Training Methodology

1. **Problem Statement:**

   Given a text and a "phrase" from it, detect the sentiment expressed towards the "phrase" in the text instance.

   Examples

   - Cannot rely on both milk delivery and grocery | milk | Negative
   - Your customer service is terrible! | customer service | Negative
   - I love notion as a tool | tool | Positive
   - Notion is a great site and an Iphone app. | notion | Positive
   - Asked for a workspace name or billing email address | billing | Neutral

2. **Objective:**

   Aspects are important words that would matter to our customers, we want to be able to provide our customers with insights on how their customers feel about these important words.

3. **Literature Survey:**

   The scope for the research for this topic is quite large. There are several advanced techniques available for this type of problem statement. But the basic difference between the given problem statement and what already available is that the user knows the specific aspect word already. So, there is no need to find out the topic or aspect separately this thing helps to reduce the complexity of the given problem statement. Most of the early works the aspect word is found out by most frequent word or noun or adjectives by Madhoushi (2019). Hu & Liu (2004) first determines all frequent noun phrases from full text reviews as candidate aspects. Then two pruning methods are applied to remove those candidate aspects with meaningless string, based on association rule mining, and those which are subsets of others (redundant). This is a reasonably effective and popular baseline (Liu 2012). Marrese-Taylor et al. (2014) improves the algorithm to estimate the orientation of sentence for compound aspects. In Twitter context, Lek & Poo (2013) takes all nouns, abbreviations, @mentions, or hashtags as candidate aspects. Closest adjective, verb, adverb, or hashtag in the right and left of each aspect considered as sentiment word. Poria et al. (2014) uses implicit aspect corpus. For each implicit aspect, synonyms and antonyms were obtained from WordNet and Semantics extracted from SenticNet. Then aspect parser is built based on several rules. Lizhen et al. (2014) uses dependency parser without extracting aspects and creates 6 tuple feature vector including feature, sentiment words, number of over-modifiers of sentiment word, average score of general modifiers of sentiment word, number of negation words and the punctuation of the sentence. A new feature weighting algorithm is presented that improves TF (Term Frequency) (Luhn 1957) and TF-IDF (Term Frequency-Inverse Document Frequency (Sparck Jones 1972). Also, there are some advanced techniques are available such as bert. Bidirectional Encoder Representations from Transformers is a transformer-based machine learning technique for natural language processing pre-training developed by Google. BERT was created and published in 2018 by Jacob Devlin and his colleagues from Google. These are some popular techniques that can be used to train or build the aspect-based sentiment analyzer.

4. Methodology:

This section gives the detailed approach for solving the problem statement in step-by-step manner. As shown in fig1 first, we have find the insights of the given data, then we have to convert that data to machine understandable language and finally we have to use various approaches to get the best solution.

