****

**PROJECT PROPOSAL -3**

**BAN5600-02-F24– Advanced Big Data Computing and Programming**

**App Store Vs Play Store**

**GROUP MEMBERS:**

**AARYAN PANKAJ DABHADE**

**MRIDULA KOIRALA**

**Prof. Hamidreza Ahady Dolatsara**



**Motivation**

The era of smartphone apps has unearthed a great deal of data regarding app downloads, user interactions, and feedback (available on official repositories like the App Store and Play Store). However, this data must be properly studied by businesses, app developers, and marketers to know better potential customers' tendencies and attitudes– the so-called marketing. Every decision; big or small is almost impossible to make under conditions where data sources are not nonexistent. It is essential because a sample which is Big Data is the key to satisfaction from the product, marketing, and customer feelings, therefore, providing Big Data with many great benefits for every business, anything that guarantees potential performance excellence. This research work will involve designing appropriate Big Data applications to analyse Big Data to maximize expected results in terms of time spent in using an application and earning more income from applications.

**Introducing the Data and Topic**

Mobile Applications have become an important part of our daily lives in the digital world era, there are thousands of apps on App Store and Play Store, etc. App developers, marketers, and businesses must manage the increasing complexity of user preferences, app performance, and market trends to succeed. The data generated from app downloads are perhaps the largest volume and variety of data, enabling data-driven decision-making. In this project, we will take a closer look at the collected App Store and Play Store datasets to try to come up with some ideas based on the number of apps, relationships between them or review sentiment. Using Big Data analytics, it will delve deep into patterns that could be used to enhance app discovery and visibility, waning user retention, as well as broader market strategy.

**Project Scope**

**Data Collection & Preprocessing:** Acquiring data with respect to app statistics, ratings, comments, and other relevant information from the App Store and the Play Store. Checking the quality of the gathered information and preparing it for analysis.

**Descriptive Analysis:** Making summaries based on data such as app genres, trend in downloads and key measures of performance, that is important in this research. This will lead to comprehension of the most rated application genres, their percentage distribution and outdoors, and their relation to the time of the year.

**Predictive Modeling:** This involves the development of models such as regressions to predict various dependent variances like app download trends, user retention and app rankings. Such assistance will also form the basis for the assessment made by software implementers in order to better the app functioning.

**Recommendation System:** Implementing a recommender system based on app download patterns and user preferences, which can recommend users to other similar apps and which can help maximize the user engagement in the app.

**Reporting & Visualization:** Creating user-friendly visual representations and more descriptive assumptions which will help in faster appreciation of the insights and the trends thus enabling the decision makers to approach the issues with confidence.

**Data Source and Description**

**Data Source:** The dataset was found on Kaggle with App Store 7,197 number of rows and 7 columns and for Playstore it's 7,197 number of rows and 9 number of columns. The dataset is a number of downloads and categories of applications unique, covering various transactions made on the platform. This proprietary dataset has been anonymized and aggregated to respect privacy and confidentiality.

**Description of the Dataset**

**PLAY STORE**

● Gender: Male and Female

● Age: Ranges from 14-70 years

● App: Application names

● Rating: Numeric app rating

● Type: Free and Paid

● Price: cost of the app in numbers.

● Content Rating: Age-appropriateness category

● Genres: Application category

**APP STORE**

● Gender: Male and Female

● Age: Ranges from 14-70 years

● track\_name: Application names

● price: Cost of the app in numbers.

● user\_rating: Numeric rating given by users

● cont\_rating: Content rating category

● prime\_genre: Primary genre or category of the app

**Literature Review**

Analysis of the Play Store versus App Store datasets and the mobile applications ecosystems adds useful understanding of the users, their networks, and the developers across the platforms. Among other topics, Gill investigates the B2B mobile apps that engage users and therefore aim at increasing sales and business development through strong customer relations. While this focus is B2B, it provides a methodological framework that merits application in consumer apps in the understanding of relevant engagement metrics, including app ratings and reviews, in the overall success of an app. Zeydan (2021) specializes in comparative analysis of Android and iOS platforms, also describing the mobile network performance factors of the sites, download speed, and latency. The present analysis shows that iOS is great in terms of latency while Android does well in download speed which in one way or the other, determines the application usability and success across app stores globally. Wang et al. (2018) performed a comprehensive analysis of Chinese Android app markets relative to Google Play and found out that the behavioral practices of developers as well as rate of malicious application presence vary from one market to the other. The study invites attention to how these variables might in the first instance compromise the quality and safety of apps, thereby determining the users’ level of trust and ratings of the apps. Each study offers distinct contributions on the web of factors that affect app's success on various platforms. Gill et al. Help investigate the web of factors that determine user participation, Zeydan helps look at the importance of network while Wang et al. help in understanding the influence of developer and security practices towards apps and apps ecosystems. Your findings would also help in outlining your analysis of data from the Play Store and App Store especially regarding user interaction aspects, differences of their performance and developers.

In the last studies comparing the App Store and Play Store, the focus has primarily been on platform-based differences, such as the availability of apps or general app store characteristics. However, these studies often lack a deep analysis of factors such as app ratings, app popularity, or the impact of user demographics on app preferences. In this project, we aim to fill that gap by conducting a more detailed analysis based on gender, the type of application (paid or free), and app ratings. By using the datasets provided by both the Play Store and App Store, we will explore how these factors affect app popularity and user engagement across platforms.

**Research Question**

Q1- Compare apps on both platforms to see how their ratings and installs differ between iOS and Android users.

Q2- How do gender preferences influence app choices and ratings across both platforms and do male or female users prefer certain types of apps more?

**The significance of this study**

The significance of this study lies in its detailed comparative analysis of app ecosystems on the Play Store and App Store platforms, focusing on key factors like user demographics, monetization models (free vs. paid apps), and user ratings. This study bridges the gap in the literature by not only comparing general platform characteristics but also by delving deeper into specific metrics such as app popularity, gender preferences, and the influence of ratings across the platforms.

**Initial Visualization**

**1- Gender Distribution based on the number of downloads (Male & Female)**

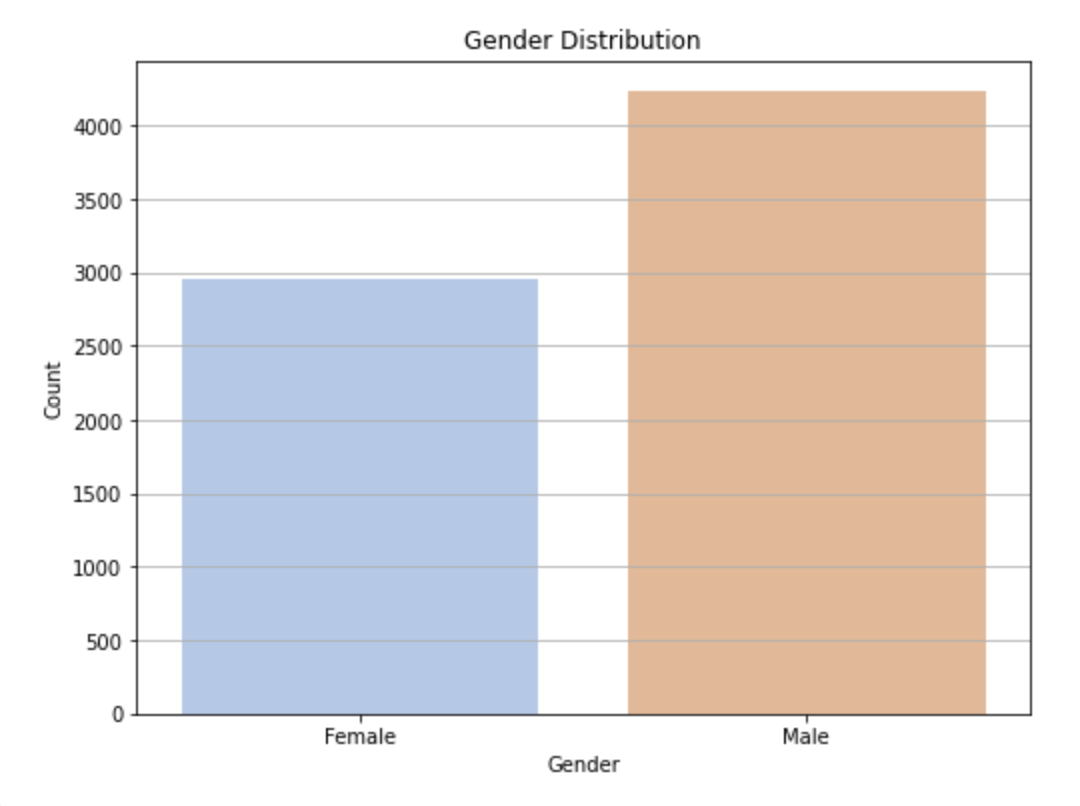
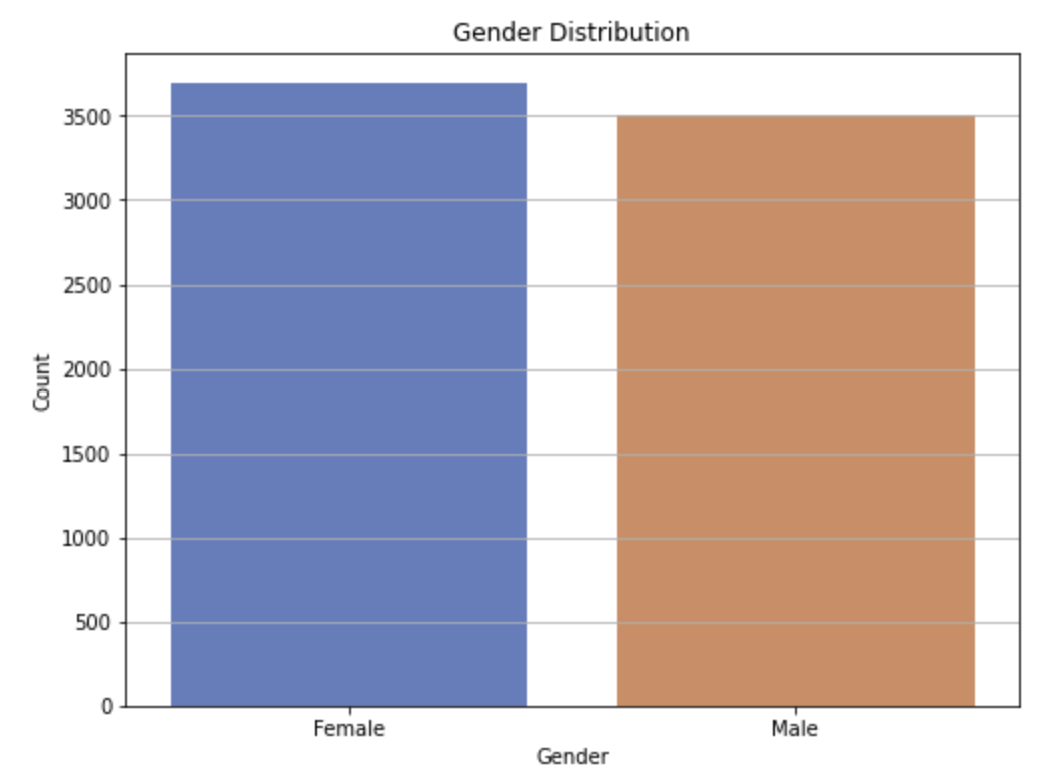


Fig-1.1 Fig-1.2

**Source code**

# Convert to Pandas DataFrame

pdf = df.toPandas()

**#Here, we convert the PySpark DataFrame to a Pandas DataFrame (pdf). This step is necessary because Seaborn and Matplotlib are built to work with Pandas DataFrames or other native Python data structures, not PySpark DataFrames.**

# Count Plot of Gender Distribution

plt.figure(figsize=(8, 6))

sns.countplot(x='Gender ', data=pdf, palette='muted')

# Title and labels

plt.title('Gender Distribution')

plt.xlabel('Gender')

plt.ylabel('Count')

plt.grid(axis='y')

plt.show()

**Explanation**

The study of Play Store users gender chart suggests that both men and women are represented in equal numbers with each gender having the number of close to three thousand five hundred. This balance indicates that the users of the Play Store are people from all walks of life since both men and women are being embraced at the same level. This implies that the applications uploaded on Play Store are able to meet the needs of relatively many people. In a different case, App Store gender distribution shows that males are more, with their number going over four thousand while females number slightly over three thousand. Such a difference suggests that male users are more on App Store. This also means that applications on this platform are meant for male users, perhaps, restructuring the design, content and marketing of the app.

**Real-time usage**

App developers can use gender distribution insights to design apps that cater to the needs of their primary user base. For instance, if App Store has a male-dominant user base, developers can optimize app features, themes, or functionalities to cater more to this demographic. Conversely, knowing that the Play Store has a balanced gender distribution could inspire a more universal design.

