Programming Project 3: Classification and Regression Decision Trees

Ricca D. Callis

July 6th, 2020

Abstract

This project sought to implement two different nonparametric Supervised Machine Learning Decision Tree Algorithms, Iterative Dichotomiser 3 (ID3) and Classification and Regression Tree (CART). The ID3 algorithm was applied to 3 classification problems, using gain-ratio as the splitting criterion. Additionally, ID3 utilized post-fit pruning in order to prevent overfitting. The CART algorithm was applied to 3 regression problems, using mean squared error as the splitting criterion. Additionally, CART utilized a validation set was used to tune an early stopping parameter in order to prevent overfitting. All Decision Trees were applied to input data obtained from the UCI Machine Learning Repository.

Introduction

This project provided students enrolled in an Introduction to Machine Learning course (605.649.83.SU20), at Johns Hopkins University, the opportunity to implement Decision Tree Classification and Regression algorithms. Decision Trees are nonparametric supervised machine learning techniques which partition a feature space in order to explain a target variable (Alpaydin, 2020; Mitchell, 2013; Quinlan, 1986). Decision Trees create a training model used to predict the class or value of a target variable by learning simple decision rules inferred from prior (i.e., training) data. Thus, Decision Trees can be applied to both classification and regression problems and will be differentiated by the type of target variable: categorical or continuous.

Decision Trees use a tree-like structure to represent classifications of data instances, where leaves become graphical nodes (or attributes) and branches become decision splits (e.g., Mitchell, 2013). The topmost part of the tree is represented by the root node, which includes the entire population or sample of data. Since each attribute can take on a set of values, during the training of a decision tree, data instances are split into subtrees (or subsets), based on the values of the attributes. Data is continuously split into smaller and smaller subsets (recursively) until each subset consists of a single class label. Thus, instances are classified by sorting down the tree from the root to some leaf node, which provides the final classification of the instance. Thus, each node in the tree specifies a test of some attribute of the instance, and each descending branch corresponds to one of the possible values for the attribute. For a class, every branch from the root of the tree to a leaf node having the same class is conjunction (product) of values, different branches ending in that class form a disjunction (sum). This allows us to classify new test instances based on the rules defined by the Decision Tree.

Algorithms and Experimental Methods

Iterative Dichotomizer 3 (ID3)

The Iterative Dichotomizer 3 (ID3) algorithm is a top-down greedy search algorithm used for classification problems (Quinlan, 1986). Given a set of features $X = (x_1, \dots, x_k)$, where x_i is a vector of *n* observations of feature *i,* and a target variable y, composed of the class labels for the *n* observations, the algorithm proceeds as follows:

1. For each feature *i*, find the optimal splitting points.
2. Partition data based on optimal splitting point
3. Repeat process on each partition
4. Continue recursively until all of the *y* observations in a given leaf node are of the same class, the *y* observations cannot be further split, or until an early stopping threshold is triggered.

We see that the process begins with the root node, where each instance attribute is evaluated using a statistical test to determine how well it alone classifies the training examples. The best attribute is selected and used as the test at the root node of the tree. A descendant of the root node is then created for each possible value of this attribute, and the training examples are sorted to the appropriate descendant node. As a top-down greedy search algorithm, ID3 never backtracks to reconsider earlier choices.

Discrete-Valued Features. If feature *i* is a discrete-value, there is only one split to consider, and each of the values gets its own split. An optimally-chosen split partitions X and y into rows where $x_i = c$, for each of the *c* possible values of x_i.

Continuous-Valued Features. If feature *i* is a continuous-value, all possible splits are considered and each of the values gets its own split. To find the optimal split point, the feature x_i is sorted, unique midpoints between feature rows are calculated, and each midpoint is tested as a possible split point (except midpoints between two identical values of *y*). After testing each midpoint, the most optimal is selected. An optimally-chosen split partitions X and y into rows where $x_i > c$, where *c* is the optimal spit point.

Entropy. ID3 uses entropy to measure the heterogeneity of the data set. Given a set of *y* values, entropy is defined as:

$$I(y) = -\sum_{i=1}^{C} \frac{C_i}{n} log \frac{C_i}{n}$$

Where,

$$C = number\ of\ unique\ values\ in\ y$$

$$C_i = number\ of\ points\ in\ y\ that\ are\ in\ class\ i$$

$$n = number\ of\ points\ in\ y$$

The entropy of a set of values *y* will be largest when each class has the same number of elements in *y.* Thus, the optimal split will minimize entropy by selecting a split that best separates the classes in *y* at each step.

Information Gain. The expected information gain from split f_i is defined as:

$$E(f_i) = \sum_{j=1}^{m_i} \frac{C_{\pi,1}^j + \cdots + C_{\pi,k}^j}{C_{\pi,i} + \cdots + C_{\pi,k}} I\left(C_{\pi,1}^j, \ldots, C_{\pi,k}^j\right)$$

Where,

$$m_i = number\ of\ partitions\ created\ by\ split\ f_i$$

Here we see that for each partition, the ratio of total points included within the partition is multiplied by the entropy of the partition. This creates a weighted average entropy calculation.

