Programming Project 1: Winnow-2 and Naïve Bayes Algorithms for Linear Classification Problems

Ricca D. Callis

June 8, 2020

Abstract

Supervised Machine Learning Algorithms require structured (or labeled) input data in order to learn a function between inputs and outputs. This Project sought to implement two Supervised Machine Learning Algorithms, Winnow-2 and Naïve Bayes, in the context of classification. Both algorithms were applied to categorical (discrete) as well as continuous-valued input data using five data sets obtained from the UCI Machine Learning Repository.

Introduction

Both Machine Learning and Statistics create models from data. While Statistics aims at describing the structure of data (i.e., descriptive statistics), as well as approximating, and understanding the data-generating process (i.e., inferential statistics). Inferential statistics create and fit a probability model using certain assumptions about the data-generating system. Therefore, a statistician is primarily concerned with model validity, accurate estimation of model parameters, and inference from the model. Machine Learning, on the other hand, aims to build models that yield accurate predictions aimed at forecasting unobserved outcomes or future behavior. Machine Learning concentrates on prediction by using general purpose learning algorithms, which make minimal assumptions about the data-generating system, in order to find patterns in data.

Machine Learning is further classified based on the structure, or lack thereof, of the input data. Supervised Machine Learning algorithms learn a function between structured input data and it's output. Supervised Machine Learning algorithms are typically applied to classification problems or regression problems. Unsupervised Machine Learning algorithms have no target variable and may be applied to clustering-type problems.

This project provided students enrolled in an Introduction to Machine Learning course (605.649.83.SU20), at Johns Hopkins University, the opportunity to implement two Supervised Machine Learning Algorithms – Winnow-2 and Naïve Bayes. Both of these algorithms are used for classification predictive modeling, where a class label (i.e., membership) is assigned to input examples. In classification tasks, your job is to build a function $y' = f(x)$ that takes in a vector of features x (also called "inputs") and predicts a label y (also called the "class" or "output").

Features are known, whereas labels are what the algorithm is attempting to learn. Generally, classification problems can be further delineated based on the number of class predictions made. Binary classification refers to predicting one of two class labels, whereas multi-class classification involves predicting one of more than two classes. Algorithms that are designed for binary classification can be adapted for use for multi-class problems by fitting multiple binary classification models for each class vs all other classes (called one-vs-rest) or one model for each pair of classes (called one-vs-one).

Winnow-2 is a Supervised Machine Learning Algorithm for binary classification (Littlestone, 1988). In Winnow-2, the learner receives a data instance (a vector binary attribute) and then makes a prediction for the data instance by assigning a class label of 0 (doesn't belong) or 1 (does belong). Then, the learner is told whether the prediction is accurate. If the learner made an accurate prediction, nothing happens. If, however, an inaccurate prediction was made, learning occurs.

Naïve Bayes is also a Supervised Machine Learning Algorithm capable of predicting binary classifications as well as multivalued discrete and continuous inputs. Unlike Winnow-2, Naïve Bayes is a probabilistic classifier that assigns class labels using a Bayesian decision rule.

For this project, both Winnow-2 and Naïve Bayes were compared using Boolean/binary inputs. Only Naïve Bayes, however, was applied to multivalued inputs, both discrete and continuous. Neither algorithm treats interactions between variables. Winnow-2 merely adjusts all active weights while Naïve Bayes assumes the feature values are conditionally independent on the class. Therefore, large differences in performance between the two are not expected.

Algorithms and Experimental Methods

Winnow-2

As previously mentioned, Winnow-2 is a linear-threshold binary classifier (e.g., Littlestone, 1988). Specifically, Winnow-2 assumes that all class labels are binary $(\forall_i\, y^i \in \{0,1\})$ and all features are binary $(\forall_i\, x^i \in \{0,1\})$. Winnow-2 is an online algorithm (i.e., supervised) for learning a function from a concept class Y (in the form of $x_1 \vee x_2 \vee \ldots \vee x_i$). The algorithm is given examples, x_i , drawn from a larger set of *d* features (or variables) and must predict the value of the target class $y' = f(X)\ on\ X, Y$. After each step, it learns whether the prediction was correct and then updates future predictions accordingly.

Specifically, Winnow-2 is given a binary/Boolean input vector, $X = (x_1, x_2, \ldots, x_i)$ and assigns a weight, w_i, to each x_i. The vector of weights $W = (w_1, w_2, \ldots, w_i)$ is initialized to 1. Winnow-2 also initializes two other parameters: $\theta = 0.5\ and\ \alpha = 2$. For each training instance, x^t, Winnow-2 predicts $y' = f(x^t)$ by calculating the weighted sum over the dimensions of each attribute $f(x^t) = \sum_{i=1}^{d} x_i w_i$. If $f(x^t) \geq \theta$, Winnow-2 predicts $y' = 1$ (i.e., instance belongs to class). If $f(x^t) < \theta$, Winnow-2 predicts $y' = 0$ (i.e., does not belong to class). Thus,

$$y' = f(x^t) = \begin{cases} 1 & if\ f(x^t) \geq 0 \\ 0 & otherwise \end{cases}$$

Afterward, the prediction, y', is compared to the actual class label, y. If an incorrect prediction was made, learning occurs. If the prediction (y') = 1, but the actual class label (y) = 0 (also known as a false positive), all the weights that contributed to the incorrect prediction are demoted: $w_i' \leftarrow w_i * \alpha\ \forall_i | \ x_i = 1$. If the prediction ($y'$) = 0, but the actual class label (y) = 1 (also known as a false negative), all the weights that contributed to the incorrect prediction are

promoted: $w_i' \leftarrow \frac{w_i}{\alpha} \ \forall_i | \ x_i = 1$. If no mistake was made, then $y = y'$, no updates to the weight vector are made. This means the algorithm can be used without even knowing the full set of variables. This is important because it allows the algorithm to train continuously on new, unseen inputs without any adjustment.

Naïve Bayes

Naïve Bayes is a probabilistic classifier which applies Bayes' Rule and strong (naïve) independence assumptions (e.g., Hand & Yu, 2001; Zhang, 2004). A natural way to define the classification prediction function, $y' = f(X)$, is to predict the label with the highest conditional probability: choose $f(X) = argmax_y P(y' = y|x)$. Let's first review Bayes' Theorem.

