

Lower Back Pain Symptoms

Die(Christy) Hu

Summary

The objectives of this study is to first identify the most important/relevant attributes that should be taken into consideration when describing whether a person suffers from lower back pain, and construct a interpretable model that acheives reasonable prediction accuracy. This study aims to answer these two questions showing the process of feature selection and describing three possible methods to accomplish this task: lasso, ridge and elastic net regression. In particular, the focus is on feature selection and model construction using the lasso method. As a result, the six metrics that are found to be most relevant include the degree of spondylolisthesis, the pelvic radius, the sacral slope, the pelvic tilt, the pelvic slope and the scoliosis slope. Among which the degree of spondylolisthesis has the most significant influence. In addition, the lasso logistic regression model is able to make future prediction with a 0.823 accuracy rate and an AUC of 0.881.

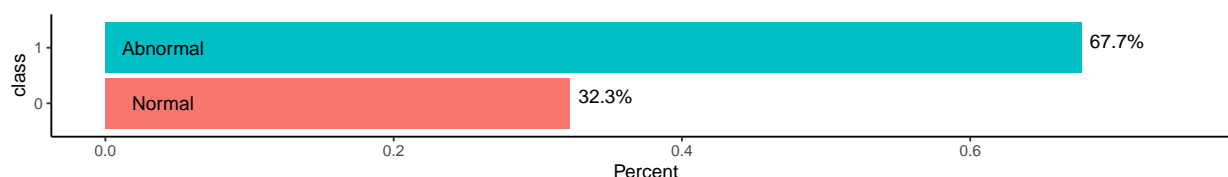
Introduction

Despite the fact that lower back pain is noticeably common, the symptoms and severity of Lower back pain vary greatly, and can be caused by problems with any parts of the complicated, interconnected network in the lumbar spine. Problems with spinal muscles, nerves, bones, discs or tendons can also cause back muscle spasms or severe issues that radiates to other parts of the body, which can result in extreme pain and disability. A simple lower back muscle strain might be agonizing enough to call for an emergency visit, while a degenerating disc might cause only moderate, sporadic discomfort. The goal of this study is to determine the relevant physical spine metrics that are directly associated with a person being abnormal and having lower back pain, as well as to help make future predictions based on those relevant metrics.

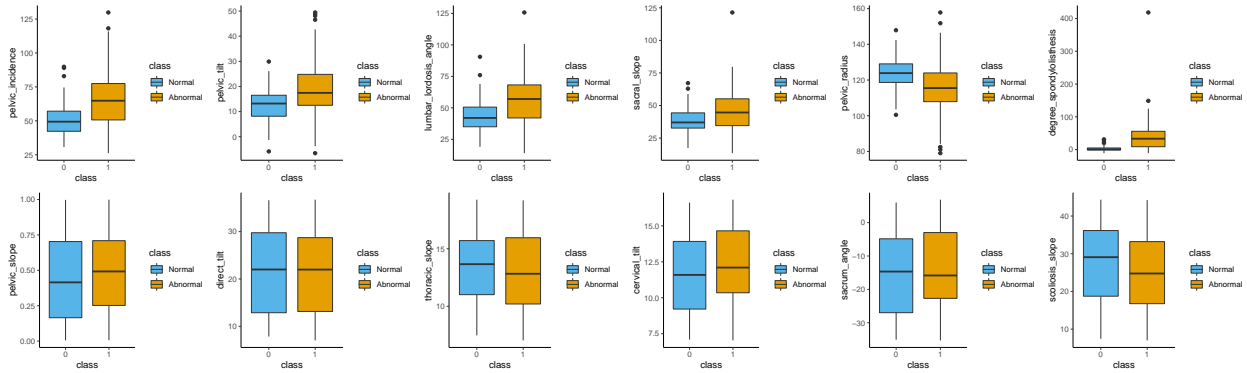
Data

The dataset for this study - “Lower Back Pain Symptoms” is retrieved from the Kaggle platform. It contains 310 observations and 13 attributes, including twelve numeric predictors and one binary outcome variable. Specifically, the predictor variables in the study are: *pelvic_incidence*, *pelvic_tilt*, *lumbar_lordosis_angle*, *sacral_slope*, *pelvic_radius*, *degree_spondylolisthesis*, *pelvic_slope*, *direct_tilt*, *thoracic_slope*, *cervical_tilt*, *sacrum_angle*, *scoliosis_slope*. The *class* attribute identifies a person is abnormal or normal using collected physical spine details and is served as the response variable. There is no obvious observation of missing or erroneous values in the dataset through initial exploration. In order to examine the inferential question of interest and measure the performance and predictive accuracy of the model, the data is randomly split into a *training set* and a *testing set* based on the 80-20 ratio. Due to the limited number of observations in this dataset, a larger portion of the data is assigned for training.

