# Song Data Analytics

## Table of Contents

1. Objective
2. Result
3. Future Development
4. Repository
5. Data Source
6. Deliverables
   1. Single Bash Script
   2. Exploratory Data Analysis (EDA)
   3. Song Analytics Data Challenge Pipeline
   4. Questions and Answers
7. Validation

## Objective

Set up reproduceable and semi-automated sets of extraction, transformation, and loading operations to answer questions specified by the instruction utilizing commonly used scripting languages

## Result

All questions are provided with answers and some discussion, as well as adequate explanations about the code and thought process used to navigate the questions. Some alternative algorithms are also provided.

Results can be found in section 6. Section 6 also includes other main deliverables, questions and answers, and data validation. The scripts are in Python 3.0 and Jupyter Notebook. File configurations, formatting’s and server-side application are based on JSON as an open-standard file format.

## Future Development

The items highlighted as follows can be completed as needed.

1. Data Ingestion Automation from the data source:

Dataflow between the source and the designated data warehouse (the destination system) needs to be designed for this project. Data can be extracted, downloaded and loaded in different manner including continuous, asynchronous, real-time or batched. Depending on the source format and protocol, the process may require some additional type of transformation or conversion. Incremental batch loading with a given frequency, or bulk batch with truncating and reloading are recommended based on the data’s size, source, and latency.

1. Integration with the Front-end and Deployment of the model server:

For the client’s front-end a server/instance is needed to be dedicated/shared to the task/schedule.

A Single Page Application (SPA) is a common choice for building front-end applications for this task. It can be built through using AngularJS or React frameworks. For a faster turnaround a micro web framework for Python, such as Flask, is recommended for building the RESTful APIs. Using the Flask and Server-side sessions and serializing with Pickle library is aligned with the standard and best practice of containerization in the production environment. This enables the model to be readily available if/when production phase starts.

## Repository

A public GitHub repository located at the link below contains all component deliverables. Your team should be able to clone this repository and reproduce the analysis.

All configurations and requirements can be found in the repository along with this document.

The link to the git-hub Repository: <https://github.com/BijanVafaei1992/Song_Analytics>

## Data Source

The available data source was provided in csv format. It can be accessed through the project repository at the link below;

<https://github.com/BijanVafaei1992/Song_Analytics/blob/master/data/chart2000-songmonth-0-3-0054.csv>

The dataset stores Music Billboard Chart data across multiple countries from Jan 2000 to June 2019. The Dataset creates an aggregated scoring system to assess how popular music was performing worldwide in each month. Each row of data in the chart represents a song’s placement on the top music charts in one given month, including the artist, its aggregated score, and its position on the charts in various countries around the world. Each datapoint may provide the position of the song in one given month in six countries of US, Germany, France, Canada, and Australia. The countries’ specific position may be blank which is indicated by “-”. There were 11,700 datapoints available in this dataset with 11 features. A quick QA/QC and cross validation was conducted and the datapoint provided in the dataset was assumed accurate.

TABLE 1 – The Snapshot of the Music Chart Dataset (Jan 2000 to June 2019)

