Project Based Learning Report

on

Create a Spotify Music Analysis Visualization Using Python Pandas

Submitted in the partial fulfillment of the requirements
For the Project based learning in (**Essentials of Data Science**)
in
Electronics & Communication Engineering

By
Name of Students in Alphabetical order with Seat Number /PRN Number

PRN	Name of the Student
2214110421	Nilesh Kumar
2214110435	Shayan Kumar
2214110441	Vishal Kumar

Under the guidance of

Prof. Vikas Kaduskar

Department of Electronics & Communication Engineering

Bharati Vidyapeeth
(Deemed to be University)
College of Engineering,
Pune – 4110043

Academic Year: 2021-22

Bharati Vidyapeeth
(Deemed to be University)
College of Engineering,
Pune – 411043

DEPARTMENT OF ELECTRONICS & COMMUNICATION ENGINEERING

CERTIFICATE

Certified that the Project Based Learning report entitled, **"Create a Spotify Music Analysis Visualization using Python Pandas"** is work done by

PRN	Name of the Student
2214110421	Nilesh Kumar
2214110435	Shayan Kumar
2214110441	Vishal Kumar

in partial fulfillment of the requirements for the award of credits for Project Based Learning (PBL) in "**Essentials of Data Science"** of Bachelor of Technology Semester IV, in ECE Div-1.

Date:

Prof. Vikas Kaduskar — **Course In-charge**

Dr. Arundhati Shinde — **Professor & Head**

Problem statement:-

Create a Spotify Music Analysis Visualization using Python Pandas.

Solution :-

What is song recommendation?

Song recommendation refers to the process of suggesting songs to a listener based on their musical preferences, listening history, and other relevant factors. This can be done through various means, such as through music streaming platforms that use algorithms to suggest songs based on a listener's listening history, or through human-curated playlists that are created by music experts or enthusiasts.
Song recommendations are useful for discovering new music that a listener may enjoy, and can help to broaden their musical horizons. They can also be a way for artists to reach new audiences and gain exposure for their music.

What is role of data science in song recommendation?

Data science plays a crucial role in song recommendation. Music streaming services use data science techniques to analyze a user's listening behavior, such as the songs they listen to, the artists they follow, the playlists they create, and the genres they prefer. This data is then used to build a personalized music recommendation system.
Data science algorithms are used to identify patterns and relationships in the data that can be used to make predictions about what songs a user might enjoy. These algorithms use various techniques such as collaborative filtering, content-based filtering, and deep learning to make recommendations.

Collaborative filtering is a popular technique used in music recommendation systems, which involves analyzing the listening behavior of other users who have similar preferences to the current user. This helps to identify songs that the user may like but has not yet discovered.
Content-based filtering involves analyzing the characteristics of the music itself, such as the tempo, melody, and lyrics, and recommending similar songs to the user based on those characteristics.
Deep learning techniques are also used in music recommendation systems, which involve training neural networks on large datasets of music to learn patterns and relationships in the data, which can be used to make more accurate recommendations.

In summary, data science plays a crucial role in developing effective song recommendation systems that can provide personalized music recommendations to users, thereby enhancing their overall music streaming experience.

what are different data sciencve techniques in song recommendation systems with example?
There are several data science techniques that can be used in song recommendation systems. Here are some examples:

Collaborative Filtering: Collaborative filtering involves recommending songs based on the listening behavior of other users who have similar preferences to the current user. For example, if User A and User B have similar listening habits and both enjoy a certain song, the system may recommend that song to User A.
Content-Based Filtering: Content-based filtering involves recommending songs that are similar in characteristics to the songs that a user has previously enjoyed. For example, if a user enjoys songs with a fast tempo and upbeat melody, the system may recommend similar songs that match those characteristics.

Matrix Factorization: Matrix factorization is a technique used to identify hidden factors or features in a dataset. In the context of music recommendation systems, matrix factorization can be used to identify the underlying musical features that are important to a user, such as tempo, melody, and genre.

Deep Learning: Deep learning involves training neural networks on large datasets to learn patterns and relationships in the data. In music recommendation systems, deep learning can be used to analyze the characteristics of songs and identify patterns in the listening behavior of users. For example, a neural network could be trained to identify songs that are popular among users who have similar listening habits.

Hybrid Approaches: Hybrid approaches combine different data science techniques to provide more accurate recommendations. For example, a hybrid approach could combine collaborative filtering and content-based filtering to provide recommendations based on both user behavior and song characteristics.

In summary, there are several data science techniques that can be used in song recommendation systems, and the choice of technique depends on the specific requirements of the system and the type of data available.

Spotify music visuallisation:-

Spotify audio streaming and media services provider founded on 23 April 2006 by Daniel Ek and Martin Lorentzon. It is one of the largest music streaming service providers, with over 422 million monthly active users, including 182 million paying subscribers, as of March 2022..

Spotify offers digital copyright restricted recorded music and podcasts, including more than 82 million songs, from record labels and media companies. Spotify is currently available in 180+ countries, as of October 2021. Users can search for music based on artist, album, or genre, and can create, edit, and share playlists.

