Mini Project 2 Report

Classification of Textual Data

COMP 551: Applied Machine Learning

Maxime Buteau	260868661
Charles Couture	260923463
Eric Pelletier	260895863

McGill University
March 7th, 2022

Abstract

A key element in machine learning is the ability to predict outcomes with the highest accuracy possible. In this project, we investigated the performance of two machine learning models —*Multinomial Naive Bayes* and *Softmax Regression*— on two benchmark datasets —*20 News groups* and *Sentiment140*—. During the experiments, we conducted hyperparameter tuning of our models on the two datasets with the aid of 5-fold cross validation to evaluate the performance of our models using accuracy. In addition, we evaluated the effects that different sizes of training sets have on our models' accuracy. In the end, we found that the Multinomial Naive Bayes model performed better for the 20 News groups dataset, while the *Softmax Regression* model performed better for the *Sentiment140* dataset. Overall, the best performing model was the *Softmax Regression* model.

1 Introduction

During this project, we conducted multiclass classification on the 20 News group and Sentiment140 datasets by implementing a multinomial naive Bayes model from scratch and comparing it to scikit learn's softmax regression model. In addition, we implemented a K-fold validation for tuning of the models' hyper-parameters and these algorithms were trained using the provided training sets in the datasets. In the context of the 20 News group dataset, the algorithm is used to classify news group documents from the test set into 20 different news groups, each corresponding to a different topic. In the context of the Sentiment140 dataset, the algorithm is used to classify tweets from the corresponding test set based on their polarity —negative and positive—.

Feature preparation is crucial and can be more important than the choice of classification algorithm. A common approach is the bag-of-words, where words in the text are counted out of context and their frequency is calculated. It has been shown that using Inverse Document Frequency with this approach improves performance [Salton and Buckley, 1988]. Using this term frequency-inverse document frequency (TF-IDF) approach allows to reduce the impact of words that appear very often in different documents (such as common English words).

The multinomial implementation of the naive Bayes model was selected in this project since we are dealing with word counts which are discrete. For the likelihood function of this model, we add up all the occurrences of a particular word in the documents of a same class, and divide by the total number of words in that class to get a probability. We noticed that this approach works even better with TF-IDF, even if this gives continuous frequencies. The only hyperparameter of the multinomial naive Bayes model is alpha, which determines how many "fake" occurrences of words are added to the training set for Laplace smoothing. This procedure prevents 0 probabilities when a word has not appeared in a certain category.

Since our implementation of Naive Bayes can take up to 20 minutes to run with the entire training and test sets, we decided to use SciKit Learn's implementation of Multinomial Naive Bayes for the experiments (Task 3) in this project. Both our implementation and SciKit Learn's achieved very similar accuracies on the twenty news group dataset. We could not compare for the Sentiment140 dataset, as our implementation would just run out of memory. The main difference seems to be that SciKit Learn's implementation can handle python sparse matrices, while ours needs to convert these to arrays. This requires a lot more memory.

On the other hand, the softmax implementation of the logistic regression model was selected in this project since we are dealing with multiple classes. The softmax regression model was implemented through the SciKit Learn library and contains 15 different hyper-parameters. During the experiments, the tuning of the hyper-parameters was performed with a 5-fold cross validation for both models.

By testing different values of the hyperparameter alpha in the multinomial naive Bayes model, we evaluated the effects that this value has on the model's accuracy. As well, by testing different values of the hyper-parameters C and intercept scaling in the softmax regression model, we evaluated the effects that these values have on the model's accuracy.

We found that better performing model for the 20 news group dataset was the multinomial naive Bayes model and that the better performing model for the Sentiment140 dataset was the softmax regression model. However, the best overall performing model was the softmax regression model.

2 Datasets

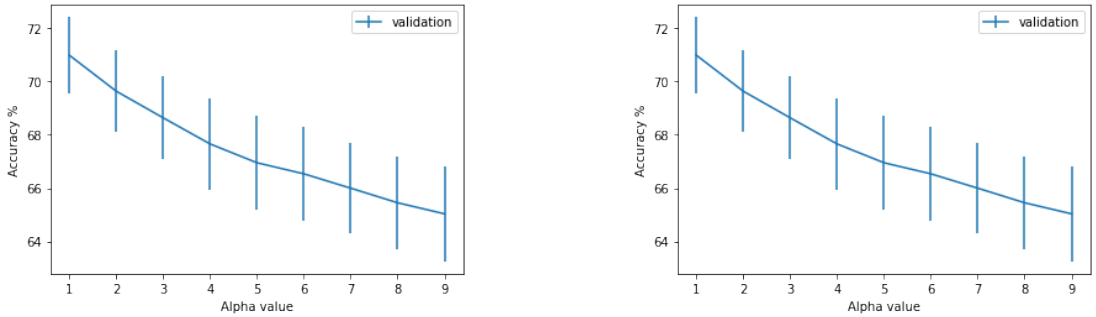
As stated above, the two datasets processed during this project are the 20 News groups and Sentiment140 datasets. Following the information provided on *qwone.com*, the 20 news groups dataset, originally collected by Ken Lang, contains around 20,000 news group documents which are evenly partitioned across 20 different news groups. Each news group corresponds to a different topic. On the other hand, the Sentiment140 contains over 1,600,000 different tweets. The sentiment140 website states that each tweet contains a polarity, id, date, query, user and text.

