Income Earning Potential

Elinnoel Nuñez, Nadeem Karachi, Sara Shiha, Hung Nguyen, Nimet Ozkan

2023-04-25

Introduction

The task at hand is to predict whether an individual's annual income is over $50,000 USD (income) based solely on data obtained from the U.S. Census in 1994. The census data contains various attributes such as occupation type (occupation), education level (education), race, sex, marital status (marital_status), and country of origin (country), which are either categorical or continuous values. Our objective is to use this information to determine if an individual falls into the high-income category.

Categorical Variables: workclass, education, marital_status, occupation, relationship, race, sex, country (can also be treated as continuous)

Continuous Variables: age, education_num, capital_gain, capital_loss, hours_per_week, fnlwgt (can also be treated as categorical)

Response Variable: income: >50k, <=50k

Question

Our task is to predict if an individual's income will be over $50K/yr (qualitative) using the given attributes.

Clean Data

Before we begin using our models, we need to clean the data by removing missing values and setting categorical variables.

Reading Data Read the data first.

```
set.seed(1)
base_adult <- read.csv("adult.csv", stringsAsFactors = TRUE)
adult <- read.csv("adult.csv")
summary(base_adult)
```

Remove rows with missing values In the given dataset, missing values are denoted by the "?" character. As the missing data only appears in our categorical variables, it is not appropriate to substitute them with a median or mean. Therefore, we have decided to eliminate those observations from our analysis. This has resulted in a reduced dataset with 45,222 observations. There are also several categorical variables that require conversion to numeric format for use in our analysis. Additionally, we will need to create a binary variables for the response variable to facilitate the modeling process.

```
unknown_val_cols <- c("occupation", "workclass", "country")
base_adult <- base_adult[rowSums(base_adult[, unknown_val_cols] == " ?") == 0, ]
base_adult$income <- as.integer(base_adult$income != "<=50K") # 0 for <50k, 1 for >= 50k
base_adult$income <- as.factor(base_adult$income)
base_adult$country <- as.integer(base_adult$country != " United-States") # 0 for US, 1 everything else
base_adult$country <- as.factor(base_adult$country)
base_adult <- na.omit(base_adult)
summary(base_adult)
```

```
##       age                     workclass         fnlwgt
##  Min.   :17.00    Private         :33307   Min.   :  13492
##  1st Qu.:28.00    Self-emp-not-inc: 3796   1st Qu.: 117388
##  Median :37.00    Local-gov       : 3100   Median : 178316
##  Mean   :38.55    State-gov       : 1946   Mean   : 189735
##  3rd Qu.:47.00    Self-emp-inc    : 1646   3rd Qu.: 237926
##  Max.   :90.00    Federal-gov     : 1406   Max.   :1490400
##                  (Other)          :   21
##          education     education_num                  marital_status
##   HS-grad     :14783   Min.   : 1.00    Divorced             : 6297
##   Some-college: 9899   1st Qu.: 9.00    Married-AF-spouse    :   32
##   Bachelors   : 7570   Median :10.00    Married-civ-spouse   :21055
##   Masters     : 2514   Mean   :10.12    Married-spouse-absent:  552
##   Assoc-voc   : 1959   3rd Qu.:13.00    Never-married        :14598
##  11th         : 1619   Max.   :16.00    Separated            : 1411
##  (Other)      : 6878                    Widowed              : 1277
##             occupation            relationship                    race
##   Craft-repair   : 6020    Husband       :18666    Amer-Indian-Eskimo:  435
##   Prof-specialty : 6008    Not-in-family :11702    Asian-Pac-Islander: 1303
##   Exec-managerial: 5984    Other-relative: 1349    Black              : 4228
##   Adm-clerical   : 5540    Own-child     : 6626    Other              :  353
##   Sales          : 5408    Unmarried     : 4788    White              :38903
##   Other-service  : 4808    Wife          : 2091
##  (Other)         :11454
##       sex          capital_gain    capital_loss    hours_per_week  country
##   Female:14695   Min.   :    0   Min.   :   0.0   Min.   : 1.00   0:41292
##   Male  :30527   1st Qu.:    0   1st Qu.:   0.0   1st Qu.:40.00   1: 3930
##                  Median :    0   Median :   0.0   Median :40.00
##                  Mean   : 1101   Mean   :  88.6   Mean   :40.94
##                  3rd Qu.:    0   3rd Qu.:   0.0   3rd Qu.:45.00
##                  Max.   :99999   Max.   :4356.0   Max.   :99.00
##
##  income
##  0:22654
##  1:22568
##
##
##
```

```
##
##
```

