Twitter Sentiment Analysis

(Sentiment140 dataset with 1.6 million tweets)

Group K

RAJ KALPESH SANGHAVI
XIAO WU

Introduction

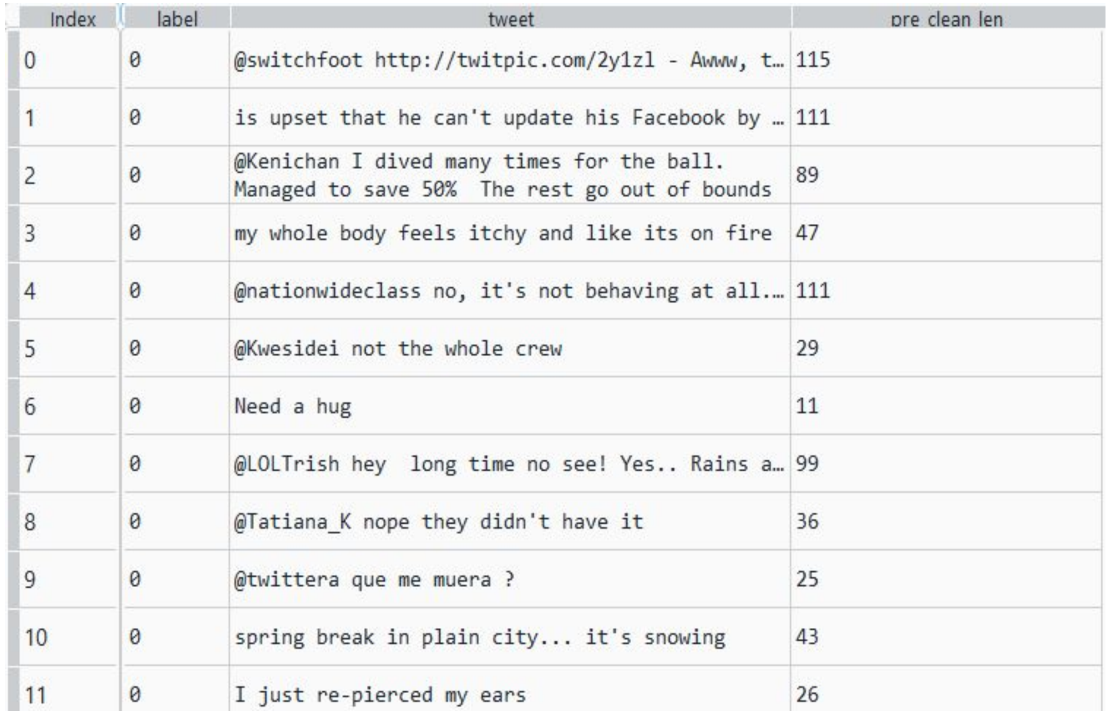
Sentimental Analysis is a process of ‘computationally’ determining whether a piece of writing is positive or negative.

In this project we will be using twitter sentiment data to analyse emotions based on the text and using natural language processing. Our database includes 1.6 million tweets and each one is labeled as negative or positive. (99578 negative tweets and 99513 positive tweets). We Split the training set and the test set in 80/20 ratio.

Steps Involved In Classifying Setimental Analysis

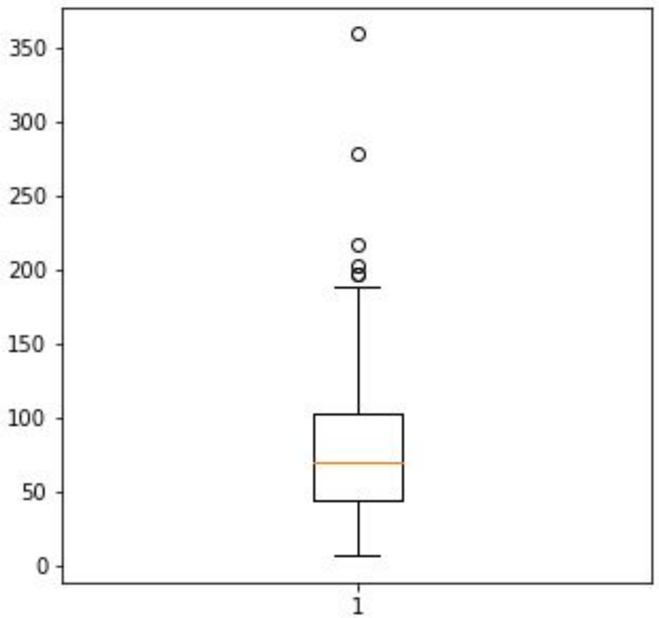
- Dataset Observation
- Data Cleaning
- Exploratory Analysis
- Data Preprocessing
- Classifying tweets to negative and positive words using various Machine Learning algorithms.

Data Observations

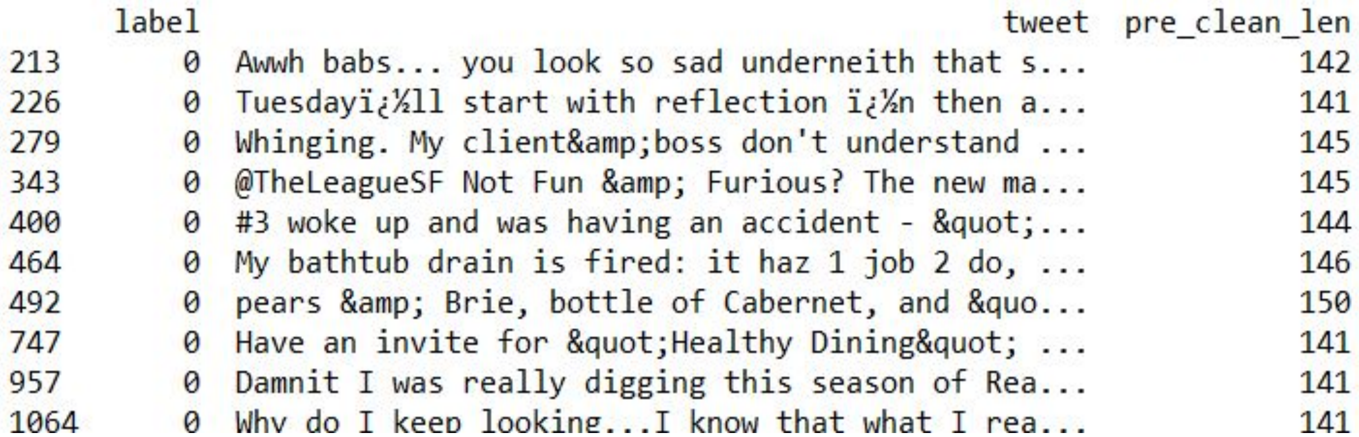


Index	label	tweet	pre clean len
0	0	@switchfoot http://twitpic.com/2y1zl - Awww, t…	115
1	0	is upset that he can't update his Facebook by …	111
2	0	@Kenichan I dived many times for the ball. Managed to save 50% The rest go out of bounds	89
3	0	my whole body feels itchy and like its on fire	47
4	0	@nationwideclass no, it's not behaving at all.…	111
5	0	@Kwesidei not the whole crew	29
6	0	Need a hug	11
7	0	@LOLTrish hey long time no see! Yes.. Rains a…	99
8	0	@Tatiana_K nope they didn't have it	36
9	0	@twittera que me muera ?	25
10	0	spring break in plain city... it's snowing	43
11	0	I just re-pierced my ears	26
12	0	@caregiving I couldn't bear to watch it. And I thought the UA loss was embarrassing	94

Box plot of tweets length

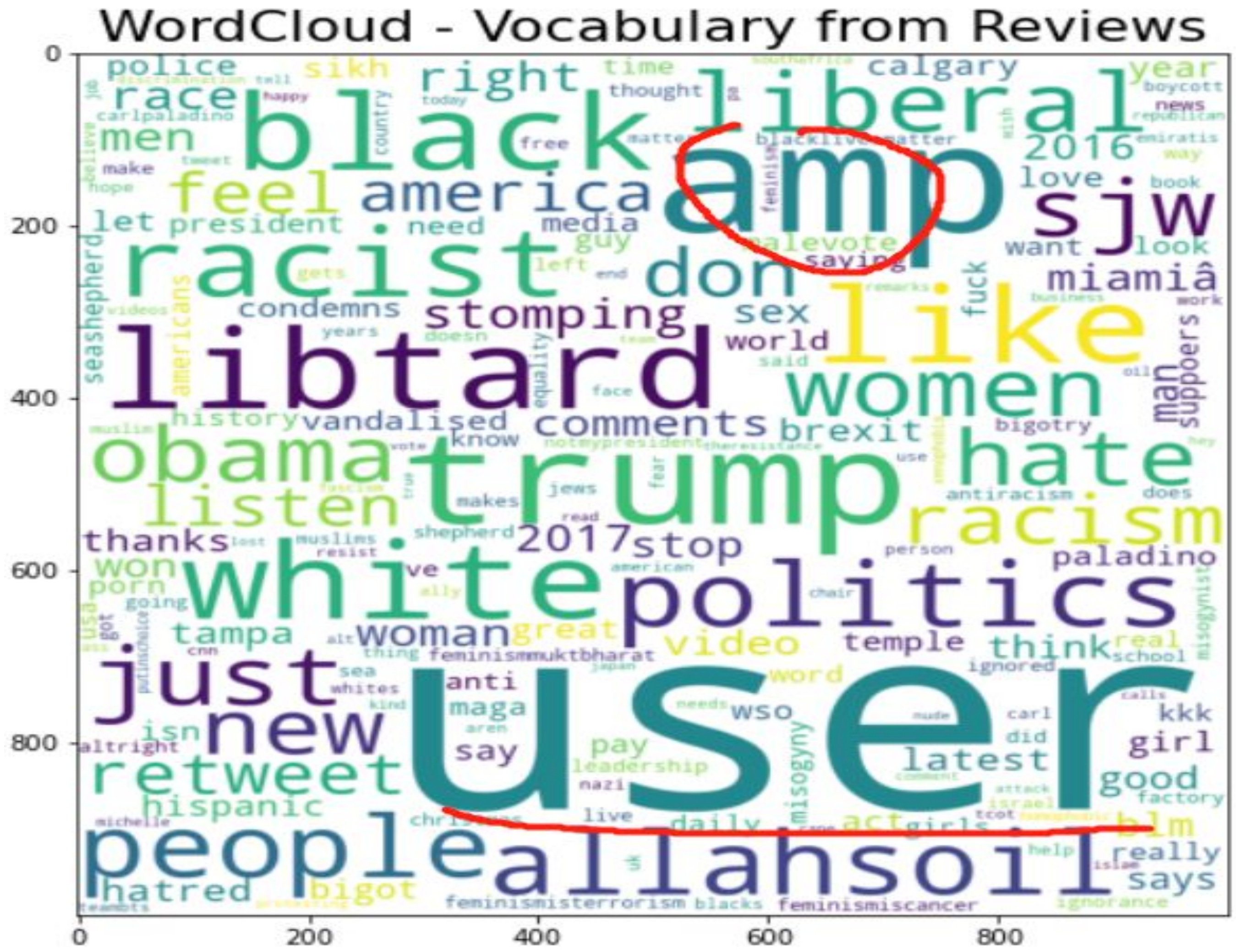


Since twitter has characters limitation of 140, check why there are tweets more than 140 characters:



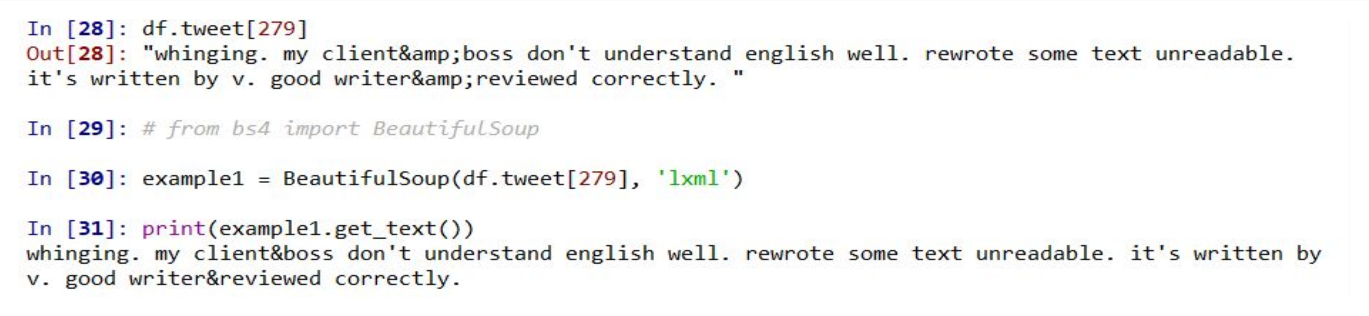
```
      label                                              tweet  pre_clean_len
213       0  Awwh babs... you look so sad underneith that s...            142
226       0  Tuesdayï¿½ll start with reflection ï¿½n then a...            141
279       0  Whinging. My client&amp;boss don't understand ...            145
343       0  @TheLeagueSF Not Fun &amp; Furious? The new ma...            145
400       0  #3 woke up and was having an accident - &quot;...            144
464       0  My bathtub drain is fired: it haz 1 job 2 do, ...            146
492       0  pears &amp; Brie, bottle of Cabernet, and &quo...            150
747       0  Have an invite for &quot;Healthy Dining&quot; ...            141
957       0  Damnit I was really digging this season of Rea...            141
1064      0  Why do I keep looking...I know that what I rea...            141
```

Dataset observation on word counts



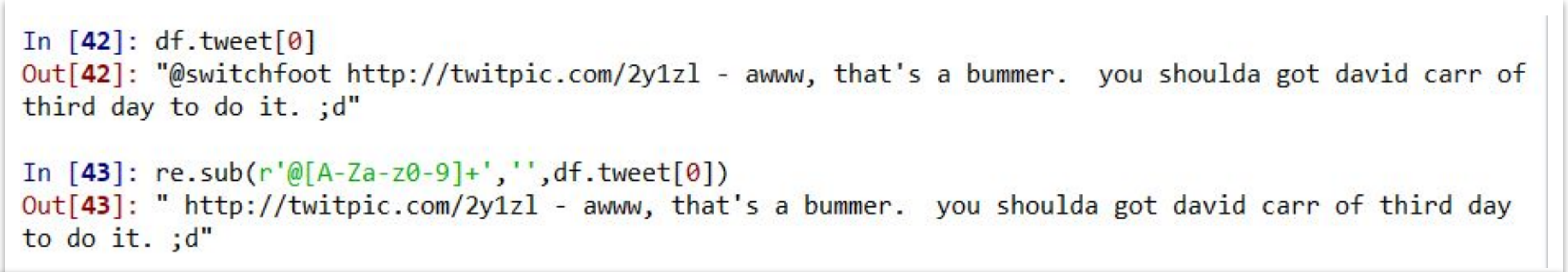
Cleaning the data

- Filter out the empty cells! - 199276 tweets left
- Making statement text in lower case
- HTML decoding for ‘&amp’ , ‘&quot’ etc.‘

