Team -1

Apple Music marketing analysis

Overview

How Apple Music can increase sales and compete with Spotify which is currently the leading music streaming app

The problem

A. Company -

With Apple Music, you can access millions of songs, curated playlists and radio, and music recommendations.

B. Context -

Spotify has gained popularity and is currently ruling the market.

We need to bridge this growth gap and device a marketing strategy to become the number 1 music streaming app.

C. Problem statement

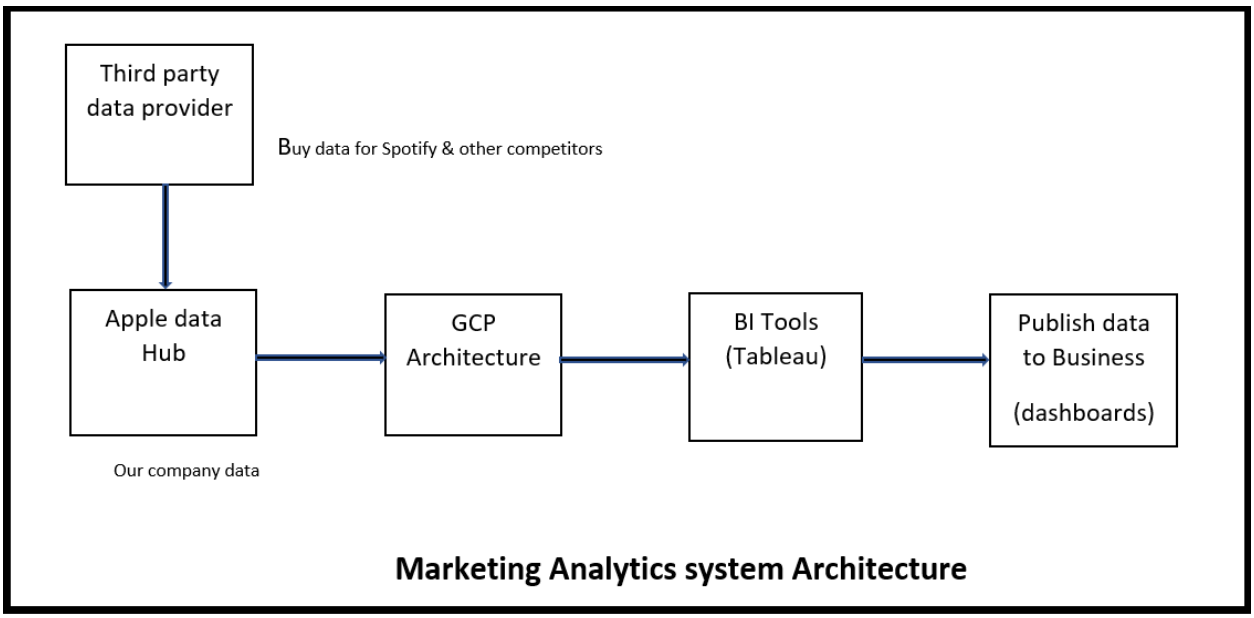
Customer base of apple music is declined by 21%.

Total number of music subscribers is extremely less compared to Spotify. Thus, revenue has been affected considerably.

Solution

Build a system using Google Cloud Platform which will ingest statistical marketing data of Apple and Spotify for analyzing trends and differences.

