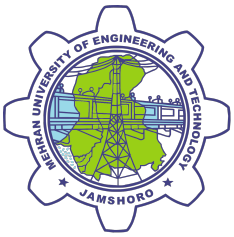
GOOGLE PLAYSTORE APPS ANALYSIS AND VISUALIZATION



A thesis submitted by

AEMON JABBAR (GL) (18SW34)
ZAIN ALI (18SW39)

Supervisor
Engr Salahuddin Saddar

In the partial fulfillment of the requirements for the degree of
Bachelor of Engineering in software engineering

Department of Software Engineering

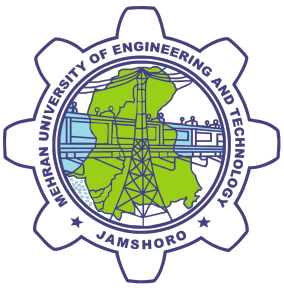
MEHRAN UNIVERSITY OF ENGINEERING &
TECHNOLOGY, JAMSHORO

October, 2022

DEDICATION

This humble effort is
DEDICATED
to our **PARENTS** and
TEACHERS With
GRATITUDE and RESPECT

DEPARTMENT OF SOFTWARE ENGINEERING



CERTIFICATE OF APPROVAL

This is to certify that, Project/Thesis report on "**Google Playstore Apps Analysis And Visualization**" is submitted in the partial fulfilment of the requirements for Bachelor's degree in Software Engineering by the following students:

AEMON JABBAR (GL) (18SW34)

ZAIN ALI (18SW39)

Project/Thesis Supervisor	Chairman
Engr. Salahuddin Saddar	Dr. Naeem Mahoto

Dated: _____

ACKNOWLEDGEMENT

First and foremost, we gratefully thank Almighty Allah for providing us with the power and capability to do this tough endeavor. Our families' prayers also assisted us in completing this project, for which we are grateful. We are grateful to our supervisor, Sir Salahuddin Saddar, for assisting us in completing our final year challenge during the course. We were able to meet the challenge's objectives thanks to their guidance, assistance, and inspiration

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

ABSTRACT

This project is based on the numerical analysis of the Google applications' rating for the computation of their reviews, number of installations, application's rating, price of the application etc. The dataset for this project has been downloaded from Kaggle.com (an authorized platform of python and ML datasets). The dataset has been cleaned and modified for acquiring nominal outcomes pertaining to the detailed analysis of various Google Applications' ratings. This project will be beneficial for all the developers, project managers and the local people to know the worth and value of any application and its ratings. Such type of numerical analysis and visualization performed can also sort out all the relevant rating details of applications which are widely used and downloaded by the people. The python libraries (Numpy, Pandas, OS, Matplotlib, Seaborn) will be used to import the data and the data cleaning will be performed to clean the data of the dataset taken from an authorized source. After that, the same data will help analyze the data from the directory for data loading as well as the data analysis will help predict the shape of the data. Then comes the main goal of this project to find and compute the rating of the applications through plotting a graph from the dataset, where the null and missing values will be sorted

out column wise. Finally, the graph will be plotted and sketched on the basis of category-wise rankings and prices in order to predict the visualization of google application rating and to know which one is the best- trending application in terms of rating, download rate, price and reviews.

CHAPTER 1

INTRODUCTION

In this chapter, we've explained our study and provided a comprehensive summary of our work, which includes an introduction to the project, a problem statement, the project's rationale, a description of the data, a brief discussion of the methodology, and goals and objectives. Additionally, other sections of this report include additional details on our analysis and research.

1.1 INTRODUCTION

The market for mobile apps is expanding quickly, and this has been shown to have a significant influence on digital technology. Mobile applications play a crucial function in our daily lives as people experience them. Today Android apps are used by all of us. Numerous Android applications are used by people, including messenger, social media, games, and browsers. One of the fastest-growing subsectors of software applications is mobile apps. We select the Google Play Store in every area due of its recent rapid growth and rising popularity. We will analyze and evaluate Google Play store applications in accordance with that goal. The use of search engines, video gaming consoles, and e-chatting services is a part of life for half of the earths

population. Users are encouraged to download a wide variety of apps from established categories on the Google Play store. More than one million mobile programmes, often known as "mobile apps," are accessible to users of mobile devices through this online market. When compared to the Apple App Store, the Google Play Store produces more than twice as many downloads, but just half as much revenue. In this century, a barrier has been the accessibility of use of products and services with only one click. Applications available through the Google Play store seek to accomplish the same thing. It has not only emerged as the most widely used platform for downloading applications but is also challenging for alternative services to attract in new users due to its ease of use and global accessibility. At every stage of the decision-making process, we consider the perspectives of others. Prior to the World Wide Web's (WWW) broad use, we used to check our loved ones before using anything. However, in modern times, we must seek advice from others. More specifically, Android device owners often select their necessary apps from the Google Play Store. It has been observed that consumers typically choose an app based on its numerical rating. The rating is the average of all the ratings submitted by other users by stars, the minimum rating is 1 star and the maximum rating is 5 stars. This python project aim

to visual analytical concepts to earn insights into how applications are becoming successful and maintain higher user ratings. So, we scratched data from the Kaggle to do our research on it.

1.2 PROBLEM STATEMENT

Smartphone development is fueling the rapid expansion of mobile app shops. The two biggest international venues for the distribution of programmes are now Apple's App Store (for iOS users) and Google Play Store (official app store for the Android users). The Google Play store was our choice for our research, and we carefully considered the features it provided that would help us forecast the success of a particular app. The degree of competition is rising as the mobile business grow so aggressively. Nevertheless, more competition also results in more failures. Because companies cannot afford for an application to fail, a significant amount of time, effort, and money must be put by the developers in additional research. Project managers and programmers were uncertain about what things are working out for them and what are not, and even the reviews were not completely able to categorize. However, the lack of a clear understanding of the inner workings and dynamics of popular app markets influences developers and users.

1.3 MOTIVATION

The Mobile App Industry is profitable in a significant way and Consequently, there is increasing competition among individuals who build apps in the growing community. This is kind of a caution to all developers to optimise their applications to the highest possible level because more competition will raise the probability that the value of their apps in the play store will decrease. Because of this, even if an app is released and has a significant number of installations in addition to a normal rating, it will not be enough to maintain its position in the Play store indefinitely because many other applications will be offering customers the exact same features, if not better. We decided to conduct research relevant to this industry due to the market's growth and competition, so we carefully analyzed our data on the Google Play Store and developed our own success metric. We believe this will be a significant contribution for the developers because they will be able to determine which features should be kept and which ones should be changed based on the app's present condition to this success metric. Finding anything significant throughout the analysis that was significant to the developers, producers, and end users was the primary motivating factor for this study. Many researchers have looked at issues like business

ratings, sentiment ratings, version ratings, and many more analyses of data from the Google Play store. Since of this, even though the topic we selected had only been the subject of a few different types of research, we decided to move on with it because we thought it would be more interesting for our research.

1.4 DATA DESCRIPTION

Google Play Store uses knowledgeable modern-day techniques (like dynamic page load) using JQuery making scratching more challenging. Each application's row has values for the app name, category, rating, size, number of installs, type, price, content rating, genres, last updated, current version, and android version. This information collected is from the Google Play Store. The different data fields in the dataset are described below:

App: The name of the application.

Category: The category of application to which the application belongs.

Rating: Number of user ratings of the application.

Reviews: Number of user reviews for the application.

Size: Size of the application.

Installs: Number of user downloads/installs for the application.

Type: A binary variable that denotes whether an application is free or paid.

Price: The cost of the application in US Dollars.

Content Rating: Age group to whom the app is targeted at i.e Children / Mature / Adult

Genres: An application can belong to more than one genre (apart from its main category). For example, a musical family game will belong to the Music, Game, and Family genres.

Last Updated: Date when the application was last updated on Play Store.

Current Version: What is the current version of the application available on the Play Store.

Android Version: Minimum required Android version.

In total there are 13 attributes and 10841 Records in the original raw dataset. Each row corresponds to a single application and its performance across the 13 parameters. The 12 other attributes were used to predict "Rating" which is the user rating, where 5.0 is the maximum and 0 corresponds to the minimum user rating for the application.

