A PROJECT REPORT ON

GOOGLE PLAY STORE DATA ANALYSIS AND RATING PREDICTION

BY

Mayuri Shivshette **Roll No-1327**

Vaibhavi Naphade **Roll No -1349**

Under the Guidance of

Mr. AKSHAY TILEKAR

POST GRADUATE DIPLOMA IN BIG DATA ANALYTICS FEB-2020

INSTITUTE FOR ADVANCED COMPUTING AND SOFTWARE DEVELOPMENT, AKURDI, PUNE

CERTIFICATE

This is to certify that the Project Entitled

GOOGLE PLAY STORE DATA ANALYSIS AND RATING PREDICTION

Submitted by:

Mayuri Shivshette **Roll No-1327**

Vaibhavi Naphade **Roll No -1349**

is a bonafide student of this institute and the work has been carried out by him/her under the supervision of Mr. AKSHAY TILEKAR and it is approved for the partial fulfillment of the requirement of Post Graduate Diploma in Big Data Analytics.

Mr. Prashant Karhale
Centre Coordinator

Mr. Akshay Tilekar
Project Guide

ACKNOWLEDGEMENT

It gives us great pleasure in presenting the preliminary project report on ‘GOOGLE PLAY STORE DATA ANALYSIS AND RATING PREDICTION’.

I would like to take his opportunity to thank my internal guide Mr. AKSHAY TILEKAR for giving me all the help and guidance I needed. I am really grateful to them for their kind support. Their valuable suggestions were very helpful.

I am also grateful to Mr. PRASHANT KARHALE, Centre Coordinator, Akurdi , Pune for his indispensable support, suggestions. In the end our special thanks to IACSD for providing various resources such as laboratory with all needed software platforms, continuous Internet connection, for our Project.

MAYURI SHIVSHETTE

VAIBHAVI NAPHADE

INDEX
Table of Contents

List of Figures and Tables

Chapter-1

Introduction

1.1 Introduction

The ability to use services and products on the go has been increased from past few years. Applications on the Google play store aim to do exactly that. As it has provided worldwide accessibility and the ease of use, it has not only become the most popular application download destination but also a hotbed for competing services to attract and gain customers.

Application (App) ratings are feedback provided voluntarily by users and function important evaluation criteria for apps. However, these ratings can often be biased due to insufficient or missing votes

This project aims to employ machine learning & visual analytics concepts to gain insights into how applications become successful and achieve high user ratings, to predict the ratings of Google Play Store apps, various machine learning Algorithms has been used to perform Data Analysis and prediction into the Google Play store application dataset which has been collected from Kaggle.

Using Machine Learning Algorithms, discover the relationships among various attributes present in dataset such as which application is free or paid, about the rating of the application etc.,

So in order to Predict the rating of application we have used various Machine learning algorithms which will analyze the each feature from given dataset and taken each point into consideration it will test the model and predict the rating of the application which will be lunched on Play Store.

1.1.1 Analysis and Prediction of Google Play store Apps

In today's scenario we can see that mobile apps playing an important role in any individuals life. It has been seen that the development of the mobile application advertise has an incredible effect on advanced innovation

With enormous challenge from everywhere throughout the globe, it is basic for a designer to realize that he is continuing in the right heading. To hold this income and their place in the market the application designers may need to figure out how to stick into their present position.

Hence, we are analyzing the different features from the application and their relationship with various other features from the application, to understand the pattern formed in the features. Considering the new discoveries we will predict the Rating of the new applications. To achieve this prediction we have used different Machine learning concepts and algorithms and find the accuracy of the various models and at last we will take into consideration only that model which gave us the best accuracy.

Below are the steps for Rating prediction and Data Analysis

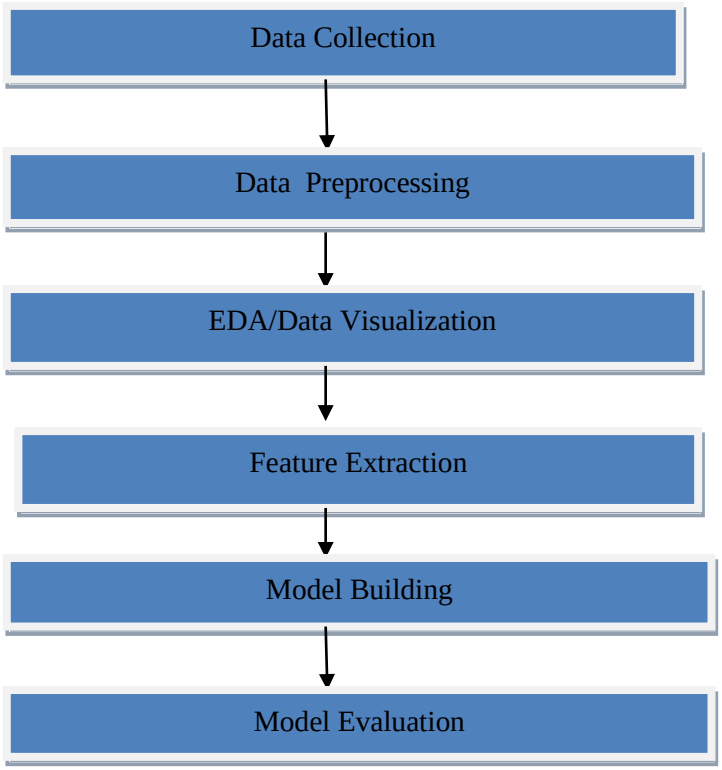

Fig.1 Steps for Rating analysis

1.1.2 Google Play store Data

The dataset taken is of Google play store application and is taken from Kaggle. This dataset is for Web scratched information of Play Store applications to analyze the market of android. This data set contains 1118136 rows and 23 columns which are Category, Installs, Size, Rating, Content Rating etc.

