DataSet3:Amazon Reviews for Sentiment Analysis

Dataset Structure:

The Dataset given in the assignment consisting of 0.4 Million Customer Reviews on different Products.

There is two Columns of the dataset, one is label and other is text.

Each Review in the dataset is classified Either in Positive or Negative Class/label.

According to Analysis there is equal number of Positive and negative Review Labels i.e, 0.2 each.

Problem:

We have to train a Model from the given dataset, so that it can classify the coming testing data reviews into Negative or positive Labels with high accuracy and bigger ROC Curve Area.

Dataset Selection:

Due to limited resources and limited computing capability of our notebook, we have decided to choose 10000 dataset among 0.4 million .

The criteria we have chosen to select our dataset is that we want to pick equal number of positive and negative Reviews dataset.

Among 10000,80% of the dataset is our training data and 20 percent we have chosen as test data

- **Stratified Random Sampling**

For this purpose, we have done stratified Random sampling and taken training dataset with equal ratio of positive and negative review labels so that the model doesn’t get biased for a particular class label during training.

-We have chosen Random 5k positive and 5k negative reviews , and combined them to create a dataset of 10k.

Among this 10k we have chosen 1k Positive dataset and 1k negative dataset and combined them to make our test data of 2K, this type of separation is also known as **Hold Out Cross Validation.**

TrainingData - > Random 4000 Positive Reviews + Random 4000 Negative Reviews.

TestData -> Random 1000 Positive Reviews + Random 1000 Negative Reviews.

Process flow We Followed:

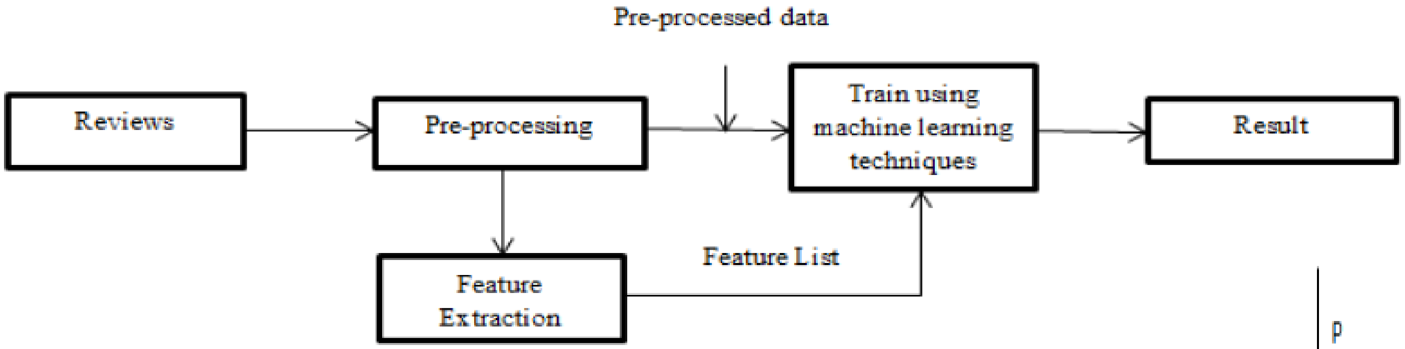
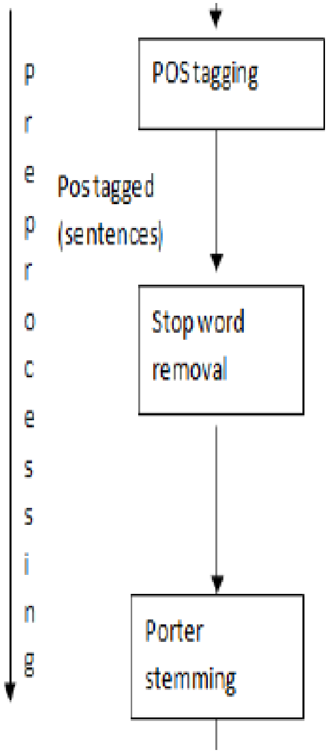


Fig.1 A general process flow using machine learning techniques



Preprocessing:

The dataset is unstructured; it may contain repetitive words, large number of words that are not at all needed in summarizing of opinions.

a)Capitalization:

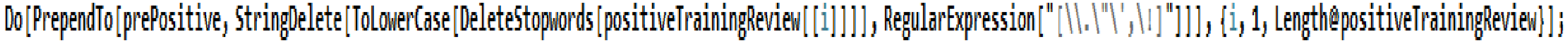
We have capitalize all the words in a review to lowercase .So,that the same words in lower case and upper case in same review should not be distinguished.

b)Stop Word Removal:

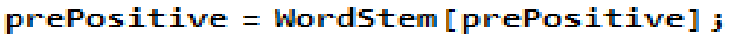
Pre-processing involves removal of stop words such as ‘and’, ‘or’, ‘that’ etc .

c)Porter Stemming:

Porter stemming which involves simplifying target words to base words by removal of suffixes such as – ed, ate, ion, ional, ment, ator, ssess, es, ance or conversion from ator to ate etc. For example, “replacement” is stemmed to replac; “troubled” to trouble ; “happy” to happi ; “operator” to operate. The raw data is pre-processed to improve quality.



```
Do[PrependTo[prePositive, StringDelete[ToLowerCase[DeleteStopwords[positiveTrainingReview[[i]]]], RegularExpression["[\\.\"\',\!]"]]], {i, 1, Length@positiveTrainingReview}];
```

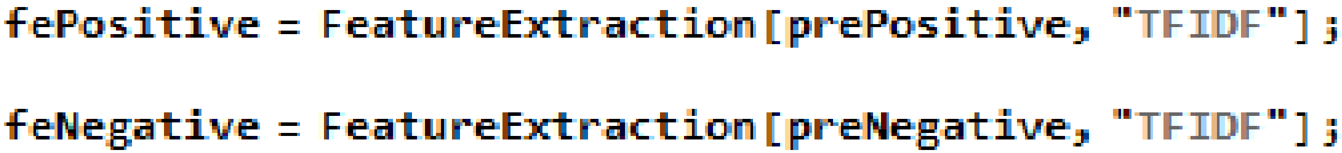


```
prePositive = WordStem[prePositive];
```

Feature Extraction:

Features in reviews are extracted so that it helps customer to know which feature has positive comment and which one has negative. Since, overall conclusion about product is much needed but there is also situation where customer requirements come into the scenario.

We have used “TFIDF” Method->Term Frequency Inverse Document Frequency of *Feature Extract Function Of Mathematica.*



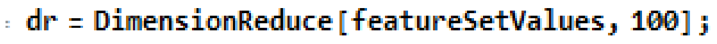
```
fePositive = FeatureExtraction[prePositive, "TFIDF"];

feNegative = FeatureExtraction[preNegative, "TFIDF"];
```

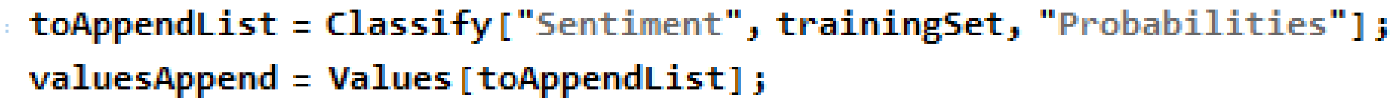
- -The TFIDF Method gives me around 37000 Features to train on my model, as it is too large for processing we have used the following Method:
- DimensionReduce[*examples*,*n*]

It projects onto an approximating manifold in *n*-dimensional space.

- We have also utilized Inbuilt Sentiment polarity of Text Feature of Mathematica, that Classify the sentiment of texts as positive or negative in terms of numerical probability contexts.

