PLAYSTORE DATA ANALYSIS USING PYSPARK

INTRODUCTION

This project focuses on leveraging the power of PySpark, a robust and scalable data processing framework, to perform a comprehensive analysis of Google Play Store reviews. By utilizing PySpark within the Google Colab environment, we can efficiently handle large datasets, perform complex transformations, and run SQL queries to extract meaningful information.

LEARNING PROJECT

PySpark is the Python API for Apache Spark, an open-source, distributed computing system designed for big data processing and analytics.

Google Colab provides free access to powerful computational resources, including GPUs and TPUs, which can be useful for intensive PySpark operations.

SQL is a widely-used language for querying and managing data. Many data analysts and engineers are already familiar with SQL, making it easier to adopt PySpark for big data processing.

FIRST WE HAVE TO CHOOSE OUR DATA

When working with PySpark in Google Colab, you can choose your data from various formats. PySpark supports a wide range of data sources and file formats.

FILE CAN BE IN FORM OF :

- CSV (Comma-Separated Values)
- JSON (JavaScript Object Notation)
- Parquet
- ORC (Optimized Row Columnar)
- Avro
- Text Files
- SQL Databases

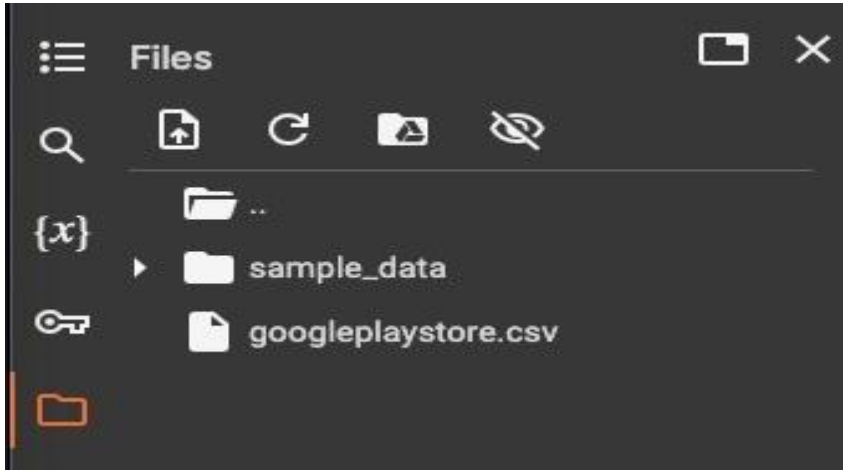
IN THIS MODEL WE ARE USING KAGGLE DATASET

DATASOURCE LINK - https://www.kaggle.com/datasets/lava18/google-play-store-apps

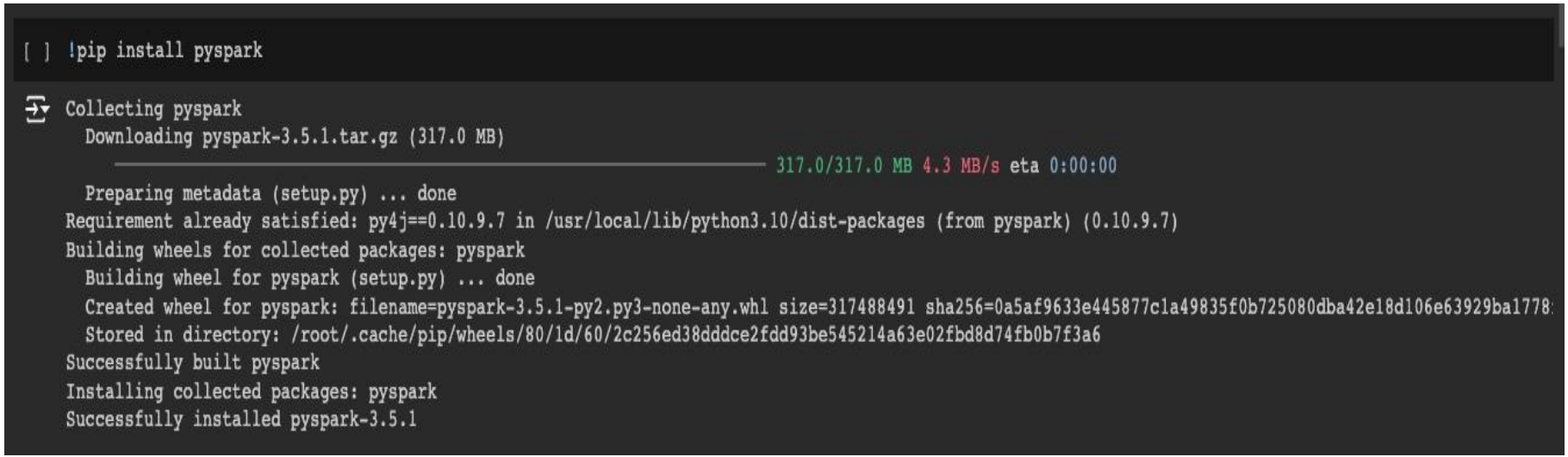
DOWNLOAD THE CSV FILE IN YOUR SYSTEM

This file contains approx 10360 rows and no. of columns with a comma separated values having a fixed schema.

UPLOAD YOUR CSV FILE IN YOUR GOOGLE COLAB NOTEBOOK



SET UP PYSPARK IN YOUR GOOGLE COLAB NOTEBOOK



NOW CREATE A SPARK SESSION

```
[ ] from pyspark.sql import SparkSession
    spark = SparkSession.builder.appName("test_spark").getOrCreate()
```

Read the csv file

```
df= spark.read.csv("googleplaystore.csv",header=True,inferSchema=True) #Read the csv and load it into dataframe
```

Command is used in PySpark to create a temporary view called "apps" from the DataFrame df