Implementation



1. Capture the marketing data of Apple Music and Spotify

2. Build a system in which cloud pub/sub will ingest data files in batches

3. Cloud data flow transforms the JSON file into structured, schema-based data.

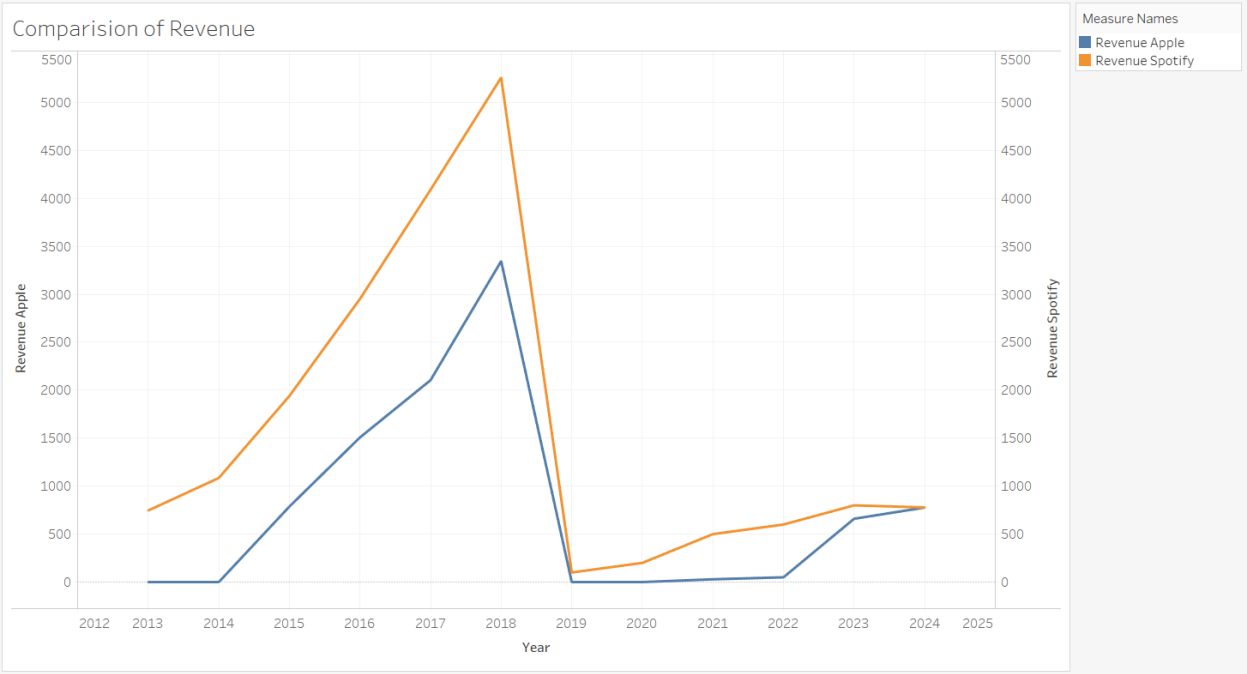
4. That data will be loaded into the BigQuery analytics engine

5. This data will then be fed to BI tools (Tableau/PowerBI) for data analysis.

6. The data set endpoints will be provided to other teams for their analysis.

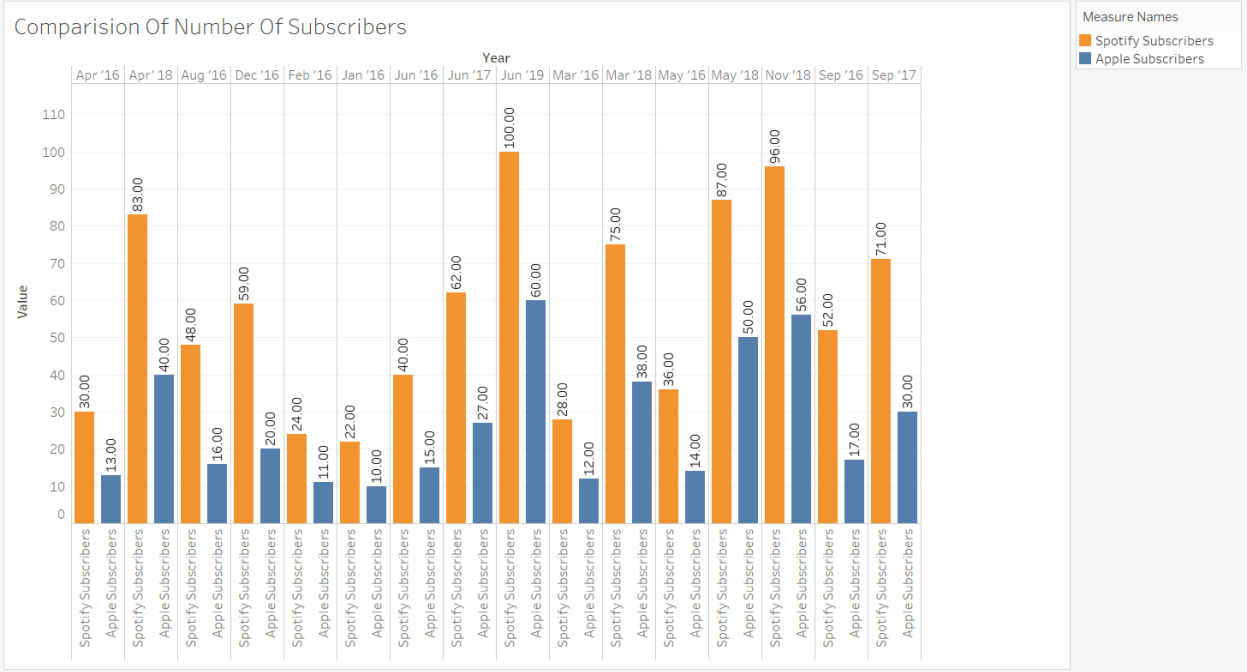
7. We will be presenting descriptive Analysis of the data.

