Supervised Learning with ID3 Decision Tree

Introduction

The advent of computer technology and the degree in the Informatization of computer network infrastructure, improves and increases the willingness of people to use information systems for the collection and handling of information and data. In these data, the detailed Information and awareness are tacit, which people do not realize in moving ahead, but theoretically useful. Currently, the decision tree has become an essential form of data exploration. The decision tree classification algorithm is one of the most common approaches in the field of information and data mining advancement. ID3 algorithm, as a heuristic estimate, was suggested in 1986 by Quinlan, acclaimed and well regarded in the field of building the trees for decision making. In the decision trees, a model is built based on existing data for which the target feature values are known to predict the target feature of an unknown query case (Rajeshkanna & Arunesh, 2020). In addition, we understand that this model can forecast unknown query instances because it models the relationship between known descriptive characteristics. The theory of the ID3 algorithm is to pick and assign the attribute with the most knowledge or information gain (i.e. the attribute that mostly contributes to the classifier) based on entropy as the current grouping characteristic, and then recursively expand the selected tree sections until the entire tree has been fully generated. In the decision tree methodology, the path to data gain is typically used for evaluating an effective property for each node of the tree being created. We may then pick an attribute with the greatest information gain (the reduction of entropy in the level of maximum information). To measure the uncertainty of a dataset, the entropy of the dataset is used and this sort of create a measure of impurity in the dataset. There are also some kinds of metrics that can be used to assess the collection of data. The most popular are the following: Gini score, Chi-Square, ratio of information gain, variance. The type which is used in this paper is the entropy or uncertainty which is given by the formula.

$$H(x) = -\sum_{for\ k\ \in target} (P(x = k) * log_2(P(x = k)))$$

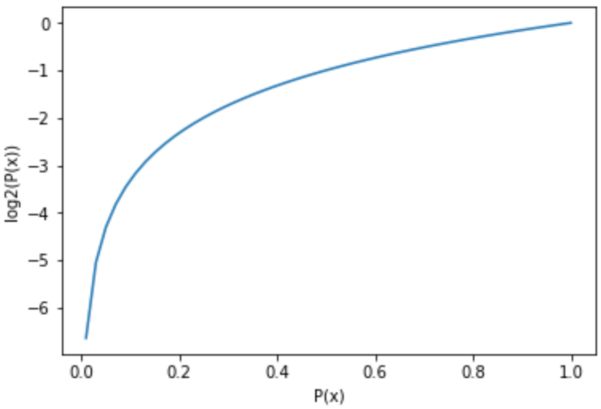

Figure 1: Entropy and Probability Representation

In simpler terms, the concept behind entropy is the following: Assume that you have a basket that contains 100 apples. Since only apples are used, the selection of fruit inside the basket can be said to be completely pure. In the language of entropy, this group of fruits has an entropy of 0 to express this (we can also say zero impurity). However, if 30 of these fruits are replaced by Oranges and 20 replaced by bananas and we now look at this basket, the probability of picking an Apple has reduced from 1.0 to 0.5. The purity decreased as the impurity increased, so the entropy increased as well. We may also assume that the more "impure" a dataset, the greater the entropy, and the less "impure" a dataset, the lower the entropy. The entropy model of Shannon uses the logarithm function (log2(P(x))) to quantify the entropy and thus the impurity of a dataset, since the greater the chance of a given outcome = P(x) (randomly drawing an apple from the basket), the closer the binary logarithm is to 1 (Ming, Wenying, & Xu). The id3 decision tree is built along attributes starting with attribute with the highest entropy and down to the decision or classification of each attributes, an illustration is shown in figure 2. In this paper, we examine the id3 algorithm on an iris data and a cancer data and we perform a cross-validation to determine how much generalization the algorithm got right.

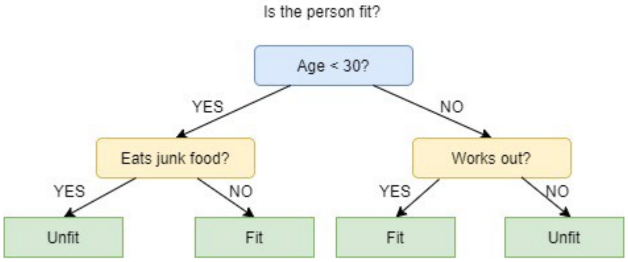

Figure 2: ID3 Decision Tree Representation

Methodology

There are two datasets used in this experiment obtained from the machine learning repositories (Dua & Graff, 2017). The first dataset is the Iris dataset. The data set contains 3 classes of 50 instances each, where each class refers to a type of iris plant. One class is linearly separable from the other 2; the latter are NOT linearly separable from each other. The attributes information are; sepal length in cm, sepal width in cm, petal length in cm 4, and petal width in cm. The classes are: -- Iris Setosa -- Iris Versicolour -- Iris Virginica
The second data set is the Cancer dataset which has a total of 104 information and 9 attributes which are classified into 6 different types of cancers. The attributes are; IO (Impedivity (ohm) at zero frequency) , PA500 (phase angle at 500 KHz), HFS (high-frequency slope of phase angle), DA (impedance distance between spectral ends), AREA (area under spectrum), A/DA (area normalized by DA), MAX IP (maximum of the spectrum DR distance between I0 and real part of the maximum frequency point), P (length of the spectral curve). The classes of cancers are; class car(carcinoma), fad (fibro-adenoma), mas (mastopathy), gla (glandular), con (connective), adi (adipose).