Abstractly, the probability model for a classifier is a conditional model: $P(C|F_1, \dots, F_n)$, where the class variable C is a small number of outcomes or classes, conditional on several feature variables F_1 through F_n. To make the model more tractable for a large number of features *n* or multi-value features, we use Baye's Rule:

$$P(C|F_1, \dots, F_n) = \frac{P(C) * P(F_1, \dots, F_n|C)}{P(F_1, \dots, F_n)}$$

Said another way,

$$Posterior = \frac{Prior * Likelihood}{Evidence}$$

In practice, for supervised learning classification problems, we only really care to calculate the numerator of that fraction. The denominator does not depend on C and the values of the features F_i are given. The numerator, then, becomes the joint probability $P(C, F_1, \dots, F_n)$ .Naïve Bayes assumes conditional independence, in that it assumes each feature F_i is

conditionally independent of every other feature $F_j \; for \; j \neq i$. Thus, $P(F_i|C, F_j) = P(F_i|C)$. As a result, the joint model can be expressed as:

$P(C, F_1, \dots, F_n) = P(C) * P(F_1|C) * P(F_2|C) * P(F_n|C) = P(C) \prod_{i=1}^{n} P(F_i|C)$. And under the independent assumption, the conditional distribution over the class variable C can be expressed as: $P(C|F_1, \dots, F_n) = \frac{1}{Z} P(C) \prod_{i=1}^{n} P(F_i|C)$, where Z (the evidence) is a scaling factor dependent only on $F_1, \dots, F_n$ (i.e., a constant if the values of the feature variable as known).

This makes the model much more manageable, calculating the model parameters for the class prior $P(C)$ and independent probability distributions $P(F_i|C)$. These model parameters can be approximated using maximum likelihood estimates. The class prior may be estimated from the training set: $P(C) = \frac{number \; of \; samples \; in \; the \; class}{total \; number \; of \; samples}$. To estimate the parameters for a feature's distribution, one must assume a distribution or generate nonparametric models for the features from the training set. If one is dealing with continuous data, a typical assumption is that the continuous values associated with each class are distributed according to a Gaussian distribution.

Naïve Bayes defines the classification prediction function, $y' = f(X)$, by predicting the label with the highest conditional probability: choose $f(X) = argmax_y P(y' = y|x)$. To compute this probability, Naïve Bayes uses training data consisting of examples of feature-label pairs (x, y). More specifically, Naïve Bayes takes *n* pairs: $(x_i^1, y_i^1), (x_i^2, y_i^2), \dots, (x_i^n, y_i^n)$, where x^i is a vector of *m* discrete features for the i^{th} training example and y^i is the discrete label for the i^{th} training example. The training objective is to estimate the prior class probability $\hat{P}(y')$ and the independent conditional probability distribution $\hat{P}(X_j|y')$ for all $2 \geq j \leq m$ features. Using a maximum likelihood estimate,

$$\hat{P}(X_i = x_i | y' = y) = \frac{\#\ training\ examples\ where\ X_j = x_i\ and\ y' = y}{training\ examples\ where\ y' = y}$$

For an example with $x = [x_1, x_2, \ldots, x_m]$, Naïve Bayes estimates the value of y as:

$y' = f(X) = argmax\ \hat{P}(y')\hat{P}(X|y')$ which is equal to $argmax\ \hat{P}(y' = y|X)$

And due to the Naïve Bayes Assumption, we use the Maximum A Posteriori (MAP) decision rule:

$$y' = f(X) = argmax\ \hat{P}(y' = y) \prod_{j=1}^{m} \hat{p}\left(X_j = x_j | y' = y\right)$$

As mentioned earlier, Naïve Bayes is also capable of learning classifications for continuous attributes. To do so, the algorithm first segments the data by class and then computes the mean and variance of *x* in each class. Let μ_y, be the mean of the values in x associated with class y and let σ_y^2,be the variance of the values in *x* associated with class y. Then, the probability of some value given a class, $P(X = x_i|y) = \frac{1}{\sqrt{2\pi\sigma_y^2}} e^{-\frac{(x_i - \mu_y)^2}{2\sigma_y^2}}$ (which is the equation for a normal distribution; see for example Alpaydın, 2020). Alternatively, continuous values may be binned into discrete values.

Although an inaccurate real-world assumption, the Naïve Bayes conditional independence assumptions eliminates the need for data sets that scale exponentially with the number of features. Furthermore, as a probabilistic MAP classifier, class probabilities do not necessarily need to estimated well so long as the correct class is more probably than any other class. (e.g., Hand & Yu, 2001; Zhang, 2004).

Data Sets

This analysis was conducted on 5 data sets, each obtained from the UCI Machine Learning Repository:

(1) Breast Cancer Data Set

(2) Glass Data Set

(3) Iris Data Set

(4) Soybean Data Set

(5) House Voting Data Set

For each data set, discrete inputs were dummy/boolean coded using one-hot encoding. Both Winnow-2 and Naïve Bayes analyzed these discrete Boolean input data sets. Continuous values were discretized based on logical bin boundaries separated by class. This was done in an attempt to better separate classes for one-hot encoding. One vs Rest Naïve Bayes classifiers were fit to multivalued outputs. As described above, Winnow-2 used the threshold $\theta = 0.5$ for all classes. Naïve Bayes selected the highest likelihood class using the MAP decision rule.

For comparison, Winnow-2 and Naïve Bayes algorithms were applied to boolean/binary inputs. In addition, Naïve Bayes was fit using multivalued discrete and continuous attributes. For each experiment, a 5-fold randomly-shuffled cross-validation approach was used to evaluate the models. Not only does this ensure that the data that was used to train the model came from the same distribution as the data that will be used to model at testing, but it also yields multiple estimates of the evaluation metrics.

Breast Cancer Data Set

Data Description: Classifies tumors as either malignant or benign, based on 10 feature attributes: id number, clump thickness, uniformity of cell size, uniformity of cell shape, marginal adhesion, single epithelial cell size, bare nuclei, bland chromatin, normal nucleoli, and mitosis (Wolberg, 1992; Wolberg & Mangasarian, 1990). This is a multivariate data set with 699 instances, where each attribute's instance is represented as a discrete value integer, ranging from 1 to 10.

Data Cleaning & Transformation: All instances with missing data were dropped from the data set (16 rows out of 699). The attribute id number was also dropped from the data set, as it represented a unique identifier that would not serve to teach class attributes. Due to the fact that inputs were all multivalued discrete, one-hot encoding representations were made for each attribute column.

Exploratory Data Analysis: Roughly 65.47% of the actual data instances were classified as benign, where as roughly 34.53% of the actual data instances were classified as malignant (see Figure 1). Each feature was also described by class (descriptive statistics included mean, standard deviation, minimum, maximum, range, 1st quartile, median, and 3rd quartile) and were plotted using a box-plot (See for example Figure 2). The means for each feature by class can be observed in Table 1.

