Twitter Sentiment Analysis

Final Report

Problem

The aim of this project is to extract the sentiment of tweets and classifying it as either a positive or a negative sentiment. A solution to this problem is important to a wide variety of domains as will be discussed in the following section.

Client

Sentiment analysis is a multidisciplinary necessity. The following are only some of the potential uses of sentiment analysis.

- **Online Commerce:** The sentiments can be extracted from the reviews that the people give about products to understand the general like or dislike of a certain feature or a product. This can be used to make improvements to the particular products
- **Voice of the Market (VOM):** Voice of the Market is about determining what customers are feeling about products or services of competitors. Accurate and timely information from the Voice of the Market helps in gaining competitive advantage and new product development. Detection of such information as early as possible helps in direct and target key marketing campaigns
- **Voice of the Customer (VOC):** Voice of the Customer is concern about what individual customers are saying about products or services. It means analyzing the reviews and feedback of the customers. VOC is a key element of Customer Experience Management. VOC helps in identifying new opportunities for product inventions. Extracting customer opinions also helps identify functional requirements of the products and some non-functional requirements like performance and cost.
- **Brand Reputation Management:** Brand Reputation Management is concern about managing your reputation in market. Opinions from customers or any other parties can damage or enhance your reputation. Brand Reputation Management (BRM) is a product and company focused rather than customer. Now, one-to-many conversations are taking place online at a high rate. That creates opportunities for organizations to manage and strengthen brand reputation.
- **Government:** Sentiment analysis helps government in assessing their strength and weaknesses by analyzing opinions from public

Dataset

The data that I will be using is of 100,000 tweets each marked with either a positive or a negative tweet. The dataset can be easily downloaded from here.

The dataset is available as a csv file and has essentially 2 columns of interest.

1. Sentiment - 0,1 (Negative,Positive)
2. Tweet - The text associated with the sentiment

Data Preprocessing

The tweets were processed according to the following and in the same order.

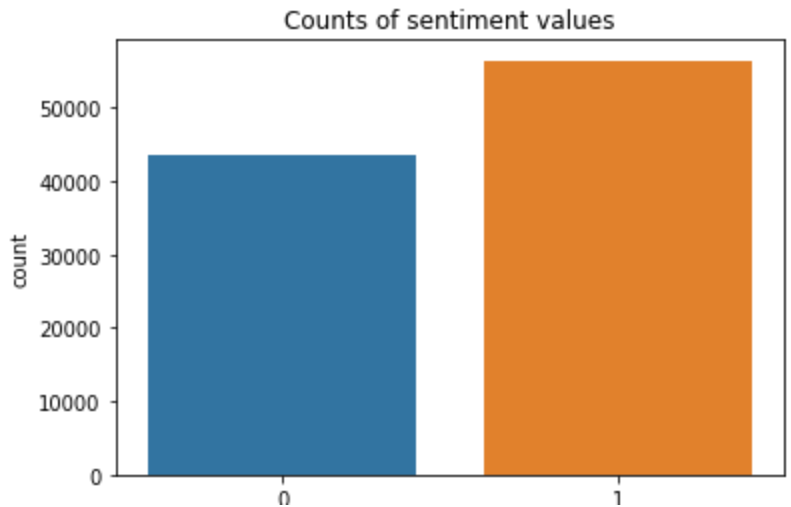
1. Tags removal (i.e '@')
2. Lowercasing
3. Numbers removal
4. HTTP links removal
5. Emojis were processed. Emojis were labelled EMO_POS or EMO_NEG
6. Punctuation removal
7. Removed extra white spaces
8. Words which did not consist of alphabets were removed
9. Stop words were removed (These words add no value to the sentiment of the tweet e.g The, He etc)
10. Character repetitions were removed e.g *funnny* was changed to *funny*
11. Words were lemmatized to bring to their basic form e.g adventurous changed to adventure

After performing these steps I checked to see if there were any tweets which were reduced to an empty string. I could not find any such cases.

