Music Recommendation System

Milestone 1

Problem Definition

The context - Why is this problem important to solve?
The objectives - What is the intended goal?
The key questions - What are the key questions that need to be answered?
The problem formulation - What is it that we are trying to solve using data science?

Data Dictionary

The core data is the Taste Profile Subset released by The Echo Nest as part of the Million Song Dataset. There are two files in this dataset. One contains the details about the song id, titles, release, artist name and the year of release. Second file contains the user id, song id and the play count of users.

song_data

song_id - A unique id given to every song

title - Title of the song

Release - Name of the released album

Artist_name - Name of the artist

year - Year of release

count_data

user _id - A unique id given to the user

song_id - A unique id given to the song

play_count - Number of times the song was played

Data Source

http://millionsongdataset.com/

Important Notes

- This notebook can be considered a guide to refer to while solving the problem. The evaluation will be as per the Rubric shared for each Milestone. Unlike previous courses, it does not follow the pattern of the graded questions in different sections. This notebook would give you a direction on what steps need to be taken in order to get a viable solution to the problem. Please note that this is just one way of doing this. There can be other 'creative' ways to solve the problem and we urge you to feel free and explore them as an 'optional' exercise.
- In the notebook, there are markdown cells called - Observations and Insights. It is a good practice to provide observations and extract insights from the outputs.
- The naming convention for different variables can vary. Please consider the code provided in this notebook as a sample code.
- All the outputs in the notebook are just for reference and can be different if you follow a different approach.
- There are sections called **Think About It** in the notebook that will help you get a better understanding of the reasoning behind a particular technique/step. Interested learners can take alternative approaches if they want to explore different techniques.

Importing Libraries and the Dataset

In [ ]:
```
#Mounting the drive
from google.colab import drive
drive.mount('/content/drive')
```

In [ ]:
```
import warnings #Used to ignore the warning given as output of the code.
warnings.filterwarnings('ignore')

import numpy as np # Basic libraries of python for numeric and dataframe computations.
import pandas as pd

import matplotlib.pyplot as plt #Basic library for data visualization.
import seaborn as sns #Slightly advanced library for data visualization

from sklearn.metrics.pairwise import cosine_similarity #To compute the cosine similarity between two vectors.
from collections import defaultdict #A dictionary output that does not raise a key error

from sklearn.metrics import mean_squared_error # A performance metrics in sklearn.
```

In [ ]:
```
#importing the datasets
count_df = pd.read_csv('/content/drive/MyDrive/Capstone Project - Recommendation
Systems/capstone/count_data.csv')
song_df = pd.read_csv('/content/drive/MyDrive/Capstone Project - Recommendation Systems/song_data.csv')
```

Understanding the data by viewing a few observations

In [ ]:
```
# See top 10 records of count_df data
```

In [ ]:
```
# See top 10 records of song_df data
```

Let us check the data types and and missing values of each column

In [ ]:
```
# See the info of the count_df data
```

In [ ]:
```
# See the info of the song_df data
```

Observations and Insights:_

In [ ]:
```
# Left merge the count_df and song_df data on "song_id". Drop duplicates from song_df data simultaneously.


# Drop the column 'Unnamed: 0'
```

Think About It: As the user_id and song_id are encrypted. Can they be encoded to numeric features?

In [ ]:
```
# Apply label encoding for "user_id" and "song_id"
```

Think About It: As the data also contains users who have listened to very few songs and vice versa, is it required to filter the data so that it contains users who have listened to a good count of songs and vice versa?

In [ ]:
```
# Get the column containing the users
users = df.user_id
# Create a dictionary from users to their number of songs
ratings_count = dict()
for user in users:
    # If we already have the user, just add 1 to their rating count
    if user in ratings_count:
        ratings_count[user] += 1
    # Otherwise, set their rating count to 1
    else:
        ratings_count[user] = 1
```

In [ ]:
```
# We want our users to have listened at least 90 songs
RATINGS_CUTOFF = 90
remove_users = []
for user, num_ratings in ratings_count.items():
    if num_ratings < RATINGS_CUTOFF:
        remove_users.append(user)
df = df.loc[~df.user_id.isin(remove_users)]
```

In [ ]:
```
# Get the column containing the songs
songs = df.song_id
# Create a dictionary from songs to their number of users
ratings_count = dict()
for song in songs:
    # If we already have the song, just add 1 to their rating count
    if song in ratings_count:
        ratings_count[song] += 1
    # Otherwise, set their rating count to 1
    else:
        ratings_count[song] = 1
```

In [ ]:
```
# We want our song to be listened by atleast 120 users to be considred
RATINGS_CUTOFF = 120
remove_songs = []
for song, num_ratings in ratings_count.items():
    if num_ratings < RATINGS_CUTOFF:
        remove_songs.append(song)
df_final= df.loc[~df.song_id.isin(remove_songs)]
```

In [ ]:
```
# Drop records with play_count more than(>) 5
df_final=_________
```

In [ ]:
```
# Check the shape of the data
```

Exploratory Data Analysis

Let's check the total number of unique users, songs, artists in the data

Total number of unique user id

In [ ]:
```
# Display total number of unique user_id
```

Total number of unique song id

In [ ]:
```
# Display total number of unique song_id
```

Total number of unique artists

In [ ]:
```
# Display total number of unique artists
```

Observations and Insights:__

Let's find out about the most interacted songs and interacted users

Most interacted songs

In [ ]:
```

```

Most interacted users

In [ ]:
```

```

Observations and Insights:___

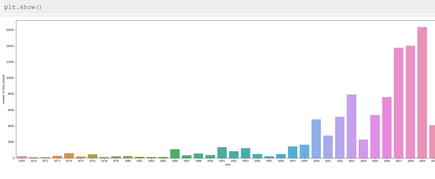
Songs played in a year

In [ ]:
```
count_songs = df_final.groupby('year').count()['title']
count = pd.DataFrame(count_songs)
count.drop(count.index[0], inplace=True)
count.tail()
```

Out [ ]:

year	title
2006	7592
2007	13750
2008	14031
2009	16351
2010	4087

In [ ]:
```
plt.figure(figsize=(30,10))
sns.barplot(x = count.index,
            y = 'title',
            data = count,
            estimator = np.median)
plt.ylabel('number of titles played')
# Show the plot
plt.show()
```



Observations and Insights:__

Think About It: What other insights can be drawn using exploratory data analysis?

Proposed approach

Potential techniques - What different techniques should be explored?
Overall solution design - What is the potential solution design?
Measures of success - What are the key measures of success to compare different potential techniques?