**Project Report\_Kimela Conway**

# **GitHub URL**

**https://github.com/KimelaC/UCDPA\_KimelaConway**

# **Abstract**

This purpose of this project is to study the collection, analysis and interpretation of dataset Top 100 Songs 2010 to 2019.

I will illustrate my process using python by outlining how data is collected, the preparation and validation of the data, the analysis of data including use of data visualisation to explore data and thus draw conclusions about this data.

# **Introduction**

I choose this project use case as I have genuine interest in music. I recognized the data was labeled well and would be amenable for merging to another dataset but also allow for a varied type of analysis. The only point of note is its unclear from the data source whether this is global top 100 playlist or whether its regional to U.S. or Canada.

The other interesting thing about the data is analysis was possible on how energetic, acoustic or loud the top most popular songs were not just the year they were in the top 100 or the most popular genres which I definitely found more interesting.

# **Dataset**

I choose the data set Spotify Top 100 2010 to 2019 from Kaggle as I love music and figured I would analyse data that was most interesting to me. The other points of note were the size of the dataset met the project requirement threshold. The data set had continuous values, and clear data within columns that would allow for adequate analysis within python.

Source: <https://www.kaggle.com/datasets/muhmores/spotify-top-100-songs-of-20152019>

# **Implementation Process:** Describe your entire process in detail

I first created new jupyter notebook and set out creating the environment I needed to work. I did this by importing the following libraries available in python.

import pandas as pd : Used to read excel or csv file along with data processing

import numpy as np : recommendation system

import matplotlib.pyplot as plt : data visualisation method

import seaborn as sns : data visualisation method

import requests : Used to request API or web scrape

Next I imported my selected resource excel file into jupyter notebook using the pd.read method, I then assign name to my dataframe.

I then want to get a view of my data so call on a number of methods to identify, number of rows, no. of columns, data types within the dataset and column headings.

df.info()

1. Shows the dataset consists of 1000 rows and 17 Columns.

1. Data type for columns no. 3, 5-15 are integer' including ' year released'
2. data type for all other columns is 'object'
3. Data appears to have no missing values

df.head()

1. Shows the first 5 rows of the dataset
2. column 'artist type' returns more than one value

df.tail()

1. Shows the last 5 rows of data

2. Column type seems consistent in each row but note ‘artist type’ and ‘top genre’ have more than one value.

df.columns

This provides description of column names; it is here that I identify that column names do not immediately relate to column headings and check data source again to insert descriptions and decide to alter column names to clarify the true meaning of the data within the columns.

df.describe()

This method returns description of the data in the Data Frame. If the Data Frame contains numerical data, the description contains these information for each column: count - The number of not-empty values. mean - The average (mean) value.

1. #oldest song: 1975 year of release
2. #youngest song: 2021 year of release
3. #mean song: 2014 year of release

df.rename

As discussed up above, I’m using this method to rename by column headings to provide clarity around the data content within the column so that when I perform further analysis on the data it will be clear and interpreted correctly. E.g. ‘Val, represents How positive a song is which is not very clear so have changed name to ‘Mood Positivity’ so data can be interpreted more accurately.

I recalled the head.() method again to ensure column names were changed.

**Preparation**

Spotify\_df = df.copy() -To prepare data frame for analysis I assign new name to data frame by using .copy() method.

Spotify\_df.info() - I then call on .info() to recheck information in resulting data frame matches df data frame.

Spotify\_df.duplicated() – I use this method to check for duplicates within my data frame. It returns a Boolean value (True/False) but does only return head and tail of data frame.

Spotify\_df.duplicated().sum()

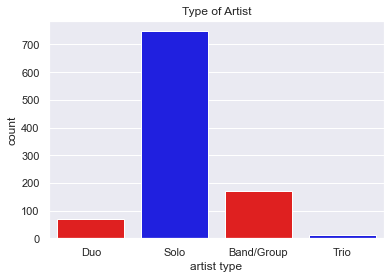
To confirms there are no duplicates within the entire dataset I include .sum() with the .duplicated() method and this results in ‘0’

**Results:**

Graph No.1

Given the different values available in ‘artist type’ I was first intrigued to identify which artist type was most prevalent in the top 100 list over the years. Solo, duo, trio or band.

This chart clearly shows of the 1000 row data set 750 of artists are Solo Artists who outweigh duo's, bands, or groups and trios significantly in terms of songs that have hit the top 100 playlist between 2010 to 2019.



Graph No. 2

Next I looked at the number of songs from the top 100 playlists that sit in each genre

The graph below clearly shows top four genres within the top 100 songs between 2010 and 2019 is Dance Pop, pop, hip hop, and art pop with the remaining genres evenly matched across the stats. This result highlighted the high number of different genres in the data frame.

Chart, bar chart

Description automatically generated with medium confidence

count\_by\_genre = Spotify\_df['top genre'].value\_counts() - I decided to explore further exactly how many different genres there were within the dataframe.

Graph 3

I next plotted scatter graph to see if I could see a correlation between ‘how positive a song was? Versus how danceable it was? You will note from results below there definitely is some correlation between the two.

Chart, scatter chart

Description automatically generated

**Graph 4**

I next looked at the variance between the ‘date a song entered the top 100 playlist’ versus the year the song was released. There was often quite a large variance between these dates with the obvious outlier being the hit released in 1975 but entering the playlist in 2020.

A picture containing shape

Description automatically generated

**Graph 5**

sns.lineplot(data=Spotify\_df, x='top year', y='artist type')

I then looked at ‘artist type’ within the top 100 list each year which showed consistent fluctuations but not surprisingly higher level of solo artist given 750 of the 100 rows were solo artists.