Companies can devise more effective retention strategies based on gender data. For example, if male users dominate the App Store, developers might offer more frequent promotions or content updates that appeal to male interests, such as sports, technology, or gaming, to maintain user engagement.

**2- Age distribution and users for App Store & Play Store**

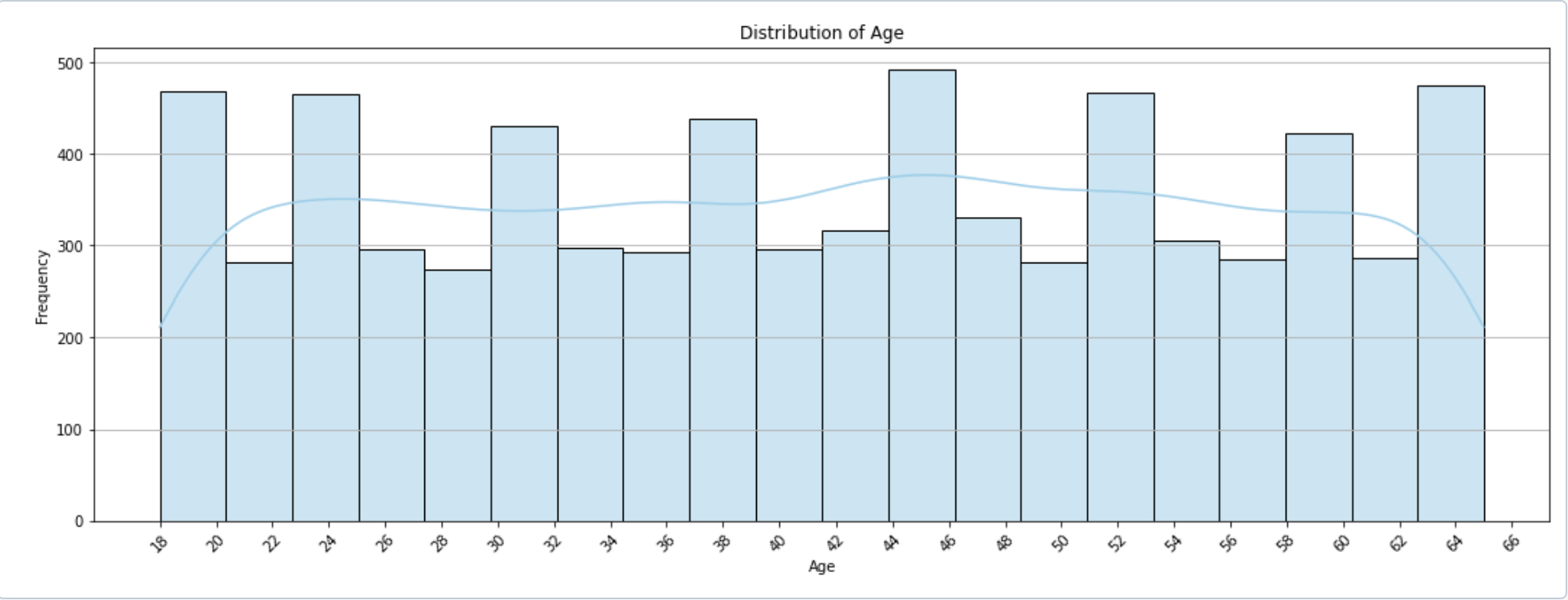
****

Fig-2.1

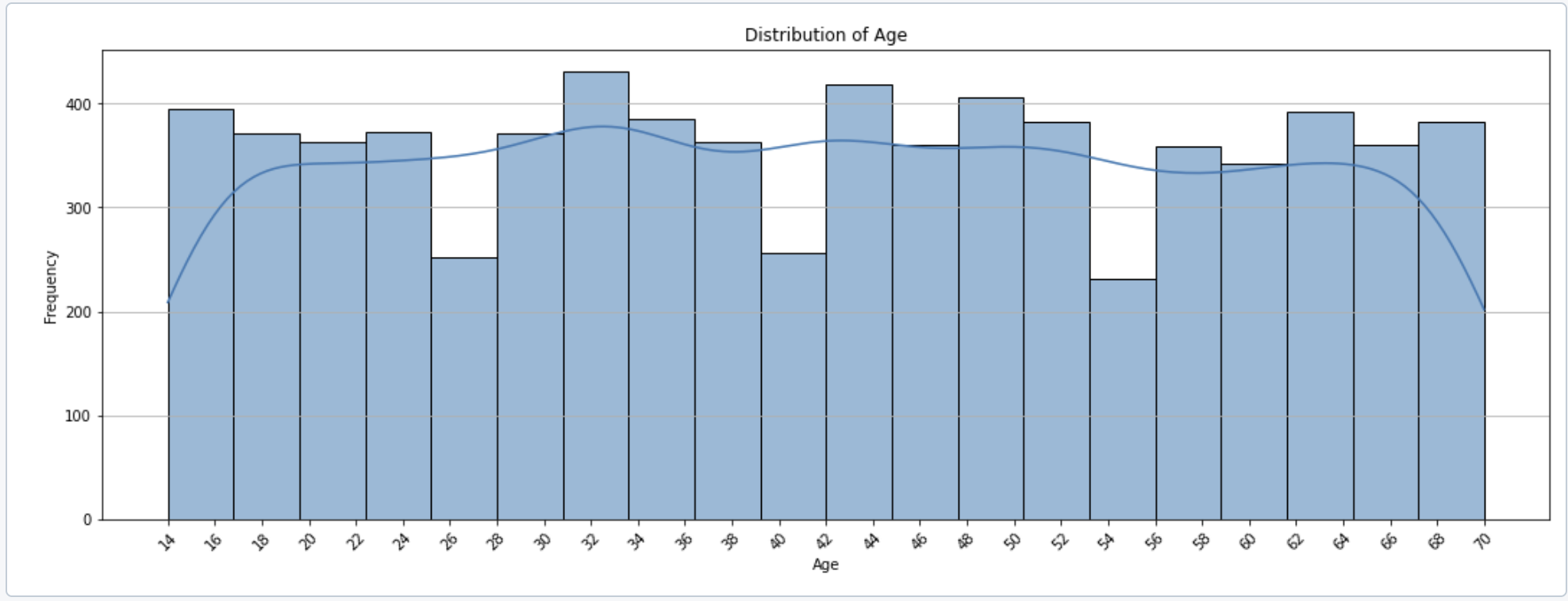
****

Fig-2.2

**Source code**

# Histogram of Age

plt.figure(figsize=(18, 6))

sns.histplot(pdf['Age'], bins=20, kde=True, palette='muted')

# Title and labels

plt.title('Distribution of Age')

plt.xlabel('Age')

plt.ylabel('Frequency')

# Set x-ticks to show even numbers from 18 to the maximum age in the data

max\_age = pdf['Age'].max()

plt.xticks(ticks=range(14, max\_age + 2, 2), rotation=45)

plt.grid(axis='y')

plt.show()

**Explanation**

The figure 2.1 and 2.2, the Play Store age distribution chart also indicates the number of users at a given time set, which is proportional to the age of the users in a fairly uniform manner. Ages 15 through 19 and 30 through 39 have the highest frequencies of 300 and 400 users respectively. This implies that the audience of the application is fairly well tempered in terms of demographics and no one party can avoid all the content in the application or makes extreme contributions to its content. Nevertheless, people below 20 and 30-39 year olds seem to be the most likely users of Play Store. On the other hand, age distribution of App Store users seems to have similar age group centers in the middle and older age brackets, but much less even spread of members age group distribution. It is observed that, older age groups register around their late twenties and some thirties, while younger demographics range around eighteen simply don’t seem so engaged with the App Store. This means that the App Store is not only for older people, but information in it seems to be preferred with more twenty and thirty year olds rather than sixty build App Store which is ideal language the audience isn’t targeted. Lower registrations from younger and older demographics turns out that information and content in the App Store appears to be much more engaging for adults simply because apps are based around adulthood life situations or their early career orientation.

**Real-time usage**

Marketers can design age-targeted advertising campaigns by focusing on specific age groups. For instance, if an app is aimed at younger users, more resources could be allocated to Play Store ads, targeting teens and young adults. On the other hand, for apps that cater to professionals or working adults, App Store campaigns can be geared toward individuals in their 20s and 30s.

Advertisers can craft relevant ad content for different age groups. For example, Play Store ads might feature more vibrant, interactive, or playful content to resonate with younger users, while App Store ads might highlight productivity tools or financial management apps for older, professional audiences.

