Google Play Store Apps

COMP-7745
Machine Learning
Spring 2023

Akhila Bachu
#U00884957
Contact Email: abachu@memphis.edu

Sainath Reddy Addula
#U00865655
Contact Email: Saddula1@memphis.edu

Table of Contents

Executive Summary:

The main goal of the paper is to predict the number of downloads an application(app) will have on the Google Play Store. Google play store, the Google's digital distribution service in this era is the world's largest app store. Google play store is hosting around 3 million plus apps, this record is based on year 2020. Android app store boosts 2.5 billion active android users. Google app store has become a brand itself with around 2+ billion active monthly users. To meet the traffic of the app download from Google play store. To predict the result the downloads of apps from Google Play Store different machine learning algorithms are developed in this project. Different machine learning algorithms are used like Random Forest, Gradient Boosting Classifier, KNeighbour Classifier, Voting Classifier. This project completely focuses on Data cleaning, EDA (Exploratory Data Analysis), Feature Engineering and Machine learning model building.

Project Motivation:

In this fast-moving world, people are expecting things to be done from a single click. All their requirements to be addressed at their foot step. From gamming to business, education to entertainment almost all categories have respective apps which are already helping people in their own way. There are different Android app provides where Google Play Store is world's largest app provider of around 2 million plus apps. It is a challenge for new app developers to find out which apps are trending and the apps that are most wanted by the people to satisfy their requirements in a better way.

Key Questions

1. What kind of apps being downloaded through Google Play Store?
2. What kind of apps are being installed per Install Bracket?
3. How many numbers of apps that already exist per a particular category through which a developer can analyze if he/ she can start building an app over it or not?

These questions are answered in the report with the help of data visualization techniques and machine learning algorithms.

Data Source:

The data file was downloaded from the Kaggle website, the link is shown below,

(https://www.kaggle.com/datasets/gauthamp10/google-playstore-apps?select=Google-Playstore-Full.csv). The datafile contains more than 2K+ rows and 11+ columns where columns represent different features of apps like

- App Name
- App Id
- Category
- Rating
- Rating Count
- Installs
- Minimum installs
- Maximum Installs
- Price
- Content Rating
- Last Updated
- Minimum Version
- Latest Version

Data Description:

The current datafile consists of different features of the Google play store apps. The datafile consists of varied features as mentioned above.

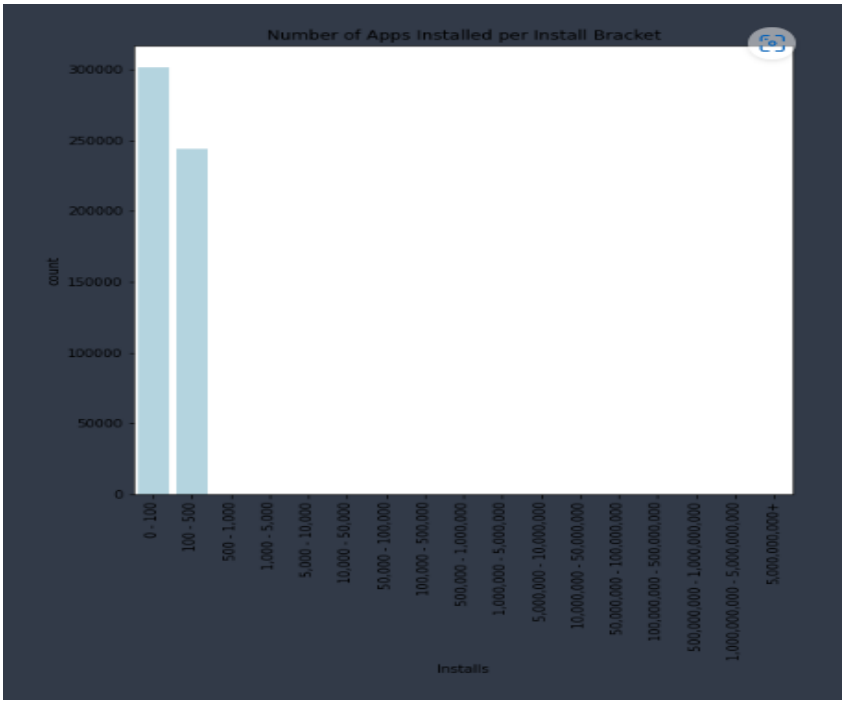
To work on the datafile required python libraries are imported in the beginning, the datafile is initially downloaded and stored in the drive, later on its path is added to the workspace. The datafile is read through the code read.csv command. The column heading can be known using the command DataFile.head() command which is used to display the first five rows of the datafile. Datafile.Shape command is used to display number of column entities in the datafile. Datafile.info() command is used to display the complete information of the datafile columns. The null value is checked at this point for all the missing values in the data file using the command datafile.isnull().sum() command is used.

