Twitter Sentiment Analysis using XGBoost Machine Learning Algorithm in Python

Abstract

The purpose of this project is to predict sentiment for the given Twitter post using Python and Plotly(for analysis). Sentiment analysis can predict many different emotions but in this project only 3 major were considered: positive, negative and neutral. Applying Extreme Gradient Boosting algorithm, the classification accuracy.

Topics Covered

1. Introduction

Twitter users around the world post around 350,000 new Tweets every minute, creating 6,000 140-character long pieces of information every second. Twitter is now a hugely valuable resource from which you can extract insights by using text mining tools like sentiment analysis.

With sentiment analysis, we can generate insights about consumers reactions to announcement, opinions on products or brands, and even track opinion about events as they unfold. They classify Tweets for a query term into negative or positive sentiment. They collect training dataset automatically from Twitter. To collect positive and negative tweets, they query twitter for happy and sad emoticons.

- Happy emoticons are different versions of smiling face, like ":)", ":-)", ":)", ":D", "=)" etc.
- Sad emoticons include frowns, like ":(", ":-(", ":(" etc.

For this reason, you'll often hear sentiment analysis referred to as "opinion mining".

There is a lot of domain in which we use sentiment analysis.

i. Ecommerce

In ecommerce activity, websites allows their users to submit their experience about shopping and product qualities. They provide summary for the product and different features of the product by assigning ratings or scores.

ii. Market Voice

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.

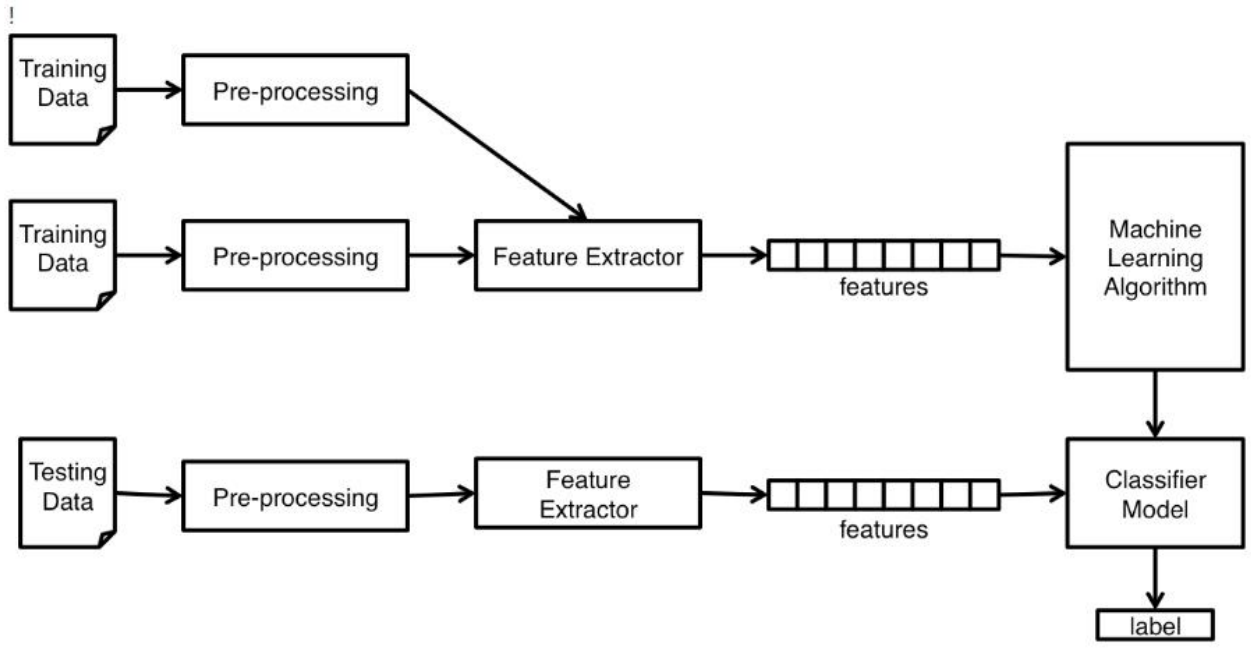
iii. Customer Voice

The voice of customer is related to what individual customer is saying about products and services i.e analysing the reviews and feedback of customers.

iv. Government

Sentiment analysis helps government in assessing their strength and weaknesses by analyzing opinions from public.

2. Approach for Twitter Sentiment Analysis



3. Twitter sentiment Corpus

An essential part of creating a Sentiment Analysis algorithm is to have a comprehensive dataset or corpus to learn from, as well as a test dataset to ensure that the accuracy of your algorithm meets the standards you expect.

