Models

```r
library(caret)
```

```
Loading required package: ggplot2

Loading required package: lattice
```

```r
library(readr)

Air_Quality <- read_csv("/Users/yegireddimounika/Desktop/AirQuality/Air_Quality_CleanedData.c
```

```
Rows: 5811 Columns: 15

-- Column specification --------------------------------------------------------
Delimiter: ","
dbl (15): RecordID, AQI, PM10, PM2_5, NO2, SO2, O3, Temperature, Humidity, W...

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.
```

```r
# Handle missing values
Air_Quality <- na.omit(Air_Quality)

# Split data into training and testing sets
set.seed(123)
sample_indices <- sample(1:nrow(Air_Quality), size = 0.7 * nrow(Air_Quality))
train_data <- Air_Quality[sample_indices, ]
test_data <- Air_Quality[-sample_indices, ]

# Enhanced Linear Regression Model
```

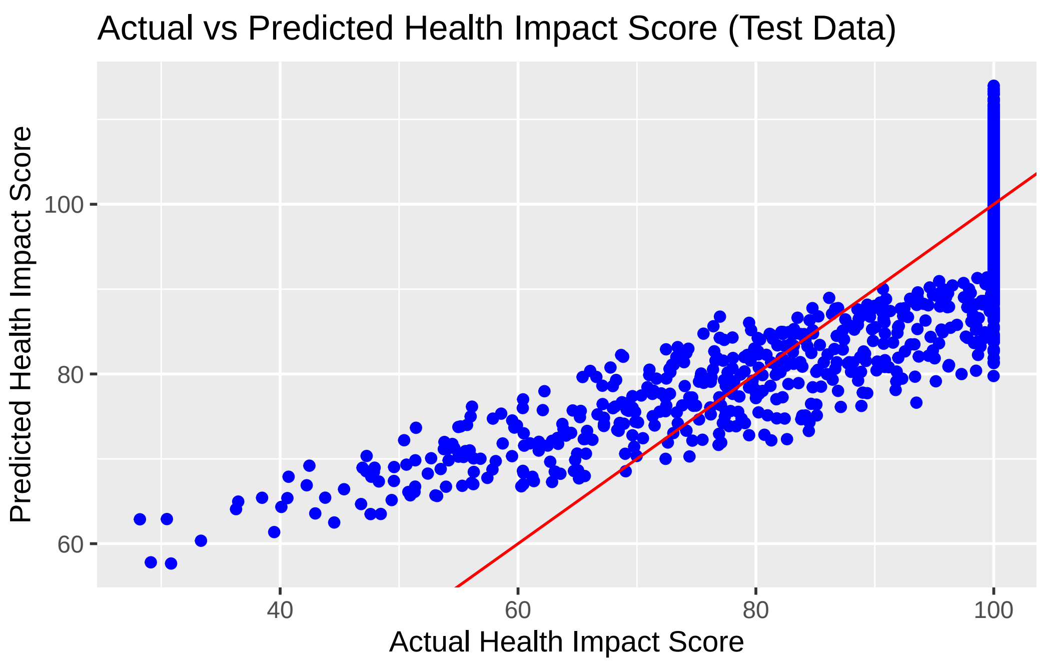
```r
lm_model <- lm(HealthImpactScore ~ AQI + PM2_5 + PM10 + NO2 + SO2 + O3 +
               I(AQI^2) + I(PM2_5 * NO2), data = train_data)

# Model summary
summary(lm_model)
```

```
Call:
lm(formula = HealthImpactScore ~ AQI + PM2_5 + PM10 + NO2 + SO2 +
    O3 + I(AQI^2) + I(PM2_5 * NO2), data = train_data)

Residuals:
    Min      1Q  Median      3Q     Max
-35.996  -4.198   0.411   5.069  23.282

Coefficients:
                 Estimate Std. Error t value Pr(>|t|)
(Intercept)     4.823e+01  6.688e-01  72.123  < 2e-16 ***
AQI             2.156e-01  3.193e-03  67.535  < 2e-16 ***
PM2_5           7.116e-02  4.028e-03  17.667  < 2e-16 ***
PM10            2.624e-02  1.359e-03  19.314  < 2e-16 ***
NO2             4.696e-02  3.953e-03  11.880  < 2e-16 ***
SO2             1.657e-02  4.066e-03   4.076 4.67e-05 ***
O3              2.482e-02  1.342e-03  18.489  < 2e-16 ***
I(AQI^2)       -3.207e-04  6.190e-06 -51.811  < 2e-16 ***
I(PM2_5 * NO2) -1.822e-04  3.415e-05  -5.334 1.01e-07 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 7.385 on 4058 degrees of freedom
Multiple R-squared:  0.6928,    Adjusted R-squared:  0.6922
F-statistic:  1144 on 8 and 4058 DF,  p-value: < 2.2e-16
```

```r
# Predict and add to test data
predictions <- predict(lm_model, newdata = test_data, interval = "confidence")
test_data$PredictedHealthImpactScore <- as.numeric(predictions[, "fit"])

# Calculate RMSE
test_data <- na.omit(test_data)
rmse <- sqrt(mean((test_data$HealthImpactScore - test_data$PredictedHealthImpactScore)^2))
print(paste("Model RMSE:", rmse))
```

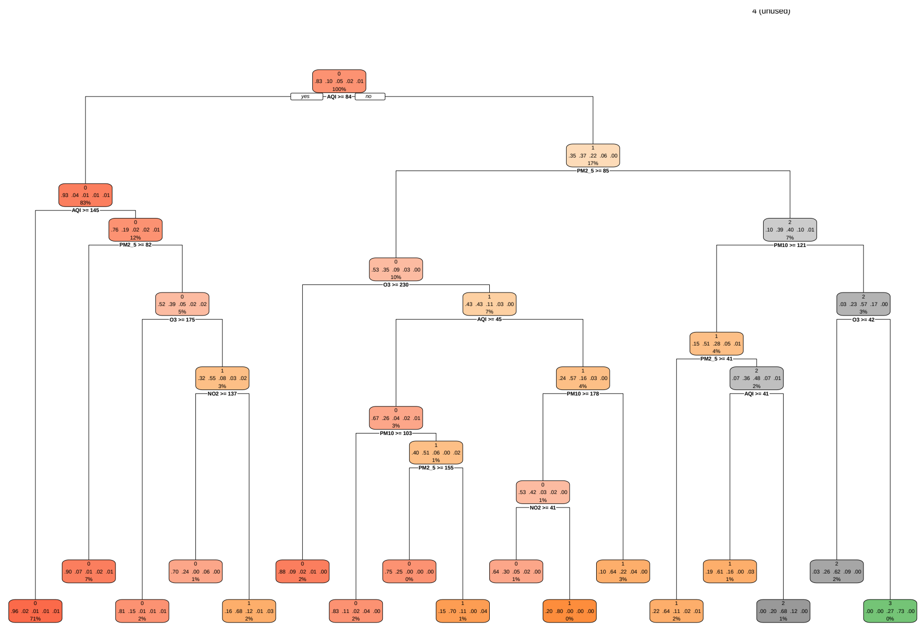
```
[1] "Model RMSE: 7.24565551810403"
```

```
# Plot actual vs predicted values
ggplot(test_data, aes(x = HealthImpactScore, y = PredictedHealthImpactScore)) +
  geom_point(color = "blue", na.rm = TRUE) +
