

**INSTITUTE FOR ADVANCED COMPUTING AND**

**SOFTWARE DEVELOPMENT, AKURDI, PUNE**

**“Mobile app Data Analysis”**

PG-DBDA March 2024

Submitted By:

Group No: 26

Roll No. Name:

243528 Mukesh Kumar Yadav

243541 Rahul Patel

**Mr. Abhijeet Nagargoje Mr. Rohit Puranik**

Project Guide Centre Coordinator

**ABSTRACT**

Mobile app data analysis is essential for understanding user behaviour, optimizing user experience, and driving business decisions. By analyzing data generated from user interactions, developers can identify patterns, preferences, and potential issues within the app. This analysis helps in enhancing app performance, personalizing content, and improving retention rates. Additionally, it enables businesses to measure the effectiveness of marketing campaigns, track key performance indicators (KPIs), and ultimately increase revenue. Data-driven insights gained from mobile app analytics are crucial for staying competitive in a rapidly evolving digital landscape.

I conducted Exploratory Data Analysis (EDA) to identify key patterns, distributions, and relationships within the dataset. I then applied linear regression to model continuous outcomes and used logistic regression for binary classification tasks. To enhance prediction accuracy, I implemented a random forest algorithm, leveraging its ensemble approach. Additionally, I employed XGBoost to further improve model performance through gradient boosting, known for its efficiency and precision.

Analyzing mobile app data enables personalized user experiences, optimizes app performance, and predicts user churn. It informs targeted marketing, guides product development, and refines monetization strategies. Additionally, it aids in detecting security threats and fraud, ensuring a safer and more engaging app environment.

**ACKNOWLEDGEMENT**

I would like to extend my sincere and heartfelt thanks to my esteemed guide, Mr. Abhijeet Nagargoje, whose invaluable guidance, advice, and support have been instrumental in navigating the crucial stages of this work. His insights and encouragement have been pivotal in shaping the direction and outcome of this project. I am also profoundly grateful to our respected Centre Co-Ordinator, Mr. Rohit Puranik, for granting us access to the necessary facilities and for his continuous support throughout this endeavor. His cooperation made it possible for us to work effectively and efficiently. I would also like to express my appreciation to the other faculty members, whose knowledge and encouragement have contributed to the successful completion of this project. Lastly, I wish to extend my heartfelt thanks to my friends and family for their unwavering support and encouragement throughout this journey. Their belief in me has been a source of strength and motivation.

Mukesh Kr. Yadav(240341225027)

Rahul Patel (240341225040)

**Table of Contents**

|  |  |  |  |
| --- | --- | --- | --- |
| **Sr.No** | **Description** | **Page No.** | |
| 1 | Introduction | 1 | |
| 2 | Requirements | 2 | |
| 3 | About Dataset | 3 | |
| 4 | Work Flow | 4 | |
| 5 | Data Loading and Exploration | 4 | |
| 6 | Data Cleaning and Pre-processing | 5 | |
| 7 | Exploratory Data Analysis | 6 | |
| 8 | Visualization | 12 | |
| 9 | Evaluation Metric | 15 | |
| 10 | Model Building | 17 | |
| 11 | Conclusion | 25 | |
| 12 | References | 26 | |
|  | | |  | |

**List of Figures**

|  |  |
| --- | --- |
| Name | Page Number |
| Figure 1 Workflow Diagram | 1 |
| Figure 2 category vs count | 6 |
| Figure 3 Word Cloud | 7 |
| Figure 4 paid vs free pie chart | 8 |
| Figure 5 Scatter plot of size vs rating | 8 |
| Figure 6 Pie Chart of App Categories | 9 |
| Figure 7 Time Series plot of Last Updates | 10 |
| Figure 8 avg user rating for free & paid apps | 11 |
| Figure 9 free and paid apps comparision | 12 |
| Figure 10 Donut chart for different categories | 13 |
| Figure 11 Donut chart for different categories | 14 |
| Figure 12 Linear Regression | 18 |
| Figure 13 KNeighborsRegressor | 18 |
| Figure 14 RandomForestRegressor | 19 |
| Figure 15 Model Performance Comparison | 20 |
| Figure 16 Logistic Regression | 21 |
| Figure 17 KNeighbors Classifier | 22 |
| Figure 18 RandomForestClassifier | 22 |
| Figure 19 Model Comparision | 23 |
| Figure 20 XGBOOST | 24 |

**1. INTRODUCTION**

* 1. **PROBLEM STATEMENT:**

Analyze Mobile App Data on various platforms and generate recommendations based on the analysis and various metrics available in the datasets using Python

* 1. **PRODUCT SCOPE:**

The scope of my mobile data analysis project includes understanding user behavior, monitoring app performance, and predicting churn. It also involves personalizing user experiences, optimizing marketing strategies, and refining monetization efforts. Additionally, the project aims to detect security threats and guide feature enhancements, all with the goal of improving user satisfaction and supporting informed decision-making.

* 1. **AIMS & OBJECTIVES:**

The aims of my mobile data analysis project are to enhance user experience and engagement, optimize app performance, and support informed business decisions. The objectives include analyzing user interaction data to identify key engagement metrics and usage patterns, predicting user churn to develop retention strategies, and personalizing content based on user preferences. The project also focuses on improving app performance by addressing issues such as crashes and slow load times, refining marketing strategies through user segmentation, monitoring and mitigating security threats, and guiding the development of new features and improvements based on user data.