Split Data Set

To evaluate the accuracy of our prediction model, we will divide the dataset into two sets: a training set and a testing set. The dataset will be split into an 80% training set and a 20% testing set.

```
# generates a vector of random values between 0 and 1 of length equal to the number of rows in data.
train_rows <- runif(nrow(base_adult)) < 0.8
train <- base_adult[train_rows, ]
test <- base_adult[!train_rows, ]
print(nrow(train))
```

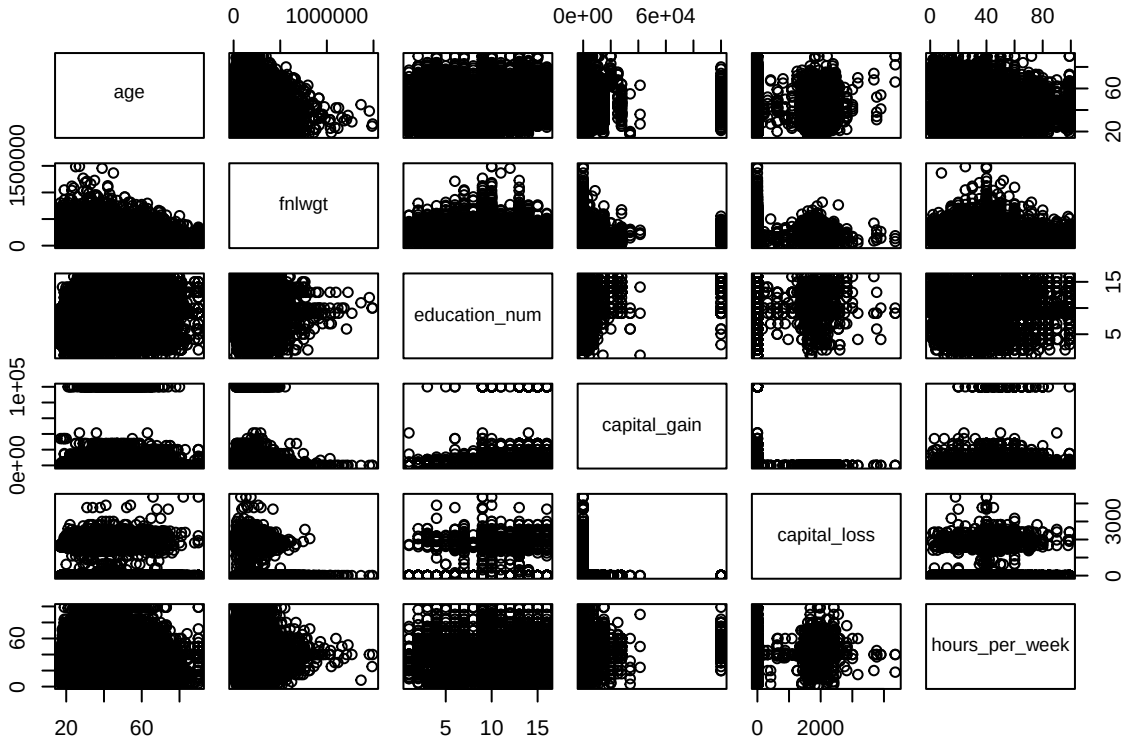
```
## [1] 36212
```

```
print(nrow(test))
```

```
## [1] 9010
```