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **index** | **month** | **position** | **artist** | **song** | **score** | **us** | **uk** | **de** | **fr** | **ca** | **au** |
| 0 | Jan-00 | 25 | Will Smith | Will 2K | 1750.797 | 57 | 21 | 62 | 32 | - | 3 |
| 1 | Jan-00 | 11 | Whitney Houston | My Love Is Your Love | 2772.888 | 4 | - | - | 97 | - | - |
| 2 | Jan-00 | 23 | Train | Meet Virginia | 1838.804 | 20 | - | - | - | - | - |
| 3 | Jan-00 | 28 | TLC | Unpretty | 1697.405 | 25 | - | - | - | - | 41 |
| 4 | Jan-00 | 43 | The Goo Goo Dolls | Black Balloon | 1478.309 | 33 | - | - | - | - | - |
| 5 | Jan-00 | 22 | The Foo Fighters | Learn To Fly | 1883.219 | 19 | - | - | - | - | 48 |
| 6 | Jan-00 | 46 | The Dixie Chicks | Cowboy Take Me Away | 1428.473 | 27 | - | - | - | - | - |
| 7 | Jan-00 | 44 | The Backstreet Boys | Show Me The Meaning Of Being Lonely | 1457.522 | 16 | 97 | - | - | - | - |
| 8 | Jan-00 | 39 | Sugar Ray | Someday | 1557.48 | 27 | - | - | - | - | - |
| 9 | Jan-00 | 16 | Smash Mouth | Then The Morning Comes | 2161.426 | 11 | - | - | - | - | - |
| … | … | … | … | … | … | … | … | … | … | … | … |
| 11695 | Jun-19 | 8 | Ava Max | Sweet But Psycho | 1661.839 | 10 | 45 | 39 | 69 | 12 | 32 |
| 11696 | Jun-19 | 14 | Ariana Grande | 7 Rings | 1346.183 | 12 | 80 | 99 | 27 | 9 | 30 |
| 11697 | Jun-19 | 35 | Ariana Grande | Break Up With Your Girlfriend, I'm | 875.253 | 20 | 96 | - | - | 33 | - |
| 11698 | Jun-19 | 38 | A Boogie Wit da Hoodie | Look Back At It | 849.373 | 28 | - | - | 45 | - | - |
| 11699 | Jun-19 | 43 | 5 Seconds of Summer | Easier | 781.932 | 48 | 27 | - | - | 37 | 12 |

A summary of descriptive statistics that quantitatively describes the features of the dataset is provided in the Table-2 as shown below.

TABLE 2- Summary of Descriptive Statistics of the features

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Features** | **Data Type** | **No. of Value** | **No. of Null Value** | **No. of Unique Value** | **Mode** | **Frequency** | **mean** | **std.** | **min** | **50%** | **75%** | **max** |
| month | Datetime | 11700 | 0 | 234 | - | 50 | - | - | Jan 2000 | - | - | June 2019 |
| position | Integer | 11700 | 0 | 50 | - | - | 25.5 | 14.43 | 1 | 25.5 | 38 | 50 |
| artist | String | 11700 | 0 | 1722 | Rihanna | 123 | - | - | - | - | - | - |
| song | String | 11700 | 0 | 2964 | Perfect | 23 | - | - | - | - | - | - |
| score | Float | 11700 | 0 | - | - | - | 1366.7 | 710.3 | 343.3 | 1225.7 | 1688.2 | 5524.5 |
| us | Float | 10813 | 887 | 101 | - | - | 20.96 | 17.15 | 1 | 17 | 29 | 100 |
| uk | Float | 6667 | 5033 | 101 | - | - | 29.37 | 25.31 | 1 | 22 | 44 | 100 |
| de | Float | 6149 | 5551 | 101 | - | - | 28.45 | 25.6 | 1 | 21 | 44 | 100 |
| fr | Float | 5062 | 6638 | 201 | - | - | 44.4 | 44.76 | 1 | 29 | 63 | 200 |
| ca | Float | 5708 | 5992 | 101 | - | - | 19.18 | 14.85 | 1 | 16 | 28 | 100 |
| au | Float | 5771 | 5929 | 51 | - | - | 18.16 | 14.03 | 1 | 15 | 29 | 50 |
| Total N0. of Rows | | 11700 |  |  |  |  |  |  |  |  |  |  |

Data can be loaded through the standard procedure as described below using the JSON format and Pandas framework’s ability to read csv formatted files.



## Deliverables

The main deliverables of this submittals are a single bash script, ETL pipeline, Exploratory Data Analysis (EDA), Preprocessing, Encoding, and Data Manipulation (as needed), and the Song Analytics Data Challenge Pipeline.

The model also provided some alternative solutions as well as more in-depth details on the answers and also expanded on data exploratory/explanatory.

### Single Bash Script

a single bash script named "run.sh" that will allow you to run the entire pipeline:

run.sh can be found in the root directory of the repository. The following script is simply used in the bash script.

### Exploratory Data Analysis (EDA)

The repository contains the following Jupyter notebook(s) for exploratory / explanatory components. Comments are provided in the notebook(s): Song\_Analytics\_EDA.py.

Some explanatory and analytical components are provided which lead to the preprocessing and better understanding the dataset.

Any additional instructions necessary to ensure that the team can reproduce the result or to direct their attention to the right places can be provided upon request for expansion.

