CHAPTER 1

INTRODUCTION

The Google Play Store is a vast marketplace hosting millions of mobile applications across various categories. As an essential platform for app developers and users alike, understanding the factors that influence app ratings can be instrumental in improving app success and user satisfaction. In this mini-project, we aim to conduct an in-depth analysis of apps available on the Google Play Store to identify the key factors that contribute to higher or lower app ratings.

1.1 MOTIVATION FOR THE WORK

- Success Recognition: The desire to be recognized and appreciated for achievements.
- Personal Growth: Continuous improvement and development of skills.
- Intrinsic Interest: Enjoyment and passion for the work itself.
- Sense of Purpose: Feeling that the work has a meaningful impact.
- Team Collaboration: Building strong relationships and working together.
- Challenging Tasks: Engaging in stimulating and rewarding challenges.
- Autonomy: Having control and freedom in decision-making.
- Work-Life Balance: Striving for a healthy balance between work and personal life.

CHAPTER 2

SYSTEM ANALYSIS

2.1 EXISTING SYSTEM

The existing system is designed to provide a comprehensive understanding of the Google Play Store's app ecosystem. It aims to identify trends, preferences, and potential factors that influence app ratings and success. The outcomes of this mini-project are valuable insights that can be used for strategic planning, app development, and market analysis.

2.2 PROPOSED SYSTEM

The proposed system for the Google Play Store app data analysis aims to build upon the existing system by incorporating advanced techniques and focusing on specific aspects of the app ecosystem. The proposed system includes the following key components:

- Enhanced Data Collection
- Sentiment Analysis
- App Recommendation System
- User Segmentation
- Time-Series Analysis
- Machine Learning Models
- Interactive Data Visualization

2.3 FEASIBILITY

- Conduct a feasibility study to assess the viability of the proposed project.
- Evaluate technical, economic, operational, legal, and scheduling factors.
- Identify potential risks, benefits, and challenges
- Determine whether the project is achievable within constraints.
- Based on the findings, make an informed decision on project initiation.

2.3.1 Data Feasibility

Data feasibility in the Google Play Store app data analysis mini project involves assessing the availability and suitability of data for conducting the analysis. Here are key considerations for data feasibility.

- Data Availability
- Data Quality
- Data Accessibility
- Data Privacy and Legal Compliance
- Data Volume and Complexity

2.3.2 Technical Feasibility

Technical feasibility in the Google Play Store app analysis mini project involves evaluating the technological requirements and capabilities necessary for conducting the analysis. Here are key considerations for technical feasibility:

- Data Collection Methods
- Data Storage and Management
- Computing Power
- Data Processing Tools
- Machine Learning Models
- Visualization Tools

2.3.3 Operational Feasibility

Operational feasibility in the Google Play Store app data mini project involves evaluating whether the project can be successfully executed within the organisation's existing operations and resources. Here are key considerations for operational feasibility:

- Resource Availability
- Team Readiness
- Collaboration and Coordination
- Time Constraints
- User Acceptance
- Cost Considerations

2.3.4 Financial Feasibility

Financial feasibility in the Google Play Store app analysis mini project involves assessing the financial viability and cost-effectiveness of conducting the data analysis. Here are key considerations for financial feasibility:

- Budget Allocation:
- Cost of Data Collection
- Data Processing Costs
- Tools and Software Costs
- Human Resource Expenses
- Training Costs
- Cost-Benefit Analysis

2.3.5 Legal Feasibility

Legal feasibility in the Google Play Store app analysis involves evaluating whether the project complies with relevant laws, regulations, and ethical guidelines. Here are key considerations for legal feasibility:

- Data Privacy and Consent:
- Terms of Use
- Intellectual Property Rights
- User Anonymity
- Ethical Data Handling
- Regulatory Compliance
- Third-Party Data

CHAPTER 3

SYSTEM SPECIFICATIONS

3.1 HARDWARE SPECIFICATIONS

- Processor : Intel core i3
- RAM : Min 4GB
- Hard Disk : Min 100GB
- Key Board : Standard Windows Keyboard
- Mouse : Two Button Mouse
- Monitor : SVGA

3.2 SOFTWARE SPECIFICATIONS

- Operating System : Windows family
- Technology : Python3.7
- IDE : Jupyter Notebook

CHAPTER 4

SOFTWARE DESCRIPTION

4.1 PYTHON

Python is an interpreted high-level general-purpose programming language. Pythondesign philosophy emphasises code readability with its notable use of significant indentation. Its language constructs as well as its object-oriented approach aim to help programmers write clear, logical code for small and large-scale projects. Python is dynamically-typed and garbage-collected. It supports multiple programming paradigms, including structured (particularly, procedural),object-oriented and functional programming. Python is often described as a "batteries included" language due to its comprehensive standard library. Guido van Rossum began working on Python in the late 1980s, as a successor to the ABC programming language, and first released it in 1991 as Python 0.9.0. Python 2.0 was released in 2000 and introduced new features, such as list comprehensions and a garbage collection system using reference counting. Python 3.0 was released in 2008.

Advantages of Python

1) Simple and Easy to learn

Python is extremely easy and simple to learn, so python is easy to read or easy to learn. It closely resembles the English language. Therefore, it comes in one of the greatest advantages of python. It is a very powerful language, and it takes no skills to learn python, so python is free and open source. It is a high – level language.

2) Portable and Extensible

Since python is portable, it is supported by all the platforms of the industries like Windows, Linux, Macintosh, and play stations support python. And with the extensibility of python, it can completely integrate java as well as dot net components even it can invoke C and C++ libraries as well, and these are all the advantages of python.

3) Object-Oriented Programming

Python supports orientation programming; it permits polymorphism and inheritance. Python user get to use the shareable categories thus, code may be reusable and additionally provide the protection mechanism by abstracting knowledge. It is additionally wide accustomed to developing prototypes that modify the computer used to straightforward scan and write.

4) Artificial Intelligence

Artificial Intelligence means that a machine program that acts or responds to human brain intelligence is done through lots of algorithm or programs. It is combined with scikit-learn python, which can do complex calculations with just a single statement. Furthermore, libraries such as Keras and TensorFlow ping machine learning functionality into the mix. Python also has libraries such as open CV that helps in image recognition, such as computer vision, and another feature of python. It can detect face or speech recognitions.

5) Testing Framework

Python is nice for good ideas or product for a startup is a company currently making a code isn't a simple task because it contains several methods ranging from its style its code additionally the checked cases and code testing doubtless the foremost vital process of the code life cycle it's also the difficult task for a code tester to settle on the simplest programming language for automation testing and also the python is that the best resolution for this downside thus it's several integrate testing frameworks that cover debugging and quickest workflows currently their area unit several tools and modules to create the items a lot of easier like chemical element that is the style in automation tool, and thus that have got a splinter, so it supports testing with cross platform and cross-browser with frameworks like Pytest and Henry M. Robert frame book currently it's several nice code testing supports.

6) Data Science

Python is the leading language for several information scientists currently for years. Academy students and PY field researchers were exploiting the MATLAB language for research project currently that each one began to the modification with the discharge of fight the numerical engines such as NumPy and Pandas python additionally deals with the tabular matrix likewise as applied mathematics information. It also visualises it with common libraries such as matplotlib and Sea bourn.

4.2 ANACONDA

Anaconda is a free and open-source distribution of the Python and R programming languages for scientific computing (data science, machine learning applications, large scale data processing, predictive analytics, etc.), that aims to simplify package management and deployment. Package versions are managed by the package management system. The Anaconda distribution includes data-science packages suitable for Windows,Linux, and MacOS. Anaconda Navigator is a desktop graphical user interface (GUI) included in Anaconda distribution that allows users to launch applications and manage conda packages, environments and channels without using command-line commands. Navigatorcan search for packages on Anaconda Cloud or in a local Anaconda Repository, install them in an environment, run the packages and update them. It is available for Windows, MacOS and Linux.

In order to run, many scientific packages depend on specific versions of other packages. Data scientists often use multiple versions of many packages and use multiple environments to separate these different versions. The command-line program conda is both a package manager and an environment manager. This helps data scientists ensure that each version of each package has all the dependencies it requires and works correctly.

Navigator is an easy, point-and-click way to work with packages and environments without needing to type conda commands in a terminal window. It can use it to find the packages that want, install them in an environment, run the packages, and update them – all inside Navigator.

