CS 6350: Mid-Term Report

Savanna Wolvin

October 26th, 2021

1 Summary

With US Census data, we plan to predict if a citizen's annual income is above or below $50,000. The predictors used are age, their work class, fnlwgt, education, education number, martial status, occupation, relationship, race, sex, capital gains and losses, hours per week worked, and native country. We have a data set used to train our models and a data set for us to predict.

In this report, we sought out to answer three questions:

1. How does the treatment of unknown values effect the average prediction error?
2. How well does a Random Forest Algorithm predict the annual income?
3. How does the Decision Tree Prediction Errors Compare to that of Linear Regression?

2 Pre-Processing the Data for Decision Trees

The data sets given include both numeric and categorical data, therefore, in decision making, we left the categorical data alone and grouped the numerical data into different categories. For the age, fnlwgt, education number, and hours worked per week, these attributes were placed into four categories based on quartiles. Meaning, each attribute label was placed into either the first, second, third or fourth quartile. Specifically for the capitol gains and loss categories, there was an over-welling number of examples with a value of zero. Therefore, all zero values were categorized as 'none'. While the quantile values were calculated from the examples with non-zero capitol gains and losses, and the examples with capitol gains or losses greater than zero were categorized by their quantile.

3 Assessing Standard Decision Tree Algorithm

First, I decided to process the census data through a decision tree algorithm to evaluate the training prediction error. Figure 1 shows how the prediction error of the training dataset changes at increasing levels of branches between treating unknown values as its own attribute or by replacing it with the most common label.

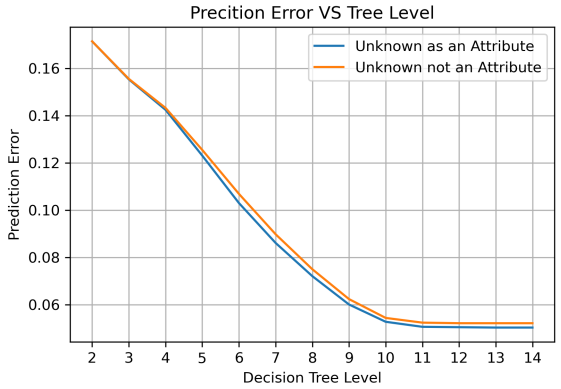

Figure 1: Prediction Error at Each Level Depending on How Unknown Values are Treated

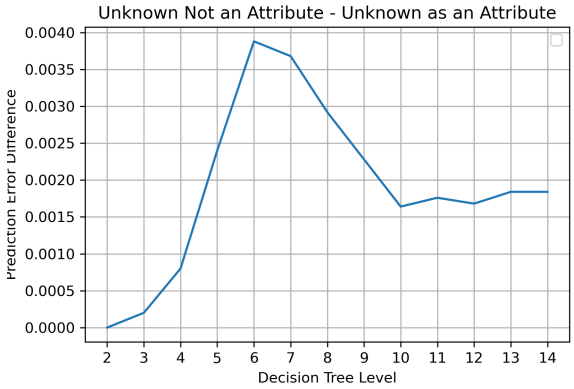

Figure 2: Difference Between the Prediction Errors on How Unknown Attributes are Treated

Initially, the prediction error between unknown as an attribute and unknown not as an attribute is the same, with a value of 0.1714. As the number of levels increase, treating unknown values as an attribute decreases the decision tree prediction error at a faster rate. Overall, treating unknown values as its own attribute creates lower prediction errors, ranging up to 0.0039 lower than treating unknown not as its own attribute, shown in Figure 2.

At the final level of the decision tree, the prediction errors of the decision tree by treating unknowns as an attribute and not are 0.0503 and 0.0522, respectively. This is a 3.6398% increase in prediction errors, 46 new mistakes, due to how unknowns are treated.

