Spotify Data Analysis Using Python

A Mini-Project Report submitted in partial fulfilment of the requirements
for the award of degree of

MASTER OF COMPUTER APPLICATIONS
of

Visvesvaraya Technological University

By

DEEKSHITH KUMAR C L (1CD21MC017)

CHETHANKUMAR K (1CD21MC015)

Under the Guidance of

Prof. Sunil Kumar K N

Department of Master of Computer Applications,
Cambridge Institute of Technology,
K R Puram,
Bangalore-560036.
MARCH 2023

Department of Master of Computer Applications,
Cambridge Institute of Technology,
KR Puram, Bangalore-560036.

CERTIFICATE

This is to Certify that DEEKSHITH KUMAR C L and CHETHANKUMAR K *has completed there III semester Mini Project work Entitled "**Spotify Data Analysis Using Python**" as a partial fulfilment for the award of Master of Computer Applications degree, during the academic year 2022-23 under my supervision.*

Signature of Guide
Prof. Sunil Kumar K N
Asst. Prof., MCA,
Cambridge Institute of Technology

Signature of HOD
Prof. Diwakar Karadi
HOD, MCA,
Cambridge Institute of Technology

HEAD OF THE DEPARTMENT
DEPARTMENT OF MCA
Cambridge Institute of Technology
K.R.Puram, Bangalore - 560 036

External Viva

Name of Examiners **Signature with Date**

1.

2.

DECLARATION

DEEKSHITH KUMAR C L and **CHETHANKUMAR K**, student of 3rd Semester MCA, Cambridge Institute of Technology, bearing hereby declare that the project entitled "**Spotify Data Analysis Using Python**" has been carried out by us under the supervision of Prof. Monika M., Asst. Prof., MCA, Cambridge Institute of Technology and submitted in partial fulfilment of the requirements for the award of the Degree of Master of Computer Applications by the Visvesvaraya Technological University during the academic year 2022-23. This report has not been submitted to any other Organization/University for any award of degree or certificate.

Project Member:

DEEKSHITH KUMAR C L (1CD21MC017)

CHETHANKUMAR K (1CD21MC015)

ACKNOWLEDGMENT

At the very outset I am very much thankful to my **Chairman, Shri. D. K. Mohan, CEO, Shri. Nithin Mohan** for being a strong support to our college and providing a well-equipped department (MCA).

I am thankful and grateful to our **Principal, Dr. Indumathi G**, for her kind permission and co-operation for making me to take up this mini project and make use of facility, which are available in our college.

I express my heartiest thanks to my beloved **Head of the Department, Prof. Diwakar Karadi** for being source of motivation during the development of my project.

And I am grateful for the generous help from my internal guide, **Prof. Sunil Kumar K N**, Department of MCA, Cambridge Institute of technology for the valuable guidance as the project guide for successful completion of this project work.

I would also like to express my sincere thanks to my parents for supporting me throughout my life and making me stand where I am right now. Finally, I thank my friends who have given their valuable time with co-operation and help throughout this project work.

CONTENT:

CHAPTER-1

ABSTRACT

The Spotify dataset provides insight into users data about which songs they listen to, and not just the popularity of tracks, but also features of the tracks they have in their library is recorded in their database.

In this project I will be analyzing a track's popularity based on several audio features

Provided in the dataset and find whether I can predict a track's popularity from key features about the song

Analysing user's listening profile to enable Spotify to suggest and acquire similar tracks on their platform to improve user experience

The plan is to analyse relationship between popularity and different features of the song, and maybe later perform cluster analysis using K-means method to provide song recommendation based on recent user listening on Spotify.

This is mainly useful to market songs to the Spotify users and improve their experience while using it. This analysis will help better understand the different clusters and enable Spotify to make a better targeted content distribution that would be helpful for the developers and the marketing team to analyse trends and help them to segments users better and try to increase profits and provide a better user experience

CHAPTER-2

INTRODUCTION

Spotify is a popular music streaming platform that provides access to millions of songs. With the help of Python, it's possible to analyse Spotify data and gain insights into user behaviour, music trends, and more

To get started with Spotify data analysis, you can use the Spotify Web API to access data about tracks, artists, albums, and playlists. You can also use the Spotify library, a Python wrapper for the Spotify Web API, to make it easier to access this data

Once you have access to the data, you can use Python's data analysis libraries such as pandas,numpy and matplotlib to perform data exploration, cleaning, transformation, and visualization. You can also use machine learning and natural language processing techniques to build predictive models, sentiment analysis, and personalized recommendations

Overall, Python is a powerful tool for analysing Spotify data, and it can help you gain valuable insights into user behaviour, music trends, and more.

CHAPTER-3

REQUIREMENT ANALYSIS

Requirement analysis for data analysis encompasses three major tasks: formulation, requirements gathering and analysis modeling. During formulation, the basic motivation and goals are identified, and the categories of users are defined. In the requirements gathering phase, the content and functional requirements are listed and interaction scenarios written from end-user's point-of-view are developed. This intent is to establish a basic understanding of why the analysis is done, who will use it, and what problems it will solve for its users.

3.1 SOFTWARE REQUIREMENT SPECIFICATION

Operating System	:	Windows XP/7
Application	:	Anaconda Navigator
Platform	:	Jupyter Notebook
Language	:	Python

3.2 HARDWARE REQUIREMENT SPECIFICATION

Processor	:	Standard processor with a speed of 2.0GHz
RAM	:	4BG
Hard Disk	:	50 GB or more
Monitor	:	Standard colour monitor
Keyboard	:	Standard keyboard

CHAPTER-4

SYSTEM ANALYSIS

System analysis is the process of gathering and interpreting facts, diagnosing problems and using the information to recommend improvements on the system. System analysis is problem solving activity that requires intensive communication between the system users and system developers.

It is a process of collecting and interpreting facts, identifying the problems, and decomposition of a system into its components.
System analysis is conducted for the purpose of studying a system or its parts in order to identify its objectives. It is a problem solving technique that improves the system and ensures that all the components of the system work efficiently to accomplish their purpose

System analysis or study is an important phase of any system development process. The system is viewed as a whole, the inputs are identified and the system is subjected to close study to identify the problem areas. The solutions are given as a proposal. The proposal is reviewed on user request and suitable changes are made. This loop ends as soon as the user is satisfied with the proposal.

CHAPTER-5

TECHNOLOGIES USED

5.1 Anaconda Navigator:

Anaconda Navigator is a desktop graphical user interface (GUI) included in Anaconda distribution that allows users to launch applications and manage CONDA packages, environments and channels without using command-line commands. Navigator can search for packages on Anaconda Cloud or in a local Anaconda Repository, install them in an environment, run the packages and update them. It is available for Windows, macOS and Linux.

Anaconda distribution comes with over 250 packages automatically installed, and over 7,500 additional open-source packages can be installed from PYPI as well as the CONDA package and virtual environment manager. It also includes a GUI, Anaconda Navigator as a graphical alternative to the command-line interface (CLI).

The following applications are available by default in Navigator:

- JUPYTER Lab
- JUPYTER Notebook
- Qt Console
- Spyder
- Glue
- Orange
- RStudio
- Visual Studio Code
- Visual Studio Code

5.2 JUPYTER Notebook:

The JUPYTER Notebook is an open-source web application that you can use to create and share documents that contain live code, equations, visualizations, and text. Jupyter Notebooks are a spin-off project from the IPython project, which used to have an IPython Notebook project itself. The name, Jupyter, comes from the core supported programming languages that it supports: Julia, Python, and R. Jupyter ships with the IPython kernel, which allows you to write your programs in Python, but there are currently over 100 other kernels that you can also use.

5.3 Python programming:

Python is a computer programming language often used to build websites and software, automate tasks, and conduct data analysis. Python is a general-purpose language, meaning it can be used to create a variety of different programs and isn't specialized for any specific problems. This versatility, along with its beginner-friendliness, has made it one of the most-used programming languages today. A survey conducted by industry analyst firm Red Monk found that it was the most popular programming language among developers in 2020

Python is commonly used for developing websites and software, task automation, data analysis, and data visualization. Since it's relatively easy to learn, Python has been adopted by many non-programmers such as accountants and scientists, for a variety of everyday tasks, like organizing finances.

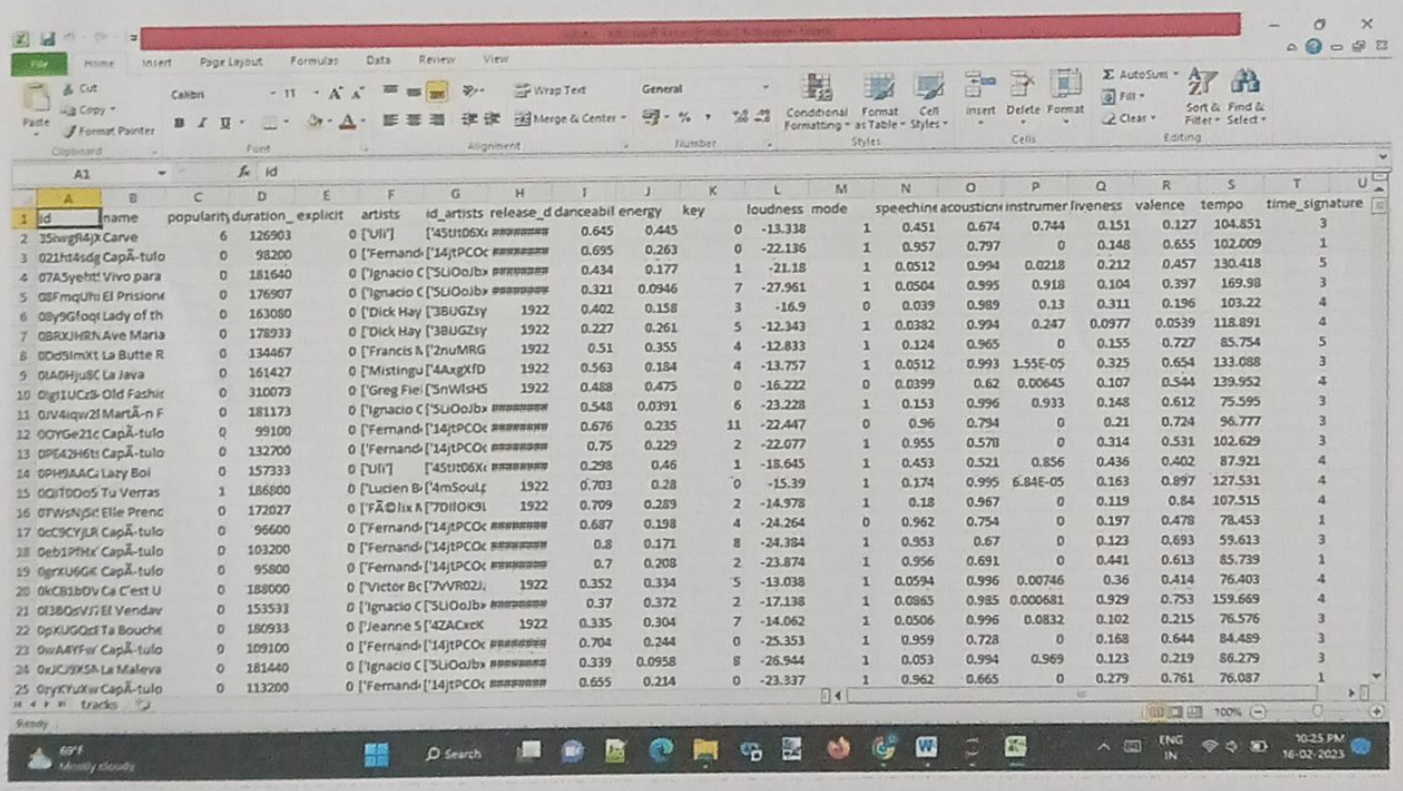
Some of the common ways Python is used

- Data analysis and machine learning
- Web development
- Automation or scripting
- Software testing and prototyping

5.4 DATASET:

A dataset (or data set) is a collection of data, usually presented in tabular form. Each column represents a particular variable. Each row corresponds to a given member of the dataset in question. It lists values for each of the variables, such as height and weight of an object. Each value is known as a datum. The dataset may comprise data for one or more members, corresponding to the number of rows.