- JupyterLab
- Jupyter Notebook
- Spyder
- PyCharm

4.3 JUPYTER NOTEBOOK

- Jupyter Notebook is an interactive computational environment that allows users to create and share documents that contain live code, equations, visualizations, and narrative text. It supports a wide range of programming languages, including Python, R, and Julia.
- In a Jupyter Notebook, you can write and execute code in cells, and the output of each cell is displayed inline. This makes it a powerful tool for data exploration, prototyping, and collaboration.
- Jupyter Notebooks can be used for various tasks such as data cleaning, visualization, statistical modeling, machine learning, and more. It is widely used by data scientists, researchers, and educators.
- One of the main advantages of Jupyter Notebook is its ability to create reproducible research. By sharing the notebook with others, they can reproduce your work and verify your findings.
- Jupyter Notebook is an open-source project and can be installed on your computer or used online through services like Google Colab and Microsoft Azure.

4.4 JUPYTER LAB

JupyterLab is an interactive development environment (IDE) for working with Jupyter Notebooks, code, and data. It is an evolution of the Jupyter Notebook interface, with additional features and improvements. JupyterLab provides a modern, web-based interface that allows you to work with multiple notebooks, code files, and consoles in a single, tabbed interface. It also includes a file browser, a terminal, and a powerful text editor with syntax highlighting, auto-completion, and other features.

One of the main advantages of JupyterLab over the classic Jupyter Notebook interface is its extensibility. JupyterLab allows you to customize the interface by installing extensions, which can add new functionality to the IDE, such as a table of contents, a debugger, or a code formatter. JupyterLab also includes features for collaboration, such as a shared clipboard and a feature for live collaboration on notebooks

CHAPTER 5

PROJECT DESCRIPTION

5.1 PROBLEM DEFINITION

The Google Play Store App Analysis Mini-Project aims to conduct a comprehensive analysis of apps available on the Google Play Store. The primary objective is to gain valuable insights into app attributes, user preferences, and factors influencing app ratings. By leveraging data analysis and visualization techniques, this project will provide actionable recommendations for app developers and businesses to improve app performance, enhance user satisfaction, and make data-driven decisions.

5.2 OVERVIEW OF THE PROJECT

The Google Play Store App Analysis Mini-Project aims to conduct a comprehensive analysis of the vast and diverse collection of mobile applications available on the Google Play Store. The project seeks to gain valuable insights into various aspects of the app ecosystem, including app categories, user ratings, download trends, and factors influencing app success. Through data analysis, visualization, and machine learning techniques, the project will provide actionable recommendations for app developers and businesses to enhance their app offerings, improve user satisfaction, and make informed decisions based on data-driven insights.

5.3. MODULE DESCRIPTION

- App Metadata: Information about apps, including app names, categories, sizes, prices, and the number of downloads.
- User Reviews and Ratings: Data on user reviews and corresponding ratings for different apps.
- App Updates History: Historical records of app updates, including release dates and changelogs.
- App Permissions: Details about the permissions requested by each app.
- App Install Statistics: Data on the number of installations for each app.
- App Version Data: Information about different versions of the same app.
- App Pricing Information: Data on app prices, including free and paid apps.

5.3.1 Collections of datasets

In a Google Play Store app analysis mini project, you can use various datasets to gain insights into the app ecosystem. Here are some potential datasets that can be collected and utilized:

- App Metadata
- User Reviews and Ratings
- App Updates History
- App Permissions
- Developer Information
- App Description Text
- App Install Statistics
- App Version Data
- App Pricing Information
- App Release Dates

5.3.2 Selections of Attributes

In a Google Play Store app analysis mini project, the selection of attributes (features) is crucial for gaining meaningful insights. Here are some key attributes to consider:

- App Name
- Category
- Size
- Price
- User Reviews
- App Permissions
- Last Update Date
- App Version
- Developer Information
- Content Rating
- App Description

Category	Rating	Reviews	Size	Installs	Type	Price	Content Rating	Genres	Last Updated	Current Ver	Android Ver
FAMILY	4.5	38	53M	5,000+	Free	0	Everyone	Education	July 25, 2017	1.48	4.1 and up
FAMILY	5.0	4	3.6M	100+	Free	0	Everyone	Education	July 6, 2018	1.0	4.1 and up
MEDICAL	NaN	3	9.5M	1,000+	Free	0	Everyone	Medical	January 20, 2017	1.0	2.2 and up
BOOKS_AND_REFERENCE	4.5	114	Varies with device	1,000+	Free	0	Mature 17+	Books & Reference	January 19, 2015	Varies with device	Varies with device
LIFESTYLE	4.5	398307	19M	10,000,000+	Free	0	Everyone	Lifestyle	July 25, 2018	Varies with device	Varies with device

5.3.3. Preprocessing of data

Preprocessing of data in the Google Play Store app analysis mini project is a crucial step to ensure the data is clean, consistent, and ready for analysis. Here are the key preprocessing steps

- Handling Missing Data
- Removing Duplicates
- Data Conversion and Formatting
- Normalization and Scaling
- Text Data Processing
- Handling Categorical Data
- Feature Selection
- Balancing class distribution
- Splitting Data

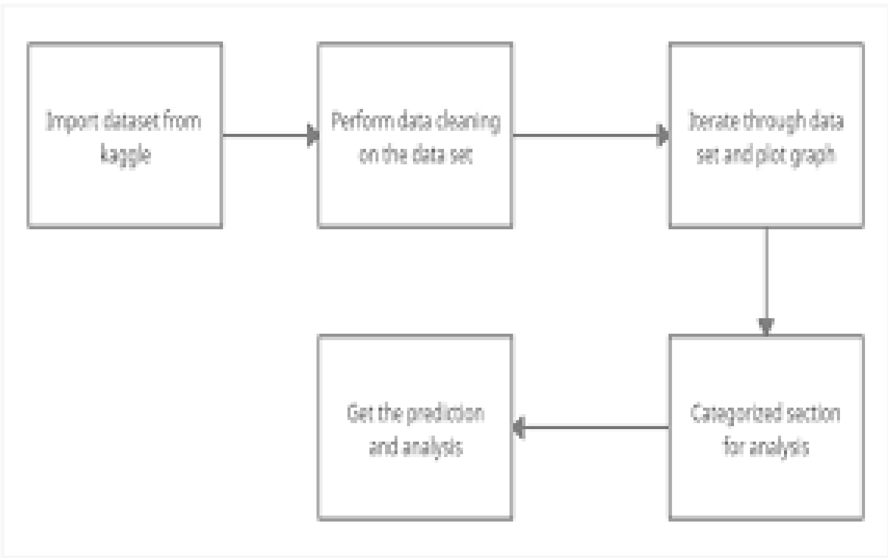
5.3.4. Prediction of Analysis

In the Google Play Store app analysis mini project, the primary prediction or outcome of the analysis would be the ability to forecast or predict app ratings based on various app attributes. Here are the key predictions and analysis outcomes:

- App Ratings Prediction
- Identifying Factors Influencing Ratings
- Popular App Categories
- User Satisfaction Analysis
- App Performance Trends
- Correlation Analysis
- User Segmentation

