
Twitter
Sentiment
&
Emotion
Analysis
Group number:- 01

Group Members

Name	Roll Number
Prathamesh Chikane	06
Subodh Halpatrao	18
Sutej Kulkarni	35

What is Sentiment Analysis?

It is classification of the polarity of a given text in the document, sentence or a phrase

The goal is to determine whether the expressed opinion in the text is positive, neutral or negative.

Sentiment Analysis

Tweet – unemployement rate is decreasing as economic recovery takes place

Sentiment – Positive

Tweet - Stay home everyone, to ensure safety

Sentiment - Neutral

Tweet - More people are dying due to the pandemic

Sentiment - Negative

What is Emotion Analysis

Emotions are described as intense feelings that are directed at something or someone in response to internal or external events having a particular significance for the individual. And the internet, today, has become a key medium through which people express their emotions, feelings and opinions

Every event, news or activity around the world, is shared, discussed, posted and commented on social media, by millions of people.

The goal is to determine whether the tweet or sentence is a worry, happy, love or hate based tweet

Why is Sentiment Analysis Important ?

- Microblogging has become popular communication tool
- Opinion of the mass is important
- Political party may want to know whether people support their program or not.
- Before investing into a company, one can leverage the sentiment of the people for the company to find out where it stands.
- A company might want to find out the reviews of its products

Why we need Emotion Analysis ?

There are plenty of research works that have focused on Sentiment Analysis and provide a 2-way or a 3-way classification of text.

But in recent times there has been more and more types of specifications and classifications needed to get a more specialized result

Capturing the emotions in text, especially those posted or circulated on social media, can be a source of precious information, which can be used to study how different people react to different situations and events.

Using Twitter for Sentiment & Emotion Analysis

- Popular microblogging site
- 240+ million active users
- Twitter audience varies from common people to celebrities
- Users often discuss current affairs and share personal views on various subjects
- Tweets are small in length and hence unambiguous

Problem Statement

The problem statement at hand consists of two subtasks:

- Phrase Level Sentiment and Emotion Analysis in Twitter:

Given a message containing a marked instance of a word or a phrase, determine whether that instance is positive, negative or neutral in that context and whether its worry, happy, love and hate in emotion

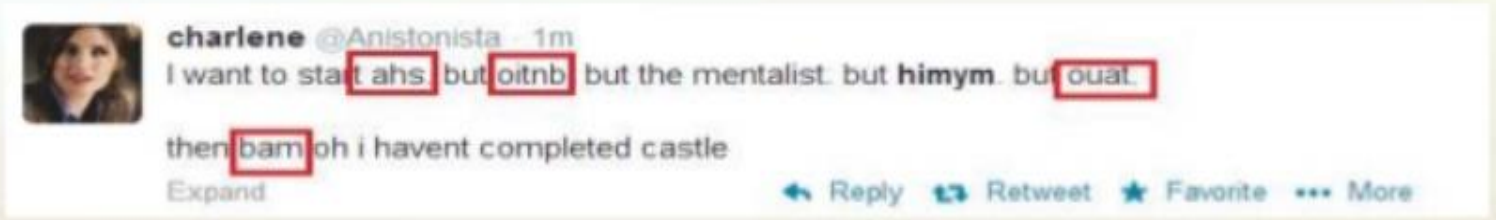
- Sentence Level Sentiment and Emotion Analysis in Twitter:

Given a message, decide whether the message is of positive, negative or neutral sentiment or a happy, worry, love or hate in emotion. For messages conveying mixed sentiment and emotion, whichever is the stronger sentiment should be chosen