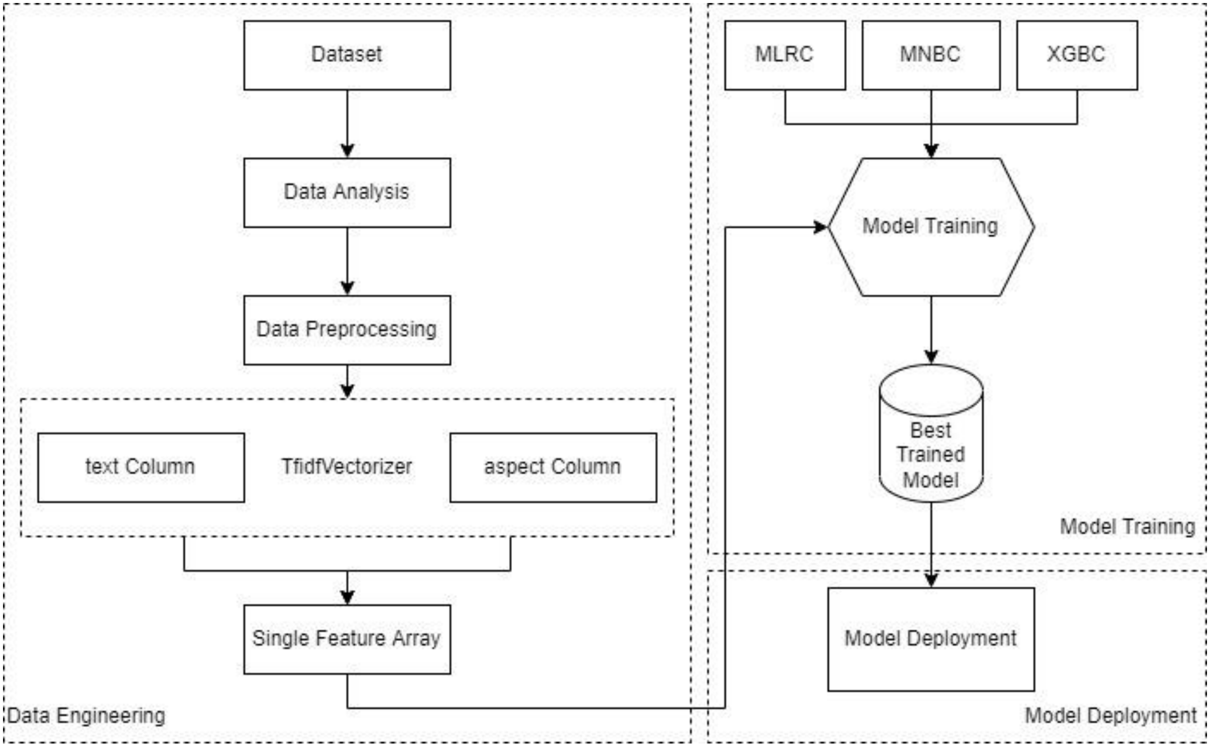

Fig 1: Block Diagram

4.1 Data Analysis:

The most important thing is data analysis. The given data contains no null values with moderate preprocessing on text data. The training data set contains 4000 samples with 3 columns named “text”, “aspect” and “label”.

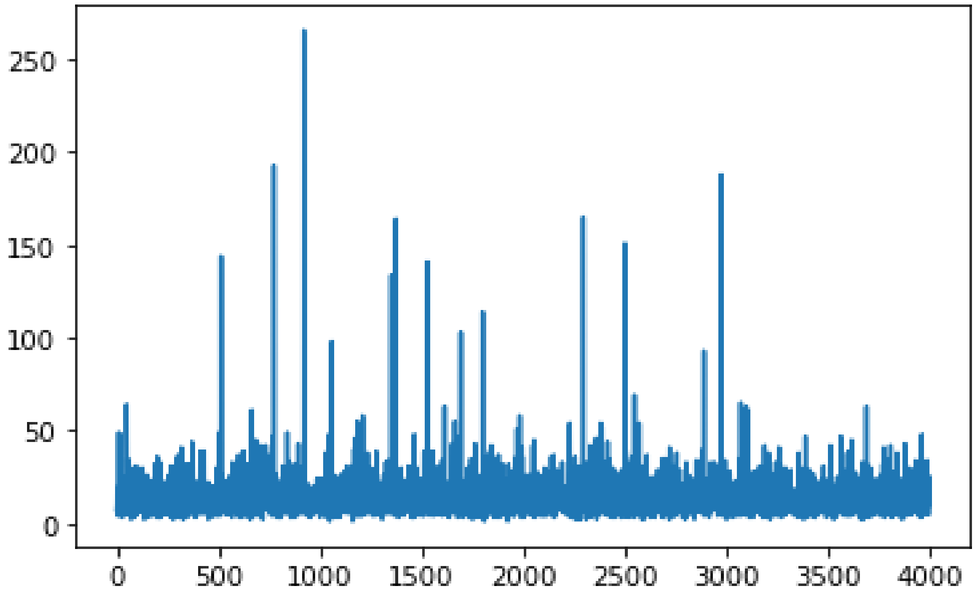

Fig 2: Plot of text lengths.

If we observe the fig 2 carefully, we can conclude that the variation of length of the various sentences in the given text column is less. Also, the maximum length of the sentence in text column is 1472 characters with 266 words. So, there is no need to divide the paragraph into number of sentences and create one different list. Also, more than 95% dataset have word length up to 40-50 words. This signifies the given data is continuous.

