Table of Contents

1 Introduction

Sentiment classification is typical text classification scenario where sentences are classified based on opinions they hold. The assignment deals with the product reviews of the restaurants where each review is classified into one of the three different categories.

The following case provides an example of text classification carried out throughout the assignment

Two of the cases in the data of Restaurant reviews are as follows,

Case1:

"A very good Greek restaurant with tasty food. I tried the chicken kabob, which was tender and juicy. The rice was very flavourful. Also tried the gyros, and they were the most amazing I've ever had! The meat was well-flavoured, went great with the cream on top. Together with tomatoes and onions, everything perfectly matched with the pita bread. My only complaint that they are a bit slow in service, but maybe it was because there was only one person with a full room of customers the night I went. But she was friendly and kind :)"

Case2:

'Website says open, Google says open, Yelp says open on Sundays. Our delivery was cancelled suddenly and no one is answering the phone. Shame'

Understanding the opinions of the reviewer, involves several tasks like splitting the sentences into words, understanding the polarity of words and classifying them into 5 different categories namely Strongly positive, positive, neutral, negative and strongly negative which are discussed in detail in the below sections. The Case1 classifies as strongly positive because of the words like very good, tender and juicy, perfectly matched, well-flavoured, friendly and kind whereas Case is classified as negative because of the words like cancelled and Shame.

Therefore, the main goal of the assignment to classify the given Restaurant reviews into five different categories**.**

2 Exploratory Analysis

2.1 Distribution of sentiment classification(labels)

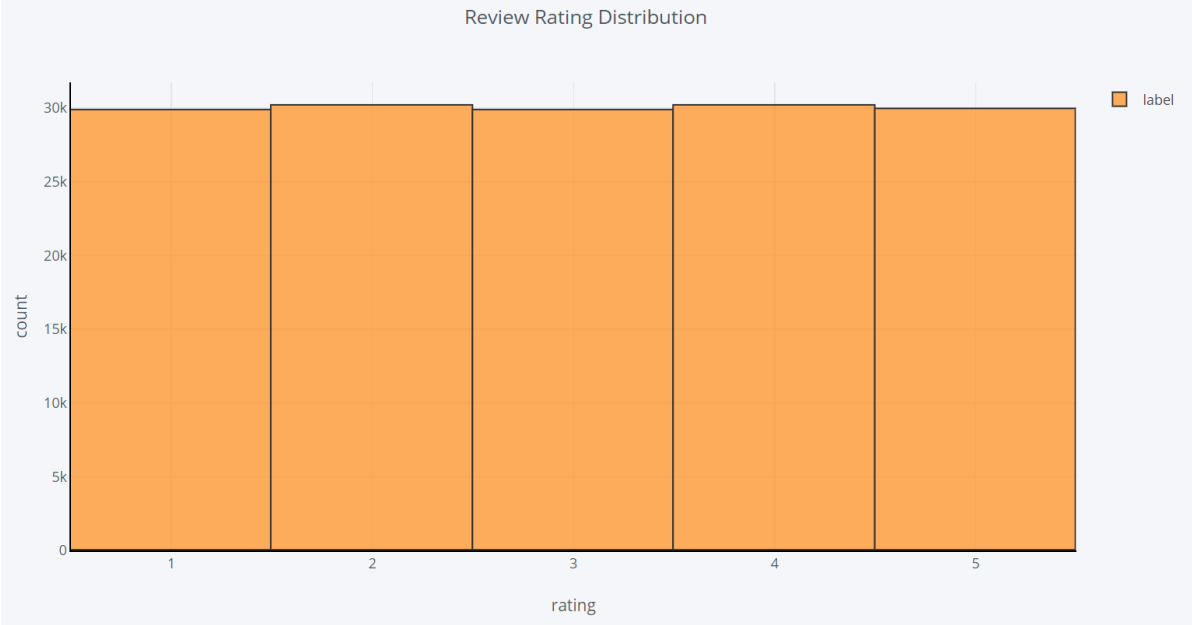

Figure-1

The above histogram shows the rating distribution in train data. We can infer from the plot that train data has almost equal distribution of ratings.