  geom_abline(intercept = 0, slope = 1, color = "red") +
  ggtitle("Actual vs Predicted Health Impact Score (Test Data)") +
  xlab("Actual Health Impact Score") +
  ylab("Predicted Health Impact Score")
```




```
# Load necessary libraries
library(caret)
library(rpart)
library(rpart.plot)

# Prepare the data
set.seed(123) # For reproducibility

# Split the dataset into training (70%) and testing (30%)
train_index <- createDataPartition(Air_Quality$HealthImpactClass, p = 0.7, list = FALSE)
train_data <- Air_Quality[train_index, ]
test_data <- Air_Quality[-train_index, ]
```

```r
# Train the Decision Tree model
tree_model <- rpart(
  HealthImpactClass ~ AQI + PM10 + PM2_5 + NO2 + SO2 + O3 + Temperature + Humidity + WindSpe
  data = train_data,
  method = "class"
)

# Visualize the Decision Tree
rpart.plot(tree_model)
```



```r
# Ensure the target variable in the test set is a factor
test_data$HealthImpactClass <- factor(test_data$HealthImpactClass)

# Generate predictions
predictions <- predict(tree_model, test_data, type = "class")

# Ensure predictions are factors with the same levels as the actual target variable
predictions <- factor(predictions, levels = levels(test_data$HealthImpactClass))

# Evaluate the model
conf_matrix <- confusionMatrix(predictions, test_data$HealthImpactClass)
```

```r
# Print the confusion matrix
print("Confusion Matrix:")
```

```
[1] "Confusion Matrix:"
```

```r
print(conf_matrix)
```

```
Confusion Matrix and Statistics

          Reference
Prediction    0    1    2    3    4
         0 1370   77   22   15   21
         1   44   88   25    2    4
         2    5   25   31    8    0
         3    0    0    5    1    0
         4    0    0    0    0    0

Overall Statistics

               Accuracy : 0.8548
                 95% CI : (0.8374, 0.8711)
    No Information Rate : 0.8141
    P-Value [Acc > NIR] : 3.919e-06

                  Kappa : 0.4906

 Mcnemar's Test P-Value : NA

Statistics by Class:

                     Class: 0 Class: 1 Class: 2  Class: 3 Class: 4
Sensitivity            0.9655  0.46316  0.37349 0.0384615  0.00000
Specificity            0.5833  0.95171  0.97711 0.9970879  1.00000
Pos Pred Value         0.9103  0.53988  0.44928 0.1666667      NaN
Neg Pred Value         0.7941  0.93544  0.96894 0.9856074  0.98566
Prevalence             0.8141  0.10901  0.04762 0.0149168  0.01434
Detection Rate         0.7860  0.05049  0.01779 0.0005737  0.00000
Detection Prevalence   0.8635  0.09352  0.03959 0.0034423  0.00000
Balanced Accuracy      0.7744  0.70743  0.67530 0.5177747  0.50000
```

```r
# Model Accuracy
accuracy <- conf_matrix$overall["Accuracy"]
print(paste("Model Accuracy:", round(accuracy, 2)))
```

```
[1] "Model Accuracy: 0.85"
```