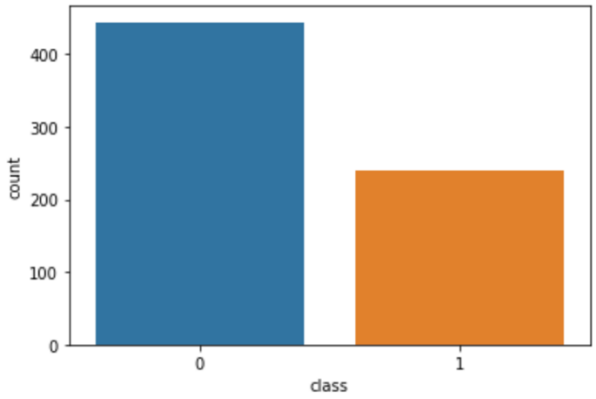

FIGURE 1. Breast cancer data set plot showing the actual raw count values for tumor classification (where 0 = benign; 1 = malignant).

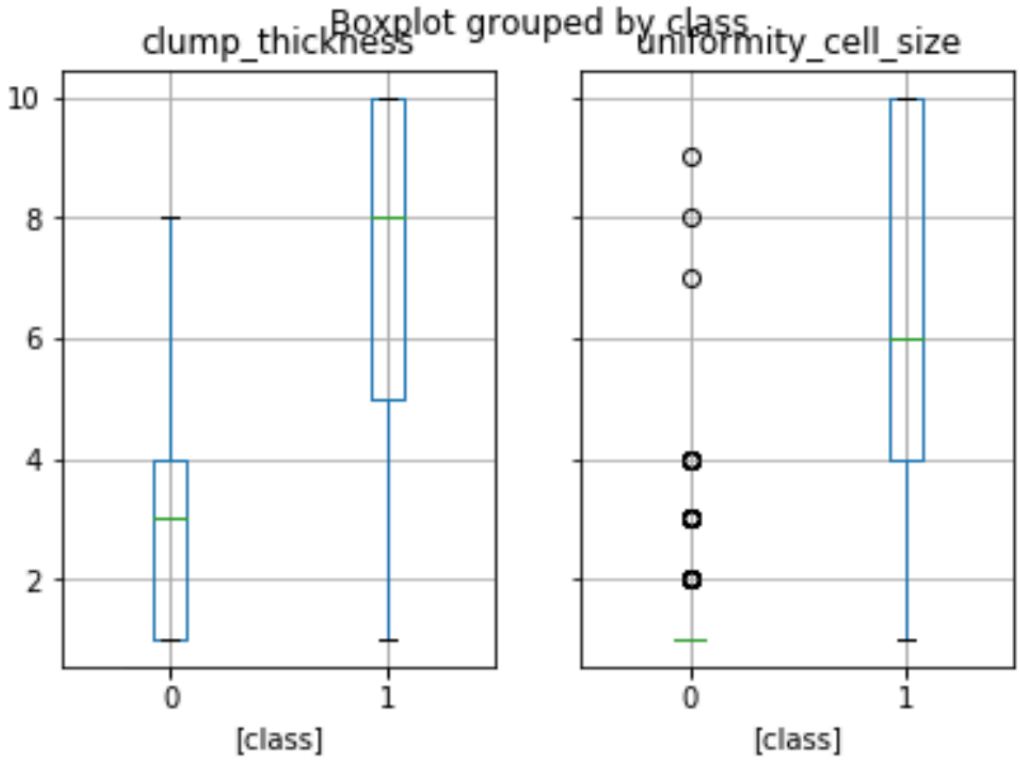

FIGURE 2. Breast cancer data set boxplot showing attribute grouped by class. Above example depicts the features clump thickness and uniformity in cell size by class = 0 (benign) and class = 1 (malignant)

	Clump Thickness	Uniformity Cell Size	Uniformity Cell Shape	Marginal Adhesion	Single Epithelial Cell Size	Bare Nuclei	Bland Chromatin	Normal Nucleoli	Mitosis
Benign	2.963	1.306	1.414	1.346	2.108	1.346	2.083	1.261	1.065
Malignant	7.188	6.577	6.560	5.587	5.326	7.627	5.974	5.857	2.602

TABLE 1. Mean values of each feature by tumor classification in the Breast Cancer Data Set.

Glass Data Set

Data Description: Classifies origin of broken glass, based on 10 feature attributes of the broken shards: id number, refractive index, sodium, magnesium, aluminum, silicon, potassium, calcium, barium, and iron (German, 1987). The class attribute has 6 classifications: building windows float processed, building windows nonfloat processed, vehicle windows float processed, containers, tableware, or headlamp. This is a multivariate data set with 214 instances, where each attribute's instance is represented as a continuous value float.

Data Cleaning & Transformation: All instances with missing data were dropped from the data set (16 rows out of 699). The attribute id number was also dropped from the data set, as it represented a unique identifier that would not serve to teach class attributes. Due to the fact that inputs were all continuous values, bins boundaries were created for each attribute so that data could be discretized before applying one-hot encoding to each attribute. The following bins were used: refractive index [1.518], sodium 10.5, 12.5, 14.5], magnesium [1, 2, 3, 4], aluminum [0.5, 1.5, 2, 2.5], silicon [72.5, 73.0, 73.5, 74.0], potassium [0.2, 0.6, 1.2, 1.6], calcium [6, 9, 12], barium [0.5, 1.0, 2.0], and iron [0.2, 0.6].

Exploratory Data Analysis: There were 76 instances classified as 2 (building windows nonfloat processed), 70 instances classified as 1 (building windows float processed), 29 instances classified as 7 (headlamps), 17 instances classified as 3 (vehicle windows float processed), 13

instances classified as 5 (containers), and 9 instances classified as 6 (tableware; see Figure 3). Each feature was also described by class (descriptive statistics included mean, standard deviation, minimum, maximum, range, 1st quartile, median, and 3rd quartile) and were plotted using a box-plot (See for example Figure 4). The means for each feature by class can be observed in Table 2.

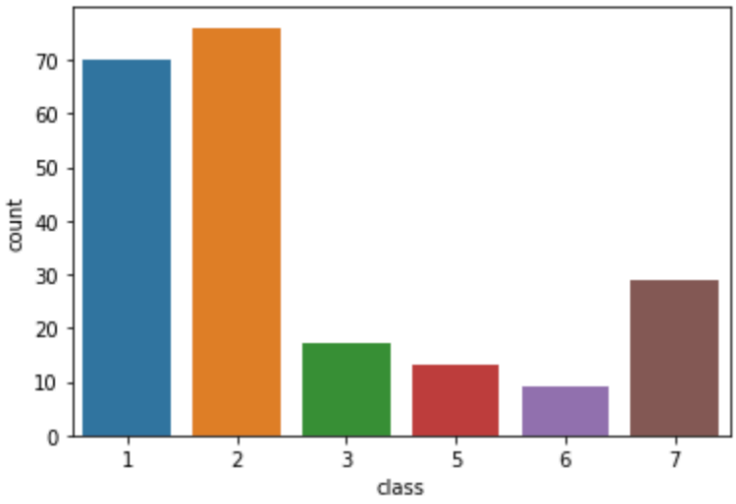

FIGURE 3. Glass data set plot showing the actual raw count values for glass classification (where 1 = building windows float processed; 2 = building windows nonfloat processed; 3 = vehicle windows float processed; 5 = containers; 6 = tablewear; 7 = headlamps).

