Classification of Textual Data

Yu Yun Liu; Ruoyu Deng; Pengyu Xue

March 8, 2022

Abstract

In this project, we investigated the performance of linear classification models on two benchmark datasets, 20 News Group and Sentiment140 datasets. The datasets are composed of multiple texts. We implemented the Naive Bayes model and k-fold cross validation functions from scratch. We trained the models with 5-fold cross validation to find the best hyperparameters, which we predict on the unseen datasets. We measured the performances of Multinomial Naive Bayes and Logistic (Softmax) Regression models with their best hyperparameters on two different test datasets. From observations of experiments, we found that the logistic regression approach achieved better accuracy than Naive Bayes on a small size vocabulary set (Sentiment140 datasets). On the other hand, the Naive Bayes model can obtain higher test accuracy than the logistics regression model on a large size vocabulary set (20 newsgroup dataset). The model with higher accuracy will take a longer time to train and make predictions.

1 Introduction

This project aims to compare two linear classification models, Naive Bayes and softmax regression. To do so, we implemented a multinomial Naive Bayes model from scratch, and we trained it using the k-fold cross validation method, which we also implemented from scratch. For this experiment, we chose a k equal to 5. We used the softmax regression from the Scikit-Learn package for the latter model. We used two textual datasets, 20 News Group and Sentiment140, to train the models. We trained the Naive Bayes model using different smoothing parameter (alpha) values. Under different alpha values, we observed a difference inaccuracy in the model. For the softmax regression, we selected the Penalty and C hyper parameters to receive the accuracy for each datasets. We picked the best values of hyper-parameters for both models in the training process. We then proceeded to test our model on the testing dataset we had. The predictions obtained from testing the models show that the model results in better accuracy than the model.

2 Data sets

We were given two datasets, the 20 News Group and the Sentiment140 datasets. The first dataset comprises 11 314 newsgroup texts/documents. Each one belongs to one of the 20 possible different newsgroups.

```
['alt.atheism',
 'comp.graphics',
 'comp.os.ms-windows.misc',
 'comp.sys.ibm.pc.hardware',
 'comp.sys.mac.hardware',
 'comp.windows.x',
 'misc.forsale',
 'rec.autos',
 'rec.motorcycles',
 'rec.sport.baseball',
 'rec.sport.hockey',
 'sci.crypt',
 'sci.electronics',
 'sci.med',
 'sci.space',
 'soc.religion.christian',
 'talk.politics.guns',
 'talk.politics.mideast',
 'talk.politics.misc',
 'talk.religion.misc']
```

Figure 1: Sample News Groups.

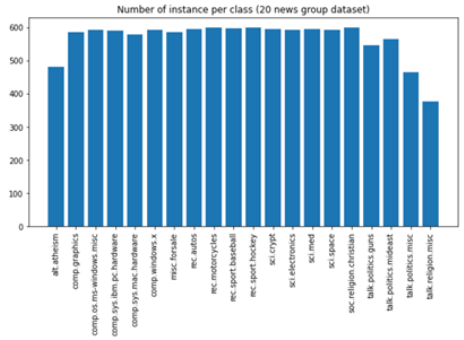

Figure 2: Number of instance per class (20 news group Dataset)

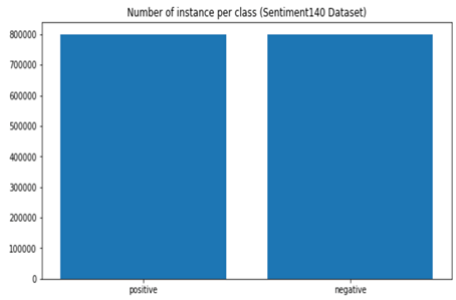

Figure 3: Number of instance per class (Sentiment140 Dataset)

The latter one describes 1 599 999 tweet texts along with their corresponding polarity, i.e. whether the tweet is negative or positive. To be able to manipulate both text data and use it to train our model, we had to turn them into numerical features. We used the bags of words representation to represent the data which means that for every possible word in the whole dataset, there is an index associated with this word. We store the word count that appears in each text data at that index.
In the newsgroup dataset, we have a total of 101 631 distinct words across the whole dataset, and in the sentiment dataset, we have a total of 26 216 distinct words. We plotted the distribution on the number of text data in each class to better understand the datasets. We notice that for both datasets, the data mainly was evenly distributed between each class, i.e. each class had around the same amount of text data.

3 Results

Since the training data is relatively enormous, our experiment training data is estimated based on an appropriate portion of the total training data of the datasets. Specifically, we sampled 2000 instances from both original training datasets, and we selected 500 instances from the testing data of the 20 newsgroup dataset. We then started running the experiment by conducting multiclass classification on the 20 news group and Sentiment140 datasets. The hyper parameter of multinomial Naive Bayes is alpha, also called the smoothing parameter.