2.2 Distribution of total tokens in each review.

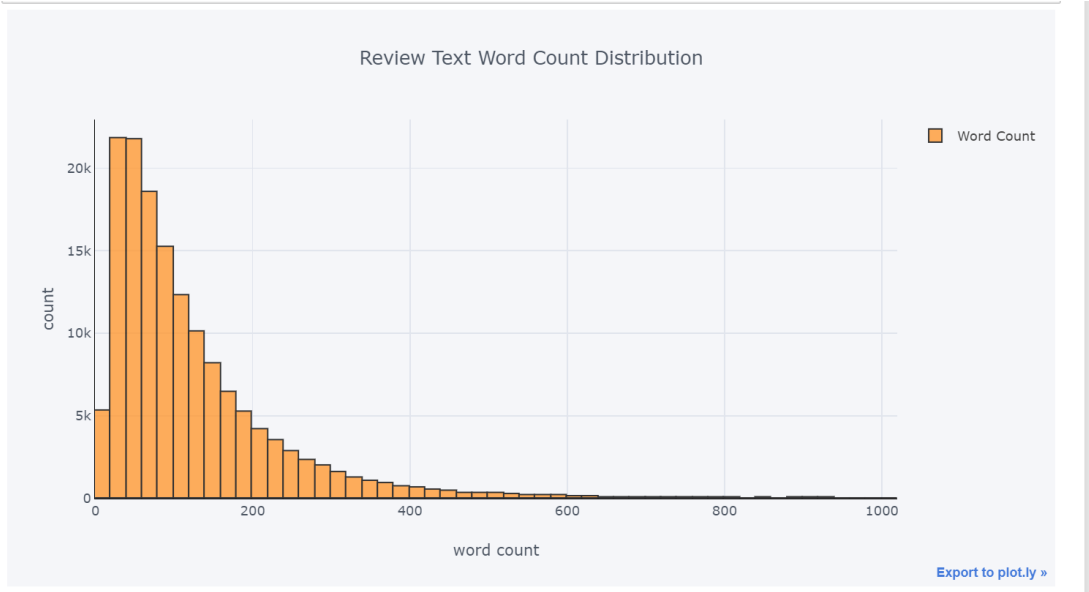

Figure-2

The above shows distribution of word counts in each document and we can its right skewed with average of 25k documents have 30-50 words.

2.3 Words Distribution through Word Cloud

Figure-3

From the Positive word cloud, we can see that most frequent words like good food, good place etc in the corpus.

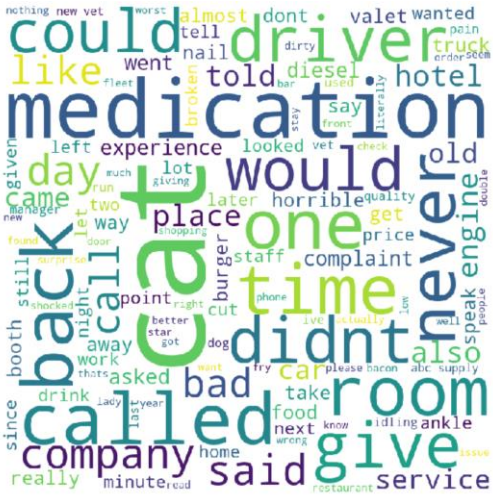

Figure-4

From the above negative cloud, we can infer there are words like bad service like horrible expenses etc in the corpus.