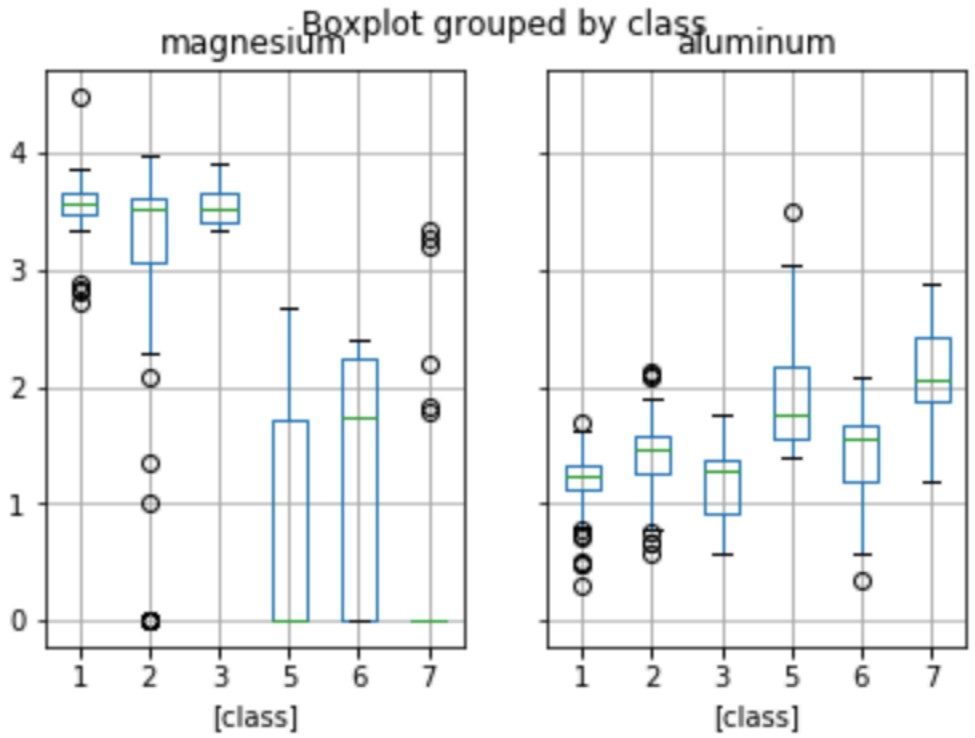

FIGURE 4. Glass data set boxplot showing attribute grouped by class. Above example depicts the features magnesium and aluminum by class (where 1 = building windows float processed; 2 = building windows nonfloat processed; 3 = vehicle windows float processed; 5 = containers; 6 = tablewear; 7 = headlamps).

	Refractive Index	Sodium	Magnesium	Aluminum	Silicon	Potassium	Calcium	Barium	Iron
Building Windows Float Processed	1.518	13.242	3.552	1.163	72.619	0.447	8.797	0.012	0.057
Building Windows Nonfloat Processed	1.518	13.111	3.002	1.408	72.598	0.521	9.073	0.050	0.079
Vehicle Windows Float Processed	1.517	13.437	3.543	1.201	72.404	0.404	8.782	0.008	0.057
Containers	1.518	12.827	0.773	2.033	72.366	1.470	10.123	0.187	0.060
Tablewear	1.517	14.646	1.305	1.366	73.206	0.000	9.3566	0.000	0.000
Headlamps	1.517	14.442	0.538	2.122	72.965	0.325	8.491	1.040	0.013

TABLE 2. Mean values of each feature by glass classification in the glass Data Set.

Iris Data Set

Data Description: Classifies Iris species (Iris Setosa, Iris Versicolour, or Iris Virginica) based on 4 feature attributes from leaf measurements: sepal length, sepal width, petal length, and petal width (Fisher, 1988). As mentioned, the class attribute has 3 classifications: Iris Setosa, Iris Versicolour, or Iris Virginica. This is a multivariate data set with 150 instances, where each attribute's instance is represented as a continuous value float. Due to the fact that inputs were all continuous values, bins boundaries were created for each attribute so that data could be discretized before applying one-hot encoding to each attribute. The following bins were used: sepal length [4.5, 5.5, 6.5, 7.5], sepal width [2, 3, 4], petal width [0.5, 1, 1.5, 2], petal length [1, 2, 4, 6].

Exploratory Data Analysis: There were 50 instances classified as Iris Versicolor, 50 instances classified as Iris Virginica, and 49 instances classified as Iris Setosa (see Figure 5). Each feature was also described by class (descriptive statistics included mean, standard deviation, minimum, maximum, range, 1st quartile, median, and 3rd quartile) and were plotted using a box-plot (See for example Figure 6). The means for each feature by class can be observed in Table 3.

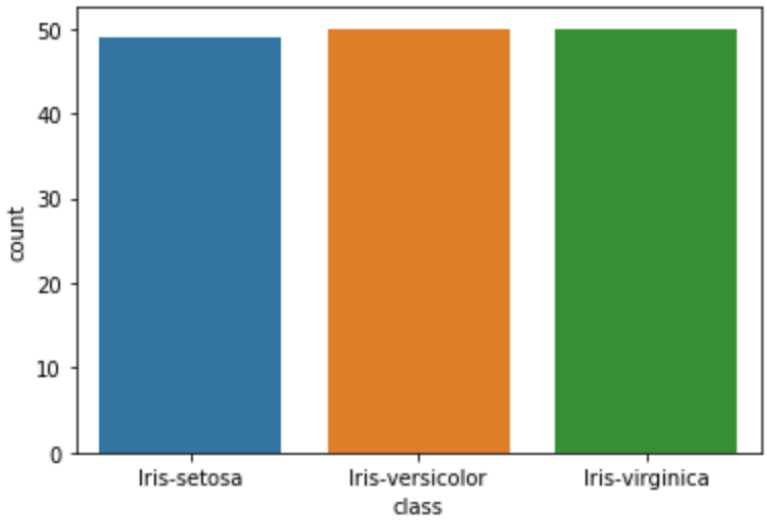

FIGURE 5. Iris data set plot showing the actual raw count values for Iris classification.

