

# Sentiment Analysis (Elon Musk & Twitter)

...

Rohan Mukerji & Michael Poma

# Executive Summary

- Implement a sentiment analysis framework that analyzes the sentiment of tweets from Elon Musk
  - Analyzing Elon Musk's Sentiments
  - Public Perception
  - Assessing Impact on Stock Market or other markets
- Application of ML/NLP techniques in “real-world” application
  - Individual Impact & Volatility
  - Wide Net of Impact (Stocks, Tech, Cryptocurrency, Politics, etc.)
- Approach is directed, yet complex



# Background

- Major Timeline: January 2022 to November 2022
- Volatility, Humor and Criticism

References:

A timeline of Elon Musk's tumultuous Twitter acquisition (ABC News)



# Technical Challenges

- Tweet Data
- Model Selection
- Data Pre-processing
- Real-time Analysis
- Depth of Exploration



# Implementation

## DATASET

- Musk Tweets downloaded from Kaggle
- Extracted using Tweepy

## HARDWARE / PLATFORM

- Jupyter Notebook, Google Colab

## FRAMEWORK

- Unsupervised and Supervised Learning

## POSITIVES / LIMITATIONS

- Fairly balanced dataset (accuracy)
- Small dataset

## 1) TF-IDF Text Vectorization

- Word must appear in 60+ Tweets
- Removed common stop\_words

## 2) Dimensionality Reduction

- PCA and t-SNE for 2D solution

## 3) Clustering

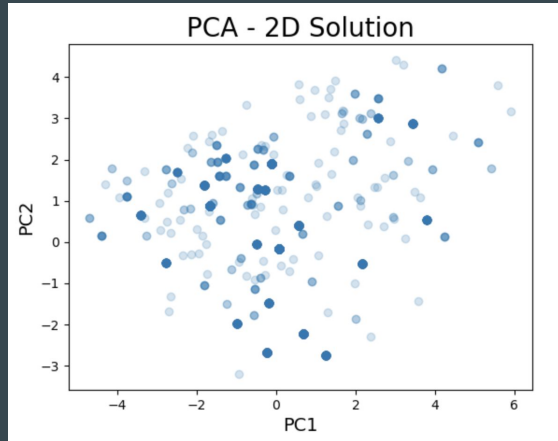
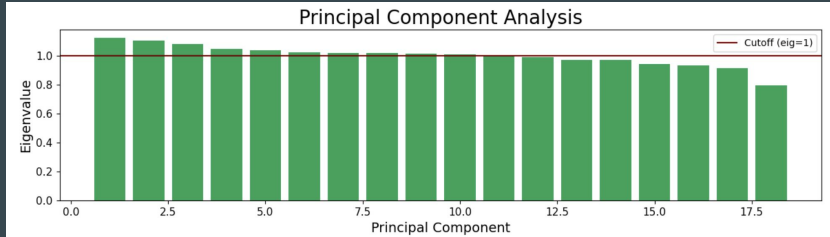
- dbSCAN to identify groups/labels

## 4) Modeling and Tuning

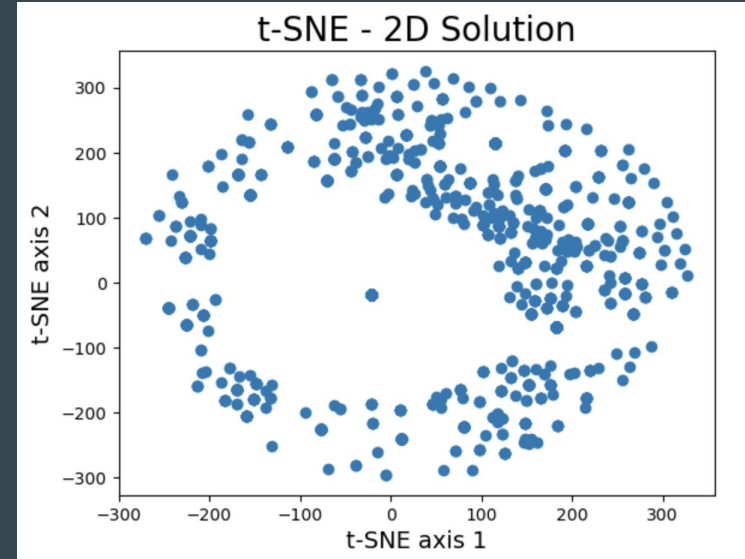
- LR, SVM, Tree, RF, Neural Network
- Hyperparameter Tuned best model