Exploratory Data Analysis

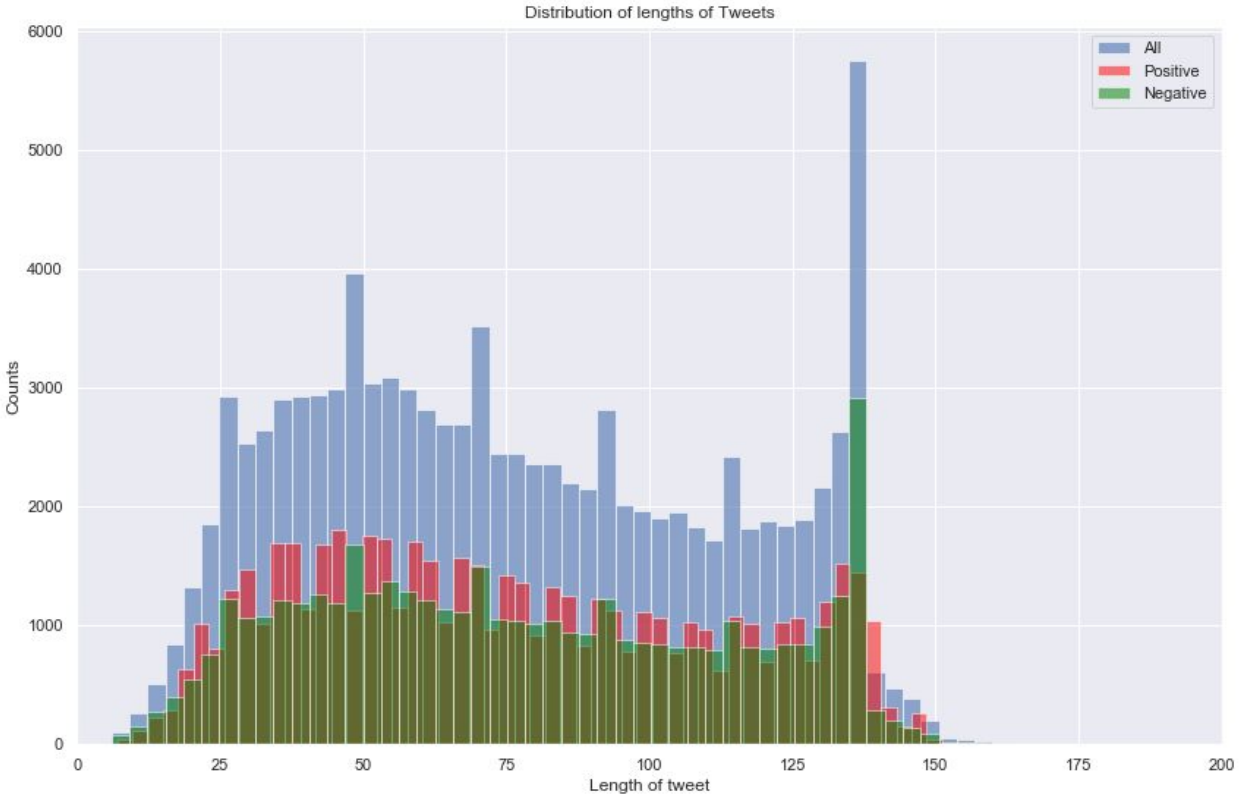
Distribution of sentiments

As can be seen below, both sentiments are somewhat balanced



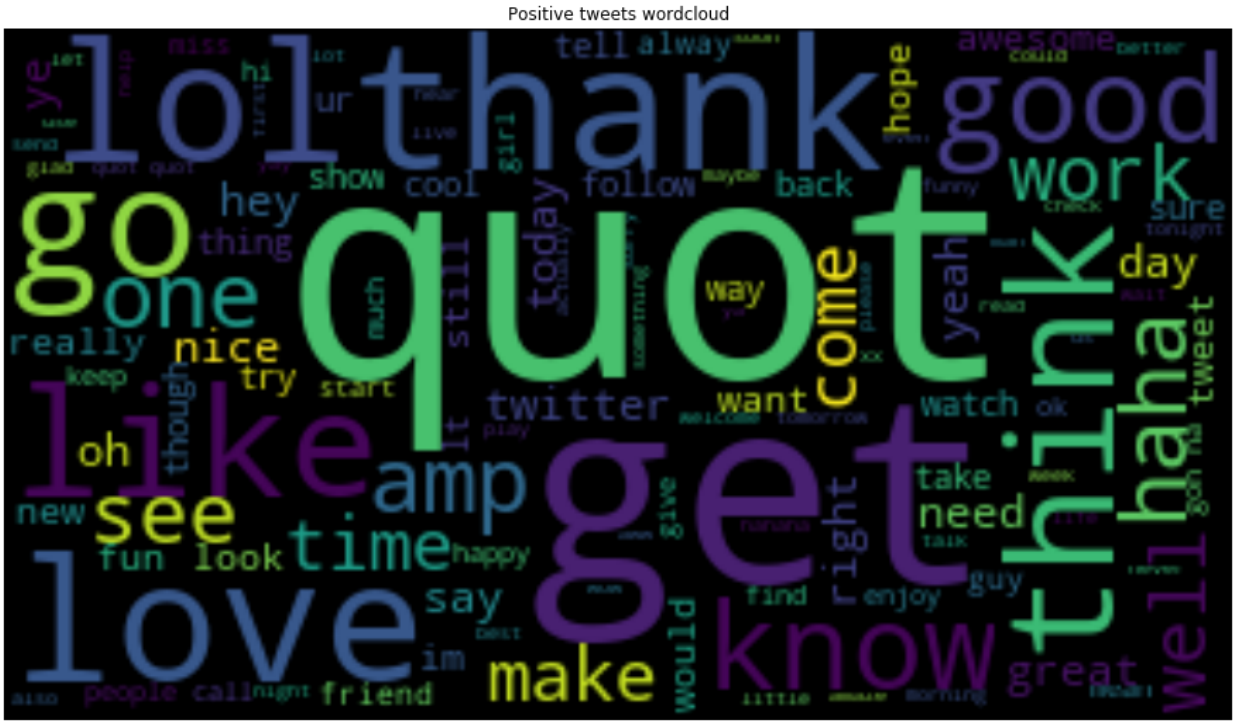
