Billboard Hot 100 Analysis &

Machine Learning Project

Name:

- Cesario Angel Ibarra

Details

Files and Data Dictionary

The files, are:

- chart.csv:
- Hot 100 Audio Features.csv:

The data dictionary for the columns in the files are:

- SongID Performer Song spotify_track_duration_ms danceability energy key loudness mode speechiness acousticness instrumentalness liveness valence tempo time_signature spotify_track_popularity date rank last-week peak-rank weeks-on-board

Data Cleaning

In order to clean the data, you will need to perform the following steps:

- Import chart.csv and Hot 100 Audio Features.csv
- Merge, Drop NaN, and Drop unecessary columns
- Convert mode and key columns for Visual Analysis
- Convert the duration of tracks from miliseconds to seconds, to make the interpretation more straightforward
- For further analysis we create a new column "song_performer"
- Next, create a year variable. In this case the format of the date variable has to be changed and then the year can be extracted
- Remove duplicates from song_performer column since the same song shows up multiple times
- Pickle df for Part 2

In [1]:
```
import matplotlib.pyplot as plt
import matplotlib as mpl
from matplotlib import cm
import numpy as np
import pandas as pd
import os
%matplotlib inline
mpl.rc('axes', labelsize=14)
mpl.rc('xtick', labelsize=12)
mpl.rc('ytick', labelsize=12)
```

```
# Where to save the figures
PROJECT_ROOT_DIR = "."
FOLDER = "figures"
IMAGES_PATH = os.path.join(PROJECT_ROOT_DIR, FOLDER)
os.makedirs(IMAGES_PATH, exist_ok=True)

def save_fig(fig_id, tight_layout=True, fig_extension="png", resolution=300):
    path = os.path.join(IMAGES_PATH, fig_id + "." + fig_extension)
    print("Saving figure", fig_id)
    if tight_layout:
        plt.tight_layout()
    plt.savefig(path, format=fig_extension, dpi=resolution)

# Set columns view to max
pd.set_option('display.max_columns', None)
```

Import Data

In [2]:
```
# Import files
file1 = 'charts.csv'
file2 = 'Hot 100 Audio Features.xlsx'
# DF's
charts_df = pd.read_csv(file1)
audiofeat_df = pd.read_excel(file2)
```

In [3]:
```
# View df's and rename columns for both
charts_df = charts_df.rename(columns={'artist': 'Performer'})
```

In [4]:
```
charts_df = charts_df.rename(columns={'song': 'Song'})
```

Merge DF's

In [5]:
```
hot100_df1 = audiofeat_df.merge(charts_df, on=['Song', 'Performer'])
hot100_df1
```

Out[5]:

	SongID	Performer	Song	spotify_genre	spotify_track_id	spotify_track_preview_url	spotify
0	-twistin'-White Silver SandsBill Black's Combo	Bill Black's Combo	-twistin'-White Silver Sands	[]	NaN	NaN	
1	-twistin'-White Silver SandsBill Black's Combo	Bill Black's Combo	-twistin'-White Silver Sands	[]	NaN	NaN	
2	¿Dònde Està Santa Claus? (Where Is Santa Claus...	Augie Rios	¿Dònde Està Santa Claus? (Where Is Santa Claus?)	['novelty']	NaN	NaN	

3	¿Dònde Està Santa Claus? (Where Is Santa Claus...	Augie Rios	¿Dònde Està Santa Claus? (Where Is Santa Claus?)	['novelty']	NaN	NaN
4	¿Dònde Està Santa Claus? (Where Is Santa Claus...	Augie Rios	¿Dònde Està Santa Claus? (Where Is Santa Claus?)	['novelty']	NaN	NaN
...	...	...	...	...	...	...
328981	Zunga ZengK7	K7	Zunga Zeng	['freestyle']	0XevPPcCBPovknaBw3lFvh	https://p.scdn.co/mp3-preview/8d5174aeb7d6b740...
328982	Zunga ZengK7	K7	Zunga Zeng	['freestyle']	0XevPPcCBPovknaBw3lFvh	https://p.scdn.co/mp3-preview/8d5174aeb7d6b740...
328983	Zunga ZengK7	K7	Zunga Zeng	['freestyle']	0XevPPcCBPovknaBw3lFvh	https://p.scdn.co/mp3-preview/8d5174aeb7d6b740...
328984	Zunga ZengK7	K7	Zunga Zeng	['freestyle']	0XevPPcCBPovknaBw3lFvh	https://p.scdn.co/mp3-preview/8d5174aeb7d6b740...
328985	Zunga ZengK7	K7	Zunga Zeng	['freestyle']	0XevPPcCBPovknaBw3lFvh	https://p.scdn.co/mp3-preview/8d5174aeb7d6b740...

