

MACHINE LEARNING - SCREENSHOTS OF RESULTS

Student ID: 700725723

Student Name: Navasahitha

Subject Code: CS5710

CRN: 12675

The image displays two screenshots of a Jupyter Notebook titled "Spotify_song_popularity_prediction (1)".

Top Screenshot: The notebook is at the "2. Reading the Data" section. The code cell shows the following:

```
# Importing our libraries
# ---
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
%matplotlib inline

# To display all columns
pd.set_option('display.max.columns', None)

# To display the entire contents of a cell
pd.set_option('display.max_colwidth', None)

# Dataset url = https://bit.ly/spotifyongsds
# ---
df = pd.read_csv("https://bit.ly/spotifyongsds")

# Checking the first 5 rows of data
# ---
df.head()
```

The output shows the first 5 rows of the dataset:

track_id	track_name	track_artist	track_popularity	track_album_id	track_album_name	track_album_release_date	playlist
6	I Don't Care (with Justin Bieber) - Justin Bieber	Justin Bieber	86	2uCu0DST80660H428G0e	I Don't Care (with Justin Bieber) (Justin Bieber)	2019-06-14	Pop
1	Memories - Maroon 5	Maroon 5	87	5CMF0C8H4u9V14885z9d8	Memories (Maroon 5)	2019-12-12	Pop
2	All the Time - Zara Larsson	Zara Larsson	70	1Hd8m9dL8u9V14885z9d8	All the Time (Zara Larsson)	2019-07-05	Pop
3	Call You Mine - The Chainsmokers	The Chainsmokers	80	1u9V14885z9d8	Call You Mine - The Chainsmokers	2019-07-19	Pop
4	Someone You Loved - Lewis Capaldi	Lewis Capaldi	89	1u9V14885z9d8	Someone You Loved (Lewis Capaldi)	2019-03-08	Pop

Bottom Screenshot: The notebook is at the "3. Checking the last 5 rows of data" section. The code cell shows the following:

```
df = pd.read_csv("https://bit.ly/spotifyongsds")

# Checking the first 5 rows of data
df.head()

# Checking the last 5 rows of data
# ---
df.tail()
```

The output shows the last 5 rows of the dataset:

track_id	track_name	track_artist	track_popularity	track_album_id	track_album_name	track_album_release_date	playlist
6	I Don't Care (with Justin Bieber) - Justin Bieber	Justin Bieber	86	2uCu0DST80660H428G0e	I Don't Care (with Justin Bieber) (Justin Bieber)	2019-06-14	Pop
1	Memories - Maroon 5	Maroon 5	87	5CMF0C8H4u9V14885z9d8	Memories (Maroon 5)	2019-12-12	Pop
2	All the Time - Zara Larsson	Zara Larsson	70	1Hd8m9dL8u9V14885z9d8	All the Time (Zara Larsson)	2019-07-05	Pop
3	Call You Mine - The Chainsmokers	The Chainsmokers	80	1u9V14885z9d8	Call You Mine - The Chainsmokers	2019-07-19	Pop
4	Someone You Loved - Lewis Capaldi	Lewis Capaldi	89	1u9V14885z9d8	Someone You Loved (Lewis Capaldi)	2019-03-08	Pop

localhost:8888/notebooks/Downloads/spotify_song_popularity_prediction/spotify_song_popularity_prediction.ipynb

jupyter Spotify_song_popularity_prediction (1) Last Checkpoint: 11/10/2022 (autosaved)

File Edit View Insert Cell Kernel Widgets Help Not Trusted Python 3 (ipykernel)

23855 2b7d9b3a7b3b2c2p0b0 Typhoon - Original Mix Julian Casar 27 633m0c0b5b0f7Poc0b0c0 Typhoon/Storm 2014-03-05

```
In [ ]: # Sample 10 rows of data
df.sample(10)
```

```
Out[5]:
```