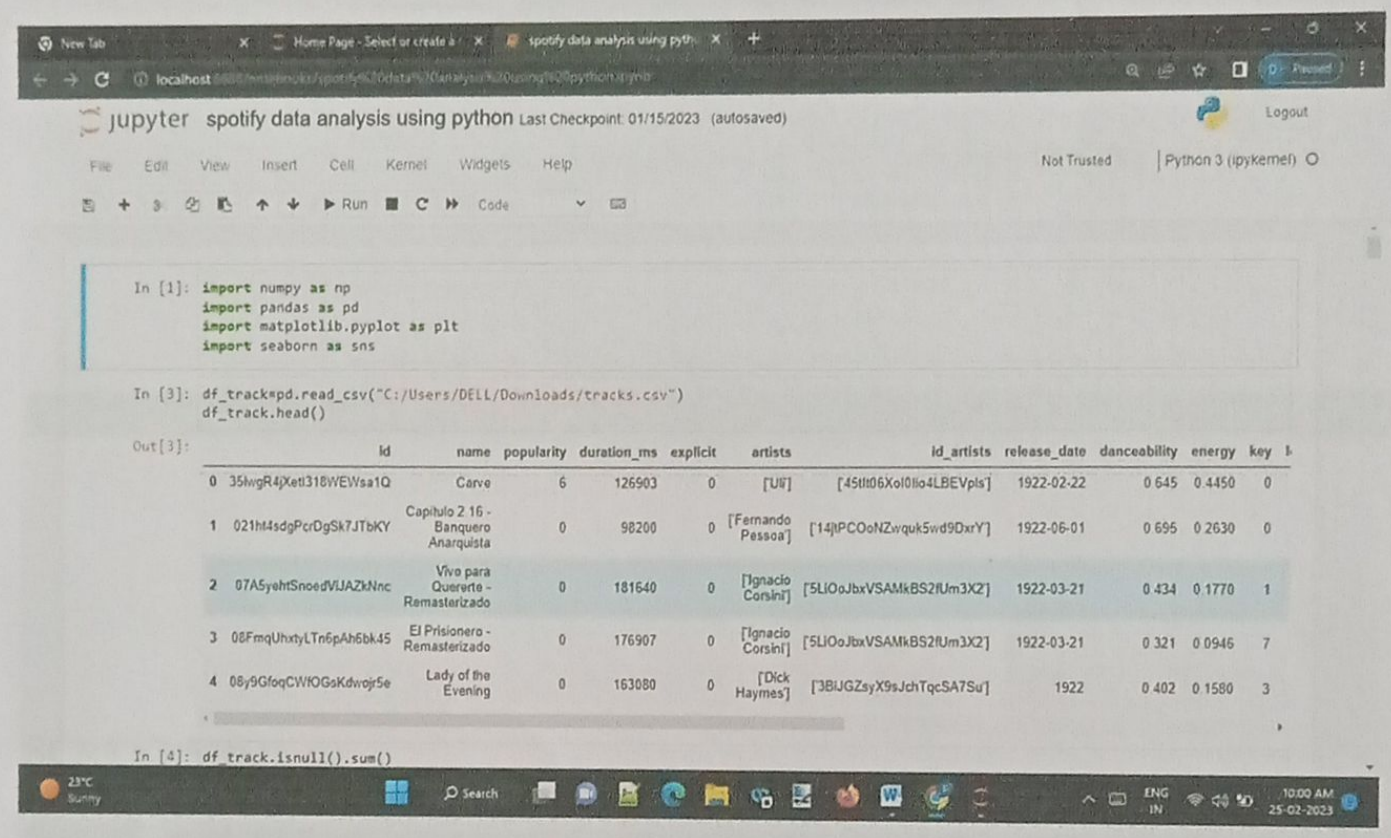
The dataset used in project is TRACKS.csv file is a excel sheet Dataset contains more than 160.000 songs collected from Spotify Web API. The features include song, artist, release date as well as some characteristics of song such as acoustic ness, danceability, loudness, tempo and so on. Date range is from 1921 to 2020.



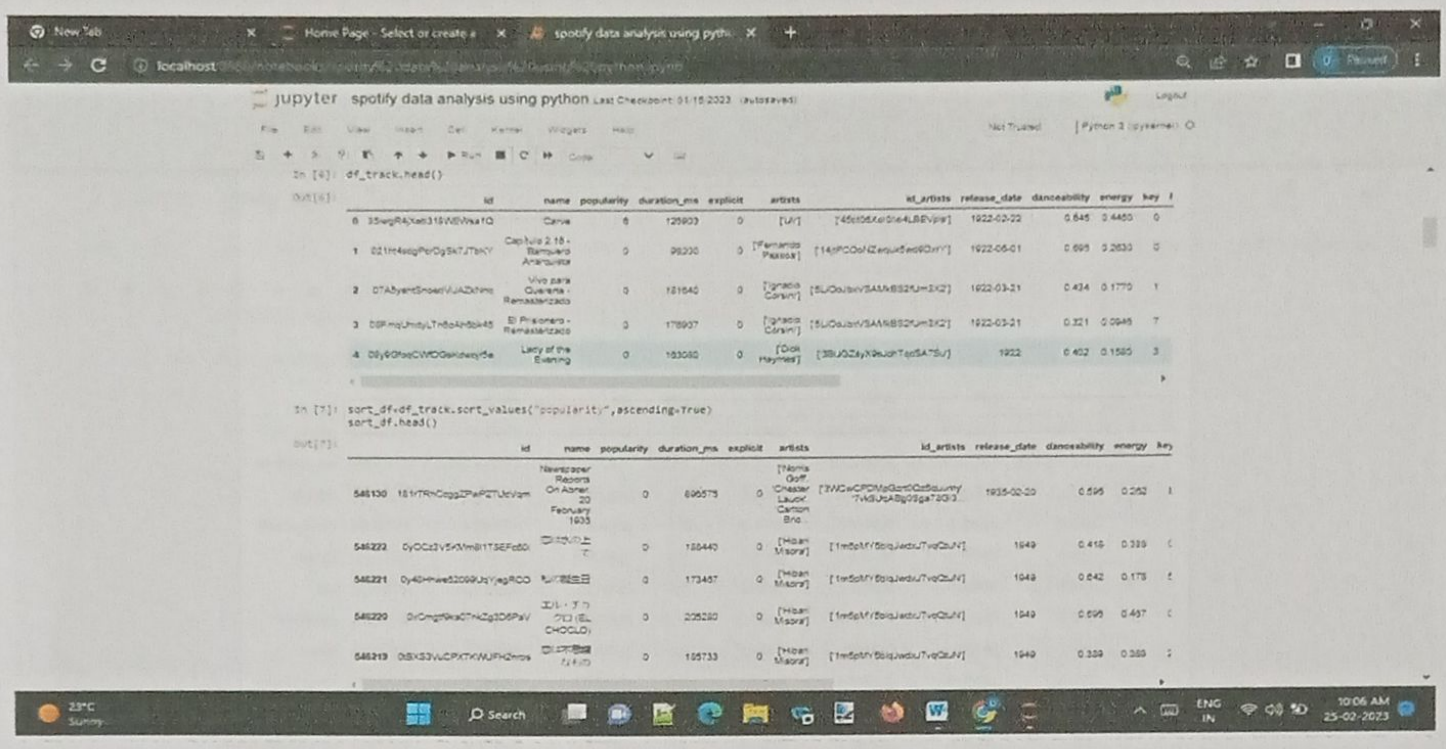
CHAPTER-6

SYSTEM DESIGN AND IMPLEMENTATION

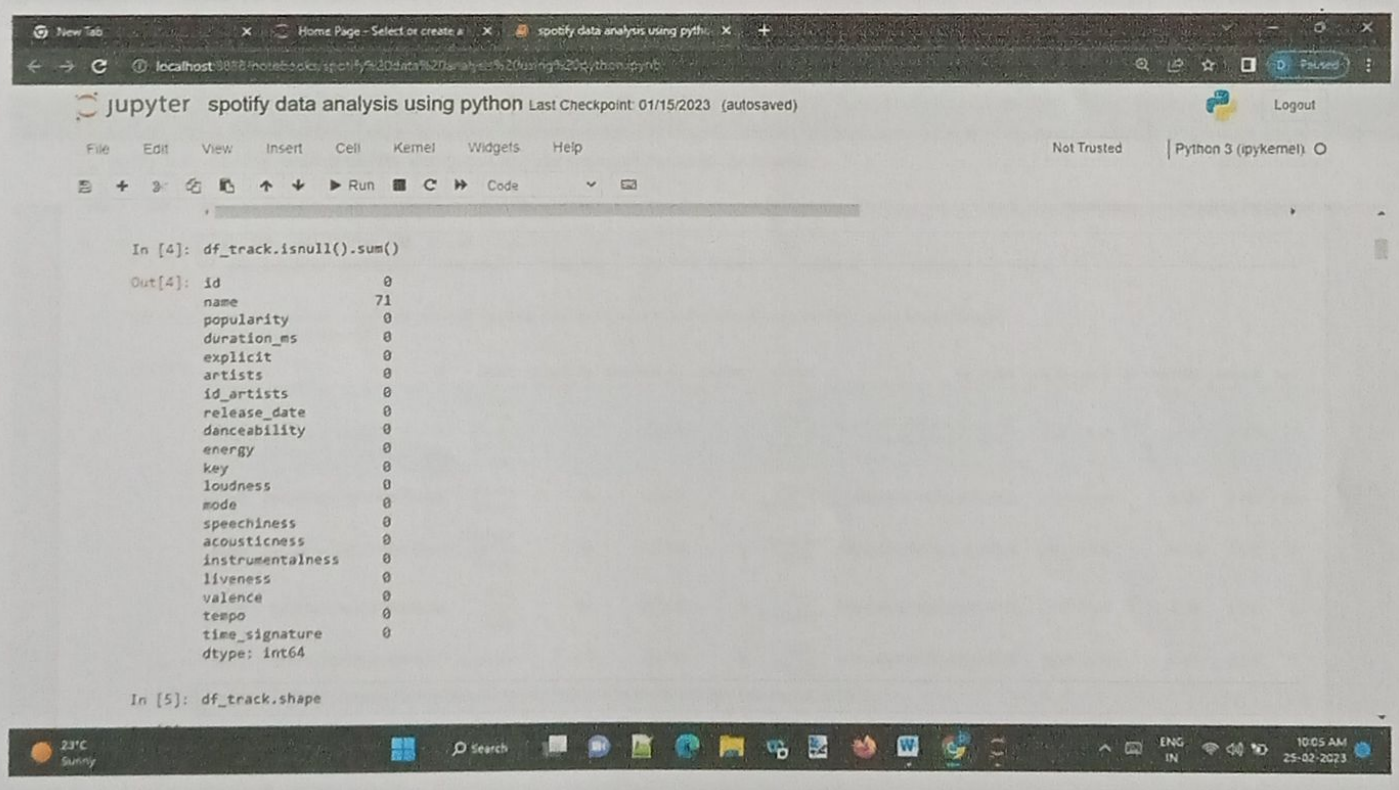
SCREENSHOT-01:



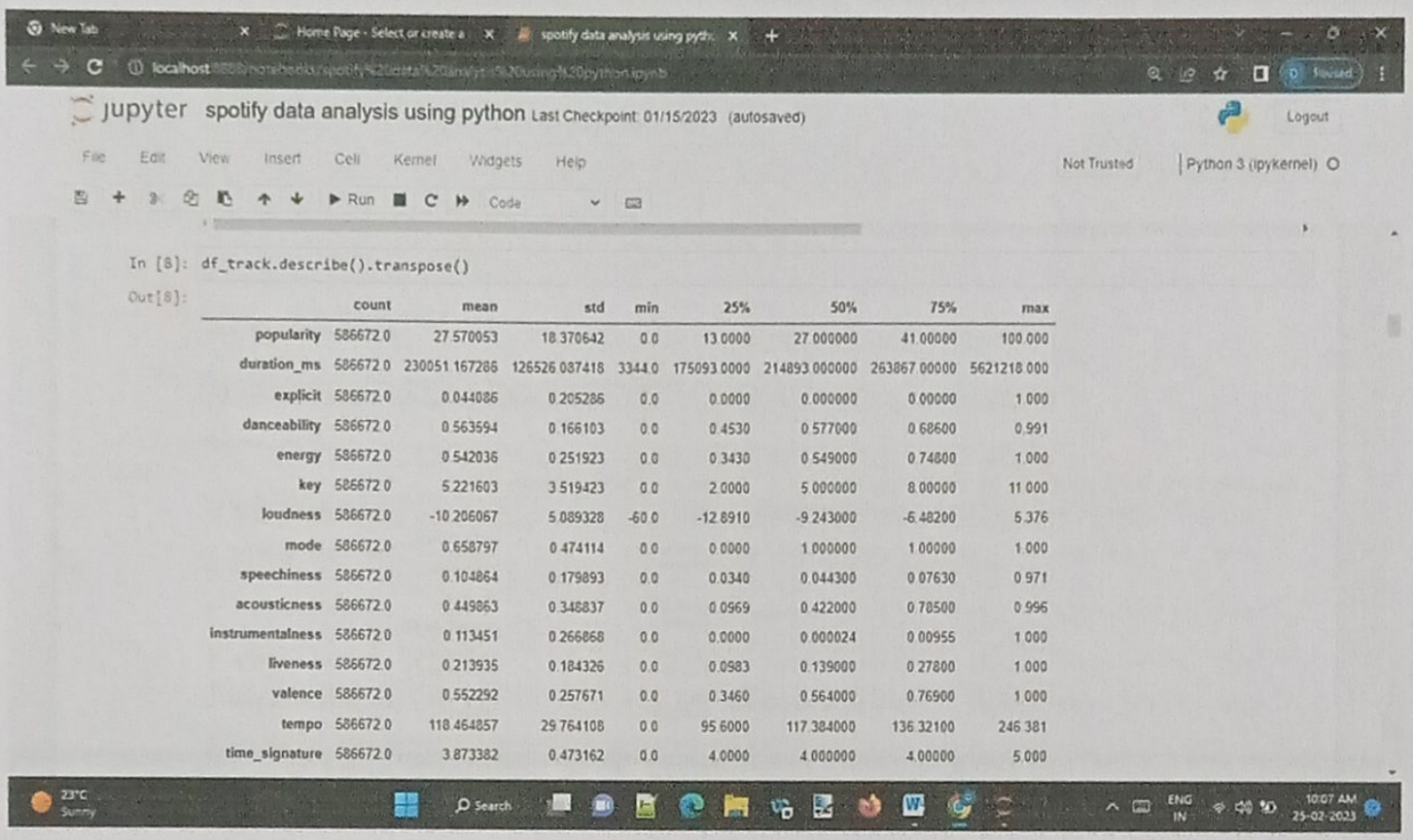
SCREENSHOT-02:



SCREENSHOT-03



SCREENSHOT-04:



jupyter spotify data analysis using python Last Checkpoint: 01/15/2023 (autosaved)

File Edit View Insert Cell Kernel Widgets Help

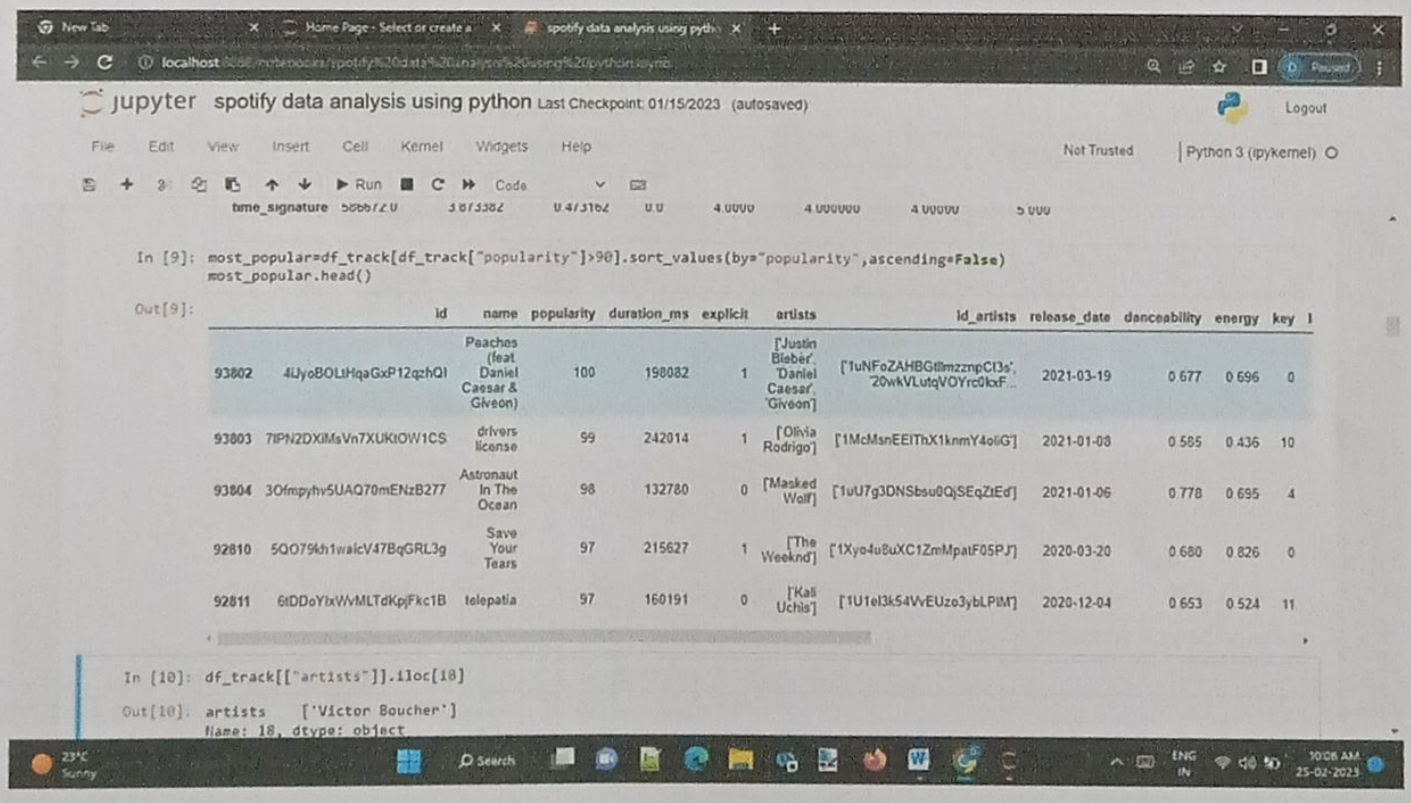
Not Trusted | Python 3 (ipykernel)

```
In [8]: df_track.describe().transpose()
```

Out[8]:

	count	mean	std	min	25%	50%	75%	max
popularity	586672.0	27.570053	18.370642	0.0	13.0000	27.000000	41.00000	100.000
duration_ms	586672.0	230051.167286	126526.087418	3344.0	175093.0000	214893.000000	263867.00000	5621218.000
explicit	586672.0	0.044086	0.205286	0.0	0.0000	0.000000	0.00000	1.000
danceability	586672.0	0.563594	0.166103	0.0	0.4530	0.577000	0.68600	0.991
energy	586672.0	0.542036	0.251923	0.0	0.3430	0.549000	0.74800	1.000
key	586672.0	5.221603	3.519423	0.0	2.0000	5.000000	8.00000	11.000
loudness	586672.0	-10.206067	5.089328	-60.0	-12.8910	-9.243000	-6.48200	5.376
mode	586672.0	0.658797	0.474114	0.0	0.0000	1.000000	1.00000	1.000
speechiness	586672.0	0.104864	0.179893	0.0	0.0340	0.044300	0.07630	0.971
acousticness	586672.0	0.449863	0.346837	0.0	0.0969	0.422000	0.78500	0.996
instrumentalness	586672.0	0.113451	0.266868	0.0	0.0000	0.000024	0.00955	1.000
liveness	586672.0	0.213935	0.184326	0.0	0.0983	0.139000	0.27800	1.000
valence	586672.0	0.552292	0.257671	0.0	0.3460	0.564000	0.76900	1.000
tempo	586672.0	118.464857	29.764108	0.0	95.6000	117.384000	136.32100	246.381
time_signature	586672.0	3.873382	0.473162	0.0	4.0000	4.000000	4.00000	5.000

SCREENSHOT-05:



jupyter spotify data analysis using python Last Checkpoint: 01/15/2023 (autosaved)

File Edit View Insert Cell Kernel Widgets Help

Not Trusted | Python 3 (ipykernel)