### **Song Analytics** Challenge Pipeline

The goal was defined to set up a reproduceable set of extraction, transformation, and loading operations to answers the questions given in the instruction document.

The class of “Song Analytics Data Challenge” from “Song Analytics” containing five generic, eight question-specific, and one plotting functions was prepared. The “Song Analytics Data Challenge” Class, and its underlying functions were imported, and utilized in the Model\_run.py to provide the output for each question.

Five generic functions were defined to conduct the following tasks;

1. Reading and loading the data to the Pandas dataframe: read\_data
2. Showing the unique values of a given feature (column): unique\_value
3. Filtering the dataframe to its desired feature(s): filter\_col
4. Filtering the dataframe to the desired value(s) in each feature(s): filter\_value
5. Aggregating the dataframe through group by and count: agg



In addition to the generic functions, eight question-specific functions, and one plotting function were designed to provide the answers for each question.

### . Questions and Answers

Questions and answers are as follows,

#### Question 1

a) How many unique artists are represented in the charts?

b) How many unique songs are represented?

#### Answer 1

Function q1 is designed to pass as many as arguments desired to the unique\_value function and return the number of unique values for those arguments.

By Passing the name of ‘artist’, and ‘song’ features to q1 function, it returns the number of unique value for the artist and song features.



There are 1722 artists represented in the charts from January 2000 to June 2019, and there are 2964 unique songs in the charts for the same period.

#### Question 2

a) What song has the highest one-month score of all time?

b) What month was this?

c) Who was the artist?

#### Answer 2

Q2 function is defined to find the maximum score exist in the dataframe and filter the dataset through using the filter\_value function with the maximum score as the argument. Q2 function returns the required features for the selected song.



The song named ‘Yeah!’ by Usher, Lil’ Jon & Ludacris has the highest one-month score of all time approximately about 5524.5 point. This song was No. one in April 2004.

#### Question 3

As I am writing this, I am listening to the song “Safe and Sound” by Capital Cities.

a) How many months was this song on the charts in the United States?

b) What was its peak US chart position?

c) In what month did it achieve this peak US chart position?

#### Answer 3

Q3 Function receives as many as desired key value pairs (\*\*kwargs) and pass those through the filter\_value function to identify the related datapoints to the given attributes (in this questions song and artist). The specified song/artist is called the favorite song in this function. Q3 Function finds the number of months that the favorite song was on chart in US by using to\_list () method of the series of fav\_song[‘month]. Then the function identifies the best position that the favorite song hits in US through utilizing the .idxmin() method and in the end, it finds out what month was the best month for this song that it achieve the peak position in the US chart.

****

‘Safe And Sound’ song by ‘Capital Cities’ was on US chart for 10 month and it secure the 8th position on the US chart as the best position in September 2013.

#### Question 4

a) Which artist has spent the most time on the charts as defined by chart-months?

b) How many chart-months have they scored?

#### Answer 4

In order to answer to this question, q4 function is designed to aggregate the dataframe on the ‘artist’ feature and count the number of months that the artist has been on the chart. After aggregating the dataset, the function finds the artist with most number of months on chart through using .iloc() and .idxmax() methods. Q4 Function returns the name of the artist with the highest number of months on chart and the total number of month he/she stayed on the chart.

Per the definition of the chart\_month from the instruction, a “chart-month” is a row in the data that represents one song spending one month on the charts. If a song is represented on the chart for multiple months, it collects multiple chart\_months as many months as it is stays on the chart. If an artist has multiple songs on the chart, the artist will receive the same number of chart\_months as many songs as they have on the chart.



Rihanna was present on the charts for 123 months as the artist with the most time spent on the charts.

Rihanna’s record is followed by Tylor Swift, Pink, Katy Perry and the Black Eyed Peas. The top five artists with the most time spent on the charts and the number of months they stayed on the charts are shown in Table 3.

TABLE 3 – Top Five Artist Who Spent the most time on the Charts

|  |  |  |
| --- | --- | --- |
| **Rank** | **Artist** | **No. of Month on Charts** |
| 1 | Rihanna | 123 |
| 2 | Taylor Swift | 109 |
| 3 | Pink | 107 |
| 4 | Katy Perry | 93 |
| 5 | The Black Eyed Peas | 87 |

#### Question 5

How many #1 hits does Drake have in Canada?

