National College of Ireland

BSc Computing

Data Analytics

2020/2021

Conor Carroll

X14535037

X14535037@student.ncirl.ie

Commercialism in Popular Music: Analysing brand mentions in song lyrics. Using Machine Learning to create lyrics in the style of different genres.

Technical Report

Contents

Executive Summary

In my report I will document my process of building a tool from scratch which is capable of identifying brand mentions within song lyrics. I will show how I get my raw data from the API provided by Genius.com, parse the data and create a data warehouse with all the brand mentions among other items of data.

I will also document how I created my list of brands to check for, using R programming language to web scrape lists from various sources on the net. The process I have built is usable for future artists going forward and could be modified to be used to check for other pieces of information within the music lyrics. I will be showing how I go about visualising the end dataset using Tableau.

Some analysis will be performed on the finished dataset and findings reporting. Some of the findings include, the rap genre has significantly more brand mentions than pop and country music and female and male rap artists tend to mention brands a similar number of times. I will also be looking at the correlation between Google Trends data and the number of times a brand is mentioned in that location.

As well as this, I will be documenting the creation of a Recurrent Neural Network in Python which is able to generate lyrics in the style of the three different genres looked at in this project. The Neural Network trains itself on the lyrics of genres and produces its own, based on the input data.

1.0 Introduction

1.1. Background

Commercial Brands have tapped into hip-hop artists for more than 25 years to promote their products. An example of a ground-breaking marketing campaign launched in the 90's revolving around hip-hop and urban culture was Sprite's "Obey Your Thirst" campaign. It revolved around hip-hop artists, basketball players and other urban themes. The campaign helped Sprite cultivate an image of the drink of hip, sometimes cynical young people everywhere. In terms of sales impact, the campaign was very successful, increasing their sales volume by 13% and increasing their soft drink market share (Sunset, 2008).

Other brands have not been so quick to embrace the image of a brand related to hip-hop/urban culture. The champagne brand "Cristal" was a favourite among rappers in the early and late 2000s, with rappers like Jay-Z claiming in his lyrics to be drinking "Cristal by the bottle". However, Louis Roederer, the makers of Cristal, did not welcome their association with rap music. In an interview with the economist, Frederic Rouzaud, the managing director of Louis Roederer, when asked about their association with rap music and if it could be detrimental said:

"*That's a good question, but what can we do? We can't forbid people from buying it. I'm sure Dom Perignon or Krug would be delighted to have their business.*"

This led to many big names in the industry boycotting the brand. It also directly led to Jay-Z starting his own brand of champagne and competitor to Cristal, "Ace of Spades". He also refused to sell it anymore at his range of bars in Manhattan (Roberts, 2009).

Although it seems that as of more recent times, brands who would have had a similar viewpoint as Cristal, in a sense that they wanted to distance themselves from rap music are starting to make a U-turn. For example, the Atlanta based rap artist Radrick Davis who goes by the stage name "Gucci Mane", claimed he received over 100 cease and desist letters in the early stage of his career (2006-2011) from the famed Italian fashion designer Gucci. The fashion brand Gucci's upmarket image did not want to be associated with the rap artist, whose lyrics referenced the harsh realities and violence associated with growing up in the low-income eastern suburbs of Atlanta. However, the culture has changed and evolved. In a complete turnaround in the relationship, Radrick Davis went on to become the face of Gucci's 2020 "Cruise Collection" (Hore-Thorburn, 2020).

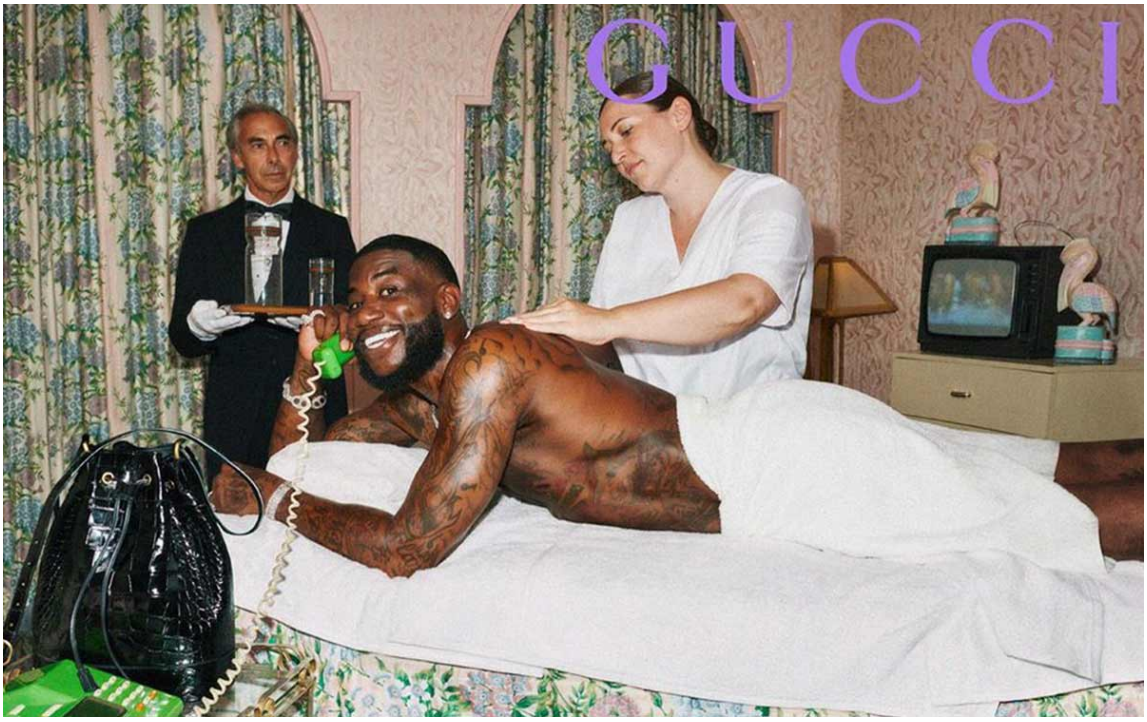


Fig.1 Gucci Mane models for Gucci's "Cruise Collection"

As of late, these hip-hop brand partnerships have really ramped up, Post-Malone is the poster boy for Bud Light, Snoop Dogg has partnered up with Just-Eat to promote their services, Cardi B went from reality show participant to platinum selling artist and the face of campaigns for Reebok and Pepsi. That is just to name a few, there are countless examples. This may be due to the fact that rap music has become more mainstream. In my lifetime, rap music has gone from a somewhat niche genre and subculture to the cultural zeitgeist it is today. In 2017, rap music became the most popular music in the world.

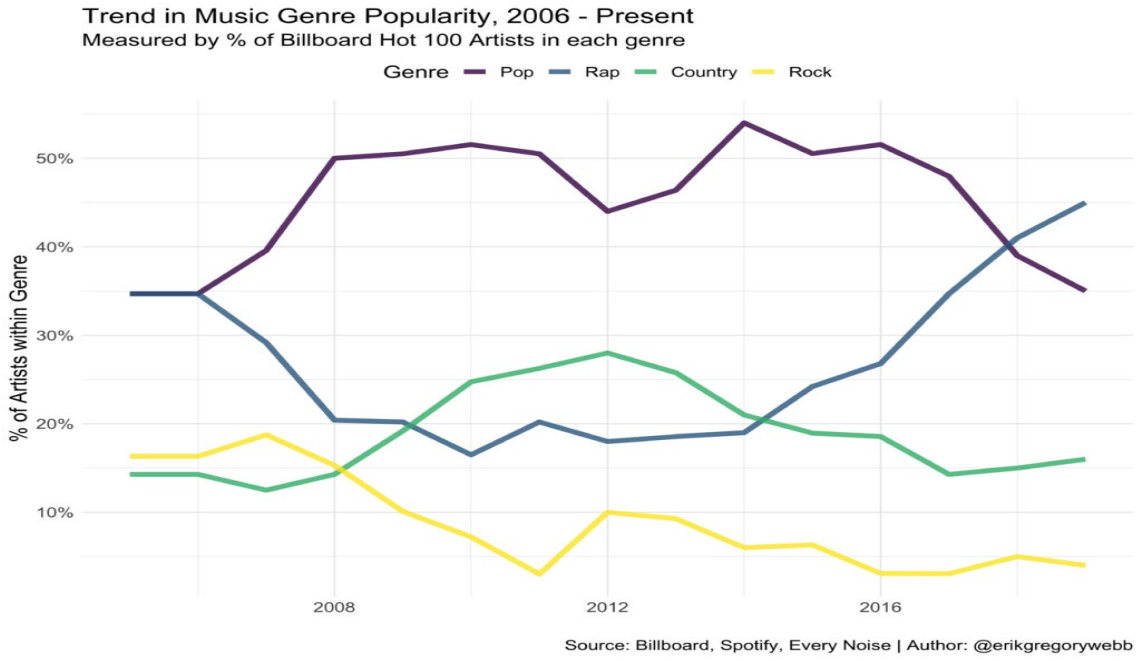


Fig. 2 Trend chart showing Rap music overtaking Pop

This trend of brands partnering up is not expected to stop anytime soon (Kelly, 2020). Kelly Bayett, co-founder of Barking Owl music group said that these partnerships have slowed due to the coronavirus pandemic but:

"*Once things calm down, I anticipate the hip-hop-based marketing and brand collaborations will skyrocket. People will be ready to let loose and have a good time*"

1.2. Motivation

For the project, I decided to do some research on what brands are most frequently mentioned by artists within their lyrics. I found a number newspaper articles and research papers which analysed this (This Is the Most Name-Dropped Brand in Music, 2017). The greatest number of songs analysed in one study that I found was 560 songs (Craig, Flynn and Holody, 2017). I felt that this was a rather small amount and that I could do this on a bigger scale. I wanted to create a data warehouse which held data on artists, and which brands they mentioned and the frequency of mentions. I wanted also to create the tool so it could be used to analyse any lyrics going forward and could be done with only a few clicks of a button.

I also felt that this could have commercial usage. Marketing departments for brands can check and see with which artists/genres/locations they are popular in and make decisions based on this. In order to achieve this, I decided to build a tableau dashboard, which will visualise the data from the data warehouse.

I will explore the difference between genres when it comes to brand mentions, their frequency and type of brands mentioned. I wanted to see if there is a difference between genders when it comes to the brand mentions. I will also be exploring if the amount of brand mentions within music correlates with Google trends data for the same region/time.

As well as analysing the content of the lyrics, I want to use AI in order to generate music lyrics from our dataset. The generated lyrics will be done on a genre-by-genre basis, using a Neural Network.

I have chosen to undertake this project as it integrates multiple aspects of my learning from my time in college. I feel it encapsulates and showcases multiple disciplines such as data pre-processing, data warehousing, machine learning, data visualisation and business data analysis.

1.3. Aims

Aim 1: **Create a data warehouse which stores information on artists, and which brands they are mentioning in their lyrics/how frequently they are being mentioned.**

Aim 2: **Provide a comparison between Genres to see what brands/types of brands are mentioned and how frequently they are mentioned.**

Aim 3: **Allow users to search for brands/artists to see where they are popular and with which artists/genres they are popular with.**

Aim 4: **Generate lyrics based on a style of genre, using a machine learning algorithm. Perform the lyrics using a text to speech tool over a backing track.**

Aim 5: **Find out if there is a statistically significant difference between the genres when it comes to frequency of brand mentions.**

Aim 6: **Find out if there is a statistically significant difference between the genders when it comes to frequency of brand mentions.**

Aim 7: **Determine if there is a correlation between frequency of brand mentions and Google trends data.**

1.4. Technology

Python 3.8 – Python is a high-level, general-purpose programming language. Python is used to connect to the API to download datasets. The Neural Network will be created in a Python Notebook.

TensorFlow – TensorFlow is an open-source software library for machine learning. I will be using it to build the neural network for the lyric generation aspect of the project.

R – R is a statistical computing programming language often used in the world of data science. R will be used to web scrape data to create our list of brands dataset. It will also be used to perform statistical analysis on our finished data set.

Alteryx – Alteryx is a data manipulation tool which allows for data transformation, data mining and visualization. Alteryx is used to parse the JSON files downloaded from the API and mine through the data for mentions of brands. It will also upload the data to our SQL server.

Tableau – Tableau is a data visualisation tool focused on business intelligence. It is used in this project to visualize our data after we have processed it and stored it in our data warehouse.

Microsoft SQL Server – Microsoft SQL Server is a relational database management system. Once the data is processed, it will be stored here for use for Tableau and R.

Genius API – This is an open API from Genius.com, it allows connection to their database so users can request data on any artist.

CRISP-DM – Stands for Cross-industry standard process for data mining. It is an approach for data mining. It is the most widely used analytics model.

1.5. Structure

I will first address the data used in the project, where it is drawn from and the schema of it. I will then focus on my methodology and how I went about formatting and analysing my data. Then I will explore the findings within the data, share the results and state my conclusions based on my tests.

2.0 Data

Genius API

Initially when I had the initial idea for the project, I assumed I would need to web scrape the lyrics using R. I was familiar with the website Genius.com, from looking up music lyrics. Genius.com hosts music lyrics from nearly every music artist. When researching the project, I did a google search to see if others had tried web scraping from Genius.com, this is when I became aware of the Genius API. It is an open API which allows users to connect to it through a Python script and can be used to download data from any artist hosted in its database.

Once the script is finished running, it will drop a JSON file in the directory of the Python script. The JSON file contains a vast amount of data, including the lyrics. The JSON files tend to be around 15000/20000 lines. It will need a lot of pre-processing in order to mine the lyrics for brand mentions, this will be done within Alteryx.

My entire dataset was 209 individual JSON files, one per artist. Each file contained the lyrics to 40 songs per artist, which works out as 8,360 songs. The breakdown per genre is 103 Rap artists, 55 Pop artists and 51 Country Artists.

Brand List

The other dataset used in my project I have compiled from scratch. To get a dataset of some of the most popular brands, I web scraped lists from multiple sources online using R programming language with the rvest and dplyr packages. The different sources were:

- **Various Luxury Brands:** https://en.wikipedia.org/wiki/Category:Luxury_brands
- **Beer:** https://eu.usatoday.com/story/money/2019/06/19/beer-brands-americas-31-most-popular-beers/39490347/
- **Tobacco:** https://brandirectory.com/rankings/tobacco/2017/table
- **Whiskey:** https://vinepair.com/articles/20-most-popular-whiskey-brands-america-2018/
- **Clothing:** https://www.endclothing.com/eu/brands
- **Soft Drinks:** https://en.wikipedia.org/wiki/List_of_brand_name_soft_drink_products
- **Miscellaneous Brands:** https://ig.ft.com/top-100-global-brands/2018/

Once the data was scraped in R, I added them all to a singular csv file. The final dataset had three columns, one for the actual name of the brand, one for the word it is looking for in the lyrics and the last column categorised the brand (car, clothing etc.). The list required some manual intervention regarding adding nicknames for some brands. It is not uncommon for brands to be referred to by names other than their proper name. To name a few, “Lambo” for Lamborghini, “Rollie” for Rolex etc. I had to go add these in manually to the best of my knowledge. The final dataset looked for 175 different mentions of brands within the lyrics. The dataset was stored within the Alteryx workflow which checks for the mentions.

Google Trends

For my analysis section I will be checking if there is a correlation between frequency of mentions in lyrics and popularity with google searches. To check this, I will be downloading trend data from google trends. This comes in a CSV format, with a breakdown of popularity on google from all 50 states in the USA. It ranks the popularity of a score out of 100.

3.0 Methodology

For the project I will be using CRISP-DM. It is a six-phase model which has business needs at the heart of it. As my project could have real world value for businesses, I felt it was appropriate to choose this model. My project consists of two separate components, the lyrics analysis and the lyrics generation, I will do CRISP-DM for each separately.

Business Understanding

Lyrical Analysis – The end goal from a business standpoint is to create a tool which allows lyrics from an artist to be analysed and checked for brand mentions within seconds. We want to create a relational database from the JSON files which we have downloaded from the Genius API. We want to then visualise the final dataset on an interactive Tableau dashboard which will give insights into which artists/genres the brand is popular with and where geographically the brand is getting the most mentions.

Lyrical Generation – The end goal from a business standpoint is to create a Python script which is capable of mimicking the style of a certain genre's lyrics and output them with the click of a button.

Data Understanding

Lyrical Analysis – We will collect all our data related to artist's (lyrics, gender, genre etc.) by connecting to the Genius API using a Python script. The dataset is 209 different JSON files, each containing the data for a single artist. The data will require a lot of preparation in order to be analysed for brand mentions. We will need to create a dataset which contains all the brand names which are being looked for. I will do this by web scraping multiple sources which have lists of popular brands and combining them all into one cell.

Lyrical Generation – For our lyric generation, the only input dataset that is required is a .txt file which contains the lyrics for it to train on. We will create this dataset by taking the lyrics out of the JSON files for the artist and outputting it as a .txt file.

Data Preparation

Lyrical Analysis – In each JSON file there is a huge amount of redundant data which is not relevant to the project, we will need to select the data which pertains to the lyrics, artist name, song year and description of the artist as that will contain all the data we require for our project. We will then need to take the lyrics from the JSON file, tokenize and clean the data, which will be done in Alteryx.

Lyrical Generation – We will take out the lyrics from the JSON files and output them as .txt format. We will host the dataset online for our neural network to connect to.

Modelling

Lyrical Analysis – For the brand analysis we will build a model which checks the words in the lyrics and searches for patterns which contain the brand mentions.

Lyrical Generation – We will use a Long Short-Term Memory Recurrent Neural Network for this. We have chosen this as it has a proven track record in generating speech, as it has an onus on the sequence in which words are generated.

