**CAPSTONE FINAL REPORT**

**CLASSIFICATION PROJECT ON APPLE APPS RATINGS**

**1.0 Problem Summary**

With millions of apps with similar functionalities around nowadays, mobile app analytics is a great way to understand the existing strategy to drive growth and retention of future users. Developers and investors can rely on users’ ratings to predict whether the overall rating for an app is more than 4 stars (which is assumed to be a good app) or not. The performance of an app based on user rating should inform decision-making on version update and functionality upgrade amongst other things. Hence, the goal of this project is to predict whether the overall rating for an app is more than 4 stars or not. This project set out to classify best performing iOS applications based on users’ ratings. On a general note, ratings serve as protection for both app users and investors to guard against time and financial investments in products/services that do not perform well in the market. Hence, the goal of this project is to predict whether the overall rating for an app is more than 4 stars (1=yes, 0=no). For this project, an app with 4 or more stars is assumed to be a very good app.

**2.0 Approach Taken**

The project was addressed with a data frame sourced from Kaggle and it was gotten from the iTunes search API at the Apple Inc website. The original data frame was made up of 5197 observations with 16 features (Table 1 lists all the 15 features excluding the rating column, which is the label’s column). The labeled column has Boolean variables of 1 and 0 that represented if the applications received ratings of more than 4 or not. Going by the nature of the label’s column, it could be seen that the project was a classification problem.

Table

Description automatically generated

Table 1: List of variables in the original data frame.

The data obtained was cleaned up to prepare it for modeling. All the apps were priced in USD. As a result, the currency column was dropped as it will have little to no impact on the model. Features ‘track\_name’ and ‘desc’ were recorded in different languages, and they were processed with Natural Language (NL) processing technique using sklearn’s TfidfVectorizer. New features were generated from these two variables and the data frame columns increased from 16 to 4897. ‘track\_name’ and ‘desc’ were dropped from the data frame.

The content rating column contained only 4 unique values and they were of string data type. These

It should be noted that none of the observations had missing values in the data frame. This could be because the data was partly cleaned from the source before it was obtained for this project.

2.1 Exploratory Data Analysis

The mean size of apps in the data frame was 197 MB. The biggest was 4gb and the smallest was 0.62mb. Other variables captured in descriptive statistics of the data frame can be seen in Table 2 below.

Table

Description automatically generated

Table 2: Descriptive analysis of variables in the data frame.

The pricing of the applications in the data frame ranged from 0 to $299.99. The mean price of the application in the data frame was $1.74. Also, the least and maximum supporting devices for any app were 9 and 47 respectively, while the average was 37. Perhaps, when the data frame was cleaned, missing observations in the language column were replaced with 0. This is a possible explanation for the least number of languages supported to be 0. On average, 5 languages are supported by the apps.

Graphical user interface, website

Description automatically generated

Figure 2: Heatmap of features in the data frame.

Using Spearman's rank correlation coefficient, the heatmap above was generated. No strong relationship can be seen between any of the variables in the data frame.

values were converted to integers so that they could be used as part of features for the modeling.

There were 23 genres the apps were associated with game-based applications were the highest in the data frame, followed by entertainment and education. The least genre of apps in the data frame was catalog. The distribution of genres in the data frame can be seen in Table 1 below:

A picture containing waterfall chart

Description automatically generated

Figure 1: Prime Genre in the data frame.

Based on the bar plot below, Health & Fitness, Medical, Photo & Video, and Games rank highest amongst genres with the total number of ratings greater than 4. The second plot further shows that of all the application genres present in the data frame, game applications are 2824. Using this value, roughly half of the users gave more than 4 stars rating for the application.

Table

Description automatically generated

Table 3: Prime Genre classified based on rating.

Chart, bar chart

Description automatically generated

Chart

Description automatically generated

The data frame was grouped by rating and price and the table below was generated. 56.5% of the apps received a rating of less than 4 or less by users. The mean price of apps that received ratings above 4 stars was $0.1 higher than those with a lesser rating. However, the prices of apps with lesser than 5 stars are more widespread based on their standard deviation. The most expensive application of the data frame had a rating of 0 (less than or equal to 4 stars). From the box plot below, based on price, more of the applications received favorable ratings of above 4. Also, applications with favorable ratings tend to be more expensive in general compared to those with lower ratings.

Graphical user interface, application

Description automatically generated

Due to the extreme prices of applications distorting the distribution of the data frame, extreme observations based on prices were dropped. As a result of this, the total number of observations was reduced to 4605. As a result of this, the box plot above was generated.

In terms of the supported number of languages per application, applications with a higher number of supporting languages had more favorable ratings higher than 4 stars. This is depicted in the image below.

Chart, box and whisker chart

Description automatically generated

As captured in the table below, applications with a rating of 1 had more users give ratings to such applications. This seems intuitive as consumers that enjoy using a service/product tend to willingly recommend such products to other consumers.  
Table

Description automatically generated

Table 4: Descriptive statistics between rating and total rating counts.

The values of the prime\_genre column were converted to dummy variables. This further increased the total number of features to 4918.