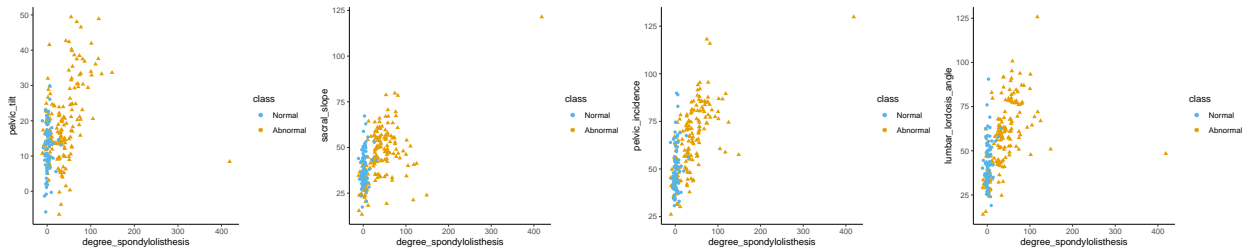
A closer inspection of the response variable *class* reveals the imbalanced data problem that requires further investigation and if necessary, needs to be addressed before model construction. The bar graph below shows that there is 67.7% of data within the *class* attribute being labelled as *Abnormal*, with the rest of 32.3% being labelled as *Normal*.



The observation of the distribution of each predictor by *class* with the assistance of boxplots indicates that there are five predictors can be potentially considered as the most relevant attributes to determine the decision boundary of the response, that is, to separate people who are *Abnormal* from those are *Normal*. The five predictors that exhibits evident difference between two classes are *sacral_slope*, *degree_spondylolisthesis*, *lumbar_lordosis_angle*, *pelvic_tilt*, and *pelvic_incidence*.



Drilling down further into the *class* distribution of each paired predictors out of the five predictors identified by the aforementioned boxplots, the scatterplots show that although the majority data points can be segregated from one class to another, there are still plenty of overlapping areas in each scatterplot for different combination of paired predictors. This indicates that at least more than two most relevant predictors are required to predict *class*. No conclusive comments can be made based on the above exploratory data analysis alone.

