Real Time sentiment analysis with machine learning

Shrikant Patro[1], Harshit Matur[2], Makubhai Shahin Shoukat[3]

[1]*Vellore Institute of Technology, Chennai, India*

[2]*Vellore Institute of Technology, Chennai, India*

[3]*Vellore Institute of Technology, Chennai, India*

E-mail: shrikantjagannath.2018@vitstudent.ac.in

Abstract

Startup and Business houses are growing and the customer base also. So, it is very important for the Business to flourish and the customer satisfaction level and their views on the product is very important. It is especially crucial when something goes wrong with the product. Sentimental analysis is the well-known Natural language processing problem which goal is to determine whether particular text is positive, negative and neutral. The variety of emotions is richer what makes this issue harder to solve, but for our purposes sentiment is just enough. Twitter was our first choice, when we thought about the media to be monitored, as it is commonly used and gives an opportunity to automatically retrieve the messages which include given phrase or hashtag. It is also a tool which people use to inform the others about things happening right now. It looks like designed for real-time analytics.

Literature Review

In this paper, This paper look at one such popular micro-blog called Twitter and build models for classifying "tweets" into positive, negative and neutral sentiment. It build models for two classification tasks: a binary task of classifying sentiment into positive and negative classes and a 3-way task of classifying sentiment into positive, negative and neutral classes. We experiment with three types of models: unigram model, a feature based model and a tree kernel based model. For the feature based model we use some of the features proposed in past literature and propose new features. For the tree kernel based model we design a new tree representation for tweets. Here, feature based model that uses only 100 features achieves similar accuracy as the unigram model that uses over 10,000 features. Our tree kernel based model outperforms both these models by a significant margin. Unigram model is explored [1].

Sentiment analysis is the process of extracting emotions or opinions from a piece of text for a given topic it allow us to understand the attitudes, opinions and emotions in the text. In it user's likes and dislikes are captured from web content. It involves predicting or analyzing the hidden information present in the text. This hidden information is very useful to get insights of user's likes and dislikes. The aim of sentiment analysis is to determine the attitudes of a writer or a speaker for a given topic. Sentiment analysis can also be applied to audio, images and videos. Important notions are Subjectivity/ Objectivity, Polarity and Sentiment level. Polarity is further divided into 3 category and this category are positive, negative and neutral tweet. Sentiment level tweet are Document level , Sentence level and phrase level [2].

A massive volume of both structured and unstructured multimedia data is being uploaded on the Internet due to rapidly growing ubiquitous web access over the world. However, analyzing those raw media resources to discover their hidden semantics is becoming a challenging task. It results into a difficult to retrieve the right type of media to satisfy multimedia content consumers. So, improving

the search ability of multimedia contents on the web is one of the most appealing demands, especially for online audio/video content providers. Even if there are a lot of effective approaches for indexing textual contents, they cannot be applied to index media type such as audio and video, unless we transform them to some form of text, and add advanced metadata annotations using contextual information around the target media. This problem motivated for the genesis of the ongoing EU research project called Media in Context (MICO). MICO mainly aims at providing cross-media analysis framework, including orchestrated chain analysis components to extract semantics from the media in a cross media context. We are mainly concerned with the textual analysis aspect of MICO, including sentiment and discourse analysis, language identification, and named entity recognition [6].

Sentiment analysis copes with the task of opinion mining from text. With the growth of user generated texts on the web, exploring the method to automatically extract and classify opinions from those texts would be enormously helpful to individuals, business and government intelligence and in decision-making. Some of the early research works in this area include in these works different methods have been used for detecting the polarity of product reviews and movie reviews respectively.[6]

Existing System

The existing system proposes the sentiment analysis using two broadly classified approach as machine learning approach and Lexicon based approach. In Machine learning approach, it used both supervised and unsupervised learning technique. The supervised technique is classified as Decision tree classifier, Linear Classifier, Rule-Based classifier and probabilistic classifier. Algorithm included are, Support vector Machine, Neural Network, Naïve Bayes, Bayesian Network and Maximum Entropy. The lexicon-based Approach. This include Dictionary-based approach, corpus-based approach, statistical and Semantic. The Feature selection in sentimental classification include performing Terms presence and frequency, Parts of Speech (POS), Opinion words and phrase and Negations.