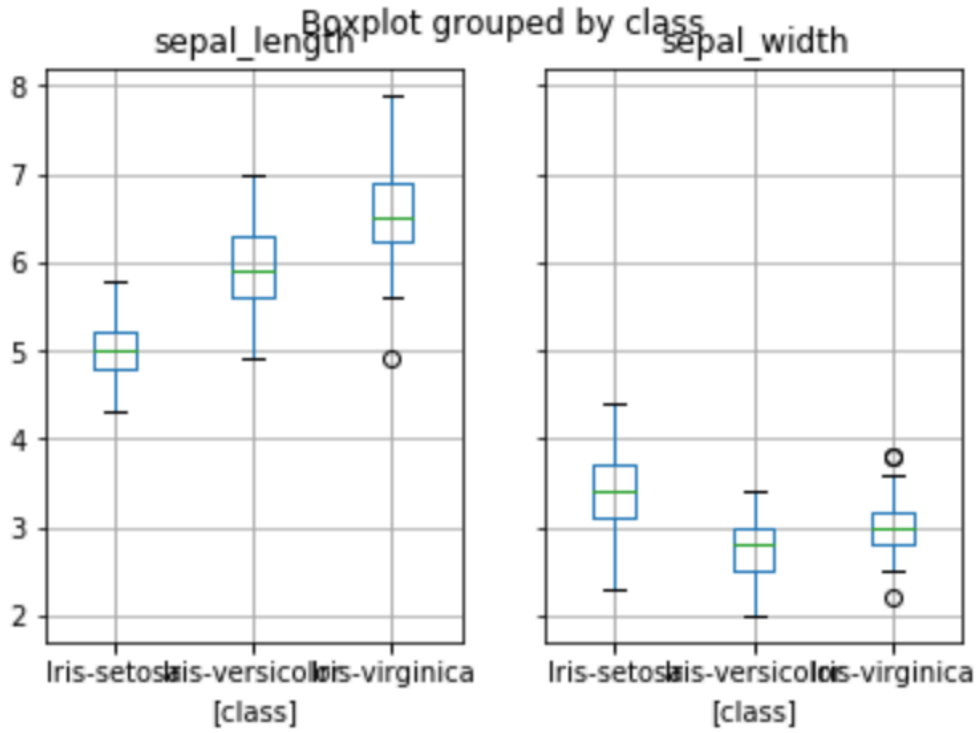

FIGURE 6. Iris data set boxplot showing attribute grouped by class. Above example depicts the features sepal length and sepal width by class.

	Sepal Length	Sepal Width	Petal Length	Petal Width
Iris Setosa	5.004	3.416	1.465	0.244
Iris Versicolor	5.936	2.770	4.260	1.326
Iris Virginica	6.588	2.974	5.552	2.026

TABLE 3. Mean values of each feature by Iris classification in the Irirs Data Set.

Soybean Data Set

Data Description: Classifies soybean disease based on 36 feature attributes of the crop: date, plant stand, precipitation, temperature, hail, crop history, area damaged, severity, seed tmt, germination, plant growth, leaves, leafspots-halo, leafspots-marg, leafspot size, leaf shred, leaf

malf, leaf mild, stem, lodging, stem cankers, canker-lesion, fruiting bodies, external decay, mucelium, int-discolor, sclerotia, fruit pods, fruit spots, seed, mold growth, seed discolor, seed size, shriveling, and roots (Michalski, 1987). The class attribute has 4 classifications: D0, D1, D2, or D3. This is a multivariate data set with 47 instances, where each attribute's instance is represented as a discrete value integer. Some attributes only had a single value and were dropped due to the fact that these algorithms would be unable to learn classifications that way. All remaining attributes were mapped into one-hot encodings.

Exploratory Data Analysis: There were 17 instances classified as D3, 10 instances classified as D2, 10 instances classified as D1, and 9 instances classified as D0 (see Figure 7). Each feature was also described by class (descriptive statistics included mean, standard deviation, minimum, maximum, range, 1st quartile, median, and 3rd quartile) and were plotted using a box-plot (See for example Figure 8). The means for a few of the feature by class can be observed in Table 4.

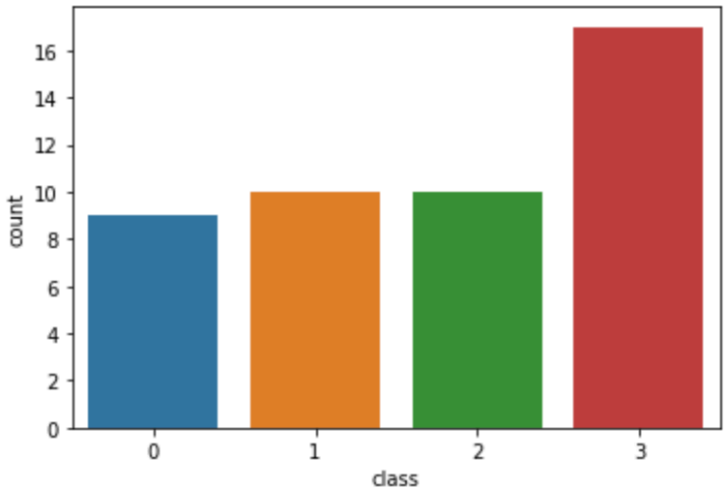

FIGURE 7. Soybean data set plot showing the actual raw count values for disease/rot classification.

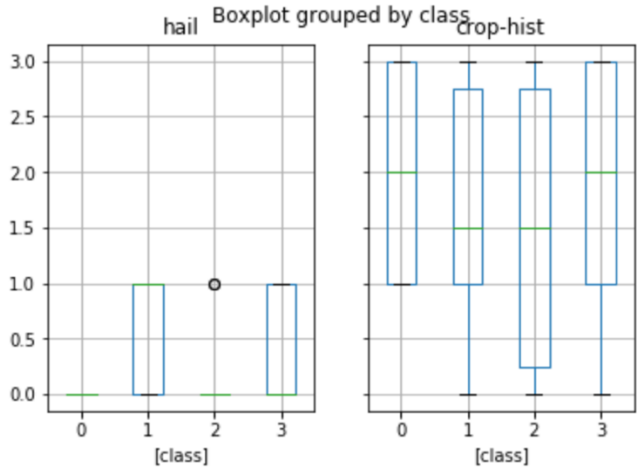

FIGURE 8. Soybean data set boxplot showing attribute grouped by class. Above example depicts the features hail and crop history by class.

	Date	Plant Stand	Precip	Temp	Hail	Crop hist	Area damaged	Severity	Seed tmt	Germination
D0	4.55	0.0	2.00	1.00	0.00	2.00	1.33	0.55	1.22	0.44
D1	4.70	0.0	0.00	1.60	0.60	1.60	2.50	1.00	0.50	0.90
D2	1.30	0.8	2.00	0.00	0.20	1.50	1.00	1.50	0.40	1.50
D3	1.29	1.0	1.76	0.58	0.35	1.82	1.11	1.64	0.52	0.94

TABLE 4. Mean values of a few of the feature by soybean disease/rot classification in the Soybean Data Set.