TABLEAU ANALYSIS

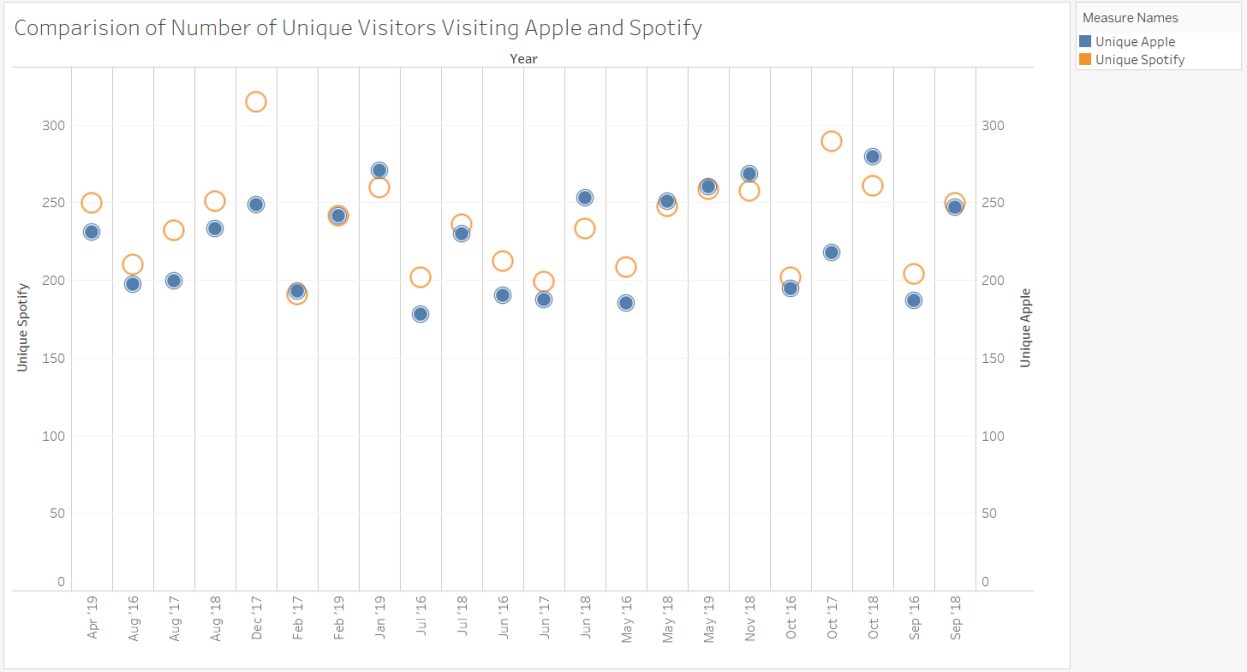


This Graph is a comparison between the Revenues of Apple Music and Spotify.

