Machine Learning
CS-7333

Project 1: Naive Bayes

Instructor:
Prof. Sandip Sen

Author:
Selim Karaoglu

February 16, 2022

In this assignment; we implement two Naive Bayes event models, Bernoulli and Multinomial. Bernoulli and Multinomial Naive Bayes classifiers are trained and tested on different datasets. We present the accuracy scores for both Bernoulli and Multinomial Naive Bayes classifiers on different datasets and compare them to see which model performed better.

1 Introduction

This project focuses on implementing two different classifiers and make comparison between them. The classifiers employed in this project are both Naive Bayes models. We implement Bernoulli event model and Multonimial event model Naive Bayes classifiers for this project. The detailed implementation of both classifiers explained with detail in the Methods section. After implementing the Bernoulli and Multinomial Naive Bayes classifiers, we present the datasets we experimented on. Naive Bayes based classification models are based on some assumptions on the data, therefore the type of datasets we use would affect the accuracy of both models. Hence, we provide some information about the datasets in the following section to illuminate the behavior of classifiers.

2 Methods

In this section, we provide detailed information about our classifiers and datasets. First we present the classifiers by explaining the Naive Bayes assumption and further building Bernoulli and Multinomial models on top of it. Both Bernoulli and Multinomial Naive Bayes classifiers explained following the base Naive Bayes implementation. Following the explanation of the classifiers, we show the datasets utilized in this project.

2.1 Naive Bayes

Naive Bayes is a machine learning algorithm, more specifically, it is a classification method. This means that Naive Bayes is used when the output variable is discrete. The underlying methods of the algorithm are based on the Bayes Theorem that states:

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)}$$

To make it easier to understand, this equation can be written using the features (attributes, input values) and the targets (output values). In simpler terms; this function can be employed to solve for the probability of y with given features X as:

$$P(y|X) = \frac{P(X|y)P(y)}{P(X)}$$

The Naive assumption takes place here on the formula. Naive Bayes method works on cases where different feature values are independent from each other. Therefore the formula can be rewritten as:

$$P(X|y) = P(X_1|y)P(X_2|y)...P(X_n|y)$$

It is clear that the goal is to solve for y, in this circumstance P(X) is a constant value and it can be removed from the equation. In addition to that, by implementing the proportional nature of the $X_1, X_2, ..., X_n$ the equation can be presented as:

$$P(y|X) = P(y)\prod_{i=1}^{n} P(X_i|y)$$

Since the values across the datapoints can be calculated, the last step should be the finding the maximum of these calculated output values of y. The maximum can be obtained with the $\arg\max$ function:

$$y = \arg\max_{y}[P(y)\prod_{i=1}^{n} P(X_i|y)]$$

With taking consideration on provided information about Naive Bayes, the classification method can be explained as; first, create a frequency table and then a ratio table so that the values for P(X) and P(y|X) can be calculated, furthermore, for a given set of input features X, compute the proportionality of P(y|X) for each class y. In this example, the implementation should be binary or simply True or False output. Now as the last important part of the Naive Bayes method, there could be different assumptions when calculating $P(X|y)$ like Multinomial model, Bernoulli model and Gaussian model. This implementation focuses on Bernoulli and Multinomial models, each model are explained in detail in following subsection. Since both Bernoulli and Multinomial methods are based on Naive Bayes assumption, the code implementation structured accordingly. We presented a class for Naive Bayes base algorithm that both Bernoulli and Multinomial classes refer to. Naive Bayes base algorithm implementation for this work follows this algorithm:

Algorithm 1 Naive Bayes Base Algorithm

```
procedure NaiveBayesBase
    function predict
        returns the class predictions for the given joint log likelihood results.
    function _log_proba
        calculates log probabilities of features with log-sum-exp method.
    function fit
        fits the Naive Bayes classifier according to features and targets
    function score
        returns accuracy score between predicted and true classes.
```