Exploratory Data Analysis:

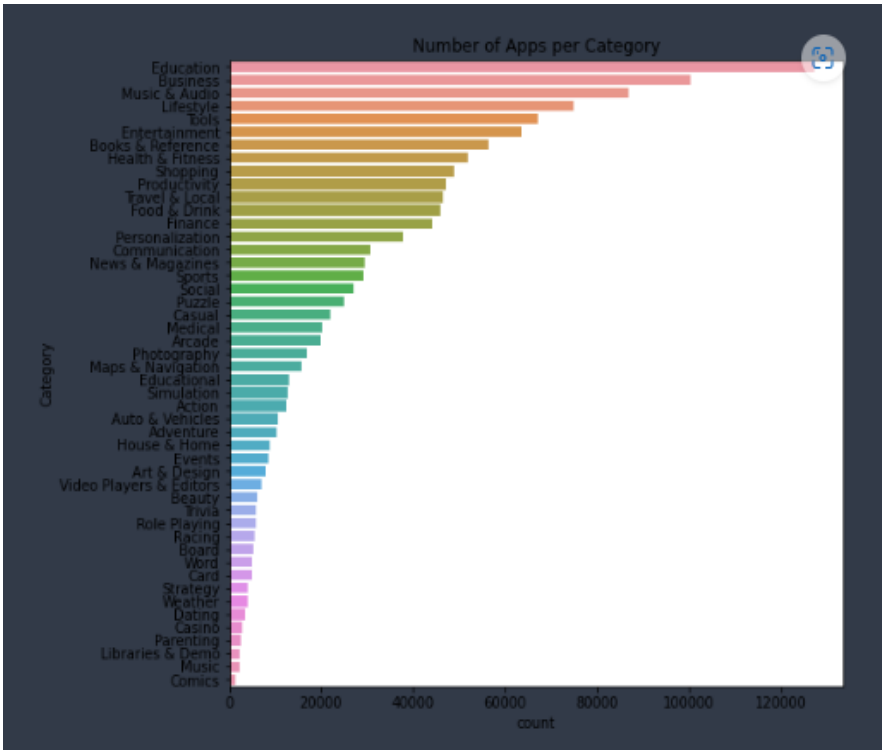
The main aim of the project is to provide a detailed visualization or visual analysis of the Google Play Store to see the relation between the insight and features. To reach the final product first and foremost the process starts with exploring and analyzing the datafile. Initially the mean and median of different categories are found out like Average rating per Category, Average app size (Megabytes) per Category, Number of Installs per Install Bracket

The data analysis is done for above three points and now let us look into the visualization part of these.

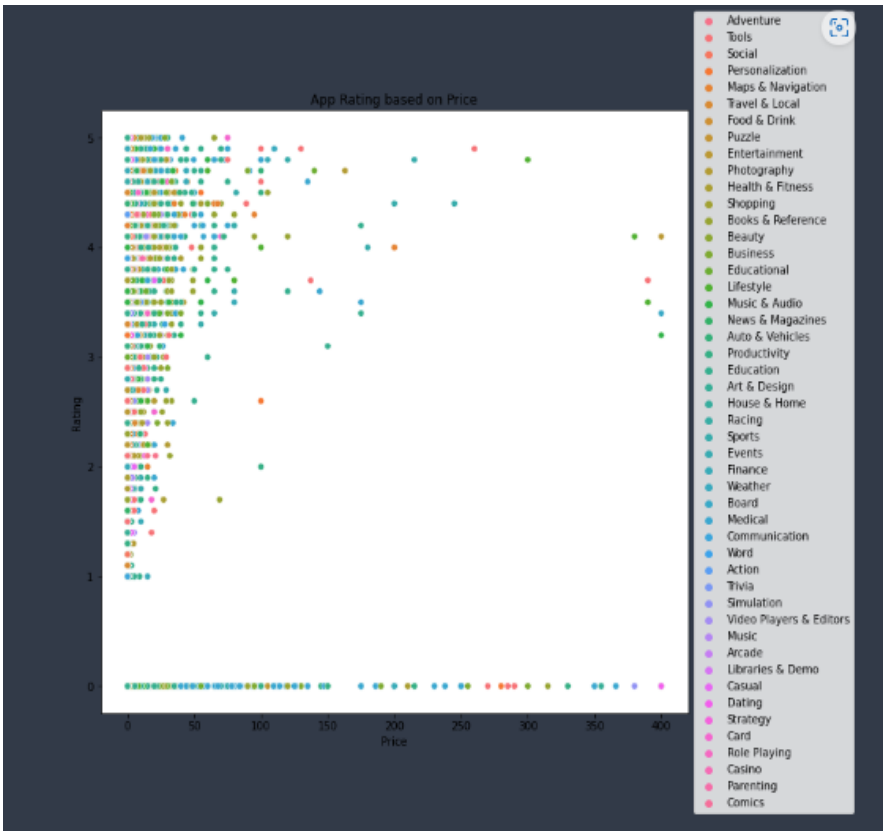
- Number of Apps Installed per Install Bracket:



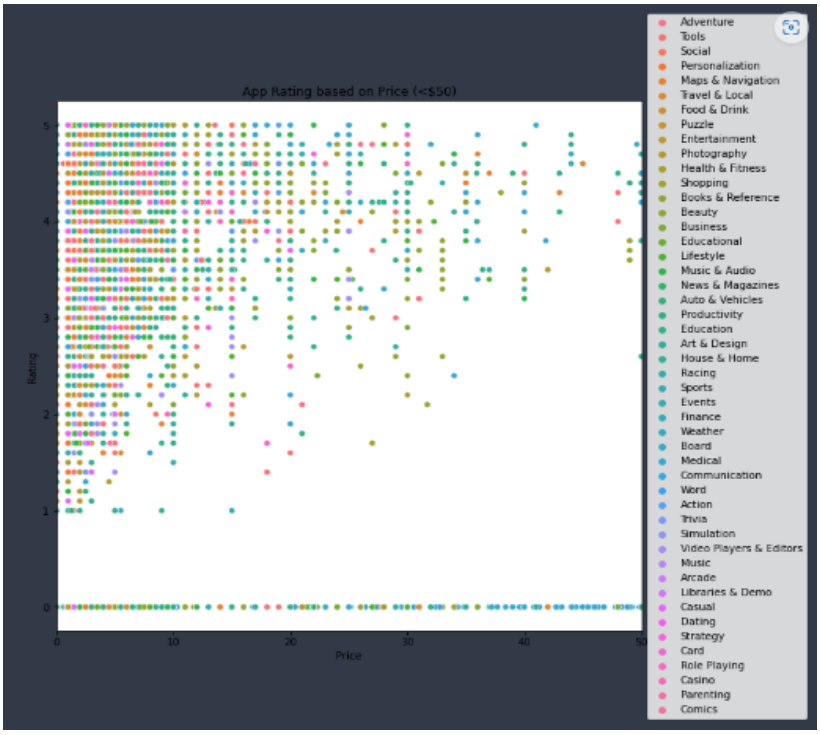
- Number of Apps per Category



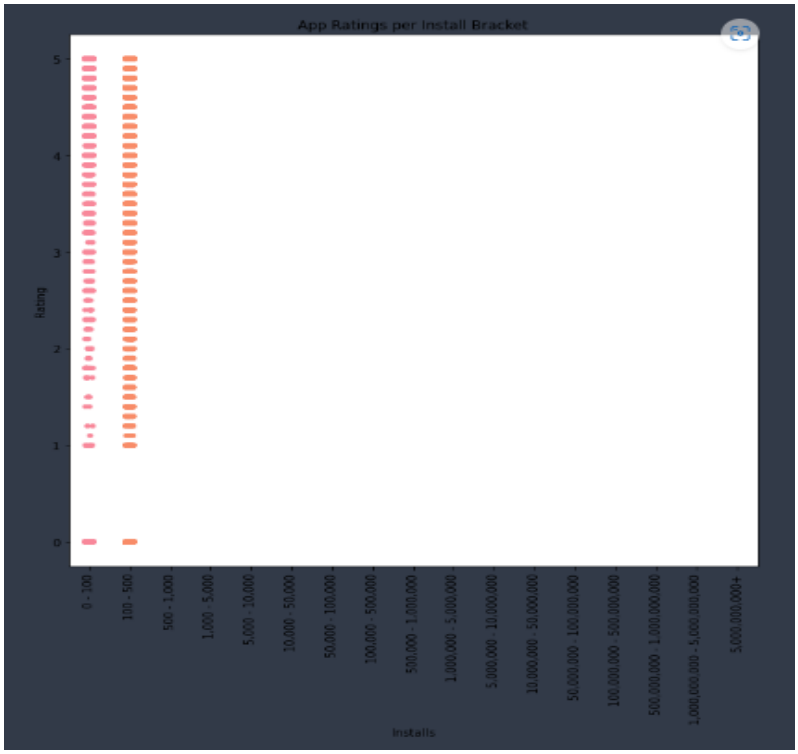
- App rating based on size



- App rating based on Pricing model

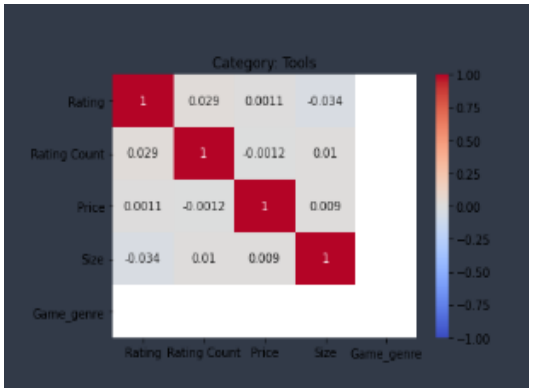


- App rating based on Install bracket



These are the data analytic points carried out in this project and now let us look into the correlation matrix to check for multicollinearity for the entire Data frame.

Now let us see the correlation between the top 5 categories i.e., Education, Business, Music & Audio, Lifestyle, Tools.



Machine Learning Algorithms:

There are many different types of machine learning algorithms in real time, which have been designed in dynamic times to solve many real time or real-world complex problems. The machine learning algorithms are smart enough well automated and self-modifying to continue to improve themselves over the period of time.

Machine learning algorithms are very basically divided into four types Supervised, Unsupervised, Semi-supervised and reinforcement learnings.

- **Linear Regression:** An algorithm that establishes relationship between independent and dependent variables by fitting them into a line. This line is called as regression line.
- **Logistic Regression:** An algorithm to estimate discrete values from a set of independent variables. Helps to predict probability of an event by lifting data to logit function.
- **Decision Tree:** This is a supervised algorithm which is used in classifying through split the population into two or more homogeneous sets based on the most significant independent variable or the attribute.
- **SVM algorithm:** Support Vector Machine algorithm, plotting is done using raw data in n-dimensional space. The value of feature is tied to the coordinates which makes classification easier.
- **Naïve Bayes Algorithm:** It is a classifier algorithm that assumes the presence of particular feature in a class, which is unrelated to the presence of any other feature.