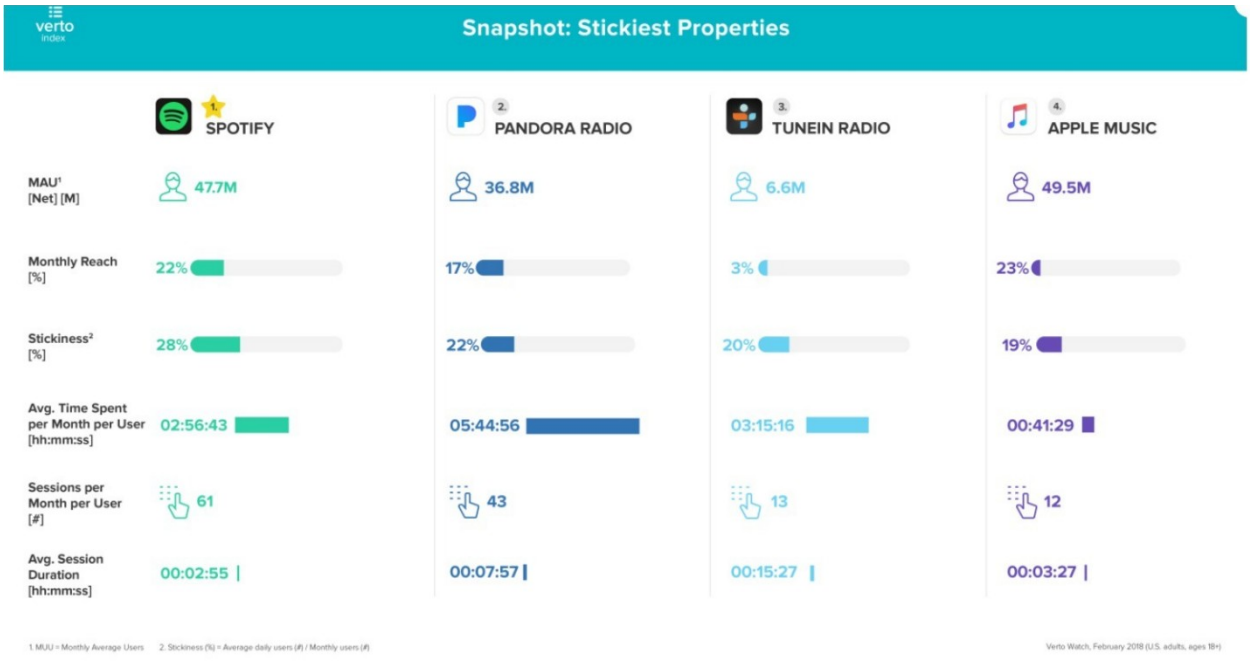
The trend is not downwards, it was just for the purpose of showing the real time update between Python and Tableau



This Chart Represents the Comparison Between the Number of Subscribers at Different Point in Time.



The above chart compares the Number of Unique Visitors that visit either Spotify or Apple Music.



The Snapshot above represents the comparison between various Music Streaming services.
This Analysis shows us that **Apple Music has the max Monthly Average Users as well as Highest Monthly Reach.**

Its main issue is **Stickiness**, which is defined as Average daily Users/ Monthly Users. The higher the stickiness rate the better.

verto index

Music Streaming Properties, February 2018

POSITION & SUBJECT	MONTHLY UNIQUE USERS [Net] [M]	MONTHLY REACH [%]	STICKINESS[1] [%]	AVERAGE SESSION DURATION [MM:SS]	SESSIONS PER MONTH PER USER [#]	SMARTPHONE USERS [M]	TABLET USERS [M]	PC USERS [M]	MOBILE ONLY USERS [M]	PERCENTAGE OF MOBILE-ONLY USERS
1. Apple Music	49.5M	23%	19%	03:27	12	44.6M	7.7M	0.6M	48.9M	99%
2. Spotify	47.7M	22%	28%	02:55	61	32.6M	6.3M	21.6M	26.1M	55%
3. Pandora	36.8M	17%	22%	07:57	43	31.9M	3.4M	5.7M	31.1M	84%
4. SoundCloud	34.2M	16%	9%	01:26	11	17.6M	3.2M	15.3M	18.9M	55%
5. Google Play Music	21.9M	10%	14%	01:15	21	19.5M	2.4M	1.2M	20.7M	95%
6. iHeartRadio	19.9M	9%	14%	02:53	19	13.9M	1.6M	6.3M	13.7M	69%
7. Amazon Music	12.7M	6%	18%	02:17	26	9.6M	0.4M	3.1M	9.6M	76%
8. Shazam	10.6M	5%	12%	01:22	7	10.6M	0.3M		10.6M	100%
9. SiriusXM	7.6M	4%	10%	01:18	14	4.6M	0.4M	3.9M	3.7M	48%
10. TuneIn Radio	6.6M	3%	20%	15:27	13	4.5M	0.7M	1.6M	5.0M	75%

