Orthopedic Patients Diagnosis System

Jack Jiang

06/12/2020

Executive Summary

In this project, we will try to build a medical prediction system to diagnose orthopedic patients based on six clinical bio-mechanical attributes derived from the shape and orientation of the pelvis and lumbar spine of a patient. The diagnostic result is either Normal or Abnormal. The data set is downloaded from the Kaggle website (please see the Reference at the end of this report for more details), which will be split into a train set and a test set. We will use two different models to build the system. The first is the logistic regression model and the second is the k nearest neighbours model. The prediction models will be trained on the train set and then will be applied to the test set to calculate the overall accuracy as a measure of their predictive powers. The better performing model will be re-trained on the entire data set (instead of the train set) to build the final prediction model ready for diagnosing usage. However, the final prediction model will not be further evaluated of its accuracy within the scope of this project due to the lack of more independent clinical data from new patients.

Methods and Analysis

First we will load the CSV source data file into a R data frame and do some initial examinations and analysis.

```
#Load the data file, and mutate the class column from string type to factor type.
data <- read.csv("Kaggle-source-data.csv") %>% mutate(class = as.factor(class))

#Examine the first few rows of the data set.
head(data)
```

```
##   pelvic_incidence pelvic_tilt.numeric lumbar_lordosis_angle sacral_slope
## 1         63.02782           22.552586              39.60912     40.47523
## 2         39.05695           10.060991              25.01538     28.99596
## 3         68.83202           22.218482              50.09219     46.61354
## 4         69.29701           24.652878              44.31124     44.64413
## 5         49.71286            9.652075              28.31741     40.06078
## 6         40.25020           13.921907              25.12495     26.32829
##   pelvic_radius degree_spondylolisthesis    class
## 1      98.67292                -0.254400 Abnormal
## 2     114.40543                 4.564259 Abnormal
## 3     105.98514                -3.530317 Abnormal
## 4     101.86850                11.211523 Abnormal
## 5     108.16872                 7.918501 Abnormal
## 6     130.32787                 2.230652 Abnormal
```

```r
#Check how many rows there are in the data set.
number_of_rows <- nrow(data)
number_of_rows
```

```
## [1] 310
```

There are six clinical measurements (the numerical columns) derived from the shape and orientation of the pelvis and lumbar spine with the accompanying diagnostic result (either Abnormal or Normal as contained in the column named class) of each of the 310 patients.

We will check what is the percentage of normal diagnosis in the data.

```r
#Calculate the percentage of normal diagnosis in the data set.
percentage_normal <- mean(data$class == "Normal")
percentage_normal
```

```
## [1] 0.3225806
```

The next step is to split the whole data set into the train and test sets. Since we do not have a large data set, we will randomly pick 15% of the rows as the test set and the remaining 85% as the train set.

```r
# Split the data set into the test (15%) and train (85%) sets.
set.seed(1, sample.kind="Rounding")
test_index <- createDataPartition(y = data$class, times = 1, p = 0.15, list = FALSE)
train_set <- data[-test_index,]
test_set <- data[test_index,]
```

Model 1: Logistic Regression Model

In this model, we will try to predict the Normal or Abnormal result based on the six clinical predictors using the logistic regression model. We will train the model on the train set, use it on the test set to make predictions, and then evaluate its diagnostic accuracy on the test set.

```r
# Train the logistic model on the train set.
glm_fit <- train(class ~ ., method = "glm", data = train_set)

# Make predictions on the test set using the trained model.
glm_predict <- predict(glm_fit, test_set, type = "raw")

# Calculate the prediction accuracy on the test set.
glm_accuracy <- confusionMatrix(glm_predict, test_set$class)$overall[["Accuracy"]]
glm_accuracy
```

```
## [1] 0.8723404
```

