📓 Big Data Analytics Project: Google Play Store App Analysis

[Python] [PySpark] [MIT License]

🌟 Project Overview

This project is a comprehensive Big Data Analytics study focused on analyzing the Google Play Store ecosystem using PySpark. The analysis cleans, transforms, and explores app metadata to uncover key patterns in category dominance, monetization strategies, user engagement, and factors correlated with app success.

🎯 Key Objectives

1. Category Insights: Identify the most saturated and active app categories and genres.
2. Market Structure: Quantify the distribution of Free vs. Paid apps and pricing patterns.
3. Engagement Analysis: Study relationships between installs, ratings, and reviews.
4. Success Profiling: Filter and examine "highly successful" apps (Rating ≥ 4.5 and Installs > 1,000,000).
5. Visualization: Build clear charts to communicate trends and correlations.

💡 Key Analytical Findings

The PySpark analysis transformed raw app data into the following insights:

Finding	Detail	Implication
Market Composition	Free apps form the overwhelming majority of listings (≈ 90%+).	Freemium and ads/in-app purchases dominate monetization.
Category Dominance	FAMILY and GAME categories contain the highest number of apps.	Highly competitive space; differentiation is crucial.
Engagement Flywheel	Strong positive correlation between Installs and Reviews.	Growth and visibility are reinforced by user engagement.
Ratings Landscape	Most apps cluster around 4.0–4.7 ratings.	Very low-rated apps are rare/screened out; quality matters.
Success Profile	Top apps found not only in GAME but also COMMUNICATION, SOCIAL, PHOTOGRAPHY.	High-quality apps can succeed across multiple verticals.

🛠️ Tools and Technologies

- Primary Framework: Apache PySpark (distributed data processing and transformations).
- Environment: Jupyter Notebook (.ipynb).

- Libraries: pandas, matplotlib, seaborn (for EDA and visualization).

💾 Dataset

- File: googleplaystore.csv
- Scope: App metadata and performance metrics (approx. 10K+ apps).
- Key Fields: App, Category, Genres, Rating, Reviews, Installs, Type (Free/Paid), Price, Size, Content Rating, Android Ver, Last Updated.

🧹 Data Cleaning and Preprocessing

- Removed a known corrupted row: "Life Made WI-Fi Touchscreen Photo Frame".
- Standardized numerics:
    - Installs: stripped “+” and “,”, cast to Integer.
    - Price: stripped “$”, cast to Float.
    - Size: converted “M” to MB, “k” to MB/1024; filled nulls with average size.
- Casted dtypes: Rating → Float, Reviews → Integer, Price → Float, Installs → Integer, Size → Float.
- Filtered Type to valid values: Free or Paid.
- Dropped null ratings; enforced valid schema before analysis.

📊 Exploratory Data Analysis

- Category distribution (Top 10/15 categories).
- Free vs Paid breakdown.
- Rating distribution histogram.
- Scatter plot (Reviews vs Installs) on log-log scale.
- Correlation heatmap for numeric features.

🎯 Focused Analysis: Highly Successful Apps

- Criteria: Rating ≥ 4.5 and Installs > 1,000,000.
- Outputs:
    - Category and genre breakdown of elite apps.
    - Saved curated dataset to CSV (single file) using a temp directory + coalesce(1) pattern.

🚀 How to Run the Project

Prerequisites

- Python 3.9+
- Apache Spark (or use PySpark in local mode)
- Install libraries:

pip install pyspark pandas matplotlib seaborn

Execution

1. Launch Jupyter and open the notebook:

jupyter notebook

2. Open: notebooks/GooglePlaystoreAnalytics.ipynb
3. Run cells in order:
    - Initialize SparkSession.
    - Load CSV (set header=True).
    - Perform cleaning and dtype casting.
    - Run EDA + visualizations.
    - Filter and export “Highly Successful Apps”.

Saving a Single CSV

Use this helper approach to save a single-file CSV:

```
import os, shutil


def save_spark_df_as_single_csv(df, temp_dir, final_csv_path):
    df.coalesce(1).write.option("header", True).mode("overwrite").csv(temp_dir)
    part = [f for f in os.listdir(temp_dir) if f.startswith("part-") and f.endswith(".csv")][0]
    shutil.move(os.path.join(temp_dir, part), final_csv_path)
    shutil.rmtree(temp_dir)
```

Example:

```
save_spark_df_as_single_csv(
    highly_successful_apps_df,
    r"C:\Users\yourname\Downloads\temp_success",
    r"C:\Users\yourname\Downloads\highly_successful_google_play_apps.csv"
)
```

📊 Visualizations Included

- Top Categories by App Count (bar chart).
- Rating Distribution (histogram with KDE).
- Free vs Paid Distribution (pie chart).
- Box Plot: Ratings by Top Categories.
- Scatter: Reviews vs Installs (log-log).

- Correlation Heatmap: Rating, Reviews, Installs, Price, Size.

📝 Results & Deliverables

- Cleaned data with correct dtypes and standardized values.
- Visual dashboards highlighting key trends.
- Curated CSV of "Highly Successful Apps."
- Presentation (PPTX) and report (PDF) for submission-ready documentation.

📌 Future Work

- Sentiment analysis of user reviews (NLP).
- Time-series updates and changelog analysis.
- Cross-platform comparison (iOS vs Android).
- ASO features: title length, keywords, and impact on discoverability.

👤 Submitted By

K.madhukar reddy (2211CS010290)
Course: Big Data Analytics (Minor Project)
Title: Google Play Store Data Analysis using PySpark