Comp 551 Mini Project 2

Javin Liu, Yuxuan Tian, Eric Chen

February, 27, 2021

1 Abstract

For this project, we investigated the performance between Naive Bayes and logistic regression. Logistic regression takes a longer amount of time to train than Naive Bayes. We found that using Tfidf increased the accuracy of logistic regression by a substantial amount. Logistic regression performs better on larger data sets than Naive Bayes but naive Bayes is very time efficient to run and easier to implement. Overall we found that the logistic regression is generally better. Picking the right data and data processing can change the accuracy and run time so it is very important for good results. As the data set gets bigger the accuracy generally improves.

2 Introduction

For this project we were required to implement the Naive Bayes algorithm and compare it to the logistic regression. We also implemented the cross validation code. We found that the best hyper parameter value for the Naive Bayes was 1. We found that the best hyper parameter for the logistic regression is C = 100, 'max_iter' = 100. Using Tfidf increased the accuracy of logistic regression by around 10 percent. Usually we have alpha and beta as the hyper pearameters for laplace smoothing, but for simplicity and convience to plot, we only used one hyperpermater "sigma", and defined the smoothing to be (numerator + sigma)/(denominator + number of classes *sigma).

Citations:

Cross Validation Sample TA Code
Naive Bayes Sample TA Code
Linear Regression Sample TA Code
Reference on processing data and naive Bayes training:
Tfidf and why it is useful

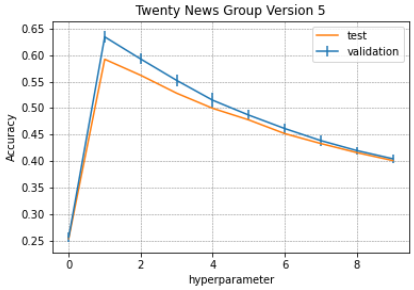


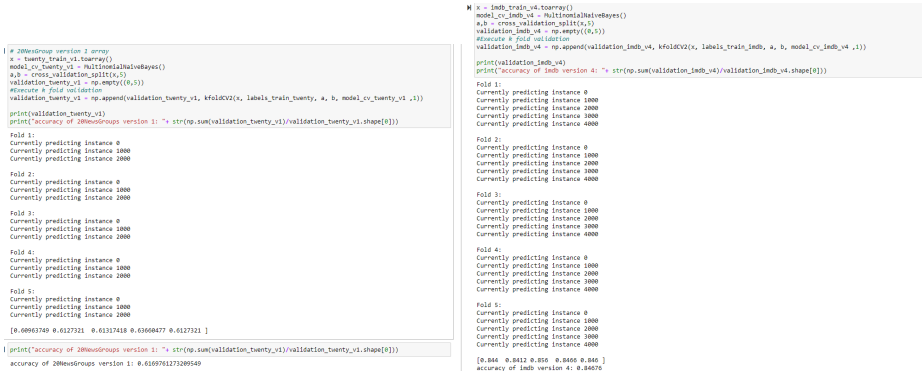
Figure 1: The effect on the Accuracy of naive Bayes when the hyper parameter is increased

3 Datasets

We processed the Datasets using the bags of words representation using the scikit-learn function: CountVectorizer. For the extra features, we implemented the data processing in five ways. Version 1 is the simplest with no conditions. For the version 2 we take care of only stop_words and meaningless combination of numbers and letters such as 32mn and 000. For the version 3, we take care of high frequency words. For the version 4, we take care of low frequency words. For the version 5, we removed stop_words, words with integers, and also got rid of low frequency words. We also implemented the Tfidf and it works by choosing proper features depending on the weights of words. We also experimented with the max_df value as inputs for the CountVectorizer() function and we found that max_df = 0.5 gave the best results. However, all max_df values causes only small changes on the features.

4 Results

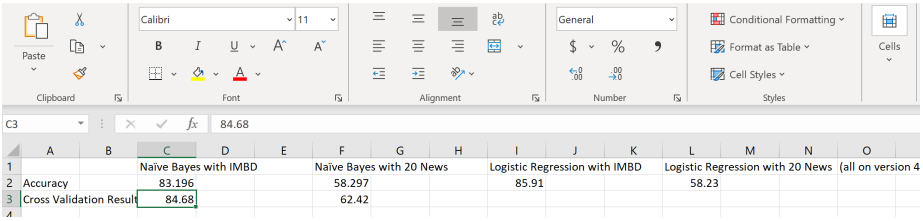
For part1 of Task 3 we found that the 20 News Group data set gave an accuracy of 61.70 percent and the IMBD gave an accuracy of 86.68 percent.