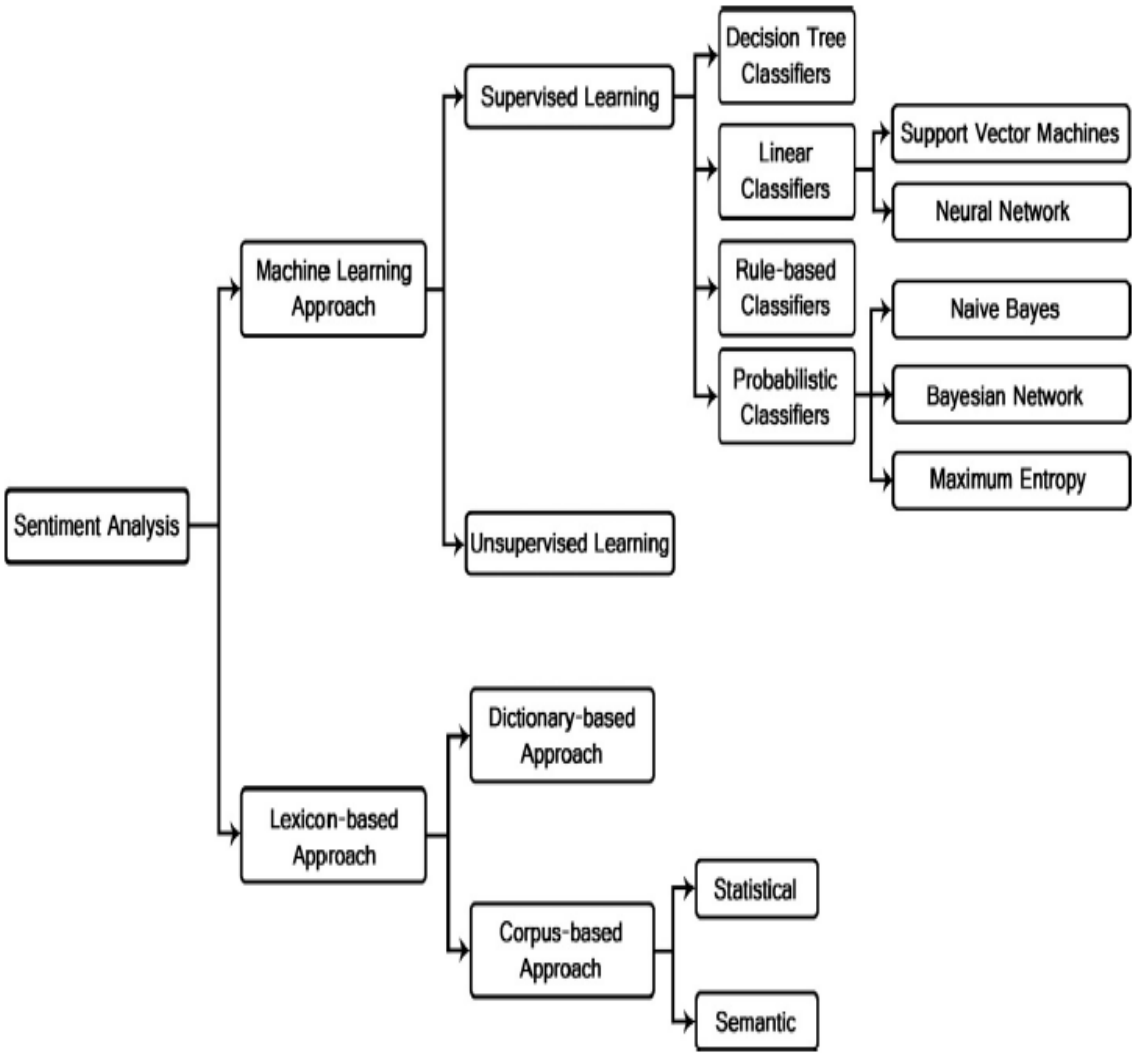


Fig 1. Existing System architecture

Pros

Usage of Machine learning provide scope for different ML Algorithm for Sentimental analysis.

Different approach are suitable for the analysis are supported by machine learning. Any of the approaches can be selected based on the type of Data.

Cons

When Data Size is extremely high then due to high dimension, Machine learning technique faces the problem of the curse of Dimensionality.

Proposed Work

The proposed system is based on the Deep Learning Technique. This technique will overcome the drawback of the ML. This system include fitting a deep learning model with Keras, Identify and deal with overfitting, use word embedding and build on the pre trained model. The packages required for the system include the basic packages like panadas, numpy, re, collections and matplotlib. Packages required for preprocessing and modeling of the data include sklearn, nltk, keras, sklearn, models, layers and regularizers.