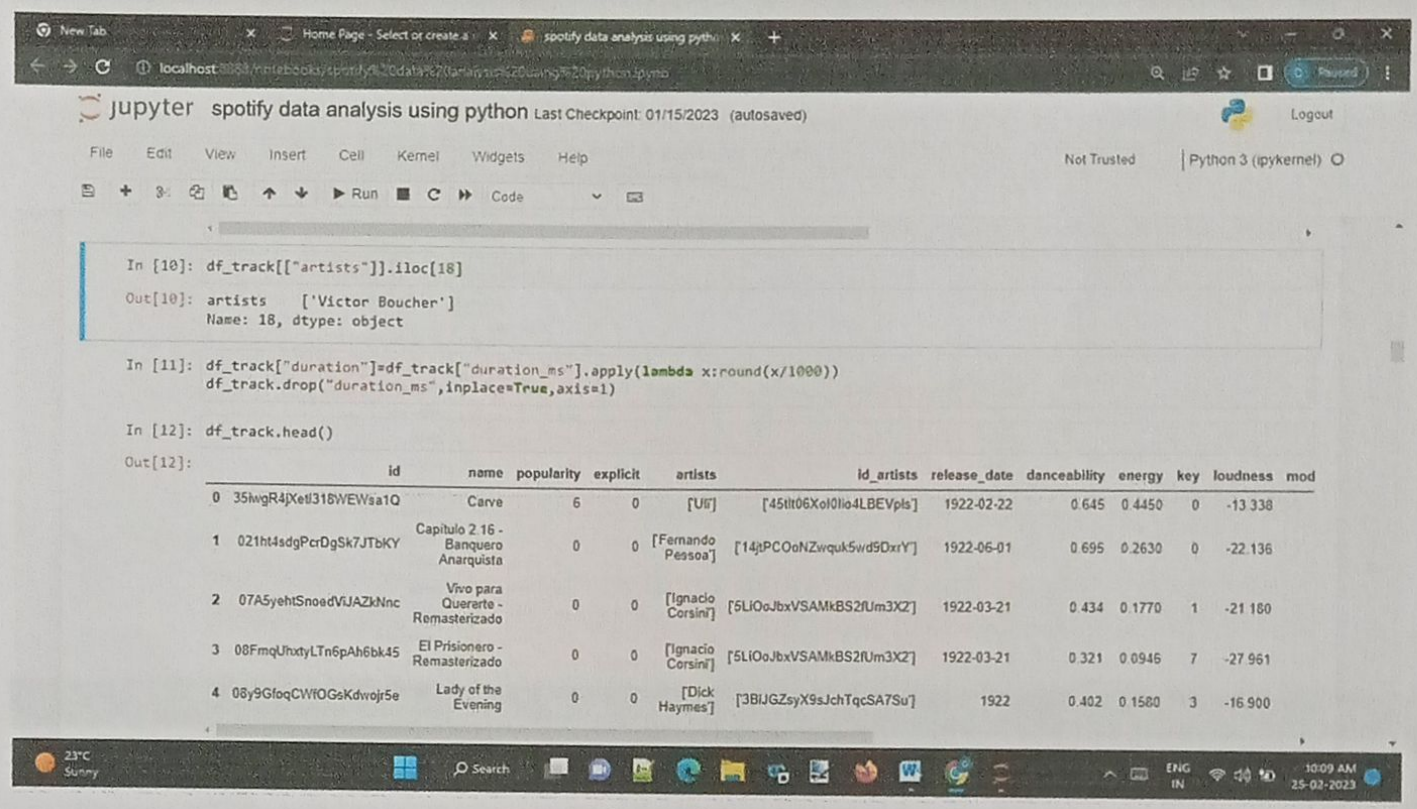
```
In [9]: most_popular=df_track[df_track["popularity"]>90].sort_values(by="popularity",ascending=False)
        most_popular.head()
```

Out[9]:

	id	name	popularity	duration_ms	explicit	artists	id_artists	release_date	danceability	energy	key	l
93802	4iJyoBOLtHqaGxP12qzhQI	Peaches (feat. Daniel Caesar & Giveon)	100	198082	1	['Justin Bieber', 'Daniel Caesar', 'Giveon']	['1uNFoZAHBGtllmzznpCI3s', '20wkVLutqVOYrc0kxF...	2021-03-19	0.677	0.696	0	
93803	7tPN2DXiMsVn7XUKtOW1CS	drivers license	99	242014	1	['Olivia Rodrigo']	['1McMsnEElThX1knmY4oliG']	2021-01-08	0.585	0.436	10	
93804	3Ofmpyhv5UAQ70mENzB277	Astronaut In The Ocean	98	132780	0	['Masked Wolf']	['1uU7g3DNSbsu0QjSEqZtEd']	2021-01-06	0.778	0.695	4	
92810	5QO79kh1waicV47BqGRL3g	Save Your Tears	97	215627	1	['The Weeknd']	['1Xyo4u8uXC1ZmMpatF05PJ']	2020-03-20	0.680	0.826	0	
92811	6tDDoYIxWvMLTdKpjFkc1B	telepatía	97	160191	0	['Kali Uchis']	['1U1el3k54VvEUzo3ybLPlM']	2020-12-04	0.653	0.524	11	

```
In [10]: df_track[["artists"]].iloc[18]

Out[10]: artists    ['Victor Boucher']
         Name: 18, dtype: object
```

SCREENSHOT-06:



jupyter spotify data analysis using python Last Checkpoint: 01/15/2023 (autosaved)

Logout

File Edit View Insert Cell Kernel Widgets Help

Not Trusted | Python 3 (ipykernel)

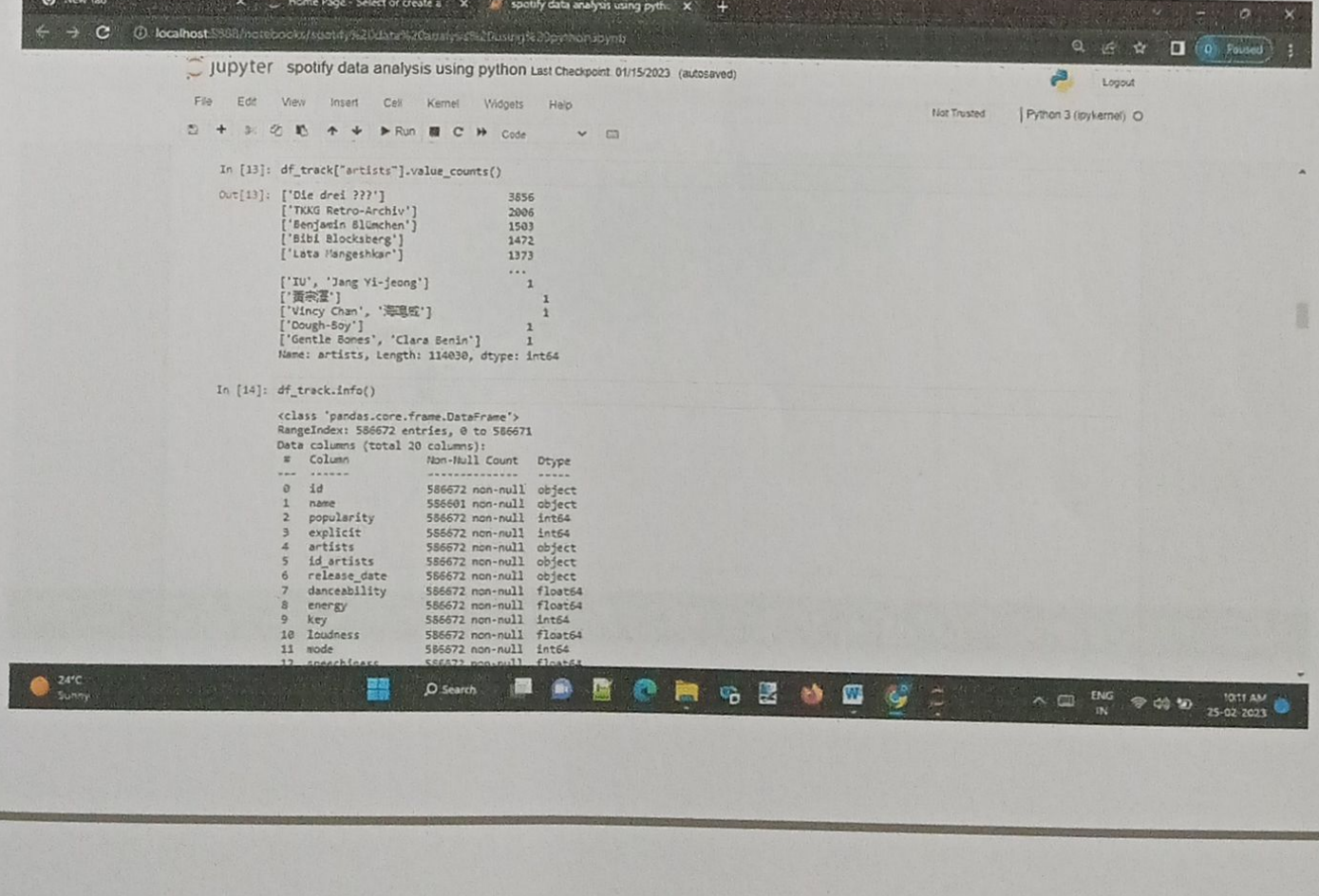
Run Code

```
In [10]: df_track[["artists"]].iloc[18]

Out[10]: artists    ['Victor Boucher']
         Name: 18, dtype: object

In [11]: df_track["duration"]=df_track["duration_ms"].apply(lambda x:round(x/1000))
         df_track.drop("duration_ms",inplace=True,axis=1)

In [12]: df_track.head()
```

Out[12]:

	id	name	popularity	explicit	artists	id_artists	release_date	danceability	energy	key	loudness	mod
0	35iwgR4jXetI318WEWsa1Q	Carve	6	0	['Uli']	['45tIt06XoI0Iio4LBEVpls']	1922-02-22	0.645	0.4450	0	-13.338	
1	021ht4sdgPcrDgSk7JTbKY	Capítulo 2.16 - Banquero Anarquista	0	0	['Fernando Pessoa']	['14jtPCOoNZwquk5wd9DxrY']	1922-06-01	0.695	0.2630	0	-22.136	
2	07A5yehtSnoedViJAZkNnc	Vivo para Quererte - Remasterizado	0	0	['Ignacio Corsini']	['5LiOoJbxVSAMkBS2fUm3XZ']	1922-03-21	0.434	0.1770	1	-21.180	
3	08FmqUhxtyLTn6pAh6bk45	El Prisionero - Remasterizado	0	0	['Ignacio Corsini']	['5LiOoJbxVSAMkBS2fUm3XZ']	1922-03-21	0.321	0.0946	7	-27.961	
4	08y9GfoqCWfOGsKdwojr5e	Lady of the Evening	0	0	['Dick Haymes']	['3BIJGZsyX9sJchTqcSA7Su']	1922	0.402	0.1580	3	-16.900	

SCREENSHOT-07:



jupyter spotify data analysis using python Last Checkpoint: 01/15/2023 (autosaved)

