IMDB Reviews – Sentiment Analysis

Syed Muhammad Maisam

Abstract

The objective of this analysis is to apply and evaluate various text classification models and techniques to accurately gauge the sentiment in the provided dataset. For this purpose, we have used Logistic Regression and Support Vector Classification models on our IMDB Dataset. Throughout the report we will explore different text pre-processing and factorization techniques that will help us build our models. We will also explore different parametric tuning techniques to further improve our model. Finally, we will explore different evaluation methods to gauge the performance of our model.

1. INTRODUCTION

Text classification is one of the fundamental tasks in Natural Language Processing. The goal is to assign labels to text. It has broad applications including topic labeling [WANG AND MANNING, 2012], sentiment classification [MAASETAL.,2011], and spam detection [SAHAMI ET AL., 1998].

In our analysis, we are primarily concerned with sentiment analysis. Sentiment analysis, also known as opinion mining, is a subfield of natural language processing (NLP) that aims to identify and extract subjective information from text data. The primary goal of sentiment analysis is to determine the sentiment or emotional tone expressed in a piece of text, such as positive, negative, or neutral. By analyzing textual data, sentiment analysis systems can automatically classify the sentiment conveyed in documents, reviews, social media posts, and other forms of text-based communication [PANG, B., & LEE, L., 2008]. There are different methods to perform sentiment analysis such as: lexicon-based methods, supervised machine learning approaches (logistic regression, support vector machines) and deep learning models (convolutional neural network) [LIU, B., 2012].

2. DATA PRE-PROCESSING & PRELIMINARY ANALYSIS

2.1 Loading and Understanding the Dataset

For our analysis we are using IMDB Dataset, compiled by Stanford AI, which roughly contains 50,000 reviews. The dataset can be found here: https://ai.stanford.edu/~amaas/data/sentiment/

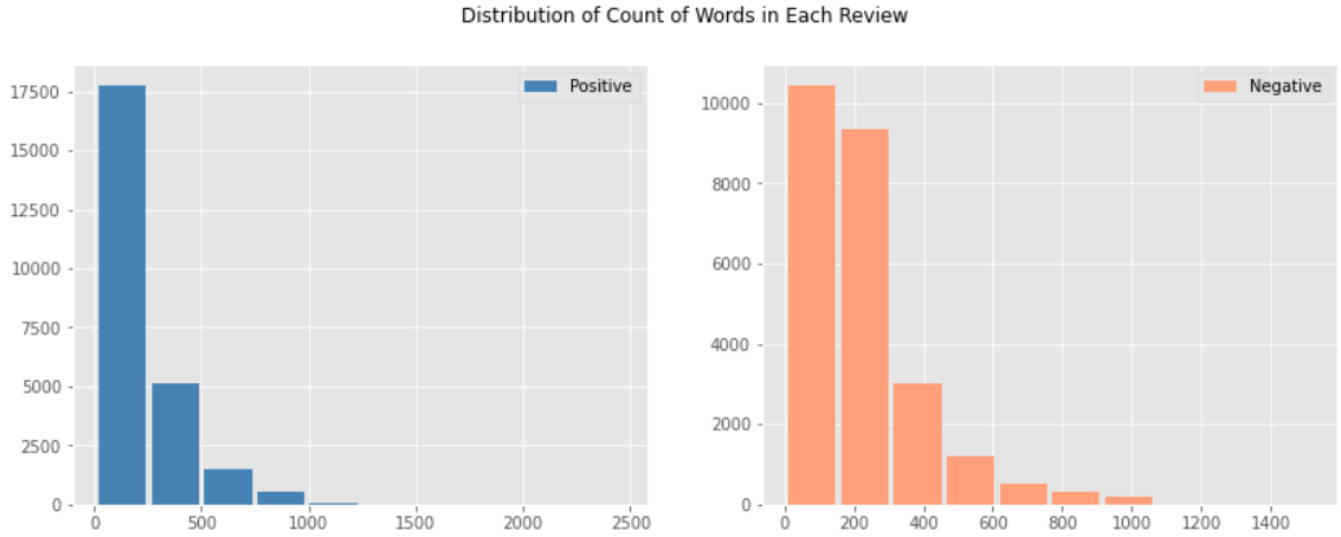
In this dataset, alongside each review, we also have a “positive” or a “negative” sentiment label associated to that review. A restricted tabular representation of our dataset is as below:

	review	sentiment
0	One of the other reviewers has mentioned that ...	positive
1	A wonderful little production. The...	positive
2	I thought this was a wonderful way to spend ti...	positive
3	Basically there's a family where a little boy ...	negative
4	Petter Mattei's "Love in the Time of Money" is...	positive
5	Probably my all-time favorite movie, a story o...	positive
6	I sure would like to see a resurrection of a u...	positive
7	This show was an amazing, fresh & innovative i...	negative
8	Encouraged by the positive comments about this...	negative
9	If you like original gut wrenching laughter yo...	positive