2.2 Bernoulli

Bernoulli assumption takes its name from famous mathematician Jacob Bernoulli. Bernoulli distribution is a specific case of Naive Bayes assumption where event X results in a binary outcome. This can be interpreted as event X can only be resulted as 0 and 1. Coin toss can be the perfect example for this type of outcome, since it only can be resulted with heads or tails. Any event that has a binary outcome can be classified with Bernoulli Naive Bayes classifier and when an event follows Bernoulli distribution, it is called a Bernoulli trial. In Bernoulli model; following the Naive Bayes assumption presented previously, a value p is attributed to the probability of an outcome sums to 1. Therefore, the probability of not having this outcome is equal to $1 - p$. This can be interpreted as:

$$Ber(X|p) = \begin{cases} p & \text{if X=1} \\ 1-p & \text{if X=0} \end{cases}$$

However, Bernoulli model is tailored for binary outcomes, therefore it is not able to use on multi-class classification problems. There is a way to modify the Bernoulli Naive Bayes classifier to solve multi-class classification tasks, although, that is called Multinoulli and it is out of context for our assignment. The Bernoulli Naive Bayes classifier implementation in this experiment only focuses on binary classification (as it should be). Bernoulli Naive Bayes classifier algorithm implementation for this work follows this algorithm:

Algorithm 2 Bernoulli Naive Bayes Classifier Algorithm

```
procedure BernoulliNB(NaiveBayesBase)
    function _count
        returns the feature and class counts of the dataset.
    function _update_f_log_prob
        applies Laplace (alpha) smoothing for feature and class counts and updates log-probabilities.
    function _joint_log_likelihood
        takes the feature and class log-probabilities and returns joint log-likelihood.
```

2.3 Multinomial

Multinomial Naive Bayes is based on the assumption that the data is distributed with multinomial distribution. Multinomial distribution is a generalization of binomial distribution, but instead of binary outcomes, multinomial distribution is able to represent multi-class structures. In machine learning literature, Multinomial Naive Bayes mainly employed for Natural Language Processing (NLP) tasks since it can model the counts of words in given data. This is mainly because Multinomial model is designed for multi-class outputs and can calculate probabilities for a whole dictionary. Let's assume that a sample X may result with k possible outcomes (reminder, it was 2 in Bernoulli) with each outcome has the probability p_k after n trials can be formulated as:

$$p(X = k) = \frac{n!}{x_1!x_2!...x_k!} p_1^{x_1} p_2^{x_2} ... p_k^{x_k}$$

In this formula, n represents the number of trials (number of entries for our case), x_i is the number of event i occurs and p_i is equal to the probability of observing event i at each single trial. Multinomial Naive Bayes classifier algorithm implementation for this work follows this algorithm:

Algorithm 3 Multinomial Naive Bayes Classifier Algorithm

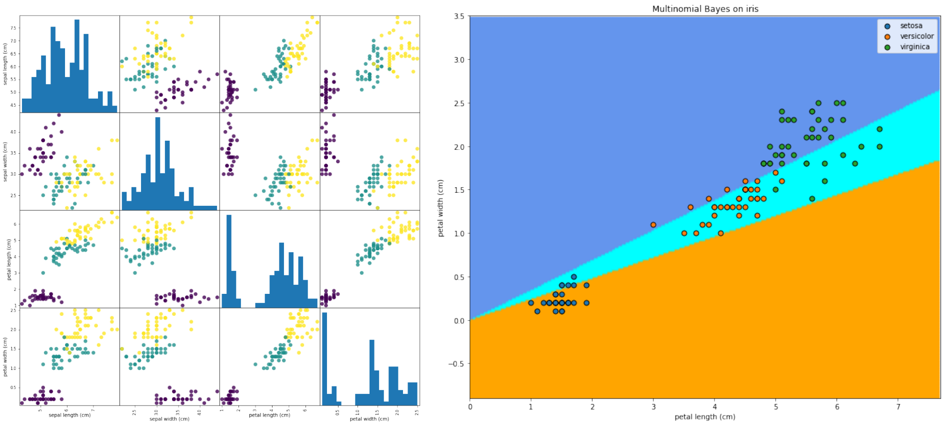
```
procedure MULTINOMIALNB(NAIVEBAYESBASE)
    function _COUNT
        returns the feature and class counts of the dataset.
    function _UPDATE_F_LOG_PROB
        applies Laplace (alpha) smoothing for feature counts and updates log-probabilities for both features and classes.
    function _JOINT_LOG_LIKELIHOOD
        takes the feature and class log-probabilities and returns joint log-likelihood.
```