Distribution of lengths of tweets

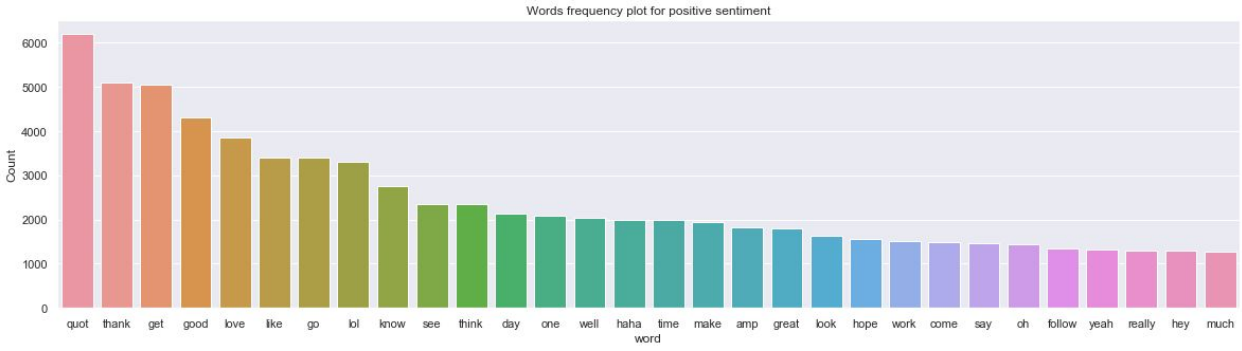
We can also see below that the lengths of both the sentiments follow the same distribution.



