# MiniProject 2: IMDB Sentiment Analysis

COMP 551

McGill University

Submitted to

Professor William L. Hamilton

Submitted by

Marcos Cardenas-Zelaya

Viet Nguyen

Gagandeep Singh

February 24, 2019

## Abstract

Sentiment analysis is an ongoing and widespread practice both in academia and industry. The need for adequate text analyzers is an ongoing challenge. In this paper we approach a binary sentiment analysis by predicting movie reviews sentiment as positive or negative on the IMDB Movie Review Dataset. We make use of various feature extraction processes such as N-grams and term frequency * inverse document frequency (TF*IDF). We then assess and compare the performance of Bernoulli Naive Bayes (BNB), support vector machine (SVM), logistic regression (LREG), extreme gradient boosting (XGB), dense shallow neural network (NN), and bidirectional LSTM (biLSTM) on this data set. Finally, we compare and analyze each model's performance on our sentiment analysis task.

## Introduction

Sentiment analysis is a classic natural language processing (NLP) task in which one predicts a positive or negative sentiment given the review texts. Although state of the art deep learning models is gradually achieving higher accuracy scores on standard datasets, we demonstrate in this paper that the more traditional feature extraction and transformation methods with classic models perform just as competitively. An accurate sentiment analysis allows many businesses across different industries to understand their customers better. Knowing the target audience allows for better products and services offered to consumers. In our study, we train our models on the IMDB dataset, consisting of 12,500 negative and 12,500 positive reviews. We find that overall, by selecting features from combined uni-bigram counts into TF*IDF transformation, most models were able to achieve higher F1-scores and accuracy scores, reinforcing the importance of manual preprocessing and feature extraction/selection in NLP. We address the ease of indirect overfitting of our models through hyperparameter tuning.

We approach the classification task using feature extraction practices such as count vectorization (binary occurrences and standard count), N-grams, feature transformation methods such TF*IDF, word embeddings, feature selection processes using principal component analysis (PCA) and Chi-squared testing. We assess and compare the performance of BNB, SVM, LREG, XGB, NN, and biLSTM. We report accuracy and F1-score on different 10-fold validation splits. We find that SVM on unigram and bigram features with TF*IDF and feature scaling on 100,000 features achieved an average accuracy of 0.922 (+/-0.016), with an F1-score of 0.924 on a held-out test set at 1/5th of the training set size.

## Related work

Tripathy et al. considered several models and feature selection on IMDB data including NB, Maximum Entropy and SVMs using N-grams and TF*IDF feature selection. They observe that as the N-gram length increases, the classification accuracy decreases. They note that the best results were obtained with unigram and bigram, but that the accuracy quickly declines when using tri-grams and above. Further, they find that an SVM unigram and bigram performs best with an accuracy of 88.8 and an F-1 score of .89 (Tripathy et al, 2016).

Wang and Manning propose a Naive Bayes Support Vector Machine (NBSVM). They show that NB is well suited for shorter documents and SVM better suited for longer document. They propose the NBSVM that is otherwise identical to the SVM except that its features are drawn from the multinomial NB model. This model is an interpolation between the NB and SVM model. The interpolation can be understood as a form of regularization; in which the algorithm trusts NB unless the SVM is very confident (Wang and Manning, 2012).

## Dataset and Setup

The IMDb dataset consists of 12,500 negative reviews and 12,500 positive reviews. This balanced data set will prevent certain classifiers from producing biased results as some are sensitive to data proportions which favour the majority class. Although the data is balanced, from figure 1, we note the distributions in text lengths is skewed. Review have an average letter count of 1278 with a standard deviation of 967. For all models employed, the common pre-processing methods included removing stop-words, non-ascii keys as they showed up frequently in the corpus, lemmatization, normalization of words, changing to lower-case and tokenizing words to extract features such as count, binary count, and TF*IDF weighting.