track_id	track_name	track_artist	track_popularity	track_album_id	track_album_name	track_album_release_date
21086	SCM5N9u0f7wqy7QyV0H	Shells for You	37	5U4BT2a5M4k0y7b0c0	The Parish Runners EP	2015-04-08
5004	5e0a2pF08b0p0v0c0b0	Forward Backward	40	2452a5P5K0M7b0b0k0c0	Fixed July	2019-12-10
20194	5b0f7c0a2pV0V00c0b0c0	Love Robin	62	5f0a2pF08b0p0v0c0b0	Love Robin	2019-04-18
4025	7a7b7a0f0e0b7c0a2p	Lucky	75	5e0a2pF08b0p0v0c0b0	Lucky	2019-11-21
11216	7a7b7a0f0e0b7c0a2p	Cherry Bomb	65	5U4BT2a5M4k0y7b0c0	The Runaways	1976
30436	7a7b7a0f0e0b7c0a2p	My Love	34	1X2b0f55b0m7c0b0k0p0	Who Is Jill Scott? (Words And Sounds Volume 1)	2000-07-18
10844	2u0b0p0c0b0k0a2p0b0c0	Outta Me	51	60544440c0c0c0c0c0c0	Outta Me	2019-08-26
21124	2e7a0f0c0b0k0a2p0b0c0	Wanted - Unreleased Remix	30	7b0a2pF08b0p0v0c0b0	Wanted (Remixes)	2014-08-08
5257	2a0c0c0b0c0c0c0c0c0c0	Metal	35	5f0a2pF08b0p0v0c0b0	Artemis (Deluxe)	2019-02-02
1111	4b0a2pF08b0p0v0c0b0	Paris	63	1u0b0a2pF08b0p0v0c0b0	Before the Waves	2014-07-07

```
In [ ]: # Checking number of rows and columns
df.shape
```

localhost:8888/notebooks/Downloads/spotify_song_popularity_prediction/spotify_song_popularity_prediction.ipynb

jupyter Spotify_song_popularity_prediction (1) Last Checkpoint: 11/10/2022 (autosaved)

File Edit View Insert Cell Kernel Widgets Help Not Trusted Python 3 (ipykernel)

```
In [ ]: # checking number of rows and columns
df.shape
```

```
Out[6]: (1083, 10)
```

```
In [ ]: # Checking datatypes
# ---
df.dtypes
```

```
Out[7]:
```

track_id	object
track_name	object
track_artist	object
track_popularity	int64
track_album_id	object
track_album_name	object
track_album_release_date	object
playlist_name	object
playlist_id	object
playlist_genre	object
playlist_subgenre	object
danceability	float64
energy	float64
key	int64
loudness	float64
mode	int64
speechiness	float64
acousticness	float64
instrumentalness	float64
liveness	float64
valence	float64
tempo	float64
duration_ms	int64
dtype:	object

Downloaded Spotify_song_popularity_prediction (1) Last checkpoint: 11/10/2022 (auto saved)

File Edit View Insert Cell Kernel Widgets Help Not Trained Python 3 (ipykernel)

```
In [ ]: df.dtypes
```

```
Out[4]:
```

	track_popularity	danceability	energy	key	loudness	mode	speechiness	acousticness	instrumentalness	liveness
count	32833.000000	32833.000000	32833.000000	32833.000000	32833.000000	32833.000000	32833.000000	32833.000000	32833.000000	32833.000000
mean	42.477081	0.694880	0.698639	0.374471	-8.710489	0.888719	0.107068	0.175324	0.084747	0.190179
std	24.984074	0.145085	0.182910	2.071957	2.988436	0.499271	0.101214	0.279930	0.234230	0.154217
min	0.000000	0.000000	0.001176	0.000000	-16.448000	0.000000	0.000000	0.000000	0.000000	0.000000
25%	24.000000	0.560000	0.561000	2.000000	-8.171000	0.000000	0.041000	0.011000	0.000000	0.060700
50%	46.000000	0.670000	0.721000	6.000000	-6.198000	1.000000	0.062000	0.080000	0.000000	0.127000
75%	62.000000	0.761000	0.840000	9.000000	-4.645000	1.000000	0.110000	0.250000	0.004000	0.248000
max	100.000000	0.943000	1.000000	11.000000	1.276000	1.000000	0.518000	0.984000	0.984000	0.996000

From the dataset provided we are able to make the following observations:

- The data set has 32833 records and various used datatypes including object, float and integers.

3. External Data Source Validation

From our data I can confirm that the information about tracks in the dataset is valid because the data is evidenced by records from the Spotify mobile app.