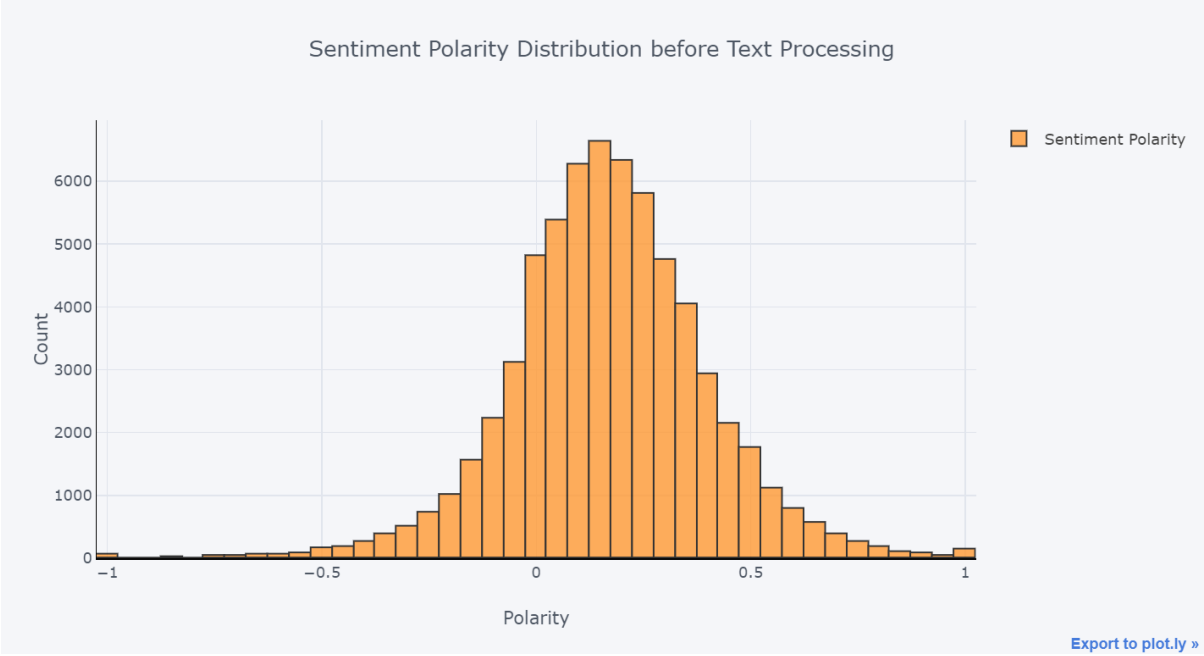

Figure-5

The above histogram shows the polarity distribution of sentiments for each document before the pre-processing of the texts.

3 Pre-processing and feature generation

The Classification of the above described review is not a magic, it happens with lot of text processing in the backend. Transforming/cleaning the text into analysable format plays as important role in getting the classification right. There are different methods used to clean up. Methods we are using is as follows

- Case Normalization – It is process of converting all the uppercase characters into lowercase. This is done reduce the number of words in the final vocabulary list which would be helpful in the later stages of analysis. For instance, word “food and Food” mean the same in the given context hence normalising these words will not will not result in loss of information.

- Tokenization is breaking up of sentence or paragraph into meaningful pieces such as words, keywords or phrases etc.

  Tokenization is fundamental in all text processing algorithms. It is comparatively easy to do analysis when we split large chunks of data into smaller units. For instance, analysis of frequencies of each word, sentiment analysis or stylometric analysis etc would be easier if there are tokenised.

- Concordance and Apostrophes – A concordance view shows us every occurrence of a given word, together with some context. For instance, if we are looking for word “don’t” it looks up the occurrences of word with different contexts. In the given context “don’t” shows up as negative word and it means the same all over. Hence, we are converting the word “don’t’ to “do not” .

  Similar kind of words can be found in the below link and those are words which has apostrophes and hence we are converting all the words like that to extended form.
  https://drive.google.com/file/d/0B1yuv8YaUVlZZ1RzMFJmc1ZsQmM/view

- Lemmatization – It is the process of grouping different forms the word together so that it can analysed in a simple way. For example, “Prices” in the text is converted to “Price”. We are doing this to get the root form of the word so that even if the words are in different form it makes into one and reduces the redundant words in vocabulary.

  The only reason for choosing lemmatization over stemming, we need the to keep the meaning of the words same to determine the sentiment and also to classify.

3.1 Pre-processing

1. Case Normalization: The below screenshot shows text and lowered text (clean text)

	trn_id	text	label	clean_text
74427	trn_74428	Food is pretty good especially the Lechon kawa...	4	food is pretty good especially the lechon kawa...
396633	trn_396634	We have had Bulwark for six years and they hav...	5	we have had bulwark for six years and they hav...
45099	trn_45100	Ordered the pork ramen, but instead got a bowl...	1	ordered the pork ramen, but instead got a bowl...
172087	trn_172088	Excellent value. The food is not awesome, but ...	4	excellent value. the food is not awesome, but ...
155164	trn_155165	Bought a Tennessee style beef brisket.. well w...	1	bought a tennessee style beef brisket.. well w...

