“SPOTIFY DATA ANALYSIS Using Python With Global Data”

The project is public, you can find it here: [shrish-tech/PAS\_Spotify (github.com)](https://github.com/shrish-tech/PAS_Spotify)

Abstract:

Digital music distribution is increasingly powered by automated mechanisms that contin-

uously capture, sort and analyze large amounts of Web-based data. This paper deals with the man-

agement of songs audio features from a statistical point of view. In particular, it explores the data

catching mechanisms enabled by Spotify Web API, and suggests statistical tools for the analysis of

these data. Special attention is devoted to songs popularity and a Beta model including random effects

is proposed in order to give a ﬁrst answer to questions like: which are the determinants of popular-

ity? The identiﬁcation of a model able to describe this relationship, the determination within the set

of characteristics of those considered most important in making a song popular is a very interesting

topic for those who aim to predict the success of new products

Digital music distribution is increasingly powered by automated mechanisms that contin-

uously capture, sort and analyze large amounts of Web-based data. This paper deals with the man-

agement of songs audio features from a statistical point of view. In particular, it explores the data

catching mechanisms enabled by Spotify Web API, and suggests statistical tools for the analysis of

these data. Special attention is devoted to songs popularity and a Beta model including random effects

is proposed in order to give a ﬁrst answer to questions like: which are the determinants of popular-

ity? The identiﬁcation of a model able to describe this relationship, the determination within the set

of characteristics of those considered most important in making a song popular is a very interesting

topic for those who aim to predict the success of new products

Digital music distribution is increasingly powered by automated mechanisms that contin-

uously capture, sort and analyze large amounts of Web-based data. This paper deals with the man-

agement of songs audio features from a statistical point of view. In particular, it explores the data

catching mechanisms enabled by Spotify Web API, and suggests statistical tools for the analysis of

these data.

Digital music distribution is increasingly powered by automated mechanisms that contin-

uously capture, sort and analyze large amounts of Web-based data.

Digital music distribution is increasingly powered by automated mechanisms that contin-

uously capture, sort and analyze large amounts of Web-based data.

uously capture, sort and analyze large amounts of Web-based data. This paper deals with the man-

Digital music distribution is increasingly powered by automated mechanisms that continuously capture, sort, and analyze large amounts of Web-based data. This paper deals with the management of songs audio features from a statistical point of view. In particular, it explores the data catching mechanisms enabled by Spotify Web API and suggests statistical tools for the analysis of these data. Special attention is devoted to songs popularity and a Beta model including random effects is proposed to give the first answer to questions like which are the determinants of popularity? The identification of a model able to describe this relationship, the determination within the set of characteristics of those considered most important in making a song popular is a very interesting topic for those who aim to predict the success of new products.

Keywords:

Spotify Web API, Audio features, Popularity Index, Beta GLMM, Danceability, Acousticity.

Introduction:

Spotify in today’s age has grown all sorts and is a regular name when streaming/listing songs, podcasts and what not. It has tremendously useful features which curates a better experience for the user which increases the retainability rate of the users. It has over 50 million tracks which user can browse using various parameters like artist, album, genre, or via playlists. It pays the artists or the right holders via royalties which is approximately 70% of their revenue. Thus, it is also a good platform for musicians to not only showcase their talent but also to make some money. It is the world's largest music streaming service provider, with over 381 million monthly active users, including 172 million paying subscribers, as of September 2021. So basically, they store a hell lot of data which is not necessarily in a very good shape or form. This data is cleaned and technically bifurcated by “data scientists/ analysts” which helps the company know what most users love listening to. We in the project will be attempting to do the same, sort of analyze the data using Python which includes many inbuilt libraries which are useful for data analysis such as pandas, SciPy, Scikit-learn. Also, some offer graph-based analysis such as Matplotlib.

Methodology:

From a musician point of view, we have following problem statements:

1. To find popular singer.
2. To find popular genre.
3. To find if there is a relationship between artist and genre.
4. To find the average duration of popular songs.
5. To find if there are specific parameters of a song that makes a particular genre.

Answering these questions might help a musician to find a particular singer and composer song that will increase the probability of becoming popular.

We are using Spotify dataset that was collected from a trustworthy website which is a daily name for data analysts, Kaggle. We begin with the analyses of data structure and cleaning and processing the data for further use. Next, we explore the data to find various relationship between variables using lists, tables, and visualization.

Something about .CSV files before starting……

A Comma Separated Values (CSV) file is a plain text file that contains a list of data. These files are often used for exchanging data between different applications. For example, databases and contact managers often support CSV files.

These files may sometimes be called Character Separated Values or Comma Delimited files. They mostly use the comma character to separate (or delimit) data, but sometimes use other characters, like semicolons. The idea is that you can export complex data from one application to a CSV file, and then import the data in that CSV file into another application.

In our case, the .csv file looks something like,Text

Description automatically generated

Having seen that, pretty much vague looking data, there seems a strong need to analyze the data and make it understandable to a common person.

Thus, we first cleaned the data which included several steps then displayed the initially clean data and the made some graphical representations of the same.