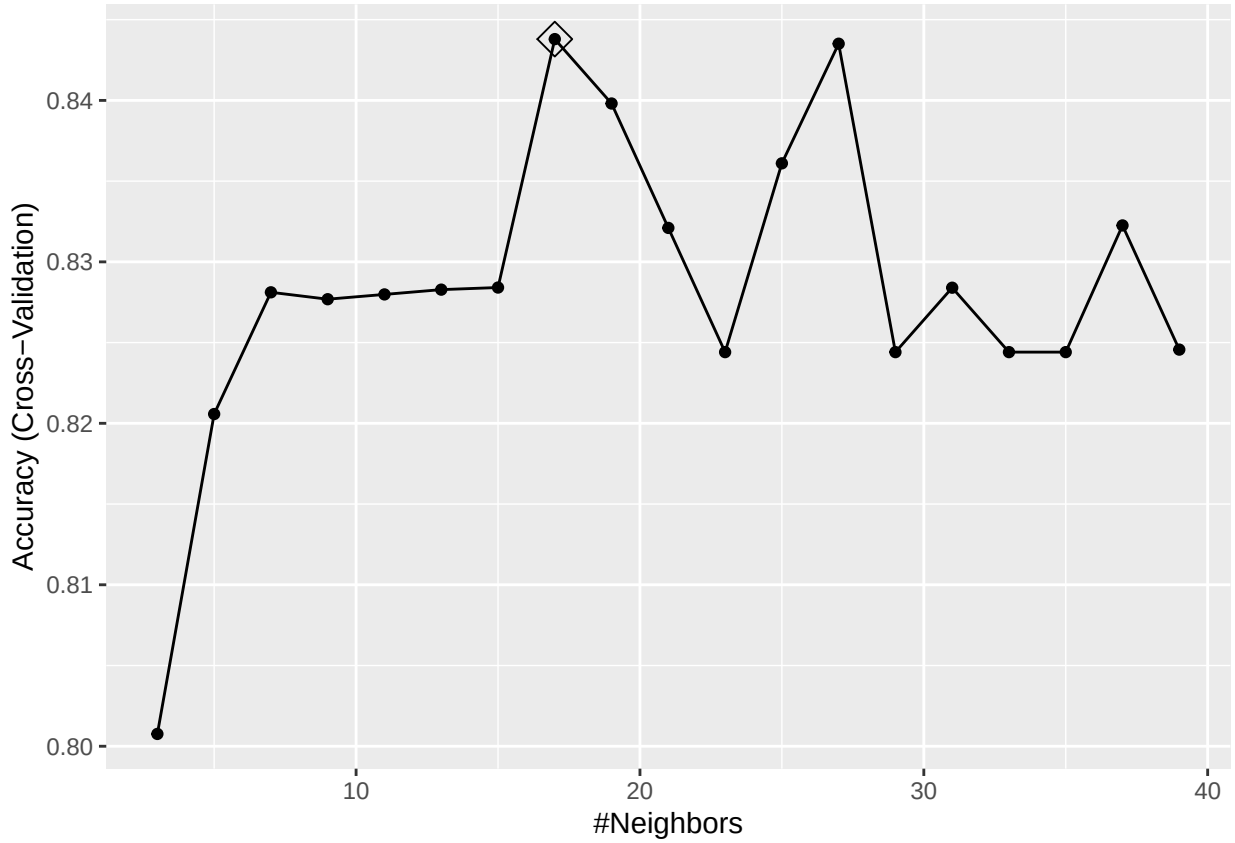
Model 2: K Nearest Neighbours Model

In this model, we will try to predict the Normal or Abnormal result based on the six clinical predictors using the k nearest neighbours model. We will use 10-fold cross validation on the train set to train the model, use it on the test set to make predictions, and then evaluate its diagnostic accuracy on the test set. We will tune a series of odd integers ranging from 3 to 39 to find the optimal value of the parameter k,

```r
# Set the cross validation to be 10-fold.
control <- trainControl(method = "cv", number = 10, p = 0.9)

# Train the k nearest neighbours model on the train set
# with a tuning grid of k ranging from 3 to 39.
knn_fit <- train(class ~ ., method = "knn", tuneGrid = data.frame(k = seq(3, 39, 2)),
