Billboard Music Trends by Twitter

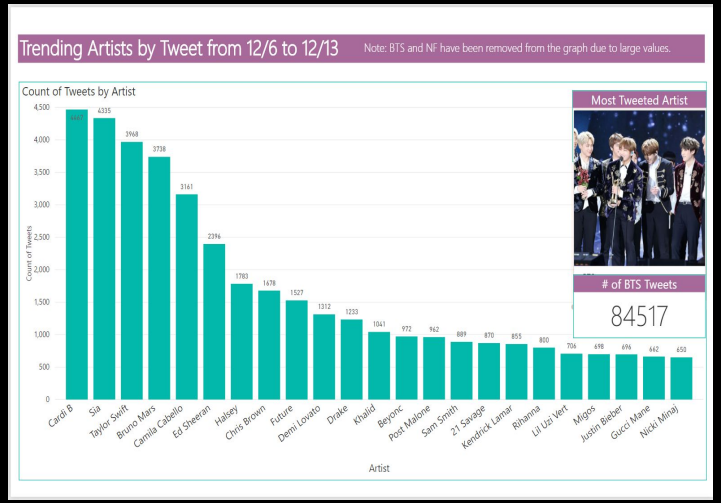
W205 Storing and Retrieving Data - Final Project

Anusha Munjuluri, Cathy Zhou, Harry Xu

Project Scope

- Music Trends in Twitter
- Understanding User Listening Habits

Project Scope and Definition:



3 Stages of Architecture:

1 | Data Ingestion

- Twitter
  - Tweepy
  - Twitter API
  - Apache Storm
  - Pyscopg2
- Billboard
  - Billboard API
  - Mozenda
- Spotify
  - Spotify API
- PostgreSQL DB

2 | Data Transformation

- Extract from tweet:
  - Artist
  - Album
  - Song
  - Channel
- Extract Sentiment using Text Blob
- Parsing JSON, array formatted fields

3 | Data Visualization

- Moved data to Spark for fast processing
- Power BI
  - PostgreSQL DB
  - Flat Files

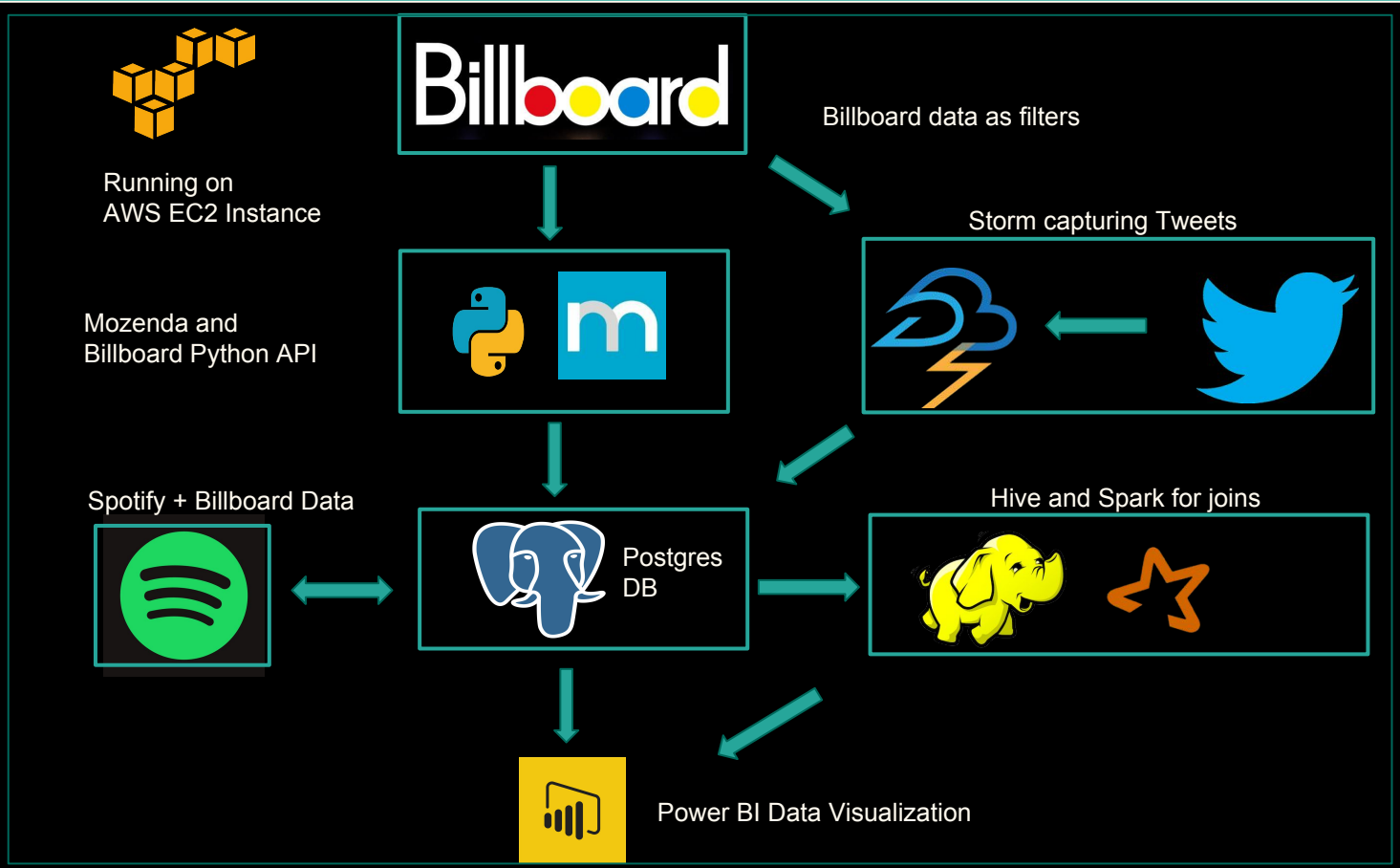
Architecture Diagram:

