Google Play Store Apps Analysis

STA 6714

Lamia Alshahrani

About Google Play

Why ?

- Mobile apps one of the sector that will have a high potential growth in the future .
- As a user I was curious to know more about Apps and what is the most popular one.

1. Dataset :

- This dataset it is about Google play store Apps it is a web scraped data of 10k Play Store apps from Kaggle.com [1].

- It has 12 columns that contains details about the application : (*App,Category,Rating,Reviews,Size,Installs,Type,Price,Content Rating,Genres,Last Updated,Current Ver,Android Ver*)

1.2 Storing Data in Sqlite :

```
[29]
      # Import sqlite3 module into
      # this program as sq
      import sqlite3 as sq

      # Import pandas module into
      # this program as pd
      import pandas as pd

      # Create a connection object,
      # Make a new db if not exist already
      # and connect it, if exist then connect.
      connection = sq.connect('information.db')

      # Create a cursor object
      curs = connection.cursor()

      # Run create table sql query
      #curs.execute("create table if not exists studentInfo" + "(Date,RowID integer, OrderID ,Sh

      # Load CSV data into Pandas DataFrame
      student = pd.read_csv('googleplay.csv', encoding= 'unicode_escape')

      # Write the data to a sqlite db table
      student.to_sql('studentInfo', connection, if_exists='replace', index=False)

      # Run select sql query
      curs.execute('select * from studentInfo')

      # Fetch all records
      # as list of tuples
      records = curs.fetchall()

      # Display result
      for row in records:
          # show row
          print(row)

      # Close connection to SQLite database
      connection.close()
```

1.2 Storing Data in Sqlite :

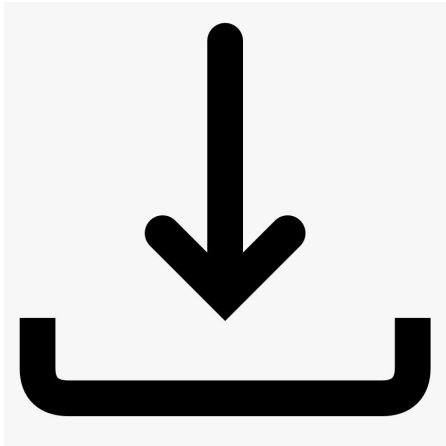
```
[ ] recordata=recordata.rename(columns={"":"",0:'App',1:'Category', 2: 'Rating',3:'Reviews',4:'Size', 5:'Installs',6:'Type',7:'Price',8:'Content Rating',9: 'Genres', 10:'Last Updated',11:'Current Ver'  , 12:'Android Ver'})
    df = pd.DataFrame(recordata)
    df
```

	App	Category	Rating	Reviews	Size	Installs	Type	Price	Content Rating	Genres	Last Updated	Current Ver	Android Ver
0	Photo Editor & Candy Camera & Grid & ScrapBook	ART_AND_DESIGN	4.1	159	19M	10,000+	Free	0	Everyone	Art & Design	7-Jan-18	1.0.0	4.0.3 and up
1	Coloring book moana	ART_AND_DESIGN	3.9	967	14M	500,000+	Free	0	Everyone	Art & Design;Pretend Play	15-Jan-18	2.0.0	4.0.3 and up
2	U Launcher Lite â□□ FREE Live Cool Themes, Hid...	ART_AND_DESIGN	4.7	87510	8.7M	5,000,000+	Free	0	Everyone	Art & Design	1-Aug-18	1.2.4	4.0.3 and up
3	Sketch - Draw & Paint	ART_AND_DESIGN	4.5	215644	25M	50,000,000+	Free	0	Teen	Art & Design	8-Jun-18	Varies with device	4.2 and up
4	Pixel Draw - Number Art Coloring Book	ART_AND_DESIGN	4.3	967	2.8M	100,000+	Free	0	Everyone	Art & Design;Creativity	20-Jun-18	1.1	4.4 and up
...	...	...	...	...	...	...	...	...	...	...	...	...	...
10835	Sya9a Maroc - FR	FAMILY	4.5	38	53M	5,000+	Free	0	Everyone	Education	25-Jul-17	1.48	4.1 and up
10836	Fr. Mike Schmitz Audio Teachings	FAMILY	5.0	4	3.6M	100+	Free	0	Everyone	Education	6-Jul-18	1	4.1 and up
10837	Parkinson Exercices FR	MEDICAL	NaN	3	9.5M	1,000+	Free	0	Everyone	Medical	20-Jan-17	1	2.2 and up
10838	The SCP Foundation DB fr nn5n	BOOKS_AND_REFERENCE	4.5	114	Varies with device	1,000+	Free	0	Mature 17+	Books & Reference	19-Jan-15	Varies with device	Varies with device
10839	iHoroscope - 2018 Daily Horoscope & Astrology	LIFESTYLE	4.5	398307	19M	10,000,000+	Free	0	Everyone	Lifestyle	25-Jul-18	Varies with device	Varies with device