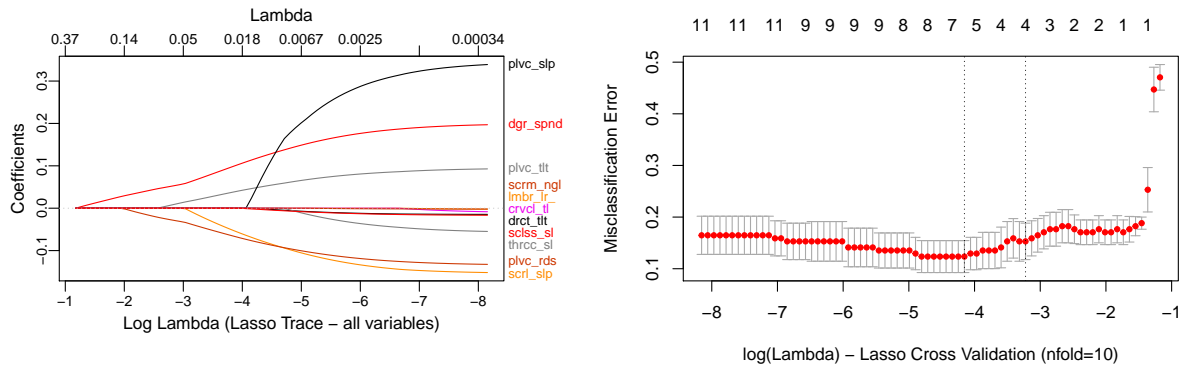


Model

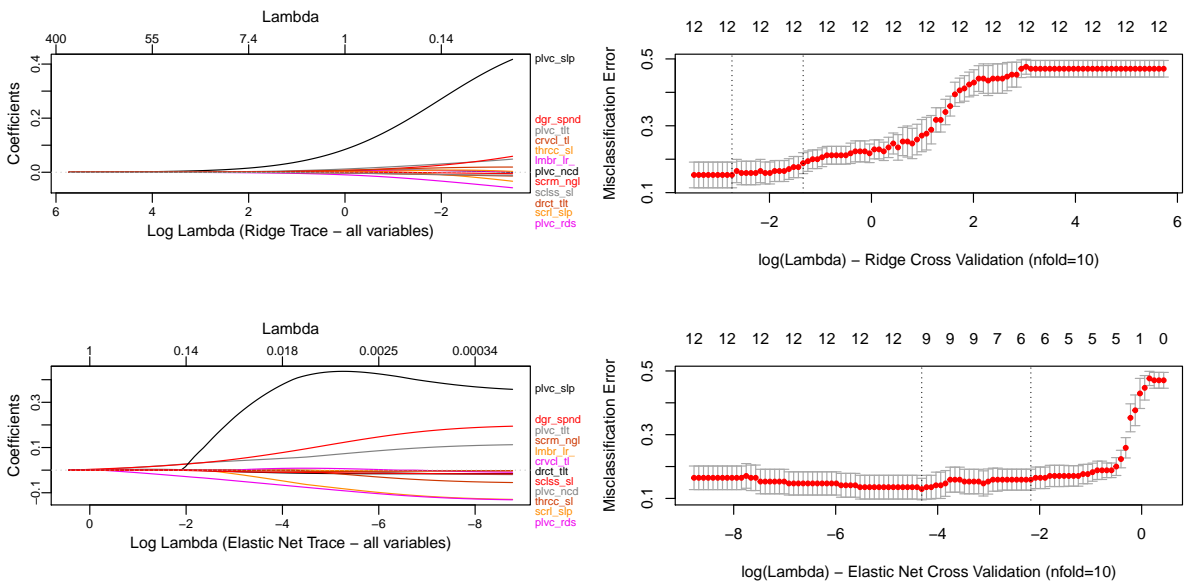
A detailed description of the model used, how you selected the model, how you selected the variables, model assessment, model validation, and presentation of the model results. What are your overall conclusions in context of the inferential problem(s)?

The implementation of feature selection and model construction in this study is achieved by comparing *Lasso*, *Ridge* and *Elastic Net* regression. The Lasso (*Least Absolute Shrinkage and Selection Operator*) method puts a constraint on the sum of the absolute values of the model parameters, commonly referred to as the L1 penalty term. To improve the prediction accuracy and interpretability of regression models, the method applies a shrinking/regularization process where it penalizes the coefficients of the regression variables by shrinking some of them to zero. In comparison, Ridge regression aims to penalize the sum of the squared coefficients, which is commonly referred to as the L2 penalty term and does not set any coefficients to zero. Elastic Net regularization is a combination of the effects of the lasso and Ridge penalties. Through the utilization of these shrinkage methods, computation burden can be greatly reduced since feature selection and model construction only happen once, in contrast to wrapper methods such as stepwise BIC or AIC which performs and compares the results of different models before making a selection.

The lasso trace graph shown below presents how each predictor affects the response variable at different values of lambda, which controls the strength of the penalty. Each line in the graph represents one predictor variable and when it enters the model. Based on the plot, *degree_spondylolisthesis* seems to be the most significant variable because it enters the model first and steadily positively affect the response variable. The second most important variable is *pelvic_radius* that enters later in the model but negatively affect the response variable. Other important variables include *sacral_slope*, *pelvic_tilt*, *pelvic_slope* and *scoliosis_slope*. To select the most appropriate value for lambda, 10-fold cross validation is employed and the relevant plot is also shown below. The lasso method extracts different values for lambda, the first vertical dotted line in the graph gives the minimum cross-validated misclassification error and the second vertical dotted line gives a model such that the error is within one standard error of the minimum. In this study, the lambda is selected to minimize the misclassification error to increase prediction accuracy.



Similar procedures are applied to ridge and elastic net regression. The below graphs help illustrate the aforementioned properties of the two methods being different from lasso. Specifically, ridge does not shrink any coefficients to zero therefore all variables are included in the model. Elastic Net tends to select number of variables that lie somewhere in between those two methods based on an alpha parameter indicating which method the model is leaning towards. This study favors lasso since feature selection is the primary objective.



