Tweets’ Sentimental Analysis with Natural Language Processing

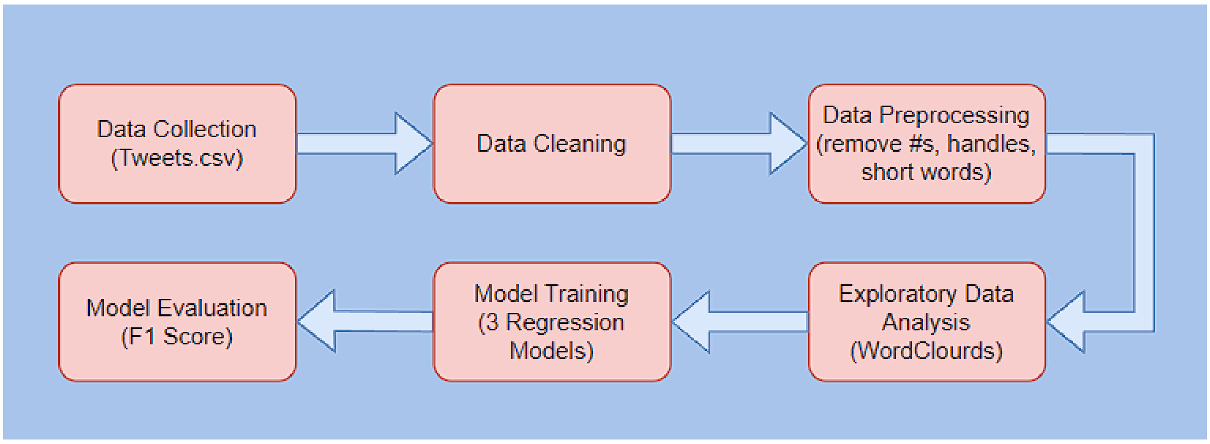
Introduction: A Twitter sentiment analysis is the process of determining the emotional tone behind a series of words, specifically on Twitter. The objective of this task is to detect hate speech in tweets. A Twitter sentiment analysis identifies negative, positive, emotions within the text of a tweet. It is a text analysis using natural language processing (NLP) and machine learning.

Problem Statement: Natural Language Processing (NLP) is a hotbed of research in data science these days and one of the most common applications of NLP is sentiment analysis. From opinion polls to creating entire marketing strategies, this domain has completely reshaped the way businesses work, which is why this is an area every data scientist must be familiar with.

Thousands of text documents can be processed for sentiment (and other features including named entities, topics, themes, etc.) in seconds, compared to the hours it would take a team of people to manually complete the same task.

We will do so by following a sequence of steps needed to solve a general sentiment analysis problem. We will start with pre-processing and cleaning of the raw text of the tweets. Then we will explore the cleaned text and try to get some intuition about the context of the tweets. After that, we will extract numerical features from the data and finally use these feature sets to train models and identify the sentiments of the tweets.

Workflow


Data Collection (Tweets.csv)
Data Cleaning
Data Preprocessing (remove #s, handles, short words)
Exploratory Data Analysis (WordClourds)
Model Training (3 Regression Models)
Model Evaluation (F1 Score)

Dataset: The dataset used for the Twitter sentiment analysis AI model consists of tweets. The tweets will be pre-processed using standard techniques such as tokenization and stemming.

The dataset has been divided into 2 files:

→ train.csv-The Training Dataset comprises of 75% of our twitter dataset.

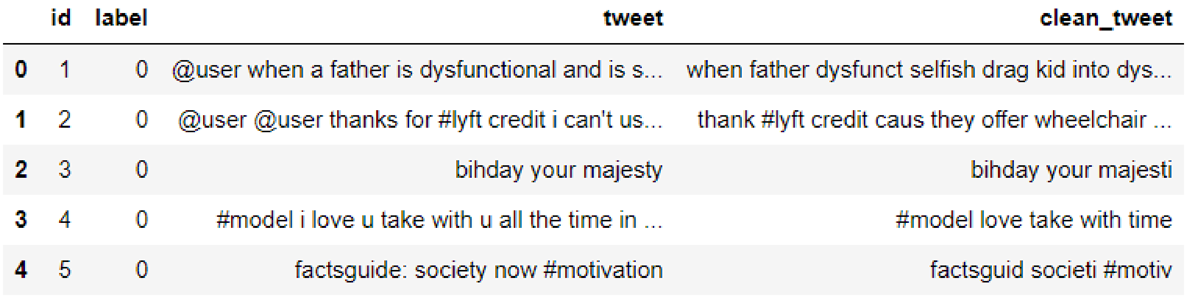
→ test.csv- The Test Dataset comprises of 25% of our twitter dataset.

