Baseball5_TY

Thomas Young

2025-04-07

Load Hitters DATA

Hitters Data Set from Kaggle: https://www.kaggle.com/datasets/floser/hitters

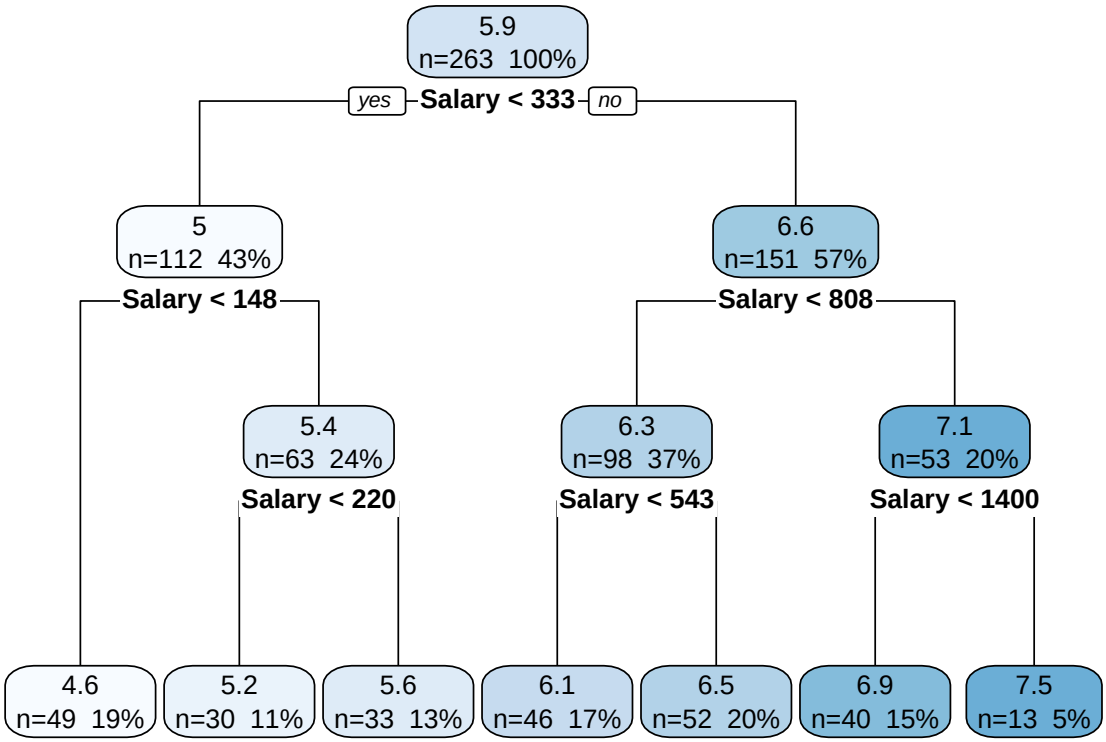
```
hitters <- read_csv("Hitters.csv")
```

```
## Rows: 322 Columns: 20
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr  (3): League, Division, NewLeague
## dbl (17): AtBat, Hits, HmRun, Runs, RBI, Walks, Years, CAtBat, CHits, CHmRun...
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
# clean
hitters <- hitters |>  drop_na()

# Create variable for log salary
hitters <- hitters |>  mutate(LogSalary = log(Salary))
```

Build models

```
set.seed(123)
tree_model <- rpart(LogSalary ~ ., data = hitters,
                    control = rpart.control(cp = 0.01))

rf_model <- randomForest(LogSalary ~ ., data = hitters)

# Visualize the full tree
rpart.plot(tree_model, extra = 101) # extra=101 for node labels
```




```
print(rf_model)
```

```
##
## Call:
##  randomForest(formula = LogSalary ~ ., data = hitters)
##                Type of random forest: regression
##                      Number of trees: 500
## No. of variables tried at each split: 6
##
##           Mean of squared residuals: 0.03549914
##                     % Var explained: 95.49
```

Check Performance

```
# CART R-squared
r_squared <- 1 - (sum((hitters$LogSalary - predict(tree_model, hitters))^2) /
                  sum((hitters$LogSalary - mean(hitters$LogSalary))^2))

# Random forest R-squared
r_squared_rf <- 1 - (sum((hitters$LogSalary - predict(rf_model, hitters))^2) /
                      sum((hitters$LogSalary - mean(hitters$LogSalary))^2))

# Compare R-squared values
print(paste("Regression Tree R-squared:", r_squared))
```

```
## [1] "Regression Tree R-squared: 0.969783114157923"
```

```r
print(paste("Random Forest R-squared:", r_squared_rf))
```

```
## [1] "Random Forest R-squared: 0.991052330662277"
```

```r
# CART RMSE
rmse <- sqrt(mean((hitters$LogSalary - predict(tree_model, hitters))^2))

# # Random forest RMSE
rmse_rf <- sqrt(mean((hitters$LogSalary - predict(rf_model, hitters))^2))

# Compare RMSE values
print(paste("Regression Tree RMSE:", rmse))
```

```
## [1] "Regression Tree RMSE: 0.154274220152242"
```

```r
print(paste("Random Forest RMSE:", rmse_rf))
```

```
## [1] "Random Forest RMSE: 0.0839505355488118"
```

Interpretation

Lower RMSE is better: RMSE (Root Mean Squared Error) measures the average difference between the predicted values and the actual values. The random forest model had a lower RMSE indicates that the model's predictions are closer to the true values.

Improvement in R-squared: The random forest model had a higher R-squared value than the regression tree, suggesting that it explains more of the variance in LogSalary