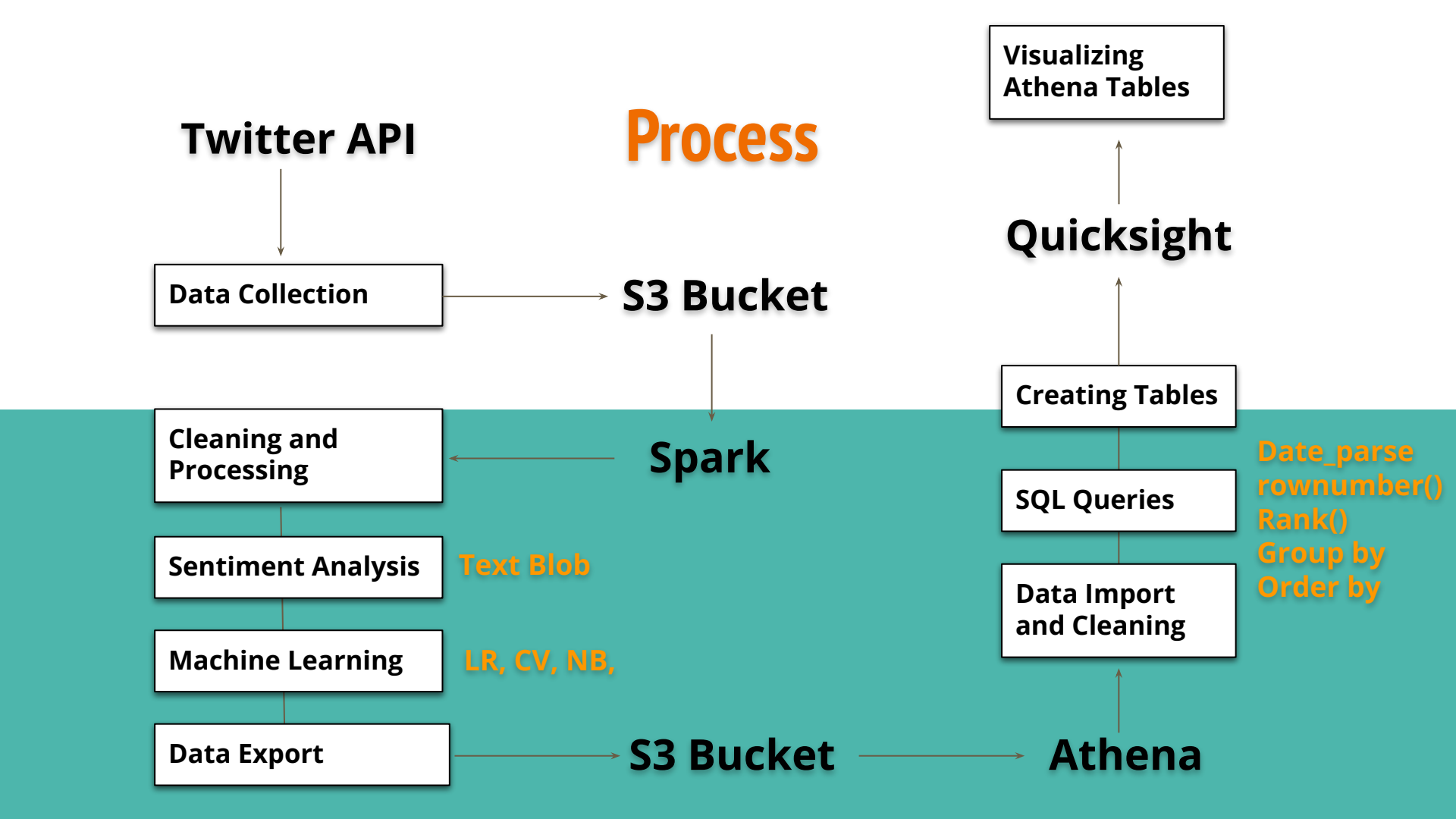
Big Data Project:
Twitter Sentiment Analysis

