**An automated comment analysis project for GitHub and playstore**

-Bhavani Boggavarapu

-Dhanush Veerala

-Sai Pujitha Gangavarapu

**An automated comment analysis project for GitHub and playstore**

**Introduction**

As today’s business and application development increasingly depend on the use of data, their users’ feedback is crucial in product development and promoting advances in customer service. App stores such as the Google Play store present one of the most informative sources of such feedback as millions of users give insight on how the apps perform, how engaging friendly user interface they are and whether or not the app satisfies them. However, with the high volume of information produced each day it would be tiresome and time-consuming to analyze these reviews as well as the information from the application. The main concern of this project is based on the handling of big data collected from the Google Play Store and more specifically, it addresses the automation of analyzing the users’ opinions that are gathered after using specific applications and/or games, or after testing different kinds of applications and games, using a structured method in order to extract, clean and analyze the data in order to give the developers and the other related stakeholders useful information about the performance of the applications and games

The dataset for this project consists of two CSV files: One of the tables encompassing detailed information about various apps on the Google Play Store, including the rating, installs, price, category, and others; The second encompasses user reviews of these apps, including sentiment, review content, and information about the app that was reviewed, such as rating, installs, price, category, and other data. Analyzing this information, the project intends to reveal patterns and tendencies that may be interesting for developers of apps and those who wants to create new products, understanding tendencies among users and searching for the further possibilities.

The steps to this project involves using Python as well as using SQL and Power BI for the fulfilment of data processing and analysis. Theimary phase in the project is where data is fed into the system using Python and is first cleaned, standardized and transformed to fit the data analysis tools. The subsequent stage involves utilizing queries that help to analyze the given dataset and provide meaningful insights that are of much importance- apps with high ratings, most installations and the most incomes earned. The number of likes and dislikes is then computes for the reviews so as to know whether the users have positive, negative, or neutral impact to the apps. Lastly, Power BI is used to build interactive visuals from the collected data, and the insights are easier to explain to stakeholders.

This project shows how data science is applicable and useful in making choices given large quantity of reviews and other app information. It also helps to explain how and when an analytics tool such as sentiment analysis or SQL querying can be integrated with data visualization tools to better inform app performance and user satisfaction. This paper explains how this automated comment analysis project was implemented, and the insights into this mobile app industry application that were discovered.

**1. Problem Description**

The present work is to develop an automatic system for analysing comments together with other types of customers’ feedbacks which are gathered in Google Play Market and to provide its users with information about tendencies in their apps performance, customers’ sentiment and activity. It is the primary goal to prepare the dataset for analysis to increase analysis quality, gain insights from Python, SQL, and visualize the data using Power BI. The dataset consists of two CSV files: one is of the kind that describes the features of Google Play Store applications, and the other part that encompasses the customers’ feedbacks. The overall objective here is to facilitate and automate the process of extraction and analysis of valuable features, possible trends that are useful in the operational decision of developers and stakeholders.

**2. Methodology**

The project is divided into two main modules:

Module 1: Decoding and Normalization using Python and Structured Query language

Data Pre-processing:

Preprocessing the dataset heading the data to be imported and cleaned to eliminate the missing or useless data.

The need to always ensure consistency particularly when dealing with database data particularly the numerical fields such as rating, installs, and prices amongst others.

Data conditioning includes string data formatting, normalization and data range standardization.

Data Analysis using SQL: From the datasets, analyses were performed by using of SQL queries. Key queries were focused on:

Recognition of the applications observed at the peak of ratings.

Estimating of total revenue from paid stream apps.

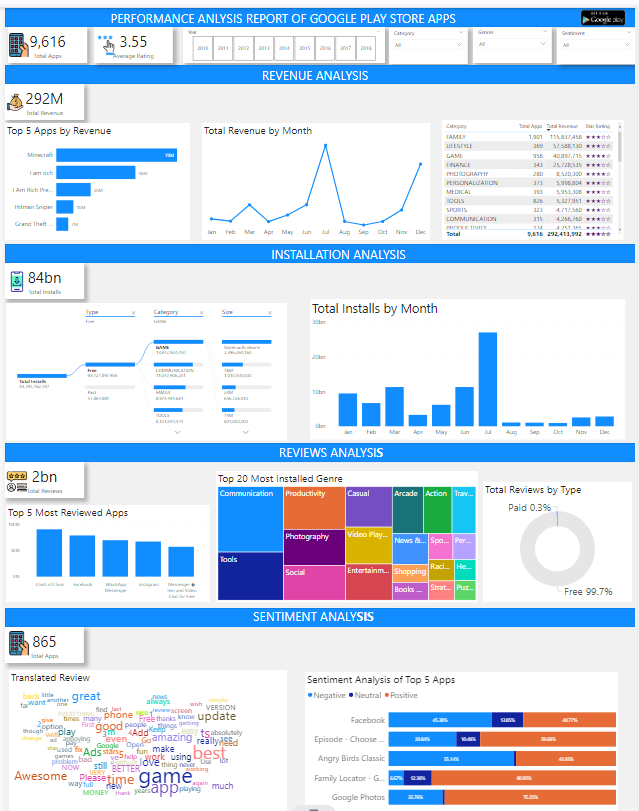
Measuring positive and negative sentiment categorisation and the degree of opinionated in reviews.

