Avrupa Bilim ve Teknoloji Dergisi
Sayı 20, S. 805-814, Aralık 2020

**Araştırma Makalesi**

www.ejosat.com ISSN:2148-2683

European Journal of Science and Technology
No. 20, pp. 805-814, December 2020

**Research Article**

# Could Mobile Applications' Success be Increased via Machine Learning and Business Intelligence Methods?

Murat Kılınç[1*], Çiğdem Tarhan[2], Can Aydın[3]

[1] Manisa Celal Bayar University, Computer Research and Application Center, Manisa, Turkey (ORCID: 0000-0003-4092-5967), kilinc.murat@cbu.edu.tr
[2] Dokuz Eylul University, Department of Management Information Systems, Izmir, Turkey (ORCID: 0000-0002-5891-0635), cigdem.tarhan@deu.edu.tr
[3] Dokuz Eylul University, Department of Management Information Systems, Izmir, Turkey (ORCID: 0000-0002-0133-9634), can.aydin@deu.edu.tr

(First received 10 September 2020 and in final form 5 December 2020)

(**DOI:** 10.31590/ejosat.793069)

**ATIF/REFERENCE:** Kılınç, M., Tarhan, Ç. & Aydın, C. (2020). Could Mobile Applications' Success be Increased via Machine Learning and Business Intelligence Methods? *European Journal of Science and Technology*, (20), 805-814.

**Abstract**

The number of applications developed for mobile platforms is increasing as developer interest is largely directed towards mobile platforms. In addition to the IOS platform, Google Play Store, one of the main platforms where developed mobile applications are published, also has a lot of developer interest, especially because it is open source. But there is no platform that developers can benefit from for elements such as the success that the developed application can provide or what features it should have. In this study, this problem was addressed. Accordingly, it is aimed to estimate and classify success according to the characteristics of the developed application. In addition, the evaluation of the developed application within the scope of business intelligence according to the characteristics of the previously developed applications is one of the main points of the study. Within the scope of the research, Decision Tree Regressor (DTC), Random Forest Regressor (RFR), K-Neighbors Regressor (KNN) and AdaBoost Regressor (ABR) were used for application rating estimates and the accuracy of the metrics were tested with R square score (R2), Mean Square Error (MSE) and Root Mean square Error (RMSE). Estimates for classification Random forest classification (RFC) decision tree classification (DTC), the K-Neighbors Classification (KNC), Classification MLP (MLP), AdaBoost Classification (ABC) and naïve Bayes (GNB) has been tested with the algorithms used and the accuracy of confusion matrix metrics. In this context, DTR with 80.73% and RFR with 82.89% gave the best results for rating estimation, DTC with 86.08% and RFC algorithms with 89.83% gave the best results for success classification. All predictions made with machine learning management in the scope of the study are dynamically shown in the web interface using the Flask framework. Therefore, a platform was created where developers could get Decision Support with business intelligence and the resulting results were analyzed and transferred into the work. In this way, mobile application developers will be able to see their shortcomings, if any, and have a prediction in terms of success.

**Keywords:** Business Intelligence, Machine Learning, Web Applications, Decision Support System.

# Makine Öğrenmesi ve İş Zekâsı Yöntemleriyle Mobil Uygulamaların Başarısı Arttırılabilir mi?

**Öz**

Mobil platformlar için geliştirilen uygulamaların sayısı, geliştirici ilgisinin büyük ölçüde mobil platformlara yönelmesiyle birlikte artış göstermektedir. IOS platformunun yanısıra, geliştirilen mobil uygulamaların yayınlandığı ana platformlardan birisi olan Google Play Store'da da özellikle açık kaynak kodlu olması sebebiyle, yoğun bir geliştirici ilgisi mevcuttur. Fakat geliştirilen uygulamanın sağlayabileceği başarı ya da hangi özelliklere sahip olması gerektiği gibi unsurlar için geliştiricilerin yararlanabileceği bir platform bulunmamaktadır. Bu çalışmada da bu sorun üzerine gidilmiştir. Bu doğrultuda, geliştirilen uygulamanın özelliklerine göre bir başarı tahminlemesi ve sınıflandırma yapılması amaçlanmıştır. Ayrıca geliştirilen uygulamanın, daha önce geliştirilmiş olan uygulamaların özelliklerine göre iş zekâsı kapsamında değerlendirilmesi de çalışmanın dayanak noktalarından biridir. Araştırma kapsamında, uygulama rating tahminleri için Decision Tree Regressor (DTC), Random Forest Regressor (RFR), K-Neighbors Regressor (KNN) ve AdaBoost Regressor (ABR) kullanılmış ve metriklerin doğruluğu R kare skoru (R2), Mean Square Error (MSE) ve Root Mean Square Error (RMSE) ile test edilmiştir. Sınıflandırma tahminleri için ise Random Forest Classification (RFC), Decision Tree Classification (DTC), K-Neighbors Classification (KNC), MLP Classification (MLP), AdaBoost Classification (ABC) ve Naive Bayes (GNB) algoritmaları kullanılmış ve metriklerin doğruluğu confusion matrix ile test edilmiştir. Bu kapsamda rating tahmini için en iyi sonuçları

[*] Corresponding Author: Manisa Celal Bayar Üniversity, Computer Research and Application Center, Manisa, Turkey, ORCID: 0000-0003-4092-5967, kilinc.murat@cbu.edu.tr

%80.73 ile DTR ve %82.89 ile RFR, başarı sınıflandırması için en iyi sonuçları ise %86.08 ile DTC, %89.83 ile RFC algoritmaları vermiştir. Çalışma kapsamındaki makine öğrenmesi yönetimleriyle yapılan tüm tahminlemeler dinamik bir şekilde Flask framework kullanılarak web arayüzünde gösterilmiştir. Dolayısıyla, iş zekâsı ile geliştiricilerin karar desteği alabileceği bir platform oluşturulmuş ve ortaya çıkan sonuçlar analiz edilerek çalışma içerisine aktarılmıştır. Bu sayede, mobil uygulama geliştiricileri varsa eksikliklerini görebilecekler ve başarı anlamında bir öngörüye sahip olabileceklerdir.

