Own project: Biomechanical Features of Orthopedic Patients

Edx Data Science Capstone Project - HarvardX PH125.9x

Charles Leung

2025-12-09

Contents

1 Introduction

This report starts with outlining the background and objectives, followed by the data preparation. After that, an exploratory analysis is presented and show the development of the two machine learning models, logistic regression and random forest, for which both of them can perform binary classification. The models are expected to predict whether a patient has a normal spine or an orthopedic issues based on different biomechanical measurements. The accuracy of the predicted results are then obtained, with the conclusion part to wrap up the whole project.

1.1 Overview

Spinal issues have been common among humans. As many problems tackling spine conditions remain unsolved, and spine is an essential to human mechanics, an expanding number of biomechanical research has conducted focusing on the spine (Oxland, 2016).

Orthopedic patients, who encounter conditions in their musculoskeletal system including bones and muscles, often exhibit several affected biomechanical features in pelvis orientation or spinal alignment. The severity or degree of exhibiting the altered characteristics facilitates in the classification of the patients, whether or not they are diagnosed with specific conditions related to spine.

In this project, the dataset is retrieved from Kaggle, with both Disk Hernia and Spondylolisthesis conditions group into a single category called "abnormal", so that the patients are only classified into two groups, a "normal" group with 100 patients or an "abnormal" group with 210 patients.

1.2 Objective

The aim of this project is to develop and compare at least two machine learning models that classify if a patient has spinal abnormality using only non-invasive biomechanical measurements. The higher the accuracy of the model, the greater confidence we have to determine the clinical status of the patient.

1.3 Key Steps

To start with, the dataset is loaded from the online source. The dataset is then explored and visualized to observe any special aspects, and is subsequently split into a training set and a test set to help developing the prediction models. The results of the models are analyzed from the corresponding confusion matrices. It is expected that both models are comparable to each other after using the test set for prediction.

2 Methods and Analysis

2.1 Creating dataset from online sources

The required dataset in this project can be downloaded to R using the "RKaggle" package, and the website providing the dataset is:

https://www.kaggle.com/datasets/uciml/biomechanical-features-of-orthopedic-patients

```
# Installing and loading the required libraries
if(!require(tidyverse)) install.packages("tidyverse", repos = "http://cran.us.r-project.org")
if(!require(caret)) install.packages("caret", repos = "http://cran.us.r-project.org")
if(!require(randomForest)) install.packages("randomForest", repos = "http://cran.us.r-project.org")
if(!require(RKaggle)) install.packages("RKaggle", repos = "http://cran.us.r-project.org")

library(tidyverse)
library(caret)
library(randomForest)
library(RKaggle)

set.seed(2025)
```

2.1.1 Direct download from Kaggle

As we are using one dataset only (patients classified into two classes only), one list is extracted from the two datasets.

```
# Downloading the dataset
datasets <- get_dataset("uciml/biomechanical-features-of-orthopedic-patients")
df_two_class <- datasets[[1]]
```

2.1.2 Creating binary target and rename a column

The dataset is added with a "target" column so as to prepare the data as a type of factor for binary classification.

```
# Adding a "target" column
ortho <- df_two_class %>%
  mutate(target = factor(if_else(class == "Normal", 0, 1),
                         levels = c(0, 1),
                         labels = c("Normal", "Abnormal")))
```

The column "pelvic_tilt numeric" is renamed to "pelvic_tilt_numeric", just to replace the space with an underscore for smooth coding.

```
# Adding an underscore
ortho <- ortho %>%
  rename(pelvic_tilt_numeric = `pelvic_tilt numeric`)
```

2.2 Exploring the data with visualization

Before building the models, let's explore what is inside the dataset.

```
# Exploring the data
glimpse(ortho)
```

```
## Rows: 310
## Columns: 8
## $ pelvic_incidence         <dbl> 63.02782, 39.05695, 68.83202, 69.29701, 49.71~
## $ pelvic_tilt_numeric      <dbl> 22.552586, 10.060991, 22.218482, 24.652878, 9~
## $ lumbar_lordosis_angle    <dbl> 39.60912, 25.01538, 50.09219, 44.31124, 28.31~
## $ sacral_slope             <dbl> 40.47523, 28.99596, 46.61354, 44.64413, 40.06~
## $ pelvic_radius            <dbl> 98.67292, 114.40543, 105.98514, 101.86850, 10~
## $ degree_spondylolisthesis <dbl> -0.2544000, 4.5642586, -3.5303173, 11.2115234~
## $ class                    <chr> "Abnormal", "Abnormal", "Abnormal", "Abnormal~
## $ target                   <fct> Abnormal, Abnormal, Abnormal, Abnormal, Abnor~
```

In the previous section (2.1.2), a "target" column is added. You can now see that our dataset has 310 rows and 8 columns, including the new "target" column, which is essentially the same as the class column but only converted to factor with labels "Normal" and "Abnormal".