Unlike physical or download sales, which pay artists a fixed price per song or album sold, Spotify pays royalties based on the number of artist streams as a proportion of total songs streamed. It distributes approximately 70% of its total revenue to rights holders (often record labels), who then pay artists based on individual agreements. According to Ben Sisario of *The New York Times*, approximately 13,000 out of seven million artists on Spotify generated $50,000 or more in payments in 2020.

DATASET:-

We have downloaded dataset about spotify music from github.com site which is

Spotify_Dataset.csv.

We have performed analysis visulization on google collab.

Library used:-

For mathematical computation:-

1) Numpy library - numpy is used to perform various mathematical operations on arrays.
2) Pandas Library - pandas provides various data structures and operations for manipulating numerical data and time series.

3)Scipy-stats - All of the statistics functions are located in the sub-package scipy.stats and a fairly complete listing of these functions can be obtained using info(stats) function. A list of random variables available can also be obtained from the docstring for the stats sub-package.

For data visuallisation:-

1) Matplotlib library from which pyplot module is used for plotting library used for 2D graphics.
2) Seaborn library - seaborn is a library for making statistical graphics in Python.
3) Plotly - Plotly is a Montreal based technical computing company involved in development of data analytics and visualisation tools such as Dash and Chart Studio. It has also developed open source graphing Application Programming Interface (API) libraries for Python.

Software used is **Google Collab**

Google is quite aggressive in AI research. Over many years, Google developed AI framework called **TensorFlow** and a development tool called **Colaboratory**. Today TensorFlow is open-sourced and since 2017, Google made Colaboratory free for public use. Colaboratory is now known as Google Colab or simply **Colab**.

Colab is a free Jupyter notebook environment that runs entirely in the cloud. Most importantly, it does not require a setup and the notebooks that you create can be simultaneously edited by your

team members - just the way you edit documents in Google Docs. Colab supports many popular machine learning libraries which can be easily loaded in your notebook.

What Colab Offers us?

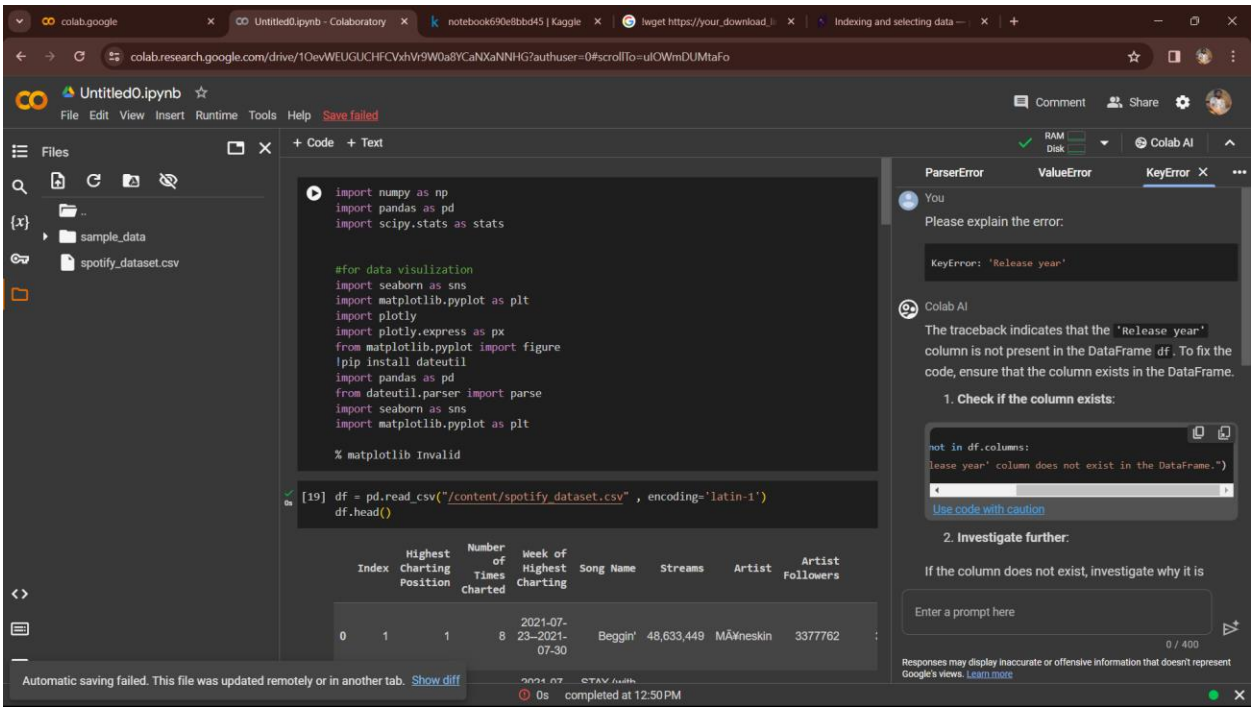
As a programmer, you can perform the following using Google Colab.