**3- Avg user rating by genres**

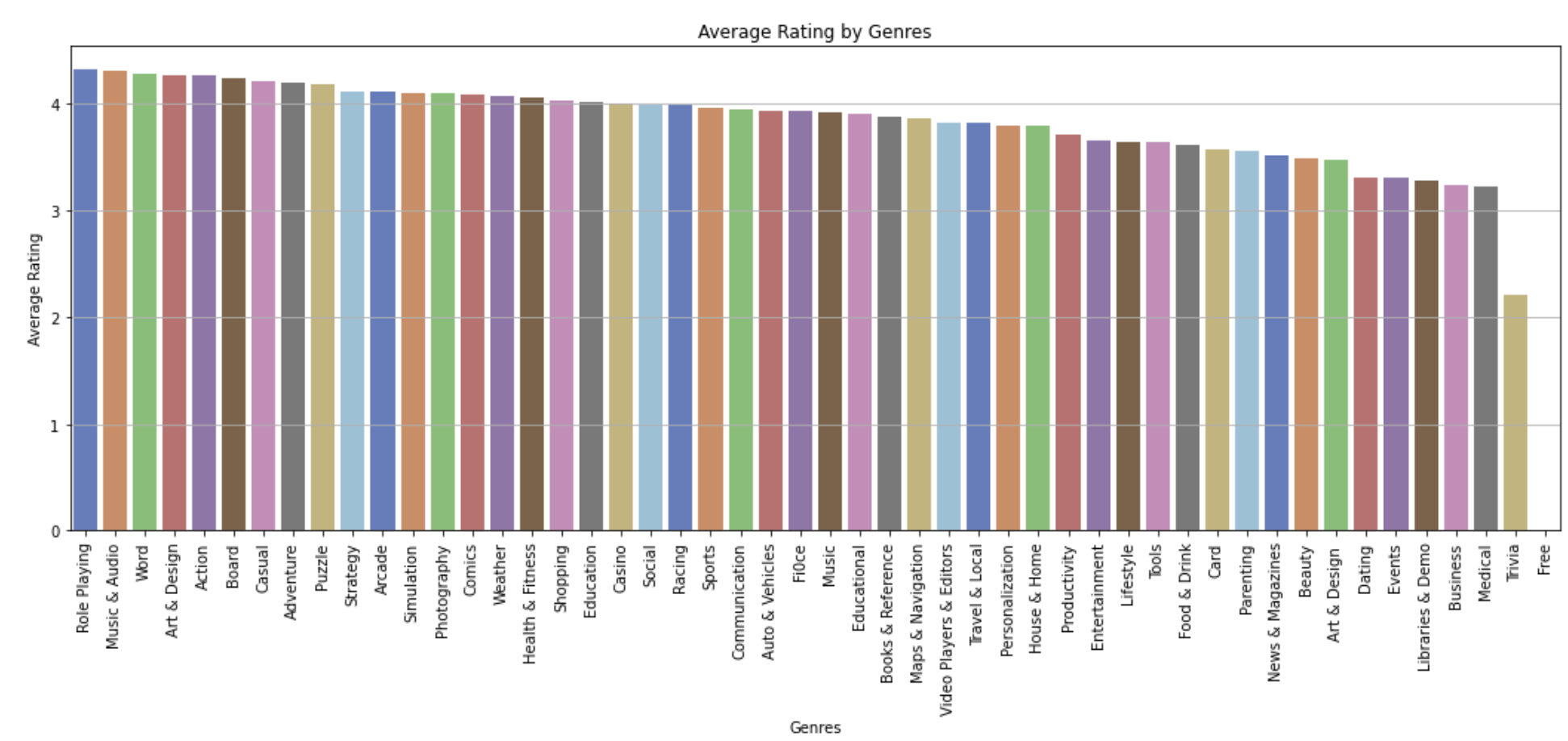
****

Fig-3.1

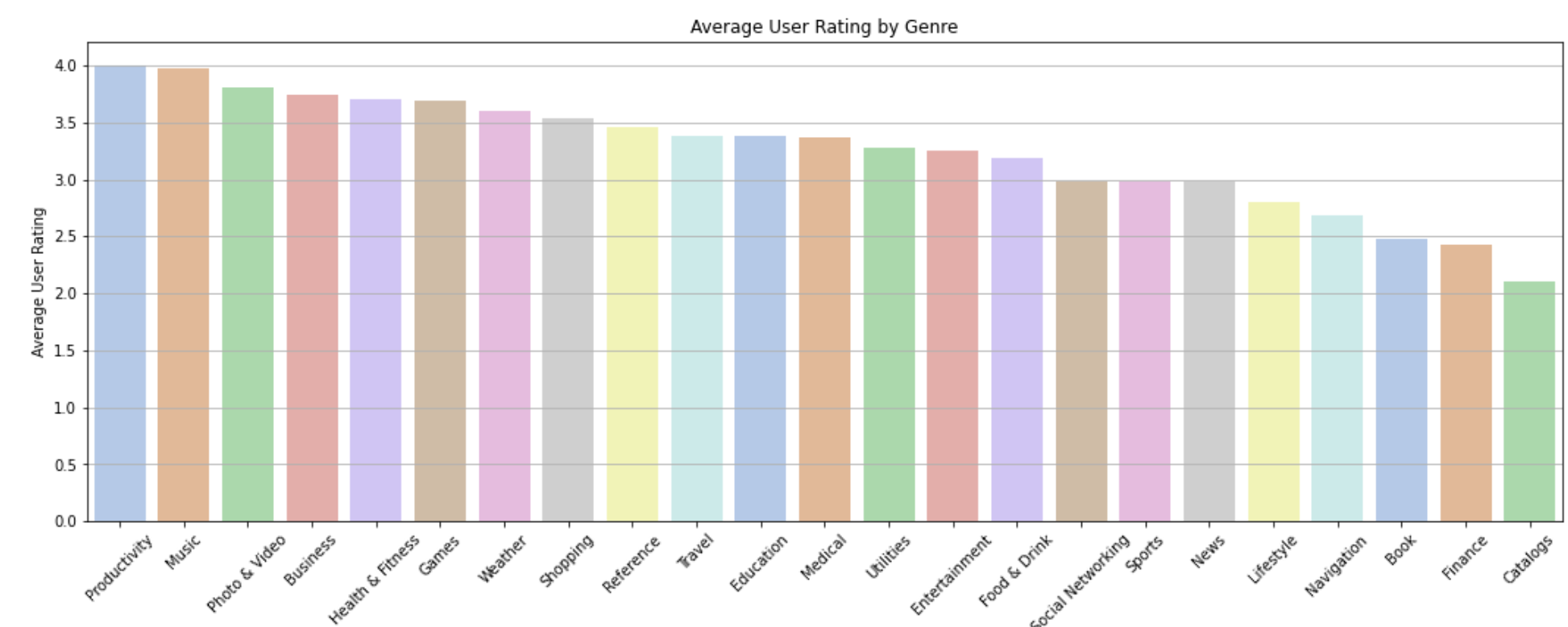
****

Fig-3.2

**Source code**

# Calculate the average user rating by genre

average\_ratings = pdf.groupby('Genres')['Rating'].mean().reset\_index()

average\_ratings.columns = ['Genres', 'Average Rating']

# Sort the average ratings in descending order

average\_ratings = average\_ratings.sort\_values(by='Average Rating', ascending=False)

# Visualization

plt.figure(figsize=(18, 6))

sns.barplot(x='Genres', y='Average Rating', data=average\_ratings, palette='muted')

# Title and labels

plt.title('Average Rating by Genres')

plt.xlabel('Genres')

plt.ylabel('Average Rating')

plt.xticks(rotation=90)

plt.grid(axis='y')

plt.show()

**Explanation**

The first graph 3.1 illustrates the average user ratings for different app genres on the Play Store, with genres sorted in descending order of their average ratings. Categories like Role Playing, Music & Audio, and Word emerge as the highest-rated genres, indicating that users are generally more satisfied with apps in these areas. This information is valuable for app developers and marketers as it highlights genres that tend to receive higher user satisfaction. By focusing on genres such as Role Playing or Music & Audio, developers can prioritize these areas for future app development or feature enhancement to meet user expectations and improve ratings. The second graph 3.2 displays the average user ratings for various app genres on the Apple Store. Similar to the Play Store data, this helps identify which genres are most appreciated by users. Categories like Productivity, Music, and Photo & Video receive the highest ratings, reflecting their strong user engagement and satisfaction. For app developers, focusing on enhancing features or developing apps in these popular categories could yield positive user feedback and higher ratings. By leveraging these insights, developers can make data-driven decisions to align app development with user preferences across both platforms.

**Real-time usage**

App developers can focus on genres with higher average ratings when planning new applications or updates. For example, the high ratings in Music & Audio suggest strong user engagement, so adding new features in this space could lead to higher downloads and ratings.

Marketers can use insights to tailor their campaigns to focus on popular genres. For instance, promoting apps in the Role Playing or Productivity categories may result in better user engagement and returns on investment, since users tend to rate these apps higher.

**4- Descriptive Statistics**

**App Store (numerical columns)**

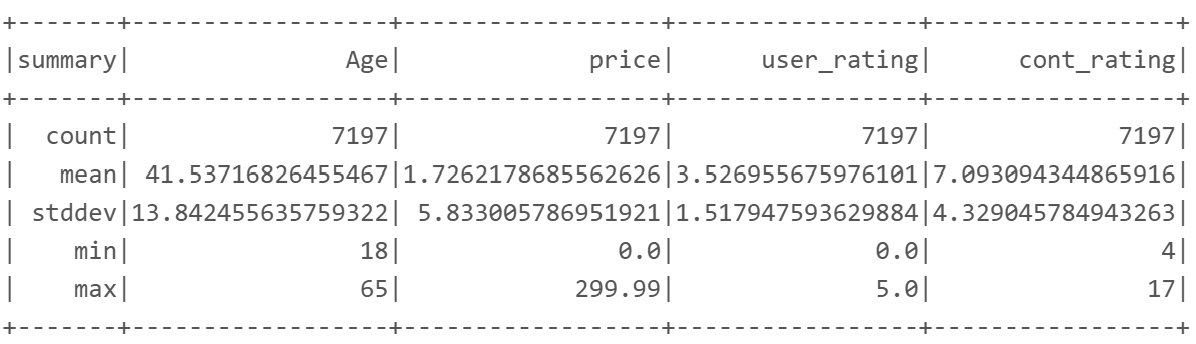


Fig- 4.1

**Source code**

# Descriptive statistics for numeric columns

numeric\_columns = ['Age', 'price', 'user\_rating', 'cont\_rating']

numeric\_stats = df.select(numeric\_columns).describe()

numeric\_stats.show()

**Explanation**

Figure 4.1 indicates that the average age of app users is 41.54 years, with ages ranging from 18 to 65. The standard deviation of 13.84 suggests there is quite a diverse age group using these apps. Additionally, the average price for paid apps is $1.73, but with a high standard deviation of 5.83, this points to a significant variation in app pricing, including a maximum price of $299.99. User satisfaction shows an average rating of 3.53 out of 5. Furthermore, the average content rating is 7.09, indicating most of the apps are made for people over 7 years.

**Real-time usage**

These statistics give a clearer picture of who our users are and how much they’re willing to pay for the app. This information helps app developers and marketers to make informed choices.

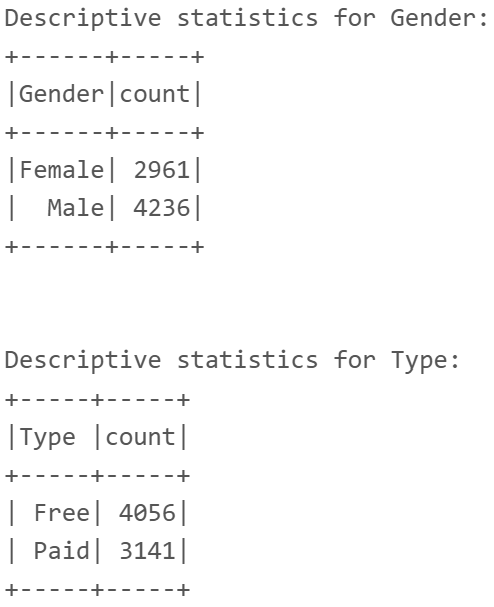
**App Store (Categorical column)**

Fig- 4.2

**Source code**

# Descriptive statistics for categorical columns

categorical\_columns = ['Gender', 'Type ', 'track\_name', 'prime\_genre']

# Calculate and display the descriptive statistics

for column in categorical\_columns:

print(f"\nDescriptive statistics for {column.strip()}:")

df.groupBy(column).agg(F.count("\*").alias("count")).show()

**Explanation**

In Fig- 4.2 there are 2,961 female and 4,236 male users. This indicates a higher proportion of male users in the dataset. There are 4,056 free apps and 3,141 paid apps, showing that more free apps are available than paid ones.

**Real-time usage**

This data helps understand user demographics' distribution and the types of apps offered, which can inform app development and marketing strategies based on user preferences and app monetization models.

**Play Store (numerical columns)**

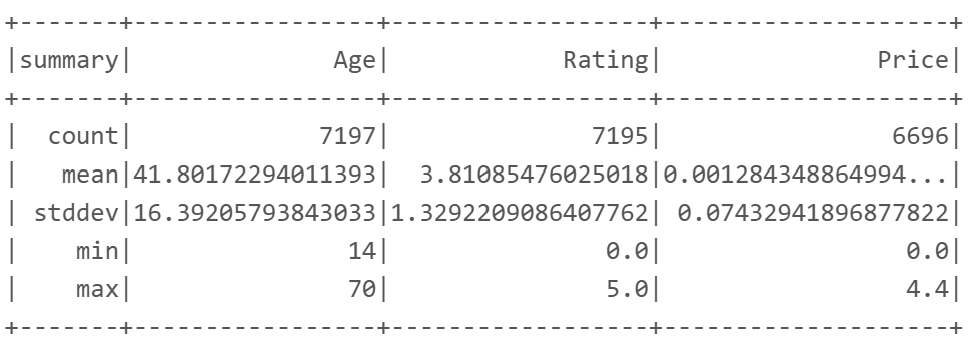
****

Fig- 4.3

**Source code**

# Descriptive statistics for numeric columns

numeric\_columns = ['Age', 'Rating', 'Price',]

numeric\_stats = df.select(numeric\_columns).describe()

numeric\_stats.show()

**Explanation**

Fig- 5.1 displays the average user age is 41.80 years, with 14 being the lowest and 70 being the highest. The high standard deviation of 16.39 indicates a widespread of the data. The average app rating is 3.81, with a minimum rating of 0 and a maximum of 5. The standard deviation of 1.33 indicates moderate variation in-app ratings. It also shows that the average price for paid apps is very low, which is because the maximum of the apps is free. The maximum price is $4.4, indicating relatively affordable pricing for the Play Store application.

**Real-time usage**

The variety in ratings and ages indicates that user experiences differ greatly, highlighting opportunities to improve apps that aren't performing well and tailor offerings to better meet the needs of various age groups.

**Play Store (categorical columns)**

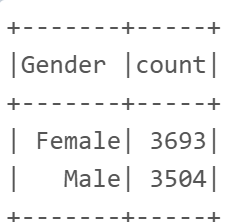
****

Fig- 4.4

**Source code**

# Descriptive statistics for categorical columns

# Count unique values for categorical columns

gender\_counts = df.groupBy("Gender ").count()

app\_counts = df.groupBy("App").count()

type\_counts = df.groupBy("Type").count()

content\_rating\_counts = df.groupBy("Content Rating").count()

genre\_counts = df.groupBy("Genres").count()

**Explanation**

There are 3,693 female users and 3,504 male users in the dataset, showing an even gender distribution.

**Real-time usage**

The above distribution can help developers and marketers tailor their strategies to better engage with both the demographics.

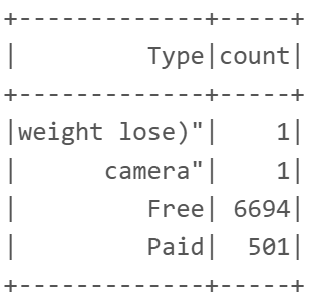
****

Fig- 4.5

**Source code**

# Show the counts for categorical columns

gender\_counts.show()

**Explanation**

Fig- 4.5 shows the count of apps categorized as "Free" with 6,694 count and "Paid” with 501 counts.

**Real-time usage**

This diagram gives us a clear picture of how free apps are so popular. Paid apps on the other hand are present but in small proportion. It can be inferred that developers are turning to other ways to earn, such as in-app purchases or ads.