**2. REQUIREMENTS**

**2.1 Hardware Requirement**

• 500 GB hard drive (Minimum requirement)

• 8 GB RAM (Minimum requirement)

• PC x64-bit CPU

**2.2 Software Requirement:**

• Windows/Mac/Linux

• Python-3.10.12

• Google Colab

**2.3 Libraries:**

* Numpy 1.26.4
* Pandas 2.1.4
* Matplotlib 3.7.1

**3. ABOUT DATASET**

* App : The name of the app
* Category : The category of the app
* Rating : The rating of the app in the Play Store
* Reviews : The number of reviews of the app
* Size : The size of the app
* Install : The number of installs of the app
* Type : The type of the app (Free/Paid)
* Price : The price of the app (0 if it is Free)
* Content Rating : The appropriate target audience of the app
* Genres: The genre of the app
* Last Updated : The date when the app was last updated
* Current Ver : The current version of the app
* Android Ver : The minimum Android version required to run the app

**3.1.Name of Columns and their data types**

* currency object
* price float64
* rating\_count\_tot int64
* rating\_count\_ver int64
* user\_rating float64
* user\_rating\_ver float64
* ver object
* cont\_rating object
* prime\_genre object
* sup\_devices.num int64
* ipadSc\_urls.num int64
* lang.num int64
* vpp\_lic int64
* app\_desc object

**4.WORKFLOW**

**4.1 Workflow of Project:**

Figure 1 Workflow Diagram

* **Data Loading and Exploration:** The first step involves loading the dataset and performing an initial exploration to understand its structure, features, and any potential issues.
* **Cleaning & Preprocessing:** After loading the data, it needs to be cleaned and preprocessed. This includes handling missing values, encoding categorical variables, scaling features, and possibly transforming the data to make it suitable for analysis.
* **Exploratory Data Analysis (EDA):** EDA is conducted to uncover patterns, correlations, and insights from the data. This step helps in understanding the relationships between variables and identifying any anomalies.
* **Visualization:** Visualizations are created to present the findings from EDA in an intuitive and comprehensible manner, often using charts, graphs, and plots.
* **Model Building:** Based on the insights gathered, a predictive model is built. This step involves selecting appropriate algorithms, training the model on the data, and evaluating its performance.

**5. DATA LOADING AND EXPLORATION:**

* Load the Dataset: Import the dataset and display basic information.
* Exploratory Data Analysis (EDA):
* Check for missing values.
* Summarize statistics and visualize feature distributions.
* Analyze correlations between features and the target variable (app ratings).

**6. DATA CLEANING AND PRE-PROCESSING:**

**6.1 Data cleaning:**

The data can have many null values, missing values for categorical and numerical features. To handle this part, data cleaning is done.

**6.2 Filling values in categorical column:**

To fill null values, present in categorical columns with mode value. Columns like [App, Category, Installs, Rating etc.]

**6.3 Filling values in numerical column:**

To handle missing values present in numerical columns, filling it with median. Columns like [Installs, Rating, Size etc.]

**6.4 Dropping duplicates:**

The data may contain multiple duplicates in dataset columns. Thus, duplicate values must be dropped for efficiency and quality of data.

6.5 **Data Preprocessing:**

* Handling Missing Values: Impute or remove missing values.
* Encoding Categorical Variables: Convert categorical features into numerical values.
* Feature Scaling: Normalize or standardize features if necessary.
* Feature Engineering: Create or modify features to enhance model performance

**6.6 Label encoding**:

Label encoding is a method for converting categories into numbers. Each unique category is assigned a different number so that algorithms can work with these values. This is useful for turning text data into a format that can be processed by machine learning models.

**7. EXPLORATORY DATA ANALYSIS:**

Exploratory Data Analysis (EDA) is the process of analysing and visualizing data to understand its structure, patterns, and relationships. It involves summarizing key statistics, detecting anomalies, and uncovering insights to guide further analysis. EDA helps in making informed decisions about data preprocessing and model selection.

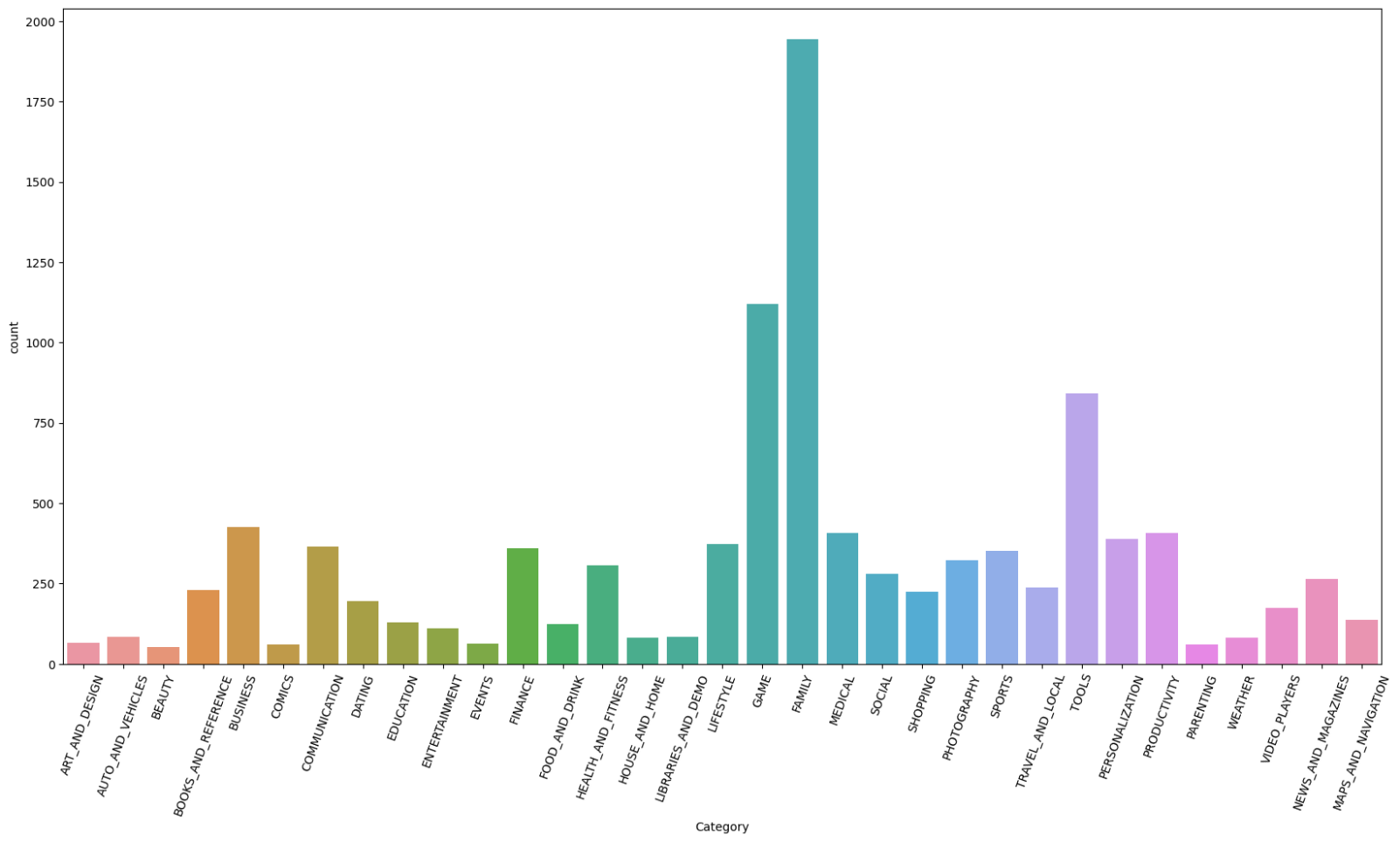


Figure 2 category vs count

The bar chart shows the count of different categories of apps on the Google Play Store. The category with the highest count is Medical, followed by Communication and Books and Reference. The remaining categories have counts ranging from 0 to 500.

