Salary Projection

Jacob Hurley, Sofia Gray, and Trevor Hoshiwara

Data 498 Capstone Final Project

Professor Zhang

May 10, 2022

The national median salary in the United States is about $50,000. In lower cost areas, this is a great salary, but in higher cost areas this might not be adequate to live comfortably. With this project, we aim to understand what ensures that an individual secures a job with a salary of 50k or more through the power of machine learning.

To conduct this analysis, we will be working with a CSV dataset containing several fields pertaining to living conditions. The "Adult Income" dataset comes from Kaggle, and contains over 33,000 records pulled from the United States Census Bureau in 1994. In total there are fifteen attributes, including the salary target variable. The nine categorical variables are as follows: *workclass*, *highest education*, *marital status*, *occupation*, *relationship*, *race*, *sex*, *native country*, and *salary*. The remaining six variables are quantitative: *age*, *final weight*, *education years*, *capital gain*, *capital loss*, and *work hours per week*.

This indicates some immediate concerns for collinearity. For example, it is evident that *highest education* and *education years* will be directly related to one another. Therefore the *highest education* categorical attribute has been excluded from the pool of predictors under consideration. Additionally, the *native country* variable was excluded for some models due to the fact that it generates over 40 dummy variables, causing limitations. One such limitation exists with decision trees in R, which only support up to 32 factors for a single categorical variable.

Our target variable, *salary*, is indicated by "Yes" or "No" if the person's salary is above $50,000 or below $50,000 respectively. Based upon the format of the data in combination with the primary objective, we will perform classification under supervised learning with several different approaches.

Because our response variable is binary, taking values of 0 for salary under 50k and 1 for salary over 50k, we thought the most obvious approach would be to build a logistic regression model. First, we plotted a correlation matrix as seen below.

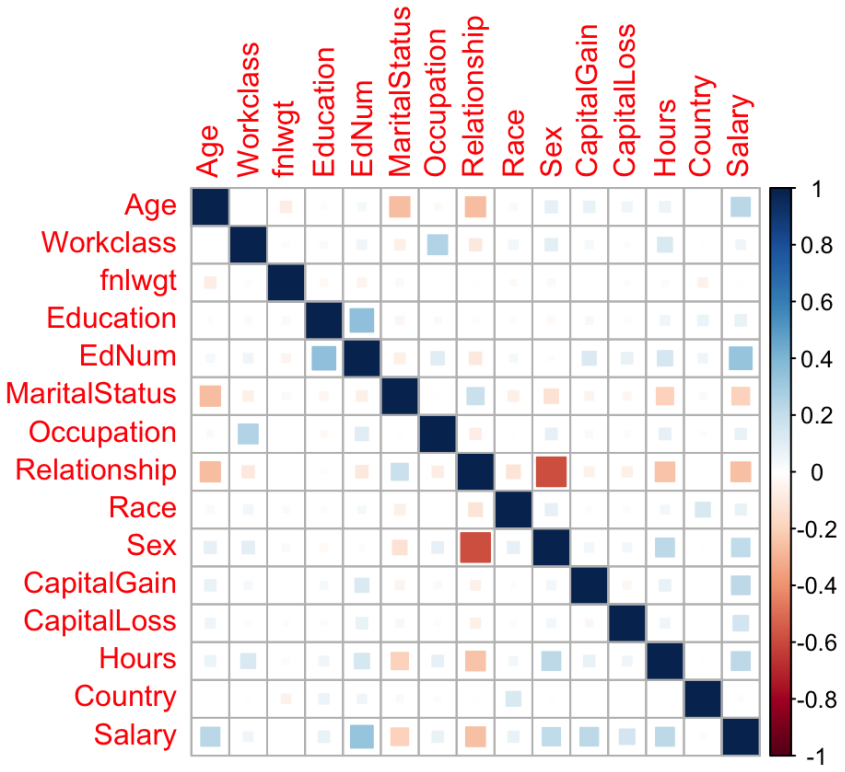

Figure 1.1

This matrix allows us to see at first glance if there are any positive or negative linear correlations between our explanatory variables and our response variable. From the matrix, we can see that quite a few variables have zero correlation with each other, which could potentially affect our model.