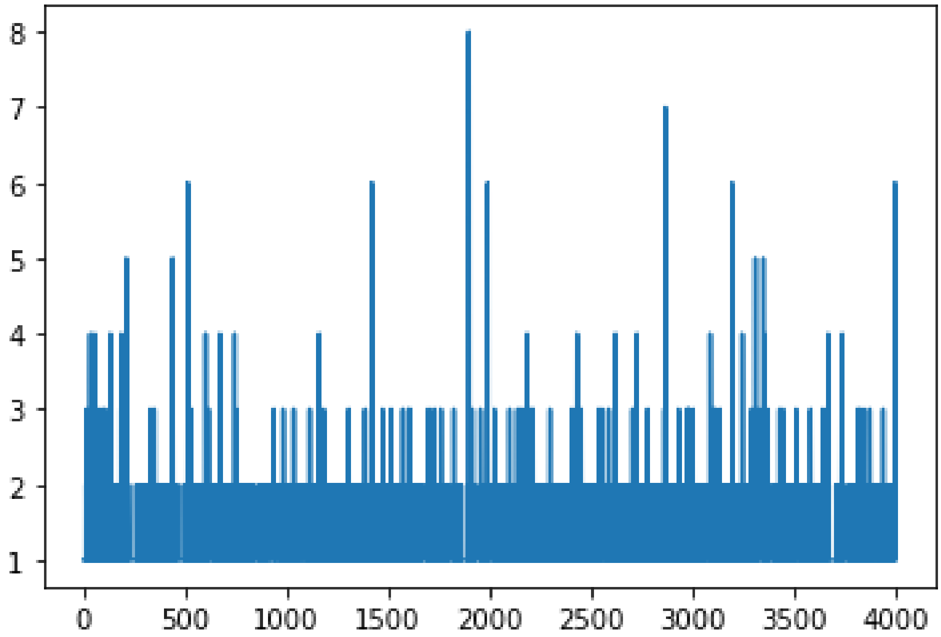

Fig 3: Plot of aspect length.

Similarly, one can observe the different types of lengths of the "aspect" column from in fig 3. From this we can conclude that the "aspect" column is also continuous with maximum length of 42 characters, 8 words and average length of 2 words. This signifies that there is no need to preprocess the given "aspect" column. It looks already effective column.

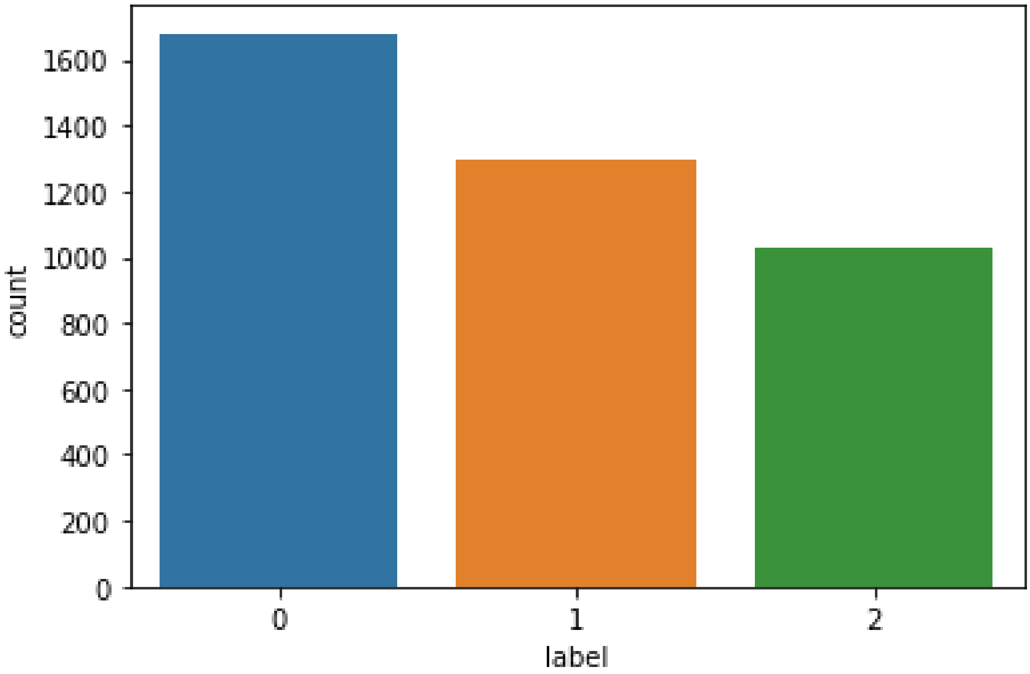

Fig 4: Data Distribution for different labels.