The final logistic regression model is presented below, where X is the model/design matrix constructed based on the covariates and β is the vector of all the corresponding coefficients for the variables. The L1 and L2 penalty terms are also indicated below for lasso and ridge regression respectively.

$$\log\left(\frac{\pi_i}{1-\pi_i}\right) = \beta X \quad \text{Lasso L1 penalty: } \lambda \sum |\beta| \quad \text{Ridge L2 penalty: } \lambda \sum \beta^2$$

The two tables below display the variables selected by each model and listed their coefficients in a descending order. The six variables being selected by lasso are *degree_spondylolisthesis*, *pelvic_radius*, *sacral_slope*, *pelvic_tilt*, *pelvic_slope* and *scoliosis_slope*, with *degree_spondylolisthesis* being the most important variable. Lasso excludes the other six variables in the model and forces their coefficients to zeros, which are displayed at the bottom of the table. The ridge model retains all variables and arrives at a similar result to lasso, with *pelvic_slope*, *pelvic_radius*, *degree_spondylolisthesis*, *pelvic_tilt*, *sacral_slope* being the top five most significant variables. Although not displayed here due to space limitation, the elastic net model also outputs identical results.

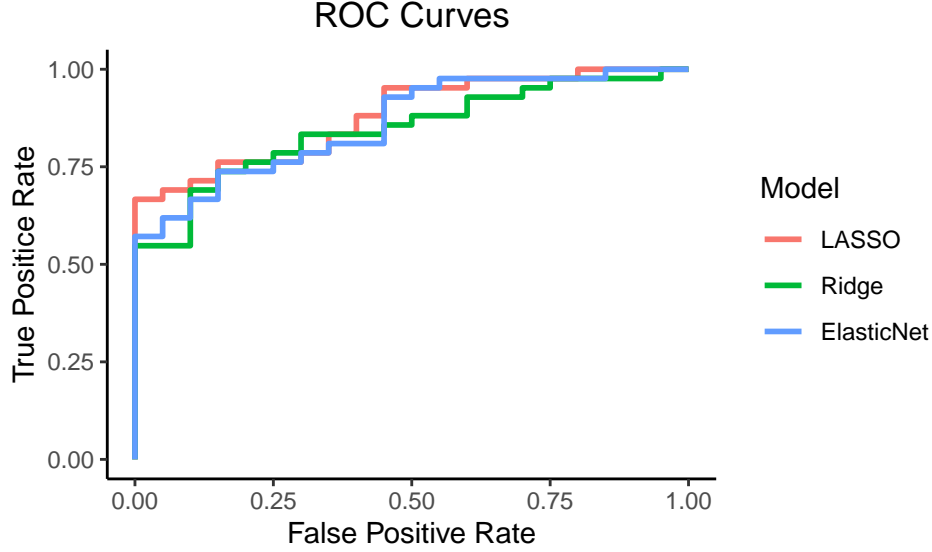
Table 1: Lasso Model - Coefficients

	Overall
degree_spondylolisthesis	0.114
pelvic_radius	0.077
sacral_slope	0.069
pelvic_tilt	0.046
pelvic_slope	0.024
scoliosis_slope	0.002
pelvic_incidence	0.000
lumbar_lordosis_angle	0.000
direct_tilt	0.000
thoracic_slope	0.000
cervical_tilt	0.000
sacrum_angle	0.000

Table 2: Ridge Model - Coefficients

	Overall
pelvic_slope	0.351
pelvic_radius	0.045
degree_spondylolisthesis	0.040
pelvic_tilt	0.039
sacral_slope	0.019
cervical_tilt	0.018
direct_tilt	0.012
scoliosis_slope	0.010
thoracic_slope	0.007
lumbar_lordosis_angle	0.005
pelvic_incidence	0.004
sacrum_angle	0.004

The ROC curves shown below aims to measure the model performance on the testing set. Out of the three models, lasso seems to outperform both ridge and elastic net by achieving an AUC of 0.881 with 0.823 accuracy, followed by elastic net(AUC: 0.862 Accuracy:0.806) and finally ridge regression(AUC: 0.845 Accuracy:0.79). This indicates that the lasso model can reasonably predict whether a person is normal or abnormal based on the selected variables.