**Anahtar Kelimeler:** İş Zekâsı, Makine Öğrenmesi, Web Uygulamaları, Karar Destek Sistemi.

# 1. Introduction

With the increase in the use of smart phones, the mobile application ecosystem has become the most important part of the software industry's cake. Because when evaluated in terms of usage rate, mobile and tablet usage is higher than the desktop usage rate with 53,1% (*Business of Apps*, 2020; *Statcounter*, 2020a; *Statista*, 2020). This situation has led to the creation of application markets such as Google Play and App Store in Android and IOS mobile operating systems and the proliferation of applications developed for these markets (Holzer & Ondrus, 2011). The interest in these markets is higher, with 74,6% on the Android side, especially since most of them are open source. The usage rate in IOS, another large platform, appears to be 24,8% (*Statcounter*, 2020b). With these rates, there are 2.2 million developed applications in the App Store, while there are around 3 million applications in Google Play. According to download statistics, applications on both platforms are expected to be downloaded more than 350 billion times in 2021. While there are an intense developer and user interest for both platforms, the applications developed must be sufficient and sustainable in terms of quality, security and design (Franke & Weise, 2011). IOS and Android users show different behaviours at the point of choosing the application (Zimbra et al., 2017). IOS users are willing to pay extra for sustainable applications in terms of quality, security, and design. So subscription and in-app purchase model is standard. Android users have adopted application models based on ads (Lamhaddab et al., 2019). Advertising-based applications alone may be insufficient in terms of quality and security. For this reason, applications in Google Play must be taested for success before or after they are published, and their deficiencies must be eliminated. However, mobile application developers' inability to successfully test their applications is a problem that concerns both developers and stakeholders investing in the application. For testing in terms of success, a model should be created after analyzing the data of other applications developed. Data is the critical component in the process of creating the model and showing it on the application. Reports, inferences, and predictions that can increase applications' success are possible by analyzing the data comprehensively. This whole process has been made possible by a significant increase in the number of recorded data with the development of information and communication technologies. With this increase, useful and meaningful information could be obtained from the data, and data analysis became more significant (Birant, 2011; Lam, 2004; Olson et al., 2012; Onan, 2015). Popular data centres such as Kaggle can be used for these analyzes, which are both practical and trendy today. Accordingly, the Google Play Applications data set consisting of approximately 10,000 data was used for the research's data-based machine learning and business intelligence model.

When Google Play applications are examined within the scope of success prediction with business intelligence and machine learning, which are the research method, many studies have been found. In a study conducted in the field of health in 2015, 65 mobile applications were examined using a survey method, and mobile applications were classified and divided into specific groups. (Stoyanov et al., 2015). In another study, the classification process was carried out by analyzing the texts written about the applications on the web pages to classify mobile applications (Zhu et al., 2012). In another study conducted in 2017, sentiment analysis was performed using machine learning algorithms with the reviews of more than 200,000 applications. According to the study results, the Support Vector Machine (SVM) algorithm gives the best analysis result with 94%. (Day & Lin, 2017). As seen in the classification studies of mobile applications, the data required for classification have been revealed by text mining, usually using comments written on mobile application stores and web pages about applications (García-Peñalvo et al., 2014). In addition to mobile applications' comments, the inclusion of data roaming in machine learning methods is also included in the literature. In another study conducted by Alfonso and his team in 2015, it was estimated whether there is malware according to the features of mobile applications on Google Play (Munoz et al., 2015). In a similar study, more than 80,000 Android apps flagged as malware were analyzed. As a result of the analysis, applications of unknown type are classified as adware or malware with a rate of 84%. (Martín et al., 2019). In another study conducted in 2020, success was predicted with artificial neural networks using the characteristics of 100 successful and 100 unsuccessful applications (Dehkordi et al., 2020). In another study, approximately 1500 Android applications were examined, and security risk estimation was made with machine learning methods. In the estimation, it was seen that the Random Forest Algorithm (RF) performed better than other algorithms at all risk levels (Cui et al., 2020). Finally, in a study conducted in 2020, it was emphasized that the performance problems that could prevent the effective use of Android applications could be prevented with early solutions. Thus the success of the applications will increase. (Das et al., 2020). After examining similar studies in the literature, machine learning algorithms were used after obtaining information such as scoring, price, number of downloads, number of comments, category, and content of mobile applications in application stores. After the success estimation and classification processes were performed, reporting was made on a management panel using business intelligence tools. Combining both methods and creating a decision support system on the web reveals the study's innovative side that differs from other studies.

# 2. Material and Method

Within the scope of the business intelligence and machine learning processes creating a decision support structure on the web environment, first of all, the system design was introduced, and the processes in the system design were implemented step by step. In this direction, the Cross-Industry Standard Process for Data Mining (CRISP-DM) standard, which is widely used in data mining and machine learning projects, has been gradually applied in system design. (Gentner et al., 2018). CRISP-DM process model levels consist of 6 stages. These stages are called business understanding, data understanding, data preparation, modelling, evaluation, and deployment; the existing data is fully integrated into the system lifecycle process (Figure 1).

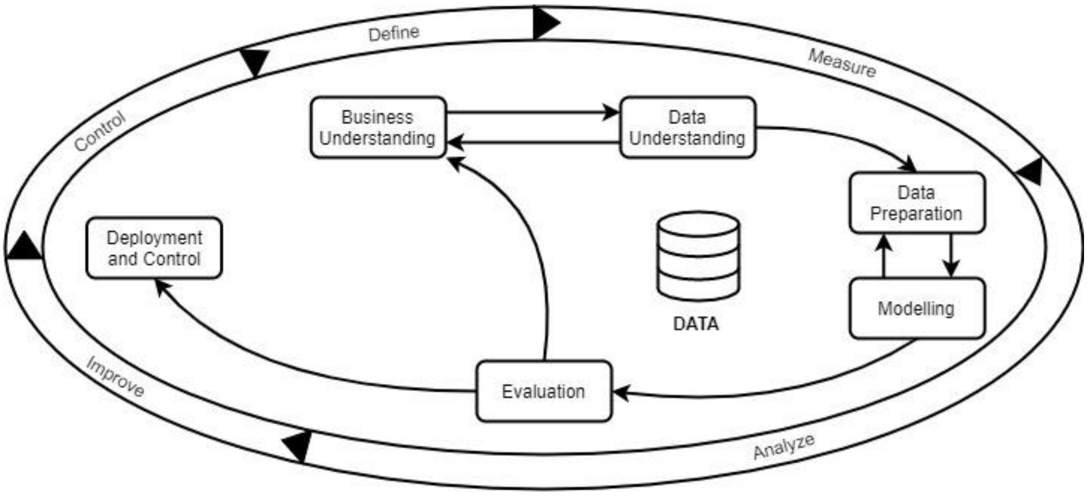


*Figure 1. Cross industry standard process for data mining* (De Mast & Lokkerbol, 2012; Huber et al., 2019)

## 2.1. Business-Data Understanding

At the stage of understanding the work, it was clarified what kind of contribution the developed system would contribute to the literature and which software libraries to be used were determined. Also, the problems that may arise during the development of the software have been sought. At the stage of data understanding, the data; how it is in terms of quality, accessibility and sustainability have been evaluated. Therefore, the system design scheme is laid out, and it is planned by specifying how the data will follow in terms of system operation. The web interface developed in this direction has been evaluated together in terms of business intelligence and machine learning. This evaluation aims to support the suggestions, reports, and inferences to be presented to the user with business intelligence and machine learning. In this context, the data integrated into the system was first analyzed on object-oriented programming languages PHP and Python (Figure 2). Later, visualization, estimation, and classification processes were carried out on the web interface. After all these processes, a decision support system was created by combining business intelligence and machine learning capabilities.