Laiba Shah
Part Time Data Science Bootcamp

Objectives

- **BIG DATA COLLECTION:** Collect all tweets related to *Elon Musk* using Twitter API
- **DATA PROCESSING:** Import tweets by mounting s3 bucket and cleaning data
- **SENTIMENT ANALYSIS:** Perform sentiment analysis using TextBlob, an NLP library
- **PREDICTION:** Predict sentiment using Machine learning techniques
- **VISUALIZATION:** Query data using Athena and visualize it using Quicksight


Twitter API
Process
Visualizing Athena Tables
Quicksight
Data Collection
S3 Bucket
Creating Tables
Cleaning and Processing
Spark
Date_parse
rownumber()
Rank()
Group by
Order by
SQL Queries
Sentiment Analysis
Text Blob
Data Import and Cleaning
Machine Learning
LR, CV, NB,
Data Export
S3 Bucket
Athena

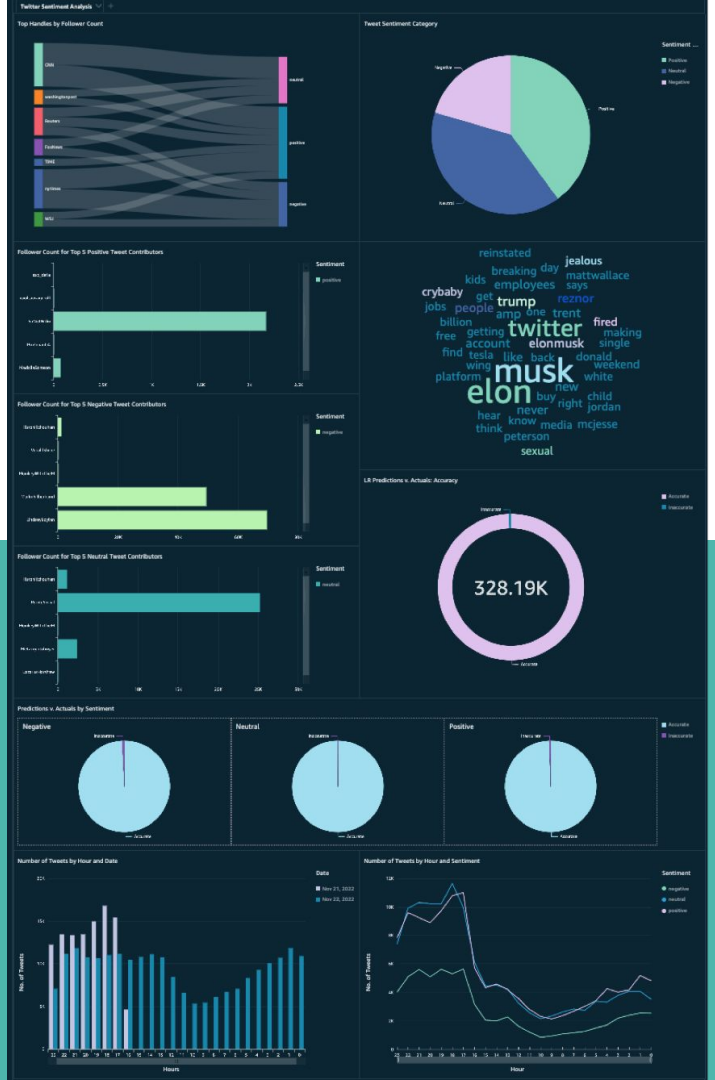
Machine Learning Analysis

- Feature transformation (tokenizer, stopword removal, count vectorizer, TF-IDF vectorization, label encoder)
- Logistic regression classification model fit and MulticlassClassificationEvaluator
- Modeling a pipeline reusable for predicting future tweet sentiment by putting all transformers and estimator

With only feature transformation:
Logistic Regression Accuracy Score: 0.9952

Train, Test Split: 70/30:
Logistic Regression Accuracy Score: 0.9392
CrossValidator Accuracy Score: 0.9516 (LR after hyperparameter tuning)
Naive Bayes Accuracy Score: 0.9109