**Machine Learning model on Play Store**

**1- K-Mean**

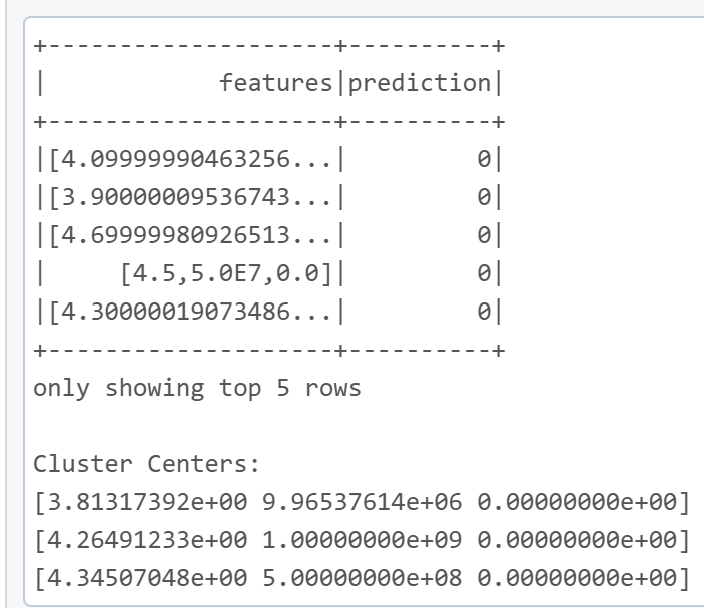
****

Fig- 5.1

**Source code**

from pyspark.ml.clustering import KMeans

from pyspark.ml.feature import VectorAssembler

# Check if 'features' column exists and remove it if it does

if 'features' in data.columns:

data = data.drop('features')

# Assemble features for KMeans clustering

assembler = VectorAssembler(inputCols=["Rating", "Installs", "Price"], outputCol="features")

data = assembler.transform(data)

# Train KMeans Model with 3 clusters

kmeans = KMeans(featuresCol="features", k=3)

model = kmeans.fit(data)

# Make predictions and show clusters

predictions\_cluster = model.transform(data)

predictions\_cluster.select("features", "prediction").show(5)

# Display cluster centers

print("Cluster Centers:")

for center in model.clusterCenters():

print(center)

**Explanation**

Figure 5.1 illustrates the outcomes of K-means clustering, which involved multiple cluster centers. K-means was employed to group applications according to features like ratings, installations, and possibly pricing. All data points were classified as part of cluster 0, suggesting that the algorithm assigned them to a single group. This could imply that the selected features lacked sufficient variability for effective clustering, or that the parameters chosen require refinement.

**Real-time usage**

Clustering can help the Play Store team understand segments of apps, as better-defined clusters can guide targeted marketing or feature development for specific app categories.

**2- Logistic Regression**

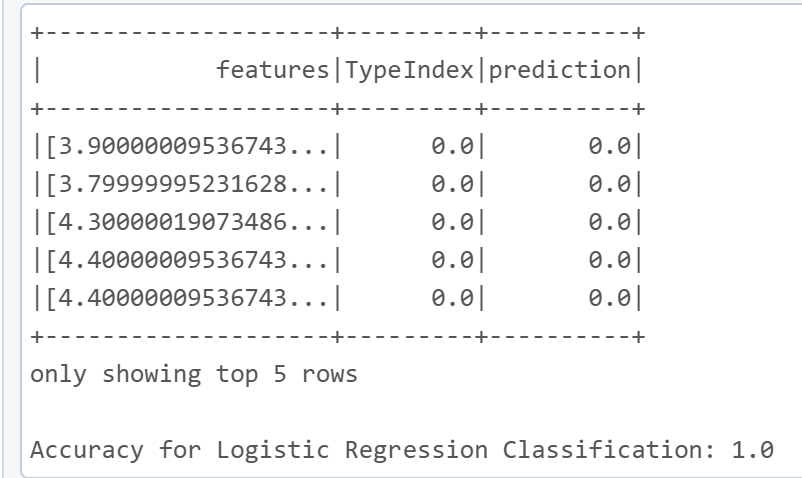
****

Fig- 5.2

**Source code**

from pyspark.ml.classification import LogisticRegression

from pyspark.ml.feature import VectorAssembler

from pyspark.ml.evaluation import MulticlassClassificationEvaluator

# Check if 'features' column exists and remove it if it does

if 'features' in data.columns:

data = data.drop('features')

# Assemble features for logistic regression to predict 'TypeIndex'

assembler = VectorAssembler(inputCols=["Rating", "Installs", "Price"], outputCol="features")

data = assembler.transform(data)

# Split the data into training and testing sets

train\_data, test\_data = data.randomSplit([0.8, 0.2], seed=1234)

# Train Logistic Regression Model

lr\_class = LogisticRegression(featuresCol="features", labelCol="TypeIndex")

lr\_class\_model = lr\_class.fit(train\_data)

# Make predictions on the test data

predictions\_class = lr\_class\_model.transform(test\_data)

predictions\_class.select("features", "TypeIndex", "prediction").show(5)

# Evaluate the model accuracy

evaluator\_class = MulticlassClassificationEvaluator(labelCol="TypeIndex", predictionCol="prediction", metricName="accuracy")

accuracy = evaluator\_class.evaluate(predictions\_class)

print(f"Accuracy for Logistic Regression Classification: {accuracy}")

**Explanation**

The logistic regression score in Figure 5.2 is 1.0, indicating that the model has attained a flawless score. All entries, though, are categorized as 0. This can suggest that the classifier was trained on a little amount of data or that it is unbalanced. This situation arrived because of overfitting or a dataset with only one class.

**Real-time usage**

The Play Store team can help users discover new applications more easily, by classifying different types of apps accurately. They can offer recommendations that truly match what users are interested in. By automatically sorting new apps based on ratings and user preferences, customers can find exactly what they want without any hassle.

**3- Linear Regression**

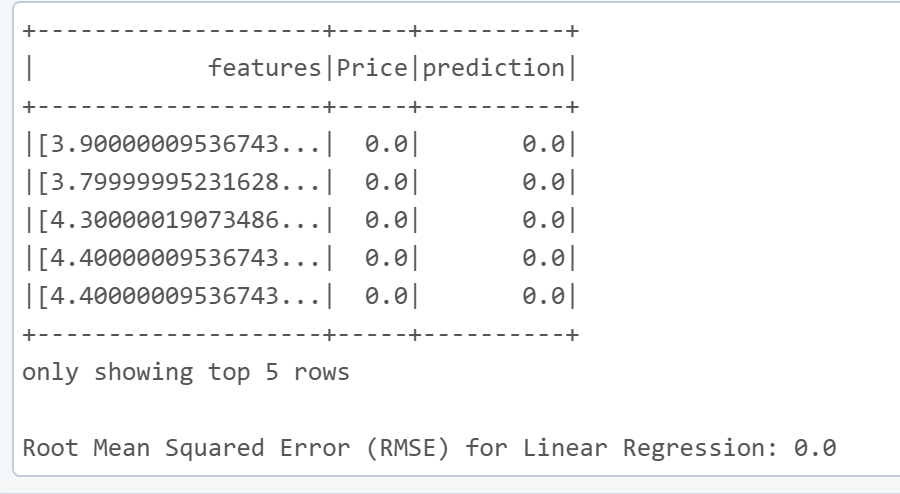
****

Fig- 5.3

**Source code**

# Split data into training and testing sets

train\_data, test\_data = data.randomSplit([0.8, 0.2], seed=1234)

# Linear Regression Model

lr = LinearRegression(featuresCol="features", labelCol="Price")

lr\_model = lr.fit(train\_data)

# Make predictions and evaluate

predictions = lr\_model.transform(test\_data)

predictions.select("features", "Price", "prediction").show(5)

# Evaluate the model using RMSE

evaluator = RegressionEvaluator(labelCol="Price", predictionCol="prediction", metricName="rmse")

rmse = evaluator.evaluate(predictions)

print(f"Root Mean Squared Error (RMSE) for Linear Regression: {rmse}")

**Explanation**

Figure 5.3 illustrates how a linear regression model predicts app prices. The RMSE (Root Mean Squared Error) is 0, indicating that all predictions matched the actual values perfectly. This could indicate that all of the apps in the dataset were free (price 0), making it simple for the model to predict.

**Real-time usage**

Predicting app prices can help the Play Store team recommend the best prices for new applications. However, because most apps are free, you may need to use additional features or a different subset of data to get useful results.

**Machine Learning model on App Store**

**1- K-Mean**

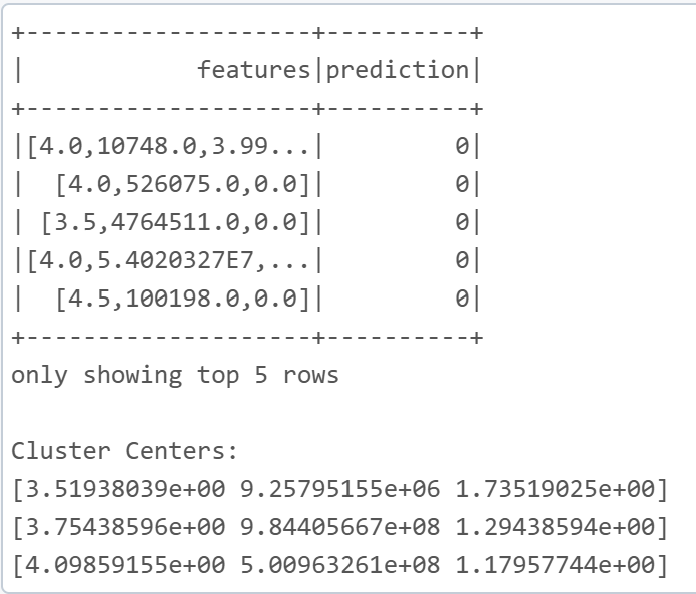
****

Fig- 6.1

**Source code**

from pyspark.ml.clustering import KMeans

from pyspark.ml.feature import VectorAssembler

# Check if 'features' column exists and remove it if it does

if 'features' in data.columns:

data = data.drop('features')

# Assemble features for KMeans clustering

assembler = VectorAssembler(inputCols=["Rating", "Installs", "Price"], outputCol="features")

data = assembler.transform(data)

# Train KMeans Model with 3 clusters

kmeans = KMeans(featuresCol="features", k=3)

model = kmeans.fit(data)

# Make predictions and show clusters

predictions\_cluster = model.transform(data)

predictions\_cluster.select("features", "prediction").show(5)

# Display cluster centers

print("Cluster Centers:")

for center in model.clusterCenters():

print(center)

**Explanation**

In Fig. 6.1, every entry seems to be projected to belong to cluster 0, indicating that either the clustering parameters need to be adjusted for meaningful grouping, or the majority of the data points share similar qualities.

**Real-time usage**

Grouping apps into categories improves user recommendations and marketing efforts. For example, clustering can reveal highly rated apps that aren’t frequently installed, uncovering hidden gems. By identifying these clusters, the team can promote these apps, helping users find new favorites they might have overlooked.

**2- Logistic Regression**

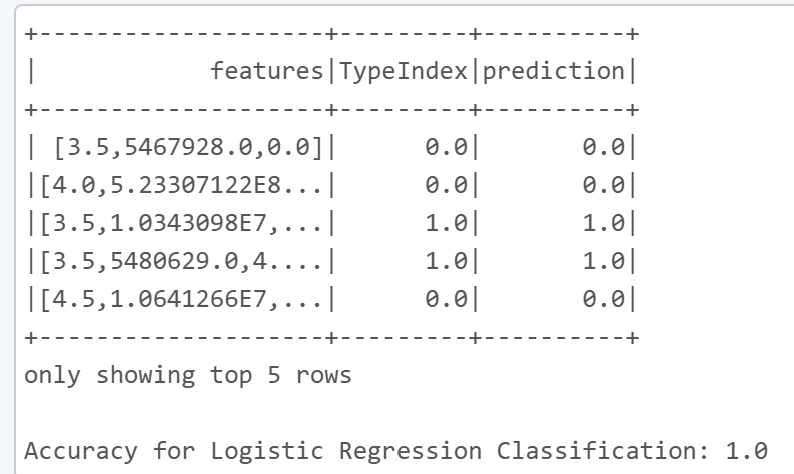
****

Fig- 6.2

**Source code**

from pyspark.ml.classification import LogisticRegression

from pyspark.ml.feature import VectorAssembler

from pyspark.ml.evaluation import MulticlassClassificationEvaluator

# Check if 'features' column exists and remove it if it does

if 'features' in data.columns:

data = data.drop('features')

# Assemble features for logistic regression to predict 'TypeIndex'

assembler = VectorAssembler(inputCols=["Rating", "Installs", "Price"], outputCol="features")

data = assembler.transform(data)

# Split the data into training and testing sets

train\_data, test\_data = data.randomSplit([0.8, 0.2], seed=1234)

# Train Logistic Regression Model

lr\_class = LogisticRegression(featuresCol="features", labelCol="TypeIndex")

lr\_class\_model = lr\_class.fit(train\_data)

# Make predictions on the test data

predictions\_class = lr\_class\_model.transform(test\_data)

predictions\_class.select("features", "TypeIndex", "prediction").show(5)

# Evaluate the model accuracy

evaluator\_class = MulticlassClassificationEvaluator(labelCol="TypeIndex", predictionCol="prediction", metricName="accuracy")

accuracy = evaluator\_class.evaluate(predictions\_class)

print(f"Accuracy for Logistic Regression Classification: {accuracy}")

**Explanation**

Figure 6.2 demonstrates that the logistic regression model attained a perfect accuracy score of 1.0, indicating flawless classification. This model can categorize apps as free or paid, or based on attributes like content ratings. However, this perfect accuracy may indicate overfitting, a lack of diversity in the classes, or that the classification task is relatively simple.