Word Clouds and Word frequency plots

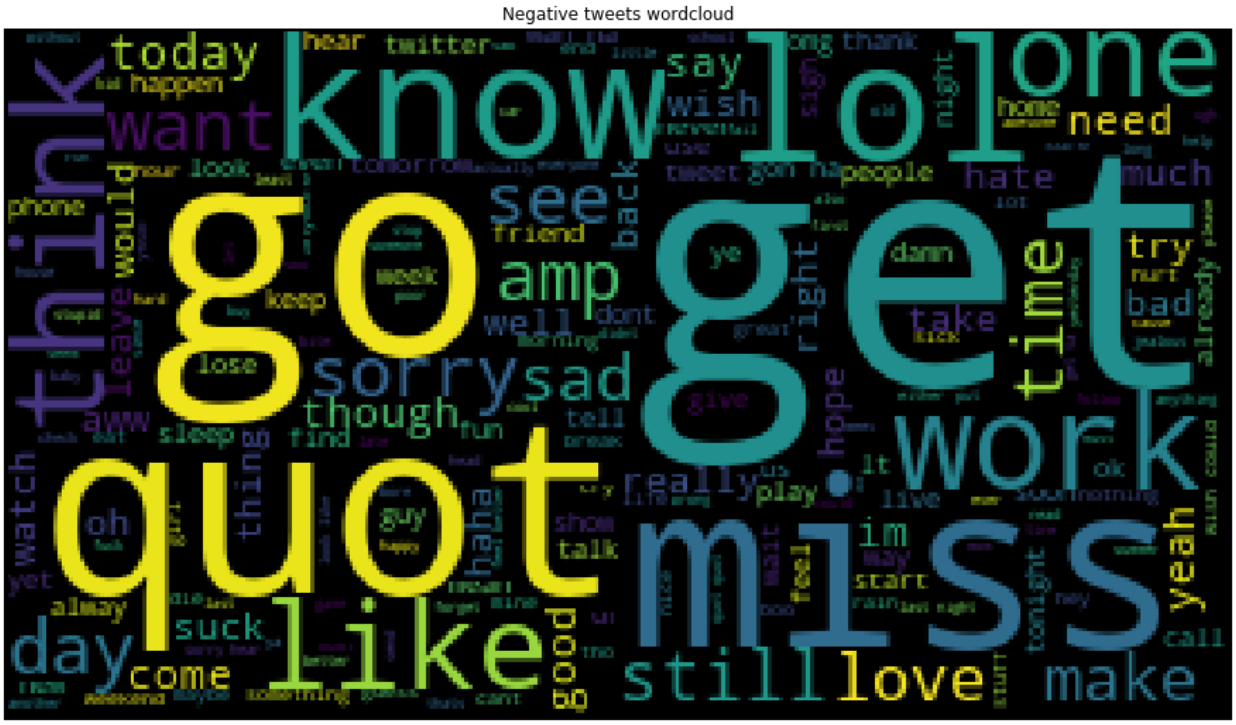
Positive Sentiments

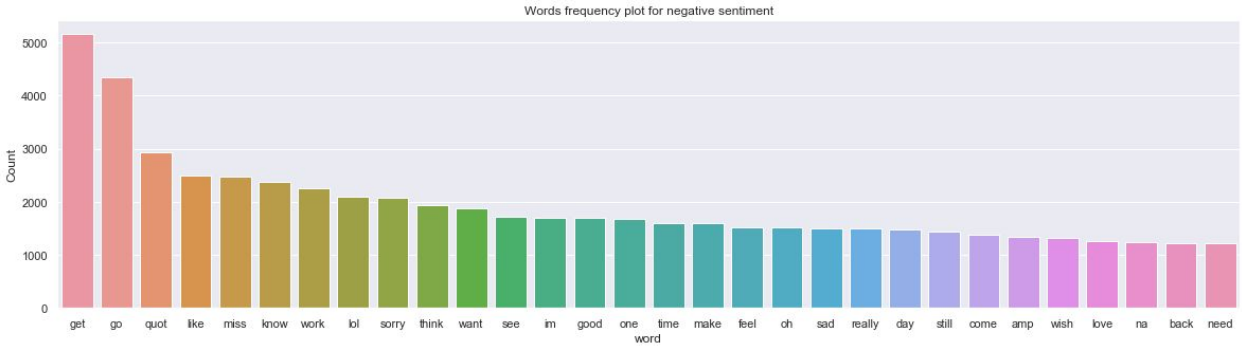




We can see that for the positive sentiment words like lol,thank, love, good show up more frequently.

Negative Sentiments





For the negative sentiment words like miss, work, sad, wish are more frequent.

Modeling strategy

I have approached the sentiment extraction through 3 methods.

