Laboratory Practice 1 :
Data Analytics Mini Project
Data Analysis Of Google App's Rating

Project ID: 13

Project created by :
Prathamesh Chaudhari (50)
Sanket Sutar(49)
Rupesh Deshmukh(52)
Piyush Patil(51)

Project Guide: Prof. N. G. Sharma

December 14, 2020

Contents

List of Figures

Chapter 1

Problem Statement

Predicting google app rating with the help of machine learning algorithms, so that we can derive business insights which will be useful for development of the application. To solve this problem we can use data visualization tools and perform data analysis on given data, to get insights as fast as possible and without much manual work. It can give accurate insights.

.

Chapter 2

Objectives

- To do data pre-processing on google app's dataset (Removal of Missing data and outlier).
- To do data analysis and visualization on dataset.
- To predict google app rating with the help of KNN and Random Forrest.

Chapter 3

Data Preprocessing

Preprocessing is important into transitioning raw data into a more desirable format. Undergoing the preprocessing process can help with completeness and compellability. For instance, you can see if certain values were recorded or not. Also, you can see how trustable the data is. It could also help with finding how consistent the values are. We need preprocessing because most real-world data are dirty. Data can be noisy i.e. the data can contain outliers or just errors in general. Data can also be incomplete i.e. there can be some missing values.

Data storage is getting easier and easier, most organizations are storing a bulk of data, for various purposes like auditing, information, prediction etc. Which turns out to be beneficial to the organization. But at the same time, bulk data said as big data increases the possibility of noisy data. Also, as this data would be coming from different sources, there is a higher chance of inconsistency. Even in smaller data set there are missing entries and inconsistent formats. All these anomalies hamper the cause of storing data, it can cause low quality mining, the prediction analysis might get drifted to the wrong side. While violets the cause of storing data, hence it is important to make the data as required which we say as preprocessing. This brings in data accuracy, completeness, consistency, timeliness, believability and interpretability

Chapter 4

Algorithms

- K Nearest Neighbor
- Random forest Algorithm

4.1 K Nearest Neighbor

- The k-nearest neighbors algorithm (k-NN) is a non-parametric method used for classification and regression
- K-Nearest Neighbour is one of the simplest Machine Learning algorithms based on Supervised Learning technique.
- K-NN algorithm can be used for Regression as well as for Classification but mostly it is used for the Classification problem.
- K-NN is a non-parametric algorithm, which means it does not make any assumption on underlying data.
- KNN algorithm at the training phase just stores the dataset and when it gets new data, then it classifies that data into a category that is much similar to the new data.

4.2 Random Forest Algorithm

- Random Forest is a classifier that contains a number of decision trees on various subsets of the given dataset and takes the average to improve the predictive accuracy of that dataset.

- Instead of relying on one decision tree, the random forest takes the prediction from each tree and based on the majority votes of predictions, and it predicts the final output. decision forests correct for decision trees' habit of overfitting to their training set .

- Random forests are frequently used as "blackbox" models in businesses, as they generate reasonable predictions across a wide range of data while requiring little configuration in packages such as scikit-learn.

Chapter 5

Screenshot of Output

```
In [7]: # Exploring missing data, checking to see if any has NaN values
        plt.figure(figsize=(7, 5))
        sns.heatmap(df.isnull(), cmap='viridis')
        df.isnull().any()

Out[7]: App              False
        Category         False
        Rating            True
        Reviews          False
        Size             False
        Installs         False
        Type              True
        Price            False
        Content Rating    True
        Genres           False
        Last Updated     False
        Current Ver       True
        Android Ver       True
        dtype: bool
```