Evaluation

Lyrical Analysis – We will evaluate our lyrical analysis by seeing how much data we have in our final dataset. I will manually check the accuracy of a few artists to check for false positives and adjusts the model accordingly.

Lyrical Generation – We will assess this by checking our loss figure. The lower the loss, the more accurate the predicted text is to the input dataset. We will reduce the loss by adjusting the number of epochs the model trains.

4.0 Implementation

4.1. ETL

To create our data warehouse with the brand mentions and information for each artist using ETL, we must extract our data from multiple sources.

Extraction

Connecting to Genius API with Python

The first step of the project is to gather our dataset of our JSON files by connecting to the Genius API. The Genius API has a lot of documentation and plenty of resources online to help you with using it which was ideal for the project as I had no prior knowledge of it. The only thing that was required before using the API was to generate an access Token to connect to it. This was free and all it required me to do was create an account on their website. Once I had my key to connect to the API, I was free to write my Python script which looked like this:

```
#Assign your Genius.com credentials and select your artist
import lyricsgenius as genius
geniusCreds = "eaIZHw-d7dxFiQsoUoPyimMC3N19a_9ix8qYFmbwMZniPCNg4auwJ0lo-u6uhMs_"
artist_name = "Sam Smith"
artist_name1 = "Kesha"
artist_name2 = "Shawn Mendes"
artist_name3 = "Mumford & Sons"
artist_name4 = "Selena Gomes"
artist_name5 = "Meghan Trainor"
artist_name6 = "Pitbull"
artist_name7 = "Mihcael Buble"
artist_name8 = "Jason Derulo"
artist_name9 = "Chainsmokers"
artist_name10 = "Halsey"

api = genius.Genius(geniusCreds)
import os
os.getcwd()
artist = api.search_artist(artist_name, max_songs=40)
artist.save_lyrics()
artist1 = api.search_artist(artist_name1, max_songs=40)
artist1.save_lyrics()
artist2 = api.search_artist(artist_name2, max_songs=40)
artist2.save_lyrics()
artist3 = api.search_artist(artist_name3, max_songs=40)
artist3.save_lyrics()
artist4 = api.search_artist(artist_name4, max_songs=40)
artist4.save_lyrics()
artist5 = api.search_artist(artist_name5, max_songs=40)
artist5.save_lyrics()
artist6 = api.search_artist(artist_name6, max_songs=40)
artist6.save_lyrics()
artist7 = api.search_artist(artist_name7, max_songs=40)
artist7.save_lyrics()
artist8 = api.search_artist(artist_name8, max_songs=40)
artist8.save_lyrics()
artist9 = api.search_artist(artist_name9, max_songs=40)
artist9.save_lyrics()
artist10 = api.search_artist(artist_name10, max_songs=40)
artist10.save_lyrics()
```

Fig.3 Python script connecting to the Genius API

First, we import the LyricsGenius Python client for the API and use add in our access token which we received by creating an account on the Genius website. The next step is to select our artists and assign them to “artist_name”. Then we connect to the API again using our credentials and import the working directory. We then search for the artist’s name in the API and set the max number of songs it will download the lyrics for to 40. The script will start search for the artist and start listing the songs until it reaches its 40-song limit.

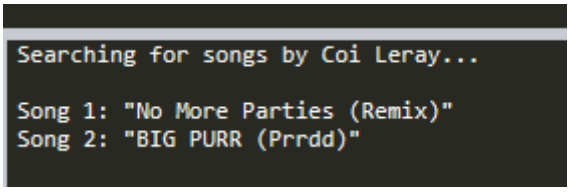


Fig.4 Python script connected to the Genius API listing artist’s songs

It will take a couple of minutes for each artist to complete. Once the script is finished running, it will drop a JSON file into the working directory. One issue I ran into while completing this step is that the connection to the API can time out and stop working. This had a few ramifications that I had to work around. Originally, I had the script set up that it collects all the artists songs then saves the lyrics.

```
11  api = genius.Genius(geniusCreds)
12  import os
13  os.getcwd()
14  artist = api.search_artist(artist_name, max_songs=40)
15  artist1 = api.search_artist(artist_name1, max_songs=40)
16  artist2 = api.search_artist(artist_name2, max_songs=40)
17  artist3 = api.search_artist(artist_name3, max_songs=40)
18  artist4 = api.search_artist(artist_name4, max_songs=40)
19
20  artist.save_lyrics()
21  artist1.save_lyrics()
22  artist2.save_lyrics()
23  artist3.save_lyrics()
24  artist4.save_lyrics()
```

Fig.5 Original Python script structure

However, when a timeout occurred while searching for artist’s songs, none of the previous artist’s it had searched for would be saved. I worked around this by saving the JSON file immediately after searching like in figure 3.

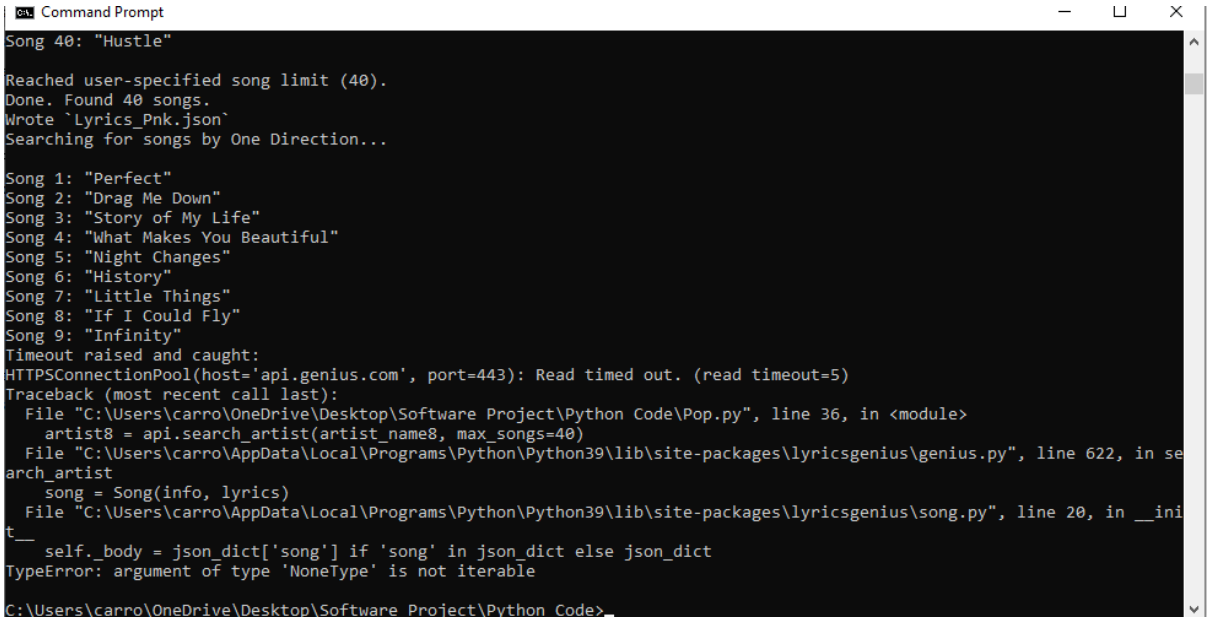


Fig.6 An example of the connection to the API timing out

Another ramification this had was that I could not make one Python script which contained every artist I wanted to analyse for my project. This meant I had to create many scripts with 10 or so artists and re-run it from the artist that the timeout occurred if one did occur. This meant that the collection of JSON files was rather time consuming, the script took up to 20/30 minutes to run for 10 artists, I would also have to supervise it as opposed to letting it run without interruption. This meant that the collection of JSON files was a couple days work. Once we had ran all the Python scripts, we had 209 JSON files from different artist's.

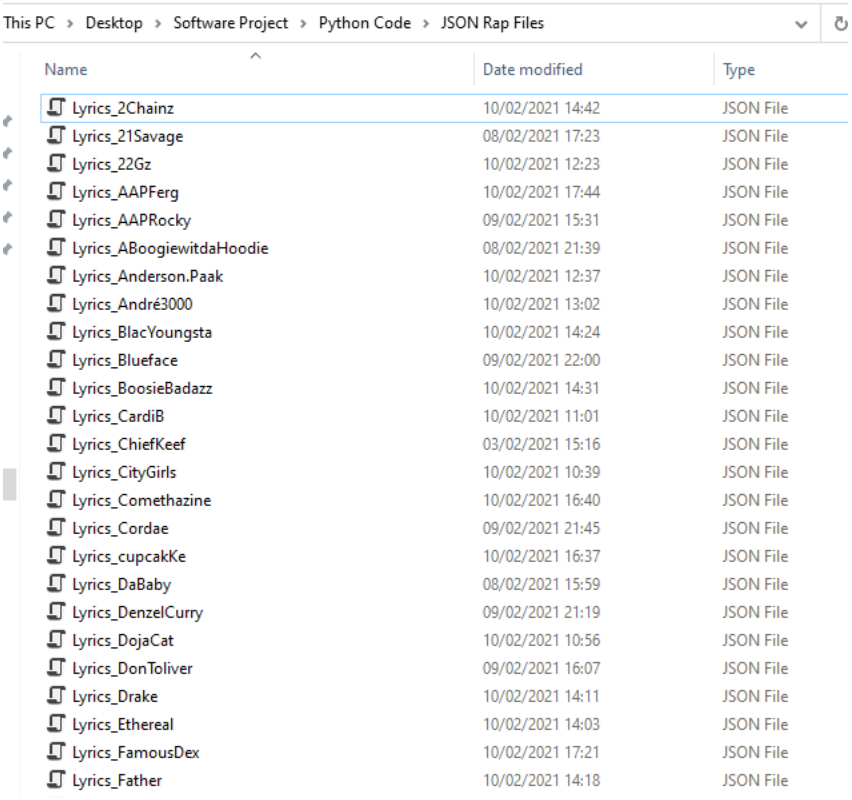


Fig.7 Folder hosting the downloaded JSON files

Once we had our artist related data downloaded from the Genius API, we needed to create our list of brands to search for within the lyrics.

Web Scraping Brand List

This was another rather time-consuming part of the project. I tried to find suitable datasets online which contained brand names that I could use for the project on websites like Kaggle. There were several CSV files which contained millions of brand names, most of which I had not heard of before and were not relevant to the project. After failing to find a premade file which I could use, I decided that I needed to create the dataset myself. I had originally planned on web scraping the music lyrics, so I had researched prior on how to do this. I performed the web scraping in R-Studio.

```
1  library(rvest)
2  library(dplyr)
3
4  link = "https://brandirectory.com/rankings/tobacco/2017/table"
5  page = read_html(link)
6
7  name = page %>% html_nodes(".tight-text") %>% html_text()
8
9  brands = data.frame(name)
10
11 write.csv(brands, "Tobaccobrands.csv")
```

Fig.8 R script web scraping the list of tobacco brands

This was achieved using the 'rvest' package which allows web scraping to be done. We then use the 'dplyr' package in order to organise the data into a data frame. The web scraping itself was a quick process but getting suitable lists to web scrape from different part of the web was a challenge. Once the web scraping had taken place, we would be left with a dataset like this:

	name2
1	Goyard
2	Graff (jewellers)
3	Grand Dorsett
4	Grant Macdonald
5	Gravati
6	Great Greenland Furhouse
7	Greubel Forsey
8	Grotrian-Steinweg
9	H. Moser & Cie
10	H.J. Cave & Sons
11	H.Stern
12	Haerfest (fashion brand)
13	Halston
14	Hamilton Watch Company
15	Hardy Brothers
16	Pierre Hardy (fashion designer)
17	Harry Winston, Inc.
18	Hästens
19	Hédiard
20	Heming (company)

Fig.9 List of fashion brands after being web scraped in R

This is only step one for our brand list dataset. As our data mining model can only check one word at a time, we have to choose one word which the model will look for. This was done by me manually. For example, a brand that was frequently mentioned was the fashion brand “Maison Margiela”. I had to have a cell with only “Margiela” in it for the model to check for in the lyrics. As well as that, I added another cell of what category the brand is in.

	Brand	Actual Name	Type
1	Gucci	Gucci	Clothing
2	Bentley	Bentley	Car
3	Xanny	Xanax	Pharmaceutical
4	Bentleys	Bentley	Car
5	Audemars	Audemar Piguet	Watch Maker
6	AP	Audemar Piguet	Watch Maker
7	Goyard	Maison Goyard	Bag Maker
8	Benz	Mercedes-Benz	Car
9	Patek	Patek Philippe	Watch Maker
10	Pateks	Patek Philippe	Watch Maker
11	Bugatti	Bugatti Veron	Car
12	OffWhite	Off-White	Clothing
13	Bape	A Bathing Ape	Clothing

Fig.10 Sample of our brand dataset

As above, the left column is what is being checked for in the lyrics. Another factor I had to consider was that often in lyrics brands will be referred to by a nickname. I had to add the nicknames to brands manually to the best of my ability, this meant from my own knowledge and also reading through some lyrics from my dataset. I stored this dataset within the Alteryx workflow

Transform

Alteryx Workflow

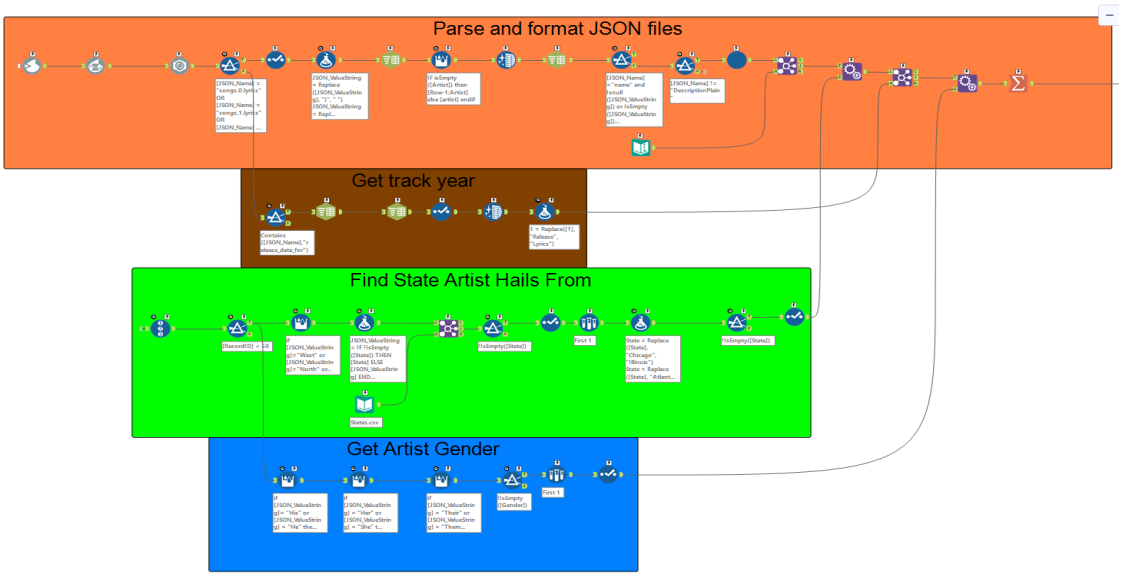


Fig.11 Alteryx Workflow Layout

There are four separate parts of my workflow as labelled. Each gets a different piece of data relevant to the project, that we can use for our analysis. During my internship I had a lot of experience using Alteryx, so I did not need to learn it from scratch. The first challenge I faced when building the Alteryx workflow was it does not accept JSON files as input with the standard input tool. This meant that I had to input the JSON file as a blob and convert the blob to a text and finally use the JSON parse tool to parse it. This left us with our data like this:

5 of 5 Fields | Cell Viewer | * 5,936 of 15,030 records displayed(partial results)

Record	JSON_Name	JSON_ValueString
1	alternate_names.0	Jatavia Shakara Johnson & Caresha Romeka Brownlee
2	alternate_names.1	Jatavia Johnson & Caresha Brownlee
3	alternate_names.2	JT & Yung Miami
4	api_path	/artists/1277050
5	description.plain	City Girls, which comprises of JT and Yung Miami, is a Miami-based rap duo—most notab...
6	facebook_name	[Null]
7	followers_count	[Null]
8	header_image_url	https://images.genius.com/88b20f414f0f9c79800d030ff92d7995.640x640x1.jpg
9	id	[Null]
10	image_url	https://images.genius.com/62454a82e6ad9112c51d8a574c0b5357.640x640x1.jpg
11	instagram_name	citygirls
12	is_meme_verified	[Null]
13	is_verified	[Null]
14	name	City Girls
15	translation_artist	[Null]
16	twitter_name	[Null]
17	url	https://genius.com/artists/City-girls
18	current_user_metadata.permissions.0	view_activity_stream

Fig.12 JSON parsed in Alteryx

In the above example, there are 15,030 rows. All the JSON files had similar amounts of rows, the vast majority of them being redundant and not useful for the project. The next step was to select only the rows which we needed. This required me to go through the file structure and find out where the relevant data is stored. We do this by selecting the cells in the "JSON_Name" Column using a filter tool. I will break this down for each section of the workflow.