WE WILL RUN OUR QUERIES ON THE TABLE “apps”

```
df.show()
```

```
+--------------------+--------------+------+-------+----+-----------+----+-----+--------------+--------------------+------------------+----------------+
|                 App|      Category|Rating|Reviews|Size|   Installs|Type|Price|Content Rating|              Genres|      Last Updated|     Current Ve|
+--------------------+--------------+------+-------+----+-----------+----+-----+--------------+--------------------+------------------+----------------+
|Photo Editor & Ca...|ART_AND_DESIGN|   4.1|    159| 19M|    10,000+|Free|    0|      Everyone|        Art & Design|   January 7, 2018|           1.0.
| Coloring book moana|ART_AND_DESIGN|   3.9|    967| 14M|   500,000+|Free|    0|      Everyone|Art & Design;Pret...|  January 15, 2018|           2.0.
|U Launcher Lite –...|ART_AND_DESIGN|   4.7|  87510|8.7M| 5,000,000+|Free|    0|      Everyone|        Art & Design|    August 1, 2018|           1.2.
|Sketch - Draw & P...|ART_AND_DESIGN|   4.5| 215644| 25M|50,000,000+|Free|    0|          Teen|        Art & Design|      June 8, 2018|Varies with devic
|Pixel Draw - Numb...|ART_AND_DESIGN|   4.3|    967|2.8M|   100,000+|Free|    0|      Everyone|Art & Design;Crea...|     June 20, 2018|              1.
|Paper flowers ins...|ART_AND_DESIGN|   4.4|    167|5.6M|    50,000+|Free|    0|      Everyone|        Art & Design|    March 26, 2017|              1.
|Smoke Effect Phot...|ART_AND_DESIGN|   3.8|    178| 19M|    50,000+|Free|    0|      Everyone|        Art & Design|    April 26, 2018|              1.
|    Infinite Painter|ART_AND_DESIGN|   4.1|  36815| 29M| 1,000,000+|Free|    0|      Everyone|        Art & Design|    June 14, 2018|          6.1.61.
|Garden Coloring Book|ART_AND_DESIGN|   4.4|  13791| 33M| 1,000,000+|Free|    0|      Everyone|        Art & Design|September 20, 2017|            2.9.
|Kids Paint Free -...|ART_AND_DESIGN|   4.7|    121|3.1M|    10,000+|Free|    0|      Everyone|Art & Design;Crea...|     July 3, 2018|              2.
|Text on Photo - F...|ART_AND_DESIGN|   4.4|  13880| 28M| 1,000,000+|Free|    0|      Everyone|        Art & Design|  October 27, 2017|           1.0.
|Name Art Photo Ed...|ART_AND_DESIGN|   4.4|   8788| 12M| 1,000,000+|Free|    0|      Everyone|        Art & Design|    July 31, 2018|          1.0.1
|Tattoo Name On My...|ART_AND_DESIGN|   4.2|  44829| 20M|10,000,000+|Free|    0|          Teen|        Art & Design|   April 2, 2018|              3.
|Mandala Coloring ...|ART_AND_DESIGN|   4.6|   4326| 21M|   100,000+|Free|    0|      Everyone|        Art & Design|    June 26, 2018|           1.0.
|3D Color Pixel by...|ART_AND_DESIGN|   4.4|   1518| 37M|   100,000+|Free|    0|      Everyone|        Art & Design|    August 3, 2018|           1.2.
|Learn To Draw Kaw...|ART_AND_DESIGN|   3.2|     55|2.7M|     5,000+|Free|    0|      Everyone|        Art & Design|     June 6, 2018|              Na
|Photo Designer - ...|ART_AND_DESIGN|   4.7|   3632|5.5M|   500,000+|Free|    0|      Everyone|        Art & Design|    July 31, 2018|              3.
|350 Diy Room Deco...|ART_AND_DESIGN|   4.5|     27| 17M|    10,000+|Free|    0|      Everyone|        Art & Design|November 7, 2017|              1.
|FlipaClip - Carto...|ART_AND_DESIGN|   4.3| 194216| 39M| 5,000,000+|Free|    0|      Everyone|        Art & Design|    August 3, 2018|           2.2.
|       ibis Paint X|ART_AND_DESIGN|   4.6| 224399| 31M|10,000,000+|Free|    0|      Everyone|        Art & Design|    July 30, 2018|           5.5.
+--------------------+--------------+------+-------+----+-----------+----+-----+--------------+--------------------+------------------+----------------+
only showing top 20 rows
```

This gives the schema of dataset

```
df.printSchema()
```

```
root
 |-- App: string (nullable = true)
 |-- Category: string (nullable = true)
 |-- Rating: string (nullable = true)
 |-- Reviews: string (nullable = true)
 |-- Size: string (nullable = true)
 |-- Installs: string (nullable = true)
 |-- Type: string (nullable = true)
 |-- Price: string (nullable = true)
 |-- Content Rating: string (nullable = true)
 |-- Genres: string (nullable = true)
 |-- Last Updated: string (nullable = true)
 |-- Current Ver: string (nullable = true)
 |-- Android Ver: string (nullable = true)
```

DATA CLEANING

```
#DATA CLEANING CODE
# 1. Drop rows with null 'App' values
df = df.dropna(subset=['App'])

# 2. Fill null values for 'Category' with a default value 'Unknown'
df = df.fillna({'Category': 'Unknown'})

# 3. Convert 'Rating' to float
df = df.withColumn("Rating", col("Rating").cast("float"))

# 4. Convert 'Reviews' to integer
df = df.withColumn("Reviews", col("Reviews").cast("integer"))

# 5. Remove 'M' from 'Size' and convert to float (assuming size is in MB)
df = df.withColumn("Size", regexp_replace("Size", "M", "").cast("float"))

# 6. Remove '+' and ',' from 'Installs' and convert to integer
df = df.withColumn("Installs", regexp_replace(regexp_replace("Installs", "\\+", ""), ",", "").cast("integer"))

# 7. Convert 'Price' to float (remove '$' for conversion)
df = df.withColumn("Price", when(col("Price") == "0", 0.0).otherwise(regexp_replace("Price", "\\$", "").cast("float")))

# 8. Standardize 'Last Updated' to a date format
df = df.withColumn("Last Updated", regexp_replace("Last Updated", " ", "-").cast("date"))
```

```
# 9. Fill null values for other columns with appropriate defaults
df = df.fillna({
    "Rating": 0.0,
    "Reviews": 0,
    "Size": 0.0,
    "Installs": 0,
    "Price": 0.0,
    "Content Rating": "Unknown",
    "Genres": "Unknown",
    "Last Updated": "2000-01-01",
    "Current Ver": "Unknown",
    "Android Ver": "Unknown"
})

# Show cleaned DataFrame
df.show()
```