To start building our model, we turned all the categorical variables into numeric values so that the glm function would take these as parameters. Thus, our first logistic regression model predicts whether salary is over or under 50k using all the variables initially provided by the data set. This model yielded a 82.42% accuracy rate. By removing the insignificant variables, which

our first model told us were country and workclass, we ran the model again and this time it yielded 82.5%, only less than a tenth of a percent difference. These accuracies were calculated by making a prediction accuracy table. Additionally, we performed a 10 fold cross validation to further analyze the accuracy of our logistic regression model, and this yielded an R-squared value of 1 with a mean squared error of essentially 0. So overall, our logistic regression model was good at predicting the target salary, but it failed to single out the most important and significant predictors since our model retained all the variables save for two.

Next, we performed LASSO variable selection to single out the more significant predictors and adjust for multicollinearity, therefore improving the accuracy rate of our logistic regression. However, our LASSO selection only predicted the target salary 62.5% percent of the time, a significant decrease from our original logistic 82.5% accuracy rate. The plot below shows the mean squared error of our LASSO selection.

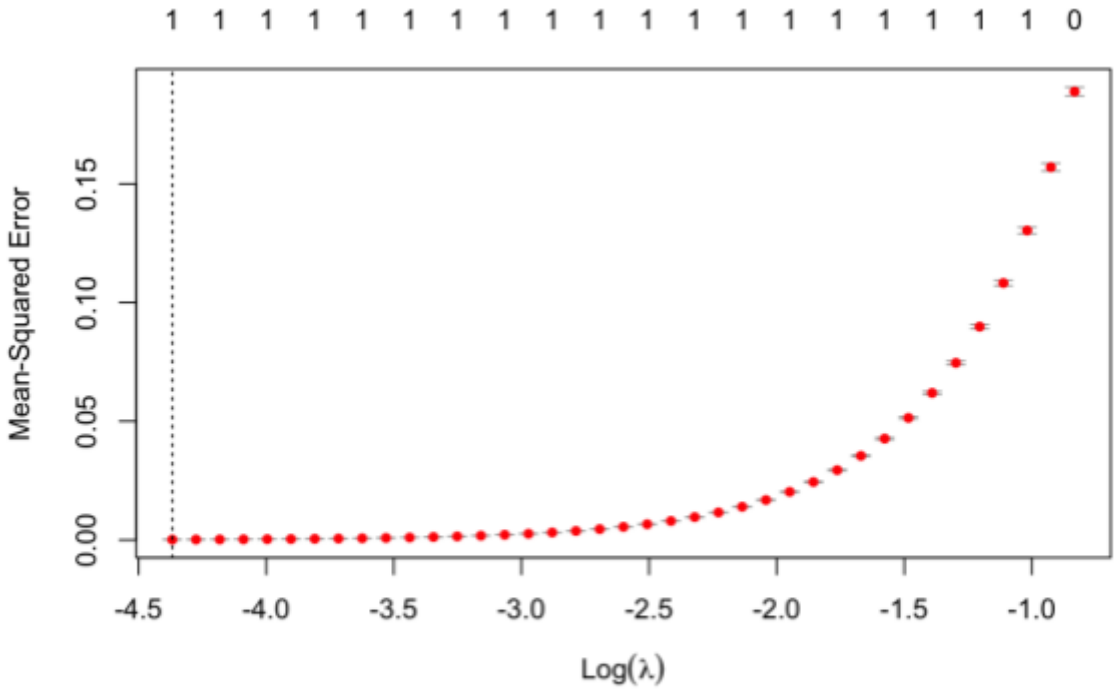

Figure 1.2

The second approach that we decided to attempt was a Naive Bayes with a Gaussian prior. First, for this algorithm to work properly, we are under the assumption that none of our variables are correlated and that they follow a normal distribution. Figure 1.1 shows a correlation matrix using a Pearson correlation test to determine relationship between variables. The darker the box was, the stronger the negative correlation, and the brighter the box the stronger the positive correlation. The most concerning correlations would be between education and education-Num, and relationship and sex. However, these variables were not used in the final working set. Figure 1.2 shows the distribution of variables in the data set.