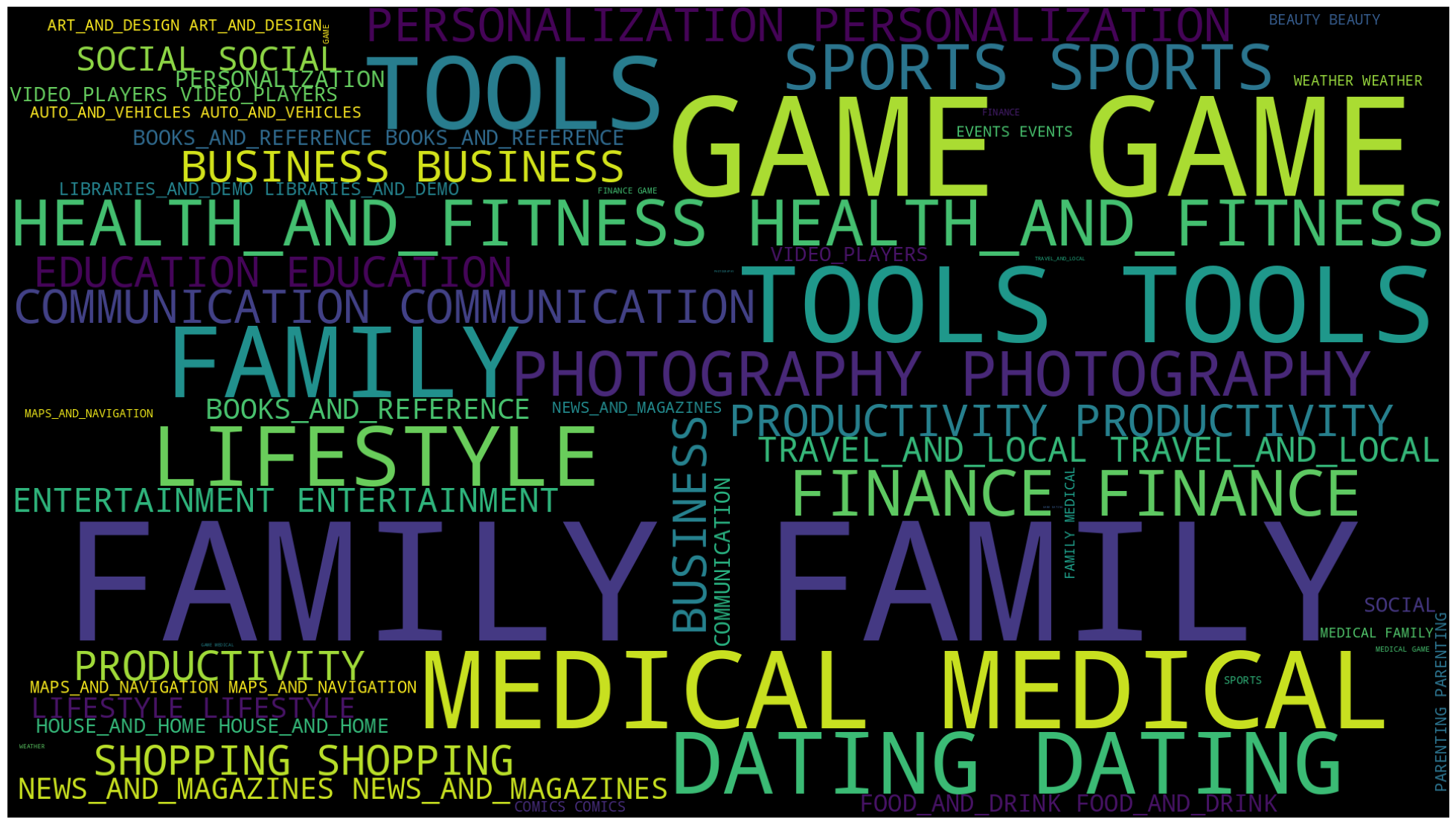


Figure 3 Word Cloud

The image is a word cloud representing different app categories on the Google Play Store. The most frequent categories are "Game", "Tools", "Medical" and "Family".

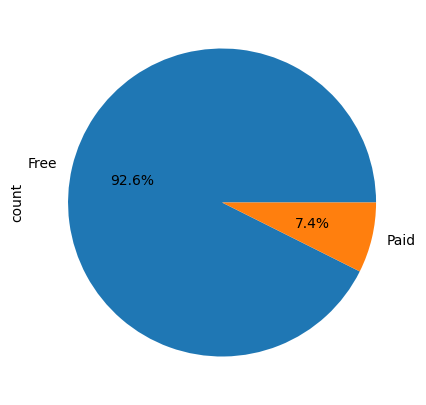


Figure 4 paid vs free pie chart

The pie chart displays the distribution of free and paid apps, with free apps comprising 92.6% and paid apps accounting for 7.4% of the total.