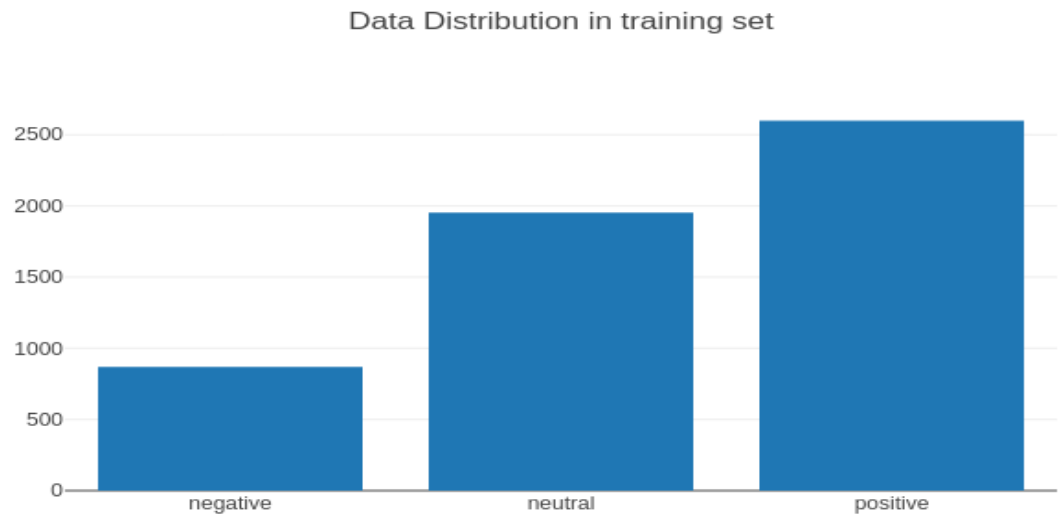
Positive For showing positive sentiment towards the topic

Positive For showing no or mixed or weak sentiments towards the topic

Negative For showing negative sentiment towards the topic

Irrelevant For non English text or off-topic comments

We get the statistics of the negative, positive and neutral words as below,



4. Pre Processing

User-generated content on the web is seldom present in a form usable for learning. It becomes important to normalize the text by applying a series of pre-processing steps. We have applied an extensive set of pre-processing steps to decrease the size of the feature set to make it suitable for learning algorithms.

4.1 Hashtags

A hashtag is a word or an un-spaced phrase prefixed with the hash symbol (#). These are used to both naming subjects and phrases that are currently in trending topics. For example, #mobile, #AI

Regular Expression: #(\w+)

Replace Expression: HASH_\1

4.2 Handles

Every Twitter user has a unique username. Any thing directed towards that user can be indicated be writing their username preceded by '@'. Thus, these are like proper nouns. For example, @Apple
Regular Expression: @(\w+)
Replace Expression: HNDL_\1

4.3 URLs

Users often share hyperlinks in their tweets.. Presence of a URL can be an important feature. Regular expression for detecting a URL is fairly complex because of different types of URLs that can be there, but because of Twitter's shortening service, we can use a relatively simple regular expression.

Regular Expression: (http|https|ftp)://[a-zA-Z0-9\\./]+
Replace Expression: URL

4.4 Emotions

Use of emoticons is very prevalent throughout the web, more so on micro blogging sites.We identify the following emoticons and replace them with a single word.

4.5 Punctuations

We replace every word boundary by a list of relevant punctuations present at that point.

4.6 Repeating Characters

People often use repeating characters while using colloquial language, like "Hellooo", "Hiiiii" As our final pre-processing step, we replace characters repeating more than twice as two characters.
Regular Expression: (.)\1{1,}
Replace Expression: \1\1
If we apply the above steps in dataset we get results as:

	id	emotion	text
1	635930169241374720	neutral	IOS App Transport Security Mm need to check if...
2	635950258682523648	neutral	Mar if you have an iOS device you should downl...
3	636030803433009153	negative	my phone does not run on latest IOS which may ...
4	636100906224848896	positive	Not sure how to start your publication on iOS ...
5	636176272947744772	neutral	Two Dollar Tuesday is here with Forklift Quick...

4.7 Lexicon Normalization

Another type of textual noise is about the multiple representations exhibited by single word.

Stemming Algorithm

Stemming is the process of reducing a word into its stem, i.e. its root form. The root form is not necessarily a word by itself, but it can be used to generate words by concatenating the right suffix.