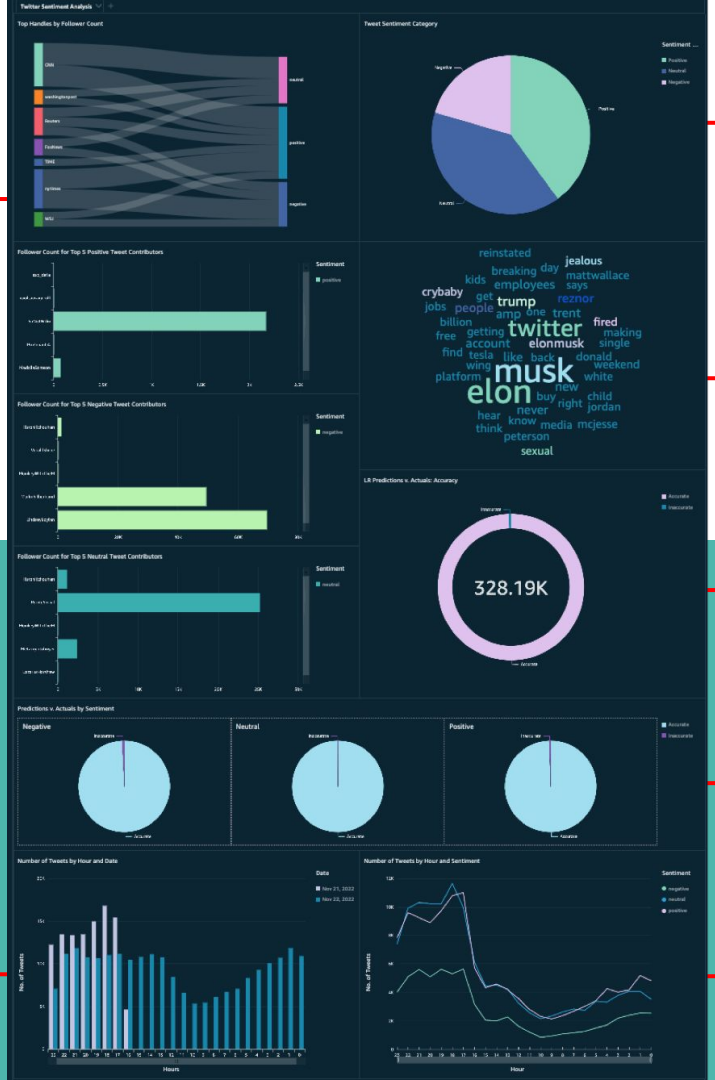
Quicksight Dashboard



Top handles by follower count broken down by sentiment of tweet

Follower count for top 5 positive, negative and neutral tweet contributors

No. of tweets by hour broken down by date



No. of tweets by sentiment