1.5 METHODOLOGY

1. Data collection: Find a suitable dataset that fulfills the requirement to apply analysis on the provided attributes.

2. Pre-process data: To make data in the correct format some filtration functions have been applied.

3. Explore data: Fix if there is any irrelevant value, sparsity, null, repetition, or error.

4. Filter data: Remove extra columns that affect computation time and memory utilization.

5. Visualizing data: Visualization of data on each one of the columns described.

6. Data Evaluation: Comparison

7. Observe data: Analyze results.

1.6 AIM AND OBJECTIVES

The aim of this project is develop dashboard that is beneficial for all the developers, project managers, users to know the worth and value of any application and its ratings.

The main objectives are:

- To find and explore the dataset.

- To compute the ratings of the applications on google application forum.
- To predict visualize trending application in terms of rating, download rate, price and reviews.

1.7 THESIS SUMMARY

There are six chapters in this thesis booklet. Introduction, Motivation, Problem Statement, Data Descriptiom, Methodology, Aim and Objectives, and Thesis Summary are all included in the first chapter. The Literature Review, Related Work, Similar Types of Google Play Store Applications Systems, and Our System are all included in Chapter 2. The GPS Methodology is covered in Chapter 3. The Tools and Technologies utilized in the overall system are covered in Chapter 4. The project's results and discussions are covered in Chapter 5. The project's future work and conclusion is discussed in Chapter 6.

CHAPTER 2

LITERATURE REVIEW

2.1 INTRODUCTION

In this chapter we have presented the working of relevant papers and defined different perspectives of author of similar systems.

2.2 RELATED WORK

In different research papers, Google Play Store Applications Analysis and visualization using Python-Machine Learning have been discussed by various authors some of the research papers are discussed below:

The author of this research [1] discovered through analysis that there was no association between programme attributes such app size, average rating, installations, or even pricing and ratings. The number of instals and the number of reviews had a significant inverse relationship. The majority of the applications were classified as tools in the entertainment, education, business, and medical categories. The author separated the apps into successful and failed groups based on the quantity of installations. The Gaussian Naive Bayes model had the lowest accuracy (88.45%), while Decision Tree

had the best accuracy (95.32%). The simplicity of the decision tree's architecture and its inclusion of an extremely crucial feature—the total number of installs—helped it perform successfully. The Gaussian Naive Bayes model, which has a high feature independence assumption, also gave the author the lowest accuracy. Naive Bayes is less accurate because it makes the erroneous assumption that all characteristics are independent, which seldom holds true in reality. The dataset included 33 distinct categories in all, and the author discovered that the overall average star rating for all the applications was 4.0. The applications with most high demand or used belonged to categories of "Entertainment" and "Auto And Vehicle"

In paper [3], the authors created a variety of algorithms that can automatically determine the rating-review conflict in an effort to depict the rating-review discrepancy. The authors used two machine learning methodologies, along with a variety of classifiers, such as the Naive Bayes Classifier, Decision tree, Decision stump, and Decision table, as well as a few additional algorithms, to get away of this discrepancy. Another approach concentrated on deep learning techniques. To estimate what end users and developers would think about this mismatch, they have done a number of polls. The survey's findings were quite predictable; end users and developers of

mobile applications both agreed that a mobile application's rating and review should correlate, and they also opted to purchase an automated system to look for any discrepancies between the two. The approach to depict the review-rating mismatch was recommended in this research. Because of the work's inspiration, the author tried to create a feature based on the difference and eliminate those reviews with the highest percentage of difference. However, only 2000 out of the 199763 reviews could be carefully annotated by three among their members because to time restrictions. As a result, the author was unable to achieve the desired outcome and rejected the plan.

The author of a separate article [10] that addresses the same issue claims that there is a significant percentage of difference between the numerical ratings, such as stars given by users, and the reviews provided by the same users. As a result, the author has proposed a rating system that will eliminate the uncertainty caused by the discrepancy between the rating and appropriate review given by the same user. According to the author, individuals install applications based on the ratings that are given for each particular piece of data since it has been seen how much we, as users, rely on other people's opinions when making decisions. The problem is divided into two sub-problems by the author of this study. First, there is uncer-

tainty, and second, there are biases in the users' summed ratings. He continued to clarify the issue by saying that before, individuals would merely take the rating from the comments, leaving out the star rating that went along with it. He developed a technique to address the issue that will first analyse the user evaluations for sentiment before generating a numerical rating based on polarity. As a result, the final rating will be the average of the sentiment analysis and the end-user-provided star ratings. This idea will lessen user misunderstandings and provide for a final rating based on both reviews and star ratings. As a result of this paper's demonstration of the close connection between customer reviews and star ratings, we were inspired to employ user reviews of the application in our own work. To that point, we properly studied the reviews and provided explanations for each one. On the other hand, Jong [11] used a Yelp dataset that included 150 000 reviews and their accompanying ratings for apps found on the Google Play store. The author's research revealed that text reviews had a higher quantitative value than star ratings, and that adding a star rating to a list of text reviews of mobile apps will produce a comprehensive quantitative estimate of the level of service satisfaction. He declared all ratings above 3.6 to be positive sentiment and all ratings below 3.6 to be negative senti-

ment because the average rating for the dataset was 3.6 stars. They applied a new learning method, which is comprised of two processes: identifying semantic similarities and modelling after sentiment, but they still used Naive Bayes and SVM, identical to the work of Pang and Lee. They predicted ratings with related sentiment analysis of 20,000 unique reviews using seven different training set sizes. The Naive Bayes classification method produced the greatest results in their research out of the three.

Fu, Lin, and Li's research [6] gave us the concept to remove inconsistent reviews due to the significance of polarity values. This would greatly reduce dataset noise, increase sentiment analysis performance, and allow us to obtain more accurate polarity values. A system called WisCom, designed by Fu, Lin, and Li, can evaluate at least 10 million user reviews and comments from application markets at three completely separate levels. Their system's main function is to identify reviews that are inconsistent, and they also looked at why individuals dislike particular applications as well as how reviews change with time. They expanded on their analysis to offer a perceptive view into the whole application sector, which causes substantial issues for clients. Additionally, the information will be used by app developers to better understand why customers detest

their app, that will finally aid in the app's quality improvement. It develops methods for reducing and analysing reviews so that users may select the best application without having to read the remark. Most frequently, it analyses reviews at three different levels: first, at the local scale (using a standard linear regression model), then at the microlevel (using LDA and topic analysis on various time segments to assess how reviews change over time), and finally, at the level of all the apps available on the market (macro level). They discussed the advantages that would be gained by end users, developers, and the broader mobile ecosystem while highlighting each of their three levels of approaches.

We also looked for articles that were written in different fields but that carried out related work in order to better understand how reviews differ from one domain to another and to be able to explain the selection of each feature in our dataset that was essential in predicting success using the success metrics suggested in [15]. In a research [13], Pang and Lee collected movie reviews and classified the retrieved scores into neutral, positive, or negative categories. They asked two graduate students in computer science to volunteer for the record of words, and they asked them to come up with outstanding identifying terms for favorable and negative feel-

ing in a review for the film. Then, using seven keywords for each positive and negative emotion, they constructed a more condensed collection of words. They started by appropriately identifying the overall negative and positive feeling from the words given. These 3 algorithms—Naive Bayes, Support Vector Machine (SVM), and Maximum Entropy—were used to examine 8 different feature combinations, and the outcomes were compared. SVM outperformed the competition the best, while Naive Bayes outperformed the rest. They claim that while accuracy for SVM and Maximum Entropy remains constant, adding Parts of Speech tags modestly increases accuracy for Naive Bayes. Because app reviews and movie reviews are frequently not the same thing, their work varies from ours in several ways [6].

The authors of this article [14] provide a system that enables developers to sort, condense, and evaluate user evaluations of programmes. Authors take pertinent information from app evaluations, such as functions, issues, and needs, and then examine the tone of each feature. Three primary building components are presented in this study: topic modelling, sentiment analysis, and summarization interface. The topic modelling block looks for semantic subjects in text written comments and extracts characteristics that support the

most important terms for each topic. Every characteristic that has been found is subject to sentiment analysis.

All of the essential researches that were previously mentioned enabled us to further explore all of the concepts that we used in our work, which allowed us to present it in a more ordered manner and, ultimately, enabled us to provide a better analysis of the characteristics that are present in our dataset.