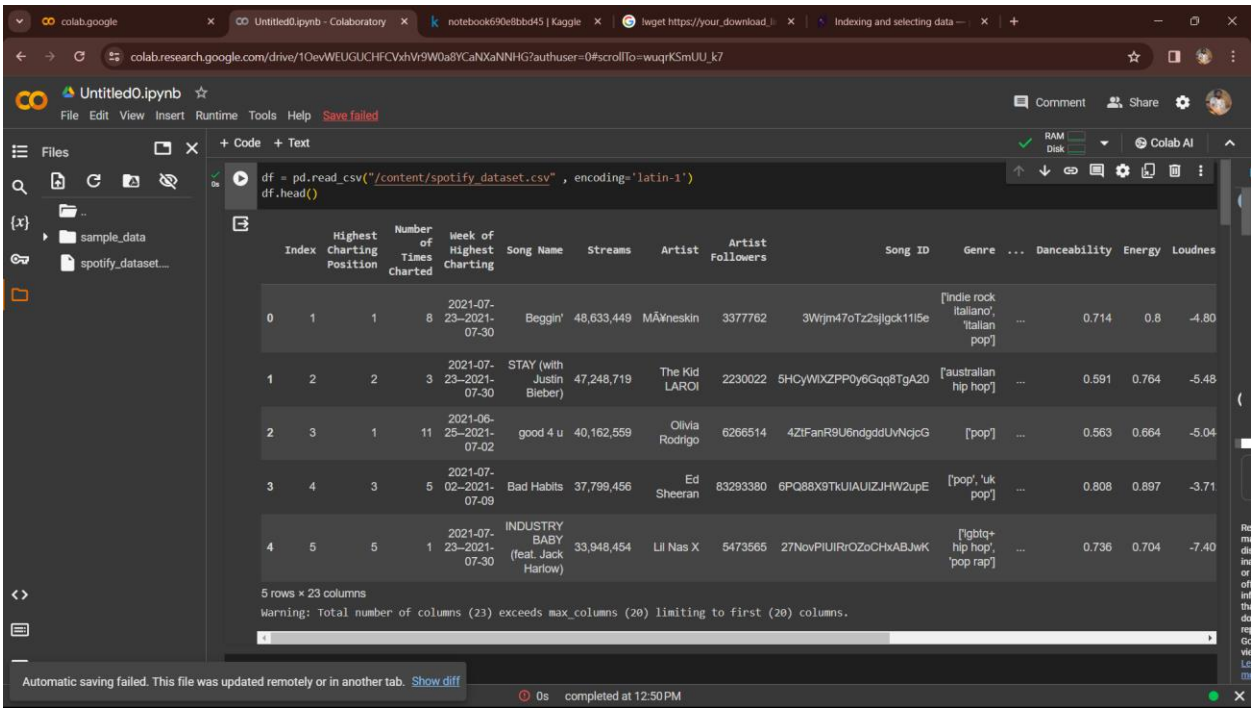
- Write and execute code in Python
- Document your code that supports mathematical equations
- Create/Upload/Share notebooks
- Import/Save notebooks from/to Google Drive
- Import/Publish notebooks from GitHub
- Import external datasets e.g. from Kaggle
- Integrate PyTorch, TensorFlow, Keras, OpenCV
- Free Cloud service with free GPU