Also, one can observe that the label column is already in numeric form so there is no need to do labelling for that column. If we see the fig 4 one can conclude that the data having little more negative samples than the positive and neutral. But the difference is small so we can ignore the model biasing effect for this dataset. The next step is data preprocessing of the "text" column.

4.2 Data Preprocessing:

I have used two different approaches to preprocess the "text" column. One is usual by loading the stopwords and removing the spaces, special symbols and stopwords. But, I have found the problem in this approach is that the preprocessed text removes the important words such as but, and, than and not. These words are not important when we do normal text sentiment analysis. But, these words create impact for aspect based sentiment analysis. Ex. The sentence is "Satya is good but Sundar is not good because he always keep eye on our data". Now, if we preprocess the sentence by using normal approach it gives us a preprocessed text as "satya good sundar good keep eye data". Now, if we choose the aspect word as "satya" it will give us positive result which is right but if someone chooses the aspect word as "sundar" then also it will give positive because the "not" word is already vanished. This small thing can lead to misclassification. So, I have created one small list of stop words and preprocessed the data by stemming and lemmatization respectively. The next and most important step is to make the given text data to vectorized or numerical form so one can feed this data to machine learning model and get the effective trained model.

4.3 Word Embedding Technique:

There are lots of popular techniques are already available in the data science which can help in word embedding like wordtovec, counterizer, TfIdfVectorizer, bert GloVe etc. The length of the text in given data is smaller so I decided not to go with bert and GloVe. Similarly, counterizer couldn't help us while gathering features from the "aspect" column. So, I have two options go with wordtovec or TfIdfVectorizer. There are already predeveloped approaches available for this type of problem statement.

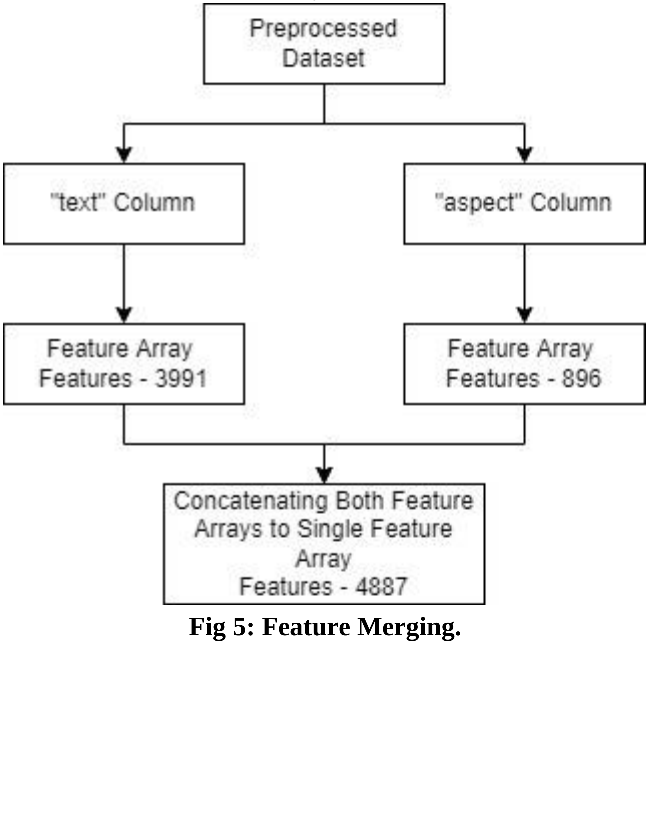
Preprocessed Dataset
"text" Column
"aspect" Column
Feature Array Features - 3991
Feature Array Features - 896
Concatenating Both Feature Arrays to Single Feature Array Features - 4887

Fig 5: Feature Merging.

So, I thought I can do little change and If I get good enough result then I will try to polish more and more that method. I used simple technique that is feature merging. In feature merging I gathered the features from the "text" column and "aspect" column by using TfIdfVectorizer. After, that I merged these two features in one single feature as shown in fig 5. Now we have single feature array with divided features. The main problem is that while taking the input from user we must extract same number of features from the user input and merge them as shown in fig 5. Now this is newly created feature can reduce most of the complexity.