DATA TABLE AFTER CLEANING

App	Category	Rating	Reviews	Size	Installs	Type	Price	Content Rating	Genres	Last Updated	Current Ver	Androi
Photo Editor & Ca...	ART_AND_DESIGN	4.1	159	19.0	10000	Free	0.0	Everyone	Art & Design	2000-01-01	1.0.0	4.0.3 a
Coloring book moana	ART_AND_DESIGN	3.9	967	14.0	500000	Free	0.0	Everyone	Art & Design;Pret...	2000-01-01	2.0.0	4.0.3 a
U Launcher Lite –...	ART_AND_DESIGN	4.7	87510	8.7	5000000	Free	0.0	Everyone	Art & Design	2000-01-01	1.2.4	4.0.3 a
Sketch - Draw & P...	ART_AND_DESIGN	4.5	215644	25.0	50000000	Free	0.0	Teen	Art & Design	2000-01-01	Varies with device	4.2 a
Pixel Draw - Numb...	ART_AND_DESIGN	4.3	967	2.8	100000	Free	0.0	Everyone	Art & Design;Crea...	2000-01-01	1.1	4.4 a
Paper flowers ins...	ART_AND_DESIGN	4.4	167	5.6	50000	Free	0.0	Everyone	Art & Design	2000-01-01	1.0	2.3 a
Smoke Effect Phot...	ART_AND_DESIGN	3.8	178	19.0	50000	Free	0.0	Everyone	Art & Design	2000-01-01	1.1	4.0.3 a
Infinite Painter	ART_AND_DESIGN	4.1	36815	29.0	1000000	Free	0.0	Everyone	Art & Design	2000-01-01	6.1.61.1	4.2 a
Garden Coloring Book	ART_AND_DESIGN	4.4	13791	33.0	1000000	Free	0.0	Everyone	Art & Design	2000-01-01	2.9.2	3.0 a
Kids Paint Free -...	ART_AND_DESIGN	4.7	121	3.1	10000	Free	0.0	Everyone	Art & Design;Crea...	2000-01-01	2.8	4.0.3 a
Text on Photo - F...	ART_AND_DESIGN	4.4	13880	28.0	1000000	Free	0.0	Everyone	Art & Design	2000-01-01	1.0.4	4.1 a
Name Art Photo Ed...	ART_AND_DESIGN	4.4	8788	12.0	1000000	Free	0.0	Everyone	Art & Design	2000-01-01	1.0.15	4.0 a
Tattoo Name On My...	ART_AND_DESIGN	4.2	44829	20.0	10000000	Free	0.0	Teen	Art & Design	2000-01-01	3.8	4.1 a
Mandala Coloring ...	ART_AND_DESIGN	4.6	4326	21.0	100000	Free	0.0	Everyone	Art & Design	2000-01-01	1.0.4	4.4 a
3D Color Pixel by...	ART_AND_DESIGN	4.4	1518	37.0	100000	Free	0.0	Everyone	Art & Design	2000-01-01	1.2.3	2.3 a
Learn To Draw Kaw...	ART_AND_DESIGN	3.2	55	2.7	5000	Free	0.0	Everyone	Art & Design	2000-01-01	NaN	4.2 a
Photo Designer - ...	ART_AND_DESIGN	4.7	3632	5.5	500000	Free	0.0	Everyone	Art & Design	2000-01-01	3.1	4.1 a
350 Diy Room Deco...	ART_AND_DESIGN	4.5	27	17.0	10000	Free	0.0	Everyone	Art & Design	2000-01-01	1.0	2.3 a
FlipaClip - Carto...	ART_AND_DESIGN	4.3	194216	39.0	5000000	Free	0.0	Everyone	Art & Design	2000-01-01	2.2.5	4.0.3 a
ibis Paint X	ART_AND_DESIGN	4.6	224399	31.0	10000000	Free	0.0	Everyone	Art & Design	2000-01-01	5.5.4	4.1 a