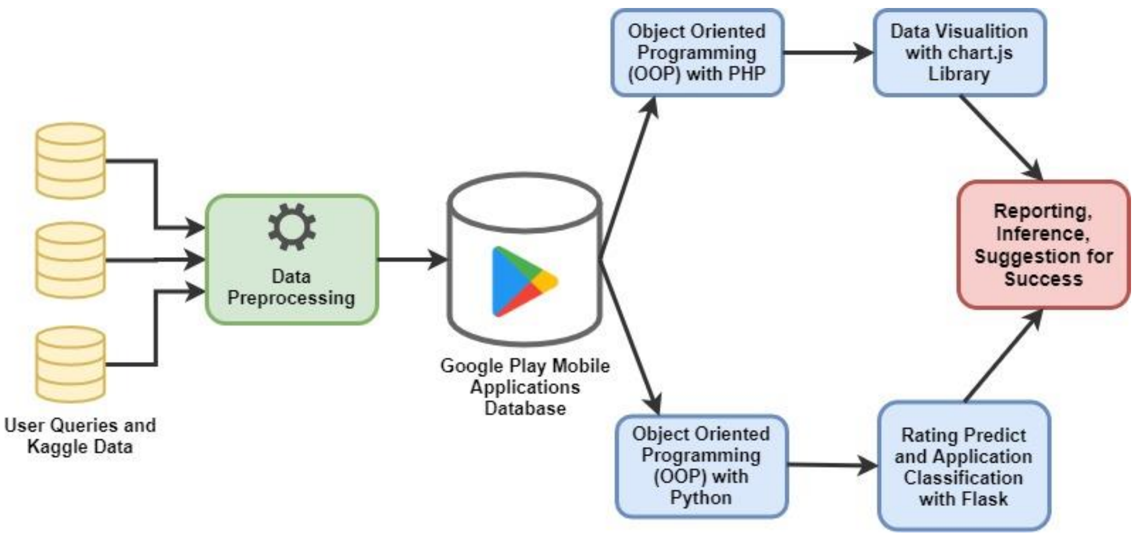


*Figure 2. Data path and analysis process in the context of business intelligence*

### *2.1.1. Software Libraries and Structures Used*

Application development was carried out in 3 separate processes. For this reason, the software libraries and structures used in each process differ. While developing the web application, HTML, CSS on the frontend, and the Chart.js library compiled with object-oriented programming languages PHP, Python, and Javascript on the Bootstrap backend were used. The Flask library, a web framework of the Python programming language, was used for the rating estimation and classification of the application (Table 1). Pandas and Numpy libraries, which provide features such as reading the data file and performing mathematical operations, were used in the data pre-processing process. The process of making the data ready for machine learning methods was done in this process, and the compilation of all processes was done in Jupyter. Finally, in the machine learning process, the sci-kit learn library structures were used for prediction and classification methods. Success predict and classification analyzes were made in this process, and Spyder was used as the compiler.

*Table 1. Components of the Developed Web Application*

| | **Web Application** | **Machine Learning** | **Data Preprocessing** |
|---|---|---|---|
| **Python Libraries and Structures** | Flask Framework | Sklearn Libraries | Pandas, Numpy Libraries |
| **Object-Oriented Programming Languages** | PHP, Javascript, Python (It was compiled with VS Code.) | Python (It was compiled with Spyder and VS Code.) | Python (It was compiled with Jupyter.) |
| **Other Tools and Libraries** | HTML, CSS, Bootstrap, Chart.js | Google Colab | Weka (Feature Selection) |

## 2.2. Business-Data Understanding

During the data preparation, since the basis of the research is based on the data source, to analyze a sufficient number of data sets in all processes in the system, the "Google Play Store Apps" data set containing many mobile application data was used. (Gupta, 2020). Mobile applications in Google Play have gone through the pre-processing data stage before being analyzed within the scope of various machine learning methods and business intelligence and become ready for analysis. The main finding aimed at this point is to make the data set entirely numerical. Because in the process of using machine learning business intelligence methods on this data set, the data must be completely numerical. In addition, data must be numerical within the scope of compatibility with the database because the convenience that arises in the transfer to the database stands out clearly (Table 2). Before the pre-processing process, the data set, which was 10.841 rows, 13 columns, and 1.1 MB in size, was numerical and feature selection was made through Weka software. As a result of data pre-processing, missing or incorrect lines were separated from the data set and the number of lines decreased to 9360. In the number of columns, as a result of the feature selection, three columns were deleted from the data set, and the numerical versions of the category and type columns were added to the main table, and a new data set of 12 columns in total was obtained. Also, the new size of the data set was 804 KB (Table 2). Analysis operations can be done more efficiently, thanks to the decreasing size as the data becomes numerical.

*Table 2. Data Set Status Before and After Data Preprocessing*

| **Before Preprocessing Data** | | | **After Preprocessing Data** | | |
|---|---|---|---|---|---|
| **10.841 line** | 13 column | Size: 1.1 MB | **9360 line** | 12 column | Size: 804 KB |

Before preprocessing:

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 10841 entries, 0 to 10840
Data columns (total 13 columns):
App               10841 non-null object
Category          10841 non-null object
Rating            9367 non-null float64
Reviews           10841 non-null object
Size              10841 non-null object
Installs          10841 non-null object
Type              10840 non-null object
Price             10841 non-null object
Content Rating    10840 non-null object
Genres            10841 non-null object
Last Updated      10841 non-null object
Current Ver       10833 non-null object
Android Ver       10838 non-null object
dtypes: float64(1), object(12)
memory usage: 1.1+ MB
```

After preprocessing:

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 9360 entries, 0 to 10840
Data columns (total 12 columns):
App               9360 non-null object
Category          9360 non-null object
Rating            9360 non-null float64
Reviews           9360 non-null int32
Size              9360 non-null float64
Installs          9360 non-null int64
Type              9360 non-null int64
Price             9360 non-null float64
Content Rating    9360 non-null int32
Genres            9360 non-null object
Category_c        9360 non-null int32
Genres_c          9360 non-null int32
dtypes: float64(3), int32(4), int64(2), object(3)
memory usage: 804.4+ KB
```

On the other hand, there can be significant differences between the data set's property column values . These large differences can cause properties with small values to be ignored during the classification process. For this reason, the data set should be normalized by compressing it between 0-1 values. For this purpose, the numerical data has been normalized.

$$X\_std = \frac{(X - X.min(axis = 0))}{(X.max(axis = 0) - X.min(axis = 0))} \quad (1)$$