Algorithm Implementation

The implementation of this algorithm uses Python programming language and the Numpy library. The first method calculates the entropy using the probability distribution of the class as described in the background, we then iterate over the attributes on the dataset to calculate the entropy. In this case, the attributes contain continuous values which cannot be explicitly differentiated from the other. The example explained in the background uses a distinct name like apple, orange and banana. To calculate the entropy, we first sort the data with the attribute index, then split the data into two at a value. This value is determined by the number of values in the attribute index which are the same. We then take the average of the higest value in the first split data and the second split data so that the first data values are less than the split value and the other part of the data have values higher than the split value to create a node. The node will have a child that is either less than this value or greater. A recursive method is then called on both split data and the same operation is performed. The node implementation is a python class that has objects like the, "value", "decision", the "less" child, the "greater" child and the "column" number. To build the tree from the data, we will call the "train_model" method on the data, which runs the entropy process and build the tree based on the entropy information calculated from the data. To predict what class a data belongs, we call the "makeDecision" method which recursively iterate through the tree by checking the values at the column index from the tree of the data against the "value" object of the node. If it is less, the iteration will go through the "less" route and if its greater than this value, the iteration will go through the "greater" route. The algorithm returns a decision when it hits a dead end.

Cross Validation

Cross Validation in this experiment is a methodology used to ascertain that our data can categorize well and right and are not overfitted or underfitted; overfitted is when the data can classify the known data but cannot classify unknown data correctly while underfitted is not having enough data for generalization. We run the algorithm on the datasets by splitting the data into two which are tagged the training data and the testing dataset. The training dataset is used to build the tree and the testing dataset is used to test and make predictions. The datasets were split in 100 different ways so that every case of the split will be considered. This is due to that fact that the split will have large effect on the result, practically the entropy value will change if there are less group. We then calculate the **mean and standard error (stddev / sqrt(100))** of the percentage of testing examples correctly classified by the decision trees for each data set and the **False Positive Rate (FPR) (+/- 1.96*standard error)** for each iris species class (0,1,2) and for the cancer class (0).

Results and Discussion

Figure 3 is the Mean accuracy of the IRIS data set using different number of split datasets. In the experiment, we created the number of testing dataset from main dataset as shown in the x-axis of the figure. The mean accuracy shows how much each training dataset were able to categorize the testing dataset. The Classification did worst in the case where the testing data was 149 and the training data was only 1 and did equally well with insignificant difference in cases where the test sizes are 1, 10, 25, 50 and 75. We can conclude in this case that, we are able to predict correctly at an high percentage when the training data is either 149, 140, 125, 100 and 75. However to avoid overfitting as described earlier, we will choose to go with a training data of 100 and testing data of 50.

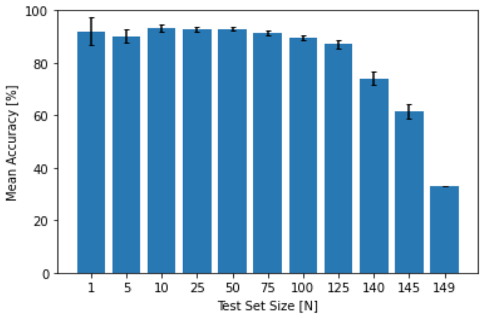

Figure 3: Mean Accuracy for the IRIS data Set

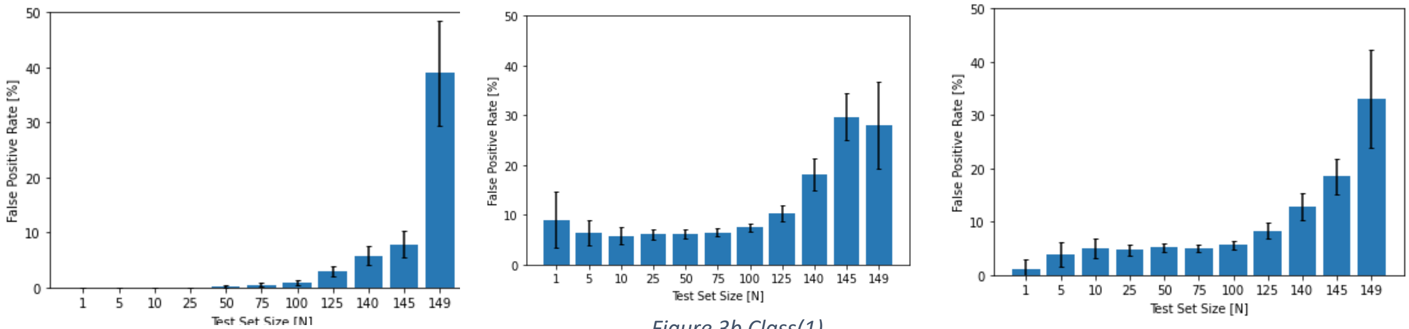

Figure 3a Class(0)