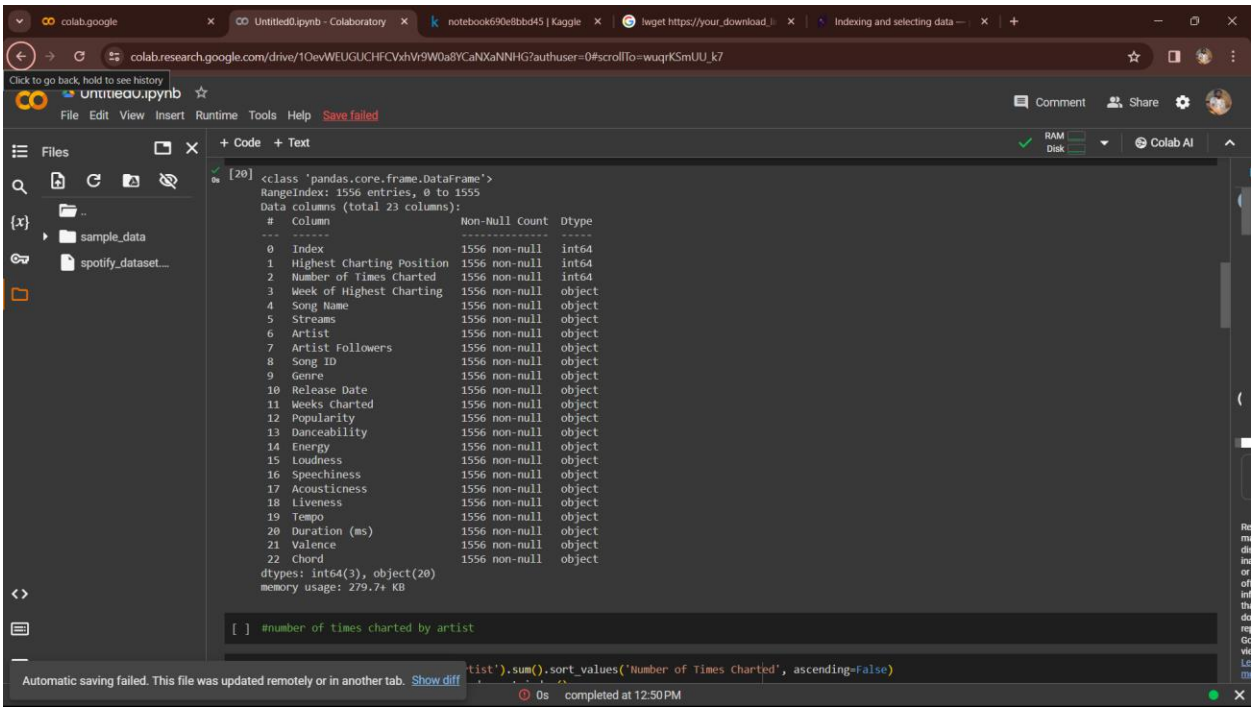
Result with analysis

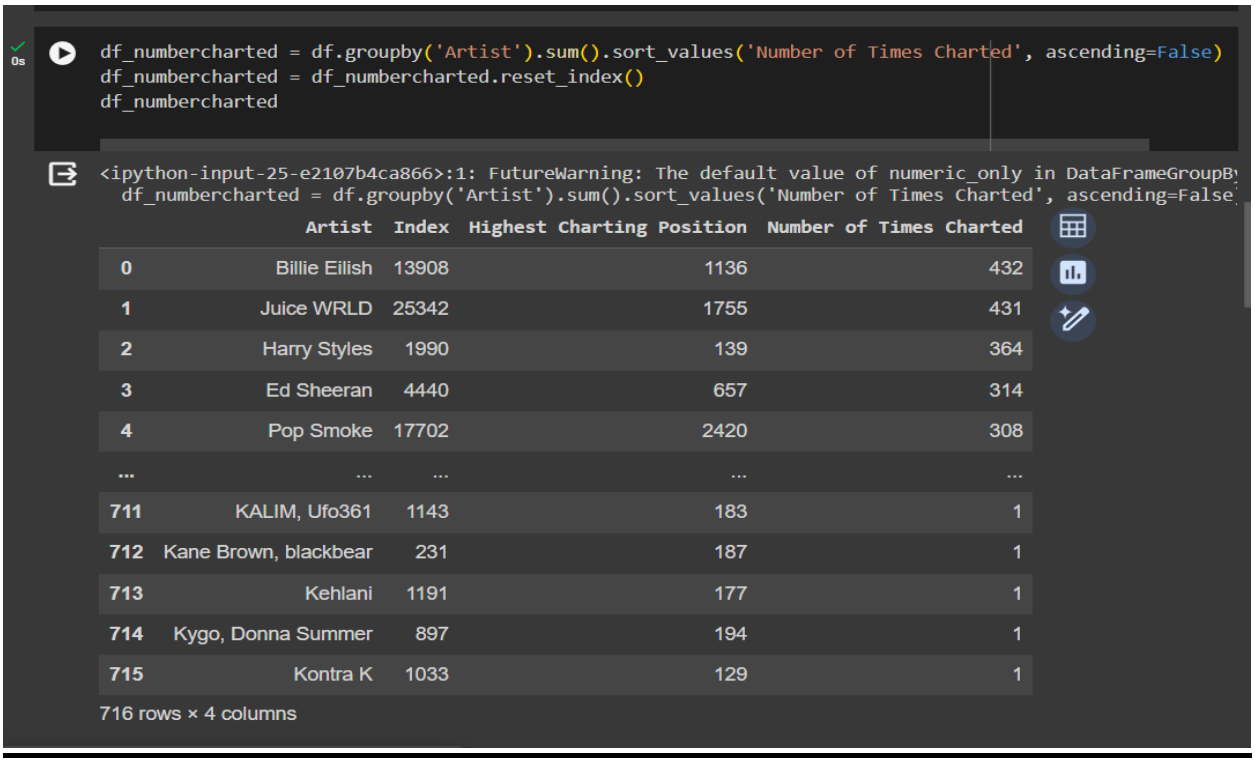
Analysis of the code: -

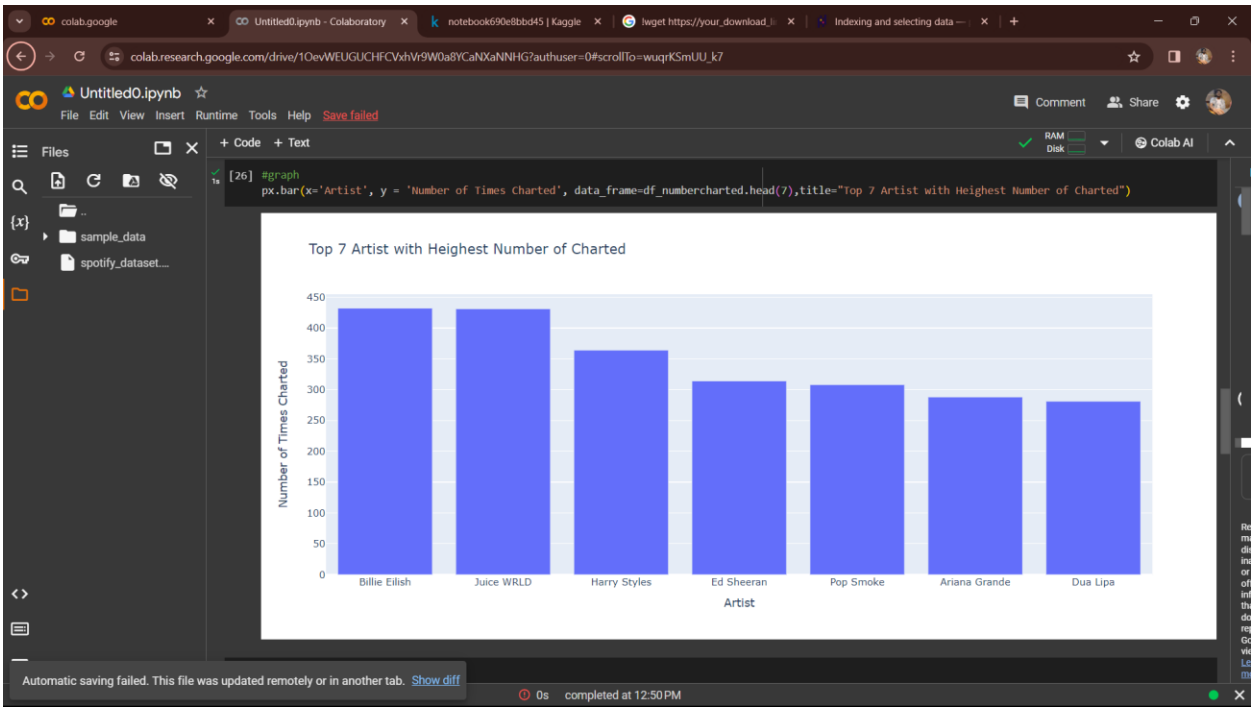
- First, we Import the libraries
- Secondly, Download the dataset and add that to the path to load the dataset. we use panda library and used head() function for displaying first five row of dataset.
- We get more information by using df.info().then for checking null values in dataset we used is null() function.
- Then we find the graph of number of time charted by artist by using px.bar() function.
- Then we create a correlation using heatmap()
- Then we use the library plotly to plot the graph of danceability by use px.line().
- Then we plot graph by using px.bar()
- At last we use pandas library to get information about genre and plot the pie chart.


colab.google
Untitled0.ipynb - Colaboratory
notebook690e8bbd45 | Kaggle
!wget https://your_download_l
Indexing and selecting data —
colab.research.google.com/drive/1OevWEUGUCHFCVxhVr9W0a8YCaNXaNNHG?authuser=0#scrollTo=ulOWmDUMtaFo
Untitled0.ipynb
File Edit View Insert Runtime Tools Help Save failed
Comment
Share
Files
sample_data
spotify_dataset.csv
+ Code + Text
RAM
Disk
Colab AI
import numpy as np
import pandas as pd
import scipy.stats as stats