Intrinsic Value. The intrinsic value of a feature is defined as:

$$IV(f_i) = -\sum_{j=1}^{m_i} \frac{C_{i,1} + \cdots + C_{i,k}}{C_1 + \cdots + C_k} \log \frac{C_{i,1} + \cdots + C_{i,k}}{C_1 + \cdots + C_k}$$

Where,

$$m_i = number\ of\ partitions\ created\ by\ split\ f_i$$

$$C_{i,k} = number\ of\ points\ in\ class\ C_k\ in\ partion\ i$$

Gain Ratio. Gain Ratio is used to select the optimal split f_i^*, and is defined as:

$$f_i^* = argmax_{f_i} \frac{E(f_i)}{IV(f_i)}$$

Pruning. To prevent overfitting data, Post-Pruning is implemented whereby any subtree which does not create a reduction in entropy is removed from the tree. To do this, a validation set is created separate from the train and test sets. The Decision Tree is fit to completion on the training set and then performance on the validation set is calculated over the entire tire. Beginning with the root node, each of the children is collapsed from a subtree to a leaf node. Prediction is made over the entire tree, with one node truncated into a leaf. If performance on the validation set improves relative to the original tree, the leaf remains. Otherwise, it is swapped back to the original node. This process continues recursively over all the subtrees, and continues iteratively from the root node until no changes are made over the entire tree.

Classification and Regression Tree Algorithm (CART)

The Classification and Regression Tree (CART) algorithm is also top-down greedy algorithm but can be modified to apply to both classification and regression problems by dividing the space via recursive binary splitting (e.g., Friedman, 1977; Breiman, et al., 1984). Unlike ID3, CART is represented as a binary tree and it replaces entropy with Mean Squared Error (MSE), which is defined as:

$$MSE(y) = \frac{1}{n}\sum_{i=1}^{n}(y - \bar{y})^2$$

Where,

$$\bar{y} = mean\ of\ y$$

We see that the above attribute selection metric attempts to minimize the variance of the target variables in each node. For this project, discrete multi-valued attributes were one-hot encoded and then treated as continuous splits.

Early Stopping Parameter. Due to the fact that CART recursively fits the data, it is also prone to overfitting. As a result, a criterion is utilized to indicate when the recursive binary splitting procedure should stop. Here, given an early-stopping parameter θ, fitting will proceed until either no further splits can be made, or until the gain associated with a split is less than θ. The larger θ, the shorter the tree. This project utilized a holdout validation set in order to determine the best value of θ.

Data Sets

This analysis was conducted on 6 data sets, each obtained from the UCI Machine Learning Repository:

(1) Abalone Data Set

(2) Car Evaluation Data Set

(3) Image Segmentation Data Set

(4) Computer Hardware Data Set

(5) Forest Fires Data Set

(6) Wine Quality Data Set

For each data set, descriptive statistics were calculated for all features and for each feature grouped by class label. For all classification experiments, both continuous and discrete features were handled by the ID3 algorithm using gain ratio as the splitting criterion. To prevent overfitting, post-pruning set aside 10% of the dataset for a validation set. Five-fold stratified cross-validation was used to estimate the out-of-sample performance on various subsets of the data.

As previously mentioned, for all regression experiments, discrete values were mapped into one-hot encoded dummy values, ensuring binary splits at each step. To prevent overfitting, 10% of the dataset was used as a validation set and the stopping threshold was tuned on this validation set. Thresholds set to zero indicated no early stopping. Five-fold stratified cross-validation was used to estimate the out-of-sample performance on various subsets of the data.

Abalone Data Set

Data Description: Classifies the age of an abalone, based on 8 feature attributes: sex, length (of shell, measured in in mm), diameter (perpendicular to length, measured in mm), height (with meat in shell, measured in mm), whole weight (measured in grams), shucked weight (weight of meat, measured in grams), viscera weight (gut weight after bleeding, measured in grams), and shell weight (weight after being dried, measured in grams; Waugh1995). The class is the number of rings on the abalone shell, used to measure age (number of rings + 1.5 = age in years). This is a large multivariate data set with 4177 instances, where each attribute's instance is represented as a continuous-value float.

Data Cleaning & Transformation: There was no missing data to handle.

Exploratory Data Analysis: The continuous-valued target variable (number of rings) was slightly skewed (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).

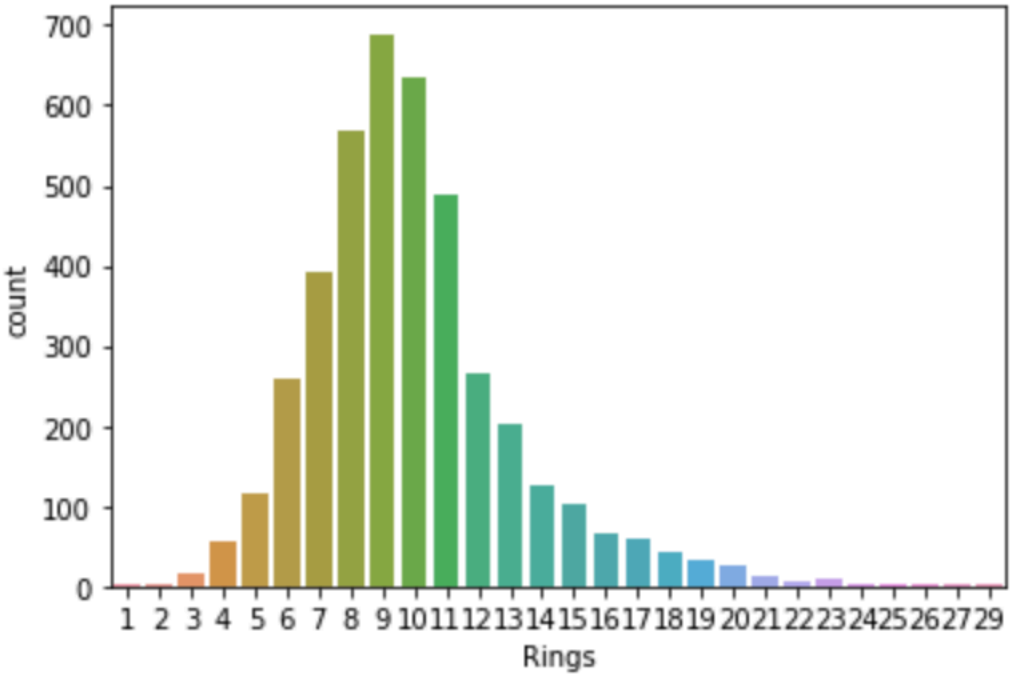

FIGURE 1. Abalone data set plot showing the distribution of the target value (number of rings).