This normalization may differ in variables that have a scaled value. Ratings of applications are in the range of 0-5. However, since the data in the data set is generally in the range of 3 to 5 values, the placement made between 0-1 is distributed only within these limits. As a solution to this situation, the mathematical formula introduced for normalization in the sklearn library was used (1). With the application of the formula to the rating column, all rating values between 0-5 are placed between 0-1 (Figure 3).

```
[9360 rows x 10 columns]
          0             2         3  ...         8        9  Rating
0  0.000000  2.021538e-06  0.189931  ...  0.000000  0.00000    0.82
1  0.000092  1.235953e-05  0.139927  ...  0.008772  0.00000    0.78
2  0.000185  1.119638e-03  0.086922  ...  0.000000  0.00000    0.94
3  0.000277  2.759054e-03  0.249936  ...  0.000000  0.00000    0.90
4  0.000369  1.235953e-05  0.027917  ...  0.017544  0.00000    0.86
5  0.000461  2.123895e-06  0.055920  ...  0.000000  0.00000    0.88
6  0.000554  2.264635e-06  0.189931  ...  0.000000  0.00000    0.76
```

*Figure 3. Post-normalization data set*

## 2.3. Modelling

In the modelling process carried out in a supervised learning process, modelling techniques have been applied in a structure that is most suitable for the desired job. Considered in a general context, which algorithms will be used in the modelling process, how to follow the path with these algorithms, and the system's working logic is evaluated. In the process of reaching the decision support system, which is the primary purpose of modelling in practice, it is to evaluate it in a learning model trained with an extensive data set by taking the values obtained from the user (Figure 4). In this context, using 15 estimation and classification methods, the choices are left to the user. However, some methods were removed from the web interface because they delayed the operation of the webserver. After the model creation, the data set was divided into training and test set to determine the quality of the model.

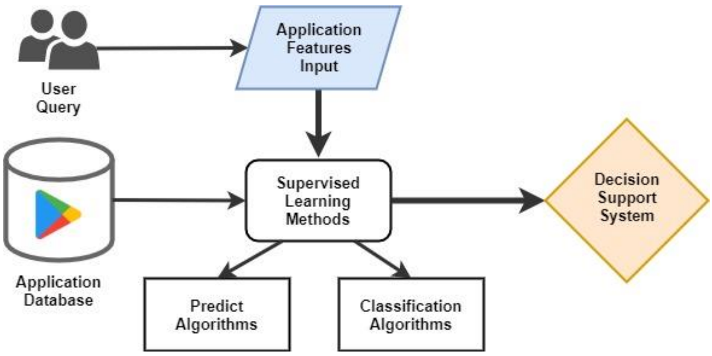


*Figure 4. Modeling process for supervised learning methods*

Therefore, the data pre-processed in the machine learning process were randomly selected and separated for training and testing and included in the process. In this context, the test data size in the cross-validation section was selected as "test_size = 0.20". A dynamic structure has been created by providing the transfer of the values entered in the web environment to the Python environment with the Flask framework structure. The web application, which passes through the machine learning process with all the system data, is designed to analyze the new incoming data in machine learning processes (Figure 4).

## 2.4. Evaluation

The regression and classification algorithms used in the research were tested with model evaluation metrics before being used in the application. Accordingly, the accuracy values for the classification algorithms were measured with F1 Score, Precision and Recall, and the accuracy values of the regression algorithms were measured with R2 Score, MSE, and RMSE. First of all, the confusion matrix must be presented to evaluate the accuracy of the classification algorithms. Then, the classification metrics can be evaluated (Table 3).

*Table 3. Confusion Matrix*

| | | **Real Values** | |
|---|---|---|---|
| | | Positive (1) | Negative (0) |
| **Predicted Values** | Positive (1) | **TP** | **FP** |
| | Negative (0) | **FN** | **TN** |

**TP:** *True Positive → The algorithm returns yes, the main result is yes*
**FP:** *False Positive → The algorithm returns yes, but the actual result is no*
**FN:** *False Negative → The algorithm returns no, the main result is yes*
**TN:** *True Negative → The algorithm returns no, the main result is no*

In this direction, it should be stated how much of the proposed model results were correctly estimated. For this, accuracy rate measurement is mostly used for classification methods. (Yang et al., 2020). The accuracy rate is obtained by dividing the correctly classified data into all data (2).

$$Accuracy\ Rate = \frac{\mathrm{TP} + \mathrm{TN}}{\mathrm{TP} + \mathrm{FP} + \mathrm{FN} + \mathrm{TN}} \quad (2)$$

After the accuracy ratio, the other two calculations using the confusion matrix are Recall and Precision measurements. The ratio of the number of correctly classified positive samples to the total number of correctly classified positive predicted samples and incorrectly classified negative samples is called sensitivity measurement (Üstün, 2019) (3). The measurement is considered as a metric that indicates how much of the values that need to be positively predicted are positively predicted. It is mainly used in cases where the cost of false negative estimation is high.

$$Recall = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN}} \quad (3) \qquad Precision = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FP}} \quad (4)$$

Based on the results obtained from the confusion matrix, precision measurement is calculated as the ratio of the number of correctly classified positive samples to the total positive samples (4). In other words, it is a measurement parameter that shows how many of the positively predicted values are positive. F1 score is a classification method evaluation metric in which extreme cases are not ignored. Measurement of the metric is obtained by the harmonic mean of the precision and recall values (5).

$$F1\ Score = \frac{2\ \times \mathrm{Recall}\ \times \mathrm{Precision}}{\mathrm{Recall} + \mathrm{Precision}} \quad (5)$$

Metrics such as R2, MSE, and RMSE are frequently used to evaluate regression algorithms. First, the R2 score is calculated by the formula (6). According to the formula, R2 score, which can take a value between -1 and +1, the higher the model fit, the better. When evaluated in terms of the relationship between variables, the R2 score between 0-0.25 is considered to be very weak, between 0.26-0.50 as weak, 0.50-0.69 as a medium, 0.70-0.89 as high, and between 0.90-1.00 as a very high agreement ( Erdal, 2015).

$$R^2\ Score = 1 - \frac{\sum_i (y_i - y_i')^2}{\sum_i (y_i - \bar{y})^2} \quad (6)$$

$$Mean\ Square\ Error(MSE) = \frac{1}{n}\sum_{i=0}^{n}(y_i - y_i')^2 \quad (7)$$

Second, MSE (7) shows how close a regression algorithm is to a set of points. The closer the always-positive MSE is to zero, the better the algorithm can predict. Finally, RMSE is a metric frequently used to find the distance between estimated values and actual values (8). The fact that the RMSE value is close to zero indicates that the algorithm used can make accurate predictions (Aydın, 2018).

$$Root\ Mean\ Square\ Error\ (RMSE) = \sqrt{\frac{1}{n}\sum_{i=0}^{n}(y_i - y_i')^2} \quad (8)$$

Results of the evaluation of regression algorithms used in the application are shown in the table below (Table 4). According to the results, DTR algorithms with 87% and RFR algorithms, with 91% give the highest accuracy rates. Also, it can be said that these algorithms make an excellent rating estimation because the R2 score is high, and the MSE and RMSE values are close to zero.