#for data visulization
import seaborn as sns
import matplotlib.pyplot as plt
import plotly
import plotly.express as px
from matplotlib.pyplot import figure
!pip install dateutil
import pandas as pd
from dateutil.parser import parse
import seaborn as sns
import matplotlib.pyplot as plt
% matplotlib Invalid
[19] df = pd.read_csv("/content/spotify_dataset.csv" , encoding='latin-1')
df.head()
Index
Highest Charting Position
Number of Times Charted
Week of Highest Charting
Song Name
Streams
Artist
Artist Followers
0
1
1
8
2021-07-23--2021-07-30
Beggin'
48,633,449
MÃ¥neskin
3377762
ParserError
ValueError
KeyError
You
Please explain the error:
KeyError: 'Release year'
Colab AI
The traceback indicates that the 'Release year' column is not present in the DataFrame df. To fix the code, ensure that the column exists in the DataFrame.
1. Check if the column exists:
not in df.columns:
lease year' column does not exist in the DataFrame.")
Use code with caution
2. Investigate further:
If the column does not exist, investigate why it is
Enter a prompt here
0 / 400
Responses may display inaccurate or offensive information that doesn't represent Google's views. Learn more
Automatic saving failed. This file was updated remotely or in another tab. Show diff
0s completed at 12:50 PM