A picture containing chart

Description automatically generated

**Duration**

Spotify\_df['duration of song'].value\_counts() – Then used count method to check the different durations of songs within the top 100. This showed

Artist\_df = Spotify\_df[(Spotify\_df['artist type'] == 'Solo')] – To explore duration of song further I subset ‘solo artist’ from artist type to create new data set.

Artist\_df.head() – check information within new dataset to ensure adequate data available to check for duration of songs.

Artist\_df['duration of song'].value\_counts() – use count method to check duration of songs relating to solo artist and note the results are similar to count completed on entire dataset which is expected given the largest number of rows on data set was for ‘solo artists’.

Graph 5

I use new data frame Artist\_df to create a graph using seaborn to illustrate the distribution of duration of songs for solo artists.

I note the largest number of songs are between 175 (2 mins 55 sec) to 220 seconds (3mins 40 seconds).

I note these timeframes are logical but wonder has this changed over the years.

Chart

Description automatically generated

Graph 6

Then use groupby method on subset data frame ‘Artist\_df’ to view a trend of song duration by year for solo artists. This shows steady decline in duration of a song from 2014 onwards. The increase shown in 2020 may not be reflective of trend given data was only entered up until 2019.

Chart, line chart

Description automatically generated

Graph 7

I further explore no of songs that reach top 100 list released in each year by solo artist by calling count method on songs and sort data to reveal trend shows fluctuations in number of songs released by solo artists between 2009 to 2016

There was a sharp increase in songs released from Solo Artists that hit the top 100 lists between 2016 and 2018

The peak release year was 2018 before sharp fall in 2019 and 2020, however data gathered only includes songs up and including 2019 which would explain this decline

Chart, line chart

Description automatically generated

**Merging**

In order to merge my data frame to new dataset I read in new csv file - Streams = pd.read\_csv('C:\\Users\\kim.conway\\Spotify\_final\_dataset.csv')

Streams.head()

I then have a look at the data by using method head() and note similar column headings within the dataset.

Streams.info()

To further investigate the column titles I call .info() to identify no. of columns

And I note two columns similar data content but different column headings 'Artist name' v 'Artist' and 'Song Name' v 'title'

I call the .rename method used previously to change column names so that I can merge the datasets using an inner join as we know have columns in common.

Streams.info() -I call the info method again to check column names have been changed.

Streams.isnull().sum() – check for null values – found 4 in column ‘title’

Although I know there are only 4 missing values I want to demonstrate a custom function so I identify % missing rows in each column. This identifies only a tiny percentage of missing values in ‘Song Name’ column which if dropped would not affect the entire dataset.

I use the .dropna() method to drop all rows with missing values and overwrite the dataset

Streams.isnull().sum() – I call .isnull().sum() to confirm missing rows have been removed.

Having now checked my second dataset I merge my two data frames using inner join on the column ‘title’ which now appears in both datasets. I call head to review first five lines of data.

Spotify\_Streams\_df.isnull().sum() – I do final check to ensure there is no missing values within the newly merged dataframe

Spotify\_Streams\_df.shape – call this method to review new data frame and not the additional columns and reduction in rows.

I sort the newly formed data frame in descending order, so as output should reflect most popular song to be streamed. I call head.(20) so I can see the top 20 records and immediately note repeat in some artists, i.e. Drake appeared 5 out of the 20 top 20 streams.

# **Insights**

1. My first insight was in to the importance of identifying relevant dataset for your analysis and how important it is to know as much as possible about your resource. Knowing whether the top 100 playlist was regional versus global would give a new viewpoint to the resulting analytics carried out on this dataset.
2. It is evident from the dataset that the ‘solo artists’ appear to release 75% of songs that appear in the top 100 Spotify playlist. This valuable analytical information could be used to predict success of future songs based on ‘artist type’
3. When considering what the most popular genre, we identified one clear winner ‘dance pop’ which represented 350 of the total dataset records. Second genre was ‘pop’ 60 and third ‘dance pop’ 40, all other genres were 35 and below and had fairly even distribution of songs within the top 100. I assume these classifications are as set by ‘Spotify App’ which is another thing to consider when considering the analytical data.
4. Duration of song was another interesting analytic and showed most songs are between 175 (2 mins 55 sec) to 220 seconds (3mins 40 seconds). When we looked at the trend between 2010 to 2019 in relation the duration of a song we could see there was a downward decline from 2013 onwards. The peak at the end I suspect would change once ore data from 2020 was entered into the dataset.

Given today’s trend of short form content this is not surprising, the success of new apps like ‘tik tok’ and updates to ‘you tube’ with ‘shorts’ versus long form content it is apparent that today’s society enjoys entertainment in shorter spurts and it is no surprise the duration of a song has followed this trend.

1. The final insight I want to share is in relation to How positive a song is and how danceable it is, this dataset could provide many more analytics in relation to use of spoken word, how loud the most popular songs were etc. but the correlation between how positive a song is danceable it is are undeniable. This interests me as I believe music has a very positive effect on mental health and this dataset if explored further could go further in supporting that belief with this analytic given an initial insight to that possibility.

**Machine Learning**

Given Deep Learning uses a complex structure of algorithms, requires a large amount of data and allows for the processing of unstructured data such as documents, images and text I would opt for machine learning to create liner correlations within my smaller dataset.

Machine Learning Regression is a technique for investigating the relationship between independent variables or features and a dependent variable or outcome. It's used as a method for predictive modelling in machine learning, in which an algorithm is used to predict continuous outcome and is most suited to the data frames above allowing predictions on ‘top songs released by solo artists’, likely ‘genres’ and future trends in durations of songs released in future years.

# References