Challenges

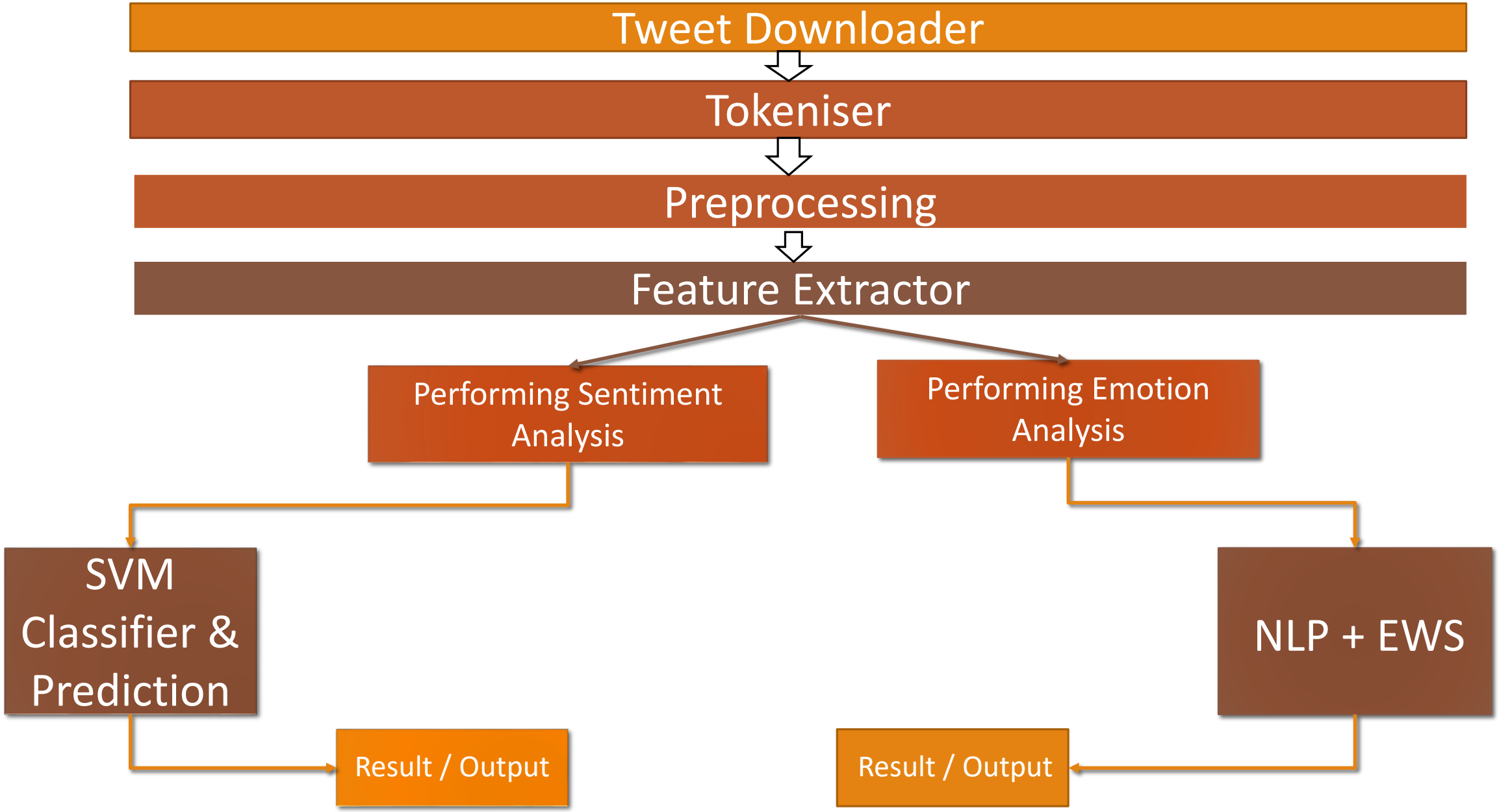
- Tweets are highly unstructured and also non-grammatical

- Out of Vocabulary Words



- Lexical Variation

- Extensive usage of acronyms like *asap, lol, afaik*


Tweet Downloader
Tokeniser
Preprocessing
Feature Extractor
Performing Sentiment Analysis
Performing Emotion Analysis
SVM Classifier & Prediction
NLP + EWS
Result / Output
Result / Output

Approach

Tweet Downloader:- Download the tweets using Twitter API

Tokenisation:- Twitter specific POS tagger developed by Social Media Search

Preprocessing :- Removing Non-English Tweets, Replacing Emoticons by their polarity, Remove URL, Target Mentions, Hashtags.
Remove Nouns and Prepositions, Replace sequence of repeated characters.