**Real-time usage**

Logistic regression can be a valuable tool for predicting app classifications, which helps make app recommendations more effective. The Play Store team could use this model to automatically sort new apps, making it easier for users to find what they’re looking for and boosting their overall engagement.

**3- Linear Regression**

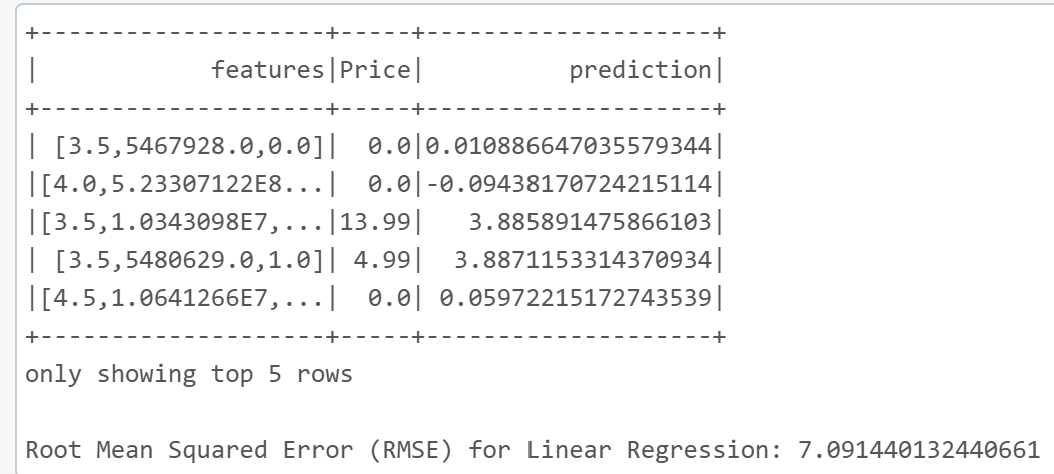
****

Fig- 6.3

**Source code-**

from pyspark.ml.feature import VectorAssembler

from pyspark.ml.regression import LinearRegression

# Assemble features for regression model

assembler = VectorAssembler(inputCols=["Rating", "Installs", "TypeIndex"], outputCol="features")

data = assembler.transform(data)

# Split data

train\_data, test\_data = data.randomSplit([0.8, 0.2], seed=1234)

# Linear Regression Model

lr = LinearRegression(featuresCol="features", labelCol="Price")

lr\_model = lr.fit(train\_data)

# Make predictions and evaluate

predictions = lr\_model.transform(test\_data)

predictions.select("features", "Price", "prediction").show(5)

from pyspark.ml.evaluation import RegressionEvaluator

evaluator = RegressionEvaluator(labelCol="Price", predictionCol="prediction", metricName="rmse")

rmse = evaluator.evaluate(predictions)

print(f"Root Mean Squared Error (RMSE) for Linear Regression: {rmse}")

**Explanation**

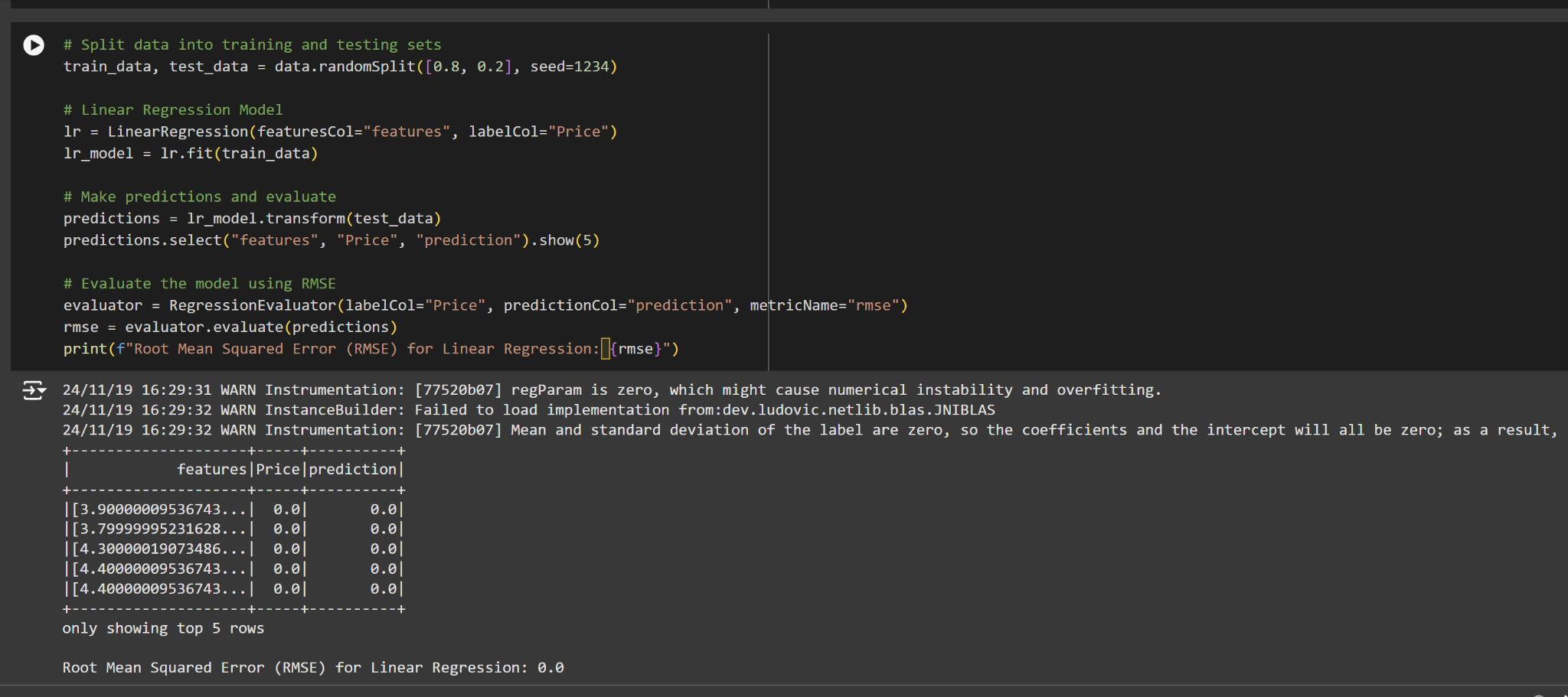
The linear regression model illustrated in Figure 6.3 demonstrates an RMSE (Root Mean Squared Error) of approximately 7.09, reflecting a reasonable level of accuracy in forecasting app prices. This model considers variables such as ratings and installation counts; however, discrepancies between the predicted and actual prices remain. This indicates that incorporating additional features or employing more sophisticated modeling techniques could enhance the accuracy of the predictions.

