# Naive Bayes With Data Noise

**Michael Downs** MICHAELD20312@GMAIL.COM
*Division of Computer Science*
*Montana State University*
*Bozeman, MT 59715, USA*

**Max Hymer** HYMERMAX1@GMAIL.COM
*Department of Electrical and Computer Engineering*
*Montana State University*
*Bozeman, MT 59715, USA*

## Abstract

In our paper, we explore the idea of Naive Bayes, which is a classification algorithm that is used in machine learning. We tested this algorithm's accuracy under conditions with slightly inaccurate data to help interpret mistakes in data that could be present. To evaluate our algorithm, we used 0/1 Loss and an $F_1$ score. We implemented a 10-fold cross-validation method to randomize our data and negate patterns present. We found that the Naive Bayes algorithm performs best with large, categorical, even-distribution data sets. We found that it performs worst with small, continuous, skewed-distribution data sets. Large data sets gave the algorithm more data to train with which allowed for more accurate results. Data sets that require binning (continuous data sets) are inherently less representative of the than data than data sets that are already categorical. Even-distribution data sets give the algorithm enough data in each category for it to accurately predict correct classes while skewed-distribution data sets do not.

**Keywords:** Cross Validation, Classification, Algorithm

## 1 Introduction

This is an important problem to tackle and attempt to improve due to the wide use of classification. From face identification that unlocks your phone to showing you media in your feed that you are likely to consume, it is all around us and an important field to understand. One key problem that is always in question is the accuracy of the data, which is why we are doing our project based on this issue. We sought to find out how noise or, in other words, how slightly inaccurate data as well as how different types of data and sizes of data sets affected such a widely used algorithm. We expected that large data sets (breast cancer) would have better results compared to smaller data sets (soybeans) which would have poor results due to the small amount of data, and continuous data sets would have poor results if the data was skewed in a way that would make it difficult to evenly bin. In conclusion, we hypothesized that the breast cancer data would be most accurate, then voting records, then iris plants, then glass, then soybeans would be least accurate. We think this because the breast cancer data set was the largest and was both categorical and evenly distributed. The voting records data set was also fairly large, categorical, and evenly distributed. The iris plant data set was not as large but it was very evenly distributed after we put it through our binning process. The glass data set, although large, had a very skewed distribution even after our binning process so we do not expect it to perform particularly well. The soybean data set is the smallest by far, so we do not expect that it has enough data to both train and test the classifier with any degree of accuracy.

## 2 Experimental Design

In this section, we will review our program's driver designs, loss functions, classification algorithm, and most importantly, why we chose to approach our project in a particular manner.

### 2.1 Preprocessing

We designed quite a few drivers that would be specifically made to interpret the five different datasets. This was important because some of the data sets contained character values, integers, doubles, and unknown values. Although all of the data was stored in comma-separated values (CSV), we still decided to make separate drivers for each of the datasets. We designed our code this way because we wanted to be able to have a uniform process for our classification process. All of our drivers process data into arrays of type Object. This way, the data can be processed regardless of its type. Four of the drivers (two of the data sets) perform some type of data imputation. The breast cancer drivers replace instances of "?" in the data with a random integer with a value 1-10 as that is the range of the data and we wanted to keep our data imputation approach simple. The iris drivers get rid of an empty line at the end of the data set so that a null line is not read in to the data and label arrays. Four of the drivers also perform a process called binning. Since the Naive Bayes algorithm works based on the likelihood of categories, continuous data sets cannot be fed into the algorithm. So, for the continuous data sets (iris and glass), we place the continuous values into categories that fit the data. Through this binning method, we can evaluate continuous data sets with a reasonable degree of accuracy with Naive Bayes. The votes drivers don't require data imputation or binning, so data is simply read into the label and data arrays; no complex preprocessing is required.

### 2.2 Loss Functions

0/1 loss function

This algorithm takes all of the guesses and records them as a 0 when they are incorrect and a 1 when they are correct. Then once all of the predictions are made the total incorrect guesses is divided by the total number of guesses. This gives us a percentage of incorrect guesses that will quickly tell us how accurate the algorithm is in predicting classes based on the feature information and more precisely the percentage of estimating the class incorrectly.

$$L(G) = \frac{\sum I}{T} * 100$$

where L(G) is our loss percentage (Incorrect Guessing percentage), G is our Guess, T is the total number of guesses, and $\sum I$ is our total number of incorrect guesses.

$f_1$

This algorithm is 2 multiplied by the product of precision and recall, then divided by the sum of precision and recall. It is particularly useful in situations where there is a class imbalance. The F1 score is defined as the harmonic mean of precision and recall:

$$\text{F1 Score} = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}$$

$$\text{Recall} = \frac{TP}{TP + FN}$$

$$\text{Precision} = \frac{TP}{TP + FP}$$

where TP is true positive, FP is false positive, and FN is false negative.