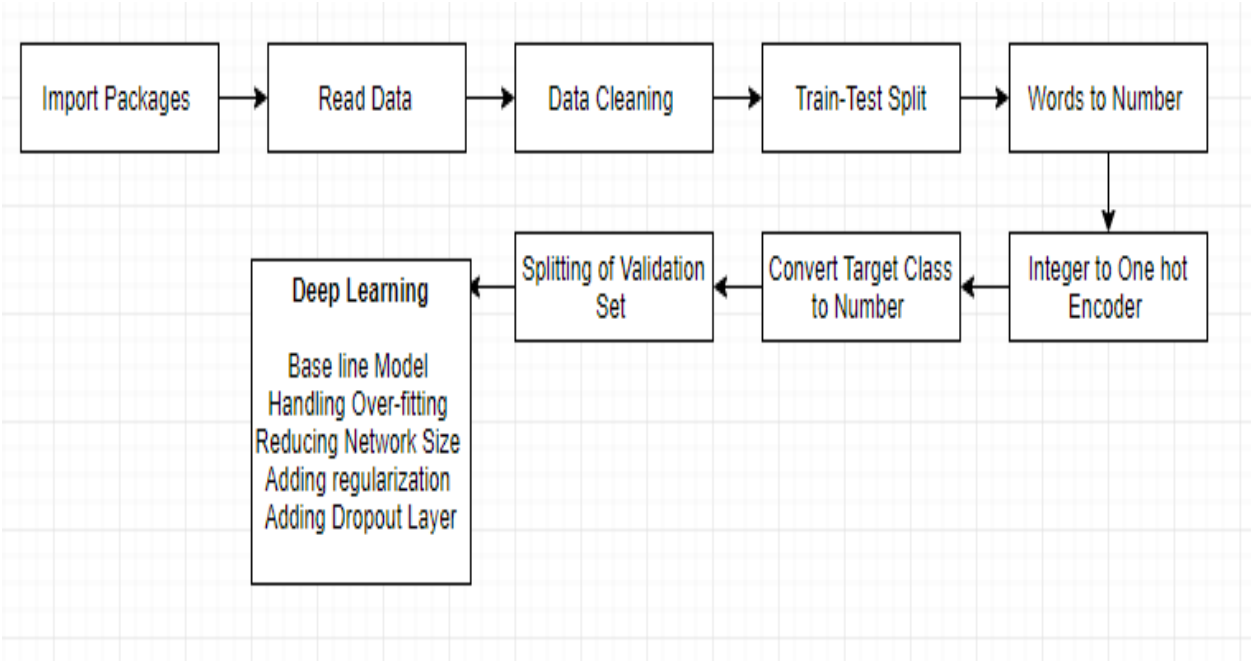
Data preparation process include Data Cleaning as remove_stopwords, remove mentions. The evolution of the model performance needs to be done on different test set. As such, we can estimate how well the model generalizes. This is done with the Train-Test split of scikit learn. To convert words to number in order use this text as input to model. We first need to convert the tweet's words into tokens, which simply means converting the words to integers that refer to an index in a dictionary. Here we will only keep the most frequent words in the train set. After having created the dictionary we can convert the text to integer indexes. This is done with the text to sequences method of the Tokenizer. Converting the target classes to one hot encoder with the to_categorical method in keras.

Deep, learning implementation include the baseline model, Handling overfitting and reducing the network size. Addition of the dropout layer we will try to add dropout layer. The model with dropout layers starts overfitting bit later then baseline model.

We examine sentiment analysis on Twitter data. The contributions of this paper are:
We introduce POS-specific prior polarity features. We explore the use of a tree kernel to obviate the need for tedious feature engineering. The new features (in conjunction with previously proposed features) and the tree kernel perform approximately at the same level, both outperforming the state-of-the-art baseline.

Deep learning is used to overcome Drawback of the Machine learning algorithm. When Data set size is extremely high and Textual data then the performance of the Deep learning is better the Machine Learning.

System Architecture



Modules

1. Data Preparation.

2. Deep learning

Module 1: Data Preparation

Import Basic. Data Modeling and Preparation packages. Data Cleaning will remove the stop words. These word do not have any value for predicting the sentiment. As model is developed it can be used to clean other raw dataset. Train-Test Split is used to evaluate the performance needs to be done on a separate test set. We can estimate how such model generalizes. This is done with tarin_test_split method of scikit-learn. For application of Machine learning technique the document consisting of words must be converted into number.

To use the text as input for a model. Convert tweet word into tokens, which simply means converting the words to an integer that refer to its syntax in dictionary. Here, we will only keep the frequent words in train set. The words can be cleaned up the text by applying filters and putting the words to lowercase. Words are separated by spaces. After having created dictionary we can convert the text to a list of integer indexes. This is done with the text_to_sequences method of the Tokenizer.

This integer now are converted into one hot encoded feature. We need to convert the target classes to numbers as well, which in turn are one-hot-encoded with the to_categorical method in keras. Splitting of a validation set. Now that our data is ready, we split of a validation set. The Validation set will be used to evaluate the model performance when we tune the performance of the model.