Data Sources:
Twitter, Spotify, Billboard

Data Storage:
Postgres, Hive, Spark

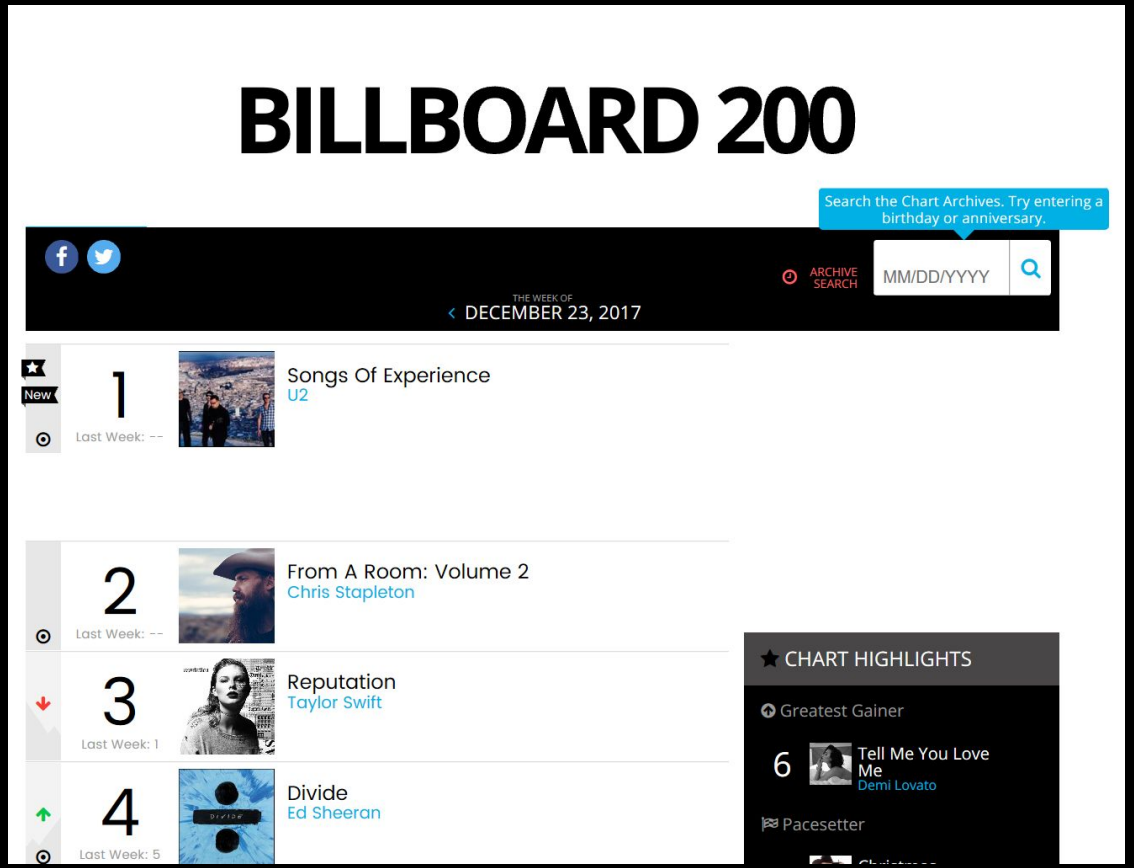
Parsing:
Python, Tweepy, Psycopg,TextBlob

Data Visualization:
Power BI



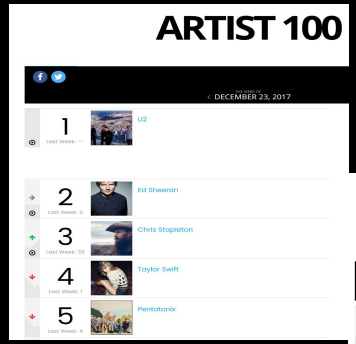
Data Ingestion

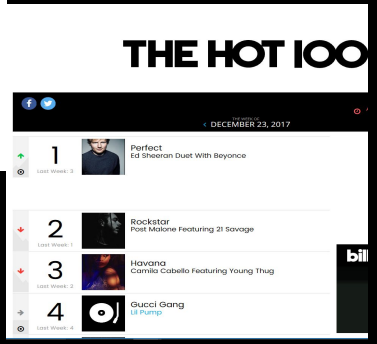
Data Source 1 : Billboard Ingestion



3 Billboard charts used:

1. Top 200 Album
2. Hot 100 Songs
3. Top 100 Artists





Data Source 1 : Billboard Ingestion (cont.)

Billboard

billboard.py

build passing

billboard.py is a Python API for accessing music charts from Billboard.com.

https://github.com/guoguo12/billboard-chartshttps://github.com/guoguo12/billboard-charts

Quickstart

To download a *Billboard* chart, we use the `ChartData()` constructor.

Let's fetch the current Hot 100 chart.

```
>>> import billboard
>>> chart = billboard.ChartData('hot-100')
```

Now we can look at the chart entries, which are of type `ChartEntry` and have attributes like `artist` and `title`:

```
>>> song = chart[0]  # Get no. 1 song on chart
>>> song.title
u'Despacito'
>>> song.artist
u'Luis Fonsi & Daddy Yankee Featuring Justin Bieber'
>>> song.weeks  # Number of weeks on chart
30
```

We can also `print` the entire chart:

```
>>> print chart
hot-100 chart from 2017-08-26
----------------------------
1. 'Despacito' by Luis Fonsi & Daddy Yankee Featuring Justin Bieber
2. 'Wild Thoughts' by DJ Khaled Featuring Rihanna & Bryson Tiller
3. 'Unforgettable' by French Montana Featuring Swae Lee
4. 'Believer' by Imagine Dragons
# ...
```

Data Source 1 : Billboard Ingestion (cont.)

mozenda | Agents | Collections | Folders | Upgrade | Processing credits: 499

BillBoard 200 / 1002
Billboard

Ready

*Ran successfully **7 minutes, 32 seconds** a...*

1 of 1
Return to agent group

Data | Find | In | All Fields | Most recent completed run | Default

200 items | Add item

ItemID	Album Name	Artist Name
1001	Reputation	Taylor Swift
1002	A Pentatonix Christmas	Pentatonix
1003	The Thrill Of It All	Sam Smith
1004	The Anthology: Part I, The Fi...	Garth Brooks
1005	Divide	Ed Sheeran
1006	Stoney	Post Malone
1007	Beautiful Trauma	P!nk
1008	Luv Is Rage 2	Lil Uzi Vert
1009	Christmas	Michael Buble
1010	Friday On Elm Street	Fabolous & Jadakiss
1011	Heartbreak On A Full Moon	Chris Brown
1012	Project Baby Two	Kodak Black

Page 1 of 1 | **1 - 200** of **200** | 250 Items per page

Help

AGENT DASHBOARD

Agents collect data from websites and store the data in Collections.

Use the Agent Builder to create a new Agent or to modify an existing Agent.