only showing top 20 rows

```
[9] #compute the total number of reviews for each app and then sort the apps by the total number of reviews in descending order
    total_reviews = spark.sql("SELECT App, SUM(Reviews) as Total_Reviews FROM apps GROUP BY 1 ORDER BY 2 DESC")
```

```
total_reviews.show()
```

```
+--------------------+-------------+
|                 App|Total_Reviews|
+--------------------+-------------+
|           Instagram| 2.66241989E8|
|  WhatsApp Messenger| 2.07348304E8|
|      Clash of Clans| 1.79558781E8|
|Messenger – Text ...| 1.69932272E8|
|      Subway Surfers| 1.66331958E8|
|    Candy Crush Saga| 1.56993136E8|
|            Facebook| 1.56286514E8|
|         8 Ball Pool|  9.9386198E7|
|        Clash Royale|  9.2530298E7|
|            Snapchat|   6.804501E7|
|     Viber Messenger|  5.6675481E7|
|UC Browser - Fast...|  5.3140694E7|
|             YouTube|  5.1278853E7|
|        Temple Run 2|   4.871093E7|
|Sniper 3D Gun Sho...|  4.6022233E7|
|      My Talking Tom|  4.4668928E7|
|Duolingo: Learn L...|  4.4047832E7|
|       Google Photos|  4.3423827E7|
|Clean Master- Spa...|  4.2916526E7|
|                 Pou|  4.1939801E7|
+--------------------+-------------+
only showing top 20 rows
```

```
[ ] #calculates the total number of installs for each app and its corresponding category, then orders the results by the total number of installs in descen
    No_of_Installs = spark.sql("SELECT App, Category, SUM(Installs) as No_of_Installs FROM apps GROUP BY App, Category ORDER BY No_of_Installs DESC")
```

```
No_of_Installs.show()
```

```
+--------------------+-----------------+-------------+
|                 App|         Category|No_of_Installs|
+--------------------+-----------------+-------------+
|"Yanosik: ""antyr...|      traffic jams|          4.4|
|"Women""s Health ...|             Face|          4.2|
|Command & Conquer...|           FAMILY|          0.0|
|Police Detector (...| AUTO_AND_VEHICLES|         NULL|
|Natural recipes f...|           BEAUTY|         NULL|
|Eyeliner step by ...|           BEAUTY|         NULL|
|                 Box|         BUSINESS|         NULL|
|           following|           DATING|         NULL|
|LOBSTR - go on a ...|           DATING|         NULL|
|Programming Hub, ...|        EDUCATION|         NULL|
| Cinematic Cinematic|    ENTERTAINMENT|         NULL|
|Sarajevo Film Fes...|           EVENTS|         NULL|
|Zombies, Run! 5k ...|HEALTH_AND_FITNESS|         NULL|
|             Happify|HEALTH_AND_FITNESS|         NULL|
|             eBiblio|LIBRARIES_AND_DEMO|         NULL|
|Easy Makeup Tutor...|        LIFESTYLE|         NULL|
|          Toon Blast|             GAME|         NULL|
|Jewels classic Pr...|             GAME|         NULL|
|Papumba Academy -...|           FAMILY|         NULL|
|My Oasis - Calmin...|           FAMILY|         NULL|
+--------------------+-----------------+-------------+
only showing top 20 rows
```

```
[ ] #calculates the total number of installs for each app and orders the results by the total number of installs in descending order
    No_of_Installs = spark.sql("SELECT App, SUM(Installs) as No_of_Installs FROM apps GROUP BY App ORDER BY No_of_Installs DESC")
```

```
No_of_Installs.show()
```

```
+--------------------+--------------+
|                 App|No_of_Installs|
+--------------------+--------------+
|"Yanosik: ""antyr...|           4.4|
|"Women""s Health ...|           4.2|
|Command & Conquer...|           0.0|
|Google Chrome: Fa...|          NULL|
|free video calls ...|          NULL|
|Toddler Learning ...|          NULL|
|             MyChart|          NULL|
|Davis's Drug Guid...|          NULL|
|    Diabetes Testing|          NULL|
|Mercari: The Sell...|          NULL|
|Find&Save - Local...|          NULL|
|                SNCF|          NULL|
|Learn the letters...|          NULL|
|Nigeria News NAIJ...|          NULL|
|    Basketball Stars|          NULL|
|          C Examples|          NULL|
|            Q Wunder|          NULL|
|         Q-Ticketing|          NULL|
|Learn R Language ...|          NULL|
| Al-Quran Al-Muallim|          NULL|
+--------------------+--------------+
only showing top 20 rows
```

```
#names and prices of all paid apps in your dataset.
paid_apps = spark.sql("SELECT App, Price FROM apps WHERE `Type` = 'Paid'")
paid_apps.show()
```

```
+--------------------+-----+
|                 App|Price|
+--------------------+-----+
|TurboScan: scan d...|$4.99|
|Tiny Scanner Pro:...|$4.99|
|TurboScan: scan d...|$4.99|
|Tiny Scanner Pro:...|$4.99|
|  Puffin Browser Pro|$3.99|
|Moco+ - Chat, Mee...|$3.99|
|          Calculator|$6.99|
|   Truth or Dare Pro|$1.49|
|Private Dating, H...|$2.99|
|Ad Blocker for SayHi|$3.99|
|AMBW Dating App: ...|$7.99|
|Moco+ - Chat, Mee...|$3.99|
| Sago Mini Hat Maker|$3.99|
|Fuzzy Numbers: Pr...|$5.99|
|     Toca Life: City|$3.99|
| Toca Life: Hospital|$3.99|
|      My Talking Pet|$4.99|
|      Meme Generator|$2.99|
|My CookBook Pro (...|$3.49|
|Paprika Recipe Ma...|$4.99|
+--------------------+-----+
only showing top 20 rows
```

```
#display the names and ratings of all apps in dataset that have ratings between 4 and 4.5 inclusive.
selected_apps = spark.sql("SELECT App, Rating FROM apps WHERE Rating BETWEEN 4 AND 4.5")
selected_apps.show()
```

```
+--------------------+------+
|                 App|Rating|
+--------------------+------+
|Photo Editor & Ca...|   4.1|
|Sketch - Draw & P...|   4.5|
|Pixel Draw - Numb...|   4.3|
|Paper flowers ins...|   4.4|
|    Infinite Painter|   4.1|
|Garden Coloring Book|   4.4|
|Text on Photo - F...|   4.4|
|Name Art Photo Ed...|   4.4|
|Tattoo Name On My...|   4.2|
|3D Color Pixel by...|   4.4|
|350 Diy Room Deco...|   4.5|
|FlipaClip - Carto...|   4.3|
|Logo Maker - Smal...|   4.0|
|Boys Photo Editor...|   4.1|
|Animated Photo Ed...|   4.1|
|Easy Realistic Dr...|   4.1|
|Pink Silver Bow K...|   4.2|
|   Art Drawing Ideas|   4.1|
|Anime Manga Color...|   4.5|
|  Easy Origami Ideas|   4.2|
+--------------------+------+
only showing top 20 rows
```

```
#display the names and number of installs of the top 5 apps with the highest number of installs in dataset.
top_5_apps = spark.sql("SELECT App, Installs FROM apps ORDER BY Installs DESC LIMIT 5")
top_5_apps.show()
```

```
+--------------------+-----------+
|                 App|   Installs|
+--------------------+-----------+
|Life Made WI-Fi T...|       Free|
|     Viber Messenger|500,000,000+|
|imo free video ca...|500,000,000+|
|imo free video ca...|500,000,000+|
|Google Duo - High...|500,000,000+|
+--------------------+-----------+
```