Approach

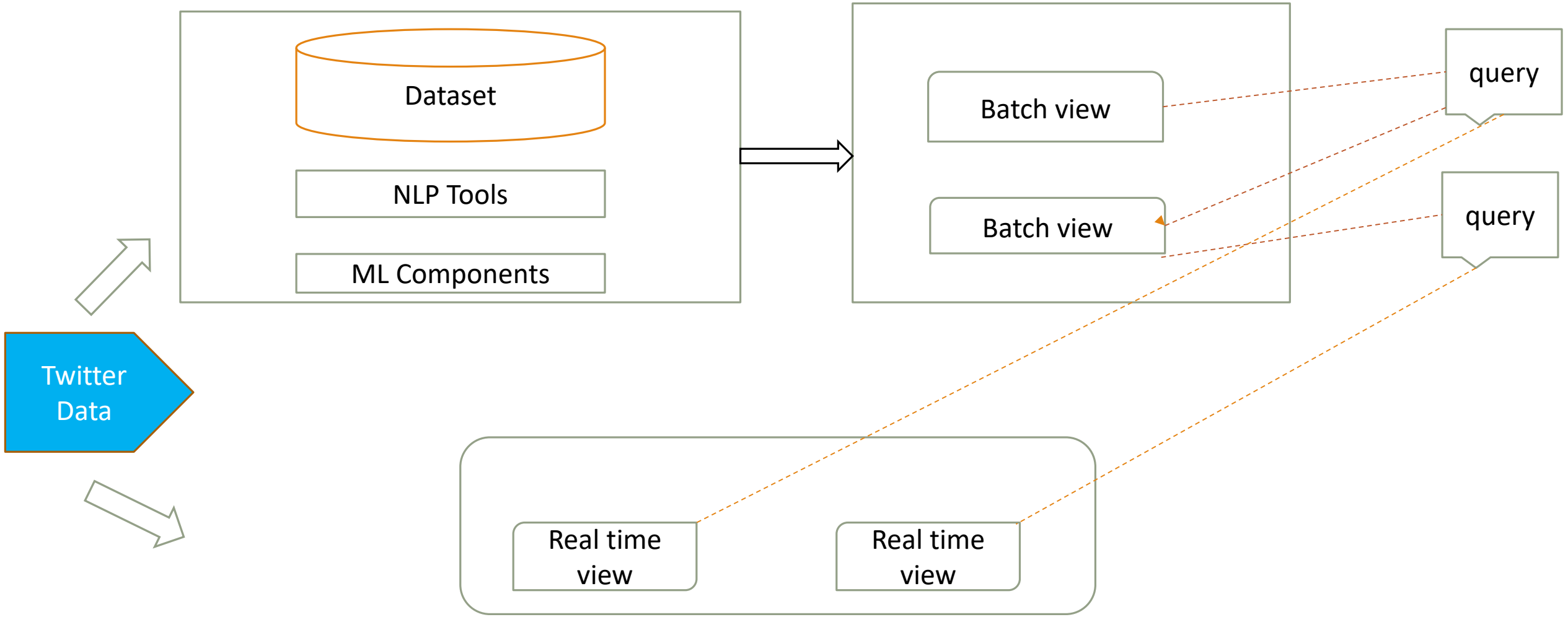
Feature Extractor
> Polarity Score of the tweet
> Percentage of capitalised words
> Number of hashtags
> Number of negations

Classifier and Prediction

The Feature extracted are next passed on to SVM classifier for Sentiment Analysis and to the NLP model which carries on the Emotion Analysis

The Model built is used to predict the sentiment and emotion of the tweets.

Design Plan


Dataset
NLP Tools
ML Components
Batch view
Batch view
query
query
Twitter Data
Real time view
Real time view

Implementation Plan

Here’s what our Implementation plan will look like:

- Gather real-time live tweets from Twitter
- Preprocessing (stopword removal)
- Applying the SVM algorithm for Sentiment Analysis & NLP on Emotion Word Set (EWS) for Emotion Analysis
- Analyze the results
- Discuss further improvement and next steps

Extracting Tweets

We’ll start by grabbing the tweets we want from Twitter, In one of the later stages, we will be extracting numeric features from our Twitter text data. This feature space is created using all the unique words present in the entire data. So, if we preprocess our data well, then we would be able to get a better quality feature space.

Pre-processing Tweets

The preprocessing of the text data is an essential step as it makes the raw text ready for mining, i.e., it becomes easier to extract information from the text and apply machine learning algorithms to it. If we skip this step then there is a higher chance that you are working with noisy and inconsistent data. The objective of this step is to clean noise those are less relevant to find the sentiment of tweets such as punctuation, special characters, numbers, and terms which don't carry much weightage in context to the text.