CHAPTER 3

METHODOLOGY

A project which follows systematic development goes through certain steps which are closely observed and documented at each step, this helps in follow-up of the project, to make certain changes or to take inspiration for similar sort of projects. We have followed following 5 stages:

3.1 STEP 1: DEFINE QUESTIONS GOALS

The first step in data analysis is to clearly define your questions and goals. So we thought about the parameters that are important to project managers, developer, and users. To research regarding this we did research on Google play store, Microsoft store and apple store. And then we discussed about how to put these parameter in a way that brings value to our target audience.

3.2 STEP 2: COLLECT DATA

There needs to be data available before you can begin analysing. We tried to collect data from multiple resources namely kaggle, nasdaq and data gov and then we did comparative analysis and found kaggle's data more fit for our purpose[2].

	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	January 7, 2018	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	January 15, 2018	2.0.0	4.0.3 and up
2	U Launcher Lite – FREE Live Cool Themes, Hide ...	ART_AND_DESIGN	4.7	87510	8.7M	5,000,000+	Free	0	Everyone	Art & Design	August 1, 2018	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	June 8, 2018	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	June 20, 2018	1.1	4.4 and up
5	Paper flowers instructions	ART_AND_DESIGN	4.4	167	5.6M	50,000+	Free	0	Everyone	Art & Design	March 26, 2017	1	2.3 and up
6	Smoke Effect Photo Maker - Smoke Editor	ART_AND_DESIGN	3.8	178	19M	50,000+	Free	0	Everyone	Art & Design	April 26, 2018	1.1	4.0.3 and up
7	Infinite Painter	ART_AND_DESIGN	4.1	36815	29M	1,000,000+	Free	0	Everyone	Art & Design	June 14, 2018	6.1.61.1	4.2 and up
8	Garden Coloring Book	ART_AND_DESIGN	4.4	13791	33M	1,000,000+	Free	0	Everyone	Art & Design	September 20, 2017	2.9.2	3.0 and up
9	Kids Paint Free - Drawing Fun	ART_AND_DESIGN	4.7	121	3.1M	10,000+	Free	0	Everyone	Art & Design;Creativity	July 3, 2018	2.8	4.0.3 and up

Figure 3.1: Data Set

3.3 STEP 3: DATA WRANGLING

Once all of your data is in one place, it is essential to clean the data before starting this procedure' analysis phase. Making sure the data is in a format that can be used is a big component of the cleaning process. This comprises looking for data that could have been entered incorrectly, dealing with null values, and looking for outliers. According to the project's requirements, we cleaned the gathered data by doing the following cleaning steps:

- **Null values:**

We found alot of missing column and that could have become problem in the later stage of analysis.

```
df.isnull().sum()
```

```
App                0
Category           0
Rating          1474
Reviews            0
Size               0
Installs           0
Type               1
Price              0
Content Rating     1
Genres             0
Last Updated       0
Current Ver        8
Android Ver        3
dtype: int64
```

Figure 3.2: Null Values

- **Symbols nullify:**

  In the collected data we found abundance of symbols such as '+', $ and ',' because the data is about applications and it's parameter are described by these symbols, so for the ease of analysis we removed them.

```
# Fixing inconsistent formatting
df_cleaned = df.loc[df["Rating"].notnull()]
df_cleaned = df_cleaned.loc[df["Rating"] <= 5]

df_cleaned["Price"] = df_cleaned["Price"].apply(lambda x: x.replace('$', ''))
df_cleaned["Installs"] = df_cleaned["Installs"].apply(lambda x: x.replace('+', ''))
df_cleaned['Installs'] = df_cleaned['Installs'].apply(lambda a: str(a).replace(',', '') if ',' in str(a) else a)
df_cleaned['Installs'] = df_cleaned['Installs'].apply(lambda a : int(a))
df_cleaned['Reviews'] = df_cleaned['Reviews'].apply(lambda a : int(a))

genres = df_cleaned["Genres"].value_counts().head().index

str_cols = ["Size", "Price"]

for col in str_cols:
    df_cleaned[[col]] = df_cleaned[[col]].fillna(value="")
    df_cleaned[col] = df_cleaned[col].apply(value_to_float)
```

Figure 3.3: Symbols Nullify

- Conversion of k into thousands

  In modern times K is being mostly used as a replacement of 1000 and its multiple's but k won't be suitable for analysis as we are processing numerically.

```
def value_to_float(x):
    # Convert the string feature into float/Integer
    #If there is 'K' or 'M' in the string, convert it to the corresponding number (1000 or 1000000)

    if type(x) == float or type(x) == int:
        return x
    if 'K' in x:
        if len(x) > 1:
            return float(x.replace('K', '')) * 10**3
        return 1000.0
```

Figure 3.4: Conversion of k into thousands

- **Mbs converted to kbs:**

  We have set our project standard as kbs because it is easier to express file size is shorter measuring standard.

```
if 'M' in x:
    if len(x) > 1:
        return float(x.replace('M', '')) * 10**6
    return 1000000.0

# If the string cannot be converted, return 0 instead
try:
    parsed_val = float(x)
except ValueError:
    parsed_val = 0.0
return parsed_val
```

Figure 3.5: Mbs converted to kbs

- **Deleted records having more than 5 rating:**

  As its not possible to have rating more than 5 because one can only rate upto 5 stars, so all those records with more than 5 rating were deleted.

```
In [17]: # Checking rating over 5
         df_cleaned.loc[df_cleaned["Rating"] > 5].values

Out[17]: array([], shape=(0, 13), dtype=object)
```

Figure 3.6: Mbs converted to kbs

- **7 records(review) greater than no: of installs:**

  In order to get pure insights we have standardized illusion that reviers can't be more than the number of installs, so we have deleted at approx 7 records having more reviews than number of installs.

```
In [18]: # There are 7 records where Reviews are greater than Installs
         df_cleaned[df_cleaned['Reviews'] > df_cleaned['Installs']].shape

Out[18]: (7, 13)

In [19]: # Dropping 7 records that have greater Reviews than Installs
         df_cleaned.drop(df_cleaned[df_cleaned['Reviews'] > df_cleaned['Installs']].index,inplace=True)
         df_cleaned[df_cleaned['Reviews'] > df_cleaned['Installs']].shape

Out[19]: (0, 13)
```

Figure 3.7: 7 records(review) greater than no: of installs

3.4 STEP 4: DETERMINE ANALYSIS

It is time to examine once the relevant data has been collected and cleared. The type of analysis we chose was strongly influenced by the questions or goals we identified in the first step. Data is used in diagnostic analysis to find the reason and solution to an issue. Descriptive analysis is a technique of describing essential areas of data to understand it. Predictive analysis provides future performance predictions for particular metrics using historical data and statistical modelling. For example, if you wanted to determine how the created application will change by fiscal quarter, then might utilise predictive analysis on data from previous years. The observed trends might then be used to predict revenue for the next year. We have gone for the descriptive type of analysis to find what is happening with the applications and its growth parameters. This would help us to identify and differentiate factors on which growth is directly or indirectly dependent.

3.4.1 APPLICATION TYPE:

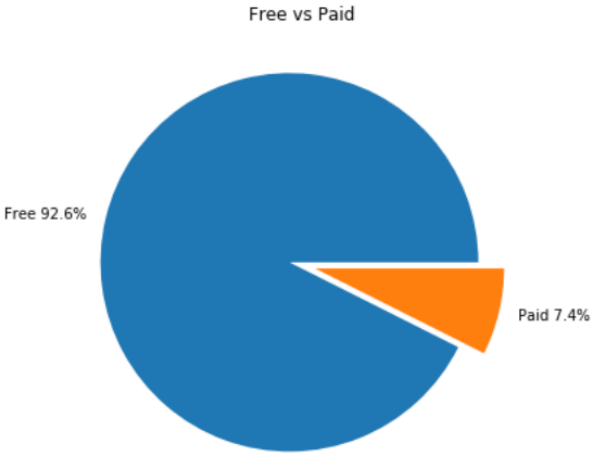


Figure 3.8: Application Type

92.6% app are Free on Play Store and only 7.4% are Paid

3.4.2 TOP CATEGORIES ON GOOGLE PLAYSTORE:

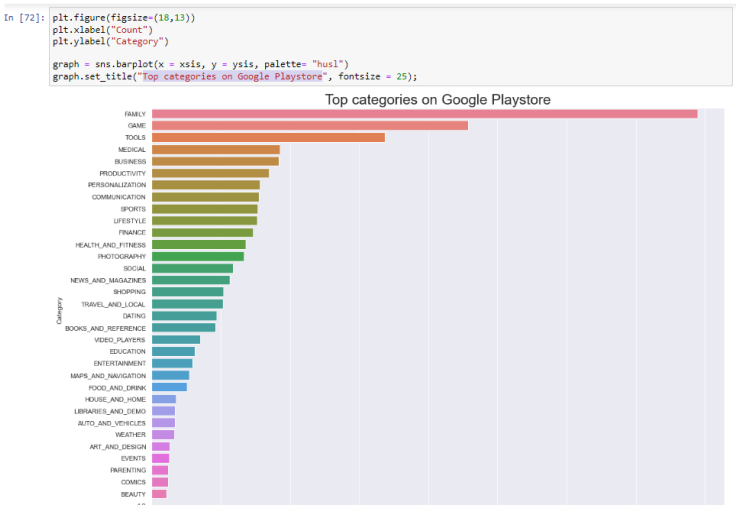


Figure 3.9: Top categories on Google Playstore:

3.4.3 CATEGORIES ARE OF TYPE PAID

```
paid_data = df[df.Type=='Paid']
px.bar(paid_data,x = paid_data['Category'].value_counts().index, y = paid_data['Category'].value_counts().values, color = paid_da
```

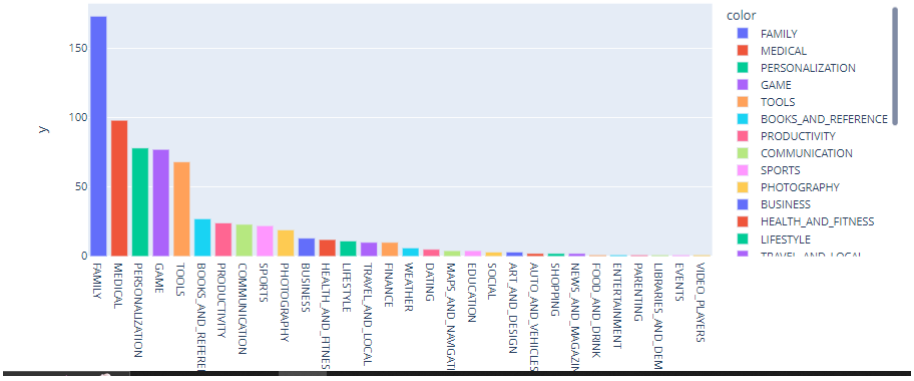


Figure 3.10: categories are of Type paid

3.4.4 WHICH CATEGORY WITH PAID APPS HAS HIGHEST RATINGS

```
px.bar(paid_data, x = paid_data['Rating'].groupby(by = paid_data['Category']).mean().sort_values().dropna().index,
       y = paid_data['Rating'].groupby(by = paid_data['Category']).mean().sort_values().dropna().values,
       color = paid_data['Rating'].groupby(by = paid_data['Category']).mean().sort_values().dropna().index)
```

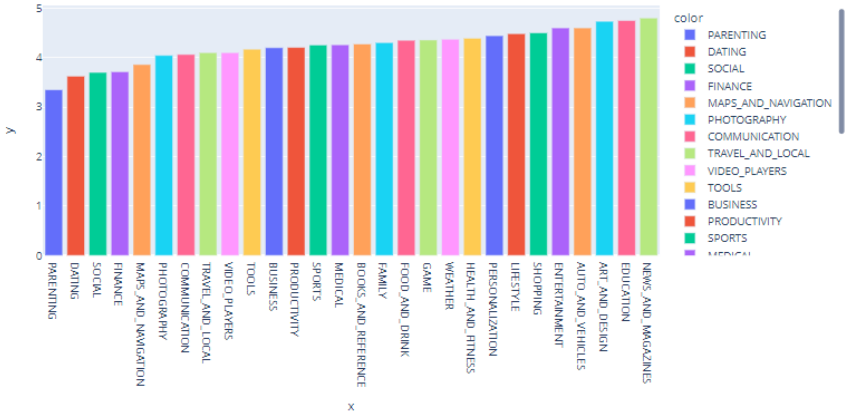


Figure 3.11: category with paid apps has highest ratings

3.4.5 RATINGS OF THE FREE VS PAID APP:

```
In [65]: col='Content Rating'
         v1=d1[col].value_counts().reset_index()
         v1=v1.rename(columns={col:'count','index':col})
         v1['percent']=v1['count'].apply(lambda x : 100*x/sum(v1['count']))
         v1=v1.astype(str).sort_values(col)
         v2=d2[col].value_counts().reset_index()
         v2=v2.rename(columns={col:'count','index':col})
         v2['percent']=v2['count'].apply(lambda x : 100*x/sum(v2['count']))
         v2=v2.sort_values(col)
         trace1 = go.Scatter(x=v1[col], y=v1["count"], name="Free", marker=dict(color="#a678de"))
         trace2 = go.Scatter(x=v2[col], y=v2["count"], name="Paid", marker=dict(color="#6ad49b"))
         y = [trace1, trace2]
         layout={'title':"Ratings of the free vs paid app",'xaxis':{'title':"Ratings"}}
         fig = go.Figure(data=y, layout=layout)
         iplot(fig)
```

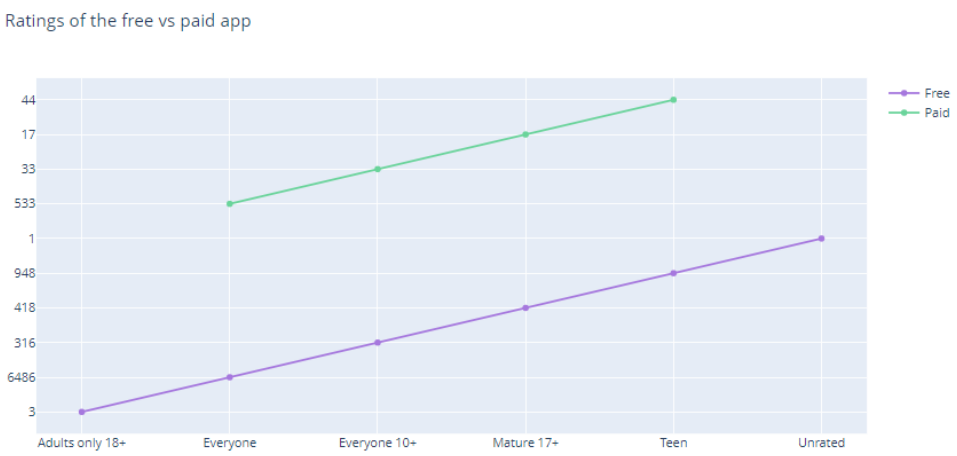


Figure 3.12: category with paid apps has highest ratings

3.5 STEP 5: INTERPRET RESULTS

Analyzing the results after data analysis is essential. In other words, what are you taking away from the findings of your analysis? We have interpreted our results by plotting visualizations on dashboard to indicate the features, factors and growth rate. We have use plotly, matplotlib and seaborn for visaulizations. With that we have also used Business intelligence tool named "Qlik Sense" to build dashboard.

Qlik sense iss further discussed in detail in the Chapter No:4.

CHAPTER 4

TOOLS AND TECHNOLOGIES

4.1 TOOLS

4.1.1 SERVER TOOLS

4.1.1.1 Jupyter-Notebook

In the early years, scientists utilized a lab notebook to record test findings, progress, and conclusions. Jupyter notebook is the latest and modern application that admits data scientists to finish analytic processes in same way that other scientists handled lab notebooks earlier. The Jupyter notebook was constituted as an iPython project. The iPython project was created to present interactive online approah to Python. Over opportunity, it enhanced help to connect with other data analytics tools, such as are in the same way that the break from Python resulted in the present incarnation of Jupyter. Jupyter Notebopok is a package manager that is both cross-platform and likewise language agnostic. We can use anaconda to install any third party packages. Jupyter Notebook is an interactive Web User Interface environment to construct notebook documents for R language and python language. Jupyter notebook Features:

• Ease of Use

- Straight-Out-the-Box Functionality
- File Management
- Patching Updates
- Help Guides

Use of Jupyter-Notebook in Our Project:

We used Jupyter-Notebook to showe our analysis work by using different charts or graphs.

