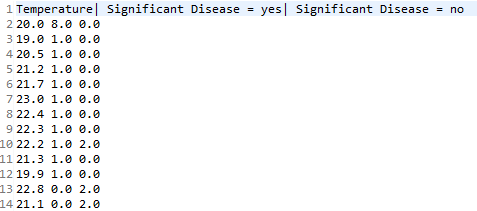
|  |
| --- |
| CSCI 330  Coffee Rust Prediction Using the Naïve Bayes Algorithm  Rebecca Crust, Michael Garcia, and Jessalene Ea  Abstract  In tropical countries coffee is a major economic and cultural crop. It is often threatened by the coffee rust fungus, *Hemileia vastatrix*, which can decrease the production of coffee beans in severe infections. There are preventative measures such as fungicide available but they are economically expensive. If severe outbreaks of rust could be predicted, then fungicide use could be minimized. Our goal was to use machine learning algorithms such as Naïve Bayes to accurately predict rust severity based on temperature, humidity, and precipitation.  We collected data by compiling weather data that corresponded with the times and places that coffee rust epidemic and non-epidemic seasons occurred in countries that grow coffee as a major crop. Our first model used the discrete Naïve Bayes algorithm and resulted in an accuracy of 68.8%. Our second model used a Gaussian Naïve Bayes algorithm and attained the same results. Compared to models such as the Gaussian process used by Weka, our models do not reach the same performance level, and so there is room for improvement. |

# Introduction

Modeling plant disease using machine learning techniques has become more popular recently among scientists who study plant diseases, known as plant pathologists. Management of diseases in crop systems is economically important across the globe. One of the diseases that has been drawing attention in the world is coffee rust. Coffee rust is caused by the fungus *Hemileia vastatrix,* which infects the leaves of *Arabica* and *Robusta* coffee cultivars and decreases bean production in severe outbreaks. Plant pathologists know that the ideal conditions for rust symptoms are temperatures around 22° C, high humidity, and prolonged moisture or rain. Preventative measures are known and include fungicide application prior to infection, planting multiple cultivars in a field, and planting cultivars that have genetic resistance to rust (Amerson, 2000). Management strategies such as fungicide are effective but expensive, and so being able to predict rust severity could decrease the amount of fungicide that coffee growers would need to use.

To address this problem, we compiled a dataset of weather conditions in countries where coffee is a major crop, at the times and places where epidemic and non-epidemic rust seasons occurred (Sainato, 2015). We decided that an outbreak was considered severe if it reached an economic threshold, meaning that the coffee growers lost money that growing season overall. We used Naïve Bayes algorithms because of our multiclass data set and the known trends in rust symptoms based on those classes.

**Figure 1**: An example model for the Discrete Naïve Bayes algorithm

# Methods

In this project, two main methods were used, one being the discrete Naive Bayes algorithm and the other being the continuous Gaussian Naive Bayes algorithm. For the discrete Naive Bayes algorithm, a discrete dataset is used to create the Naive Bayes model.

Once all of the data is pre-processed, the training method will go through and count the total occurrences of each data value that corresponds to each possible output. For this project, there were only two possible outputs, 1 if there was significant coffee rust present, and 0 if the coffee rust was not significant enough to reach an economic threshold. Once all of the occurrences for each data value was counted, a Naive Bayes model was created to predict, based off of weather conditions, if coffee rust was significant or not. An example of the output model of the temperature attribute for a discrete Naive Bayes model can be seen in Figure 1.

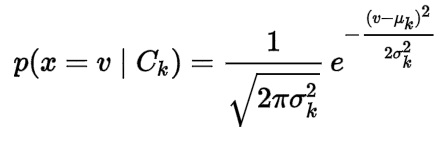
Once this model has been obtained, the program is able to predict whether or not future weather conditions will lead to significant coffee rust. This is done by comparing the probability of the output being 1 given the current attributes with the probability of the output being 0. Figure 2 illustrates an examples of how these probabilities are calculated.

The greater probability is then the predicted output. For the example given in Figure 2, the predicted output would be that, given the following values for each class, the output is most likely 1, coffee rust is significant.

**Figure 2**: An example of how predictions are calculated using the discrete Naïve Bayes algorithm.

P(Yes|x’) ≈ [P(Temperature=23|Yes)P(Humidity=75|Yes)P(Rainfall=0.2|Yes)]P(Significant Disease|Yes)= 0.0053

P(No|x’) ≈ [P(Temperature=23|No)P(Humidity=75|No)P(Rainfall=0.2|No)]P(Significant Disease|No)= 0.0023

The second implemented method of this project was the Gaussian Naïve Bayes algorithm. This method works by finding the mean and standard deviation of each class within the data. Once the mean and standard deviation for each class is calculated, the probability of each output is calculated using the formula in Figure 3.