Brand mentions – We need the column that relates to the lyrics and the artist's name.

```
[JSON_Name] = "songs.0.lyrics" OR
[JSON_Name] = "songs.1.lyrics" OR
[JSON_Name] = "songs.2.lyrics" OR
[JSON_Name] = "songs.3.lyrics" OR
[JSON_Name] = "songs.4.lyrics" OR
[JSON_Name] = "songs.5.lyrics" OR
[JSON_Name] = "songs.6.lyrics" OR
[JSON_Name] = "songs.7.lyrics" OR
[JSON_Name] = "songs.8.lyrics" OR
[JSON_Name] = "songs.9.lyrics" OR
[JSON_Name] = "songs.10.lyrics" OR
[JSON_Name] = "songs.11.lyrics" OR
[JSON_Name] = "songs.12.lyrics" OR
[JSON_Name] = "songs.13.lyrics" OR
[JSON_Name] = "songs.14.lyrics" OR
[JSON_Name] = "songs.15.lyrics" OR
[JSON_Name] = "songs.17.lyrics" OR
[JSON_Name] = "songs.18.lyrics" OR
[JSON_Name] = "songs.19.lyrics" OR
[JSON_Name] = "songs.20.lyrics" OR
[JSON_Name] = "songs.21.lyrics" OR
[JSON_Name] = "songs.22.lyrics" OR
[JSON_Name] = "songs.23.lyrics" OR
[JSON_Name] = "songs.24.lyrics" OR
[JSON_Name] = "songs.25.lyrics" OR
[JSON_Name] = "songs.26.lyrics" OR
[JSON_Name] = "songs.27.lyrics" OR
[JSON_Name] = "songs.28.lyrics" OR
[JSON_Name] = "songs.29.lyrics" OR
[JSON_Name] = "songs.30.lyrics" OR
[JSON_Name] = "songs.31.lyrics" OR
```

Results - Filter (9) - Out - True

5 of 5 Fields | Cell Viewer | 41 records displayed

Record	JSON_Name	JSON_ValueString
1	description.plain	City Girls, which comprises of JT and Yung Miami...
2	name	City Girls
3	songs.0.lyrics	[Intro]
4	songs.1.lyrics	[Intro: J.T.]
5	songs.2.lyrics	[Intro: Yung Miami & Cardi B]
6	songs.3.lyrics	[Intro: Yung Miami]
7	songs.4.lyrics	[Intro: J.T.]
8	songs.5.lyrics	[Intro: Yung Miami]
9	songs.6.lyrics	[Chorus: JT & Yung Miami]
10	songs.7.lyrics	[Intro: JT, Pee & Yung Miami]
11	songs.8.lyrics	[Chorus: J.T. &
12	songs.9.lyrics	[Chorus: J.T.]
13	songs.10.lyrics	[Verse 1: Yung Miami]
14	songs.11.lyrics	[Intro: J.T.]
15	songs.12.lyrics	[Intro]
16	songs.13.lyrics	[Intro: J.T. &
17	songs.14.lyrics	[Intro]
18	songs.15.lyrics	[Chorus: Ball Greezy,

Fig.13 Filter tool configured to pick the relevant data

Once we have our relevant data, we need to break it down to single cells per word in order to allow it to be searched through for the brand mentions. We start this by splitting the cells to rows, using the space between words as a separator.

ds | Cell Viewer | * 18,807 of 20,845 records displayed(partial results)

	JSON_Name	JSON_ValueString
2	songs.0.lyrics	
3	songs.0.lyrics	
4	songs.0.lyrics	on
5	songs.0.lyrics	the
6	songs.0.lyrics	beat
7	songs.0.lyrics	
8	songs.0.lyrics	1:
9	songs.0.lyrics	JT
0	songs.0.lyrics	&
1	songs.0.lyrics	
2	songs.0.lyrics	Miami
3	songs.0.lyrics	
4	songs.0.lyrics	ass
5	songs.0.lyrics	bitch,
6	songs.0.lyrics	give
7	songs.0.lyrics	a
8	songs.0.lyrics	fuck
9	songs.0.lyrics	'bout
0	songs.0.lyrics	a
1	songs.0.lyrics	nigga
2	songs.0.lyrics	Birkin
3	songs.0.lyrics	bag,
4	songs.0.lyrics	hold
5	songs.0.lyrics	five,
6	songs.0.lyrics	six

Fig.14 Data after being split to rows, highlighted is a brand mention (Birkin is a bag produced by Hermès)

Once we have each word in a cell, we filter out the blank cells and remove all punctuation and other symbols. We also need to add a column for the artist's name, which is contained in one of the cells we have selected.

3 of 3 Fields | Cell Viewer | * 14,633 of 21,064 records displayed(partial results)

Record	JSON_Name	JSON_ValueString	Artist
4	Songs0Lyrics	Earl	City Girls
5	Songs0Lyrics	On	City Girls
6	Songs0Lyrics	The	City Girls
7	Songs0Lyrics	Beat	City Girls
8	Songs0Lyrics	Verse	City Girls
9	Songs0Lyrics	1	City Girls
10	Songs0Lyrics	Jt	City Girls
11	Songs0Lyrics	Yung	City Girls
12	Songs0Lyrics	Miami	City Girls
13	Songs0Lyrics	Real	City Girls
14	Songs0Lyrics	Ass	City Girls
15	Songs0Lyrics	Bitch	City Girls

Fig.15 Our data ready to be searched through for bran mentions

We are then left with our data ready to be searched through for brand mentions. Once our workflow has identified the brand mentions by checking it against our other dataset that we created we are left with the following:

6 of 6 Fields | Cell Viewer | 120 records displayed

Record	JSON_Name	JSON_ValueString	Artist	Brand	Actual Name	Type
4	Songs7Lyrics	Audemars	City Girls	Audemars	Audemar Piguet	Watch Maker
5	Songs37Lyrics	Audemars	City Girls	Audemars	Audemar Piguet	Watch Maker
6	Songs3Lyrics	Bentleys	City Girls	Bentleys	Bentley	Car
7	Songs3Lyrics	Bentleys	City Girls	Bentleys	Bentley	Car
8	Songs3Lyrics	Bentleys	City Girls	Bentleys	Bentley	Car
9	Songs3Lyrics	Bentleys	City Girls	Bentleys	Bentley	Car
10	Songs3Lyrics	Benz	City Girls	Benz	Mercedes-Benz	Car
11	Songs3Lyrics	Benz	City Girls	Benz	Mercedes-Benz	Car
12	Songs3Lyrics	Benz	City Girls	Benz	Mercedes-Benz	Car
13	Songs3Lyrics	Benz	City Girls	Benz	Mercedes-Benz	Car
14	Songs4Lyrics	Benz	City Girls	Benz	Mercedes-Benz	Car
15	Songs4Lyrics	Benz	City Girls	Benz	Mercedes-Benz	Car
16	Songs4Lyrics	Benz	City Girls	Benz	Mercedes-Benz	Car
17	Songs5Lyrics	Benz	City Girls	Benz	Mercedes-Benz	Car
18	Songs6Lyrics	Benz	City Girls	Benz	Mercedes-Benz	Car
19	Songs13Lyrics	Benz	City Girls	Benz	Mercedes-Benz	Car
20	Songs19Lyrics	Benz	City Girls	Benz	Mercedes-Benz	Car
21	Songs19Lyrics	Benz	City Girls	Benz	Mercedes-Benz	Car
22	Songs19Lyrics	Benz	City Girls	Benz	Mercedes-Benz	Car
23	Songs21Lyrics	Benz	City Girls	Benz	Mercedes-Benz	Car
24	Songs24Lyrics	Benz	City Girls	Benz	Mercedes-Benz	Car
25	Songs30Lyrics	Benz	City Girls	Benz	Mercedes-Benz	Car
26	Songs0Lyrics	Birkin	City Girls	Birkin	Hermès	Clothing

Fig.16 How our data once the brand mentions within the lyrics have been identified

In the above sample, our model identified 120 brand mentions within the 40 songs from the artist's "City Girls". We will then count how many times each brand is mentioned per song.

Year Song was Released – The next thing we need to do is to get the year the song was made, as some of our analysis will focus on brand mentions over the years. In the "JSON_Name" column each song has its own unique number in between "Songs" and "Lyrics". In our JSON file we also have the year the song was made in the following format.

JSON_Name	JSON_ValueString
songs.0.release_date_for_display	November 16, 2018
songs.1.release_date_for_display	May 11, 2018
songs.2.release_date_for_display	November 9, 2018
songs.3.release_date_for_display	June 20, 2020
songs.4.release_date_for_display	February 23, 2018
songs.5.release_date_for_display	May 11, 2018
songs.6.release_date_for_display	November 16, 2018
songs.7.release_date_for_display	October 8, 2019
songs.8.release_date_for_display	May 11, 2018
songs.9.release_date_for_display	May 11, 2018
songs.10.release_date_for_display	June 19, 2020
songs.11.release_date_for_display	May 11, 2018
songs.12.release_date_for_display	June 20, 2020
songs.13.release_date_for_display	May 11, 2018
songs.14.release_date_for_display	November 16, 2018

Fig.17 Release date of songs within the JSON file

We start by splitting the "JSON_ValueString" Column and using the comma "," as a delimter. This leaves us with only the year in a column. In order to join the year with the brand mentions dataset we need to do it by the JSON_Name column. We again split that but use the underscore "_" as a delimiter. We then remove the punctuation and replace the word "release" with "Lyrics"

1	Year
Songs0Lyrics	2018
Songs1Lyrics	2018
Songs2Lyrics	2018
Songs3Lyrics	2020
Songs4Lyrics	2018
Songs5Lyrics	2018
Songs6Lyrics	2018
Songs7Lyrics	2019
Songs8Lyrics	2018
Songs9Lyrics	2018
Songs10Lyrics	2020
Songs11Lyrics	2018
Songs12Lyrics	2020
Songs13Lyrics	2018

Fig.18 Song year with the corresponding song lyrics identifier

We can then use a join tool and combine this with our brand mentions dataset.

Artist's State – My project focuses on artists from the United States of America. I will be visualising the data on a map of the USA, so it is necessary to get the artist's state that they are from within the data. This is not found in a single cell like the artist's name or lyrics, so I had to be a little bit creative with how I found the artist's state. There is a cell within the parsed JSON which has a blurb written about the artist which usually mentions where the 4artist hails from within the first few lines. I filtered out this cell and broke it down into

single words per cell with the same method as seen above. Here is an example of how it looks:

RecordID	JSON_Name	JSON_ValueString
1	DescriptionPlain	City
2	DescriptionPlain	Girls
3	DescriptionPlain	Which
4	DescriptionPlain	Comprises
5	DescriptionPlain	Of
6	DescriptionPlain	Jt
7	DescriptionPlain	And
8	DescriptionPlain	Yung
9	DescriptionPlain	Miami
10	DescriptionPlain	Is
11	DescriptionPlain	A
12	DescriptionPlain	Florida
13	DescriptionPlain	Rap
14	DescriptionPlain	DuoMost

Fig.19 Artist description broken down cell by cell

The workflow will then search through the words for a state, in the above example we can see in cell 12 contains the data we are after. Once we have the state, we create a column for it and append it onto the brand mentions table we have.

Artist's Gender – Another piece of the analysis of the final dataset will focus on the gender of artists. We will split them into Male, Female and Group (multiple members). In order to get this, we use the above dataset of the blurb broken down into single words. The workflow then checks to see what pronouns are being used to describe them. If it finds he/him it will categorise the artist as male, if it finds she/her then female and they/their it will categorise them as a group. In our sample City Girls are a group, here is what we have at the end of this section of our workflow:

RecordID	JSON_Name	JSON_ValueString	Artist	Gender
27	DescriptionPlain	Their		Group
43	DescriptionPlain	Their		Group

Fig.20 Artist's gender has been found

The last piece of information is the genre. This is not held anywhere within the JSON file, so I had to make three separate workflows for each genre I will be analysing. At the end of the process the workflow adds a column with the genre in it (Rap, Pop or Country).

Load

Once we have all of our relevant data, we can upload it to our SQL Server. This can be done directly within Alteryx as it has the capability to connect to Microsoft SQL Server.

Our database contains 2 tables. One for artist's data called "Artist" and one for the brand mentions called "Brand".

Fig.21 Entity-Relationship Diagram for our data warehouse

With our Artist table we ended up with 186 artists, as some of our 209 had no brand mentions that were found.

State	Gender	Genre	Artist
Alabama	Female	Rap	Flo Milli
Arizona	Female	Country	Courtney Marie Andrews
California	Female	Pop	Billie Eilish
California	Female	Pop	Katy Perry
California	Female	Pop	Kesha
California	Female	Pop	P!nk
California	Female	Rap	Doja Cat
California	Female	Rap	Saweetie
California	Male	Country	Brett Young
California	Male	Country	Dave Alvin
California	Male	Country	The White Buffalo
California	Male	Rap	Anderson .Paak
California	Male	Rap	Blueface
California	Male	Rap	Kendrick Lamar
California	Male	Rap	Roddy Ricch
California	Male	Rap	Ty Dolla $ign
California	Male	Rap	Tyga
California	Male	Rap	YG

Fig.22 Artist table in our database

Our Brand table contains 4,763 rows, with a total of 11,233 brand mentions overall.

Count	Artist	Type	Brand	Year
2	Lil Yachty	Watch Maker	Patek Philippe	2017
3	Lil Yachty	Watch Maker	Patek Philippe	2018
2	Lil Yachty	Watch Maker	Rolex SA	2016
1	Lil Yachty	Watch Maker	Rolex SA	2017
1	Lil Yachty	Watch Maker	Rolex SA	2018
1	Lil Yachty	Weapons Manufact...	F&N	2018
4	Lil Yachty	Weapons Manufact...	Glock Ges.m.b.H.	2018
1	Logic	Alcohol	Hennessy	2014
1	Logic	Alcohol	Moët & Chandon	2013
1	Logic	Beverage	Kool-Aid	2014
1	Logic	Beverage	Kool-Aid	2018
2	Logic	Car	Cadillac	2015
1	Logic	Car	Chevy	2014
1	Logic	Car	Ford Motor Company	2019
1	Logic	Car	Mercedes-Benz	2014
1	Logic	Car	Porsche Design	2019
1	Logic	Clothing	Adidas	2014
2	Logic	Clothing	Nike Inc.	2018
1	Logic	Clothing	Rick Owens	2015

Fig.23 Brand table in our database

This is the final dataset I will be using for my analysis and visualisation. It is hosted on a Microsoft SQL Server.

4.2. Visualisation

To visualise the final dataset, I will be using Tableau. Data visualisations are an essential part to communicating the results of findings of this project. My Tableau dashboard is a critical element for the completion of my project. This is where marketing departments can jump in and delve through the brand mentions.

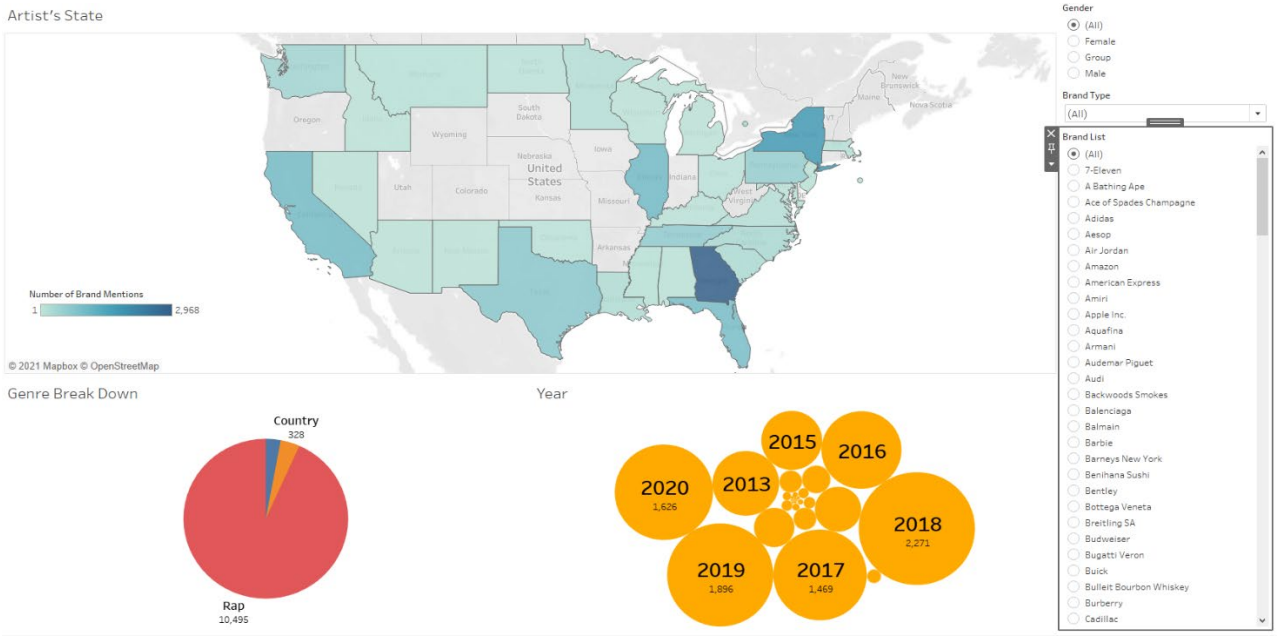


Fig.24 Tableau Dashboard 1

In figure 24 is the first Tableau dashboard, which focuses geographically where brands are being mentioned the most. It has a breakdown of genre using a pie chart and a breakdown of year using a bubble plot. On the right is some filter tools, one containing all the brands found within the lyrics. If an employee from a company's marketing department wanted to see where their brand is mentioned most frequently by state, genre, year, they could select their brand from the list and see all the breakdowns.