Figure-6

2. Tokenization:

As Part of Tokenization, we are splitting the data using space and doing string manipulation (lowering case, removing apostrophes etc) so that we get each word, punctuations and also emoticons.

We are keeping punctuations and emotions because they add significant value to the review which can also determine sentiments scores

3. Removal of Apostrophes and by checking their significance through Concordance.

I wanted to like this place, I really did. Fir...	1	i wanted to like this place, i really did. fir...
I used to come here all the time- every haircu...	3	i used to come here all the time- every haircu...
What a cute brunch spot!! It looked like a pr...	3	what a cute brunch spot!! it looked like a pr...
Didn't get to spend as much time in there as I...	3	did not get to spend as much time in there as ...

Figure 7

The above output shows the word “Didn’t” is converted to “did not”

4. Lemmatization:

	trn_id	text	label	clean_text
372655	trn_372656	If I could give progressive negative stars I w...	1	if i could give progressive negative star i wo...
635429	trn_635430	This was supposed to be a magical evening in a...	2	this wa supposed to be a magical evening in a ...
394972	trn_394973	Kathy at front desk is a sweetheart however I ...	1	kathy at front desk is a sweetheart however i ...
444883	trn_444884	Usually rave about this place to my friends be...	2	usually rave about this place to my friend bec...
505824	trn_505825	Was taken here by family who wanted to try the...	2	wa taken here by family who wanted to try the ...

Figure 8

After passing the input through lemmatization the output is generated is as shown in the above figure where “friends” is getting converted to “friend”. Therefore, lemmatizations is not changing the meaning of the sentence.

5. Identifying Most Frequent and less frequent words

The below graph contains the 25 most frequent words. We can see that it is mostly dominated by the little words of the English language which have important grammatical roles. Those words are articles, prepositions, pronouns, auxiliary webs, conjunctions, etc. They are usually referred to as function words in linguistics, which tell us nothing about the meaning of the text. Hence removing them would be helpful in the analysis.

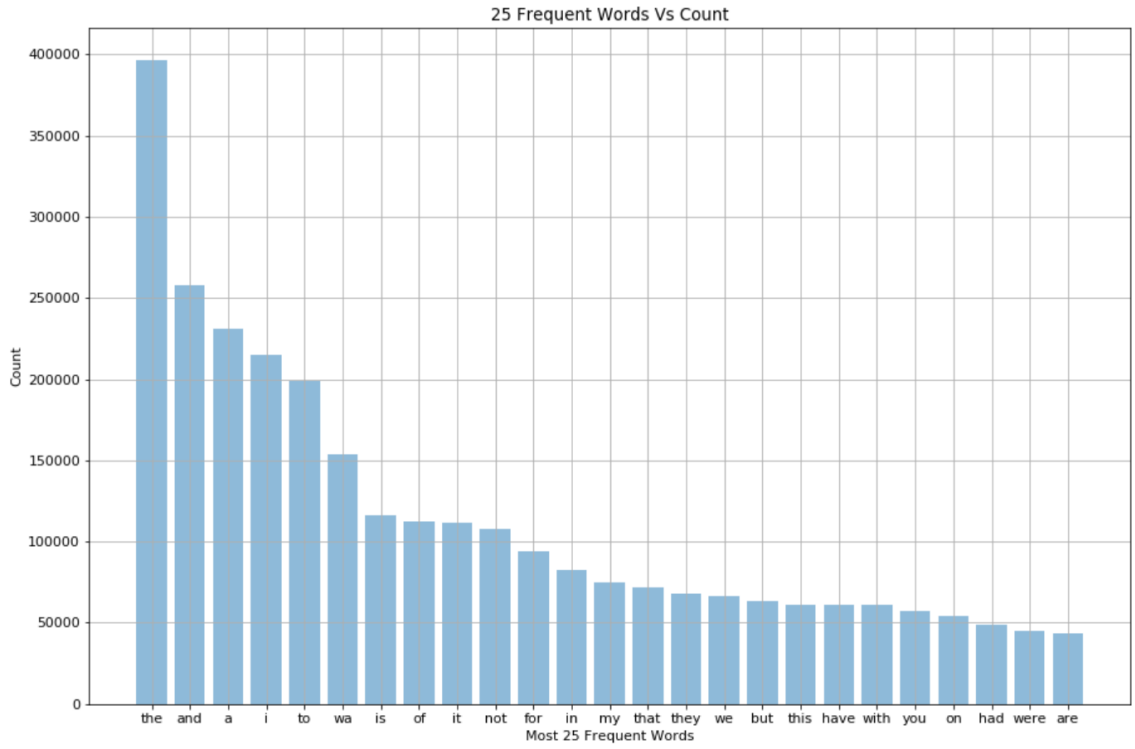

Figure-9

Here another interesting statistic to look at is the frequency of the frequencies of word types in a given corpus. We would like to see how many words appear only once, how many words appear twice, how many words appear three times, and so on.