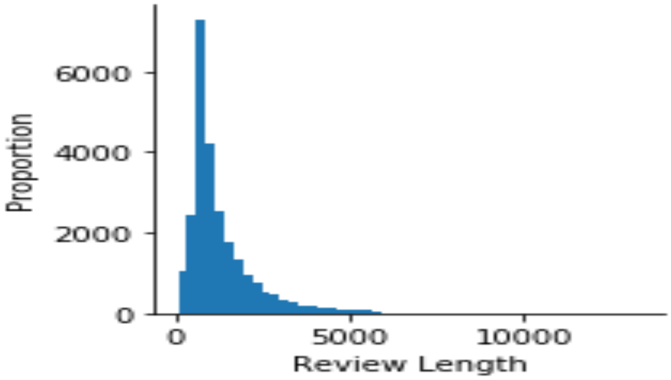

Figure 1. Distribution of review lengths in the IMDB dataset.

## Proposed approach

We propose a Bernoulli Naive Bayes (BNB) model on the word count of unigrams in the dataset. Laplace smoothing is used so that words that are not observed in the training data will have a prior probability of .5 to prevent documents containing this word from being automatically set to a 0 probability. The Bernoulli model acts as a base model for which all other models are compared to as it is the simplest in implementation and in features. Given that this model was implemented not using any libraries, we found that this limited us in the size of the vocabulary used in the feature selection, however we note that the accuracy does not change drastically when the vocabulary size increases, but the runtime increases dramatically with an increasing vocabulary.

We attempted a LREG model as the discriminative counterpart to BNB, using L2 regularization and a SAGA solver (Defazio et al, 2014). We expect logistic regression to perform just as well as the BNB model as it accounts for correlation between features whereas the latter makes the independence assumption on the model's features.

We propose a soft-margin linear SVM model using L2 regularization on our dataset, as we believe inferring a spatial separation between our feature components would give us competitive results as achieved by Wang and Manning in their paper on NBSVM, as well as the fact that SVMs tend to excel at binary classification and anomaly detection as it doesn't suffer from imbalanced data samples in each class as much as other high performant popular classifiers.

We explored the impact of a gradient boosting model using the common XGB implementation, using as base estimator's shallow decision tree (DT) classifiers with 1,500 base estimators. We believe that as misclassified labels tend to be the same ones all the time for several of our previous models, a boosting method which heavily weighs the wrongly labeled samples would deliver a good performance. XGB is strongly resistant to overfitting, and hence we conjecture that a higher accuracy could be achieved with more base estimators.

We then explore deeper methods including a NN using maximum regularization from dropout (p=0.5) and batch normalization, using rectified linear units (ReLU) and sigmoid activations and binary cross-entropy loss, and trained using the Adam optimizer. Our neural network was trained from 20k features selected based on chi-square testing, in which we downsized to 5k features with PCA. As for the biLSTM model, we followed a suggested approach and trained an 8 cell LSTM layer to produce a transformation of the original text sequence and feed it into a dense neural network with dropout (p=0.20) for the final prediction [1]. Our embedding layer size was 256, and we padded text-transformed sequences to a max length of 200 (as around 2/3 of reviews are under this length) and trained on mini-batch sizes of 100 over 3 epochs.

We implemented the NBSVM model from the proposed paper by Wang and Manning as we share the dataset used in their paper. As the authors point out, the NB is better equipped at predicting short documents while SVM is better equipped at longer documents. Given that the IMDb datasets contain longer reviews, the SVM will be better at predicting on average, however may it not be so equipped for shorter document outliers which would be better handled by NB. As such, we employ the NBSVM model. We find that using bigrams yielded the best accuracy[2].

Through our feature engineering process, we employed various feature construction techniques, the best performer being TF*IDF on uni and bigrams with minimum occurrence greater than 5 (to avoid sparsity). We noticed that the use of raw word count and binary occurrences as models on these features tend to be weaker in terms of generalization to our validation set. We used the TF*IDF for our models because it embodies the intuitions that the more often a term occurs in a document, the more it is representative of its content, and the more documents a term occurs in, the less discriminating it is. The n-grams provide a more natural context to a sentiment by bridging words whose context is highlighted by another word, and we find that combining uni and bigrams led to the best results. We performed feature selection using PCA and chi-square component testing. While PCA maps the feature space into a lower dimensional space while minimizing information loss, the chi-square captures the probability of which class a word is most likely to be associated with; we find this a natural fit for a bag of words approach. Further, the chi-square test gives an estimate of goodness of fit and selects features that best correlate with the class label of samples in our dataset, and we found that our models learned better with features selected using the chi-squared test. We select 50,000 features (100,000 for SVM) using scikit-learn's SelectKBest function, with the chi-square scoring function. Our optimal hyperparameters for our models were found through the commonly used grid search method.