With the help of this dataset we will examine various qualities like rating, free or paid and so forth utilizing ML concepts and after that we will likewise do forecast of various traits like rating etc.

1.2 Problem Statement

In this project we have focused on our 2 objectives. We have taken the dataset and observed it nicely and as per our need we have taken various attributes to analyze and further display the result. By doing this, we can clearly and easily observe the dataset. Moreover, Firstly, we will analyze different attributes given in dataset. Secondly, I will do prediction of those different attributes like predict user Rating on the application

1.3 Objective

- The main goal of this project is to analyze different attributes of given application like application name, category, rating, reviews, size, installs, type, price, content rating, last updated, current version, android version. And to find out the most rated and most reviewed apps and also to distinguish between the apps which are either free or paid using Python technology.

- The second Objective is to predict whether the user review for different application using ML Algorithms

1.4 Scope of Project

The Implementation of this Project is divided into 2 parts:

1.4.1 Analysis

In this various attributes like application name, category, rating, reviews, size, installs, type, price, content rating, genres, last updated, current version, android version are analyzed using Python.

Steps for implementing the analysis part:

- First we have selected the dataset.
- Then that information is load into Jupyter notebook in dataframe using python .
- Then in we have check the data in python
- And perform Pre-processing

In [1]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
```

In [2]:
```
#Load csv file
df=pd.read_csv("G:\project\Google-Playstore.csv")
df.head(n=5)
```

Out[2]:

	App Name	App Id	Category	Rating	Rating Count	Installs	Minimum Installs	Maximum Installs	Free	Price	...	Developer Id	Develop
0	HTTrack Website Copier	com.httrack.android	Communication	3.6	2848.0	100,000+	100000.0	351560	True	0.0	...	Xavier Roche	http://www.
1	World War 2: Offline Strategy	com.skizze.wwii	Strategy	4.3	17297.0	1,000,000+	1000000.0	2161778	True	0.0	...	Skizze Games	http://s
2	WPSApp	com.themausoft.wpsapp	Tools	4.2	488639.0	50,000,000+	50000000.0	79304739	True	0.0	...	TheMauSoft	http://www.then

```
In [7]: #Checking Duplicate Values in App Name
        duplicate=df[df.duplicated('App Name')]
        duplicate.count()
        #duplicate
```

```
Out[7]: App Name            71266
        Category            71266
        Rating              71066
        Rating Count        71066
        Installs            71264
        Minimum Installs    71264
        Maximum Installs    71266
        Free                71266
        Price               71266
        Size                71266
        Released            70992
        Last Updated        71266
        Content Rating      71266
        In App Purchases    71266
        dtype: int64
```

1.4.2 Prediction

- Initially Features has been decided and few of the feature is extracted from the dataset :

```
In [29]: #Extracting new feature Days from Released Date and Updated Date
         Days=df['Last Updated']-df['Released']
         print(Days)
```

```
0         861 days
1         641 days
2         628 days
3         775 days
4         155 days
             ...
663885    107 days
663886    326 days
663887    106 days
663888    353 days
663889    945 days
Length: 663890, dtype: timedelta64[ns]
```