10840 rows × 13 columns

Google Play

2. Question the dataset will provide answers



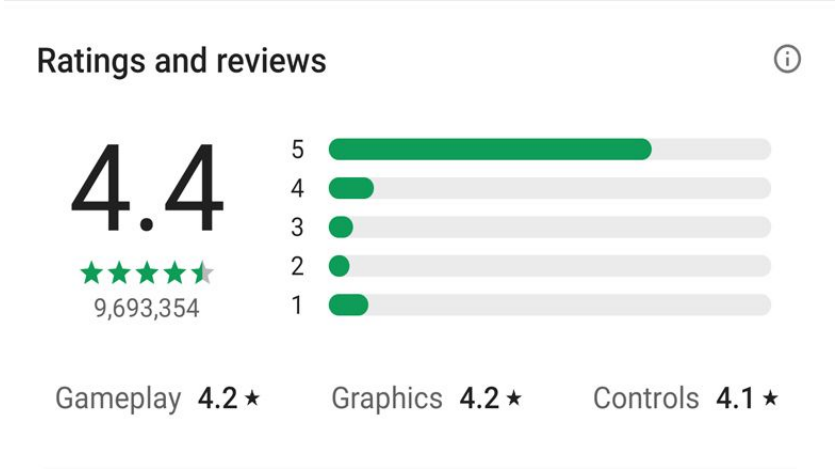
The most installed app

Most Popular App in Play Store based on the reviews

Which category of app has the highest rating

2. Question the dataset will provide answers

Predicting Rating of the Apps ?



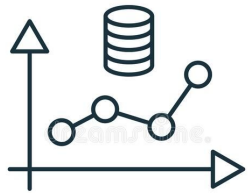
Google Play

3. Choice of methods to analyze the data

To Answer the questions I did :

Data cleaning

Data Exploration



Data Modeling

3. 1 Choice of methods to analyze the data

Data cleaning

▾ Data cleaning

```
[ ] df.isnull().sum()
```

```
App                0
Category           0
Rating          1474
Reviews            0
Size               0
Installs           0
Type               1
Price              0
Content Rating     0
Genres             0
Last Updated       0
Current Ver        8
Android Ver        2
dtype: int64
```

```
[ ] df.dropna(inplace=True)
    df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 9418 entries, 0 to 10839
Data columns (total 46 columns):
 #   Column                    Non-Null Count  Dtype
---  ------                    --------------  -----
 0   App                       9418 non-null   int64
 1   Category                  9418 non-null   object
 2   Rating                    9418 non-null   float64
 3   Reviews                   9418 non-null   int64
 4   Size                      9418 non-null   float64
 5   Installs                  9418 non-null   object
 6   Type                      9418 non-null   uint8
 7   Price                     9418 non-null   object
 8   Content Rating            9418 non-null   int64
 9   Genres                    9418 non-null   int64
 10  Last Updated              9418 non-null   object
 11  Current Ver               9418 non-null   object
 12  Android Ver               9418 non-null   object
```

▾ Converting objects to float :

```
[ ] df['Price'].replace(to_replace='0',value='$0',inplace=True)
    df['Price']=df['Price'].apply(lambda a : a[1:])
    df['price']=df['Price'].astype(float)
```