House Voting Data Set

Data Description: Classifies party (republican or democrat) based on 16 feature attributes of legislation votes: handicapped infants, water project cost sharing, adoption of the budget resolution, physician fee freeze, el Salvador aid, religious groups in schools, anti-satellite test ban, aid to Nicaraguan contras, mx missile, immigration, synfuels corporation cutback, education spending, superfund right to sue, crime, duty free exports, and export administration act south

Africa (Congress Quarterly Almanac, 1984). This is a multivariate data set with 435 instances, where each attribute's instance is represented as a discrete Boolean value, where 1 indicates that the congressperson voted for a measure and 0 indicates that the congressperson voted against a measure.

Exploratory Data Analysis: There were 124 instances classified as Republican and 108 instances classified as Democrat (see Figure 8).

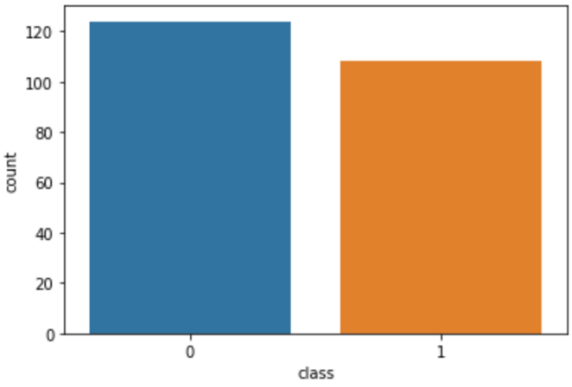

FIGURE 8. House votes data set plot showing the actual raw count values Republican (Class = 0) and Democrat (Class = 1).

Results

Breast Cancer Data Set

Winnow-2 on Breast Cancer Data Set: Winnow-2 predicted 51 malignant cases (compared to 52 actual malignant cases) and 85 benign cases (compared to 84 actual benign cases; see Table 5.). Winnow-2 made a total of 3 inaccurate predictions (out of 136 total predictions), yielding an accuracy rate of 97.79%. Of the three incorrect predictions, two were false negatives (resulting in promotion) and one was a false positive (resulting in demotion).

Winnow-2	Prediction	Actual
Benign	85	84
Malignant	51	52
Accuracy: 97.79%		

TABLE 5. Winnow-2 results on Breast Cancer Data Set. Values indicate the total number of class assignments (both predicted and actual) during 5-fold cross-validation.

Naïve Bayes (Discrete Binary Inputs) on Breast Cancer Data Set: Naïve-Bayes applied to binary/Boolean inputs predicted 49 malignant cases (compared to 45 actual malignant cases) and 87 benign cases (compared to 91 actual benign cases; see Table 6.). Naïve Bayes made a total of 4 inaccurate predictions (out of 136 total predictions), yielding an accuracy rate of 97.06%. All four inaccurate predictions were false negatives.

Naïve Bayes (Binary Input)	Prediction	Actual
Benign	87	91
Malignant	49	45
Accuracy: 97.06%		

TABLE 6. Naïve Bayes with Binary/Boolean Inputs on Breast Cancer Data Set. Values indicate the total number of class assignments (both predicted and actual) during 5-fold cross-validation.

Naïve Bayes (Multinomial Inputs) on Breast Cancer Data Set: Naïve-Bayes applied to multinomial inputs were identical to those predicted with binary/Boolean inputs. Specifically, multinomial Naïve Bayes predicted 49 malignant cases (compared to 45 actual malignant cases) and 87 benign cases (compared to 91 actual benign cases; see Table 7.). Naïve Bayes made a total of 4 inaccurate predictions (out of 137 total predictions), yielding an accuracy rate of 96.35%. All four inaccurate predictions were false negatives.

Naïve Bayes (Multinomial Input)	Prediction	Actual
Benign	87	91
Malignant	49	45
Accuracy: 97.06%		

TABLE 7. Naïve Bayes with Multinomial Inputs on Breast Cancer Data Set. Values indicate the total number of class assignments (both predicted and actual) during 5-fold cross-validation.

Overall Conclusions: Results show accuracy at 97.79% for Winnow-2, 97.06% for Boolean Naïve Bayes, and 97.06% for Multinomial Naïve Bayes. Overall, we see that all algorithms performed equally well, regardless of input type for Naïve Bayes. Since the data set inputs were non-continuous to begin with, it makes sense that Naïve Bayes did not hold an advantage.

Glass Data Set

Winnow-2 on Glass Data Set: For the class label 0, Winnow-2 predicted 32 cases, compared to 18 actual cases, indicating 14 false positive assignments to class = 0 (see Table 8). We see that this class label assignment yielded the highest miss-rate. For the class label 1, Winnow-2 predicted 2 cases, compared to 12 actual cases, indicating 10 false negative assignments to class = 1. For the class label 2, Winnow-2 predicted 0 cases, compared to 4 actual cases, indicating 4 false negative assignments to class = 2. For the class label 3, Winnow-2 predicted 7 cases, compared to 3 actual cases, indicating 4 false positive assignments to class = 3. For the class label 4, Winnow-2 predicted 0 cases, compared to 1 actual case, indicating 1 false negative assignment to class = 4. For the class label 5, the final class label, Winnow-2 predicted 1 case, compared to 0 actual cases, indicating 1 false positive assignment to class = 5.

Winnow-2 made a total of 22 inaccurate predictions (out of 42 total predictions), yielding an accuracy rate of only 52.38%.

Winnow-2	Prediction	Actual
0	32	18
1	2	12
2	0	4
3	7	3
4	0	1
5	1	0
Accuracy: 52.38%		

TABLE 8. Winnow-2 results on Glass Data Set. Values indicate the total number of class assignments (both predicted and actual) during 5-fold cross-validation.

Naïve Bayes (Discrete Boolean Inputs) on Glass Data Set: Naïve-Bayes applied to boolean inputs predicted 19 classifications of class = 0, compared to 18 actual cases, indicating 1 false positive assignment (see Table 9). Boolean Naïve Bayes predicted 14 cases of class = 1, compared to 12 actual cases, indicating 2 false positive assignments. Boolean Naïve Bayes predicted 0 cases of class = 2, compared to 4 actual cases, indicating 4 false negative assignments. Boolean Naïve Bayes predicted 2 cases of class = 3, compared to 3 actual cases, indicating 1 false negative assignment. Boolean Naïve Bayes predicted 3 cases of class = 4, compared to 1 actual case, indicating 3 false positive assignments. Finally, Boolean Naïve Bayes predicted 4 cases of class = 5, which accurately predicted all 5 actual cases. Boolean Naïve Bayes made a total of 17 inaccurate predictions (out of 42 total predictions), yielding an accuracy rate of 59.52%.

Naïve Bayes (Boolean Input)	Prediction	Actual
0	19	18
1	14	12
2	0	4
3	2	3
4	3	1
5	4	4
Accuracy: 59.52%		

TABLE 9. Naïve Bayes with Boolean Inputs on Glass Data Set. Values indicate the total number of class assignments (both predicted and actual) during 5-fold cross-validation.