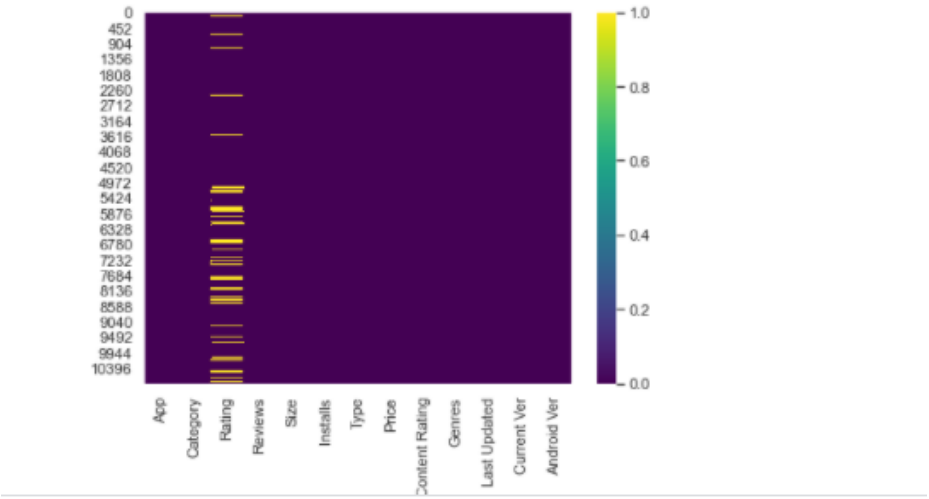

Figure 5.1: Exploring missing data

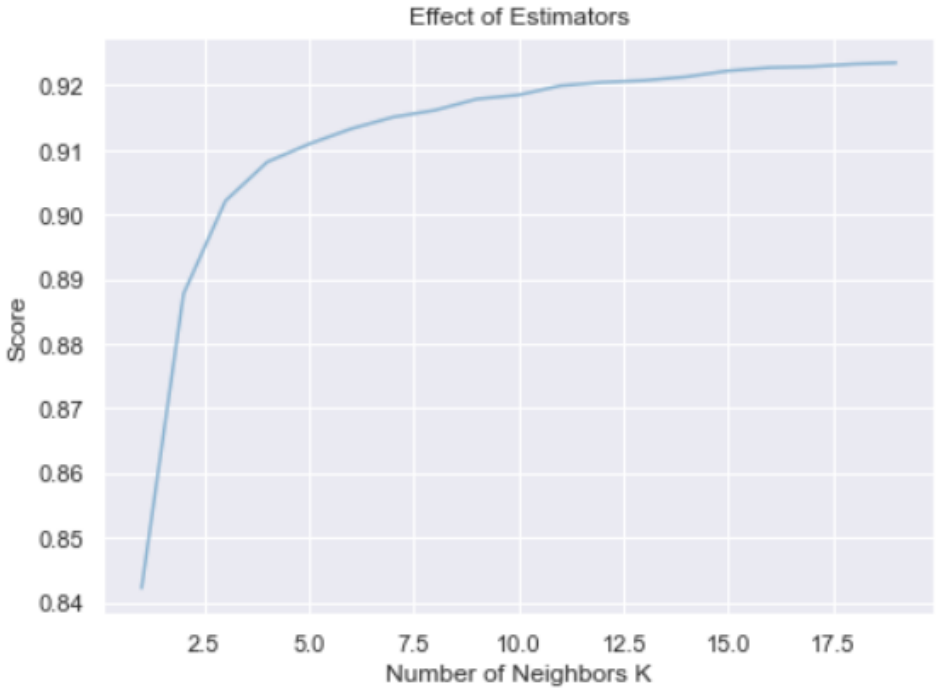

Figure 5.2: Number of Neighbors vs Score

```
In [36]: # Calculate the mean accuracy of the KNN model
         accuracy = model.score(X_test,y_test)
         'Accuracy: ' + str(np.round(accuracy*100, 2)) + '%'

Out[36]: 'Accuracy: 92.22%'
```

Figure 5.3: Accuracy of KNN algorithm

Chapter 6

Outcomes

- Google app's data is analyzed and application's rating and performance is also predicted.
- Visualized input data and understood the relation between them.
- KNN algorithm works best in our case with 92% accuracy.

Chapter 7

Conclusion

After undergoing KNN, Random Forest algorithms and process, we concluded that our hypothesis is true. Meaning you can predict the app ratings, however significant preprocessing must be done before you start the classification and regression processes.

The Play Store apps data has enormous potential to drive app-making businesses to success. Actionable insights can be drawn for developers to work on and capture the Android market! This shows that given the Size, Type, Price, Content Rating, and Genre of an app, we can predict about 91% accuracy.