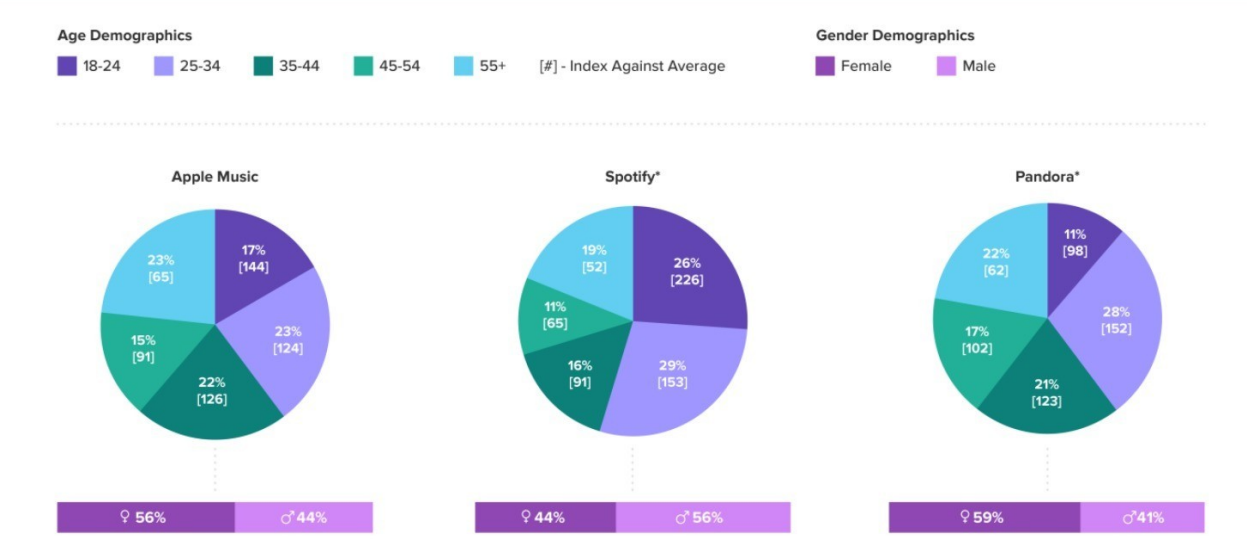
1. Stickiness (%) = Average daily users (#) / Monthly users (#)

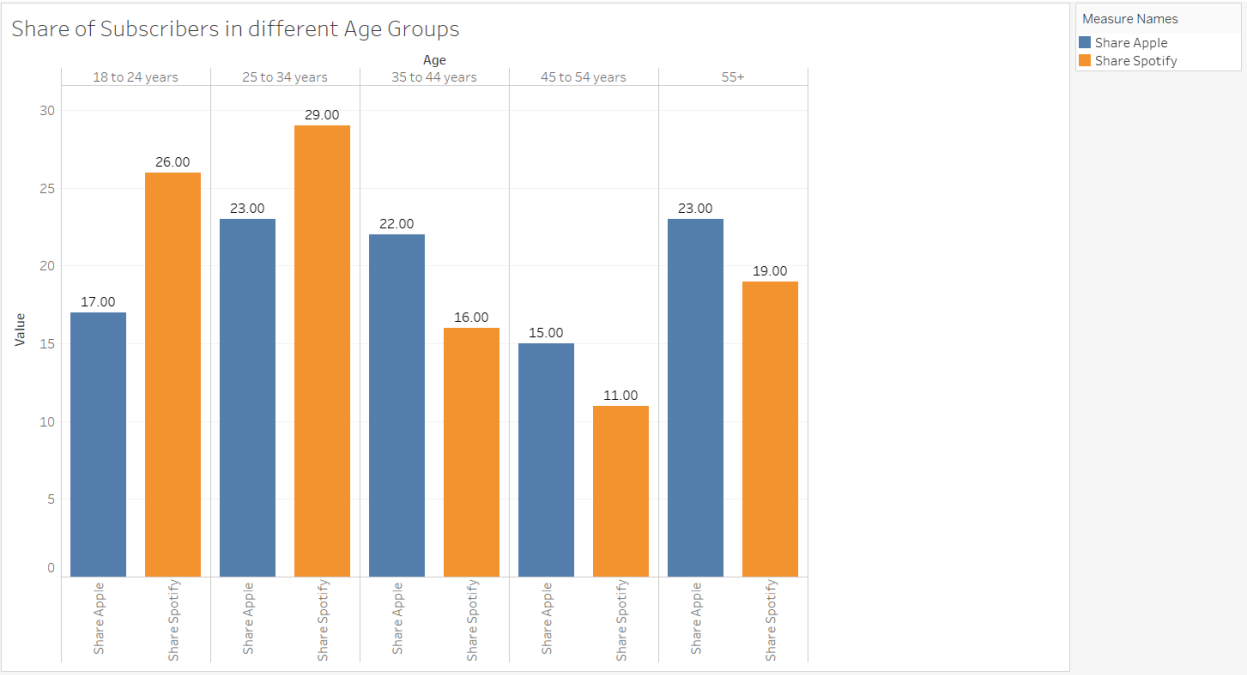
Verto Watch, February 2018 (U.S. adults, ages 18+)