Formally, given a training sample of tweets and labels, where label '1' denotes the tweet is racist/sexist and label '0' denotes the tweet is not racist/sexist, our objective is to predict the labels on the test dataset.

For training the models, we have used a labelled dataset of 31,962 tweets. The dataset is provided in the form of a csv file with each line storing its label and the tweet.

Methodology: Our Twitter sentiment analysis AI model follows a well-defined work pipeline that consists of the following steps:

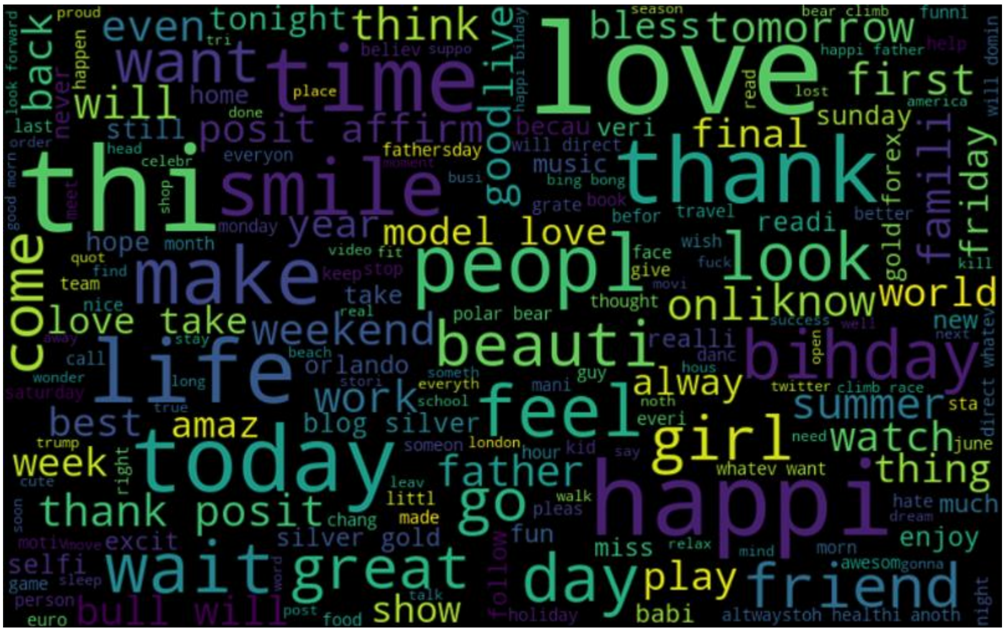
1. Data collection from various sources including Twitter API, web scraping, and manual annotation.
2. Data pre-processing including text cleaning, tokenization, stop-word removal, special characters & numbers removal, stemming, and feature extraction.



