|  |  |
| --- | --- |
| **Group 8**  **Saurabh Donthin**  **Jamie Wheeler** | **STAT 515-001**  **Final Project Report**  **12 December 2016** |

**Computer Blue**

**An Analysis of Prince’s 80’s Music**

**1. Problem description**

Our project investigated the music artist, Rogers 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.

The ability to classify songs can promote subscriber retention or value to users for streaming and music recommendation systems. The ability to identify factors of a hit is valuable for artist and record label song promotion and identifying potentially new and successful artists. Further, the ability to quantify things such as sentiment and album structure may offer new art or cultural study of an artist, song, album, and/or genre. Typically, these studies have been done almost solely by opinion, case study, and relative comparison without much quantification.

**2. Data Description**

We did not have a dataset to begin with, and had to gather all data from primary or solid secondary sources. What follows next is a description of the different variables we considered for this analysis, the method that was used to extract this data, and the data cleaning methods that we used to create the dataset.

|  |  |
| --- | --- |
| **Variable Name:** Song name | **Source:** Wikipedia |
| **Variable Description:** Name of the song | |
| **Method of Extraction**: 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 . | |
| **Cleaning Operations Performed:** 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. | |

|  |  |
| --- | --- |
| **Variable Name:** Album Name | **Source:** Wikipedia |
| **Variable Description:** The name of the album in which each song is present | |
| **Method of Extraction**: 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 . | |
| **Cleaning Operations Performed:** 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. | |

|  |  |
| --- | --- |
| **Variable Name:** Position on Album | **Source:** Wikipedia |
| **Variable Description:** The position of the song in the Album | |
| **Method of Extraction**: 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 . | |
| **Cleaning Operations Performed:** 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. | |

|  |  |
| --- | --- |
| **Variable Name:** In Hot 100 Billboard | **Source:** <http://www.billboard.com/artist/351039/prince/chart> |
| **Variable Description:** If the songs from the hot 100 list were in our dataset, we gave it a value of yes, else a value of no. | |
| **Method of Extraction**: 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. | |
| **Cleaning Operations Performed:** There was no cleaning required as Urllib2 and lxml can be used to extract the exact pieces of information that we need. | |

|  |  |
| --- | --- |
| **Variable Name:** Position on top 100 billboard | **Source:**<http://www.billboard.com/artist/351039/prince/chart> |
| **Variable Description:** The highest possible position in the hot 100 billboard list. If not, give it a value of 0. | |
| **Method of Extraction**: 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. | |
| **Cleaning Operations Performed:** There was no cleaning required as Urllib2 and lxml can be used to extract the exact pieces of information that we need. | |

|  |  |
| --- | --- |
| **Variable Name:** Year | **Source:** Wikipedia |
| **Variable Description:** The year in which the song was released. | |
| **Method of Extraction**: 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. | |
| **Cleaning Operations Performed:** No cleaning operations were necessary. | |

|  |  |
| --- | --- |
| **Variable Name:** Song length in seconds | **Source:** Wikipedia |
| **Variable Description:** The song length in seconds | |
| **Method of Extraction**: The song length was extracted from Wikipedia. After this was done, we converted the total time from Min:Sec format to seconds by performing a basic conversion in Microsoft Excel. | |
| **Cleaning Operations Performed:** No cleaning operations were necessary. | |

|  |  |
| --- | --- |
| **Variable Name:** BPM Value | **Source:** https://www.cs.ubc.ca/~davet/music/track/PRINCE\_\_\_PRN/, https://jog.fm/workout-songs/by/prince?order=desc&sort=bpm, |
| **Variable Description:** The BPM ( Beats per minute ) value of the song. | |
| **Method of Extraction**: 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. | |
| **Cleaning Operations Performed:** For the values extracted using import.io, manual cleaning was performed in Microsoft Excel to remove special characters. | |

|  |  |
| --- | --- |
| **Variable Name:** Key | **Source:** https://www.cs.ubc.ca/~davet/music/track/PRINCE\_\_\_PRN/, http://prince.org/msg/7/124627, https://beatportcharts.com/track/1701012/ |
| **Variable Description:** The key of the song. | |
| **Method of Extraction**: The key of each song was not available at a single location, so a combination of import.io and manual extraction was used. | |
| **Cleaning Operations Performed:** The values obtained through import.io were manually cleaned in Microsoft Excel for removing special characters. | |