Correlations

```
#check correlations between the numeric values
is_numeric <- sapply(adult, is.numeric)
pairs(adult[, is_numeric])
```




```
cor(adult[, is_numeric])
```

```
##                      age       fnlwgt education_num capital_gain capital_loss
## age           1.00000000 -0.076628079    0.03094038  0.077229022   0.05694383
## fnlwgt       -0.07662808  1.000000000   -0.03876068 -0.003706389  -0.00436615
## education_num 0.03094038 -0.038760684    1.00000000  0.125146459   0.08097194
## capital_gain  0.07722902 -0.003706389    0.12514646  1.000000000  -0.03144077
## capital_loss  0.05694383 -0.004366150    0.08097194 -0.031440771   1.00000000
## hours_per_week 0.07155834 -0.013518715    0.14368891  0.082157278   0.05446722
##                hours_per_week
## age                 0.07155834
## fnlwgt             -0.01351871
## education_num       0.14368891
## capital_gain        0.08215728
## capital_loss        0.05446722
## hours_per_week      1.00000000
```

Upon examining the scatterplots generated from the dataset, there appears to be little evidence of strong correlation between the variables. There is no clear pattern or trend that can be observed in the scatterplots that would suggest a significant relationship between any of the variables.

This observation suggests that the variables in the dataset are largely independent of one another, and that no single variable strongly influences the values of any other variable. However, it is still necessary to perform a thorough analysis to ensure that any potential correlation between variables is addressed.

It is important to check for correlations between variables in a dataset because highly correlated variables can cause issues in the modeling process. For example, high correlation between predictor variables can result

in instability of coefficients and decreased interpretability of the model. Therefore, it may be necessary to explore and address any potential correlation issues between variables, especially after the creation of new variables.

Logistic Regression Model

We have decided to use logistic regression for our analysis because the response variable is categorical, with a value of 1 for salaries >=50K. Logistic regression offers several advantages, including ease of implementation and interpretation, no assumptions about class distribution in the predictors, identification of the most important predictors, and reduced risk of overfitting.

However, logistic regression has some limitations, such as the inability to handle non-linear relationships between variables, and the need for low or no multicollinearity between independent variables. In high-dimensional datasets, overfitting can also be a concern, leading to less accurate results.

Logistic Regression Model Formula:

$$\frac{e^{b_0+b_1x}}{1+e^{b_0+b_1x}}$$

```
# Fit a logistic regression model
lr.glm <- glm(income ~ ., data = train, family = "binomial")
# Make predictions on the test set
lr.pred <- predict.glm(lr.glm, test, type = "response")
pred_income <- ifelse(lr.pred > 0.5, TRUE, FALSE)
# Create a confusion matrix to evaluate the performance of the model and compute accuracy metrics
confusion_matrix <- table(test$income, pred_income)
confusion_matrix
```

```
##    pred_income
##     FALSE TRUE
##   0  3086 1477
##   1  1664 2783
```

```
# Calculate misclassification rate
misclassification_rate <- (confusion_matrix[2,1] + confusion_matrix[1,2]) / sum(confusion_matrix)
misclassification_rate
```

```
## [1] 0.3486127
```

```
summary(lr.glm)
```

The misclassification rate is a measure of the model's overall accuracy and a lower rate is generally indicative of better performance. The misclassification rate of our model is 35%, which is higher than our desired level of accuracy. This indicates that our model correctly predicted only about 65% of the data in the test set, and therefore, may not be performing optimally.