As we can see, they have a similar curve to a normal distribution, however they are not properly following it. To fix this issue, we standardized the continuous variables by subtracting by the mean and dividing by the standard deviation. Now that our two assumptions are met, we

had to choose the best subset. The method we chose for this was *Exhaustive Feature Selection*. This involved creating a list of every combination of features and then training a model to each subset. When testing the model accuracy, we used the mean score from a 10-fold cross-validation and chose the subset that provided the highest score. The subset {'age', 'Education-num', 'Marital-status', 'Hours-per-week', 'Native_country'} provided the highest score with 82.47%. With the best subset known, we tested the accuracy of the model on a 25% test set and received a score of 83.04%.

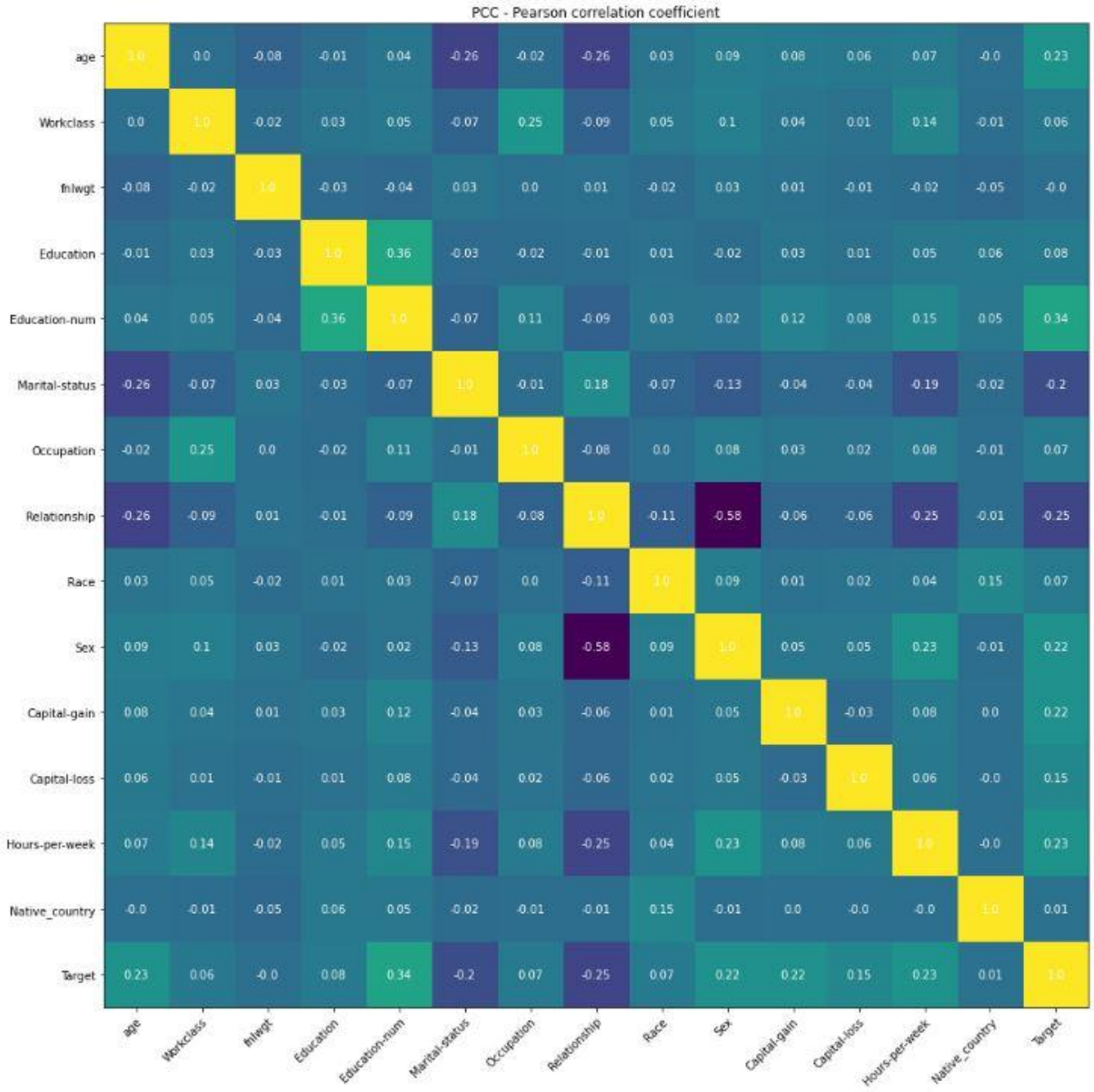
PCC - Pearson correlation coefficient
age 1.0 0.0 -0.08 -0.01 0.04 -0.26 -0.02 -0.26 0.03 0.09 0.08 0.06 0.07 -0.0 0.23
Workclass 0.0 1.0 -0.02 0.03 0.05 -0.07 0.25 -0.09 0.05 0.1 0.04 0.01 0.14 -0.01 0.06
fnlwgt -0.08 -0.02 1.0 -0.03 -0.04 0.03 0.0 0.01 -0.02 0.03 0.01 -0.01 -0.02 -0.05 -0.0