4. Data Preparation

Downloaded Spotify_song_popularity_prediction (1) Last checkpoint: 11/10/2022 (auto saved)

File Edit View Insert Cell Kernel Widgets Help Not Trained Python 3 (ipykernel)

Performing Data exploration

```
In [ ]: # checking datatypes and missing values of all the variables
# ...
df.isnull().sum()
```

```
Out[4]:
```

track_id	0
track_name	5
track_artist	5
track_popularity	0
track_album_id	0
track_album_name	5
track_album_release_date	0
playlist_name	0
playlist_id	0
playlist_genre	0
playlist_subgenre	0
danceability	0
energy	0
key	0
loudness	0
mode	0
speechiness	0
acousticness	0
instrumentalness	0
liveness	0
valence	0
tempo	0
duration_ms	0
dtype: int64	

From our observation we observe that some columns contain some missing values but that's very negligible.

Download/ Spotify_song_popularity_prediction (1) Last Checkpoint 11/10/2022 (autosaved)

File Edit View Insert Cell Kernel Widgets Help New Trusted Python 3 (ipykernel)

From our observation we observe that some columns contain some missing values but that's very negligible.

```
In [ ]: # Missing values by percentage
missing_pct = df.isna().mean().round(4)*100
missing_pct
```

```
Out[18]:
```

track_id	0.00
track_name	0.01
track_artist	0.01
track_popularity	0.00
track_album_id	0.00
track_album_name	0.01
track_album_release_date	0.00
playlist_name	0.00
playlist_id	0.00
playlist_genre	0.00
playlist_subgenre	0.00
danceability	0.00
energy	0.00
key	0.00
loudness	0.00
mode	0.00
speechiness	0.00
acousticness	0.00
instrumentalness	0.00
liveness	0.00
valence	0.00
tempo	0.00
duration_ms	0.00
dtype: float64	

```
In [ ]: # Checking how many duplicate rows are there in the data
# ---
#
```

Download/ Spotify_song_popularity_prediction (1) Last Checkpoint 11/10/2022 (autosaved)

File Edit View Insert Cell Kernel Widgets Help New Trusted Python 3 (ipykernel)

```
In [ ]: # checking how many duplicate rows are there in the data
# ---
df.duplicated().sum()
```

```
Out[19]: 0
```

We observe the following from our dataset:

- There are no duplicated rows in the dataset

```
In [ ]: # checking if any of the columns are all null
# ---
df.isna().all()
```

```
Out[20]:
```

track_id	False
track_name	False
track_artist	False
track_popularity	False
track_album_id	False
track_album_name	False
track_album_release_date	False
playlist_name	False
playlist_id	False
playlist_genre	False
playlist_subgenre	False
danceability	False
energy	False
key	False
loudness	False
mode	False
speechiness	False
acousticness	False
instrumentalness	False
liveness	False
valence	False
tempo	False
duration_ms	False

Spotify_song_popularity_prediction (1) Last Checkpoint: 11/10/2022 (auto-saved)

File Edit View Insert Cell Kernel Widgets Help

Run

Cleaning the data

Remaining columns

```
In [ ]: # standardizing your dataset i.e. variable naming just to be sure
df.columns = df.columns.str.lower()
df.head()
```

```
Out[18]:
```

	track_id	track_name	track_artist	track_popularity	track_album_id	track_album_name	track_album_release_date	playlist
0	0B0W0eNdr0P0u0VM	I Don't Care (with Justin Bieber) - Louis Luxury Remix	Ed Sheeran	86	3u0u0D0T0R000H0Z00C0	I Don't Care (with Justin Bieber) [Louis Luxury Remix]	2019-06-14	Pop
1	0Y0A0Z0V0g0T0N0A0PH	Memories - Elton Francis Barnes	Maroon 5	67	63P0000A0u0g0Y0X0000u0v0	Memories [Elton Francis Barnes]	2019-12-13	Pop
2	1r0g0N0A0H0G0e0D0E0	All the Time - One Direction	Zoe Lomon	70	1r0g0N0A0H0G0e0D0g0T0k	All the Time (One Direction)	2019-07-06	Pop
3	75P000w0m0R0U0u0G0C0T	Call You Close - The Chainsmokers	The Chainsmokers	60	1n0g0N0A0H0G0e0D0e0D0	Call You Close - The Chainsmokers	2019-07-19	Pop
4	1u0P000G0Y0u0b0P0e0g0e0k	Someone You Loved - Future Humans	Leona Capaldi	69	1r0Y0e0N0G0R0F0u0R0Z00G0	Someone You Loved (Future Humans)	2019-03-06	Pop