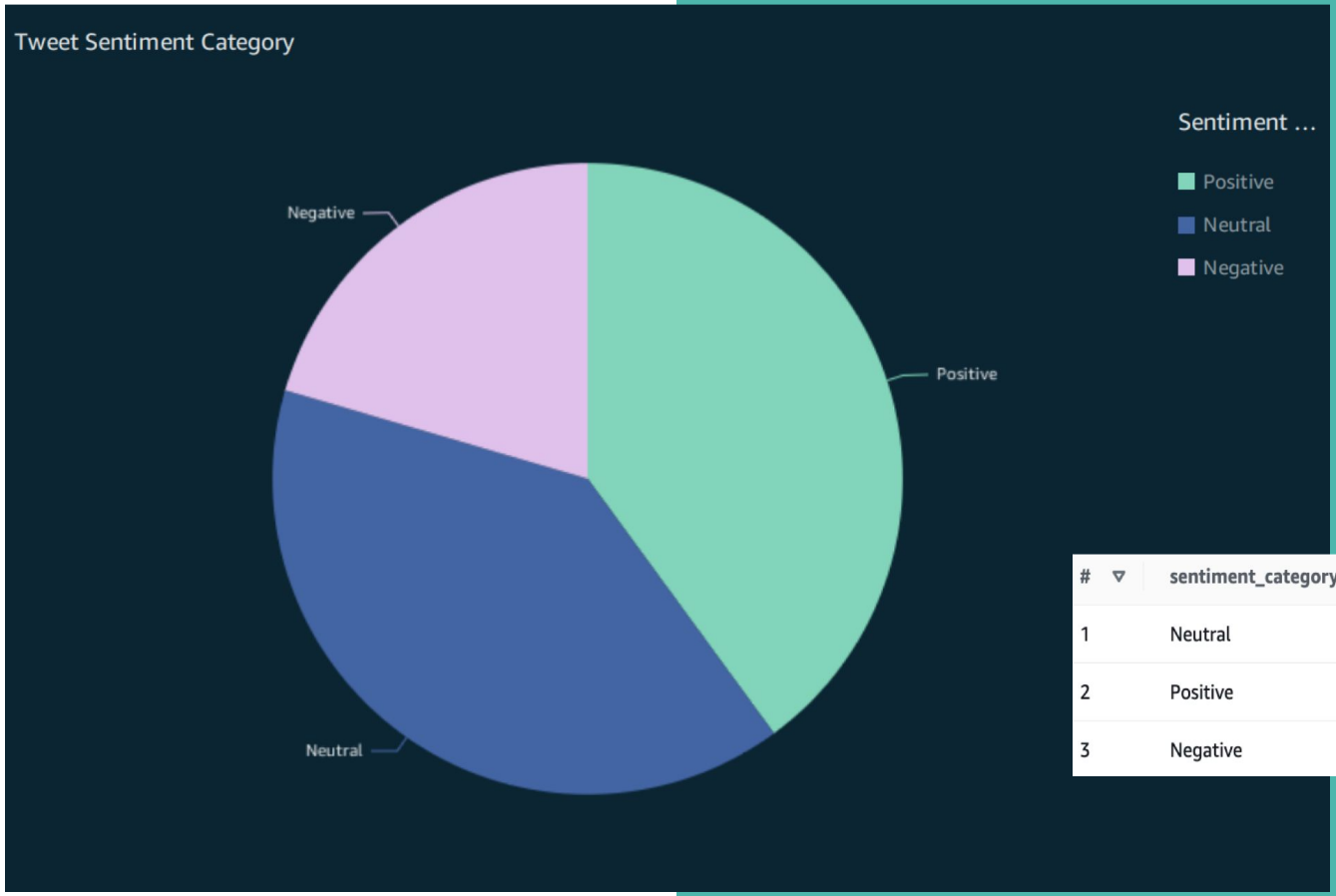
Word cloud with top 50 commonly used words in tweets

No. of tweets accurately predicted v. inaccurately predicted

No. of tweets accurately predicted v. inaccurately predicted by sentiment

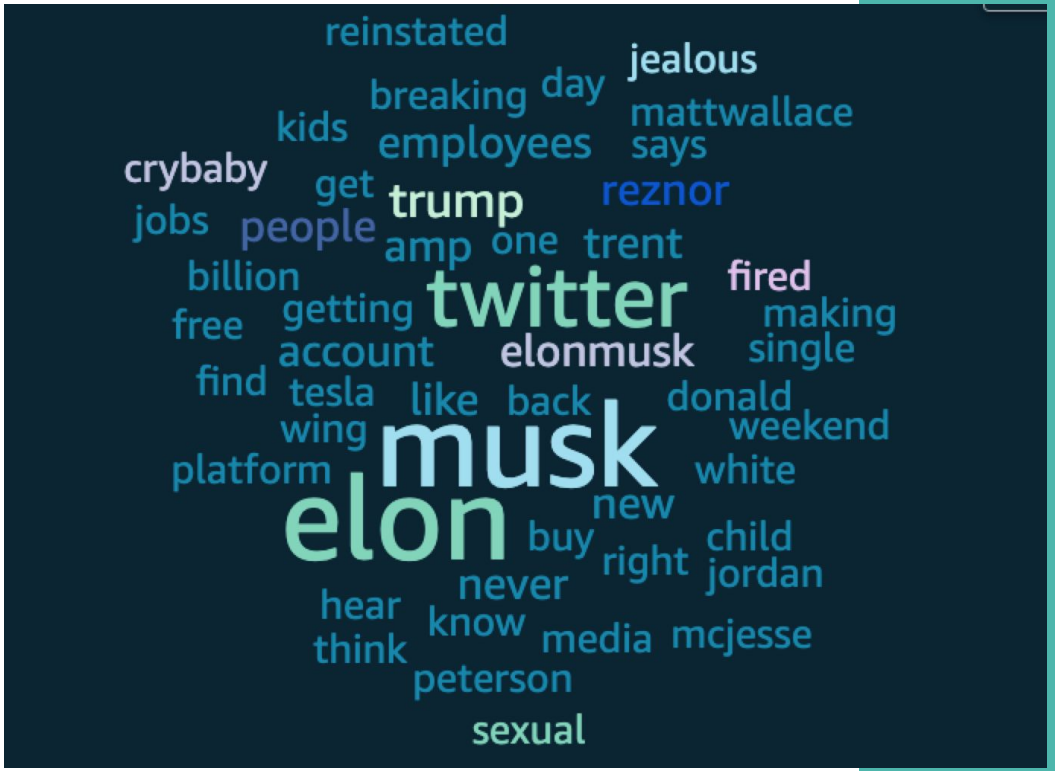
Sentiment of tweets being scraped by hour