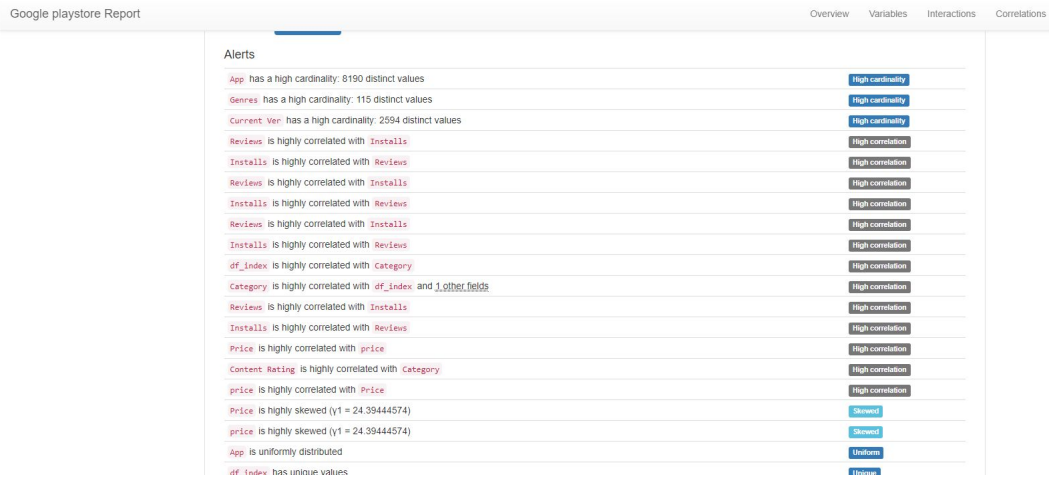
```
[ ] # converting last date
    df['Last Updated']=pd.to_datetime(df['Last Updated'])
    df['before update']=df['Last Updated'].max()-df['Last Updated']
```

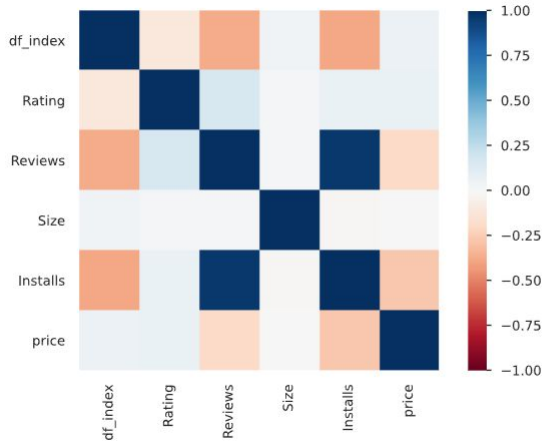
```
[ ] # converting size
    df['Size']=df['Size'].str.replace('M','e+6').str.replace('k','e+3').str.replace('Varies with device','0').astype('float')
```

```
[ ]
    df['Reviews']=df['Reviews'].astype('int')
```