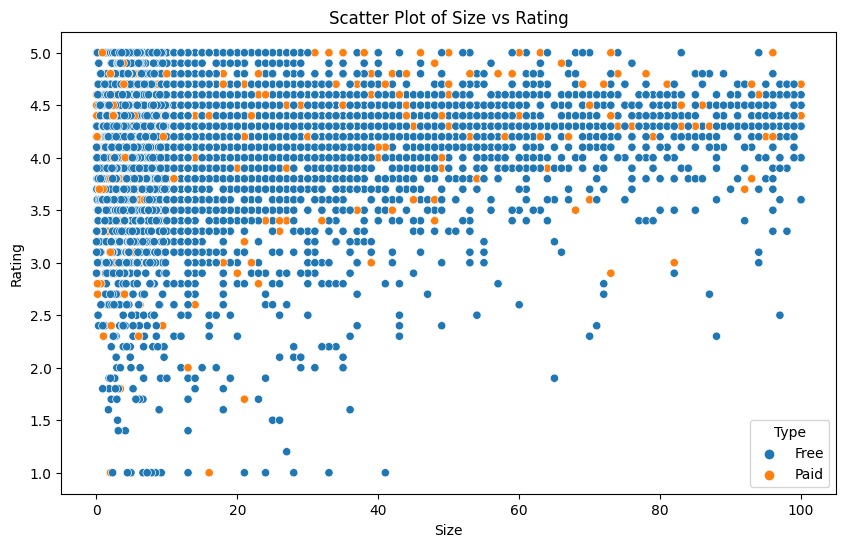


Figure 5 Scatter plot of size vs rating

The scatter plot shows the relationship between app size and rating, with data points differentiated by app type (free or paid).

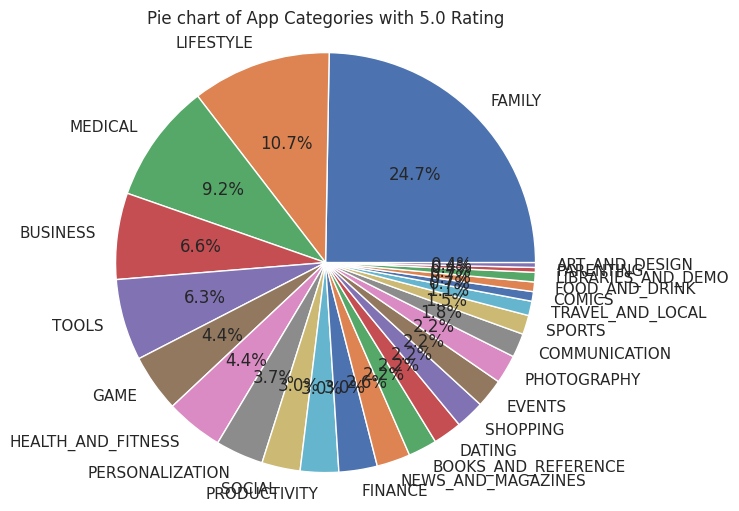


Figure 6 Pie Chart of App Categories

The pie chart shows the distribution of app categories with a 5.0 rating on the Google Play Store. Family and Medical apps have the highest ratings, together comprising over 35% of the total.

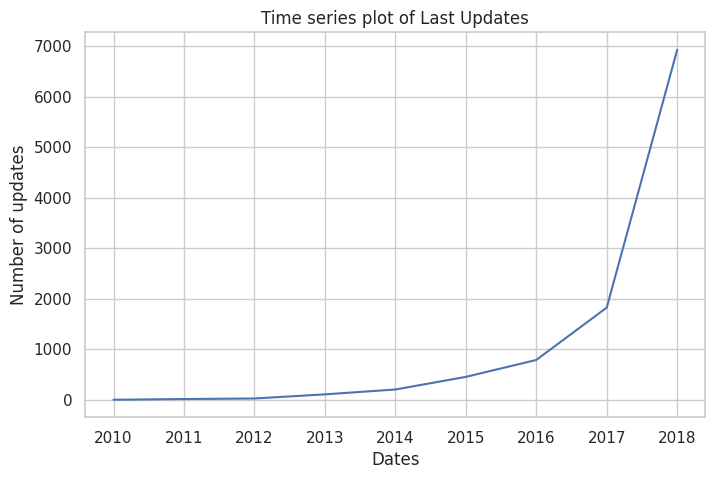


Figure 7 Time Series plot of Last Updates

The time series plot depicts the number of updates over time from 2010 to 2018. There's a significant increase in the number of updates starting from 2016.

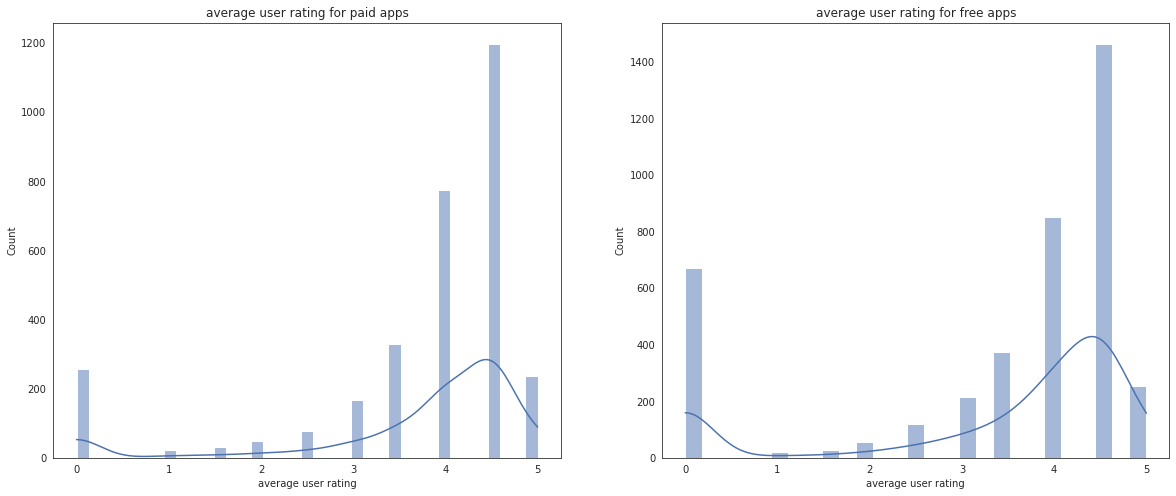


Figure 8 avg user rating for free & paid apps

The image displays two histograms comparing the distribution of average user ratings for paid and free apps. The distribution for paid apps is slightly narrower and peaks around a higher rating than that of free apps.

**8.Visualization**

Pandas offers built-in visualization capabilities for quick data exploration. It provides a simple interface to create various plot types like line, bar, histogram, boxplot, scatter, and more, directly from DataFrames and Series. While basic, it's useful for initial data understanding. For more complex and customized visualizations, libraries like Matplotlib and Seaborn are often preferred.