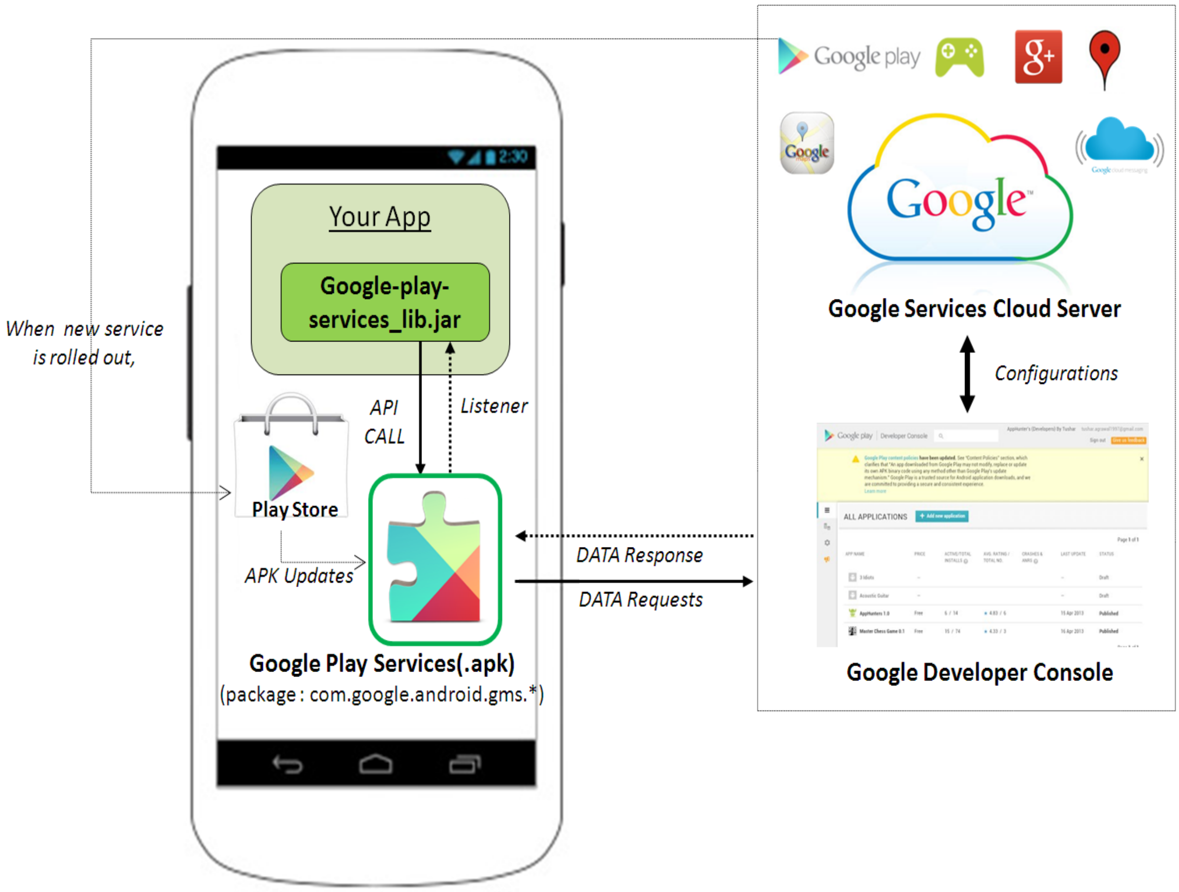
5.4 SYSTEM FLOW DIAGRAM

This diagram shows the interactions between users, developers, and Google in the context of Google Play Store app analysis. Users search for apps on Google Play Store, download and install apps, and use apps. Google provides data about apps to developers, who update apps and provide updates to Google. Google then updates app information for users. Users can also review and rate apps, which Google displays to other users.



5.5. ARCHITECTURE DIAGRAM

This diagram shows the different components involved in Google Play Store app analysis. The Google Play Store is the central component that provides access to app information and analysis. The AppInfoRepository provides access to the raw data about apps, such as the app name, description, category, and ratings. The AppAnalysisService uses this data to generate app analysis results, such as the app's popularity, its rating, and its user reviews. The AppRecommendationService uses these results to generate app recommendations for users.



CHAPTER 6

SYSTEM IMPLEMENTATION

System implementation in the Google Play Store app analysis mini project involves putting the planned analysis into action and executing the various steps to achieve the project's objectives.

6.1 SYSTEM ARCHITECTURE

The system architecture for the Google Play Store app analysis mini project outlines the high-level structure and components of the analysis system. Here's a simplified system architecture for the mini project:

- Data Collection Layer
- Data Preprocessing Layer
- Exploratory Data Analysis (EDA) Layer
- Sentiment Analysis Layer
- Feature Engineering Layer
- Correlation Analysis Layer
- Machine Learning Modeling Layer
- Data Visualization Layer
- Insights and Recommendations Layer

6.2 ALGORITHMS

6.2.1. Exploratory Data Analysis (EDA)

EDA is a statistical method that is used to explore data and identify patterns. EDA is used in the Google Play Store App Analysis project to identify the most popular app categories, the most

EDA	Kategori
Total of Applications per Categories	Games
	Education
	Tools
	Entertainment
Total of Applications per Price Categories	Free
Average Size per Application Categories	Games
	Parenting
Total of Reviews of Application Categories	Tools
	Game Action
Average Rating of Application Categories	Books & References
	Music and Audio
	Personalization

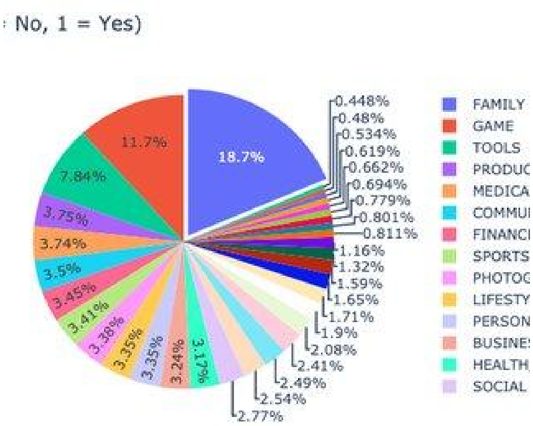
successful apps, and the factors that contribute to the success of apps.

6.2.2. Machine Learning (ML)

ML is a type of artificial intelligence that allows computers to learn from data without being explicitly programmed. ML is used in the Google Play Store App Analysis project to build models that can predict the success of apps.

6.2.3. Visualization

Visualization is the process of transforming data into a visual format that can be easily understood. Visualization is used in the Google Play Store App Analysis project to create charts, graphs, and other visualizations that help to communicate the findings of the project.



CHAPTER 7

TESTING

Software testing is a method of assessing the functionality of a software program. There are many different types of software testing but the two main categories are dynamic testing and static testing. Dynamic testing is an assessment that is conducted while the program is executed; static testing, on the other hand, is an examination of the programs code and associated documentation. Dynamic and static methods are often used together.

7.1. SYSTEM TESTING

System testing in the Google Play Store app analysis mini project is essential to verify that the analysis is accurate, reliable, and meets the project's objectives. Here are key aspects of system testing

7.2. TYPES OF TESTING

Two Testing methods are used to test our project they are

1) Usability Testing
2) A/B Testing

7.2.1 Usability Testing

Usability testing is a qualitative method that is used to identify usability problems in an app. It involves conducting user interviews, usability tests, and surveys to get feedback from users about how they interact with the app. This feedback can be used to identify usability problems, such as confusing or frustrating user interfaces.

7.2.2 A/B Testing

A/B testing is a quantitative method that is used to compare the performance of two versions of an app. It involves splitting users into two groups and showing them different versions of the app. The results of the A/B test can be used to identify which version of the app is more effective, such as which version has a higher click-through rate or conversion rate.

CHAPTER 8

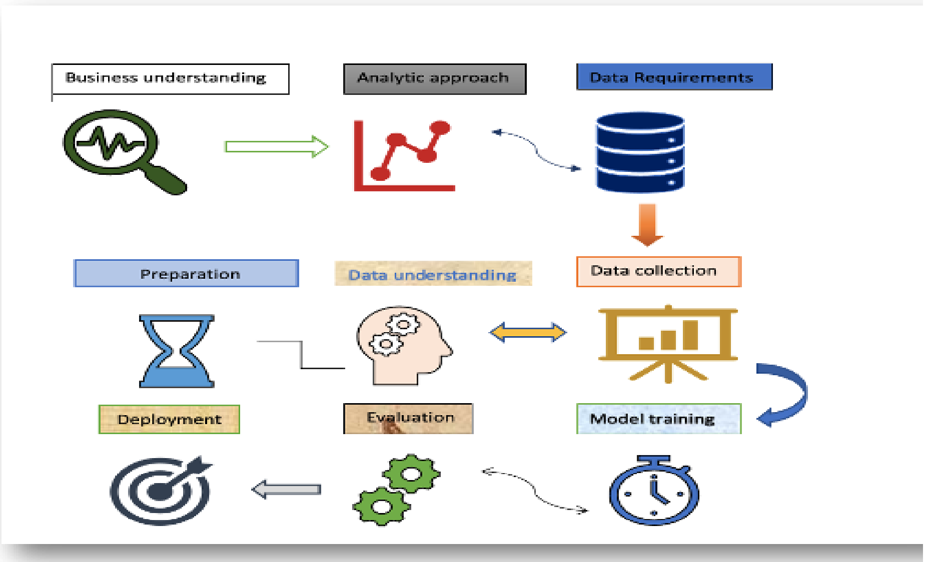
CONCLUSION AND FUTURE ENHANCEMENT

As an AI language model, I don't have access to specific content or data from your mini-project. However, I can suggest a general template for the conclusion and future enhancement appendix for a Google Play Store app analysis mini-project. Remember to tailor it to the specific findings and scope of your project.