328986 rows × 27 columns

In [6]:
```python
hot100_df1.to_pickle('hot100_total_unclean.pkl')
```

Drop NaN and Drop unecessary columns

In [7]:
```python
# Drop missing values
hot100_df1 = hot100_df1.dropna()
# Drop unecessary columns
hot100_df1 = hot100_df1.drop(['SongID','spotify_genre','spotify_track_id',
                              'spotify_track_preview_url',
                              'spotify_track_explicit',
                              'spotify_track_album',
                              'spotify_track_popularity'], axis=1)
```

In [8]:
```python
hot100_df1_Copy = hot100_df1.copy()
```

In [9]:
```python
hot100_df1_Copy = hot100_df1_Copy.rename(columns={'Performer':'performer',
                                          'Song':'song',
                                          'spotify_track_duration_ms':'track_dur
                                          'last-week':'last_week',
                                          'peak-rank':'peak_rank',
                                          'weeks-on-board':'weeks_on_board'})
```

In [10]:
```python
hot100_df1_Copy
```

Out[10]:

	performer	song	track_duration_s	danceability	energy	key	loudness	mode	speechiness	acousticne
6	Andy Williams	......And Roses	166106.0	0.154	0.185	5.0	-14.063	1.0	0.0315	0.91

		And Roses								
7	Andy Williams	......And Roses And Roses	166106.0	0.154	0.185	5.0	-14.063	1.0	0.0315	0.91
8	Andy Williams	......And Roses And Roses	166106.0	0.154	0.185	5.0	-14.063	1.0	0.0315	0.91
9	Andy Williams	......And Roses And Roses	166106.0	0.154	0.185	5.0	-14.063	1.0	0.0315	0.91
10	Andy Williams	......And Roses And Roses	166106.0	0.154	0.185	5.0	-14.063	1.0	0.0315	0.91
...	...	...	...	...	...	...	...	...	...	
328980	K7	Zunga Zeng	273000.0	0.846	0.657	1.0	-9.642	1.0	0.1400	0.04
328981	K7	Zunga Zeng	273000.0	0.846	0.657	1.0	-9.642	1.0	0.1400	0.04
328982	K7	Zunga Zeng	273000.0	0.846	0.657	1.0	-9.642	1.0	0.1400	0.04
328983	K7	Zunga Zeng	273000.0	0.846	0.657	1.0	-9.642	1.0	0.1400	0.04
328984	K7	Zunga Zeng	273000.0	0.846	0.657	1.0	-9.642	1.0	0.1400	0.04

152077 rows × 20 columns

In [11]:
```
# Convert key and mode column values from floats to intergers
hot100_df1_Copy['key'] = hot100_df1_Copy['key'].astype(int)
hot100_df1_Copy['mode'] = hot100_df1_Copy['mode'].astype(int)
```

Convert mode and key columns for Visual Analysis

-use Copy df and save hot100df1 for machine learning portion

In [12]:
```
# Convert the key column from numbers to note name labels
note_names = ["C", "C#", "D", "D#", "E", "F", "F#", "G", "G#", "A", "A#", "B"]
hot100_df1_Copy['key'] = hot100_df1_Copy['key'].map(lambda x: note_names[x])

# Convert the mode column to a factor with labeled levels
mode_labels = {0: 'minor', 1: 'major'}
hot100_df1_Copy['mode'] = hot100_df1_Copy['mode'].map(mode_labels)
```

In [13]:
```
# Create new column with key and mode columns
hot100_df1_Copy['key_signature'] = hot100_df1_Copy['key'].str.cat(hot100_df1_Copy['mode'
hot100_df1_Copy
```

Out[13]:

	performer	song	track_duration_s	danceability	energy	key	loudness	mode	speechiness	acousticne
6	Andy	......And	166106.0	0.154	0.185	F	-14.063	major	0.0315	0.91