Naïve Bayes (Multinomial Inputs) on Glass Data Set: Naïve-Bayes applied to multinomial inputs predicted 0 classifications of class = 0, compared to 18 actual cases, indicating 18 false negative assignments (see Table 10). Multinomial Naïve Bayes predicted 5 cases of class = 1, compared to 12 actual cases, indicating 7 false negative assignments. Multinomial Naïve Bayes predicted 28 cases of class = 2, compared to 4 actual cases, indicating 24 false positive assignments. Multinomial Naïve Bayes predicted 4 cases of class = 3, compared to 3 actual cases, indicating 1 false positive assignment. Multinomial Naïve Bayes predicted 2 cases of class = 4, compared to 1 actual case, indicating 1 false positive assignment. Finally, Multinomial Naïve Bayes predicted 3 cases of class = 5, compared to 4 actual cases, indicating 1 false negative assignment. Multinomial Naïve Bayes made a total of 29 inaccurate predictions (out of 42 total predictions), yielding an accuracy rate of 30.95%.

Naïve Bayes (Multinomial Input)	Prediction	Actual
0	0	18
1	5	12
2	28	4
3	4	3
4	2	1
5	3	4
Accuracy: 30.95%		

TABLE 10. Naïve Bayes with Multinomial Inputs on Glass Data Set. Values indicate the total number of class assignments (both predicted and actual) during 5-fold cross-validation.

Overall Conclusions: Results show accuracy at 52.38% for Winnow-2, 59.52% for Boolean Naïve Bayes, and 30.95% for Multinomial Naïve Bayes. Overall, we see that Boolean Naïve Bayes was slightly more accurate than Winnow-2. Surprisingly, the accuracy for Multinomial Naïve Bayes than for Boolean Naïve Bayes. Considering this was a continuous dataset, the opposite was hypothesized. We also note the lower accuracy rates for all three algorithms, compared to those observed in the previous data set analysis. Perhaps this effect is due to the higher number of classes within the Glass Data Set.

Iris Data Set

Winnow-2 on Iris Data Set: For the class label 0, Winnow-2 predicted 14 cases, compared to 182 actual cases, indicating 2 false positive assignments (see Table 11). For the class label 1, Winnow-2 predicted 4 cases, compared to 9 actual cases, indicating 5 false negative assignments to class = 1. For the class label 2, Winnow-2 predicted 11 cases, compared to 8 actual cases,

indicating 2 false positive assignments. Winnow-2 made a total of 7 inaccurate predictions (out of 29 total predictions), yielding an accuracy rate of 75.86%.

Winnow-2	Prediction	Actual
0	14	12
1	4	9
2	11	8
Accuracy: 75.86%		

TABLE 11. Winnow-2 results on Iris Data Set. Values indicate the total number of class assignments (both predicted and actual) during 5-fold cross-validation.

Naïve Bayes (Discrete Boolean Inputs) on Iris Data Set: Naïve-Bayes applied to boolean inputs predicted 12 classifications of class = 0, compared to 11 actual cases, indicating 1 false positive assignment (see Table 12). Boolean Naïve Bayes predicted 10 cases of class = 1, compared to 9 actual cases, indicating 1 false positive assignment. Boolean Naïve Bayes predicted 6 cases of class = 2, compared to 8 actual cases, indicating 2 false negative assignments. Boolean Naïve Bayes made a total of 2 inaccurate predictions (out of 29 total predictions), yielding an accuracy rate of 93.10%.

Naïve Bayes (Boolean Input)	Prediction	Actual
0	12	11
1	10	9
2	6	8
Accuracy: 93.10%		

TABLE 12. Naïve Bayes with Boolean Inputs on Iris Data Set. Values indicate the total number of class assignments (both predicted and actual) during 5-fold cross-validation.

Naïve Bayes (Multinomial Inputs) on Iris Data Set: Multinomial Naïve Bayes predicted each classification with 100% accuracy (see Table 13). Specifically, Multinomial Naïve-Bayes predicted 12 classifications of class = 0 (compared to 12 actual cases), predicted 9 cases of class = 1 (compared to 9 actual cases), and predicted 8 cases of class = 2 (compared to 8 actual cases).

Naïve Bayes (Multinomial Input)	Prediction	Actual
0	12	12
1	9	9
2	8	8
Accuracy: 100.00%		

TABLE 13. Naïve Bayes with Multinomial Inputs on Iris Data Set. Values indicate the total number of class assignments (both predicted and actual) during 5-fold cross-validation.

Overall Conclusions: Results show accuracy at 75.86% for Winnow-2, 93.10% for Boolean Naïve Bayes, and 100.00% for Multinomial Naïve Bayes. Overall, we see that Naïve Bayes more accurate than Winnow-2, perhaps because Winnow-2 had discretized bins. Clearly Multinomial Naïve Bayes was the most accurate.

Soybean Data Set

Winnow-2 on Soybean Data Set: Winnow-2 was 100% accurate in predicting class assignments (see Table 14). Specifically, Winnow-2 predicted 1 case for class = 0 (compared to 1 actual case), predicted 2 cases for class = 1 (compared to 2 actual cases), predicted 2 cases for class = 2 (compared to 2 actual cases), and predicted 4 cases for class = 3 (compared to 4 actual cases).

Winnow-2	Prediction	Actual
0	1	1
1	2	2
2	2	2
3	4	4
Accuracy: 100.00%		

TABLE 14. Winnow-2 results on Soybean Data Set. Values indicate the total number of class assignments (both predicted and actual) during 5-fold cross-validation.

Naïve Bayes (Discrete Boolean Inputs) on Soybean Data Set: Boolean Naïve-Bayes predicted all class assignments with 100% accuracy. Specifically, Boolean Naïve Bayes predicted 1 case for class = 0 (compared to 1 actual case), predicted 2 cases for class = 1 (compared to 2 actual cases), predicted 2 cases for class = 2 (compared to 2 actual cases), and predicted 4 cases for class = 3 (compared to 4 actual cases).

Naïve Bayes (Boolean Input)	Prediction	Actual
0	1	1
1	2	2
2	2	2
3	4	4
Accuracy: 100.00%		

TABLE 15. Naïve Bayes with Boolean Inputs on Soybean Data Set. Values indicate the total number of class assignments (both predicted and actual) during 5-fold cross-validation.

Naïve Bayes (Multinomial Inputs) on Soybean Data Set: Multinomial Naïve Bayes predicted each classification with 100% accuracy (see Table 16). Specifically, Multinomial Naïve-Bayes

predicted 1 classification of class = 0 (compared to 1 actual case), predicted 2 cases of class = 1 (compared to 2 actual cases), predicted 2 cases of class = 2 (compared to 2 actual cases), and predicted 4 classifications of class = 3 (compared to 4 actual cases).