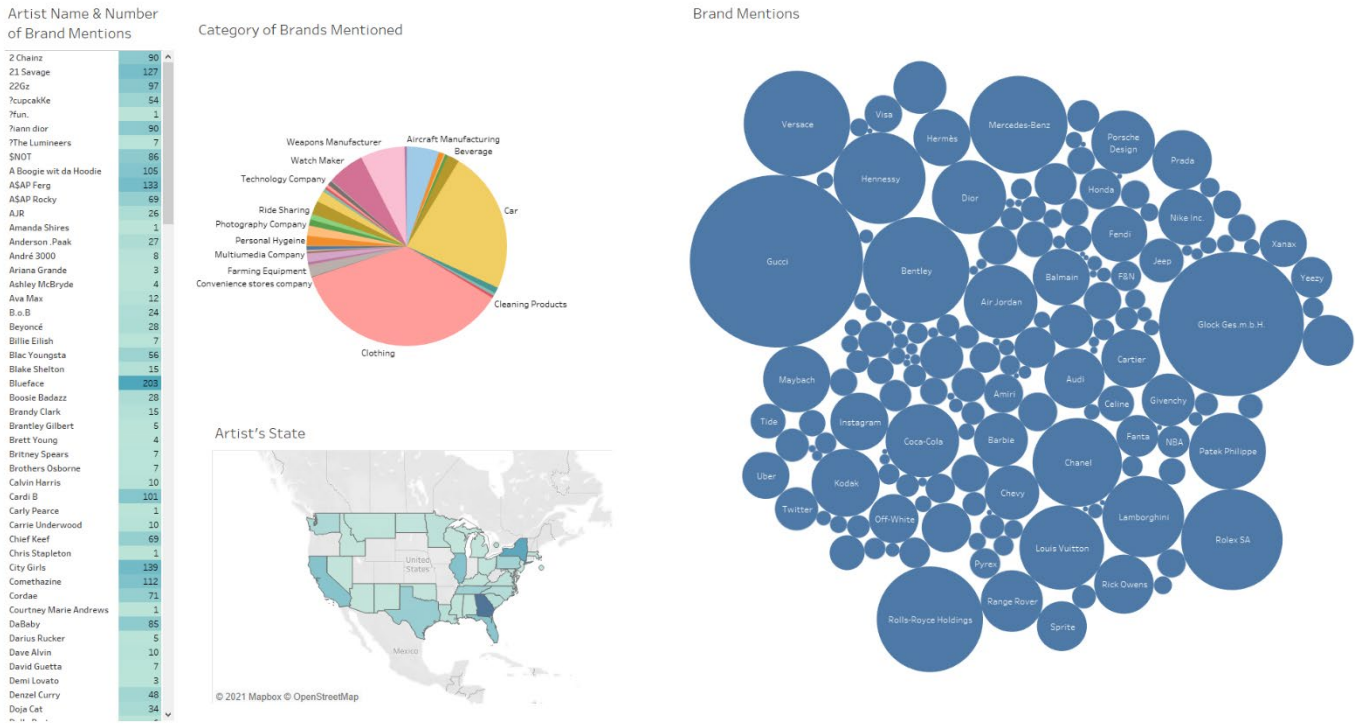


Fig.25 Tableau Dashboard 2

Figure 25 contains the second dashboard which is linked to the first dashboard. This dashboard is more artist focused. It shows the number of mentions each artist has. Category of brands are broken down is visualised in a pie chart.

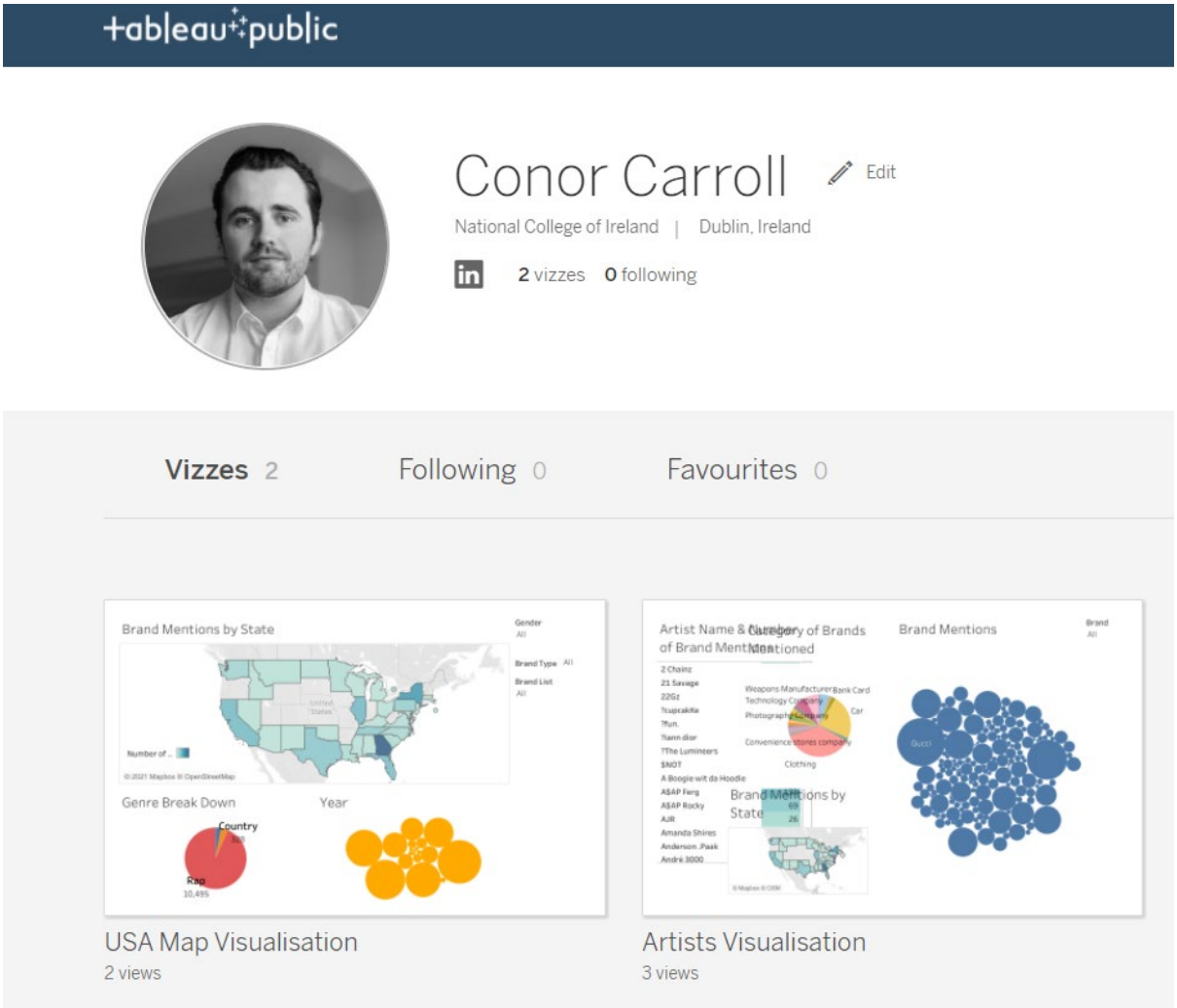


Fig.26 Tableau Public Profile

The visualisations are hosted on my Tableau Public profile and can be found at the following link:

https://public.tableau.com/profile/conor.carroll1079#!/

4.3. Neural Network

For the creation of our text generation machine learning script, we will be using Python with the TensorFlow package and creating a Recurrent Neural Network (RNN). The reason why we choose a RNN is because it recognises the sequence in which the letters are input. In language, the sequence of our words in a sentence is of the upmost importance hence why a RNN is appropriate for our task at hand (Karpathy, 2015). For my project I will be using the Python notebook created by TensorFlow and tweaking it to create music lyrics in the style of rap, pop and country.

The first thing we must do in our Python script is import TensorFlow and our dataset which is a txt file containing the lyrics.

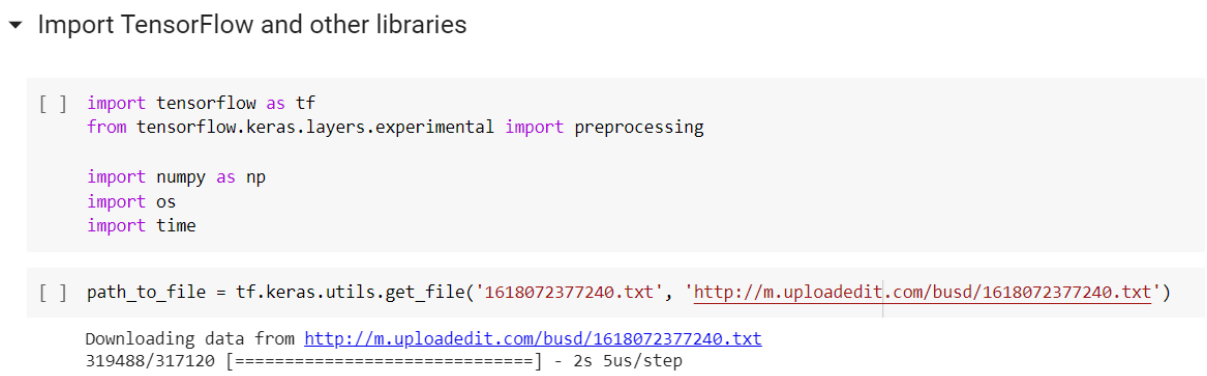


Fig.27 Importing TensorFlow and downloading our dataset

The next step is to process the text, we begin by vectorizing the text. The text is split into tokens and the characters are converted to numeric ID. Once we have the text processed, we need to divide the text into sample sequences for the model to train on, We do this by converting the text vector into a stream of character indices.

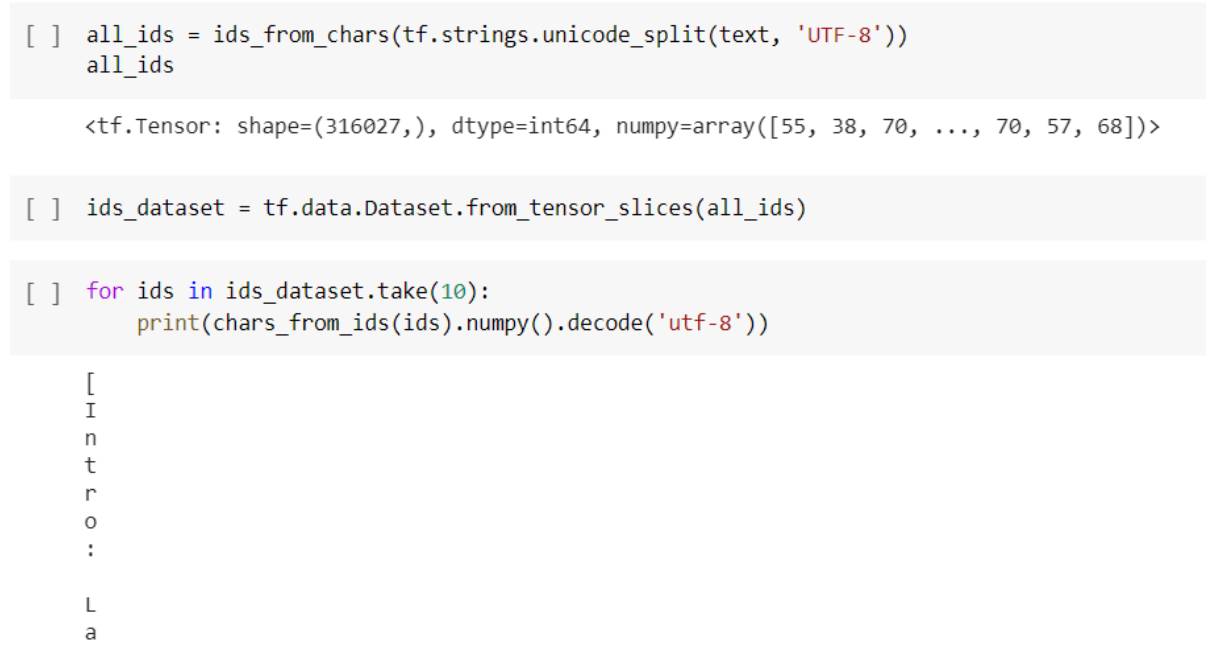


Fig.28 Text being converted to a stream for model to train on

The next step is to create a training dataset of input and label. Input and label are sequences, the model will look at the original input and learn what the next characters are going to be based on the label.

```
[ ] def split_input_target(sequence):
        input_text = sequence[:-1]
        target_text = sequence[1:]
        return input_text, target_text
```

```
[ ] split_input_target(list("Tensorflow"))
```

```
(['T', 'e', 'n', 's', 'o', 'r', 'f', 'l', 'o'],
 ['e', 'n', 's', 'o', 'r', 'f', 'l', 'o', 'w'])
```

```
[ ] dataset = sequences.map(split_input_target)
```

```
[ ] for input_example, target_example in dataset.take(1):
        print("Input :", text_from_ids(input_example).numpy())
        print("Target:", text_from_ids(target_example).numpy())
```

```
Input : b'[Intro: Lady Gaga & R. Kelly ] Yeah (Oh) Turn the mic up (Yeah, yeah) Eh eh eh eh eh eh eh (Oh) Eh e'
Target: b'Intro: Lady Gaga & R. Kelly ] Yeah (Oh) Turn the mic up (Yeah, yeah) Eh eh eh eh eh eh eh (Oh) Eh eh'
```


```
[ ] def split_input_target(sequence):
        input_text = sequence[:-1]
        target_text = sequence[1:]
        return input_text, target_text
```

```
[ ] split_input_target(list("Tensorflow"))
```

```
(['T', 'e', 'n', 's', 'o', 'r', 'f', 'l', 'o'],
 ['e', 'n', 's', 'o', 'r', 'f', 'l', 'o', 'w'])
```

```
[ ] dataset = sequences.map(split_input_target)
```

```
[ ] for input_example, target_example in dataset.take(1):
        print("Input :", text_from_ids(input_example).numpy())
        print("Target:", text_from_ids(target_example).numpy())
```

```
Input : b'[Intro: Lady Gaga & R. Kelly ] Yeah (Oh) Turn the mic up (Yeah, yeah) Eh eh eh eh eh eh eh (Oh) Eh e'
Target: b'Intro: Lady Gaga & R. Kelly ] Yeah (Oh) Turn the mic up (Yeah, yeah) Eh eh eh eh eh eh eh (Oh) Eh eh'
```


Fig.29 Text being converted to a stream for model to train on

After this we split our data into manageable sequences, we also must rearrange the data and format it into batches.

```
[ ] # Batch size
    BATCH_SIZE = 64

    # Buffer size to shuffle the dataset
    # (TF data is designed to work with possibly infinite sequences,
    # so it doesn't attempt to shuffle the entire sequence in memory. Instead,
    # it maintains a buffer in which it shuffles elements).
    BUFFER_SIZE = 10000

    dataset = (
        dataset
        .shuffle(BUFFER_SIZE)
        .batch(BATCH_SIZE, drop_remainder=True)
        .prefetch(tf.data.experimental.AUTOTUNE))

    dataset
```

```
<PrefetchDataset shapes: ((64, 100), (64, 100)), types: (tf.int64, tf.int64)>
```

Fig.30 Text converted to trainable batches

Once we have our text processed, we then need to move onto building the model. There are three layers to the model, the input layer, an RNN layer and the output layer.

```
[ ]  # Length of the vocabulary in chars
     vocab_size = len(vocab)

     # The embedding dimension
     embedding_dim = 256

     # Number of RNN units
     rnn_units = 1024
```

```
[ ]  class MyModel(tf.keras.Model):
       def __init__(self, vocab_size, embedding_dim, rnn_units):
         super().__init__(self)
         self.embedding = tf.keras.layers.Embedding(vocab_size, embedding_dim)
         self.gru = tf.keras.layers.GRU(rnn_units,
                                        return_sequences=True,
                                        return_state=True)
         self.dense = tf.keras.layers.Dense(vocab_size)

       def call(self, inputs, states=None, return_state=False, training=False):
         x = inputs
         x = self.embedding(x, training=training)
         if states is None:
           states = self.gru.get_initial_state(x)
         x, states = self.gru(x, initial_state=states, training=training)
         x = self.dense(x, training=training)

         if return_state:
           return x, states
         else:
           return x
```

```
[ ]  model = MyModel(
         # Be sure the vocabulary size matches the `StringLookup` layers.
         vocab_size=len(ids_from_chars.get_vocabulary()),
         embedding_dim=embedding_dim,
         rnn_units=rnn_units)
```

Fig.31 Three layers of our model

The next step is to attach an optimizer and loss function. This will allow our model to attempt to minimise loss and increase accuracy of our output.

```
[ ]  loss = tf.losses.SparseCategoricalCrossentropy(from_logits=True)
```

```
[ ]  example_batch_loss = loss(target_example_batch, example_batch_predictions)
     mean_loss = example_batch_loss.numpy().mean()
     print("Prediction shape: ", example_batch_predictions.shape, " # (batch_size, sequence_length, vocab_size)")
     print("Mean loss:        ", mean_loss)

     Prediction shape:  (64, 100, 99)  # (batch_size, sequence_length, vocab_size)
     Mean loss:         4.596059
```

Fig.32 Adding loss function

We then have to specify the number of epochs to train the model. I will be exploring the optimum amount for each genre in the conclusion section. Once we specify the epoch number, we allow the model to run through and train on the data.

```
[ ]  EPOCHS = 60
```

```
[ ]  history = model.fit(dataset, epochs=EPOCHS, callbacks=[checkpoint_callback])

     Epoch 1/60
     63/63 [==============================] - 5s 53ms/step - loss: 3.9481
     Epoch 2/60
     63/63 [==============================] - 4s 52ms/step - loss: 2.5394
     Epoch 3/60
     63/63 [==============================] - 4s 53ms/step - loss: 2.2177
     Epoch 4/60
     63/63 [==============================] - 4s 53ms/step - loss: 2.0174
     Epoch 5/60
     63/63 [==============================] - 4s 52ms/step - loss: 1.8382
     Epoch 6/60
     63/63 [==============================] - 4s 53ms/step - loss: 1.7227
     Epoch 7/60
     63/63 [==============================] - 4s 53ms/step - loss: 1.6202
     Epoch 8/60
     63/63 [==============================] - 4s 53ms/step - loss: 1.5155
     Epoch 9/60
     63/63 [==============================] - 4s 53ms/step - loss: 1.4298
     Epoch 10/60
```

Fig.33 EPOCH training

Once we have completed the preceding step, our model is trained on our input data. We are ready to generate our lyrics, to do this we run the model in a loop. Each time the model is called we pass in text and check the internal state. The model returns a prediction for the next character and continues on doing this, generating a batch of text.

```
start = time.time()
states = None
next_char = tf.constant(['Chorus:'])
result = [next_char]

for n in range(1000):
  next_char, states = one_step_model.generate_one_step(next_char, states=states)
  result.append(next_char)

result = tf.strings.join(result)
end = time.time()
print(result[0].numpy().decode('utf-8'), '\n\n' + '_'*80)
print('\nRun time:', end - start)
```

```
Chorus: Lil Keed] Hop in the coupe, hop in the coupe I'm going fast I'm rockin' these shows, not

________________________________________________________________________________

Run time: 2.010453224182129
```

Fig.34 Model outputting text

The process is repeatable for each genre, the only thing we must change is the input txt file to have the lyrics for the corresponding genre. Each run of the above cell will produce a new block of text in the style of the genre.