#### Answer 5

For this question, q5 function will receive the dataframe as well as several key/value pair. The q5 function will call filter\_value function and pass both the dataframe as well the key/value pairs and return the number of incidents that match with the input.



In this case, dataframe will be filtered out on Drake as the artist and the number of first position he achieves in Canada. The number of times that Drake hits #1 in Canada is up to 10.

#### Question 6

Which Lady Gaga song on the list did not chart in Australia?

#### Answer 6

Q6 Function is designed for this question to pass the dataframe and the desired number of key/value pairs to the filter\_value and agg functions as show below. First q6 function filter the dataset to the targeted artist, and then, it aggregates the datapoint to show the sum of the position in the targeted country for each song. Furthermore, it filters out the aggregated dataframe to find the list of songs with no position in chart. No position in the chart can be translated into sum of the position is equal zero. In order to get to this translation, the dataframe was modified slightly. In this question, it is required to filter the Null value (Nan in Pandas/Numpy frameworks) to find what Lady GaGa’s songs did not made to the Australia’s chart. All the null values were replaced with the nohit variable, equal to zero, through using the fillna methods of the dataframe. Before the modification, the main dataframe (df) was copied to a temporary dataframe (df\_q6).



Lady Gaga alone has appeared on chart for 68 months. If including all the combination of artists and featured songs she has, This record increases to 101 months. She has released 15 songs with a position on the charts worldwide, but the “Marry The Night” song did not make to the Australia’s Chart not even for at least one month. “Marry The Night” made the 11th position in Canada.

TABLE 4 – Lady GaGa’s Songs with accumulative rankings in Australia’s Chart

|  |  |  |  |
| --- | --- | --- | --- |
| **No.** | **artist** | **song** | **au** |
| 1 | Lady GaGa | Alejandro | 65 |
| 2 | Lady GaGa | Always Remember Us This Way | 12 |
| 3 | Lady GaGa | Applause | 27 |
| 4 | Lady GaGa | Bad Romance | 107 |
| 5 | Lady GaGa | Born This Way | 51 |
| 6 | Lady GaGa | Judas | 20 |
| 7 | Lady GaGa | LoveGame | 79 |
| 8 | Lady GaGa | Marry The Night | 0 |
| 9 | Lady GaGa | Million Reasons | 47 |
| 10 | Lady GaGa | Paparazzi | 45 |
| 11 | Lady GaGa | Perfect Illusion | 14 |
| 12 | Lady GaGa | Poker Face | 126 |
| 13 | Lady GaGa | The Cure | 48 |
| 14 | Lady GaGa | The Edge Of Glory | 86 |
| 15 | Lady GaGa | You & I | 33 |

#### Question 7

a) In what month of her career did Ariana Grande have her highest-ever number of songs

on the charts?

b) In this month, how many songs did she chart?

c) What were these songs ranked in the US, and which was the highest?

#### Answer 7

Q7 function is designed to receive the dataframe and the name of artist, and to call the filter\_value and agg functions. It groups the datapoint by the chart month and returned the aggregated number of songs for each chart month. The aggregated number of songs in “agg\_df\_artist\_month” dataframe shows how many songs the artist has placed on charts in each month. The function calculates the maximum value for the number of released songs appeared on charts in one month and with that captures the month in which the artist has released the highest-ever number of songs on chart using the filter\_value function. With this new variable, q7 function filters out the values and the columns of the dataframe using filter\_value and filter\_col functions and places the filtered values and columns in a new dataframe named “artist\_metrics”. Q7 function sorts the artist\_metric dataframe and returns it.



Ariana Grande had the highest ever numbers of songs on the charts in February 2019. In this month she had four songs on the charts. The name of songs and their position of the US chart ranking can be found in the Table 5.

Her songs have been #1, #2, #3, and #18.

TABLE 5 – Ariana Grande’s Songs on Charts and US Position in February 2019

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Song** | **Month** | **Artist** | **Song** | **US** |
| 1 | Feb 2019 | Ariana Grande | 7 Rings | 1 |
| 2 | Feb 2019 | Ariana Grande | Break Up With Your Girlfriend, I'm bored | 2 |
| 3 | Feb 2019 | Ariana Grande | Thank U, Next | 3 |
| 4 | Feb 2019 | Ariana Grande | Breathin | 18 |

Ariana Grande had made 3 hits on chart during four months of March 2019, January 2019, November 2018 and October 2018.