Both datasets were processed using a tf-idf vectorizer function in which we ignore the terms that are found in more than 50% of documents since they are likely not very informative as well as the common english words such as 'the', 'and', 'him'. The tf-idf vectorizer function starts by using the bag of words representation by assigning a fixed integer id to each word occurring in any document of the training set and then counting the number of occurrences of each word. We will then have a number of features equal to the number of distinct words in the corpus. However, as stated on SciKit Learn, "longer documents will will have higher average count values than shorter documents, even though they might talk about the same topic" [Pedregosa et al., 2011]. Therefore, to avoid these potential discrepancies, the td-idf vectorizer function divides the number of occurrences of each word in a document by the total number of words in the document which create new features called term frequencies. In addition, the td-idf vectorizer function utilizes downscale weights for words that occur in many documents which is called term frequency times inverse document frequency.

3 Results

3.1 Comparison of the Accuracy of the Models on Both Datasets

We started off by performing hyperparameter tuning for the multinomial naive Bayes model with both datasets separately by comparing the accuracy of the model with alpha values ranging from 1 to 9. The accuracies of the model for the 20 news groups and sentiment140 datasets are displayed in figure 1



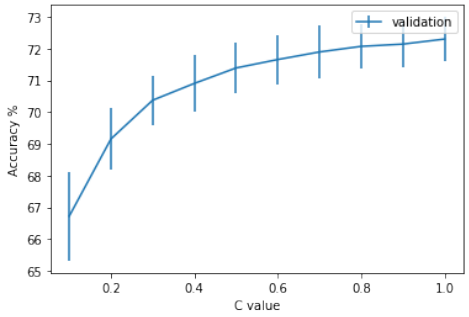
(a) 20 News Groups Dataset

(b) Sentiment140 Dataset

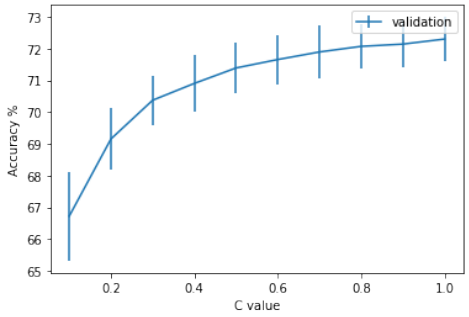
Figure 1: Accuracy of Multinomial Naive Bayes Model With Various Alpha Values

From these graphs, we can deduce that the best value of alpha is 1 for the 20 news groups dataset as well as for the sentiment140 dataset.

We then performed hyperparameter tuning for the softmax regression model with both datasets separately by comparing C values ranging from 0.1 to 1 with increments of 0.1 as well as intercept scaling values ranging from 0.1 to 1 with increments of 0.1. The accuracies of the model with the various C values and intercept scaling values for the 20 news groups and sentiment140 datasets are displayed in figures 2 and 3 respectively.

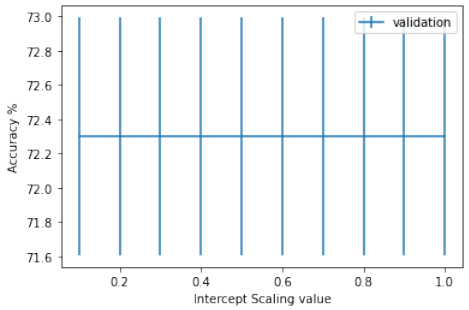


(a) 20 News Groups Dataset

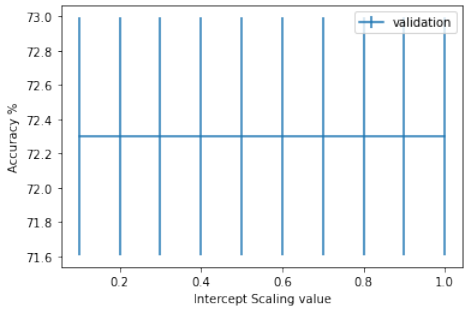


(b) Sentiment140 Dataset

Figure 2: Accuracy of Softmax Regression Model With Various C Values



(a) 20 News Groups Dataset



(b) Sentiment140 Dataset

Figure 3: Accuracy of Softmax Regression Model With Various Intercept Scaling Values

From figure 2, we notice that the best value of C seems to be 1.0 for the 20 news groups dataset and 0.8 for the sentiment140 dataset. In addition, from figure 3, the best intercept scaling value seems to be 1.0 for the 20 news groups dataset as well as for the sentiment140 dataset.

