Boston Housing Analysis

Introduction

This vignette demonstrates how to use the provided functions to analyze the Boston Housing dataset. The dataset contains information on various housing factors in the Boston area, such as crime rate, average number of rooms, and median value of owner-occupied homes.

Load and preprocess the data

```
library(LinearRegress)
boston_data <- load_boston_data("C:/Users/calvi/Documents/CUHK/course_content_new/Term_6/STA3005/Packag
#> Rows: 333 Columns: 15
#> -- Column specification --------------------------------------------------------
#> Delimiter: ","
#> dbl (15): ID, crim, zn, indus, chas, nox, rm, age, dis, rad, tax, ptratio, b...
#>
#> 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.
#> Rows: 173 Columns: 14
#> -- Column specification --------------------------------------------------------
#> Delimiter: ","
#> dbl (14): ID, crim, zn, indus, chas, nox, rm, age, dis, rad, tax, ptratio, b...
#>
#> 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.
#> Rows: 506 Columns: 15
#> -- Column specification --------------------------------------------------------
#> Delimiter: ","
#> dbl (15): ID, crim, zn, indus, chas, nox, rm, age, dis, rad, tax, ptratio, b...
#>
#> 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.
```

Next, preprocess the data by removing rows with missing values, removing the 'ID' column, centering, and scaling the predictor variables using the preprocess_boston_data() function:

```
preprocessed_data <- preprocess_boston_data(boston_data)
preprocessed_data
#> $train_data
#> # A tibble: 333 x 14
#>       crim      zn  indus   chas    nox     rm     age   dis    rad    tax
#>      <dbl>   <dbl>  <dbl>  <dbl>  <dbl>  <dbl>   <dbl> <dbl>  <dbl>  <dbl>
#>  1 -0.456  0.322  -1.28  -0.252 -0.167  0.439 -0.108  0.192 -0.988 -0.663
#>  2 -0.453 -0.471  -0.604 -0.252 -0.767  0.221  0.379  0.635 -0.873 -0.979
```

```
#>  3 -0.453 -0.471  -1.30  -0.252 -0.862  1.04  -0.797  1.19  -0.759 -1.10
#>  4 -0.448 -0.471  -1.30  -0.252 -0.862  1.25  -0.499  1.19  -0.759 -1.10
#>  5 -0.445  0.0799 -0.489 -0.252 -0.288 -0.360 -0.0578 0.934 -0.530 -0.575
#>  6 -0.426  0.0799 -0.489 -0.252 -0.288  0.158  0.927  1.33  -0.530 -0.575
#>  7 -0.441  0.0799 -0.489 -0.252 -0.288 -0.365  0.522  1.27  -0.530 -0.575
#>  8 -0.444  0.0799 -0.489 -0.252 -0.288 -0.535 -1.04   0.879 -0.530 -0.575
#>  9 -0.371 -0.471  -0.451 -0.252 -0.167 -0.450 -0.228  0.504 -0.644 -0.599
#> 10 -0.370 -0.471  -0.451 -0.252 -0.167 -0.241  0.578  0.380 -0.644 -0.599
#> # i 323 more rows
#> # i 4 more variables: ptratio <dbl>, black <dbl>, lstat <dbl>, medv <dbl>
#>
#> $test_data
#> # A tibble: 173 x 14
#>      crim      zn  indus   chas    nox     rm     age    dis    rad    tax
#>     <dbl>   <dbl>  <dbl>  <dbl>  <dbl>  <dbl>   <dbl>  <dbl>  <dbl>  <dbl>
#>  1 -0.453 -0.471  -0.604 -0.252 -0.767  1.31  -0.253  0.635  -0.873 -0.979
#>  2 -0.453 -0.471  -1.30  -0.252 -0.862  0.234 -0.339  1.19   -0.759 -1.10
#>  3 -0.437  0.0799 -0.489 -0.252 -0.288 -0.133  0.991  1.13   -0.530 -0.575
#>  4 -0.428  0.0799 -0.489 -0.252 -0.288 -0.902  1.13   1.20   -0.530 -0.575
#>  5 -0.434  0.0799 -0.489 -0.252 -0.288 -0.372  0.628  1.45   -0.530 -0.575
#>  6 -0.350 -0.471  -0.451 -0.252 -0.167 -0.392  0.479  0.277  -0.644 -0.599
#>  7 -0.358 -0.471  -0.451 -0.252 -0.167 -0.765  0.0453 0.0437 -0.644 -0.599
#>  8 -0.355 -0.471  -0.451 -0.252 -0.167 -0.485  0.920  0.348  -0.644 -0.599
#>  9 -0.343 -0.471  -0.451 -0.252 -0.167 -0.947  0.621  0.376  -0.644 -0.599
#> 10 -0.366 -0.471  -0.451 -0.252 -0.167 -0.643  0.785  0.491  -0.644 -0.599
#> # i 163 more rows
#> # i 4 more variables: ptratio <dbl>, black <dbl>, lstat <dbl>, medv <dbl>
```