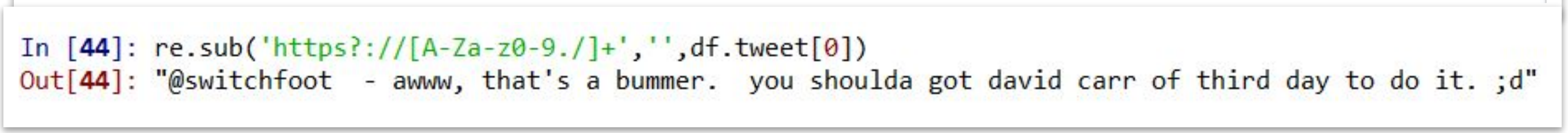


```
In [28]: df.tweet[279]
Out[28]: "whinging. my client&amp;boss don't understand english well. rewrote some text unreadable.
it's written by v. good writer&amp;reviewed correctly. "

In [29]: # from bs4 import BeautifulSoup

In [30]: example1 = BeautifulSoup(df.tweet[279], 'lxml')

In [31]: print(example1.get_text())
whinging. my client&boss don't understand english well. rewrote some text unreadable. it's written by
v. good writer&reviewed correctly.
```

- Removing URL Links and ‘@’ user:



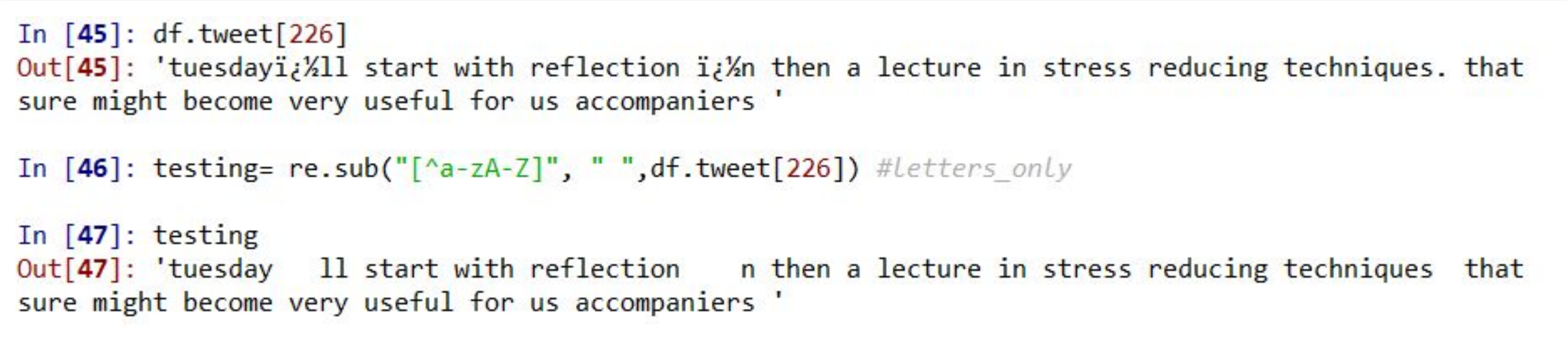
```
In [42]: df.tweet[0]
Out[42]: "@switchfoot http://twitpic.com/2y1zl - awww, that's a bummer.  you shoulda got david carr of
third day to do it. ;d"

In [43]: re.sub(r'@[A-Za-z0-9]+','',df.tweet[0])
Out[43]: " http://twitpic.com/2y1zl - awww, that's a bummer.  you shoulda got david carr of third day
to do it. ;d"
```



```
In [44]: re.sub('https?://[A-Za-z0-9./]+','',df.tweet[0])
Out[44]: "@switchfoot  - awww, that's a bummer.  you shoulda got david carr of third day to do it. ;d"
```

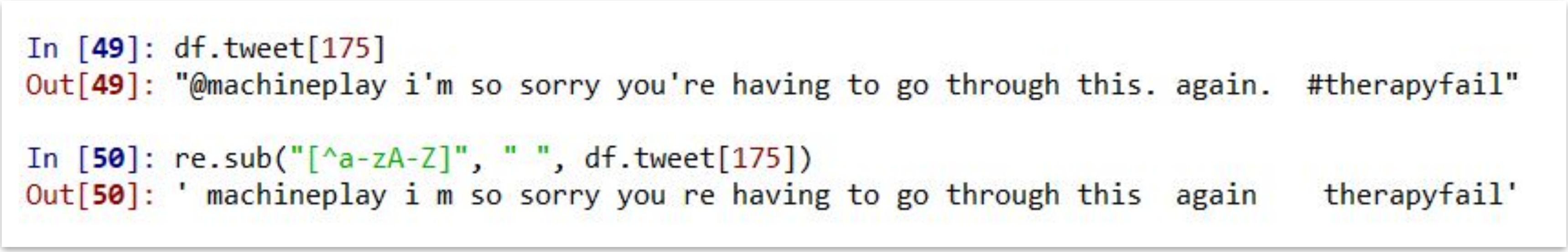
Cleaning the data:

- Removing Encoding Errors:



```
In [45]: df.tweet[226]
Out[45]: 'tuesdayï¿½ll start with reflection ï¿½n then a lecture in stress reducing techniques. that
sure might become very useful for us accompaniers '

In [46]: testing= re.sub("[^a-zA-Z]", " ",df.tweet[226]) #letters_only

In [47]: testing
Out[47]: 'tuesday   ll start with reflection    n then a lecture in stress reducing techniques  that
sure might become very useful for us accompaniers '
```

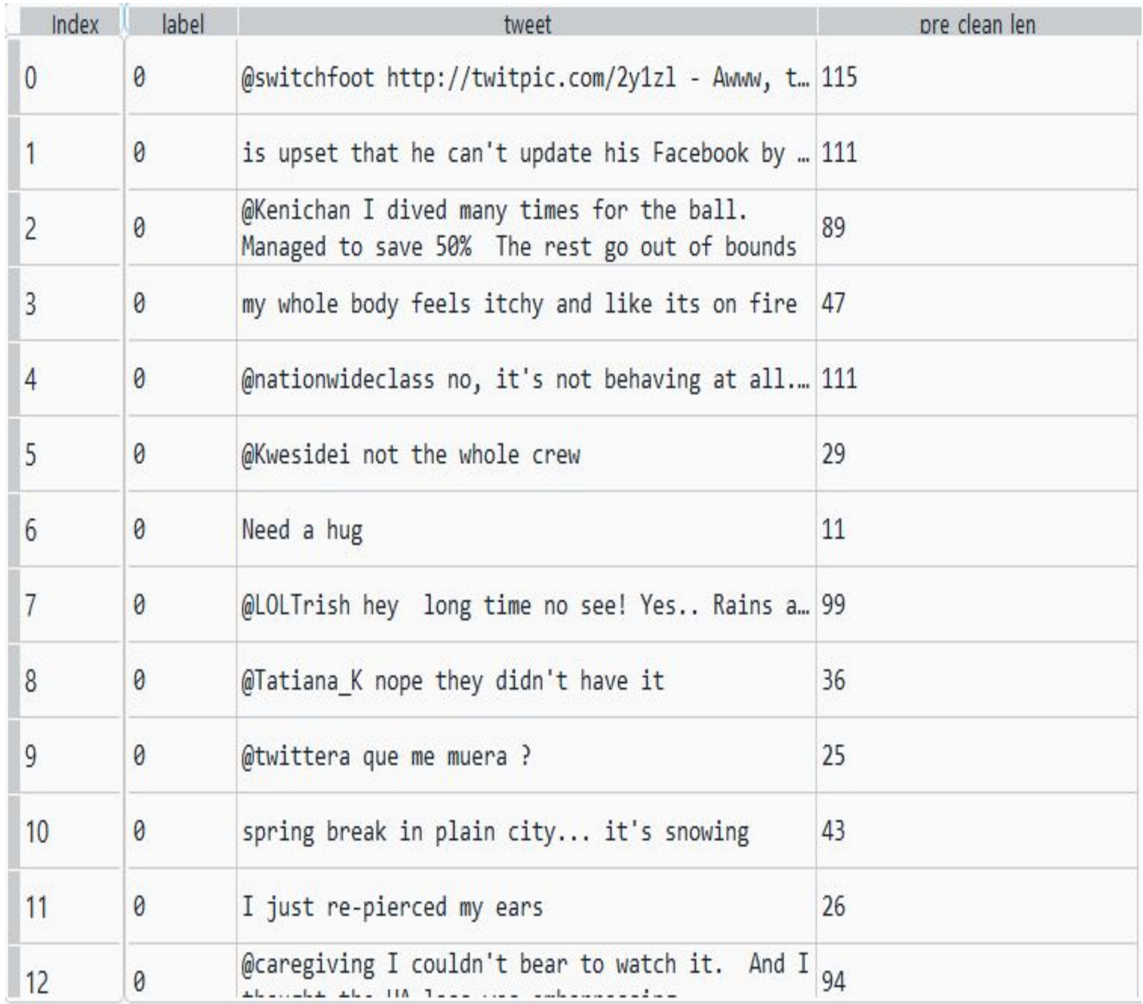
- Removing hashtag / numbers:



```
In [49]: df.tweet[175]
Out[49]: "@machineplay i'm so sorry you're having to go through this. again.  #therapyfail"

In [50]: re.sub("[^a-zA-Z]", " ", df.tweet[175])
Out[50]: ' machineplay i m so sorry you re having to go through this  again    therapyfail'
```

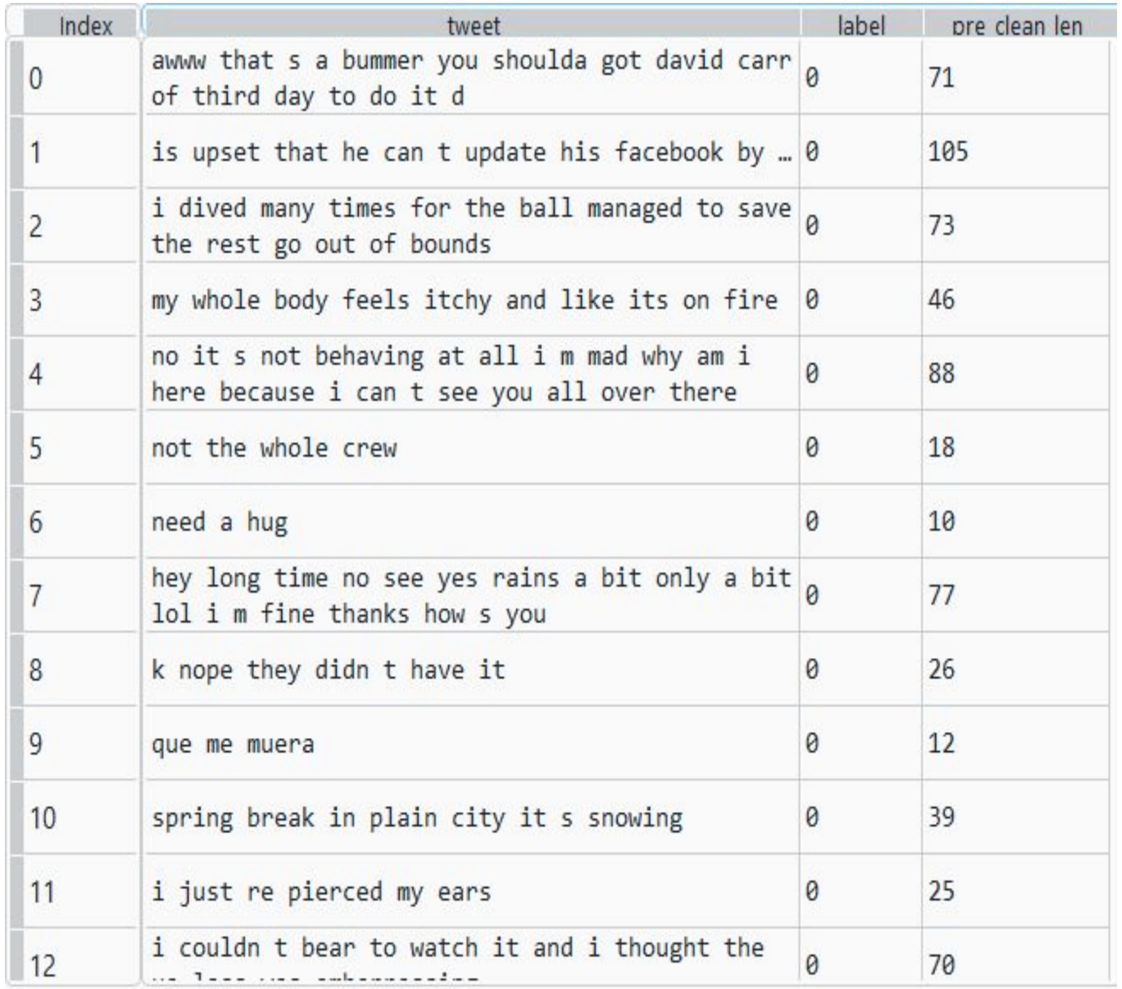
Data Results After and Before Cleaning

Original:



Index	label	tweet	pre clean len
0	0	@switchfoot http://twitpic.com/2y1zl - Awww, t…	115
1	0	is upset that he can't update his Facebook by …	111
2	0	@Kenichan I dived many times for the ball. Managed to save 50% The rest go out of bounds	89
3	0	my whole body feels itchy and like its on fire	47
4	0	@nationwideclass no, it's not behaving at all.…	111
5	0	@Kwesidei not the whole crew	29
6	0	Need a hug	11
7	0	@LOLTrish hey long time no see! Yes.. Rains a…	99
8	0	@Tatiana_K nope they didn't have it	36
9	0	@twittera que me muera ?	25
10	0	spring break in plain city... it's snowing	43
11	0	I just re-pierced my ears	26
12	0	@caregiving I couldn't bear to watch it. And I	94