4 Assessing Random Forest Algorithm

Second, I processed the census data using a random forest algorithm to predict for each citizen if they made more or less than $50,000 annually. I currently have 28 iterations of the random forest algorithm. The following figure illustrates how the prediction error decreases with the increasing iterations:

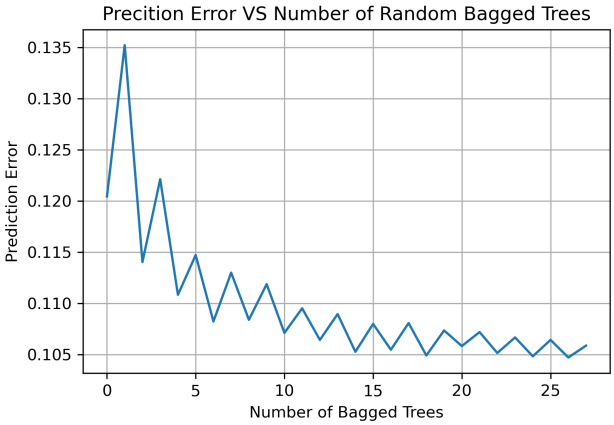

Figure 3: Prediction Error With Each Increasing Iteration

The prediction errors at my current run of my random forest model seems to be converging between 0.1047 and 0.1059. I plan to continue to run this program to the 500 iterations for a complete analysis.

5 Pre-Processing the Data for Linear Regression

The data sets given include both numeric and categorical data, but for linear regression, we used only the numerical values from the dataset. The numeric attributes are age, fnlwgt, education number, hours worked per week, and capitol gains and loss. Since these categories have very large ranges in values, we normalized each data array using the following equation:

$$X_{norm} = \frac{X - X_{min}}{X_{max} - X_{min}}$$

6 Linear Regression

For the numeric census data attributes, we used a batch gradient descent to predict if each citizen made more or less than $50,000 annually. With the batch gradient descent, we converged on a learning rate of 0.00011 using 500 iterations.

The weighted vector, the norm of the weight vector difference, and the cost function values is:

$$\mathbf{w} = \begin{bmatrix} 0.2641 & -0.4260 & 0.2898 & 1.1241 & 0.6041 & 0.0852 \end{bmatrix}$$

$$norm = 7.2529 \cdot 10^{-6}$$

$$Cost\ \ Function\ \ Value = 1909.5$$

From this matrix, we calculated the predicted value by taking the dot product of $\mathbf{w}^T$ and each example, then determining if the value is closer to 1 or 0. This is shown below:

$$PredictedValue = \begin{cases} 0 & \text{if } \mathbf{w}^T\mathbf{x}_j < 0.5 \\ 1 & \text{if } \mathbf{w}^T\mathbf{x}_j > 0.5 \end{cases}$$

The prediction error found using batch gradient descent with the training data was 0.2162.

7 Discussion

In the 3rd section, we compared how the prediction error would change depending on how to unknown value is treated. We found that the prediction error decreases by 0.0019 if unknown values are treated like their own attribute. This decrease in error is not significant. But while this decrease is not significant, only 46 more mistakes, I plan to replace the unknown values with the most likely attribute.

In the 4th section, we used a random forest algorithm to predict for each citizen example if they made more or less than $50,000 annually. The prediction errors seem to be converging around 0.1047 and 0.1059, which is 0.0544 to 0.0556 higher than that from the decision tree. Though I do plan to continue to run the random forest decision tree for a few more hundred runs to confirm this conclusion.

In the 6th section, we used linear regression of the numerical values to predict for each citizen example if they made more or less than $50,000 annually. This route was the least successful with a prediction error of 0.2162, illustrating a 429.8% increase in prediction errors in comparision to the standard decision tree. I do not plan to use linear regression in this project in the future.

For the rest of the semester, I plan to implement more of the algorithms taught in class. Once implemented, I want to compare these results to the new results. The most successful results in relation to the training results will be chosen as my final results of the project.