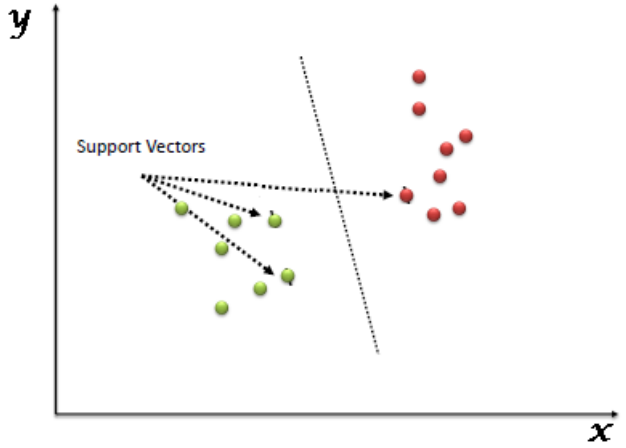
Stopwords are words that aren't integral to the meaning of a text, and are usually removed as part of a Natural Language Processing workflow. The tweets are now reassembled as sentences, but without stopwords removing these extra elements should give the sentiment analysis algorithm a better shot.

Applying the algorithm for sentiment analysis

Choosing which sentiment algorithm to use depends on a number of factors: you need to take into account the required level of detail, speed, cost, and accuracy among other things, Here we have selected the Support Vector Machine algorithm

Support Vector Machine (SVM)

"Support Vector Machine" (SVM) is a supervised machine learning algorithm which can be used for both classification or regression challenges. However, it is mostly used in classification problems. In the SVM algorithm, we plot each data item as a point in n-dimensional space (where n is number of features you have) with the value of each feature being the value of a particular coordinate. Then, we perform classification by finding the hyper-plane that differentiates the two classes very well



How SVM Works?

Unlike other classifiers, the support vector machine is explicitly told to find the best separating line. How? The support vector machine searches for the closest points, which it calls the "support vectors" (the name "support vector machine" is due to the fact that points are like vectors and that the best line "depends on" or is "supported by" the closest points).

Once it has found the closest points, the SVM draws a line connecting them. It draws this connecting line by doing vector subtraction (point A - point B). The support vector machine then declares the best separating line to be the line that bisects -- and is perpendicular to -- the connecting line.

The support vector machine is better because when you get a new sample (new points), you will have already made a line that keeps B and A as far away from each other as possible, and so it is less likely that one will spillover across the line into the other's territory.