Additional Information

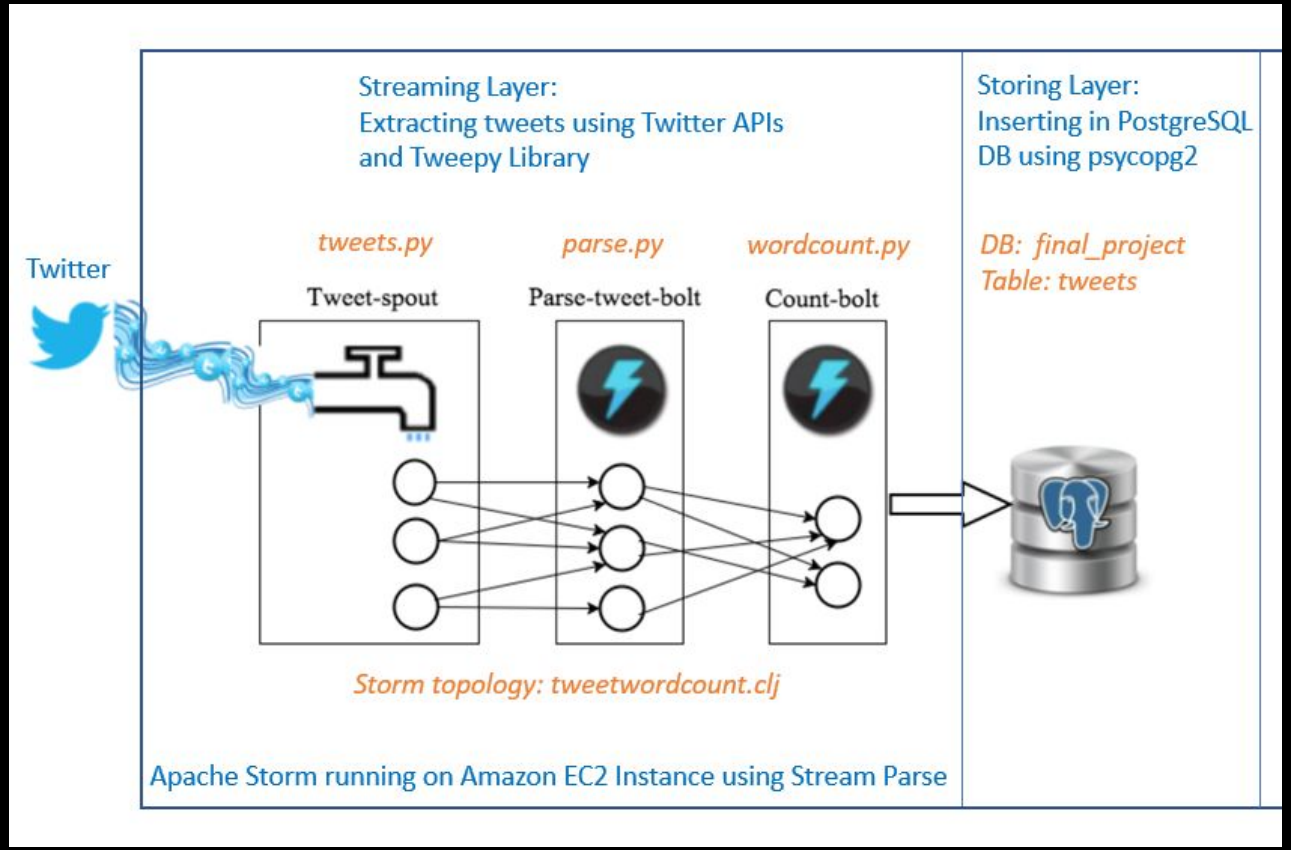
- Creating and modifying Agents
- Viewing collected data
- Viewing Job Statistics
- Status
- Schedules
- Tools
- GeoLocating an Agent

Creating and modifying Agents

Use the Agent Builder to create a new Agent or to modify an existing Agent.

You can start the Agent Builder using a variety of methods:

Data Source 2 : Twitter Ingestion



Ran 2 storms with different set of filters:

Components of Storm:
3 Spouts
3 Parse Bolts
2 Count Bolts

Run time: ~ 1 week

Number of Tweets collected: ~ 30 million

Data Source 2 : Twitter Ingestion (cont.)

```
stream.filter(track=["#NowPlaying","#ListeningTo","#Spotify", "#listenlive",
"#Grammy", "#GrammyNomination", "#Grammy2018", "#Grammys","listening to
Amazon Prime Music", "listening to Amazon Music", "listening to Apple Music",
"listening to Deezer", "listening to Gaana", "listening to Ghost Tunes", "listening
to Ghost iTunes", "listening to Google Play All playing Raaga", "now playing
Radical.fm", "now playing Yandex Music"], async = True)
```

Storm 1 Filters used:

Hashtag Filters:

#nowplaying, #listeningto
#listenlive, #grammys, #spotify

Channel Filters:

Listening to Spotify, now playing Pandora

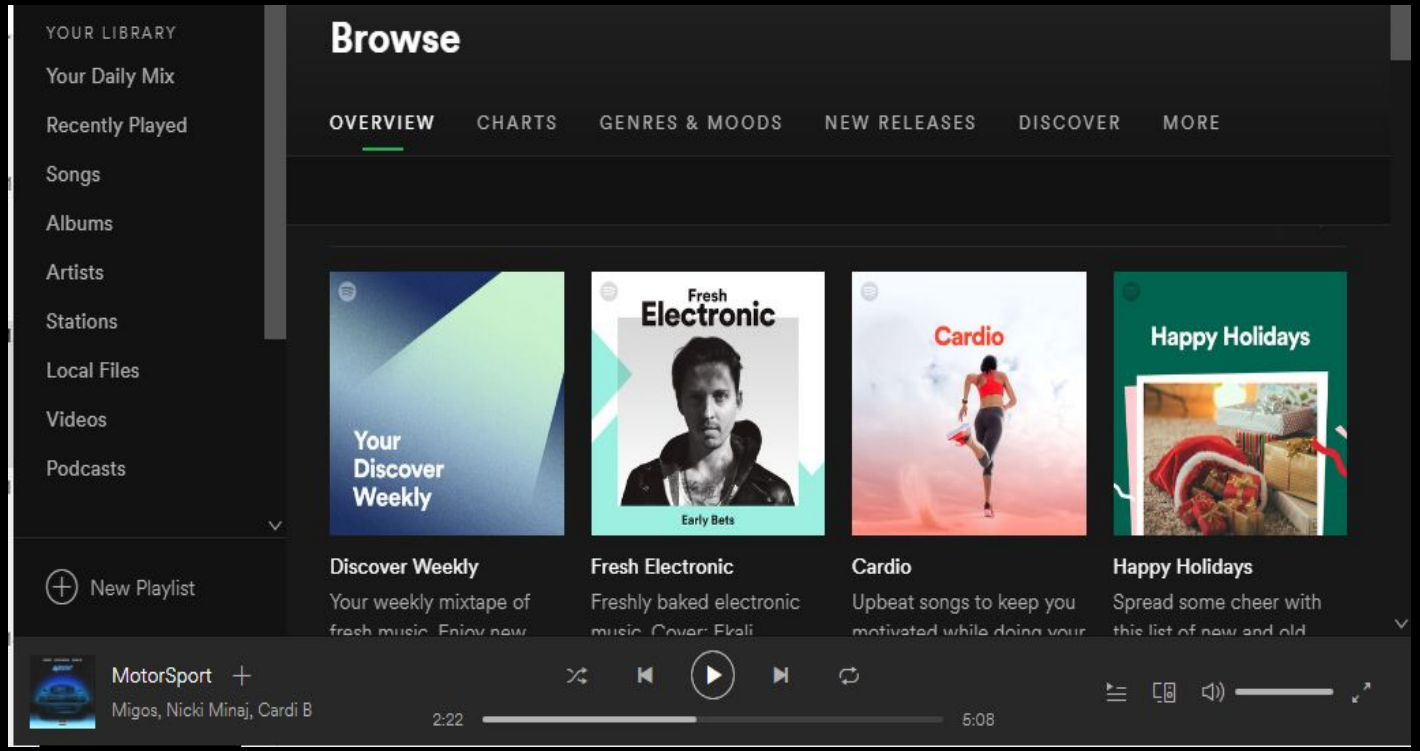
Storm 2 Filters used:

Artist Filters:
Top 20 Billboard Artists

Album Filters:
Top 20 Billboard Albums

```
stream.filter(track=["listening to Taylor Swift","listening to
Pentatonix","listening to Sam Smith","listening to Garth Brooks","listening to
Ed Sheeran",""DAMN by Kendrick Lamar","That's Christmas To Me by
Pentatonix","Without Warning by 21 Savage, Offset & Metro
Boomin","American Teen by Khalid","Lil Pump by Lil Pump","The Rest Of Our
Life by Tim McGraw & Faith Hill"], async = True)
```

Data Source 3 : Spotify Ingestion



Spotify data used:

1. Song details
2. Artist details
3. Album details

Data Source 3 : Spotify Ingestion (cont.)

Song

Album

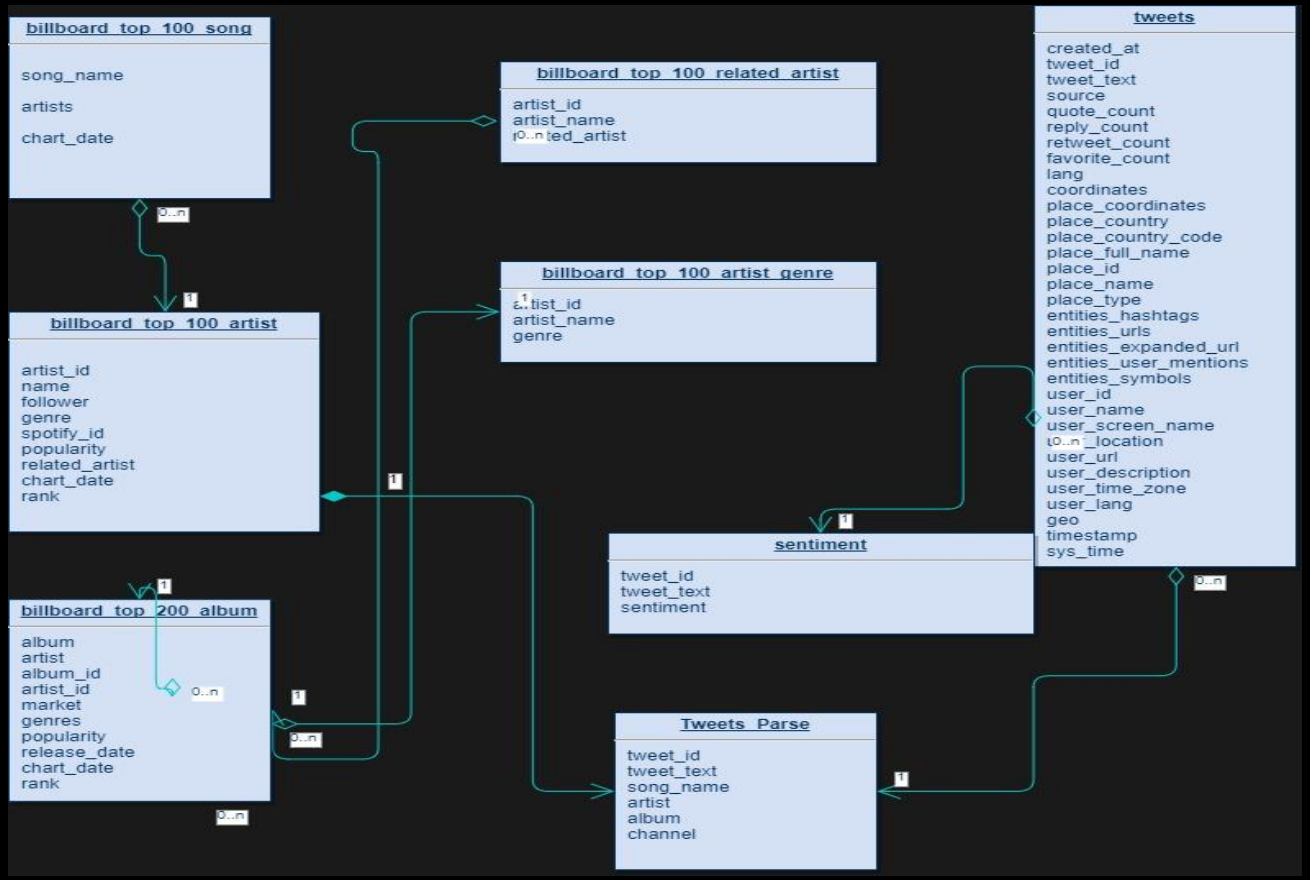
Artist

Spotify API

```
sp.search(q=i,type = 'artist')
spotify.artist(i)
sp.artist_albums(i, album_type='album')
sp.audio_features(str(song))
```