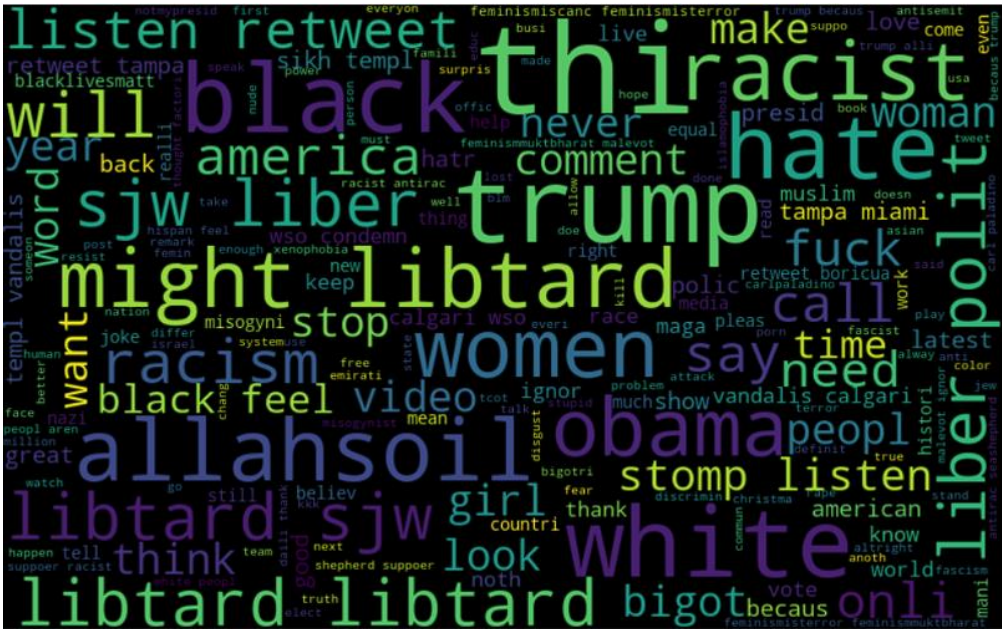
	id	label	tweet	clean_tweet
0	1	0	@user when a father is dysfunctional and is s...	when father dysfunct selfish drag kid into dys...
1	2	0	@user @user thanks for #lyft credit i can't us...	thank #lyft credit caus they offer wheelchair ...
2	3	0	bihday your majesty	bihday your majesti
3	4	0	#model i love u take with u all the time in ...	#model love take with time
4	5	0	factsguide: society now #motivation	factsguid societi #motiv

Pre-processed/Clean Tweets

3. Exploration: We have used word clouds and graphs to retrieve information regarding key word used in tweets based they have been classified as negative or positive.