The above snapshot is a more detailed comparison of Apple Music with other Streaming Services.

Use Case 1 - Analyzing data for age and gender demographics:

Our analysis shows that, age groups between 18 to 35 are maximum subscribers which is dominated by Spotify.



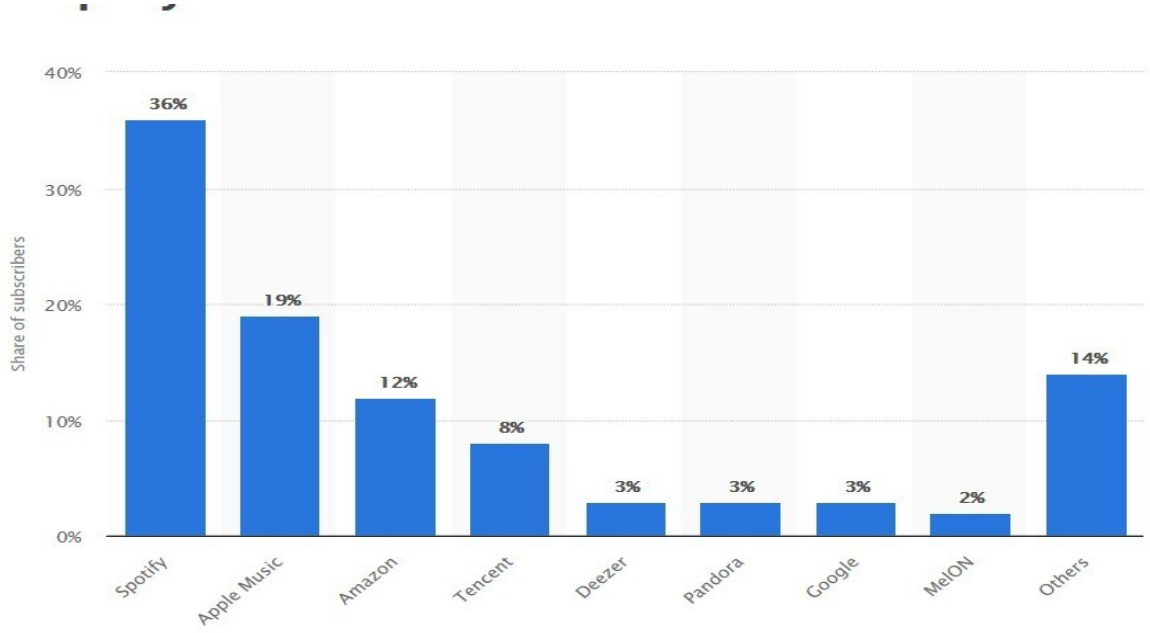


The above two Graphs details how the age group is affecting the number of Subscribers.
We can see that Age Group 18-35 have Maximum numbers of Subscribers.
This section is dominated by Spotify while remaining Age Groups are Dominated by Apple Music.