2.4 Datasets

In this part we present the datasets employed in this assignment. Each dataset structure is explained with their features, targets, distributions, number of entries etc. Since the classifiers we utilized in this project are based on some assumptions about the dataset, examining datasets in-depth can give us clues about the results and why the results make sense.

2.4.1 Iris Dataset

Iris dataset first appeared in Fisher's paper in 1936[1]. This project uses the iris dataset provided by sklearn library[1]. This dataset contains 150 instances with 3 classes, 50 instance for each class. Each instance in iris dataset has 1 class and 4 numeric attributes; sepal length, sepal width, petal length and petal width all in cm. 3 different class outputs are labeled as; Setosa, Versicolour and Virginica. There are no missing attributes in this dataset. Figure 1.a shows the distribution plot for each feature and scatter plots for each two feature combination. This subfigure shows that if petal length and petal width are taken into consideration, the class distribution is easier to seperate visually. With the advantage of this visualization; on Figure 1.b, we present Multinomial Naive Bayes classification plot for only two features; petal length and petal width.



(a) Iris Dataset Attribute Dist.

(b) Multinomial Naive Bayes boundaries

Figure 1: Iris Dataset Attributes

2.4.2 SMS Spam Collection v.1 Dataset

SMS Spam collection dataset is introduced by Almeida's paper in 2011[2]. This dataset can be found on UCI Machine Learning Repository[2] and Kaggle[3]. According to the providers; "the SMS Spam Collection is a set of SMS tagged messages that have been collected for SMS Spam research. It contains one set of SMS messages in English of 5,574 messages, tagged according being ham (legitimate) or spam". This dataset contains 5572 instances and 2 class outputs. The feature that contains the raw text messages in the dataset contains 5169 unique values. %87 of the instances are labeled as "ham" while the remaining %13 are labeled as "spam".

[1] https://scikit-learn.org/stable/modules/generated/sklearn.datasets.load_iris.html

[2] https://archive.ics.uci.edu/ml/datasets/SMS+Spam+Collection

[3] https://www.kaggle.com/datasets/uciml/sms-spam-collection-dataset

2.4.3 20 Newsgroups Text Dataset

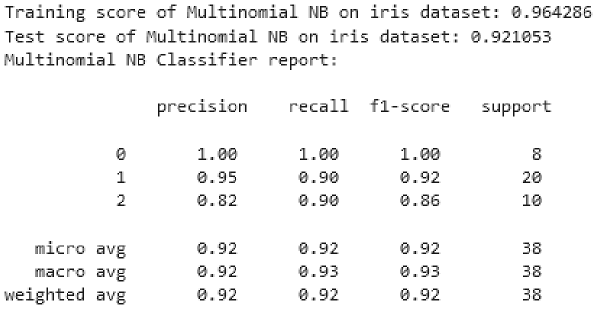
20 newsgroups text dataset is provided by sklearn[4]. The providers note; "The 20 newsgroups dataset comprises around 18000 newsgroups posts on 20 topics split in two subsets: one for training (or development) and the other one for testing (or for performance evaluation). The split between the train and test set is based upon a messages posted before and after a specific date". This dataset is built with 18846 samples with 20 classes that represents 20 news categories. Since each instance is an article, we utilized term frequency–inverse document frequency (TF-IDF) vectorizer in order to be able to train the classifiers.