	Williams	Roses And Roses								
7	Andy Williams	......And Roses And Roses	166106.0	0.154	0.185	F	-14.063	major	0.0315	0.91
8	Andy Williams	......And Roses And Roses	166106.0	0.154	0.185	F	-14.063	major	0.0315	0.91
9	Andy Williams	......And Roses And Roses	166106.0	0.154	0.185	F	-14.063	major	0.0315	0.91
10	Andy Williams	......And Roses And Roses	166106.0	0.154	0.185	F	-14.063	major	0.0315	0.91
...	...	...	...	...	...	...	...	...	...	
328980	K7	Zunga Zeng	273000.0	0.846	0.657	C#	-9.642	major	0.1400	0.04
328981	K7	Zunga Zeng	273000.0	0.846	0.657	C#	-9.642	major	0.1400	0.04
328982	K7	Zunga Zeng	273000.0	0.846	0.657	C#	-9.642	major	0.1400	0.04
328983	K7	Zunga Zeng	273000.0	0.846	0.657	C#	-9.642	major	0.1400	0.04
328984	K7	Zunga Zeng	273000.0	0.846	0.657	C#	-9.642	major	0.1400	0.04

152077 rows × 21 columns

Convert the duration of tracks from miliseconds to seconds, to make the interpretation more straightforward.

```
In [14]: hot100_df1_Copy['track_duration_s'] = hot100_df1_Copy['track_duration_s'] / 1000

         hot100_df1_Copy
```

Out[14]:

	performer	song	track_duration_s	danceability	energy	key	loudness	mode	speechiness	acousticne
6	Andy Williams	......And Roses And Roses	166.106	0.154	0.185	F	-14.063	major	0.0315	0.91
7	Andy Williams	......And Roses And Roses	166.106	0.154	0.185	F	-14.063	major	0.0315	0.91
8	Andy Williams	......And Roses And Roses	166.106	0.154	0.185	F	-14.063	major	0.0315	0.91

9	Andy Williams	......And Roses And Roses	166.106	0.154	0.185	F	-14.063	major	0.0315	0.91
10	Andy Williams	......And Roses And Roses	166.106	0.154	0.185	F	-14.063	major	0.0315	0.91
...	...	...	...	...	...	...	...	...	...	
328980	K7	Zunga Zeng	273.000	0.846	0.657	C#	-9.642	major	0.1400	0.04
328981	K7	Zunga Zeng	273.000	0.846	0.657	C#	-9.642	major	0.1400	0.04
328982	K7	Zunga Zeng	273.000	0.846	0.657	C#	-9.642	major	0.1400	0.04
328983	K7	Zunga Zeng	273.000	0.846	0.657	C#	-9.642	major	0.1400	0.04
328984	K7	Zunga Zeng	273.000	0.846	0.657	C#	-9.642	major	0.1400	0.04

152077 rows × 21 columns

For further analysis we can also create a new column “song_performer”

In [15]:
```
hot100_df1_Copy['song_performer'] = hot100_df1_Copy['song'].str.cat(hot100_df1_Copy['per
cols = list(hot100_df1_Copy.columns)
cols.insert(cols.index('song'), cols.pop(cols.index('song_performer')))
hot100_df1_Copy = hot100_df1_Copy[cols]
```

Next, create a year variable. In this case the format of the date variable has to be changed and then the year can be extracted.

In [16]:
```
hot100_df1_Copy['date'] = pd.to_datetime(hot100_df1_Copy['date'], format='%m/%d/%Y')
hot100_df1_Copy['year'] = hot100_df1_Copy['date'].dt.year
```

Remove duplicates from song_performer column since the same song shows up multiple times

In [17]:
```
hot100df_distinct = hot100_df1_Copy.drop_duplicates(subset='song_performer', keep='first
hot100df_distinct
```

Out[17]:

	performer	song_performer	song	track_duration_s	danceability	energy	key	loudness	mode	speec
6	Andy Williams	......And Roses And Roses | Andy Williams	......And Roses And Roses	166.106	0.154	0.185	F	-14.063	major	
17	Britney Spears	...Baby One More Time | Britney Spears	...Baby One More Time	211.066	0.759	0.699	C	-5.745	minor	
91	Paul Davis	'65 Love Affair |	'65	219.813	0.647	0.686	D	-4.247	minor	

