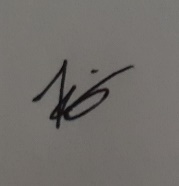
**Cover Page**

**Name:** Sena Nathaniel Luis Villaflor

**Matriculation Number:** 2101297F

**Signature: **

**Introduction**

The dataset that I have chosen is called top10s.csv and it is a dataset that shows the top songs per year between 2010 and 2019. This dataset was chosen with the intention of exploring, understanding, and finding out the popularity of songs based on their musical characteristics. With this knowledge, I will have a better understanding of how and why songs become popular and what role do such musical characteristics and factors play into contributing to their popularity

**Pre-processing of Data (Data cleaning and Data Transformation)**

**Step 1:** Import relevant modules such as pandas, matplotlib, missingno, numpy and sklearn. Then, read the dataset into a variable named df and print the first 5 rows.

Graphical user interface

Description automatically generated with medium confidence

**Step 2:** Print the shape and describe the dataset, to help better explore the dataset

**Graphical user interface, text, table

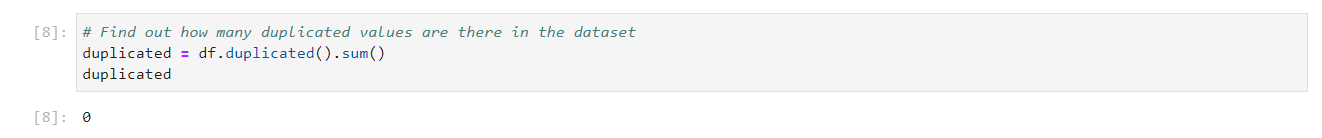
Description automatically generated with medium confidence**

**Step 3:** Replace all empty values with np.nan and drop all nan values just in case there are any in the dataset. However, in this dataset, there happens to not be any missing values. Code is still carried out just in case

**Graphical user interface, application

Description automatically generated**

**Step 4:** Find out if there are any duplicated values in the dataset. In this case, there are no duplicated values

****

**Step 5:** Find out what columns there are in the dataset. Notice how there is a column named “Unnamed”. This is supposed to be the index column for the dataset, but its name does not make sense and the column is irrelevant. Hence the column is dropped, and the index of the dataset is reset.

**A picture containing graphical user interface

Description automatically generated**

**Step 6:** After looking at the dataset, I noticed there are some outliers in the ‘pop’,’db’,’bpm’,’nrgy’ and ’dnce’ column. Hence, I identified the upper and lower limits of each of the columns and dropped all rows that are above the upper limit and below the lower limit. This is to remove any outliers. A distribution plot is plotted to check the distribution of values and check if there are still outliers for each column.

**Chart, histogram

Description automatically generated**

**Step 7:** Now, on to data transformation. I noticed that there were too many genres in the “top genre” column which could be generalised and categorised into main genres. Hence, I used a for loop to replace the genres into general categorised values like pop, hip-hop, latin, room, house, and others

Graphical user interface, application

Description automatically generated

**Step 8:** Next, I one hot encode the top genre column and then print out the datatypes to check whether the column has been one-hot encoded properlyGraphical user interface, text, application, email

Description automatically generated

**Step 9:** To make the data in the “dur” column easier to read, divide the duration column by 60 and round it to 2 decimal places to get the duration in minutes and in a rounded value.

A picture containing application

Description automatically generated Graphical user interface, text, application, chat or text message

Description automatically generated

**Step 10:** Rename the columns for better clarity and check if they have been renamed properly. ‘year’ has been changed to ‘release year’ to better specify the column. ‘nrgy’, ‘dnce’, ‘dB’, ‘live’, ‘val’, ‘dur’, ‘acous’, ‘spch’ and ‘pop’ has been renamed to their full name form to prevent confusion.

Graphical user interface, text, application, email

Description automatically generated

**Step 11:** Bin the values for bpm, energy, danceability, decibels, valence, duration, acoustic, speechiness, liveness and popularity to make it easier to read data. Plot a bar chart to better visualise the newly binned data for each columnGraphical user interface, text, application