We set aside 5,000 of our 25,000 training samples as held-out validation and ran 10-fold cross validation on the other 20,000 samples to calculate our chosen evaluation metrics and report the accuracy on the held-out validation set.

## Results

Table 1. Evaluation parameters for the implemented models.

| **Model** | **Avg. Accuracy** | **F1-score** | **Precision** | **Recall** | **# Features** |
|---|---|---|---|---|---|
| BNB | 0.812 | 0.813 | 0.814 | 0.812 | 1000 |
| **Uni-bi SVM (TF*IDF)** | **0.922** | **0.924** | **0.919** | **0.929** | **100,000** |
| Uni-bi SVM (binary) | 0.889 | 0.888 | 0.886 | 0.890 | 50,000 |
| Uni-bi NBSVM (TF*IDF) | 0.910 | 0.911 | 0.899 | 0.924 | 50,000 |
| LREG (SAGA) | 0.884 | 0.885 | 0.879 | 0.891 | 50,000 |
| XGB | 0.883 | 0.885 | 0.875 | 0.894 | 50,000 |
| NN(5k, 750, 400, 1) | 0.863 | 0.864 | 0.862 | 0.865 | 5,000 |

[1] https://www.kaggle.com/nilanml/imdb-review-deep-model-94-89-accuracy

[2] We note that results for NBSVM in this paper are produced using an implementation cited on the code

|  |  |  |  |  |  |
|---|---|---|---|---|---|
| biLSTM(8)<br>Dense NN(128, 1) | 0.860 | 0.857 | 0.855 | 0.860 | 50,000 |

Using the uni-bi SVM without feature scaling, we were able to achieve a competitive accuracy on our held-out validation set, and a **Kaggle leaderboard accuracy of 0.890**. We observe that the linear SVM delivered a worse performance on binary occurrences (F1-score of 0.888 vs. 0.924) than on transformed TF*IDF. We hypothesize that this is since term frequency and document frequency contribute significantly to making accurate predictions on the validation set. For example, a document can contain both the words "amazing" and "terrible", where amazing appears more often as an IMDb critic would want to state both good and bad opinions on a certain movie. A binary count would note that both words appear in the document, and our classifier will not be able to make sense of the reviewer's intent while a TF*IDF transformation would help the model make the connection between higher frequencies and corresponding class labels, resulting in better prediction.

We remark here that our BNB model performed significantly worse than the other models tested. We hypothesize that such underperformance is caused by strong correlations between word features that are semantically related, while the BNB model assumes independence of the extracted features. Although, we note that given the small number of features used and relative simplicity compared to the competing models it performs better than expected.

We remark that although we chose twice as many features as we have of sample points, a linear SVM would still be able to perform well as linear SVMs scale well, as their decision boundary is linear. If it were a non-linear SVM kernel, solving an optimization in an infinite dimensional space would result in a decision boundary of arbitrary complexity and thus be prone to overfitting. A similar argument can be made for our LREG model. For our NN model, dropout and batch normalization as implicit regularization methods will automatically infer the correlation between features and the corresponding output for training samples, and thus if the number of hidden layers and nodes per hidden layer is reasonable, the effects of having many features does not significantly impact the performance of our model.

