

Project – Apple app store Analysis

Problem Statement:

Analyze apple app store data to provide market insights to potential stakeholders and give them answers to basic questions based of analysis, to enable them to make data driven decisions.

We have taken a dataset from Kaggle to analyze the information on the apps available on the app store. This includes the apps name, the sizing (bytes), Ratings, supporting devices, Supported Languages, genres, rating count etc.

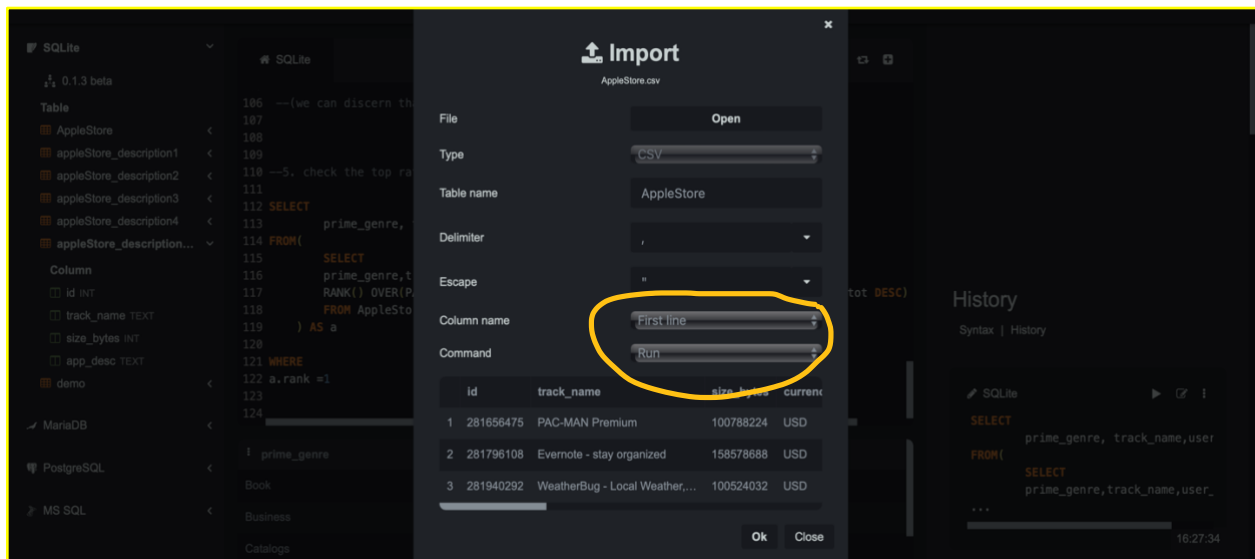
To get an idea of the app descriptions and what these apps do we have taken another dataset that stores all the app descriptions and gives us a lot of information about how the app interfaces with its users and how it could perform in the market.

Connect To Database:

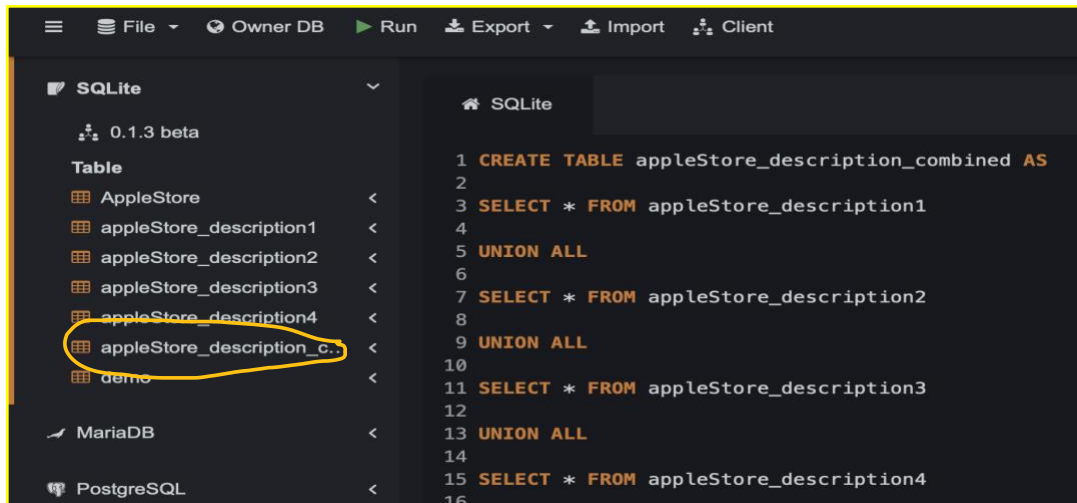
For the execution of the project and ease of access I have chosen to work with SQL lite, which gives me the ease of analyzing my dataset without and installations necessary.

