Final Report: Google Play Store Data Analytics

1. Introduction

The objective of this project is to analyze the Google Play Store dataset and derive meaningful insights using Python. This analysis includes data cleaning, feature engineering, and visualization, ensuring adherence to specific time-based display restrictions for graphs.

2. Background

Google Play Store hosts millions of applications across various categories. The dataset includes information such as app ratings, installs, reviews, and prices. By analyzing this dataset, we aim to understand trends, user engagement, and other key business insights.

3. Learning Objectives

- Data Preprocessing & Cleaning
- Data Visualization with Python
- Implementing Time-Based Graph Restrictions

4. Activities and Tasks

Step 1: Data Loading and Exploration

- Loaded the dataset using Pandas.
- Checked for missing values and inconsistencies.
- Explored data types and performed initial analysis.

Step 2: Data Cleaning

- Handled missing values by removing or imputing them.
- Converted non-numeric columns (e.g., 3.0M, $4.99) into numerical format.
- Removed duplicate entries to maintain dataset integrity.
- Ensured categorical consistency across data fields.

Step 3: Data Visualization

- Implemented various visualizations:
    - **Top-rated categories**
    - **Distribution of app installs**
    - **Price vs. Ratings correlation**
    - **Number of apps per category**
- Enforced time-based visibility (3 PM - 5 PM IST restriction) for graphs.

Step 4: Time-Based Graph Restrictions

- Ensured all visualizations follow the assignment's specific display time constraints.
- Used Python logic to restrict the visibility of charts outside allowed time frames.

5. Skills and Competencies Developed

- **Python for Data Analysis**: Pandas, NumPy, and data wrangling.
- **Data Cleaning & Preprocessing**: Handling missing and inconsistent data.
- **Data Visualization**: Matplotlib, Seaborn for effective graphical representation.
- **Conditional Graph Display**: Implementing time-based restrictions in dashboards.

6. Challenges and Solutions

Challenge: Handling non-standard numerical formats (3.0M, $4.99)

Solution: Used string manipulation techniques to convert text-based numbers into integers and floats.

Challenge: Implementing time-based visibility constraints on graphs

Solution: Applied Python's datetime module to control when graphs should be displayed.

Challenge: Missing and inconsistent data

Solution: Used data imputation and filtering techniques to clean the dataset.

7. Outcomes and Impact

- Successfully cleaned and analyzed a real-world dataset.
- Gained hands-on experience in data visualization and feature engineering.
- Implemented real-time dashboard constraints based on business requirements.
- Prepared insights that can be useful for app developers and marketers.

8. Conclusion

This project helped in understanding the end-to-end data analytics workflow, from preprocessing to visualization. The key takeaway is the ability to transform raw data into actionable insights while adhering to specific analytical requirements.

Final Submission Checklist:

✅ Cleaned dataset ✅ Graphs with time restrictions ✅ Internship Report ✅ GitHub Repository

✅ Code Documentation