FIT5149 – Applied Data Analysis

Sentiment Classification on Product Reviews

Semester 2, 2019

Siddhant Sharma
Student Id: 29047137
Email: ssha0045@student.monash.edu
Master of Data Science
Batch: 2018-19

Shubham Diwe
Student Id: 29329604
Email: sdiw0001@student.monash.edu
Master of Data Science
Batch: 2018-19

Suyash Sadanand Sathe
Student Id: 29279208
Email: ssat0005@student.monash.edu
Master of Data Science
Batch: 2018-19

Table of Contents

1 Introduction

1.1 Brief Overview of Sentiment Analysis

Today, with increase in huge customer base for various products being used and purchased daily by a variety of customers, it has become very important to understand the feedback which is being constantly provided by the customers around the globe. This leads to a use-case in Natural Language Processing domain – Sentiment Analysis which is a study to classify various reviews and customer feedback into different categories. Sentiment Analysis is very important from perspective of organizations which directly interact with end customer as part of their business model. The importance of this area of study can be reflected based on the following:

- Running targeted marketing campaign and lead generation programs
- Enhancing the sales of the product in-turn increase the revenue of business
- Maintaining check on product quality and to keep up with changing customer demands
- Ensuing the best possible customer service to maintain customer engagement
- Enables businesses to manage crisis-situation and anti-organization social campaigns

1.2 Objectives of the Project

The main objectives outlined as part of this project are follows:

- Analyzing the product reviews collected from YELP crowd sources reviews.
- Performing feature extraction steps for getting the most relevant features
- Implementing text pre-processing which includes N-gram extraction, TF-IDF generation, tokenization and stop word removal.
- Implementation of supervised and unsupervised models for classification.
- Identification of best algorithm which provides best results in the project setup
- Analysis of results obtained from different models along with their comparison.

1.3 Reflection of Work

Achievements – Team was able to implement classification models with high accuracy.

Learnings from the project – It had been an amazing experience for each team member as it gave an opportunity for us to learn more about the natural language processing domain along with the understanding and implementation knowledge for various learning algorithms.

Allocation of work – The project was evenly divided within the team members. Each team member initially performed EDA individually and then based on results the EDA methods and techniques were adopted by the team. Same was with the learning models. Once each team member developed different models and tested the accuracy, model with best accuracy was finalized by the team.

2 Pre-Processing and Feature Generation

Starting step for any natural language processing problem is to preprocess the textual information and transform the same into a format which could be easily utilized by the machine translation algorithms to learn information from the corpus and assist in finding an optimal solution for the NLP task in hand. Majorly the different steps and tasks associated in the process of text processing are as below.

2.1 Raw Data Analysis

It is always best to understand and get the high-level information about the data on which we are trying to implement any machine learning algorithms. It is required as it guides in the direction of selecting the most appropriate machine learning algorithm which could be used for that problem and dataset. The different analysis performed as part of the research work as follows:

- Checking for the **sentiment polarity score** to analyze and review the positive, negative and neutral reviews from dataset. performed to understand how do reviews of different polarity score different from each other.
- Checking for the **distribution of sentiment polarity** score to check the skewness if any in the review comments. Based on the analysis the dataset provided from training which includes both the labelled and unlabeled data where **slightly right skewed** showing a slight more number of positive reviews.
- It was analyzed that each type of rating in the labelled dataset where having **similar number of reviews** ranging from 1(negative) to 5(positive).
- Distribution for **length of sentences** for each type of label along with length of sentences for unlabeled data was analyzed. It can be observed that the **distribution is left skewed** and having a long tail on the right side of the distribution.
- Analyzing the distribution of unigram and bigrams with the corpus.

2.2 Data Cleaning

Next step as part of pre-processing was to clean data so that a standardized text corpus is available for all integration and transformation steps. The steps undertaken as part of this phase of text pre-processing were as follows:

- Implemented **text case normalization** to have words in the corpus as lowercase.
- **Removing the punctuation** marks for the text corpus.
- Both **most common words** with a threshold value of 95% and **rare words** with a threshold value of 5% were removed from the corpus. It is computed based on the number of times a word occurs in different reviews.