1. Bag of Words - Each processed tweet is considered a set of words. (Used the 3000 most frequent words)
2. Term Frequency Inverse Document Frequency - **TF-IDF** is a numerical statistic that is intended to reflect how important a word is to a document in a collection or corpus. Here each word is assigned a weight that reflects how frequent it is in that tweet in comparison to all the tweets. (Used 3000 most frequent words)
3. Word Embeddings - This technique converts the words into mathematical vectors based how their similarity to other words. (1000 word embeddings were used. The vectors were rounded to 3 decimal places to reduce the training time of the models)

For each method 5 machine learning approaches are used

1. Logistic Regression
2. SVM
3. XGBoost
4. NaiveBayes
5. Decision Tree

The dataset is used as per the following scheme for each method - model iteration

	All data	Training data	Cross Validation data	Test data
Percentage	100%	72%	8%	20%
Tweets	100,000	72,000	8,000	20,000

All the models are optimized for accuracy. Each models own subset of hyperparameters are tuned using the cross-validation data. They were cross validated using a shuffled 3-fold split.

The code has been set up so that each iteration receives the same data training and validation data and is entirely reproducible.

Note:

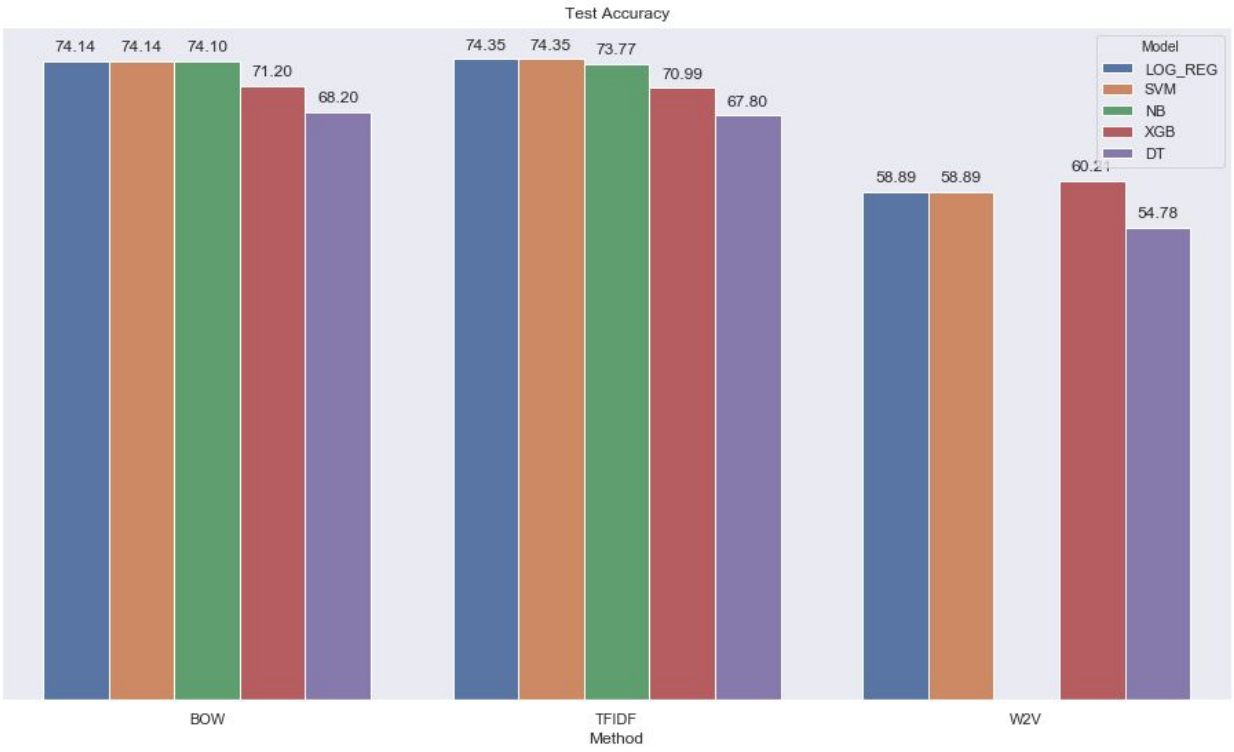
NaiveBayes model for Word2Vec was NOT trained because word vectors can take negative values and NaiveBayes does not handle them. One work around could be to apply MinMax scaling to the Word embeddings but that would void the homogeneity of the validation process.