For simplicity, we used the hyper parameter of the multinomial naive byes from 0.1 to 1, increment by 0.1.

	Alpha	K fold	Avg Acc
0	0.1	5	0.5105
1	0.2	5	0.4420
2	0.3	5	0.3800
3	0.4	5	0.3465
4	0.5	5	0.3160
5	0.6	5	0.2895
6	0.7	5	0.2700
7	0.8	5	0.2565
8	0.9	5	0.2430
9	1.0	5	0.2320

Accuracy on test dataset with the best smoothing parameter: 0.522

Figure 4: Naive Bytes for 20 news group dataset

	Alpha	K fold	Avg Acc
0	0.1	5	0.6990
1	0.2	5	0.6500
2	0.3	5	0.6085
3	0.4	5	0.5645
4	0.5	5	0.5455
5	0.6	5	0.5210
6	0.7	5	0.5040
7	0.8	5	0.4905
8	0.9	5	0.4800
9	1.0	5	0.4725

Accuracy on test dataset with the best smoothing parameter: 0.6396648044692738

Figure 5: Naive Bytes for Sentiment 140 sentiment dataset

For simplicity, we limited the range of the hyper parameter of the multinomial Naive Bayes as 10 floating point numbers ranging from 0.1 to 1, differing by 0.1. From the data listed, we found that the accuracy on the test dataset with the best smoothing parameter gives 0.522 for the 20 newsgroup dataset, and 0.6396 for the sentiment 140 dataset.

Additionally in order to choose the most appropriate hyperparameters for the softmax regression, we tried different parameters and compared the required solvers indicated in each of them. To avoid their restrictions, we eventually selected Penalty and C hyperparameters.
The penalty list has a default list of l1, l2, elasticnet and none. We set the solver to "liblinear" for our choice, therefore, to only get l1 and l2 in the list. Since C has a default value of 1 and has to be a floating point number, we selected the appropriate floating number from 1 to 1.6 for C values. In addition, the K fold value is set to 5.

line	Penalty	C_val	K fold	Avg Acc
10	l1	1.3	5	0.4890
9	l1	1.2	5	0.4865
11	l1	1.4	5	0.4850
12	l1	1.5	5	0.4845
8	l1	1.1	5	0.4845
7	l1	1.0	5	0.4840
13	l1	1.6	5	0.4835
0	l2	1.0	5	0.4760
1	l2	1.1	5	0.4760
4	l2	1.4	5	0.4755
3	l2	1.3	5	0.4750
2	l2	1.2	5	0.4745
5	l2	1.5	5	0.4745
6	l2	1.6	5	0.4730

Figure 6: Soft Max for 20 news group dataset

line	Penalty	C_val	K fold	Avg Acc
13	l1	1.6	5	0.6830
6	l2	1.6	5	0.6795
12	l1	1.5	5	0.6795
5	l2	1.5	5	0.6785
4	l2	1.4	5	0.6775
11	l1	1.4	5	0.6760
3	l2	1.3	5	0.6750
2	l2	1.2	5	0.6745
1	l2	1.1	5	0.6735
10	l1	1.3	5	0.6735
9	l1	1.2	5	0.6715
0	l2	1.0	5	0.6705
8	l1	1.1	5	0.6685
7	l1	1.0	5	0.6655

Figure 7: Soft Max for sentiment 140 dataset

We can see that the softmax regression gives the best average accuracy 0.4890 when penalty is l1 and C value is 1.3 for 20 the newsgroup datasets, and the best average accuracy of 0.6830 when penalty is l1 and C value is 1.6 for the sentiment 140 datasets.

We selected the hyperparameters giving the best accuracy for our respective final models. For the 20 newsgroup datasets, we set the logistic regression with penalty l1, C value equals to 1.3, solver "liblinear," respectively and max_iter equals 800. For the sentiment 140 datasets, we set the logistic regression with penalty l1, C value equals to 1.6, solver "liblinear," respectively and max_iter equals 800.

	Softmax_acc	Bayes_acc
0	0.617318	0.72905

Figure 8: Accuracy for 20 news group dataset

	Softmax_acc	Bayes_acc
0	0.689944	0.639665

Figure 9: Accuracy for sentiment 140 dataset

For the 20 news group datasets, the Naive Bayes model obtains an accuracy of 0.72905 higher than 0.617318 for the SoftMax model. However, for the sentiment 140 dataset, the SoftMax model receives an accuracy of 0.689944 higher than 0.639665 for the Naive Bayes model.