Train a linear regression model

Train a linear regression model on the preprocessed training data using the train_linear_regression() function:

```
train_result <- train_linear_regression(preprocessed_data$train_data)
train_result
#> $model
#>
#> Call:
#> lm(formula = medv ~ ., data = train_data)
#>
#> Coefficients:
#> (Intercept)         crim           zn        indus         chas          nox
#>     22.7688      -0.3859       1.0758       0.3769       0.9006      -1.8093
#>          rm          age          dis          rad          tax      ptratio
#>      2.6531      -0.1302      -3.0684       2.8759      -2.1981      -1.8441
#>       black        lstat
#>      1.0101      -4.2429
#>
#>
#> $train_data
#> # A tibble: 333 x 14
#>      crim      zn  indus   chas    nox     rm     age   dis    rad    tax
```

```
#>      <dbl>   <dbl>  <dbl>  <dbl>  <dbl>  <dbl>   <dbl> <dbl>  <dbl>  <dbl>
#>  1 -0.456  0.322  -1.28  -0.252 -0.167  0.439 -0.108  0.192 -0.988 -0.663
#>  2 -0.453 -0.471  -0.604 -0.252 -0.767  0.221  0.379  0.635 -0.873 -0.979
#>  3 -0.453 -0.471  -1.30  -0.252 -0.862  1.04  -0.797  1.19  -0.759 -1.10
#>  4 -0.448 -0.471  -1.30  -0.252 -0.862  1.25  -0.499  1.19  -0.759 -1.10
#>  5 -0.445  0.0799 -0.489 -0.252 -0.288 -0.360 -0.0578 0.934 -0.530 -0.575
#>  6 -0.426  0.0799 -0.489 -0.252 -0.288  0.158  0.927  1.33  -0.530 -0.575
#>  7 -0.441  0.0799 -0.489 -0.252 -0.288 -0.365  0.522  1.27  -0.530 -0.575
#>  8 -0.444  0.0799 -0.489 -0.252 -0.288 -0.535 -1.04   0.879 -0.530 -0.575
#>  9 -0.371 -0.471  -0.451 -0.252 -0.167 -0.450 -0.228  0.504 -0.644 -0.599
#> 10 -0.370 -0.471  -0.451 -0.252 -0.167 -0.241  0.578  0.380 -0.644 -0.599
#> # i 323 more rows
#> # i 4 more variables: ptratio <dbl>, black <dbl>, lstat <dbl>, medv <dbl>
```