The data set contains the following fields:

* Highest Charting Position
* Number of times charted
* Week of highest charting
* Song Name
* Streams
* Artist
* Artist followers
* Song ID
* Genre
* Release Date
* Weeks Charted
* Popularity
* Danceability
* Energy
* Loudness
* Speechiness
* Acousticness
* Liveness
* Tempo
* Duration (ms)
* Valance
* Chord

NOW DIRECTLY JUMPING ON THE CODE AND THE RESULTS.

>>Next Page>>

(Step 1) Importing all the libraries required for data analysis.

Shape, rectangle

Description automatically generated

(Step 2) Importing the data frame

Table

Description automatically generated

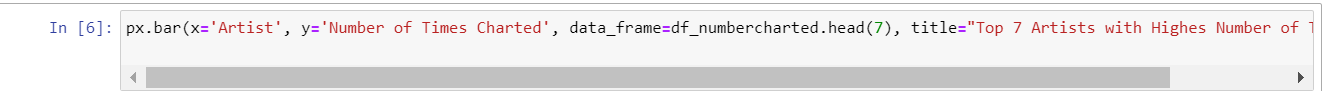
Table

Description automatically generated

Table

Description automatically generated

Top 7 Artists with Highest Number of times charted:



**Chart, bar chart

Description automatically generated**

#Checking Correlation and Cleaning DataChart, scatter chart

Description automatically generated



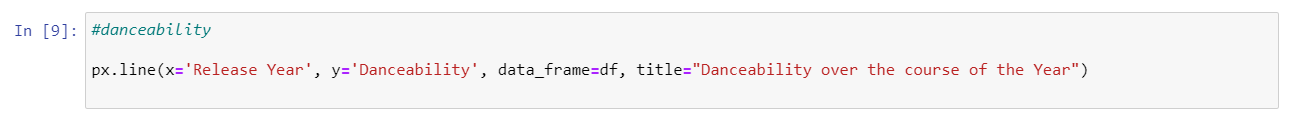
>> Next Page>>

#HEATMAP

Background pattern

Description automatically generated with low confidence

# Dancability



>> Next Page >>

**Chart

Description automatically generated**

# Number of Times Charted

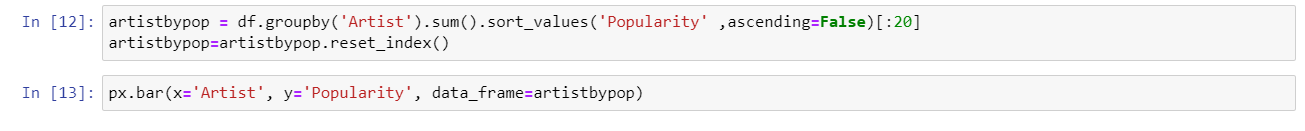
Graphical user interface, text

Description automatically generated

**Chart, bar chart

Description automatically generated**>> Next Page>>

# Popularity of Artists



**Chart, bar chart, histogram

Description automatically generated**

# Most Popular Genres

Graphical user interface, text, application, email

Description automatically generated

>> Next Page>>

The Final sorted and analyzed and processed data…..

Table

Description automatically generated

Conclusion:

1. The objective was to predict the popularity of any song.
2. Performed exploratory data analysis to derive insights from the data.
3. Mostly 2000 popular songs are identified and added every year in Spotify.
4. The most popular artist from 1921 to 2021 is Tayler Swift.
5. The most popular song is Beggin’ by Menisskin.
6. The data was split into train (80%) and test (20%) datasets for model building and evaluation, respectively.
7. Using GridSearchCV, we could find the optimal hyperparameters for the decision tree regressor model and achieved an accuracy of 76.6%.

>>Next Page>>

References:

1. Spotify API: [Privacy settings - Spotify](https://www.spotify.com/in-en/account/privacy/) (You need to put a request for their API, Dataset).
2. Sample datasets: [Dataset of songs in Spotify | Kaggle](https://www.kaggle.com/mrmorj/dataset-of-songs-in-spotify)
3. Courses referred: [Data Analysis with Python | Coursera](https://www.coursera.org/learn/data-analysis-with-python)
4. Course completion certificate: <https://coursera.org/share/dc41ad794bcef4fe785243576e25e9a2>
5. [Data Analysis with Python Course: Part 1 of 6 (हिंदी में) - YouTube](https://www.youtube.com/watch?v=YV0_puRO1u4)
6. Libraries:

* [pandas - Python Data Analysis Library (pydata.org)](https://pandas.pydata.org/)
* [Matplotlib — Visualization with Python](https://matplotlib.org/)
* [SciPy](https://scipy.org/)
* [NumPy](https://numpy.org/)
* [seaborn: statistical data visualization — seaborn 0.11.2 documentation (pydata.org)](https://seaborn.pydata.org/)
* [Plotly Python Graphing Library | Python | Plotly](https://plotly.com/python/)

1. Special Editor for the code: [Project Jupyter | Home](https://jupyter.org/)
2. Taking ref from some industry/working persons: <https://www.linkedin.com/>
3. The open book: [www.google.com](http://www.google.com)
4. GITHUB: [GitHub](https://github.com/)