Description automatically generated

**Step 12:** Check if the values have been binned properly

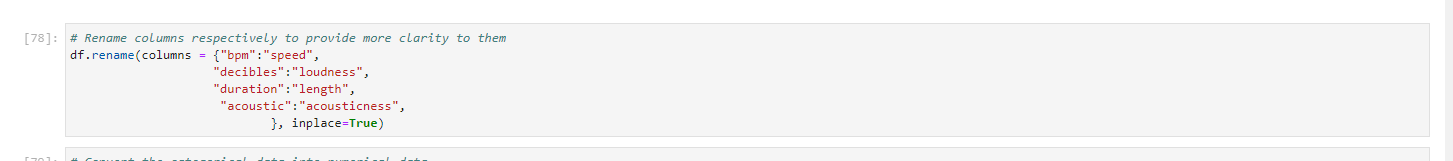
A picture containing shape

Description automatically generated

**Step 13:** Ordinally encode the categorical data in the dataset to allow for better clarity and readability

Background pattern

Description automatically generated

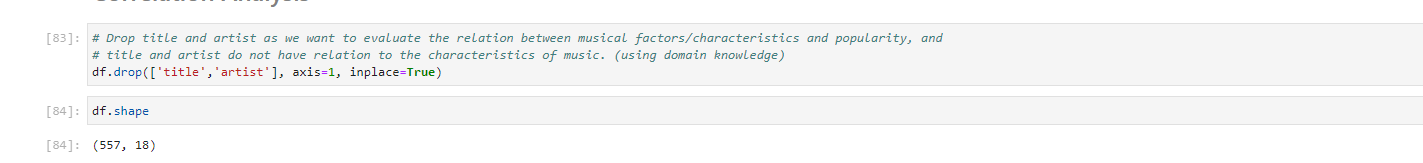
**Step 14:** Rename the columns again to provide further clarity****

**Step 15:** After ordinally encoding the categorical data in our dataset, I noticed that the data type is still categorical, hence I ran code that helped to convert the categorical data to numerical data without changing the values**Graphical user interface, application

Description automatically generated with medium confidence**

**Correlation and Feature Selection**

**Step 16:** Now, I move on toe correlation analysis. I dropped title and artist as I want to find the correlation between popularity and musical characteristics and factors. Title and artist do not have much correlation to popularity as it is not a musical characteristic, using my domain knowledge

****

**Step 17:** Find the correlation in our dataset and plot a correlation matrix

**Chart, treemap chart

Description automatically generated**

**Step 18:** Print out the columns that have correlation to popularity. These will be the relevant features

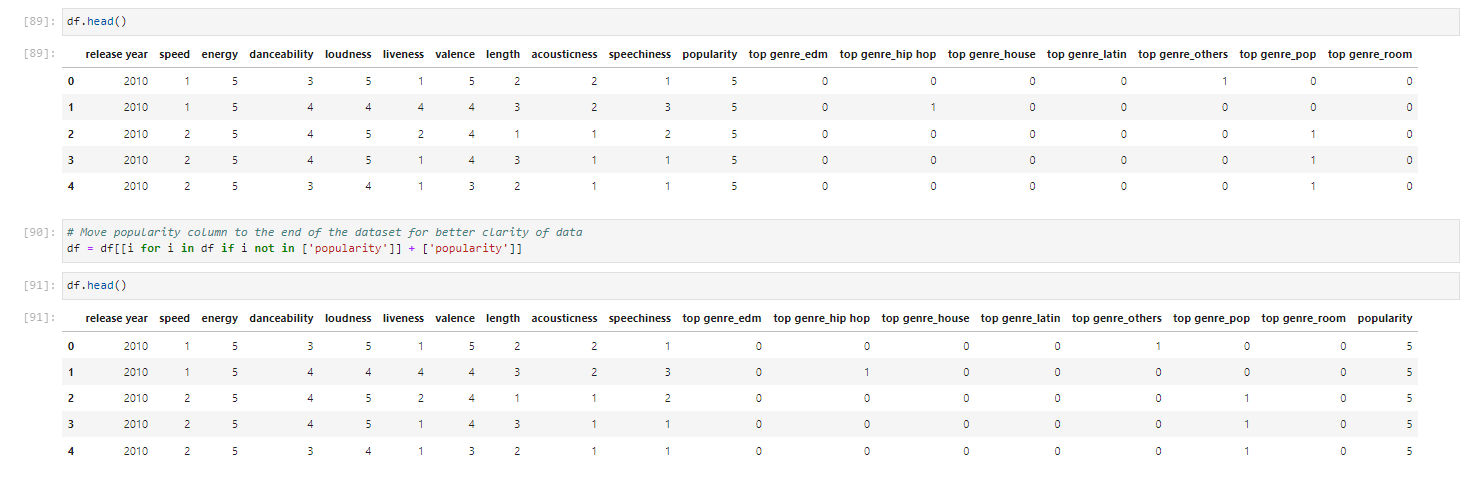
**Graphical user interface, text, application

Description automatically generated**

**Step 19:** Plot a heatmap for further visualisation of correlation

**Graphical user interface

Description automatically generated**

**Step 20:** Move the popularity column to the end of the dataset to make the dataset easier to read****

**Data Visualisation**

**Step 21:** Now, I move on to data visualisation. Firstly, I find the top 5 correlated columns in our dataset. It seems that loudness and energy, valence and danceability, energy and valence, valence and loudness and top genre\_hip hop and length have the most correlation in our dataset**Graphical user interface, text, application, email