4.2 TECHNOLOGIES

4.2.0.1 DATA COLLECTION AND CLEANING

Numpy:

A Python library called NumPy allows you to assist with arrays. It also has functions for matrices, the Transform, and the associated linear algebra. Travis Oliphant founded NumPy in the year 2005. You are free to use this open-supply responsibility. The official name of NumPy is Numerical Python. In Python, lists may be used like arrays, although they are slow to process. With NumPy, array items should be up to 50 times quicker than Python lists traditionally. Working with arrays is a simple because to a variety of helpful features that are included in NumPy's array object, also known as an array. For the project's mathematical and scientific calculations,

NumPy is used.

Pandas:

Pandas is a python library used for manipulation and analysis of datasets. It comprises of features for analyzing, cleaning, exploring, and manipulating facts. It permits us to research large facts and draw final output primarily based totally on statistical theories. It can ease up difficult fact units and convert them into be readable and relevant. Relevant facts could be vital in facts science. Pandas also can delete rows that might be beside the point or include incorrect values, consisting of empty or NULL values. This is known as fact sanitization. Pandas is used in this project for data analysis and for performing operations such as merging, reshaping as well as data cleaning.

Seaborn:

Python's Seaborn visualisation package allows for the plotting of statistical visuals. It includes basic design elements and colour schemes to make statistics charts more attractive. It is constructed on top of the Matplotlib toolkit and is closely linked with the Pandas data structures. With Seaborn, visualisation will be at the root of data exploration and comprehension. In order to better comprehend the dataset, it offers dataset-oriented APIs that allow us to move be-

tween several visual representations of the same variables.

Matplotlib:

Matplotlib is a charting package included with NumPy, a virtual big fact control tool, for the Python programming language. To combine charts in Python applications, Matplotlib uses an item-oriented API. In this project, Matplotlib is used for data visualization and displaying data in the form of graphs in order to better comprehend the datasets.

Plotly:

An open-source graphing package for Python is called Plotly. It could be a very helpful tool for clearly and confidently analyzing the data and visualising it. Plotly chart objects provide a high level, user-friendly interface for plotly. It can draw a variety of graphs and charts, including pie charts, scatter plots, box plots, histograms, and bar charts. Plotly ExpressPX is the common import name for the plotly.express module, which has functions that may instantly make out whole figures. The plotly library comes with Plotly Express, which is the preferred place to start when creating the majority of common figures.

4.3 FRONT AND BACK END TECHNOLOGIES

4.3.1 PYTHON:

Python is a high level, general-purpose interpreted language with simple syntax and dynamic semantics. Guido V Rossum designed it in 1989. Python is the language of choice for both novices and experts. Python is an open source language, which means that anybody may use it. Python may be used to create desktop, online, and mobile applications. Python has a fantastic community that is continuously creating new libraries and assisting those in need.

Features in python

- Free of cost and open source
- Simple to implement
- Contains large number of libraries
- Embeddable and extensible
- Interpreted
- Highly portable
- Object-oriented

4.3.2 QLIK SENSE:

Qlik Sense a self service BI(business intelligence) software, it lets developers or users develop applications with interactive dashboards showing visualization components such as diagrams, charts, KPI-objects (key performance indicators) and tables. A Qlik Sense application is deployed as an application that, in distinction to static Business Intelligence tools, enables interactivity with the end user. Each application contains one or more dashboards with data visualization objects showing various measures and dimensions such as time or volume. These measures and dimensions are derived from the data loaded into the Qlik Sense application and defined by scripts. Figure 1 illustrates a Qlik Sense dashboard with different visualizations.

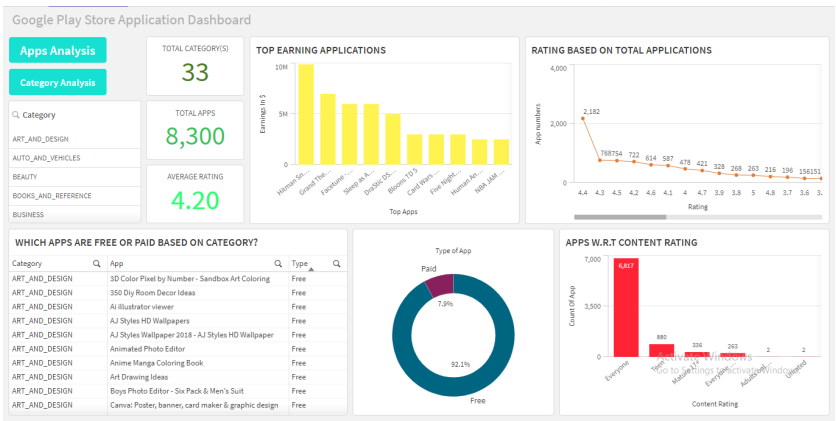


Figure 4.1: A Qlik Sense dashboard with different visualizations - a pie chart, a bar chart, line chart, and a table.

Data Set: At the start-up of an application the data is loaded into the memory of the user's computer in accordance with a load script. This data is the initial working set from which selections and calculations may be derived. To make a selection, the user just clicks on the part or any multiple parts of one or more visualizations that correspond to the subset of dataset that is of interest and the working set changes accordingly. Each selection thus works as a filter and all selections may be set and unset interchangeably.

Scripting: In order to make full use of the Qlik Sense software one must be well acquainted with its script syntax and many chart functions. This section provides information regarding when and how scripts are used along with examples of commonly used scripting functions. A full manual (QlikTech, Script syntax, 2016) of the Qlik Sense script syntax and functions can be found on their official website (QlikTech, 2016).

The Load Script: Before it is possible to create visualizations in a Qlik Sense application data must be created or loaded from one or multiple sources. The source from where the data is to be loaded as well as how this data is to be set up is defined by a load script. The simplest way to work with qlik sense is just to drag-and-drop, for sample an xls extension file consisting of the data fields, directly

into the application. Doing so will start a windows page that let the user pick and choose data fields to be loaded in it and Qlik Sense then fastly fill out the load script in accordingly with the choices of users. It is however possible to do many different manipulations on the data that is to be loaded and also to generate new data. In order to do this way the load-script must have to be edited manually.

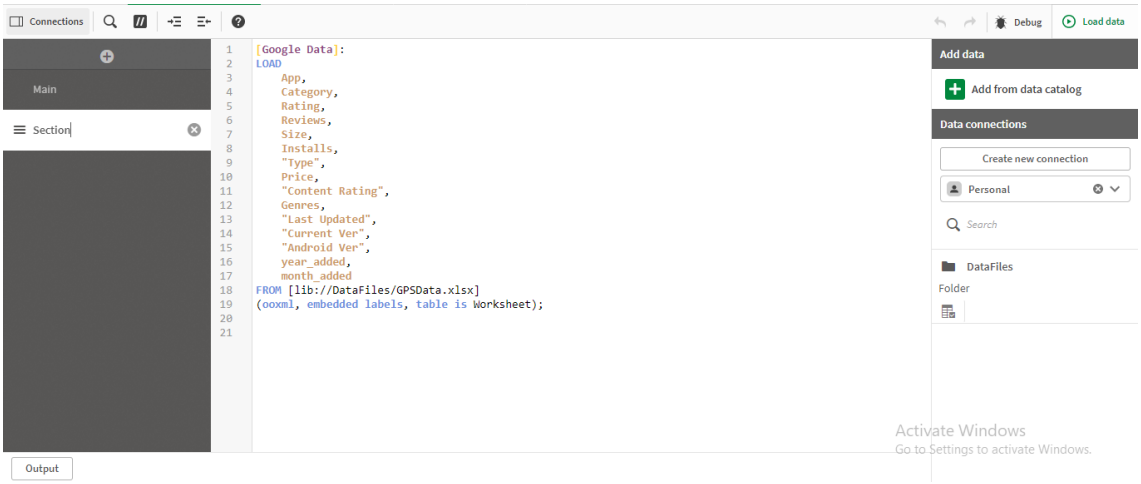


Figure 4.2: The Qlik Sense App's Data load editor where data can be loaded by uploading from datafiles or by writing scripts.