For the headers of the first 6 columns, they are the biomechanical measurements based on the form of pelvis and lumbar spine: "pelvic incidence", "pelvic tilt", "lumbar lordosis angle", "sacral slope", "pelvic radius", and "grade of spondylolisthesis". Each row represents the data of a patient (there are 310 patients in total).

Some general characteristics of this dataset will be illustrated later. Before that, data wrangling is performed to reshape the wide data (each row represents one patient including all biomechanical measurements) to tidy data (each row representing one biomechanical measurement) for plotting convenience.

The code below describes how the data is converted. The "class" column of the "ortho" data is first removed, and the "target" column showing the patient's status will not be gathered. Each biomechanical attribute (the original column name) becomes a categorical variable and is assigned to a new column "feature", and their corresponding numeric values are assigned to the new column "value".

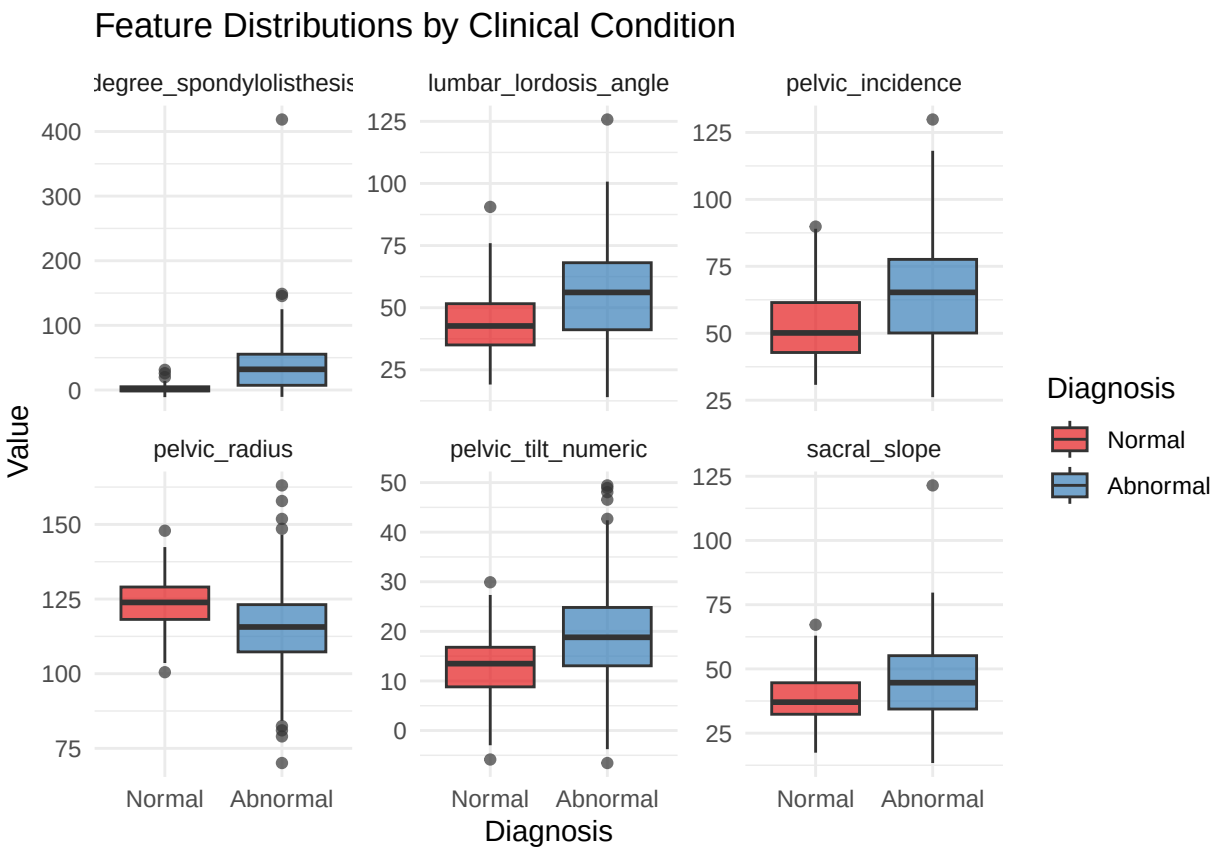
```
# Data wrangling, from wide data to tidy data
ortho_plot <- ortho %>%
  select(-class) %>%
  pivot_longer(-target, names_to = "feature", values_to = "value")

head(ortho_plot)
```

```
## # A tibble: 6 x 3
##   target   feature                   value
##   <fct>    <chr>                     <dbl>
## 1 Abnormal pelvic_incidence         63.0
## 2 Abnormal pelvic_tilt_numeric      22.6
## 3 Abnormal lumbar_lordosis_angle    39.6
## 4 Abnormal sacral_slope             40.5
## 5 Abnormal pelvic_radius            98.7
## 6 Abnormal degree_spondylolisthesis -0.254
```