*Table 4. Comparison of the Results of the Regression Algorithm*

| Regressor Algorithms | Accuracy Score | R2 | MSE | RMSE |
|---|---|---|---|---|
| *DTR* | 0.87 | 0.71 | 0.078 | 0.297 |
| *RFR* | 0.91 | 0.75 | 0.063 | 0.256 |
| *KNN* | 0.42 | 0.36 | 0.169 | 0.415 |
| *ABR* | 0.14 | 0.24 | 0.279 | 0.533 |

In the evaluation of classification methods, the algorithms that yielded the best results were DTC with 87% and RFC with 89% (Table 5). At the same time, since these algorithms give an F1 score close to the accuracy value, it is seen that DTC and RFC algorithms make a useful classification. The reason why other methods make classification with lower accuracy values is the characteristic feature of the data set. Because the data set has an unevenly distributed structure, this situation has enabled DTC and RFC algorithms that can classify better in unbalanced data sets to give better results.

*Table 5. Comparison of the Results of the Classification Algorithm*

| Classification Algorithms | Accuracy Score | F1 Score | Precision | Recall |
|---|---|---|---|---|
| *DTC* | 0.87 | 0.83 | 0.81 | 0.87 |
| *RFC* | 0.89 | 0.86 | 0.91 | 0.83 |
| *KNC* | 0.65 | 0.57 | 0.58 | 0.56 |
| *MLP* | 0.50 | 0.23 | 0.25 | 0.27 |
| *ABC* | 0.46 | 0.32 | 0.38 | 0.28 |
| *GNB* | 0.17 | 0.31 | 0.31 | 0.32 |

## 2.5. Deployment and Control

The model introduced and evaluated in the deployment process, which is the last stage of the CRISP-DM process, is implemented. In this direction, the web-based learning system has been successfully implemented for the planned job. (Huang et al., 2005). Within the scope of the study, the model was implemented in the web environment and a web application was developed. All processes control, business intelligence, and machine learning methods were transferred to mobile application developers. Developed web application; It consists of 5 sections called Dashboard, Add App Data, Data Management, Reporting, and Rating Predict. Accordingly, statistics and graphics regarding all data in the database are included on the Dashboard page. All lists and graphics are dynamically designed due to the system operating logic. When new data is entered into the system, the change can be observed instantly. On the Add App Data page, a registration form has been created for adding new applications. Each new data should be added to the system carefully, as it is dynamic and can make changes in general reports and graphics. Multiple queries can be made in line with the criteria specified on the Data Management page, and operations such as updating and deleting the results can be performed (Figure 5).

On the Reporting page, all the data obtained from the database were analyzed in line with various criteria, and the mobile application analysis was revealed in the form of questions and answers. On the Rating Predict page, a structure that performs the prediction and classification of success with machine learning methods according to the features of the developed application. The developer who tests the mobile application developed in this direction receives a rating and success classification estimation after entering the features such as size, category, price, type, content. Then, what needs to be done to improve the result is shown in the form of a report.

Therefore, with the revealed structure, a decision support system was provided, and the reporting capability of business intelligence and the estimation and classification methods of machine learning were combined with the use of the Flask framework (Figure 6). Besides, to strengthen the decision support system with visualization, all data in the system are visualized with unique criteria and presented to the user on the web interface. The data received on the interface side with PHP has been made suitable for visualization with various SQL queries. The data grouped according to various criteria were transferred red to the Chart.js library compiled with Javascript in a way to be visualized dynamically. Dynamic graphics change as data is added to the web application (Figure 7).

The graphics on the dashboard differ according to the type of data. Groups containing value range data are shown in the form of a pie chart, categorized groups in the form of a bar chart, and groups containing average values on graphs showing the average values of all values (Figure 8).

In addition, every query made in the developed system is added to the database as mobile application information. In this way, the data set in which machine learning methods can be used is increased. In other words, algorithms that make regression and classification within the scope of supervised learning will improve themselves as the query is made.