3. 1 Choice of methods to analyze the data

Data Exploration

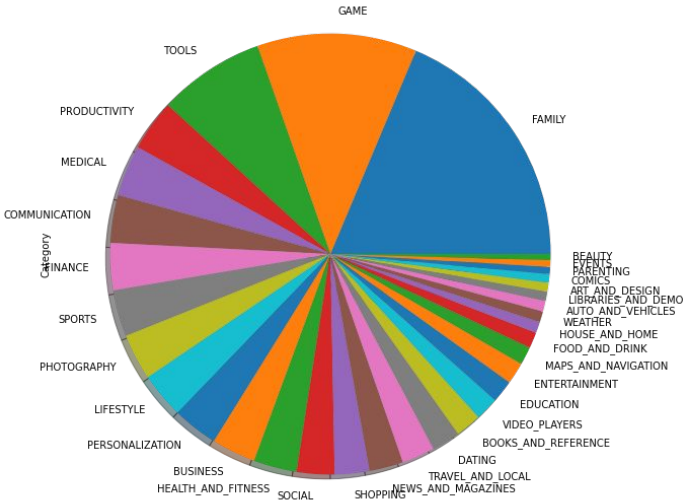




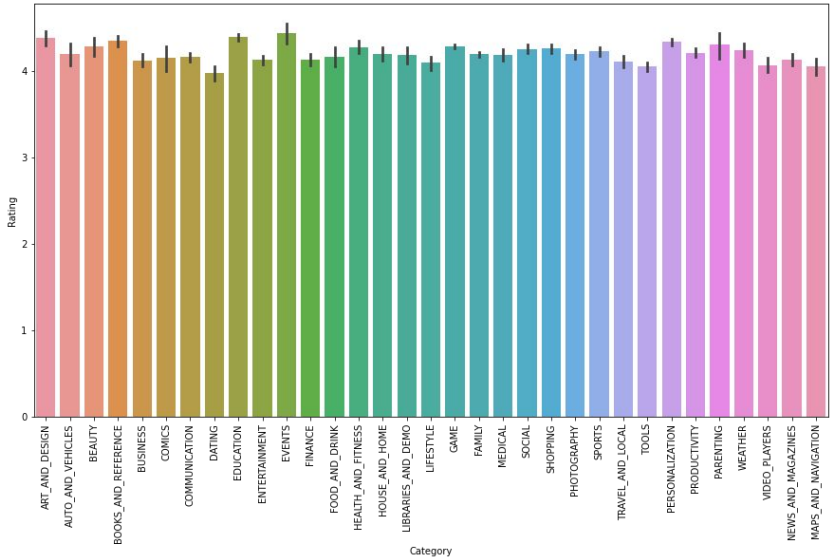
Google Play

3. 1 Choice of methods to analyze the data

Most popular category

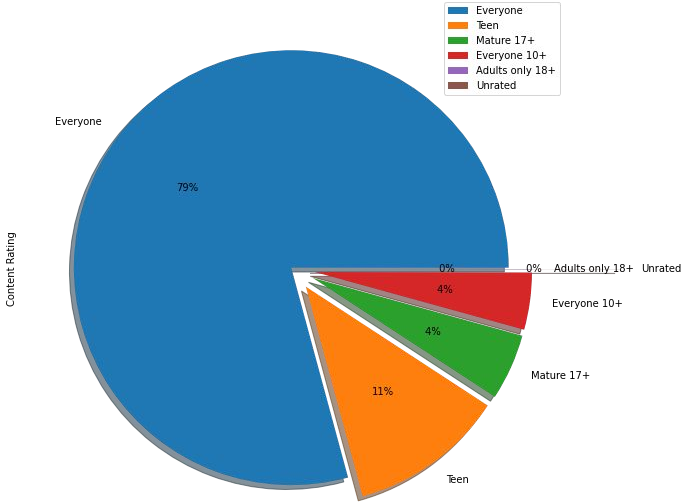


Which category has the highest rating?