DataBase

ER Diagram


billboard_top_100_song
song_name
artists
chart_date
billboard_top_100_related_artist
artist_id
artist_name
related_artist
billboard_top_100_artist_genre
artist_id
artist_name
genre
tweets
created_at
tweet_id
tweet_text
source
quote_count
reply_count
retweet_count
favorite_count
lang
coordinates
place_coordinates
place_country
place_country_code
place_full_name
place_id
place_name
place_type
entities_hashtags
entities_urls
entities_expanded_url
entities_user_mentions
entities_symbols
user_id
user_name
user_screen_name
user_location
user_url
user_description
user_time_zone
user_lang
geo
timestamp
sys_time
billboard_top_100_artist
artist_id
name
follower
genre
spotify_id
popularity
related_artist
chart_date
rank
sentiment
tweet_id
tweet_text
sentiment
billboard_top_200_album
album
artist
album_id
artist_id
market
genres
popularity
release_date
chart_date
rank
Tweets_Parse
tweet_id
tweet_text
song_name
artist
album
channel
0..n
1

Data Transformation

Step 1: Extracting Artist, Album, Song, Channel


```
{'contributors': None,
 'coordinates': None,
 'created_at': 'Mon Dec 18 02:40:19 +0000 2017',
 'entities': {'hashtags': [{'indices': [55, 66], 'text': 'NowPlaying'}],
              'symbols': [],
              'urls': [{'display_url': 'open.spotify.com/track/1SBzPKUb…',
                        'expanded_url': 'https://open.spotify.com/track/1SBzPKUbThoavBuPIbi3pn',
                        'indices': [31, 54],
                        'url': 'https://t.co/3iHzlpaJaD'}],
              'user_mentions': []},
 'favorite_count': 0,
 'favorited': False,
 'filter_level': 'low',
 'geo': None,
 'id': 942585286588092418,
 'id_str': '942585286588092418',
 'in_reply_to_screen_name': None,
 'in_reply_to_status_id': None,
 'in_reply_to_status_id_str': None,
 'in_reply_to_user_id': None,
 'in_reply_to_user_id_str': None,
 'is_quote_status': False,
 'lang': 'en',
 'place': None,
 'possibly_sensitive': False,
 'quote_count': 0,
 'reply_count': 0,
 'retweet_count': 0,
 'retweeted': False,
 'source': '<a href="http://twitter.com" rel="nofollow">Twitter Web Client</a>',
 'text': 'Say When by CHill The Producer https://t.co/3iHzlpaJaD #NowPlaying',
 'timestamp_ms': '1513564819367',
 'truncated': False,
 'user': {'contributors_enabled': False,
          'created_at': 'Tue Feb 04 06:30:55 +0000 2014',
          'default_profile': False,
          'default_profile_image': False,
          'description': 'Musician making relaxed music for #relaxation and '
                         '#meditation guitars and synth over amazing beats by '
                         '@chillthemonster available on #Spotify #iTunes and '
                         'more.',
```


Hashtag	:	#NowPlaying
Song	:	Say When
Artist	:	Chill The Producer
Channel	:	Spotify

Step 2 : Extracting Sentiment

Used Python TextBlob library for sentiment:
https://pypi.python.org/pypi/textblob

Gives sentiment in range of -1 to 1 (-ve to +ve)

```
>>> text = "Listening to countdown by @Beyonce Love love love!!! "
>>> blob = TextBlob(text)
>>> blob.sentiment
Sentiment(polarity=0.6588541666666666, subjectivity=0.6)
```

Removing song name from tweet before getting sentiment as it may change the polarity.

5	Perfect
6	Bodak Yellow (Money Moves)
7	Too Good At Goodbyes
8	Feel It Still
9	Sorry Not Sorry
10	What Lovers Do
11	No Limit
12	Mi Gente
13	1-800-273-8255

Step 3: Extracting JSON formatted fields

```
{'aritist_id:'06HL4z0CvFAxyc27GXpf02',
 'artist': 'Taylor Swift',
 'related_artist': {"Meghan Trainor","Selena Gomez","Miley Cyrus","Ariana Grande",
                    "Demi Lovato","Carly Rae Jepsen","Hailee Steinfeld",
                    "Kelly Clarkson","Katy Perry","Rachel Platten","Carrie Underwood",
                    "Little Mix","Justin Bieber",ZAYN,"Ed Sheeran",DNCE,"Fifth Harmony",
                    "Shawn Mendes",Lorde,"Lady Gaga"},
 'genre': {"dance pop",pop,"pop christmas","post-teen pop"}}
```

artist_id	artist	value
06HL4z0CvFAxyc27GXpf02	Taylor Swift	dance pop
06HL4z0CvFAxyc27GXpf02	Taylor Swift	pop christmas
06HL4z0CvFAxyc27GXpf02	Taylor Swift	pop

Extract Artist Categories/Genre

artist_id	artist	value
06HL4z0CvFAxyc27GXpf02	Taylor Swift	Meghan Trainor
06HL4z0CvFAxyc27GXpf02	Taylor Swift	Miley Cyrus
06HL4z0CvFAxyc27GXpf02	Taylor Swift	Selena Gomez

Extract Artists - Related Artists

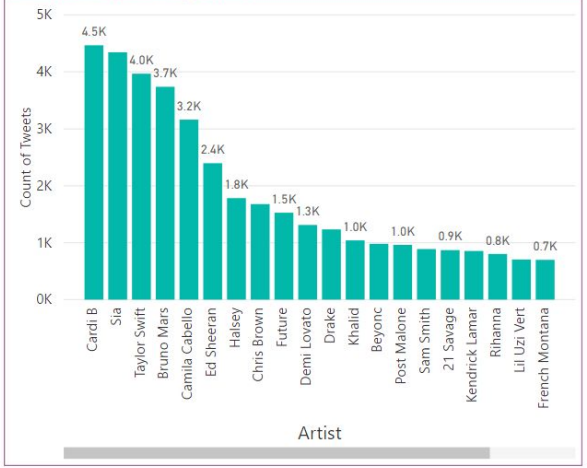
Data Visualization (Live Demo)

Trending Artist by Tweets

Trending Artists by Tweet from 12/6 to 12/13

Note: BTS and NF have been removed from the graph due to large values.