colab.google
Untitled0.ipynb - Colaboratory
notebook690e8bbd45 | Kaggle
!wget https://your_download_l
Indexing and selecting data —
colab.research.google.com/drive/1OevWEUGUCHFCVxhVr9W0a8YCaNXaNNHG?authuser=0#scrollTo=wuqrKSmUU_k7
Untitled0.ipynb
File Edit View Insert Runtime Tools Help Save failed
Comment
Share
Files
sample_data
spotify_dataset....
+ Code + Text
RAM
Disk
Colab AI
df = pd.read_csv("/content/spotify_dataset.csv" , encoding='latin-1')
df.head()
Index
Highest Charting Position
Number of Times Charted
Week of Highest Charting
Song Name
Streams
Artist
Artist Followers
Song ID
Genre
...
Danceability
Energy
Loudnes
0
1
1
8
2021-07-23--2021-07-30
Beggin'
48,633,449
MÃ¥neskin
3377762
3Wrjm47oTz2sjIgck11l5e
['indie rock italiano', 'italian pop']
...
0.714
0.8
-4.80
1
2
2
3
2021-07-23--2021-07-30
STAY (with Justin Bieber)
47,248,719
The Kid LAROI
2230022
5HCyWlXZPP0y6Gqq8TgA20
['australian hip hop']
...
0.591
0.764
-5.48
2
3
1
11
2021-06-25--2021-07-02
good 4 u
40,162,559
Olivia Rodrigo
6266514
4ZtFanR9U6ndgddUvNcjcG
['pop']
...
0.563
0.664
-5.04
3
4
3
5
2021-07-02--2021-07-09
Bad Habits
37,799,456
Ed Sheeran
83293380
6PQ88X9TkUIAUIZJHW2upE
['pop', 'uk pop']
...
0.808
0.897
-3.71
4
5
5
1
2021-07-23--2021-07-30
INDUSTRY BABY (feat. Jack Harlow)
33,948,454
Lil Nas X
5473565
27NovPIUIRrOZoCHxABJwK
['lgbtq+ hip hop', 'pop rap']
...
0.736
0.704
-7.40
5 rows × 23 columns
Warning: Total number of columns (23) exceeds max_columns (20) limiting to first (20) columns.
Automatic saving failed. This file was updated remotely or in another tab. Show diff
0s completed at 12:50 PM


colab.google
Untitled0.ipynb - Colaboratory
notebook690e8bbd45 | Kaggle
!wget https://your_download_l
Indexing and selecting data —
colab.research.google.com/drive/1OevWEUGUCHFCVxhVr9W0a8YCaNXaNNHG?authuser=0#scrollTo=wuqrKSmUU_k7
Click to go back, hold to see history
Untitled0.ipynb
File Edit View Insert Runtime Tools Help Save failed
Comment
Share
Files
+ Code + Text
RAM
Disk
Colab AI
sample_data
spotify_dataset....
[20] <class 'pandas.core.frame.DataFrame'>
RangeIndex: 1556 entries, 0 to 1555
Data columns (total 23 columns):
Column Non-Null Count Dtype
0 Index 1556 non-null int64
1 Highest Charting Position 1556 non-null int64
2 Number of Times Charted 1556 non-null int64
3 Week of Highest Charting 1556 non-null object
4 Song Name 1556 non-null object
5 Streams 1556 non-null object
6 Artist 1556 non-null object
7 Artist Followers 1556 non-null object
8 Song ID 1556 non-null object
9 Genre 1556 non-null object
10 Release Date 1556 non-null object
11 Weeks Charted 1556 non-null object
12 Popularity 1556 non-null object
13 Danceability 1556 non-null object
14 Energy 1556 non-null object
15 Loudness 1556 non-null object
16 Speechiness 1556 non-null object
17 Acousticness 1556 non-null object
18 Liveness 1556 non-null object
19 Tempo 1556 non-null object
20 Duration (ms) 1556 non-null object
21 Valence 1556 non-null object
22 Chord 1556 non-null object
dtypes: int64(3), object(20)
memory usage: 279.7+ KB
[ ] #number of times charted by artist
tist').sum().sort_values('Number of Times Charted', ascending=False)
Automatic saving failed. This file was updated remotely or in another tab. Show diff
0s completed at 12:50PM


0s
df_numbercharted = df.groupby('Artist').sum().sort_values('Number of Times Charted', ascending=False)
df_numbercharted = df_numbercharted.reset_index()
df_numbercharted
<ipython-input-25-e2107b4ca866>:1: FutureWarning: The default value of numeric_only in DataFrameGroupBy
df_numbercharted = df.groupby('Artist').sum().sort_values('Number of Times Charted', ascending=False
Artist Index Highest Charting Position Number of Times Charted
0 Billie Eilish 13908 1136 432
1 Juice WRLD 25342 1755 431
2 Harry Styles 1990 139 364
3 Ed Sheeran 4440 657 314
4 Pop Smoke 17702 2420 308
... ...
711 KALIM, Ufo361 1143 183 1
712 Kane Brown, blackbear 231 187 1
713 Kehlani 1191 177 1
714 Kygo, Donna Summer 897 194 1
715 Kontra K 1033 129 1
716 rows × 4 columns