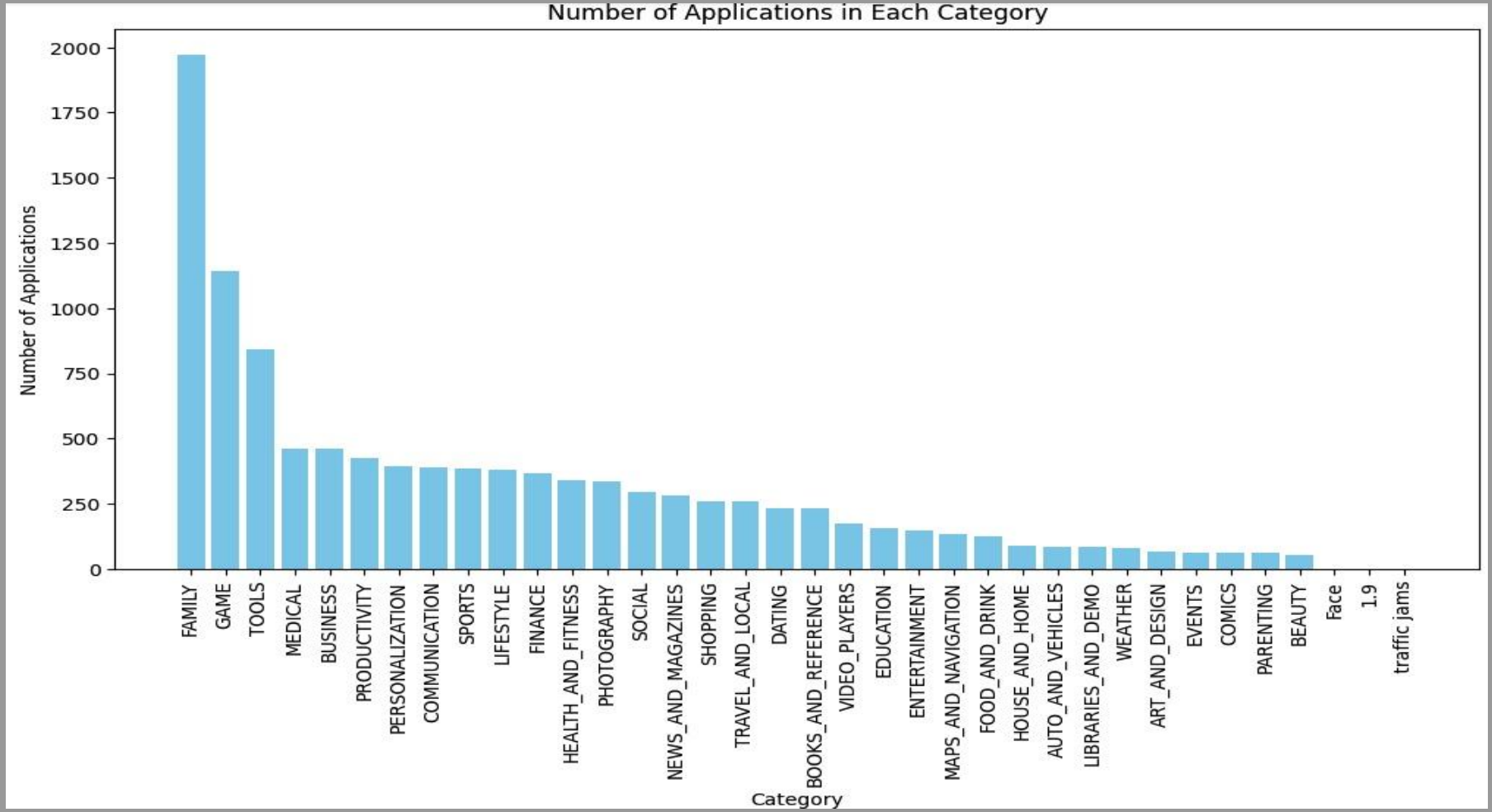
```
#display the names and number of reviews of the top 10 apps with the highest number of reviews in dataset.
top_10_apps = spark.sql("SELECT App, Reviews FROM apps ORDER BY Reviews DESC LIMIT 10")
top_10_apps.show()
```

```
+--------------------+-------------+
|                 App|      Reviews|
+--------------------+-------------+
|"Women""s Health ...|weight lose)"|
|GollerCepte Live ...|         9992|
|Ad Block REMOVER ...|          999|
| SnipSnap Coupon App|         9975|
| SnipSnap Coupon App|         9975|
|US Open Tennis Ch...|         9971|
|US Open Tennis Ch...|         9971|
|          DreamTrips|         9971|
|Adult Color by Nu...|          997|
|BSPlayer ARMv7 VF...|         9966|
+--------------------+-------------+
```

```
# Create a new DataFrame with the count of apps in each category
category_counts = df.groupBy('Category').count().orderBy('count', ascending=False)

# Convert the PySpark DataFrame to a Pandas DataFrame for plotting
category_counts_pd = category_counts.toPandas()

# Plotting the data
plt.figure(figsize=(12, 6))
plt.bar(category_counts_pd['Category'], category_counts_pd['count'], color='skyblue')
plt.xlabel('Category')
plt.ylabel('Number of Applications')
plt.title('Number of Applications in Each Category')
plt.xticks(rotation=90)
plt.show()
```


Number of Applications in Each Category
Number of Applications
2000
1750
1500
1250
1000
750
500
250
0
FAMILY
GAME
TOOLS
MEDICAL
BUSINESS
PRODUCTIVITY
PERSONALIZATION
COMMUNICATION
SPORTS
LIFESTYLE
FINANCE
HEALTH_AND_FITNESS
PHOTOGRAPHY
SOCIAL
NEWS_AND_MAGAZINES
SHOPPING
TRAVEL_AND_LOCAL
DATING
BOOKS_AND_REFERENCE
VIDEO_PLAYERS
EDUCATION
ENTERTAINMENT
MAPS_AND_NAVIGATION
FOOD_AND_DRINK
HOUSE_AND_HOME
AUTO_AND_VEHICLES
LIBRARIES_AND_DEMO
WEATHER
ART_AND_DESIGN
EVENTS
COMICS
PARENTING
BEAUTY
Face
1.9
traffic jams
Category

