Stuart School of Business
ILLINOIS INSTITUTE OF TECHNOLOGY

Google Play Analytics

MAX-595-01: Supply Chain Analytics
Professor Andrew Kumiega

Group 3:
Kobbie Antwi
Somya Garg
Tachin Ho

I. Project Overview

1. Introduction

This analysis utilizes a refined Google Play Store dataset to predict app performance, with a focus on Rating as a key indicator of user satisfaction. By leveraging actionable insights, developers can optimize app features, and businesses can target promotions and categories to enhance user satisfaction and drive app success.

2. Business Problem Statement

In the competitive app market, understanding what drives user satisfaction and app success is critical for developers and stakeholders. With thousands of apps competing for user attention, developers need to know which features to prioritize, and businesses must identify areas to invest in for growth. However, the challenge lies in analyzing a vast dataset to determine which variables have the most significant impact on app ratings. This analysis aims to address these challenges and provide clear, actionable insights to enhance app quality and user engagement.

3. Prediction Data Target

The target variable for prediction is Rating, which reflects user satisfaction and app quality. The objective is to identify the variables that have the most impact on Rating and leverage these insights to improve app performance. Key candidates include Reviews, Installs, and Rating Categories (High/Medium/Low), which collectively influence user perception and engagement.

4. Choice of Data

Focus on App Quality

- The Rating variable serves as a direct measure of user satisfaction and app quality, making it essential for predicting performance.
- Example: Understanding how features like Size and Price influence higher ratings provides clear avenues for improvement.

Actionable Insights

- Developers can enhance features such as size, pricing, and update frequency to boost retention.

- Businesses can refine strategies for promotions and focus on high-potential app categories.

High-Quality Dataset

- Cleaned and preprocessed to ensure no missing values in key variables like Rating.
- Enhanced with derived features (Log_Installs, Log_Reviews) to address skewness and improve predictive accuracy.

Business Relevance

- Provides insights to allocate resources effectively, improve app features, and address underserved categories, enabling data-driven growth strategies.

5. Data Dictionary

Column Name	Description	Data Type
Category	App Genre Category	Categorical
Rating	Average user rating of the app (range: 1.0 to 5.0).	Continuous
Reviews	Number of user reviews for the app.	Numeric (int)
Size	Size of the app in kilobytes.	Numeric (float)
Installs	Total number of app installations.	Numeric (int)
Type	Type of app (e.g., Free or Paid)	Categorical
Price	Price of the app (0 for free apps)	Numeric (float)

Content.Rating	Recommended user age group (e.g., Everyone, Teen)	Categorical
Genres	Primary app genre (e.g., Puzzle, Action)	Categorical
Last.Updated	Date when the app was last updated	Date
days_since_update	Days since the app was last updated	Numeric (int)
Price_Installs	Interaction effect combining price and installs	Numeric (float)
Log_Installs	Log-transformed value of the Installs column	Numeric (float)
Log_Reviews	Log-transformed value of the Reviews column	Numeric (float)
Rating_Category	Categorized version of Rating (e.g., High, Medium, Low)	Categorical
Scaled_Rating	Normalized Rating column	Numeric (float)
Scaled_Reviews	Normalized Reviews column	Numeric (float)
Scaled_Size	Normalized Size column	Numeric (float)
Scaled_Installs	Normalized Installs column	Numeric (float)
Scaled_Price	Normalized Price column	Numeric (float)

II. Data Transformation

Overview

The data transformation process focused on standardizing, cleaning, and preparing the dataset to ensure it was suitable for modeling. Key transformations included imputing missing values, converting data types, and creating new features to enhance predictive performance.