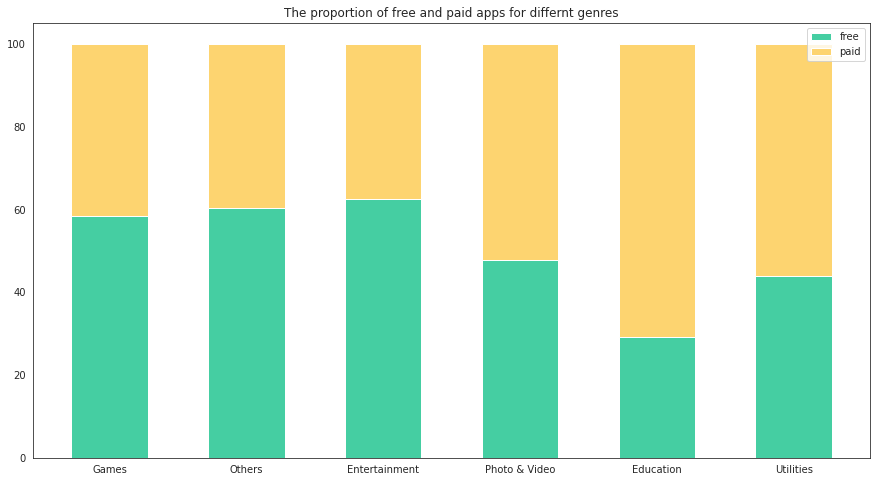


Figure 9 free and paid apps comparision

The stacked bar chart visualizes the proportion of free and paid apps across different genres. Games, Others, and Entertainment genres have the highest proportion of free apps.

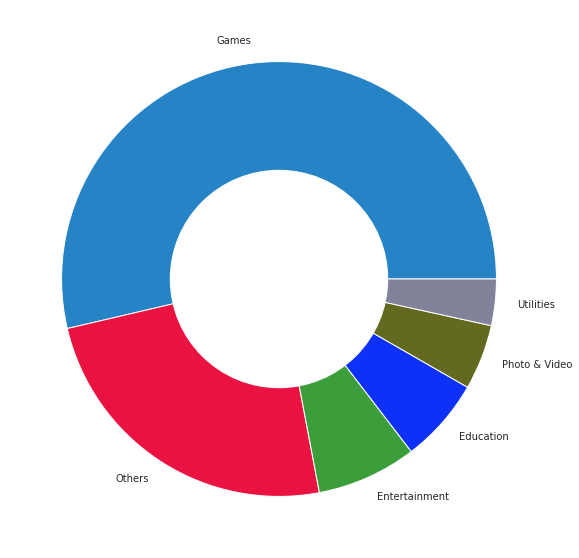


Figure 10 Donut chart for different categories

The pie chart shows the distribution of app categories on the Google Play Store. Games is the largest category, followed by Others and Photo & Video.

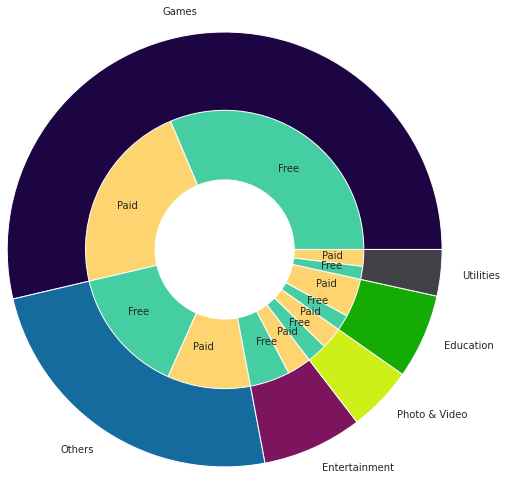


Figure 11 Donut chart for different categories for paid & free

The image shows a donut chart representing app categories and their distribution into free and paid versions. Games is the largest category, followed by Others and Photo & Video, with varying proportions of free and paid apps within each category.

**9.EVALUATION METRIC FOR REGRESSION**

The essential step in any machine learning model is to evaluate the accuracy of the model. In regression, evaluation metrics assess how well a model predicts continuous outcomes. Common metrics include Mean Absolute Error (MAE), Mean Squared Error (MSE), R-squared error (R²). Using these metrics helps in understanding the model's predictive performance and refining it for better accuracy.

**9.1 Mean Absolute Error:**

Mean Absolute Error (MAE) is a key metric for evaluating the accuracy of a regression model in machine learning. It calculates the average absolute difference between predicted values and actual outcomes, providing a clear measure of prediction accuracy without penalizing larger errors more than smaller ones.

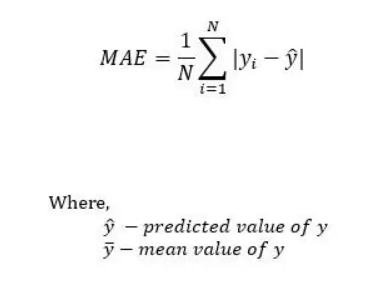
****

Figure 2: Formula of Mean Absolute Error.

**9.2 Mean Squared Error:**

Mean Squared Error (MSE) is a commonly used metric for evaluating regression models in machine learning. It measures the average squared difference between predicted values and actual outcomes, emphasizing larger errors more than smaller ones due to the squaring of residuals. This makes MSE sensitive to outliers and useful for assessing overall model accuracy, particularly when penalizing significant errors is important. Its quadratic nature can help in identifying models that are less robust to large deviations.

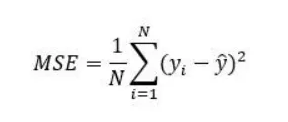


Figure 3: Formula of Mean squared error

**9.3 R-Squared Error:**

R-squared (R²) is a statistical metric that measures the proportion of variance in the dependent variable that is explained by the independent variables in a regression model. It ranges from 0 to 1, where a value of 1 indicates that the model explains all the variability of the outcome, and 0 means it explains none. R² helps to assess the goodness-of-fit of the model, providing an insight into how well the model captures the underlying data patterns.

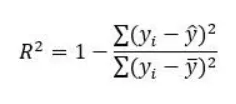


Figure 4: Formula of R- Squared Error.

**10. MODEL BUILDING**

**1 Train Test Split:**

One important aspect of all machine learning models is to determine their accuracy. Now, in order to determine their accuracy, one can train the model using the given dataset and then predict the response values for the same dataset using that model and hence, find the accuracy of the model. A better option is to split our data into two parts: first for training our machine learning model, and second one for testing our model

* Split the dataset into two pieces: a training set and a testing set.
* Train the model on the training set.
* Test the model on the testing set, and evaluate how well our model did.