2.3 Data Integration

As part of this step, the NLTK library was used to get the most updated list stop word, wordnet and punctuations. This list of words was integrated with corpus and used to remove all stop words and punctuation marks from text corpus being used for sentiment analysis task. This done to remove all the words which add very little context to the reviews and are not useful in the sentiment analysis classification task.

2.4 Data Transformation and Reduction

This is one of the most crucial steps in the complete text pre-processing phase. Text in its raw format is difficult to be used by the machine learning algorithms hence it becomes very important for it to be transformed into a format which could be used easily by the algorithms along with providing the ability to perform some statistical analysis and operations. Different steps associated to this phase of text pre-processing are as follows:

- **Word tokenization** was performed by using the regular expression ***“(?u)\\b\\w\\w+\\b”***
- There could be many different words which have the same root but present within the corpus in different forms. **Word Lemmatization** was performed in order to group the similar route words together and replace them with just one word in the corpus.
- Quite a number of words occur in co-occurrence with each other. These are the N-grams which can be extracted from the corpus and are one single word from the perspective of NLP task. **N-Gram extraction** was performed for range from 1-3.
- Another important step is to convert the word tokens into vectorized form for the machine learning algorithm to use these vectors easily for processing. **Term Frequency-Inverse Document Frequency (TF_IDF)** has been used to convert the extracted features into vectorized format.
- All the above steps have been collected together and implemented in one go as part of **TfidfVectorizer** object. This object performs all the above operation all together and returns the TF-IDF vectorized sparse matrix with features equal to the max-feature parameter value. **Max feature parameter value** enables the object to return the most relevant features which are in the request range.

The result achieved after performing all the steps on raw dataset is getting a sparse matrix which could be used for further modelling building purposes. This can be converted to array object or dataframe object but due to computational limitations, it is better to use the sparse matrix as is when required in the model building.

3 Models

3.1 Model 1: Cross validated Logistic Regression Models

In this attempt, we trained a Logistic Regression model for multi-class classification with cross-validation to predict the labels of the unlabeled data. Once we have the entire corpus of reviews with their labels, we can use it to train the logistic regression model again and then predict the labels of the test data set.

3.1.1 Model Implementation

Firstly, we extract the word features of the labeled data set to train the model on these features so that the model learns the hidden weights of the text data distribution. The logistic model instance is created and fed with the sparse vector word representation of the labeled data in the form of the term frequency-inverse document frequency sparse matrix.

While doing so, we also perform cross validation on the training data set. Meaning, we divide the training data in 10 equal parts and pass 9 out of these 10 parts to train the model and predict the labels for the 10th part using this trained model and compute the misclassification rate. Every time, we pass different 9 parts of the data such that the model predicts the label for each data point at least once. Cross validation is done to get an idea of how the model will generalize to about of sample data point estimation. This improves the generalization of the model, prevents overfitting and usually improves the accuracy.

Once the model learns these data parameters, we can use the model to predict the classes of the unlabeled data set. We tried this with some other methods like the Multinomial Naïve Bayes, K-Means clustering, but the best results were obtained with logistic regression. So, once we have the labels for the unlabeled data set, we have huge labelled data set to fit our model to. Though the accuracy is not guaranteed, there should be more correct labels once we merge the unlabeled and labeled data sets.

Finally, we use this large data set to fit a logistic regression model and then predict the labels of the test data.

3.1.2 Model Assumptions and Advantages

The logistic regression model assumes that all the features of the data set are independent even if they are correlated. However, it is a great classifier as it does not assume the normality, homoscedasticity or linearity in the data. Thus, it is a great generalization model used for classification.

The advantages of logistic regression are:

1. Easy to implement.
2. Computationally efficient.
3. No unrealistic assumptions like the normality of the data.
4. No need to scale the input features or tuning of any parameters.

3.2 Model 2: Logistic Regression with Ranked Automatic Keyword Extraction

Ranked automatic keyword extraction (RAKE) is a power tool for sentiment analysis and is used in a lot of tasks where sentiment analysis is required. In this task, RAKE was used for label propagation on the unlabeled data. Then, on the entire corpus, we train the logistic regression model to predict the labels for the test data.

3.2.1 Model Implementation

Rake is a good method of label propagation for text classification as it extracts the keywords from labeled data and then maps the labels to unlabeled data based on the similarity in the keywords for the unlabeled and labeled data. However, it is not a machine learning model that can learn the features. It is just a similarity index for label propagation. Hence, it cannot be expected to generalize to all the data set. So, Rake was used to get the labels for the unlabeled data.