### 2.3 Algorithm

Multinomial Naive Bayes is the algorithm that we use for classification tasks. This includes prior probability where $N_{C_i}$ is the number of instances of a feature value in class $C_i$ and $N$ is the total number of feature value instances:

$$P(C_i) = \frac{N_{C_i}}{N}$$

For each feature $x_f$, calculate the conditional probability of feature $x_f$ in terms of class $C_i$ where $C_i$ is going to represent each class because we need to find this probability for all of the classes.

$$P(x_f|C_i) = \frac{N_{x_f,C_i} + \alpha}{N_{C_i} + \alpha \cdot V}$$

where $N_{x_j,C_i}$ is the count of feature $x_j$ in documents of class $C_i$, $N_{C_i}$ is the total count of features in class $C_i$, $\alpha$ is a smoothing parameter that we set to 1, and $V$ is the number of unique features. Setting our $\alpha$ value to one is important because if an instance of a feature that is connected to a class is present in testing that was not present in training, the likelihood will always be zero. By putting a one there we will never get a likelihood of 0 for any of our classes, which we hoped to help negate the noise that is added to our datasets.

$$P(C_i|\mathbf{x}) \propto P(C_i) \prod_{j=1}^{n} P(x_j|C_i)$$

Next, We take all of the feature likelihoods and multiply them together. Then the predicted class $y$ is the class that returns the largest number which is represented by the argmax statement.

$$y = \arg\max_{C_i} \left( P(C_i) \prod_{j=1}^{n} P(x_j|C_i) \right)$$

### 2.4 Experimental process

Each data set was ran through 10-fold cross validation five times. The 10-fold cross validation ran 10 times and used each of the 10 folds as the test fold (with the remaining 9 as training folds). Then, it averaged the 0/1 loss and F1 scores of all 10 test folds. So, our averages represented in the tables below are averages of 50 folds of testing (5 iterations of testing and averaging 10 folds). Because of this testing strategy, our data should accurately represent our classifier's real potential as the testing is performed so thoroughly.

## 3 Results

This section will review all of the results we found for each dataset.

### 3.1 Breast Cancer Identification

This database from Wisconsin Breast Cancer (January 8, 1991) shows data related to tumors found in breasts. Each tumor instance has 9 feature values that are integers on a scale from one to ten and a feature that tells us whether the tumor is malignant or benign through a specified integer value. This was one of our larger datasets, as there are about 700 instances of tumors, with about 450 cases being benign and about 250 cases being malignant. Plus, there are a few instances where the data is missing some attribute within a feature.

| **Breast Cancer** | | | **Breast Cancer Noise** | |
|---|---|---|---|---|
| **Iteration** | **0/1 Loss** | **$F_1$ Malignant (4)** | **0/1 Loss** | **$F_1$ Malignant (4)** |
| 1 | 0.0275 | 0.9595 | 0.0290 | 0.9592 |
| 2 | 0.0261 | 0.9629 | 0.0290 | 0.9585 |
| 3 | 0.0275 | 0.9626 | 0.0319 | 0.9542 |
| 4 | 0.0261 | 0.9646 | 0.0304 | 0.9570 |
| 5 | 0.0275 | 0.9607 | 0.0304 | 0.9542 |
| **Avg.** | **0.0263** | **0.9621** | **0.0301** | **0.9566** |

Table 1: Figure 3.1

### 3.2 Congressional Voting Identification

The 1984 United States Congressional Voting Records Database records voting decisions based on issues that differentiate Democrats from Republicans. This data shows 435 people on record, 267 of whom are Democrats and 168 of whom are Republicans. This data also involves more information than just voting yes, no, or ”?”. Yes involved voted for, paired for and announced for. No involved voted against, paired against, and announced against. Then unknown involved congress who voted present, voted present to avoid conflict of interest, and did not vote or otherwise make a position known.

| **Voting** | | | **Voting Noise** | |
|---|---|---|---|---|
| **Iteration** | **0/1 Loss** | **$F_1$ Republican** | **0/1 Loss** | **$F_1$ Republican** |
| 1 | 0.0953 | 0.8778 | 0.0977 | 0.8697 |
| 2 | 0.1023 | 0.8670 | 0.0953 | 0.8717 |
| 3 | 0.1023 | 0.8703 | 0.1000 | 0.8694 |
| 4 | 0.1000 | 0.8768 | 0.1000 | 0.8692 |
| 5 | 0.0953 | 0.8752 | 0.0977 | 0.8798 |
| **Avg.** | **0.0990** | **0.8734** | **0.0981** | **0.8720** |