Make predictions

Use the trained linear regression model to make predictions on the preprocessed test data with the make_predictions() function:

```
predictions <- make_predictions(preprocessed_data$test_data, train_result)
predictions
#> # A tibble: 173 x 15
#>      crim      zn  indus   chas    nox     rm     age    dis    rad    tax
#>     <dbl>   <dbl>  <dbl>  <dbl>  <dbl>  <dbl>   <dbl>  <dbl>  <dbl>  <dbl>
#>  1 -0.453 -0.471  -0.604 -0.252 -0.767  1.31  -0.253  0.635  -0.873 -0.979
#>  2 -0.453 -0.471  -1.30  -0.252 -0.862  0.234 -0.339  1.19   -0.759 -1.10
#>  3 -0.437  0.0799 -0.489 -0.252 -0.288 -0.133  0.991  1.13   -0.530 -0.575
#>  4 -0.428  0.0799 -0.489 -0.252 -0.288 -0.902  1.13   1.20   -0.530 -0.575
#>  5 -0.434  0.0799 -0.489 -0.252 -0.288 -0.372  0.628  1.45   -0.530 -0.575
#>  6 -0.350 -0.471  -0.451 -0.252 -0.167 -0.392  0.479  0.277  -0.644 -0.599
#>  7 -0.358 -0.471  -0.451 -0.252 -0.167 -0.765  0.0453 0.0437 -0.644 -0.599
#>  8 -0.355 -0.471  -0.451 -0.252 -0.167 -0.485  0.920  0.348  -0.644 -0.599
#>  9 -0.343 -0.471  -0.451 -0.252 -0.167 -0.947  0.621  0.376  -0.644 -0.599
#> 10 -0.366 -0.471  -0.451 -0.252 -0.167 -0.643  0.785  0.491  -0.644 -0.599
#> # i 163 more rows
#> # i 5 more variables: ptratio <dbl>, black <dbl>, lstat <dbl>, medv <dbl>,
#> #   pred_medv <dbl>
```

Evaluate the model

Calculate the Mean Squared Error (MSE) of the predictions using the evaluate_model() function:

```
mse <- evaluate_model(predictions)
print(mse)
#> [1] 22.73999
```

Perform k-Fold Cross-Validation

Perform k-Fold cross-validation on the linear regression model to obtain the average MSE across all folds using the cross_validate() function:

```
avg_mse <- cross_validate(preprocessed_data$train_data, 5)
#> Loading required package: ggplot2
#> Loading required package: lattice
print(avg_mse)
#> [1] 25.03804
```

Calculate the R-squared Value

Calculate the R-squared value of the linear regression model using the calculate_r_squared() function:

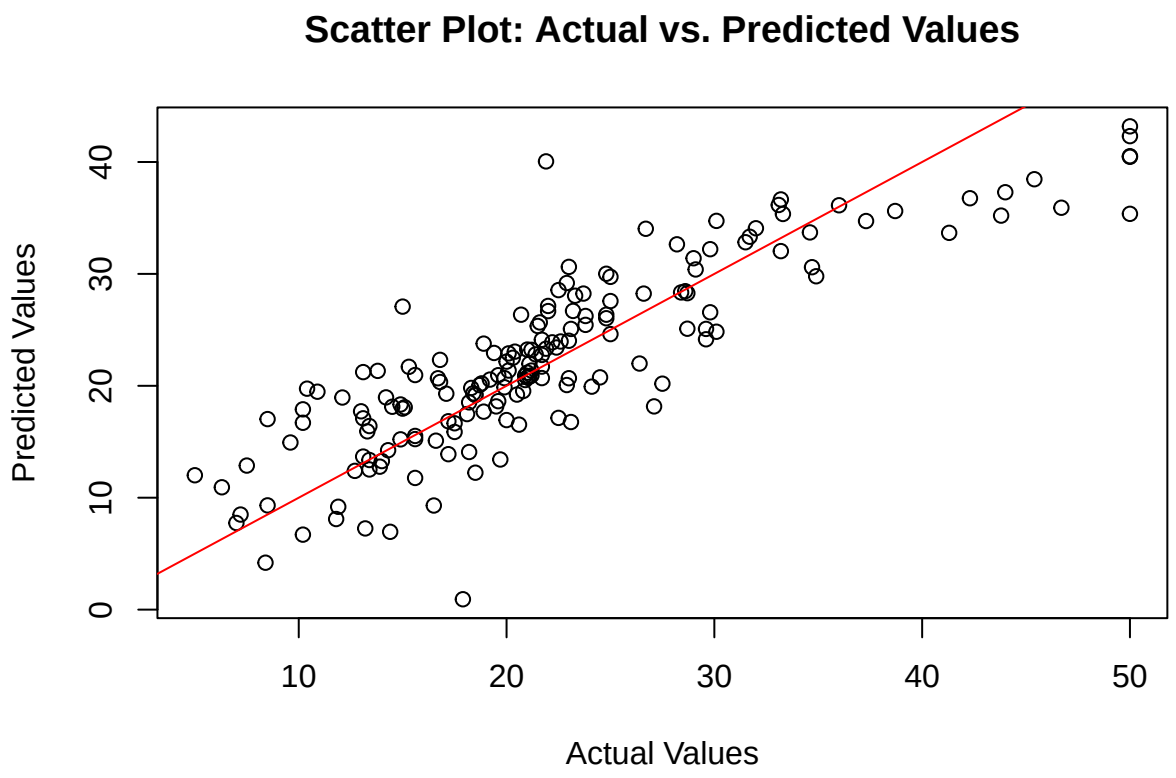
```
r_squared_value <- calculate_r_squared(train_result$model, preprocessed_data$test_data)
print(r_squared_value)
#> [1] 0.7328056
```