Description automatically generated**

**Step 22:** Graphical user interface, application, Word

Description automatically generated

**Step 23:**

**Graphical user interface, application

Description automatically generated with medium confidence**

**Step 24:**

**Graphical user interface, text, application

Description automatically generated**

**Step 25:**

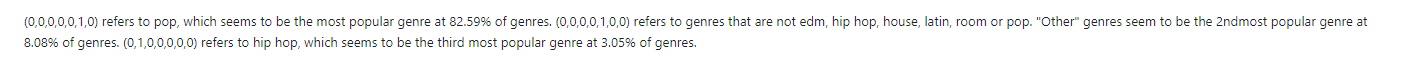
**Graphical user interface, application

Description automatically generated**

**Step 26:**

**Chart, pie chart

Description automatically generated**

****

**Methods and Improvements (and Modelling)**

**Step 27:** Now, I will move on to data modelling. Firstly, I will import all relevant modules. Next, I will shuffle and split the dataset to predict popularity, splitting the dataset at a 80:20 ratio and setting a random state of 8.

**Graphical user interface, text, application

Description automatically generated**

**Step 28:** Using default hyperparameters for the different algorithms, I then train the models, evaluate them, and find which model is the most accurate with default hyperparameters. From this, I concluded that LDA had the most accuracy using default hyperparameters.

****

**Step 29:** Next, I used grid search to determine the best and most efficient hyperparameters for the LR, LDA, KNN, CART and SVM model, and then created a new model for each one using those newly found hyperparameters.

**Graphical user interface, text, application

Description automatically generated**

****

**Text

Description automatically generated with medium confidence**

**A picture containing scatter chart

Description automatically generated**

**Application

Description automatically generated with low confidence**

**Results and Analysis**

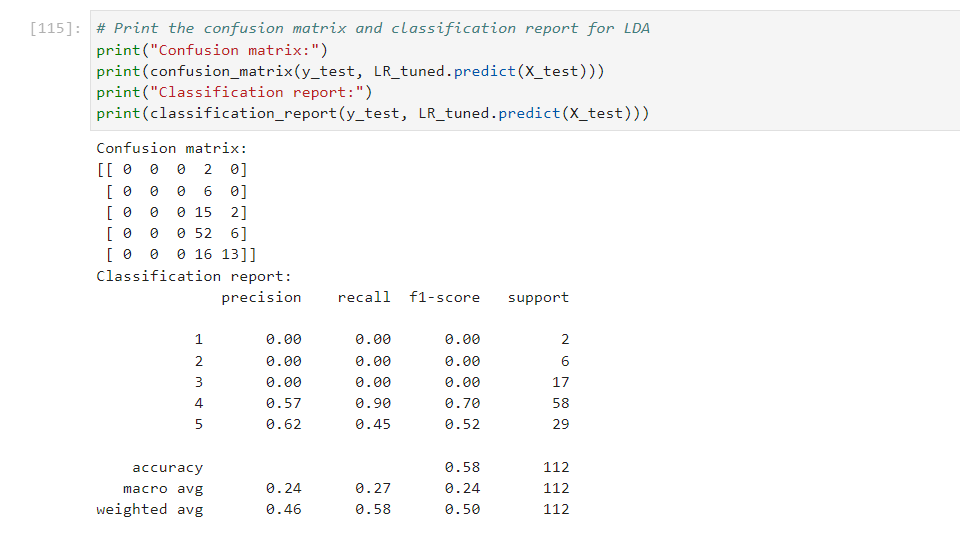
**Step 30:** Next, I trained the newly tuned models and found which one had the most accuracy score. LDA and LR were tied with the highest accuracy score. After evaluating, I concluded that LR is the more appropriate model to use as logistic regression produces robust estimations regardless of assumptions as logistic regression does not make any assumptions about distributions of variables or relationships, unlike linear discriminant analysis, where if the assumptions of LDA are violated, the model would not make as robust estimations. Furthermore, the requirements of multivariate normalities would disrupt LDA’s classifications more than LR, hence I chose to use LR over LDA.

Graphical user interface, text, application

Description automatically generated

**Graphical user interface, text, application

Description automatically generated**

**Step 31:** Print the confusion matrix and classification report for the tuned logistic regression model****

**Step 31:** Use flask to implement model into a website

Graphical user interface, text, application, email

Description automatically generated

Text

Description automatically generated

A picture containing graphical user interface

Description automatically generated

**Conclusion**

To conclude, this project has given me a better insight into what roles do factors and characteristics of songs play in determining the popularity of a song. I have learnt many valuable lessons when it comes to pre-processing and processing a dataset and in the future I will be able to better understand and work with other datasets.

**References**

Antonogeorgos, G., Panagiotakos, D. B., Priftis, K. N., & Tzonou, A. (2009). *Logistic regression and linear discriminant analyses in evaluating factors associated with asthma prevalence among 10- to 12-years-old children: Divergence and similarity of the two statistical methods*. International journal of pediatrics. Retrieved July 27, 2022, from https://www.ncbi.nlm.nih.gov/pmc/articles/PMC2798100/#:~:text=Thus%2C%20linear%20discriminant%20analysis%20and,logit%20of%20the%20logistic%20regression.