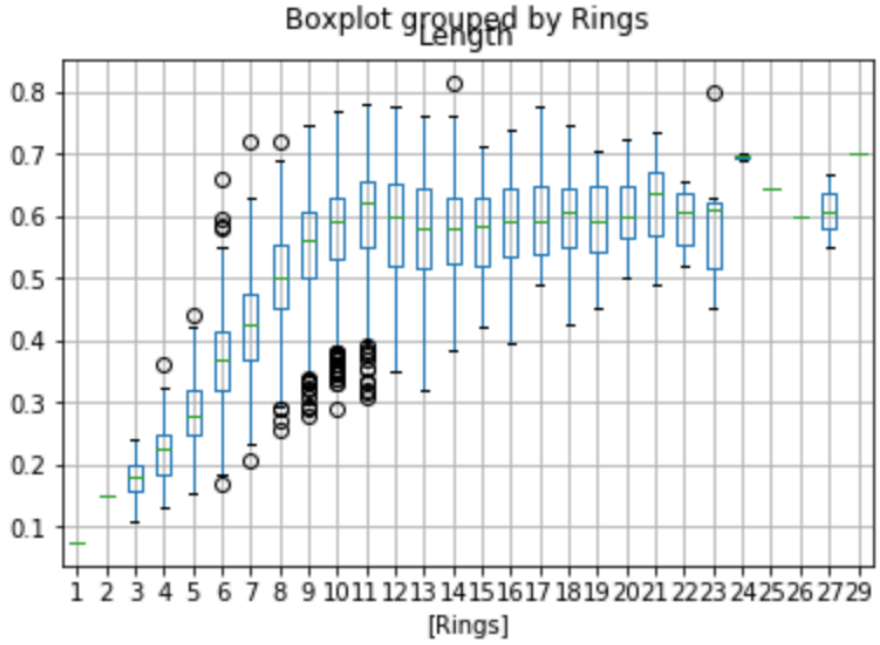

FIGURE 2. Abalone data set boxplot showing attribute grouped by class. Above example depicts the feature length by class label number of rings.

Car Evaluation Data Set

Data Description: Classifies the evaluation of a car, based on 6 feature attributes: buying (i.e., purchase price), maint (i.e., maintenance cost), doors (i.e., number of doors: 2, 3, 4, or 5+), persons (i.e., maximum passenger capacity: 2, 4, or more), lug boot (i.e., the size of the luggage boot: either small, medium, or big), and safety (either low, medium, or high; Bohanec & Zupan, 1997). The class is the categorical variable 'acceptable', indicating whether the car is evaluated as either unacceptable, acceptable, good, or very good. This is a multivariate data set with 1728 instances, where each attribute's instance is represented as a categorical variable and each class label is also represented as a discrete-value categorical variable.

Data Cleaning & Transformation: There was no missing data to handle. The attribute id was also dropped from the data set, as it represented a unique identifier that would not serve to teach class attributes.

Exploratory Data Analysis: There were 4 class labels with the following assignment frequencies: 1210 unaccepable, 384 accepable, 69 good, 65 very good (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).

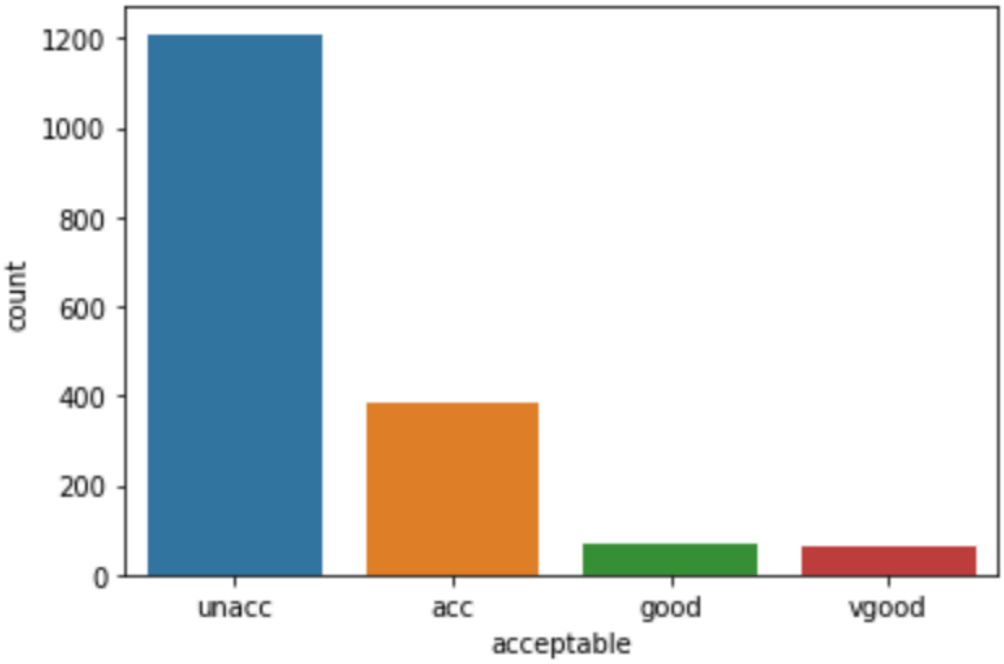

FIGURE 3. Car Evaluation data set plot showing the actual raw count values for the acceptable classification.