Steps in Data Transformation

1. **Handling Missing and Invalid Values**
    - Removed rows where the Rating was missing or greater than 5 (invalid values).
    - Replaced "Varies with device" in the Size column with NA and imputed missing values using the median size within each category.
2. **Standardization**
    - Converted the Size column to a uniform unit (kilobytes) for consistency.
    - Removed special characters in Reviews, Installs, and Price columns (+, ,, $) and converted them to numeric types.
3. **Feature Engineering**
    - Created Log_Installs and Log_Reviews by log-transforming the highly skewed Installs and Reviews columns to address skewness.
    - Generated days_since_update to capture recency as the difference in days between the current date and the Last.Updated field.
    - Designed an interaction feature, Price_Installs, to measure the combined effect of app price and installation count.
4. **Categorical Simplification**
    - Extracted the primary genre from the Genres column by keeping only the first entry (e.g., "Puzzle; Action" → "Puzzle").
    - Grouped rare genres into an "Other" category to reduce noise.
5. **Scaling**
    - Standardized numeric columns (Rating, Reviews, Size, Installs, Price) to ensure comparability using z-scores.
    - Scaled variables were added as new columns (Scaled_Rating, Scaled_Reviews, etc.) without overwriting the original data.
6. **Binning**
    - Categorized Rating into Rating_Category with three levels: "Low" (1-3), "Medium" (3-4), and "High" (4-5) for better interpretability.

Key Outcomes

The transformations resulted in a cleaner and more uniform dataset, with enhanced features to improve model interpretability and predictive power.

III. Data Exploration

1. Univariate Analysis

- **Objective**: Understand the distribution and characteristics of individual variables.
- **Insights**:
    - **Rating**: Most apps received ratings between 4 and 5, indicating a positive bias in user reviews.
    - **Installs**: Highly skewed, with a majority of apps having fewer installations, while a few popular apps had millions of installs.
    - **Reviews**: Similar skewness as Installs, with a few apps garnering the majority of reviews.
    - **Price**: Most apps were free, with only a small proportion of paid apps priced under $10.

2. Bivariate Analysis

- **Objective**: Explore relationships between variables to identify potential predictors of Rating.
- **Insights**:
    - **Rating vs. Installs**: Positive correlation observed; apps with higher installations tend to have better ratings.
    - **Rating vs. Price**: Free apps generally have higher ratings compared to paid apps.
    - **Rating vs. Reviews**: Strong positive relationship; more reviews indicate higher user engagement and satisfaction.

3. Multivariate Analysis

- **Objective**: Examine interactions between variables to uncover deeper insights.
- **Insights**:
    - Apps with high Installs and frequent updates (days_since_update) tended to maintain higher ratings.
    - Paid apps with low Price_Installs often had lower ratings, suggesting value-for-money considerations.

    - Genres like "Productivity" and "Games" dominated high-rating categories, while niche genres showed mixed results.

4. Visualization Summary

- **Rating Distribution**: A histogram revealed a left-skewed distribution, with most ratings concentrated in the 4-5 range.
- **Correlation Heatmap**: Highlighted significant relationships between Installs, Reviews, and Rating.
- **Scatter Plots**: Explored relationships such as Rating vs. Log_Installs and Rating vs. days_since_update, reinforcing earlier observations.

5. Conclusion

The data exploration phase highlighted critical variables influencing app ratings, such as Installs, Reviews, and Price. These insights guided feature selection and informed the subsequent modeling phase.

IV. Models + Cross-Validation

To determine the factors influencing app ratings, we utilized an eXtreme Gradient Boosted Trees (XGBoost) Regressor. This model is well-suited for handling structured data and provides excellent interpretability through feature importance analysis. Below are the key aspects of the model development process and other model compared with; XGBoost Regressor, a powerful algorithm for structured data, complemented by Random Forest Regressor for comparison.