|  |  |
| --- | --- |
| **Variable Name:** Lyrics | **Source:**  [www.princelyrics.co.uk](http://www.princelyrics.co.uk/) |
| **Variable Description:** The lyrics for each song. | |
| **Method of Extraction**: 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. | |
| **Cleaning Operations Performed:** Stopwords like "I", "me" etc were removed. Special characters like ".",",","!" were also removed using basic find and replace operations in excel. | |

|  |  |
| --- | --- |
| **Variable Name:** Word Count | **Source:**  [www.princelyrics.co.uk](http://www.princelyrics.co.uk/) |
| **Variable Description:** The word count for each song. | |
| **Method of Extraction**: 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. | |
| **Cleaning Operations Performed:** Stop words and special characters had been removed. | |

|  |  |
| --- | --- |
| **Variable Name:** Sentiment | **Source:**  [www.princelyrics.co.uk](http://www.princelyrics.co.uk/) |
| **Variable Description:** The overall sentiment of the song. The process used to get the value for this will be explained below. | |
| **Method of Extraction**: Explained below. | |
| **Cleaning Operations Performed:** Explained below. | |

|  |  |
| --- | --- |
| **Variable Name:** Category | **Source:**  [www.princelyrics.co.uk](http://www.princelyrics.co.uk/) |
| **Variable Description:** Based on the overall sentiment value, the song is classified into one of four categories | |
| **Method of Extraction**: The basis for the category value was the sentiment score that was obtained, which means that the data extraction had already been done. | |
| **Cleaning Operations Performed:** No cleaning operations were necessary. | |

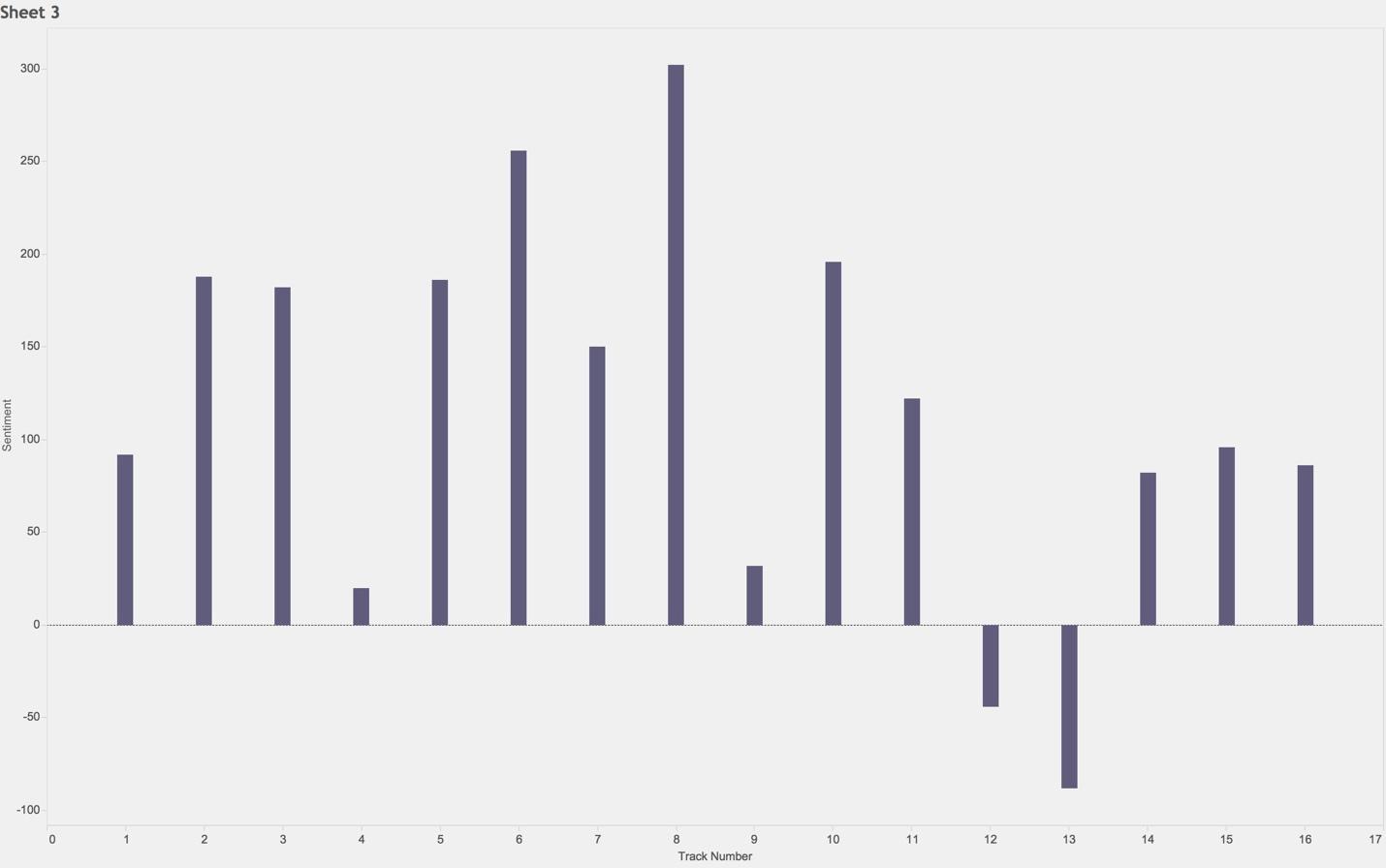
Import.io description – Import.io is a standalone desktop application through which it is possible to extract data from websites without any code necessary. It is also possible to save the functionality as an API so the same operations can be performed on multiple web pages. This is done by teaching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