Conclusion:

In this mini-project, we conducted an analysis of apps available on the Google Play Store. We explored various aspects, including app categories, ratings, reviews, and download statistics. Based on our analysis, the following conclusions can be drawn:

- ➢ **Popular Categories:** The top app categories with the highest number of downloads and user ratings are [insert categories], indicating a strong demand for these types of applications.
- ➢ **Impact of Ratings:** Apps with higher user ratings tend to have more downloads, demonstrating the importance of positive user experiences in driving app popularity and success.
- ➢ **Monetization Strategies:** We observed that certain app categories are more likely to employ monetization strategies such as in-app purchases and advertisements, while others rely on upfront payment models.
- ➢ **User Reviews:** User reviews play a crucial role in app success, as apps with positive reviews attract more downloads and engagement.
- ➢ **Competitive Landscape:** The Google Play Store hosts a highly competitive environment, with numerous apps vying for user attention. Developers need to focus on unique features and effective marketing strategies to stand out from the crowd.



Future Enhancements

Based on the insights gained from this analysis, there are several potential avenues for future enhancement and research

- ➢ **Sentiment Analysis**: Conduct sentiment analysis on user reviews to gain deeper insights into users' sentiments and identify common pain points and areas for improvement.
- ➢ **Feature Importance:** Perform feature importance analysis to understand which app characteristics (e.g., size, content rating, etc.) have the most significant impact on user engagement and app success.
- ➢ **App Update Impact:** Investigate the impact of app updates on user engagement and ratings to determine whether regular updates positively affect app performance.
- ➢ **User Demographics:** Explore the demographics of users engaging with different app categories to tailor apps more effectively to specific target audiences.
- ➢ Machine Learning Prediction: Utilize machine learning algorithms to predict the success of an app based on various factors, helping developers make data-driven decisions.
- ➢ **Competitor Analysis:** Extend the analysis to include a comprehensive competitor analysis to identify the strengths and weaknesses of competing apps in the same category.
- ➢ **Global vs. Local Markets:** Compare app performance in different countries or regions to identify potential opportunities for expansion.
- ➢ **User Behavior Analysis:** Analyze user behavior within the app, such as the most frequently used features or user churn, to optimize user experiences

CHAPTER 9

APPENDIX

9.1 SOURCE CODE AND SCREENSHOT

Introduction

In [1]:

```
#import library
import pandas as pd # data processing, CSV file I/O (e.g. pd.read_csv)
import numpy as np # linear algebra
import matplotlib.pyplot as plt
import seaborn as sns  # visualization tool
# plotly
import plotly.plotly as py
from plotly.offline import init_notebook_mode, iplot
init_notebook_mode(connected=True)
import plotly.graph_objs as go

# word cloud library
from wordcloud import WordCloud
```

In [2]:

```
#read to csv
data = pd.read_csv("../input/googleplaystore.csv")
```

In [3]:

```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 10841 entries, 0 to 10840
Data columns (total 13 columns):
App               10841 non-null object
Category          10841 non-null object
Rating            9367 non-null float64
Reviews           10841 non-null object
Size              10841 non-null object
Installs          10841 non-null object
Type              10840 non-null object
Price             10841 non-null object
Content Rating    10840 non-null object
Genres            10841 non-null object
Last Updated      10841 non-null object
```

Current Ver 10833 non-null object
Android Ver 10838 non-null object
dtypes: float64(1), object(12)
memory usage: 1.1+ MB

Out[4]:

Index(['App', 'Category', 'Rating', 'Reviews', 'Size', 'Installs', 'Type', 'Price', 'Content Rating', 'Genres', 'Last Updated', 'Current Ver', 'Android Ver'],

dtype='object')

In [5]:

data.shape

Out[5]:

(10841, 13)

In [6]:

data.head()
Out[6]:

[6]:
```
data.head()
```
ıt[6]:

	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

In [7]:
data.tail()
Out[7]:

In [7]:

```
data.tail()
```

Out[7]:

	App	Category	Rating	Reviews	Size	Installs	Type	Price	Content Rating	Genres	Last Updated	Current Ver	Android Ver
10836	Sya9a Maroc - FR	FAMILY	4.5	38	53M	5,000+	Free	0	Everyone	Education	July 25, 2017	1.48	4.1 and up
10837	Fr. Mike Schmitz Audio Teachings	FAMILY	5.0	4	3.6M	100+	Free	0	Everyone	Education	July 6, 2018	1.0	4.1 and up
10838	Parkinson Exercices FR	MEDICAL	NaN	3	9.5M	1,000+	Free	0	Everyone	Medical	January 20, 2017	1.0	2.2 and up
10839	The SCP Foundation DB fr nn5n	BOOKS_AND_REFERENCE	4.5	114	Varies with device	1,000+	Free	0	Mature 17+	Books & Reference	January 19, 2015	Varies with device	Varies with device
10840	iHoroscope - 2018 Daily Horoscope & Astrology	LIFESTYLE	4.5	398307	19M	10,000,000+	Free	0	Everyone	Lifestyle	July 25, 2018	Varies with device	Varies with device

In [8]:

```
data1 = data.head()
data2 = data.tail()
concat_data = pd.concat([data1,data2],axis=0,ignore_index=True)
concat_data
```

Out[8]:Cleaning Data

Out[8]:

	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	Sya9a Maroc - FR	FAMILY	4.5	38	53M	5,000+	Free	0	Everyone	Education	July 25, 2017	1.48	4.1 and up
6	Fr. Mike Schmitz Audio Teachings	FAMILY	5.0	4	3.6M	100+	Free	0	Everyone	Education	July 6, 2018	1.0	4.1 and up
7	Parkinson Exercices FR	MEDICAL	NaN	3	9.5M	1,000+	Free	0	Everyone	Medical	January 20, 2017	1.0	2.2 and up
8	The SCP Foundation DB fr nn5n	BOOKS_AND_REFERENCE	4.5	114	Varies with device	1,000+	Free	0	Mature 17+	Books & Reference	January 19, 2015	Varies with device	Varies with device
9	iHoroscope - 2018 Daily Horoscope & Astrology	LIFESTYLE	4.5	398307	19M	10,000,000+	Free	0	Everyone	Lifestyle	July 25, 2018	Varies with device	Varies with device

In [9]:

```
data['Category'].unique()
```

Out[9]:

```
array(['ART_AND_DESIGN', 'AUTO_AND_VEHICLES', 'BEAUTY',
       'BOOKS_AND_REFERENCE', 'BUSINESS', 'COMICS', 'COMMUNICATION',
       'DATING', 'EDUCATION', 'ENTERTAINMENT', 'EVENTS', 'FINANCE',
       'FOOD_AND_DRINK', 'HEALTH_AND_FITNESS', 'HOUSE_AND_HOME',
       'LIBRARIES_AND_DEMO', 'LIFESTYLE', 'GAME', 'FAMILY', 'MEDICAL',
       'SOCIAL', 'SHOPPING', 'PHOTOGRAPHY', 'SPORTS', 'TRAVEL_AND_LOCAL',
       'TOOLS', 'PERSONALIZATION', 'PRODUCTIVITY', 'PARENTING', 'WEATHER',
       'VIDEO_PLAYERS', 'NEWS_AND_MAGAZINES', 'MAPS_AND_NAVIGATION'])
```

In [10]:

```
data[data['Category'] == '1.9']
```

Out[10]:

In [10]:

```
data[data['Category'] == '1.9']
```

Out[10]:

	App	Category	Rating	Reviews	Size	Installs	Type	Price	Content Rating	Genres	Last Updated	Current Ver	Android Ver
10472	Life Made WI-Fi Touchscreen Photo Frame	1.9	19.0	3.0M	1,000+	Free	0	Everyone	NaN	February 11, 2018	1.0.19	4.0 and up	NaN