Number of Tweets Broken Down by Sentiment



#	sentiment_category	tweet_count
1	Neutral	129941
2	Positive	131055
3	Negative	67195

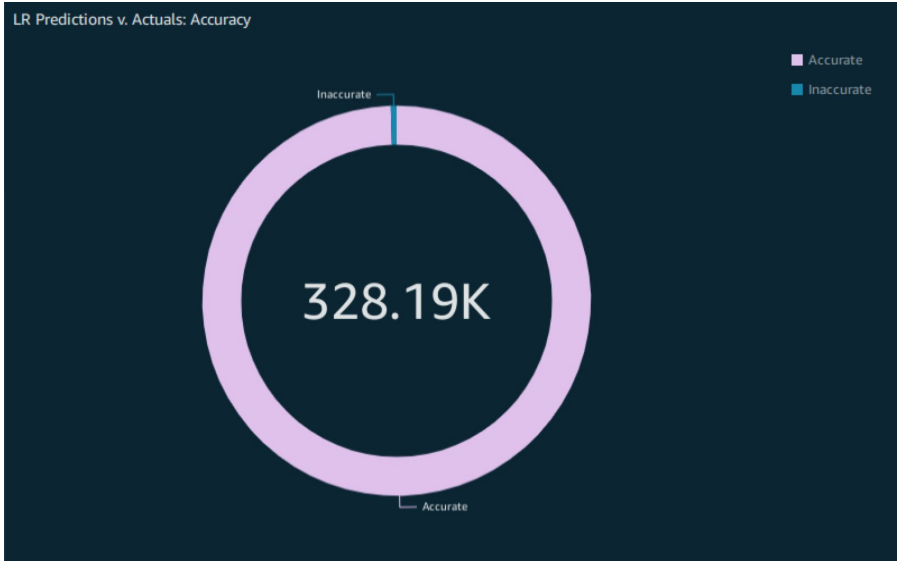
Word Cloud



#	word	word_count
1	elon	252499
2	thread	5583
3	make	4325
4	clear	3004
5	personal	5100
6	kanye	2276
7	freedom	2564
8	disgusting	427

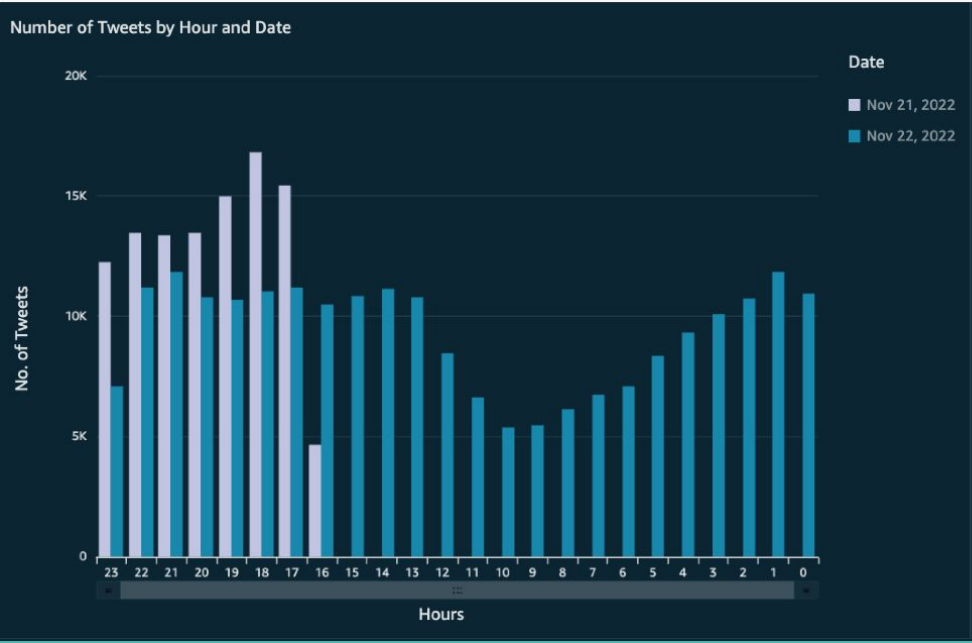
```
--USING FILTERED ARRAY COLUMN FROM PREDICTIONS DATASET TO CREATE A LIST OF WORDS AND THEIR COUNT TO USE FOR WORDCLOUD
CREATE TABLE words AS(
SELECT word, COUNT(word) AS word_count
FROM em_predictions,unnest(filtered) AS t(word) --unnest the array and save each word as a row
GROUP BY word); --group and save total count per word for cloud
```