Extract Significant Variables

```
# Extract p-values of coefficients
p_values <- summary(lr.glm)$coef[, 4]
# Identify significant variables based on p-values
sig_vars <- names(p_values[p_values < 0.05]) # pick based off this
sig_vars
```

```
##  [1] "(Intercept)"                         "age"
##  [3] "workclass Local-gov"                 "workclass Private"
##  [5] "workclass Self-emp-not-inc"          "workclass State-gov"
##  [7] "fnlwgt"                              "education Bachelors"
##  [9] "education Doctorate"                 "education HS-grad"
## [11] "education Masters"                   "education Prof-school"
## [13] "education10th"                       "education1st-4th"
## [15] "education7th-8th"                    "education9th"
## [17] "marital_status Married-civ-spouse"   "occupation Exec-managerial"
## [19] "occupation Farming-fishing"          "occupation Priv-house-serv"
## [21] "occupation Prof-specialty"           "occupation Protective-serv"
## [23] "occupation Sales"                    "occupation Tech-support"
## [25] "occupation Transport-moving"         "relationship Not-in-family"
## [27] "relationship Other-relative"         "relationship Own-child"
## [29] "relationship Unmarried"              "relationship Wife"
## [31] "sex Male"                            "capital_gain"
## [33] "capital_loss"                        "hours_per_week"
## [35] "country1"
```

To improve the performance of our model and decrease the misclassification rate, by running summary(lr.glm) we should identify variables that have little impact on predicting the response variable and consider removing them from the model. This process of feature selection can help to simplify the model and improve its overall performance.

```
# Subset train and test data using significant variables
sig <- c("age","workclass", "education","occupation","relationship","sex","capital_gain","capital_loss"
train_sig <- train[, sig]
test_sig <- test[, sig]
# Refit logistic regression model using significant variables
lr_sig <- glm(income ~ ., train_sig, family = "binomial")
# Make predictions on test data
glm.pred <- predict(lr_sig, test_sig, type = "response")
y_hat <- ifelse(glm.pred > 0.5, 1, 0)
# Create a confusion matrix to evaluate the performance of the model and compute accuracy metrics
conf_matrix <- table(test_sig$income, y_hat)
conf_matrix
```

```
##    y_hat
##        0    1
##   0 3105 1458
##   1 1672 2775
```

```
# Calculate misclassification rate
misclass_rate <- mean(y_hat != test_sig$income)
misclass_rate
```

```
## [1] 0.3473918
```

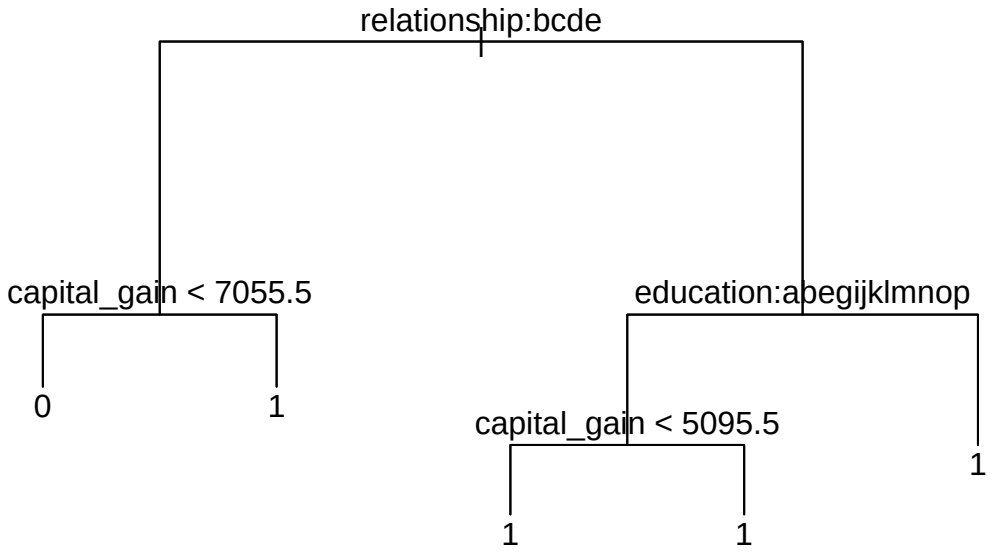
After removing non-significant variables from the model, the misclassification rate was found to be 0.3473918, which is almost identical to the rate obtained from the original model. This suggests that the removed variables had little impact on predicting the response variable. The remaining variables are likely the most important in determining income levels, and should be further analyzed to understand their individual contributions to the model.