Table 2: Figure 3.2

### 3.3 Soybean Identification

The Small Soybean Database (1987) contains information about diseased soybeans. The dataset has only 47 instances, with 35 features represented by integers. Four possible classes are represented by D4 (17 cases), D3 (10 cases), D2 (10 cases), and D1 (10 cases). It is also important to note that we have no $F_1$ score for this data because not all of our folds contained all classes and because of this we were not able to calculate precision and recall.

| **Soybean** | | |
|---|---|---|
| **Iteration** | **0/1 Loss** | **0/1 w/ Noise** |
| 1 | 0.0250 | 0.2750 |
| 2 | 0.1750 | 0.2750 |
| 3 | 0.1750 | 0.3250 |
| 4 | 0.4000 | 0.1000 |
| 5 | 0.2000 | 0.2750 |
| **Avg.** | **0.1950** | **0.2500** |

Table 3: Figure 3.3

### 3.4 Iris Plant Identification

The Iris Plants Database (1988) contained information about 3 different iris plants, each class having 50 instances. Features involved measuring the plants' sepals and petals. This database contained no missing values and was represented with decimal value measurements.

| **Iris** | | | **Iris Noise** | |
|---|---|---|---|---|
| **Iteration** | **0/1 Loss** | **$F_1$ Virginica** | **0/1 Loss** | **$F_1$ Virginica** |
| 1 | 0.0733 | 0.8859 | 0.0533 | 0.9056 |
| 2 | 0.0733 | 0.8661 | 0.0533 | 0.9254 |
| 3 | 0.0733 | 0.8686 | 0.0533 | 0.9154 |
| 4 | 0.0733 | 0.8766 | 1.399 | 0.7454 |
| 5 | 0.0733 | 0.8945 | 0.6600 | 0.9240 |
| **Avg.** | **0.0733** | **0.8783** | **0.0720** | **0.8832** |

Table 4: Figure 3.4

### 3.5 Glass Identification

The Glass Identification Database (1987), used in forensic science, contained information about elements present in glass and its refractive index. Based on the feature data, the class specified what type of glass was present. This database had no missing attribute values, with 214 total instances.

| | **Glass** | | **Glass Noise** | |
|---|---|---|---|---|
| **Iteration** | **0/1 Loss** | **$F_1$ class 1** | **0/1 Loss** | **$F_1$ Class 1** |
| 1 | 0.3238 | 0.7323 | 0.3381 | 0.7013 |
| 2 | 0.3333 | 0.7240 | 0.3381 | 0.7125 |
| 3 | 0.3095 | 0.7481 | 0.3190 | 0.7270 |
| 4 | 0.3048 | 0.7263 | 0.3190 | 0.7374 |
| 5 | 0.3286 | 0.7132 | 0.3333 | 0.7001 |
| **Avg.** | **0.3200** | **0.7288** | **0.3295** | **0.7157** |

Table 5: Figure 3.5

## 4 Conclusions

Overall, our algorithm was very successful relative to what we were hoping for. It was interesting to find that the data containing noise was not extremely harmful compared to the raw data. In a couple of cases, the noise even improved the accuracy on average over a number of times running. This is likely because some data that is very close to being interpreted between classes, is randomly modified to make it contain data that is more similar to the instances class. In terms of our hypothesis, it was relatively accurate in terms of average accuracy other than the glass identification. The Naive Bayes algorithm did perform best (measured by 0/1 loss) on the breast cancer data set. We believe this is because the data set was essentially ideal for this algorithm; it was categorical with evenly distributed data and was a very large data set. It performed well on the voting and iris data sets as well for similar reasons; they were both fairly large data sets with evenly distributed data (voting with categorical data, iris with evenly distributed binning). We think that the glass data set performed so poorly because the data was so skewed; it was almost impossible to bin in a way that filled bins even close to evenly. The soybean data set did perform poorly just like we expected it to because it was such a small data set. In terms of our limitations within this experiment, it is important to note that more data would likely be more helpful. As reflecting on our hypothesis seemed to be true to an extent as accuracy loosely was aligned with the amount of data present. One clear limitation of our algorithm that could take away from the accuracy of classification and the accuracy of our hypothesis was the fact that some data was continuous. Continuous data, in this case, was binned, and it was difficult to tell what binning method would be the most successful for each set of data and patterns that are present within the classes. For this reason, we believe that the glass accuracy was lower than expected.

## 5 Appendix

Max Hymer:
- Paper (loss functions, algorithm, results, conclusions)
- Code (10-fold cross validation, binning, Naive Bayes debugging)
- 50% effort

Michael Downs:
- Paper (abstract, introduction, preprocessing, experimental process)
- Code (Naive Bayes implementation, preprocessing, noise implementation, loss functions)
- 50% effort

Combined Work:
- Video commentary and recording
- Did all work together, there was collaboration between most parts