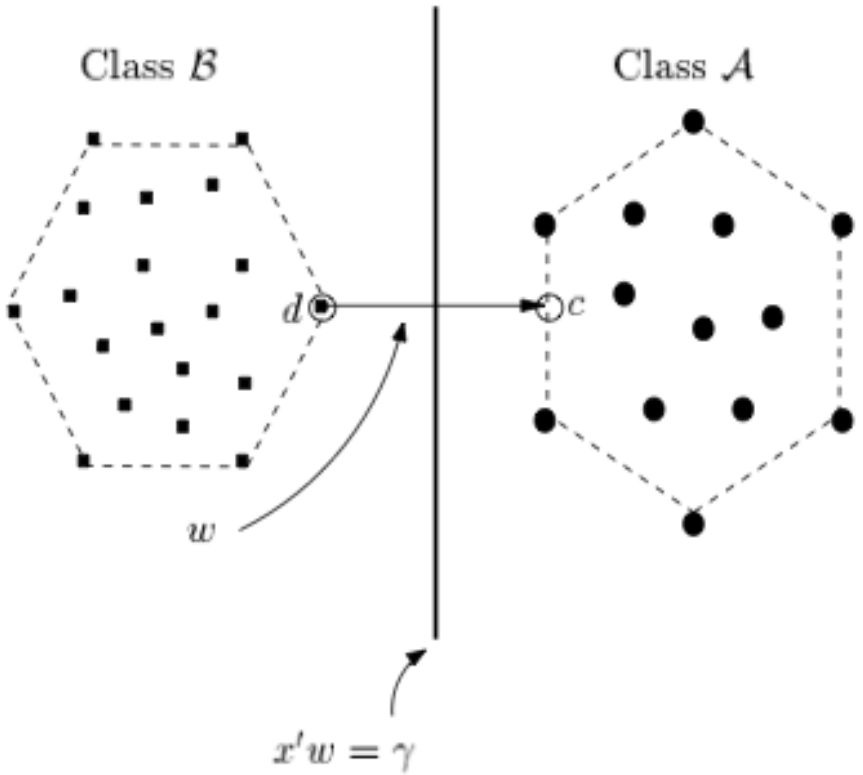
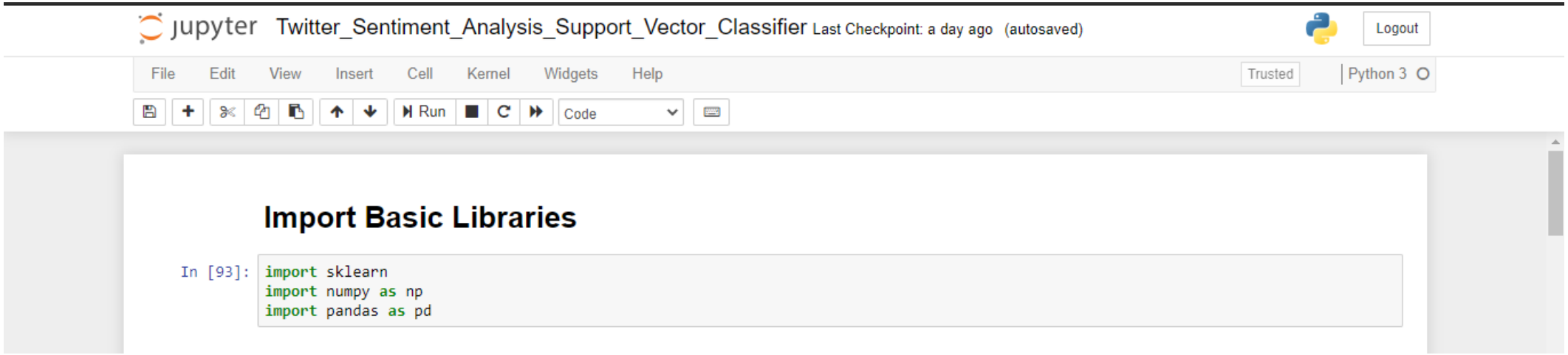


Figure 2. The two closest points of the convex hulls determine the separating plane.

SVM Implementation



Data cleaning

```
#install tweet-preprocessor to clean tweets
!pip install tweet-preprocessor
```

```
Requirement already satisfied: tweet-preprocessor in c:\users\kulka\anaconda3\lib\site-packages (0.6.0)
```

```
distributed 1.21.8 requires msgpack, which is not installed.
You are using pip version 10.0.1, however version 20.3.3 is available.
You should consider upgrading via the 'python -m pip install --upgrade pip' command.
```

```
# remove special characters using the regular expression library
import re

#set up punctuations we want to be replaced
REPLACE_NO_SPACE = re.compile("(\.)|(\;)|(\:)|(\!)|(\')|(\?)|(\,)|(\")|(\|)|(\()|(\))|(\[)|(\])|(\%)|(\$)|(\>)|(\<)|(\{)|(\})")
REPLACE_WITH_SPACE = re.compile("(<br\s/><br\s/?)|(-)|(/)|(:).")
```

```
import preprocessor as p

# custum function to clean the dataset (combining tweet_preprocessor and reguar expression)
def clean_tweets(df):
  tempArr = []
  for line in df:
    # send to tweet_processor
    tmpL = p.clean(line)
    # remove puctuation
    tmpL = REPLACE_NO_SPACE.sub("", tmpL.lower()) # convert all tweets to lower cases
    tmpL = REPLACE_WITH_SPACE.sub(" ", tmpL)
    tempArr.append(tmpL)
  return tempArr
```

```
# clean training data
train_tweet = clean_tweets(train["tweet"])
train_tweet = pd.DataFrame(train_tweet)
```

```
# append cleaned tweets to the training data
train["clean_tweet"] = train_tweet

# compare the cleaned and uncleaned tweets
train.head(5)
```

	id	label	tweet	clean_tweet
0	1	0	@user when a father is dysfunctional and is s...	when a father is dysfunctional and is so selfi...
1	2	0	@user @user thanks for #lyft credit i can't us...	thanks for credit i cant use cause they dont o...
2	3	0	bihday your majesty	bihday your majesty
3	4	0	#model i love u take with u all the time in ...	i love u take with u all the time in ur
4	5	0	factsguide: society now #motivation	factsguide society now