Module 2: Deep Learning

Baseline model

We start with a model with 2 densely connected layers of 64 hidden elements. The *input_shape* for the first layer is equal to the number of words we allowed in the dictionary and for which we created one-hot-encoded features.As we need to predict 3 different sentiment classes, the last layer has 3 hidden elements. The *softmax* activation function makes sure the three probabilities sum up to 1.

In the first layer we need to estimate 640064 weights. This is determined by (nb inputs * nb hidden elements) + nb bias terms, or (10000 x 64) + 64 = 640064
In the second layer we estimate (64 x 64) + 64 = 4160 weights
In the last layer we estimate (64 x 3) + 3 = 195 weights. Because this project is a multi-class, single-label prediction,

Reducing the network's size

We reduce the network's size by removing additional neurons use categorical_crossentropy as the loss function and softmax as the final activation function. We fit the model on the remaining train data and validate on the validation set. We run for a predetermined number of epochs and will see when the model starts to overfit.

To evaluate the model performance, we will look at the training and validation loss and accuracy. the validation loss starts to increase as from epoch 4. The training loss continues to lower, which is normal as the model is trained to fit the train data as good as possible. Just as with the validation loss, the validation accuracy peaks at an early epoch. After that, it goes down slightly. So to conclude, we can say that the model starts overfitting as from epoch 4.

Handling overfitting

Now, we can try to do something about the overfitting. There are different options to do that.

- Option 1: reduce the network's size by removing layers or reducing the number of hidden elements in the layers
- Option 2: add regularization, which comes down to adding a cost to the loss function for large weights

Option 3: adding dropout layers, which will randomly remove certain features by setting them to zeroving one layer and lowering the number of hidden elements in the remaining layer to 32. We can see that it takes more epochs before the reduced model starts overfitting (around epoch 10). Moreover, the loss increases much slower after that epoch compared to the baseline model. To address overfitting, we can also add regularization to the model. Let's try with L2 regularization. For the regularized model we notice that it starts overfitting earlier than the baseline model. However, the loss increases much slower afterwards. The model with dropout layers starts overfitting a bit later than the baseline model. The loss also increases slower than the baseline model.

Design and Analysis

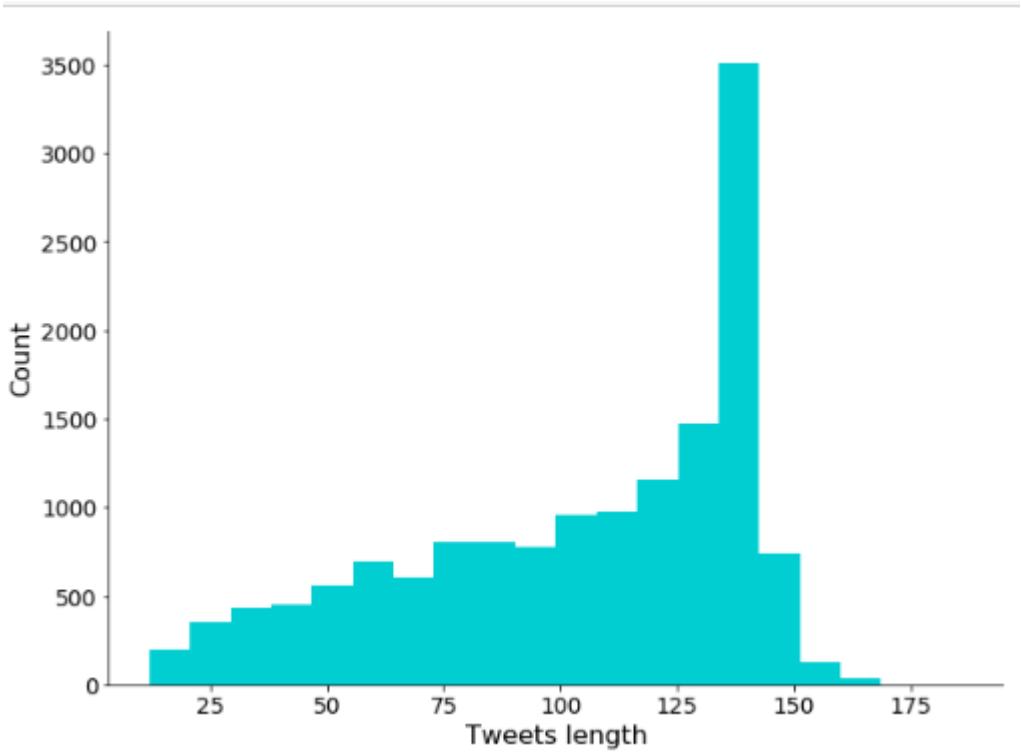
Python is the programming language and the Deep learning technique is used together to achieve class of reviews whether it is good, bad or Neutral. Python is a popular programming language. It was created by Guido van Rossum, and released in 1991. It is used for web development (server-side), software development, mathematics, system scripting. Python can be used on a server to create web applications.