3 Experiment

In this section, we share the experiment. By this point of the assignment, we defined two Naive Bayes classifiers; Bernoulli and Multinomial models, in addition to that, we presented the datasets we utilized in this work. Now we present the experiment results for each dataset in detail.

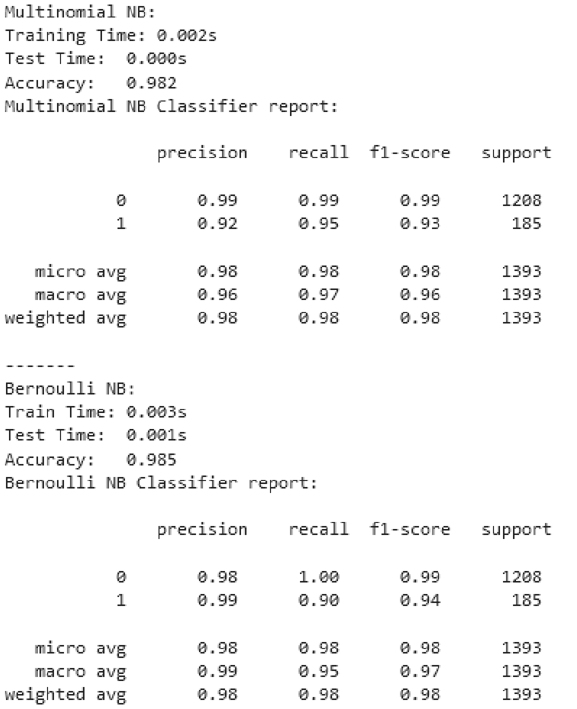
Iris Dataset is a multi-class output dataset. Since Bernoulli Naive Bayes classifier is not designed to classify non-binary outputs, we only trained Multinomial Naive Bayes classifier on Iris dataset (the reason for this explained further). Figure 2 shows the classification report for Multinomial Naive Bayes classifier on Iris dataset. Training and test split is applied with %25 test size. Our classifier achieved %92 weighted average accuracy on instantaneous training times as 0.003 seconds.

In SMS Spam dataset, there are only 2 class values; "ham" or "spam". This is a binary output and both Bernoulli and Multinomial Naive Bayes classifiers can be employed with this data. However, the main difference between the two models is the number of class values, the distributions of both models are still almost the same. The only difference (can be seen on code) is the application of Laplace smoothing; in Multinomial model, Laplace smoothing is only applied to features but in Bernoulli model, Laplace smoothing is applied for both features and classes. The experiment results clarify how much this difference would affect the accuracy scores. Training and test split is applied with %25 test size. Figure 3 shows the classification report on both Bernoulli and Multinomial Naive Bayes classifiers. The results are almost the same, only difference is on the macro average scores; Bernoulli model scored slightly higher on macro average precision and f1-scores and Multinomial model scored slightly higher on macro average recall score. Both models have almost perfect accuracy up to %98. That slight score differences can only



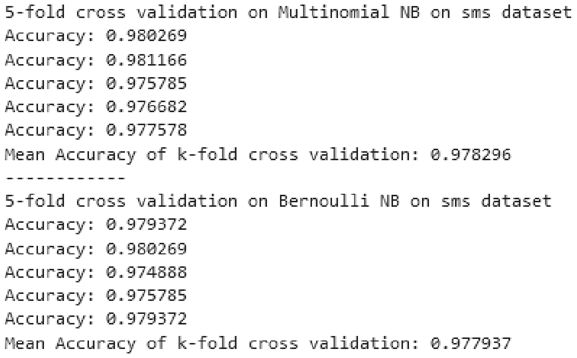
```
Training score of Multinomial NB on iris dataset: 0.964286
Test score of Multinomial NB on iris dataset: 0.921053
Multinomial NB Classifier report:

              precision    recall  f1-score   support

           0       1.00      1.00      1.00         8
           1       0.95      0.90      0.92        20
           2       0.82      0.90      0.86        10

   micro avg       0.92      0.92      0.92        38
   macro avg       0.92      0.93      0.93        38
weighted avg       0.92      0.92      0.92        38
```