From our initial exploration, we can observe that our class label *“sentiment”* is split equally.

Before diving into text pre-processing, we looked at the distribution of “*length of reviews”* to understand which techniques to apply during the text pre-processing stage.

The breakup of the distribution can be observed below:



2.2 Text Pre-processing and Visualization

Text preprocessing in NLP involves cleaning and transforming raw text data before analysis. It includes tasks such as tokenization, lowercasing, removing punctuation and stop-words, stemming/lemmatization, and handling special characters, URLs, and HTML tags. These steps help improve the quality and usability of text data for downstream NLP tasks and modeling [Bird, S., Klein, E., & Loper, E., 2009].

In our dataset, a sample review can be observed to have such noise as well.

```
Review Example:  [3]
Basically there's a family where a little boy (Jake)
ie is slower than a soap opera... and suddenly, Jake
a film you must Decide if its a thriller or a drama!
have Jake with his closet which totally ruins all the
gless thriller spots.<br /><br />3 out of 10 just for

Sentiment:  negative
```

Therefore, we have applied most of the methods listed above to transform each review. A simple comparison of "*word count*" can demonstrate the significant change in composition of each review.

			Pre	Post
	review	sentiment	word_count	word_count
0	One of the other reviewers has mentioned that ...	positive	307	168
1	A wonderful little production. The...	positive	162	84
2	I thought this was a wonderful way to spend ti...	positive	166	86
3	Basically there's a family where a little boy ...	negative	138	67
4	Petter Mattei's "Love in the Time of Money" is...	positive	230	125
5	Probably my all-time favorite movie, a story o...	positive	119	58

A simple pictorial representation through word clouds or frequency distribution of most used words can give us an intuitive idea of the sentiment carried through in the transformed reviews.

3. TEXT CLASSIFICATION MODELS

3.1 Factorizing and Splitting the Dataset

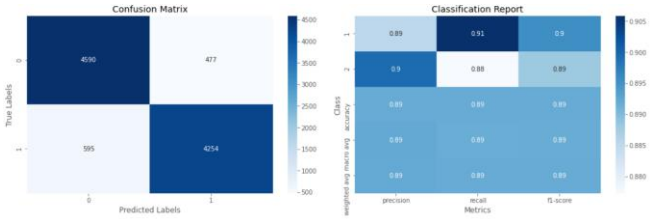
To develop our model, we started by splitting our dataset into "*train and test dataset*". 80% of the dataset ~ 40,000 reviews were used as training dataset and 20% of the dataset ~ 20,000 reviews were used for testing.

We have also performed "*tf-idf vectorization.*" It is used to convert text documents to matrix of tf-idf features. The term frequency-inverse document frequency statistic is a numerical measure of how essential a word is to a document in a collection. [Manning, C. D., Raghavan, P., & Schütze, H , 2008].

3.2 Building Logistic Regression Model

Logistic regression is a statistical model used for binary classification tasks, where the output variable is categorical and has only two possible outcomes. It estimates the probability that a given input belongs to one of the classes using the logistic function [HOSMER JR, D. W., LEMESHOW, S., & STURDIVANT, R. X , 2013].

To evaluate our logistic regression model, we simply created a confusion matrix and classification report that highlights all essential details about the model.

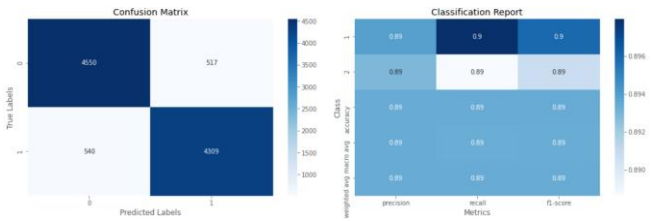


The logistic regression model had test accuracy of ~**89.2%**. The precision, recall and f1-score were all roughly ~**89%** for both class labels. Overall, looking at the parameters, we can conclude that the model's performance is acceptable.