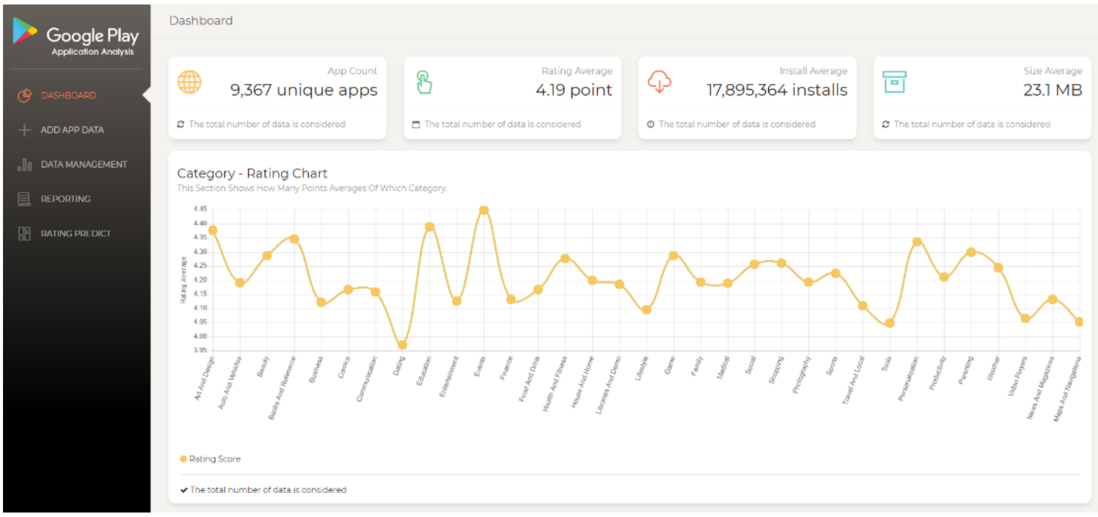


*Figure 5. A view from the developed web application interface*

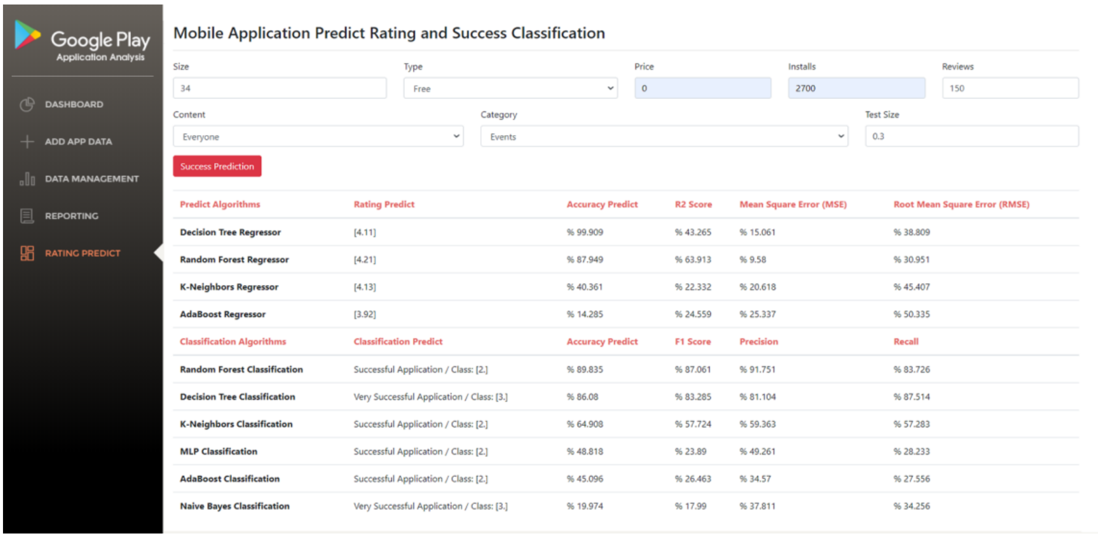


*Figure 6. Demonstration of machine learning methods on the web environment with the flask framework*