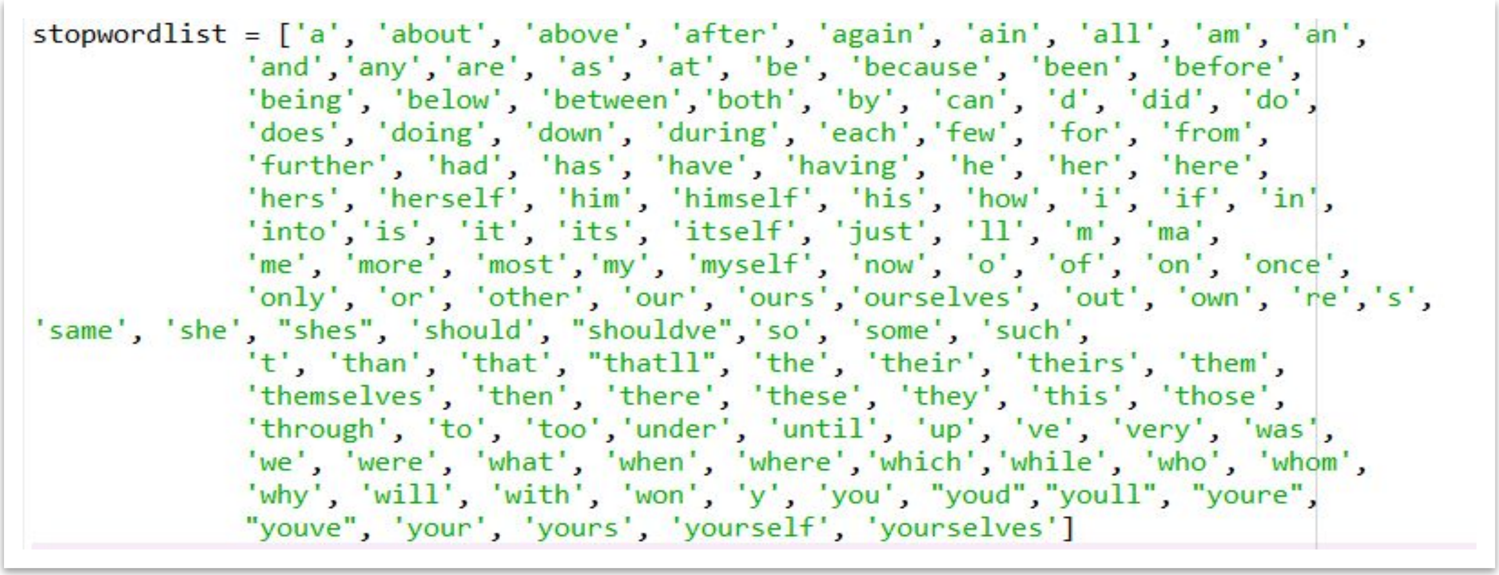
Format cleaning:



Index	tweet	label	pre clean len
0	awww that s a bummer you shoulda got david carr of third day to do it d	0	71
1	is upset that he can t update his facebook by …	0	105
2	i dived many times for the ball managed to save the rest go out of bounds	0	73
3	my whole body feels itchy and like its on fire	0	46
4	no it s not behaving at all i m mad why am i here because i can t see you all over there	0	88
5	not the whole crew	0	18
6	need a hug	0	10
7	hey long time no see yes rains a bit only a bit lol i m fine thanks how s you	0	77
8	k nope they didn t have it	0	26
9	que me muera	0	12
10	spring break in plain city it s snowing	0	39
11	i just re pierced my ears	0	25
12	i couldn t bear to watch it and i thought the	0	70

More cleaning on the words level:

- Defining set containing all stopwords in English.
- To clear the Personal Pronoun, ‘the’, and other common words that with no meanings.

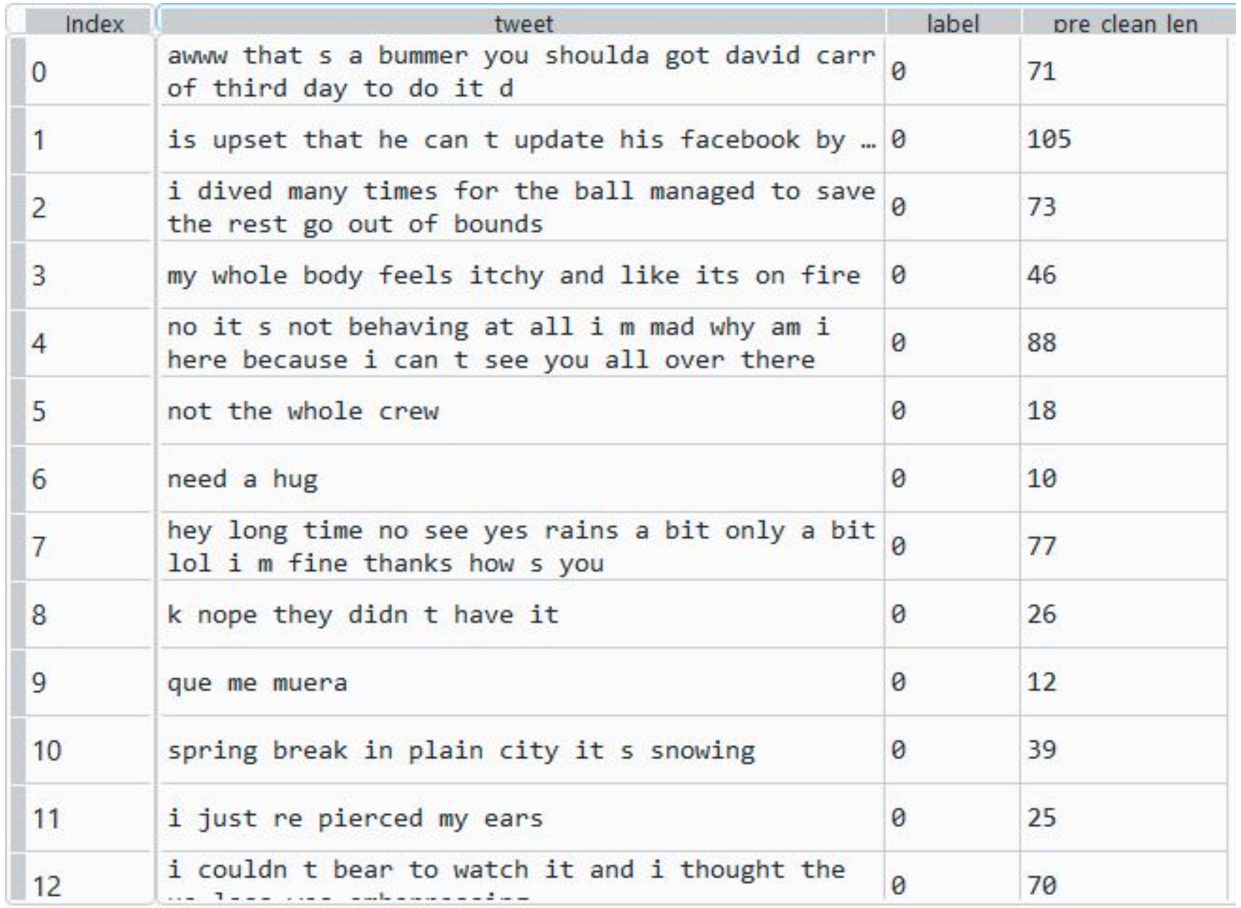


```
stopwordlist = ['a', 'about', 'above', 'after', 'again', 'ain', 'all', 'am', 'an',
             'and','any','are', 'as', 'at', 'be', 'because', 'been', 'before',
             'being', 'below', 'between','both', 'by', 'can', 'd', 'did', 'do',
             'does', 'doing', 'down', 'during', 'each','few', 'for', 'from',
             'further', 'had', 'has', 'have', 'having', 'he', 'her', 'here',
             'hers', 'herself', 'him', 'himself', 'his', 'how', 'i', 'if', 'in',
             'into','is', 'it', 'its', 'itself', 'just', 'll', 'm', 'ma',
             'me', 'more', 'most','my', 'myself', 'now', 'o', 'of', 'on', 'once',
             'only', 'or', 'other', 'our', 'ours','ourselves', 'out', 'own', 're','s',
'same', 'she', "shes", 'should', "shouldve",'so', 'some', 'such',
             't', 'than', 'that', "thatll", 'the', 'their', 'theirs', 'them',
             'themselves', 'then', 'there', 'these', 'they', 'this', 'those',
             'through', 'to', 'too','under', 'until', 'up', 've', 'very', 'was',
             'we', 'were', 'what', 'when', 'where','which','while', 'who', 'whom',
             'why', 'will', 'with', 'won', 'y', 'you', "youd","youll", "youre",
             "youve", 'your', 'yours', 'yourself', 'yourselves']
```

- Removing short words that are less than 1 letter (I feel ‘hmm’ and ‘oh’ may be useful for the test, so I am going to keep them):

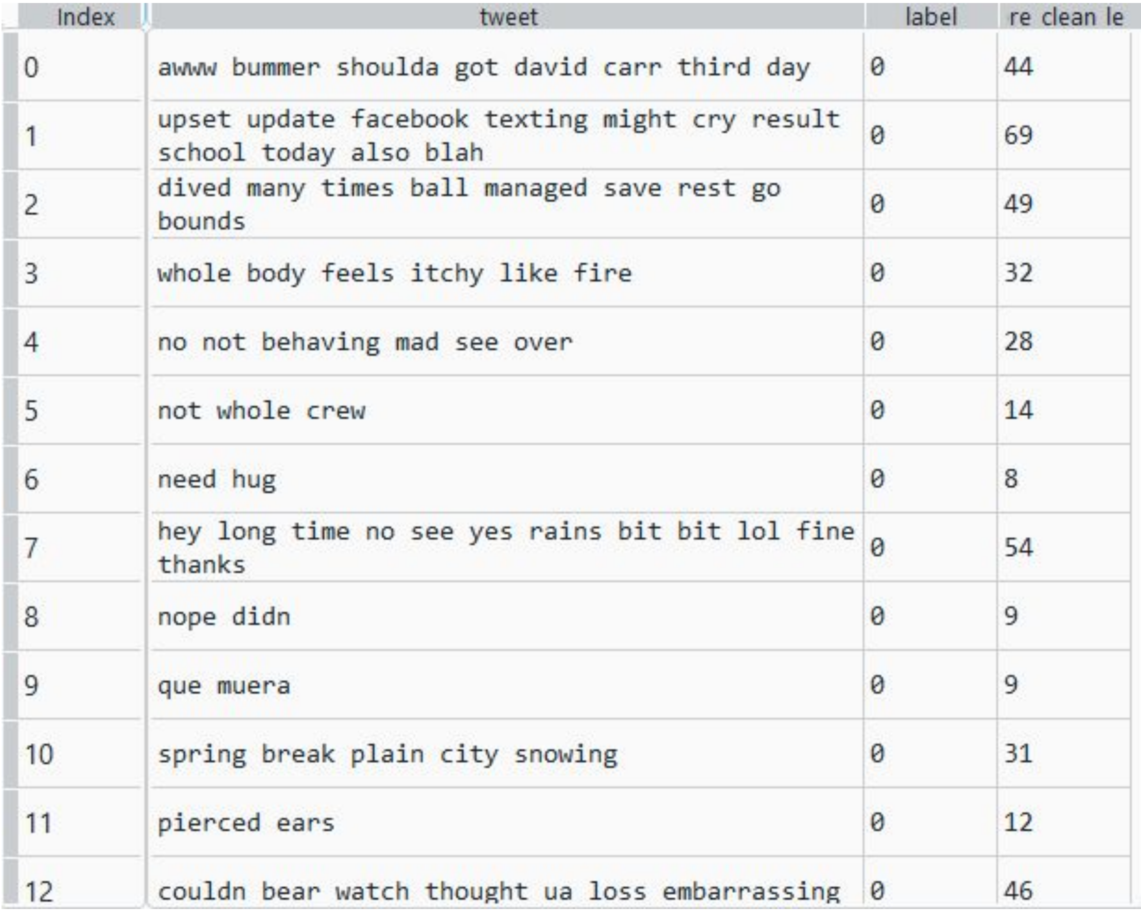
Data Cleaning on word level:

Format cleaning:1



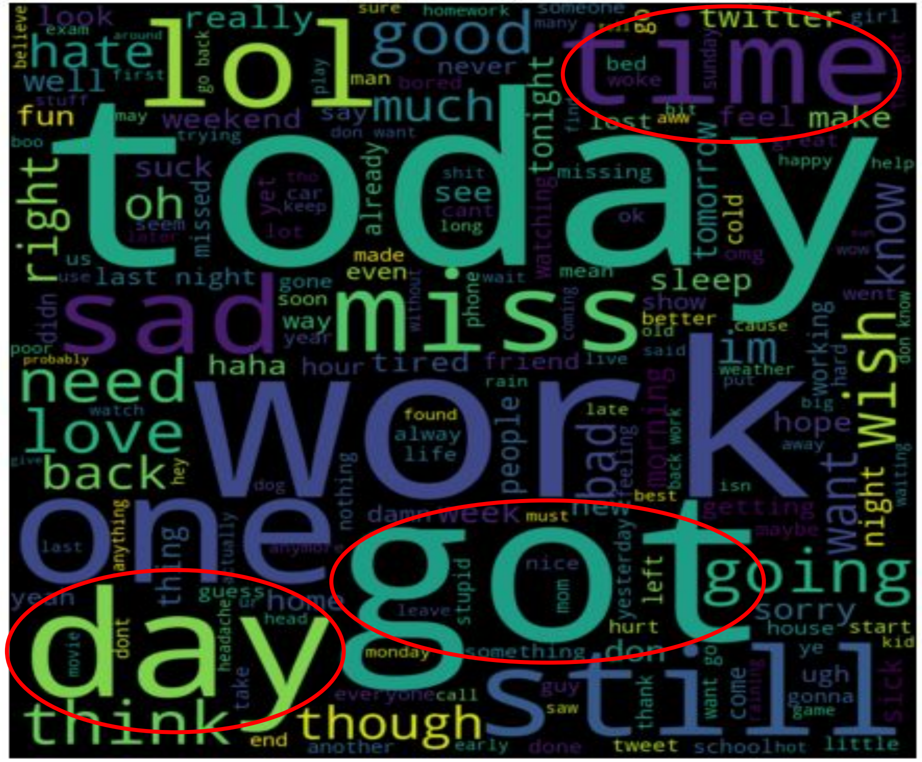
Index	tweet	label	pre clean len
0	aww that s a bummer you shoulda got david carr of third day to do it d	0	71
1	is upset that he can t update his facebook by …	0	105
2	i dived many times for the ball managed to save the rest go out of bounds	0	73
3	my whole body feels itchy and like its on fire	0	46
4	no it s not behaving at all i m mad why am i here because i can t see you all over there	0	88
5	not the whole crew	0	18
6	need a hug	0	10
7	hey long time no see yes rains a bit only a bit lol i m fine thanks how s you	0	77
8	k nope they didn t have it	0	26
9	que me muera	0	12
10	spring break in plain city it s snowing	0	39
11	i just re pierced my ears	0	25
12	i couldn t bear to watch it and i thought the ua loss was embarrassing	0	70