Count of Tweets by Artist



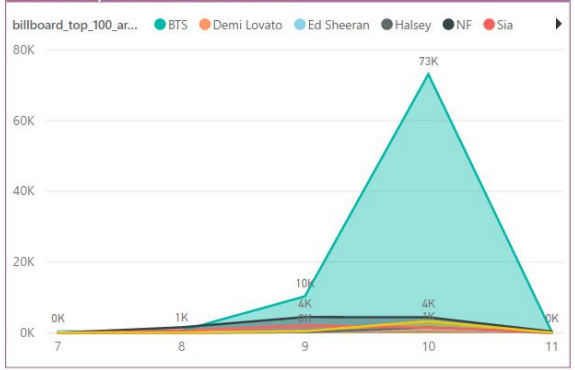
Most Tweeted Artist

BTS

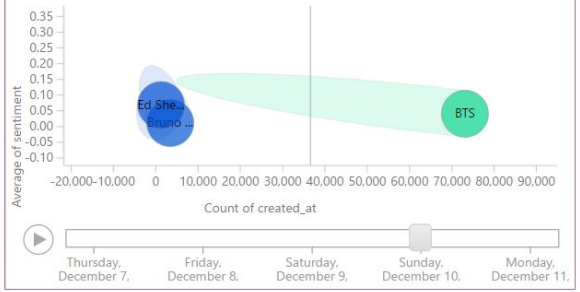
of BTS Tweets

84517

Sudden Spike in Tweets for BTS



Count of Tweets and Sentiment by Date for BTS



BTS Final Tour Date: December 10th, 2017

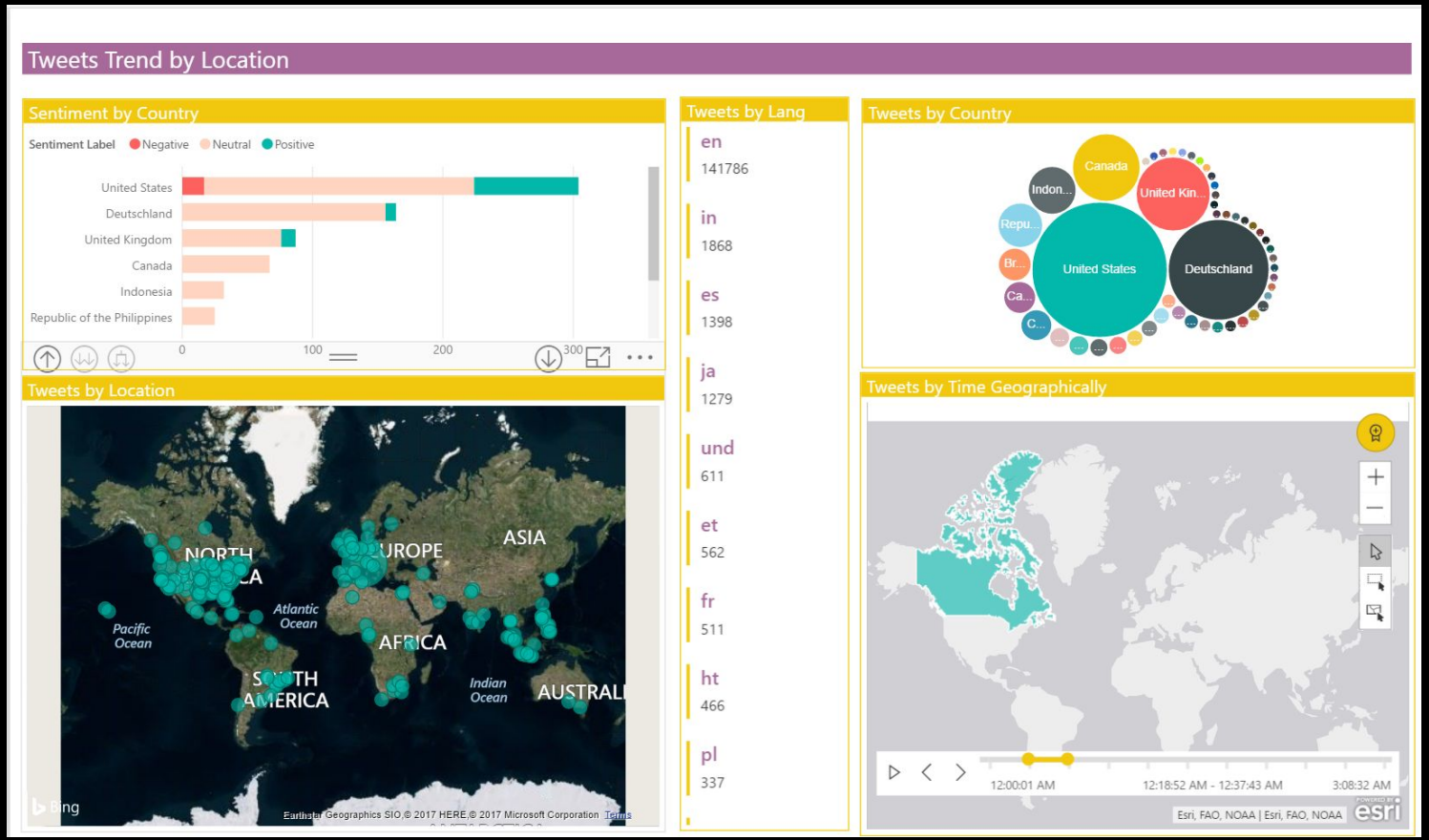
December 8, 2017 December 9, 2017 December 10, 2017	Seoul	South Korea	Gocheok Sky Dome	60,000[32]