Classification Decision Tree Model

We have chosen to utilize decision tree as our second model due to the large dimensions of our dataset, which it is capable of managing more effectively than logistic regression. Our objective is to classify individuals as 1 if their salary is greater than or equal to 50K, and 0 if it is below. Decision tree models have several advantages, such as their simplicity and ease of interpretation, minimal data preparation requirements, and ability to handle both categorical and numerical data. However, they are prone to overfitting, sensitive to changes in the dataset, and feature reduction is often necessary when dealing with large datasets. Despite these limitations, decision trees remain a valuable tool in machine learning for classification tasks and are particularly useful in identifying individuals with a salary greater than or equal to 50K in the adult.csv dataset. Their accessibility and ability to handle complex relationships between features make them a popular choice in a variety of applications.

We'll create the tree model $y = income\ x1 + x2 + x3 + ...$, where the response variable is "income", and xi to xn are the variables used in prediction, and then plot it.

```
# Load library
library(tree)
# Build the tree
btree <- tree(income ~ ., data = train)
# Plot tree
plot(btree)
text(btree)
```




```
# Get summary of base tree
base_tree <- summary(btree)
base_tree # current terminal nodes: 5, variables used: relationship, capital_gain, education
```

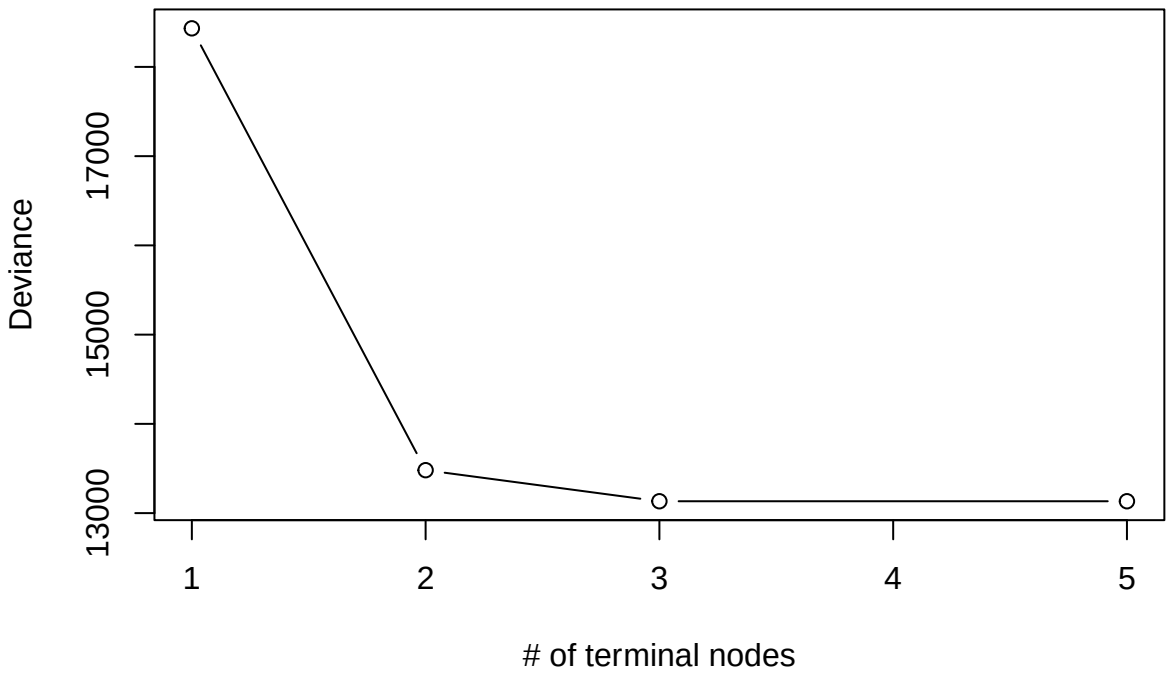
```
##
## Classification tree:
## tree(formula = income ~ ., data = train)
## Variables actually used in tree construction:
## [1] "relationship" "capital_gain" "education"
## Number of terminal nodes:  5
## Residual mean deviance:  1.254 = 45390 / 36210
## Misclassification error rate: 0.3627 = 13133 / 36212
```