Education -0.01 0.03 -0.03 1.0 0.36 -0.03 -0.02 -0.01 0.01 -0.02 0.03 0.01 0.05 0.06 0.08
Education-num 0.04 0.05 -0.04 0.36 1.0 -0.07 0.11 -0.09 0.03 0.02 0.12 0.08 0.15 0.05 0.34
Marital-status -0.26 -0.07 0.03 -0.03 -0.07 1.0 -0.01 0.18 -0.07 -0.13 -0.04 -0.04 -0.19 -0.02 -0.2
Occupation -0.02 0.25 0.0 -0.02 0.11 -0.01 1.0 -0.08 0.0 0.08 0.03 0.02 0.08 -0.01 0.07
Relationship -0.26 -0.09 0.01 -0.01 -0.09 0.18 -0.08 1.0 -0.11 -0.58 -0.06 -0.06 -0.25 -0.01 -0.25
Race 0.03 0.05 -0.02 0.01 0.03 -0.07 0.0 -0.11 1.0 0.09 0.01 0.02 0.04 0.15 0.07
Sex 0.09 0.1 0.03 -0.02 0.02 -0.13 0.08 -0.58 0.09 1.0 0.05 0.05 0.23 -0.01 0.22
Capital-gain 0.08 0.04 0.01 0.03 0.12 -0.04 0.03 -0.06 0.01 0.05 1.0 -0.03 0.08 0.0 0.22
Capital-loss 0.06 0.01 -0.01 0.01 0.08 -0.04 0.02 -0.06 0.02 0.05 -0.03 1.0 0.06 -0.0 0.15
Hours-per-week 0.07 0.14 -0.02 0.05 0.15 -0.19 0.08 -0.25 0.04 0.23 0.08 0.06 1.0 -0.0 0.23
Native_country -0.0 -0.01 -0.05 0.06 0.05 -0.02 -0.01 -0.01 0.15 -0.01 0.0 -0.0 -0.0 1.0 0.01
Target 0.23 0.06 -0.0 0.08 0.34 -0.2 0.07 -0.25 0.07 0.22 0.22 0.15 0.23 0.01 1.0
age
Workclass
fnlwgt
Education
Education-num
Marital-status
Occupation
Relationship
Race
Sex
Capital-gain
Capital-loss
Hours-per-week
Native_country
Target