Visualizations

Scatter plot of Actual vs. Predicted Values

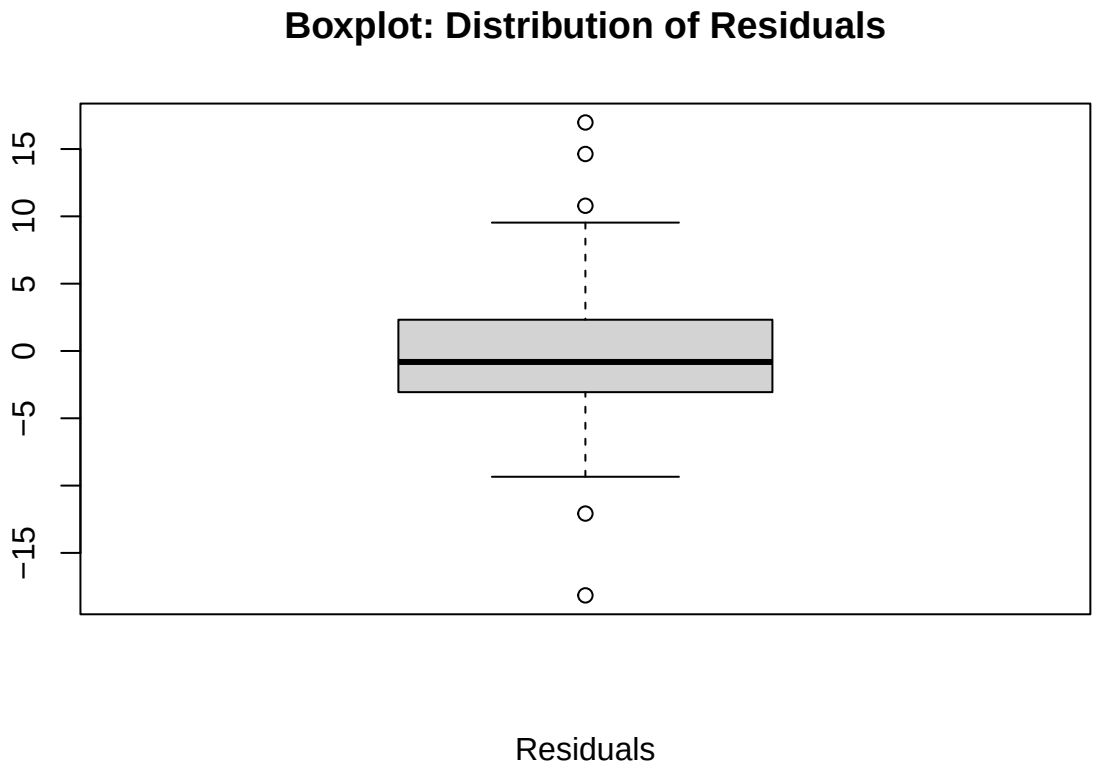
Create a scatter plot comparing actual and predicted values of the target variable using the scatter_actual_vs_predicted() function:

```
actual_values <- preprocessed_data$test_data$medv
predicted_values <- predictions$pred_medv
scatter_actual_vs_predicted(actual_values, predicted_values)
```