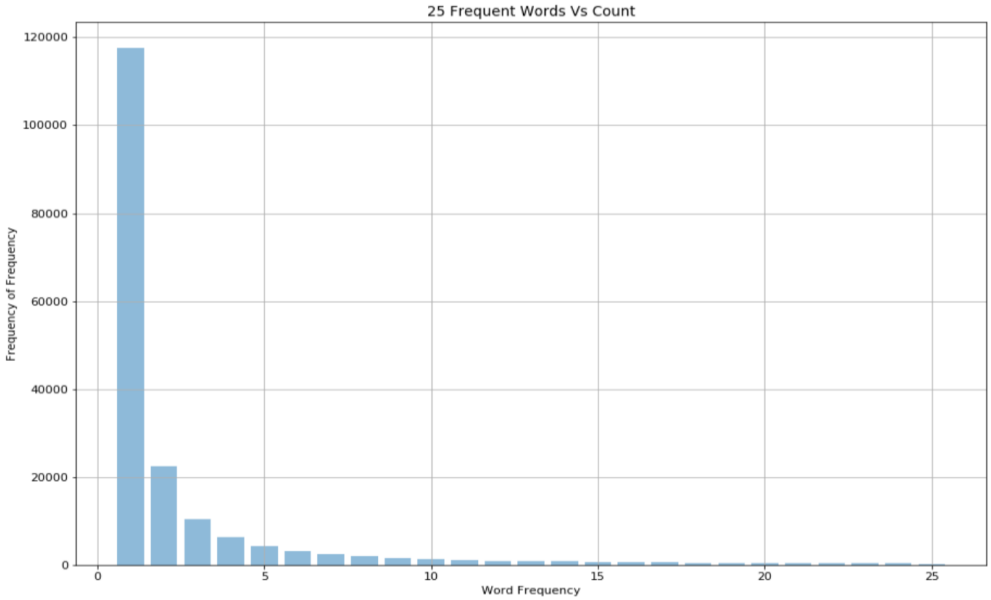

Figure-10

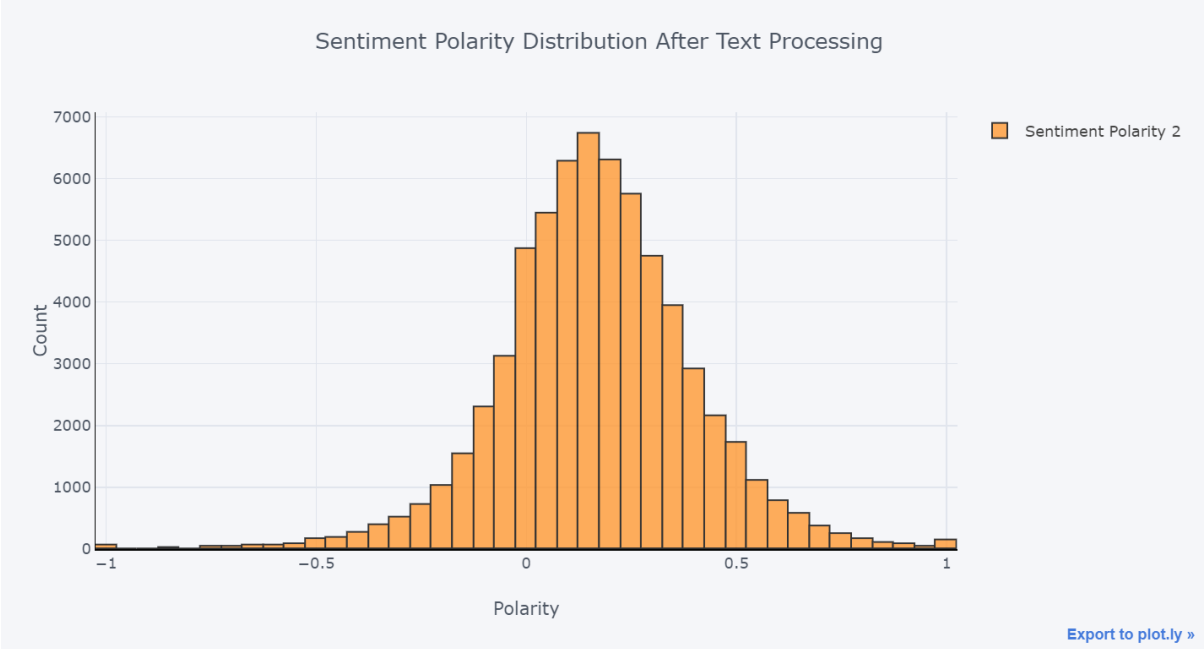

Figure-11

The above histogram shows the polarity distribution of sentiments for each document after the pre-processing of the texts. This shows that the distribution has remained constant throughout the pre-processing stages.

3.2 Feature generation

It is very necessary to extract the features/ predictors to build the model, if not extracted it ends up in the noise. It becomes more important if the features are large. It is helpful to the model if only required number if features are sent across or else the time required to train the model increases and thus cost.

The main reasons to this task are:

- If the appropriate set is chosen it improves the accuracy of the model.
- Variance decreases and hence reduces overfitting.
- It can train the model faster reducing the cost.

There are different methods of features selection like

✓ Filter methods which involves chi-square, Anova, LDA.

It is the statistical test which determines the correlation between the features and associations using their frequency distribution.
We are using Chi-Square as one of the feature selection methods to select correct attributes.

✓ Wrapper method which involves forward and backward selections
These methods are iterative and hence applying these methods where there are more features are computationally expensive. Hence, we have not chosen any wrapper method.

- ✓ Embedded methods which involves Lasso and Ridge regression which basically adds penalty to magnitude of the coefficients. We are using Ridge regression as one our feature selection approach.

  L1 Regularizations adds penalty to absolute value of magnitude of the coefficients
  L2 Regularizations adds penalty to square of the magnitude of the coefficients.

  We are adding the ridge regression which performs L2 regularization TF-IDF vectorizer functions as below.

```