4.4 Model Training:

For model training I have used three popular machine learning algorithms. Such as "Multinomial Logistic Regression (MLRC)", "Multinomial Naïve Bayes Classifier (MNBC)" and "eXtreme Gradient Boosting Classifier (XGBoost)" algorithms. The detailed information is given below sections:

4.4.1 Multinomial Logistic Regression Classifier (MLRC):

It is the advanced form of logistic regression for multiclass classification. For best parameters I have used two methods gridsearch and randomsearch. The best parameters that I got are *{'C': 0.6888888888888889, 'class_weight': 'balanced', 'multi_class': 'multinomial', 'penalty': 'l2', 'solver': 'sag'}*. As shown in fig 6 of confusion matrix one can clearly conclude that the overall accuracy is moderate. We got nearly 70.75% test set accuracy and 87.125% as training set accuracy. It's not very good but It can be useful.

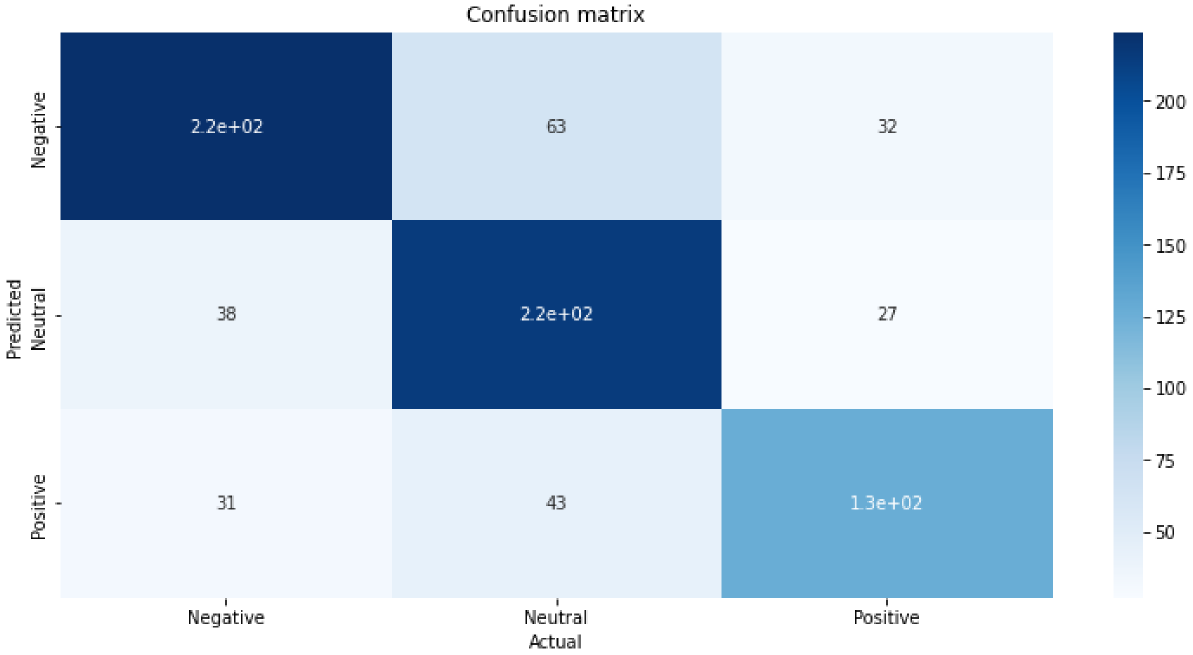

Fig 6: Confusion Matrix of MLRC.

4.4.2 Multinomial Naïve Bayes Classifier (MNBC):

Naïve bayes classifier works on probability. So, our earlier approach of adding features together by taking TfIdfVectorizer might be helpful for this training algorithm. By seeing the fig 7 one can conclude that the overall classification is not that much good but it is moderate. We got test set accuracy as 69.125% and training set accuracy as 83.09%.

