**Introduction**

Our intent was to examine the characteristics of the music of artist Roger Prince Nelson, also known as Prince. We considered all albums that Prince produced during the eighties. The reason for doing so was that the eighties was when most of Prince’s iconic albums were released. In other words, there was a higher probability of understanding which song would be a hit and which song would not be a hit, simply because of the number of songs that were a hit during this period. We wanted to understand if there were any defining patterns and critical features in his songs that could be used to determine if the song would be a hit or not. Apart from this, we also wanted to understand if there was a relation between the sentiment of his songs and the song being a hit or not. Lastly, we wanted to understand if any of the metrics that we considered could be used to predict if a song produced by him would be a hit or not.

**Data Description**

We did not have a dataset to begin with, and had to gather all of the data from scratch. What follows in the next paragraph is a description of the different variables we considered for this analysis, along with the method that was used to extract this data. We will also mention the data cleaning methods that we used to ensure we had a dataset.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Variable Name** | **Variable Description** | **Method of Extraction** | **Cleaning Operations Performed** | **Source** |
| Song name | The name of the song | We googled for each album name, and went to the Wikipedia entry for the album. After this was done, we extracted the song name, the position on the album and the total song length using import.io . | The Wikipedia entries had a lot of special characters that were also imported and had to be removed. This was done using Microsoft Excel's find and replace function. | Wikipedia |
| Album Name | The name of the album in which each song is present | We googled for each album name, and went to the Wikipedia entry for the album. After this was done, we extracted the song name, the position on the album and the total song length using import.io . | The Wikipedia entries had a lot of special characters that were also imported and had to be removed. This was done using Microsoft Excel's find and replace function. | Wikipedia |
| Position on Album | The position of the song in the Album | We googled for each album name, and went to the Wikipedia entry for the album. After this was done, we extracted the song name, the position on the album and the total song length using import.io . | The Wikipedia entries had a lot of special characters that were also imported and had to be removed. This was done using Microsoft Excel's find and replace function. | Wikipedia |
| In Hot 100 Billboard | If the songs from the hot 100 list were in our dataset, we gave it a value of yes, else a value of no. | The language that was used to extract these values was Python. The packages that were used were urllib2 and lxml. The name of the song and the position were extracted. | There was no cleaning required as Urllib2 and lxml can be used to extract the exact pieces of information that we need. | http://www.billboard.com/artist/351039/prince/chart |
| Position on top 100 billboard | The highest possible position in the hot 100 billboard list. If not, give it a value of 0. | The songs were extracted from http://www.billboard.com/artist/351039/prince/chart . The language that was used to extract these values was Python. The packages that were used were urllib2 and lxml. The name of the song and the position were extracted. | There was no cleaning required as Urllib2 and lxml can be used to extract the exact pieces of information that we need. | http://www.billboard.com/artist/351039/prince/chart |
| Year | The year in which the song was released. | The year in which the album was extracted manually. There was no need to automate this process as the number of albums we had were limited. | No cleaning operations were necessary. | Wikipedia |
| Song length in seconds | The song length in seconds. | The song length was extracted from Wikipedia. After this was done, we converted the total time from Min:Sec format to minutes by performing a basic conversion in Microsoft Excel. | No cleaning operations were necessary. | Wikipedia |
| BPM Value | The BPM ( Beats per minute ) value of the song. | The BPM values were extracted using multiple methods, because all of them were not available at one place. Some were extracted using import.io, while others were extracted manually because each song's BPM value had to be searched for. | For the values extracted using import.io, manual cleaning was performed in Microsoft Excel to remove special characters. | https://www.cs.ubc.ca/~davet/music/track/PRINCE\_\_\_PRN/, https://jog.fm/workout-songs/by/prince?order=desc&sort=bpm, |
| Key | The key of the song. | The key of each song was not available at a single location, so a combination of import.io and manual extraction was used. | The values obtained through import.io were manually cleaned in Microsoft Excel for removing special characters. | https://www.cs.ubc.ca/~davet/music/track/PRINCE\_\_\_PRN/, http://prince.org/msg/7/124627, https://beatportcharts.com/track/1701012/ |
| Lyrics | The lyrics for each song. | The lyrics were extracted using import.io from princelyrics.co.uk. The song lyrics were not in a format which allowed them to be processed. | Stopwords like "I", "me" etc were removed. Special characters like ".",",","!" were also removed using basic find and replace operations in excel. | [www.princelyrics.co.uk](http://www.princelyrics.co.uk/) |
| Word Count | The word count for each song. | The word count was extracted from the lyrics. First, the excel transpose function was used to put all of the song lines into a single cell. Then the formula =IF(LEN(TRIM(A2))=0,0,LEN(TRIM(A2))-LEN(SUBSTITUTE(A2," ",""))+1) was used to get the word count. | Stop words and special characters had been removed. | [www.princelyrics.co.uk](http://www.princelyrics.co.uk/) |
| Sentiment | The overall sentiment of the song. The process used to get the value for this will be explained below. | Explained below. | Explained below. | [www.princelyrics.co.uk](http://www.princelyrics.co.uk/) |
| Category | Based on the overall sentiment value, the song is classified into one of four categories. | The basis for the category value was the sentiment score that was obtained, which means that the data extraction had already been done. | No cleaning operations were necessary. | [www.princelyrics.co.uk](http://www.princelyrics.co.uk/) |

Import.io description – Import.io is a standalone desktop application through which it is possible to extract data from websites without any code necessary. It is also possible to save the functionality as an API so the same operations can be performed on multiple web pages. This is done by teaching the application to extract data by highlighting examples by which the software understands and generalizes from these examples to extract data from the web pages.

Sentiment analysis description – for the sentiment analysis part, we saved the lyrics for each song in a separate csv file. This was after each song was cleaned by removing all the stop words and special characters. After this was done, we used R to load this file into the workspace. We then loaded a corpus of positive and negative keywords into the database and compared the lyrics against the corpus. For every positive word that the lyrics had, we increased the score by +1. We did the same with the negative words corpus by giving a score of -1. The total score was calculated by summing the positive and negative score. After this was done, we got the net sentiment score by dividing the sentiment score with the total number of words.