Billboard 200 Data Analysis

CIS 3389 Programming for Data Processing

Alexis Benitez-Martinez, Jeffrey Medina, Naga Nithin Matta, Pedro Batdorf

Data Collection

- → Our data comes from Billboard's top 200 chart, which ranks the top 200 popular music albums and EP's in the United States
- → In order to gather our data, we web scrape their Billboard 200 page and save the information into a csv file
  - → https://www.billboard.com/charts/billboard-200
- → This page is continuously updated week after week with the newest music releases
- → This data could be utilized to track specific artist performance data

Research Questions

→ 1. What is the frequency of each artist on this roster? Plot the top 10 artists that have the most appearances

→ 2. What is the rank difference from the last week to the current week?

→ 3. Slice the first letter of each artist's name and compute the number of times each letter appears as the first character in a name. Visualize this data.

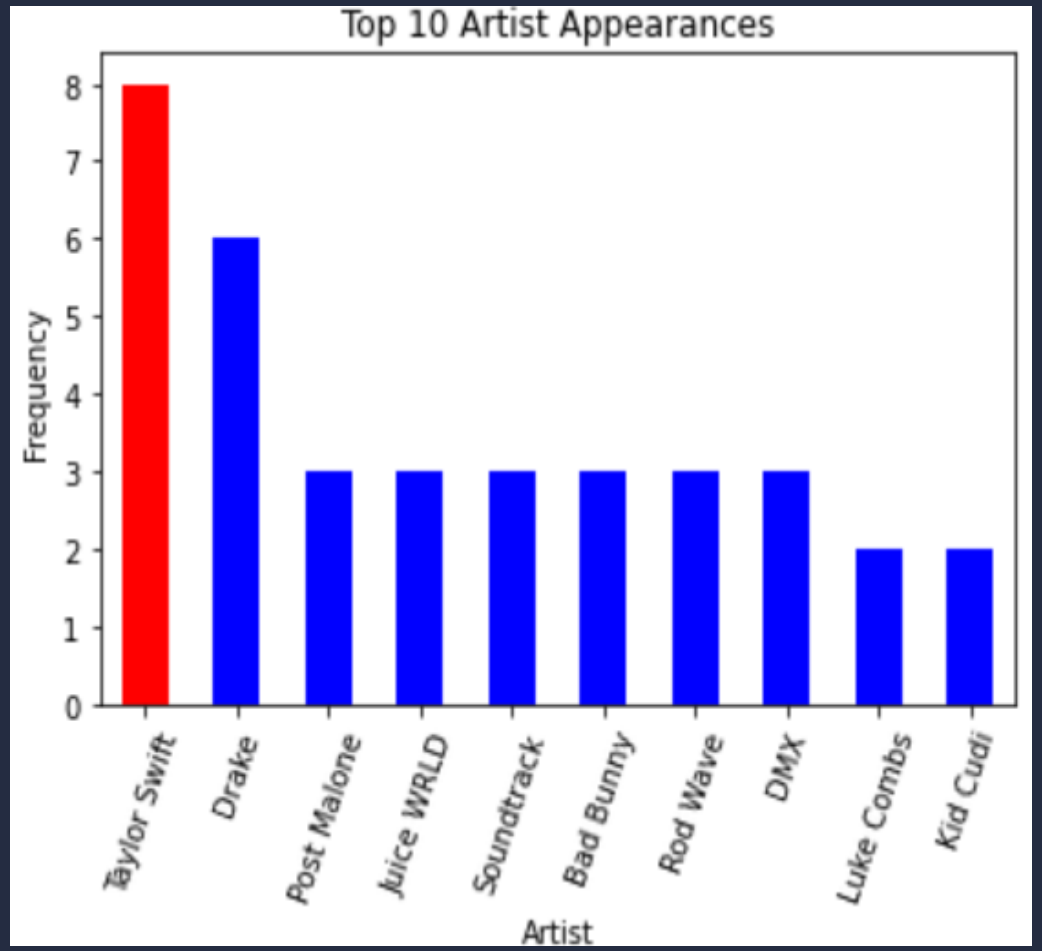
Data Summary

- The variables we will be looking at are:
  - **Artist**
  - **Title** (Album name)
  - **Rank**
  - **Last week’s rank**

Data Pre-processing

- We first need to import the necessary libraries to begin the web scraping process
- Utilize BeautifulSoup to scrape the content from the Billboard 200 web page
- Create lists for each piece of data we are interested in and using Pandas to create a data frame with those lists
  - For our last week rank list, we needed to create a data cleaning condition that would transform non-numerical values to 0
- Export the data frame to a csv file