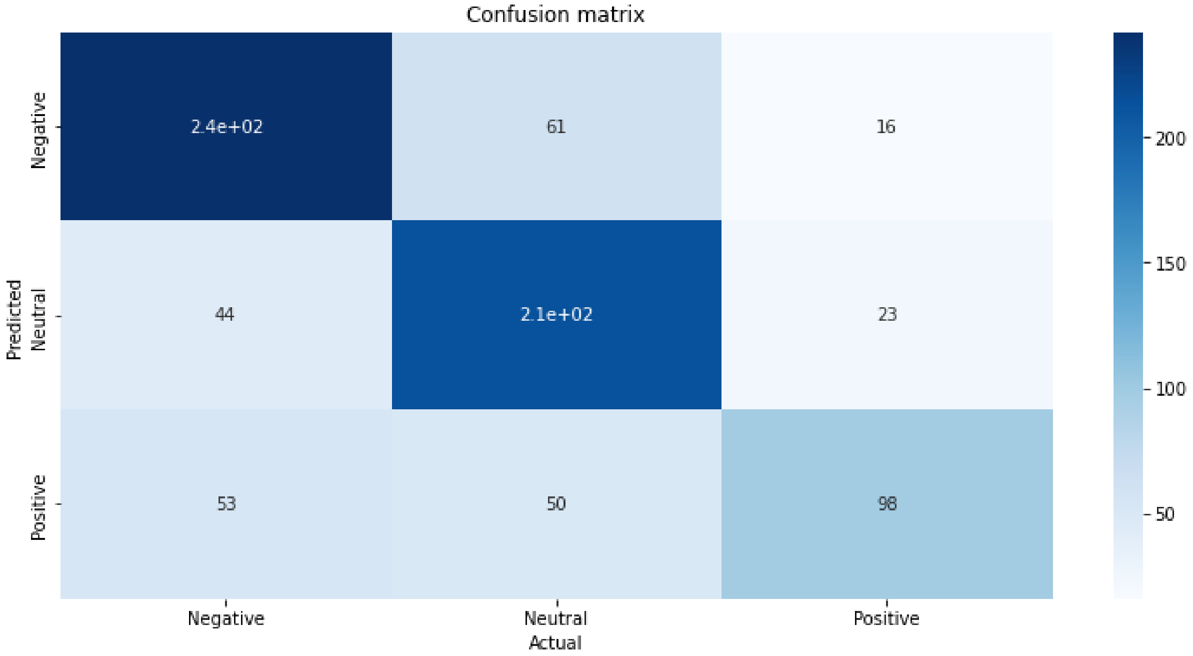

Fig 7: Confusion Matrix of MNBC.

4.4.3 Extreme Gradient Boosting Classifier (XGBC):

It's an advanced version of Decision Tree Algorithm. By observing the fig 8 one can conclude that the overall classification is very poor. We got test set accuracy as 67.25% and training set accuracy as 90.03%. From these results we can say the model is overfitted. So, for deployment we have to choose between MLRC or MNBC.