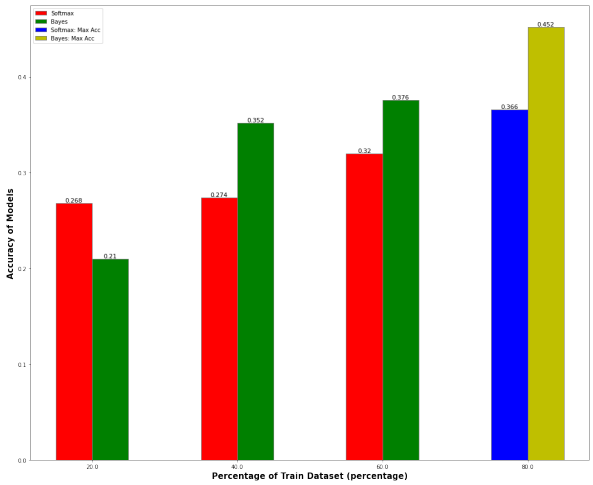

Figure 10: Accuracy for 20 news group dataset

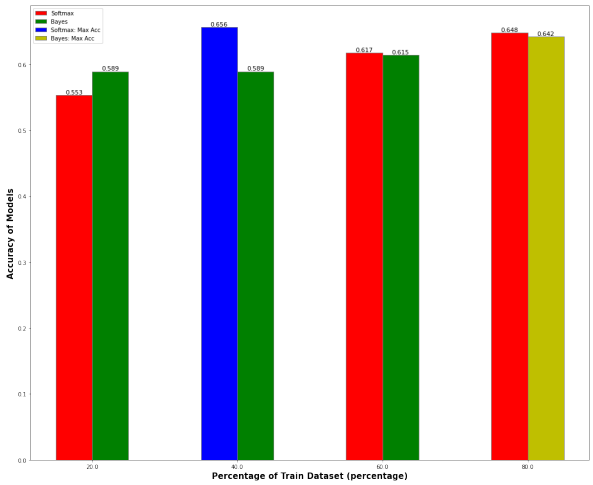

Figure 11: Accuracy for sentiment 140 dataset

To compare the accuracy of the two models as a function of the size of the dataset, we plotted a bar plot to illustrate a clearer view. We selected a fair portion of the enormous datasets and tested which model works more accurately for each. The accuracy for each portion of the dataset is highlighted in different colours, and the best accuracy of the portion is highlighted in different colours.

From the bar plot, we can tell that the Naive Bytes model gives the best accuracy with a larger portion of the data for both datasets. However, SoftMax usually produces a higher accuracy when selecting a smaller portion of datasets. Furthermore, the accuracy we got for the Sentiment140 dataset is very close, whether we have a different portion of the dataset or different models. For the 20 newsgroups datasets, the accuracy shows a huge difference between portion selection and models.
On the other hand, we compared the run time of each model by testing out the fit time and the average of 10 prediction times. SoftMax regression has a shorter average prediction time and a longer fit time than Naive Bytes for both datasets. However, SoftMax has a longer total run time in the sentiment 140 dataset.

	Test accuracy	Avg_Predict Time	Fit Time	Total Time
Softmax	0.614525	0.132500	3.225377	3.357877
Naive Bayes	0.729050	4.492618	0.655248	5.147866

Figure 12: Runtime for 20 news group dataset

	Test accuracy	Avg_Predict Time	Fit Time	Total Time
Softmax	0.689944	0.025048	0.561696	0.586744
Naive Bayes	0.639665	0.064723	0.171540	0.236264

Figure 13: Runtime for sentiment 140 dataset

4 Discussion and Conclusion

Comparing the accuracy of each model with their best hyperparameters we obtained from each data set shows that the SoftMax model receives better accuracy than Naive Bayes does in the Sentiment 140 data set, and the Naive Bayes model receives better accuracy than the SoftMax model does in 20 newsgroups data set.
For the 20 newsgroups dataset, Naive Bayes takes longer time to obtain the results, but its accuracy is better. On the other hand, SoftMax produces a higher accuracy with longer time in Sentiment140 datasets.
SoftMax model generally performs better than the Naive Bayes model on sentiment 140 data set when selecting appropriate portion of data. However, Naive Bayes works better in 20 news group data set. For different portions of Sentiment140 datasets, the selection of models won't affect the accuracy of the result, but the Naive Bayes model works better when handling most size of it. The different portions of the 20 newsgroups dataset estimate accuracy with a noticeable difference. The Naive Bayes model results a higher accuracy and the SoftMax model works more stable for this particular dataset.

5 Statement of Contributions

Yu Yun worked on the first task, acquired, prepossessed and analyzed the data. She also contributed to writing the abstract, intro and dataset part of the report.
Ruoyu has implemented the Multinomial Naive Bayes and K-cross validation functions from scratch. In addition, he helped Pengyu with some coding problems in task 3 as well.
Pengyu has designed and contributed on running experiments from task 3. He has also formatted the final version of the writeup into overleaf.