Based on the current tree model, there are five terminal nodes, or final decision points, that are determined by the values of three predictor variables: relationship, capital_gain, and education. These variables have been identified as the most important in predicting income, and they are used to make the final classification decision at each terminal node. The resulting decision tree provides a clear and interpretable model for predicting income based on these key features.

Optmize by Pruning

Optimizing decision trees through pruning and cross-validation is crucial to prevent overfitting and improve the model's generalization performance. Pruning simplifies the tree by removing unnecessary branches, while cross-validation evaluates the model's performance on different subsets of data. This helps to obtain a more accurate estimate of the model's true performance and prevent overfitting, resulting in a more accurate and interpretable model for predicting income based on key predictor variables.

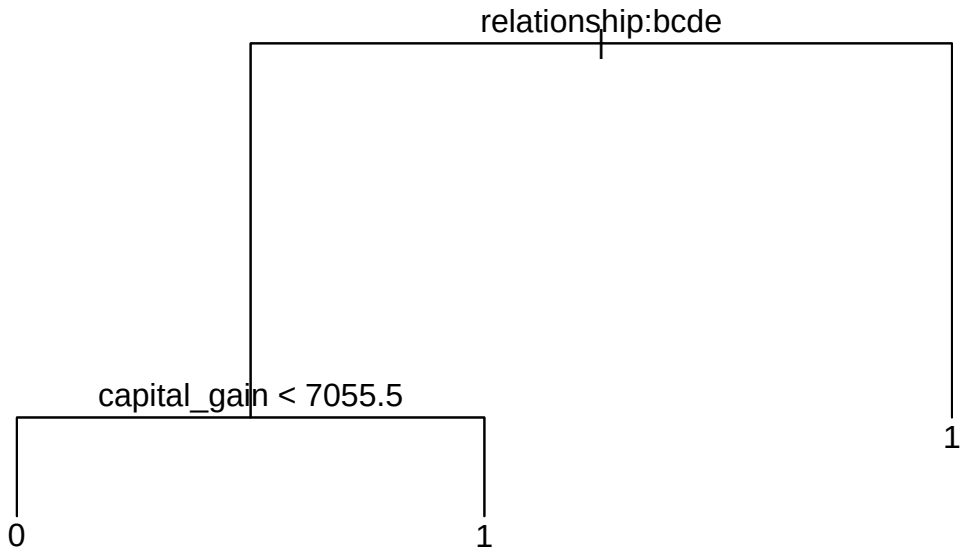
```r
# Define a function to find the pruning parameter with the lowest deviance
best_pruning <- function(cv_results) {
  min_dev <- min(cv_results$dev)
  min_dev_idx <- which(cv_results$dev == min_dev)
  return(cv_results$size[min_dev_idx])
}
# Cross-validate and find the best pruning parameter
class_cv <- cv.tree(btree, FUN = prune.misclass)
best_param <- best_pruning(class_cv)
# Plot the cross-validation results
plot(class_cv$size, class_cv$dev, type = "b", xlab = "# of terminal nodes", ylab = "Deviance")
```




```
# Prune the tree using the best parameter
prunedTree <- prune.misclass(btree, best = best_param)
# Print the summary of the pruned tree
summary(prunedTree) # variables: relationship, capital_gain
```

```
##
## Classification tree:
## snip.tree(tree = btree, nodes = 3L)
## Variables actually used in tree construction:
## [1] "relationship" "capital_gain"
## Number of terminal nodes:  3
## Residual mean deviance:  1.302 = 47160 / 36210
## Misclassification error rate: 0.3627 = 13133 / 36212
```

```
# Plot the pruned tree and label the nodes
plot(prunedTree)
text(prunedTree)
```



Although the misclassification error rate remains at 36.3% after pruning, the optimization process has still improved the decision tree model's generalization performance and interpretability. Pruning has simplified the model by removing unnecessary branches and nodes, resulting in a more interpretable model. Moreover, the model's accuracy may still be satisfactory for some applications, and further fine-tuning may lead to better results. Overall, the optimization process through pruning and cross-validation has been successful in creating a more accurate and interpretable decision tree model for predicting income based on key predictor variables.