# Convert the text into Tfidf vector form with max features as 7000 and appending bigrams and trigrams.
vectorizer = TfidfVectorizer(max_features=2000,sublinear_tf=True,norm='l2',ngram_range=(1,3),analyzer="word")
X = vectorizer.fit_transform(train["clean_text"])
```

```
# Convert the text into Tfidf vector form with max features as 7000 and appending bigrams and trigrams.
vectorizer = TfidfVectorizer(max_df=0.95,min_df=0.95,sublinear_tf=True,norm='l2',ngram_range=(1,3),analyzer="word")
X = vectorizer.fit_transform(train["clean_text"])
```

  TF-IDF scales down the impact of the words which appear more frequent and also less frequent which are basically less informative.

  Max_df: When building the vocabulary, it ignores terms that have a document frequency strictly higher than the given threshold.

  Min_df: ignores terms that have a document frequency (presence in % of documents) strictly lower than the given threshold.

  Sublinear_tf: Applies scaling term frequencies makes in 1+log(tf)

  n-grams: Add ngrams to vocabulary

  analyser: If the input has tokens then analyser is activated as word

  Max_features : Selects the top n words as vocabulary

4. Models

4.1 Linear Support Vector Machines

Support Vector Machine are a set of Machine learning algorithms which are used for the classification, regression and detection of the outliers.

The main advantages of using SVM are:

1. **High Dimensional Data**:

It is very effective in high dimensional data even where the number of samples are less than the dimension that is very much like our dataset as we have got a large set of predictors.

2. **Memory Efficiency**:

It is also very memory efficient in the decision function it uses a subset of training points also called Support Vector which makes the Model more efficient and very helpful for our dataset as we must feed a lot of data.

3. **Versatile**:

SVM is very flexible as it provides the different kernel functions infect the Common Kernels are provided but we can easily customise the Kernel according to our requirement.

4. **Accuracy**: SVM works well when the data has a clear margin of separation.