Following the tuning of the hyper-parameters, we evaluated the accuracies of the models on the test sets of both datasets using the best hyper-parameters values stated above. We additionally calculated the overall accuracy of each model by taking their average accuracy between both datasets. The accuracy of the models are illustrated in figure 4

	Model	20 news group accuracy	Sentiment140 accuracy	Overall accuracy
0	Multinomial Naive Bayes	67.604886	80.779944	74.192415
1	Softmax Regression	67.087095	81.615599	74.351347

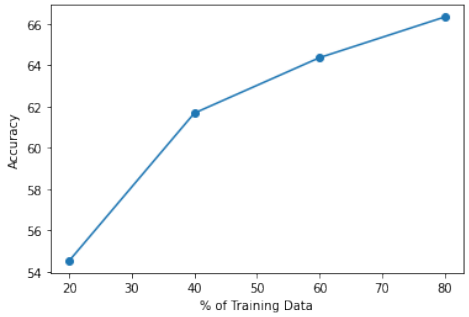
Figure 4: Comparison of Accuracy of Multinomial Naive Bayes and Softmax Regression Models

As this table illustrates, the better performing model for the 20 news groups dataset was the multinomial naive Bayes model. On the other hand, the better performing model for the sentiment140 dataset was the softmax regression model. Overall, the software regression model performed better than the multinomial naive Bayes model.

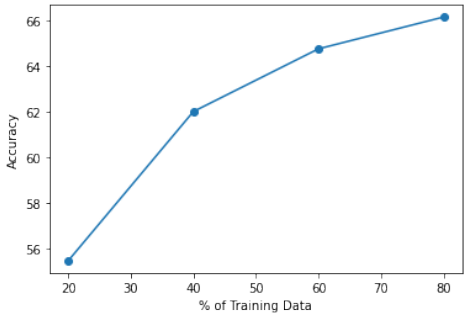
3.2 Accuracy Based on Different Data Size

3.2.1 20 News Groups Dataset

We tested both models performance with various percentages of the total training set size. We tested values of 20%, 40%, 60% and 80%. The accuracies of the models can be seen in the figure 5.



(a) Multinomial Naive Bayes Model



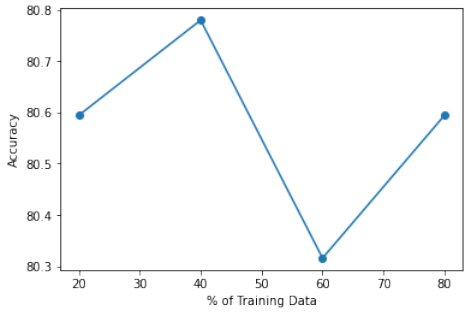
(b) Softmax Regression Model

Figure 5: Accuracy of multinomial naive Bayes and Softmax Regressions model with various training set sizes

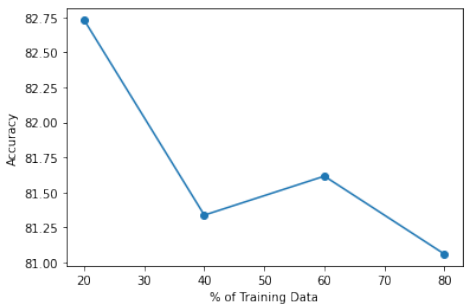
From this figure, we notice both models behave very similarly when working on a subset of the train data. However, softmax regression seems to have a slight edge over Naive Bayes at lower percentages, while Naive Bayes performs slightly better when reaching higher percentages.

3.2.2 Sentiment140 Dataset

Next, we tested both models' performance with various percentages of the total training set size as seen on the 20 news groups dataset. We tested values of 20%, 40%, 60% and 80%. The accuracies of the models can be seen in the figure 6.



(a) Multinomial Naive Bayes Model



(b) Softmax Regression Model

Figure 6: Accuracy of multinomial naive Bayes and Softmax Regressions model with various training set sizes

For the Sentiment140 dataset, the size of the training set does not seem to affect the accuracy very much since 20% of the training data is still more than 300,000 documents, and each document is rather short. This means that a good vocabulary that covers most important words can be established with a subset of the training data. It is also worth noting that there are only 2 classes in the sentiment140 dataset compared to the 20 classes in the 20 news group dataset

4 Discussion and Conclusion

Overall, the multinomial naive Bayes and softmax regression models performed very similarly on the given datasets. The multinomial naive Bayes model provided a slightly better accuracy while working with the 20 news group dataset. On the other hand, the softmax regression model outputted a better accuracy than the competing model while working with the sentiment140 dataset and was found to be the better performing model overall. As we were not able to test out all of the various hyper-parameters of the softmax regression model due to its large number of features, we could further expand our research on this model by testing

other features. In addition, we could further our research by using the count vectorizer instead of the tf-idf vectorizer and tuning some of the parameters of these vectors.

5 Statement of Contributions

We all contributed equally.

References

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

Gerard Salton and Christopher Buckley. Term-weighting approaches in automatic text retrieval. *Information Processing & Management*, 24(5):513–523, jan 1988. doi: 10.1016/0306-4573(88)90021-0. URL `https://doi.org/10.1016%2F0306-4573%2888%2990021-0`.