In [11]:

```
data.loc[10472] = data.loc[10472].shift()
data['App'].loc[10472] = data['Category'].loc[10472]
data['Category'].loc[10472] = np.nan
data.loc[10472]
```

Out[11]:

```
App                Life Made WI-Fi Touchscreen Photo Frame
Category                                               NaN
Rating                                                 1.9
Reviews                                                 19
Size                                                  3.0M
Installs                                            1,000+
Type                                                  Free
Price                                                    0
Content Rating                                    Everyone
Genres                                                 NaN
Last Updated                             February 11, 2018
Current Ver                                         1.0.19
```

In [12]:

```
data['Rating'].unique()
```

Out[12]:

```
array([4.1, 3.9, 4.7, 4.5, 4.3, 4.4, 3.8, 4.2, 4.6, 3.2, 4.0, nan, 4.8,
       4.9, 3.6, 3.7, 3.3, 3.4, 3.5, 3.1, 5.0, 2.6, 3.0, 1.9, 2.5, 2.8,
       2.7, 1.0, 2.9, 2.3, 2.2, 1.7, 2.0, 1.8, 2.4, 1.6, 2.1, 1.4, 1.5,

       1.2, '1.9'], dtype=object)
```

In [13]:

```
data['Rating'] = pd.to_numeric(data['Rating'], errors='coerce')
data['Rating'].dtype
```

Out[13]:

```
dtype('float64')
```

Reviews

In [14]:

```
data['Reviews'].unique()
```

Out[14]:

```
array(['159', '967', '87510', ..., '603', '1195', '398307'], dtype=object)
```

In [15]:

```
data[data['Reviews'] =='3.0M']
```

Out[15]:

In [15]:

```
data[data['Reviews'] =='3.0M']
```

Out[15]:

	App	Category	Rating	Reviews	Size	Installs	Type	Price	Content Rating	Genres	Last Updated	Current Ver	Android Ver

In [16]:

```
data['Reviews'] = data.Reviews.replace("0.0",0)
data['Reviews'] = data.Reviews.replace("3.0M",3000000.0)
```

```
data['Reviews'] = data['Reviews'].astype(float)
data['Reviews'].dtype
```

Out[16]:

```
dtype('float64')
```

Size

In [17]:

```
data['Size'].unique()
```

Out[17]:

```
array(['19M', '14M', '8.7M', '25M', '2.8M', '5.6M', '29M', '33M', '3.1M',
       '28M', '12M', '20M', '21M', '37M', '2.7M', '5.5M', '17M', '39M',
       '31M', '4.2M', '7.0M', '23M', '6.0M', '6.1M', '4.6M', '9.2M',
       '5.2M', '11M', '24M', 'Varies with device', '9.4M', '15M', '10M',
       '1.2M', '26M', '8.0M', '7.9M', '56M', '57M', '35M', '54M', '201k',
       '3.6M', '5.7M', '8.6M', '2.4M', '27M', '2.5M', '16M', '3.4M',
       '8.9M', '3.9M', '2.9M', '38M', '32M', '5.4M', '18M', '1.1M',
       '2.2M', '4.5M', '9.8M', '52M', '9.0M', '6.7M', '30M', '2.6M',
       '7.1M', '3.7M', '22M', '7.4M', '6.4M', '3.2M', '8.2M', '9.9M',
       '4.9M', '9.5M', '5.0M', '5.9M', '13M', '73M', '6.8M', '3.5M',
       '4.0M', '2.3M', '7.2M', '2.1M', '42M', '7.3M', '9.1M', '55M',
       '23k', '6.5M', '1.5M', '7.5M', '51M', '41M', '48M', '8.5M', '46M',
       '8.3M', '4.3M', '4.7M', '3.3M', '40M', '7.8M', '8.8M', '6.6M',
       '5.1M', '61M', '66M', '79k', '8.4M', '118k', '44M', '695k', '1.6M',
       '6.2M', '18k', '53M', '1.4M', '3.0M', '5.8M', '3.8M', '9.6M',
       '45M', '63M', '49M', '77M', '4.4M', '4.8M', '70M', '6.9M', '9.3M',
       '10.0M', '8.1M', '36M', '84M', '97M', '2.0M', '1.9M', '1.8M',
       '5.3M', '47M', '556k', '526k', '76M', '7.6M', '59M', '9.7M', '78M',
       '72M', '43M', '7.7M', '6.3M', '334k', '34M', '93M', '65M', '79M',
       '100M', '58M', '50M', '68M', '64M', '67M', '60M', '94M', '232k',
       '99M', '624k', '95M', '8.5k', '41k', '292k', '11k', '80M', '1.7M',
       '74M', '62M', '69M', '75M', '98M', '85M', '82M', '96M', '87M',
       '71M', '86M', '91M', '81M', '92M', '83M', '88M', '704k', '862k',
       '899k', '378k', '266k', '375k', '1.3M', '975k', '980k', '4.1M',
       '89M', '696k', '544k', '525k', '920k', '779k', '853k', '720k',
       '713k', '772k', '318k', '58k', '241k', '196k', '857k', '51k',
       '953k', '865k', '251k', '930k', '540k', '313k', '746k', '203k',
       '26k', '314k', '239k', '371k', '220k', '730k', '756k', '91k',
       '293k', '17k', '74k', '14k', '317k', '78k', '924k', '902k', '818k',
       '81k', '939k', '169k', '45k', '475k', '965k', '90M', '545k', '61k',
       '283k', '655k', '714k', '93k', '872k', '121k', '322k', '1.0M',
       '976k', '172k', '238k', '549k', '206k', '954k', '444k', '717k',
```

```
'210k', '609k', '308k', '705k', '306k', '904k', '473k', '175k',
'350k', '383k', '454k', '421k', '70k', '812k', '442k', '842k',
'417k', '412k', '459k', '478k', '335k', '782k', '721k', '430k',
'429k', '192k', '200k', '460k', '728k', '496k', '816k', '414k',
'506k', '887k', '613k', '243k', '569k', '778k', '683k', '592k',
'319k', '186k', '840k', '647k', '191k', '373k', '437k', '598k',
'716k', '585k', '982k', '222k', '219k', '55k', '948k', '323k',
'691k', '511k', '951k', '963k', '25k', '554k', '351k', '27k',
'82k', '208k', '913k', '514k', '551k', '29k', '103k', '898k',
'743k', '116k', '153k', '209k', '353k', '499k', '173k', '597k',
'809k', '122k', '411k', '400k', '801k', '787k', '237k', '50k',
'643k', '986k', '97k', '516k', '837k', '780k', '961k', '269k',
'20k', '498k', '600k', '749k', '642k', '881k', '72k', '656k',
'601k', '221k', '228k', '108k', '940k', '176k', '33k', '663k',
'34k', '942k', '259k', '164k', '458k', '245k', '629k', '28k',
'288k', '775k', '785k', '636k', '916k', '994k', '309k', '485k',
'914k', '903k', '608k', '500k', '54k', '562k', '847k', '957k',
'688k', '811k', '270k', '48k', '329k', '523k', '921k', '874k',
'981k', '784k', '280k', '24k', '518k', '754k', '892k', '154k',
'860k', '364k', '387k', '626k', '161k', '879k', '39k', '970k',
'170k', '141k', '160k', '144k', '143k', '190k', '376k', '193k',
'246k', '73k', '658k', '992k', '253k', '420k', '404k', '470k',
'226k', '240k', '89k', '234k', '257k', '861k', '467k', '157k',
'44k', '676k', '67k', '552k', '885k', '1020k', '582k', '619k'],

dtype=object)
```

In [18]:

```python
data['Size'] = data.Size.replace("Varies with device",np.nan)
data['Size'] = data.Size.str.replace("M","000") # All size values became the kilobyte type.
data['Size'] = data.Size.str.replace("k","")
data['Size'] = data.Size.replace("1,000+",1000)
data['Size'] =data['Size'].astype(float)
data['Size'].dtype
```

Out[18]:

```
dtype('float64')
```

Installs

In [19]:

```python
data['Installs'].unique()
```

Out[19]:

```
array(['10,000+', '500,000+', '5,000,000+', '50,000,000+', '100,000+',
       '50,000+', '1,000,000+', '10,000,000+', '5,000+', '100,000,000+',
       '1,000,000,000+', '1,000+', '500,000,000+', '50+', '100+', '500+',

       '10+', '1+', '5+', '0+', '0'], dtype=object)
```