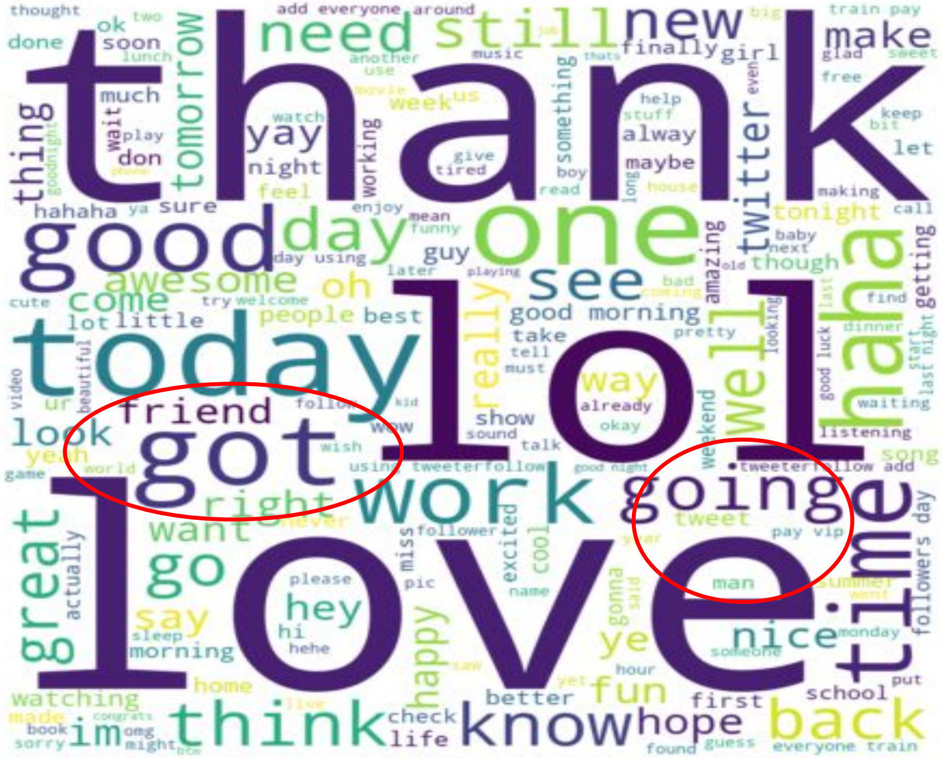
Format cleaning 2



Index	tweet	label	re clean le
0	awww bummer shoulda got david carr third day	0	44
1	upset update facebook texting might cry result school today also blah	0	69
2	dived many times ball managed save rest go bounds	0	49
3	whole body feels itchy like fire	0	32
4	no not behaving mad see over	0	28
5	not whole crew	0	14
6	need hug	0	8
7	hey long time no see yes rains bit bit lol fine thanks	0	54
8	nope didn	0	9
9	que muera	0	9
10	spring break plain city snowing	0	31
11	pierced ears	0	12
12	couldn bear watch thought ua loss embarrassing	0	46

Detection of Meaningless Words

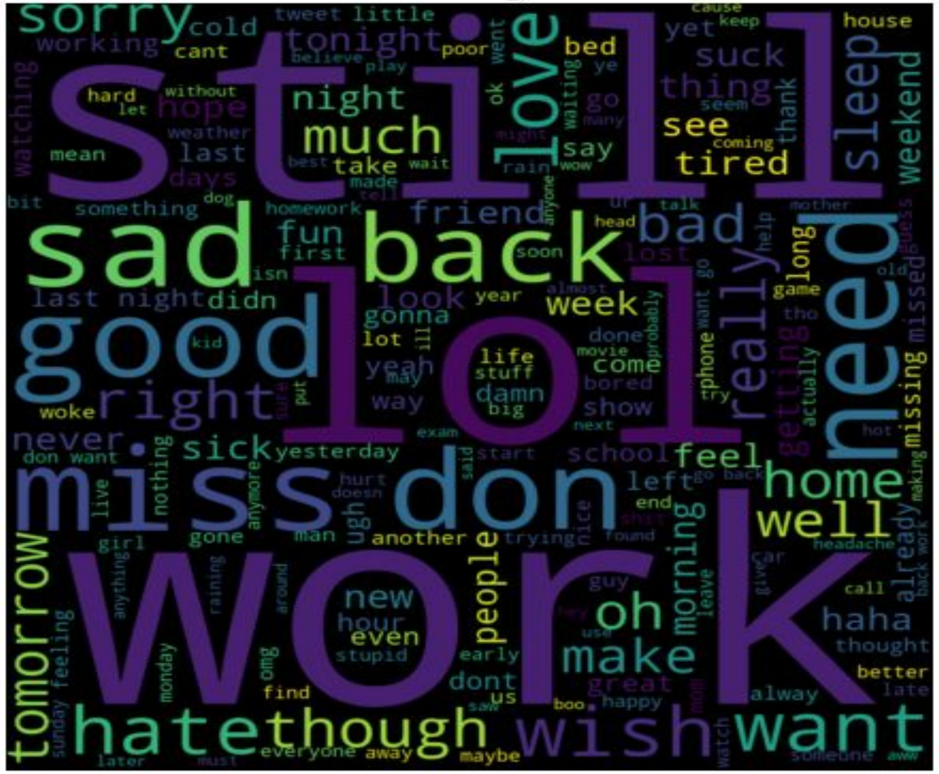

WordCloud - negative words


WordCloud - positive words

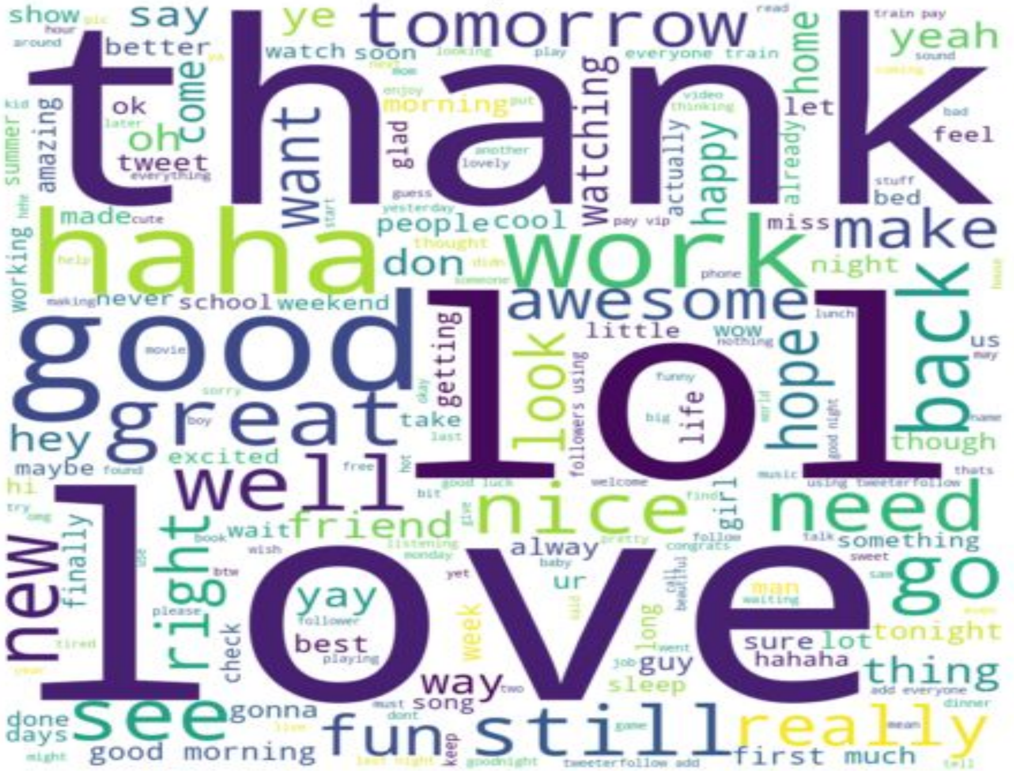
- Removing words : 'day','today','got','going','time','im','think','one','think','know’, ‘twitter’

Removing meaningless words

WordCloud - negative words

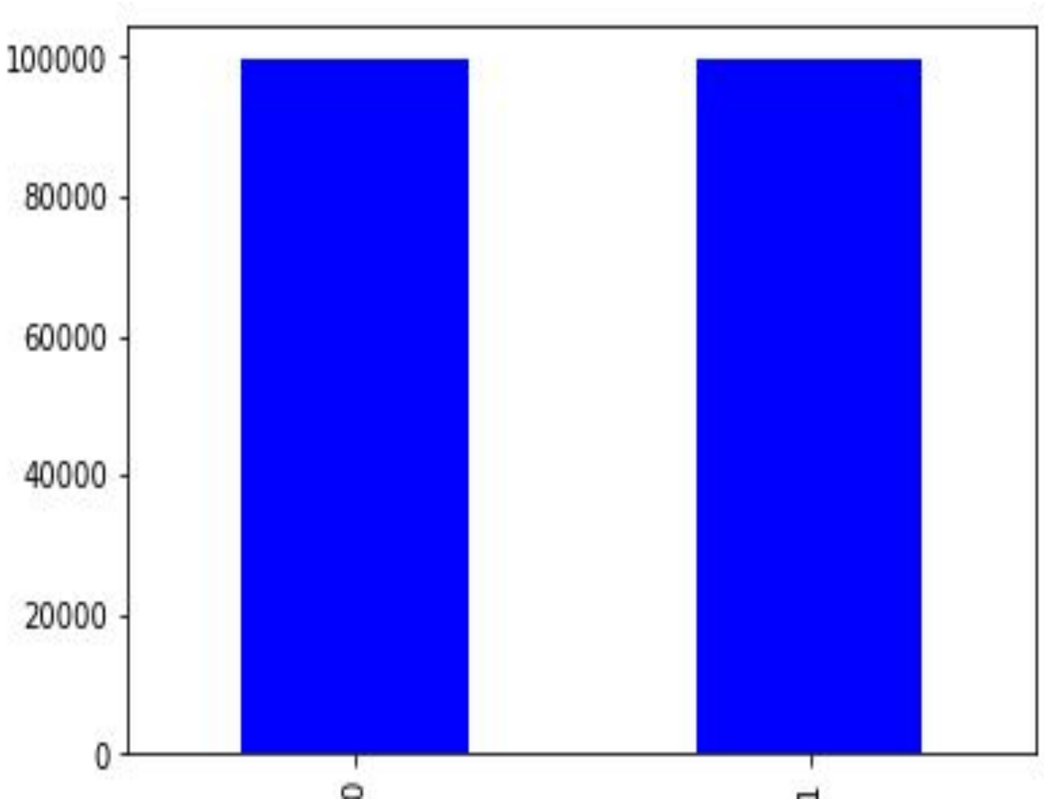


WordCloud - positive words



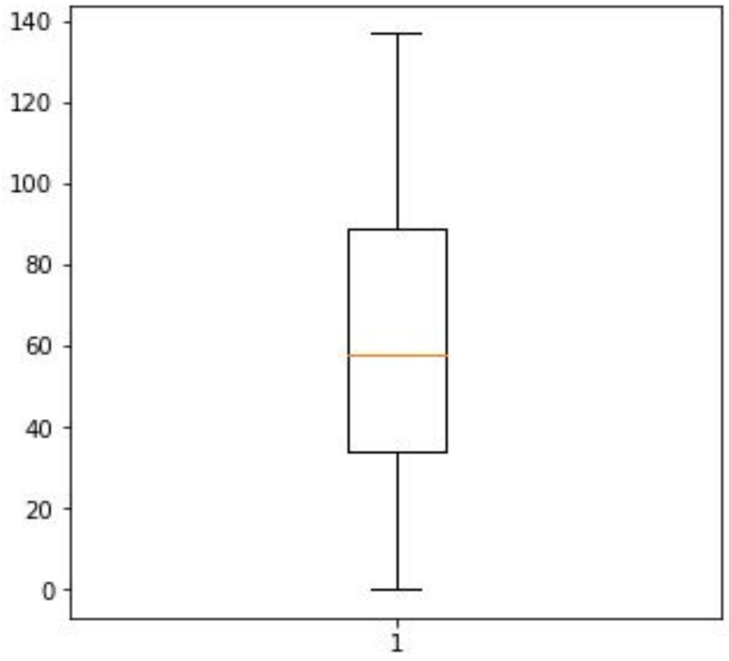
Exploratory Analysis

- Total number of tweets: 199091
- Total number of words :77692
- Data is not skewed as we have equal numbers of negative and positive tweets.



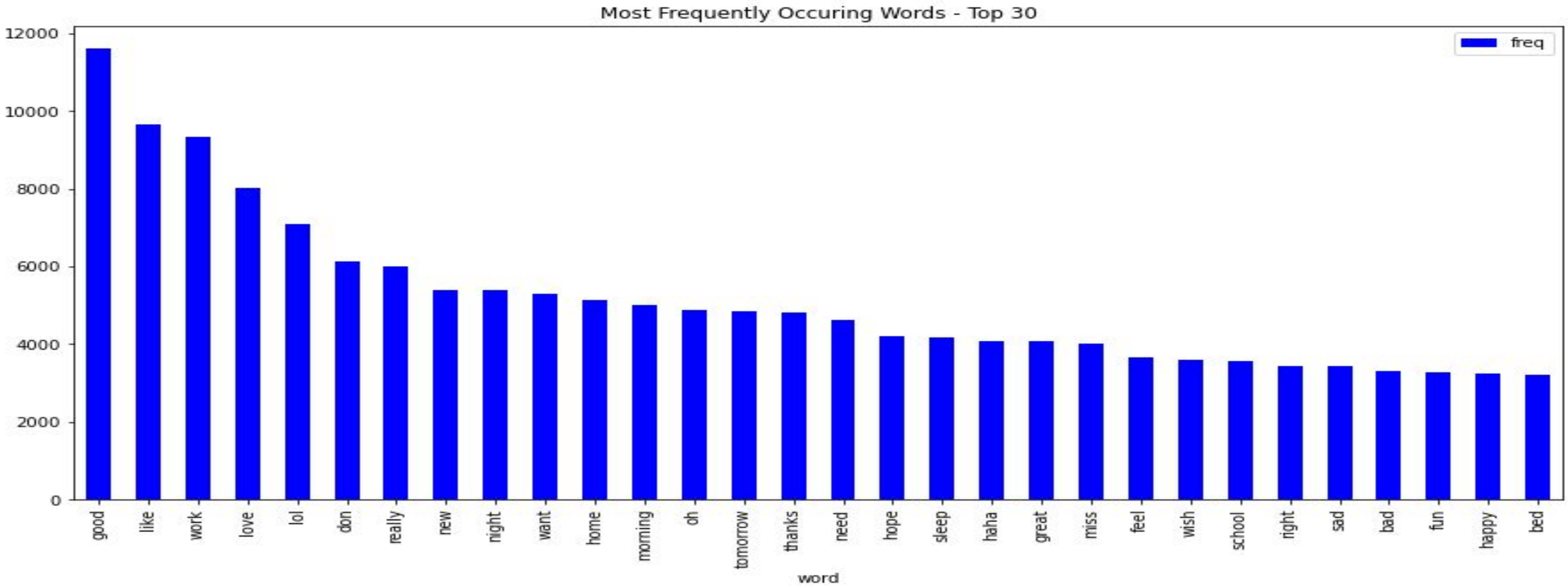
- Box plot of string length:

there are no text more than 140 characters.