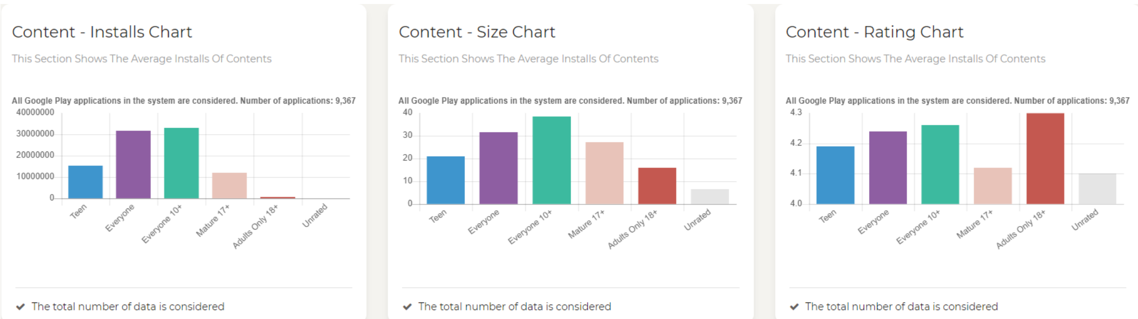


*Figure 7. Visualization of data in web application-1*

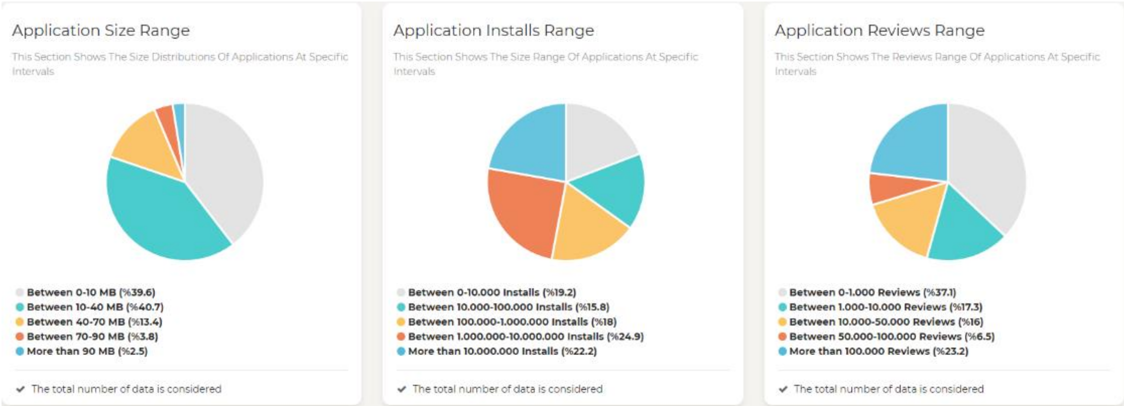


*Figure 8. Visualization of data in web application-2*

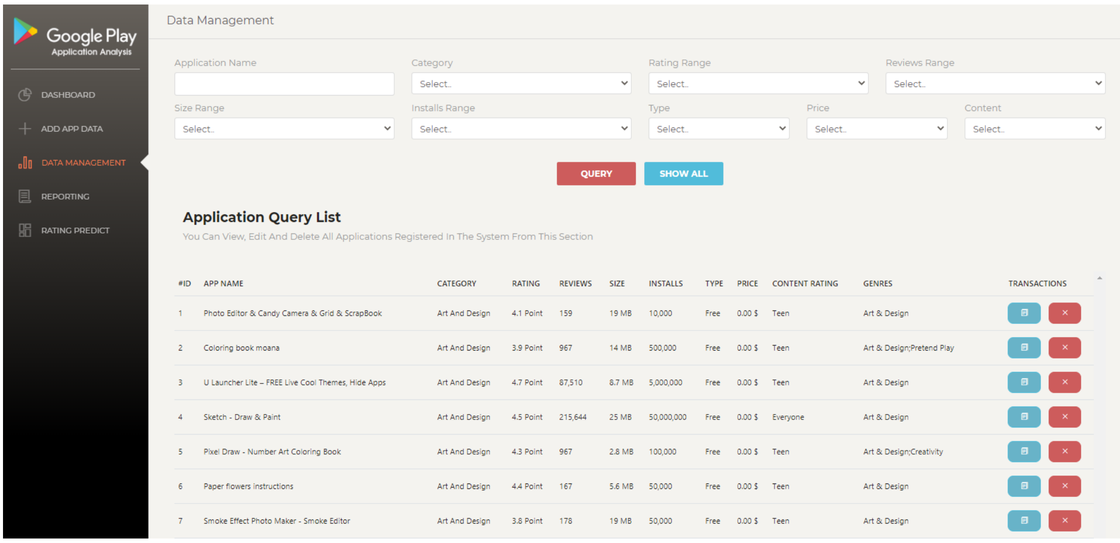


*Figure 9. Data management in web application*

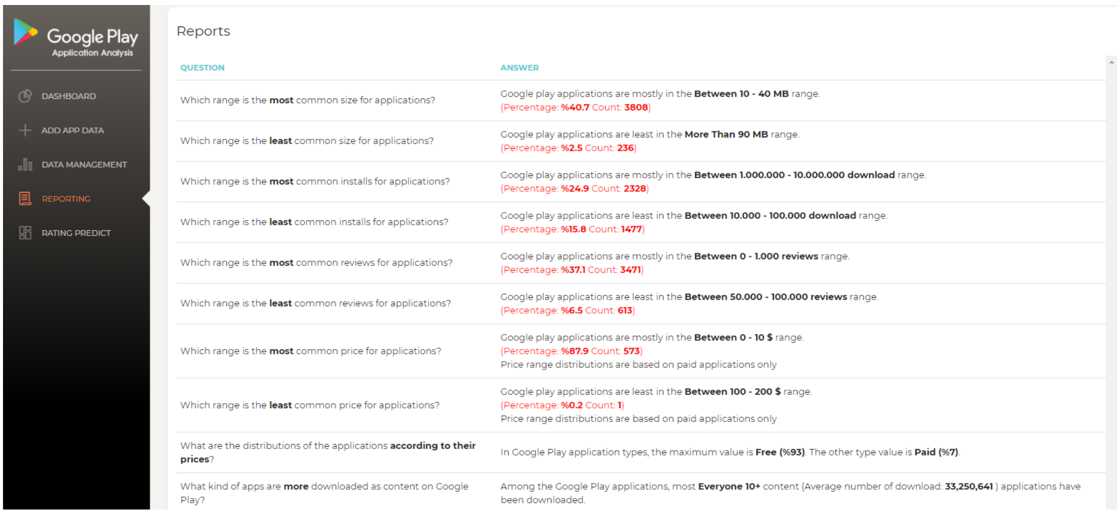


*Figure 10. Detailed reporting of mobile application data*

The mobile applications included in the platform developed by querying and the existing applications in the system are controlled via the data management page in the interface (Figure 9). Mobile application developers can get information about other applications by searching by category and other features through this page.

Finally, detailed statistics are presented to the developers on the reporting page of the developed application (Figure 10). The fact that the statistics presented is in the form of questions and answers makes the analysis processes more active. In addition, when the results on the rating predict page and the statistics on the reporting screen are combined, it is seen that information that can increase the decision support of mobile application developers is revealed. For example, a developer whose application entered the system is predicted to be unsuccessful can find the steps to take to eliminate this failure on the reporting page. In this respect, the developed application should be considered as an environment where applications are tested before being added to application markets.

## 3. Conclusions

The web-based application developed with business intelligence and machine learning methods allows mobile application developers to test and evaluate their applications. For this reason, the web platform developed is a decision support system. Because the system is designed with the inference and reporting capability of business intelligence, as well as the prediction and classification methods provided by machine learning. In this way, developers will be able to get support in determining the application features through the system developed both before and after the application development phase.

After making comparisons between this platform provided in the field of mobile application development and various methods for estimation and classification in machine learning methods, the choice of the methods that give the most accurate result is left to the user. Because a dynamic structure is created, as the application data is added to the system, the results of the algorithms may vary. However, based on the existing applications in the system, the algorithms that give the best results in rating estimation and success classification were determined. In this context, the rating estimation methods; Decision Tree Regressor (DTR), Random Forest Regressor (RFR), K-Nearest Neighbor Regressor (KNR), Adaboost Regressor (ABR) were applied on the system, and the accuracy percentages of each method were presented to the user besides the estimation made by the model. Among the algorithms used, the best rating estimates were DTR with 87% and RFR with 91%. In the classification methods, Decision Tree Classifier (DTC), Random Forest Classifier (RFC), K-Nearest Neighbor Classifier, Multi-Layer Perception Classifier (MLP), Adaboost Classifier (ABC), Gaussian Naive Bayes (GNB) were used. As a result of the model training made with application data in the system, the best classification algorithms were RFC with 87% and RFC with 89%. In addition, according to the data entered by the application developer, they were provided to test their application in the web environment as "0 = Very Unsuccessful", "1 = Unsuccessful", "2 = Successful", "3 = Very Successful". Within the scope of classification and estimation methods, entering data into the system via web application has been created in a way to strengthen the predictions to be made later. Also, since the entire data set is evaluated within the scope of business intelligence, a decision support structure can be created with dynamic reporting and inferences.

As a result, the study revealed that machine learning and business intelligence work in an integrated manner, creating a decision support structure in mobile application development. Nowadays, it is possible to increase the success factor with the study that can be applied to other fields together with many standards. For this reason, similar systems must be meticulously created and maintained.

The earlier version of the paper was presented in the 6th International Management Information Systems Conference (Kılınç et al., 2019). The previous work only revealed the accuracy values. Additionally, this study shows how this is evaluated with a confusion matrix as well as accuracy values. In this context, besides the classification metrics F1 score, sensitivity and recall values, the regression evaluation metrics R2 Score, MSE and RMSE values were also added to the study. In contrast to the previous study, classification and regression were examined in different tables in this study. Therefore, the rating predict and application classification areas in the web interface were also changed and presented to the developers.

## 4. Future Works and Recommendations

Many developers develop applications on Android because it is an open-source to improve their skills, but this should not cause the quality to decrease. Every application published on Google Play must be published after it exceeds a certain quality level. For this, there is a need for predictive and classifying systems for possible success. Thanks to these systems, both developer interest and overall application quality in Google Play will increase. Studies on mobile applications on Google Play are usually done on detecting malware. The average rating of 4.19 can be further increased by efforts to increase the quality and success of the applications. Also, a structure for predicting success can be made mandatory before the application is added to the market. With the decision support to be provided at this level, the quality of applications on Android will increase positively. Otherwise, this may cause Android, which is the most used operating system today, to become an application dump.

On the other hand, the data we use in our study constitutes a small part of the applications in Google Play. With better hardware, the system we have created with machine learning methods can be strengthened using all applications in Google Play. Besides, image processing methods can be used to increase decision support by including other features and images of the applications in the system.