In [20]:

```
data['Installs'] = data.Installs.str.replace(",","")
data['Installs'] = data.Installs.str.replace("+","")
data['Installs'] = data.Installs.replace("Free",np.nan)
data['Installs'] = data['Installs'].astype(float)
data['Installs'].dtype
```

Out[20]:

```
dtype('float64')
```

Price

In [21]:

```
data['Price'].unique()
```

Out[21]:

```
array(['0', '$4.99', '$3.99', '$6.99', '$1.49', '$2.99', '$7.99', '$5.99',
       '$3.49', '$1.99', '$9.99', '$7.49', '$0.99', '$9.00', '$5.49',
       '$10.00', '$24.99', '$11.99', '$79.99', '$16.99', '$14.99',
       '$1.00', '$29.99', '$12.99', '$2.49', '$10.99', '$1.50', '$19.99',
       '$15.99', '$33.99', '$74.99', '$39.99', '$3.95', '$4.49', '$1.70',
       '$8.99', '$2.00', '$3.88', '$25.99', '$399.99', '$17.99',
       '$400.00', '$3.02', '$1.76', '$4.84', '$4.77', '$1.61', '$2.50',
       '$1.59', '$6.49', '$1.29', '$5.00', '$13.99', '$299.99', '$379.99',
       '$37.99', '$18.99', '$389.99', '$19.90', '$8.49', '$1.75',
       '$14.00', '$4.85', '$46.99', '$109.99', '$154.99', '$3.08',
       '$2.59', '$4.80', '$1.96', '$19.40', '$3.90', '$4.59', '$15.46',
       '$3.04', '$4.29', '$2.60', '$3.28', '$4.60', '$28.99', '$2.95',
       '$2.90', '$1.97', '$200.00', '$89.99', '$2.56', '$30.99', '$3.61'
        '$394.99', '$1.26', '$1.20', '$1.04'], dtype=object)
```

In [22]:

```
data['Price'] = data.Price.replace("Everyone",np.nan)
data['Price'] = data.Price.str.replace("$","").astype(float)
data['Price'].dtype
```

Out[22]:

dtype('float64')

Last Updated

In [23]:

```
data['Last Updated'].unique()
```

Out[23]:

```
array(['January 7, 2018', 'January 15, 2018', 'August 1, 2018', ...,
       'January 20, 2014', 'February 16, 2014', 'March 23, 2014'],

      dtype=object)
```

In [24]:

```
data['Last Updated'] = pd.to_datetime(data['Last Updated'])
data['Last Updated']
```

Out[24]:

```
0       2018-01-07
1       2018-01-15
2       2018-08-01
3       2018-06-08
4       2018-06-20
5       2017-03-26
6       2018-04-26
7       2018-06-14
8       2017-09-20
9       2018-07-03
10      2017-10-27
11      2018-07-31
12      2018-04-02
13      2018-06-26
14      2018-08-03
15      2018-06-06
16      2018-07-31
17      2017-11-07
18      2018-08-03
19      2018-07-30
20      2018-04-20
21      2018-03-20
22      2018-07-12
23      2018-03-07
```

24 2018-07-07
25 2018-04-25
26 2017-10-11
27 2018-03-21
28 2018-07-12
29 2017-08-22
...
10811 2018-07-24
10812 2018-06-13
10813 2018-07-17
10814 2015-06-03
10815 2018-07-19
10816 2018-08-06
10817 2015-10-30
10818 2016-05-19
10819 2017-08-05
10820 2018-06-27
10821 2018-05-29
10822 2017-12-01
10823 2018-04-22
10824 2018-07-31
10825 2018-02-02
10826 2018-03-23
10827 2018-06-13
10828 2017-05-15
10829 2016-06-19
10830 2014-01-20
10831 2018-06-13
10832 2014-02-16
10833 2014-03-23
10834 2017-06-18
10835 2016-09-29
10836 2017-07-25
10837 2018-07-06
10838 2017-01-20
10839 2015-01-19
10840 2018-07-25

Exploratory Data Analysis

After, I prepared to analyze our data, somewhat let's explore the datas. :)

- **corr() : It returns correlation.**
- **describe (): It returns number of entries, average of entries, outlier values, standart deviation, minimum and maximum entry.**

In [25]:

data.corr()

Out[25]:

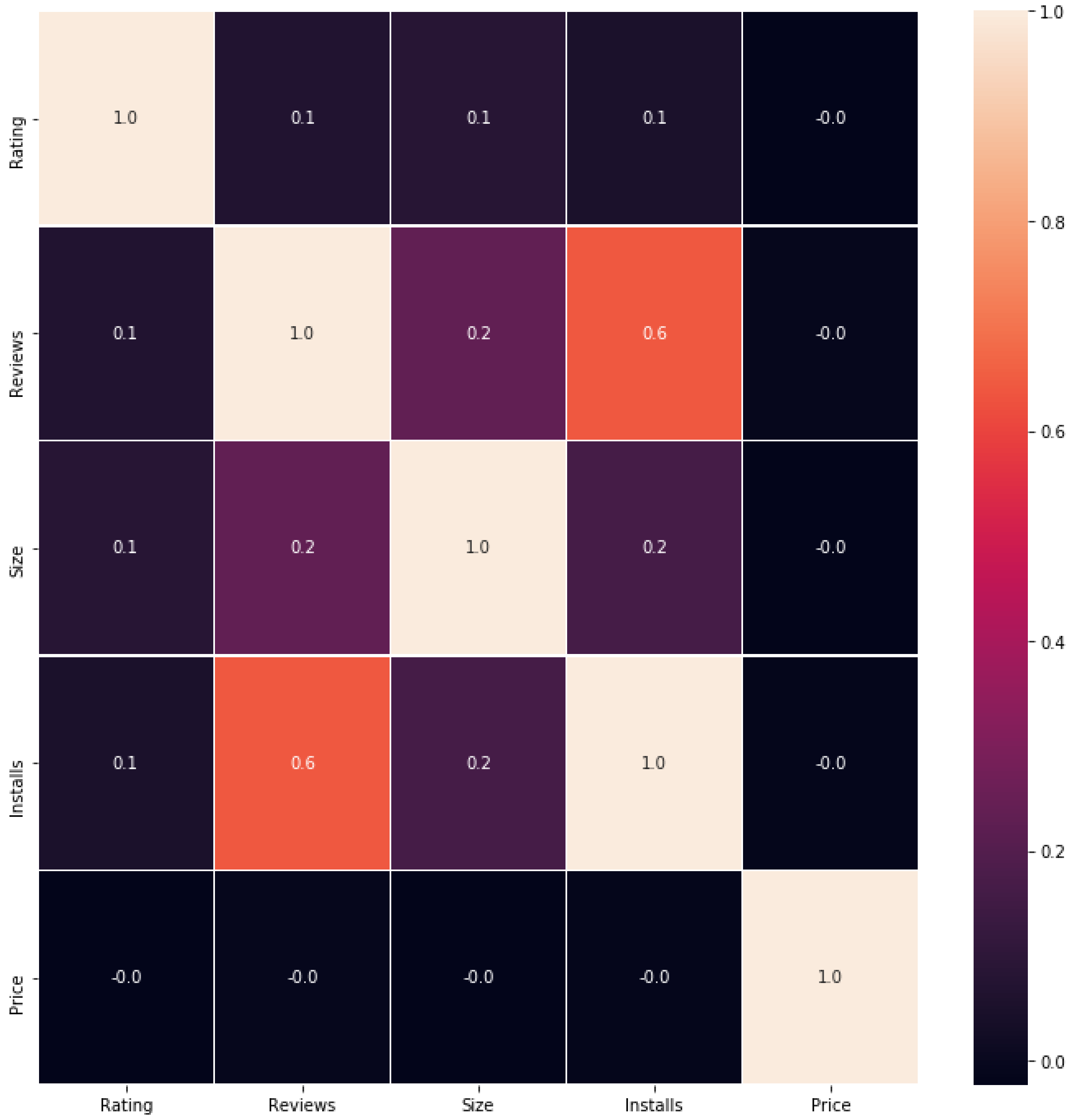
```
In [25]:  data.corr()
```

Out[25]:

	Rating	Reviews	Size	Installs	Price
Rating	1.000000	0.068147	0.082495	0.051393	-0.021851
Reviews	0.068147	1.000000	0.233795	0.643123	-0.009666
Size	0.082495	0.233795	1.000000	0.164616	-0.022766
Installs	0.051393	0.643123	0.164616	1.000000	-0.011688
Price	-0.021851	-0.009666	-0.022766	-0.011688	1.000000

In [26]:

```
#correlation map
fax = plt.subplots(figsize=(12, 12))
sns.heatmap(data.corr(), annot=True, linewidths=.5, fmt= '.1f',ax=ax)
plt.show()
```



In [27]:

data.describe()

Out[27]:

In [27]:

```
data.describe()
```

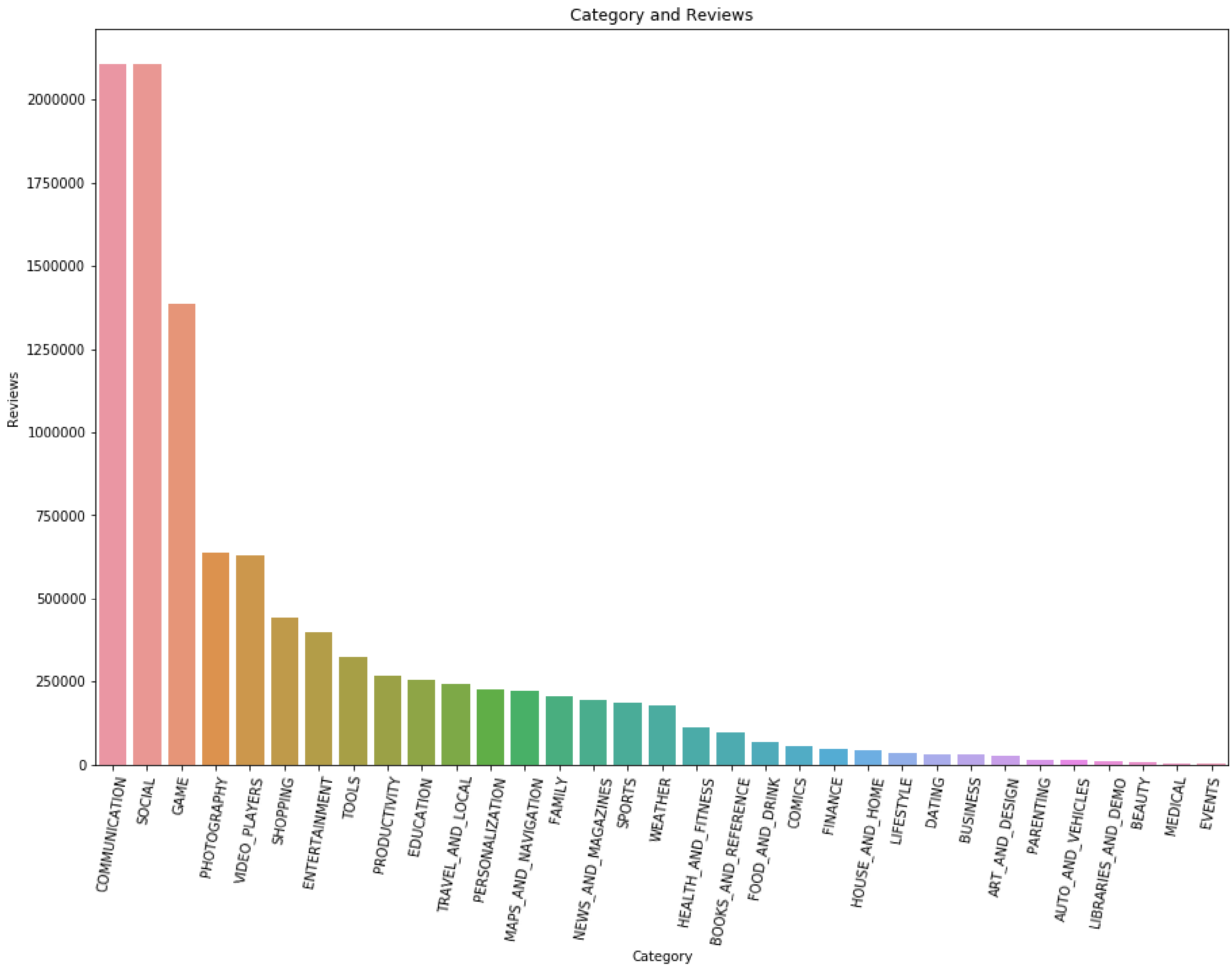
Out[27]:

	Rating	Reviews	Size	Installs	Price
count	9367.000000	1.084100e+04	9146.000000	1.084100e+04	10841.000000
mean	4.191513	4.441119e+05	19577.279477	1.546291e+07	1.027273
std	0.515735	2.927629e+06	24041.618956	8.502557e+07	15.948971
min	1.000000	0.000000e+00	1.000000	0.000000e+00	0.000000
25%	4.000000	3.800000e+01	5.600000	1.000000e+03	0.000000
50%	4.300000	2.094000e+03	13000.000000	1.000000e+05	0.000000
75%	4.500000	5.476800e+04	30000.000000	5.000000e+06	0.000000
max	5.000000	7.815831e+07	100000.000000	1.000000e+09	400.000000

Category and Reviews

In [28]:

```
category_list = list(data['Category'].unique())
category_review = []
for i in category_list:
    x = data[data['Category'] == i]
    if(len(x)!=0):
        review = sum(x.Reviews)/len(x)
        category_review.append(review)
    else:
        review = sum(x.Reviews)
        category_review.append(review)
#sorting
data_category_reviews = pd.DataFrame({'category': category_list,'review':category_review})
new_index = (data_category_reviews['review'].sort_values(ascending=False)).index.values
sorted_data =data_category_reviews.reindex(new_index)

# visualization
plt.figure(figsize=(15,10))
sns.barplot(x=sorted_data['category'], y=sorted_data['review'])
plt.xticks(rotation=80)
plt.xlabel("Category")
plt.ylabel("Reviews")
plt.title("Category and Reviews")
plt.show()
```