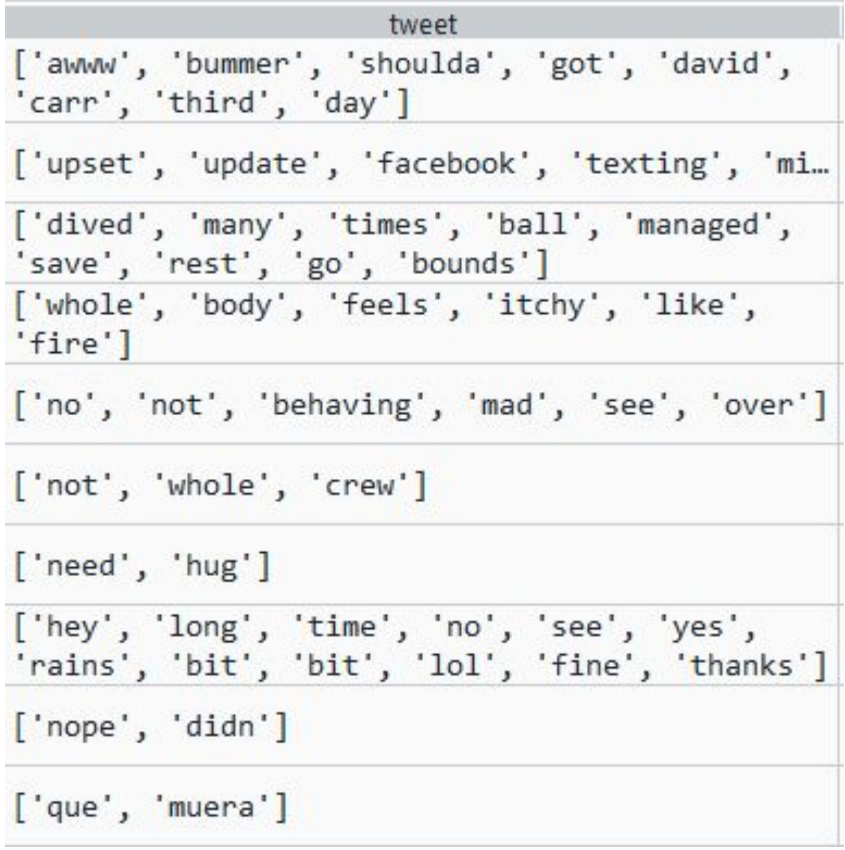
Exploratory Analysis

- Top 30 most frequency words in the data using Count Vectorize method



Preparation Data for Classification

- Do Word Tokenization (splitting a piece of a text into individual word based on a certain delimiter)
- Applying Stemming (to make all the format of words uniformed, remove e.g "ing", "s", "ed"...)
- Applying Lemmatization (morphological analysis)



tweet
['awww', 'bummer', 'shoulda', 'got', 'david', 'carr', 'third', 'day']
['upset', 'update', 'facebook', 'texting', 'mi…
['dived', 'many', 'times', 'ball', 'managed', 'save', 'rest', 'go', 'bounds']
['whole', 'body', 'feels', 'itchy', 'like', 'fire']
['no', 'not', 'behaving', 'mad', 'see', 'over']
['not', 'whole', 'crew']
['need', 'hug']
['hey', 'long', 'time', 'no', 'see', 'yes', 'rains', 'bit', 'bit', 'lol', 'fine', 'thanks']
['nope', 'didn']
['que', 'muera']

Feature Extractions-2500 Features

- One-Hot method :
  - Text data representation: (frequency is 0 or 1)

- CV Method :
  - Transform a given text into a vector on the basis of the frequency (count) of each word that occurs in the entire text.

- TF-IDF (term frequency–inverse document frequency) Method :
  - Evaluates how relevant a word is to a document in a collection of documents, the value is not an integer.

Appling ML Models

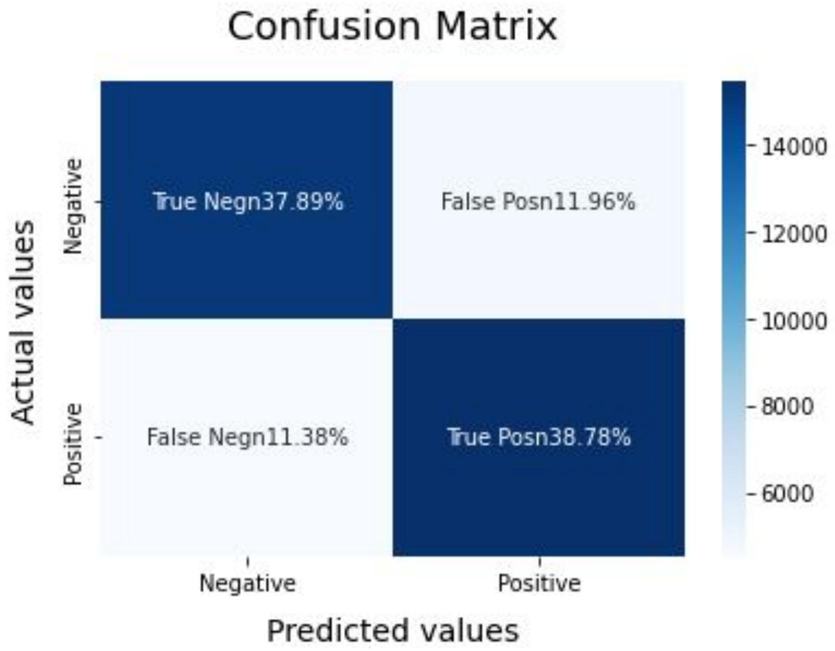
- Naive Bayes
  - BernoulliNB
  - MultiNomialNB
- K- Nearest Neighbour
- CART (Classification and Regression Tree)
- Random Forest
- Logistic Regression
  - With PCA and without PCA
  - Using L1 and L2 Regularization with SAG and SAGA Optimizations
- Neural Network
  - With PCA and without PCA

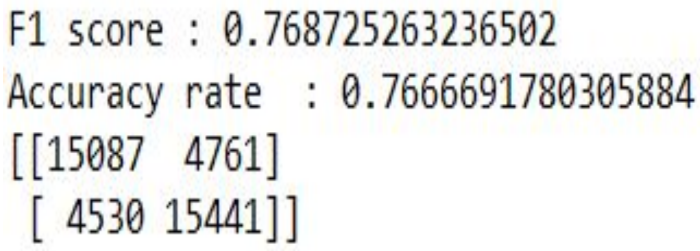
Naive Bayes:

- Results:
- One-hot (frequency is 0 or 1):
- Bernoulli NB using alpha=0 accuracy rate=0.7553
- Bernoulli NB using alpha=1 accuracy rate=0.7553
- Multinomial NB accuracy rate=0.7501

- Cv:
- Bernoulli NB using alpha=0 accuracy rate=0.7550
- Bernoulli NB using alpha=1 accuracy rate=0.7550
- Multinomial NB accuracy rate=0.7490

- TF-IDF:
- Bernoulli NB using alpha=0 accuracy rate=0.7667
- Bernoulli NB using alpha=1 accuracy rate=0.7667
- Multinomial NB accuracy rate=0.7602





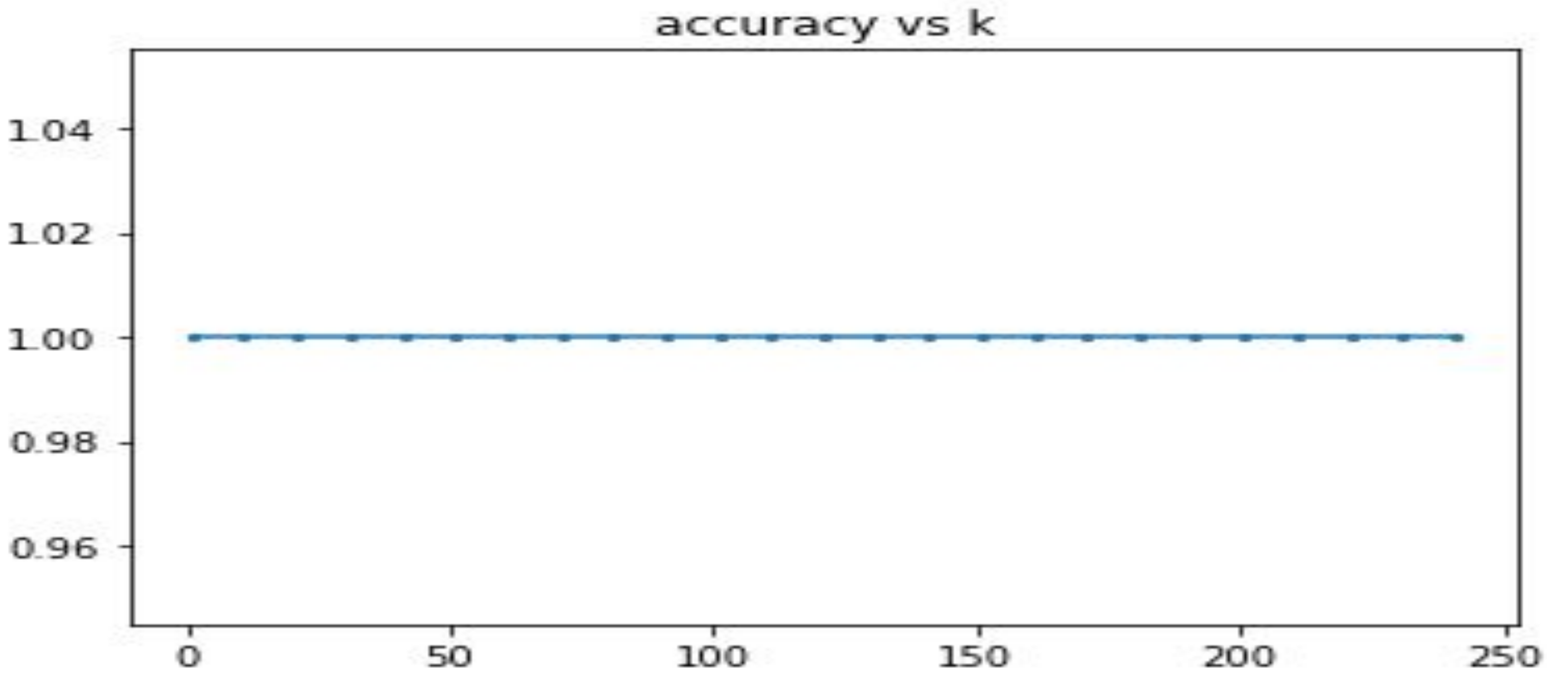
```
F1 score : 0.768725263236502
Accuracy rate  : 0.7666691780305884
[[15087  4761]
 [ 4530 15441]]
```

Naive Bayes Result:

- It seems TF-IDF works better for Naive Bayes.
- One-hot and CV works better for Bernoulli NB than Multinomial NB which makes sense as each tweet is short, and most of the frequency in this dataset is 0 or 1.
- Bernoulli NB works slightly better than Multinomial NB.
- We have a base-line of a 0.7667 accuracy. Using Bernoulli NB with TF-IDF feature extraction.

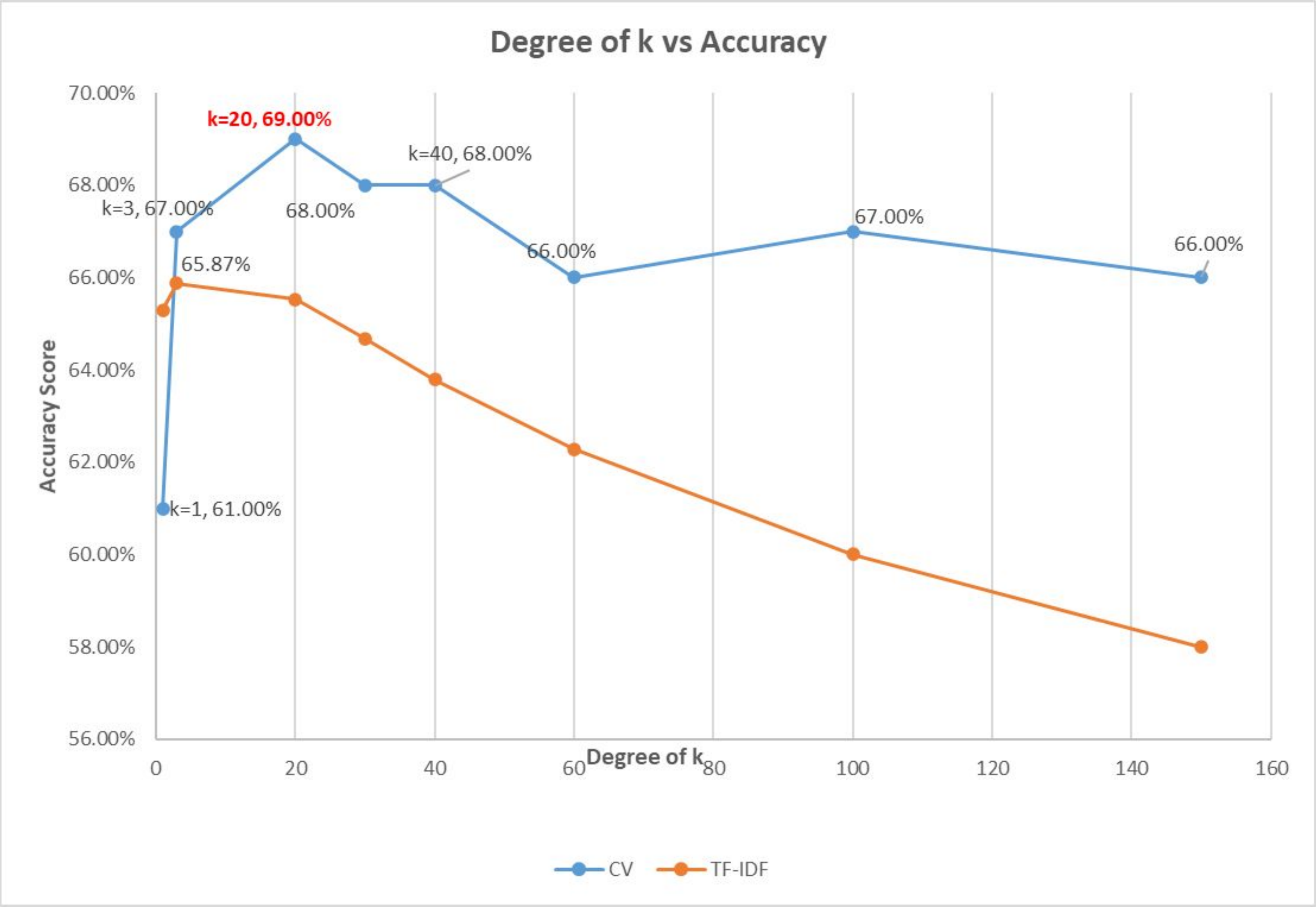
KNN- small subset with 5000 observation and 100 features

- Since the dataset is too large, I made a smaller set to see the pattern of the best K: By using the Cross Validation, the degree of k from 1 to 250 has the same result which seems useless to our data. So We test some k for our model.