5.0 Analysis

Correlation Test of Brand Mentions vs Google Trends

I will be conducting an analysis on the final dataset to see if there is a correlation between the frequency of brand mentions in a state and the popularity on Google based off Google trends data. In order to do this, I calculated the top 10 most mentioned brands grouped by state and year, which left me with the following dataset.

A	B	C	D
Year	State	Brand	Count
2013	Georgia	Versace	274
2018	Florida	Gucci	219
2017	Florida	Gucci	75
2020	Texas	Glock Ges.	68
2019	Texas	Glock Ges.	58
2017	South Carc	Kodak	58
2016	California	Gucci	50
2017	Illinois	Wockhard	50
2018	New York	Bentley	49
2020	Illinois	Glock Ges.	47
2018	Georgia	Gucci	46
2018	Georgia	Chanel	43

Fig.35 Top 10 most mentioned brands by state/year

I then needed to get the related data for search queries from Google Trends. Google Trends allows users to search by year/subregion for popularity of a search query.

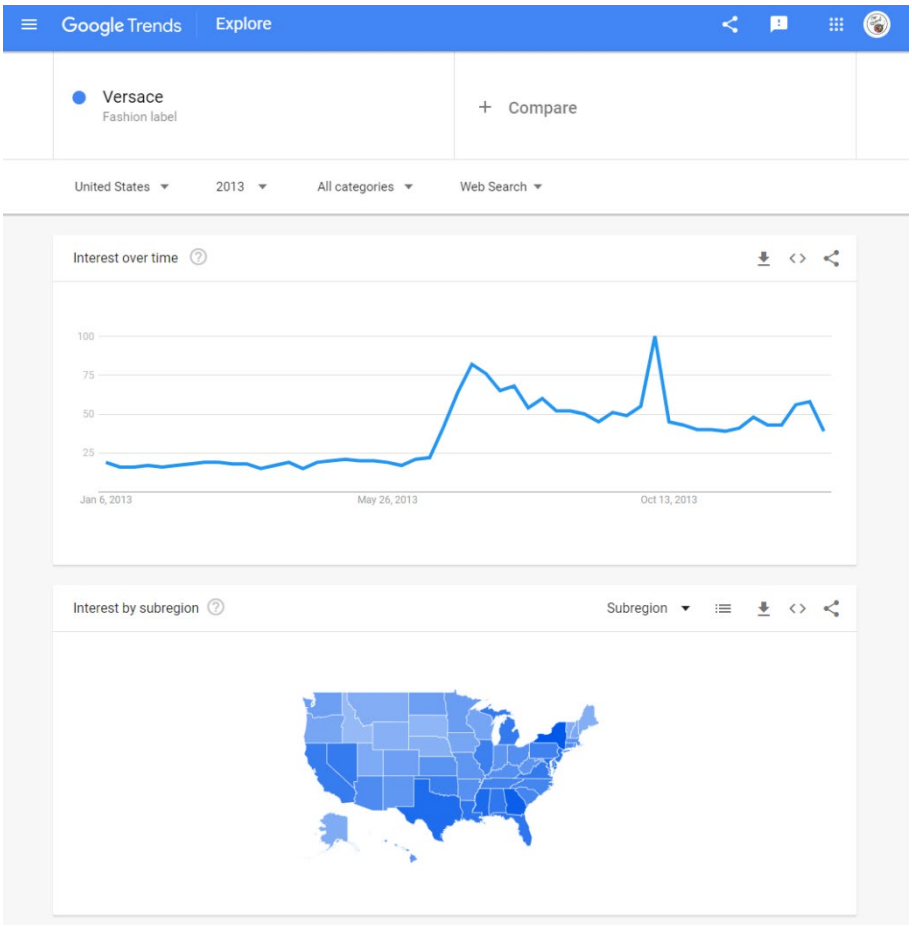


Fig.36 Search popularity for Versace for the year 2013

You can then download a csv file which ranks popularity for search query on a scale from 0-100. Here is the schema of the csv file.

A	B
Category:	All categories
Region	Versace: (201
New York	100
Mississippi	83
Delaware	81
Florida	81
Texas	80
New Jerse	79
Georgia	78
Maryland	76
South Carc	75
Virginia	75
Nevada	73
California	70
District of	69
Louisiana	69
Michigan	68
Illinois	66
Pennsylvar	62
Alabama	61
Oklahoma	60

Fig.37 csv file schema from Google Trends

We then Join this with our original dataset and get a column for the search popularity. Once we have done this, we are left with the following dataset which is ready to be analysed.

Year	State	Brand	Count	Trends Data
2013	Georgia	Versace	274	134
2018	Florida	Gucci	219	115
2017	Florida	Gucci	75	114
2020	Texas	Glock Ges.	68	98
2019	Texas	Glock Ges.	58	88
2017	South Carc	Kodak	58	117
2016	California	Gucci	50	118
2017	Illinois	Wockhard	50	95
2018	New York	Bentley	49	113
2020	Illinois	Glock Ges.	47	64
2018	Georgia	Gucci	46	127
2018	Georgia	Chanel	43	89

Fig.38 Dataset ready to be analysed

In order to test for correlation, I will be performing a Pearson's Correlation Coefficient (Pearson's r) test within R studio. This is a measure of linear correlation between two variables. To determine the strength of the relationship between the two variables we will see if a statistically significant correlation can be observed.

Pearson's Correlation Coefficient formula:

$$r = \frac{\sum_{i=1}^{n} \left(\left(x_i - \bar{x} \right) \left(y_i - \bar{y} \right) \right)}{\sqrt{\sum_{i=1}^{n} \left(x_i - \bar{x} \right)^2 \sum_{i=1}^{n} \left(y_i - \bar{y} \right)^2}}$$

Fig.39 Pearson's Correlation Coefficient

In figure 39 is the formula which is used to measure for Pearson's r where:

r = Correlation coefficient.

N = The number of observations which occur.

∑xy = Sum of products of paired scores.

∑x = Sum of x score.

∑y = Sum of y score.

∑x2= Sum of squared x scores.

∑y2= Sum of squared y scores.

Comparison of Brand Mentions by Genre

Another piece of analysis that will be conducted will be a test to see if there is a significant difference between the amount of brand references by each genre. For this we will be performing a Kruskal-Wallis H Test in r programming language. Kruskal-Wallis test is also known as Non-Parametric Anova, as it an alternative to One-way Anova for non-parametric datasets.

In order to do this, we will be taking a sample of 40 artists from each genre and summing their total brand mentions across the 40 songs from each artist. We are then left with a dataset with the following schema:

A	B	C
Pop	Rap	Country
7	257	1
1	53	4
26	115	15
3	22	15
12	17	5
24	180	4
28	36	7
7	429	1
7	109	5
10	100	1
5	95	1
7	100	5
3	204	10
9	101	6
8	34	2
3	75	33
9	207	2
10	69	1
22	54	2

Fig.40 Sample of our dataset ready for Kruskal-Wallis H test

The first thing we need to do before performing our Kruskal-Wallis test is to check each variable to see if it is non-normally distributed. We start by visualising the distribution with a QQ Plot.

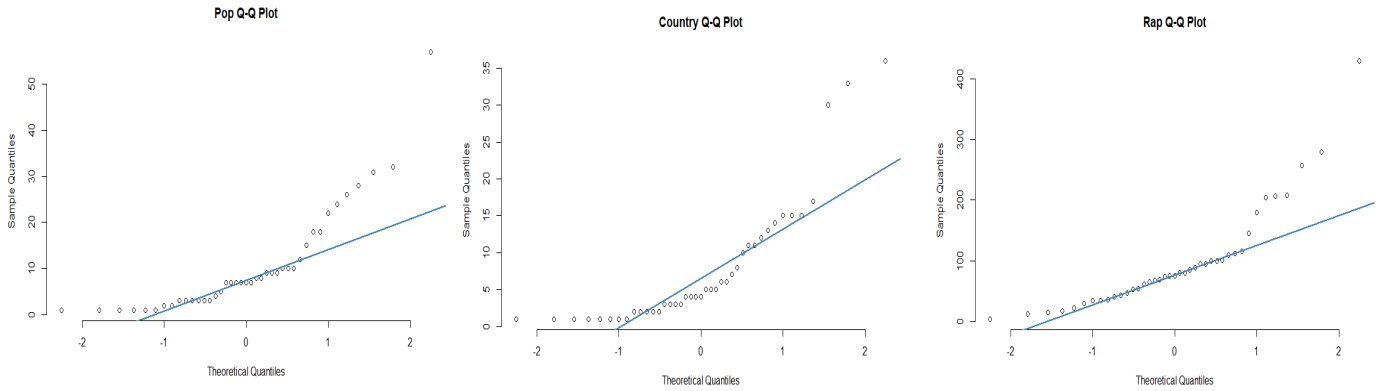


Fig.41 Q-Q Plot for all 3 variables

Once we have confirmed our dataset is non-normally distributed, we can perform our test with the following formula.

$$H = \left[\frac{12}{n(n+1)} \sum_{j=1}^{c} \frac{T_j^2}{n_j} \right] - 3(n+1)$$

Fig.35 Kruskal-Wallis H test formula

Where:

n = sum of sample sizes for all samples

c = number of samples

Tj = sum of ranks in the j^{th} sample

nj = size of the j^{th} sample

Male vs Female Brand Mentions in Rap

I will be comparing if there is a statistical difference between the amount of brand references for male and female artists in the rap genre. In a research Article titled "Hegemonic Masculinity in Hip-Hop Music? Difference in Brand Mention in Rap Music Based on the Rapper's Gender" (Mohammed-baksh and Callison, 2015), they conclude that male artists had significantly higher amount of brand references within their lyrics when compared to their female counterparts. They tested this on a dataset of 200 songs. In my analysis I will be taking 10 female rappers and 10 male rappers with 40 songs each, a total of 800 songs. I will perform a T-test in order to determine if there is a significant difference between the number of mentions between the two groups. Again, we must check for normality between the two groups. When performing a Shapiro-Wilks test for normality in r, we get the following results.

```
> shapiro.test(Data$Male)

        Shapiro-Wilk normality test

data:  Data$Male
W = 0.95636, p-value = 0.7437

> shapiro.test(Data$Female)

        Shapiro-Wilk normality test

data:  Data$Female
W = 0.96194, p-value = 0.8077
```

Fig.42 Results of normality test in r

As our P-value is above 0.05 we can accept that our data is normally distributed, we can continue with our T-test. The formula for the T-test is as follows.

$$t = \frac{(\bar{X}_1 - \bar{X}_2) - (\mu_1 - \mu_2)}{\sqrt{\left(\frac{s_1^2}{n_1} + \frac{s_2^2}{n_2}\right)}}$$

Fig.43 T-test formula

Where:

$\boldsymbol{X\ 1}$ is the mean for sample Group 1

$\boldsymbol{X2}$ is the mean for sample Group 2

$\boldsymbol{\mu 1}$ is the mean for Population 1

$\boldsymbol{\mu 2}$ is the mean for Population 2

$\boldsymbol{s1\text{^}2}$ is the variance of Group 1

$\boldsymbol{s2\text{^}2}$ is the variance of Group 2

$\boldsymbol{n1}$ is the number of participants in Group 1

$\boldsymbol{n2}$ is the number of participants in Group 2

6.0 Results

Correlation Test of Brand Mentions vs Google Trends

For our correlation analysis we first must set out the null and alternate hypothesis.

H0: There is not a correlation between brand mentions in lyrics and Google Trends data

H1: There is a correlation between brand mentions in lyrics and Google Trends data

We are calculating Pearson's r value, so the null and alternate hypothesis is.

H0: r = 0

H1: r =/= 0

When we run the test in R-studio, we get the following output:

```
> View(Correlation)
> cor(Correlation$Count, Correlation$`Trends Data`)
[1] 0.4821791
```

Fig.44 T-test formula

This gives us an output of an r value of 0.48, this would imply a moderate correlation between the two variables. As the r value is above 0, we can reject the null hypothesis and accept the alternate hypothesis as true.

We can visualise the results on a scatterplot within R-studio, which gives us the following output visual.

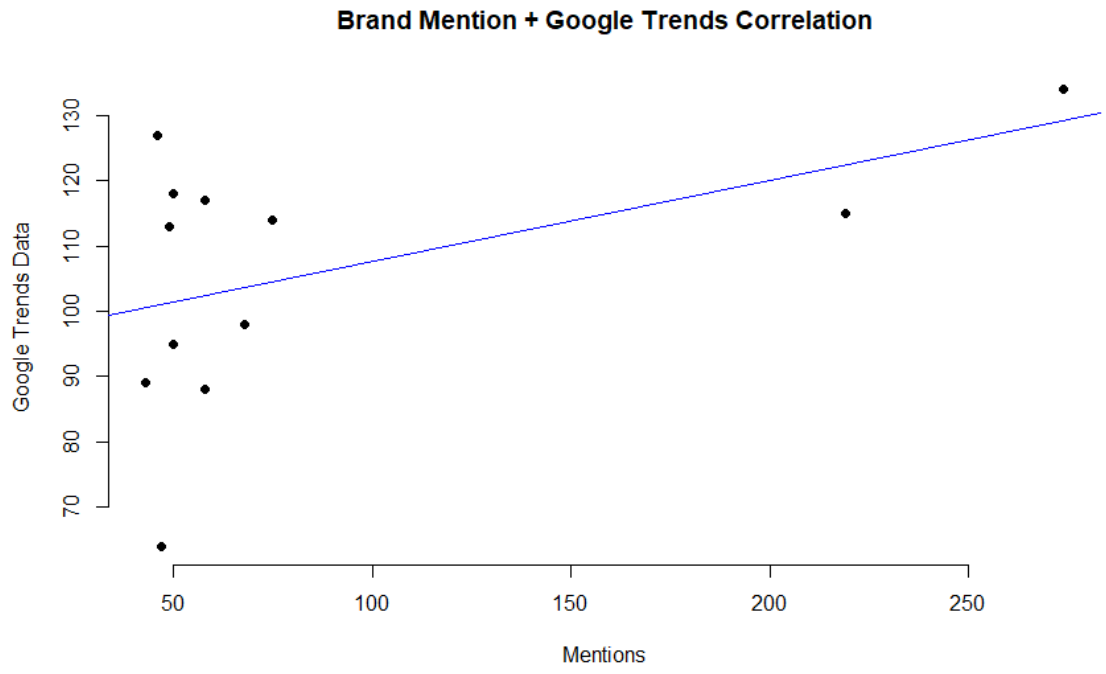


Fig.45 Scatterplot showing correlation between the two variables

Comparison of Brand Mentions by Genre

For our Genre comparison analysis, we will set out the null and alternate hypothesis:

H0: There is no difference in amount of brand mentions in lyrics of rap, pop and country

H1: There is a difference in amount of brand mentions in lyrics of rap, pop and country

The null hypothesis is expecting the mean value of each genre to be the same.

H0: μ Rap = μ Pop = μ Country

H1: μ Rap =/= μ Pop =/= μ Country

When we perform our Kruskal-Wallis H test, we will be looking for the p-value to see if the difference in means is statistically significant. Anything below p-value of 0.05 is considered statistically significant. Therefore, our null and alternate hypothesis for p value

H0: P > 0.05

H1: P < 0.05

The output from our Kruskal-Wallis H test in R-studio is:

```
        Kruskal-Wallis rank sum test

data:  Genre_Comparison
Kruskal-Wallis chi-squared = 70.963, df
= 2, p-value = 3.895e-16
```

Fig.46 Output of our Kruskal-Wallis test in R

Our P-value when written as a real number is 0.0000000000000003895. Therefore, we can reject the null hypothesis and accept the alternate hypothesis that there is a significant difference in the means of each group.