Advantages of train/test split:

* Model can be trained and tested on different data than the one used for training.
* Response values are known for the test dataset, hence predictions can be evaluated
* Testing accuracy is a better estimate than training accuracy of out-of-sample performance.

**2.Building Model:**

Machine learning consists of algorithms that can automate analytical model building. Using algorithms that iteratively learn from data, machine learning models facilitate computers to find hidden insights from Big Data without being explicitly programmed where to look.

We have used the following algorithms to build predictive model.

* 1. **. Linear regression:**

Linear regression is a statistical method used to model the relationship between a dependent variable and one or more independent variables. It aims to find the best-fitting line through the data points, which minimizes the difference between observed and predicted values. This technique helps in making predictions and understanding how changes in independent variables impact the dependent variable.

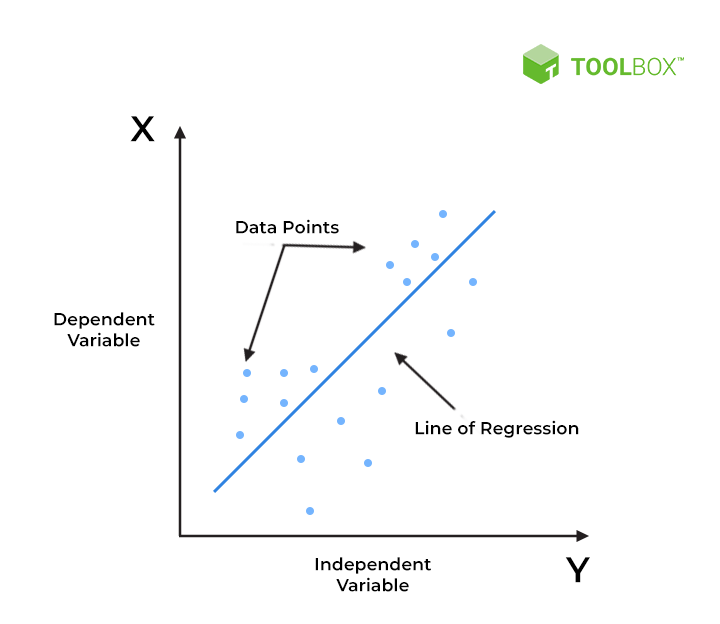


Figure 12 Linear Regression

**10.2.KNeighborsRegressor:**

The KNeighborsRegressor is a regression model in which the prediction for a given data point is based on the average of the target values of its nearest neighbors in the feature space. The number of neighbors (k) is a parameter that you can choose. The model essentially assumes that similar input data points will have similar output values.

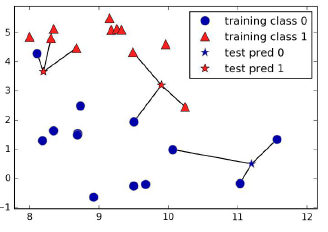


Figure 13 KNeighborsRegressor

**10.3 RandomForestRegressor:**

The RandomForestRegressor is an ensemble model that combines the predictions of multiple decision trees to make a final prediction. Each tree is trained on a random subset of the data and features, which helps to reduce overfitting and improve accuracy. The final prediction is typically the average of the predictions from all the individual trees**.**

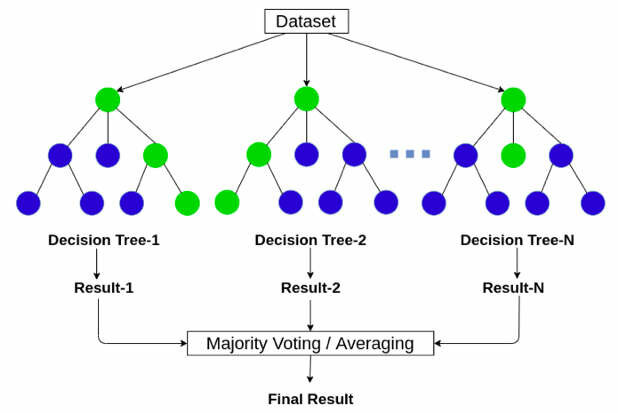


Figure 14 RandomForestRegressor

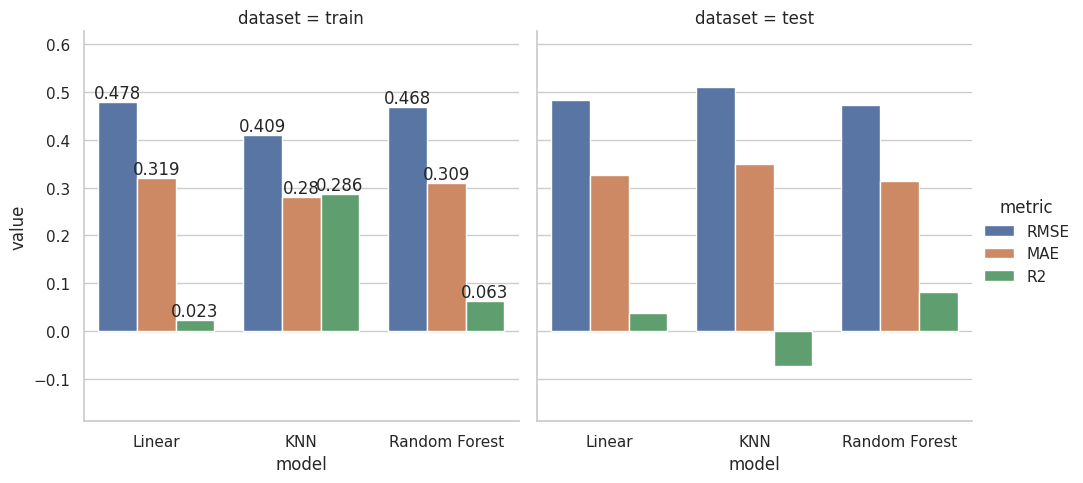


Figure 15 Model Performance Comparison

The image presents a comparison of three machine learning models (Linear, KNN, Random Forest) on two datasets (train and test) using three evaluation metrics (RMSE, MAE, R2).

Interpretation:

* **RMSE and MAE:** Lower values indicate better performance, suggesting Random Forest generally outperforms Linear and KNN models on both datasets.
* **R2:** Higher values are better, indicating Random Forest again shows superior performance in explaining the variance in the data.
* **Train vs. Test:** The performance difference between train and test sets can give insights into model overfitting or underfitting.