Logout

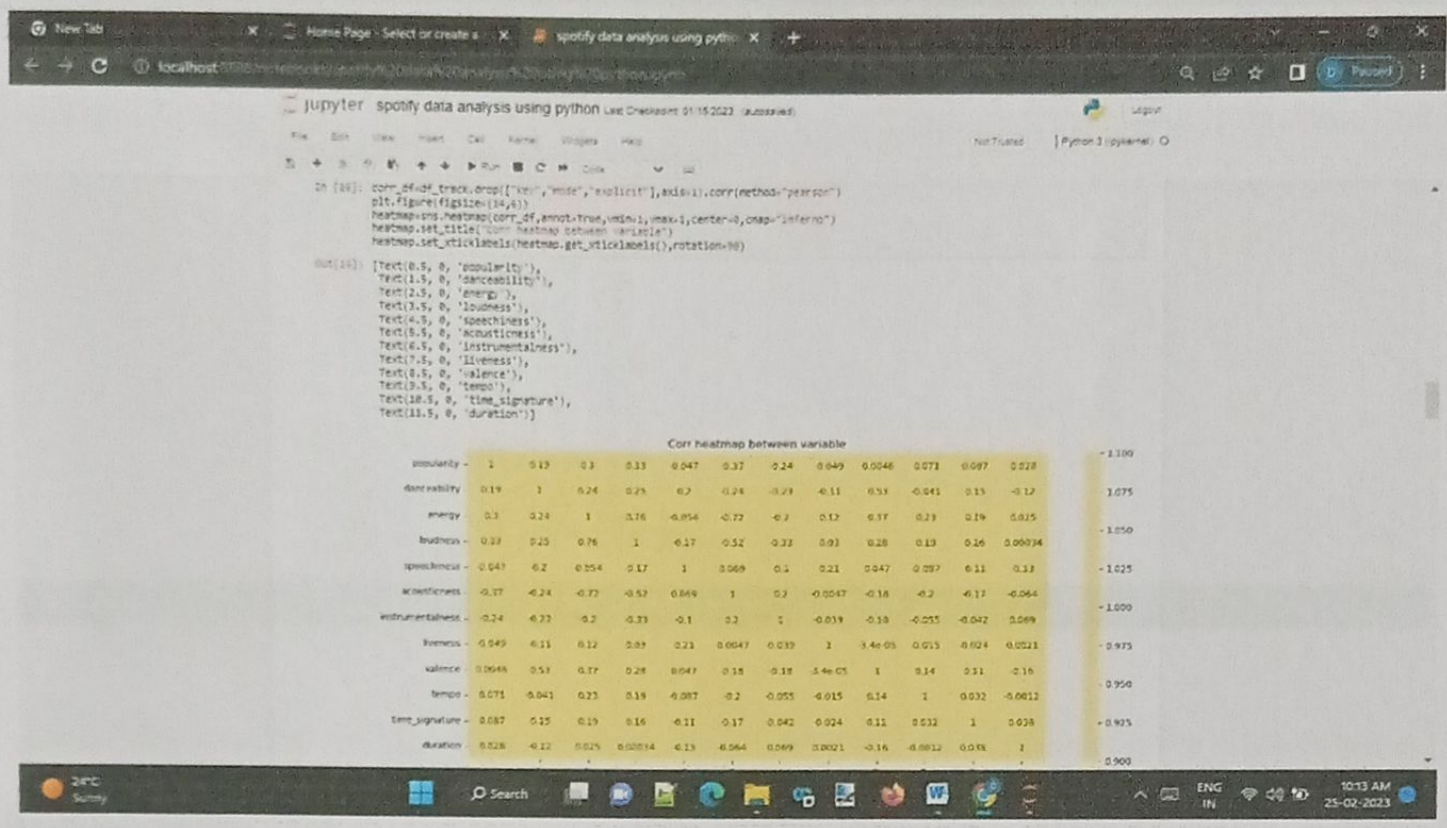
File Edit View Insert Cell Kernel Widgets Help

Not Trusted | Python 3 (ipykernel)

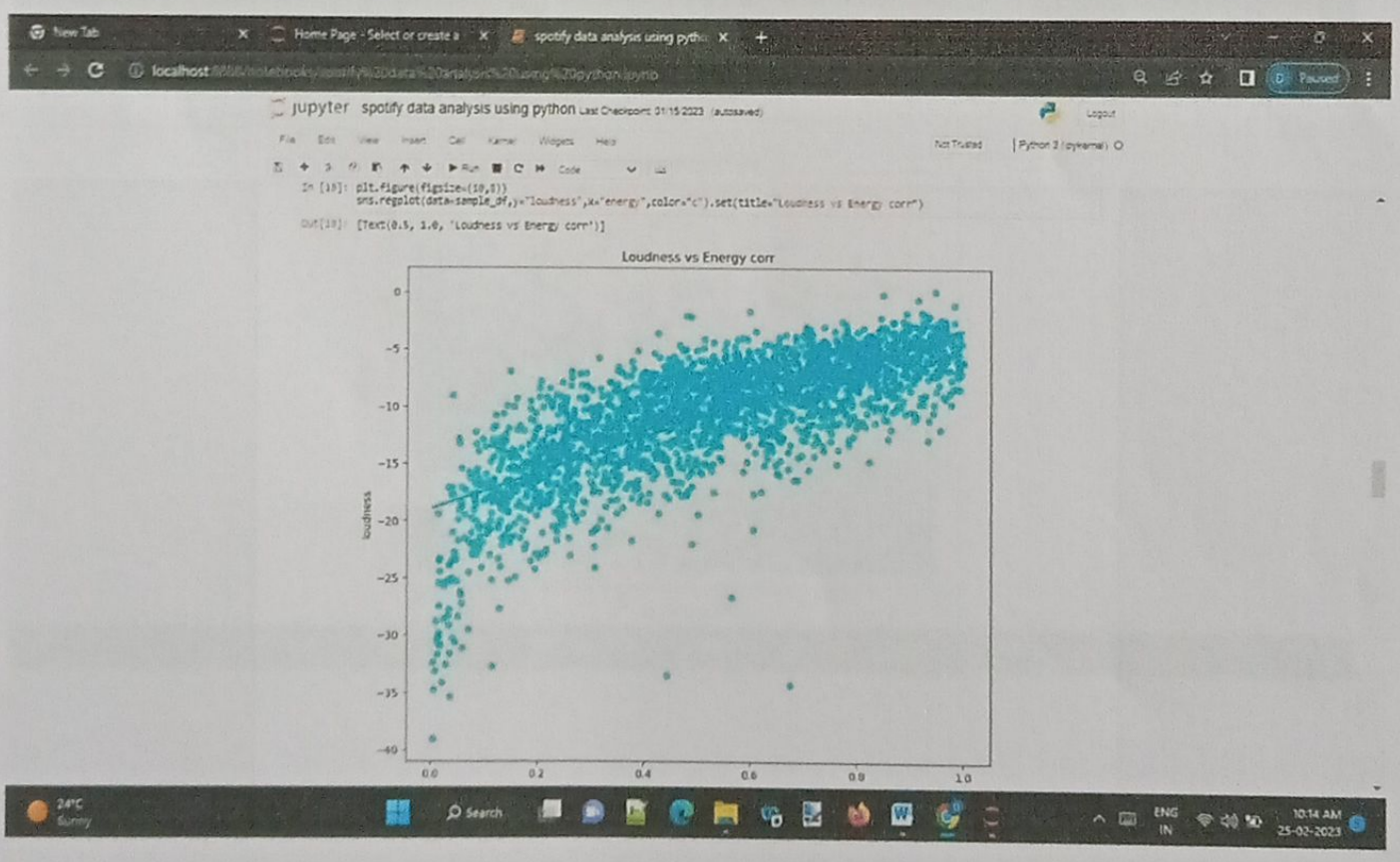
Run Code

```
In [13]: df_track["artists"].value_counts()

Out[13]: ['Die drei ???']                3656
         ['TKKG Retro-Archiv']           2006
         ['Benjamin Blümchen']           1503
         ['Bibi Blocksberg']             1472
         ['Lata Mangeshkar']             1373
                                         ...
         ['IU', 'Jang Yi-jeong']            1
         ['黃宗澤']                           1
         ['Vincy Chan', '陳慧敏']             1
         ['Dough-Boy']                      1
         ['Gentle Bones', 'Clara Benin']    1
         Name: artists, Length: 114030, dtype: int64

In [14]: df_track.info()

         <class 'pandas.core.frame.DataFrame'>
         RangeIndex: 586672 entries, 0 to 586671
         Data columns (total 20 columns):
          #   Column            Non-Null Count   Dtype
         ---  ------            --------------   -----
          0   id                586672 non-null  object
          1   name              586601 non-null  object
          2   popularity        586672 non-null  int64
          3   explicit          586672 non-null  int64
          4   artists           586672 non-null  object
          5   id_artists        586672 non-null  object
          6   release_date      586672 non-null  object
          7   danceability      586672 non-null  float64
          8   energy            586672 non-null  float64
          9   key               586672 non-null  int64
          10  loudness          586672 non-null  float64
          11  mode              586672 non-null  int64
          12  speechiness       586672 non-null  float64
```

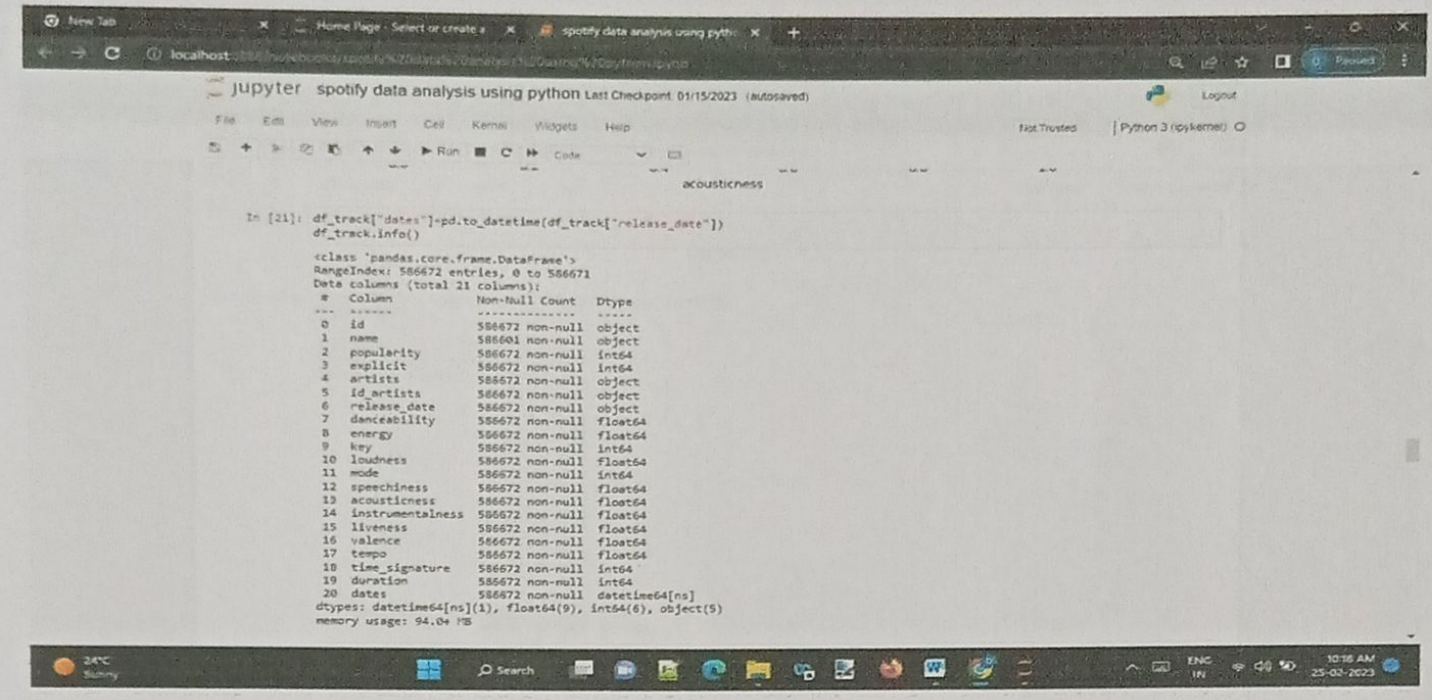
SCREENSHOT-08:



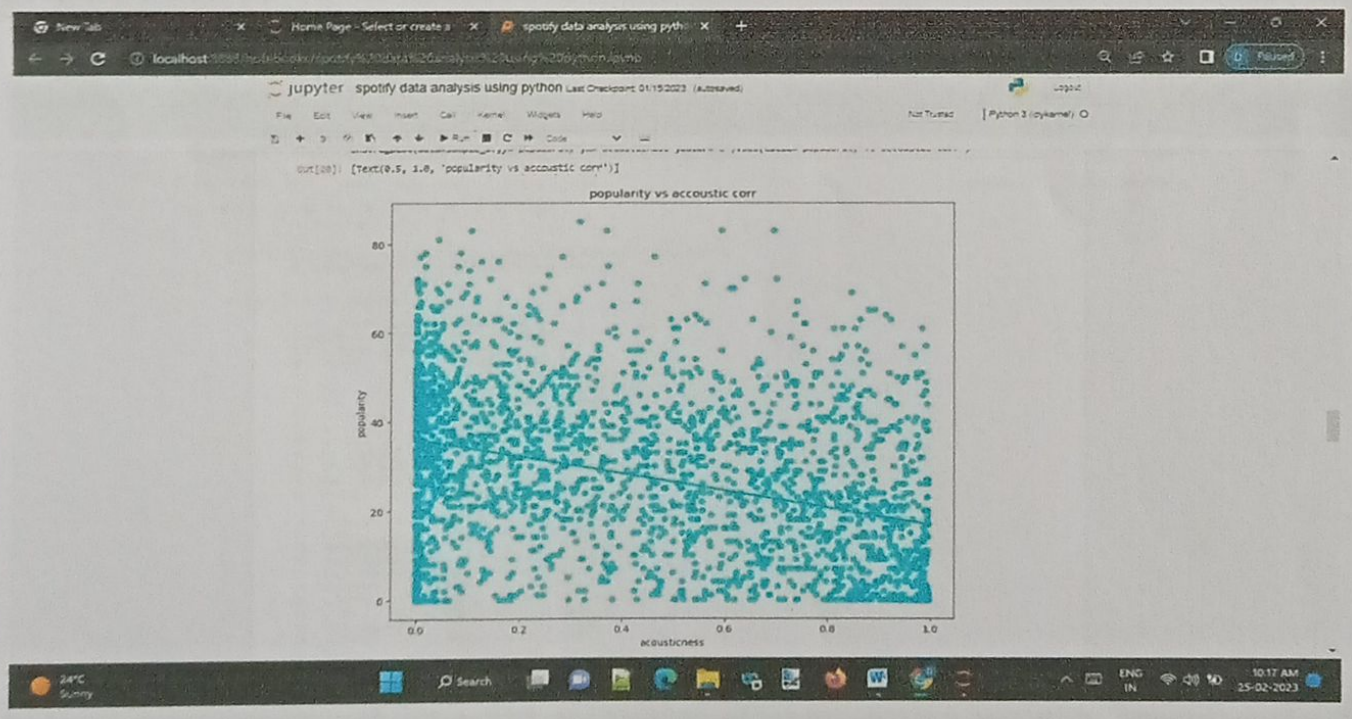
SCREENSHOT-09



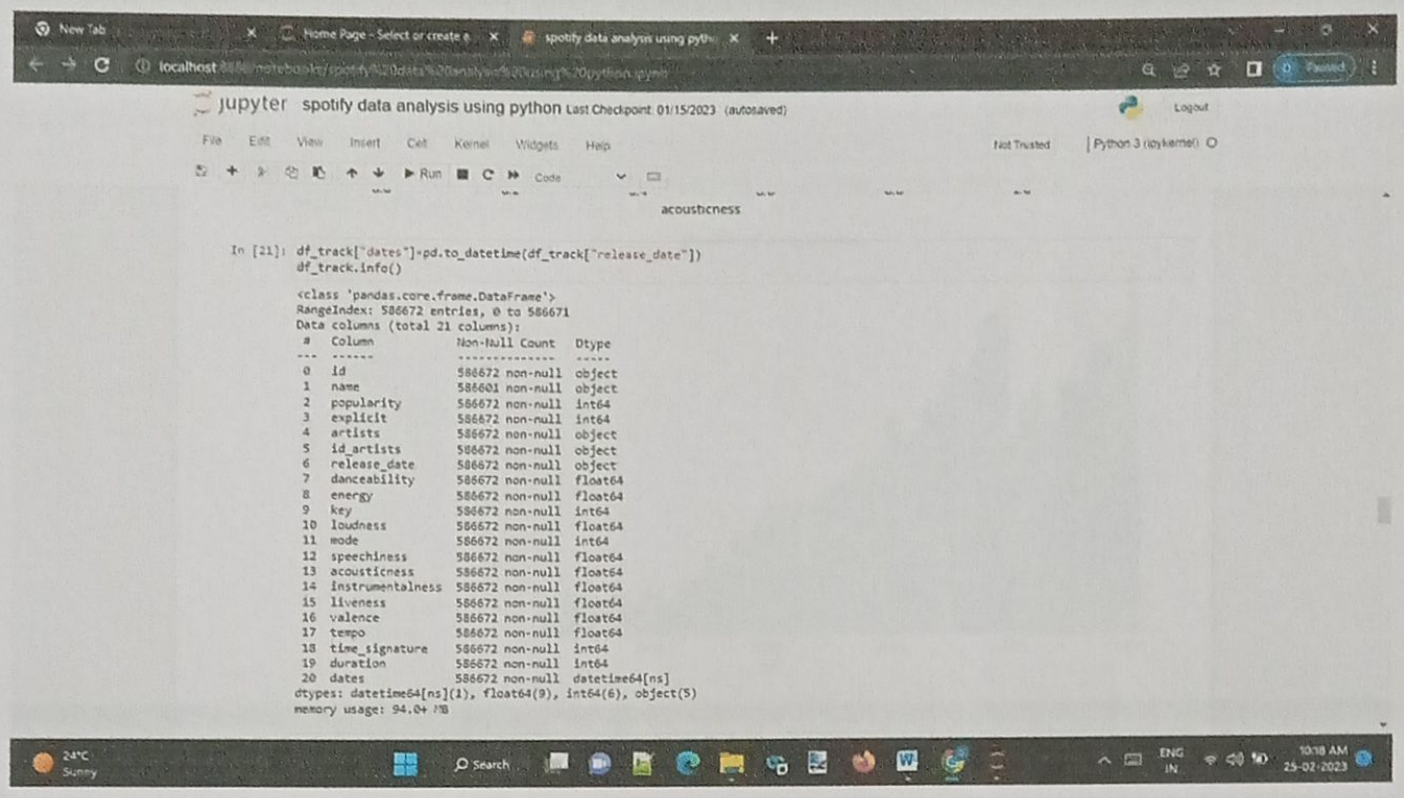
SCREENSHOT-10



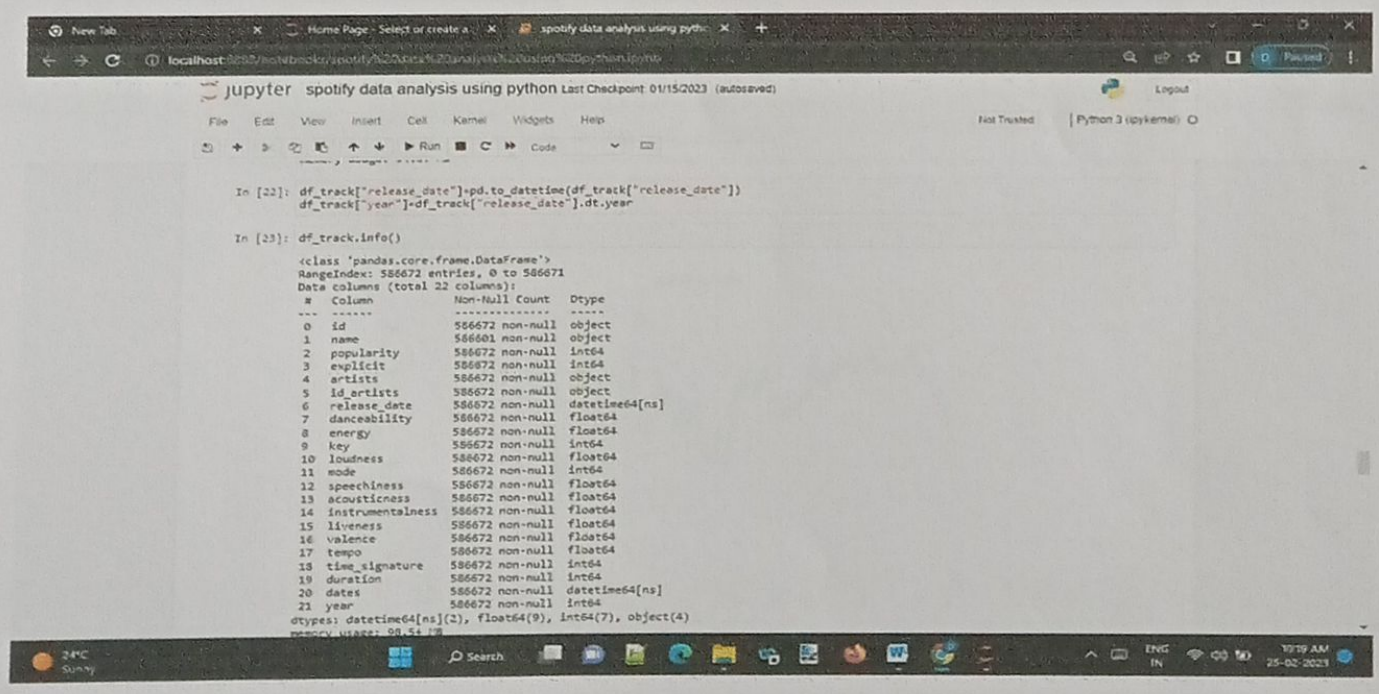
SCREENSHOT-11



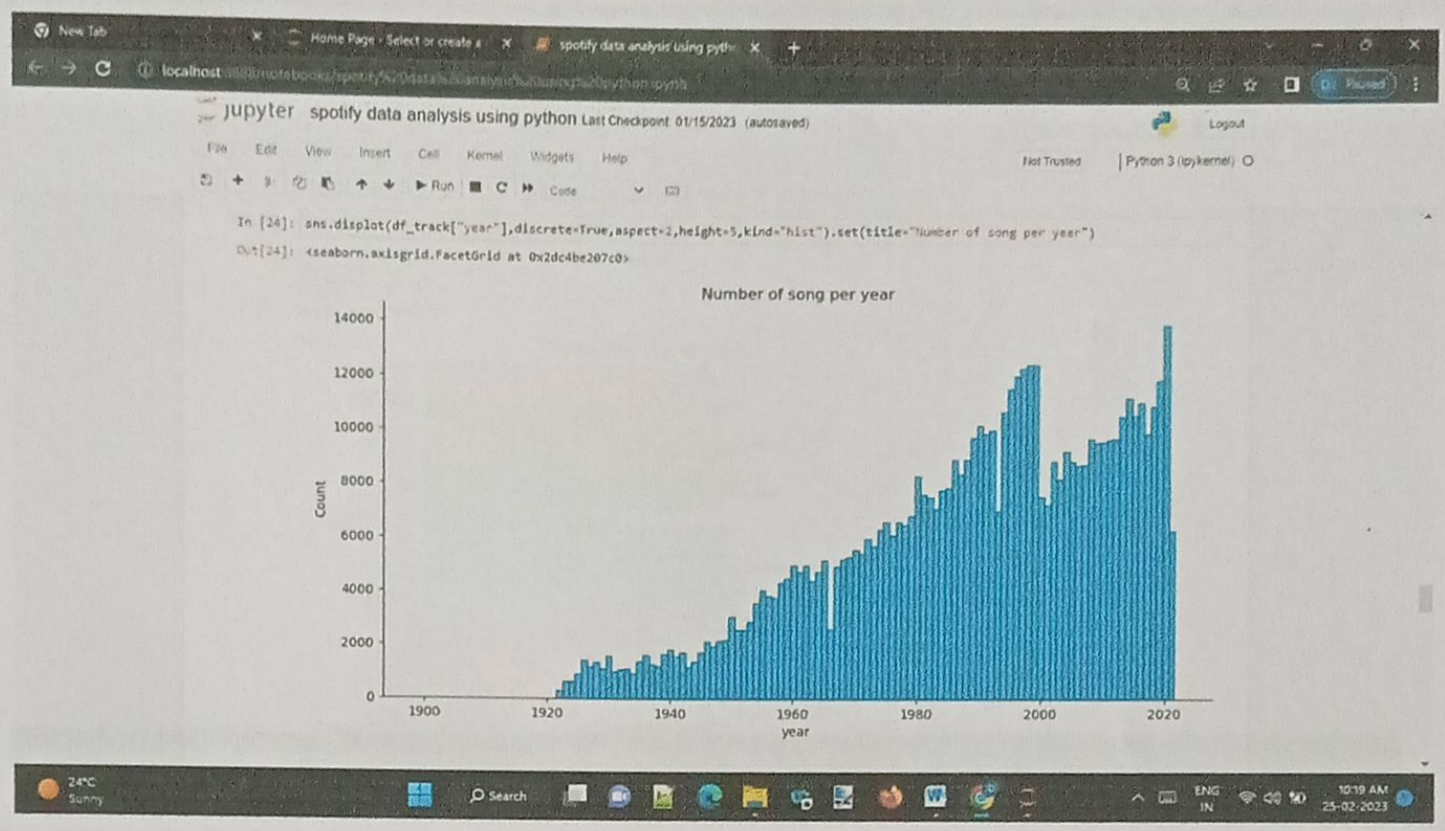
SCREENSHOT-12



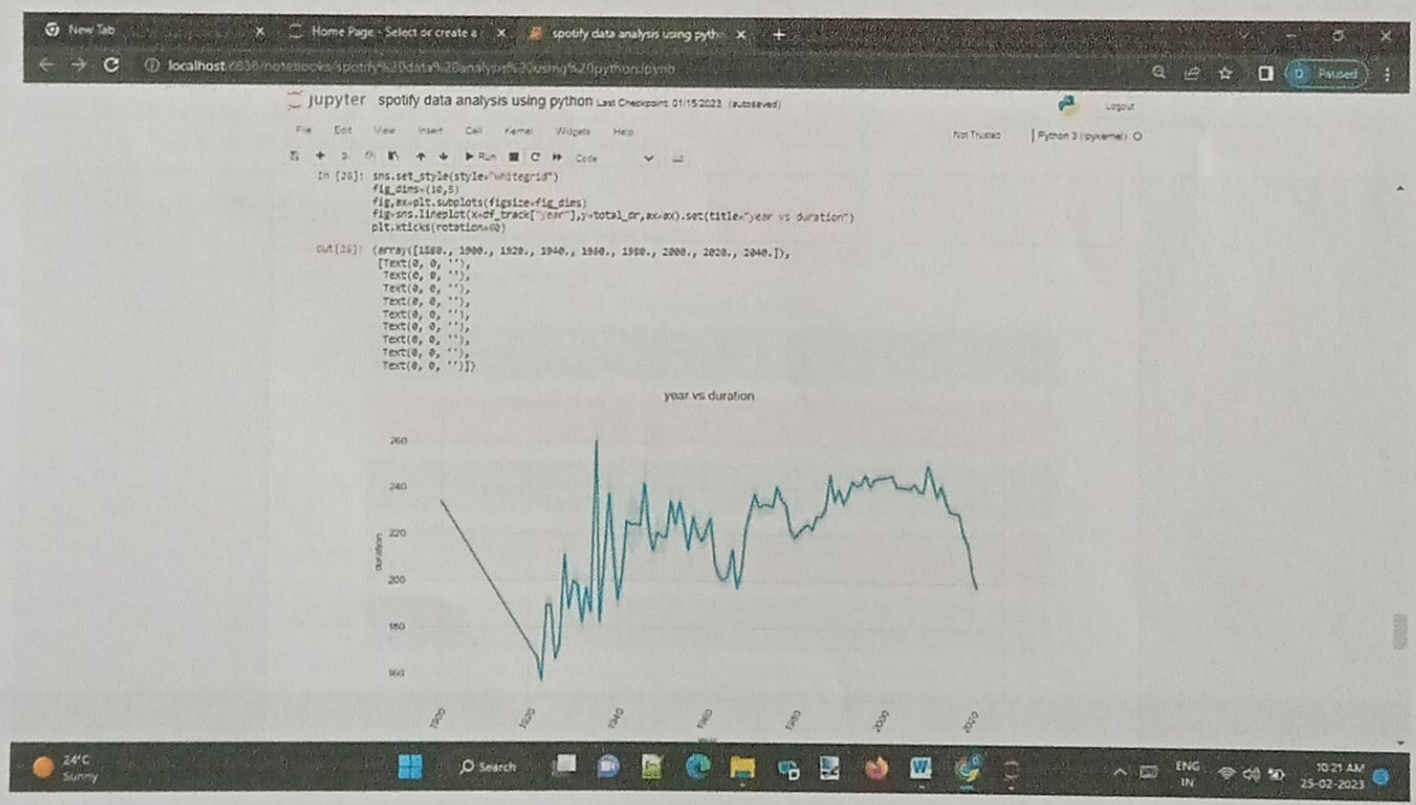
SCREENSHOT-13



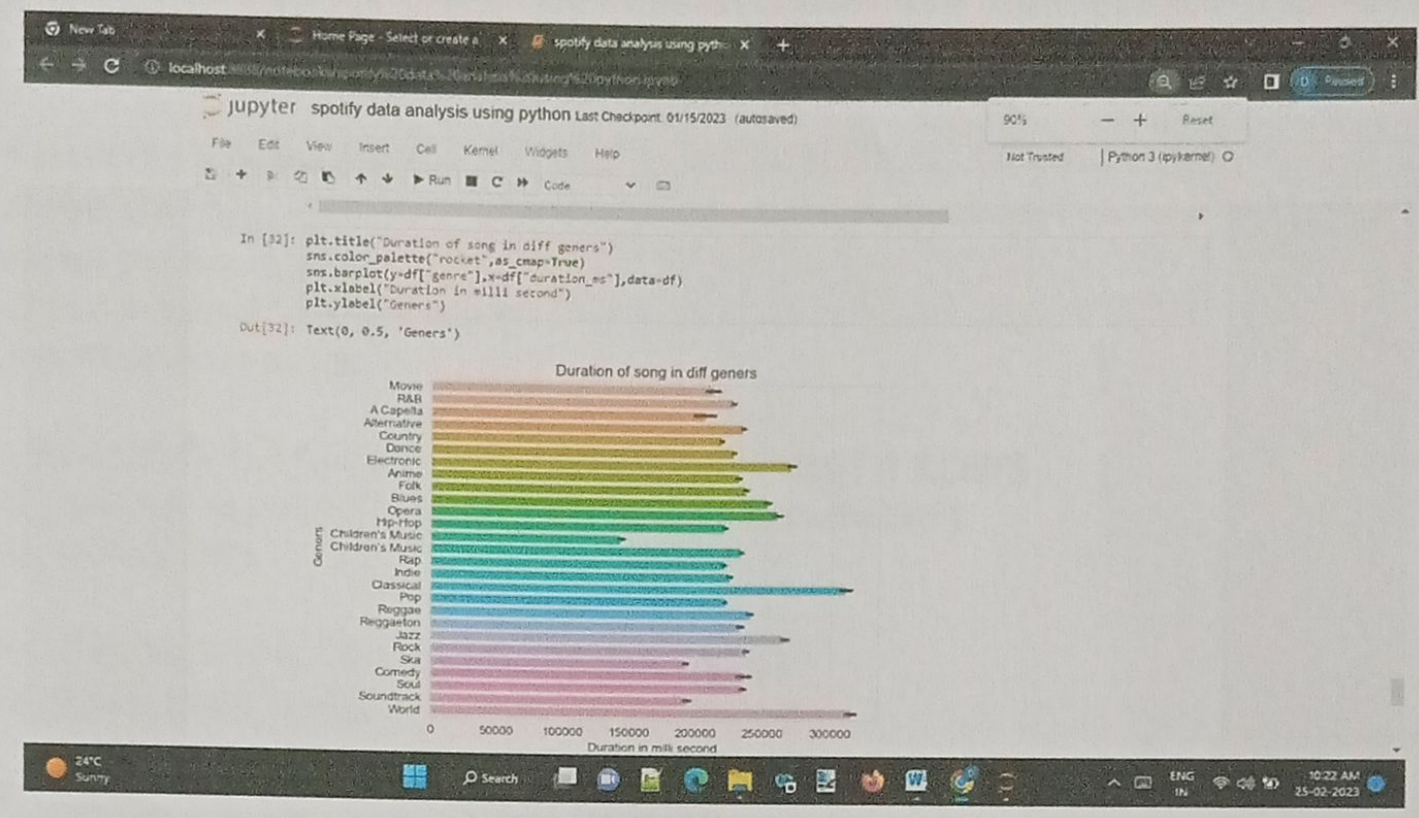
SCREENSHOT-14



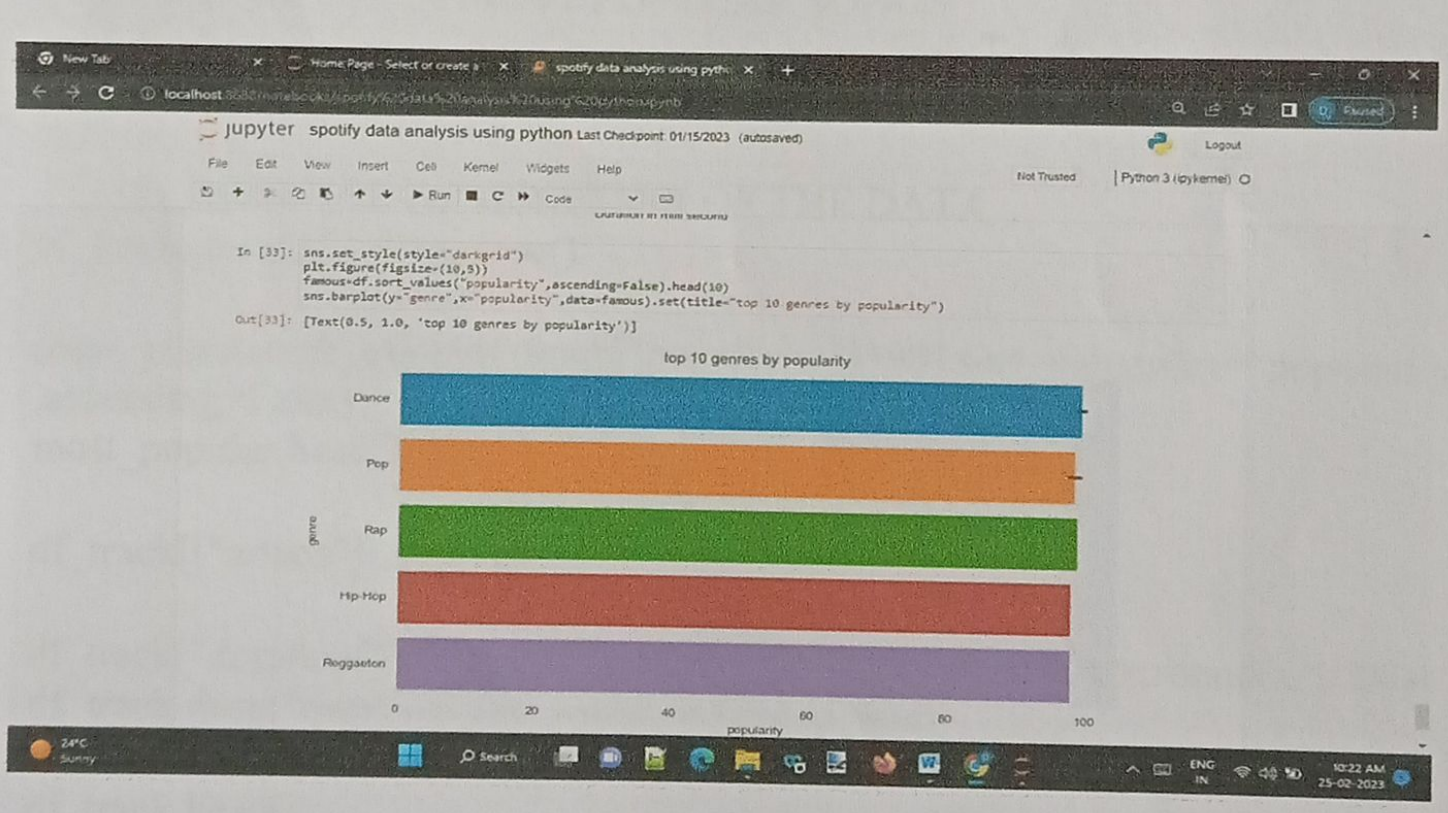
SCREENSHOT-15



SCREENSHOT-16



SCREENSHOT-17



CHAPTER-07

CODING

```
# IMPORT LIBRARIES
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns

# READING DATASET AND CHECKING FIRST 5 ROWS
df_track=pd.read_csv("C:/Users/DELL/Downloads/tracks.csv")
df_track.head()

# CHECKING NULL VALUES IN DATASET
df_track.isnull().sum()

# CHECKING SHAPE OF AN ARRAY
df_track.shape

# CHECKING TOP 10 MOST POPULAR SONGS
sort_df=df_track.sort_values("popularity",ascending=True)
sort_df.head()

# CHECKING THE DESCRIPTION OF THE DATA
df_track.describe().transpose()

most_popular=df_track[df_track["popularity"]>90].sort_values(by="popularity"
,ascending=False)
most_popular.head()

df_track[["artists"]].iloc[18]

df_track["duration"]=df_track["duration_ms"].apply(lambda x:round(x/1000))
df_track.drop("duration_ms",inplace=True,axis=1)

df_track.head()

df_track["artists"].value_counts()

df_track.info()

df_track[df_track["duration"]==5621]["artists"]
```

```
corr_df=df_track.drop(["key","mode","explicit"],axis=1).corr(method="pearson
")
plt.figure(figsize=(14,6))
heatmap=sns.heatmap(corr_df,annot=True,vmin=1,vmax=1,center=0,cmap="inf
erno")
heatmap.set_title("Corr heatmap between variable")
heatmap.set_xticklabels(heatmap.get_xticklabels(),rotation=90)

sample_df=df_track.sample(int(0.004*len(df_track)))
sample_df.head()

len(sample_df)

plt.figure(figsize=(10,8))
sns.regplot(data=sample_df,y="loudness",x="energy",color="c").set(title="Lou
dness vs Energy corr")

plt.figure(figsize=(10,8))
sns.regplot(data=sample_df,y="popularity",x="acousticness",color="c").set(title
="popularity vs accoustic corr")
```

```
df_track["dates"]=pd.to_datetime(df_track["release_date"])
df_track.info()


df_track["release_date"]=pd.to_datetime(df_track["release_date"])
df_track["year"]=df_track["release_date"].dt.year

df_track.info()

sns.displot(df_track["year"],discrete=True,aspect=2,height=5,kind="hist").set(ti
tle="Number of song per year")

total_dr=df_track["duration"]
fig,ax=plt.subplots()
fig=sns.barplot(x=df_track["year"],y=total_dr,ax=ax,errwidth=False).set(title="
Year vs Duration")

sns.set_style(style="whitegrid")
fig_dims=(10,5)
fig,ax=plt.subplots(figsize=fig_dims)
```

```
fig=sns.lineplot(x=df_track["year"],y=total_dr,ax=ax).set(title="year vs
duration")
plt.xticks(rotation=60)



df=pd.read_csv("C:/Users/DELL/Downloads/SpotifyFeatures.csv")
df.head()