                 data = train_set, trControl = control)

#Visualize the tuning results of k.
ggplot(knn_fit, highlight = TRUE)
```




```r
#Show the trained model.
knn_fit$finalModel
```

```
## 17-nearest neighbor model
## Training set outcome distribution:
##
## Abnormal   Normal
##      178       85
```

```r
optimal_k_on_train_set <- knn_fit$finalModel$k
optimal_k_on_train_set
```

```
## [1] 17
```

```r
# Make predictions on the test set using the trained model.
knn_predict <- predict(knn_fit, test_set, type = "raw")

# Calculate the prediction accuracy on the test set.
knn_accuracy <- confusionMatrix(knn_predict, test_set$class)$overall[["Accuracy"]]
knn_accuracy
```

```
## [1] 0.9148936
```

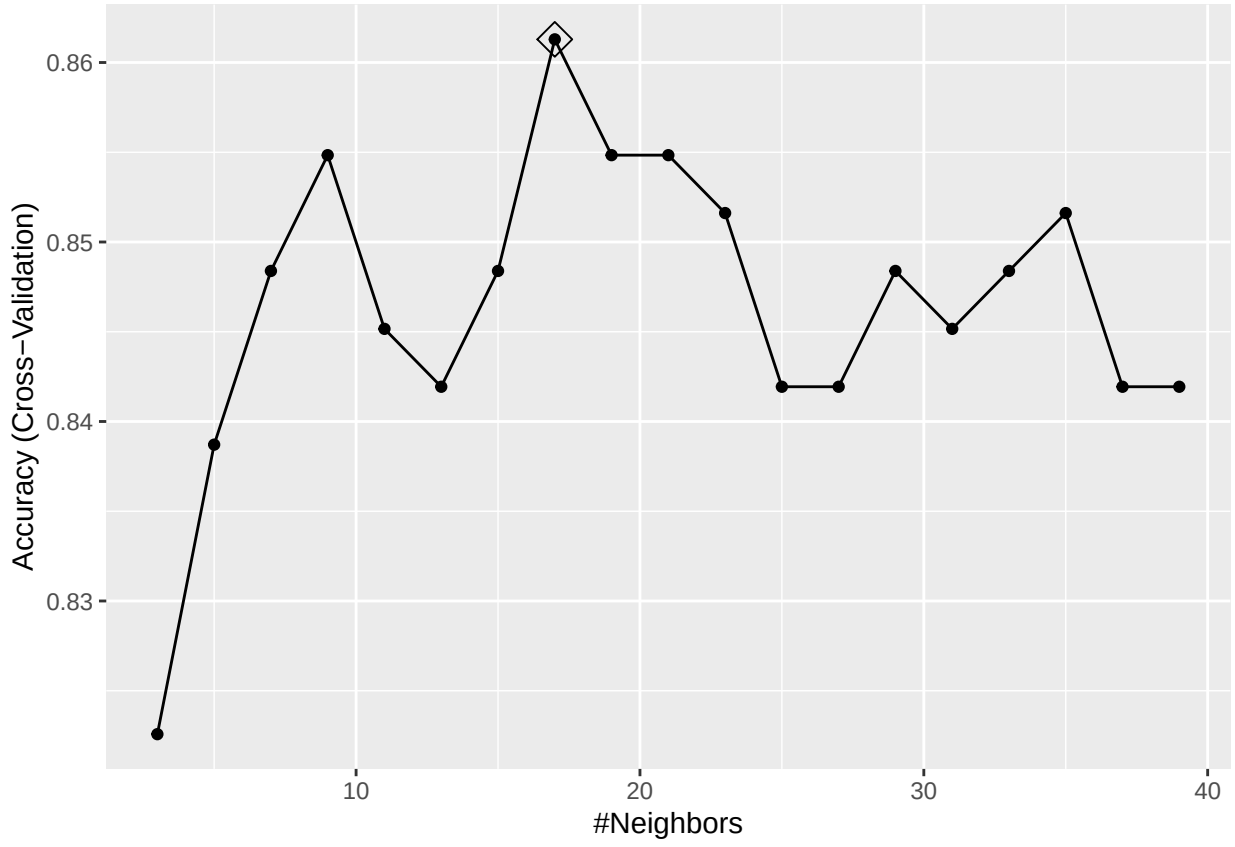
The final prediction model: K Nearest Neighbours Model with 10-fold cross validation