Below shows the boxplots demonstrating each of the features that stratified by the diagnostic results of the patients. Each facet describes a single feature with a free y axis allowing different measurement ranges. This process highlights the differences of features measured between the healthy and abnormal individuals.

```
# Data visualization
ortho_plot %>% ggplot(aes(x = target, y = value, fill = target)) +
  geom_boxplot(alpha = 0.7) +
  facet_wrap(~feature, scales = "free_y") +
  theme_minimal() +
  labs(title = "Feature Distributions by Clinical Condition",
       x = "Diagnosis",
       y = "Value",
       fill = "Diagnosis") +
  scale_fill_brewer(palette = "Set1")
```



2.2.1 Key insights

From the graph visualized above, several observations can be made. With the exception of pelvic_radius, all biomechanical features exhibit higher values in the abnormal group (blue) compared to the normal group (red). The abnormal group generally has greater variability, including a wider interquartile range (IQR), longer whiskers, and more extreme outliers. Among all the attributes, the degree of spondylolisthesis shows the clearest separation, while other features exhibit certain level of overlapping between the two groups.

Consequently, the classification models generated from the later sections can be used to predict the patient's conditions by making good use of the discrepancies between groups.

2.3 Subsequent modelling approaches

As mentioned in Section 1.3 (Key Steps), there will be approaches to develop models for predictions, but first, let's process the Kaggle dataset and work towards the goal step-by-step.

2.3.1 Create a test set from the Kaggle dataset

To develop the models and understand how well they perform, the dataset is first separated into a training set and a test set using a test index. Since there is limited data (310 observations in this project), the dataset is split into two subsets: 80% forms the training set and the remaining 20% is the test set. This provides a reasonably sufficient training data (248 cases) for model training while keeping enough size for testing the algorithms (62 cases) by metrics performance.

```r
# Splitting the data
test_index <- createDataPartition(y = ortho$target, times = 1, p = 0.2, list = FALSE)
train_set <- ortho[-test_index, ]
test_set  <- ortho[test_index, ]
```

Here shows the proportion of normal and abnormal groups for both the training set and the test set:

```r
# Checking the proportion of the data
prop.table(table(train_set$target))
```

```
##
##    Normal  Abnormal
## 0.3225806 0.6774194
```

```r
prop.table(table(test_set$target))
```

```
##
##    Normal  Abnormal
## 0.3225806 0.6774194
```

It appears that both training set and test set have identical class proportions, ensuring that both sets follow the same distributions as the original dataset and thus, providing an unbiased estimates of the performance metrics.

2.3.2 Creating the logistic regression model

The first model we are going to construct is a logistic regression model. Since our outcome is categorical ("Normal" or "Abnormal"), logistic regression is an appropriate choice for binary classification tasks. Unlike linear regression, which predicts continuous numerical values, logistic regression predicts the probability that a patient belongs to one of the classes.

Specifically, the model works by modelling the log-odds (logit function, natural logarithm of the odds), then applying a sigmoid (logistic) function that transforms the logit back to a probability within 0 to 1. The final prediction is generated by selecting the group with higher probability. That is, if the probability is greater than 0.5, the sample is considered as "Abnormal"; otherwise, it is regarded as "Normal".