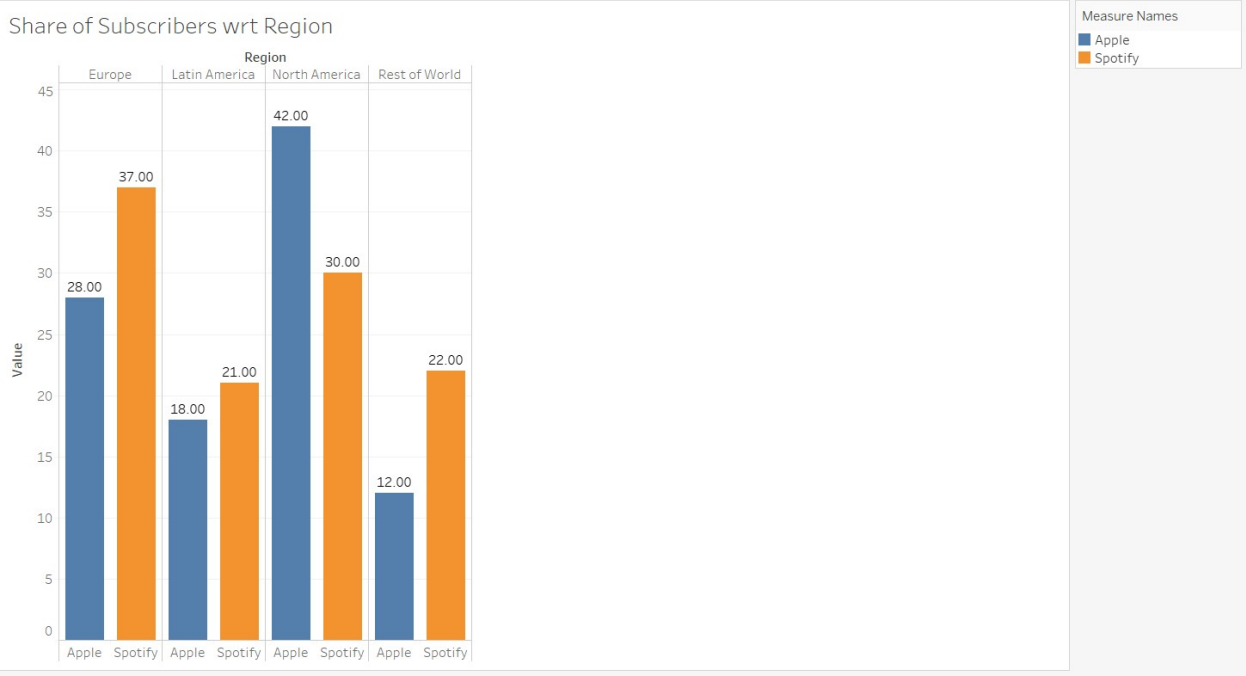
The First Graph, additionally highlights the Distribution between Gender.

Use Case 2 - Analyzing region wise data:

Our analysis shows that, Apple Music dominates in the USA but in other global regions it lags behind Spotify.



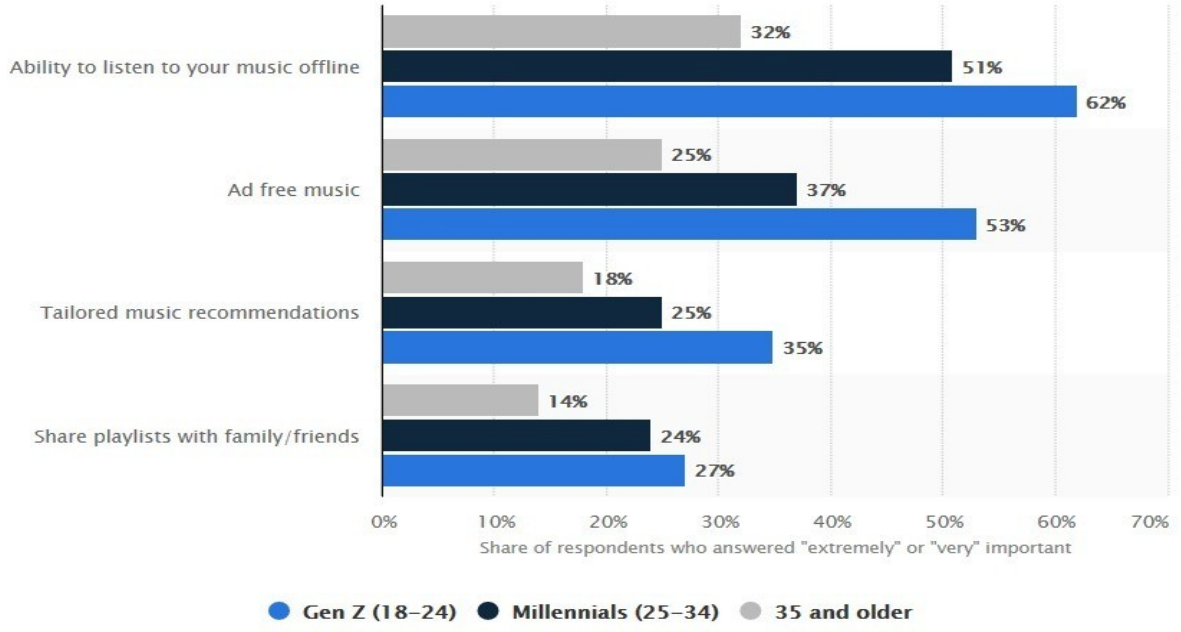
The above Graph Shows the division between various Music Streaming Services in terms of Subscriber Share.
As we can clearly see that Spotify ranks first and Apple Music has a lot of Catching to do.
One of the main reason for that is World wide reach.