Figure 2.1

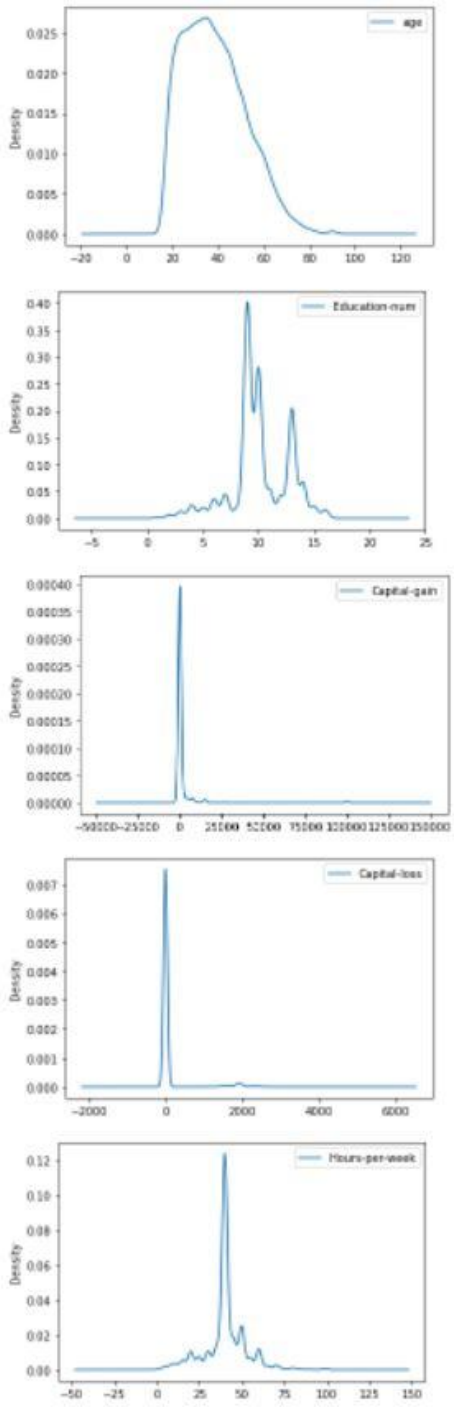
age
Density
Education-num
Density
Capital-gain
Density
Capital-loss
Density
Hours-per-week
Density