The following code shows how a logistic regression model is built. In R, the model can be fitted using the function "glm", generalized linear models. Compared to logistic regression, this function is more general so the desired model should be specified (in this case, binomial):

```r
# Generalized linear models
glm_fit <- train_set %>%
  glm(target ~ pelvic_incidence + pelvic_tilt_numeric +
        lumbar_lordosis_angle + sacral_slope +
        pelvic_radius + degree_spondylolisthesis,
      data = .,
      family = "binomial")
```

Here shows the summary of the model, including the coefficients such as standard error, z-value and probability of greater than the absolute z-value:

```r
# Summary of glm model
summary(glm_fit)
```

```
##
## Call:
## glm(formula = target ~ pelvic_incidence + pelvic_tilt_numeric +
##     lumbar_lordosis_angle + sacral_slope + pelvic_radius + degree_spondylolisthesis,
##     family = "binomial", data = .)
##
## Coefficients:
##                            Estimate Std. Error z value Pr(>|z|)
## (Intercept)               1.630e+01  3.877e+00   4.205 2.61e-05 ***
## pelvic_incidence         -2.826e+07  5.262e+07  -0.537    0.591
## pelvic_tilt_numeric       2.826e+07  5.262e+07   0.537    0.591
## lumbar_lordosis_angle     2.345e-03  2.807e-02   0.084    0.933
## sacral_slope              2.826e+07  5.262e+07   0.537    0.591
## pelvic_radius            -1.168e-01  2.729e-02  -4.279 1.88e-05 ***
## degree_spondylolisthesis  1.965e-01  3.150e-02   6.236 4.48e-10 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 311.88  on 247  degrees of freedom
## Residual deviance: 123.55  on 241  degrees of freedom
## AIC: 137.55
##
## Number of Fisher Scoring iterations: 8
```

2.3.3 Creating the random forest model

Regarding the second model, random forest improves the prediction model by taking the average of multiple decision trees after predicting categorical outcomes. The name of this model actually comes from two features: the "random" generation of samples using the bootstrap method, in which different trees are generated using the same set of data but the samples are drawn repeatedly with replacement and randomly, and the tree combinations will form the "forest".

Here shows the random forest model in R. The function "randomForest" is used after we have downloaded the package with the identical name.

From the code below, "mtry", the number of variables randomly sampled at each split when building a tree, can be tuned to give the best evaluation metrics. Cross-validation is used as the resampling method here for model evaluation with the help of the function "trainControl".