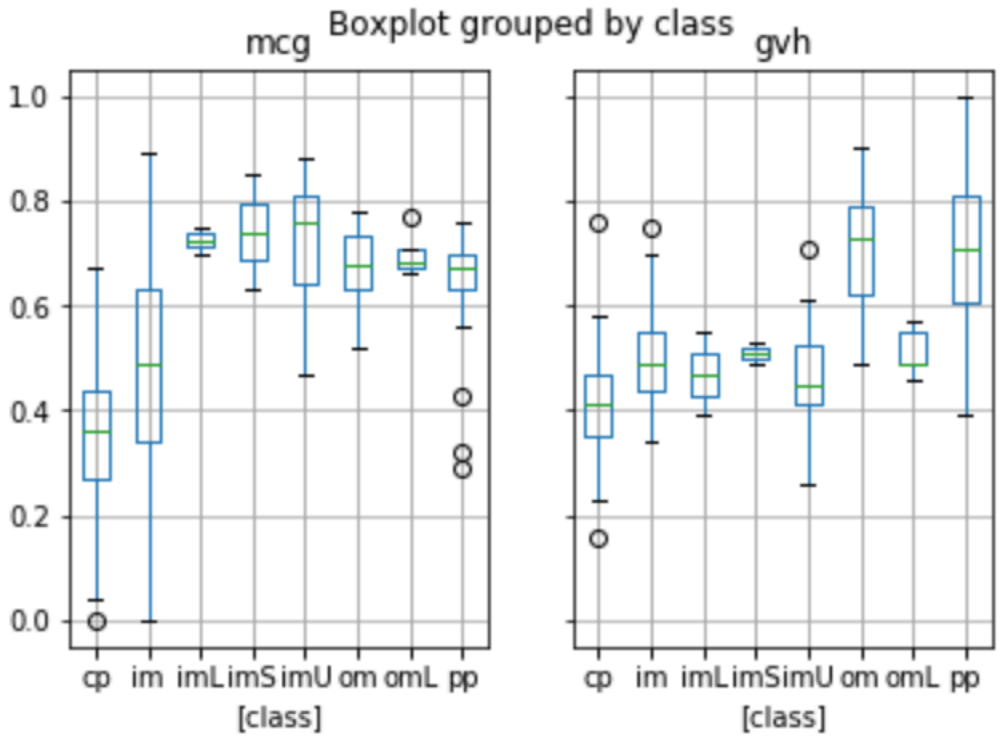

FIGURE 4. Car Evaluation data set boxplot showing attribute grouped by class. Above example depicts the features mcg and gvh by class label assignment.

Image Segmentation Set

Data Description: Classifies part of an image based on a given pixel, based on 19 feature attributes of pixels: region-centroid-col, region-centroid-row, region-pixel-count, short-line-density-5, short-line-density-2, vedge-mean, vedge-sd, hedge-mean, hedge-sd, intensity-mean, rawred-mean, rawblue-mean, rawgreen-mean, exred-mean, exblue-mean, exgreen-mean, value-mean, saturation-mean, and hue-mean (Vision Group, 1990). The class attribute has 7 labels (representing different pictures), each with 30 instances. This is a multivariate data set with 210 instances, where each attribute's instance is represented as a continuous value float and each class label is represented as a discrete-value integer using one-hot encoding.

Data Cleaning & Transformation: There were no missing data. The attribute region-pixel-count was also dropped from the data set, as it had no variance.

Exploratory Data Analysis: There were 30 instances for each class label (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 a few of the feature by class can be observed in Table 1.