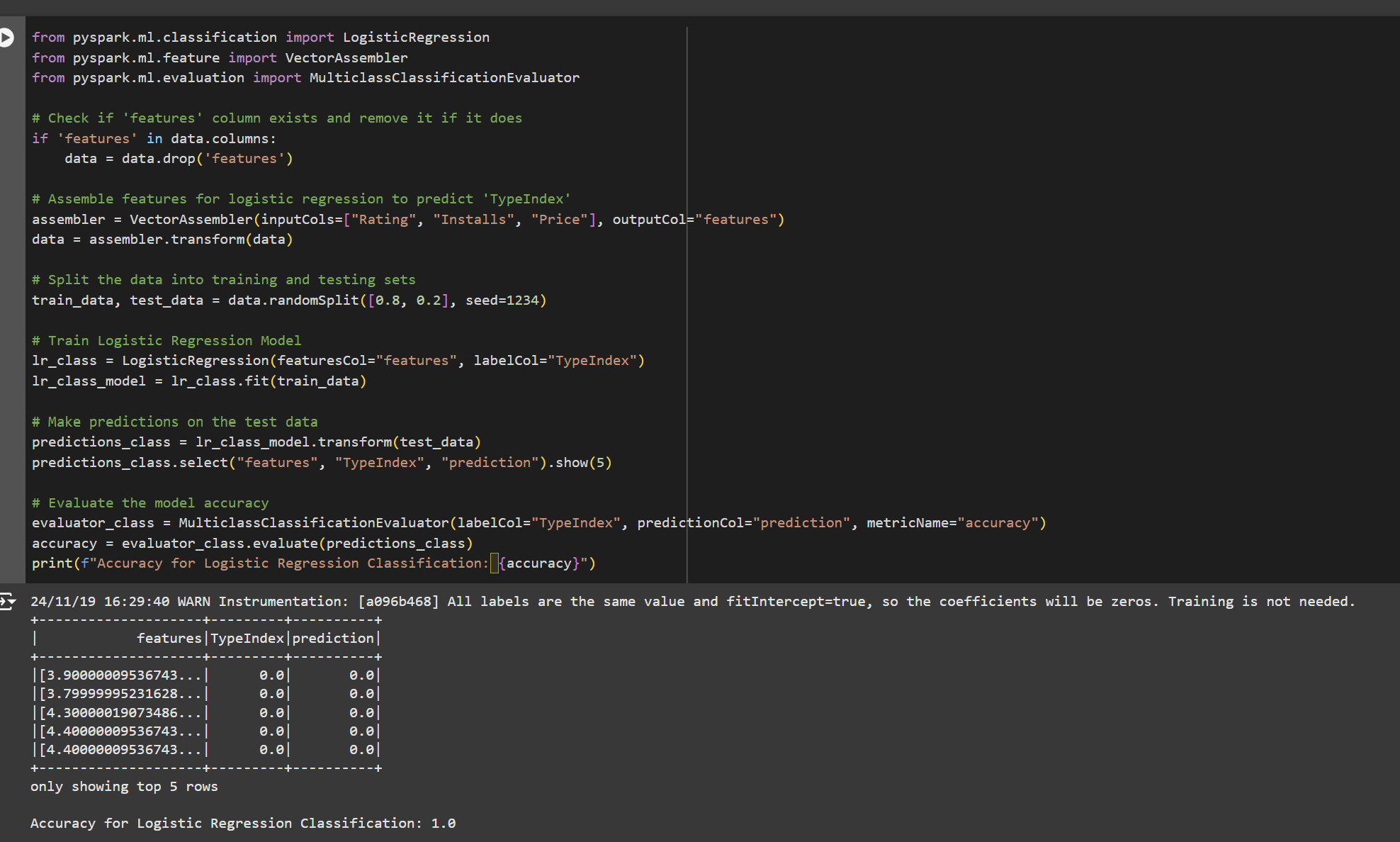
**Real-time usage**

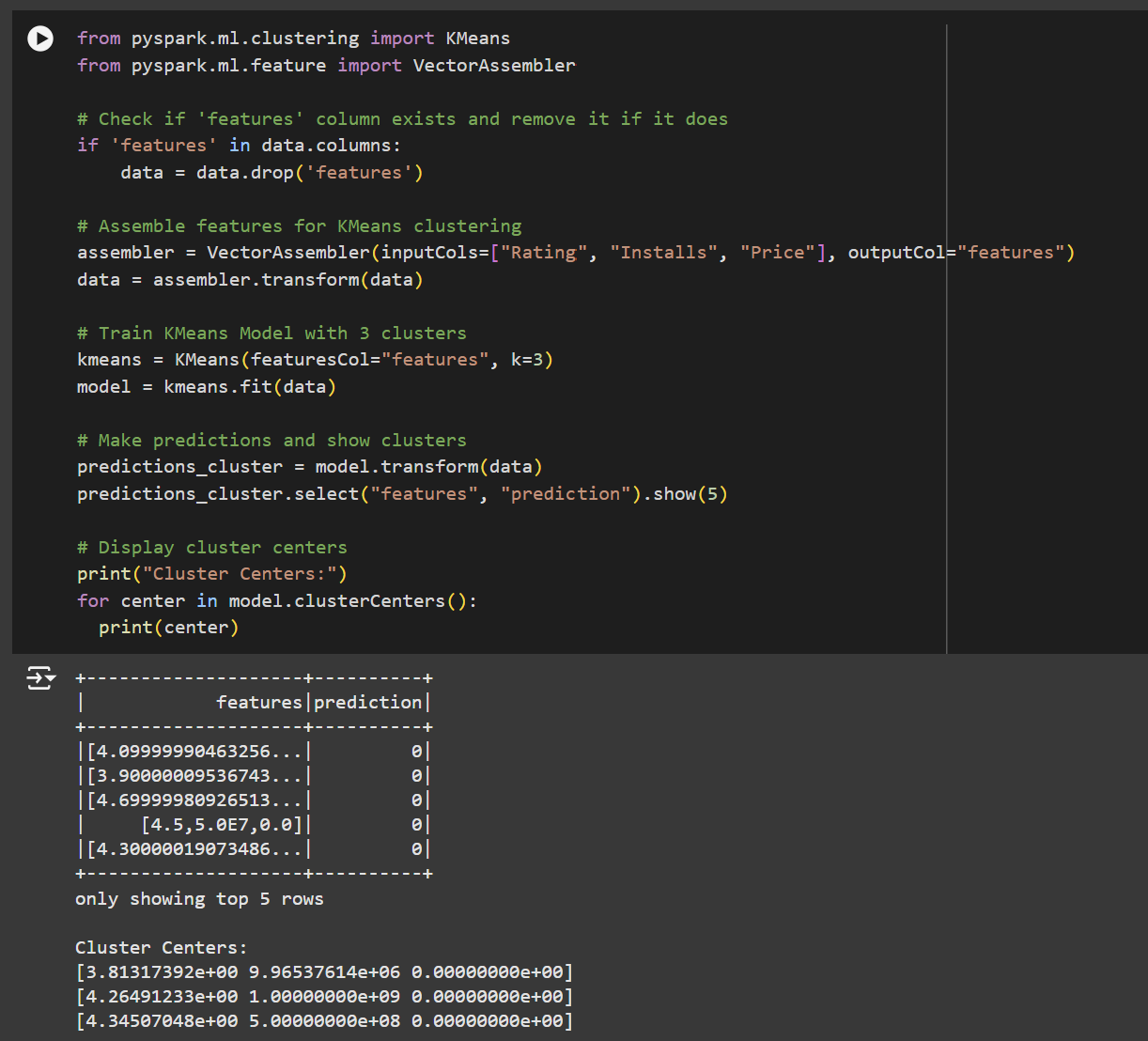
From a practical standpoint, predicting app prices enables the team to propose effective pricing strategies for newly launched applications. By analyzing trends in user reviews, installation figures, and pricing, developers can establish competitive pricing that may result in increased downloads and enhanced revenue.