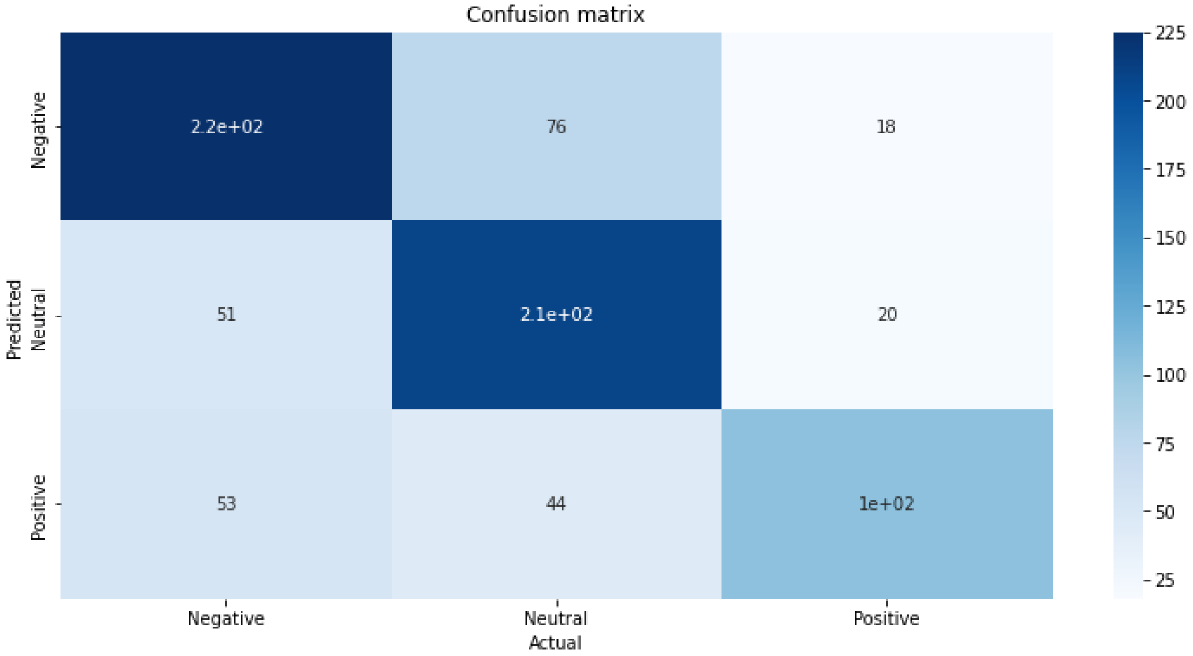

Fig 7: Confusion Matrix of XGBC.

4.4.4 Best Model Selection:

Choosing best model from these three models is quite difficult because the test set accuracies are nearly like each other. By observing the table 1 and comparing the training set accuracy and test set accuracy. I decided to use MLRC for deployment. Because this model is giving the highest test accuracy compared with the other models.

Model/ Algorithm	Training Set Accuracy	Test Set Accuracy
MLRC	0.871250	0.70750
MNBC	0.830937	0.69125
XGBC	0.900312	0.67250

Table 1: Training and Test Set Accuracies of all Models.

5. Conclusion:

There are various other good approaches are available already to solve this type of problem statement. My focus is on how one can get good results with effective use of algorithms and the avoidance of small mistakes that might leads to misclassification of the data. I have deployed this model on cloud platform for live inferencing. The detail of live inferencing is given in deployment document. The model is working good for less complex sentences. There are some limitations are present for this model where it fails due to its less accuracy. The situations like when we have to find the sentiment of aspect word where the aspect word is in between the two contradict statements. One can mix this approach with bert or any other advanced NLP techniques and can get the good results. Also, this problem statement has good future scope.

References:

1) Madhoushi, Zohreh & Razak, Abdul & Zainudin, Suhaila. (2019). Aspect-Based Sentiment Analysis Methods in Recent Years. Asia-Pacific Journal of Information Technology & Multimedia. 08. 10.17576/apjitm-2019-0801-07.
2) Liu, B. 2012. Sentiment Analysis and Opinion Mining. Synthesis Lectures on Human Language Technologies. 5(1): 1-167.
3) Liu, K., Xu, L. & Zhao, J. 2012. Opinion Target Extraction Using Word-Based Translation Model. Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning, pp. 1346-1356.
4) Yaakub, M. R., Li, Y., Algarni, A. & Peng, B. 2012. Integration of Opinion into Customer Analysis Model. Web Intelligence and Intelligent Agent Technology (WI-IAT), 2012 IEEE/WIC/ACM International Conferences on, pp. 164-168.
5) Cambria, E., Schuller, B., Xia, Y. & Havasi, C. 2013. New Avenues in Opinion Mining and Sentiment Analysis,"in IEEE Intelligent Systems, 28(2): 15-21. doi: 10.1109/MIS.2013.30
6) Bagheri, A., Saraee, M. & De Jong, F. 2014. ADM-LDA: An Aspect Detection Model Based on Topic Modelling Using the Structure of Review Sentences. Journal of Information Science. 40(5): 621-636.
7) Chen, Z., Mukherjee, A. & Liu, B. 2014. Aspect Extraction with Automated Prior Knowledge Learning. Proceedings of ACL, pp. 347-358.
8) Luhn, H. P. 1957. A Statistical Approach to Mechanized Encoding and Searching of Literary Information. IBM Journal of Research and Development, 1(4): 309-317.
9) https://www.wonderflow.ai/blog/challenges-in-aspect-based-sentiment-analysis-absa