1. Model Performance

- eXtreme Gradient Boosted Trees (XGBoost) Regressor
  Training Metrics: R^2
  Validation: 0.8210
  Cross-Validation: 0.8132
  Holdout: 0.8007

- eXtreme Gradient Boosted Trees (XGBoost) Regressor
  Training Metrics: MAPE
  Validation: 4.6528
  Cross-Validation: 4.6815
  Holdout: 4.7216

- RandomForest Regressor
  Training Metrics: R^2
  Validation: 0.8156
  Cross-Validation: 0.8019
  Holdout: 0.7951

- RandomForest Regressor
  Training Metrics: MAPE
  Validation: 4.7270
  Cross-Validation: 4.7660
  Holdout: 4.7688

Key Points:

- **Metrics and Results:**
  - **R^2 Scores:**
    - Validation: 0.8210 (XGBoost) vs. 0.8156 (Random Forest)
    - Cross-Validation: 0.8132 (XGBoost) vs. 0.8019 (Random Forest)
    - Holdout: 0.8007 (XGBoost) vs. 0.7951 (Random Forest)
  - **MAPE (Mean Absolute Percentage Error):**
    - Validation: 4.6528 (XGBoost) vs. 4.7270 (Random Forest)
    - Cross-Validation: 4.6815 (XGBoost) vs. 4.7660 (Random Forest)
    - Holdout: 4.7216 (XGBoost) vs. 4.7688 (Random Forest)

XGBoost outperforms Random Forest in both R^2 and MAPE metrics, indicating better accuracy and lower prediction error. The high R^2 values across all datasets confirm strong generalization ability. Lift charts validate the parity between predicted and actual ratings.

Explanation: This section compares model performances, highlighting XGBoost's superior capability for prediction accuracy and variance explanation in app ratings.

The high R^2 values across validation, cross-validation, and holdout sets indicate that the model generalizes well and captures a significant portion of the variance in-app ratings. The parity between predicted and actual values, as shown in the lift chart, validates the reliability of the model.

2. Model Insights

The SHAP feature importance analysis highlights the top predictors of app ratings:

- Rating Categories (High/Medium/Low): The most impactful variable, explaining the majority of the variance in ratings.

- Installs: Strongly correlated with app popularity and user engagement, making it the second most important factor.
- Reviews: Serves as a proxy for user satisfaction, further influencing ratings.

Additional features like days_since_update and content_rating showed moderate impact but were less significant than the top three variables.

3. Cross-Validation Process

- A 5-fold cross-validation approach was used to assess model robustness and avoid overfitting. This ensured that the performance metrics were consistent across different subsets of the data.
- The cross-validation R^2 score of 0.8132 confirms the model's stability when applied to unseen data.

4. Residual Analysis

- The residual plot shows a tight clustering of residuals around zero, with no significant outliers. This indicates that the model predictions are unbiased and accurately capture the target variable across the dataset.
- The histogram of residuals further supports this, displaying a normal distribution with minimal deviations.

5. Model Strengths

- Early Stopping: Implemented during training to prevent overfitting while maintaining optimal performance.
- Scalability: The XGBoost model efficiently handles the dataset's size and complexity.
- Feature Interpretability: The SHAP analysis provides actionable insights, enabling targeted interventions to improve ratings.

V. Results and Conclusion

Challenges

Selecting the appropriate data posed challenges, particularly with cleaning the dataset, identifying valuable insights, and determining relevant features to analyze. Ensuring the removal of data leakage while maintaining high predictive accuracy was crucial.

Results

Our analysis identified the following top three variables influencing app ratings:

1. **Rating Categories (High/Medium/Low):** A significant predictor, enabling a nuanced understanding of user sentiment.
2. **Installs:** Reflecting app visibility and popularity, directly impacting user trust and perception.
3. **Reviews:** Acting as a key indicator of user satisfaction and feedback.

Using the Extreme Gradient Boosted Trees Regressor, we achieved the following performance metrics:

- **Validation R^2:** 0.8210
- **Cross-validation R^2:** 0.8132
- **Holdout R^2:** 0.8007

The residual analysis and lift data confirm the model's robustness, with predictions closely aligning with actual values.

Solutions

- **Optimize Reviews and Installs:** Enhance app visibility and user satisfaction by incentivizing reviews and promoting app installations through targeted campaigns.
- **Leverage Rating Categories:** Focus on specific areas for improvement (e.g., feature enhancements) and segment user groups for personalized engagement strategies.

These insights empower app developers and businesses to allocate resources effectively, improve app features, and target high-potential categories for sustained growth.