As an alternative you can also use MySQL workbench and setup a root user and connect to the IDE as a localhost.

Due to size restrictions, we have had to split the app store descriptions dataset in 4 parts and will **upload** them all individually and rejoin them later using a UNION ALL function.



Use the UNION ALL function for combining all the 4 files that we split



Stakeholders:

As per our dataset and use case, stakeholders are [aspiring app developers](#) the need data driven insights. To decide what type of app to build and launch, they are seeking answers to questions such as:

1. What price should be set for the app?
2. What categories have the most popular apps based on ratings?
3. How can they maximize rating?

Exploratory Data Analysis:

Aimed at understanding the structure and characteristics of the dataset, it reveals if the dataset happens to have any issues that need to be addressed before further analysis.

Some of the issues can include missing data, incorrect data formats, inconsistent data, errors and outliers.

1. Check the number of unique apps in both tables.

(To ensure we are dealing with the same set of apps in both the datasets, discrepancies could mean missing data in either table)

```

20
21 SELECT COUNT (DISTINCT id) AS uniqueappids
22 FROM AppleStore
23
24 SELECT COUNT (DISTINCT id) AS uniqueappids
25 FROM appleStore_description_combined
26
27 --check for missing values in key fields--
28 SELECT COUNT (*)
29 FROM AppleStore
30 WHERE track_name IS null OR user_rating IS null OR prime_genre IS NULL
31
32 SELECT COUNT (*)
33
: uniqueappids
7197

```

We can discern that both numbers are same so we conclude there is no missing data between the two datasets

2. Check if values are missing in a few key columns.

(we are taking the prime genre, rating and track name for the apple store dataset and app description column for the app description dataset)

```

27 --check for missing values in key fields--
28 SELECT COUNT (*)
29 FROM AppleStore
30 WHERE track_name IS null OR user_rating IS null OR prime_genre IS NULL
31
32 SELECT COUNT (*)
33 FROM appleStore_description_combined
34 WHERE app_desc IS NULL
35
36 --find out the number of apps per genre--
37 SELECT prime_genre, COUNT(*) AS numapps
38 FROM AppleStore
39 GROUP BY prime_genre
40
: count (*)
0

```

Here, the count is 0 for both the tables so we can conclude that there are no missing values.

3. Find out the number of apps per genre.

(Gives us a nice overview of the genre distribution in the Appstore, helping us identify dominant genres)

The screenshot shows a database client interface with two SQL queries and their results.

Query 1: Counts the number of apps per genre from the AppleStore table, excluding NULL descriptions. The results are ordered by the number of apps in descending order.

prime_genre	numapps
Games	3862
Entertainment	535
Education	453
Photo & Video	349
Utilities	248
Health & Fitness	180
Productivity	178

Query 2: Provides an overview of app ratings by selecting the minimum, maximum, and average user ratings from the AppleStore table.

minrating	maxrating	avgrating
0	5	3.526955675976101

We see that the Games & Entertainment industry is clearly leading with the number of apps present in the Appstore.

4. Get an overview of the rating trends of the Appstore.

(We will take some basic measures such as the minimum, maximum and average of the user ratings)

The screenshot shows a SQL query and its result in a database client.

Query: Selects the minimum, maximum, and average user ratings from the AppleStore table.

minrating	maxrating	avgrating
0	5	3.526955675976101

The average rating is 3.52, the maximum being 5.

Finding Insights and Data Analysis:

a. Do paid apps have a higher rating than free apps?

(To determine this, we will use a CASE WHEN statement and select the average user rating)

```

50 **DATA ANALYSIS**
51
52 --1.Determine whether paid apps have a higher rating than free apps--
53 SELECT CASE
54     WHEN price > 0 THEN 'paid'
55     ELSE 'free'
56     END AS App_type,
57     avg(user_rating) AS avg_rating
58 FROM AppleStore
59 GROUP BY App_type
60
61

```

App_type	avg_rating
free	3.3767258382642997
paid	3.720948742438714

On an average the ratings of **paid apps** are slightly higher than free apps.

- b. Determine of apps with a greater number of supported languages get higher ratings?**
(To determine this, we will use a CASE WHEN statement and select the average user rating)

```

63 -- 2. Do apps that support more languages get higher rating?--
64
65 SELECT CASE
66     WHEN lang_num < 10 THEN 'less than 10'
67     WHEN lang_num BETWEEN 10 AND 30 THEN '10-30 languages'
68     ELSE '>30 languages'
69     END AS language_pool,
70     avg(user_rating) AS Avg_Rating
71 FROM AppleStore
72 GROUP BY language_pool
73 ORDER BY Avg_Rating DESC
74
75 --( we can see 10-30 languages are sufficient enough to get good ratings)--
76
77

```

language_pool	Avg_Rating
10-30 languages	4.1305120910384066
>30 languages	3.7777777777777777
less than 10	3.368327402135231

Based on the language bucket with range of 10-30 having the highest ratings we can conclude that we don't need to focus hard on language support barring a few essential and popular languages.