https://en.wikipedia.org/wiki/2017_BTS_Live_Trilogy_Episode_III:_The_Wings_Tour

Sudden spike noticed in Tweets for music band BTS on last day of tour Dec 10th

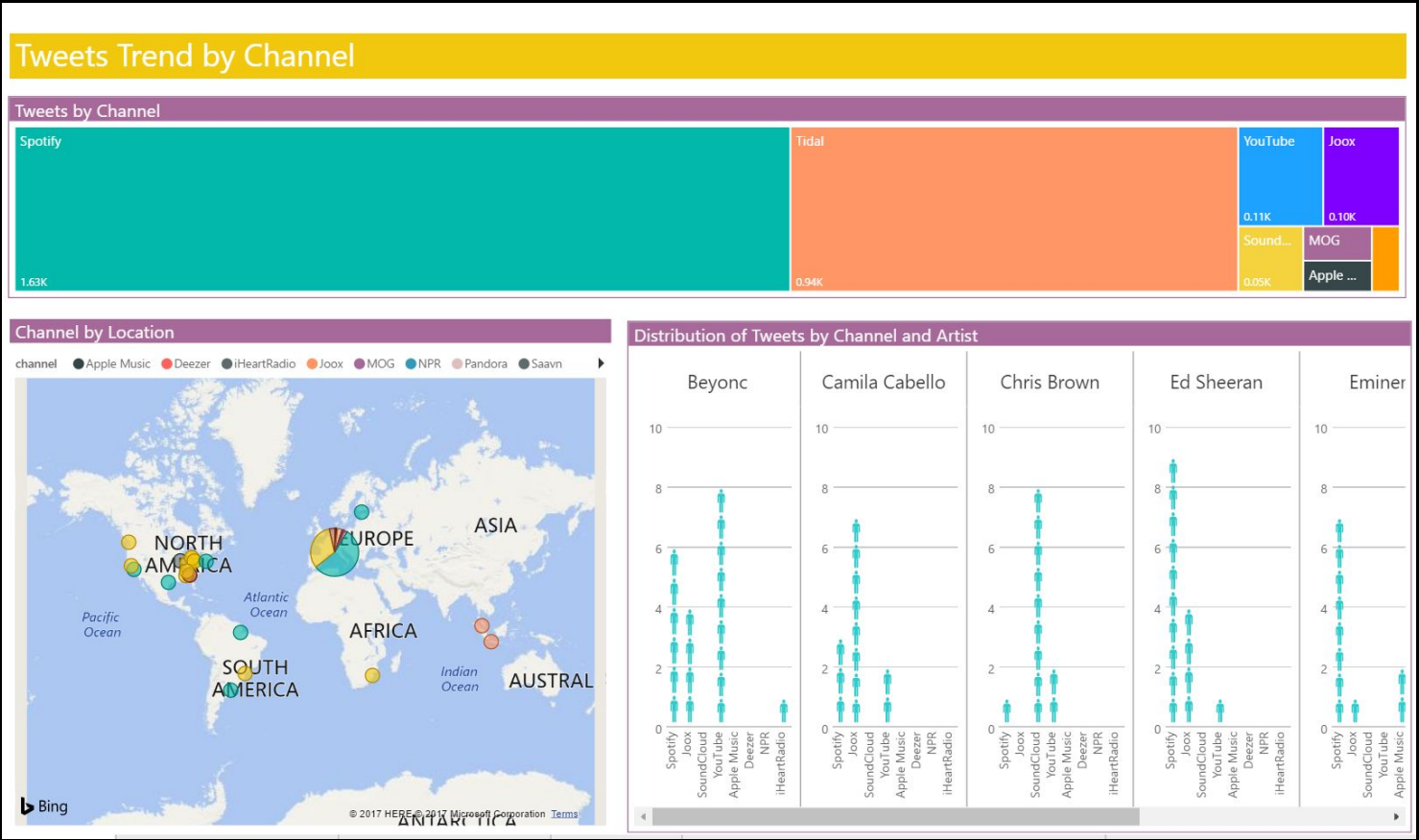
Average sentiment goes down as users express sadness over tour coming to an end

Tweets Trend by Time and Location

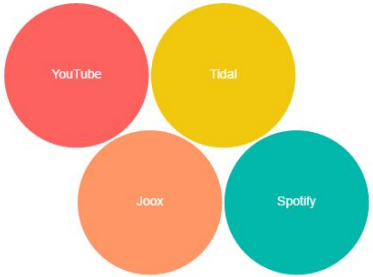


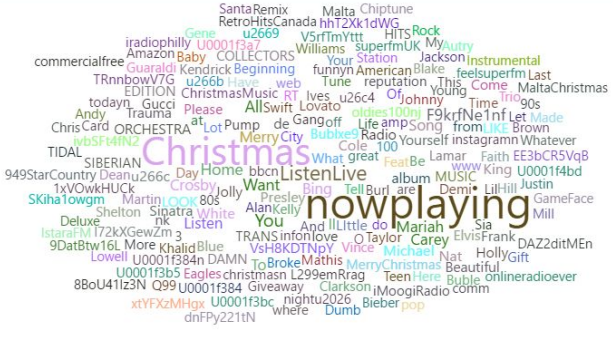
Users tend to Tweet more in the evenings than morning about music.

Distribution of Tweets by Channel



Spotify and Tidal are most preferred music streaming channels.





Featured Album Trends and Sentiment

Songs Listening To

billboard_top_100_song.song_name
...Ready For It?
Bad At Love
Beautiful Trauma
Gucci Gang
Home
How Long

Tweets

tweet_text
#NowPlaying "Young, Dumb And Broke" by Khalid from the album American Teen. https://t.co/wAhHXfySc0
#nowplaying #bingcrosby ~ Bing Crosby | White Christmas ||| Radio TEDDY - In #Kassel #Hessen #GER auf 91.7
#nowplaying #bingcrosby ~ Bing Crosby | White Christmas ||| Radio TEDDY - In #Stralsund #HST #MV #GER auf 93.0
#NowPlaying #FrequencyRad All I Want For Christmas Is You - Mariah Carey Feat Justin Bieber - Le Meilleur de la Mu2026 https://t.co/bzWQhi3pOC

Song Spotify Details

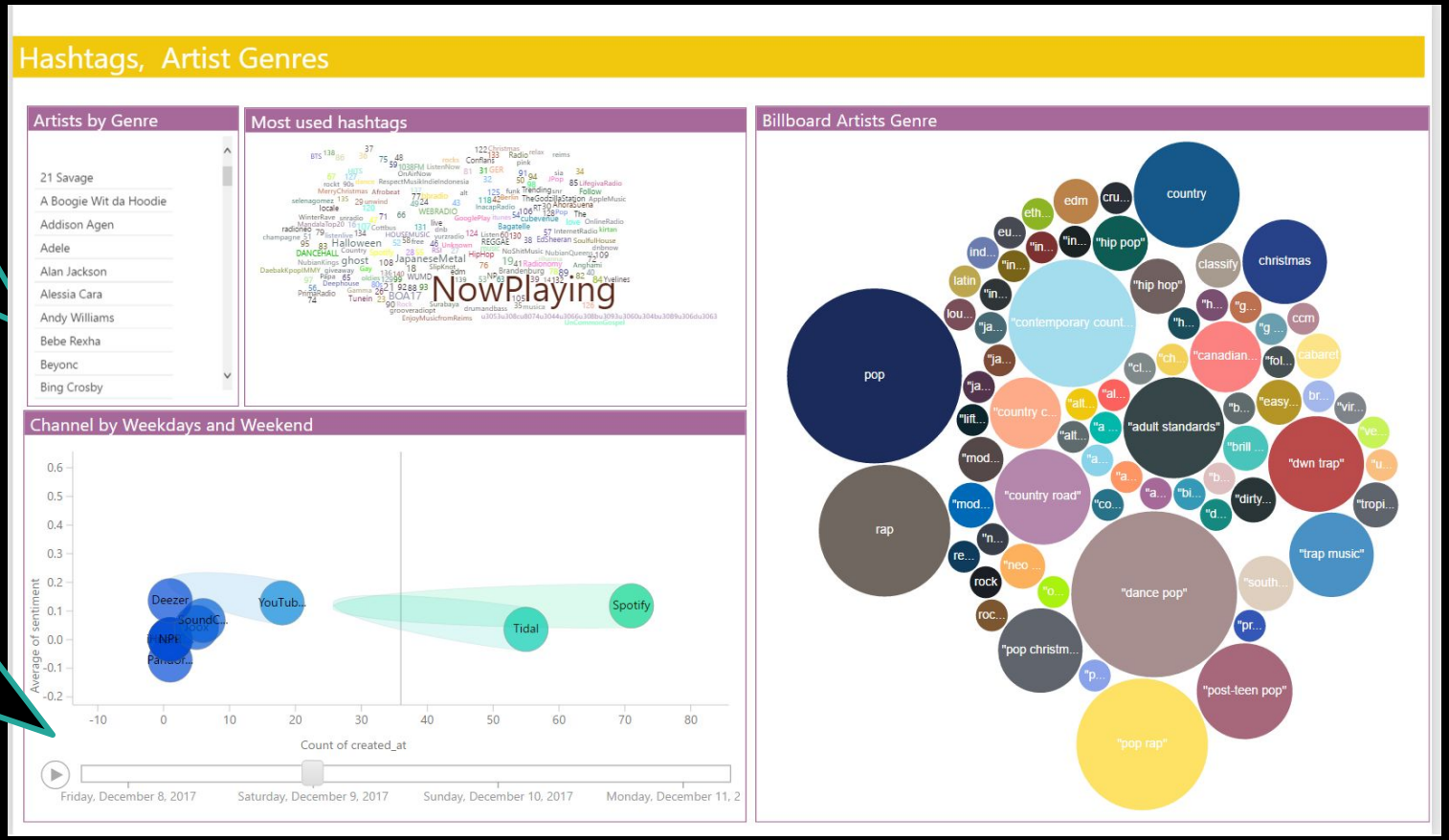
song_name	energy	loudness	liveness
Attention	0.63	-4.43	0.08
Awful Things	0.80	-9.85	0.14
Bad At Love	0.74	-7.21	0.18
Bank Account	0.35	-8.23	0.09
Believer	0.78	-4.29	0.12
Betrayed	0.42	-10.36	0.11