- **KNN Algorithm:** It can be applied to both classification and regression. It stores all available cases and classifies new cases by taking majority vote of its K neighbor.
- **Random Forest Algorithm:** A collective of decision trees is called as Random Forest. To classify an object based on its attribute, each tree is classified and the tree votes for that class. The forest chooses the one with majority voting.
- **K-Means:** It is an unsupervised machine learning algorithm, where datasets are classified into a particular number of clusters in a pattern so that all datapoints in the cluster are homogeneous and heterogeneous from the data in other clusters.
- **Dimension Reduction Algorithm:** This Dimension reduction algorithms like Decision tree, Factor Analysis, Missing value ratio and Random Forest can help in finding relevant details

- **Gradient Boosting and Adaboosting algorithm:** Different boosting algorithms are used when massive loads of data have to be handled to make predictions with very high or perfect accuracy

Models and Analysis:

The main aim of the project, to build a model that predicts a meaningful and accurate result for young and aspiring app creators to know weather the features are the most important when maximizing installs.

Some of the test data in the datafile is restructured like dropping the null values, drop few features, dividing the datafile into 20-80 test-train data.

The target labels are breakdowns and count were as shown below

- Installs Brackets
- 0-1000
- 1,000-10,000
- 10,000-100,000
- 1,000,000-10,000,000
- 10,000,000-100,000,000
- 100,000,000-1,000,000,000
- 1,000,000,000+

The data is judged based on the measure of Accuracy.

To build perfect model for this project, the data needs to be prepared well for the above analysis. The complete datafile needs to be divided into test and train data. Generally, this split goes in 20-80 percentage i.e., 20 for testing and 80 for training. The training data is taken more so that the model can be trained for number of combinations. After the datafile is split for train and test data different models are applied on the machine learning code.