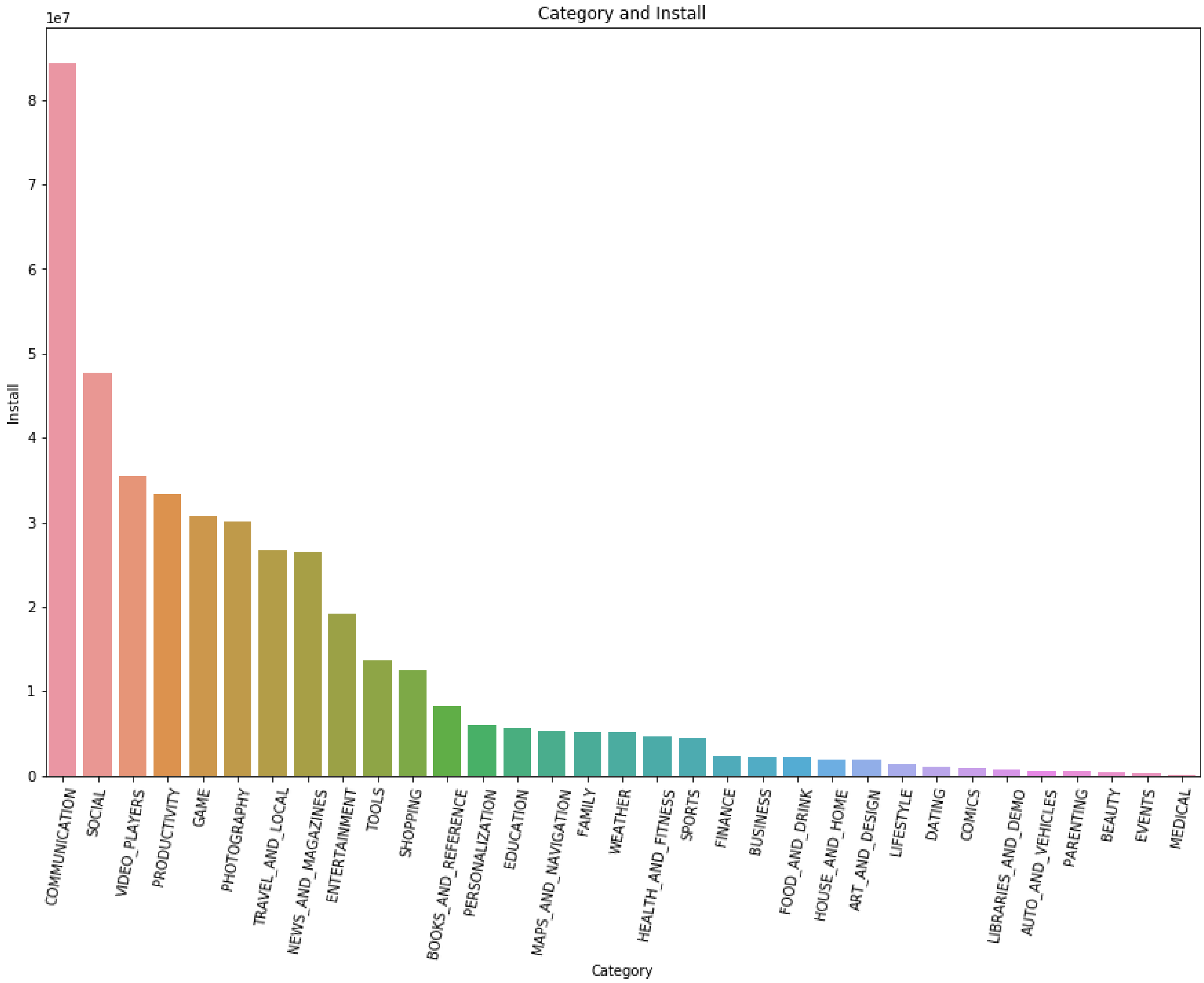


Category and Installs

In [29]:

```
category_list = list(data['Category'].unique())
category_install = []
for i in category_list:
    x = data[data['Category'] == i]
    if(len(x)!=0):
        install = sum(x.Installs)/len(x)
        category_install.append(install)
    else:
        install = sum(x.Installs)
        category_install.append(install)

#sorting
data_category_install = pd.DataFrame({'category': category_list,'install':category_install})
```

```
new_index = (data_category_install['install'].sort_values(ascending=False)).index.values
sorted_data =data_category_install.reindex(new_index)

# visualization
plt.figure(figsize=(15,10))
sns.barplot(x=sorted_data['category'], y=sorted_data['install'])
plt.xticks(rotation=80)
plt.xlabel("Category")
plt.ylabel("Install")
plt.title("Category and Install")
plt.show()
```



Word Cloud

In [30]:

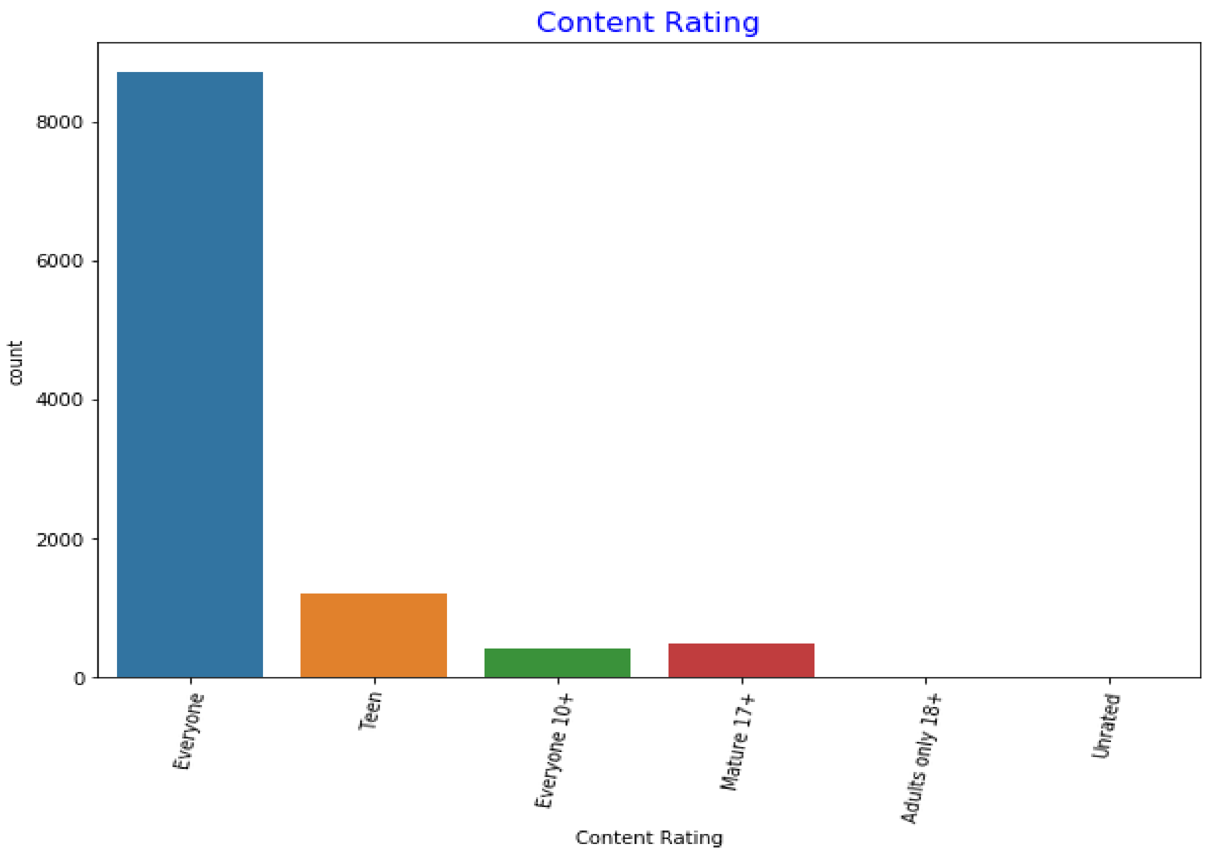
```
plt.subplots(figsize=(8,8))
wordcloud = WordCloud(
                    background_color='white',
                    width=512,
                    height=384
                   ).generate(" ".join(data))
plt.imshow(wordcloud)
plt.axis('off')
plt.savefig('graph.png')

plt.show()
```

Content Rating

In [31]:

```
plt.figure(figsize=(10,7))
sns.countplot(data=data, x='Content Rating')
plt.xticks(rotation=80)
```

```
plt.title('Content Rating',color = 'blue',fontsize=15)
plt.show()
```



CHAPTER 10

REFERENCES

[1] Google Play Developer Console: https://developer.android.com/distribute/console/

[2] **Statista**: https://www.statista.com/

- Provides statistics and data on app downloads, revenue, and user demographics.

[3] **App Annie:** https://www.appannie.com/

- Offers app market data and insights to understand app performance, rankings, and market trends.

[4] **Sensor Tower:** https://sensortower.com/

- Provides app store optimization (ASO) and market intelligence tools for app developers.

[5] **Android Developers Blog:** https://android-developers.googleblog.com/

- Official blog for Android developers, providing updates, best practices, and insights into app development.

[6] **Research Papers and Academic Journals:**

35 Look for academic papers and research articles related to mobile app analysis, user behaviour, and app performance on platforms like Google Scholar.

[7] **Google Play Developer Console Documentation:**
The official documentation provided by Google for app developers includes valuable insights into app performance, user engagement, and various metrics available through the console.

[8] **App Annie:**
App Annie offers market data and analytics on app performance, rankings, and trends. They provide valuable insights into app downloads, revenue, and user demographics.

[9] **Sensor Tower:**
Sensor Tower provides app store optimization (ASO) tools and market intelligence for app developers. Their data can be useful for understanding app rankings, keyword performance, and competitor analysis.

[10] **Statista:**
Statista is a platform that provides statistics and data on various industries, including app downloads, usage, and revenue. It can be a good source for market trends and user behaviour insights.

[11] **Academic Research Papers:**
Look for academic papers and research articles on mobile app analysis, user behaviour, and app performance. Websites like Google Scholar can be helpful for finding relevant research papers.

[12] **Android Developers Blog:** The official blog for Android developers offers updates, best practices, and insights into app development, which can be relevant for your mini-project.