Figure 2: MultinomialNB classification report on Iris dataset



```
Multinomial NB:
Training Time: 0.002s
Test Time:  0.000s
Accuracy:   0.982
Multinomial NB Classifier report:

              precision    recall  f1-score   support

           0       0.99      0.99      0.99      1208
           1       0.92      0.95      0.93       185

   micro avg       0.98      0.98      0.98      1393
   macro avg       0.96      0.97      0.96      1393
weighted avg       0.98      0.98      0.98      1393

-------
Bernoulli NB:
Train Time: 0.003s
Test Time:  0.001s
Accuracy:   0.985
Bernoulli NB Classifier report:

              precision    recall  f1-score   support

           0       0.98      1.00      0.99      1208
           1       0.99      0.90      0.94       185

   micro avg       0.98      0.98      0.98      1393
   macro avg       0.99      0.95      0.97      1393
weighted avg       0.98      0.98      0.98      1393
```

Figure 3: MultinomialNB & BernoulliNB classification report on SMS dataset



```
5-fold cross validation on Multinomial NB on sms dataset
Accuracy: 0.980269
Accuracy: 0.981166
Accuracy: 0.975785
Accuracy: 0.976682
Accuracy: 0.977578
Mean Accuracy of k-fold cross validation: 0.978296
------------
5-fold cross validation on Bernoulli NB on sms dataset
Accuracy: 0.979372
Accuracy: 0.980269
Accuracy: 0.974888
Accuracy: 0.975785
Accuracy: 0.979372
Mean Accuracy of k-fold cross validation: 0.977937
```

Figure 4: MultinomialNB & BernoulliNB 5-fold cross validation on SMS dataset

[4] https://scikit-learn.org/stable/modules/generated/sklearn.datasets.fetch_20newsgroups.html

be caused by the different Laplace smoothing applications for the classifiers. We present the k-fold cross validation results for both models on SMS Spam dataset on Figure 4 (k=5). Both models classified with almost perfect accuracy scores, the Multinomial model scored %0.000359 higher than Bernoulli model.

20 news groups dataset is a large dataset that contains more than 18800 samples. These samples are divided to 20 different topics, therefore the number of classes for this dataset is 20. This is a perfect dataset for Multinomial Naive Bayes classifier, however Bernoulli Naive Bayes classifier is not designed to classify on datasets with multi-class outputs. Without concerning about the number of classes, we trained and tested both Bernoulli and Multinomial models on this dataset. Figure 5 shows the confusion matrix created on class values of 20 news groups dataset. This figure represents a comparison between the predicted values from Multinomial Naive Bayes classifier and data labels. According to the confusion matrix from Figure 5, the most confused topics are "Social:Religion:Christian" and "Talk:Religion:Miscellaneous", which are easy to confuse since they share a lot of similar concepts inside. Sharing the same number of confusions, "Talk:Politic:Guns" and "Talk:Politics:Misc" are classes also caused some confusion for our classifier owing to the fact that topics and the contents are similar to each other. As expected, Bernoulli Naive Bayes classifier resulted with very low average scores with weighted average precision being %54. Although, when the classifier report is examined carefully, it can be spotted that there is one class predicted with very high score (class 11 for our case). Since Bernoulli Naive Bayes classifier is employed for datasets with binary classes, it still is able to classify 1 class with high accuracy scores. For class with output value 11, Bernoulli Naive Bayes achieved %93 precision score, %77 recall score and %84 f1-score. Other results are significantly inaccurate. Multinomial Naive Bayes classifier performs with a %82 weighted average score. Both models took approximately 2.8 seconds to train and 1.5 seconds to test. Figure 6 shows the training time, test time and accuracy scores for Bernoulli and Multinomial Naive Bayes classifiers on 20 newsgroups dataset.