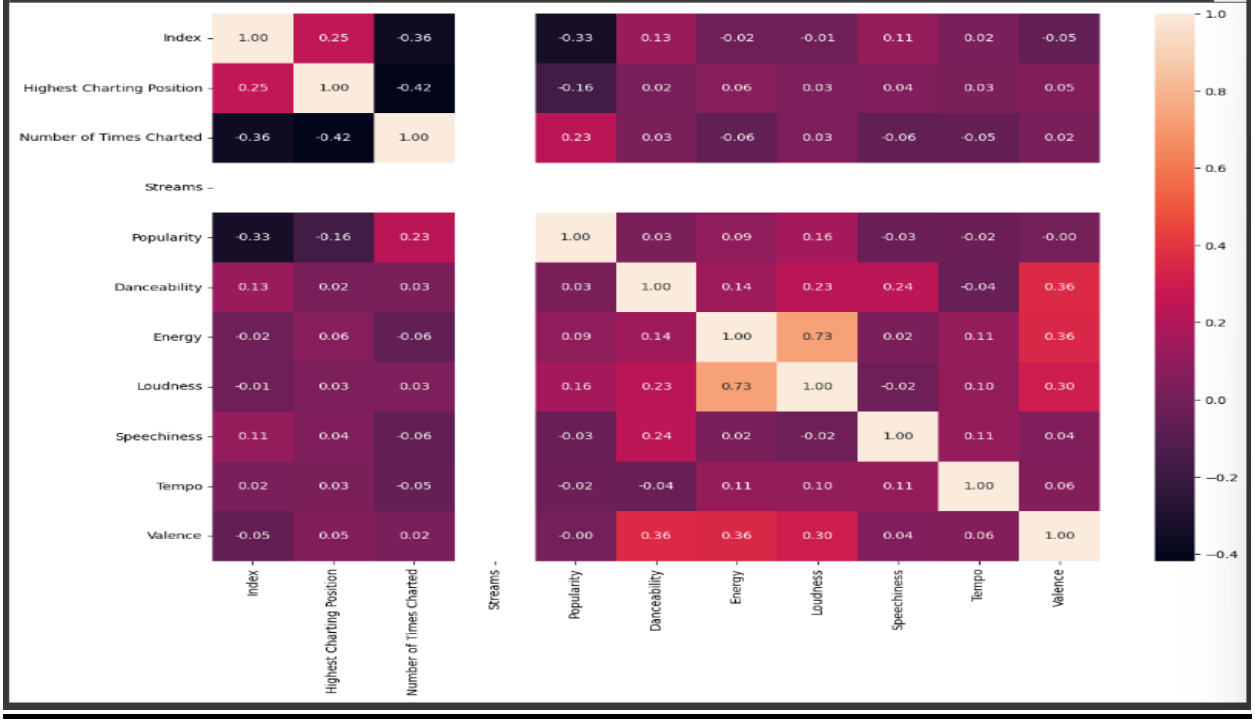


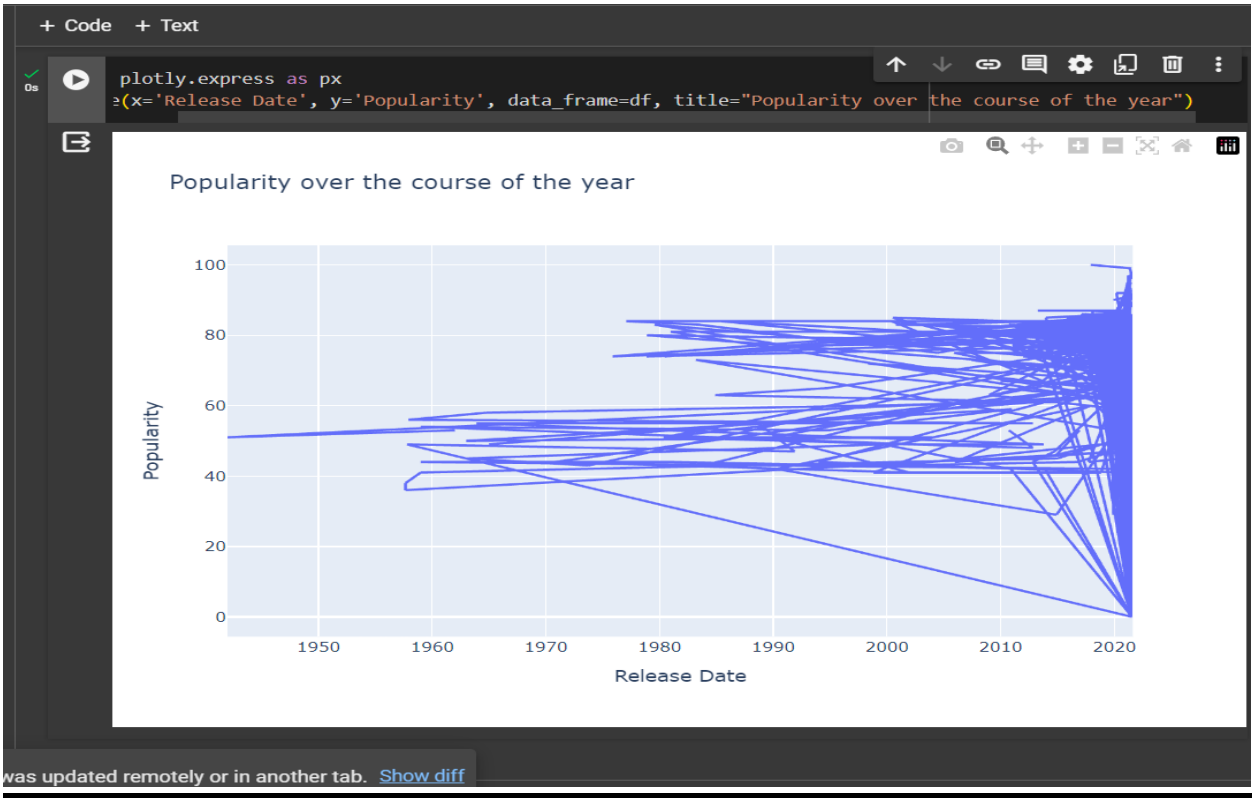

```
[ ] df=df.fillna(' ')
    df=df.replace(' ' , ' ')
    df['Streams']=df['Streams'].str.replace(',' , ' ')

[ ] #now convert all coloumns to numeric
    df[['Highest Charting Position', 'Number of Times Charted','Streams','Popularity','

    df['Release Date'] = pd.to_datetime(df['Release Date'], errors='coerce')
```