- Python can be used alongside software to create workflows.
- Python can connect to database systems. It can also read and modify files.
- Python can be used to handle big data and perform complex mathematics.
- Python can be used for rapid prototyping, or for production-ready software development.
- Python works on different platforms (Windows, Mac, Linux, Raspberry Pi, etc).
- Python has a simple syntax similar to the English language.
- Python has syntax that allows developers to write programs with fewer lines than some other programming languages.
- Python runs on an interpreter system, meaning that code can be executed as soon as it is written. This means that prototyping can be very quick.
- Python can be treated in a procedural way, an object-orientated way or a functional way.

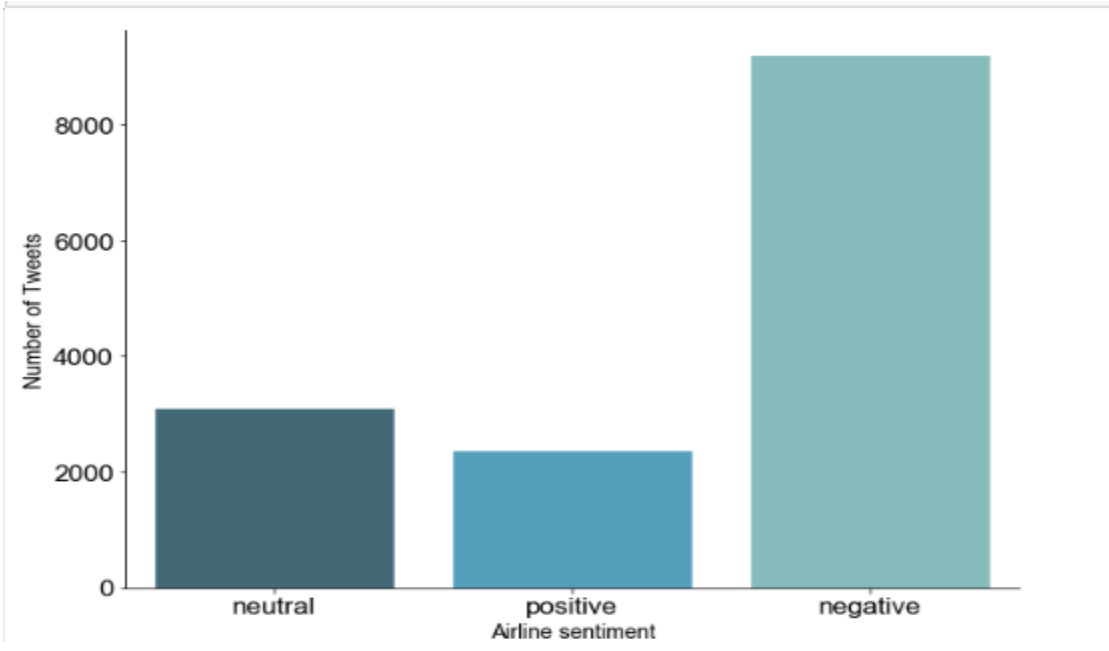
The most recent major version of Python is Python 3, which we shall be using in this tutorial. However, Python 2, although not being updated with anything other than security updates, is still quite popular.In this tutorial Python will be written in a text editor. It is possible to write Python in an Integrated Development Environment, such as Thonny, Pycharm, Netbeans or Eclipse which are particularly useful when managing larger collections of Python files. Python was designed for readability, and has some similarities to the English language with influence from mathematics.Python uses new lines to complete a command, as opposed to other programming languages which often use semicolons or parentheses. Python relies on indentation, using whitespace, to define scope; such as the scope of loops, functions and classes. Other programming languages often use curly-brackets for this purpose.