We can get some insight into the difference in each group by using some visualisations

Boxplot of three groups

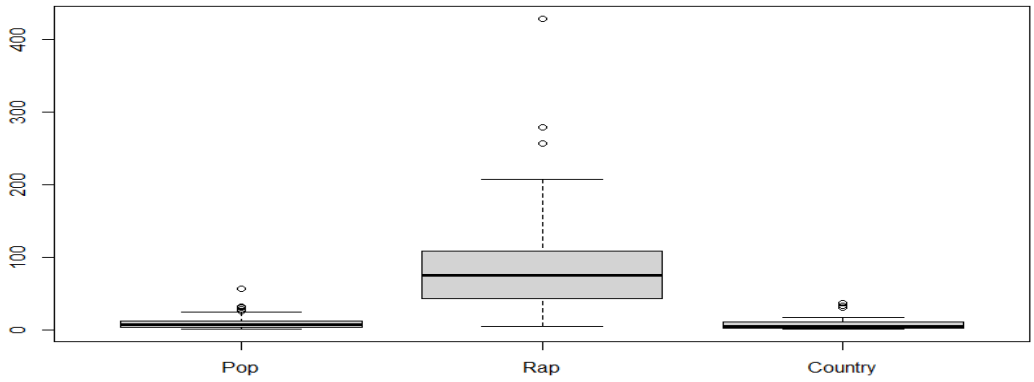


Fig.47 Box Plot of the three genres

We can see the pop and country have relatively similar amounts of brand mentions; however, rap has a significantly higher amount of brand references within its lyrics.

Histogram for each group

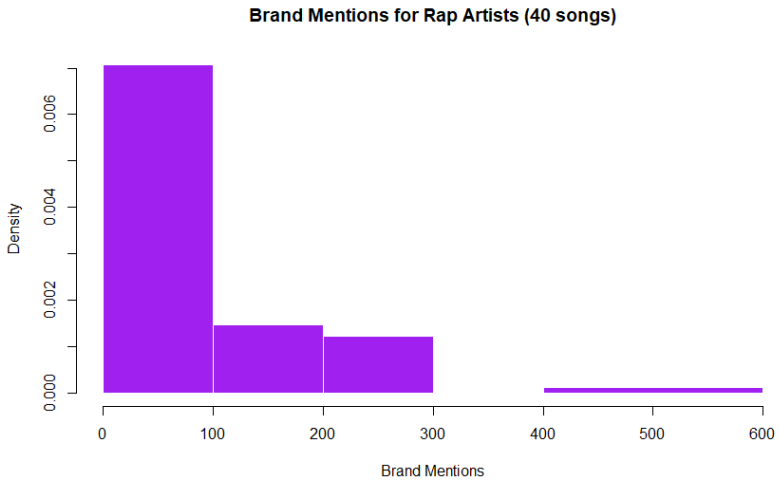


Fig.48 Rap histogram

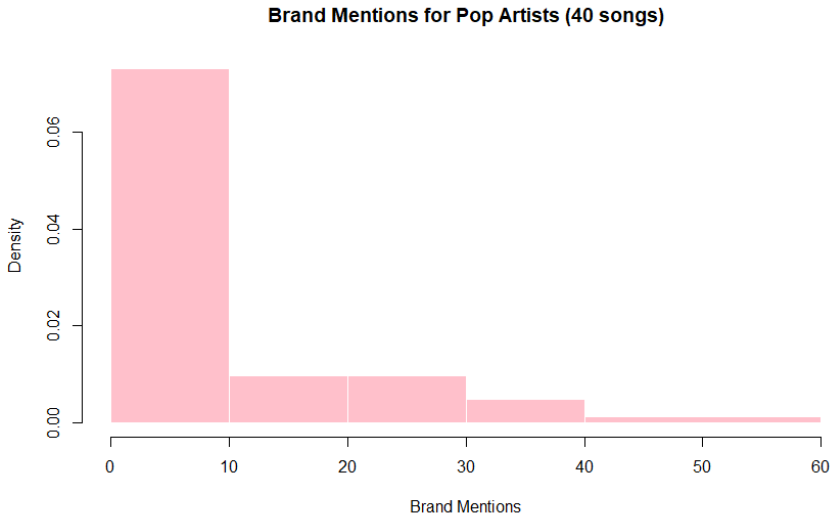


Fig.49 Pop histogram

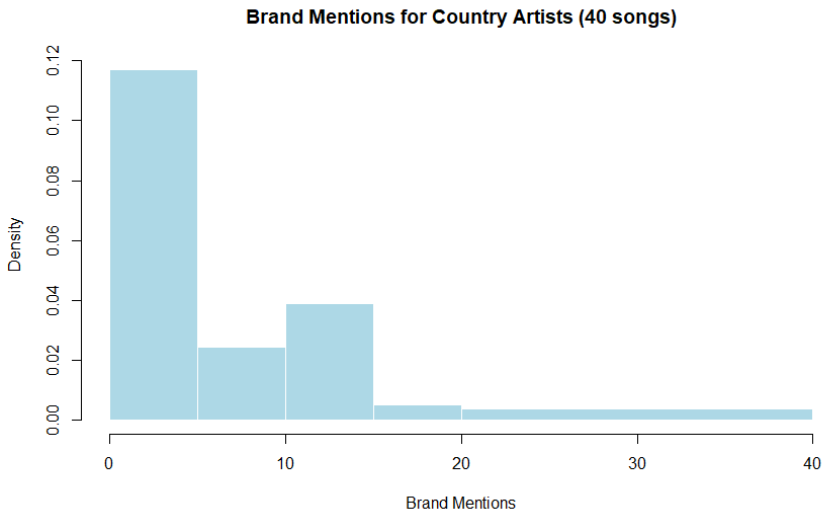


Fig.50 Country histogram

Pie chart for category of brands referenced for each genre

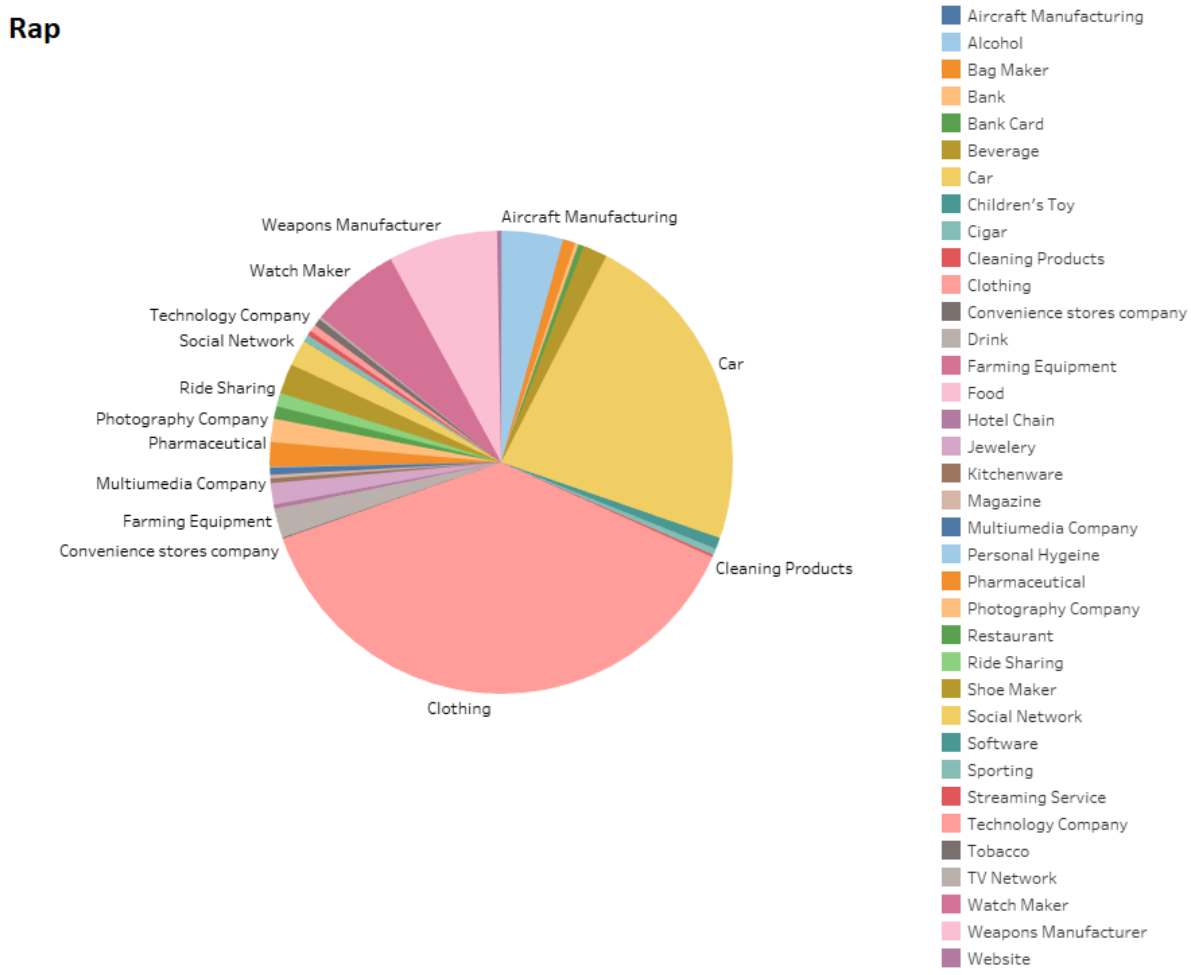


Fig.51 Rap pie chart showing different brand categories mentioned

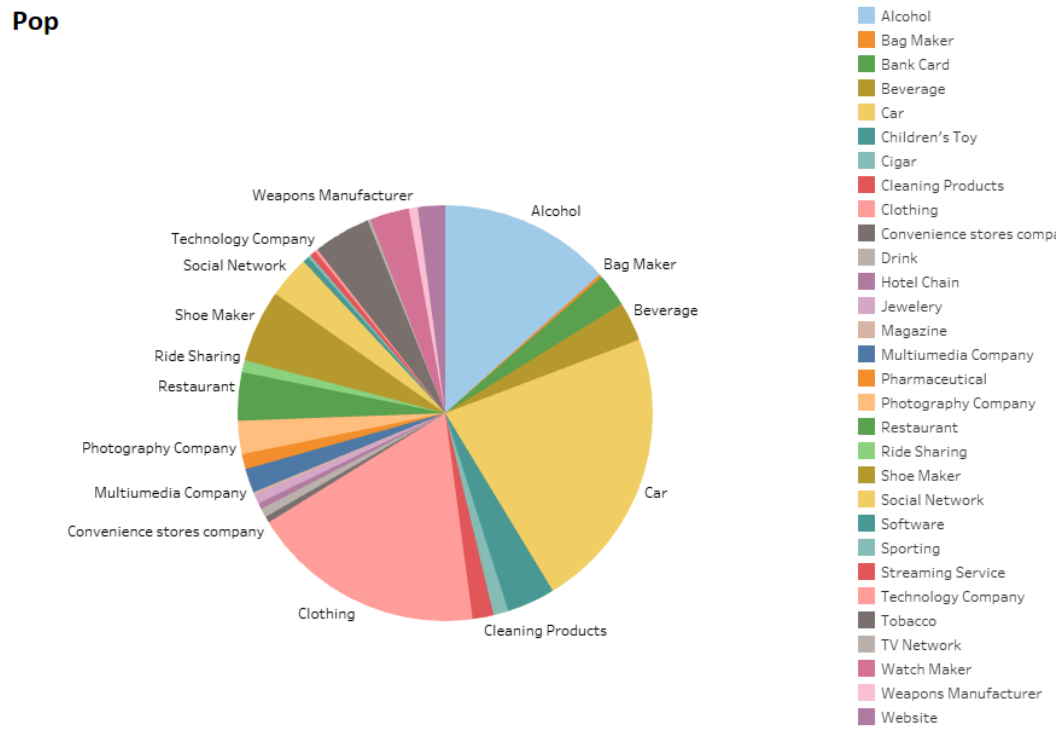


Fig.52 Pop pie chart showing different brand categories mentioned

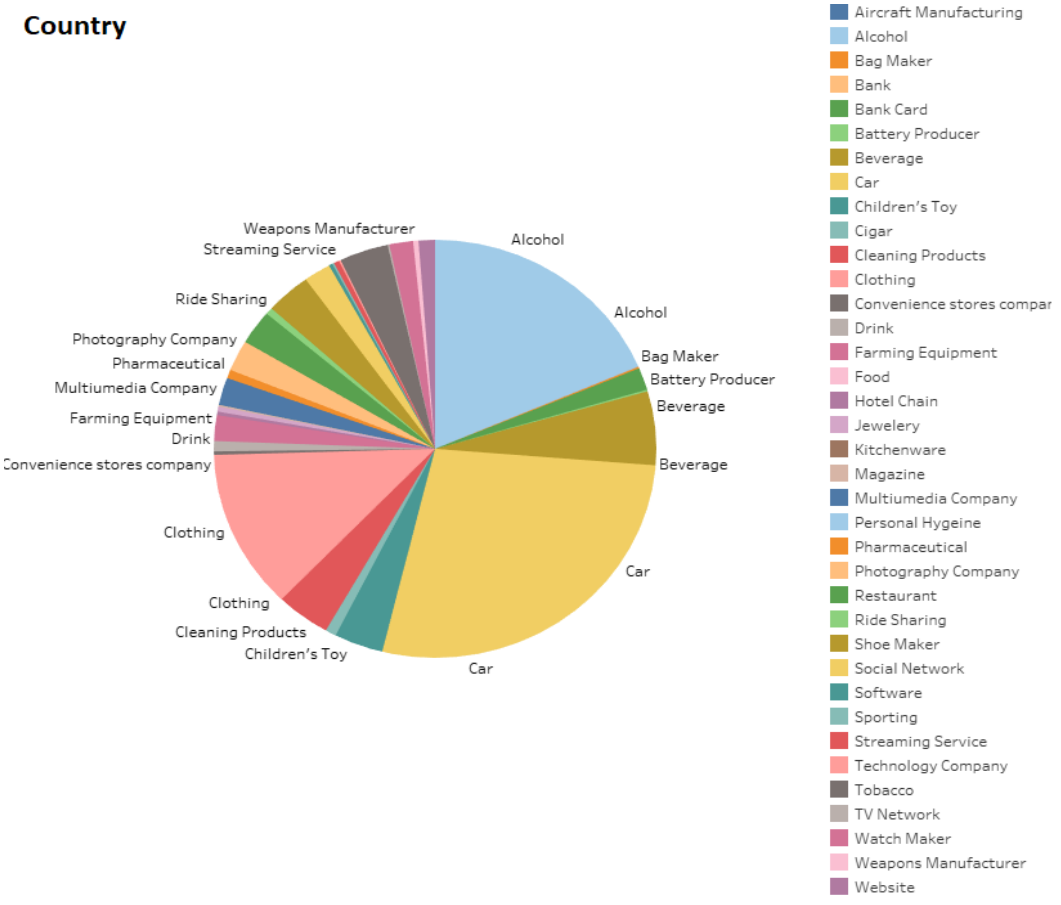


Fig.53 Country pie chart showing different brand categories mentioned

Male vs Female Brand Mentions in Rap

For our gender comparison analysis, we will set out the null and alternate hypothesis:

H0: There is no difference in amount of brand mentions in the lyrics of male and female artists

H1: There is a difference in amount of brand mentions in the lyrics of male and female artists

The null hypothesis is expecting the mean value of each gender to be the same.

H0: μ Male = μ Female

H1: μ Male =/= μ Female

When we perform the t-test, we will be looking for the p-value to see if the difference in means is statistically significant. Anything below p-value of 0.05 is considered statistically significant. Therefore, our null and alternate hypothesis for p value

H0: P > 0.05

H1: P < 0.05

When we run our t-test in R-studio, we get the following output.

```
> t.test(Data$Male, Data$Female)

        Welch Two Sample t-test

data:  Data$Male and Data$Female
t = 1.8002, df = 15.255, p-value = 0.09164
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 -7.036722 84.236722
sample estimates:
mean of x mean of y
    106.5      67.9
```

Fig.54 R-studio output for t-test

As our p-value is above 0.05 we can accept the null hypothesis as true.

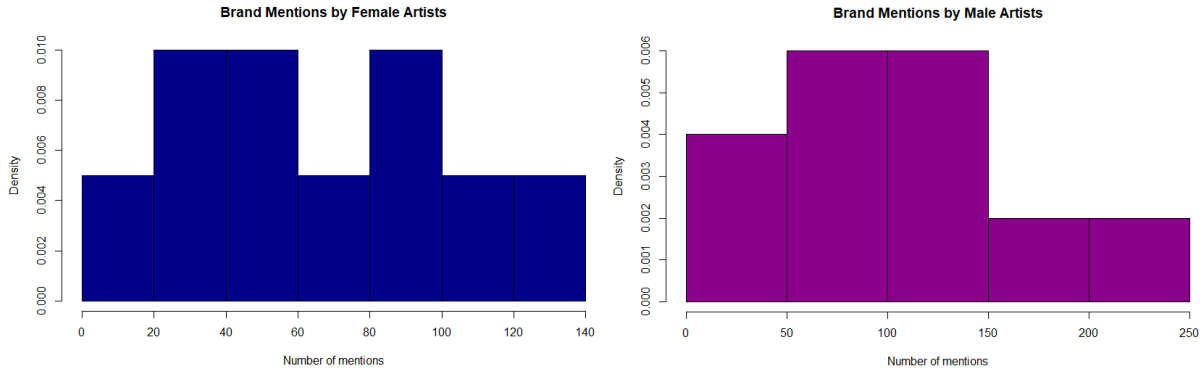


Fig.55 Histogram for female and male artists

7.0 Conclusions

7.1. Problem Addressed

The main problem I set out to address at the start of this project was to create a tool which can be used to quickly identify brand mentions within music lyrics. I achieved this by downloading data from the Genius API, the data is then transformed using Alteryx workflow and checked for brand mentions. It is then loaded onto a Microsoft SQL server.

I also set out to answer several research questions with our dataset, which I achieved by using our final dataset and running statistical analyses on it using r programming language.