K-Nearest Neighbour

- The best k is 20 and the accuracy rate is 69% with CV. Since the dataset is very large, and the accuracy of knn given k=20 is lower than Naive Bayes method, it seems knn is not a good method for this analysis.



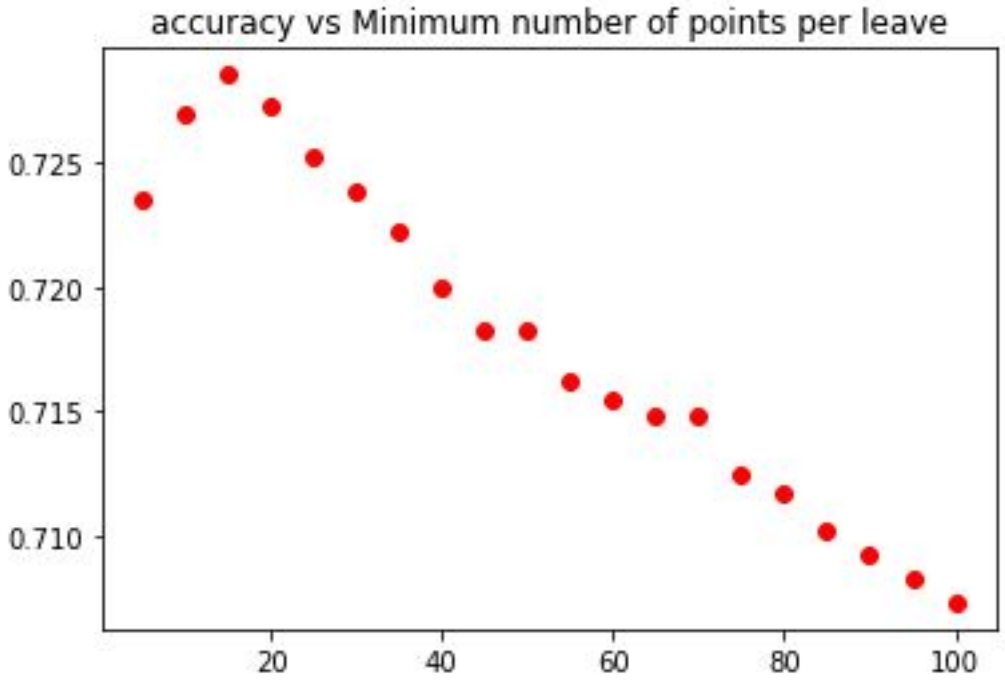
Classification & Regression Trees (CART)

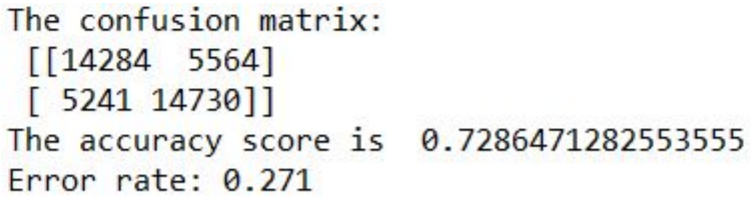
- We found changing minimum number of points per leave can improve model (below is using CV method):
  - The accuracy of Max depth=5 tree is 55%.
  - The accuracy of Max depth=10 tree is 57%.
  - The accuracy of Minimum number of points per leaf= 2000 tree is 60%.
  - The accuracy of Minimum number of points per leaf= 1000 tree is 62%.
  - The accuracy of Minimum number of points per leaf= 500 tree is 65%.
  - The accuracy of Minimum number of points per leaf= 100 tree is 70%

CART

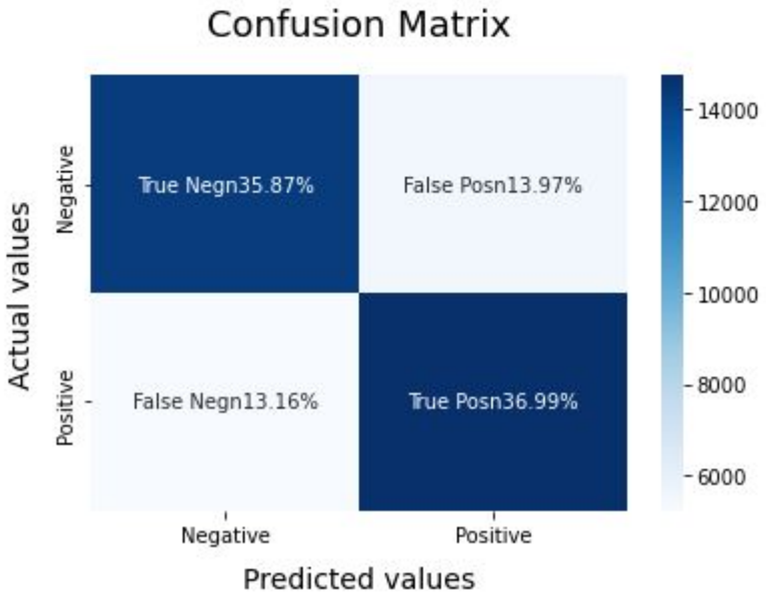
The best tree model is when the minimum number of points per leaf= 15, the accuracy rate is around 72.86% with CV.

Below graph is accuracy vs minimum number of point per leaf from 5 to 100 jumping every 5 values:





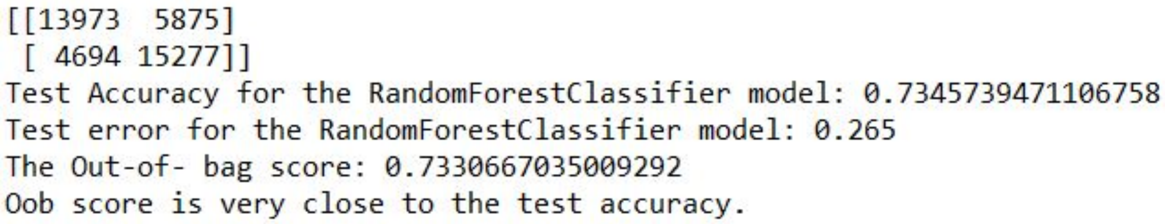
```
The confusion matrix:
 [[14284  5564]
 [ 5241 14730]]
The accuracy score is  0.7286471282553555
Error rate: 0.271
```



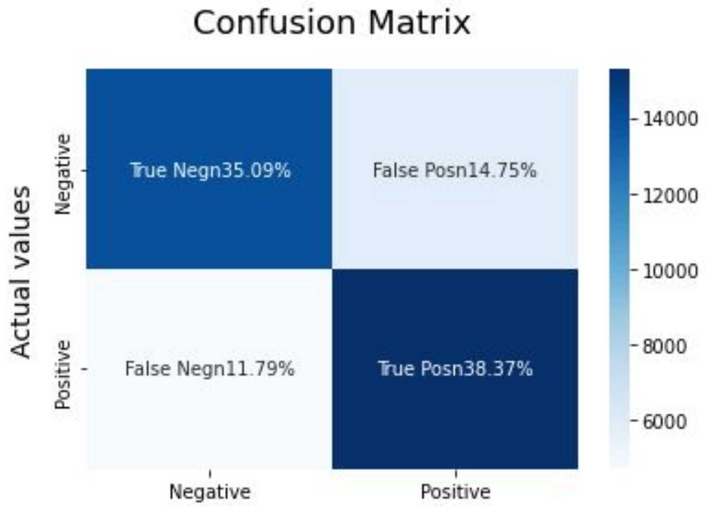
The best tree of TF-IDF is when minimum number of points per leaf =35, score=72.2%

Random Forest

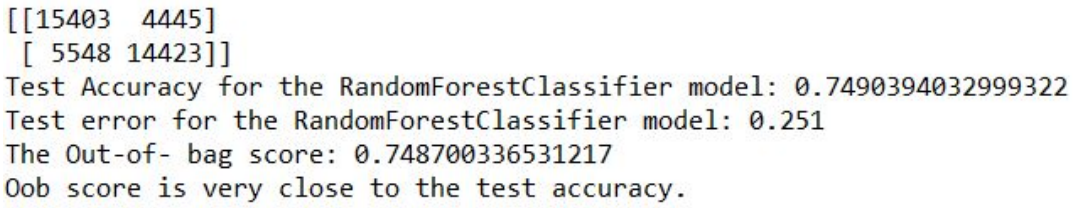
The random forest given number of estimators 500, and with minimum number items per leaf equal to 20 has an accuracy around 73.4% (CV).



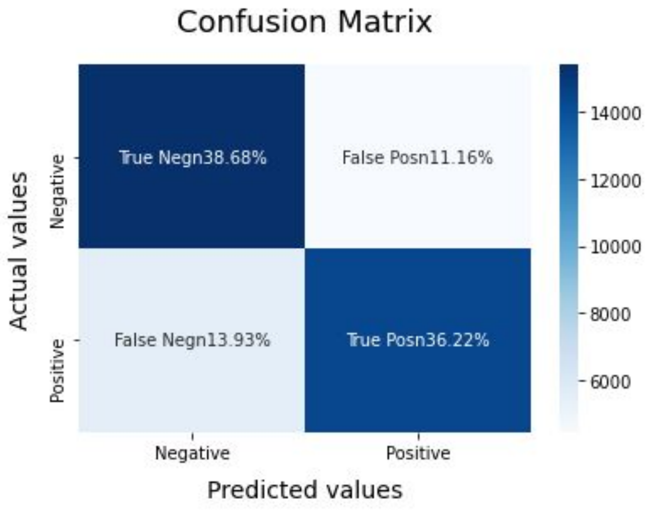
```
[[13973  5875]
 [ 4694 15277]]
Test Accuracy for the RandomForestClassifier model: 0.7345739471106758
Test error for the RandomForestClassifier model: 0.265
The Out-of- bag score: 0.7330667035009292
Oob score is very close to the test accuracy.
```



The random forest given number of estimators 500, and with minimum number items per leaf equal to 20 has an accuracy around 74.88% (TF-IDF).



```
[[15403  4445]
 [ 5548 14423]]
Test Accuracy for the RandomForestClassifier model: 0.7490394032999322
Test error for the RandomForestClassifier model: 0.251
The Out-of- bag score: 0.748700336531217
Oob score is very close to the test accuracy.
```



Logistic Regression without PCA

<u>**L2 regularization with the SAG optimization method:**</u>

c = [0.0001,0.001,0.01,0.1,1,10,100]

CV result:

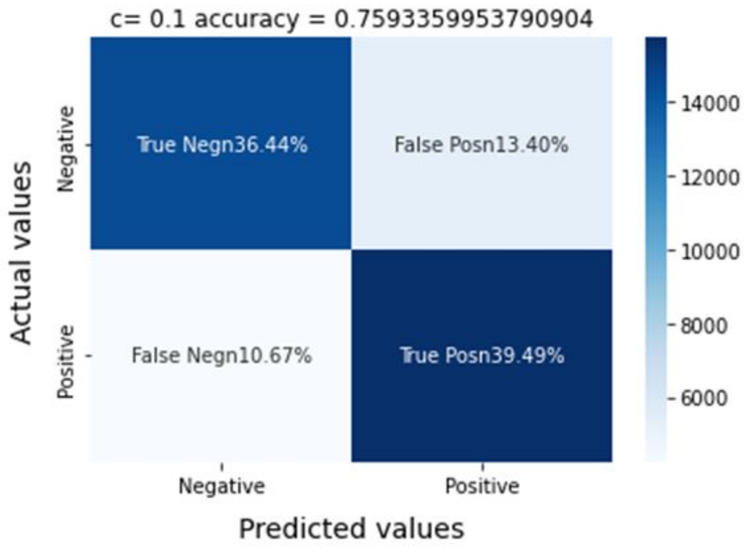
- The accuracy rate of each c is

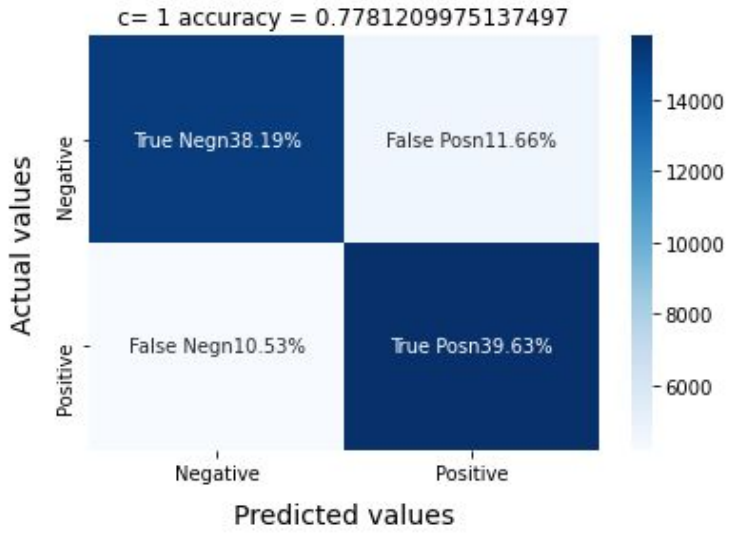
[0.7171, 0.7334, 0.7529, 0.7593, 0.7589, 0.7587, 0.7587]

- The highest accuracy is **0.7593** when c= 0.1.