Visualizations and Dashboards: Qlik Sense offers 13 different visualization objects such as line charts, pie charts and the Text and image object for presenting data and information in various ways or for making selections. To create a new visualization the developer simply drags it from the library pane onto a dashboard. Most visualization graphs have one or more than one dimensions and one or more than one measures. These dimensions and measures are de-

fined by script fields. The dimension fields can either be simply a name of a data field such as “App” or “Category” or a more or less complicated script as to display a subset of one or more fields’ category items or a conditional script for e.g. using a different dimension depending on the current data selection. The measure field always uses scripts that defines what is to be measured and how, such as the sum of e.g. “Price” or the count of elements of a certain field. In visualization objects that also have a dimension the measure will per default be calculated for each of the dimension’s category items respectively.

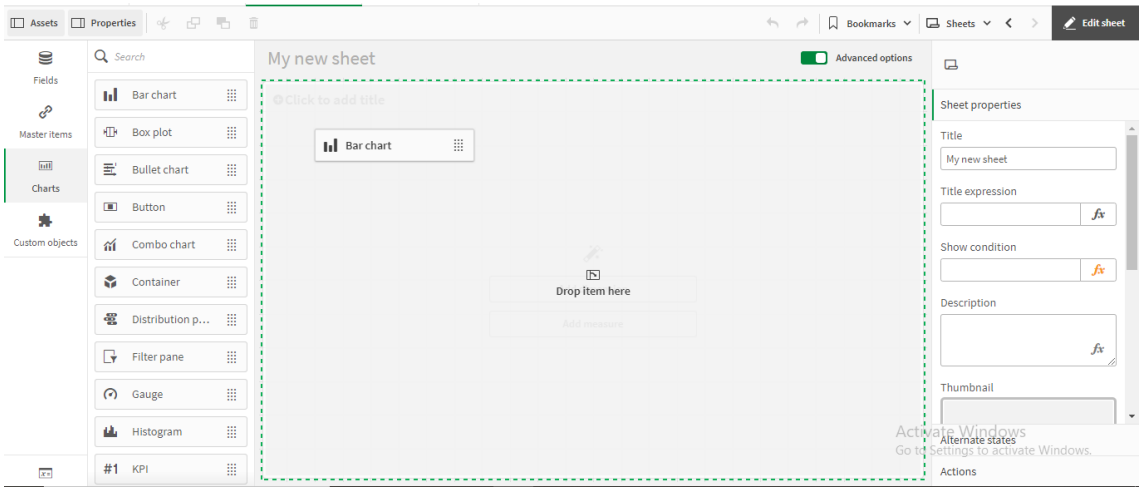


Figure 4.3: The Qlik Sense App Overview with an empty dashboard grid and a bar chart being dragged into the empty grid from the library pane.

Master Items: Qlik Sense supports front-end customizability by implementing master items, alternative measures along with the ability to create new visualizations or to drag-and-drop existing visu-

alizations onto dashboards. The master items, which consist of measures, dimensions and visualizations predefined by the developer, are all accessible from the library pane. The purpose of the master items is to facilitate reusability and allow for non-developer end users to create customized dashboards and visualizations.

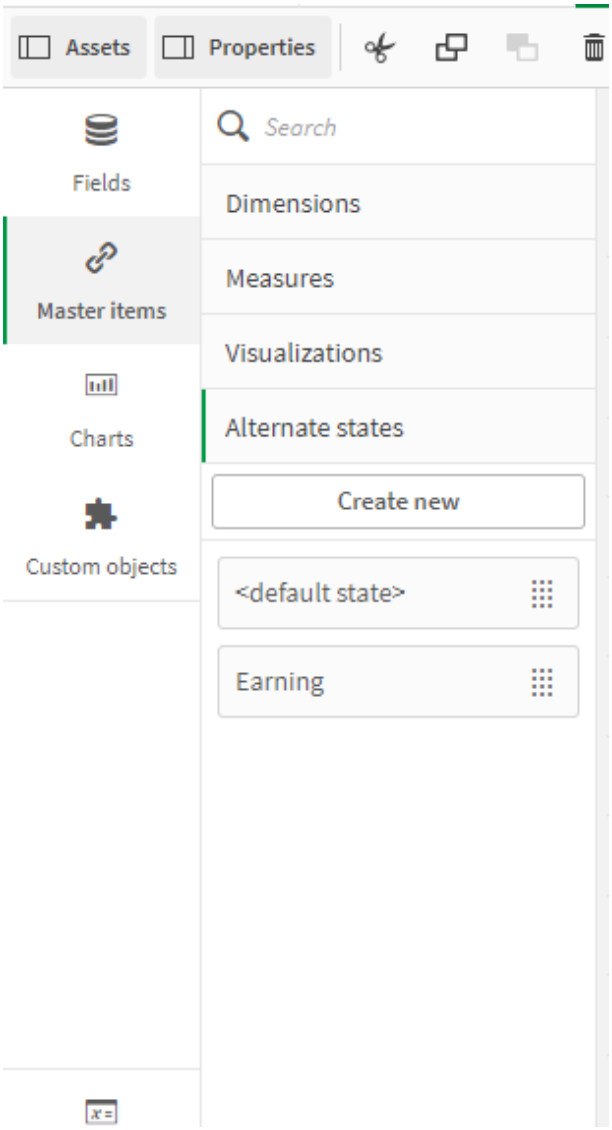


Figure 4.4: Showing the library pane tab Master items and its search function.

The master items structure in Qlik Sense is made of three alphabetically ordered lists where one contains dimensions, one contains measures and one contains visualizations. Within each one of these master item lists, no hierarchy exists and the items cannot be reordered. Qlik Sense however provides a search function which matches the searched term with both the names of the master items as well as the tags assigned to them. Figure 4 illustrates the use of this search function.

Choosing the Right Data Visualization: To show visualization different types of graphs can be used and it's important to choose suitable one for both the data to be displayed and the message that is to be communicated. Different business intelligence tools come with different advantages and limitation and offers different visualization selections . The available visualization objects in Qlik Sense are listed in Table 1 along with their purposes.

Bar charts are suitable for comparing values of multiple category items. The bar chart visualization in Qlik Sense has a limitation of up to two dimensions for one measure and up to 15 measures when only one dimension is used . The standard bar chart uses one dimension (e.g. Apps) and one measure (e.g. sum of price). The stacked and grouped bar charts use multiple measures such as

sum of sales per region and sum of sales of a specific item etcetera and are suitable for communicating multiple relations. Grouped bar charts are more suitable when the specific value of each measure is of importance while stacked bar charts better show each measures in relation to the whole. The dimension of a bar chart in Qlik Sense can be placed on either x axis for vertical allignment, or on the y axis for horizontal allignment.

Line charts are suitable for showing trends of continuous measures over evenly spaced time or categories. Line charts has the same limitations in Qlik Sense for number of dimensions and measures as bar charts. The data set must have at least two measure values as to draw a line. In Qlik Sense the data values that connect the lines may be shown as points. Missing values may be presented as either gaps, connections or zeros. There is also an option to show the area beneath the lines and if used, to either stack or overlay these areas.

Combo charts uses a combination of the line chart and the bar chart. The combo chart is suitable for displaying values of different measure scales as it may have one scale at the left side of hand and another at the right side of hand. It is also suitable for distinguishing multiple measures that tend to have equal values at many following dimension steps since its measures may be depicted as either trend

lines, bars or data points of various shapes. The combo chart in Qlik Sense only allows for one dimension to be used.

Gauges are used to visualize states of a single measure as to help the user interpret its value. It is usually used for showing the states good (green), satisfactory/neutral (yellow) and bad (red) of a measure but Qlik Sense also allows for other colors along with a legend and single color gauges.

KPI stands for key performance indicator and the KPI-object in Qlik Sense simply show wither one or two measures without any dimension. It is suitable for highlighting values of importance. In Qlik Sense the color of the KPI-objects' values can be coded as to indicate its states and the elements can also be set as a clickable link to another dashboard.

Filter pane The filter pane object in Qlik Sense is used to let the user define the current Qlik Sense data selection. Any number of dimensions may be used as filters and the filter pane will show a green bar for each dimension that represents how many of the dimensions total elements that are currently in the data set.

Tree map Tree maps display hierarchical data as nested rectangles. It uses one measure and one or many dimensions. It is suitable for showing the relative size of measures over different dimension

as each rectangle's size corresponds to its dimension's relative contribution to the whole of its containing rectangle. Qlik Sense also allows for different colors to be used to distinguish between dimensions/categories at different hierarchal levels.