TABLE 6 – Ariana Grande’s number of songs on chart in one month

|  |  |  |
| --- | --- | --- |
| **Item** | **Month** | **No. of song** |
| 1 | Feb-19 | 4 |
| 2 | Mar-19 | 3 |
| 3 | Jan-19 | 3 |
| 4 | Nov-18 | 3 |
| 5 | Oct-18 | 3 |

#### Question 8

a) Which ten artists had the most unique songs make the charts between 2013 and 2017?

b) Make a bar graph that includes each artist and how many songs they charted. Please

label, title, and color your graph appropriately.

#### Answer 8

Q8 function was prepared for this question as shown below. It utilizes the filter\_col function to reduce the size of dataframe following by a modification on the month feature in order to make it ready for the designed mask. The mask is a Boolean variable to filter the dataset to just have the data between the given dates (start\_date and end\_date). The defined mask variable is used to create the dataframe for the targeted dates, between start\_date and end\_date. The datapoints in the df\_yr can be grouped by artist name to show the number of unique songs each artist got on chart. The .agg(‘nunique’) method makes this computation readily available.

Finally, the q8 function sorts and returns the aggregated dataframe of the desired period (df\_yr\_agg). The output of this function is the top ten artists with the highest number of unique songs on the charts during the desired period.



The top ten artists with the highest number of unique songs represented on charts between 2013 to 2017 are shown in the Table 6. It was anticipated these artists should be remarkable and well-known internationally. Ed Sheeran with 11 songs had the greatest number of unique songs released and on charts. Taylor Swift and One Direction are following Ed with 10 songs during the period. Bruno Mars, Drake and Rihanna race after the mentioned music producers and singers. Selena Gomez, Avicii each secure 7 unique songs on charts. Two famous band, Maroon 5 and Florida Georgia Line are also made to this list.

TABLE 7- Top Ten Artists with the highest number of unique songs on Charts (2013 – 2017)

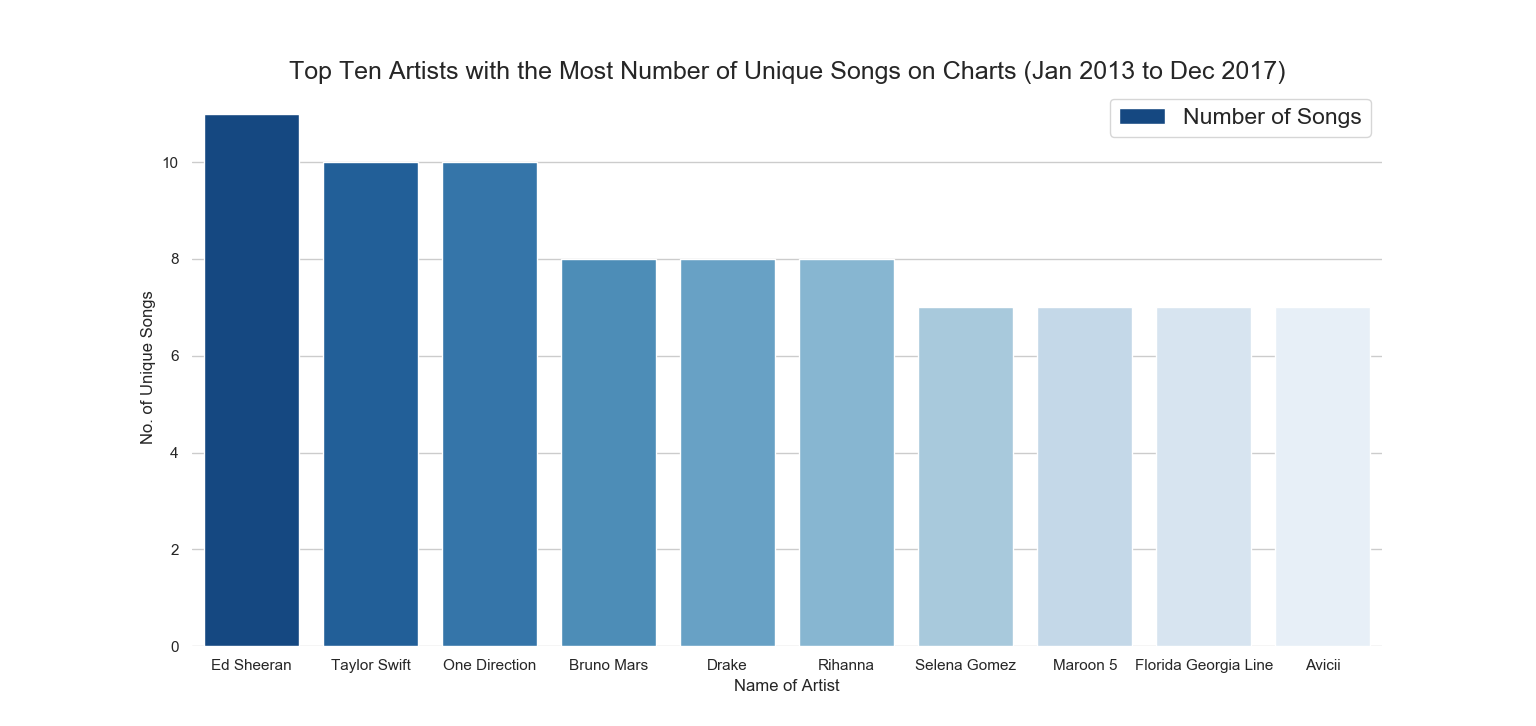
|  |  |  |
| --- | --- | --- |
| **Rank** | **artist** | **song** |
| 1 | Ed Sheeran | 11 |
| 2 | Taylor Swift | 10 |
| 3 | One Direction | 10 |
| 4 | Bruno Mars | 8 |
| 5 | Drake | 8 |
| 6 | Rihanna | 8 |
| 7 | Selena Gomez | 7 |
| 8 | Maroon 5 | 7 |
| 9 | Florida Georgia Line | 7 |
| 10 | Avicii | 7 |