Decision Tree Classifier:

This Decision tree classifier is used as the baseline for the model

```
dt = DecisionTreeClassifier(max_depth=8, max_features='sqrt')
dt.fit(X_train, y_train)
y_pred = dt.predict(X_train)

acc = accuracy_score(y_train, y_pred)
print("Decision Tree train data accuracy: {:.2f}".format(acc))
```

Random Forest Classifier:

Random forest was chosen due to its ability to handle multi-classification problems and its reliability

```
rf = RandomForestClassifier(max_depth=20, n_estimators=100)
rf.fit(X_train, y_train)
y_pred = rf.predict(X_train)

acc = accuracy_score(y_train, y_pred)
print("Random Forest Train data accuracy: {:.2f}".format(acc))
```

Gradient Boosting Classifier

Gradient Booster was used in measurement of different accuracies between Random Forest and a more complex and intensive computing model.

```
gbm = GradientBoostingClassifier(learning_rate=0.05, max_depth=8, n_estimators=400, verbose=1)
gbm.fit(X_train, y_train)

y_pred = gbm.predict(X_test)

acc = accuracy_score(y_test.astype(str), y_pred)
print("Gradient Boosting Classifier Test data accuracy: {:.5f}".format(acc))
```

KNeighbor classifier

K nearest is the another baseline used as reference to understand what accuracy the model is reaching to.

```
knn = KNeighborsClassifier()
knn.fit(X_train, y_train)
print('knn train data accuracy', knn.score(X_train, y_train))
print('knn test data accuracy', knn.score(X_test, y_test.astype(str)))
print('knn cross validation accuracy', np.mean(cross_val_score(knn, X, y, cv=5)))
```

Finding and Results:

S.no	Classifier Algorithm	Accuracy
1	Decision Tree Classifier	**0.63**
2	K Nearest Neighbor	**0.7674**
3	Random Forest	**0.79157**
4	Gradient Boosting Classifier	**0.79958**

Conclusion:

In this project, we have performed data cleaning on google play store dataset and we have saved the cleaned data. Also, performed exploratory data analysis on the cleaned data and trained with four different models where Gradient Boosting classifier gives highest accuracy of 79.99.

References:

[1] Kaggle.com. (2018). Google Play Store Apps.[online]https://www.kaggle.com/lava18/google-play-store-apps [Accessed 3 Mar. 2020].

[2] "Mining and Analysis of Apps in Google Play," Pro-ceedings of the 9th International Conference on WebInformation Systems and Technologies, 2013.

[3] Google play store number of apps [online] https://www.statista.com/statistics/266210/number-of-available-applications-in-the-google-play-store/ [Accessed 3 Mar. 2020].

[4] Amit Chile, Dr. P. R. Gundalwar.(2019). Anal-ysis of Google Play Store Application.[online]http://ijraset.com/fileserve.php?FID=24134 [Accessed3 Mar. 2020]

[5] T. Denoeux, M. Skarstein-Bjanger, Induction of deci-sion trees from partially classified data, in: Proceedingsof the 2000 IEEE International Conference on Systems,Man and Cybernetics (SMC'00), IEEE, Nashville, TN,2000, pp. 2923–2928.

[6] Harman, M., Jia, Y., and Zhang, Y. (2012). App store mining and analysis: Msr for app stores. In 2012 9thIEEE Working Conference on Mining Software Repos-itories (MSR),pages 108–111.

[7] R. P. Rajeswari, K. Juliet, and Aradhana, "Text Classification for Student Data Set using NaiveBayes Classifier and KNN Classifier," Int. J. Com-put. Trends Technol., vol. 43, no. 1, pp. 8–12, 2017.https://doi.org/10.14445/22312803/ijctt-v43p103

[8] Jong, J. (2011). Predicting rating with sentiment anal-ysis. [online] http://cs229.stanford.edu/proj2011/Jong-PredictingRatingwithSentimentAnalysis.pdf.

[9] [2015].Grover, S. 3 apps that failed (and whatthey teach us about app marketing). [online]https://blog.placeit.net/apps-fail-teach-us-app-marketing/.

[10] H. G. Schnack, M. Nieuwenhuis, N. E. van Haren,L. Abramovic, T. W. Scheewe, R. M. Brouwer, H. E.Hulshoff Pol, and R. S. Kahn, "Can structural MRI aidin clinical classification? A machine learning study intwo independent samples of patients with schizophre-nia, bipolar disorder and healthy subjects," NeuroIm-age, vol. 84, pp. 299–306, jan 2014