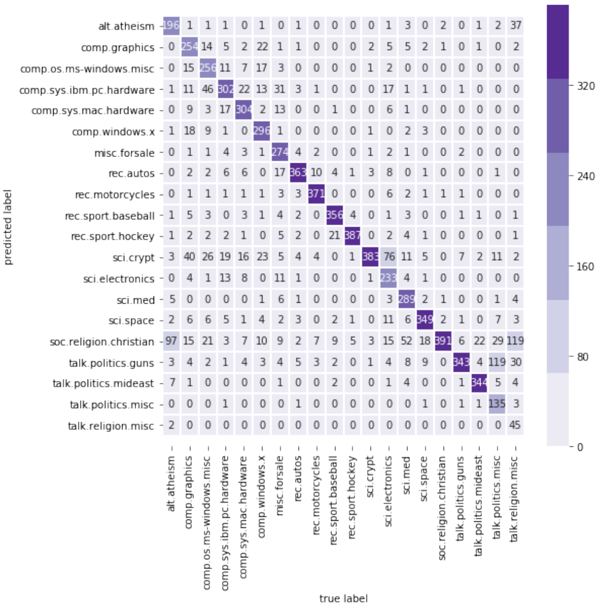


Figure 5: MultinomialNB classification report on 20 Newsgroups dataset

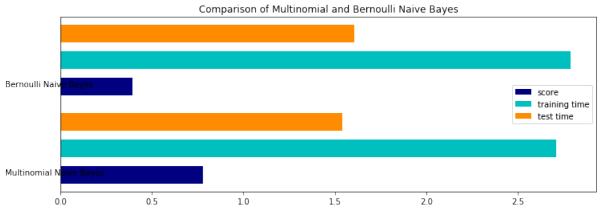


Figure 6: MultinomialNB & BernoulliNB time & accuracy on 20 Newsgroups dataset

4 Conclusion

In this work, we presented two Naive Bayes classifiers; Bernoulli and Multinomial models. Bernoulli model is designed to work on binary labels and Multinomial model is designed to work on multi-class labels. After we presented the implementation of both classifiers, we provide basic information about the datasets. Since the structure of the dataset affects the accuracy of the classifier, the information about the dataset can illuminate some questions about the results. The experiment part provides in-depth information about how the experiment is conducted and how it resulted. Iris dataset is only classified with Multinomial model, and the model achieved very high scores. SMS Spam dataset is used for both Bernoulli and Multinomial models and both achieved almost perfect accuracy scores. 20 news groups dataset is built with 20 different class labels, hence it's not a good fit for Bernoulli model. We experimented the Bernoulli model on 20 news groups dataset and observed that 1 of the 20 classes (11th class) is classified with very high accuracy scores (comparing to other classes). This suggest that Bernoulli model classified the dataset as 11 and non-11 classes and therefore achieved high accuracy scores on 11th class. Multinomial model achieved %82 weighted average accuracy score for 20 news groups dataset. This experiment suggests that, one of the most important aspects of the classification problems is to employ the correct classifier. Both models are based on Naive Bayes assumption, and has almost the same calculation; the main difference between them is the number of classes they can classify.

References

[1] Fisher, R.A. “The use of multiple measurements in taxonomic problems” Annual Eugenics, 7, Part II, 179-188 (1936); also in “Contributions to Mathematical Statistics” (John Wiley, NY, 1950).

[2] Almeida, T.A., et al. "Contributions to the Study of SMS Spam Filtering: New Collection and Results." Proceedings of the 2011 ACM Symposium on Document Engineering (DOCENG'11), Mountain View, CA, USA, 2011.