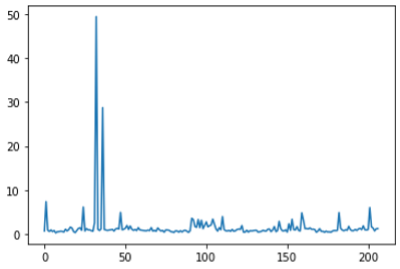


Figure 1 Correlation Matrix Obtained From The data set




```
count    207.000000
mean       1.611598
std        3.975200
min        0.310548
25%        0.747834
50%        0.954987
75%        1.284750
max       49.366090
Name: bölüm, dtype: float64
```

Figure 2 Efor Formülünün Son Halinin Gerçek Efora Oranı

of technical debt compared to the cost of developing all the source code obtained from the regression analysis from scratch, and the 7 constant multipliers obtained from the ratio of the coefficients of the formula created so far to the formula.

The resulting effort formula was compared with the COCOMO model and its accuracy was examined. The R2 value was found to be 0.92. The obtained formula provides effort estimation with the same accuracy as COCOMO, using different metrics from the models in the literature to obtain the effort value.

REFERENCES

[1] V. Nguyen, "Improved size and effort estimation models for software maintenance," in *2010 IEEE International Conference on Software Maintenance*, 2010, pp. 1–2.

[2] C. Syavasya, "Evaluation of changes on annual change traffic in calculating maintenance cost in man-months based on constructive cost model," *www.ijcst.com*, vol. 4, 01 2013.

[3] D. Tran-Cao and G. Lévesque, "1 maintenance effort and cost estimation using software functional sizes," 2003.