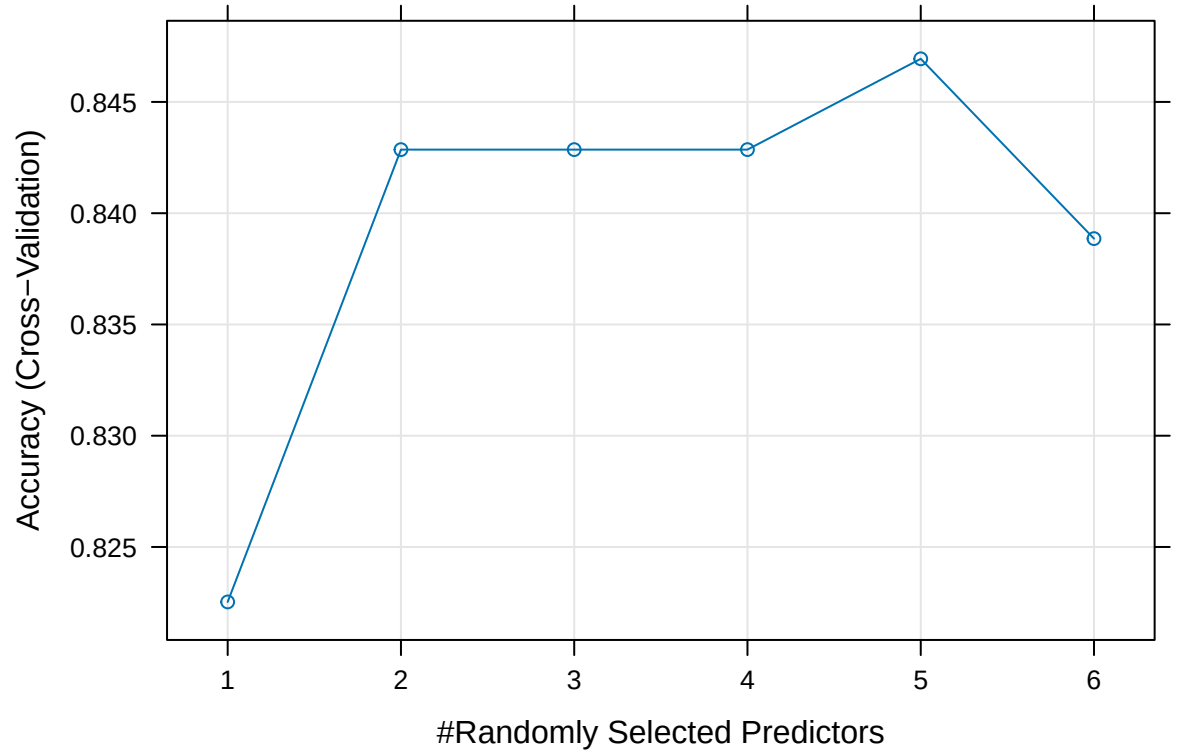
```r
# Cross validation
control <- trainControl(method = "cv", number = 5) # five-fold cross validation

# Tune mtry
mtry_grid <- data.frame(mtry = 1:6)

set.seed(2025)
train_rf <- train(target ~ pelvic_incidence + pelvic_tilt_numeric +
                          lumbar_lordosis_angle + sacral_slope +
```

```
                        pelvic_radius + degree_spondylolisthesis,
                    data = train_set,
                    method = "rf",
                    tuneGrid = mtry_grid,
                    trControl = control,
                    importance = TRUE)

plot(train_rf)
```




```
best_mtry <- train_rf$bestTune
best_mtry
```

```
##   mtry
## 5    5
```

After running the above code, the best mtry is 5.

To fit in the model:

```
# Random forest model
fit_rf <- randomForest(target ~ pelvic_incidence + pelvic_tilt_numeric +
                         lumbar_lordosis_angle + sacral_slope +
                         pelvic_radius + degree_spondylolisthesis,
                    data = train_set,
                    mtry = best_mtry$mtry,
                    importance = TRUE)
```

3 Results

We have generated both models in Section 2.3. Now, we are going to present the performance of the trained models and summarize their predictive metrics in this result part.

3.1 Prediction models

The predictions of both models can be obtained using the "predict" function. Both the predicted status and the probabilities for every patient in test set are recorded. The part of the code 'type = "response"' in the logistic regression prediction is required for generating the conditional probabilities as the logistic transformed values are the default settings.

```
# Logistic regression predictions using test_set
pred_glm_prob <- predict(glm_fit, newdata = test_set, type = "response")
pred_glm <- factor(if_else(pred_glm_prob > 0.5, "Abnormal", "Normal"),
                   levels = c("Normal", "Abnormal"))
```

```
# Random forest predictions
pred_rf_prob <- predict(fit_rf, newdata = test_set, type = "prob")[, "Abnormal"]
pred_rf <- predict(fit_rf, newdata = test_set)
```

3.2 Confusion matrices

In this section, the classification prediction results are visualized using the confusion matrix, in which the combinations between the predicted and the actual values are tabulated.

The following codes compute the confusion matrix for both logistic regression and random forest models on the test set, comparing the predicted values (pred_glm/pred_rf) to the true diagnosis (test_set$target), with the positive results as the "abnormal" cases.

```
# Confusion matrices
cm_glm <- confusionMatrix(data = pred_glm,
                          reference = test_set$target,
                          positive = "Abnormal")

cm_rf  <- confusionMatrix(data = pred_rf,
                          reference = test_set$target,
                          positive = "Abnormal")
```

After that, the area under the Receiver Operating Characteristic (ROC, illustrating the binary classification model performance), AUC, are computed for both models so that the overall performance of how well a model can distinguish between classes can be summarized.

```
# AUC
glm_auc <- pROC::auc(response = test_set$target, predictor = pred_glm_prob)
rf_auc <- pROC::auc(response = test_set$target, predictor = pred_rf_prob)
```