# Experimental Evaluation

## PRINCIPAL COMPONENT ANALYSIS

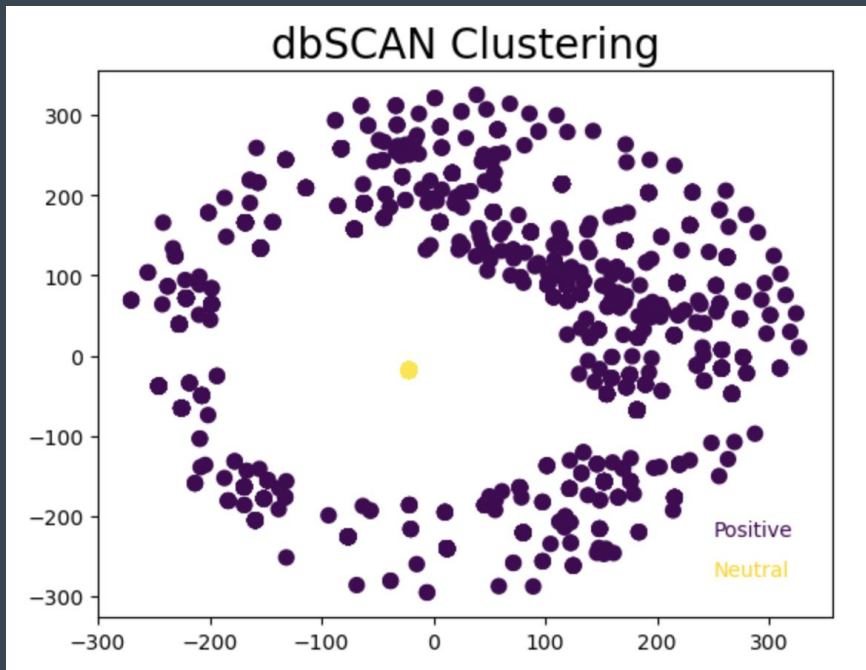


## t-SNE



# Experimental Evaluation

## dbSCAN



## EXAMPLE TWEETS

### Positive Tweets:

- 0 : @SawyerMerritt A picture is worth 1000 tweets
- 1 : @chicago\_glenn We should thank farmers more
- 2 : Thank goodness for Senator Manchin

### Negative Tweets:

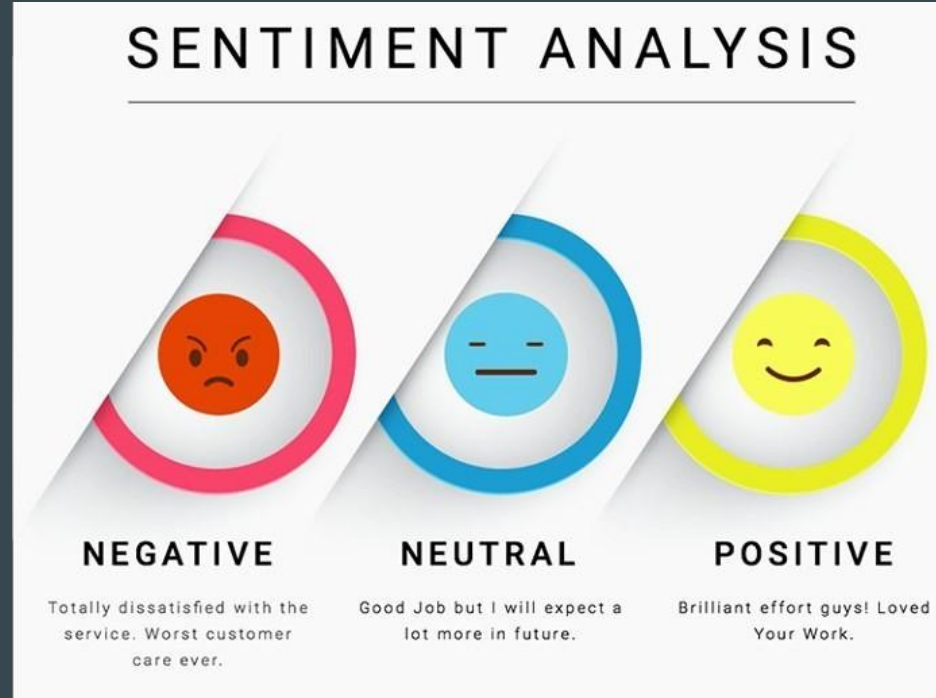
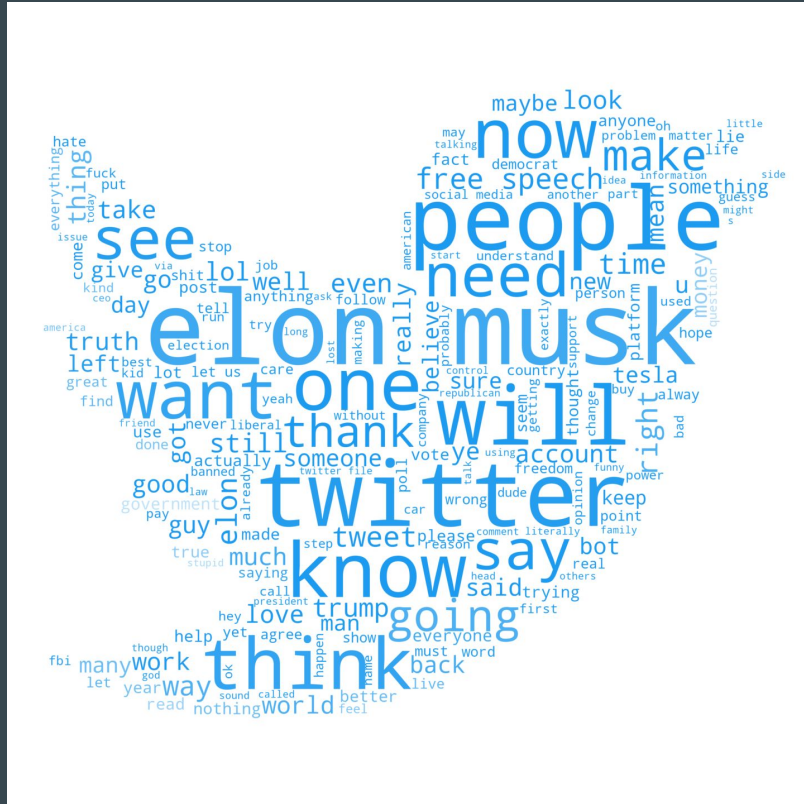
- 0 : @TaraBull808 Sounds pretty bad
- 1 : @RBReich You both an idiot and a liar
- 2 : @stillgray Agreed, that is not ok

## MODEL ACCURACIES

### Accuracy:

Logistic Regression: 0.9355  
SVM: 0.5339  
Decision Tree: 0.8102  
Random Forest: 0.9558  
Base Dense NN: 0.8998

# Conclusion





Github Link