

Project Report

1. Introduction

This project focuses on building time-sensitive, filtered, and category-driven data visualizations for app marketplace data using Python and Plotly. The visualizations aim to support interactive dashboards that adapt based on business logic and user context.

2. Background

With the growing volume of app store data, visual analytics plays a crucial role in identifying trends, optimizing monetization strategies, and improving user engagement. This project implements three types of charts—scatter plot, dual-axis chart, and violin plot—with advanced filtering and time-based visibility to enhance decision-making.

3. Learning Objectives

- Understand how to filter and prepare data for visualization based on complex business conditions.
- Learn to create different chart types in Plotly for comparative and distribution analysis.
- Implement time-bound visual logic in Python.
- Gain experience integrating logic-based constraints in a dynamic visualization workflow.

4. Activities and Tasks

- Data Preparation: Synthetic and/or real-world app data was filtered based on multiple criteria such as type, installs, revenue, ratings, Android version, and content rating.
- Visualization Development:
 - Scatter plot to explore revenue vs. installs for paid apps.
 - Dual-axis chart to compare average installs and revenue across top categories for free and paid apps.
 - Violin plot to examine rating distribution across app categories.
- Time-based Control: Implemented time-gating logic to show/hide graphs based on the current IST time window.

5. Skills and Competencies Developed

- Data cleaning and transformation using Pandas
- Advanced filtering and conditional logic
- Interactive data visualization using Plotly
- Implementation of time-sensitive logic

- Analytical thinking in interpreting visualization requirements
- Writing modular, scalable Python code for dashboards

6. Feedback and Evidence

The visualizations:

- Showed clear patterns in revenue generation relative to installs in paid apps.
- Effectively illustrated differences in monetization and engagement between free and paid apps.
- Highlighted potential quality gaps in apps with lower ratings and specific naming patterns.

These charts can serve product managers, marketing teams, and data analysts in making informed strategic decisions.

7. Challenges and Solutions

Challenge:

Handling multiple complex filter conditions while ensuring accuracy and performance. Also, implementing reliable time-based rendering logic.

Solution:

Used robust Pandas filtering pipelines and incorporated ``pytz`` and ``datetime`` modules to ensure accurate time-zone handling and logic enforcement.

8. Outcomes and Impact

- Created reusable and scalable code for a data dashboard in Python.
- Introduced a smart way to limit chart visibility based on business hours.
- Enabled meaningful comparisons across app types and categories with dynamic filtering.

9. Conclusion

This project successfully demonstrates how to combine data wrangling, visualization, and logic programming to create intelligent, user-aware dashboards. The methodology can easily be extended to production environments where real-time insights are critical.