plt.title("Duration of song in diff geners")
sns.color_palette("rocket",as_cmap=True)
sns.barplot(y=df["genre"],x=df["duration_ms"],data=df)
plt.xlabel("Duration in milli second")
plt.ylabel("Geners")

sns.set_style(style="darkgrid")
plt.figure(figsize=(10,5))
famous=df.sort_values("popularity",ascending=False).head(10)
sns.barplot(y="genre",x="popularity",data=famous).set(title="top 10 genres by
popularity")
```

CHAPTER-08

TESTING

8.1 What is Software Testing?

Software testing is an activity to check whether the actual results match the expected results and to ensure that the software system is Defect free. It involves execution of a software component or system component to evaluate one or more properties of interest.

Software testing also helps to identify errors, gaps or missing requirements in contrary to the actual requirements. It can be either done manually or using automated tools. Some prefer saying Software testing as a white box and Black Box Testing.

Different levels of testing are used in the test process; each level of testing aims to test different aspects of the system.

The First Level is **unit testing**. In this testing, individual components are tested to ensure that they operate correctly. It focuses on verification efforts.

The Second Level is **integration testing**. It is a systematic technique for constructing the program structure. In this testing, many tested modules are combined into the subsystems which are then tested. The good here is to see if the modules can be integrated properly.

The Third Level is **integration testing**? System testing is actually a series of different tests whose primary purpose is to fully exercise computer-based system.

Testing is the fundamental process of software success. Testing is not a distinct phase in system development life cycle but should be applicable throughout all phases that is design development and maintenance phase. Testing is used to show incorrectness and considered to success when an error is detected.

8.2 Software Quality Improvement

The computer and the software are mainly used for complex and critical applications and a bug or fault in software causes severe losses. So, a great consideration is required for checking for quality of software.

8.3 Verification and Validation

Verification means to test that we are building the product in right way that we are using the correct procedure for the development of software so that it can meet the user requirements. Validation means to check whether we are building the right product or not.

8.4 Software Reliability Estimation The objective is to discover the residual designing errors before delivery to the customer. The failure data during process are taken down in order to estimate the software reliability.