Naïve Bayes (Multinomial Input)	Prediction	Actual
0	1	1
1	2	2
2	2	2
3	4	4
Accuracy: 100.00%		

TABLE 16. Naïve Bayes with Multinomial Inputs on Soybean Data Set. Values indicate the total number of class assignments (both predicted and actual) during 5-fold cross-validation.

Overall Conclusions: Results show that each algorithm performed at 100% accuracy.

House Votes Data Set

Winnow-2 on House Votes Data Set: Winnow-2 predicted 26 Democrat class assignments (class = 0) and 20 Republican class assignments (class = 1). Although one could mistake these results for perfect accuracy, there were 2 prediction errors in the data set (out of 46 total predictions). For x_4, Winnow-2 predicted the class label as 0 (Democrat) when, in fact, the actual class label was 1 (Republican). For x_{33}, Winnow-2 predicted the class label as 1 (Republican) when, in fact, the actual class label was 0 (Democrat). Thus, Winnow-2 yielded 95.65% accuracy.

Winnow-2	Prediction	Actual
0	26	26
1	20	20
Accuracy: 95.65%		

TABLE 17. Winnow-2 results on House Votes Data Set. Values indicate the total number of class assignments (both predicted and actual) during 5-fold cross-validation.

Naïve Bayes (Discrete Boolean Inputs) on House Votes Data Set: Boolean Naïve Bayes predicted 26 Democrat class assignments (class = 0) and 20 Republican class assignments (class = 1; see Table 18). Although one could mistake these results for perfect accuracy, there were 4 prediction errors in the data set (out of 46 total predictions). For x_4, Boolean Naïve Bayes predicted the class label as 0 (Democrat) when, in fact, the actual class label was 1 (Republican). For x_{12}, Boolean Naïve Bayes predicted the class label as 1 (Republican) when, in fact, the actual class label was 0 (Democrat). For x_{29}, Boolean Naïve Bayes predicted the class label as 1 (Republican) when, in fact, the actual class label was 0 (Democrat). For x_{30}, Boolean Naïve Bayes predicted the class label as 0 (Democrat) when, in fact, the actual class label was 1 (Republican). Thus, Boolean Naïve Bayes yielded 91.30% accuracy.

Naïve Bayes (Boolean Input)	Prediction	Actual
0	26	26
1	20	20
Accuracy: 91.30%		

TABLE 18. Naïve Bayes with Boolean Inputs on House Votes Data Set. Values indicate the total number of class assignments (both predicted and actual) during 5-fold cross-validation.

Naïve Bayes (Multinomial Inputs) on House Votes Data Set: Multinomial Naïve Bayes predicted 26 Democrat class assignments (class = 0) and 20 Republican class assignments (class = 1; see Table 19). Although one could mistake these results for perfect accuracy, there were 4 prediction errors in the data set (out of 46 total predictions). For x_4, Multinomial Naïve Bayes predicted the class label as 0 (Democrat) when, in fact, the actual class label was 1 (Republican). For x_{12}, Multinomial Naïve Bayes predicted the class label as 1 (Republican) when, in fact, the actual class label was 0 (Democrat). For x_{29}, Multinomial Naïve Bayes predicted the class label as 1 (Republican) when, in fact, the actual class label was 0 (Democrat). For x_{30}, Multinomial Naïve Bayes predicted the class label as 0 (Democrat) when, in fact, the actual class label was 1 (Republican). Thus, Multinomial Naïve Bayes yielded 91.30% accuracy.

Naïve Bayes (Multinomial Input)	Prediction	Actual
0	26	26
1	20	20
Accuracy: 91.30%		

TABLE 19. Naïve Bayes with Multinomial Inputs on House Votes Data Set. Values indicate the total number of class assignments (both predicted and actual) during 5-fold cross-validation.

Overall Conclusions: Results show accuracy at 95.65% for Winnow-2, 91.30% for Boolean Naïve Bayes, and 91.30% for Multinomial Naïve Bayes. Overall, we see that Winnow-2 was slightly more effective than Naïve Bayes. This is not surprising since the data set consisted of binary/Boolean input.

Discussion

Overall, all three algorithms performed similarly. Winnow-2 slightly outperformed Naïve Bayes with Boolean input data, whereas Naïve Bayes performed slightly better with continuous data. It also appeared that the overall accuracy of predictions, across all algorithms, decreased as the number of class labels increased.

Conclusions

This paper implemented two supervised learning algorithms: Winnow-2 and Naïve Bayes. Both algorithms were utilized to solve classification problems using 5 data sets from the UCI Machine Learning Repository. For each data set, the prediction accuracy for each algorithm was compared. Although both algorithms performed equally well, Winnow-2 slightly outperformed Naïve Bayes with Boolean input data, whereas Naïve Bayes was capable of handling multivalued continuous data.

Future Work

Future analyses may want to compare other classification metrics (e.g., F1, sensitivity, specificity, false positive rate, recall, Mean Squared Error). And, learning curves may also provide using information regarding the bias/variance tradeoff.

References

Alpaydın, E. (2020). *Introduction to Machine Learning*. Cambridge, MA: MIT Press.

Congressional Quarterly Almanac (1984). 98th Congress, 2nd Session, 1984, Volume XL: Congressional Quarterly Inc., Washington, D.C. Retrieved June 8, 2020 from https://archive.ics.uci.edu/ml/datasets/Congressional+Voting+Records

German, B. (1987, September 1). Glass Identification Data Set. Retrieved June 8, 2020, from https://archive.ics.uci.edu/ml/datasets/Glass+Identification

Hand, D. J., & Yu, K. (2001). Idiot's bayes - not so stupid after all. International Statistical Review, 69(3):385-399. ISSN 0306-7734.

Fisher, R. A. (1988). Iris Data Set. Retrieved June 8, 2020, from https://archive.ics.uci.edu/ml/datasets/Iris

Littlestone, N. (1988). Learning quickly when irrelevant attributed abound: A new linear-threshold algorithm. *Machine Learning, 2*, 285-318. URL: https://link.springer.com/content/pdf/10.1023/A:1022869011914.pdf

Michalski, R. S. (1987, January 1). Soybean (Small) Data Set. Retrieved June 8, 2020, from https://archive.ics.uci.edu/ml/datasets/Soybean+%28Small%29

Zhang, H. (2004). The optimality of naïve bayes. FLAIRS2004 conference. (available online: PDF (http://www.cs.unb.ca/profs/hzhang/ publications/FLAIRS04ZhangH.pdf))

Wolberg, W. (1992, July 15). Breast Cancer Wisconsin (Original) Data Set. Retrieved June 8, 2020, from https://archive.ics.uci.edu/ml/datasets/Breast+Cancer+Wisconsin+%28Original%29

Wolberg, W. H., & Mangasarian, O. L. (1990). Multisurface method of pattern separation for medical diagnosis applied to breast cytology. Proceedings of the National Academy of Sciences, 87, 9193-9196.