Dropping rows with missing values

Spotify_song_popularity_prediction (1) Last Checkpoint: 11/10/2022 (auto-saved)

File Edit View Insert Cell Kernel Widgets Help

Run

Dropping rows with missing values

```
In [ ]: # df.dropna(inplace = True)
df.isna().sum()
```

```
Out[17]:
```

track_id	0
track_name	0
track_artist	0
track_popularity	0
track_album_id	0
track_album_name	0
track_album_release_date	0
playlist_name	0
playlist_id	0
playlist_genre	0
playlist_subgenre	0
danceability	0
energy	0
key	0
loudness	0
mode	0
speechiness	0
acousticness	0
instrumentalness	0
liveness	0
valence	0
tempo	0
duration_ms	0
dtype: int64	

Dropping irrelevant variables

```
In [ ]: # df.drop(columns = ["track_album_id", "playlist_id", "playlist_name", "track_album_name", "track_name"], inplace = True)
```


The screenshot shows a Jupyter Notebook titled "Spotify_song_popularity_prediction (1)". The interface includes a top bar with navigation icons, a file explorer, and a toolbar with various editing and execution tools. The notebook content consists of a table of data and a code cell.

id	track	artist	album	genre	year	popularity	duration	energy	danceability	valence	loudness	tempo	key	mode	instrumentalness	liveness	speechiness	acousticness	electricity	popularity
24114	24114	24114	24114	24114	24114	24114	24114	24114	24114	24114	24114	24114	24114	24114	24114	24114	24114	24114	24114	24114
9887	9887	9887	9887	9887	9887	9887	9887	9887	9887	9887	9887	9887	9887	9887	9887	9887	9887	9887	9887	9887
12390	12390	12390	12390	12390	12390	12390	12390	12390	12390	12390	12390	12390	12390	12390	12390	12390	12390	12390	12390	12390
29405	29405	29405	29405	29405	29405	29405	29405	29405	29405	29405	29405	29405	29405	29405	29405	29405	29405	29405	29405	29405
37427	37427	37427	37427	37427	37427	37427	37427	37427	37427	37427	37427	37427	37427	37427	37427	37427	37427	37427	37427	37427
888	888	888	888	888	888	888	888	888	888	888	888	888	888	888	888	888	888	888	888	888

```

In [ ]: # Installing the pandas-profiling package from
        ! pip install https://github.com/pandas-profiling/pandas-profiling/archive/master.zip

collecting https://github.com/pandas-profiling/pandas-profiling/archive/master.zip
using cached https://github.com/pandas-profiling/pandas-profiling/archive/master.zip
Requirement already satisfied (use --upgrade to upgrade): pandas-profiling==2.9.0 from https://github.com/pandas-profiling/pandas-profiling/archive/master.zip
Requirement already satisfied: joblib in /usr/local/lib/python3.6/dist-packages (from pandas-profiling==2.9.0) (0.16.0)
Requirement already satisfied: scipy<1.4.1 in /usr/local/lib/python3.6/dist-packages (from pandas-profiling==2.9.0) (1.4.1)
Requirement already satisfied: pandas<1.0.0,>=1.0.0.1,!=1.0.2,!=1.1.0,!=0.29.3 in /usr/local/lib/python3.6/dist-packages (from pandas-profiling==2.9.0) (1.0.5)
Requirement already satisfied: matplotlib<3.3.0 in /usr/local/lib/python3.6/dist-packages (from pandas-profiling==2.9.0) (3.2.0)
Requirement already satisfied: confusable==1.0 in /usr/local/lib/python3.6/dist-packages (from pandas-profiling==2.9.0) (1.0.0)
Requirement already satisfied: jieba==2.0.2 in /usr/local/lib/python3.6/dist-packages (from pandas-profiling==2.9.0) (2.0.2)
Requirement already satisfied: vision[type_image_path]<0.5.0 in /usr/local/lib/python3.6/dist-packages (from pandas-profiling==2.9.0) (0.5.0)
Requirement already satisfied: numpy<1.18.0 in /usr/local/lib/python3.6/dist-packages (from pandas-profiling==2.9.0) (1.16.5)

```

















GITHUB Link:

https://github.com/NXI57230/Project_700725723