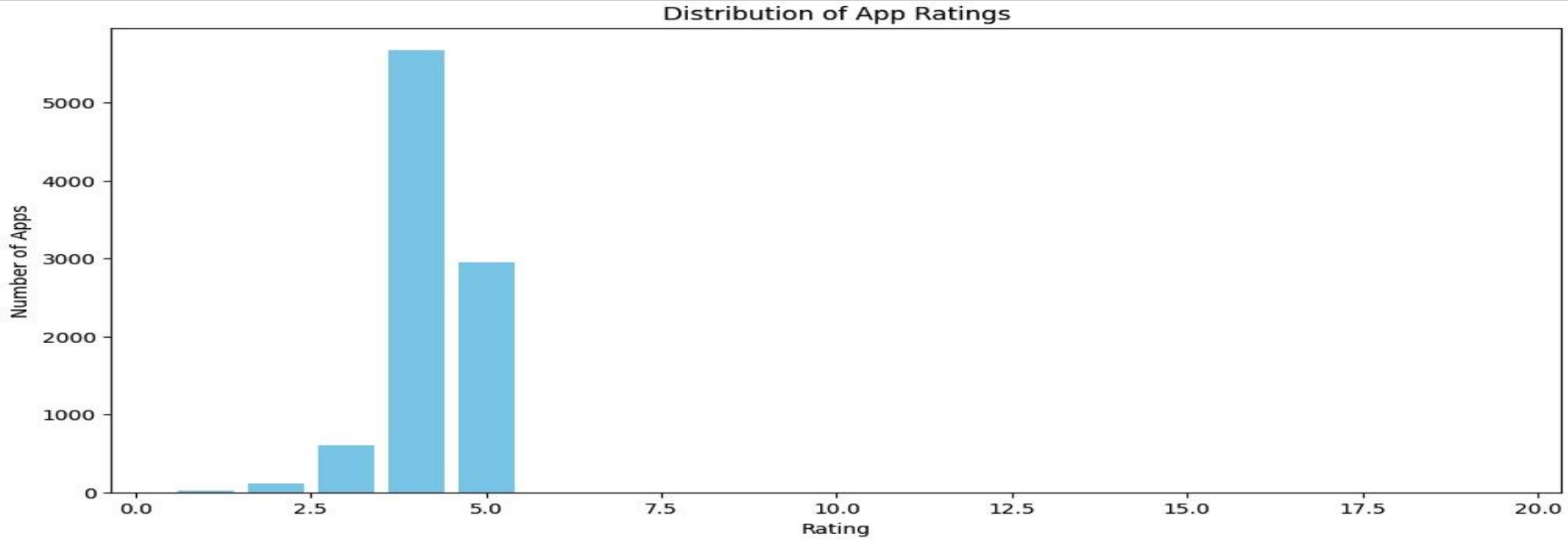
```
[ ] # Create a new DataFrame with the count of apps in each Content Rating
    content_rating_counts = spark.sql("SELECT `Content Rating`, COUNT(*) as AppCount FROM apps GROUP BY `Content Rating`")

    # Show the table
    content_rating_counts.show()
```

```
+--------------+--------+
| Content Rating|AppCount|
+--------------+--------+
|        Unrated|       2|
|           Teen|    1208|
|     5,000,000+|       1|
|           NULL|       1|
|     1,000,000+|       1|
|     Mature 17+|     498|
|   Everyone 10+|     414|
|       Everyone|    8713|
|Adults only 18+|       3|
+--------------+--------+
```

```
# Run an SQL query to get the distribution of app ratings
# The query rounds the ratings to the nearest integer, counts the number of apps for each rounded rating,
# groups by the rounded rating, and orders the results by the rounded rating
ratings_distribution = spark.sql("SELECT ROUND(Rating) AS RoundedRating, COUNT(*) AS Count FROM apps GROUP BY ROUND(Rating) ORDER BY RoundedRat
ratings_distribution_pd = ratings_distribution.toPandas()

plt.figure(figsize=(12, 6))
plt.bar(ratings_distribution_pd['RoundedRating'], ratings_distribution_pd['Count'], color='skyblue')
plt.xlabel('Rating')
plt.ylabel('Number of Apps')
plt.title('Distribution of App Ratings')
plt.show()
```

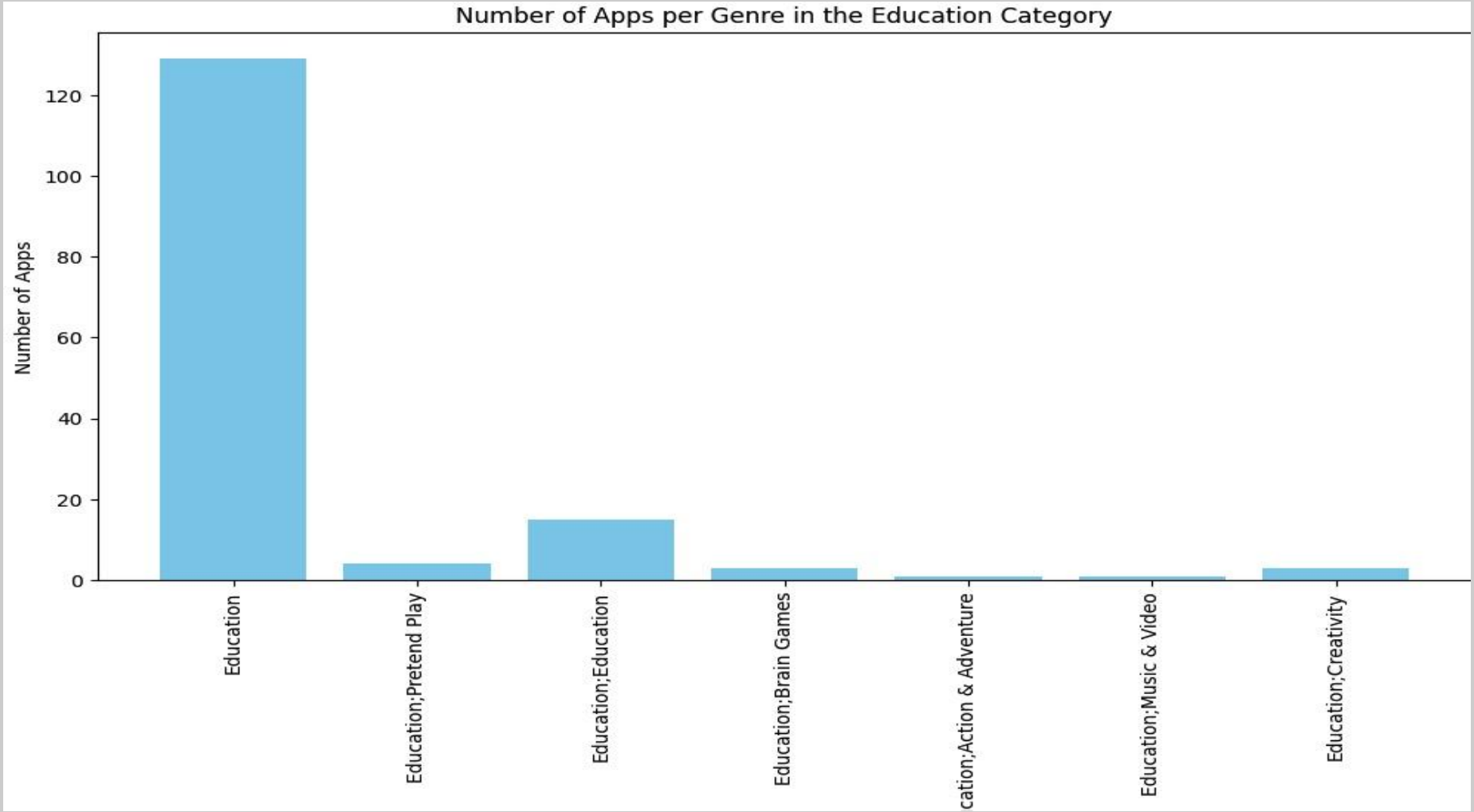


```
#The query groups the data by 'Type' (Free or Paid) and sums the 'Installs' for each type
installs_comparison = spark.sql("SELECT Type, SUM(Installs) AS TotalInstalls FROM apps GROUP BY Type")
installs_comparison.show()
```

```
+------+-------------+
|  Type|TotalInstalls|
+------+-------------+
|     0|         NULL|
|102248|          4.4|
|   NaN|          0.0|
|  Free|         NULL|
|  Paid|         NULL|
|  2509|          4.2|
+------+-------------+
```

```
# The query filters the data to include only apps in the 'EDUCATION' category,
# groups the data by 'Genres', and counts the number of apps in each genre
education_apps_genres = spark.sql("SELECT Genres, COUNT(*) AS AppCount FROM apps WHERE Category = 'EDUCATION' GROUP BY Genres")
education_apps_genres_pd = education_apps_genres.toPandas()

plt.figure(figsize=(12, 6))
plt.bar(education_apps_genres_pd['Genres'], education_apps_genres_pd['AppCount'], color='skyblue')
plt.xlabel('Genres')
plt.ylabel('Number of Apps')
plt.title('Number of Apps per Genre in the Education Category')
plt.xticks(rotation=90)
plt.show()
```