**5- Machine Learning model on Jupyter Notebook**

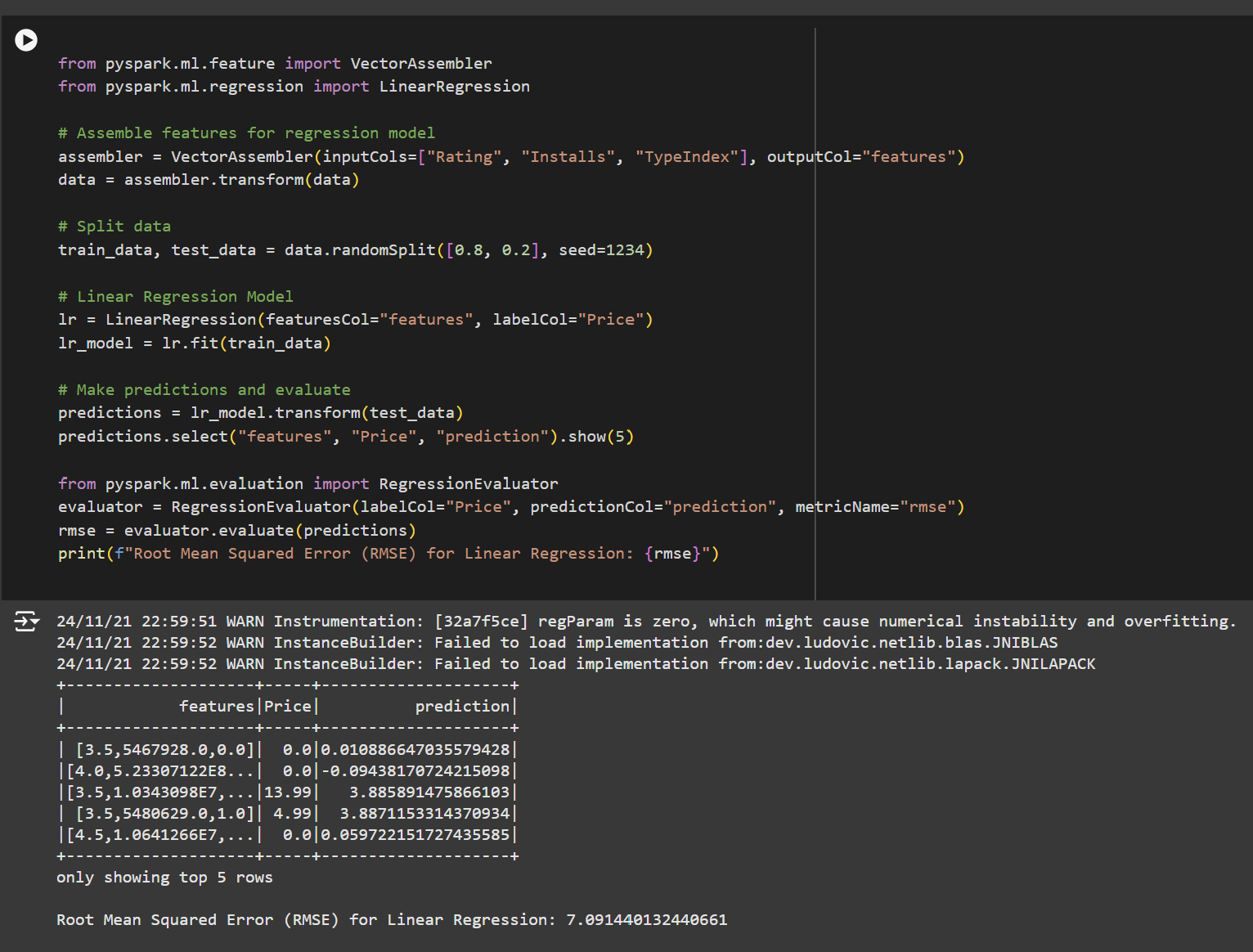
**Play Store**

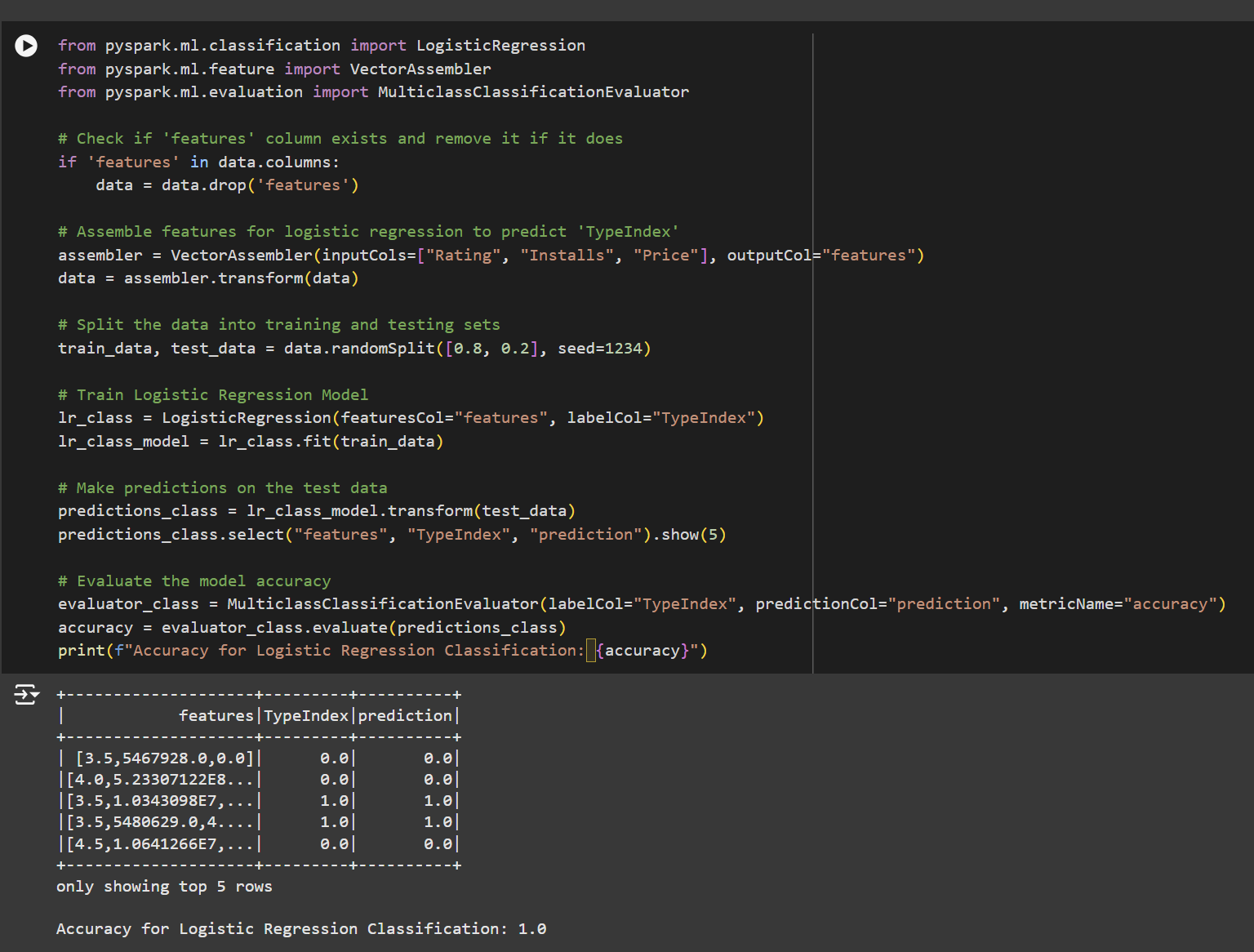


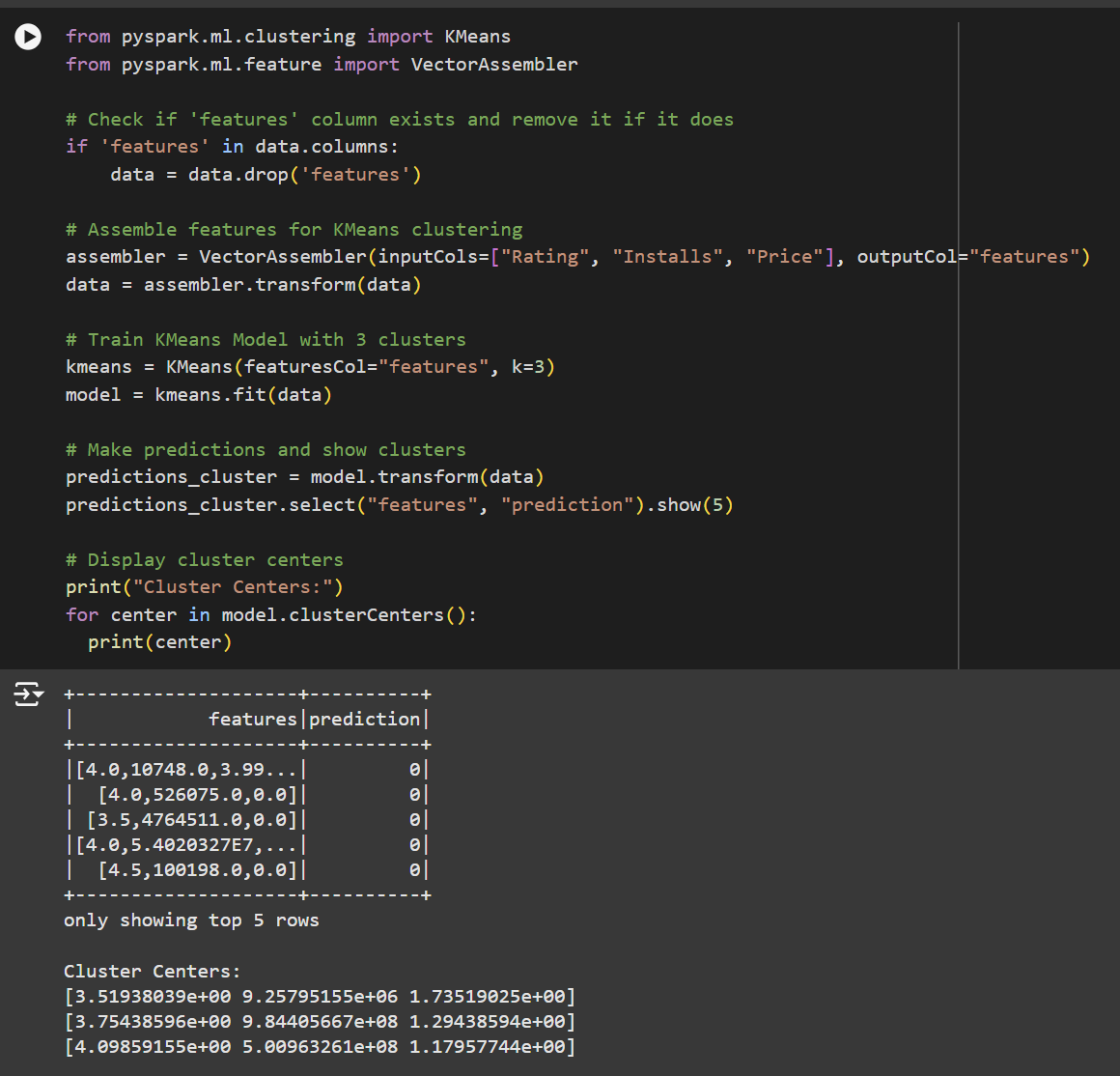




**App Store**







**Visualization Using Tableau**

**For App Store**

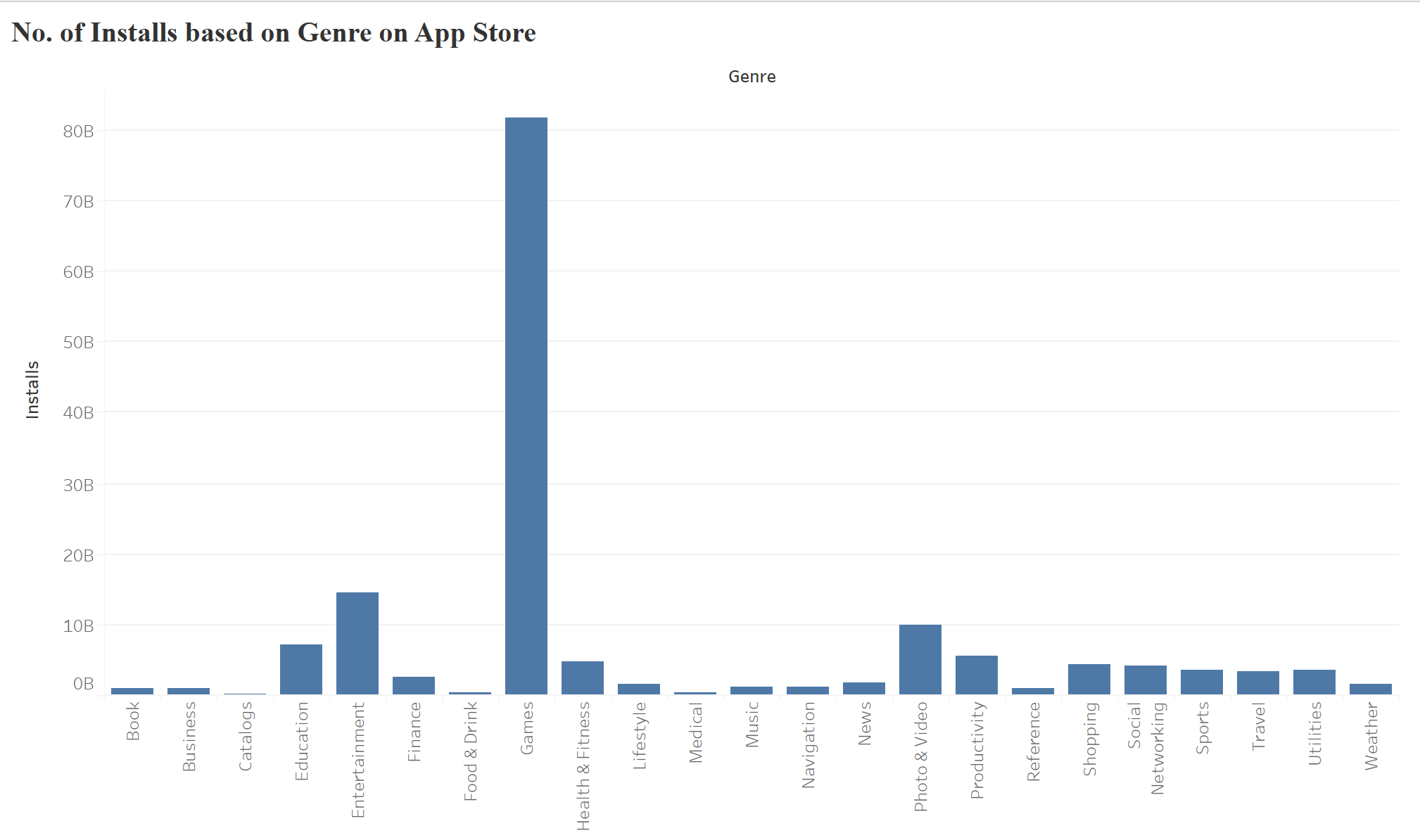


Fig- 7.1

This chart (Fig 7.1) shows installations by genre for the App Store. "Games" are by far the most downloaded category, reflecting the strong preference for gaming among iOS users. Other popular genres include "Entertainment" and "Photo & Video," which cater to iOS users’ focus on leisure and creativity. Genres like "Finance" and "Catalogs" have fewer installs, indicating more specialized use cases for these apps.

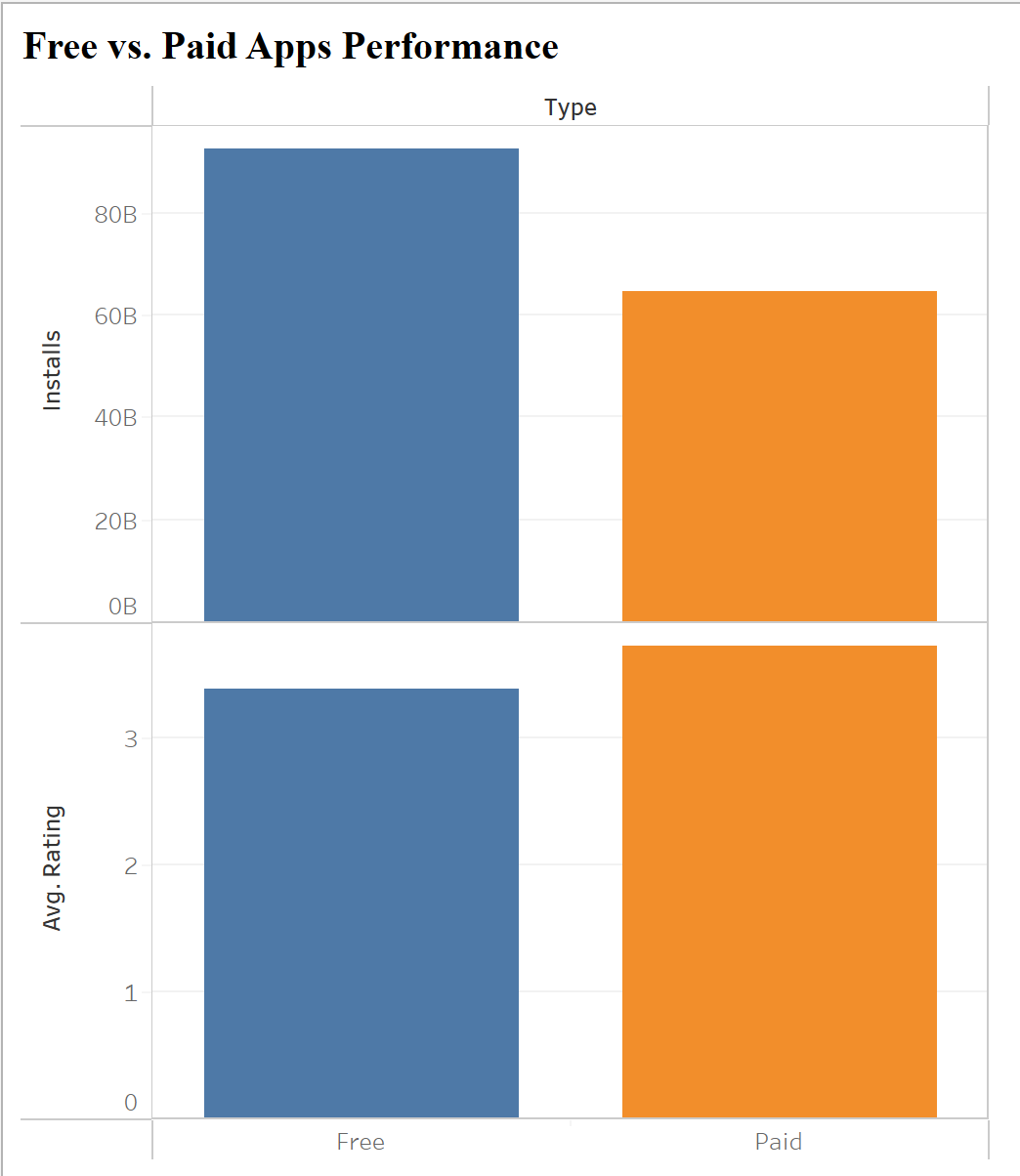


Fig- 7.2

This visualization (Fig- 7.2), akin to Graph 1, examines the correlation between the number of installs and the average user ratings for both free and paid applications. Free applications draw a more extensive user base; however, they generally receive marginally lower ratings, which may be attributed to the diverse expectations of users. In contrast, paid applications, while having a more restricted audience, tend to garner higher ratings, as users are more discerning and anticipate superior quality.

**For Play Store (Android)**

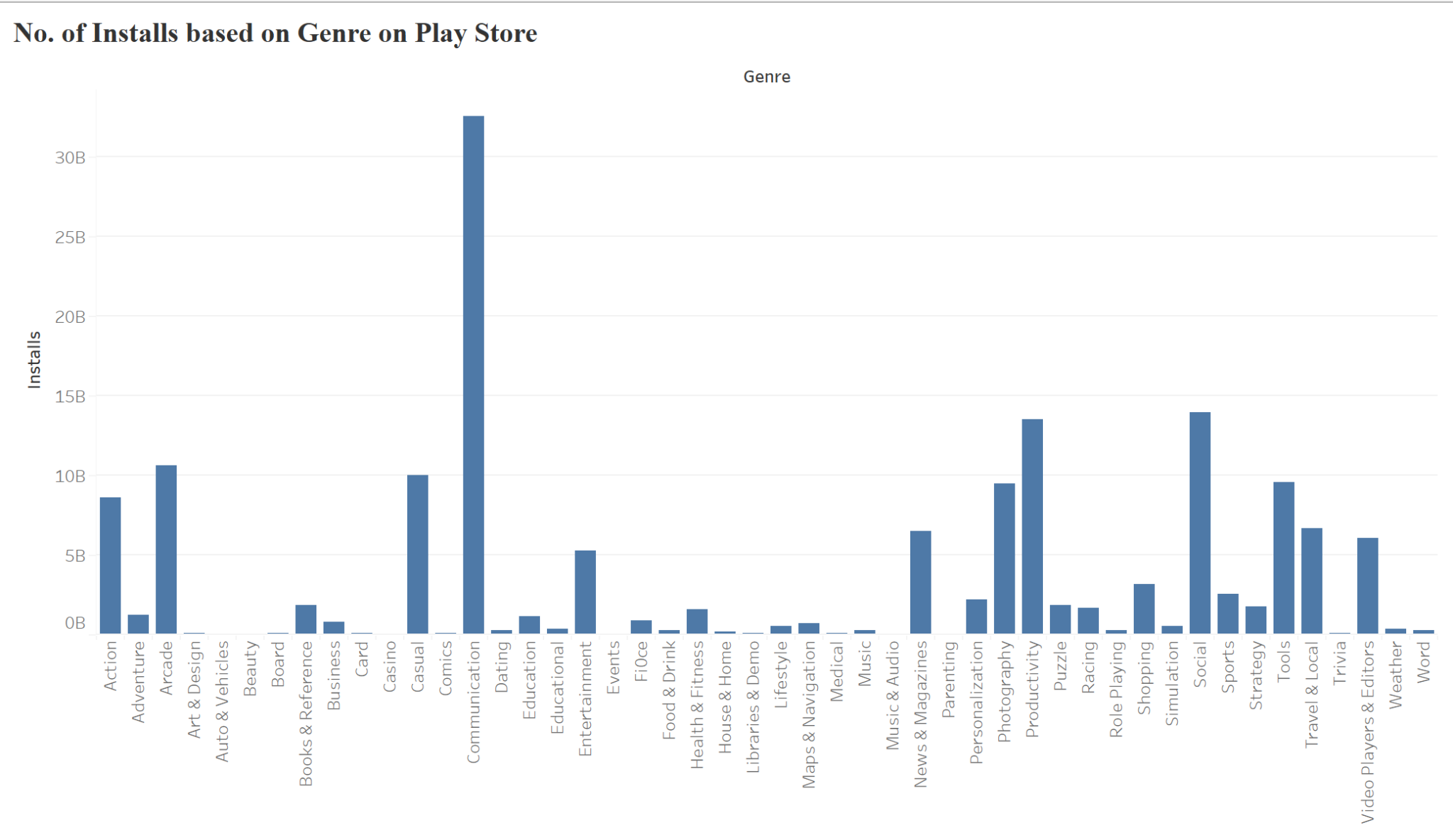


Fig- 7.3

Figure 7.3 shows the number of installations for different app genres on the Play Store. The top categories, "Communication" and "Social," highlight how popular apps like WhatsApp, Facebook, and Instagram are. At the same time, "Tools" and "Entertainment" also draw in users, demonstrating a mix of interests in both productivity and fun. In contrast, the "Books & Reference" and "Dating" categories see fewer installs, suggesting they cater to more specialized audiences.

