**SUMMARY AND OBJECTIVE**

With this project my aim is to improve my data analysis and data science skills and to create a public repository that I can use to showcase my capabilities.

For this first project I’ve chosen a topic that I’m not familiar with and that I hope it will give me the opportunity to try different data analysis tools that I haven’t been able to try on my date to date job yet. As a little bit of background, I am a manufacturing engineer with extensive experience in the aerospace sector, specifically in manufacturing.

Coming back to the project, I will split this week into different sections.

**DATA COLLECTION**

I have segregated the data collection part into a data\_creation module for tidiness purposes.

I check whether the data has already been collected (which would be stored in parquet file on the current working directory) and if not I collect the data from the following webpage using pandas:

<http://www.umdmusic.com/default.asp?Lang=English&Chart=D&ChDay=1&ChMonth=2&ChYear=1999&ChBand=&ChSong>=

I decided to collect data using parquet files from 1970s to 2020s to be able to see if there is any change in the behaviour of the data across time. I chose to scrape the data 4 times for every month in that year range to create enough granularity of the data. Because the charts are on a weekly basis, I need to collect the data weekly. I also want to create a big enough dataset to apply descriptive and machine learning analysis.

**DATA PROCESSING**

My initial objective is to find out the characteristics that differentiate the number one songs from the rest of the songs. For that I calculate the following features for each song:

* Days in charts
* Days taken to climb from bottom to top
* Days in top position
* Quantity of positions climbed from bottom to top
* Quantity of positions climbed from start to top
* Starting position

I do the work creating a summary table with one row per each song and artist. Then I add the results of the features to each of the songs in the table.

I calculate the next fields for each song and title to get the previous features:

* The maximum date in the charts
* The top position
* The date of the top position
* The bottom position prior to reaching the top position

Once the attributes are calculated, I clean the data through the following steps:

1. I select the songs where the OverallPeakPos (which came with the original data) and the calculated peak position match. I exclude songs that peaked outside of the 1970 to 2010 time range because they would not have stats reflecting the true nature of the song.
2. I select the songs where the Entry in Charts or LastDate in charts reflect that the song’s run in the charts didn´t start before or didn´t potentially finish after the time analysis range.
3. I select the songs where the original length of days in charts is at least equal or bigger than the calculated time in charts. This is because if a song has had multiple re-entries across the years in the charts, the calculated time in charts can throw silly numbers, where a song originally entered the charts in 1973 and the maximum date of the last time it entered the charts is 2016.

I realized that there is a mismatch between the Overall Total Weeks (the time that the song has been in charts according to the original data) and the calculated time in charts (by subtracting the last parsing date of that song with the entry date of the song).

For example, Mariah Carey’s song “All I want for Christmas is you” entered the charts in 2018-12-01 and exited the charts in 2019-01-01, spanning a total of 31 days. However, the original column specifies a total of 19 weeks in charts, which I think includes for future appearances in following Christmas periods and would not be suitable for the current analysis.

**DATA ANALYSIS**

Within the data analysis section, I dive a little bit deeper into the dataset. From the previous work I have been able to create a set of features to describe each of the songs.

In this module I create a summary table for each of the decades where I get the means of the features. I can see that in some of the features such as Days In Charts have a lot of variation in different decades.

Then I check whether any of the features is normally distributed. I do this for each feature across all the dataset and then across each decade individually. And none of the features is normal for any of the decades.

After checking the normality (or lack of thereof), I also want to assess if the data is equivalent when we compare a decade with another one.

I want to know for example if the features ‘Days of Song In Charts’ is equivalent for 1970 and 2010. At a later stage I want to research if a model could get better results by using a more specific set of data or whether the equivalence between sets of data is irrelevant.

In terms of equivalence there are also very little positive results.

I repeat the analysis for the n1 songs specifically and the results keep on showing little equivalence across decades.

**PREDICTION**

What type of predictions do we want to carry out?

With the entry data of a song in the charts. We want to answer the following questions:

* The top position that it’ll reach
* The time it’ll take to reach that position
* The time it’ll stay at that top position
* The time it’ll stay in the charts

What are other attributes that can determine if a song will be a n1?

* Words in lyrics
* Music genre
* Topics of music
* Song’s composition: Song’s sheet music
* Song’s recording studio
* Song’s producers

Data sources for a song’s production values:

* Music industry websites: AllMusic, Billboard, Rolling Stone, Pitchfork, NME, The fader, Stereogum, Mixmag
* Music streaming and download services: Spotipy, Apple music API, AWS API, Google Play
* Online databases of music information, such as AllMusic, Discogs, MusicBrainz
* Music industry publications and magazines, such as Rolling Stone and Billboard

Potential predictive analytics to try:

* Random Forest
* XGBoost
* K-means clustering