HW4 Presentation

Itay Akad

Values Being Predicted

Numerical Prediction Target: HoursPerYear

This is a feature I engineered that represents the total hours worked per year. It's calculated by multiplying the hours worked per week by 52 (the number of weeks in a year). Predicting this variable can provide insights into employment patterns and work commitment, which can be useful in various socio economic analyses.

Categorical Prediction Target: Marital.Status

This variable indicates an individual's marital status, categorizing them as married, divorced, never married, and other similar categories. This variable can be crucial for predicting various social and economic outcomes, as it often correlates with different life stages and financial responsibilities. Analyzing "marital.status" in conjunction with other demographic or occupational data helps in understanding broader social patterns and how marital status influences or reflects other aspects of life.

Numerical Prediction Target: HoursPerYear

The prediction model was built by segmenting the dataset based on whether or not someone was categorized as a husband. This segmentation aimed to capture distinct work patterns influenced by modern roles in the home and the need to support a family (i.e. married men are more likely to work longer hours than wives, unmarried people, etc). To enhance model accuracy, variables such as age group, education level, gender, and marital status were considered. Separate linear models were then fitted for husbands and non-husbands, and predictions from these models were combined to form a comprehensive prediction vector for all data points. The MSE for this combined approach was calculated to be 337,520.6, which showed an improvement over the benchmark models, where the lowest MSE was 386,036.4. Although this MSE is only 12.57% lower than the benchmark, far from the 50% stated in the assignment, it still indicates better accuracy, demonstrating an enhanced predictive performance. Furthermore, this small percentage change can also be attributed to the sheer size of the data, which contains tens of thousands of data points. This means that a smaller data set may have returned a better percentage.

```
> #--PREDICTING NUM VALUE--
> ##--CREATING BENCHMARK--
>  data$race <- as.factor(data$race)
>  data$marital.status <- as.factor(data$marital.status)
>  # Fit a linear model
>  lm_model <- lm(HoursPerYear ~ race + marital.status, data = data)
>  # Fit a regression tree model
>  rpart_model <- rpart(HoursPerYear ~ race + marital.status, data = data,
+                      control=rpart.control(minsplit=50, cp=0.01))
>  # Prediction using the linear model
>  pred_lm <- predict(lm_model, newdata = data)
>  # Prediction using the regression tree model
>  pred_rpart <- predict(rpart_model, newdata = data)
>  # Calculate MSE for both models
>  mse_lm <- mse(data$HoursPerYear, pred_lm)
>  mse_rpart <- mse(data$HoursPerYear, pred_rpart)
>  # Compare MSEs and determine the better benchmark
>  benchmark_mse <- min(mse_lm, mse_rpart)
>  # Print the results
>  cat("MSE from LM: ", mse_lm, "\n")
MSE from LM:  386036.4
>  cat("MSE from Rpart: ", mse_rpart, "\n")
MSE from Rpart:  389555.4
>  cat("Benchmark MSE is: ", benchmark_mse, "\n")
Benchmark MSE is:  386036.4
##--PREDICTION MODEL--
>  # Segment data based on relationship status
>  data$RelationshipSegment <- ifelse(data$relationship == "Husband", "Husband", "Other")
>  # Update models
>  model_husband <- lm(HoursPerYear ~ AgeGroup + education + sex + marital.status,
+                     data = data[data$RelationshipSegment == 'Husband',])
>  model_other <- lm(HoursPerYear ~ AgeGroup + education + sex + marital.status,
+                    data = data[data$RelationshipSegment != 'Husband',])
>  # Making predictions for each segment
>  pred_husband <- predict(model_husband, newdata = data[data$RelationshipSegment ==
'Husband',])
>  pred_other <- predict(model_other, newdata = data[data$RelationshipSegment != 'Husband',])
>  # Combine predictions back into one vector
>  pred_combined <- rep(NA, nrow(data))
>  pred_combined[data$RelationshipSegment == 'Husband'] <- pred_husband
>  pred_combined[data$RelationshipSegment == 'Other'] <- pred_other
>  # Calculating MSE
>  mse_combined <- mean((pred_combined - data$HoursPerYear)^2)
>  # Output the MSE
>  cat("Updated Model MSE is", mse_combined, "Benchmark MSE is", benchmark_mse)
Updated Model MSE is 337520.6 Benchmark MSE is 386036.4
```