Figure 3b Class(1)

Figure 3c Class(2)

Figure 4: FPR for classes (0, 1, 2) of the IRIS Dataset

The False Positive Rate (FPR) is the rate at which the model classifies the testing data incorrectly. Looking at the figures in figure 3 for each classification, the model with a testing size of 1, 5, 10, 25, 50, and 75 set classifies close to 5 percent of the 100-sampling size incorrectly. This is degree to which the model can generalize unknown data. 95 percent of the time, the model can generalize. We then conclude that a training size of 100 will be enough to generalize in the decision tree for the IRIS data.

In figure 5, the mean accuracy of the cancer dataset is plotted using a different number of the testing data as shown in the X-axis. The accuracy of the model when the training dataset is 103 and the testing data is 1 was very small at less than 60 percent while the accuracy when the training data is 100 and the testing data is 5 was closer to 60 percent. This is an indication of how complex the attribute information of the dataset is and the number of datasets may not be enough to categorize the type of cancer. The accuracy of the decision tree may not be more than 60percent at generalizing unknown parameters.

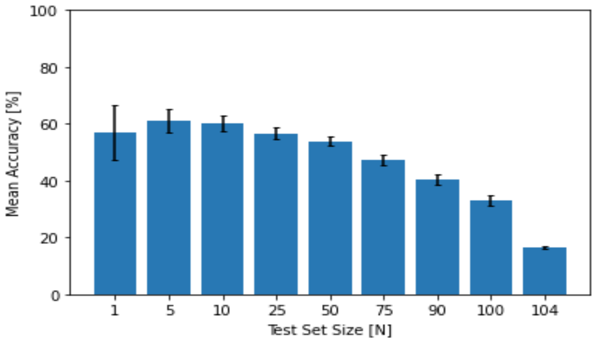

Figure 3: Mean Accuracy for Cancer Data Set

Figure 6 shows the False positive Rate of the Cancer data for class (0). It is rate at which the model predicts incorrectly the car(carcinoma) class.

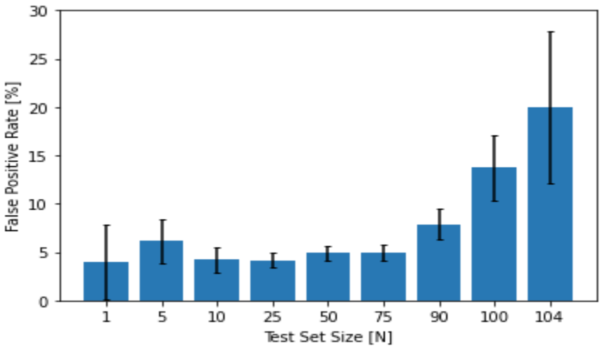

Figure 4: FPR for Cancer data class (0)

The training set for the cancer data are given in the figure and the FPR were calculated. At 10 and 25 number of the training data, the decision tree predicted less than 5% incorrectly which is similar to the case where the testing data is only 1 and the training data is 104. We may conclude that 79 sets of the data is enough to predict if a person has a cancer of the class(0) with 95 percent accuracy.

With the id3 algorithm implemented in this paper and the experimentation of the IRIS data and the Cancer data, we conclude that a 100 number of the IRIS dataset is enough to predict the class of an IRIS plant with over 95percent accuracy. However, the cancer dataset may require more data in order to predict the cancer status of a person. The decision tree may predict the class (0) correctly 95% of the time but may fall short when classifying the other classes 1 – 5.

References

Dua, D., & Graff, C. (2017). *{UCI} Machine Learning Repository*. Retrieved from University of California, Irvine, School of Information and Computer Sciences: http://archive.ics.uci.edu/ml

Ming, H., Wenying, N., & Xu, L. (n.d.). An improved Decision Tree classification algorithm based on ID3 and the application in score analysis. *2009 Chinese Control and Decision Conference, Guilin, 2009*, (pp. 1876-1879).

Rajeshkanna, A., & Arunesh, K. (2020). ID3 Decision Tree Classification: An Algorithmic Perspective based on Error rate. *2020 International Conference on Electronics and Sustainable Communication Systems (ICESC)*, (pp. 787-790). Coimbatore, India.