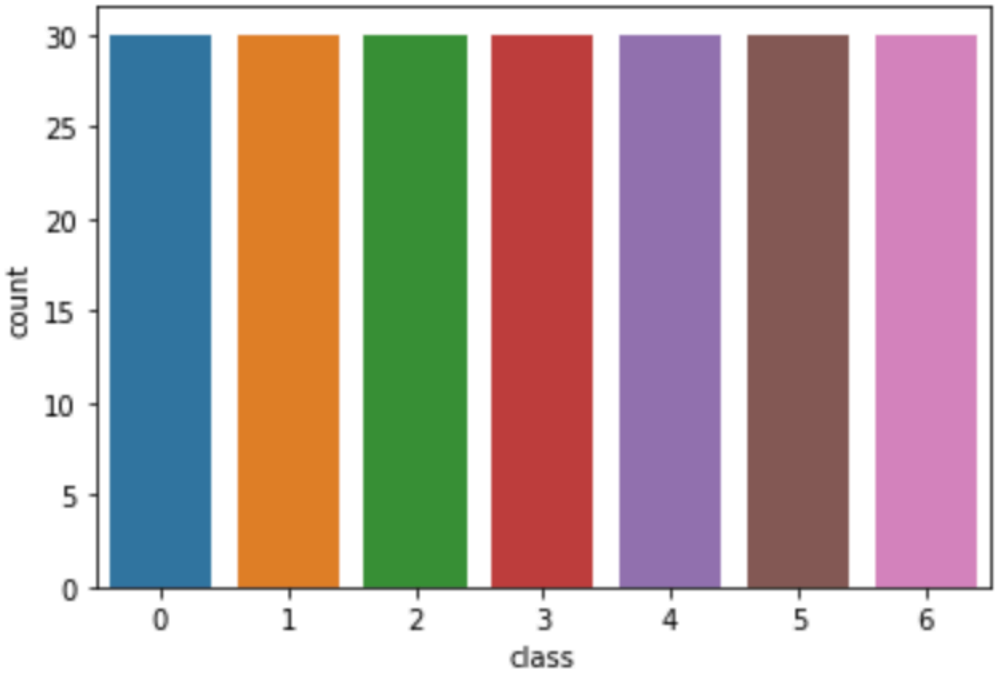

FIGURE 5. Image Segmentation data set plot showing the actual raw count values for image classification (where each classification represents a different picture).

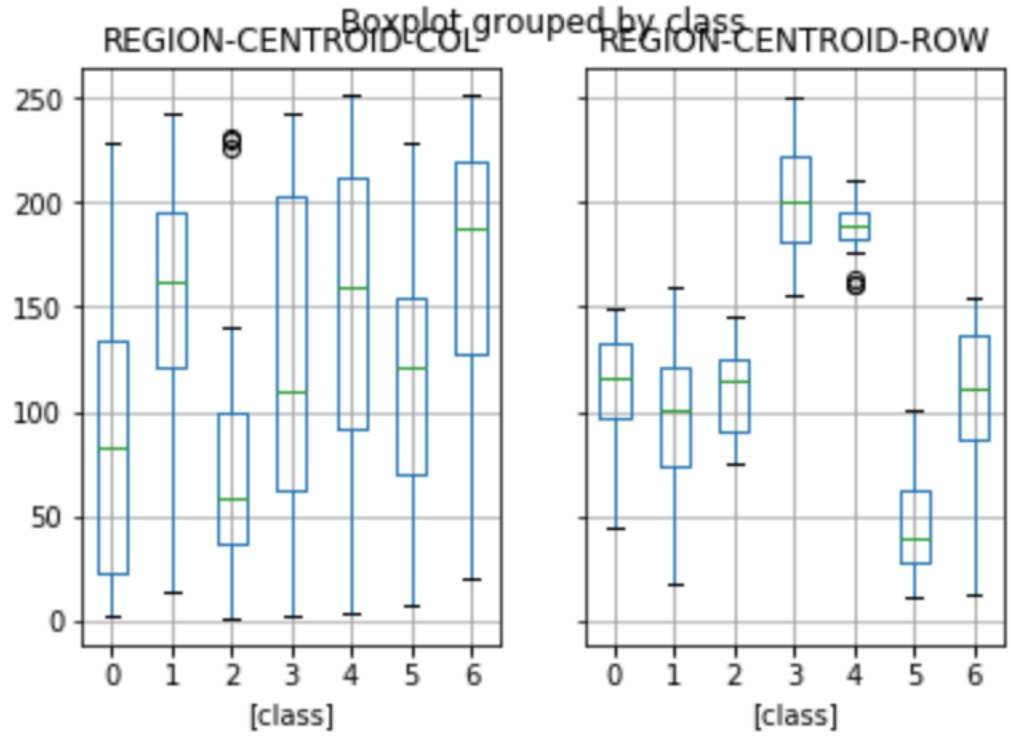

FIGURE 6. Image Segmentation data set boxplot showing attribute grouped by class. Above example depicts the features region-centroi-col and region-centroid-row by class (where each class label represents a different image/picture).

Class Labels	**Attributes/Features**								
	Region-Centroid-Col	**Region-Centroid-Row**	**Short-Line-Density5**	**Short-Line-Density-2**	**Vedge-Mean**	**Vedge-SD**	**Hedge-Mean**	**Hedge-SD**	**Intensity-Mean**
0	83.4000	109.333	0.0037	0.0000	1.03703	1.03086	1.33703	0.85160	13.16543
1	150.5000	97.16666	0.00740	0.00740	2.95185	2.43526	2.56481	3.79208	43.54691
2	76.50000	111.4000	0.00370	0.01481	3.82777	30.9057	5.29074	58.9660	10.99259
3	130.7000	203.5000	0.02592	0.00000	1.50740	1.97301	2.14259	2.06424	14.97777
4	150.1666	187.2333	0.01111	0.02222	2.40000	2.11818	4.62037	10.0159	49.49135
5	116.4000	45.86666	0.00740	0.00000	0.83148	0.58065	1.13703	0.79859	119.069
6	164.866667	104.80000	0.00000	0.00000	0.92037	0.9929	1.1370	4.9801	8.39382

TABLE 1. Mean values of each feature by image classification in the Image Segmentation Data Set.

Computer Hardware Data Set

Data Description: Predicts the performance of a given CPU, based on 9 feature attributes: vendor_name, model_name, myct, mmin, mmax, cash, chmin, chmax, prp, and erp (Ein-Dor & Feldmesser, 1987). The class attribute has 29 labels: adviser, amdahl,apollo, basf, bti, burroughs, c.r.d, cambex, cdc, dec, dg, formation, four-phase, gould, honeywell, hp, ibm, ipl, magnuson, microdata, nas, ncr, nixdorf, perkin-elmer, prime, siemens, sperry, sratus, and wang. This is a multivariate data set with 209 instances, where each attribute's instance is represented as a continuous value float.