Also, due to the magnitude of values of 'size\_bytes', 'rating\_count\_tot', 'rating\_count\_ver', and 'sup\_devices.num', the values of these features were standardized, and the new values were added to the data frame.

For the modeling purpose, the data frame was split into training and testing data set in the ratio of 75% to 25% respectively.

2.2 Modeling:

**This is a classification problem, in supervised learning. Here we have used the following classification models:**

* Logistic Regression
* K-Nearest Neighbor (KNN)
* Support vector machine (SVM)
* Random Forest
* Naive Bayes
* Gradient Boost

Evaluating the performance of a model by training and testing on the same dataset can lead to overfitting. Hence the model evaluation is based on splitting the dataset into train and validation set as done above. But the performance of the prediction result depends upon the random choice of the pair. To overcome that, the Cross-Validation procedure is used where under the k-fold CV approach, the training set is split into k smaller sets, where a model is trained using k-1 of the folds as training data and the model is validated on the remaining part.

**Classification/ Confusion Matrix:**

This matrix summarizes the correct and incorrect classifications that a classifier produced for a certain dataset. Rows and columns of the classification matrix correspond to the true and predicted classes respectively. The two diagonal cells (upper left & lower right (TP & TN respectively)) give the number of correct classifications, where the predicted class coincides with the actual class of the observation. The off-diagonal cells give the count of the misclassification (upper right & lower left (FP & FN respectively)). The classification matrix gives estimates of the true classification and misclassification rates.

* Logistic Regression: the accuracy score using a grid search was 59.98%

|  |  |
| --- | --- |
| TP: 433 | FP: 223 |
| FN: 255 | TN: 241 |

The precision rate of this classification is 66%

The recall rate of the classification is 62.9%

Using cross-validation score, the mean test and train scores are 57.24% and 64.48% respectively.

* K-Nearest Neighbor: the accuracy score using grid search was 57.90%

|  |  |
| --- | --- |
| TP: 644 | FP: 12 |
| FN: 473 | TN: 23 |

The precision rate of this classification is 98%

The recall rate of the classification is 57.65%

Using the cross-validation score, the mean test and train scores are 57.03% and 63.76% respectively.

* Support Vector Machine: the accuracy score using a grid search was 59.11%

|  |  |
| --- | --- |
| TP: 451 | FP: 205 |
| FN: 266 | TN: 230 |

The precision rate of this classification is 68.75%

The recall rate of the classification is 62.90%

Using the cross-validation score, the mean test and train scores are 56.46% and 66.71% respectively.

* Random Forest: the accuracy score using a grid search was 69.97%

|  |  |
| --- | --- |
| TP: 513 | FP: 143 |
| FN: 203 | TN: 293 |

The precision rate of this classification is 78.20%

The recall rate of the classification is 71.64%

Using the cross-validation score, the mean test and train scores are 73.17% and 79.81% respectively.

* Naïve Baise: the accuracy score using a grid search was 50.87%

|  |  |
| --- | --- |
| TP: 287 | FP: 369 |
| FN: 197 | TN: 299 |

The precision rate of this classification is 43.75%

The recall rate of the classification is 59.30%

Using the cross-validation score, the mean test and train scores are 51.40% and 57.69% respectively.

* Gradient Boosting: the accuracy score using a grid search was 69.97%

|  |  |
| --- | --- |
| TP: 508 | FP: 148 |
| FN: 195 | TN: 301 |

The precision rate of this classification is 77.44%

The recall rate of the classification is 72.26%

Using the cross-validation score, the mean test and train scores are 73.25% and 79.52% respectively.

Model Comparisons

Table

Description automatically generated

From the table above, it can be seen that Random Forest and Gradient Boost have the best scores from the 6 models used for classification. The Naive Bayes model is the worst performed of the models. The performances based on their accuracies are depicted in the graphs below.

Chart, bar chart

Description automatically generated

The hyperparameter tuning of the Random Forest algorithm improved its score from 69.97% to 71.00%. On the other hand, the most important features of the data frame were the engineered features. The ROC-AUC score for the Random Forest Algorithm is 0.7797

Chart, line chart

Description automatically generated

The hyperparameter tuning of the Gradient Boost algorithm reduced its accuracy score from 70.23% to 69.97%. On the other hand, the most important features of the data frame were the engineered features. The ROC-AUC score for the Gradient Boost algorithm is 0.7656.

Chart, line chart

Description automatically generated

A comparison of the two ROC-AUC scores of the two algorithms shows that Random Forest performs better than Gradient Boost in the classification of the test data set.

**3.0 Ideas for Future Research:**

An idea for further research on this topic is to investigate the impact of prices on application retention among similar applications.

**4.0 Recommendations**

1. Based on the best performing model in the data set, investors can compare apps and predict the possibility of their apps gaining traction in the market and giving a good return on investment in the long run.
2. Apps platforms can use the model to rank and suggest high performing apps to users. Time spent by consumers trying to reduce similar applications will be reduced through this rating ranking system.
3. For apps’ owners and apps platforms (app store), revenue generation can be properly structured to mirror the performance of the apps based on customer ratings.