The last problem I addressed was to get a machine learning model to generate music lyrics in the style of the genres analysed in our data. This was done by using a Recurrent Neural Network in Python programming language

7.2. Testing

Brand mentions

When testing the results of the final dataset, I manually went through the data to find where the matches were found, and I noticed some false positives. There were certain brands which were causing issues with the matches, mainly brands which did not have a unique word that could be looked for within the lyrics. Some examples are Lucky Strike (cigarettes), Chrome Hearts (clothing), Southern Comfort (Alcohol) among others. As the model only checks one cell with one word at a time the issue was looking for “Hearts” for chrome hearts would show a match when an artist simply was using the word heart in their lyrics.

In order to resolve this, I had to create a macro within Alteryx to alleviate the false positives. The macro will check if a cell contains one of the words from the brands giving false positives, lets for our example take Chrome Hearts. If the macro finds a cell with the word “Hearts”, it will check the previous cell and see if it says the word “Chrome”. If it finds the two words beside each other, it will merge them to one cell containing both words. This allowed us to search for two words and got rid of a large issue with false positives.

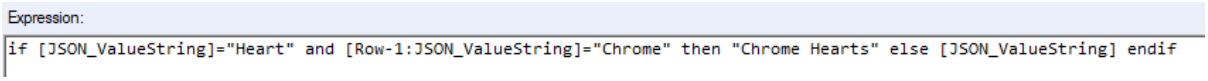


Fig.56 Expression to check for Chrome Hearts

We then repeated this process for all other brands with the same issue.

Lyric Generation

For our lyric generator, the Neural Network measures its accuracy by measuring the loss of the predicted text. The lower the loss, the more similar the text is to the input dataset. What determines the loss is the amount of training that the number of epochs we set the model to go through. An epoch is one cycle of full training through the dataset. There was some trial and error with each model for each genre, in finding the optimum number of

epochs to minimise loss. In the end this was the optimal number of epochs for each genre's dataset.

Table 1 Optimum number of epochs

Genre	Epochs	Loss
Rap	60	0.0673
Country	59	0.0694
Pop	15	0.0631

7.3. End Products

Brand mentions

The final product from this part of the project is a relational database which has analysed the lyrics of over 8000 songs. It contains all the brand mentions within our list of brands and also contains the gender, genre and home state of the artist. As well as this, there is a workflow which allows users to add many more artists and lyrics to the database with the click of a few buttons and some changing of code.

The results of the initial dataset analysis are also visualised on a Tableau dashboard which can again be found here:

https://public.tableau.com/profile/conor.carroll1079#!/

Lyric Generation

For this section of the project, there is a Python script which accepts a txt file with lyrics and will recreate its own lyrics in the style of the genre. I have used a text to speech tool to perform the lyrics over a backing track to each genre. It can be found here:

https://youtu.be/mEDt829mq40

7.4. Strengths and Limitations

Strengths

A huge strength of my project is that it integrates JSON files from the Genius API. Genius.com host the data for almost every single musical artist of note meaning that with more time, the number of songs/artists lyrics that can be analysed is almost infinite. The model itself will not require any extra tweaking to parameters or any changes to workflow etc. it is ready to work with the JSON files as is.

The Alteryx workflow also uploads directly to the Microsoft SQL server, which the Tableau dashboard draws its data from for the visualisation. This means that if more artists/songs were to be analysed, they would automatically update the Tableau dashboard and be visualised instantly upon running.

Another strength is that it could be tweaked to check for anything, not only brand mentions. For example, if someone wanted to conduct a research on the amount of curse words used within music lyrics, it would require changing out the brand mention dataset for a list of curse words to search for. The workflow is highly adaptable and can have many uses beyond the scope of analysing brand mentions.

Limitations

The main limitation of my project is that the JSON files must be manually ran through the Alteryx workflow one at a time. Running multiple files that are accepted in the normal input tool in Alteryx which accepts csv, txt, Excel etc. is easily done. However, due to the nature of the JSON files they have to be read in as blobs and the usual method of running multiple files in the one folder location does not work. I tried for many hours and did a lot of research about ways to run multiple files but did not have any success.

Another limitation is with the list of brand names. I could not find a resource which contained lists of brand nicknames that was suitable for the project. I had to manually add them as best I could. In the end the model identified over 11,000 brand mentions in 8,000 songs which was a good return, but I feel more could have been identified with a more complete list of brand nicknames.

7.5. Key Findings

Google Trends and brand mentions are moderately correlated

After conducting our Pearson's correlation test, we received an r value of 0.48 which can be deemed a moderate positive correlation.

Strength of Association	Coefficient, *r*	
	Positive	Negative
Small	.1 to .3	-0.1 to -0.3
Medium	.3 to .5	-0.3 to -0.5
Large	.5 to 1.0	-0.5 to -1.0

Fig.57 Pearson's coefficient chart

The positive relationship means that as brand mention frequency rises, so do Google Trend figures. However, it is only a moderate correlation, and our scatterplot does not show much correlation between the two.

Rap Music has significantly more mentions of brands than pop and country

From running our Kruskal-Wallis H test, we got the result that the mean mentions of brands from 40 artists across 3 genres was significantly different. When we plot our data on a box plot in figure 47 it is evident that rap music has a much higher mean than the other two genres. It also contained the most outliers.

Male and female rap artists mention the same number of brands on average

The results of our t-test showed that the difference between the mean of a sample of 10 female and male rap artists was not statistically significant. The histograms of both variables were fairly similar, and we had a p-value of 0.092. This contradicts the findings put forward by Mohammed-baksh, S. and Callison, C in their 2015 report. They claimed that men are more likely to mentioned brands within their lyrics than women. Their dataset was ¼ the size of the dataset I have used, so I would argue that my analysis gives a better representation.

8.0 Further Development or Research

8.1. More artists/genres

With additional time on the project, I would like to include more artists to analyse the lyrics of. It is quite time consuming to input all the names into the Python script and run the files through the Alteryx workflow. My project currently analyses 8,000+ songs for 200+ artists across 3 genres, however the analyses tool is made and the "hard part" of the project is done. With additional time it could be used to analyse many more genres, artists, songs as all the data that is required is available on the Genius API it just needs to be extracted and ran through the rest of the process.

8.2. Bigger Brand List

With additional time and resources, I would create a bigger list of brands and include more brand nicknames. I know there are some matches being missed due to them being referenced as nicknames within the lyrics. I would like to find a way to add these to the list without having to do it manually. Given enough time however, I could add it manually although this would be time consuming.

8.3. Accept user input

Another route I would like my project to go down is to have a GUI which asks a user to enter an artist's name. The user would click run and the JSON file would be downloaded, analysed and visualised. I researched how to achieve this; it is possible to do it using Power Automate on the Microsoft apps. The Python script could email the JSON file after downloading it, Power Automate can be triggered by the email and run the file through Alteryx. Alteryx automatically uploads to the finished data to an SQL server, which Tableau is connected to so it is automatically visualised.

8.4. Animated tableau dashboard

One feature to the visualisation which I would like to achieve with additional time is to animate some of the visualisation on Tableau. This would just be to add some creativity to the visualisation and add some entertainment factor.

9.0 References

Sunset, B., 2008. OBEY YOUR THIRST CAMPAIGN (1998). [Blog] Marketing Campaign Case Studies, Available at: http://marketing-case-studies.blogspot.com/2008/05/obey-your-thirst-campaign-1998.html

Roberts, G., 2009. Rap stars lose taste for Cristal after owner's attack on 'bling'. The Independent, [online] Available at: https://www.independent.co.uk/arts-entertainment/music/news/rap-stars-lose-taste-cristal-after-owner-s-attack-bling-6098086.html

Hore-Thorburn, I., 2020. HERE'S HOW GUCCI MANE WENT FROM CEASE AND DESIST TO PARTNERSHIP WITH GUCCI. Highsnobiety, [online] Available at: https://www.highsnobiety.com/p/gucci-gucci-mane-partnership-interview

Kelly, C., 2020. Why brands are doubling down on hip-hop talent. [online] Marketing Dive. Available at: https://www.marketingdive.com/news/hip-hop-brand-campaigns-2020/574585

Fortune. 2017. This Is the Most Name-Dropped Brand in Music. [online] Available at: https://fortune.com/2017/08/18/name-brands-pop-music-rap

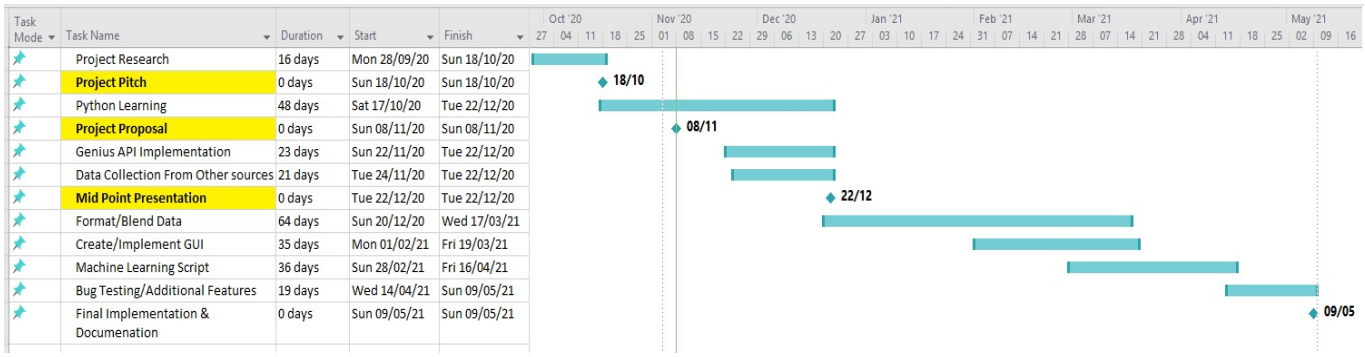
Craig, C., Flynn, M. and Holody, K., 2017. Name Dropping and Product Mentions: Branding in Popular Music Lyrics. Journal of Promotion Management, 23(2), pp.258-276.

Karpathy, A., 2015. The Unreasonable Effectiveness of Recurrent Neural Networks. [Blog] Andrej Karpathy blog, Available at: http://karpathy.github.io/2015/05/21/rnn-effectiveness

Mohammed-baksh, S. and Callison, C., 2015. Hegemonic Masculinity in Hip-Hop Music? Difference in Brand Mention in Rap Music Based on the Rapper's Gender. Journal of Promotion Management, 21(3), pp.351-370.

10.0 Appendices

10.1. Project Plan



Task Mode	Task Name	Duration	Start	Finish
	Project Research	16 days	Mon 28/09/20	Sun 18/10/20
	Project Pitch	0 days	Sun 18/10/20	Sun 18/10/20
	Python Learning	48 days	Sat 17/10/20	Tue 22/12/20
	Project Proposal	0 days	Sun 08/11/20	Sun 08/11/20
	Genius API Implementation	23 days	Sun 22/11/20	Tue 22/12/20
	Data Collection From Other sources	21 days	Tue 24/11/20	Tue 22/12/20
	Mid Point Presentation	0 days	Tue 22/12/20	Tue 22/12/20
	Format/Blend Data	64 days	Sun 20/12/20	Wed 17/03/21
	Create/Implement GUI	35 days	Mon 01/02/21	Fri 19/03/21
	Machine Learning Script	36 days	Sun 28/02/21	Fri 16/04/21
	Bug Testing/Additional Features	19 days	Wed 14/04/21	Sun 09/05/21
	Final Implementation & Documenation	0 days	Sun 09/05/21	Sun 09/05/21

Fig.58 Original Gantt chart for the project

10.2. Reflective Journals

Software Project Reflective Journal

X14535037 – Conor Carroll - October

This month has mainly been one for researching about my idea for the project and learning more about the technologies that I am planning to use; I am going to write this journal under each heading.

Research:

Originally when I came up with my idea, in my head I had the idea to use web scraping to get data from Genius.com, it was only when I started to research this that I found out they have a public API which grants users access to the metadata contained on the website. I have read through some documentation for the API as part of my research, the website contains lots of information on navigating the API which I am sure will come in handy.

I have also been researching to see if there was anything like my idea done before. In my research I came across a couple articles with infographic pictures regarding brands mentioned but nothing interactive and on the scale of my own project. I came across two YouTube videos which were close to the machine learning part of my own project. One video took a dataset of rap lyrics and then created its own, the other video used a text to speech tool to mimic the voice of musician Billie Eilish. From my research, I could not find anything which put the two aspects together.

Learning:

For the project I decided to use Python, this is because it seems to be popular on most job descriptions for roles, I am interested in. I have no prior experience with Python, so I am learning from scratch. I bought a Udemy course, and I am going through the videos, I am about 2 of 8 hours in and so far, I am finding it quite good and intuitive. As well as Python, I am going to use Alteryx for some data cleansing etc. I used Alteryx heavily during my work experience over summer, so I am fairly familiar with it. I downloaded a song lyrics data set from Kaggle.com, Then I used Alteryx to create a workflow to single out brands within the lyrics from certain artists. I did this just as a proof of concept and managed to do it easily enough.

Going forward, I am going to meet with my project supervisor to get some feedback from them and then create a plan with a timeline. This coming week I plan on playing around with the Genius API within a programme, just to get it functioning etc.

Software Project Reflective Journal

X14535037 – Conor Carroll

November

Proof of Concept

This month I created a very basic proof of concept to show what exactly it is I am trying to achieve with the data. I did some handson use with connecting to the Genius API using a Python script. The Python script downloaded a JSON file from the API which contained metadata about the artist and song lyrics. As it was only a proof of concept I chose one Artist, and 5 of his songs (which the script pulled at random).

Once the JSON file was downloaded, I used Alteryx to parse the file as it has a built in JSON parser tool which I had used before during my work placement. An issue I ran into is that the JSON parser truncates the cells with the lyrics, cutting most of it off, this will be something I will have to figure out/research to fix. After this I parsed the data by using a space as a delimiter and split the data as one line per word. This left me with a dataset of lyrics with one word per cell. It is then a case of linking it to another dataset which contains brand names and identifying the cells with the brand names. I took 3 brand names (again just as a proof of concept) and found them in the lyrics dataset. This Alteryx counts how many times the word appears in the dataset and the end result looks something like this:

Count	Actual Name
3	Bentley
2	Gucci
3	Xanax

With the brand name on the right and the amount of times it is mentioned on the left. This shows what it is that I am trying to achieve and one route on how it can be achieved.

Going Forward

My next focus now is on the Mid-point upload. I am going to flesh out the proof of concept and include more artists, brands etc. I also want to link this into an interactive map hosted in a web browser and have this ready for the mid-point upload

Software Project Reflective Journal

X14535037 – Conor Carroll

December

This month was mainly to get prepared for the midpoint presentation due on the 22nd. I had a brief proof of concept done up on how I was going to extract and search through the data via python and Alteryx done up from November. I had to expand on this, this meant broadening the list of designers I have and downloading more sets of lyrics from the Genius API.

In order to show a geographical breakdown of the data I used a responsive HTML page which contained the map of the USA, when a state was clicked it would show a Power BI dashboard breaking down brands mentioned. I want to build on this significantly, this was really only a proof of concept to show in my video presentation. It is quite a simplistic approach to it currently, I want it to be driven all within Python.

I found it hard to hit on everything I wanted to talk about within the 10-minute max video length for the midpoint. This project is very interesting to me and felt I could have filled a lot longer talking about each individual aspect. I especially wanted to elaborate more on where it was going to go in the future, but the limitations of video length did not allow me to do so.

Regarding going forward, I have the approach to the data exploration essentially completed. Now it is just a case of gathering more of the data and figuring out how to display it the best way possible. I will also have to get started on the machine learning part of the project.

Software Project Reflective Journal

X14535037 – Conor Carroll

January

This month I continued on downloading lyric data from the Genius API. I have decided that I am going to download the data for 10,000 songs. The breakdown of this is 40 songs from 250 artists, 100 from the rap genre, 50 from country genre, 50 from pop music and 50 from rock music. I should have enough data from this to analyse the difference between the genres in brand mentions.

In order to analyse geographically, I have downloaded Tableau and began learning the geographic data mapping tools with it. For the midpoint I used a HTML map which linked to a Power Bi dashboard, I feel that I could get a more seamless experience for the user upon using a Tableau dashboard. I also would like to gain more experience with Tableau as it features in a lot of job listings I have been looking at.

Another addition to my project is that I have set up a free MongoDB cluster which I will be linking my Alteryx workflow output to. After the workflow has done its cleaning and joining of data tables, the CSV file will be uploaded to the MongoDB cluster as I want to store it in the cloud as opposed to locally on my computer.

Going forward, I will need to continue on with my data gathering and processing. I will need to learn the Tableau tools for data reporting, and I will need to get my processed data uploaded to MongoDB so it is stored in the cloud.

Software Project Reflective Journal

X14535037 – Conor Carroll

February

Data Collection:

This month I completed the data collection for my project. I downloaded data from 200+ artists (over 80,000 song lyrics). Originally, I planned on doing 50 artists from the rock genre, but from my preliminary analysis, there was not enough brand mentions within the lyrics to make it worth my while.

As well as the artist data, I extended the list of brands it searches for greatly (from around 60 to 200+). I achieved this by finding different websites which hosted lists for example “50 most popular tobacco brands in the USA” and using R to web scrape the list.

At this point I have my dataset and the data mining technique done. It is a case now of looking for findings within the dataset. I have discussed a couple ideas with my project supervisor (comparing with google trends etc). I am also in the process of going through past projects in order to get some ideas for my own.

Macro Creation:

Within my Alteryx workflow, the lyric data is broken down to one word per cell. This made it difficult for it to check for brands which had two-word names (e.g “True Religion”). In order to get around this I had to create a macro which contains a tool called the “multi-row tool”. This allows the workflow to check the current cell and the one previous/ahead.