Exploratory Data Analysis: There were 32 instances of imb, 19 instances of nas, 13 instances of sperry, 13 instances of Honeywell, 13 instances of ncr, 12 instances of siemens, 9 instances of cdc, 9 instances of amdahl, 8 instances of burroughs, 7 instances of hp, 7 instances of dg, 7 instances of harris, 6 instances of dec, 6 instances of c.r.d., 6 instances of magnuson, 6 instances of ipl, 5 instances of formation, 5 instances of cambex, 5 instances of prime, 3 instances of gould, 3 instances of perkin-elmer, 3 instances of Nixdorf, 2 instances of basf, 2 instances of wang, 2 instances of Apollo, 2 instances of bti, 1 instance of sratus, 1 instance of four-phase, 1 instance of microdata, and 1 instance of adviser (see Figure 7). Each feature was also described by area (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). A sample of the means for a select few of the attributes and class labels can be observed in Table 2.

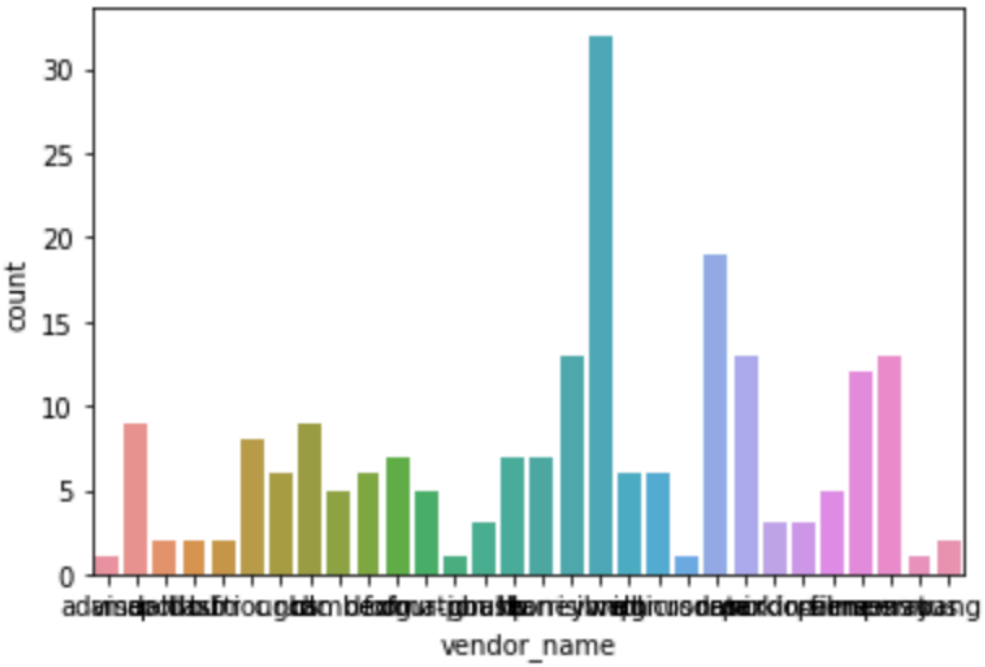

FIGURE 7. Computer Hardware data set plot showing the actual raw count values for vendor name classification.

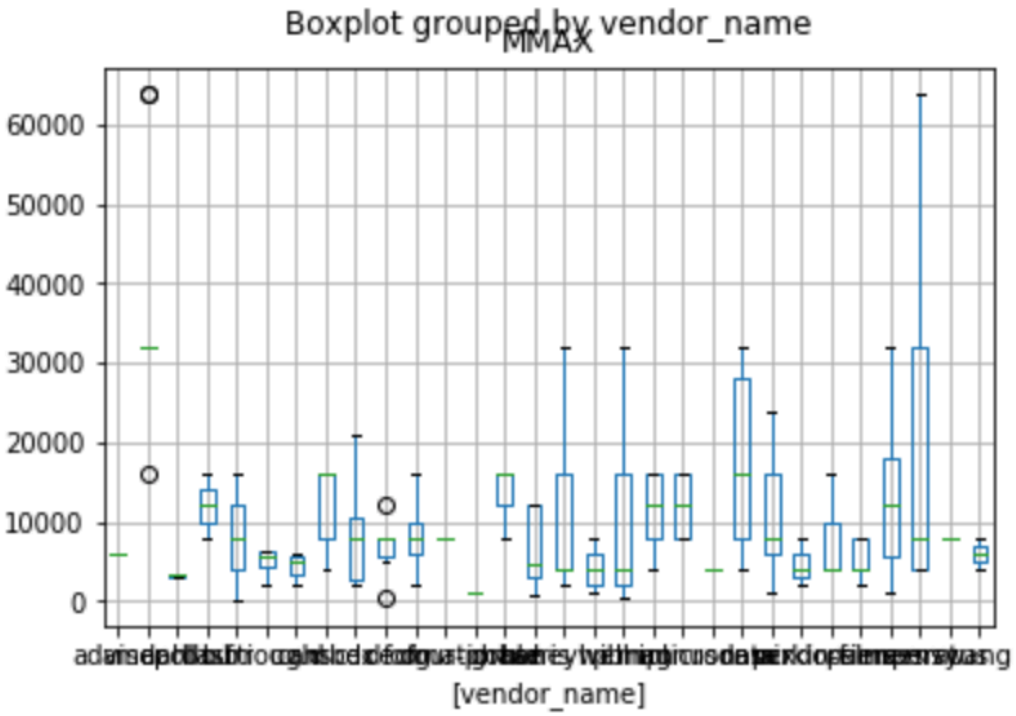

FIGURE 8. Computer Hardware data set boxplot showing attribute grouped by class. Above example depicts the feature MMAX by class (vendor namne).