Several evaluation matrices are shown by running the code below:

```r
# Summarizing the results
results <- tibble(
  Model = c("Logistic Regression", "Random Forest"),
  Accuracy = c(cm_glm$overall["Accuracy"], cm_rf$overall["Accuracy"]),
  Sensitivity = c(cm_glm$byClass["Sensitivity"], cm_rf$byClass["Sensitivity"]),
  Specificity = c(cm_glm$byClass["Specificity"], cm_rf$byClass["Specificity"]),
  F1 = c(cm_glm$byClass["F1"], cm_rf$byClass["F1"]),
  AUC = c(as.numeric(glm_auc), as.numeric(rf_auc))) %>%
  mutate(across(where(is.numeric), ~round(., 3)))

results %>%
  knitr::kable(caption = "Comparison of test set performance")
```

Table 1: Comparison of test set performance

Model	Accuracy	Sensitivity	Specificity	F1	AUC
Logistic Regression	0.758	0.786	0.7	0.815	0.864
Random Forest	0.758	0.833	0.6	0.824	0.855

As shown in the table above, a summary of the model performances on the test set is provided. 5 classification matrices (accuracy, sensitivity, specificity, F1, and AUC) are extracted. For your information, F1-score describes the harmonic average of precision, the proportion of the predicted patients having spine abnormalities are actually "abnormal", and recall, the proportion of real "abnormal" patients that the model accurately classified. This metric is useful for the imbalanced datasets, as it can be adjusted by weighing the sensitivity and specificity in different cases.

Both models achieve same accuracy (0.758), with random forest model having higher sensitivity while logistic regression model has a higher specificity. In other words, random forest model is better at detecting patients with actual orthopedic abnormalities, but at the same time, it has a higher probability of diagnosing healthy individuals with the spine conditions (false positive). Thus, random forest model is better used for medical screening purpose.

As for the F1-score, random forest is slightly higher than that of logistic regression model, showing that random forest should be adopted when the balance between precision and recall is considered. In addition, both models attain excellence in the ability of distinguishing the clinical status of the patients, with logistic regression model performing slightly better.

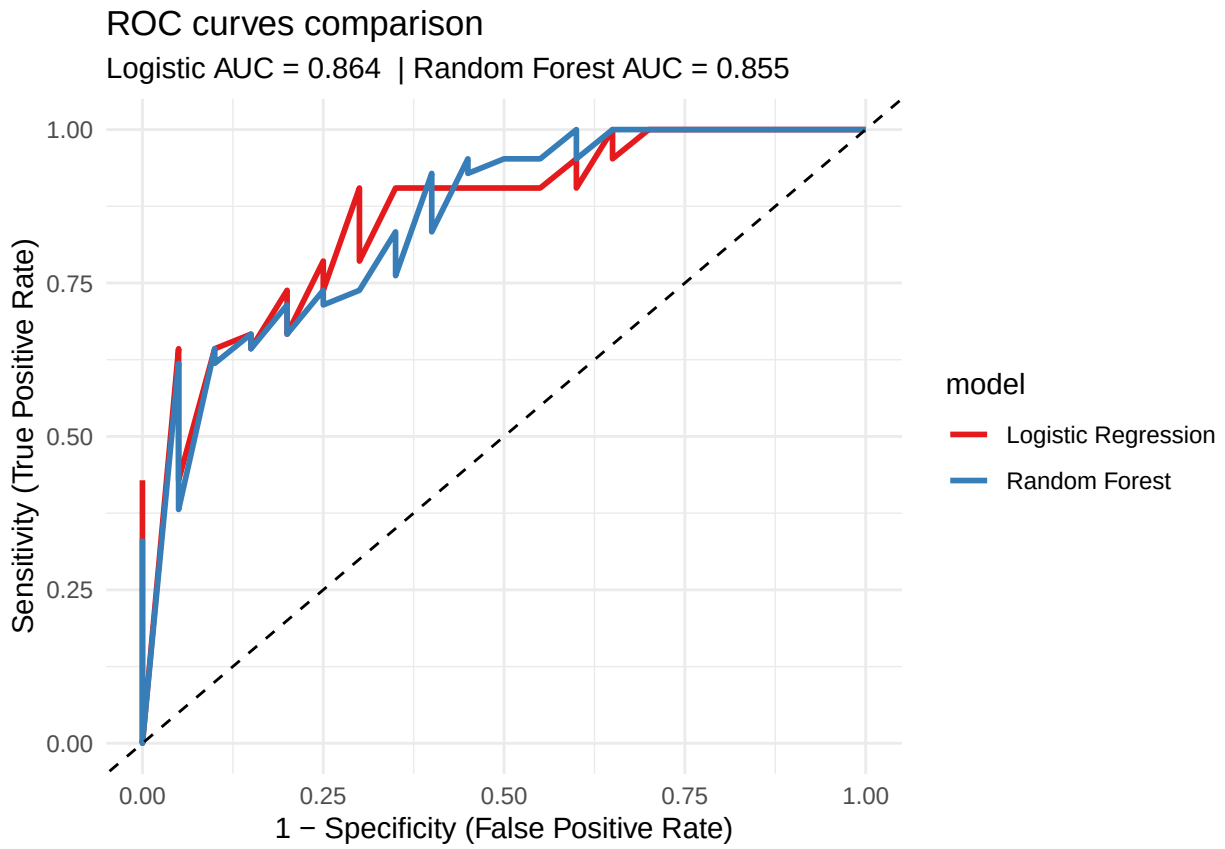
The following code demonstrates how the required values or sensitivity and specificity for both ROC curves are withdrawn for the sake of visualizing the AUC.

```r
# ROC curves
roc_glm <- pROC::roc(test_set$target, pred_glm_prob)
roc_rf <- pROC::roc(test_set$target, pred_rf_prob)

# Extracting sensitivities and specificities for the ROC objects
roc_data <- bind_rows(tibble(sensitivities = roc_glm$sensitivities,
                             specificities = roc_glm$specificities,
                             model = "Logistic Regression"),
                      tibble(sensitivities = roc_rf$sensitivities,
                             specificities = roc_rf$specificities,
                             model = "Random Forest"))
```