#### Bar Graph

The q8\_plot function creates the bar graph for the presented ranking in the Question 8 section. To this end, seaborn and matplotlib libraries are imported. Plot size, plot style, plot context, font size and line style, title, legend, and axis’s labels were all setup.



FIGURE 1- Bar Graph showing the Top Ten Artist with the most number of Unique Songs on Charts (2013 - 2017)



## Validation

Data Quality and Cross Validation (CV)

To evaluate the performance of any future machine learning model, understanding the quality of data is imperative. An overview of data cross validation was performed on the available data source.

To ensure data have undergone cleaning processes automated "validation rules", "validation constraints" and "check routines", that check for correctness, meaningfulness, and security of data are suggested.Availability, accessibility, usability, structure, reliability and consistency of data source are six criteria’s that needs to be followed. The chart-month data were well available, accessible, and well-structured. The reliability and consistency of the data will be checked by a unit-test and Integration-test approaches as well as Data Schema approach. To validate this data using the proposed system, the user defines a set of checks including (i) completeness and consistency (for example: artist\_ID and song\_title columns should have no missing values), (ii) uniqueness (for example: each row of combined artist\_ID and song\_ title value should be unique), and (iii) counting (for example: number of distinct values in the song\_title column should be less than the total number of songs in the system). Once the constraints are specified, the system converts them into actual computable metrics. For example, completeness would convert into a “fraction of non-missing values in a column” metric. The last step is to generate a report to show how all the constraints fared. The report also lists the ones that failed, along with the value that triggered a failure.

The following Cross Validation examples are interesting and worth mentioning. Following each constrain defined by the data scientist, there is an interesting song title:

An Arbitrary Constrain: Month, year, released date, numbers phone numbers should not be accidentally stored in the song title column. Checking that through a quick query, the following datapoint are some examples found in the song title column. Cross validating these songs through for example a simple google search can prove that these songs title are correctly stored.