Then, the data corpus of the labeled and unlabeled data was converted to the Tf-Idf Vector sparse matrix format. On this large corpus, we train the logistic regression model to find the correlations between the data and the corresponding labels. Finally, we use this model to predict the test data labels.

3.2.2 Model Assumption and Advantages

The Rake model assumes one keyword to be the essence of the sentence and the phrase containing that keyword to be the most important phrase in the sentence. However, if a word or phrase is repeated more, then Rake assigns high importance to it which may lead to wrong correlations in the model.

The advantages of Rake and Logistic regressions are:

1. Rake is highly scalable. We can analyze huge amounts of data easily.
2. The criteria of label propagation using Rake is consistent. The model, in itself, is highly consistent.
3. Reduces the features in the Tf-Idf matrix as instead of words, important phrases are included.
4. The logistic regression model learns very quickly, even when the amount of data available is limited.

3.3 Model 3 SVM for multi-class classification

Support Vector Machine is a linear model for classification which creates a hyperplane in an n-dimensional space to separate the data for each class. Here, it was first used to propagate labels for the unlabeled data and then to predict the classes for the test data.

3.3.1 Model Implementation

SVM is an easy to implement classifier and is almost as memory efficient as logistic regression. First, we trained the SVM on the labeled data and then predicted the labels for the unlabeled data. Then, a new SVM model was trained on this entire data set create the linear hyperplane boundaries for each class in the entire corpus of the data set.

3.3.2 Model Assumptions and Advantages

The biggest assumption of SVM is that the data is linearly separable. In real life, this may not be the case. However, SVM is great at finding a linear boundary in an n-dimensional space when 'n' is large. SVM assumes the data to be two-class classification problem. So, we need to provide a one versus rest approach for each class for multi-class SVM.

The advantages of SVM are:

1. It is a special case of optimized perceptron.
2. It is easy to implement.
3. Highly effective for high dimensional data.
4. Can perform nonlinear classification by mapping vectors into even higher dimensions.

Disadvantages:

1. Not great for large data sets.
2. Solves K binary classification problems for K class classification.

3.4 Discussion of the models

Initially, we tried K-Means clustering with training on the labeled data and label propagation on the unlabeled data. But, due to the size of the unlabeled data, it was not possible to perform label propagation. So, direct K-Means was used to get labels for the unlabeled data.

- A Logistic Regression model gave an accuracy of 60% on the test data after training on this data set.
- A Multinomial Naïve Bayes (good when small labelled and large unlabeled data) gave an accuracy of 58% after training on this model.
- An SVM gave an accuracy of 59% after training on this model.
- Using the K-Means directly to cluster test data gave an accuracy of just 36% on the test data.

We have a small amount of labeled data and a large amount of unlabeled data. This is a case of semi-supervised learning and hence, use of K-Means, EM for GMMs, etc. are the preferred choices. But due to the size of the data set, it was not possible to perform Expectation Maximization for GMM. SVM is a better choice for high dimensional data spaces but in this case, logistic regression is performing better. Usually, logistic regression is better at generalization as well as its performance is also better. So, we select the logistic regression model for final submission.

4 Experimental Setups

The above discussed model where developed by implementing series of steps in sequential manner.

Models Tried – Multiclass Logistic Regression, K-Means Clustering, Support Vector Machines, Neural Network using LSTM, CNN and GRU.

Use of Labelled-Unlabeled Data – For training the models and calculating the accuracy of unsupervised learning algorithms. Unlabeled data was used to perform K- Means clustering algorithm for getting the predicted labels for large test data-set. Assumption was that the predicted labels would be having less error and would be accurate, hence those can be used to train the supervised learning algorithms and neural networks with large datasets.

Parameter Setting – The most important setting was the number of features which would be required in different experiments. It was observed as the number of features increased the training accuracy was not increasing by any significant amount. Also, having changes in the number of neuron units within the neural networks had a significant impact on training time along with computational resources required.