Number of Apps per Genre in the Education Category
Number of Apps
0
20
40
60
80
100
120
Education
Education;Pretend Play
Education;Education
Education;Brain Games
cation;Action & Adventure
Education;Music & Video
Education;Creativity

```
#shows the maximum of categories
max_price_by_category = spark.sql("SELECT Category, MAX(Price) AS MaxPrice FROM apps GROUP BY Category")
max_price_by_category.show()
```

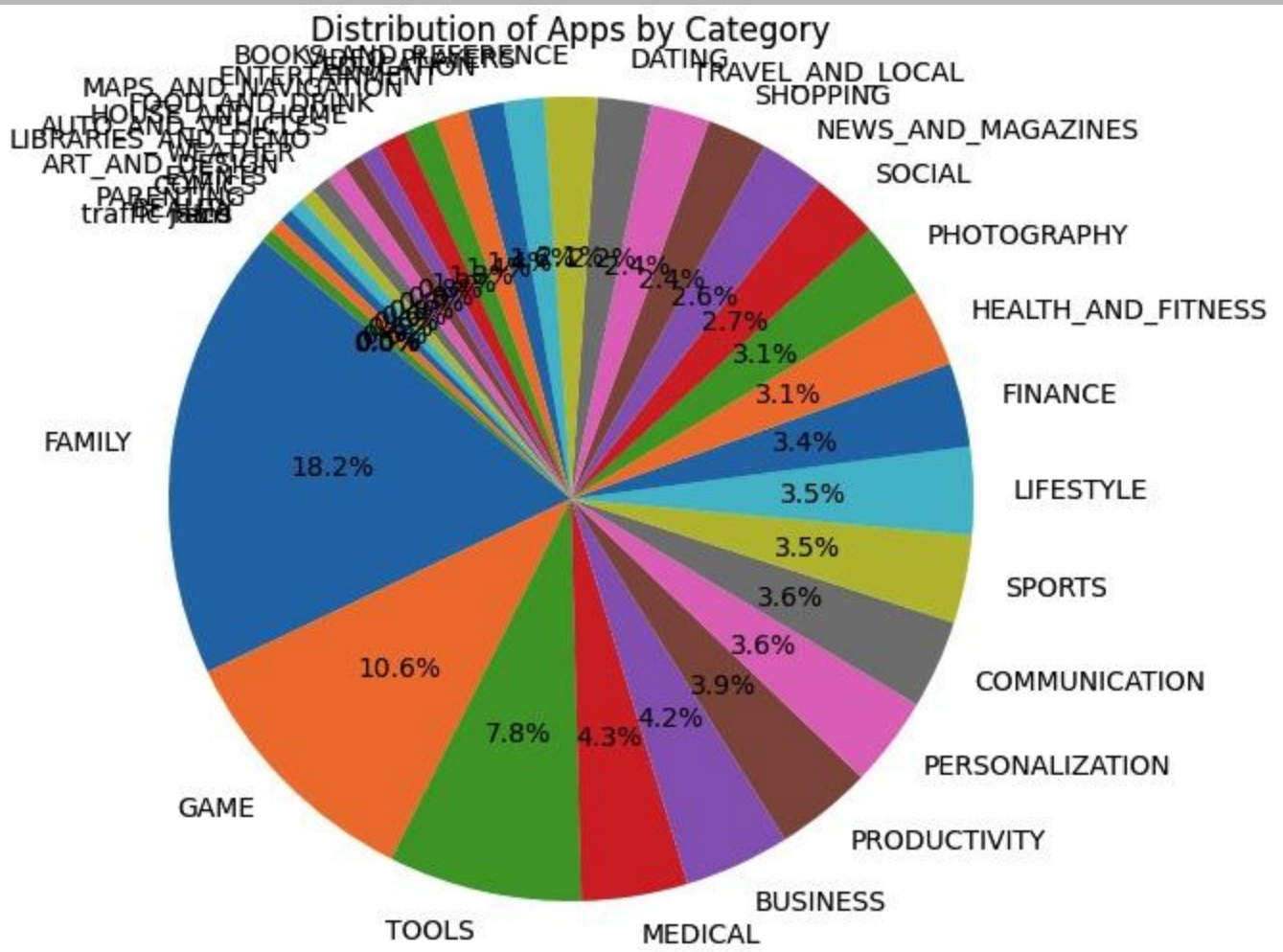
```
+------------------+------------------+
|          Category|          MaxPrice|
+------------------+------------------+
|       traffic jams|Varies with device|
|               1.9|          Everyone|
|     ART_AND_DESIGN|                 0|
|  AUTO_AND_VEHICLES|                 0|
|            BEAUTY|                 0|
|BOOKS_AND_REFERENCE|                 0|
|          BUSINESS|                 0|
|            COMICS|                 0|
|      COMMUNICATION|                 0|
|            DATING|                 0|
|          EDUCATION|                 0|
|      ENTERTAINMENT|                 0|
|            EVENTS|                 0|
|            FAMILY|                 0|
|           FINANCE|                 0|
|     FOOD_AND_DRINK|                 0|
|              Face|              5.0M|
|              GAME|                 0|
| HEALTH_AND_FITNESS|                 0|
|     HOUSE_AND_HOME|                 0|
+------------------+------------------+
only showing top 20 rows
```