3.3 Building Support Vector Classification Model

Support Vector Classifier (SVC) is a supervised learning algorithm used for classification tasks. It works by finding the hyperplane that best separates different classes in the feature space, maximizing the margin between the classes [CORTES, C., & VAPNIK, V, 1995].

To evaluate our SVC model, we used the same method as above.



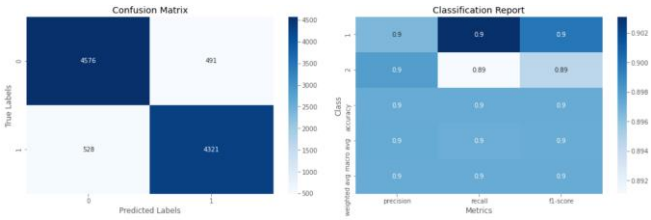
The support vector classifier model performed slightly better than our logistic regression model. The SVC model had test accuracy of **~89.4%**. The precision, recall and f1-score were all roughly **~90%** for both class labels.

3.4 Creating the Best Model

To further improve our existing Support Vector Classifier model, we performed hyper parametric tuning using Grid Search.
Grid search is a hyperparameter optimization technique used to fine-tune the performance of machine learning models by exhaustively searching through a manually specified subset of the hyperparameter space. It involves defining a grid of hyperparameter values and evaluating the model's performance using cross-validation for each combination of hyperparameters. Grid search helps identify the optimal hyperparameters that yield the best performance for a given model and dataset [JAMES, G., WITTEN, D., HASTIE, T., & TIBSHIRANI, R., 2013].
The output of such tuning resulted in:

```
Best Cross Validation Score: 0.90
Best Parameters:  {'C': 1, 'loss': 'hinge'}
```

Using these parameters, we tuned our SVC model and evaluated the performance using the same techniques.



The tuned support vector classifier model performed slightly better than our original support vector classifier model. The tuned-SVC model had test accuracy of **~89.7%**. The precision, recall and f1-score were all roughly **~91%** for both class labels.

To conclude that our model is consistent, stable and has little variability we ran multiple iterations of the same model (10 runs) and calculated standard deviation of relevant parameters such as precision, recall and f1-score. The findings are as below:

```
Precision Standard Deviation: 1.1102230246251565e-16
Recall Standard Deviation: 1.1102230246251565e-16
F1 Score Standard Deviation: 2.220446049250313e-16
```

Since standard deviation across all these metrics is insignificant, we can be confident in our model's performance across the dataset.

4. REFERENCES

Sida Wang and Christopher D. Manning. (2012). Baselines and bigrams: Simple, good sentiment and topic classification. In Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Short Papers - Volume 2, pages 90–94. Association for Computational Linguistics.

Andrew L. Maas, Raymond E. Daly, Peter T. Pham, Dan Huang, Andrew Y. Ng, and Christopher Potts. (2011). Learning word vectors for sentiment analysis. In Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies - Volume 1, pages 142–150. Association for Computational Linguistics.

Mehran Sahami, Susan Dumais, David Heckerman, and Eric Horvitz. (1998). A Bayesian approach to filtering junk e-mail. In Learning for Text Categorization: Papers from the 1998 workshop, volume 62, pages 98-105.

Pang, B., & Lee, L. (2008). Opinion mining and sentiment analysis. Foundations and Trends® in Information Retrieval, 2(1–2), 1–135.

Liu, B. (2012). Sentiment analysis and opinion mining. Synthesis Lectures on Human Language Technologies, 5(1), 1–167.

Bird, S., Klein, E., & Loper, E. (2009). Natural Language Processing with Python: Analyzing Text with the Natural Language Toolkit. O'Reilly Media.

Manning, C. D., Raghavan, P., & Schütze, H. (2008). Introduction to Information Retrieval. Cambridge University Press.

Hosmer Jr, D. W., Lemeshow, S., & Sturdivant, R. X. (2013). Applied logistic regression. John Wiley & Sons.

Cortes, C., & Vapnik, V. (1995). Support-vector networks. Machine learning, 20(3), 273-297.

James, G., Witten, D., Hastie, T., & Tibshirani, R. (2013). An Introduction to Statistical Learning: with Applications in R. Springer.