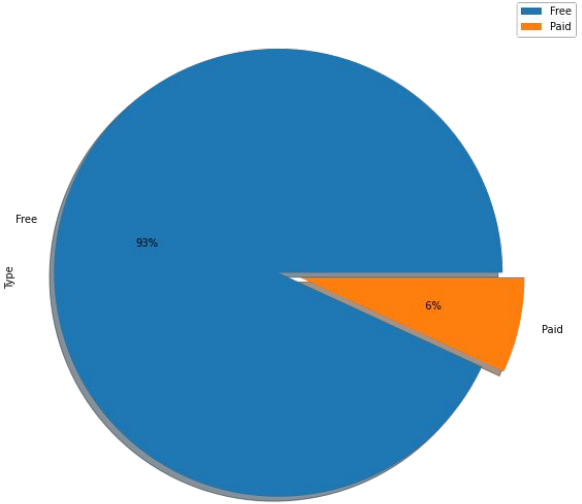


3. 1 Choice of methods to analyze the data

Content Rating

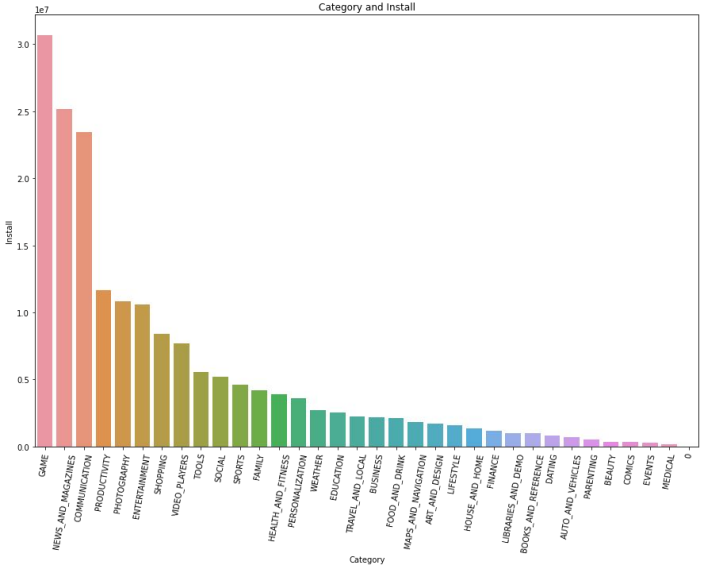


Which type is popular ?

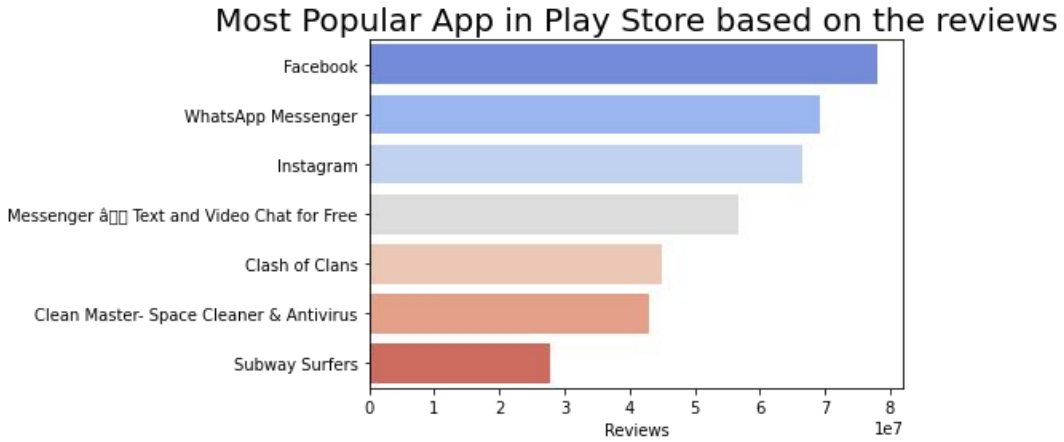


3. 1 Choice of methods to analyze the data

The most installed app?



Most Popular App in Play Store based on the review



Google Play

4. Pre-Data Modeling

Predicting Rating of the Apps ?

```
def clean_data(df):
    '''

    '''
    df = df.dropna(subset=['Rating'],axis=0)
    y = df['Rating']

    X_add = np.log(df[['Reviews','Installs']])
    #df = df.drop(['Rating','App','Current Ver','Android Ver','Day','Genre1'], axis=1)
    df = df.drop(['Rating','Current Ver','Android Ver','App','Reviews','Installs'], axis=1)

    #Numeric variables
    num_var = df.select_dtypes(include=['float','int']).columns
    for col in num_var:
        df[col].fillna((df[col].mean()),inplace=True)

    #Categorical variables
    cat_var = df.select_dtypes(include=['object']).columns
    for col in cat_var:
        df = pd.concat([df.drop(col, axis=1), pd.get_dummies(df[col],prefix=col,prefix_sep='_')],axis=1)


    X = pd.concat([df,X_add],axis=1)
    return X,y
```

Making sure all the columns can be fitted into the model :