```
In [30]: df['Days']=Days
```

```
In [31]: df['Days'].astype('timedelta64[D]')
```

```
Out[31]: 0         861.0
         1         641.0
         2         628.0
         3         775.0
         4         155.0
```

- We have broken the dataset into training and testing data. The training input data and testing input data is dummied

In [44]:
```
#Get Dummies for Category, Free, Content Rating columns
catgry=pd.get_dummies(df['Category'],prefix='catg',drop_first=True)
typ=pd.get_dummies(df['Free'],prefix='typ',drop_first=True)
cr=pd.get_dummies(df['Content Rating'],prefix='cr',drop_first=True)
frames=[df,catgry,typ,cr]
df=pd.concat(frames,axis=1)
df.drop(['Category','Free','Content Rating'],axis=1,inplace=True)
```

:
```
X=df.drop('Rating',axis=1)
y=df['Rating'].values
y=y.astype('int')
```

:
```
#Train Test Split
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=2020)
```

CHAPTER 2
Overall Description

2.1 Description

Our aim from the project is to make use of pandas, matplotlib, & seaborn libraries from python to extract insights from the data and Decision tree classifier, Random forest, Naive Bayes, Logistic Regression, Cat Boost algorithm & scikit-learn libraries for machine learning.

Secondly, to learn how to hypertune the parameters for machine learning model.

And in the end, to predict rating of the application using various ML algorithms, finalizing the optimal performing model for Rating prediction by validating the accuracy given by the various models.

2.2 Exploratory Data Analysis:

2.2.1 Correlation Plot:-