Pie charts are suitable for showing the relations between a single measure over different categories of data as well as the relation of each category to a whole. A category is represented as a piece of the pie and its size corresponds to its contribution to the total.

Tables are suitable for showing multiple measures connected to one or more categories or dimensions. It is a good choice for showing many exact values.

Pivot tables are suitable for letting the user reorganize a tables dimensions and measures as to see different subtotals of various categories.

Text and image The text and image object in Qlik Sense is used for adding static text and images to a dashboard. Measures, hyperlinks and links to dashboards can also be inserted to this object.

Scatter plots are suitable for showing values from two or three measures for category items. Two values are represented by the data points positions on the x and y axis's and a third measure can be represented by varying sizes of the data points.

Map The map object can be used when data is connected to pointers of a picture. It is usually used to present regional data.

CHAPTER 5

RESULTS AND DISCUSSIONS

After cleaning and preprocessing our data of GPS Applications, we have come up with results by displaying it on dashboard of qlik sense. As mentioned before qlik sense has good user interactivity. We will now be discussing our end results by performing some activity on dashboard. Following figure is the main sheet of dashboard which will be displayed to our users/developers.

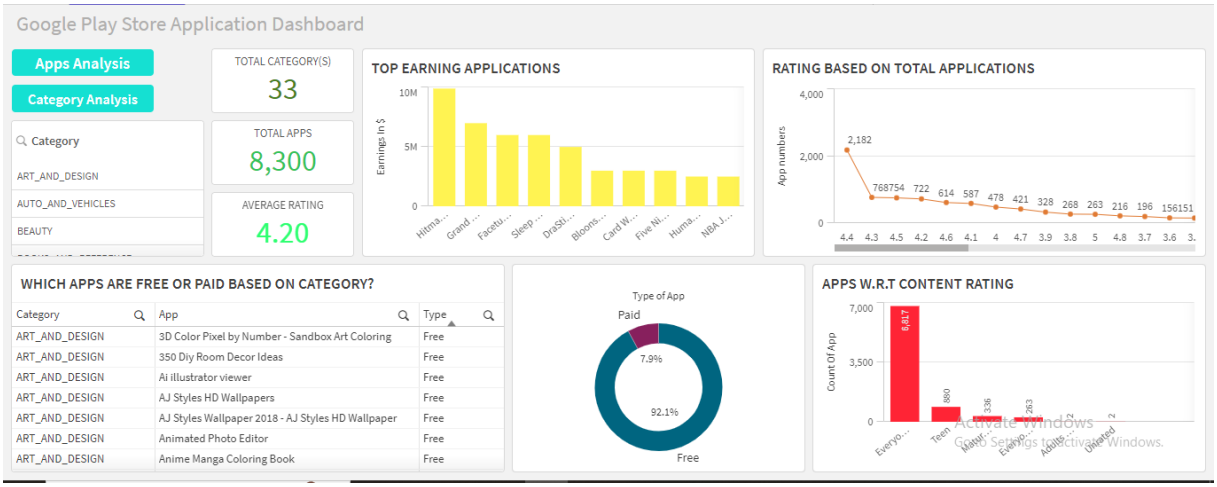


Figure 5.1: Qlik Sense Dashboard displaying all the important visualization graphs.

Analysis displayed on the main sheet of dashboard are total numbers of categories, total apps in our dataset, average rating of all the apps present in dataset, top earning applications, rating based on total applications, which apps are free or paid based on category selection, free and paid percentages,and total apps with respect to

content rating. Buttons named as Apps Analysis and Category Analysis will navigate to other sheets of the dashboard. All these analysis will help developers, users, and business developers to search for their required data and help them out in their respected fields. We got average rating 4.2 of about 8300 distinct apps of our dataset, 92.1% apps are of free type and 7.9% apps are of paid type, application named "Hitman Sniper" of category "GAME" has made hightest earning in dollars.

We can also search for results on our own choice just by doing selection where ever required. For eg. in below given figure "PHOTOGRAPHY" category is selected.

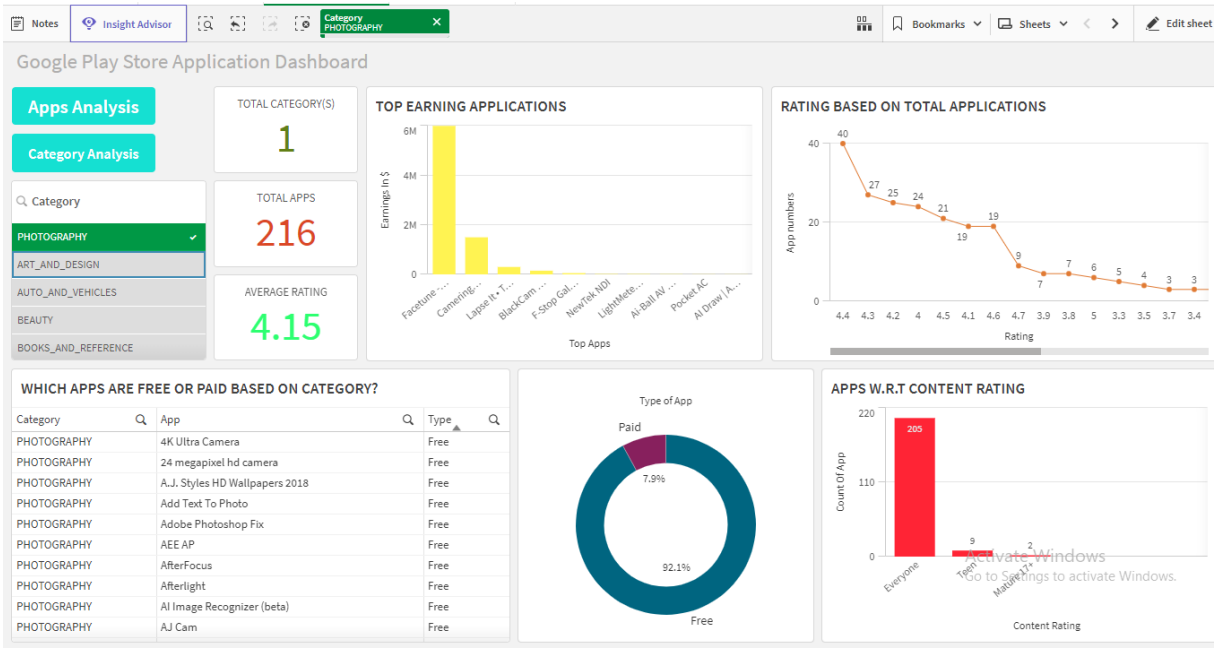


Figure 5.2: Qlik Sense Dashboard made changes according to selection.

We can now see the analysis have filtered according to selection i.e all the graphs are displaying data of photography category.