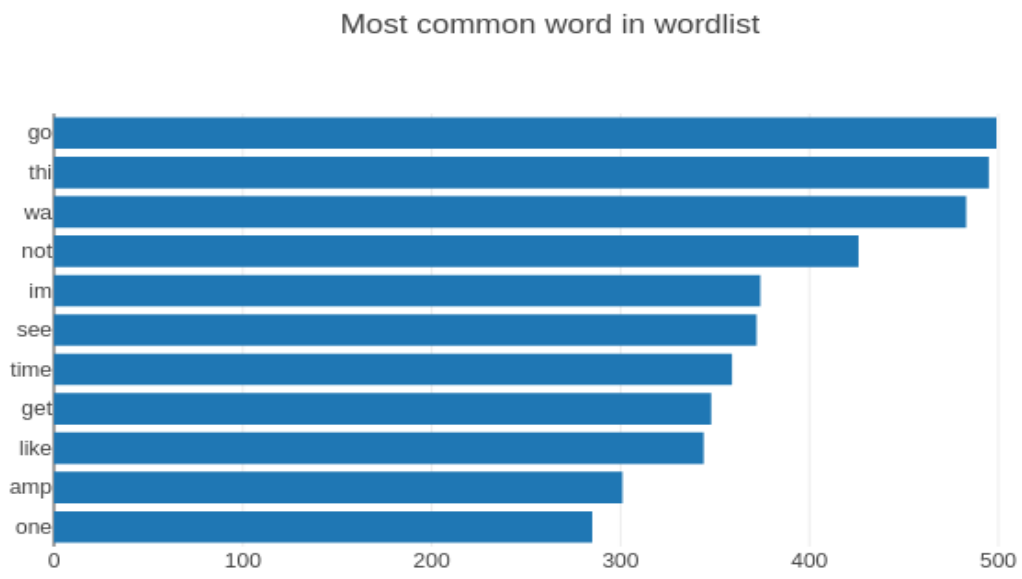
Lemmatisation

Lemmatization is the process of normalizing a word rather than just finding its stem. In the process, a suffix may not only be removed, but may also be substituted with a different one. It may also involve first determining the part-of-speech for a word and then applying normalization rules.

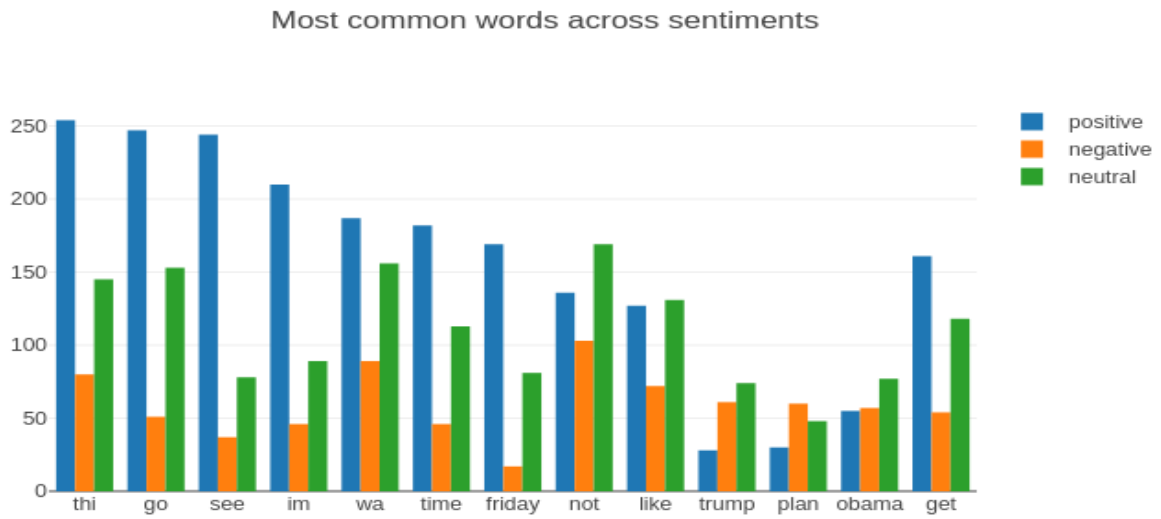
If we apply the above steps, then we can get the following output as below,

	id	emotion	text	tokenized_text
1	635930169241374720	neutral	[io, app, transport, secur, mm, need, to, chec...	[IOS, App, Transport, Security, Mm, need, to, ...
2	635950258682523648	neutral	[mar, if, you, have, an, io, devic, you, shoul...	[Mar, if, you, have, an, iOS, device, you, sho...
3	636030803433009153	negative	[my, phone, doe, not, run, on, latest, io, whi...	[my, phone, does, not, run, on, latest, IOS, w...
4	636100906224848896	positive	[not, sure, how, to, start, your, public, on, ...	[Not, sure, how, to, start, your, publication,...
5	636176272947744772	neutral	[two, dollar, tuesday, is, here, with, forklif...	[Two, Dollar, Tuesday, is, here, with, Forklif...