Below Fig.2, shows the relation between each of the feature with one another. The values which are in positive are has positive relationship with each other like Rating count and minimum installs and the values which are negative has negative relationship like rating and price.

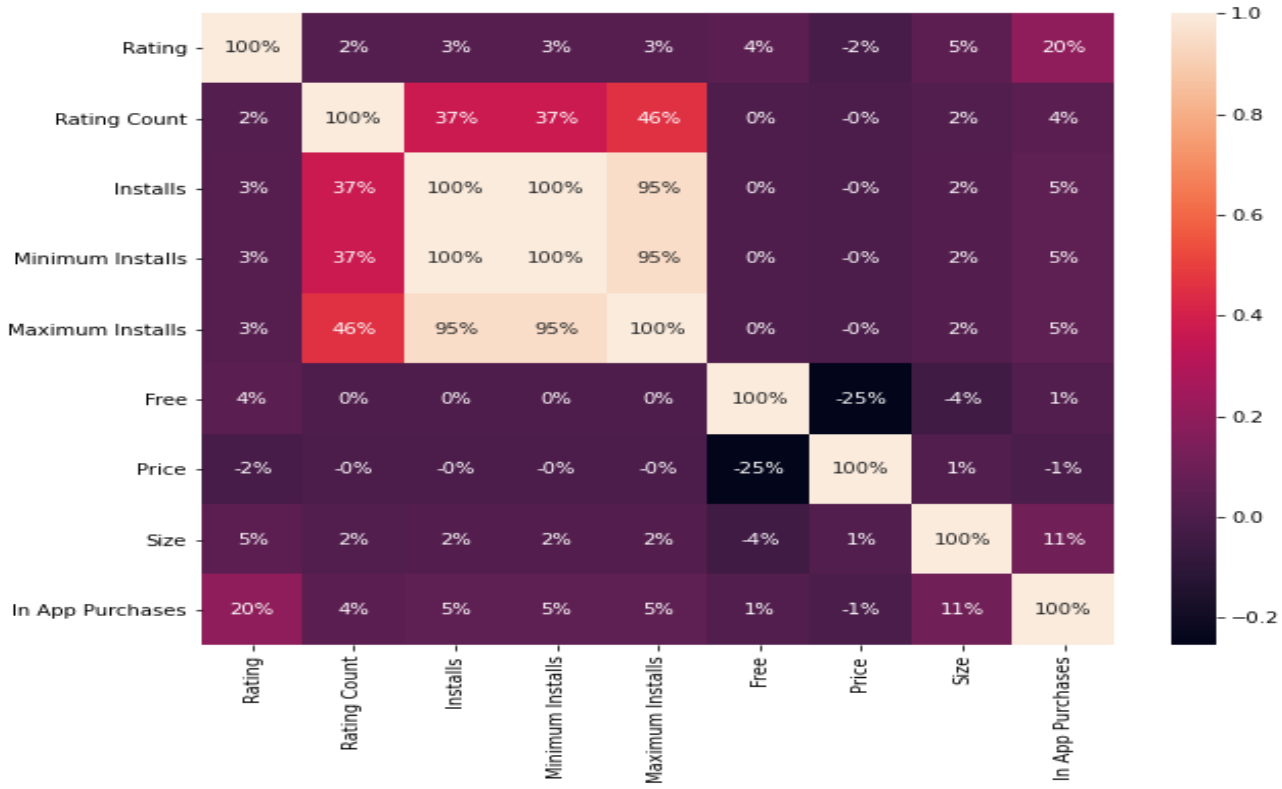

Fig 2. Correlation Matrix

2.2.2 Most Popular Category of Application:-

As per the below fig.3, Music and Audio is the most popular category followed by Education and Entertainment

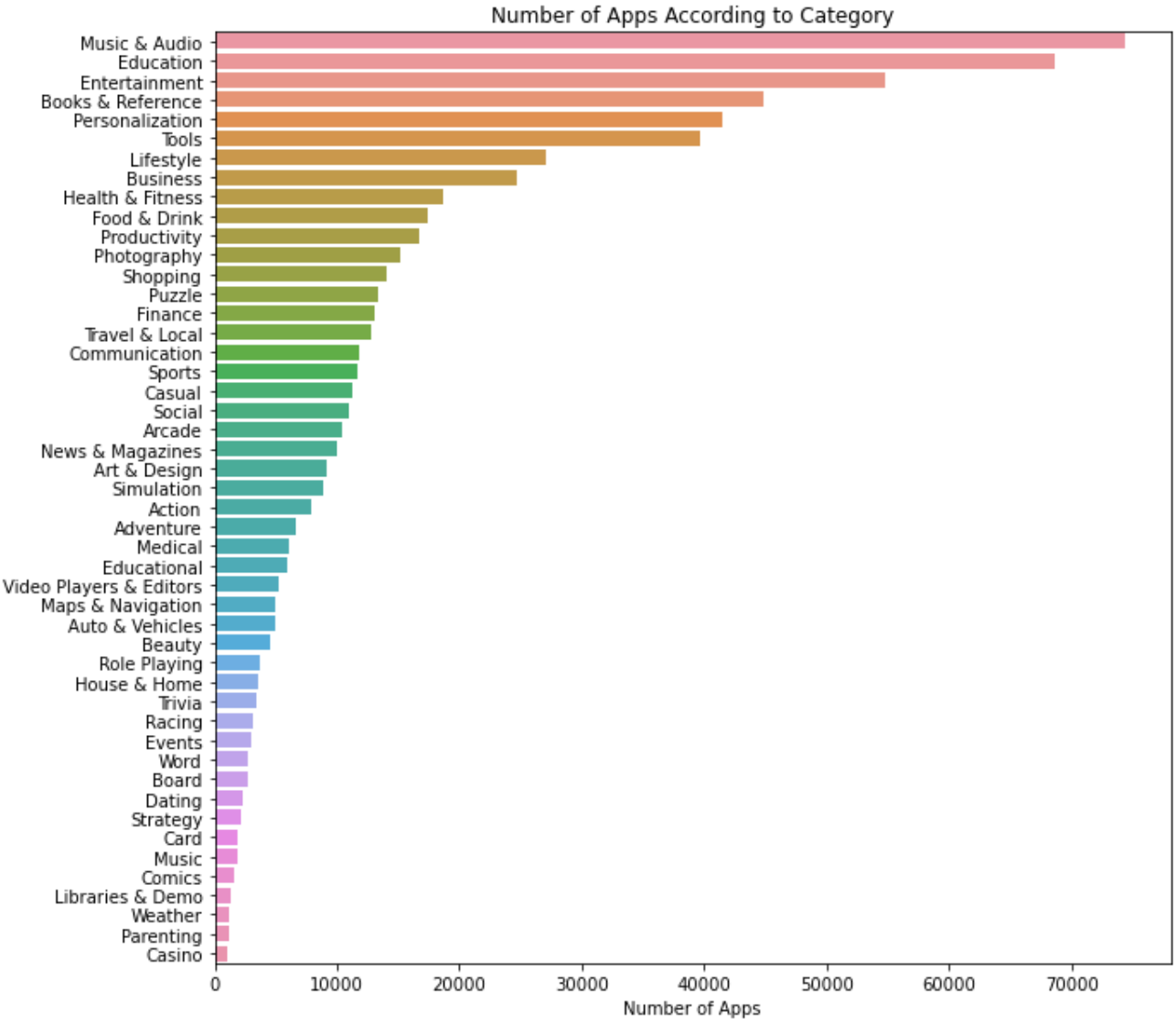

fig 3. Most Popular category

2.2.3 Category wise Installations:

As per the below fig.4, most installed category is Tools followed by Action and Arcade

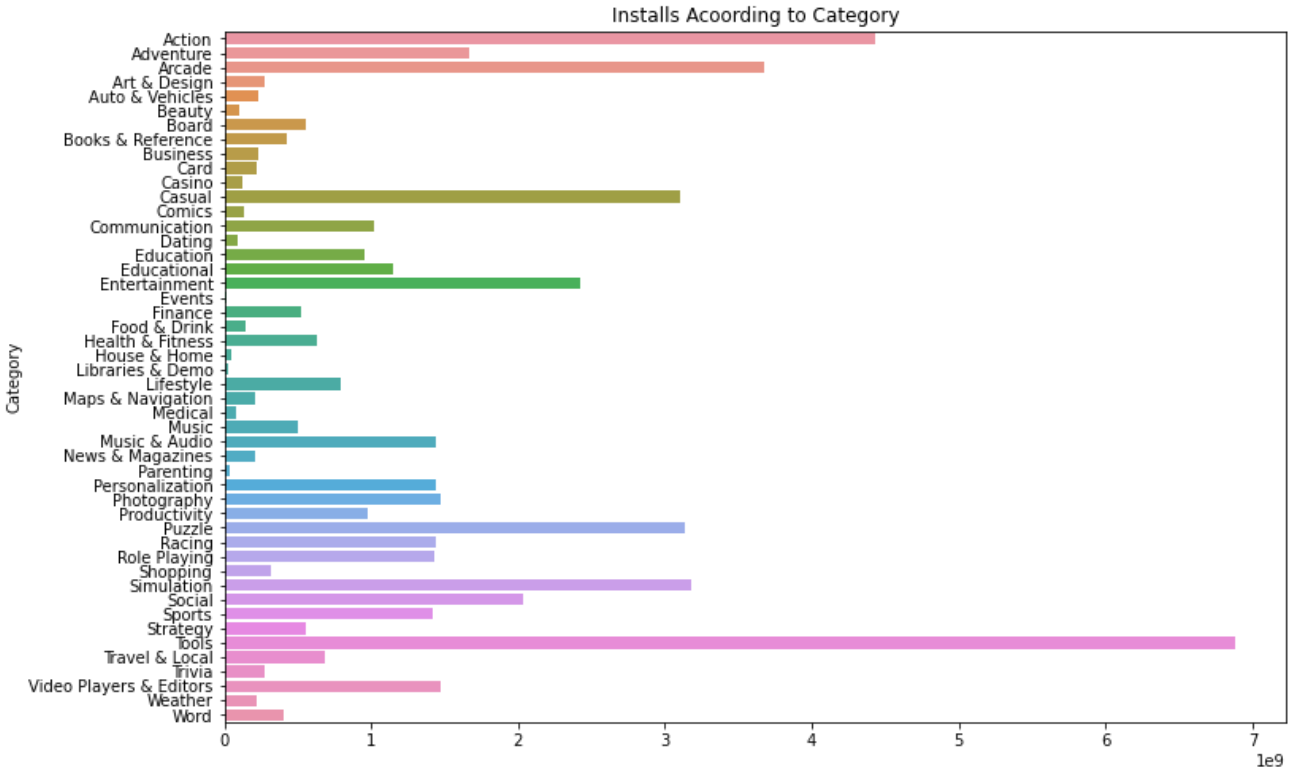

Fig .4 Category wise Installs

2.2.4 Installation count graph:

Below fig.5, shows the no of installation of the application.

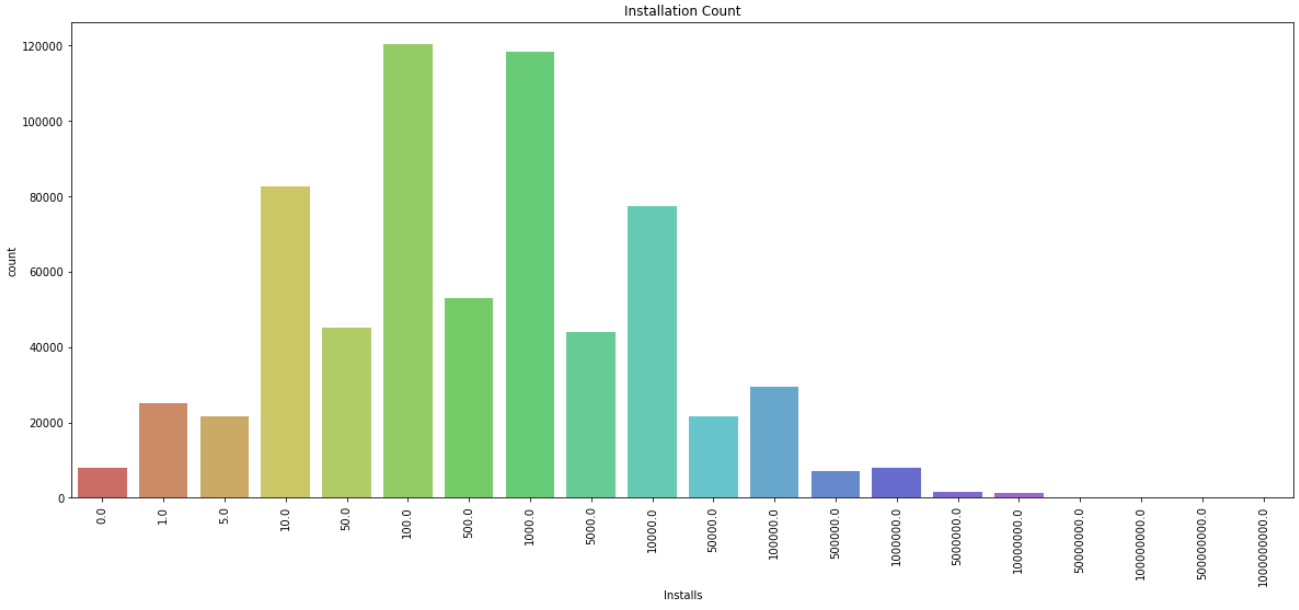

Fig.5 No of Installs

2.2.5 Content Rating :

As per the below fig.6, Everyone content type has highest count followed by Teen

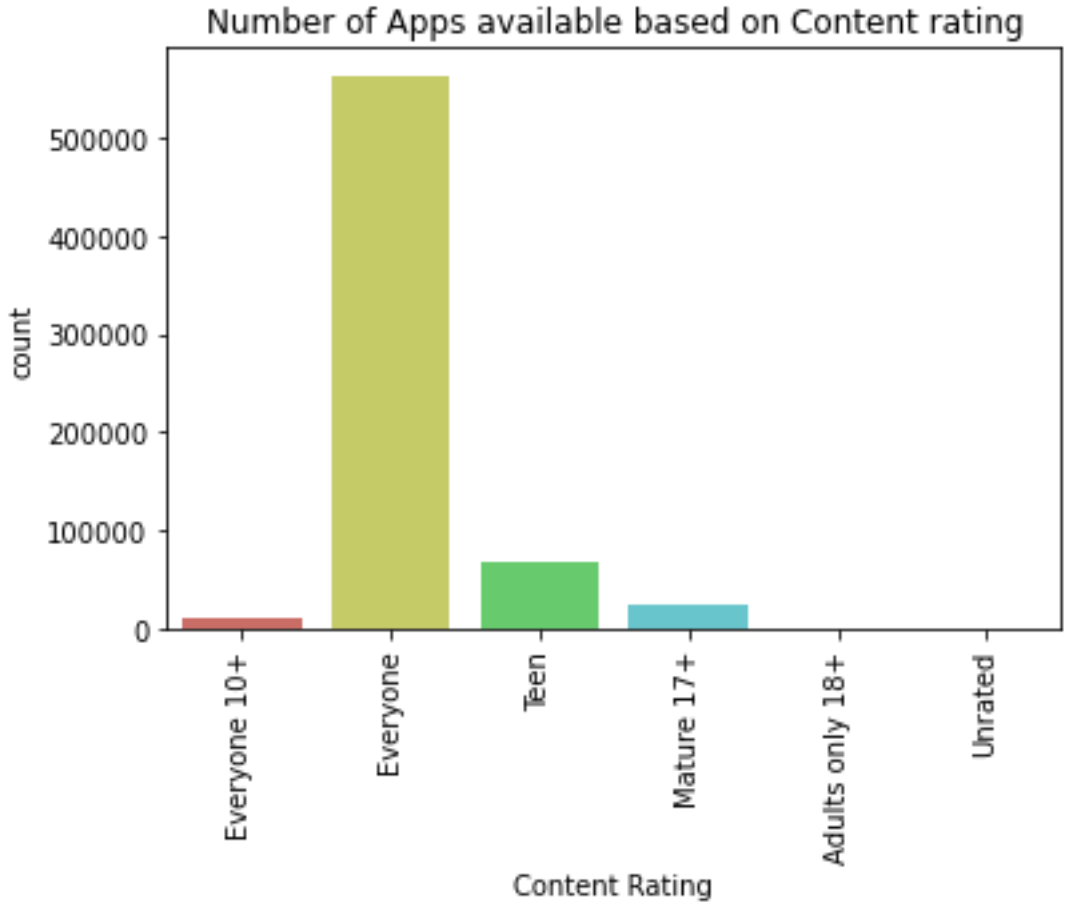

Fig. 6 Content Rating count

2.2.5 Free Vs Paid apps:

As per the below fig.7, highest no of applications are free

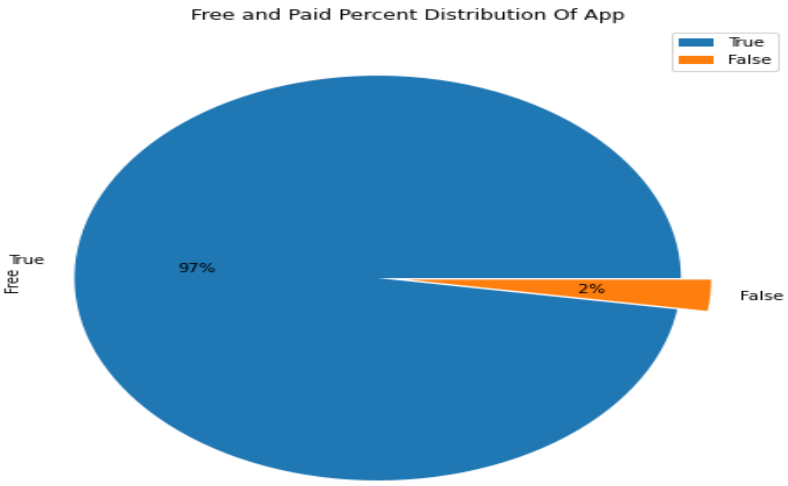

Fig 7. Free Vs Paid

CHAPTER 3

System Design

3.1 Flowchart of the System:

The flowchart of the algorithm

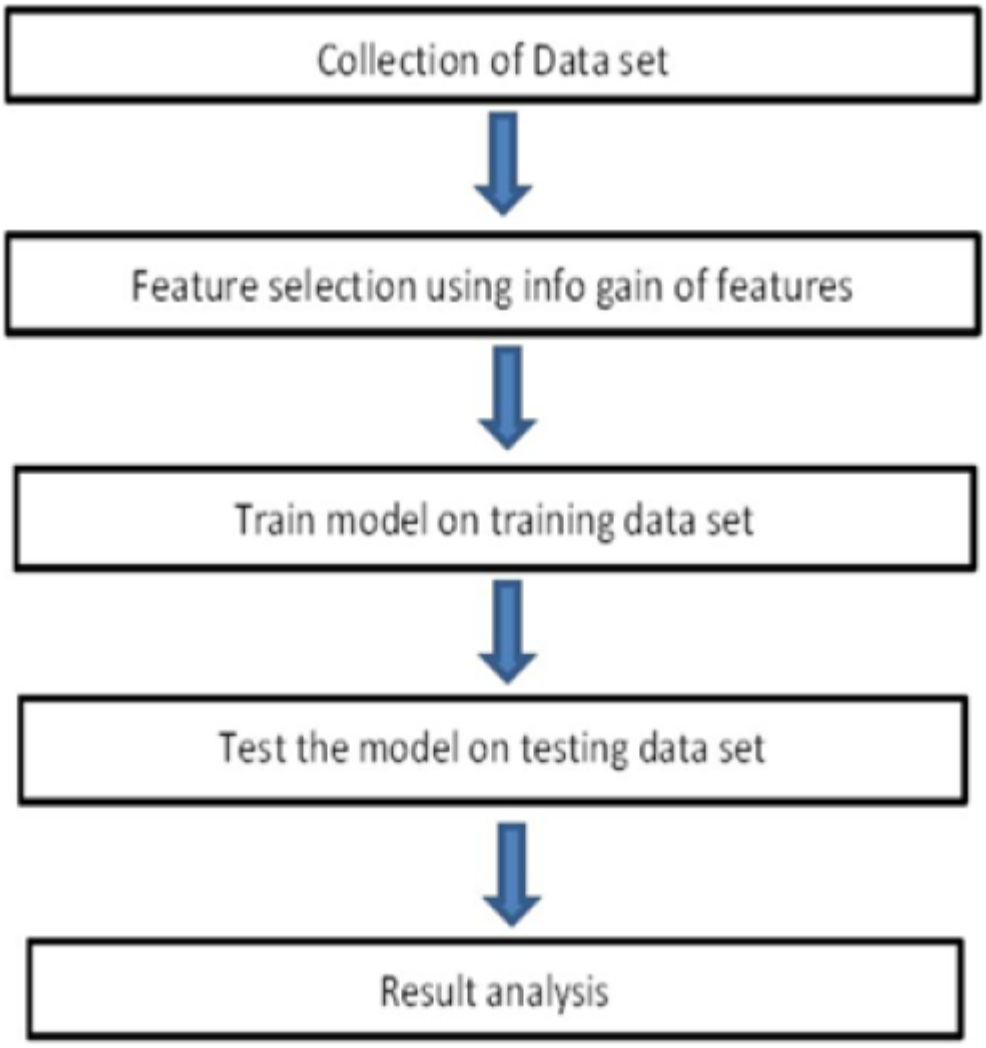

Fig.8 Flow Chart

CHAPTER 4

Model Building

4.1 Feature Prepation and Selection:

- First we have Dummied the categorical values then selected the dependent variable Rating for predication and independent variables in y and x respectively