**3. Sentiment Analysis**

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 meaning that sentiment can only be between -100 and +100.

**4. Visualization**

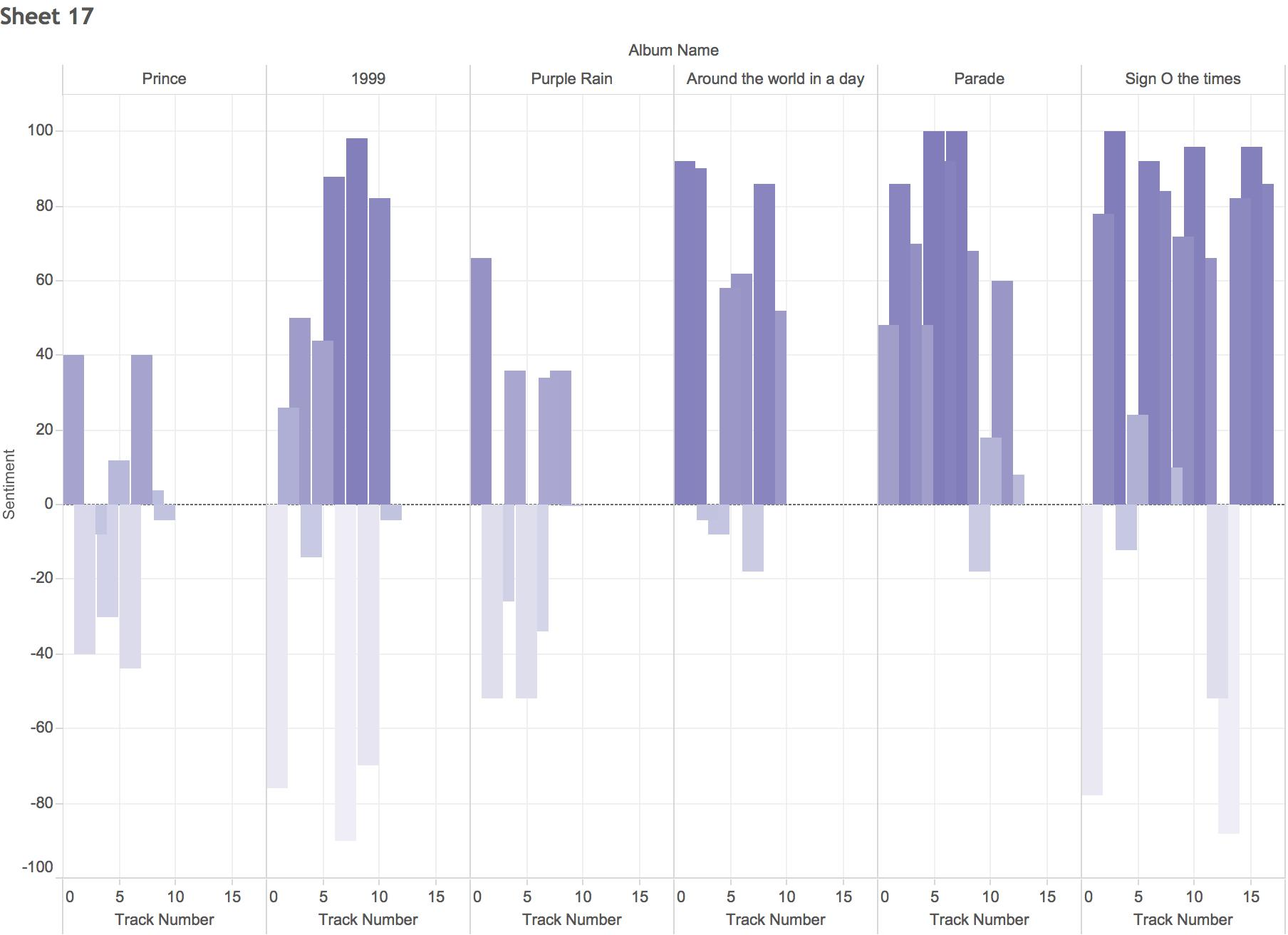
When the sentiment analysis was performed, visualizations were applied to analyze potential patterns related to the sentiment.