Results and Discussion

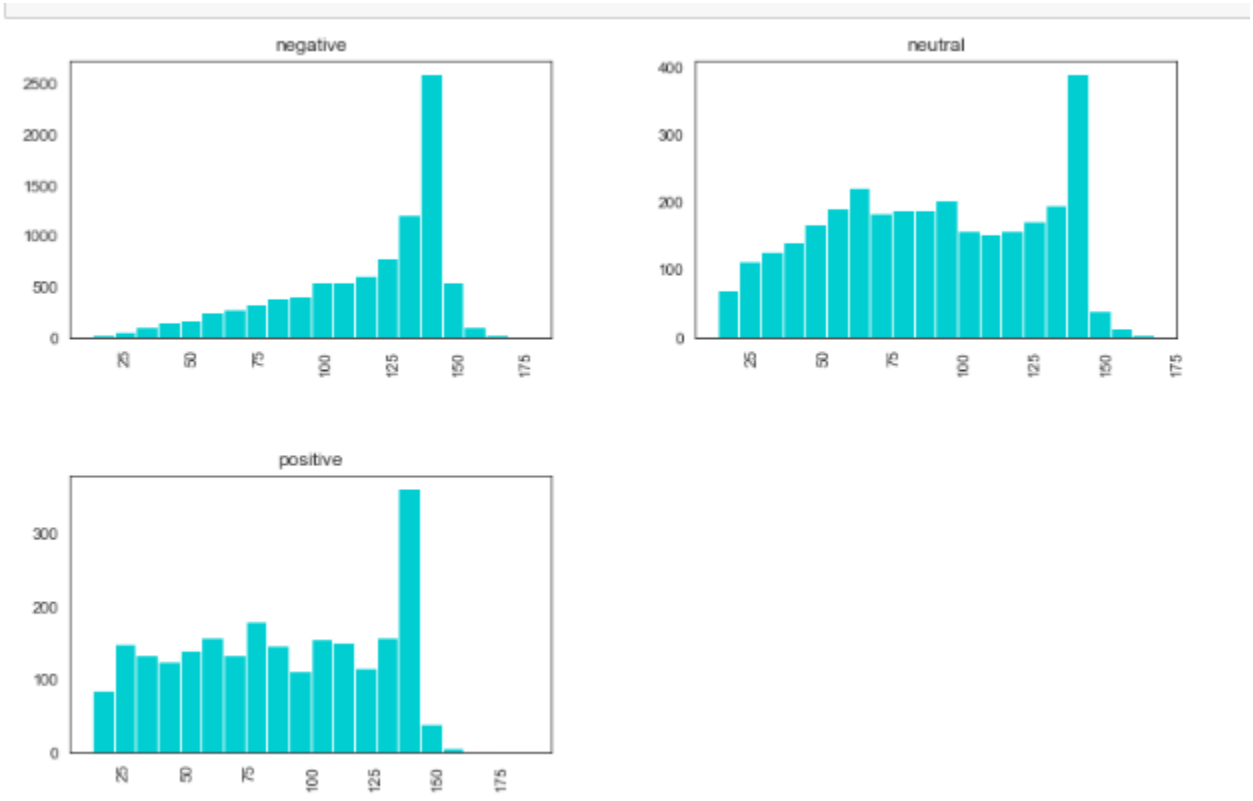
Result of Exploratory Data Analysis



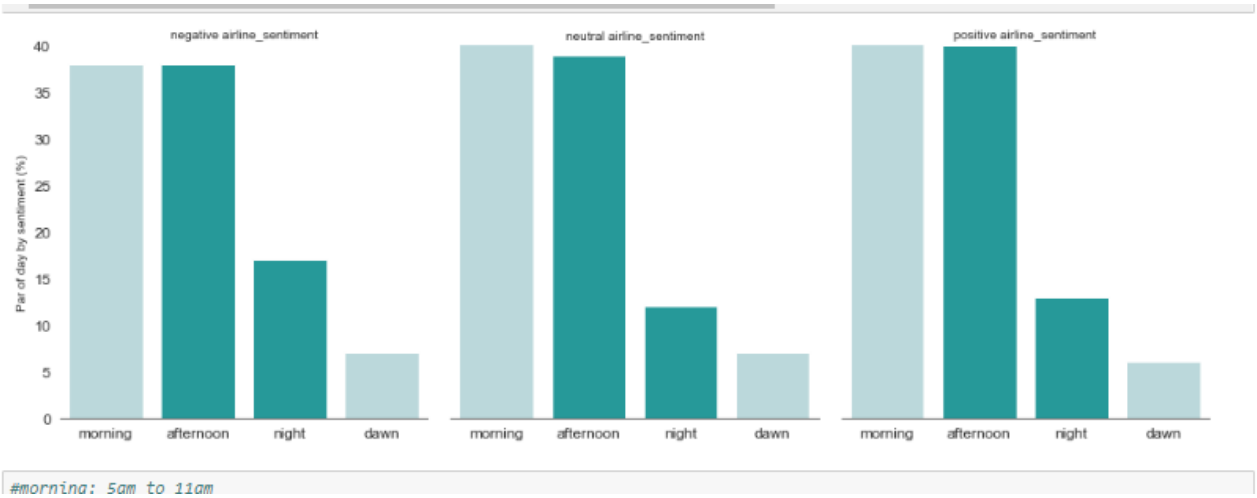
Visualizing Tweet Length



Classification of Tweet on sentiment

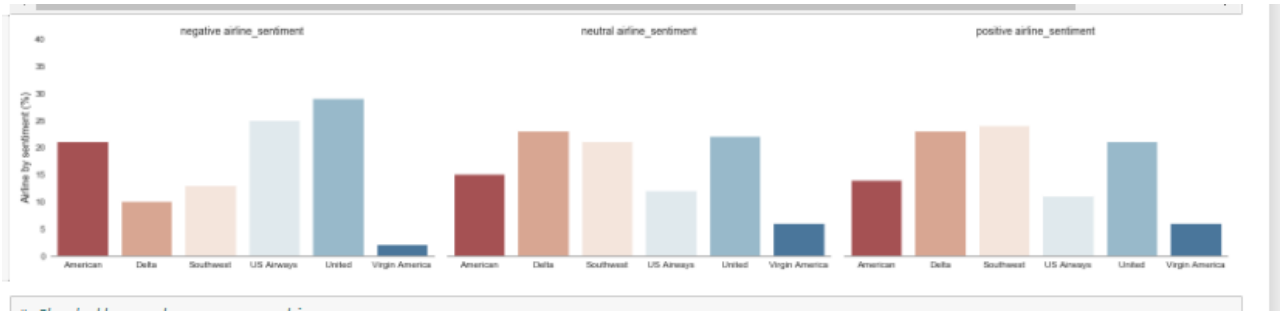


Length wise classification of Tweet



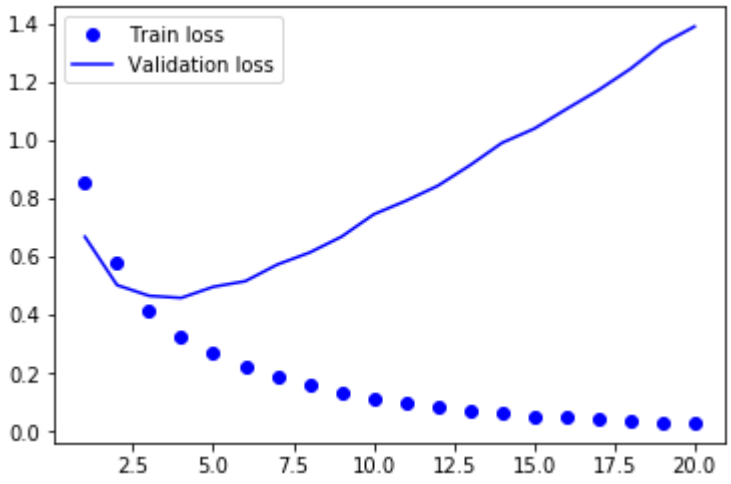

```
#morning: 5am to 11am
```

Length Wise Classification of Tweets

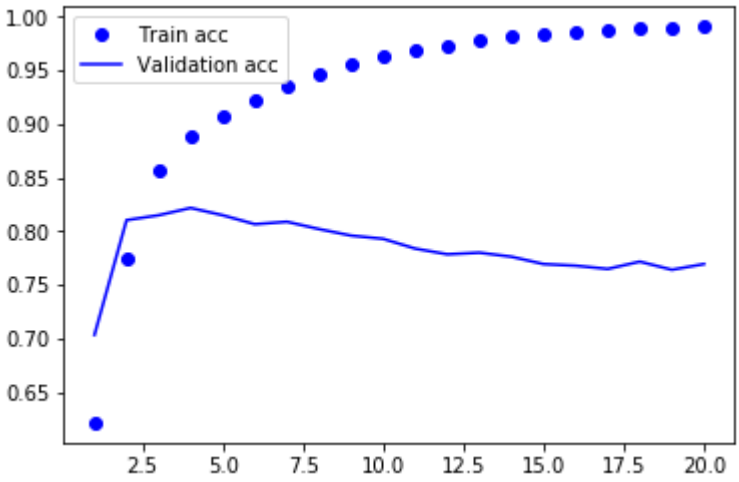


Positive Tweet Visualization

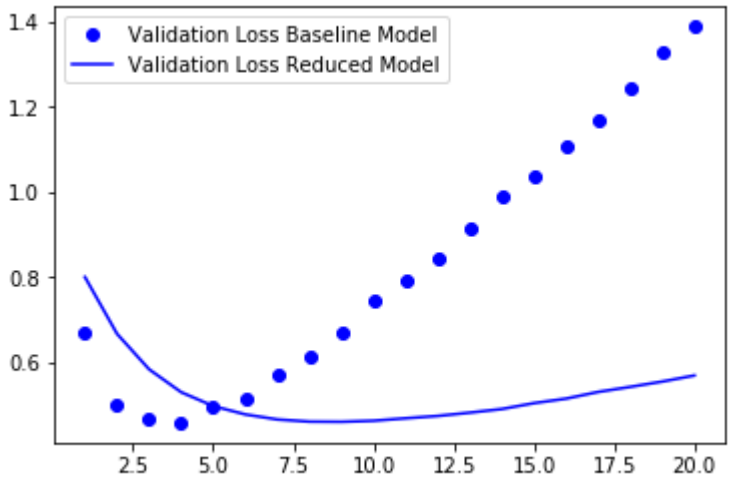
Negative Tweet Visualization



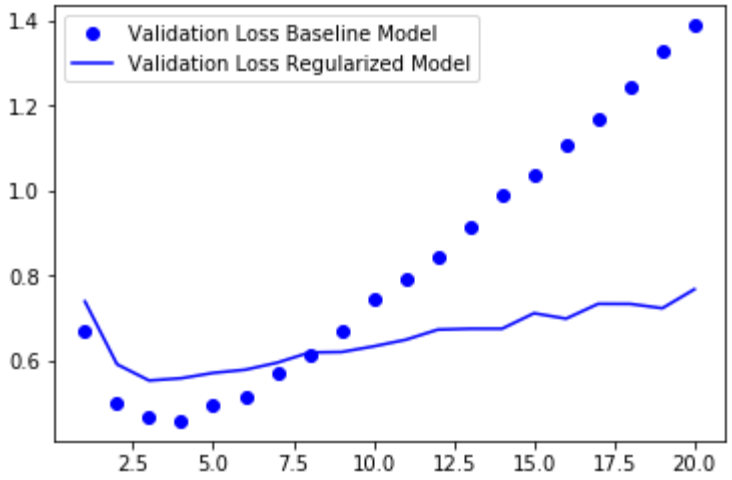
Training loss and Validation loss using Baseline Model



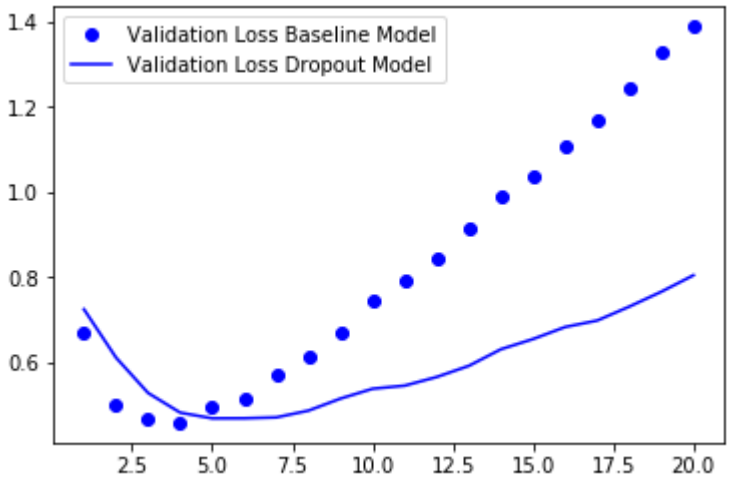