Categorical Prediction Target: Marital.Status

The prediction model for "marital.status" was developed using a decision tree approach with the rpart package in R, utilizing predictors such as workclass, occupation, and race. Initially, a benchmark decision tree model was built using these predictors across the entire dataset, achieving an accuracy of approximately 51.95%. To enhance the model's performance, the data was segmented into two age groups — "Younger" and "Older" — based on whether individuals were younger or older than 45 years. Separate decision tree models were then constructed for each age segment, hypothesizing that different age groups might exhibit distinct patterns affecting marital status. The predictions from each age-specific model were combined to create a holistic view of marital status across the dataset. This segmented approach improved the accuracy of the predictions to about 56.93%, demonstrating a significant improvement over the benchmark model. This difference also equates to a percentage change of about 9.58%, which when rounded, matches the 10% required as stated in the assignment. Similarly to the numerical value predicted, this percent would most likely be higher if the data set did not contain the thousands of data points that it does.

```
> ##--CREATING BENCHMARK--
>  # Fit a linear model
>  lm_model <- lm(HoursPerYear ~ age + education.num + sex, data = data)
> #--PREDICTING CAT VALUE--
> ##--CREATING BENCHMARK--
>  # Ensure factors are correctly specified
>  data$workclass <- as.factor(data$workclass)
>  data$occupation <- as.factor(data$occupation)
>  data$race <- as.factor(data$race)
>  data$marital.status <- as.factor(data$marital.status)
>  # Build the decision tree model using different predictors
>  marital_status_tree <- rpart(marital.status ~ workclass + occupation + race, data = data, method
= "class")
>  # Predict marital status using the model, retrieving the most probable class
>  predictions <- predict(marital_status_tree, data, type = "class")
>  # Calculate accuracy of the model
>  accuracy <- sum(data$marital.status == predictions) / nrow(data)
>  # Print the model accuracy
>  cat("Accuracy of the decision tree model predicting marital status:", accuracy, "\n")
Accuracy of the decision tree model predicting marital status: 0.5195172
> ##--PREDICTION MODEL--
>  # Segment data based on age
>  data$AgeGroup <- ifelse(data$age >= 45, "Older", "Younger")
>  # Factors must be correctly specified
>  data$AgeGroup <- as.factor(data$AgeGroup)
>  data$workclass <- as.factor(data$workclass)
>  data$occupation <- as.factor(data$occupation)
>  data$race <- as.factor(data$race)
>  # Build decision tree models for each segment
>  model_younger <- rpart(marital.status ~ workclass + occupation + race,
+                        data = data[data$AgeGroup == 'Younger',], method = "class")
>  model_older <- rpart(marital.status ~ workclass + occupation + race,
+                      data = data[data$AgeGroup == 'Older',], method = "class")
>  # Making predictions for each segment
>  pred_younger <- predict(model_younger, newdata = data[data$AgeGroup == 'Younger',], type =
"class")
>  pred_older <- predict(model_older, newdata = data[data$AgeGroup == 'Older',], type = "class")
>  # Combine predictions back into one vector
>  predictions <- rep(NA, nrow(data))
>  predictions[data$AgeGroup == 'Younger'] <- levels(data$marital.status)[pred_younger]
>  predictions[data$AgeGroup == 'Older'] <- levels(data$marital.status)[pred_older]
>  # Calculate accuracy of the combined model
>  newAccuracy <- sum(data$marital.status == predictions) / nrow(data)
>  # Print the combined model accuracy
>  cat("Combined Model Accuracy:", newAccuracy, "vs Benchmark Accuracy:", accuracy)
Combined Model Accuracy: 0.5693007 vs Benchmark Accuracy: 0.5195172
```

1 Step Cross Validation

Numerical Value	Categorical Value
[1] 30918 26957 23564 5880 12143 MSE: 402519.8	[1] 23515 19592 25395 19362 21901 Accuracy: 0.5
[2] 19380 14835 3955 14023 24189 MSE: 310937.9	[2] 24541 26524 8367 5842 21483 Accuracy: 0.55
[3] 15342 10453 5142 31185 23714 MSE: 357136.8	[3] 18354 21385 12913 12550 21309 Accuracy: 0.54
[4] 30069 24710 13640 22768 23555 MSE: 363733.9	[4] 363 6931 3463 8579 6102 Accuracy: 0.46
[5] 21965 29152 9284 3685 10076 MSE: 309857.4	[5] 26994 9976 11348 26324 10587 Accuracy: 0.49