TF-IDF result:

- The accuracy rate of each c is[0.7385, 0.7413, 0.7615, 0.7772, 0.7781, 0.7769, 0.7768]
- The highest accuracy is **0.7781** when c= 1 .





Logistic Regression without PCA

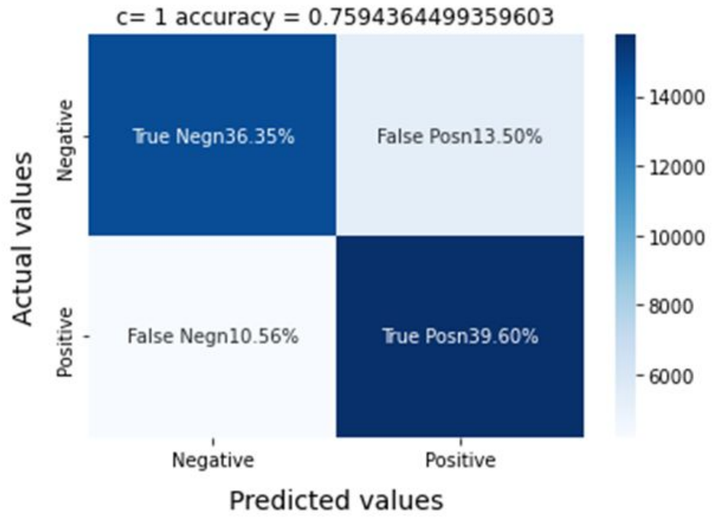
L1 lasso regularization with SAGA optimization method.
c = [0.0001,0.001,0.01,0.1,1,10,100]

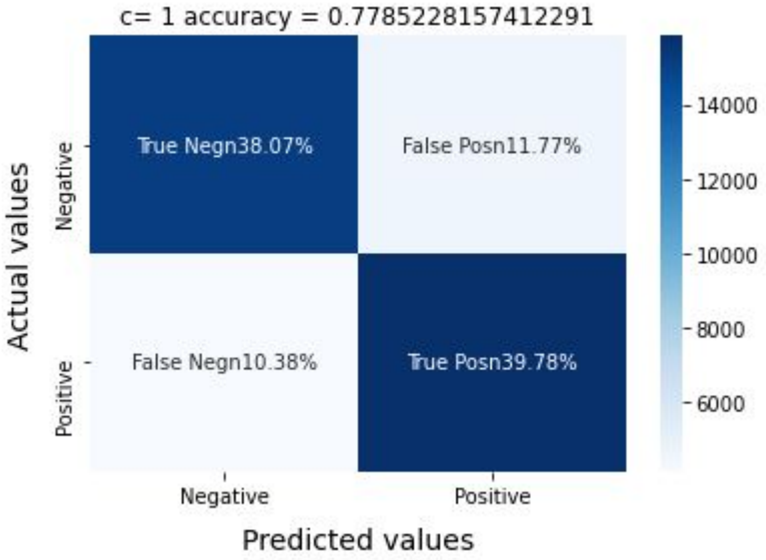
CV result:

- The accuracy rate of each c is[0.5015, 0.5509, 0.6938, 0.7545, 0.7594, 0.7590, 0.7589]
- The highest accuracy is **0.7594** when c= 1.

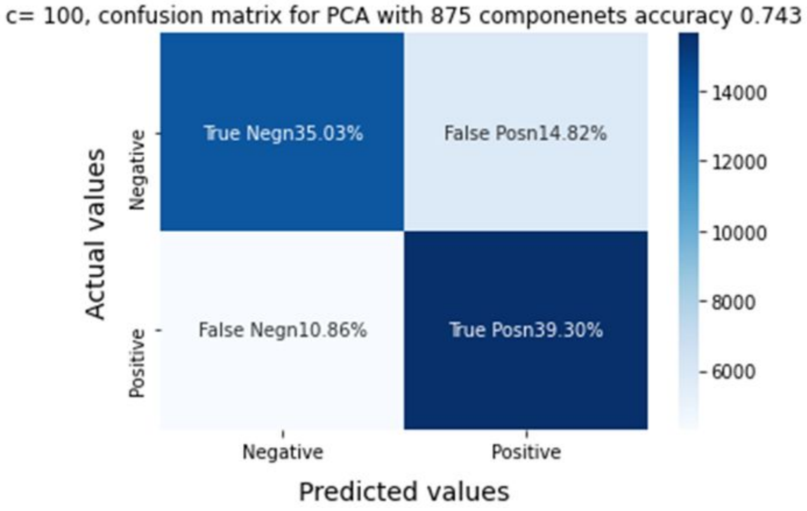
TF-IDF result:

- The accuracy rate of each c is[0.4985, 0.5015, 0.6807, 0.7648, 0.7785, 0.7770, 0.7767]
- The highest accuracy is **0.7785** when c= 1 .





Logistic Regression with PCA



L2 regularization with the SAG optimization method:

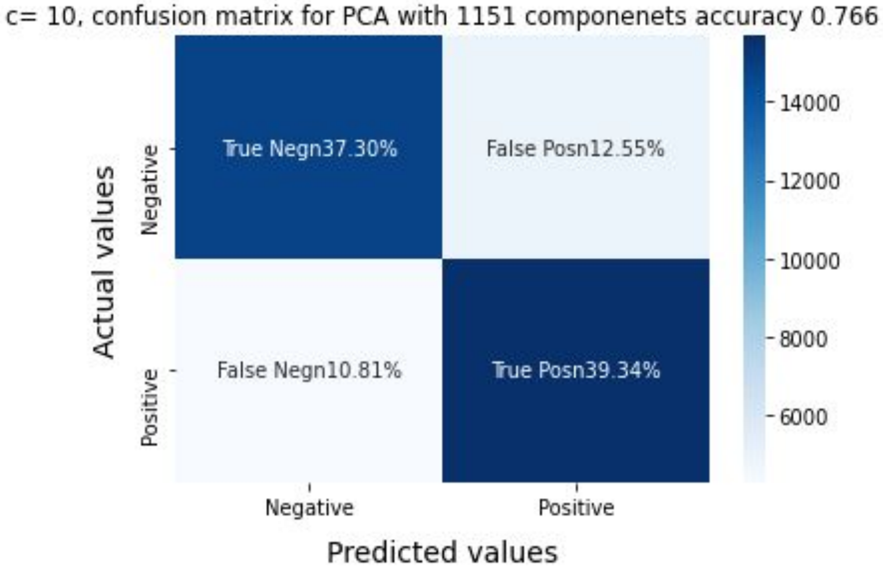
c = [0.0001,0.001,0.01,0.1,1,10,100]

CV result:

- PCA reduced 2500 features to 1151 features.
- The accuracy rate of each c is [0.7128, 0.7249, 0.7402, 0.7426, 0.7430 0.7432, 0.7432]
- The highest accuracy is **0.7432** when c= 100.

TF-IDF result :

- PCA reduced 2500 features to 1151 features.
- Accuracy rate of each c is[0.7351, 0.7386, 0.7563, 0.7649, 0.7663, 0.7664, 0.7498]
- The highest accuracy is **0.7664** when c= 10 .



Logistic Regression with PCA

L1 lasso regularization with SAGA optimization method.

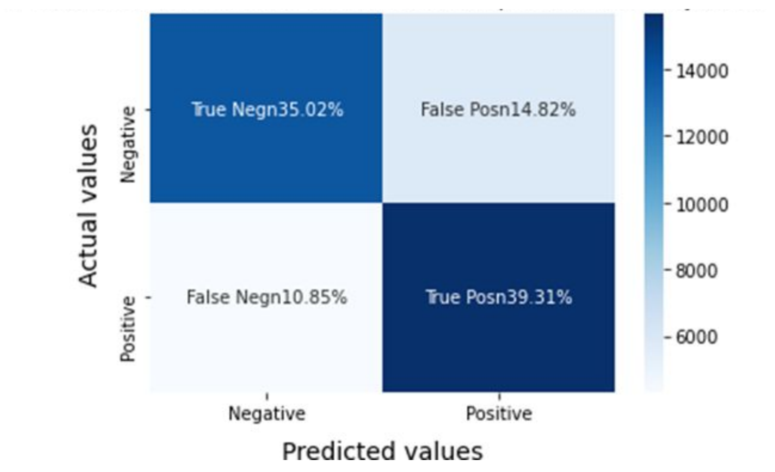
c = [0.0001,0.001,0.01,0.1,1,10,100]

CV result:

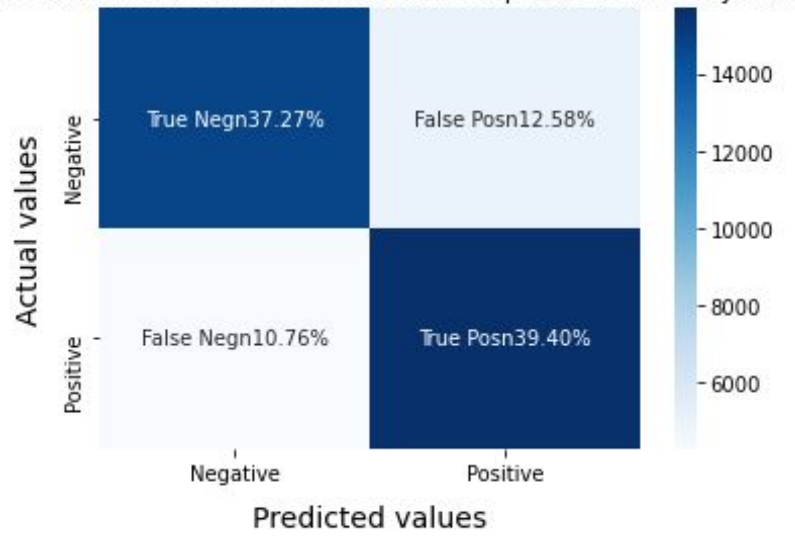
- The accuracy rate of each c is[0.4985, 0.6240, 0.7023, 0.7418, 0.7426, 0.7432, 0.7433]
- The highest accuracy is **0.7433** when c= 100.

TF-IDF result:

- The accuracy rate of each c is[0.4985, 0.6346, 0.7025, 0.7575, 0.7665, 0.7667, 0.7662]
- The highest accuracy is **0.7667** when c= 10 .



c= 10, confusion matrix for PCA with 1151 componenets accuracy 0.767

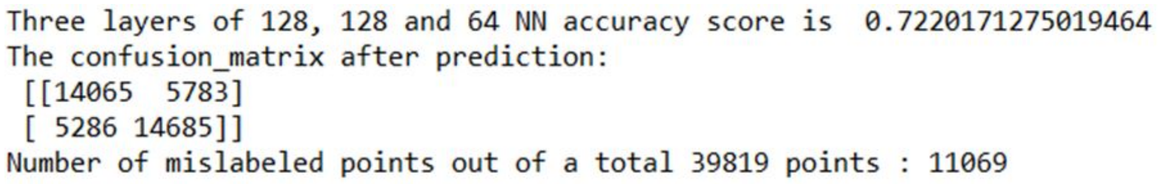


Logistic Regression Results

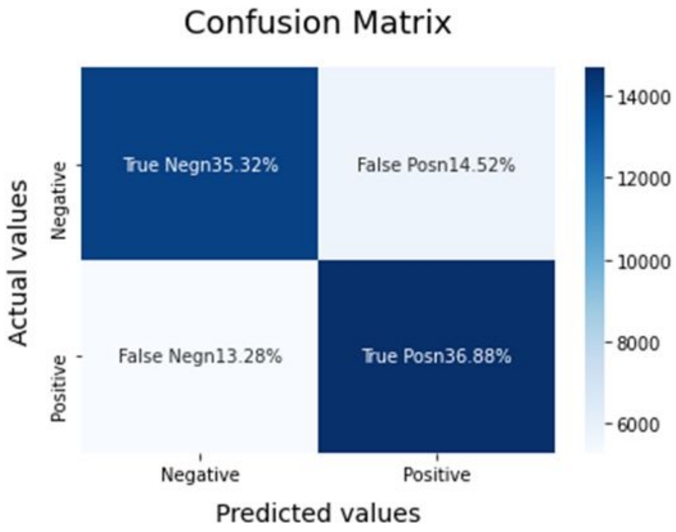
- The best result is using L1 lasso regularization with SAGA optimization method before PCA and using TF-idf representation. The accuracy rate is 77.85%, when c =1, which is slightly higher than L2 SAG, accuracy is 77.81% when c= 1.
- PCA doesn’t work well for the linear logistic regression model.
- It costs too high (memory error) for polynomial logistic regression model, will try this using Neural Network (We tried SVM, but still too slow).
- PCA doesn't work for NB, as the Naive Bayes classifier needs discrete-valued features, but the PCA breaks this property of the features.