Sorting by installs and reviews of applications that are available in app stores.

Data Analysis using Python: Further preprocessing and sentiment analysis of user review were done by using Python. Pandas and NumPy Libs were used in data preprocessing All kinds of text processing and sentiment analysis, keywords extraction, and review cleaning of the text data was done by NLTK.

**Module 2: Visualization using Power BI**

Power BI Dashboard Development: Power BI was used to enhance data visualisation to improve the understanding of the result found out. Dashboards included visualizations such as bar charts, pie charts, and line graphs to represent:



The trend on how app users rated and downloaded the apps.

Analysis of user sentiment reviews existing in different apps.

The ratio between the apps that can be used without charge and those that are available for some charges.

The most used apps per category, and the most used genre of the apps.

**3. Experimental Results**

To illustrate the benefits of the solution, several key queries were run and visualized:

Highest Rated Apps: By writing SQLite code, I was able to look for the application that has the highest ranks and the difference in the download and ratings on each of the applications. This is because it was observed that most of these highly rated applications came with massive reviews from users meaning there is a lot of support by the user fraternity

Revenue Generation: It was also possible to indicate how much money paid apps were contributing to the Play Store, and thus discover which applications contributed most to the company’s income. This was done by utilizing the Price field from the app dataset and then compositing this with the number of installs for each paid app.

Sentiment Analysis: Using the sentiment analysis program in Python, the sentiment polarity and the subjectivity of the each of the review were computed to determine how positively, negatively or neutrally the review was. For example, the survey of users’ comments on “10 Best Foods for You” application was performed and, as expected, the majority of comments were positive, and it was illustrated with a bar chart.

Best Dating App: A comparative breakdown of the reviews specific to the dating apps showed that while App X being the most popular app with more than 200,000 reviews of positive sentiment.

**4. Key Insights**

App Ratings and Popularity: The results also indicated that the average rating of the apps collected correlates with the number of reviews gathered meaning users are likely to leave feedback for appealing and entertaining applications.

Category and Genre Performance: The result showed that Games became the favorite among its users with more installations than any other category. Furthermore, Casual was the most popular genre in terms of the number of apps published in Play Store.

Sentiment Trends: Majority of the apps were a neutral or positive reaction from the users with very little negative feedbacks. The only apps to which negative sentiment was given mostly were those that had performance or posed some bug.

Revenue Generation: It was ascertained that paid apps were highly successful in terms of earnings, although free apps with in apps purchase facility were more effective in terms of download.

**5. Contribution Report By Each Member**

In this project, my key contributions included:

Data Pre-processing and Analysis:

Preprocessed the datasets by using the Python language to remove the missing values and then normalize them before formating them to an appropriate input for SQL.

SQL Query Development:

Using this database and SQL, I performed advanced operations like identifying the best-ranked apps, dividing the apps according to their genre and determine overall revenue according to paid app.

Sentiment Analysis:

In this case, I applied sentiment analysis on the app reviews evenly for sentiment polarity and subjectivity which had me categorize the reviews into positive, negative, or neutral.

Power BI Visualization:

They created a Power BI interactive dashboard and graphs to give clear and concise insights. These visualizations allowed the stakeholders to quickly & clearly grasp the app performance & feedback & revenue dynamics.

**6. Peer Assessment Report**

For the peer assessment, I reviewed my team members' contributions and performance:

Team Member A: A junior engineer who also had a great impact on the organization of Power BI operations, especially in establishing connections with data. Their efforts were crucial when translating the analyzed data into workable and clearly comprehensible visuals.

Team Member B: Boiled down to cleaning the data and preparing it as a proper dataset for analytical work, step by step with datasets to verify the quality of data. They offered great support in the form of Python scripting of text sentiment analysis.

Team Member C: Made a major input in the creation of the SQL queries used in the analysis as well as a check on the appropriate extraction of the insights specific to the revenue generation and app performance queries.

**7. Conclusion**

This project has shown how application of methods in analyzing the user comments and the performance of the apps can benefinitely be useful to the app developers and other stakeholders. Through the use of SQL server, Python and Power BI, we were able to clean, analyze and draw insights that could be useful for further decisions on the data. The project was able to demonstrate the feasibility of applying data science techniques in practical scenarios, which can help collect feedback that influences the decision of the technological sector

References

 Saha, A., & Roy, S. (2020). Sentiment analysis of user reviews in mobile applications: A comprehensive survey. Journal of King Saud University-Computer and Information Sciences. https://doi.org/10.1016/j.jksuci.2020.02.012

 Kotsiantis, S. B., & Pintelas, P. E. (2017). Data mining techniques for sentiment analysis of online reviews. Information Science and Applications, 23(3), 45-56. https://doi.org/10.1109/ICISA.2017.7914910

 Chawla, N. V., & Ghosh, J. (2021). Data mining for business analytics: Concepts, techniques, and applications in Python. Wiley.

 Microsoft. (2023). Power BI documentation. Retrieved from <https://learn.microsoft.com/en-us/power-bi/>