```
[44]: #Get Dummies for Category, Free, Content Rating columns
catgry=pd.get_dummies(df['Category'],prefix='catg',drop_first=True)
typ=pd.get_dummies(df['Free'],prefix='typ',drop_first=True)
cr=pd.get_dummies(df['Content Rating'],prefix='cr',drop_first=True)
frames=[df,catgry,typ,cr]
df=pd.concat(frames,axis=1)
df.drop(['Category','Free','Content Rating'],axis=1,inplace=True)
```

```
[47]: X=df.drop('Rating',axis=1)
y=df['Rating'].values
y=y.astype('int')
```

- Secondly we have split the data into the ration 80% and 20% for train and test split a and stored it into x and y respectively

```
In [47]: X=df.drop('Rating',axis=1)
y=df['Rating'].values
y=y.astype('int')
```

```
In [48]: #Train Test Split
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=2020)
```

4.2 Algorithm Research and Selection:

4.2.1 Logistic Regression:

- We have applied Logistic regression model on the data and receive the accuracy of the model as 50.21% with the random state 2020

```
[49]: #Applying Model Logistic Regression
logreg_c=LogisticRegression(random_state=2020)
logreg_c.fit(X_train,y_train)
logreg_pred=logreg_c.predict(X_test)
logreg_cm=confusion_matrix(y_test,logreg_pred)
logreg_ac=accuracy_score(y_test, logreg_pred)
print('LogisticRegression_accuracy:',logreg_ac)
```

```
C:\Users\vaibh\anaconda3\lib\site-packages\sklearn\
rge (status=1):
STOP: TOTAL NO. of ITERATIONS REACHED LIMIT.

Increase the number of iterations (max_iter) or sca
    https://scikit-learn.org/stable/modules/preproc
Please also refer to the documentation for alternat
    https://scikit-learn.org/stable/modules/linear_
  n_iter_i = _check_optimize_result(
```

```
LogisticRegression_accuracy: 0.5021690340267213
```

4.2.2 Decision Tree Algorithm:

- We have applied Decision tree classification model on the data and receive the accuracy of the model as 74.88% with the random state 2020

```
[50]: #Applying Model DecisionTree Classifier
dtree_c=DecisionTreeClassifier(random_state=2020)
dtree_c.fit(X_train,y_train)
dtree_pred=dtree_c.predict(X_test)
dtree_cm=confusion_matrix(y_test,dtree_pred)
dtree_ac=accuracy_score(dtree_pred,y_test)
print('DecisionTreeClassifier_Accuracy: ',dtree_ac)
```

```
DecisionTreeClassifier_Accuracy:  0.7488364036210818
```

4.2.3 Random Forest Algorithm:

- We have applied Decision tree classification model on the data and receive the accuracy of the model as 79.11% with the random state 2020

```
In [51]: #Applying Model RandomForest
         rdf_c=RandomForestClassifier(random_state=2020)
         rdf_c.fit(X_train,y_train)
         rdf_pred=rdf_c.predict(X_test)
         rdf_cm=confusion_matrix(y_test,rdf_pred)
         rdf_ac=accuracy_score(rdf_pred,y_test)
         print('RandomForest_Accuracy: ', rdf_ac)

         RandomForest_Accuracy:  0.7911702239828887
```

4.2.4 Naive Bayes Algorithm:

- We have applied Naive Bayes model on the data and receive the accuracy of the model as 79.40% with the random state 2020

```
In [52]: #Applying Model Naive Bayesian
         NB = BernoulliNB(binarize = 0.0)
         NB.fit(X_train,y_train)
         y_pred = NB.predict(X_test)
         nb_ac=accuracy_score(y_test, y_pred)
         print("Bernoulli Naive Bayes_Accuracy: ", nb_ac)

         Bernoulli Naive Bayes_Accuracy:  0.7940848634562955
```

4.2.4 Cat Boost Algorithm:

- We have applied CatBoost model on the data and receive the accuracy of the model as 81.21% with the random state 2020