Test and Train split

```
from sklearn.model_selection import train_test_split

# extract the labels from the train data
y = train.label.values

# use 70% for the training and 30% for the test
x_train, x_test, y_train, y_test = train_test_split(train.clean_tweet.values, y,
                                                    stratify=y,
                                                    random_state=1,
                                                    test_size=0.3, shuffle=True)
```

jupyter Twitter_Sentiment_Analysis_Support_Vector_Classifier Last Checkpoint: a day ago (autosaved)

File Edit View Insert Cell Kernel Widgets Help

Run Code

Vectorize tweets using CountVectorizer

CountVectorizer Example

In [107]:
```
from sklearn.feature_extraction.text import CountVectorizer
```

In [114]:
```
documents = ["Data science is my passion!"]

# initializing the countvectorizer
vectorizer = CountVectorizer()

# tokenize and make the document into a matrix
document_term_matrix = vectorizer.fit_transform(documents)

# check the result
pd.DataFrame(document_term_matrix.toarray(), columns = vectorizer.get_feature_names())
```

Out[114]:

	data	is	my	passion	science
0	1	1	1	1	1

Model building

Apply Support Vetor Machine (SVM)

```
from sklearn import svm
# classify using support vector machine
svm = svm.SVC(kernel = 'linear', probability=True)

# fit the SVC model based on the given training data
prob = svm.fit(x_train_vec, y_train).predict_proba(x_test_vec)

# perform classification and prediction on samples in x_test
y_pred_svm = svm.predict(x_test_vec)
```

Accuracy score for SVM

```
]: from sklearn.metrics import accuracy_score
   print("Accuracy score for SVM is: ", accuracy_score(y_test, y_pred_svm) * 100, '%')
```

```
Accuracy score for SVM is:  94.86912086766085 %
```

Working of Emotion Analysis Model

Our Emotion analysis is based on Natural Language Processing, and uses several textual features like emoticons, degree words and negations, Parts Of Speech and other grammatical analysis.

Natural Language Toolkit (nltk) which is used for building programs that work with human language data for applying in statistical natural language processing (NLP). It contains text processing libraries for tokenization, parsing, classification, stemming, tagging and semantic reasoning

In this approach we use the tweets in Tweets Set as well as from the live API data

Approach

pos: Parts of Speech (noun, verb, adjective, for example)

lemma: Lemmatized version of that word

Next, the annotated (and filtered) tokens are matched against the words present in the EWS. But first, all the words in the EWS are lemmatized to their base form. Eg. “happiness” and “happily” are changed to “happy”. While matching the tokens against the EWS, only the tokens that are annotated as “O” (the other entity in Named Entity Recognition) are considered, because a named entity (location, time or a person word, for example) can never be an Emotion-word.

A matched token along with all its characteristics/annotations is stored as a hit, Every hit is finally a tuple of its features that contribute in the process.

Finally using the Natural language toolkit (nltk) and textblob we then classify the lemmatized sentence containing the token into Worry, Happy, Sad, Love & Hate whichever emotion it carries.

Hosting on AWS

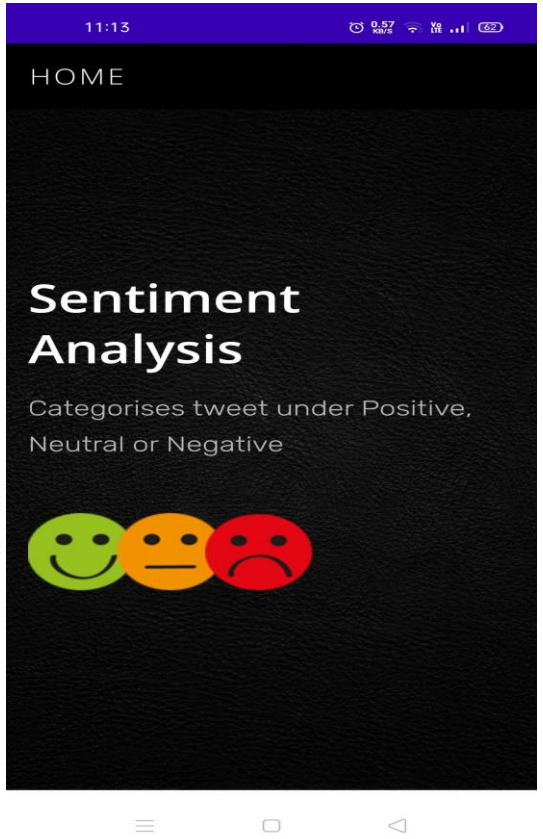
When compared to traditional web hosting services, AWS comes out as the clear winner in terms of user understandability. AWS makes sure that a user manages and maintains their website without any hassles.