Since the k nearest neighbours model showed a higher diagnostic accuracy on the test set, we will use the entire data set (instead of the train set) to train this model for diagnosing usage. However, the final prediction model will not be further evaluated of its accuracy without obtaining more independent clinical data from new patients.

```r
# Set the cross validation to be 10-fold.
control <- trainControl(method = "cv", number = 10, p = 0.9)

# Train the k nearest neighbours model on the entire data set
# with a tuning grid of k ranging from 3 to 39.
final_knn_fit <- train(class ~ ., method = "knn", tuneGrid = data.frame(k = seq(3, 39, 2)),
                 data = data, trControl = control)

# Visualize the tuning results of k.
ggplot(final_knn_fit, highlight = TRUE)
```



```
#Show the trained final prediction model.
final_knn_fit$finalModel
```

```
## 17-nearest neighbor model
## Training set outcome distribution:
##
## Abnormal   Normal
##      210      100
```

```
optimal_k_on_entire_data_set <- final_knn_fit$finalModel$k
optimal_k_on_entire_data_set
```

```
## [1] 17
```

Results

Based on the above analysis, we have the following results:

(1) The logistic regression model had an overall diagnosis accuracy of 0.8723404 on the test set.

(2) The k nearest neighbours model had an overall diagnosis accuracy of 0.9148936 on the test set.

(3) When the k nearest neighbours model was trained on the train set, the optimal value of k was 17.

(4) When the k nearest neighbours model was trained on the entire data set, the optimal value of k was 17.

(5) The final k nearest neighbours prediction model trained on the entire data set cannot be further evaluated of its diagnosing accuracy due to lack of more independent data.

The optimal k trained on the entire data set might be different from that trained on the train set, possibly due to the fact that our whole data set is relatively small (with only 310 patients). If we have a much larger data set, the anticipation is that the results in (3) and (4) above will converge and the difference (if any) will not be material.

As we can see from the above results, the logistic regression model had lower prediction power than the k nearest neighbours model in diagnosing the patients on the test set, for which reason we chose the latter to develop the final model.

Conclusion

In this project, we have tried to build a system to diagnose orthopedic patients based on six clinical indicators. Starting with the Kaggle data set, we have tried two different models: the logistic regression model and the k nearest neighbours model. Based on the results, it is reasonable to draw the conclusion that the Normal or Abnormal diagnosis can be reliably predicated based on the bio-mechanical measurements of the shape and orientation of the pelvis and lumbar spine of a patient. And the k nearest neighbours model appears to be more powerful with higher accuracy than the logistic regression model in making the diagnostic predictions.

One limitation of this project is that the size of our data set is relatively small, which might have led to lower accuracy and stability in the prediction powers of the models that we have used.

Some possible future works on this project may include obtaining more clinical data from new patients and/or using more advanced techniques to build our prediction models.

Reference

www.kaggle.com/uciml/biomechanical-features-of-orthopedic-patients?select=column_2C_weka.csv

The original data set was downloaded from UCI ML repository:

Lichman, M. (2013). UCI Machine Learning Repository [http://archive.ics.uci.edu/ml]. Irvine, CA: University of California, School of Information and Computer Science

Files were converted to CSV.