		Paul Davis	Love Affair						
118	Tammy Wynette	'til I Can Make It On My Own | Tammy Wynette	'til I Can Make It On My Own	182.080	0.450	0.294	G	-12.022	major
132	Luther Vandross	'Til My Baby Comes Home | Luther Vandross	'Til My Baby Comes Home	332.226	0.804	0.714	B	-6.714	minor
...	...	...	...	...	...	...	...	...	...
328894	The Trammps	Zing Went The Strings Of My Heart | The Trammps	Zing Went The Strings Of My Heart	202.693	0.667	0.851	E	-5.257	major
328907	The Five Americans	Zip Code | The Five Americans	Zip Code	175.040	0.393	0.594	A	-5.986	major
328928	Bad Wolves	Zombie | Bad Wolves	Zombie	254.805	0.448	0.826	D	-3.244	minor
328959	Herb Alpert & The Tijuana Brass	Zorba The Greek | Herb Alpert & The Tijuana Brass	Zorba The Greek	264.853	0.531	0.642	F	-12.702	major
328971	K7	Zunga Zeng | K7	Zunga Zeng	273.000	0.846	0.657	C#	-9.642	major

13058 rows × 23 columns

BAM cleaning is complete

- check dtypes
- check for missing values just in case
- reset index

In [18]:
```
# check hot100df_distinct dtypes and missing values
print(hot100df_distinct.dtypes)
print(hot100df_distinct.isnull().sum())
```

```
performer                     object
song_performer                object
song                          object
track_duration_s             float64
danceability                 float64
energy                       float64
key                           object
loudness                     float64
mode                          object
speechiness                  float64
acousticness                 float64
instrumentalness             float64
liveness                     float64
valence                      float64
```

```
tempo                          float64
time_signature                 float64
date                    datetime64[ns]
rank                             int64
last_week                      float64
peak_rank                        int64
weeks_on_board                   int64
key_signature                   object
year                             int64
dtype: object
performer           0
song_performer      0
song                0
track_duration_s    0
danceability        0
energy              0
key                 0
loudness            0
mode                0
speechiness         0
acousticness        0
instrumentalness    0
liveness            0
valence             0
tempo               0
time_signature      0
date                0
rank                0
last_week           0
peak_rank           0
weeks_on_board      0
key_signature       0
year                0
dtype: int64
```

In [19]:
```
hot100df_distinct = hot100df_distinct.reset_index(drop=True)
hot100df_distinct
```

Out[19]:

	performer	song_performer	song	track_duration_s	danceability	energy	key	loudness	mode	speech
0	Andy Williams	......And Roses And Roses | Andy Williams	......And Roses And Roses	166.106	0.154	0.185	F	-14.063	major	0
1	Britney Spears	...Baby One More Time | Britney Spears	...Baby One More Time	211.066	0.759	0.699	C	-5.745	minor	0
2	Paul Davis	'65 Love Affair | Paul Davis	'65 Love Affair	219.813	0.647	0.686	D	-4.247	minor	0
3	Tammy Wynette	'til I Can Make It On My Own | Tammy Wynette	'til I Can Make It On My Own	182.080	0.450	0.294	G	-12.022	major	0
4	Luther Vandross	'Til My Baby Comes Home | Luther Vandross	'Til My Baby Comes Home	332.226	0.804	0.714	B	-6.714	minor	0
...	...	...	...	...	...	...	...	...	...	

13053	The Trammps	Zing Went The Strings Of My Heart | The Trammps	Zing Went The Strings Of My Heart	202.693	0.667	0.851	E	-5.257	major	0
13054	The Five Americans	Zip Code | The Five Americans	Zip Code	175.040	0.393	0.594	A	-5.986	major	0
13055	Bad Wolves	Zombie | Bad Wolves	Zombie	254.805	0.448	0.826	D	-3.244	minor	0
13056	Herb Alpert & The Tijuana Brass	Zorba The Greek | Herb Alpert & The Tijuana Brass	Zorba The Greek	264.853	0.531	0.642	F	-12.702	major	0
13057	K7	Zunga Zeng | K7	Zunga Zeng	273.000	0.846	0.657	C#	-9.642	major	0

13058 rows × 23 columns

```
In [20]: # Pickle df and save it to a file
         hot100df_distinct.to_pickle('hot100df_distinct.pkl')
```