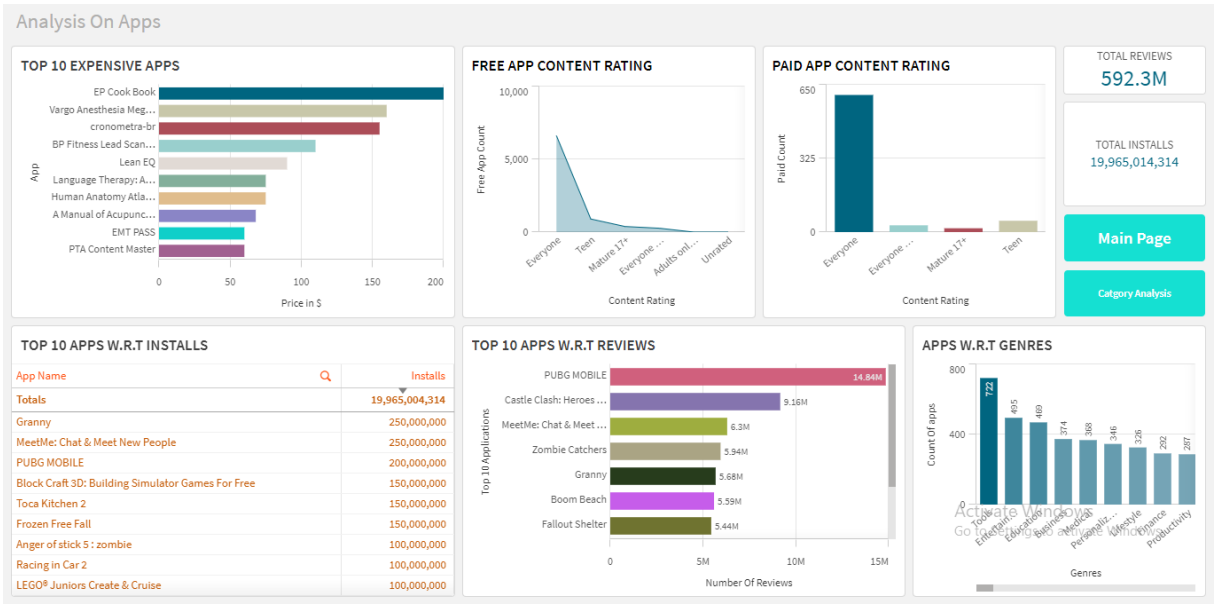


Figure 5.3: Qlik Sense Dashboard's second sheet displaying analysis on Apps

Figure 5.3: is showing all the visualizations related to applications i.e top 10 expensive apps, top 10 apps w.r.t installs, top 10 apps w.r.t reviews, free app content rating count, paid paid content rating count, and apps w.r.t genres. KPIs showing total reviews and total installs in data set. These visualizations are beneficial for those who just want to get help regarding apps only.

is showing all the visualizations related to categories i.e top 10 expensive categories, average rating per category, top 10 categories based on installs, categories w.r.t reviews and rating, and total apps per category.

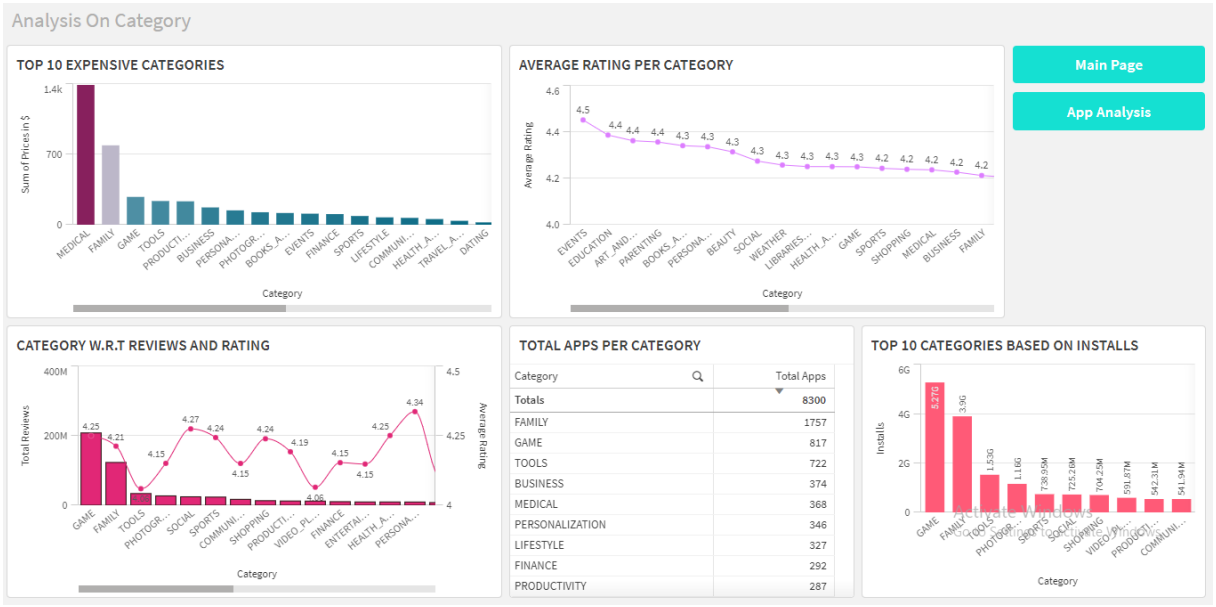


Figure 5.4: Qlik Sense Dashboard's second sheet displaying analysis on category.

These visualizations are beneficial for those who just want to get help regarding categories only.

CHAPTER 6

CONCLUSION

6.1 CONCLUSION

The Google Play Store Apps analysis offers some valuable insights on how the apps in the store are trending. According to the graph visualisations above, even though there are twice as few applications in these categories as there are in the category FAMILY, the majority of the trending apps (in terms of user installations) come from the categories of GAME, COMMUNICATION, and TOOL. These applications' popularity is most likely a result of their capacity to entertain or benefit the user. Besides, it also shows a good trend where we can see that developers from these categories are focusing on the quality instead of the quantity of the apps. In moreover, the data above suggest that the majority of applications with strong ratings of above 4.0 are generally proven to have a significant number of reviews and user installations. Although there are some price and size increases, they shouldn't be interpreted as meaning that applications with high ratings are often large and expensive because, based on the graphs, they are most likely the result of a small minority. However, the majority of the applications with a lot of reviews fall under

the SOCIAL, COMMUNICATION, and GAME categories. Examples of these apps are Facebook, Whats App Messenger, Instagram, Messenger - Text and Video Chat for Free, Clash of Clans, etc. Even while the most popular, highly rated, and reviewed applications in the categories of GAME, SOCIAL, COMMUNICATION, and TOOL match the current trend of Android users, they do not even feature as a category in the top 5 most costly apps in the market (which are mostly from FINANCE and LIFESTYLE). As a result, we discovered that the majority of the popular Android applications fall into one of three categories: helping, communicating, or entertaining.

6.2 FUTURE WORK

6.3 REFERENCES:

[1] Maredia, R. Analysis of Google Play Store Data set and predict the popularity of an app on Google Play Store , 2020.

[2] Kaggle.com.(2018).GooglePlay Store Apps.[online] https://www.kaggle.com play-storeapps [Accessed 3 Mar. 2020].

[3] Aralikatte, R., Sridhara, G., Gantayat, N., and Mani, S. (2018). Fault in your stars: an analysis of android app reviews. In Proceedings of the ACM India Joint International Conference on Data Science and Management of Data, pages 57–66. ACM.

[4]Google play store: number of apps 2018(2018). [online] https://www.statist
statistics/266210/number-of-available-applications-inthe-google-play-store/ [Accessed 3 Mar. 2020].

[5] [2015].Grover, S. 3 apps that failed (and what they teach us about app marketing). [online] https://blog.placeit.net/apps-fail-teach-us-appmarketing/.

[6] Fu, B., Lin, J., Li, L., Faloutsos, C., Hong, J., and Sadeh, N. (2013). Why people hate your app: Making sense of user feedback in a mobile app store. In Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining, pages 1276–1284. ACM

[7]Harman, M., Jia, Y., and Zhang, Y. (2012). App store mining and analysis: Msr for app stores. In 2012 9th IEEE Working Conference on Mining Software Repositories (MSR),pages 108–111

[8] Saxena, P. (2018). How much money can you make with your app. [online] https: //appinventiv.com/blog/how-much-money-can-you-earn-through-your-mobile-app.

[9] Ruiz, I. J. M., Nagappan, M., Adams, B., Berger, T., Dienst, S., and Hassan, A. E. (2016). Examining the rating system used in mobile-app stores. IEEE Software, 33(6):86– 92.

[10] Islam, M. R. (2014). Numeric rating of apps on google play

store by sentiment analysis on user reviews. In 2014 International Conference on Electrical Engineering and Information Communication Technology, pages 1–4.

[11] Jong, J. (2011). Predicting rating with sentiment analysis. [online] http://cs229.stanford. edu/proj2011/Jong-PredictingRatingwithSentim

[12] Valentine, A. (2017). 4 mobile app developer success stories. [online] https://blog. proto.io/4-mobile-app-developer-success-stories/

[13] Pang, B., Lee, L., and Vaithyanathan, S. (2002). Thumbs up?: sentiment classification using machine learning techniques. In Proceedings of the ACL-02 conference on Empirical methods in natural language processing-Volume 10, pages 79–86. Association for Computational Linguistics

[14] A. Di Sorbo et al., "What would users change in my app? summarizing app reviews for recommending software changes," in Proceedings of the 2016 24th ACM SIGSOFT International Symposium on Foundations of Software Engineering, 2016, pp. 499-510: ACM

[15] Tuckerman, C. (2014). Predicting mobile application success.