Song Sentiment by Track

There appeared to be a general trend toward increasing sentiment that dropped off after track 8. However, the higher track numbers are a bit misleading because there are only three albums have more than nine tracks and only one has more than twelve tracks.

It should be noted at this point that tracks 4 and 9 are the closest to neutral sentiment.

A visualization was then created to analyze on an individual album basis to identify trends. That visualization is on the following page.

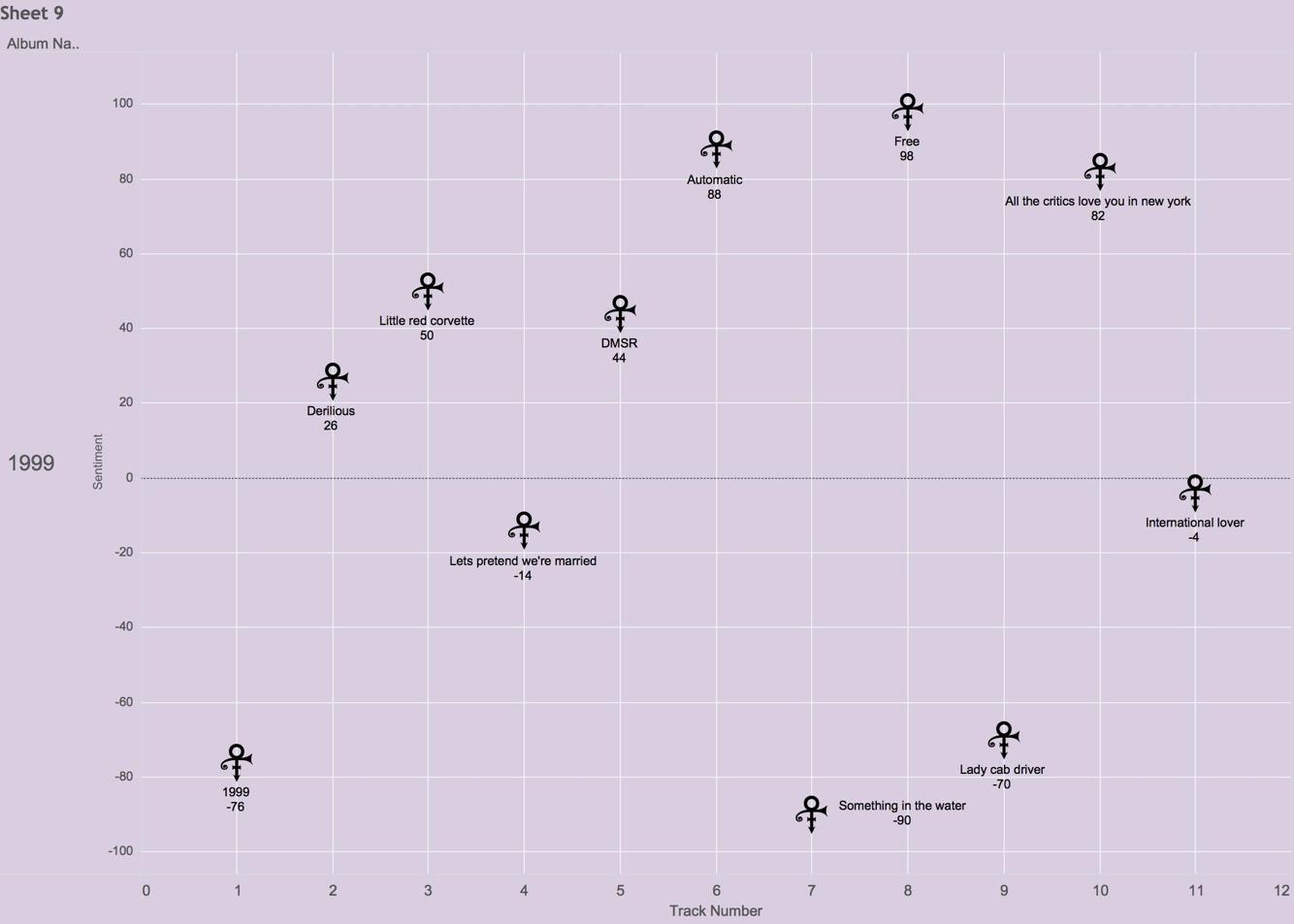


Song Sentiment by Album and Track

When sentiment rating is separated by album, we see a slight trend toward increasing sentiment as the album continues with his lower sentiment songs clearly being more skewed toward the beginning of the album.

Additionally, we see a pattern of overall album sentiment appearing with there being two eras; an early neutral sentiment era and a later positive sentiment era.

A weakness of this graph is the inability to easily identify neutral sentiment. The following visualizations were done to identify any neutral sentiment songs on albums.



We can see from the above three visualizations that there are in fact some songs that are nearly neutral sentiment. The one that stands out the most, though, is *Purple Rain.* *Purple Rain* has a sentiment rating of 0. This was unexpected due to the results of the sentiment analysis. While we rounded most sentiments to the nearest integer, they were at least four significant figures. *Purple Rain* is not rounded. It is actual zero. A further analysis of this song was done to explore the curiosity.

|  |  |
| --- | --- |
| I never meant to cause you any **sorrow** I never meant to cause you any **pain** I only wanted to one time to see you **laughing** I only wanted to see you **Laughing** in the purple rain  *[Chorus]*  I never wanted to be your weekend **lover** I only wanted to be some kind of **friend** Baby, I could never **steal** you from another It's such a **shame** our friendship had to end | **Positive Sentiment Word**  **Negative Sentiment Word**  -sorrow + -pain + laughing + Laughing + lover + friend + -steal + -shame =  -1-1+1+1+1+1-1-1 = 0 |

Not only is *Purple Rain* sentimentally neutral, it is sentimentally symmetrical. It is unlikely this perfect symmetry was accidental. It may be that Prince structured intentionally. In fact, it seems likely that some aspects were intentionally structured.