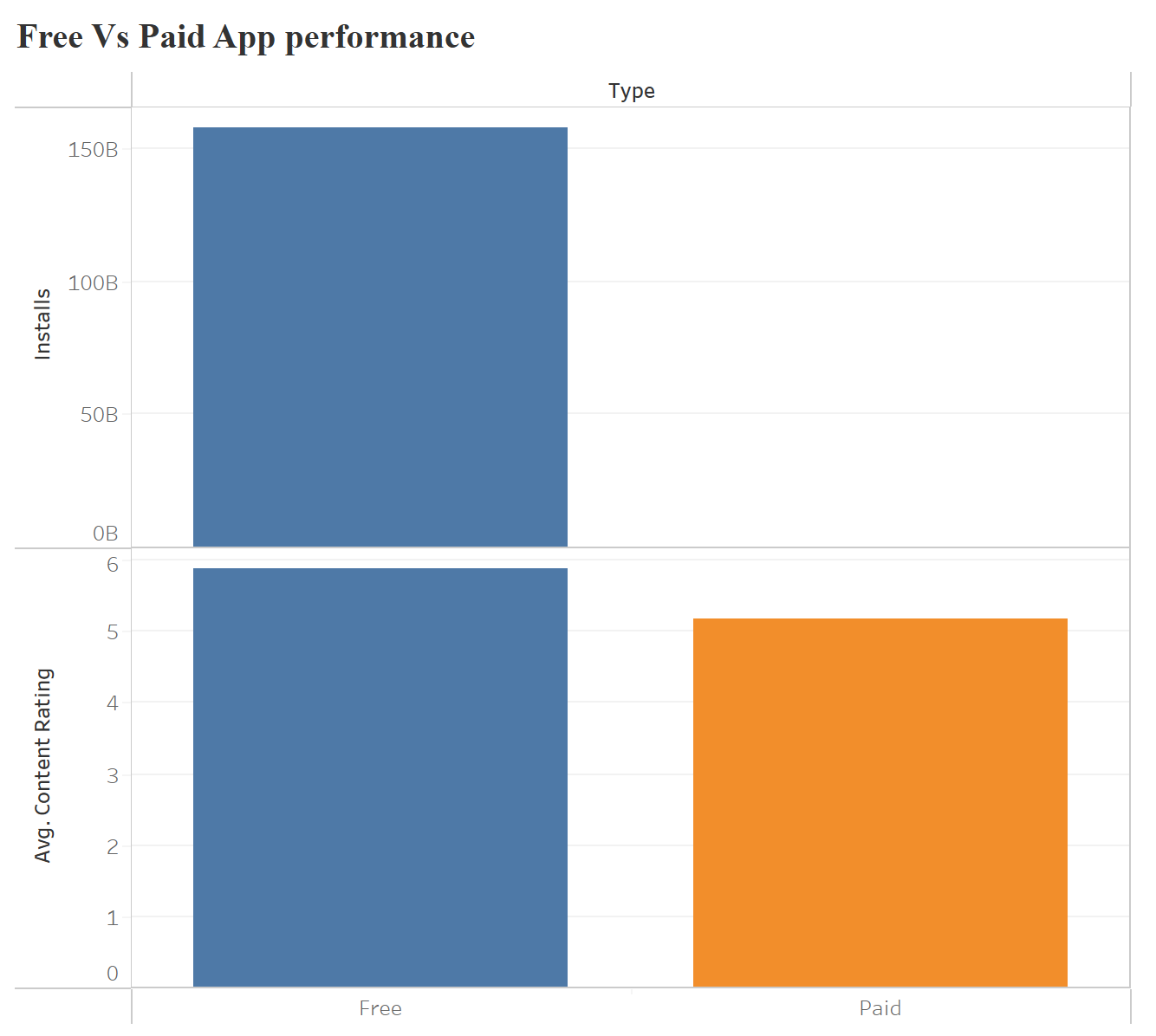


Fig- 7.4

Figure 7.4 compares total installations of free and paid apps. Free apps significantly outnumber paid ones in installations, which is expected since they are available at no cost. The chart also shows that paid apps generally receive slightly higher average ratings. This may be because the users think before actually making a decision to purchase an app.

**Addressing Research Questions**

**Q1-** Compare apps on both platforms to see how their ratings and installs differ between iOS and Android users.

**Ans-** When comparing iOS and Android apps based on ratings and installs, there are noticeable differences. Android apps have a higher number of installs due to their larger user base and global reach. In contrast, iOS applications exhibit higher average ratings, likely owing to Apple's more stringent app approval procedure and a user base with higher expectations and a greater willingness to pay for superior quality. Despite Android's superior installation numbers due to its affordability and accessibility, iOS applications frequently exhibit superior user engagement and retention. The Play Store attracts a diverse range of users, resulting in lower average ratings. However, the App Store audience tends to be more professional and financially stable, leading to higher app quality and user satisfaction.

**Advantages of this Insight:**

Developer Focus: App developers can prioritize platforms based on their target audience and revenue goals.

Market Strategy: Businesses can adjust their marketing efforts based on how users behave on each platform, such as focusing on installs for Android and aiming for quality on iOS.

Product Positioning: Insights into what users expect on each platform can help developers improve app features and design.

**Disadvantages of this Insight:**

Platform Bias: Developers who concentrate on just one platform might miss out on opportunities available on the other.

Data Homogeneity: Disparities in user demographics across platforms can bias results, rendering them less applicable to a wider market.

Limited Insight into Cross-Platform Users: This analysis may overlook the preferences of users who engage with both ecosystems, which could limit the overall understanding.

**Q2-** How do gender preferences influence app choices and ratings across both platforms and do male or female users prefer certain types of apps more?

**Ans-** The analysis of gender preferences in app choices and ratings uncovers nuanced behaviors across platforms. On the Play Store, the gender distribution is nearly equal, reflecting a diverse user base. This inclusivity drives the popularity of universal genres like gaming, social media, and productivity apps. However, on the App Store, a higher proportion of male users suggests skewed preferences. Male users dominate categories like technology, gaming, and finance, while female users gravitate toward lifestyle, health, and wellness apps. Ratings also differ by gender, with women generally providing higher ratings for apps related to productivity and creativity, while men often rate gaming and technology apps higher. This disparity suggests gender-based expectations and satisfaction levels, which developers can leverage to optimize app features and design. For example, developers targeting female users could enhance personalization and aesthetics in lifestyle apps, while apps targeting male users might emphasize functionality and competitive features.

**Advantages of this Insight:**

Targeted Marketing: Businesses can craft gender-specific marketing campaigns to improve app adoption and retention.

Feature Customization: Developers can tailor app features to better align with gender-based preferences, boosting satisfaction.

Enhanced Retention Strategies: Understanding gender-specific behaviors can help reduce churn.

**Disadvantages of this Insight:**

Risk of Stereotyping: Overgeneralization of gender preferences might lead to missed opportunities in appealing to diverse audiences.

Data Limitations: Differences in cultural or geographic factors influencing gender preferences might not be captured comprehensively.

Dynamic Preferences: Gender-based app choices may evolve rapidly, making static insights less relevant over time.

**Limitations of the Project**

**Data Limitations:**

The datasets for both the App Store and Play Store are anonymized and aggregated, which might omit critical individual-level behavioral insights such as session times or in-app purchase patterns. Some attributes, like app reviews or user feedback, are not detailed, limiting the understanding of qualitative aspects of user satisfaction.

**Platform-Specific Bias:**

The analysis primarily focuses on installs and ratings but doesn't include other important metrics like revenue generation, app retention, or churn rates, which are critical to understanding app success. Differences in user demographics across platforms, such as regional preferences or device affordability, might skew the findings.

**Generalization Issues:**

While the study identifies trends in gender and platform preferences, it assumes homogeneity in user behavior within these groups. Some genres or app categories might be underrepresented in the dataset, leading to biased conclusions about their popularity or success.

**Temporal Limitations:**

The data is static and doesn't account for temporal trends such as seasonal variations in app usage or evolving user preferences.

**Lack of Advanced Metrics:**

Advanced metrics like sentiment analysis on reviews, in-app purchase data, or feature-specific user engagement (e.g., time spent per app) are not included, which could have enriched the analysis.

**Conclusion**

This project lays a strong groundwork for comparing the app ecosystems of the Play Store and App Store. By examining factors like gender distribution, age demographics, app pricing models (free vs. paid), content rating, and user ratings, we've gained valuable insights into how user preferences and engagement differ between these platforms.

The comparative analysis of the Play Store and App Store highlights significant differences in user behaviors, ratings, and preferences. It was visible that android users prioritize affordability and accessibility, which results in higher install volumes. On the other hand, iOS users lean toward quality and exclusivity, reflected in higher app ratings. Gender preferences also influence the types of apps downloaded and rated on both platforms.

While our findings are helpful, we recognize that further analysis is needed to deepen our understanding of how these elements impact app performance. Moving forward, we plan to enhance our visualizations and take a closer look at the relationships between app installs, ratings, and user engagement. Our goal is to develop more targeted strategies for app developers and marketers, helping them optimize their approaches based on the specific behaviors of users on each platform.

**Future Work**

**Dynamic and Temporal Analysis:**

Use data points to study trends over time and different seasons. Additionally, look at trends in app updates and how they affect user engagement.

**Incorporate Revenue Data:**

Include data such as purchase method, spending habits, promotion or discounts affect app installments.

**User Segmentation:**

Add demographic segmentation such as income, family size, and professions to investigate the relations further.

**Sentiment and Review Analysis:**

Perform sentiment analysis on user reviews to understand and incorporate qualitative feedback. Then examine its correlation with ratings and installations.

**Engagement Metrics:**

Analyze app engagement metrics such as session duration, frequency of use, and retention rates.

**Cross-Platform Analysis:**

Investigate users who operate on both platforms to identify commonalities and differences in their app preferences and behavior.

**Machine Learning Models:**

Develop predictive models using advanced machine learning techniques with additional data. Then forecast app success, user retention, and potential installs based on app features and demographics.

**References**

1- Return on Engagement Initiatives: A Study of a Business-to-Business Mobile App - 4, July 2017

By- Manpreet Gill, Shrihari Sridhar, and Rajdeep Grewal <https://journals.sagepub.com/doi/epub/10.1509/jm.16.0149>

2- Two App Stores in One Smartphone：A Comparative Study on Mobile Application Stores between Google Play and T-Store - 2013 By- Andrew Dela Rosa\*\*․Hong Joo Lee\*\*

<https://koreascience.kr/article/JAKO201328053102987.pdf>

3- Apple App Store vs Google Play Store (2024 Comparison) - February 9, 2024- By- Sanja Zdraveska

<https://cybercrew.uk/software/app-store-vs-play-store/>

4- Competition In the Mobile Application Ecosystem - February 2023 - By-Department of Commerce

<https://www.ntia.gov/sites/default/files/publications/mobileappecosystemreport.pdf>

5- Beyond Google Play: A Large-Scale Comparative Study of Chinese Android App Markets - 31 October 2018 - By- Haoyu Wang, Zhe Liu, Jingyue Liang, Narseo Vallina-Rodriguez, Yao Guo, Li Li, Juan Tapiador, Jingcun Cao, Guoai Xu.

<https://dl.acm.org/doi/abs/10.1145/3278532.3278558>

6- Unveiling the Dynamics of Mobile App Adoption: Insights and Trends Across Diverse Categories by - MarketWatch on 27th December 2023. - <https://openquire.com/unveiling-the-dynamics-of-mobile-app-adoption-insights-and-trends-across-diverse-categories/>