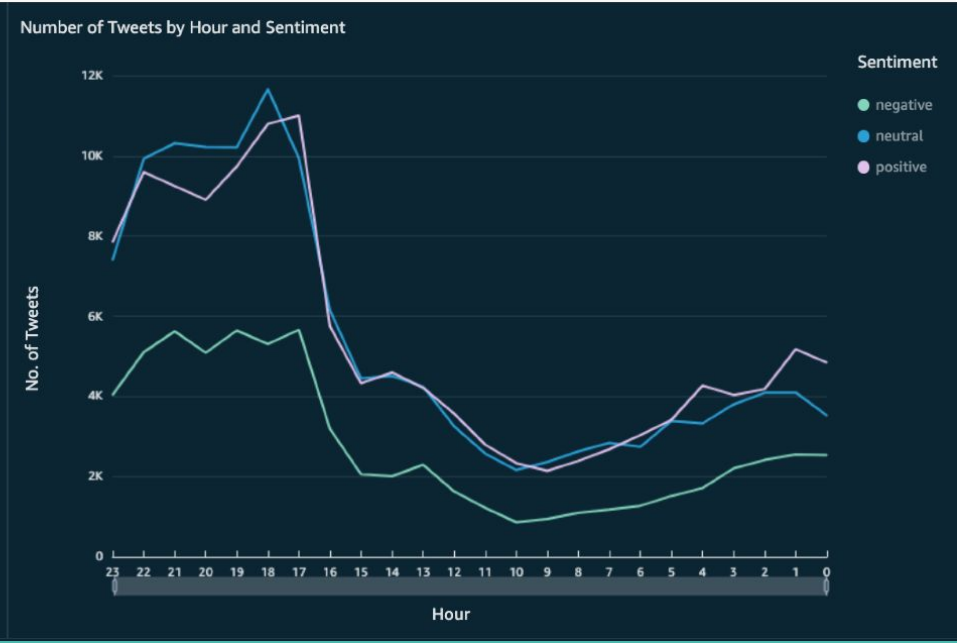
Logistic Regression:
Predictions v. Actuals



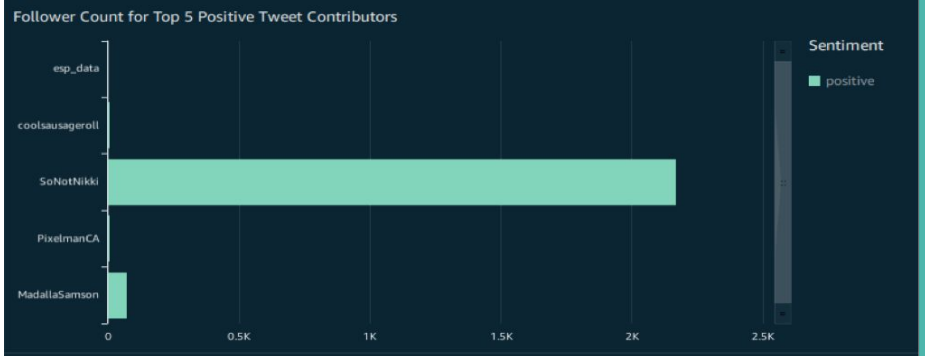

```sql
--DETERMINING WHAT PORTION OF THE PREDICTIONS WERE ACCURATE V. INACCURATE BY SENTIMENT CATEGORY
CREATE TABLE QC_sent AS(
SELECT sentiment_category, QC, COUNT(*) AS accuracy_count FROM(
SELECT sentiment_category, label,prediction,
CASE --case then to label each row as an accurate or inaccurate prediction
    WHEN label-prediction = 0 THEN 'Accurate'
    ELSE 'Inaccurate'
END AS QC
FROM em_predictions)
GROUP BY sentiment_category, QC
ORDER BY sentiment_category);
```