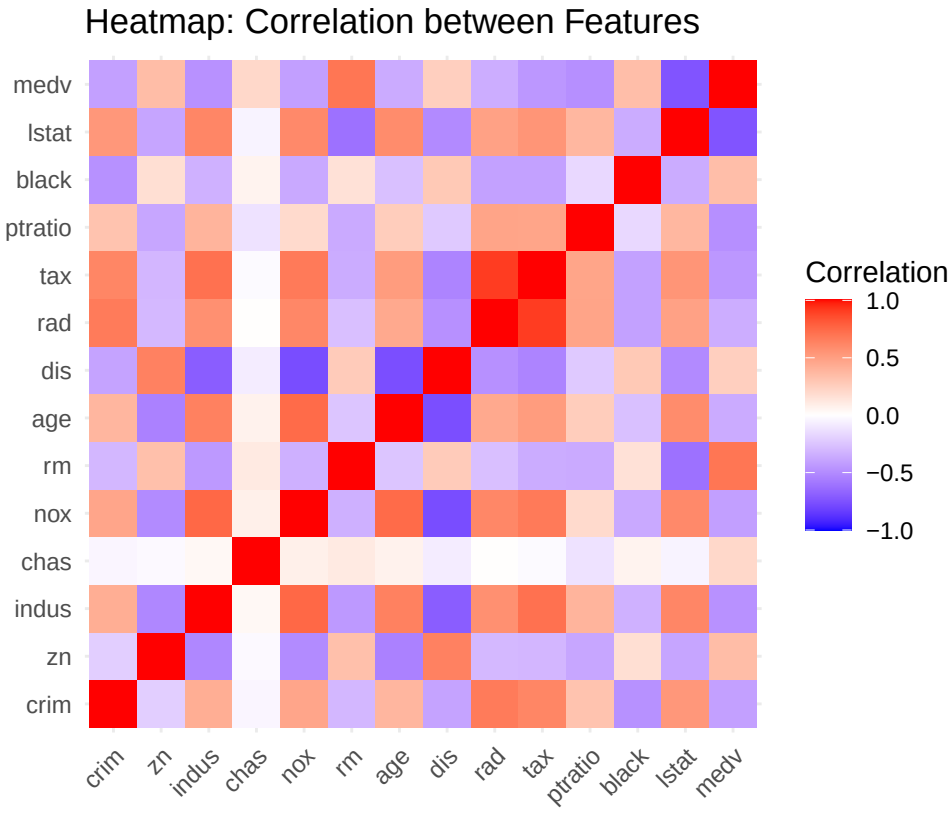
Boxplot of Residuals Create a boxplot displaying the distribution of residuals using the boxplot_residuals() function:

```
residuals <- actual_values - predicted_values
boxplot_residuals(residuals)
```



Heatmap of Correlation between Features Create a heatmap displaying the correlation between features in the dataset using the heatmap_correlation() function:

```
heatmap_correlation(preprocessed_data$train_data)
```


Heatmap: Correlation between Features
medv
lstat
black
ptratio
tax
rad
dis
age
rm
nox
chas
indus
zn
crim
crim
zn
indus
chas
nox
rm
age
dis
rad
tax
ptratio
black
lstat
medv
Correlation
1.0
0.5
0.0
−0.5
−1.0