Training accuracy and validation accuracy using baseline Model



Comparison of validation loss using baseline and reduced model



Comparison and Validation loss of the baseline model



Comparison of Validation of baseline model and Dropout Model

78.66% accuracy is achieved by using naïve bayes classifier model.

75.48% accuracy is achieved by using Baseline Deep learning model.

76.91% accuracy is achieved by using reduced Deep learning model.

77.32% accuracy is achieved by using regularized Deep learning model.

76.91% accuracy is achieved by using dropout layer in Deep learning model.

References

[1] “Sentiment Analysis of the Twitter Data”, Apoorv Agarwal, Boyi Xie, Ilia Vovsha, Owen Rambow, Rebecca Passonneau. Department of Computer Science , Columbia University. New York.
[2] “Sentiment analysis algorithms and applications: A survey”, Walla Medhat, Ahmed Hassan, Hoda korashy.
[3] https://codete.com/blog/real-time-sentiment-analysis-with-machine-learning/
[4]http://thinknook.com/twitter-sentiment-analysis-training-corpus-dataset-2012- 09-22/
[5]http://thinknook.com/twitter-sentiment-analysis-training-corpus-dataset-2012-09-22/
[6] “Sentiment analysis in cross media analysis framework”, Yonas Woldmariam, Department of computing science Umea University, Umea Sweden
[7] “A Survey of Sentiment Analysis techniques”, Harpreet Kaur, Veenu Mangat, Nidhi. UIET, Punjab University, Chandigarh, India.