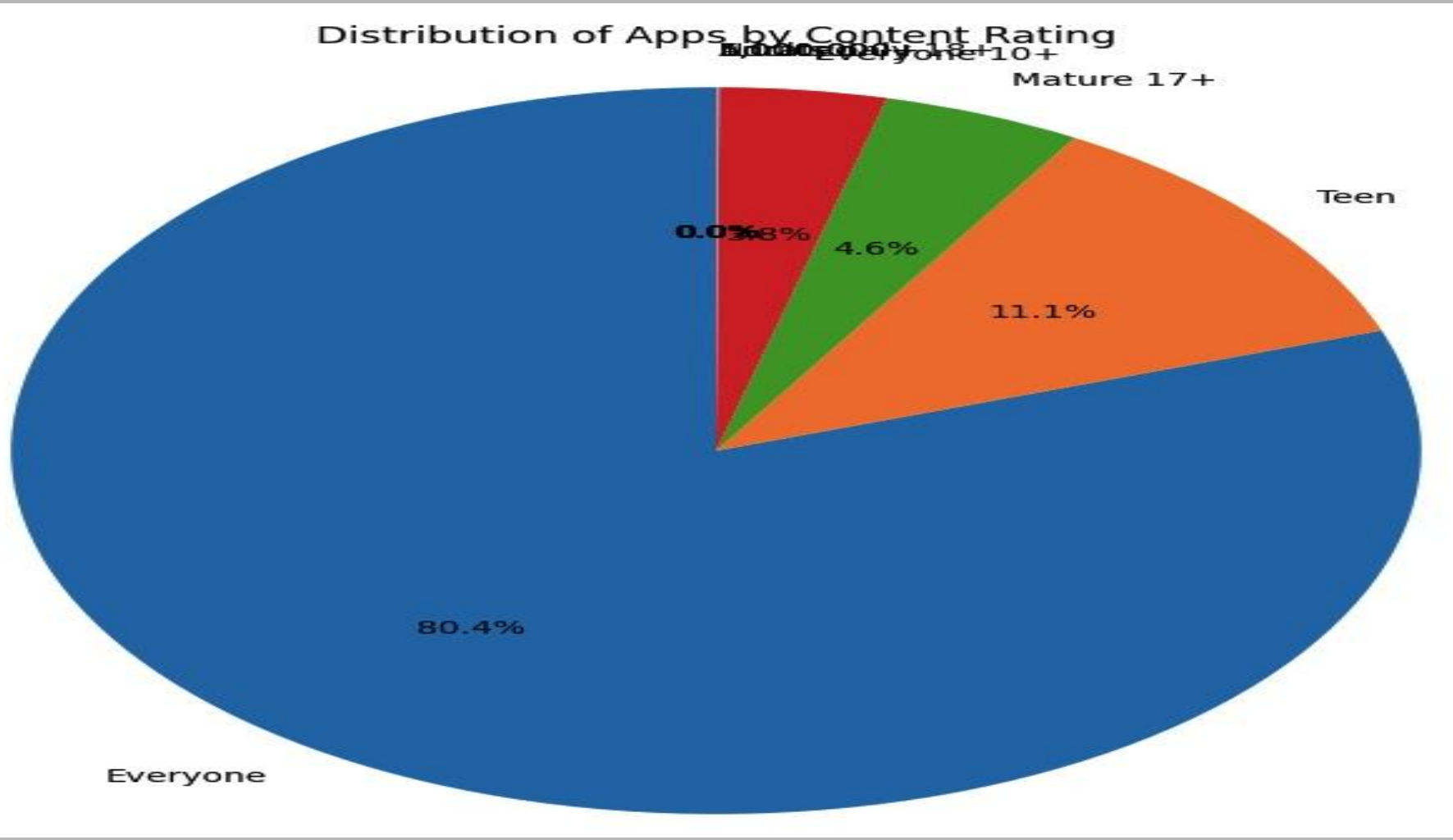
```
# Aggregate the data
category_counts = df.groupBy('Category').count().orderBy('count', ascending=False)
content_rating_counts = df.groupBy('Content Rating').count().orderBy('count', ascending=False)

# Convert PySpark DataFrame to Pandas DataFrame
category_counts_pd = category_counts.toPandas()
content_rating_counts_pd = content_rating_counts.toPandas()

# Create a pie chart for the number of apps in each category
plt.figure(figsize=(12, 6))
plt.pie(category_counts_pd['count'], labels=category_counts_pd['Category'], autopct='%1.1f%%', startangle=140)
plt.axis('equal')  # Equal aspect ratio ensures that pie is drawn as a circle
plt.title('Distribution of Apps by Category')
plt.show()

# Create a pie chart for the distribution of apps by content rating
plt.figure(figsize=(8, 8))
plt.pie(content_rating_counts_pd['count'], labels=content_rating_counts_pd['Content Rating'], autopct='%1.1f%%', startangle=90)
plt.axis('equal')
plt.title('Distribution of Apps by Content Rating')
plt.show()
```


Distribution of Apps by Category
DATING
TRAVEL_AND_LOCAL
SHOPPING
NEWS_AND_MAGAZINES
SOCIAL
PHOTOGRAPHY
HEALTH_AND_FITNESS
FINANCE
LIFESTYLE
SPORTS
COMMUNICATION
PERSONALIZATION
PRODUCTIVITY
BUSINESS
MEDICAL
TOOLS
GAME
FAMILY
18.2%
10.6%
7.8%
4.3%
4.2%
3.9%
3.6%
3.6%
3.5%
3.5%
3.4%
3.1%
3.1%
2.7%
2.6%


Distribution of Apps by Content Rating
Everyone 10+
Mature 17+
Teen
0.0%
4.6%
11.1%
80.4%
Everyone