Overall, the visualization suggests that the Random Forest model provides the best performance across both datasets and evaluation metrics.

**10.4. Logistic Regression :**

Logistic regression is a supervised machine learning algorithm used for classification tasks where the goal is to predict the probability that an instance belongs to a given class or not. Logistic regression is a statistical algorithm which analyze the relationship between two data factors. Logistic regression is used for binary [classification](https://www.geeksforgeeks.org/getting-started-with-classification/) where we use [sigmoid function](https://www.geeksforgeeks.org/derivative-of-the-sigmoid-function/), that takes input as independent variables and produces a probability value between 0 and 1.

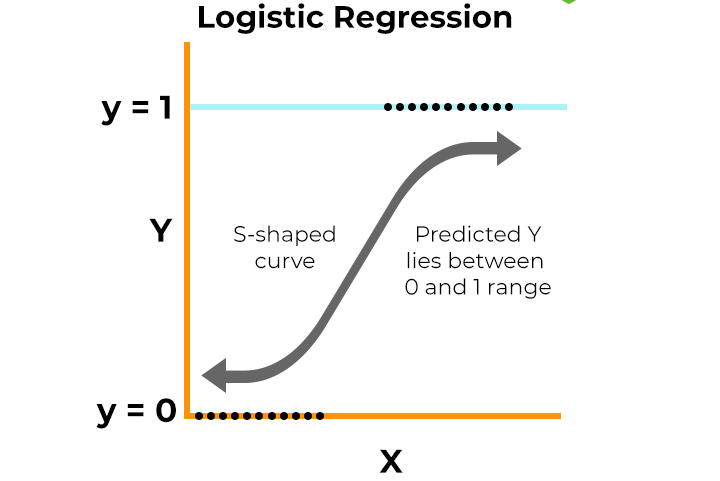


Figure 16 Logistic Regression

**10.5 . K-Nearest Neighbors Classifier:**

K-Nearest Neighbors (KNN) is a simple yet effective classification algorithm. It works by finding the K closest data points to a new, unseen data point and assigning it the class that is most frequent among those K neighbors. Think of it as classifying something based on its similarity to nearby examples.

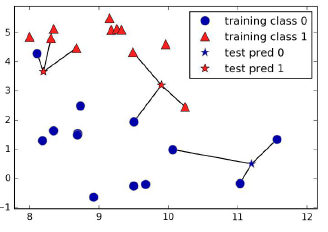


Figure 17 KNeighbors Classifier

**10.6. RandomForestClassifier:**

Random Forest Classifier is an ensemble method that combines multiple decision trees.

Diversity: Each tree is built on a random subset of the data and considers a random subset of features.

Prediction: Each tree makes a prediction, and the final prediction is determined by the majority vote of all trees.

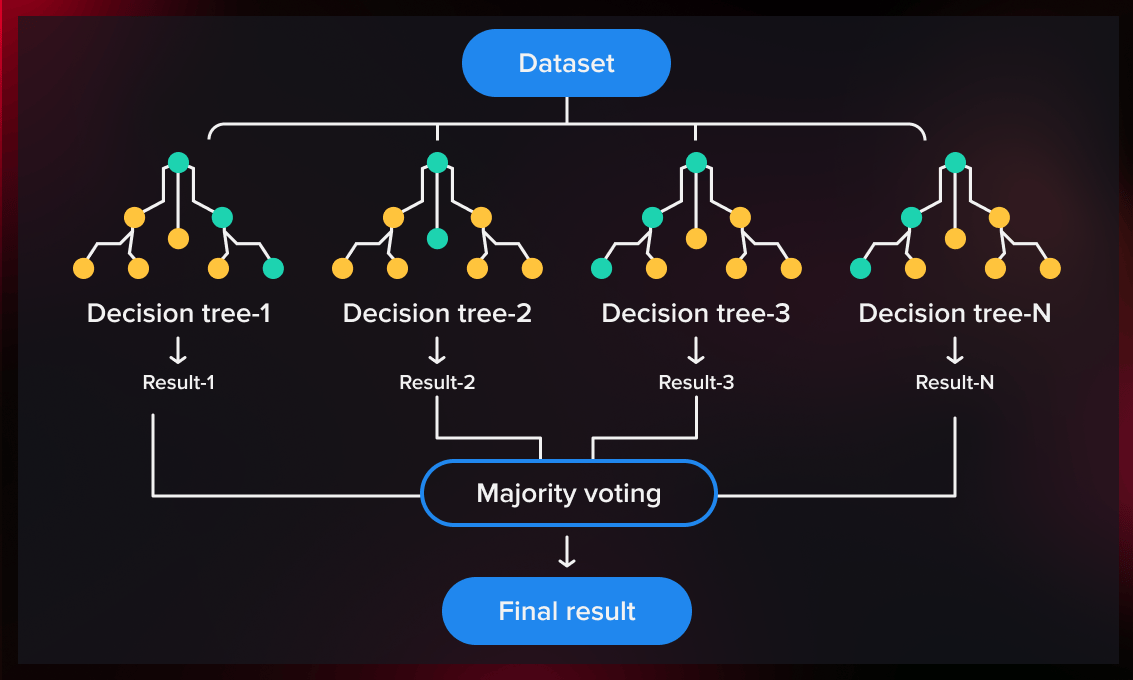
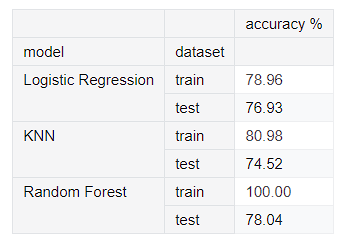