Figure 2.2

The algorithm that we decided to attempt after naive bayes was a K nearest neighbor approach. This algorithm takes in a parameter K which is the number of neighbors that a parameter has. It will check the K nearest points and decide which category the point belongs to based on the percentage of category closest to it. This algorithm can take a long time to run due to our large amount of data. Because of this, we used the same “Best subset” found from the naive bayes algorithm ({'age', 'Education-num', 'Marital-status', 'Hours-per-week', 'Native_country'}). We then had to determine what the best K value would be to provide the highest accuracy score. To do this, we looped through every K value from 1 to 24 (inclusive). Figure 3.1 shows the accuracy of the 10 fold cross-validation for each K value. The K value chosen was 23 which provided a 10 fold cross-validation score of 82.22% and a testing score of 83.51%.

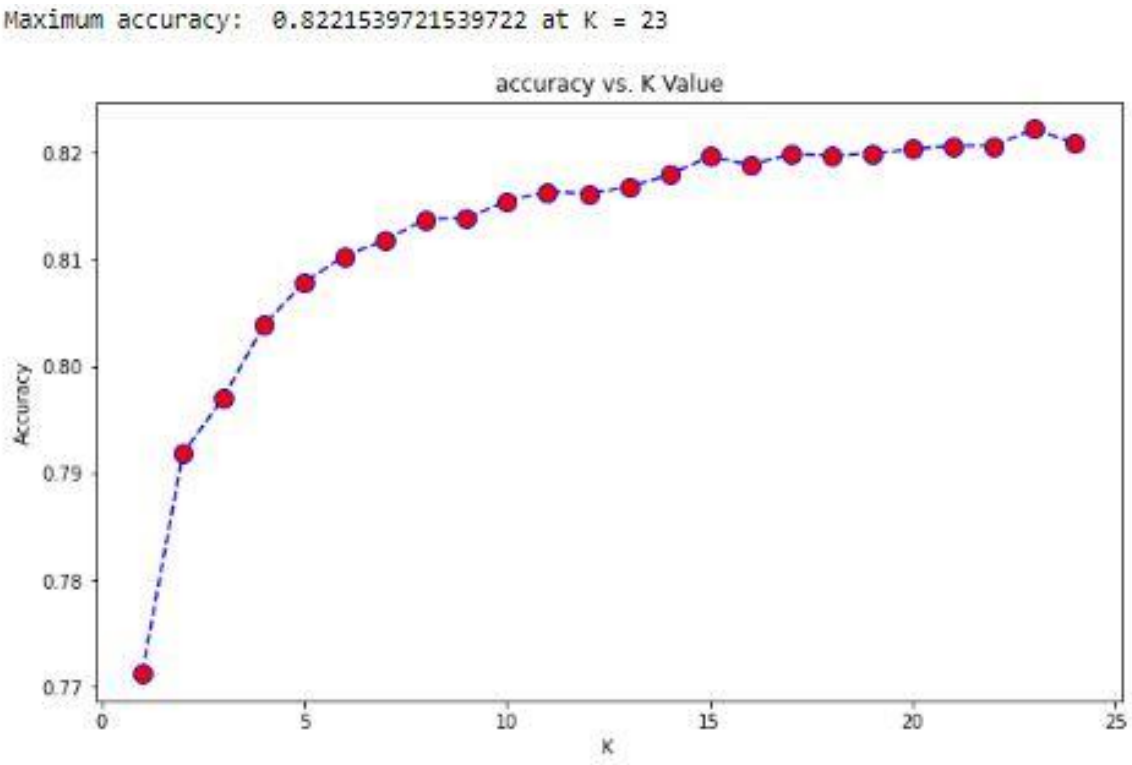

Figure 3.1

Lastly, knowing the success of many neural networks, we attempted to replicate that success. We again used the same "Best Subset" found from the naive bayes algorithm. The variable that we attempted to optimize was the number of hidden layers in the network. Figure 4.1, similar to figure 3.1, shows the accuracy of the model using a 5 fold cross-validation. We observed that the highest accuracy is observed at 25 hidden layers, however there is a lot of variability in this model as the hidden layer numbers increase. This leaves us to believe that there is not a "true" correct value for the number of hidden layers in the model and that a better variable to manipulate would be the subset used.

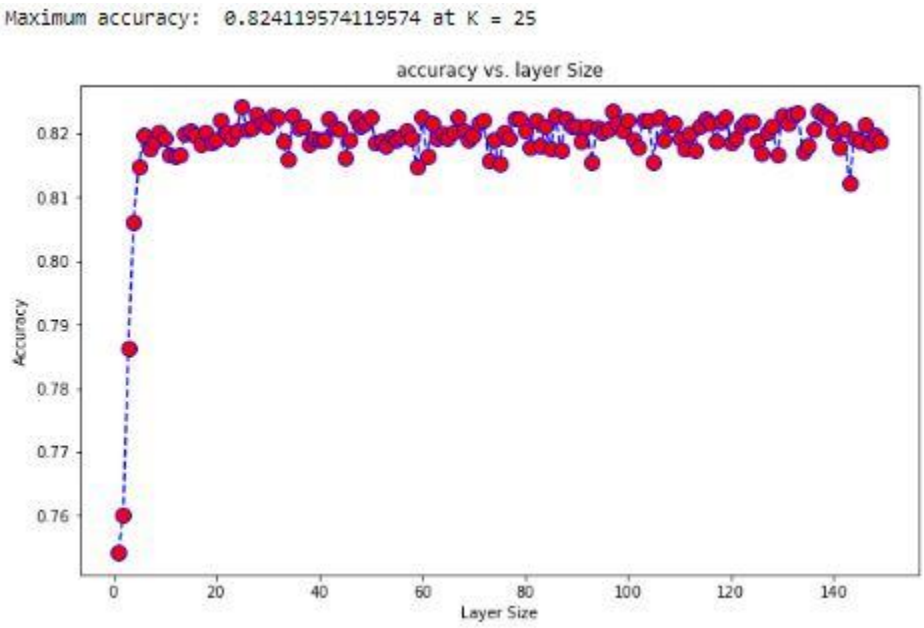

Figure 4.1

Another worthwhile non-parametric approach is a decision tree. Given that our target is either "Yes" or "No", we are dealing with the binary classification tree method. After strategically binning observations based on certain criteria, this tree's prediction will result in the majority vote for whatever region the new observation falls into. Using the *caret* package in R to create a 5-fold split, the cross validation yielded an accuracy of 84.08%. A visual of the splitting process is illustrated in *Figure 5.1*.