Christmas songs are trending now in Twitter

Most used Hashtags and Artist Genre

More pop related artists trending in Billboard

Users use more streaming services over the weekend



Challenges Faced

Challenges Faced: Data Ingestion/Storage

Issues with Postgres DB:

1. Volume:
   > 500k records Postgres DB becomes slow. Collected about 30 million rows.

2. Table Locks:
   ShareUpdateExlusiveLock when accessing tables by mutliple users.
   https://stackoverflow.com/questions/1063043/how-to-release-possible-postgres-row-locks

3. Processing Speed:
   Spark was better for doing joins across tables.
   (2 million rows table joins)

4. Restriction on length of one line of code

37

It's possible to see the locks.

Here is a view to make it a bit easier than using pg_locks directly:

```
CREATE OR REPLACE VIEW public.active_locks AS
 SELECT t.schemaname,
    t.relname,
    l.locktype,
    l.page,
    l.virtualtransaction,
    l.pid,
    l.mode,
    l.granted
   FROM pg_locks l
   JOIN pg_stat_all_tables t ON l.relation = t.relid
  WHERE t.schemaname <> 'pg_toast'::name AND t.schemaname <> 'pg_catalog'::name
  ORDER BY t.schemaname, t.relname;
```

Then you just select from the view:

```
SELECT * FROM active_locks;
```

And kill it with:

```
SELECT pg_cancel_backend('%pid%');
```

Other solutions: http://wiki.postgresql.org/wiki/Lock_Monitoring

Challenges Faced: Parsing

1. Encoding process for PostgresSQL

   E.g. Beyoncé → Beyonc

2. Free language form of the tweets :

   a. Difficult to extract accurate information

   b. Multiple ways of writing artists/song/album names

   c. Foreign language/emoji

3. Parsing process needs improvement
   1.5 Million tweets
   100 featured songs
   200 featured albums
   100 featured artists
   33 twitter related data points, 25 spotify related data points

Scaling, Automation, Future Enhancements

Scaling, Automation, Future Enhancements

1. Scaling Data Storage

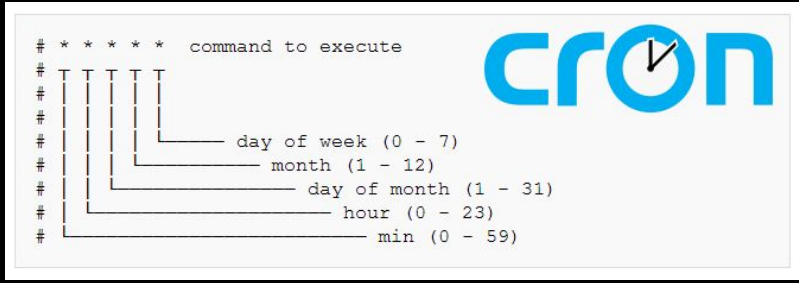
   PostgresSQL → Spark SQL

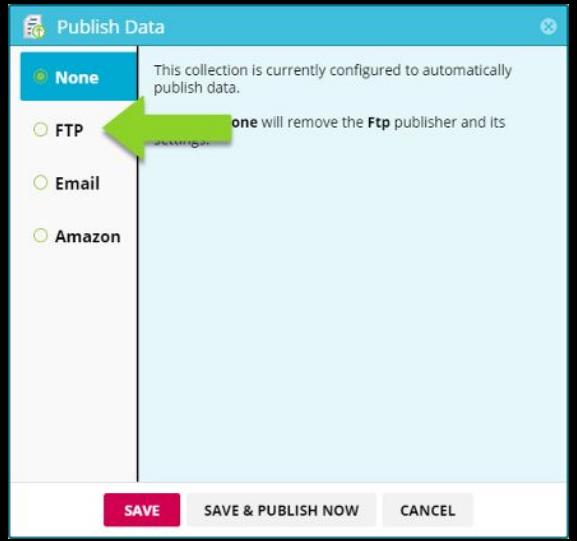
2. Automation

   a. Scheduled cron jobs for collecting updated data
   b. Auto parsing
   c. Automating Mozenda to put Billboard data on AWS

3. Future Enhancements

   a. NLP for parsing free text in depth
   b. Acquire enhanced instance/storage/database type
   c. Improve speed of parsing process





Conclusion

Conclusion

- Further research in music domain
    - By using analytics and trends from application to better understand user listening habits and preferences
- Build better recommendation systems based on user's' listening habits
- Provide artists and professionals in music domain with live updates and Twitter trends about their work

Thank you !

..was a fun project :) !