Here illustrates the ROC curves for both model after predictions, with the diagonal dotted line passing through the origin as the random classifier (guessing by chance):

```r
# Side-by-side ROC curves
roc_data %>% ggplot(aes(x = 1 - specificities, y = sensitivities, color = model)) +
  geom_line(linewidth = 1) +
  geom_abline(linetype = "dashed", color = "black", alpha = 1) +
  theme_minimal() +
  labs(title = "ROC curves comparison",
       subtitle = paste("Logistic AUC =", round(glm_auc, 3),
                        " | Random Forest AUC =", round(rf_auc, 3)),
       x = "1 - Specificity (False Positive Rate)",
       y = "Sensitivity (True Positive Rate)") +
  scale_color_brewer(palette = "Set1")
```



Both of the lines are yielding a point at the top left corner, which is the perfect classification. Therefore, both models can predict the patient's status pretty well.

4 Conclusion

By conducting this project, two machine learning models, logistic regression and random forest, are developed. Their performances are compared based on classifying orthopedic patients as "normal" or "abnormal" using 6 biomechanical features from a Kaggle dataset. The accuracy for both models are identical, with random forest edged out in sensitivity and F1-score, while logistic regression demonstrates superior specificity and AUC. This implies that ensemble approaches (using multiple small models to obtain a better estimate), such as random forest, possess great efficacy in this kind of data.

The possible effects are providing evidences of using these models for early screening in some medical disorders like orthopedic problems. This non-invasive method can reduce the reliance on the costly medical procedures like imaging scans, and hence, demonstrating the foundation of broader applications of machine learning for early detection in medical care.

However, there are some limitations in this project. The size of this dataset is relatively small, with only 310 observations included. The diversity of the population may be limited and biased to a specific populations as well. Moreover, the biomechanical aspects are assumed to be affecting the orthopedic situations, thus having the possibility of overcoming some confounding effects by the individuals themselves.

In light of this, future research could incorporate larger and multi-source datasets for enhancing model robustness and achieving more accurate predictions. Collaboration between data scientists and medical experts could ensure that advancements in data science results in improvements of diagnosis and treatment of orthopedic conditions.

5 References

Oxland T. R. (2016). Fundamental biomechanics of the spine–What we have learned in the past 25 years and future directions. Journal of biomechanics, 49(6), 817–832. https://doi.org/10.1016/j.jbiomech.2015.10.035

UCI Machine Learning. (n.d.). *Biomechanical features of orthopedic patients* [Data set]. Kaggle. https://www.kaggle.com/datasets/uciml/biomechanical-features-of-orthopedic-patients