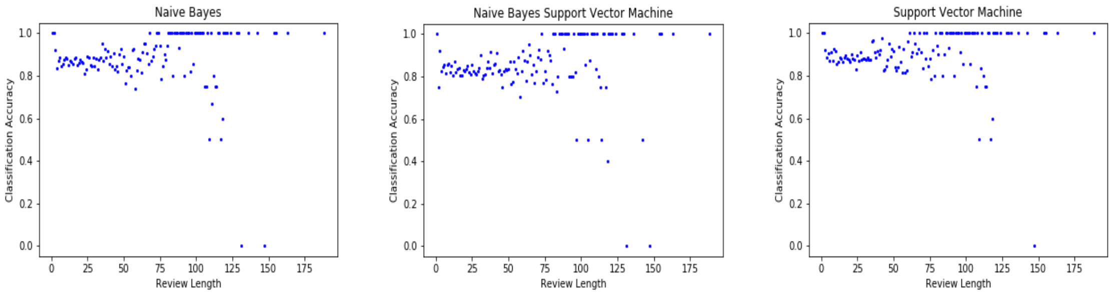

Figure 2: Classification accuracy vs. review length in tokens, grouped in bins.

According to Wang and Manning; NB, SVM and NBSVM would perform better on shorter, longer and all lengths of documents, respectively. However, we were not able to reproduce these results; from figure 2 we see that they our models performed about the same by predicting medium sized documents the best which suggests that these models are not that different with respect to document length. Further, given the similar results yet differences in complexity, this would suggest that it would be better to choose the simpler model of SVM over the more complex NBSVM. In all cases we note that accuracy decreases with the review's length.

We note that all the measures in table 1 are very similar which indicates that our models are not prone to overfitting, and its predictions can generalize accurately without making a disproportionate amount of type 1 or 2 errors. These algorithms could be extended to determine whether a movie is worth watching or not based on its reviews. In this context we favour a higher precision score over the recall score as a false negative is not a detrimental outcome since we believe watching a bad movie when the model indicated otherwise is not a life and death situation. As such we prefer to not err on the side of caution with our movie reviews. However, we note that if this model was extended to more important contexts such as tumor predictions or even sentiment analysis for legal documents that could affect the livelihoods of individuals, then we would suggest erring on the side of caution and prefer a model with a higher recall.

We note the effects of indirect overfitting to the held-out validation set as we achieved a competitive accuracy of 0.922 with the TF*IDF SVM model, while our Kaggle leaderboard score was only 0.890. Further, as we scaled our features to have 0 mean and 1 variance as a standard preprocessing step for SVM classification, we achieved a held-out validation accuracy of 0.960; which is most likely due to overfitting. More surprisingly, the individual accuracies on each of the 10-folds in our validation pipeline were all above 0.95. The same effect could be observed with the NN model where our k-fold validation accuracy would approach 0.900, while the held-out validation accuracy was 0.863.

## Discussion and Conclusion

This paper attempts to classify movie reviews using various supervised machine learning algorithms most notably Naive Bayes and Support Vector Machines. The models are applied using feature selections such as N-gram, Chi-squared and TF*IDF. We find that SVM combined with low level N-grams provided the highest accuracy score. We recognize the competitive performance of the more traditional approaches to text classification and manual feature engineering (SVMs, LREG, etc…), outperforming our biLSTM and NN models, in which feature construction happens implicitly. Future research could benefit from choosing models that are more robust to document lengths.

Various future investigations can be done on this classification task, namely training word embeddings scaled by TF*IDF weighing, or training paragraph vectors as proposed by Le and Mikolov in *Distributed Representations of Sentences and Documents (2014)*.

## Statement of Contribution

Viet: Coded the model and feature extraction pipelines as well as deriving the highest scoring models. Wrote half of the report.
Marcos: Coded the Bernoulli Naïve Bayes and NBSVM. Wrote the other half of the report.
Gagandeep: Assisted a bit in writing the report.

# *Citations*

1. Defazio, Aaron et al. (2014). SAGA: A Fast-Incremental Gradient Method With Support for Non-Strongly Convex Composite Objectives.
2. Tripathy, Abinash. Agrawal, Ankit. et al (2016). Classification of sentiment reviews using n-gram machine learning approach. Expert Systems with Applications 57, 117-126.
3. Wang, Sida. Manning, Christopher D. (2012). Baselines and Bigrams: Simple, Good Sentiment and Topic Classification. Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics, pages 90–94, Jeju, Republic of Korea, 8-14 July 2012
4. Le, Mikolov (2014). Distributed Representations of Sentences and Documents