**Figure 3**: The probability formula used in the Gaussian Naïve Bayes Algorithm

Once again, after the probability of each output is calculated given the current values for each class, the highest probability determines significant disease prediction by the model.

# Results

We used multiple algorithms and methods to get the greatest possible accuracy and to examine our results even further. We then compared our second method, Gaussian Naïve Bayes Algorithm, to Weka, a software with a collection of machine learning algorithms for data mining tasks, as a reference to compare our Gaussian Naïve Bayes algorithm. Other than accuracy, we also compared the precision, sensitivity, specificity, and F-score of all the different algorithms used.

## Discrete vs. Gaussian Naïve Bayes Algorithms

On our first trial, Discrete Naïve Bayes algorithm produced an unexpectedly high accuracy and precision rate of 68.8% and a F-score of 81.5%. For a first trial, we thought that was decent and unexpected. However, after realizing that our classes contained continuous variables rather than discrete variables, we changed our algorithm to accommodate the continuous classes. The most unexpected outcome was that Gaussian Naïve Bayes algorithm output the same predictions as the Discrete algorithm. We concluded that this was due to our small data size of 110.

**Figure 4**: Compared results of the confusion matrix between Discrete Naïve Bayes (blue), Gaussian Naïve Bayes 70/30 (orange), Gaussian Naïve Bayes 80/20 (gray), and Weka’s Gaussian Process (yellow).

## 80/20 Training to Testing Data Ratio

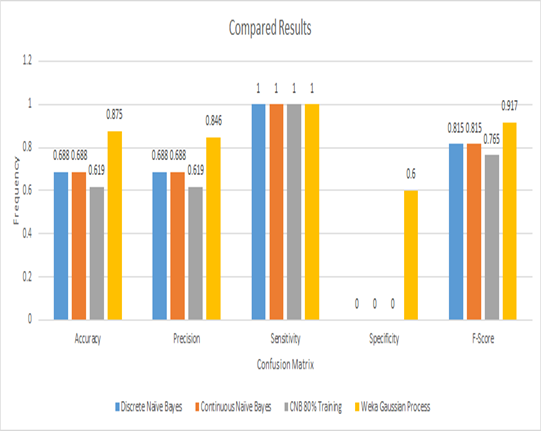
Our initial training to testing data ratio was 70/30. However, to further examine our thought of having too small of a dataset as our main problem, we changed our ratio to 80/20. After using more data to train our program, the results were worse than the outcome of the 70/30 training/testing data ratio. The accuracy and precision score for the change was 61.9% and the F-score was 76.5%. So, we concluded that the more we train our data with the same data size that we have, the worse the accuracy of our program will have. However, in theory, if we were to expand our dataset and even out the data for significant coffee rest and non-significant coffee rust data, our program’s accuracy will improve.

## Sensitivity and Specificity

Most of our data collected for areas with severe coffee rust (1) and areas with non-severe coffee rust (0) overlapped with one another, resulting in insufficient outcomes. In other words, areas without coffee rust had similar weather conditions as areas with coffee rust. When testing with our testing data from our dataset, our program was unable to correctly predict lack of presence of coffee rust. It always predicted, when there was no coffee rust significant, that coffee rust is significant, regardless of the weather conditions inputted. When testing our data with random variables, our program predicted that coffee rust was not present 2 out of 9 times. We concluded that the reason for this was our dataset had more examples of present coffee rust than examples of no coffee rust. Furthermore, the reason our program predicted two instances of no coffee rust is due to unrealistic inputs.

## Weka

Compared to Weka, all our methods resulted in worse results in accuracy, precision, specificity, and F-score. Our Gaussian Naïve Bayes algorithm had a result of a 10.2% lower F-score than Weka.



References

Amerson, P A. “Coffee Rust.” *APSnet*, The Plant Health Instructor, 2000, [www.apsnet.org/edcenter/intropp/lessons/fungi/Basidiomycetes/Pages/CoffeeRust.aspx](http://www.apsnet.org/edcenter/intropp/lessons/fungi/Basidiomycetes/Pages/CoffeeRust.aspx).

“Hemileia Vastatrix.” *Wikipedia*, Wikimedia Foundation, 5 Dec. 2017, en.wikipedia.org/wiki/Hemileia\_vastatrix.

Pedregosa, F., et al. “1.4. Support Vector Machines¶.” *1.4. Support Vector Machines — Scikit-Learn 0.19.1 Documentation*, Scikit-Learn, 2011, scikit-learn.org/stable/modules/svm.html#multi-class-classification.

Sainato, Michael. “Central America’s Slow Recovery From Coffee-Ravaging Epidemic.” *Observer*, Observer, 22 Sept. 2015, observer.com/2015/09/central-americas-slow-recovery-from-coffee-ravaging-epidemic/.

“Weather Forecast & Reports - Long Range & Local.” *Weather Underground*, [www.wunderground.com/](http://www.wunderground.com/).