Figure 18 RandomForestClassifier

Classification Evaluation



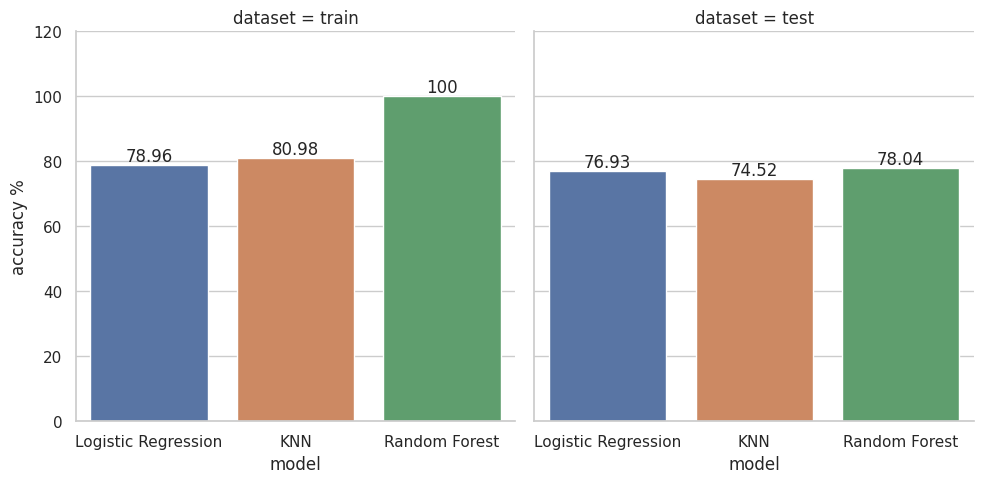


Figure 19 Model Comparision

The image presents a comparison of three machine learning models (Logistic Regression, KNN, and Random Forest) on two datasets (train and test) using accuracy as the evaluation metric.

**10.7. XGBOOST Classifier:**

XGBoost (Extreme Gradient Boosting) is an advanced implementation of the gradient boosting algorithm. It builds decision trees sequentially, where each tree tries to correct the errors made by the previous ones. The key features include regularization to prevent overfitting, handling missing data automatically, and parallel processing for faster computation. XGBoost is widely used for its high performance and scalability in machine learning tasks.

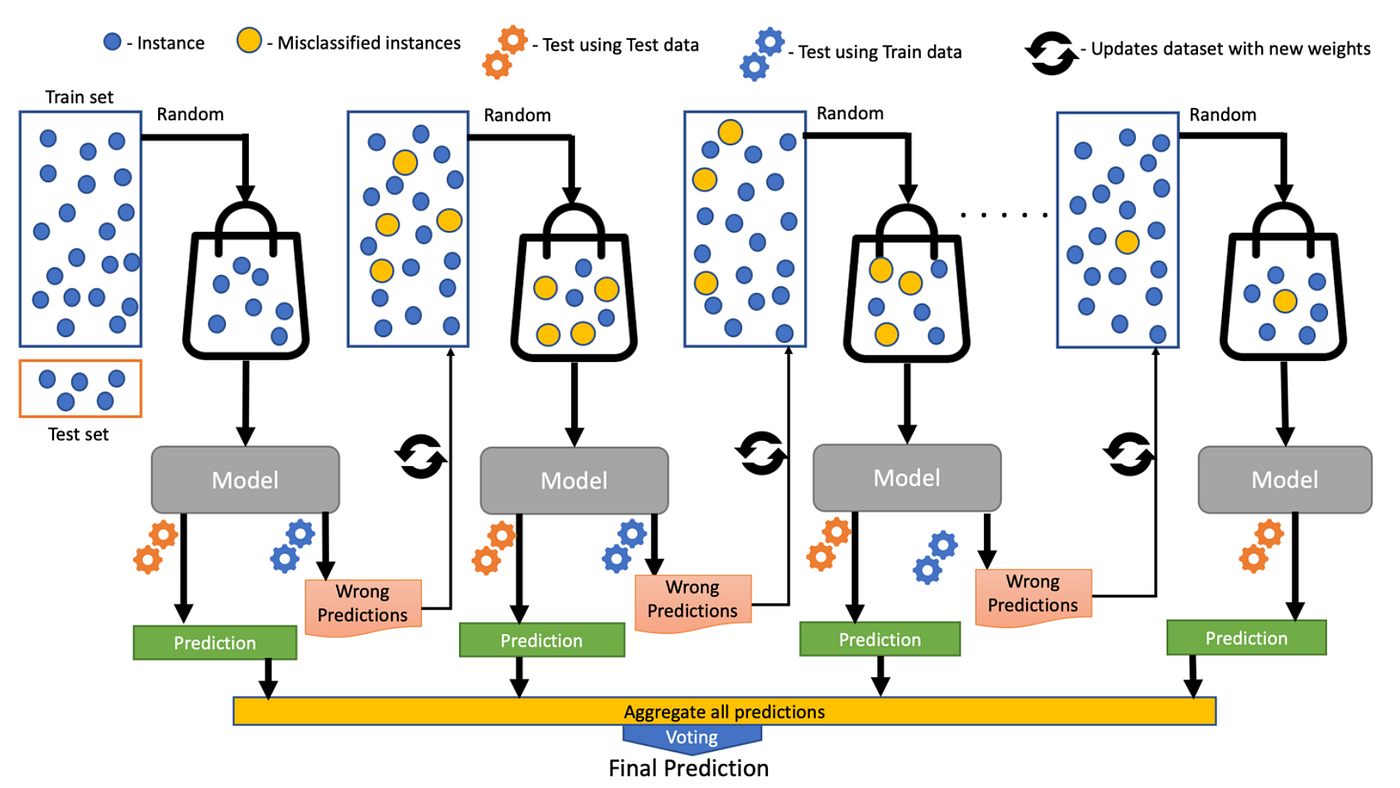


Figure.20 XGBOOST

**11.Conclusion**

Our analysis of Google Play and App Store data reveals that "Games" and Entertainment are dominant genres across both platforms. While Social Networking is a shared area of popularity, the App Store leans heavily towards practical applications like Reference, Communication, and Utilities.

Given this insight, we propose focusing on developing free, practical apps that cater to a broad audience. Potential app ideas include a comprehensive Bible app (considering the significant Christian population in the US) and a food discovery platform (leveraging the universal appeal of food and social sharing).

To further refine our app concept, detailed discussions with product managers are essential to align with user needs and market trends.

Avoiding the highly competitive gaming market, especially on iOS, due to technical constraints and smaller user base, is recommended.

**12.REFERENCES**

1. <https://www.kaggle.com/datasets/lava18/google-play-store-apps>

2. <https://www.kaggle.com/datasets/ramamet4/app-store-apple-data-set-10k-apps>

3. <https://pandas.pydata.org/docs/>

4. <https://scikit-learn.org/stable/>

5. <https://machinelearning101.readthedocs.io/en/latest/_pages/01_introduction.html>