Neural Network without PCA

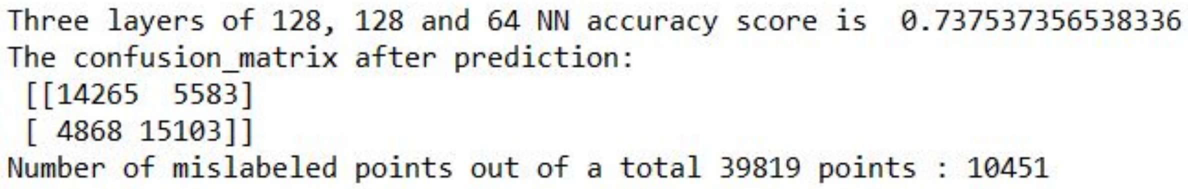
CV Three layers of 128, 128, 64 without PCA:



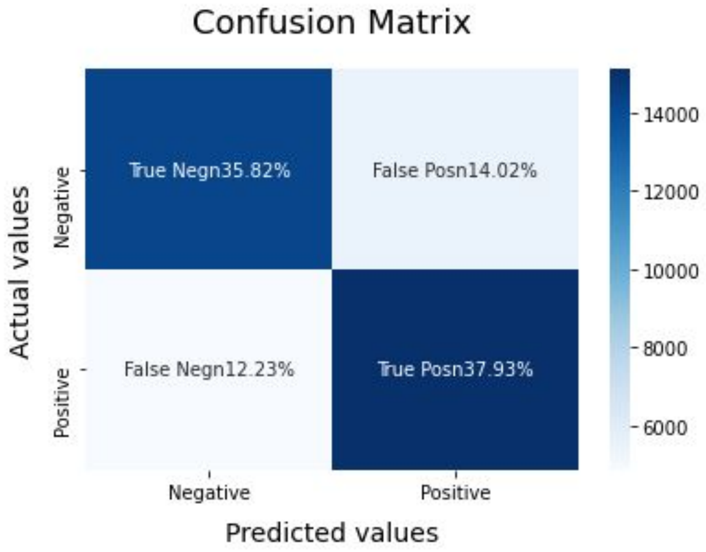
```
Three layers of 128, 128 and 64 NN accuracy score is  0.7220171275019464
The confusion_matrix after prediction:
 [[14065  5783]
 [ 5286 14685]]
Number of mislabeled points out of a total 39819 points : 11069
```



TF-IDF Three layers of 128, 128, 64 without PCA:

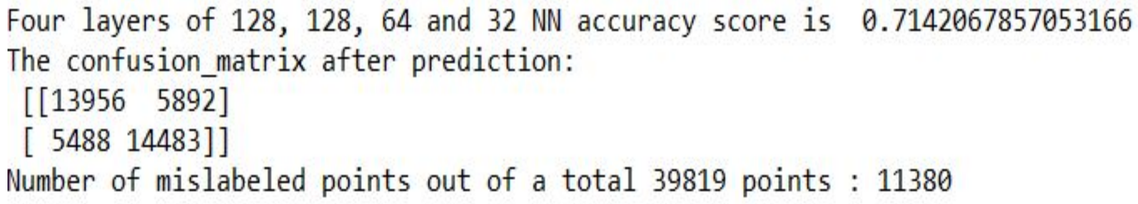


```
Three layers of 128, 128 and 64 NN accuracy score is  0.737537356538336
The confusion_matrix after prediction:
 [[14265  5583]
 [ 4868 15103]]
Number of mislabeled points out of a total 39819 points : 10451
```

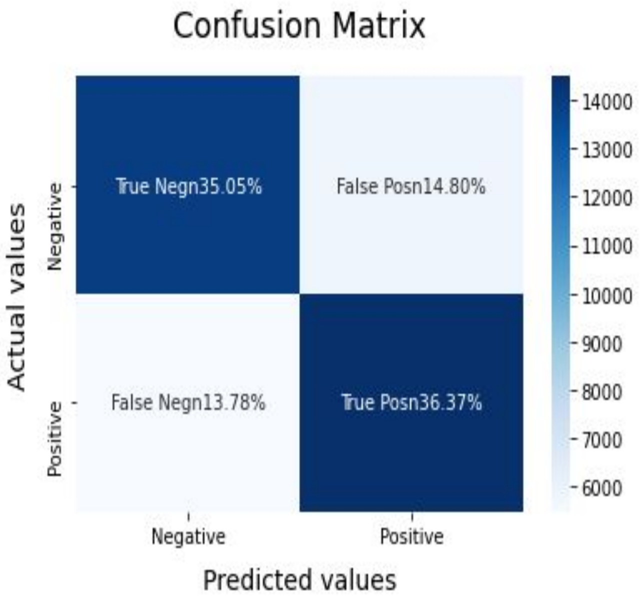


Neural Network without PCA

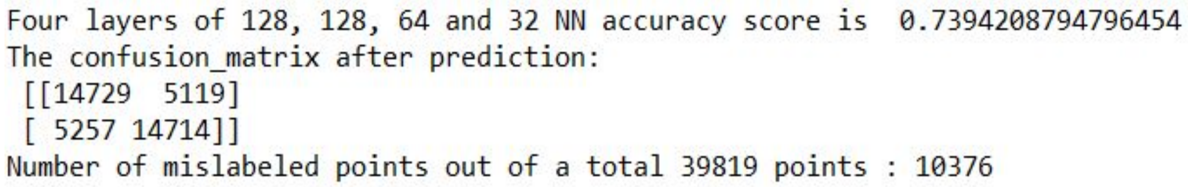
CV Four layers of 128, 128, 64, 32 without PCA:



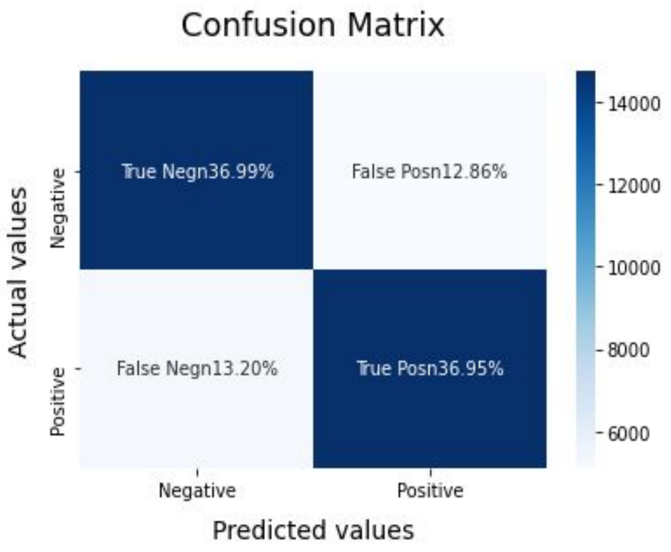
```
Four layers of 128, 128, 64 and 32 NN accuracy score is  0.7142067857053166
The confusion_matrix after prediction:
 [[13956  5892]
 [ 5488 14483]]
Number of mislabeled points out of a total 39819 points : 11380
```



TF-IDF Four layers of 128, 128, 64, 32 without PCA:

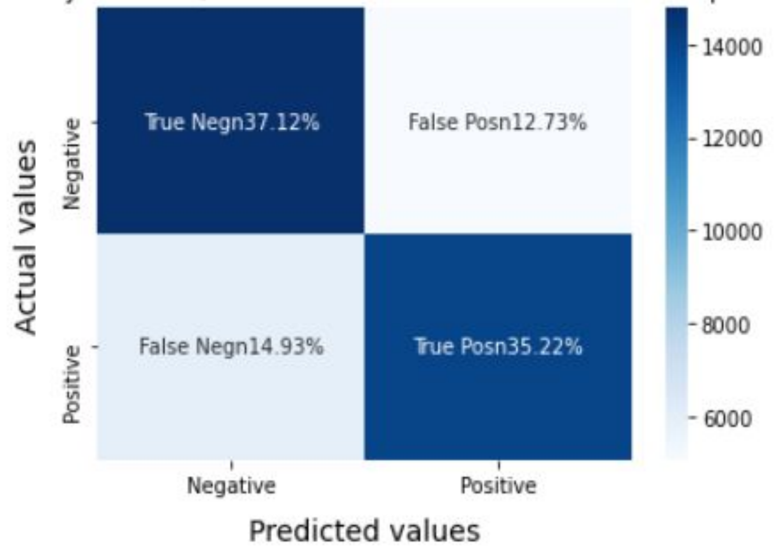


```
Four layers of 128, 128, 64 and 32 NN accuracy score is  0.7394208794796454
The confusion_matrix after prediction:
 [[14729  5119]
 [ 5257 14714]]
Number of mislabeled points out of a total 39819 points : 10376
```



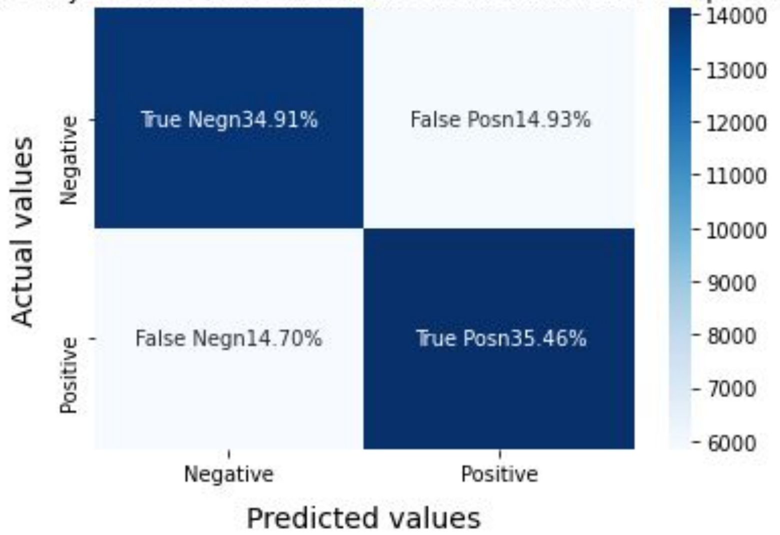
Neural Network with PCA

TF-IDF Three layers of 128, 128, 64 with PCA: The accuracy score is = 0.7234



confusion matrix for Three layers of 128, 128 and 64 NN after PCA with 1151 componenets accuracy 0.723

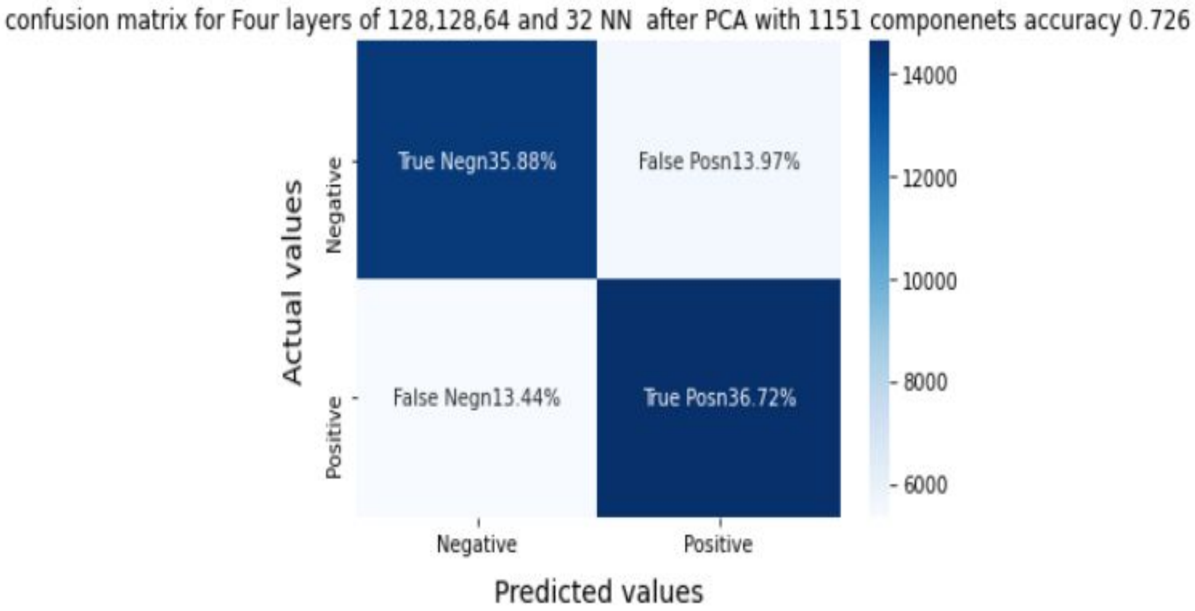
CV Three layers of 128, 128, 64 with PCA The accuracy score is = 0.7037



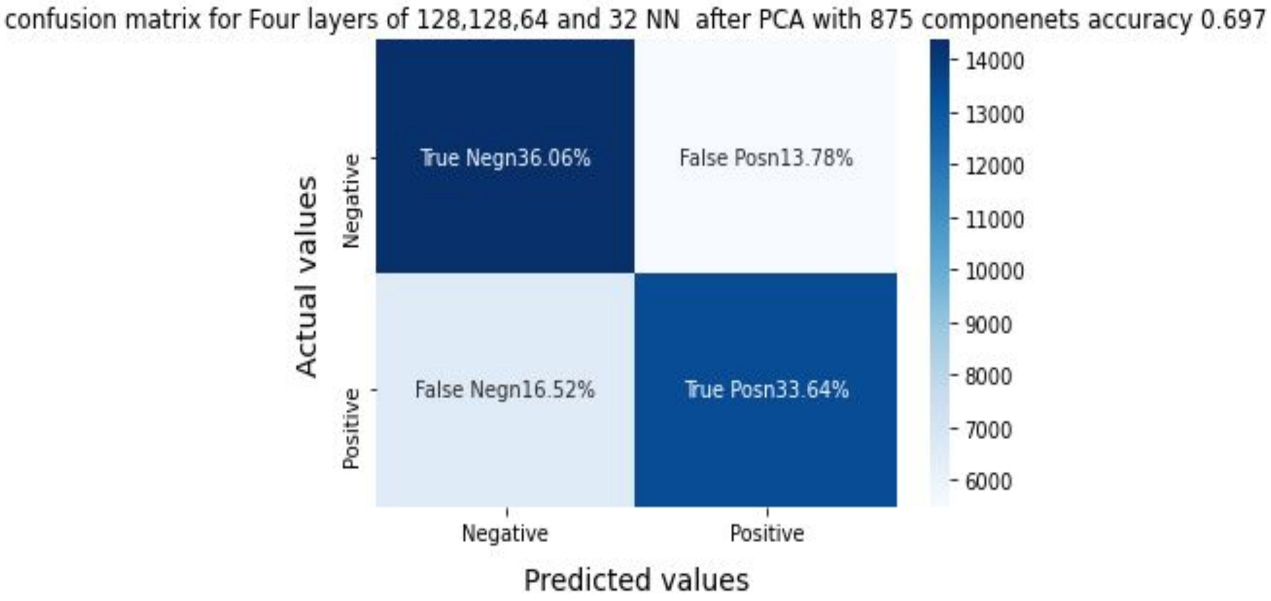
confusion matrix for Three layers of 128, 128 and 64 NN after PCA with 875 componenets accuracy 0.704

Neural Network with PCA

TF-IDF Four layers of 128, 128, 64, 32 with PCA: The accuracy score is =0.7260


confusion matrix for Four layers of 128,128,64 and 32 NN after PCA with 1151 componenets accuracy 0.726

CV Four layers of 128, 128, 64, 32 with PCA: The accuracy score is = 0.6970


confusion matrix for Four layers of 128,128,64 and 32 NN after PCA with 875 componenets accuracy 0.697

Conclusion

- The best model is L1 lasso regularization with SAGA optimization method before PCA using TF-IDF feature extraction model, the accuracy rate is 77.85%, when c =1, followed by L2 SAG PCA using TF-idf, accuracy is 77.81% when c= 1. This is an improvement to our baseline Bernoulli NB model which is 76.67%.
- TF-IDF feature extraction performed better than CV feature extraction to most of the models we have applied.
- PCA does not work well to this project.
- KNN does not work well to this project.
- We can try to increase feature size to improve the performance, or we can try CNN.

Thank you!