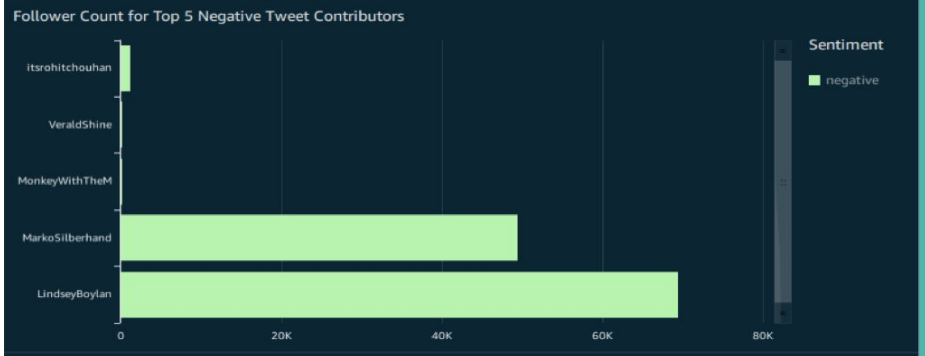
```sql
--CLEANING THE CREATED_AT STRING AND CONVERTING TO DATETIME ATTRIBUTE TO THEN GROUPING THE DATA BY SENTIMENT CATEGORY, DATE AND HOUR
CREATE TABLE datetime_sent as(
SELECT sentiment, new_date, date(new_date) as date, hour(new_date) AS hour, COUNT(*) AS no_tweets FROM(
SELECT created_at, sentiment, date_parse(created_at, '%a %b %d %H:%i:%S +0000 %Y') AS new_date --use date_parse to convert string to datetime
FROM rawdatatable
WHERE created_at not like 'Tue Nov 22 07:38:43 +0000 2022"' AND created_at LIKE '% Nov %') --clean date column and remove all unnecessary garbage values
GROUP BY sentiment, new_date, HOUR(new_date));
```