Class Labels	**Attributes/Features**			
	MYCT	MMIN	MMAX	CASH
Adviser	125.000000	256.000000	6000.000000	256.000000
Amdal	26.000000	13333.333333	37333.333333	56.888889
Apollo	400.000000	756.000000	3250.000000	2.000000

TABLE 2. Mean values of each a few features by a few vendor names (class labels) in the Computer Hardware Data Set.

Forest Fires Data Set

Data Description: Predicts a forest fire's burn area based on 10 feature attribute: month, day, ffmc, dmc, dc, isi, temp, rh, wind, and rain (Cortez & Morais, 2008). The target-value is a continuous-float variable.

Exploratory Data Analysis: Each feature was described by area (descriptive statistics included mean, standard deviation, minimum, maximum, range, 1st quartile, median, and 3rd quartile).

Wine Quality Data Set

Data Description: Classifies the quality of a wine, based on 11 feature attributes: fixed acidity, volatile acidity, citric acid, residual sugar, chlorides, free sulfur dioxide, total sulfur dioxide, density, pH, sulphates, and alcohol (Cortez, 2007). The class is a wine-quality score, based on sensory data, ranging from 0 to 10. There were two relevant datasets: one containing red wines and one containing white wines. The two datasets were combined and one-hot encoding was used to indicate whether a given bottle was a white-wine or a red-wine. The raining attribute instances were represented as continuous-value floats.

Data Cleaning & Transformation: There was no missing data to handle.

Exploratory Data Analysis: There were 7 class labels with the following assignment frequencies: 2836 instances for quality rating of 6, 2138 instances for quality rating of 5, 1079 instances for quality rating of 7, 216 instances for quality rating of 4, 193 instances for quality rating of 8, 30 instances for quality rating of 3, and 5 instances for quality rating of 5 (see Figure 9).

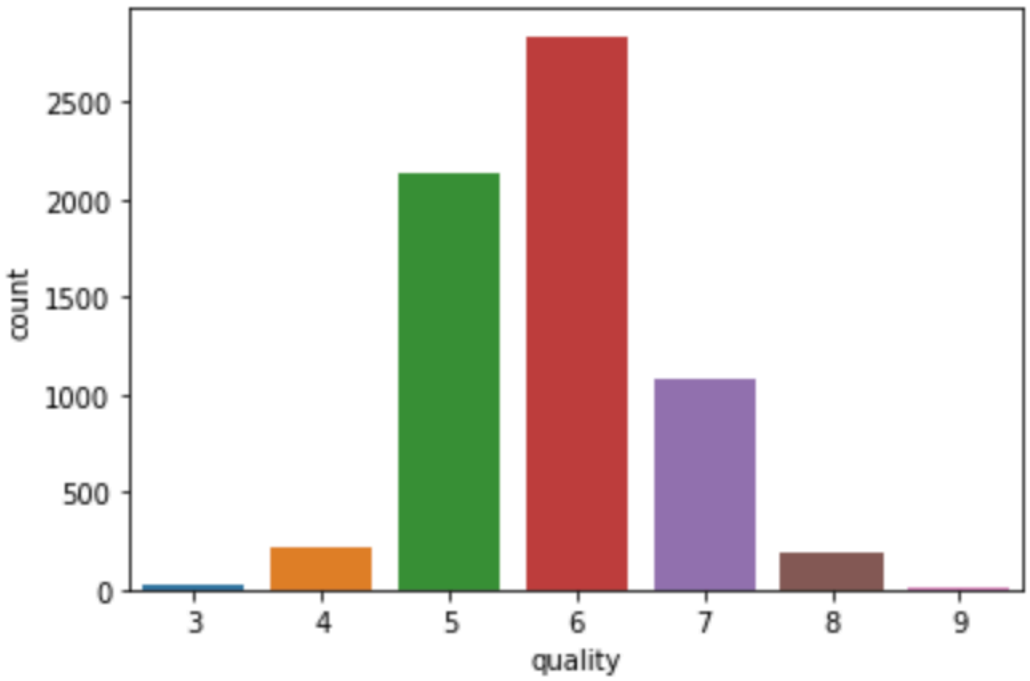

FIGURE 9. Wine Quality data set plot showing the actual raw count values for wine-quality classification.

Results

Abalone Data Set

The Abalone data set represents a classification problem used to classify age of abalone shells (with the target-variable number of rings) using 8 feature attributes. The ID3 algorithm was applied to this data set using gain ratio as the splitting criterion and results are shown in Table 3.

Fold	Accuracy
1	24.80%
2	23.16%
3	22.93%
4	21.56%
5	24.74%

TABLE 3. ID3 Decision Tree classification results on Abalone Data Set. The classification accuracy is displayed across the 5-fold cross-validation. Gain Ratio was used as the splitting criterion.

Car Evaluation Data Set

The Car Evaluation data set represents a classification problem used to classify the suitability of a car, given various features about the car. The ID3 algorithm was applied to this data set using gain ratio as the splitting criterion and results are shown in Table 4.

Fold	Accuracy
1	93.23%
2	94.52%
3	93.25%
4	95.19%
5	94.25%

TABLE 4. ID3 Decision Tree classification results on Car Evaluation Data Set. The classification accuracy is displayed across the 5-fold cross-validation. Gain Ratio was used as the splitting criterion.