Building the most common wordlist using plotly,



Another view of most common words is,



5. Experiment and Classification using XGBoost Machine Learning Algorithm

XGBoost belongs to a family of boosting algorithms that convert week learners into strong learners. A week learner is one which is slightly better than random guessing. Boosting is a sequential process: i.e. trees are grown using the information from a previously grown tree one after another. This process slowly learns from the data and tries to improve its prediction in subsequent iterations. XGBoost is an implementation of gradient boosted decision trees designed for speed and performance. It uses gradient boosting framework at core. This ML algorithm have following features as below,

Parallel Computing: It is enabled with parallel processing i.e., when you run xgboost, by default, it would use all the cores of your laptop/machine.

Regularization: Regularization is a technique used to avoid overfitting in linear and tree-based models.

Missing Values: XGBoost is designed to handle missing values internally. The missing values are treated in such a manner that if there exists any trend in missing values, it is captured by the model.

Flexibility: In addition to regression, classification, and ranking problems, it supports user-defined objective functions also. An objective function is used to measure the performance of the model given a certain set of parameters. Furthermore, it supports user defined evaluation metrics as well.

Save and Reload: XGBoost gives us a feature to save our data matrix and model and reload it later. Suppose, we have a large data set, we can simply save the model and use it in future instead of wasting time redoing the computation.

Tree Pruning: Unlike GBM, where tree pruning stops once a negative loss is encountered, XGBoost grows the tree upto max_depth and then prune backward until the improvement in loss function is below a threshold.

Python Code

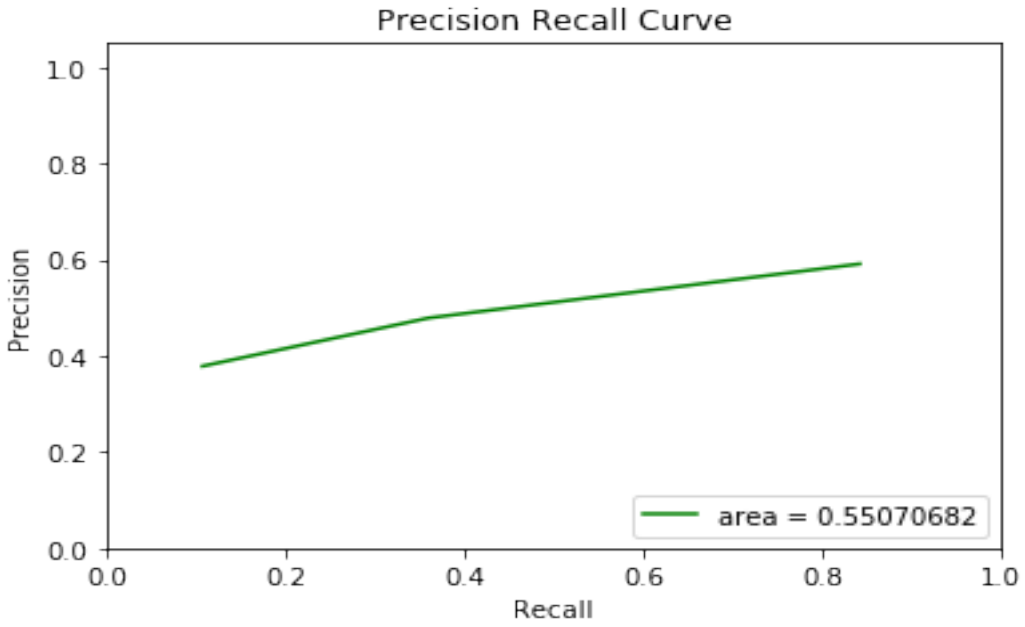
you can find code in following link:

After applying XGBoost Machine Learning Algorithm, we get final Accuracy as,

```
Testing XGBClassifier

Learing time 70.33846664428711s
Predicting time 0.1466081142425537s
************ Results ************
            Negative     Neutral     Positive
F1       [0.16716418 0.41090555 0.69450317]
Precision[0.37837838 0.47845805 0.59082734]
Recall   [0.10727969 0.36006826 0.84230769]
Accuracy 0.5507068223724647
```

Precision and Recall curve as below,



6. Conclusion and Observation

Now we can say prediction of text sentiment is non-trivial task for machine learning. A lot of preprocessing is required just to be able to run any algorithm. Simple bag-of-words was definitely not enough to obtain satisfying results, thus a lot of additional features were created basing on common sense. The thing that could possibly improve classification results will be to add a lot of additional examples like increase training dataset.