The main disadvantage of using SVM are:

1. **Overfitting**: We must deal with Overfitting when we have the number of samples are much less than the number of the number of features. It can be avoided by choosing different Kernel Function.

2. **Expensive**: The probability estimates are not directly calculated by SVM it uses an expensive method of Five-Fold Cross-Validation.

The different type of Kernel that can be used in SVM are :

1. Linear
2. Polynomial
3. RBF
4. Sigmoid

We are using SVC (**liblinear with Linear Kernel)** instead of libsvm in our Model because it works well with the large dataset as when we have high number of features in our dataset it is not necessary to map all the data to higher dimensional space due to which the non-linear mapping won't help in improving the performance of the model.

The SVC (**liblinear** with Linear Kernel) has more flexibility in the choice of penalties and the loss functions.

We are using **'balanced'** mode which utilizes the estimations of y to naturally modify loads contrarily corresponding to class frequencies in the information.

The Linear Kernel works very efficiently, and it searches for only the required.

4.2 Multinomial Naïve Bayes

Naive Bayes Model works on the principle of the Bayes Theorem that each feature which is used for the classification is independent of one another given some class.

The Naive Bayes model works on the Gaussian Probability Distribution Function which is general and can also be called as Gaussian Naive Bayes Model.

Multinomial Naive Bayes Classifier is special instance of Naive Bayes Classifier in which it uses the multinomial distribution for each feature.

We are using the Multinomial Naive Bayes Classifier as it is has more edge over the classification which has discrete features that is working with the Word Counts, Review of a customer, Text Documents and it also works with the tf-idf.

Advantages of the Multinomial Bayes:

1. Computationally fast

The efficiency and processing of the multinomial Naïve Bayes is good on a large dataset.

2. Works well with high dimensions

As in our dataset we have got a high number of features so it's better to implement the Multinomial Naïve Bayes Algorithm to work on high number of predictors.

3. Easy to implement:

Naïve Bayes is very easy to implement.

4. Smoothing Priors:

It gives you more flexibility to play with smoothing priors we use it when our test data experience a new feature which it has not experienced in training data then we use Smoothing priors.

- $\alpha=1$ is called Laplace smoothing
- $\alpha<1$ is called Lidstone smoothing.

Disadvantage of the Multinomial Bayes:

If it experiences a categorical variable in the test data which it has not experienced in the training data, then it considers it as a 'Zero Frequency'.

One of the biggest limitations of the Naïve Bayes is that it considers all of the predictors as independent where as in real life there are high chances that the predictors are highly correlated.

4.3 Logistic Regression

Logistic Regression is the most widespread algorithm for solving industry scale problems, although its flinching to other modern techniques with higher potential in efficiency and implementation ease of other complex algorithms.

Logistic regression is noise tolerant and is not affected by few cases multi-collinearity. Extreme cases of multi-collinearity can be handles by applying L2 regularization. Although it is not best choice as it has all the features in the model thus impacting time and cost.

Advantages of Logistic Regression:

- ✓ Handles Multi-collinearity
- ✓ Implementations available across all the platforms.
- ✓ Probability scores are convenient for observations.

Disadvantages:

- ✓ Takes long time to run if the features are more
- ✓ Doesn't handle large number of categorical features

4.4 Discussion of model difference(s)

We have explored SVC, Multinomial Nave Bayes and Multinomial Logistic Regression above,

We realise that Multinomial Nave Bayes treats all the features to be independent of each other. Uses probabilistic approach to categorise records.

SVC considers the interaction between features which helps in understanding the relation between the features and effect on the model. Uses geometric approach to categorise the records.

However, Logistic regression estimates the probability directly from the training dataset minimizing error which makes it a Discriminative model. Logistic regression works really well when the features are co-related which adds as an advantage in the task when we have singular and plural words. Since we have a large dataset, we take advantage of logistic regression's simple computation and implementation. We also have an advantage on utilizing lasso and ridge regression in our modelling to penalise inaccuracy.

We arrived at a better accuracy for our dataset using multinomial logistic regression. Thus, we concluded to improvise the model and train the logistic model to arrive at a better model.