Model	AUC	Accuracy
Lasso	0.881	0.823
Ridge	0.845	0.79
ElasticNet	0.862	0.806

Conclusions

Based on the above analysis, there are six variables that are found to be relevant in terms of explaining the response variable *class*, that is, whether a person is normal or abnormal (could potentially suffer from lower back pain). The six useful metrics include the degree of spondylolisthesis, the pelvic radius, the sacral slope, the pelvic tilt, the pelvic slope and the scoliosis slope. Among which the degree of spondylolisthesis has the most significant influence. It refers to a spinal condition that affects the lower vertebrae (spinal bones) and causes one of the lower vertebrae to slip forward onto the bone directly beneath it. In addition, the lasso logistic regression model is able to predict future classes with 0.823 accuracy rate and AUC of 0.881. One limitation of this analysis is that we do not have a large amount of physical spine-related data. In order to increase the predictive power of the model, more training data is preferred. Apart from collecting additional physical data, activity-related data that could potentially contribute to each physical condition that is explored in this analysis can also be helpful to incorporate. Understanding what types of daily habits may be associated with spondylolisthesis will provide beneficial ground for adopting or retiring a certain habit for instance. In addition, the research topic of this analysis requires a substantial amount of domain knowledge, which can be increased by extending and deepening the research and gathering more field information. Finally, there are other classification methods or techniques other than lasso, ridge and elastic net that are worth exploring and implementing.

Appendix

```
## ----include = FALSE-----
knitr::opts_chunk$set(echo=FALSE,
                      warning=FALSE,
                      message=FALSE,
                      fig.align="center",
                      fig.pos='H',
                      results="asis")

## ----results="hide"-----
list_of_packages = c("tidyverse",
                    "dplyr",
                    "caret",
                    "broom",
                    "glmnet",
                    "psych",
                    "regclass",
                    "knitr",
                    "xtable",
                    "kableExtra",
                    "gridExtra",
                    "pROC",
                    "corrplot",
                    "ROSE",
                    "plotmo",
                    "ROCR")

packages = list_of_packages[!(list_of_packages %in% installed.packages()[,"Package"])]
if(length(packages)){install.packages(packages)}
pkg_lib = lapply(list_of_packages, require, character.only = TRUE)
digit=3
options(digits=digit, xtable.comment=FALSE)

## ----echo=FALSE-----
# Load clean data & set seed for reproducibility
set.seed(123)
spine = read_csv("../DataSets/Dataset_spine.csv")
spine = spine %>%
  dplyr::rename('pelvic_incidence' = Col1,
               'pelvic_tilt' = Col2,
               'lumbar_lordosis_angle' = Col3,
               'sacral_slope' = Col4,
               'pelvic_radius' = Col5,
               'degree_spondylolisthesis' = Col6,
               'pelvic_slope' = Col7,
               'direct_tilt' = Col8,
               'thoracic_slope' = Col9,
               'cervical_tilt' = Col10,
               'sacrum_angle' = Col11,
               'scoliosis_slope' = Col12,
               'class' = Class_att) %>%
```

```

mutate(class = ifelse(class=='Abnormal',1,0)) %>%
  mutate(class = as.factor(class))

## ----echo=FALSE-----
# Training & Testing Split
set.seed(123)
intrain<-createDataPartition(y=spine$class, p=0.8, list=FALSE)
training<-spine[intrain,]
testing<-spine[-intrain,]
write.csv(training, "training.csv")

## ----echo=FALSE-----
# Handling Imbalanced Data
training0 = training[training$class==0,]
training1 = training[training$class==1,]
training1_down = training1[sample(length(training1$class), size = 90, replace = F),]
training_fix.balance = rbind(training0,training1_down)

## ----echo=FALSE,fig.width=10,fig.height=1.5,message=FALSE, warning=FALSE----
# EDA - imbalanced class within the response variable
#library(scales)
ggplot(training, aes(x= class, group=1)) +
  geom_bar(aes(y = ..prop.., fill = factor(..x..)), stat="count", show.legend = FALSE) +
  geom_text(aes( label = scales::percent(..prop..),
                y= ..prop.. ), stat= "count", vjust = -0, hjust=-.15) +
  labs(y = "Percent", fill="class") +
  scale_y_continuous(labels = scales::percent) +
  ylim(0,0.75) +
  coord_flip() +
  theme_classic() +
  annotate("text", x=2, y=0.04, label= "Abnormal") +
  annotate("text", x =1, y=0.04, label = "Normal")

## ----echo=FALSE,fig.width=20,fig.height=6-----
# EDA - boxplots of all predictor variables
b1 = ggplot(training, aes(x=class, y=pelvic_incidence, fill=class)) +
  scale_fill_manual(name = "class", breaks = c("0","1"), labels = c("Normal", "Abnormal"), values=c("#5
  geom_boxplot() +
  theme_classic()
b2 = ggplot(training, aes(x=class, y=pelvic_tilt, fill=class)) +
  scale_fill_manual(name = "class", breaks = c("0","1"), labels = c("Normal", "Abnormal"), values=c("#5
  geom_boxplot() +
  theme_classic()
b3 = ggplot(training, aes(x=class, y=lumbar_lordosis_angle, fill=class)) +
  scale_fill_manual(name = "class", breaks = c("0","1"), labels = c("Normal", "Abnormal"), values=c("#5
  geom_boxplot() +
  theme_classic()
b4 = ggplot(training, aes(x=class, y=sacral_slope, fill=class)) +
  scale_fill_manual(name = "class", breaks = c("0","1"), labels = c("Normal", "Abnormal"), values=c("#5