```
[53]: #Applying Model CatBoost Model
      Cat_Boost = CatBoostClassifier(verbose = 0, n_estimators = 100)
      Cat_Boost.fit(X_train, y_train)
      cb_ac=Cat_Boost.score(X_train, y_train)
      print("CatBoost_Accuracy: ",cb_ac)

      CatBoost_Accuracy:  0.8121488499600837
```

Below figure 9 shows the accuracy of each of the model applied on the Data set to predict the Rating and we have observed that CatBoost algorithm has given us the highest accuracy as 81.21%

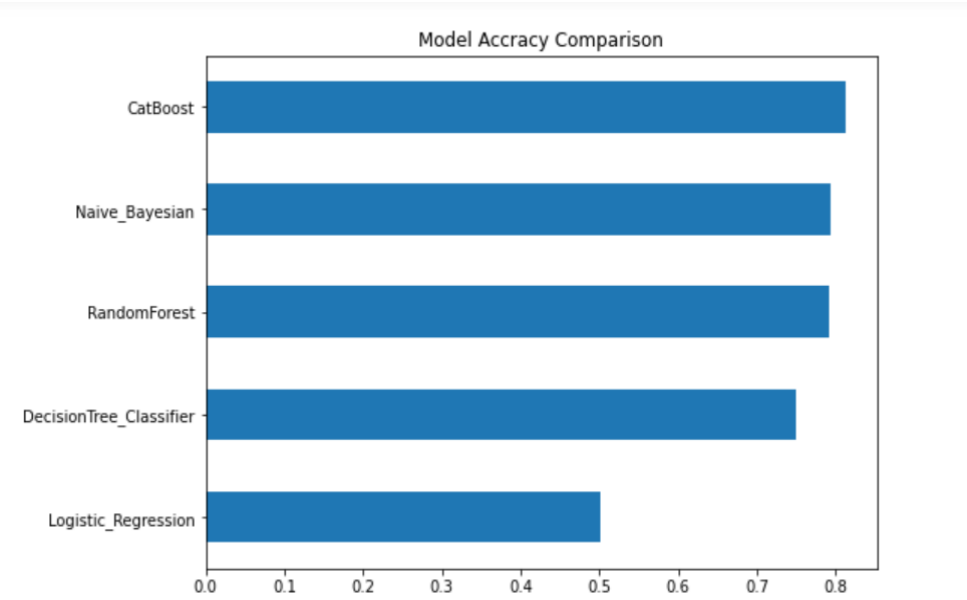

Fig.9 Accuracy Comparison of applied Models

CHAPTER 5

Results

The Google Play Store is one of the largest and most popular store for Android App. This project aims to achieve, fairly accurate rating prediction for different apps which belongs to different categories, by using machine learning algorithms and by visualizing different analytics concept using seaborn libraries and Tableau software. We have used a raw dataset of Google Play Store in which we have 23 different features that can be used for predicting whether an app will be successful or not using different feature and various Machine learning algorithms, like Logistic Regression, Decision tree Classifier, CatBoost, Random Forest and Bernoulli Naive Bayes and then compare the accuracy of each model. We have selected optimal performing model to predict ratings of apps.

The table given below shows the accuracy algorithms.

Machine Learning Algorithm	Accuracy
Logistic Regression	50%
Decision Tree Classifier	74.88%
Random Forest Classifier	79%
Bernoulli Naïve Bayesian	79.40%
CatBoost	81%

Table : Accuracy Of Algorithms

CHAPTER 6

Future Scope

UI Design for Recommendation of Apps for different categories according to rating which is given by customers who have installed it already. Predication of Installation of Application.

Prediction of maximum Installation and minimum installation for an upcoming app for particular category. Predict the days taken to reach maximum installation of app.

CHAPTER 7

Conclusion

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!

In this project according to the various machine learning algorithms it is predicted that if the app will hit on the Google Play Store on the basis of rating with best accuracy followed by 81% given by CatBoost Algorithm.

REFERENCES

[1] Statista, Number of available application in the Google Play store from December 2009 to March 2019, https://www.statista.com/ statistics/266210/numberof-available-applications-in-the-googl e-play-store/, Online: accessed 22 May 2019.

[2] Statistaa, Number of mobile app downloads worldwide in 2017, 2018 and 2020 (in billions), https://www.statista.com/statistics/ 271644/worldwide-free-and-paid-mobile-app-storedownloads/, Online: accessed 22 May 2019.

[3] J.Horrigan, Online shopping, pewinternet andAmerican life project, Washington, DC, 2018, http://www.pewinternet.org/Repor ts/2008/OnlineShopping/01-Summary-of-Findings.aspx Online: accessed 8 Aug. 2014.

[4] D. Pagano and W. Maalej, User feedback in the appstore: an empirical study, in Proc. IEEE Int. Requirements Eng. Conf. (Rio de Janeiro, Brazil), July 2013, pp. 125–134.

[5] T. Chumwatana, Using sentiment analysis technique for analyzing Thai customer satisfaction from social media, 2015.

[6] T. Thiviya et al., Mobile apps'feature extraction based on user reviews using machine learning, 2019.

[7] H. Hanyang et al., Studying the consistency of star ratings and reviews of popular free hybrid android and ios apps, Empirical Softw. Eng. 24 (2019), no. 7, 7–32