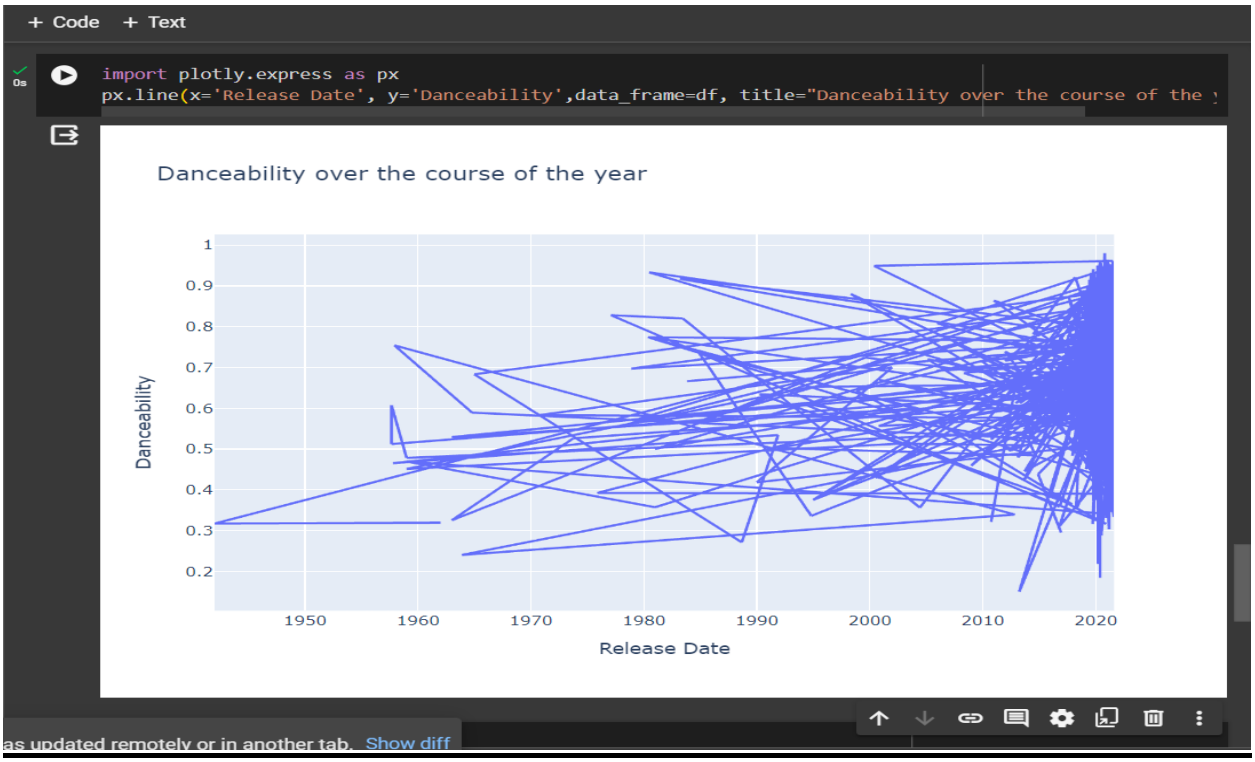
Double-click (or enter) to edit

```
[ ] %matplotlib inline
    f,ax = plt.subplots(figsize=(14,10))
    sns.heatmap(df.corr(), annot=True, fmt=".2f" , ax=ax)
    plt.show

    <ipython-input-53-6a88bf3eaf8c>:3: FutureWarning:

    The default value of numeric_only in DataFrame.corr is deprecated. In a future versi

    matplotlib.pyplot.show
    def show(*args, **kwargs)

    Display all open figures.

    Parameters
    ----------
    block : bool, optional
        Whether to wait for all figures to be closed before returning.
```

```
block : bool, optional
    Whether to wait for all figures to be closed before returning.
```






+ Code
+ Text
0s
import plotly.express as px
px.line(x='Release Date', y='Danceability',data_frame=df, title="Danceability over the course of the
Danceability over the course of the year
1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
Danceability
1950
1960
1970
1980
1990
2000
2010
2020
Release Date
as updated remotely or in another tab. Show diff

Outcome:

From this project, we learnt to describe a flow process for data science problems and classified data science problems into standard typology. We also learnt about correlating results to the solution approach followed and assessing the solution approach.

Project Conclusion:

From this project, we gained the knowledge of software – Google colab. We learnt to analyse the datasets and afterwards, visualizing them. We learnt about various plots .

PROJECT LINK :-

https://colab.research.google.com/drive/1i5XCzXKD8IQHacDhs7i7u82pYj8-uNeX?usp=sharing