CHAPTER-09

CONCLUSION

The analysis is intended to understand the characteristics of various genres of music and to identify how popular a songs would be based on its features. Along with that, I have also identified the patterns and relationship between the features that describe the songs in our dataset.

We have performed data wrangling on our Spotify dataset by removing null values, removing duplicates and transforming variables, before starting our exploratory data analysis. From our correlation plot, we observed that variables have strong correlation with each other, indicating that this dataset has multicollinearity.I have also plotted histograms and boxplot to show the relation between the variables. I have then utilized this dataset for clustering analysis to analyse a track's popularity based on several audio features provided in the dataset and find whether we could predict a track's popularity from key features about the song.

We have gained a lot of insights during this journey of exploring the data and the interesting ones relevant in our analysis are: A lot of songs in our data have a popularity score of 0, which means a lot those haven't been explored yet. Energy has high positive correlation with loudness and high negative correlation with acoustic ness. EDM genre has songs with highest energy, Rap genre has the highest dance ability factor .Rap and Latin are most lively and loudness is pretty similar for all genres. Furthermore, Even though we have six genres in the dataset, with the clustering technique I got optimal clusters count as three, which means few genres might be very similar. Lastly, less speech and high energy songs are likely to be popular among users.

With our clustering techniques we could provide more suggestions based on three clusters that we got, which would improve the user experience and finally

we have an idea of which characteristics have an impact on the popularity of the songs and we could probably build more customized playlists.

Our dataset had limited entries, so with a much larger dataset we could probably improve our analysis even more. Also additional data points in the dataset like user demographics and historical data would make our analysis even more reliable

CHAPTER-10

BIBLIOGRAPHY

1. Jake Vander plas, "Python Data Science Handbook: Essential tools for working with data",O'Reilly Publishers, I Edition.

2. Wes Mc Kinney, "Python for Data Analysis", O'Reilly Media, 2012Mark Lutz, "ProgrammingPython", O'Reilly Media, 4th edition, 2010.

Some of the links which were referred during the project

- https://www.kaggle.com/code/mrankitgupta/spotify-data-analysis-using-python

- https://www.youtube.com/watch?v=8d7ywKCm6HI

- S | Kaggle