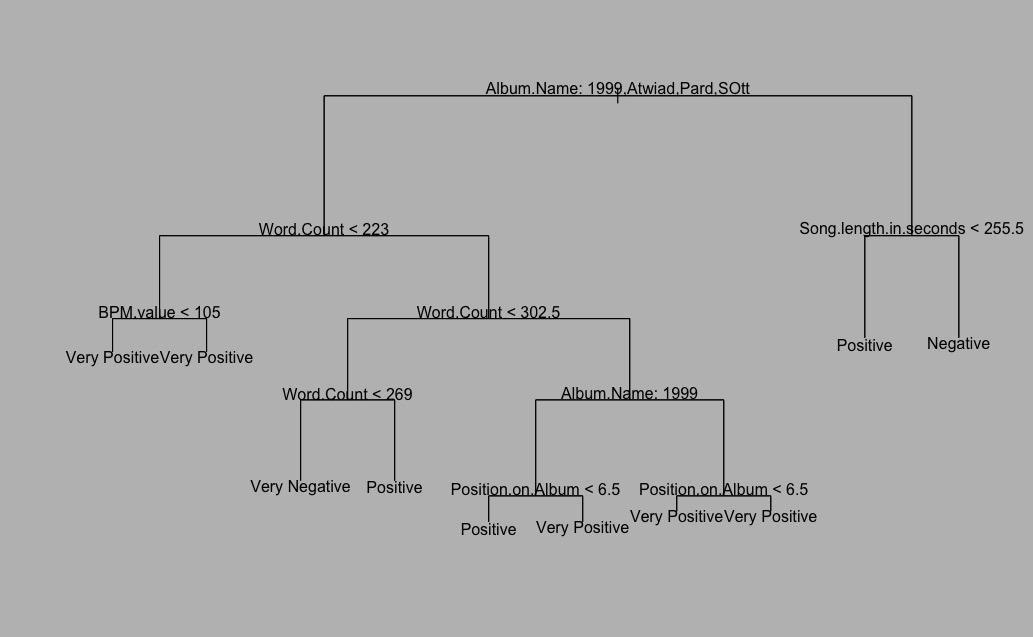
**5. Data Modeling**

We attempted to model the data using a series of methods including logistic regression, liner regression, classification, and decision tree. For logistic regression, we attempted to identify if it was a in the hot 100 or not. Linear regressions were performed for other album position, chart position, BPM, song length, and word count.

Every combination of dependent variable was attempted and very little that was statistically significant was found and even when it was statistically significant, it was of minor impact. The R code for this is provided in the appendix. This was also attempted with Taylor Swift songs from a couple of albums and more was found that was statistically significant, so we believe the models are fine. It is simply that there are too many exogenous variables and potentially randomness to make it feasible to model these factors for Prince’s song writing.

[INSERT CONFUSION MATRIX AND DISCUSSION HERE]

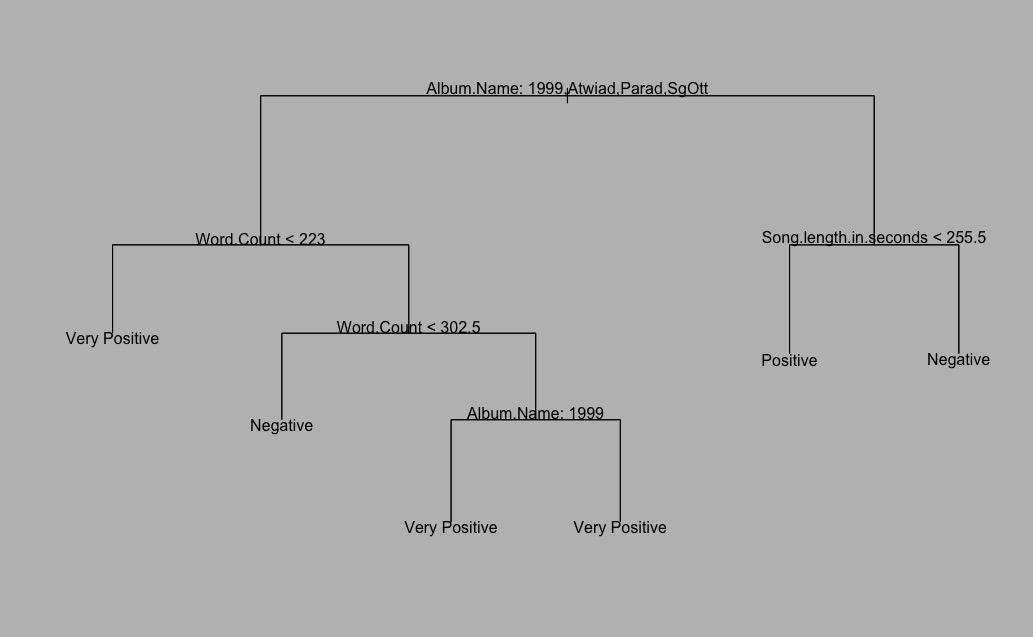
The decision tree gave us the best model of deciding if his songs would be a hit. Early models actually crashed R. We believe it is because his song names are almost as unique as the key the song is in making too many viable branches. Keys were removed and the buckets created which classified the sentiment as very negative (-100 to -51), negative (-50 to -1), neutral (0), positive (1 to 50), or very positive (51-100). The decision tree gave very clear results:



Unpruned Decision Tree

Decision trees appear to be good at identifying if a song was a hit how one could identify it, unfortunately, given the dependence upon album, it is not a very good predictive tool. It is reasonable that this would be the strongest predictor of past data since there are albums with clearly higher sentiment than others.

The pruned tree informs us of something interesting as well.



Pruned Decision Tree

The most interesting part of this tree is that albums other than 1999, Around the World in a Day, and Sign O the Times can be categorized almost strictly by song length in seconds indicating they may not have been as diverse. Also, it appears from this that there was a preference for shorter songs to be positive. It is unclear why the node “Album Name: 1999” has two branches that both terminate for very positive sentiment.

**6. Analysis**