- c. Check the top 15 genres with the least ratings.**
(Low ratings indicate that the users are unhappy that their demands are not being met and there is a good chance to create a successful app in this space)

```

77
78 --3. check genres with low ratings--
79
80 SELECT prime_genre, avg(user_rating) AS Avg_Rating
81 FROM AppleStore
82 GROUP BY prime_genre
83 ORDER BY Avg_Rating ASC
84 LIMIT 15
85
86 --(there is space to create apps in. catalogs and finance industry)--
87

```

prime_genre	Avg_Rating
Catalogs	2.1
Finance	2.4326923076923075
Book	2.4776785714285716
Navigation	2.6847826086956523
Lifestyle	2.8055555555555554
News	2.98
Sports	2.982456140350877

d. Check for correlation between app description length and user ratings.

(We will JOIN the id column in both tables and use a CASE -WHEN -END statement to define a character length condition)

```

89 --4. is there a correlation between length of app description and use ratings--
90 SELECT CASE
91     WHEN length(app_desc) <500 THEN 'Short description'
92     WHEN length(app_desc) BETWEEN 500 AND 1000 THEN 'Medium description'
93     ELSE 'long description'
94 END AS description_length_bucket,
95 avg(a.user_rating) AS avg_rating
96
97 FROM AppleStore AS a
98
99 JOIN appleStore_description_combined AS b
100
101 ON a.id = b.id
102
103 GROUP BY description_length_bucket
104 ORDER BY avg_rating DESC
105
106 --(we can discern that the longer the description the better the rating)--
107

```

description_length_bucket	avg_rating
long description	3.855946944988041
Medium description	3.232809430255403
Short description	2.533613445378151

We have found a positive correlation between the app descriptions length and the user ratings.

We can understand that the users prefer to get a good understanding of the app features before they install the applications.

e. Check the top-rated apps for each genre.

(we use the RANK function to assign a rank to a partition of genres ordered by the user ratings)

```
110 --5. check the top rated apps for a each genre--
111
112 SELECT
113     prime_genre, track_name, user_rating
114 FROM(
115     SELECT
116         prime_genre, track_name, user_rating,
117         RANK() OVER(PARTITION BY prime_genre ORDER BY user_rating DESC, rating_count_tot DESC)
118         FROM AppleStore
119     ) AS a
120
121 WHERE
122 a.rank =1
123
```

prime_genre	track_name	user_rating
Book	Color Therapy Adult Coloring Book fo...	5
Business	TurboScan™ Pro - document & recei...	5
Catalogs	CPlus for Craigslist app - mobile clas...	5
Education	Elevate - Brain Training and Games	5
Entertainment	Bruh-Button	5
Finance	Credit Karma: Free Credit Scores, Re...	5
Food & Drink	Domino's Pizza USA	5
Games	Head Soccer	5
Health & Fitness	Yoga Studio	5
Lifestyle	ipsy - Makeup, subscription and beau...	5

a.rank=1 selects only the observations with rank 1 or top row, it means it selects the highest rated apps from each genres, to break rank ties we use the total user ratings column.

We can look at top performing apps, which is a nice insight for a anyone to emulate the design flow to get better functionality and ratings.

Final Recommendations:

- Paid apps have a better rating than free apps,

- This could be due to a slew of reasons such as, users engaging more and perceiving value leading to better ratings.
- We recommend charging a small fee if the perceived app quality is good.
- Apps supporting a moderate number of languages (10-30) receive higher ratings.
 - We recommend that language quantity is not important, rather proving the widely spoken ones are.
- Finance and Books genres have very low user ratings, which means user demands are not being met.
 - This represents an opportunity to deliver a quality app in these segments, that better matches user requirements
 - This will stand a much better chance of getting a high user rating and effective market penetration.
- Length of the app description has a direct correlation with the app ratings.
 - This means that users want a clear explanation of the app functionality, features and capabilities.
 - A detailed, well-crafted app description can help set user expectations, boosting user satisfaction.
- On an average the rating for an app is 3.5
 - So, a new app should aim for achieving a higher rating than 3.5 to be better.
- Games and Entertainment genres have the highest number of functioning apps, which indicates that is area is saturated.
 - Entering these spaces may be challenging due to high competition.
 - However, it clearly indicates high demand in these sectors also.