Spotify Music Recommender System

`Project By :` PRASAD JADHAV

In [ ]:
```
import pandas as pd

import nltk
from nltk.stem.porter import PorterStemmer

from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.metrics.pairwise import cosine_similarity
```

In [2]:
```
dataset = pd.read_csv('spotify_music.csv')
```

In [3]:
```
dataset.head()
```

Out[3]:

	artist	song	link	text
0	ABBA	Ahe's My Kind Of Girl	/a/abba/ahes+my+kind+of+girl_20598417.html	Look at her face, it's a wonderful face \r\nA...
1	ABBA	Andante, Andante	/a/abba/andante+andante_20002708.html	Take it easy with me, please \r\nTouch me gen...
2	ABBA	As Good As New	/a/abba/as+good+as+new_20003033.html	I'll never know why I had to go \r\nWhy I had...
3	ABBA	Bang	/a/abba/bang_20598415.html	Making somebody happy is a question of give an...
4	ABBA	Bang-A-Boomerang	/a/abba/bang+a+boomerang_20002668.html	Making somebody happy is a question of give an...

In [4]:
```
dataset.tail()
```

Out[4]:

	artist	song	link	text
57645	Ziggy Marley	Good Old Days	/z/ziggy+marley/good+old+days_10198588.html	Irie days come on play \r\nLet the angels fly...
57646	Ziggy Marley	Hand To Mouth	/z/ziggy+marley/hand+to+mouth_20531167.html	Power to the workers \r\nMore power \r\nPowe...
57647	Zwan	Come With Me	/z/zwan/come+with+me_20148981.html	all you need \r\nis something i'll believe \...
57648	Zwan	Desire	/z/zwan/desire_20148986.html	northern star \r\nam i frightened \r\nwhere ...
57649	Zwan	Heartsong	/z/zwan/heartsong_20148991.html	come in \r\nmake yourself at home \r\ni'm a ...

In [5]:
```
dataset.shape
```

Out[5]:
```
(57650, 4)
```

In [6]:
```
dataset.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 57650 entries, 0 to 57649
Data columns (total 4 columns):
 #   Column  Non-Null Count  Dtype 
---  ------  --------------  ----- 
 0   artist  57650 non-null  object
 1   song    57650 non-null  object
 2   link    57650 non-null  object
 3   text    57650 non-null  object
dtypes: object(4)
memory usage: 1.8+ MB
```

In [7]:
```
dataset.isnull().sum()
```

Out[7]:
```
artist    0
song      0
link      0
text      0
dtype: int64
```

In [8]:
```
dataset.duplicated().sum()
```

Out[8]:
```
0
```

In [9]:
```
dataset = dataset.sample(10000).drop('link', axis=1).reset_index(drop=True)
```

In [10]:
```
dataset['text'][0]
```

Out[10]:
```
"All we seem to do is talk about it  \r\nWe always end up shouting about it
\r\nThere was a time we could overcome it  \r\nBut it's too late to say we'l
l just forget it  \r\nIt's too bad that we had to break up  \r\nAnd too much
said for us to every make up  \r\nI could get by if I could just forget you
\r\nBut things remind me and I feel so sad now  \r\n  \r\nI could say I'm so
rry  \r\nBut it's not the point is it?  \r\nYou want to play your games and
\r\nYou don't mind if I get hurt  \r\n  \r\nSame old feeling every time I se
e you  \r\nAnd every avenue I walk I'm behind you  \r\nYour back is turned a
nd your eyes are closed girl  \r\nYou move in circles that are out of my rea
ch now\r\n\r\n"
```

In [11]:
```
dataset['text'] = dataset['text'].str.lower().replace(r'^\w\s', ' ').replace(
```

In [12]:
```
stemmer = PorterStemmer()

def tokenization(txt):
    tokens = nltk.word_tokenize(txt)
    stemming = [stemmer.stem(w) for w in tokens]
    return ' '.join(stemming)
```

In [13]:
```
dataset['text'] = dataset['text'].apply(lambda x: tokenization(x))
```

In [14]:
```
tfidvector = TfidfVectorizer(analyzer='word', stop_words='english')
matrix = tfidvector.fit_transform(dataset['text'])
similarity = cosine_similarity(matrix)
```

In [15]:
```
similarity[0]
```

Out[15]:
```
array([1.        , 0.02189188, 0.12461475, ..., 0.05715449, 0.07917509,
       0.01350715])
```

In [18]:
```
dataset[dataset['song'] == 'Money'].index[0]
```

Out[18]:
```
1
```

In [19]:
```
def recommendation(song_dataset):
    idx = dataset[dataset['song'] == song_dataset].index[0]
    distances = sorted(list(enumerate(similarity[idx])), reverse=True, key=la

    songs = []
    for m_id in distances[1:11]:
        songs.append(dataset.iloc[m_id[0]].song)

    return songs
```

In [20]:
```
recommendation('Money')
```

Out[20]:
```
['Mo Money',
 'Money Makes Her Smile',
 'For The Love Of Money',
 'Money',
 'Holiday',
 'Easy Money',
 'The Big Money',
 'Sweetest Girl (Dollar Bill)',
 "I'd Rather Be Rich",
 'Silas Stingy']
```

In [23]:
```
import pickle

pickle.dump(similarity,open('similarity.pkl','wb'))
pickle.dump(dataset,open('df.pkl','wb'))
```

Thank You!