Top 10 Artist Appearances



- What are the top 10 artists that have the most appearances in the Billboard 200 list?
- First, we are importing matplotlib.pylpot in order to utilize the plt function to create a bar chart
- We can then utilize our Artist data in our data frame in conjunction with plt and other parameters to produce the top 10 artists who appear the most and their frequency
- In a business setting, this can provide excellent artist performance tracking, especially since this data is updated on a weekly basis

```
import matplotlib.pyplot as plt

topalbum['Artist'].value_counts()[:10].plot.bar(rot=70,
#LISTS TOP 10 ARTISTS WHICH HAVE MORE APPERANCES IN THE
plt.title('Top 10 Artist Appearances') #TITLE

plt.xlabel('Artist') # LABEL

plt.ylabel('Frequency') #LABEL
```

Rank Changes Week After Week

- We are given each artist's current rank, as well their rank from the prior week. We want to know how much their position has changed from the prior week to the current.
- This can be accomplished through the subtraction of the 'Rank' and 'Last week Rank' lists.
  - We ensured that each of these lists are the same shape because if they aren't, this would not produce accurate results, if any at all.
- First, we create a new list that will contain these differences in rankings
- Utilize a subtraction function to append the output of the computations. This new list will be appended to the end of the data frame

```
In [25]: eng['rank_Diff'] = eng['Last week Rank'].sub(eng['Rank'], axis = 0)
         print("\nDifference of rank and lastweekrank :\n", eng)
```

Rank Changes Week After Week

	Artist	Title Name	Rank	Last week Rank
0	Taylor Swift	Fearless (Taylor's Version)	1	0
1	DMX	The Best Of DMX	2	73
2	Justin Bieber	Justice	3	1
3	Morgan Wallen	Dangerous: The Double Album	4	4
4	Rod Wave	SoulFly	5	3

Before appending the calculated difference

	Artist	Title Name	Rank	Last week Rank	rank_Diff
0	Taylor Swift	Fearless (Taylor's Version)	1	0	-1
1	DMX	The Best Of DMX	2	73	71
2	Justin Bieber	Justice	3	1	-2
3	Morgan Wallen	Dangerous: The Double Album	4	4	0
4	Rod Wave	SoulFly	5	3	-2

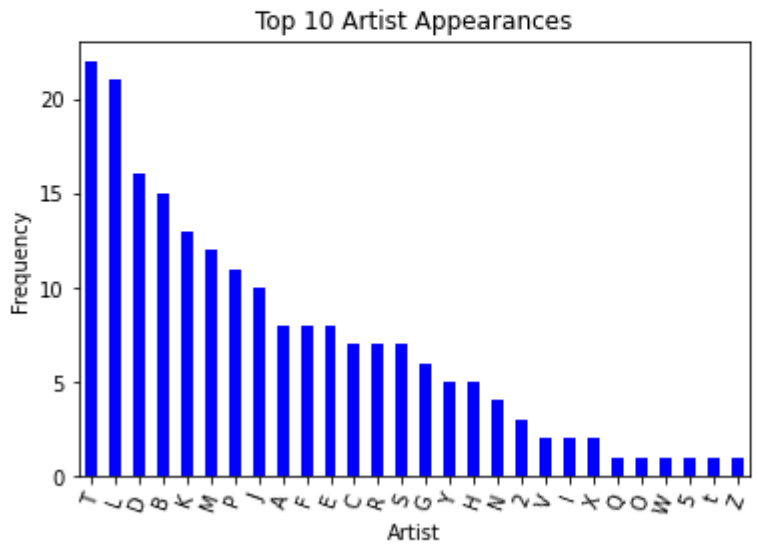
After appending the calculated difference

Frequency of Artists sorted by First Character

```
In [28]: eng['First_Letter_Analysis'] = eng['Artist'].astype(str).str[0]
         eng
```

```
eng['First_Letter_Analysis'].value_counts().plot.bar(rot=70, color =['b'])
plt.title('Top 10 Artist Appearances')
plt.xlabel('Artist')
plt.ylabel('Frequency')
```

```
Text(0, 0.5, 'Frequency')
```



Splice the first character of each artist's name and compute the number of times each letter appears as the first character in a name.

Once again, we are importing a matplot library in order to utilize the plt function to create a bar chart in descending order of frequency. Displaying the highest frequency to the lowest.

In a business setting this particular analysis can be used to utilize phonetic data for market research purposes.

Conclusion

Our main points were.....

→ To show the frequency of each artist roster.

→ To rank the differences from last week to the current week.

→ To splice the first letter of each artist name and compute the number of times each letter appears as the first character in a name.

Thank you!

Q/A

TEXAS STATE UNIVERSITY®

The rising STAR *of Texas*