Image Segmentation Data Set

The Image Segmentation data set represents a classification problem used to classify images based on pixel attributes. The ID3 algorithm was applied to this data set using gain ratio as the splitting criterion and results are shown in Table 5.

Fold	Accuracy
1	88.57%
2	88.89%
3	94.74%
4	82.05%
5	85.37%

TABLE 5. ID3 Decision Tree classification results on Car Evaluation Data Set. The classification accuracy is displayed across the 5-fold cross-validation. Gain Ratio was used as the splitting criterion.

Computer Hardware Data Set

The Computer Hardware data set represents a regression problem used to predict the relative performance given a set of CPU characteristics. The CART algorithm was applied to this data set using mean squared error (MSE) as the splitting criterion and results are shown in Table 6. To prevent overfitting, 10% of the dataset was used as a validation set and the stopping threshold was tuned on this validation set. The early stopping threshold was calculated and set at 25435.44.

Fold	MSE
1	380.45
2	174.24
3	260.18
4	4636.65
5	25.98

TABLE 6. CART Decision Tree regression results on Computer Hardware Data Set. Regression Mean Squared Error (MSE) was used as the splitting criterion and is displayed across the 5-fold cross-validation.

Forest Fires Data Set

The Forest Fires data set represents a regression problem used to predict the burn area of a forest fire given a set of feature characteristics. The CART algorithm was applied to this data set using mean squared error (MSE) as the splitting criterion and results are shown in Table 7. Month and Day attributes were one-hot encoded. To prevent overfitting, 10% of the dataset was used as a validation set and the stopping threshold was tuned on this validation set. The early stopping threshold was calculated and set at 1875.0.

Fold	MSE
1	369.42
2	712.29
3	6192.63
4	1818.19
5	12757.43

TABLE 7. CART Decision Tree regression results on Forest Fire Data Set. Regression Mean Squared Error (MSE) was used as the splitting criterion and is displayed across the 5-fold cross-validation.

Wine Quality Data Set

The Wine Quality data set represents a regression problem used to predict the quality of a wine given a set of feature characteristics. The CART algorithm was applied to this data set using mean squared error (MSE) as the splitting criterion and results are shown in Table 8. To prevent overfitting, 10% of the dataset was used as a validation set and the stopping threshold was tuned on this validation set. The early stopping threshold was calculated and set at 0.2.

Fold	MSE
1	0.67
2	0.65
3	0.61
4	0.63
5	0.65

TABLE 8. CART Decision Tree regression results on Wine Quality Data Set. Regression Mean Squared Error (MSE) was used as the splitting criterion and is displayed across the 5-fold cross-validation.

Conclusions

As nonparametric supervised machine learning algorithms, both ID3 and CART performed reasonably well on the classification and regression experiments conducted for this project. Decision Trees have a large preference bias due to their top-down greedy split approach. While this is computationally efficient, it may also mean that the algorithms presented in this project miss interaction effects. Future work may wish to examine this more closely.

Results indicate Decision Trees using that out-of-sample datasets had poorer performance. This indicates an estimation problem, more likely to occur with large, multivalued discrete feature data sets.

We see that the Decision Tree performed well on datasets with multiple irrelevant training features (e.g., the forest fires and wine quality data set). In Project 2, the KNN algorithms had reduced performance on these same datasets.

References

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

Bohanec, M., & Zupan, B. (1997, June 1). Car Evaluation Data Set. Retrieved July 6, 2020, from https://archive.ics.uci.edu/ml/datasets/Car+Evaluation

Breiman, L., Friedman, J. H., Olshen, R. A., & Stone, P. 1. (1984). Classification and regression trees. Belmont, CA: Wadsworth International Group.

Cortez, P. (2009, October 7). Wine Quality Data Set. Retrieved July 6, 2020, from https://archive.ics.uci.edu/ml/datasets/Wine+Quality

Cortez, P., & Morais, A. (2008, February 29). Forest Fires Data Set. Retrieved July 6, 2020 from https://archive.ics.uci.edu/ml/datasets/Forest+Fires

Ein-Dor, P., & Feldmesser, J. (1987, October 01). Computer Hardware Data Set. Retrieved July 6, 2020 from https://archive.ics.uci.edu/ml/datasets/Computer+Hardware

Friedman, J. H. (1977). A recursive partitioning decision rule for non-parametric classification. IEEE Transactions on Computers, 404-408.

Nash, W. J., Sellers, T. L., Talbot, S. R., Crawthorn, A. J., & Ford, W. B. (1994). The Population Biology of Abalone (_Haliotis_ species) in Tasmania. I. Blacklip Abalone (_H. rubra_) from the North Coast and Islands of Bass Strait. *Sea Fisheries Division, 48*.

Mitchell, T. M. (2013). *Machine learning*. New York: McGraw-Hill.

Quinlan, J. R. (1986). Induction of decision trees. *Machine Learning, 1*, 81-106.

Russell, S. J., & Norvig, P. (2010). *Artificial intelligence a modern approach*. Upper Saddle River (N.J.): Pearson.

Vision Group (1990, September 01). Image Segmentation Data Set. Retrieved July 6, 2020 from https://archive.ics.uci.edu/ml/datasets/Image+Segmentation

Waugh, S. (1995, December 01). Abalone Data Set. Retrieved July 6, 2020, from https://archive.ics.uci.edu/ml/datasets/Abalone