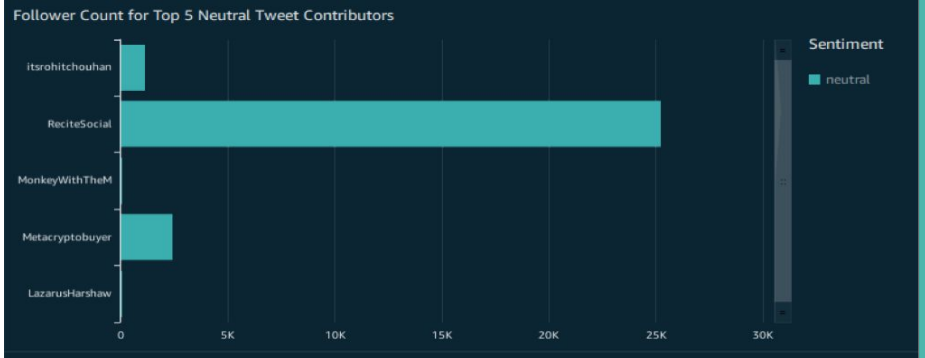




Number of Tweets: Date and Sentiment







Top 5 Handles Contribution by Tweet Sentiment

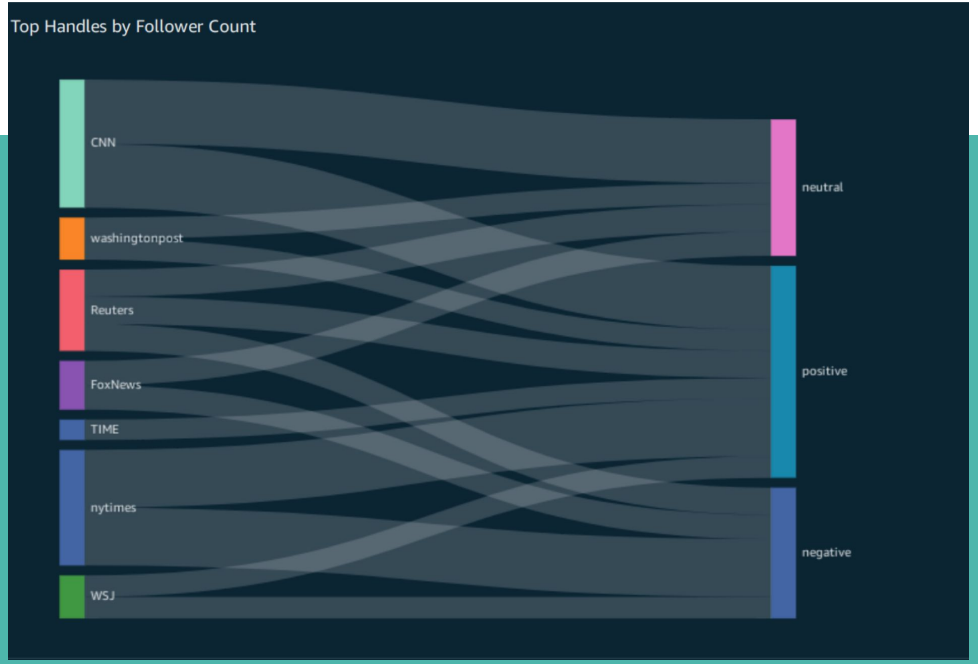
```sql
--TOP 5 HANDLES CONTRIBUTING TO EACH TWEET SENTIMENT AND THEIR FOLLOWER COUNTS
CREATE TABLE top5handles as(
WITH rws AS(
select sentiment, handle, MAX(follower_count) follower_count, COUNT(*) tweets,
row_number () over (PARTITION BY sentiment ORDER BY COUNT(*) DESC) rn
FROM rawdatatable
GROUP BY sentiment, handle)
SELECT * FROM rws
WHERE rn <= 5
ORDER BY sentiment, rn); --use nested selected statement, row number and partition to rank handles within each category by tweet count
```

#	sentiment	handle
1		omgwtfbbqdurian
2	negative	VeraldShine
3	negative	MonkeyWithTheM
4	negative	LindseyBoylan
5	negative	itsrohitchouhan
6	negative	MarkoSilberhand
7	neutral	MonkeyWithTheM
8	neutral	ReciteSocial
9	neutral	itsrohitchouhan
10	neutral	LazarusHarshaw
11	neutral	Metacryptobuyer
12	positive	PixelmanCA
13	positive	SoNotNikki
14	positive	MadallaSamson
15	positive	coolsausageroll
16	positive	esp_data

```sql
--TOP 10 HANDLES BY FOLLOWER COUNT WITHIN EACH SENTIMENT CATEGORY
CREATE TABLE top10handles AS(
SELECT sentiment, handle, follower_count, RANKING
FROM(SELECT sentiment, handle, follower_count, RANK() OVER (PARTITION BY sentiment ORDER BY sentiment, follower_count desc) RANKING
FROM rawdatatable)
WHERE RANKING <=10
ORDER BY sentiment,RANKING DESC); --use nested select statement, rank and partition to rank handles within each category by follower count
```

Top 10 Handles by Follower Count

#	sentiment	handle	follower_count
1		omgwtfbbqdurian	55
2	negative	nytimes	54676071
3	negative	nytimes	54677264
4	negative	nytimes	54677506
5	negative	nytimes	54677542
6	negative	nytimes	54685581
7	neutral	Reuters	25645839
8	neutral	Reuters	25646289
9	neutral	Reuters	25647092
10	neutral	CNN	60582381
11	neutral	CNN	60587470
12	positive	Reuters	25643509
13	positive	Reuters	25643906
14	positive	Reuters	25646699
15	positive	nytimes	54685683
16	positive	CNN	60584683



Challenges

- Computing time
- Cleaning data and garbage values
- Conjoining tables
- Limited knowledge

Thank you.