## References


Aydın, C. (2018). Makine Öğrenmesi Algoritmaları Kullanılarak İtfaiye İstasyonu İhtiyacının Sınıflandırılması. *European Journal of Science and Technology*, 169–175. https://doi.org/10.31590/ejosat.458613

Birant, D. (2011). Comparison of decision tree algorithms for predicting potential air pollutant emissions with data mining models. *Journal of Environmental Informatics*, *17*(1), 46–53. https://doi.org/10.3808/jei.201100186

*Business of Apps*. (2020). App Download and Usage Statistics. https://www.businessofapps.com/data/app-statistics/

Cui, J., Wang, L., Zhao, X., & Zhang, H. (2020). Towards predictive analysis of android vulnerability using statistical codes and machine learning for IoT applications. *Computer Communications*, *155*(February), 125–131. https://doi.org/10.1016/j.comcom.2020.02.078

Das, T., Di Penta, M., & Malavolta, I. (2020). Characterizing the evolution of statically-detectable performance issues of Android apps. *Empirical Software Engineering*, *25*(4), 2748–2808. https://doi.org/10.1007/s10664-019-09798-3

Day, M. Y., & Lin, Y. Da. (2017). Deep learning for sentiment analysis on google play consumer review. *Proceedings - 2017 IEEE International Conference on Information Reuse and Integration, IRI 2017*, *2017-Janua*, 382–388. https://doi.org/10.1109/IRI.2017.79

De Mast, J., & Lokkerbol, J. (2012). An analysis of the Six Sigma DMAIC method from the perspective of problem solving. *International Journal of Production Economics*, *139*(2), 604–614. https://doi.org/10.1016/j.ijpe.2012.05.035

Dehkordi, M. R., Seifzadeh, H., Beydoun, G., & Nadimi-Shahraki, M. H. (2020). Success prediction of android applications in a novel repository using neural networks. *Complex & Intelligent Systems*. https://doi.org/10.1007/s40747-020-00154-3

Franke, D., & Weise, C. (2011). Providing a software quality framework for testing of mobile applications. *Proceedings - 4th IEEE International Conference on Software Testing, Verification, and Validation, ICST 2011*, 431–434. https://doi.org/10.1109/ICST.2011.18

García-Peñalvo, F., Conde-González, M., & Matellán, V. (2014). Mobile Apps for Older Users. The Development of a Mobile Apps Repository for Older People. *International Conference on Learning and Collaboration Technologies*, 117–126. https://doi.org/10.1007/978-3-319-07485-6

Gentner, D., Stelzer, B., Ramosaj, B., & Brecht, L. (2018). Strategic Foresight of Future B2B Customer Opportunities through Machine Learning. *Technology Innovation Management Review*, *8*(10), 5–17. https://doi.org/10.22215/timreview/1189

Gupta, L. (2020). *Google Play Store Apps Dataset*. Web Scraped Data of 10k Play Store Apps for Analysing the Android Market (Version 6). https://www.kaggle.com/lava18/google-play-store-apps

Holzer, A., & Ondrus, J. (2011). Mobile application market: A developer's perspective. *Telematics and Informatics*, *28*(1), 22–31. https://doi.org/10.1016/j.tele.2010.05.006

Huang, C. J., Liu, M. C., Chu, S. S., & Cheng, C. L. (2005). Application of machine learning techniques to Web-based intelligent learning diagnosis system. *Proceedings - HIS'04: 4th International Conference on Hybrid Intelligent Systems*, 242–247. https://doi.org/10.1109/ichis.2004.25

Huber, S., Wiemer, H., Schneider, D., & Ihlenfeldt, S. (2019). DMME: Data mining methodology for engineering applications - A holistic extension to the CRISP-DM model. *Procedia CIRP*, *79*, 403–408. https://doi.org/10.1016/j.procir.2019.02.106

Kılınç, M., Tarhan, Ç., & Aydın, C. (2019). *6. International Management Information Systems Conference "Connectedness and Cybersecurity" 09-12. October*. https://2019.imisc.net/img/proceedings_bildiriler.pdf

Lam, M. (2004). Neural network techniques for financial performance prediction: Integrating fundamental and technical analysis. *Decision Support Systems*, *37*(4), 567–581. https://doi.org/10.1016/S0167-9236(03)00088-5

Lamhaddab, K., Lachgar, M., & Elbaamrani, K. (2019). Porting mobile apps from iOS to android: A practical experience. *Mobile Information Systems*, *2019*(iii). https://doi.org/10.1155/2019/4324871

Martín, I., Hernández, J. A., & de los Santos, S. (2019). Machine-Learning based analysis and classification of Android malware signatures. *Future Generation Computer Systems*, *97*, 295–305. https://doi.org/10.1016/j.future.2019.03.006

Munoz, A., Martin, I., Guzman, A., & Hernandez, J. A. (2015). Android malware detection from Google Play meta-data: Selection of important features. *2015 IEEE Conference on Communications and NetworkSecurity, CNS 2015*, 701–702. https://doi.org/10.1109/CNS.2015.7346893

Olson, D. L., Delen, D., & Meng, Y. (2012). Comparative analysis of data mining methods for bankruptcy prediction. *Decision Support Systems*, *52*(2), 464–473. https://doi.org/10.1016/j.dss.2011.10.007

Onan, A. (2015). Şirket İflaslarının Tahminlenmesinde Karar Ağacı Algoritmalarının Karşılaştırmalı Başarım Analizi. *Bilişim Teknolojileri Dergisi*, *8*(1), 9–19. https://doi.org/10.17671/btd.36087

*Statcounter*. (2020a). Desktop vs Mobile vs Tablet Market Share Worldwide | StatCounter Global Stats. https://gs.statcounter.com/platform-market-share/desktop-mobile-tablet/worldwide

*Statcounter*. (2020b). Mobile Operating System Market Share Worldwide | StatCounter Global Stats. https://gs.statcounter.com/os-market-share/mobile/worldwide

*Statista*. (2020). Biggest App Stores in the World 2020. https://www.statista.com/statistics/276623/number-of-apps-available-in-leading-app-stores/

Stoyanov, S. R., Hides, L., Kavanagh, D. J., Zelenko, O., Tjondronegoro, D., & Mani, M. (2015). Mobile App Rating Scale: A New Tool for Assessing the Quality of Health Mobile Apps. *JMIR MHealth and UHealth*, *3*(1), e27. https://doi.org/10.2196/mhealth.3422

Üstün, D. (2019). Identification of Breast Cancer Using the Extreme Learning Machine Assisted by Firefly Algorithm. *European Journal of Science and Technology*, *17*, 637–644. https://doi.org/10.31590/ejosat.623816

Yang, T. L., Lin, C. H., Chen, W. L., Lin, H. Y., Su, C. S., & Liang, C. K. (2020). Hash Transformation and Machine Learning-Based Decision-Making Classifier Improved the Accuracy Rate of Automated Parkinson's Disease Screening. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, *28*(1), 72–82. https://doi.org/10.1109/TNSRE.2019.2950143

Zhu, H., Cao, H., Chen, E., Xiong, H., & Tian, J. (2012). Exploiting Enriched Contextual Information for Mobile App Classification. *CIKM '12: Proceedings of the 21st ACM International Conference on Information and Knowledge Management*, 1617–1621. https://doi.org/10.1145/2396761.2398484

Zimbra, D., Sarangee, K. R., & Jindal, R. P. (2017). Movie aspects, tweet metrics, and movie revenues: The influence of iOS vs. Android. *Decision Support Systems*, *102*, 98–109. https://doi.org/10.1016/j.dss.2017.08.002