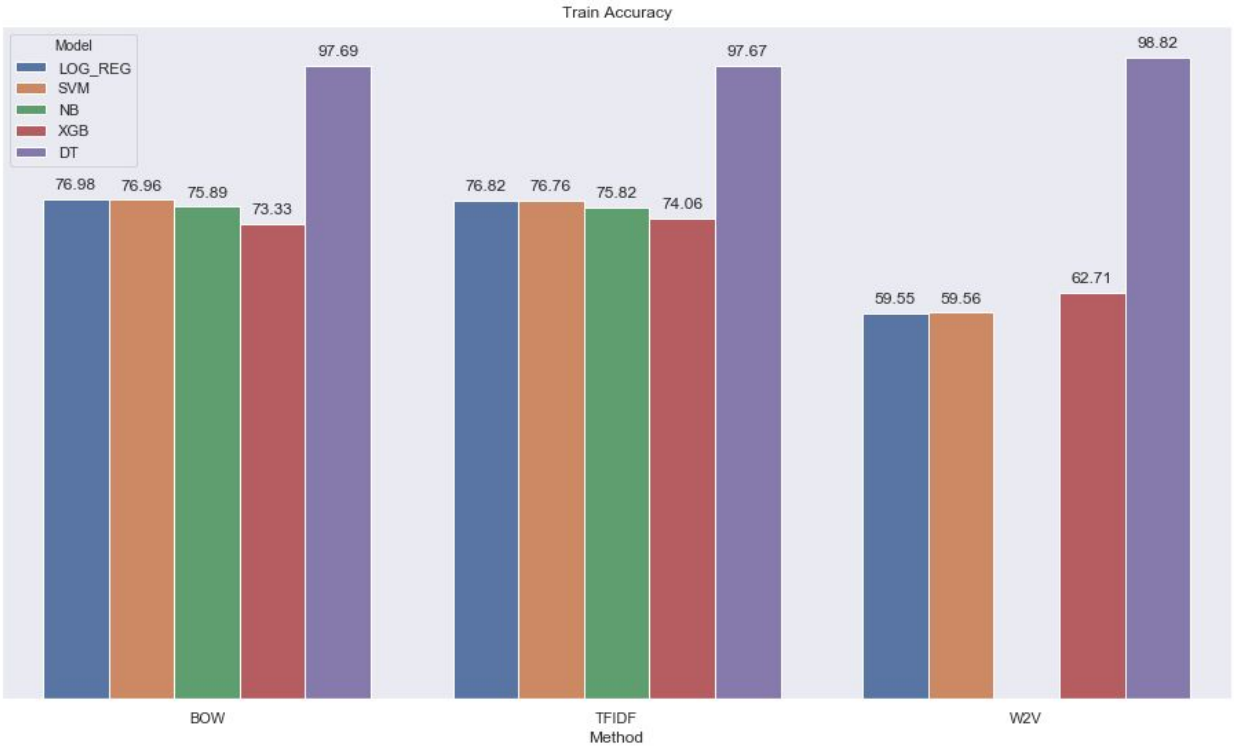
Results

The following table shows the results of all the models and methods. The best performing method-model row is highlighted

Method	Model	TrainAcc	CVAcc	TestAcc	TestF1	TestRecall	TestPrecision	TestROC_AUC
BOW	LOG_REG	0.769830	0.738096	0.741374	0.782981	0.829481	0.741418	0.815214
	SVM	0.769580	0.736983	0.741374	0.782981	0.829481	0.741418	0.815214
	NB	0.758942	0.737095	0.741024	0.776217	0.798542	0.755107	0.812639
	XGB	0.733270	0.715168	0.712021	0.775714	0.885402	0.690207	0.788394
	DT	0.976916	0.668888	0.682018	0.721792	0.733375	0.710569	0.682222
TFIDF	LOG_REG	0.768249	0.739808	0.743524	0.783303	0.824147	0.746317	0.821860
	SVM	0.767643	0.739871	0.743524	0.783303	0.824147	0.746317	0.821860
	NB	0.758217	0.733157	0.737674	0.783277	0.842817	0.731594	0.816258
	XGB	0.740615	0.715230	0.709871	0.773819	0.882379	0.689045	0.787071
	DT	0.976722	0.671601	0.678018	0.719617	0.734619	0.705215	0.671574
W2V	LOG_REG	0.595523	0.593554	0.588909	0.701218	0.857664	0.593041	0.601333
	SVM	0.595561	0.592692	0.588909	0.701218	0.857664	0.593041	0.601333
	XGB	0.627121	0.609594	0.602110	0.702397	0.834815	0.606237	0.632348
	DT	0.988205	0.551437	0.547805	0.598927	0.600284	0.597575	0.535828