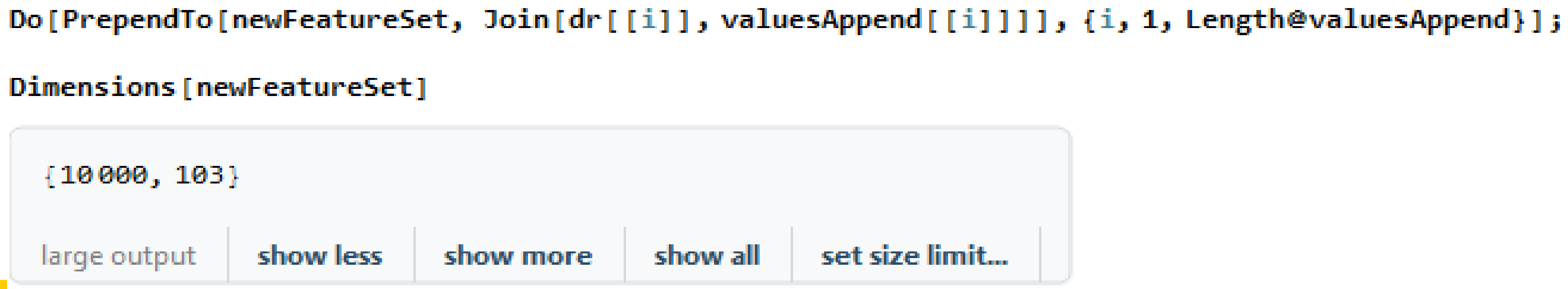


```
dr = DimensionReduce[featureSetValues, 100];
```



```
toAppendList = Classify["Sentiment", trainingSet, "Probabilities"];
valuesAppend = Values[toAppendList];
```

-We have Tested on certain DR Values like 60,80 and 100.,but the accuracy were around not satisfying i.e around 0.66.We merged the attributes we calculated from the above sentiment analysis into our feature set and train our model on 1000*103 set.



```
Do[PrependTo[newFeatureSet, Join[dr[[i]], valuesAppend[[i]]]], {i, 1, Length@valuesAppend}];

Dimensions[newFeatureSet]

{10000, 103}
```

Training And Testing:

1)**Random Forest Algorithm:**

- I have Used Random Forest Algorithm because it is a type of ensemble machine Learning Algorithm and moreover by averaging several Trees.
- It overcomes several problems with decision Tress:
- -Reduction In overfitting: By averaging several trees there is a signigicantly lower rise of overfitting.
- -By using multiple trees we reduce the chance of stumbling across a classifier that doesn’t perform well because of train and test data as it uses the bagging approach in building classification models.
- - Random Forest uses the Majority Vote method and returns the class with most votes.

2)Artificial Neural Networks:

- Neural networks are data driven self adaptive methods in that they can adjust themselves to the data without any explicit specification of functional or distributional form for the underlying model.
- ANN with Back propagation (BP) learning algorithm is widely used in solving various classification and forecasting problems. Even though BP convergence is slow but it is guranted .
- However, ANN is black box learning approach, cannot intepret relationship between input and output and cannot deal with uncertainties.

Training And Testing Contd..

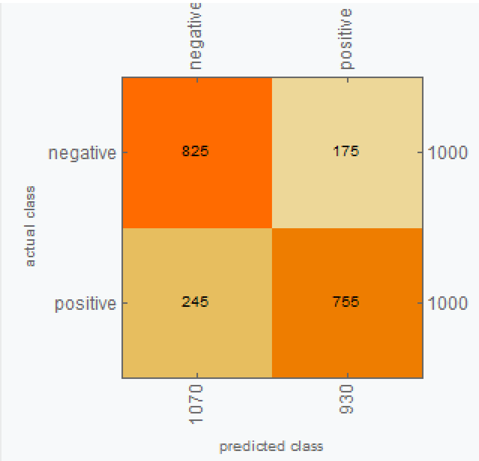
- We went through several research papers on sentiment Analysis and found that deep learning gives better results, and ANN does deep learning on the structures that why our accuracy shoots to 0.79.

3)Logistic Regression:

- Suited for binary classification model, In turn suits for the dataset given to us which has only two class labels either positive or negative .

-The features we have from Sentiment Classifier Function of Mathematic

With the default feature set helps to train our model significantly better result of accuracy 0.81.



-Logistic regression is intrinsically simple, it has low variance and so is less prone to over-fitting.