Trials

We'll run 10 trials and determine their average for both logistic regression and decision tree.

```
evaluate_trials <- function() {
  trials <- 10
  # Evaluate misclassification rate over multiple trials
  logreg_misclass_rates <- numeric(trials)
  tree_misclass_rates <- numeric(trials)
  for (i in 1:trials) {
    set.seed(i)
    # Split data into train and test sets
    train_rows_x <- runif(nrow(base_adult)) < 0.8
    train_x <<- base_adult[train_rows_x, ]
    test_x <- base_adult[!train_rows_x, ]
    # Subset train and test data using significant variables
    train_sig <- train_x[, sig]
    test_sig <- test_x[, sig]
```

```
    # Refit logistic regression model using significant variables
    lr_sig <- glm(income ~ ., train_sig, family = "binomial")
    # Make predictions on test data
    glm.pred <- predict(lr_sig, test_sig, type = "response")
    y_hat <- ifelse(glm.pred > 0.5, 1, 0)
    # Calculate misclassification rate for logistic regression
    logreg_misclass_rate <- mean(y_hat != test_sig$income)
    logreg_misclass_rates[i] = logreg_misclass_rate
    # Build the tree
    ba_tree <- tree(income ~ ., train_x)
    # Optimize by pruning
    class_cv <- cv.tree(ba_tree, FUN = prune.misclass)
    best_param <- best_pruning(class_cv)
    # Prune the tree using the best parameter
    pruned_tree <- prune.misclass(ba_tree, best = best_param)
    # Calculate misclassification rate for classification decision tree
    tree_sum <- summary(pruned_tree)
    tree_misclass_rate <- tree_sum$misclass[1]/tree_sum$misclass[2]
    tree_misclass_rates[i] <- misclass_rate
  }
  # Return the average misclassification rates for both
  return (list(mean(logreg_misclass_rates), mean(tree_misclass_rates)))
}
misclass_rate_vals = evaluate_trials()
logreg_misclass_avg <- misclass_rate_vals[[1]]
tree_misclass_avg <- misclass_rate_vals[[2]]
logreg_misclass_avg
```

```
## [1] 0.350494
```

```
tree_misclass_avg
```

```
## [1] 0.3473918
```

The average misclassification rate for the logistic regression model, based on the average of 10 trials, is 35.05%. On the other hand, the decision tree model has an average misclassification rate of 34.74%, which is slightly better than the logistic regression model. These results suggest that the decision tree model may be a more suitable algorithm for predicting income based on the available predictor variables in the dataset.

Conclusion

In conclusion, the decision tree model has shown to perform slightly better than the logistic regression model in predicting an individual's income exceeding $50,000 based on the adult dataset from the 1994 census. The most significant variables for predicting an individual's income exceeding $50,000 are "education", "capital_gain", and "relationship". These variables were used in the tree's splitting criteria and appeared in multiple levels of the tree, indicating their importance in the model's decision-making process. Other variables, such as "age", "workclass", and "occupation", also appeared in the tree but were less significant in terms of splitting criteria and had less impact on the model's predictions. The difference in performance between the two models is relatively small, and further adjustments or evaluation methods may produce different results. Ultimately, both models provide viable options for predicting income based on the available predictor variables in the dataset, and selecting the most suitable model would depend on the specific application and desired level of accuracy.

Bibliography

- https://archive-beta.ics.uci.edu/dataset/2/adult