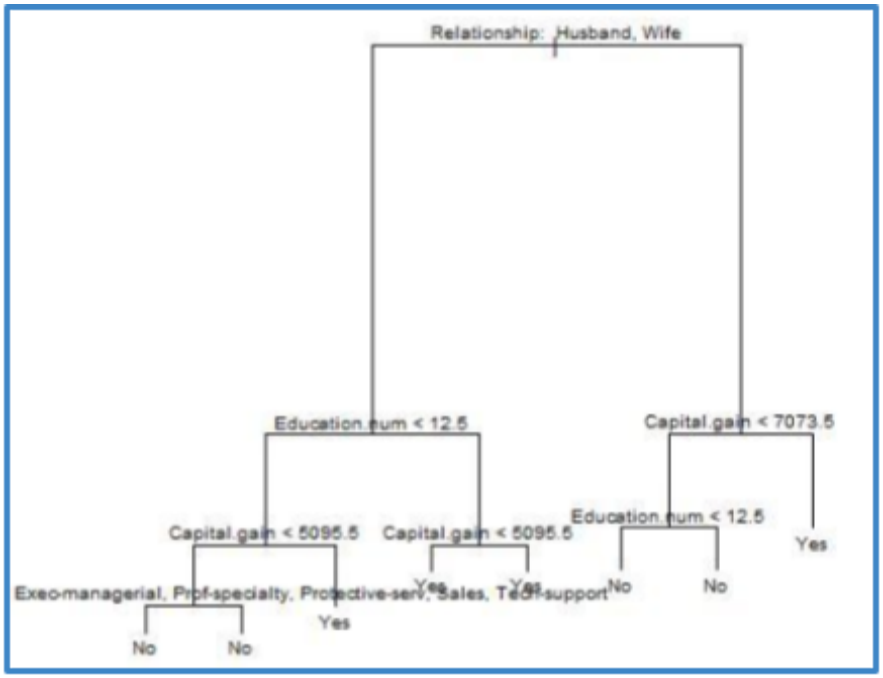

Figure 5.1:
Binary classification decision tree with 8 leaves and 84.08% accuracy.

Evidently there is some redundancy present in this tree. Redundancy occurs when a tree splits, but yields the same prediction for both its left and right children. While it may not have a substantial impact on the model performance, it is advantageous to investigate pruning for simplicity and to prevent overfitting.

This can be accomplished by restricting the number of leaves a tree may contain, and evaluating its corresponding error. *Figure 5.2* plots the cross validation error as a function of the number of leaves permitted.

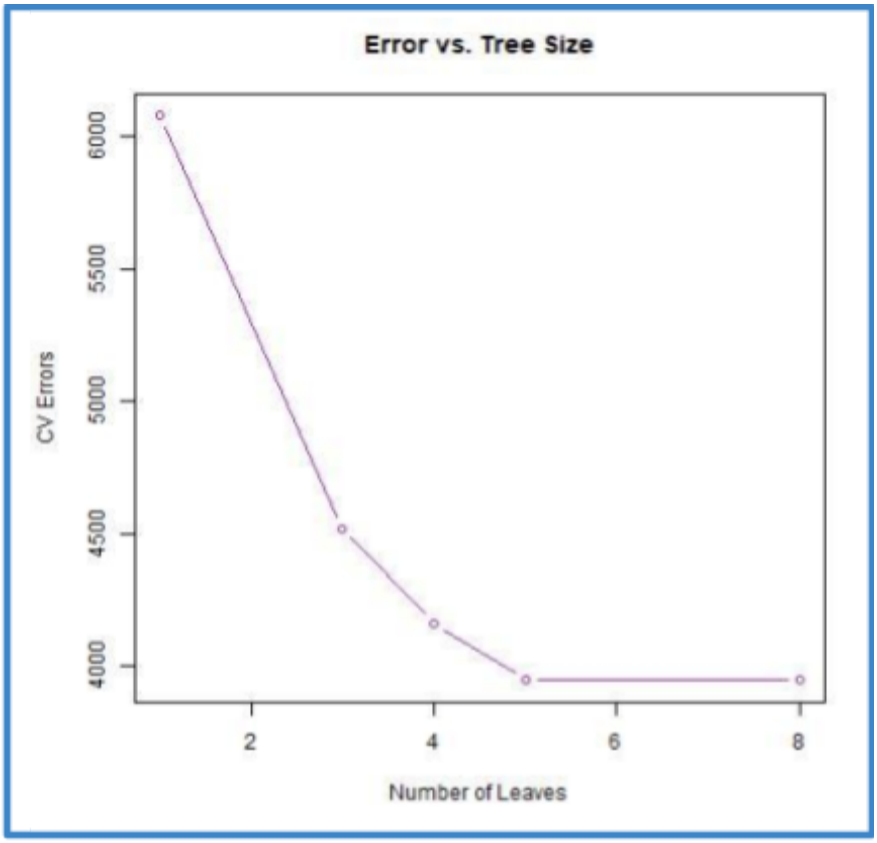

Figure 5.2:
Plotting CV Error against the number of leaves in the decision tree to determine an optimal stopping point.