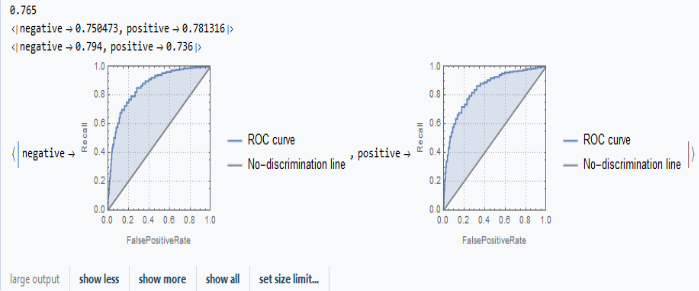
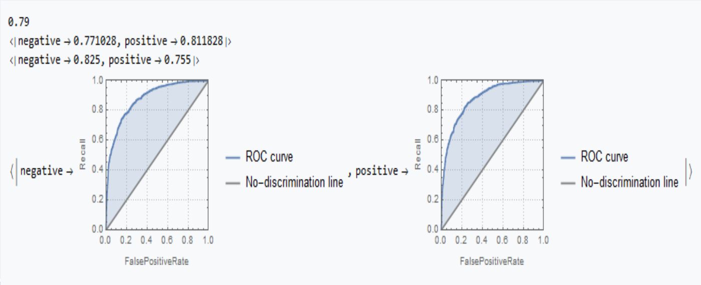
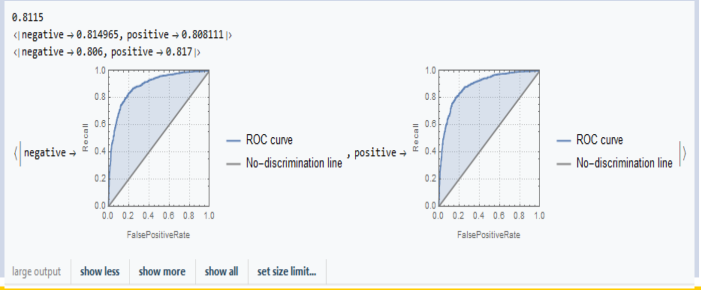
-Logistic regression will work better if there's a **single** decision boundary,

- It is a probability/risk estimator that give you discrete output or outright classes as output.

Findings , Challenges And Conclusion:

- Stricting method to “TFIDF” as a function extractor gives us satisfying result for all the applied algorithms.
- Dimension Reduction/Expansion on one category of plain text given doesn’t affects our results much due to single feature field given in our dataset.
- **Logistic Regression performs better than Naïve Bayes.**
- Various feature selection technique is applied **but ensemble of feature selection can further improve** accuracy and unigram with bag of word gives best accuracy.
- **POS tagging** identifies tagging of word and produces improved result
- Accuracy can be still improved **by doing careful feature selection and proper classification technique.**
- **Choose Non probabilistic Classifier for better results in sentiment Accuracy(Naïve bayes/Markov are not satisfying)**

- To overcome limitation of some techniques, my study concludes we should use of artificial neural networks (ANN) in sentiment classification and analysis. **Our study suggests that the ANN implementations would result in improved classification, combining the best of artificial neural network with fuzzy logic.**

Algorithm	Accuracy	Precision	Recall	ROCCurve
Random Forest	0.765	<|negative->0.750473,positive->0.781316|>	<|negative->0.794,positive->0.736|>	 0.765 <|negative → 0.750473, positive → 0.781316|> <|negative → 0.794, positive → 0.736|>
Neural Network	0.79	<|negative->0.771028,positive->0.811828|>	<|negative->0.825,positive->0.755|>	 0.79 <|negative → 0.771028, positive → 0.811828|> <|negative → 0.825, positive → 0.755|>
Logistic Regression	0.8115	<|negative->0.814965,positive->0.808111|>	<|negative->0.806,positive->0.817|>	 0.8115 <|negative → 0.814965, positive → 0.808111|> <|negative → 0.806, positive → 0.817|>

USC Viterbi
School of Engineering

Deepak Singh|5026958753|singhdee
University of Southern California