(a) 20 News data set (b) IMBD data set

Figure 2: Task 3. Part 1

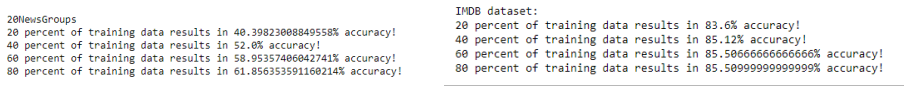
For part2 of Task 3 we found that in general logistic Regression performed dramatically better than Naive Bayes on smaller data sets. However, it performed worse than Naive Bayes on larger data sets.



	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
1			Naïve Bayes with IMBD			Naïve Bayes with 20 News			Logistic Regression with IMBD			Logistic Regression with 20 News	(all on version 4		
2	Accuracy		83.196			58.297			85.91			58.23			
3	Cross Validation Result		84.68			62.42									

Figure 3: Table of Task 3 Part 2

For part3 of Task 3 we found that as we increase the data set size, the accuracy increases for both data sets. However, we also see that the effect is more dramatic for the 20 News data set. This may be because the 20 News data set is dramatically smaller than the IMBD data set.



(a) 20 News (b) IMBD

Figure 4: Effect of size of data set

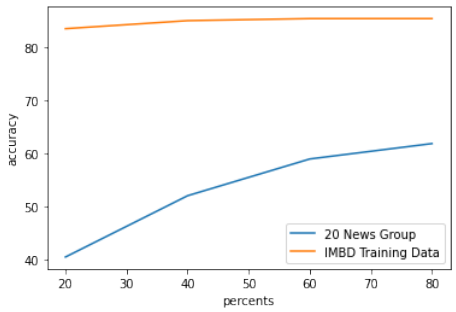


Figure 5: Diagram of the effect of training size on the two data sets

We found that out of all our different version, the logistic regression is generally better than the naive bayes. For experimentation and extra exploration, we implemented the tfidf. Using it on the IMDB data set version 3 with logistic regression gave us an accuracy of 97% , while the naive Bayes gave an accuracy of around 85.68% when we ran it on IMBD version 5. Without the tfidf, the logistic regression generally have a higher accuracy than the naive Bayes. Version 5 of the 20 news data set ran by the Naive Bayes gave the highest accuracy(63.44 percent) out of all the different versions. Version 5 of the Imdb also gave the best accuracy output.

```
imdb_train_LR_3 = LogisticRegression(random_state=0,max_iter=100).fit(imdb_train_
imdb_predict_3 = imdb_train_LR_3.predict(imdb_test_v3)
print(imdb_train_LR_3.score(imdb_test_v3,labels_test_imdb))
```

```
0.86356
```

```
[ ]  imdb_train_LR_tfidf_3_1 = LogisticRegression(random_state=0,max_iter=100,C=5.0).f
     imdb_predict_tfidf_3_1 = imdb_train_LR_tfidf_3_1.predict(imdb_test_tfidf_v3)
     print(imdb_train_LR_tfidf_3_1.score(imdb_test_tfidf_v3,labels_test_imdb))
```

```
0.97648
```

Figure 6: Tfidf helps increase Accuracy

As you see in the figure 6, using the Tfidf can have many positive benefits and using it helped to increase the accuracy by a significant amount.

5 Discussion and Conclusion

Data pre-processing is very important for a text data set. For example, running the Naive Bayes on imdb version 5 gave us an accuracy of 85.68 percent and running Naive Bayes on imdb version 5 gave us an accuracy of 84.68 percent. Because imdb has less data, the code runs faster while giving a higher accuracy.

So picking the most important data is very important. It allows us to have a faster run time because of the smaller amount of data while giving a potentially better accuracy because of more carefully selected data points. We found it a little weird that the logistic regression performed worse than the Naive bayes on the 20 News data set. The reason for this may be because logistic regression generally performs better than Naive Bayes on larger data sets but 20 News is a smaller data set.

6 Experimentation and extra exploration

For the extra exploration, we demonstrated our creativity by writing the Tfidf code. Tfidf generally gives the frequency of words and it also chooses features based on the weights. For example, words with a higher frequency will have a higher weight and words with a lower frequency will have a lower weight. We found that using tfidf improved the accuracy of logistic regression by around 5 to 10 percent which is a pretty significant increase. We also experimented with different ways to process the data (frequency of words, content of words, removing meaningless words, removing stop_words) and tested their effects on the accuracy. For logistic regression, we also did hyper parameter tuning when applying it on the IMDB dataset. We chose the best regularization strength and max_iter,being(5,100) out of a range of hyper parameters, and raised the prediction accuracy to 97%.

7 Statement of Contributions

Yuxuan Tian:1. Implemented the Multinomial Likelihood function. 2. Naive base fit and predict function. 3. Part 3 experiments 4. Helped with data processing. Eric Chen: 1. Data Reprocessing for both the 20 news group dataset and the IMDB Reviews dataset.2.Implemented the logistic regression code. Javin Liu: 1. Typed the latex write up. 2. Helped write the Naive base fit and predict function. 3.Implemented cross_validation_split and KfoldCV functions.