This plot makes it clear that the decision tree with five leaves produces the same error as the tree with eight leaves. The pruned tree can then be created as seen in *Figure 5.3*.

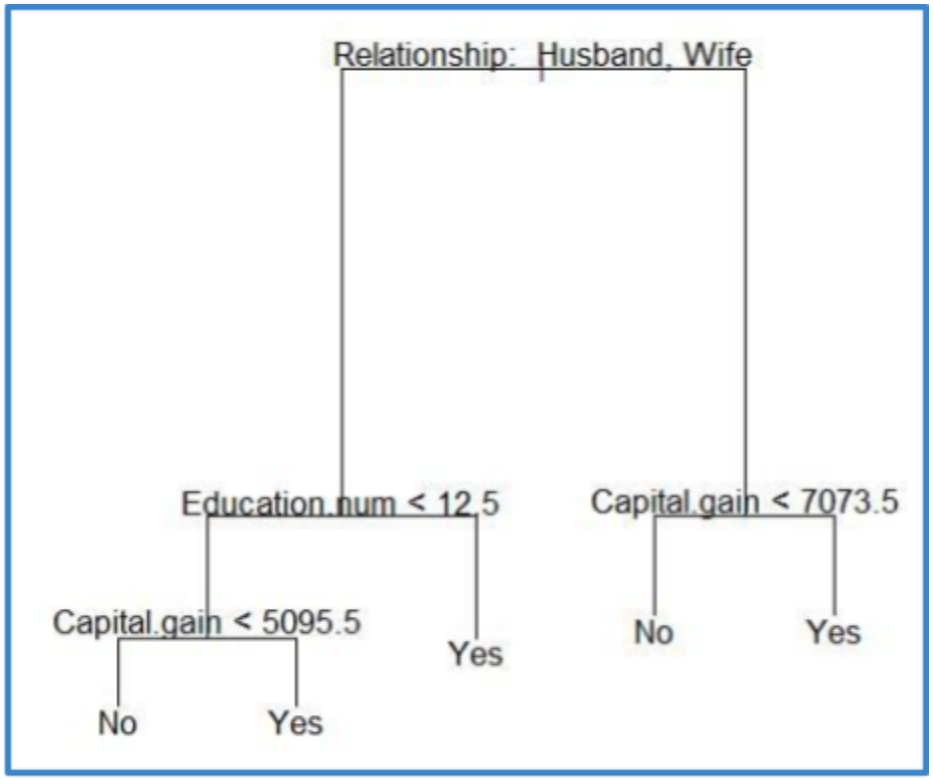

Figure 5.3:
Binary classification decision tree with 5 leaves and 84.08% accuracy.

Although the performance of the pruned tree is promising, there may be room for improvement. The current splitting criteria will select the optimal predictor to split on, which means that cross validation via Bootstrapping will result in correlated predictions. We can decorrelate the predictions with a method known as Random Forests. This is accomplished by selecting a subsample, m, of predictors rather than examining all of them for each split. A small m can be extremely useful in the case of correlated predictors. A random forest with $m = 2$ yielded an 86.37% accuracy from 5-fold cross validation.

Out of all the models tested and statistical approaches explored, we found that the random forest model with $m = 2$ proved to be the most accurate at predicting the target salary, with a 86.37% accuracy rate. The model with the lowest accuracy was the LASSO selection with a 62.5% accuracy rate. This was surprising considering it usually tends to perform better.

One concern in our experimental design is that the subset used for our KNN analysis is not the optimal subset for that algorithm since running the exhaustive search took a lot longer than expected. Additionally, as mentioned before, the logistic regression did not prove to be useful at identifying the most significant predictors while the classification trees were. We think this is because of correlation between certain variables, such as marital status and relationship, that causes some collinearity that could potentially mess with the model. Overall, we conclude the random forest model with $m = 2$ is the best model to predict target salary.

References

Shochat, Gal. “Classification Problem / Yes or No 50k Salary.” Kaggle, United States Census Bureau, 31 Mar. 2022, https://www.kaggle.com/datasets/galshochat/classification-problem-yes-or-no-50k-salary.