Ex: we need to create a dummy variables for categories to use our models

4. Data Modeling

Predicting Rating of the Apps ?

```
def Evaluationmatrix_dict(y_true, y_predict, name = 'Linear - Integer'):
    '''
    INPUT:
    y_true - Traget value in a test set
    y_predict - predicted value from the model
    OUTPUT:
    Evaluation of the model using: Mean Squared Error and Root Mean Sqaured Error
    '''
    dict_matrix = {}
    dict_matrix['Series Name'] = name
    dict_matrix['Mean Squared Error'] = metrics.mean_squared_error(y_true,y_predict)
    dict_matrix['Root Mean Squared Error'] = sqrt(mean_squared_error(y_true, y_predict))
    return dict_matrix
from sklearn import preprocessing
from math import sqrt
X_train,X_test, y_train, y_test = train_test_split(X,y, test_size = 0.3,random_state=42)
# mm_scaler = preprocessing.MinMaxScaler()
#X_train_minmax = mm_scaler.fit_transform(X_train)
#mm_scaler.transform(X_test)
# scaler = StandardScaler()
# X_train_scaled = scaler.fit_transform(X_train_minmax)
# X_test_scaled = scaler.fit_transform(X_test)
lm_model = LinearRegression(normalize=True)
lm_model.fit(X_train, y_train)
LinearRegression(copy_X=True, fit_intercept=True, n_jobs=None, normalize=True)
y_test_preds = lm_model.predict(X_test)
results = pd.DataFrame()
results = results.append(Evaluationmatrix_dict(y_test,y_test_preds, name = 'Linear Regression'),ignore_index = True)
results
```

	Mean Squared Error	Root Mean Squared Error	Series Name
0	2.865949e+28	1.692911e+14	Linear Regression

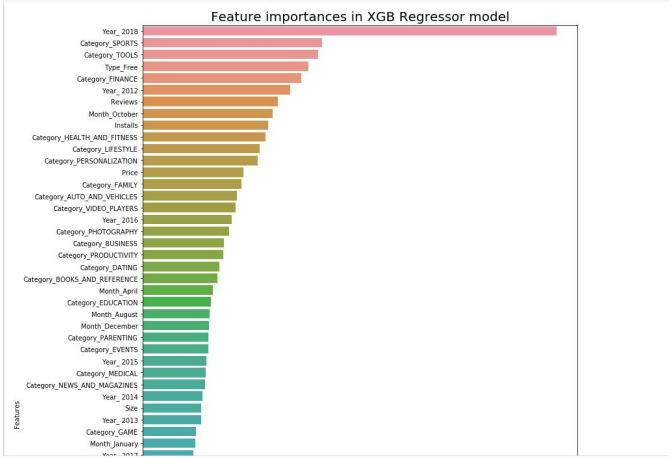
Linear Regression :

The scores are very high (overfitting)

4. Data Modeling

Predicting Rating of the Apps ?

```
[ ] from xgboost import plot_importance
    import matplotlib
    import matplotlib.pyplot as pyplot
    ax = plot_importance(gbm)
    fig = ax.figure
    fig.set_size_inches(20, 20)
```



XGboosting :

```
[ ] print('This is a RMSE score of XGBoost Regressor',rmse(gbm.predict(X_test_2, ntree_limit=gbm.best_ntree_limit),y_test_2))
```

This is a RMSE score of XGBoost Regressor 0.5250269424042028

```
[ ] print('This is MSE score of XGBoost Regressor',mse(gbm.predict(X_test_2, ntree_limit=gbm.best_ntree_limit),y_test_2))
```

This is MSE score of XGBoost Regressor 0.2756532902503061

5. Conclusions

- From the result we can say that 2018 was one of the important feature
- Sport and tools category was important and Free apps.

6.Future work :

- Doing some hyperparameter tuning for the models .
- Analysing more features and find the relations .
- Scaling and normalizing the data to get a better result for LR.
- I think doing more analysis in this field can help both users and developers.

References :

[1]. Dataset by Kaggle https://www.kaggle.com/lava18/google-play-store-apps

Google Play