5 Experiment setups

Experiment 1

Pre-Processing

- ✓ Convert to Lowercase
- ✓ Remove Stop words
- ✓ Remove Punctuations
- ✓ Remove the words which are having the length less than 3
- ✓ Lemmatization
- ✓ Removing the words which are having the document frequency>95% and less than 2%
- ✓ Using Chi-Square test to get top 500 words which act as features for the model.

Modelling

1. Linear SVC
   Setting the penalty(c) to be 0.5
   class_weight to be “balanced”

2. Multi Nominal Naïve Bayes
   With smoothing parameter =0.5 (lid stone smoothing)

3. Multi Nominal Logistic regression with parameters

   solver='newton-cg as this handle multinomial loss and used for multinomial classes
   Penalty (C) as 0.4
   class_weight="balanced",
   max_iter=500, number of times it iterates before the solver converges
   penalty='l2',
   tol=0.001 tolerance level

Experiment 2

Pre-Processing

- ✓ Convert to Lowercase.
- ✓ Removing the words which are having the document frequency>99% and less than 1% using TF IDF vectorizer using min and max
- ✓ Including Bigrams and also Trigrams in the model
- ✓ Setting up the parameters of the model as in section 2.2

Modelling

1. Linear SVC

Setting the penalty(c) to be 0.3
class_weight to be “balanced”

2. Multi Nominal Naïve Bayes
   With smoothing parameter =0.5 (lid stone smoothing)

3. Multi Nominal Logistic regression with parameters

   solver='newton-cg as this handle multinomial loss and used for multinomial classes
   Penalty (C) as 0.3
   class_weight="balanced",
   max_iter=1000, number of times it iterates before the solver converges
   penalty='l2',
   tol=0.001 tolerance level

Experiment 3:

Pre-Processing

- ✓ Convert to Lowercase.
- ✓ Removing the words which are having the document frequency>99% and less than 1% using TF IDF vectorizer using max features
- ✓ Including Bigrams and also Trigrams in the model
- ✓ Setting up the parameters of the model as in section 2.2

Modelling

1. Linear SVC
   Setting the penalty(c) to be 0.5
   class_weight to be “balanced”

2. Multi Nominal Naïve Bayes
   With smoothing parameter =0.5 (lid stone smoothing)

3. Multi Nominal Logistic regression with parameters

   solver='newton-cg as this handle multinomial loss and used for multinomial classes
   Penalty (C) as 0.6
   class_weight="balanced",
   max_iter=250, number of times it iterates before the solver converges
   penalty='l2',
   tol=0.0001 tolerance level

6 Experimental results

Feature Set	Model	Accuracy
Feature Set 1	Model 1	54.622%
	Model 2	53.82%
	Model 3	55.66921%
Feature Set 2	Model 1	57.8732%
	Model 2	58.16%
	Model 3	58.873%
Feature Set 3	Model 1	60.19%
	Model 2	59.8741%
	Model 3	62.506%

Table -1

7. Conclusion

Sentiment analysing is one of the trending topics in Machine Learning. This is still a research area and it is difficult to detect the sentiments of the reviews accurately due to complex nature of the language. For instance, sarcasm which very unlikely to classify correctly.

In this assignment the product reviews provided to classify had 6,50,000 restaurant reviews where each review had to be classified as one of the five categories. Initial Text processing has been conducted to remove unwanted words and generate features. Three experiments have been conducted in each case text processing and model development parameter has been tweaked to achieve the best accuracy.

It's been found that Multinomial Logistic Regression provides highest accuracy when compared to all the other models with 62.6% with basic feature extraction involving bigrams and trigrams

References

1. https://www.analyticsvidhya.com/blog/2018/02/the-different-methods-deal-text-data predictive-python
2. https://towardsdatascience.com/a-complete-exploratory-data-analysis-and-visualization-for-text-data-29fb1b96fb6a
3. https://www.datacamp.com/community/tutorials/wordcloud-python
4. https://www.analyticsvidhya.com/blog/2018/02/the-different-methods-deal-text-data-predictive-python/
5. https://textblob.readthedocs.io/en/dev/quickstart.html
6. https://scikit-learn.org/stable/modules/generated/sklearn.naive_bayes.MultinomialNB.html
7. https://www.nltk.org/
8. https://scikit-learn.org/stable/modules/generated/sklearn.svm.SVC.html
9. https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LogisticRegression.html