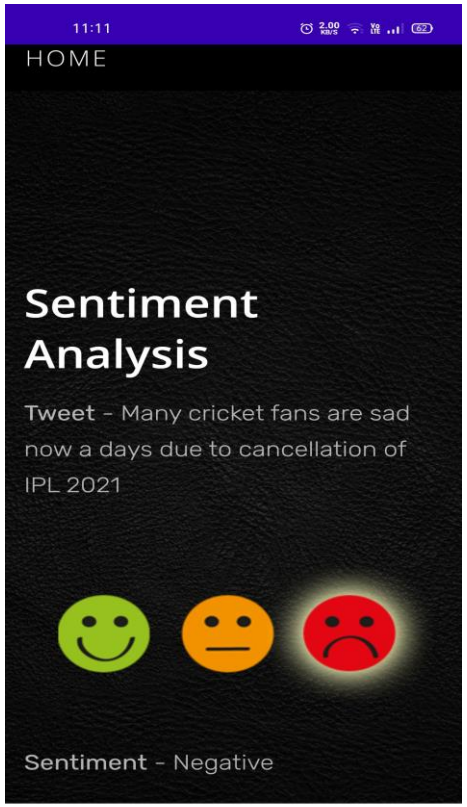
It is designed to make web-scale cloud computing easier for developers and allows maximum scalability and availability for websites and web applications.

Developing the Android App

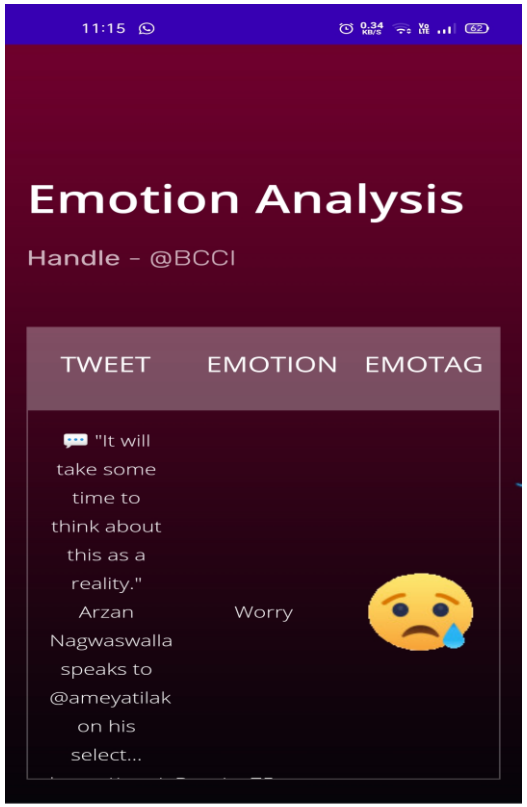
- After successful hosting of our website on AWS, we get an URL of our website to make an android application.
- With the help of webview in android and website's IP address we built an Android app of our website successfully.
- Also we implement an animation as splash screen while starting the application.

Android App Snapshots




11:11
HOME
Sentiment Analysis
Tweet - Many cricket fans are sad now a days due to cancellation of IPL 2021
Sentiment - Negative


11:15
HOME
Emotion Analysis
Categorises tweet under Love, Happiness, Worry, Sadness or Hate
@BCCI
SEARCH


11:15
Emotion Analysis
Handle - @BCCI
TWEET
EMOTION
EMOTAG
"It will take some time to think about this as a reality." Arzan Nagwaswalla speaks to @ameyatilak on his select...
Worry

Further Improvements and Next Steps

There are a few ways we can improve this pipeline to make it look more like something production ready.

Use Geolocation and Google Maps API to get location wise sentiments and emotion regarding the sentence or tweet

Using other Social Media Platform's API and integrating all the combined results

CONCLUSION

We have successfully designed and developed a sentiment and emotion analysis model using Machine learning algorithms and have hosted the project on amazon web services (aws), Also an Android App of our web model is being developed using Android Studio.