Here, We can see that Apple has a good hold in the regions of North America but everywhere else it needs to improve.

Use Case 3 - Analyzing recommendation system:

According to google, recommendation system is better for Spotify as compared to Apple. Tailored music is one of the most important features for user subscriptions.



Here, we can see that Recommendation System is the Third most Important feature when it comes to User Subscription and as we can recollect from Use Case 1 that Age Group 18-35 are the highest number of Subscribers and those are the ones who think that this feature is an important part when it comes to choosing Music Streaming Service.

Python – For Real Time

Following are the screenshots from Python Notebook which shows how we connected to the GCP and how are we updating the information in real time.

In [2]:
```
from google.oauth2 import service_account #For GCP Account connection
from google.cloud import pubsub_v1 # For PubSub Client
from google.cloud import bigquery # For BigQuery Client
from google.cloud import storage # For Cloud Storage Client
import json # For Message syntax
```

In [3]:
```
#Setting up the credential file
cred = service_account.Credentials.from_service_account_file('Dataset and Schema/AppleCred.json')
```

In [5]:
```
#Setting up the Configuration Variables:
project_id = "team-246422"
bucket_name = "applebucket"
topic_name = "AppleRevenue"
subscription_name = "AppleRevenue"
dataset_name = "Apple"
table_name = "Revenue"
```

In [6]:
```
#Realtime publish

import pandas as pd
import numpy as np
```

In [7]:
```
data_file = r'Dataset and Schema\Dataset and Schema\revenue-2013-2018.xlsx'
```

In [8]:
```
data = pd.read_excel(data_file)
```

In [9]:
```
data.shape
```

Out[9]: (6, 3)

In [39]:
```
data.head()
```

Out[39]:

	Year	Revenue_Spotify	Revenue_Apple
0	2023	800	660
1	2024	778	777

In [40]:
```
def default(o):
    if isinstance(o, np.int64): return int(o)
    raise TypeError
```

In [41]:
```
publisher = pubsub_v1.PublisherClient(credentials=cred)
```

In [42]:
```
client = pubsub_v1.PublisherClient(credentials=cred)
topic_path = publisher.topic_path(project_id, topic_name)
print(topic_path)
```

```
projects/team-246422/topics/AppleRevenue
```

In [43]:
```
subscriber = pubsub_v1.SubscriberClient(credentials = cred)
topic_path = subscriber.topic_path(project_id, topic_name)
subscription_path = subscriber.subscription_path(project_id, subscription_name)
```

In [44]:
```
for i in range(0,data.shape[0]):
    year = data['Year'][i]
    revenue_sp = data['Revenue_Spotify'][i]
    revenue_ap = data['Revenue_Apple'][i]


    data_row = {"Year":year,"Revenue_Spotify":revenue_sp,"Revenue_Apple":revenue_ap}
    #print(data_row)
    message_data = json.dumps(data_row, default=default)
    message_data = message_data.encode('utf-8')



    print(message_data)
    #Publishing a message on the PubSub Topic Created:
    response = publisher.publish(topic_path, message_data , origin='python-sample')
    print(response)
```

```
b'{"Year": 2023, "Revenue_Spotify": 800, "Revenue_Apple": 660}'
<google.cloud.pubsub_v1.publisher.futures.Future object at 0x0000023DDC4B0A58>
b'{"Year": 2024, "Revenue_Spotify": 778, "Revenue_Apple": 777}'
<google.cloud.pubsub_v1.publisher.futures.Future object at 0x0000023DDC4B0E10>
```

In [45]:
```
response.result()
```

Out[45]: '434495482878072'

In [18]:
```
#message_data
```

In [12]:
```
#topics = publisher.list_topics('team-246422')
```