Acc is short for Accuracy, **CV** is short for CrossValidation

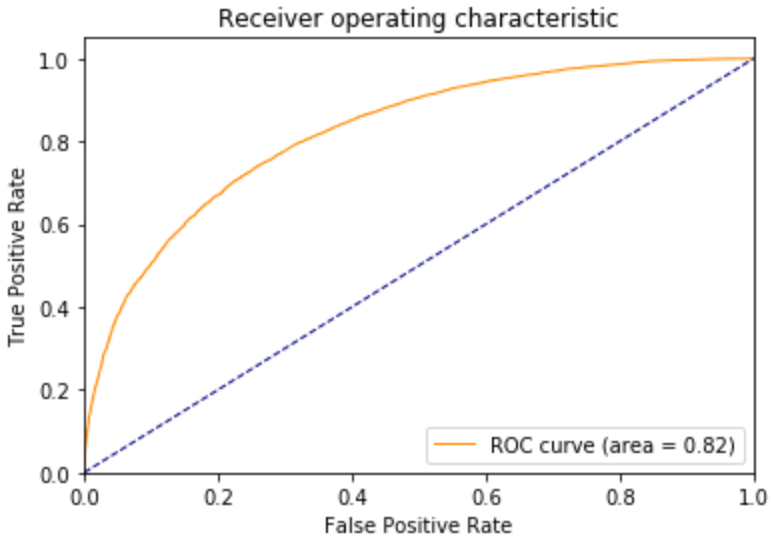




We can see from the results that the all the metrics are higher for **Logistic Regression with TF-IDF**.

We can also notice that the Decision tree models were overfit for all methods while the rest were fine.

The following are the ROC and the class-wise metrics for the best performing (Logistic Regression - TFIDF) model



	Negative	Positive	micro avg	macro avg	weighted avg
f1-score	0.705046	0.796944	0.759474	0.750995	0.756934
precision	0.756302	0.761418	0.759474	0.758860	0.759191
recall	0.660296	0.835946	0.759474	0.748121	0.759474

Error Analysis

We have seen the Logistic Regression with TF-IDF performs the best. The following is a sample of misclassifications.

	sentiment	preds	tweet
0	1	0	@chrisledlin why r u goin to the gym for r u on a diet too or r u just keepn fit?
1	0	1	@alexanderchee i dressed as wojnarowcz's rimbaud photo series for halloween - no one got it
2	0	1	i wanted to win but its allll good lol
3	0	1	@chrysshanice88 you hella drove!! lol..& bitch where you at? you dont love me nomoe
4	0	1	@coryj111 Uhm...duh. Neil Patrick Harris. The hottest thing on this earth. I bet parties for the Tony's in New York are 10 times better
5	0	1	#uknowuliveinthehood when all the corner stores and family owned businesses around when you were growin up are gone
6	1	0	@BlayzeThePro Cuz I Can,.!!!!!
7	0	1	@albertsthings im on 1800/2000 but i still need to do.. two conclusions .. and my other point. I WROTE TOO MUCH! i so cbb to edit!
8	1	0	@ Gaia by car! Try to find a damn parking spot at the WUR campus: ptvrdr mission impossible!!
9	0	1	@bridgers i have one too

Observations:

1. We can see that some tweets cannot be classified as either positive or negative. They don't have a sentiment in them. For example tweet 0 is neither positive or negative. However the truth value associated with it is positive. It should be classified as Neutral. More examples are 1,9
2. Some tweets have both a positive and a negative sentiment e.g *'i wanted to win but it's allll good lol'*. They can not be labelled either positive or negative in entirety.
3. Wrong labeling in the Training data e.g. 8. This should have been a negative sentiment.

Other than the observations highlighted further improvement can be made by tuning the model further or trying a more sophisticated technique.

Further improvements

1. Use more features to train the models for TFIDF.
2. Use a sophisticated model like deep learning where the models are trained with the context of a window of words.
3. BOW and TFIDF can be used with 2-grams or 3-grams
4. Training dataset can be refined to be more accurate.
5. Exclamation marks could be handled specially during the preprocessing of the tweets.