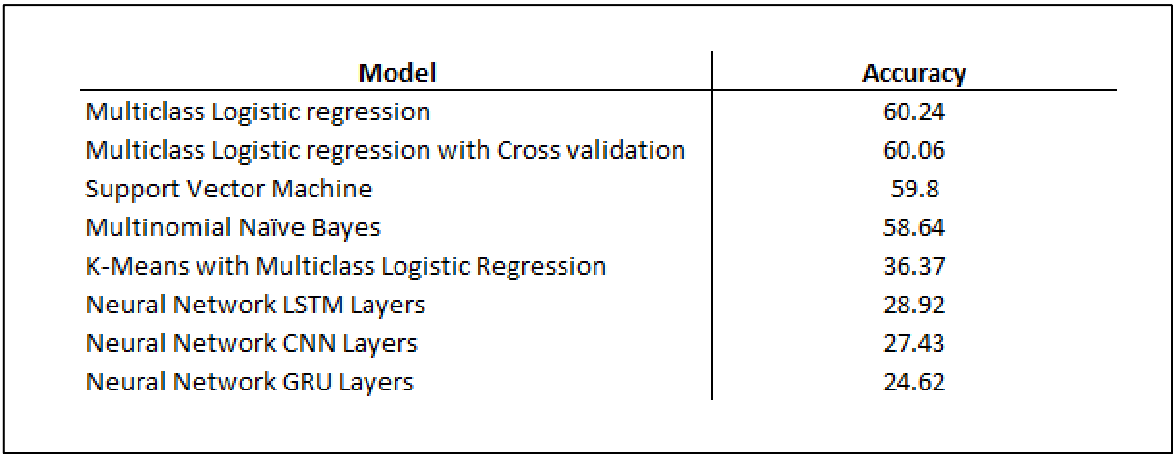
Cross-Validation/Bootstrapping – Both cross validation and bootstrapping was used in training different models. Bootstrapping was done as the number of labelled data was very less. Similarly, cross-validation was done ensure all the uncertainty while training was taken care of and a validation dataset can be made from the labelled data-set itself. The training and test validation dataset split was in 80:20 ratio.

Accuracy computation – Accuracy of trained models was computed using the "accuracy_score" method available in the SKLearn library.

5 Experimental Results

As part of this project there were quite several different experiments which were tried and tested for their accuracy. We were having both labelled and unlabeled data, but the volume of labelled data was very less as compared of unlabeled data. Based on the reading and research of the team it was identified that generally the sentiment analysis models are trained well on large volumes of labelled data which was not available as part of this project.

Unlabeled data was used during the TF-IDF feature reduction and creation step. Also, unlabeled data was used to running the unsupervised learning algorithms which were tried by the team as part of different model building and testing activity. Even though we discuss 3 models in detail in previous section, other models were also tried with an idea to get better model for sentiment analysis and classification.



Model	Accuracy
Multiclass Logistic regression	60.24
Multiclass Logistic regression with Cross validation	60.06
Support Vector Machine	59.8
Multinomial Naïve Bayes	58.64
K-Means with Multiclass Logistic Regression	36.37
Neural Network LSTM Layers	28.92
Neural Network CNN Layers	27.43
Neural Network GRU Layers	24.62

Above are the results which have been collected by performing various experiments on different models. For both Neural Networks and K-Means, the complete dataset of 6,50,000 was used due to which the accuracy of the models is very low. This includes the error in classification which is achieved from the predicted values from unsupervised learning algorithms.

6 Conclusion

From the above results and the overall attempt at sentiment analysis of the Yelp reviews data, we have learned that the traditional machine learning methods all perform in almost the same range. Though SVM is a suitable classifier for such type of problems with high dimensional vector spaces, logistic regression offers an easy to implement and highly generalizable approach.

This is an active area o research and newer ways to optimize the representations of words like the word2vec transformations using a deep neural network offers solutions for faster computations instead of the TF-IDF vectorization.

We did try to implement an RNN which we were hopeful that it will provide the most accurate results. A data size of 650,000 is sufficient to train a small RNN which can outperform traditional machine learning methods subject to condition that we have the entire 650,000 as correctly labeled data.

The major issue we faced during this implementation was the lack of computational resources. Overall, this was a great learning experience and a chance to implement and test the semi-supervised and optimization algorithms like the Expectation Maximization.

The key take-away from this project is that the logistic regression is a great generalizer and a very good classifier as it requires very less computational resources.