Machine Learning Research:

I have started the early stages of seeing what is involved regarding building the machine learning script for creation of song lyrics in genre styles. For this I will have to build a Recurrent Neural Network (RNN). This is the best machine learning for creating speech as with RNN’s the order of the data carries significant weight, much like the English language.

Software Project Reflective Journal

X14535037 – Conor Carroll

March

Finished Dataset:

After downloading all the relevant JSON files for each Genre, I ran each file through my Alteryx Workflow which checked for all brand mentions from my list along with other elements of data which left me with a dataset with the following schema:

	Count	Brand	Type	Artist	State	Gender	Year	Genre
2	1	Ford Motor Company	Car	Amanda Shires	Texas	Female	2018	Country
3	2	Coca-Cola	Beverage	Ashley McBryde	Tennessee	Female	2020	Country
4	1	General Motors	Car	Ashley McBryde	Tennessee	Female	2018	Country
5	1	Natural Ice	Alcohol	Ashley McBryde	Tennessee	Female	2020	Country
6	3	Cadillac	Car	Blake Shelton	Oklahoma	Male	2021	Country
7	3	Chevy	Car	Blake Shelton	Oklahoma	Male	2013	Country
8	1	Ford Motor Company	Car	Blake Shelton	Oklahoma	Male	2017	Country
9	1	John Deer	Farming Equipment	Blake Shelton	Oklahoma	Male	2016	Country
10	1	Keystone Light	Alcohol	Blake Shelton	Oklahoma	Male	2017	Country
11	1	Marlboro	Tobacco	Blake Shelton	Oklahoma	Male		Country
12	1	Marlboro	Tobacco	Blake Shelton	Oklahoma	Male	2013	Country
13	1	Mercedes-Benz	Car	Blake Shelton	Oklahoma	Male		Country
14	1	Nobu Restaurants	Restaurant	Blake Shelton	Oklahoma	Male	2017	Country
15	1	Visa	Bank Card	Blake Shelton	Oklahoma	Male	2017	Country
16	1	Wrangler	Clothing	Blake Shelton	Oklahoma	Male		Country
17	3	Barbie	Children's Toy	Brandy Clark	Washington	Female	2016	Country
18	1	Budweiser	Alcohol	Brandy Clark	Washington	Female	2016	Country
19	1	Cadillac	Car	Brandy Clark	Washington	Female	2012	Country
20	1	Cadillac	Car	Brandy Clark	Washington	Female	2020	Country

The final dataset is 8 columns, 4775 rows, with over 11000+ brand mentions across 3 different genres. It analysed 8000 songs; the workflow can be used to check for brand mentions for any artist downloaded from the Genius API.

Statistical Test:

Since the final dataset is completed, I was able to complete the findings section of my report. The 4 statistical tests I did was a time series analysis of brand mentions over the 3 genres, a t-test comparison of number of brand mentions for Male vs Female in rap, Kruskal-Wallis H test comparing the 3 different genres and a correlation test between google trends data and the amount of brand mentions related to that state for the top 10 most mentioned brands.

Going forward:

Next on my list is to get my Tableau dashboard fully completed, I have an idea of what I want it to look like, however this is my first time using it so I am sure there will be a learning curve. Once this is completed, I am going to move onto building my neural network. My lecturer Noel Cosgrave suggested I build a Long short-term memory recurrent neural network and kindly provided me with some resources on building it.

Software Project Reflective Journal

X14535037 – Conor Carroll

April

Microsoft SQL Server

I have decided to store my final dataset on a Microsoft SQL Server database. Originally, I wanted to use MongoDB, but I had a lot of difficulty uploading from Alteryx and linking it to Tableau. I decided to try Microsoft SQL which I got working very quickly so decided to use this for the project.

Tableau Dashboard

I completed my Tableau dashboard, there is two workbooks finished. One gives geographical insight using a map of the USA and another one gives a breakdown of the artists. It is linked to the SQL server and is uploaded on Tableau public to be viewed.

Lyric Generation

I have used the Python notebook compiled by TensorFlow and tweaked it to generate music lyrics using my dataset. The output data will be performed over a royalty free backing sample.

Going Forward

The remaining pieces of my project are to finish the write up, create the code repository and create my project poster.

10.3. Other materials used

National College of Ireland

Project Proposal

Using song lyrics as a dataset to see which brands are most frequently mentioned by breakdown of Genre/Artist/Geographically. Using a machine learning to algorithm to create a set of lyrics in the style of a specific genre.

04/11/2020

BSc Computing

Data Analytics

2020/2021

Conor Carroll

X14535037

X14535037@student.ncirl.ie

Contents

1.0 Objectives

My Project has two main objectives, a Data Analysis objective and a Machine Learning objective.

Data Analysis:

The first objective is to use the public API from Genius.com and analyse which brands are mentioned in song lyrics. Genius.com is a song lyrics website which contains metadata on nearly every musical artist in the world as well as their song lyrics from just about every song written/produced by the artist. My project will use the API to take lyrics and other metadata from the API, then combine it with another dataset containing a list of brand names to identify mentions of brands within the lyrics.

I plan on analysing the data by breaking down which brands are mentioned and how frequently they are mentioned by artists, by genre and geographically. The finished product will be interactive and allow users to search by artist and genre which will then bring up a list of mentioned brands and how frequently they are mentioned. It will also have a map of the USA with states and cities which users can click on to see which brands are mentioned most by artists hailing from the location.

Machine Learning:

The second objective is to take a set of lyrics from a chosen genre from the Genius API and output a set of lyrics using AI. The dataset would then be fed into a machine learning script which will output a set of lyrics in the style of the chosen genre. Ideally the output lyrics will use the same wording and style as the genre.

Once the lyrics are output by the machine learning script, I'll then use a text to speech tool and take a royalty free vocal sample in the style of the genre and have the lyrics read out or "sung" by the text to speech tool. The objective being have a computer create the lyrics and "sing" them and compare them to that of a human creation.

2.0 Background

Over summer I had been thinking about a potential project to do. I was doing a lot of work with data for my internship which I found challenging and rewarding so I decided I would specialize in data analysis for my final year. I had seen a data analysis project done on transcripts of Donald Trump's speeches during his campaign rallies which analysed the words he used. This was the first time I had seen a sheet of words which were read out aloud and parsing it to be used for data analysis. This put the idea in my head of taking song lyrics, parsing it, and using it as a dataset.

During my internship I got hands on experience writing R code, using data visualizers like Power BI and Alteryx. I gained a good understanding of what can be done with data and decided I would like to take on a data project with a topic I am passionate about

I have a keen interest in song lyrics and lyric writing which led me to chose this. I also have an interest in fashion and clothing, I thought an interesting topic would be to analyse which clothing brands are mentioned the most within song lyrics as lyrics today tend to be full of references to certain brands. I did some research to see if this had been done before, I found some articles which had single picture infographics with brands mentioned by certain artists, but nothing on the scale of what I am proposing. I did not come across anything which broke it down into groups I previously mentioned or with an interactive map to look at it geographically.

Originally when I thought of the idea, I planned on web scraping the data from Genius.com. It was only when I started researching the project that I came across the Genius API, purpose built by the developers at Genius.com to help developers tap into their huge library of metadata. There is good documentation on the API and a forum to ask other users questions about the API if necessary.

I have also found a dataset on Kaggle.com which contains the name of 1 million brands. This will be used for identifying the brands within the lyrics, there will be some manipulation required to avoid false positives etc. I plan on using Alteryx to sanitize the dataset.

I decided to do the second part of my project for two reasons. The main reason being I wanted to get some experience in machine learning and artificial intelligence, as it is an

interesting topic and a booming sector which will only get more in demand. The other reason is I was inspired by a YouTube video I had seen a year ago from a user called "Carykh" which is titled "AI Learns to Write Rap Lyrics!". This user posts a lot of content about AI and his process of creating the programs, which first generated my interest in the topic.

In this video he feeds in a set of lyrics from a handful of artists from the rap genre and the AI produces a mostly non-sensical list of words that had been used in the songs. I want to attempt to get the AI to understand the structure of sentences and make a more coherent version of lyrics. I also want to try different genres to see how the output lyrics differentiate in style and vocabulary.

In the previously mentioned video from "Carykh" at the end, he feeds the lyrics into a generic text-to-speech tool which robotically reads out the lyrics over a rap beat. I want to differentiate my work again by using a text to speech tool which can mimic the voice of a sound file fed into it. I found out about this tool in another YouTube video titled "I Cloned Billie Eilish's Voice & Face with AI" by a user called "Will Kwan". In this video he takes a sound clip of Pop Singer Billie Eilish speaking in an interview, feeds it into the AI and is then able to type in text which is output in her voice.

I plan on using the text-to-speech tool in this video but using a royalty free sound clip of someone singing in the style of each genre and feeding the lyrics sheet into it, so it sounds more authentic as opposed to a generic text to speech tool reading it out.

3.0 Technical Approach

I will take the following approach for my project

- **Implementing Genius API:** I will learn how to implement the API and how to get metadata and lyrics from it.

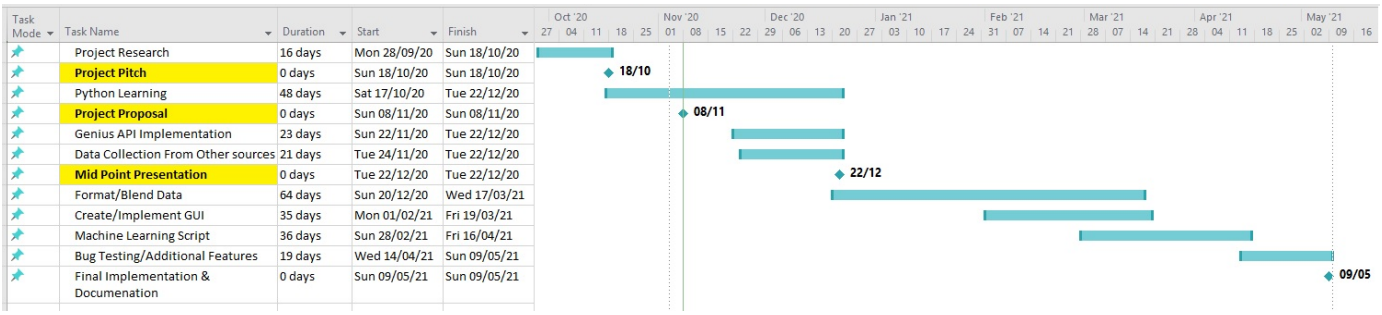
- **Cleaning data sources:** Parse lyrics into single words. Clean brands data source to allow it to combine with lyric data.

- **Combine data sources:** Once I have gathered the data and figured out how to implement the Genius API, I will combine the data.

- **Visualise results:** Develop interactive dashboard, charts and USA map.

- **Build AI script to recreate lyrics:** Create script using Python to automate the output of lyrics based on that fed into it.

4.0 Project Plan



Task Mode	Task Name	Duration	Start	Finish
	Project Research	16 days	Mon 28/09/20	Sun 18/10/20
	Project Pitch	0 days	Sun 18/10/20	Sun 18/10/20
	Python Learning	48 days	Sat 17/10/20	Tue 22/12/20
	Project Proposal	0 days	Sun 08/11/20	Sun 08/11/20
	Genius API Implementation	23 days	Sun 22/11/20	Tue 22/12/20
	Data Collection From Other sources	21 days	Tue 24/11/20	Tue 22/12/20
	Mid Point Presentation	0 days	Tue 22/12/20	Tue 22/12/20
	Format/Blend Data	64 days	Sun 20/12/20	Wed 17/03/21
	Create/Implement GUI	35 days	Mon 01/02/21	Fri 19/03/21
	Machine Learning Script	36 days	Sun 28/02/21	Fri 16/04/21
	Bug Testing/Additional Features	19 days	Wed 14/04/21	Sun 09/05/21
	Final Implementation & Documenation	0 days	Sun 09/05/21	Sun 09/05/21

- My plan is to have all my data cleaned and ready to go by the mid-point presentation. I also want to achieve a decent level of Python Proficiency by then.

- After the data is ready to go it is a case of blending the data and visualizing it/making it interactive.

- Then I will turn my attention to the machine learning aspect of my project, as I'll be proficient with the Genius API by then.

- I am allowing 3 weeks to test for bugs and resolve any issues that may arise.

5.0 Technical Details

The following technologies/languages will be used in my project:

- **Python:** Python will be used to run the scripts that interact with the API and pull information into a .csv file. The machine learning algorithm for the computer-generated lyrics will also be written in Python

- **Alteryx:** I will use Alteryx to visualize the data. I also may use it for blending and cleansing the data

- **Genius API:** This will be where the main source of data comes from. It will be implemented in a Python script.

6.0 Evaluation

I plan to evaluate the **data analysis** side of my project as follows:

- Is the data visualized correctly and in such a way that the average person can gain insight from it?
- Is the data collected from the source correctly?
- Can the collection of the data be automated?
- Can new data be added to the dataset automatically?
- Is the data from each source cleansed and blended without issue?

I plan to evaluate the **machine learning** side of my project as follows:

- Do the lyrics make sense in any way?
- Does the vocabulary used reflect that of the chosen genre?
- Does the output text-to-speech sound similar to that of a human?
- Can any genre be chosen and successfully output a set of lyrics?

Commercialism in Popular Music: Analysing Brand Mentions in Song Lyrics.

Conor Carroll
Computer Science - Data Analysis

The Project

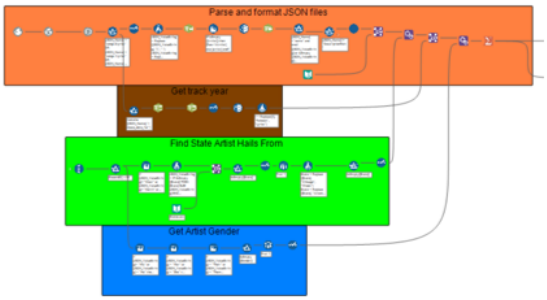
Using a range of technologies and programming languages, I have built a tool which can identify commercial brand mentions within music lyrics in seconds. My project analyses the lyrics of over 8,000+ songs across three genres and identifies over 11,000+ brand references within the lyrics. The results are then analysed using r programming language and visualised using Tableau.

Another feature of the project is a machine learning script which can generate music lyrics in the style of three different genres (rap, pop and country). Using Python with TensorFlow, a Recurrent Neural Network trains itself on a set of lyric data. It then generates a set of lyrics based on the input dataset.

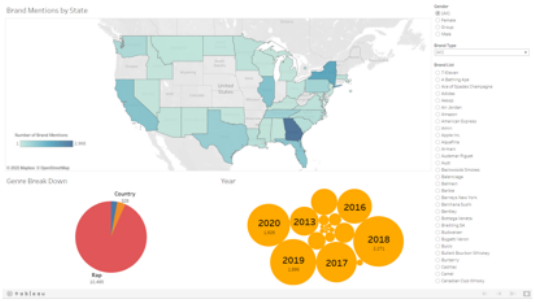
[Chorus: Young Thug]
She want Chanel, go get it
She want Chanel, go get it
She want this Fendi, go get it
She want a Birkin, go get it
She want this Gucci, she can get it
She want this Louis, go get it
Loubs with the spikes, she get it
Everything I got, she gettin' it, yeah
Bentley sedan, she get it
Hop out the Benz, she gettin' it

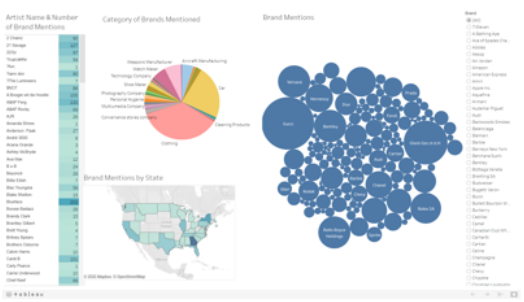
```
#Assign your Genius.com credentials and select your
import lyricsgenius as genius
geniusCreds = "eaIZHw-d7dxFiQsoUoPyimMC3N19a_9ix8qV
artist_name = "Drake"

api = genius.Genius(geniusCreds)
import os
os.getcwd()
artist = api.search_artist(artist_name, max_songs=4
artist.save_lyrics()
```

```
Searching for songs by Drake...

Song 1: "God's Plan"
Song 2: "In My Feelings"
```



Count	Actual Name	Type	Artist	State	Gender	Year
4	Balmain	Clothing	Gunna	Georgia	Male	2019
3	Barneys New York	Clothing	Gunna	Georgia	Male	
3	Barneys New York	Clothing	Gunna	Georgia	Male	2019
3	Benihana Sushi	Restaurant	Gunna	Georgia	Male	2020
1	Bentley	Car	Gunna	Georgia	Male	
3	Bentley	Car	Gunna	Georgia	Male	2018
7	Bentley	Car	Gunna	Georgia	Male	2019
4	Bentley	Car	Gunna	Georgia	Male	2020
1	Bugatti Veron	Car	Gunna	Georgia	Male	2018
1	Cartier	Jewelery	Gunna	Georgia	Male	
2	Cartier	Jewelery	Gunna	Georgia	Male	2018
1	Cartier	Jewelery	Gunna	Georgia	Male	2019
2	Chanel	Clothing	Gunna	Georgia	Male	2017
4	Chanel	Clothing	Gunna	Georgia	Male	2018
7	Chanel	Clothing	Gunna	Georgia	Male	2019
3	Chrome Hearts	Clothing	Gunna	Georgia	Male	2018
4	Chrome Hearts	Clothing	Gunna	Georgia	Male	2020





Technologies Used

python
Microsoft SQL Server
TensorFlow
GENIUS {API}
+ableau
alteryx