TABLE 7- Interesting Song Titles

|  |  |  |  |
| --- | --- | --- | --- |
| **month** | **position** | **artist** | **song** |
| 3-Jun | 4 | 50 Cent & Nate Dogg | 21 Questions |
| 5-May | 22 | Amerie | 1 Thing |
| 19-Feb | 1 | Ariana Grande | 7 Rings |
| 15-Jan | 21 | Beyonce | 43657 |
| 3-Feb | 2 | Beyonce & Jay-Z | '03 Bonnie & Clyde |
| 18-Oct | 35 | Bonez MC & RAF Camora | 500 PS |
| 4-Nov | 28 | Bowling For Soup | 1985 |
| 9-Dec | 11 | Britney Spears | 3 |
| 17-Jan | 5 | Bruno Mars | 24K Magic |
| 5-Jan | 4 | Ciara & Missy 'Misdemeanor' Elliot | 1-2 Step |
| 2-Mar | 21 | Craig David | 7 Days |
| 17-Jul | 6 | David Guetta & Justin Bieber | 2U |
| 7-Oct | 15 | Feist | 1, 2, 3, 4 |
| 15-Oct | 13 | Fetty Wap & Remy Boyz | 679 |
| 9-Aug | 16 | Green Day | 21 Guns |
| 7-Jul | 19 | Gwen Stefani | 4 In The Morning |
| 7-Oct | 9 | James Blunt | 1973 |
| 10-Oct | 34 | Jay Sean & Nicki Minaj | 2012 (It Ain't The End) |
| 4-Jun | 33 | Jay-Z | 99 Problems |
| 16-Apr | 35 | Kevin Gates | 2 Phones |
| 1-Jan | 20 | Kevon Edmonds | 24/7 |
| 7-Dec | 29 | Kylie Minogue | 2 Hearts |
| 3-Jun | 40 | Lil' Mo & Fabolous | 4 Ever |
| 11-Jan | 32 | Lil' Wayne & Corey Gunz | 6 Foot 7 Foot |
| 17-Oct | 11 | Logic, Alessia Cara & Khalid | 1-800-273-8255 |
| 3-Jan | 34 | Mark Wills | 19 Somethin' |
| 12-Apr | 49 | Mike Candys, Evelyn & Patrick Miller | 2012 (If The World Would End) |
| 13-Nov | 31 | Mike WiLL Made-It | 23 |
| 8-Jul | 28 | Miley Cyrus | 7 Things |
| 5-Jan | 32 | Ryan Cabrera | TRUE |
| 13-May | 24 | Taylor Swift | 22 |
| 8-Jul | 43 | The White Stripes | 7 Nation Army |
| 14-Aug | 49 | Tinashe & Schoolboy Q | 2 On |
| 11-Dec | 19 | T-Pain & Wiz Khalifa & Lily Allen | 5 O'Clock |
| 1-Dec | 26 | Wyclef Jean & Mary J Blige | 911 |

Missing Values:

For the future machine learning projects, the corrupted part of data e.g. missing values or inaccurate datapoint need to be preprocessed in order to be in the ideal format for producing the best performing model with acceptable accuracy.

Some missing values were observed with the following statistics. For handling the missing values, the following solution can be mentioned. Removal, replacement, or imputing are broadly used. The null value for the ranking on each song in a country means that the song did not make to the top songs in each month in a given country. For the purpose of future machine learning, they need to be accurately dealt with.

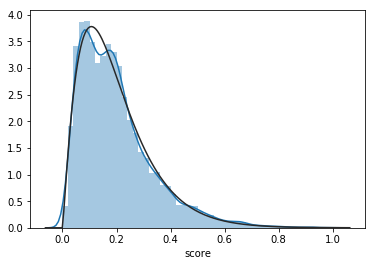
TABLE 8 – Missing Value

|  |  |  |
| --- | --- | --- |
| **Features** | **No. of Value** | **No. of Null Value** |
| Total Number of rows in dataset | 11700 | 0 |
| us | 10813 | 887 |
| uk | 6667 | 5033 |
| de | 6149 | 5551 |
| fr | 5062 | 6638 |
| ca | 5708 | 5992 |
| au | 5771 | 5929 |

Normalization:

Score variable seems to fit the Gamma distribution Ranging from 343.3 to 5524.5 with the mean of 1366.7 and standard deviation of 710.3.

FIGURE 2- Normalization of the score values



Normalization was performed on the Score amount to observe this feature by scaling each to a given range between its Min and Max amount.

This transformation normalized the feature individually such that it is in the given range e.g. between zero and one to make it ready for the training purposes for the future Machine Learning.