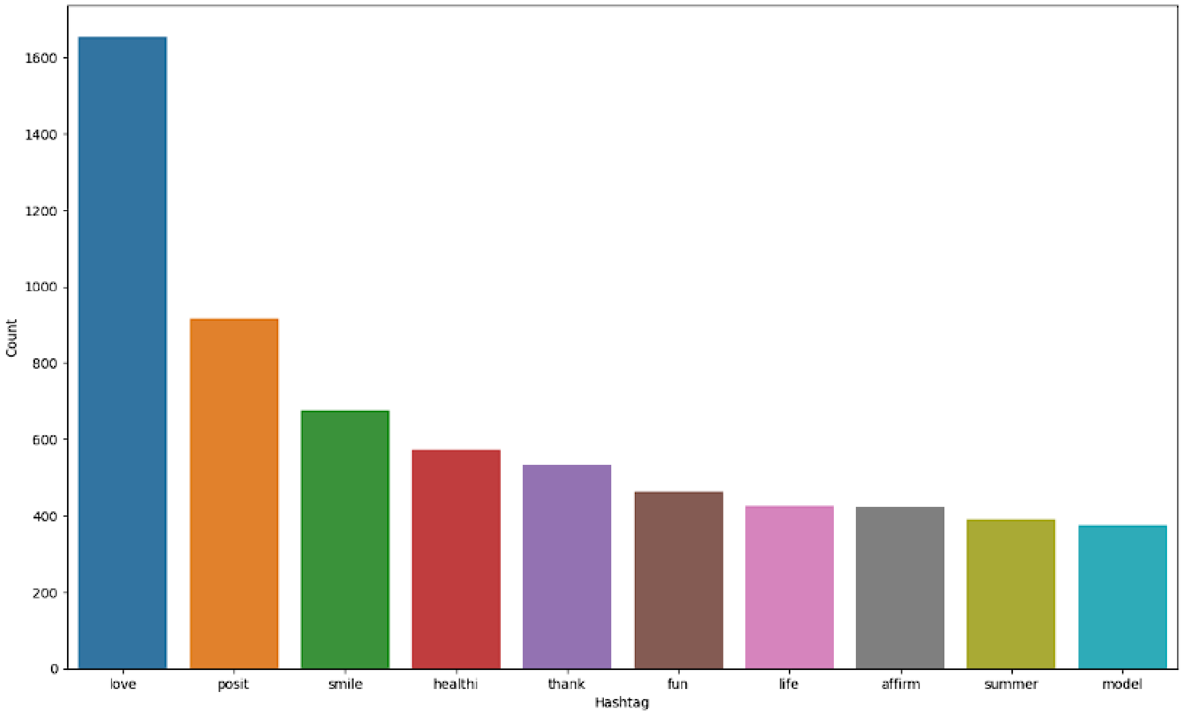


Positive Word Cloud

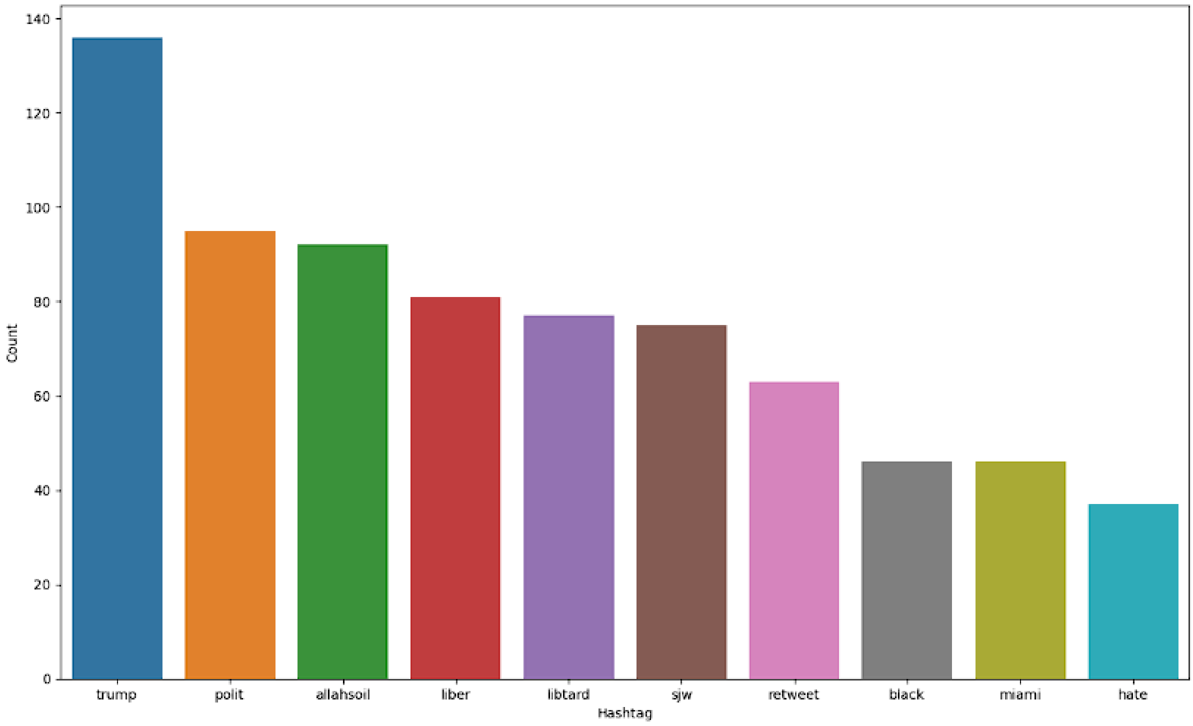


Negative Word Cloud

Visualization of top 10 hashtags used sentiment wise:



Positive Hashtags Barplot

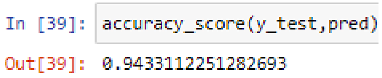


Negative Hashtags Barplot

4. Model architecture selection and implementation based on the specific problem statement and dataset characteristics.

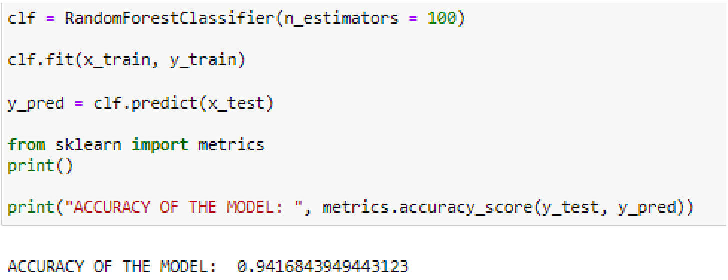
5. Model training using various machine learning algorithms and techniques such as supervised and unsupervised learning, deep learning, and transfer learning.
6. Evaluation of the model's performance using metrics such as accuracy, F1-score:

Accuracy with simple linear regression model:



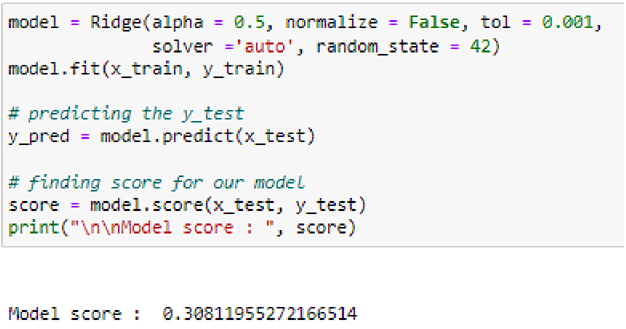
```
In [39]: accuracy_score(y_test,pred)
Out[39]: 0.9433112251282693
```

Accuracy with random forest regression model:



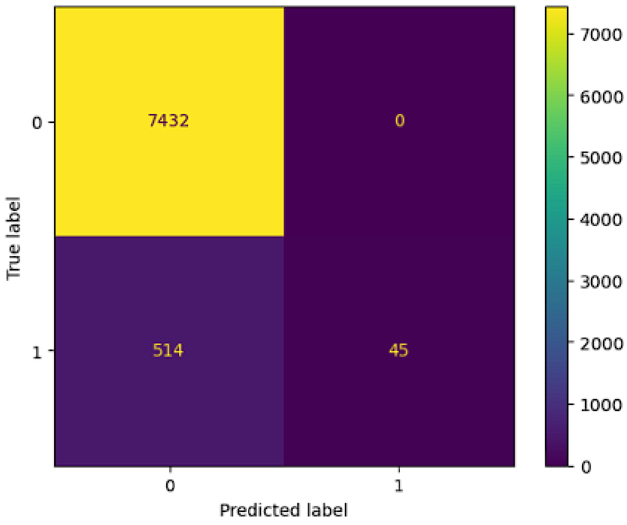
```
clf = RandomForestClassifier(n_estimators = 100)

clf.fit(x_train, y_train)

y_pred = clf.predict(x_test)

from sklearn import metrics
print()

print("ACCURACY OF THE MODEL: ", metrics.accuracy_score(y_test, y_pred))

ACCURACY OF THE MODEL:  0.9416843949443123
```

Accuracy with ridge regression model:



```
model = Ridge(alpha = 0.5, normalize = False, tol = 0.001,
              solver ='auto', random_state = 42)
model.fit(x_train, y_train)

# predicting the y_test
y_pred = model.predict(x_test)

# finding score for our model
score = model.score(x_test, y_test)
print("\n\nModel score : ", score)


Model score :  0.30811955272166514
```

Confusion Matrix for the model:



7. Real-world applications of the model in various domains such as marketing, politics, finance, and healthcare.

Outcome: With the available csv as our dataset we have create a regression model using three different regression training techniques:

- Simple Linear Regression Model
- Random Forest Regression Model
- Ridge Regression Model

The accuracy of our model was measured at 94%, indicating that it was able to accurately classify tweets as positive or sentiment.

```
Accuracy: 0.9409335502440245
Model score:  0.30811955272166514
F1 Score: 0.5545722713864307
```

Our F1-score was 0.55, which is a measure of the model's overall accuracy and balance between precision and recall.

Future Prospects: The model we have created is not perfect but it’s completely genuine and authentic and solely created by the team.

- To add multithreading
- To add more models
- To add more detailed sentimental analysis with 4 emotions

Created Solely by:

Yash Rajoria (102103526)

Varada Gupta (102103542)