```

```

    geom_boxplot() +
    theme_classic()
b5 = ggplot(training, aes(x=class, y=pelvic_radius, fill=class)) +
    scale_fill_manual(name = "class", breaks = c("0","1"), labels = c("Normal", "Abnormal"), values=c("#56b4e9", "#f08080")) +
    geom_boxplot() +
    theme_classic()
b6 = ggplot(training, aes(x=class, y=degree_spondylolisthesis, fill=class)) +
    scale_fill_manual(name = "class", breaks = c("0","1"), labels = c("Normal", "Abnormal"), values=c("#56b4e9", "#f08080")) +
    geom_boxplot() +
    theme_classic()
b7 = ggplot(training, aes(x=class, y=pelvic_slope, fill=class)) +
    scale_fill_manual(name = "class", breaks = c("0","1"), labels = c("Normal", "Abnormal"), values=c("#56b4e9", "#f08080")) +
    geom_boxplot() +
    theme_classic()
b8 = ggplot(training, aes(x=class, y=direct_tilt, fill=class)) +
    scale_fill_manual(name = "class", breaks = c("0","1"), labels = c("Normal", "Abnormal"), values=c("#56b4e9", "#f08080")) +
    geom_boxplot() +
    theme_classic()
b9 = ggplot(training, aes(x=class, y=thoracic_slope, fill=class)) +
    scale_fill_manual(name = "class", breaks = c("0","1"), labels = c("Normal", "Abnormal"), values=c("#56b4e9", "#f08080")) +
    geom_boxplot() +
    theme_classic()
b10 = ggplot(training, aes(x=class, y=cervical_tilt, fill=class)) +
    scale_fill_manual(name = "class", breaks = c("0","1"), labels = c("Normal", "Abnormal"), values=c("#56b4e9", "#f08080")) +
    geom_boxplot() +
    theme_classic()
b11 = ggplot(training, aes(x=class, y=sacrum_angle, fill=class)) +
    scale_fill_manual(name = "class", breaks = c("0","1"), labels = c("Normal", "Abnormal"), values=c("#56b4e9", "#f08080")) +
    geom_boxplot() +
    theme_classic()
b12 = ggplot(training, aes(x=class, y=scoliosis_slope, fill=class)) +
    scale_fill_manual(name = "class", breaks = c("0","1"), labels = c("Normal", "Abnormal"), values=c("#56b4e9", "#f08080")) +
    geom_boxplot() +
    theme_classic()
grid.arrange(b1, b2, b3, b4, b5, b6, b7, b8, b9, b10, b11, b12, ncol=6)

## -----echo=FALSE,fig.width=20,fig.height=4-----
# EDA - selected scatterplots
s1 = ggplot(training, aes(x=degree_spondylolisthesis, y=pelvic_tilt, shape=class, color=class)) +
    scale_color_manual(name = "class", breaks = c("0","1"), labels = c("Normal", "Abnormal"), values=c("#56b4e9", "#f08080")) +
    geom_point() +
    guides(shape = FALSE) +
    theme_classic()
s2 = ggplot(training, aes(x=degree_spondylolisthesis, y=sacral_slope, shape=class, color=class)) +
    scale_color_manual(name = "class", breaks = c("0","1"), labels = c("Normal", "Abnormal"), values=c("#56b4e9", "#f08080")) +
    geom_point() +
    guides(shape = FALSE) +
    theme_classic()
s3 = ggplot(training, aes(x=degree_spondylolisthesis, y=pelvic_incidence, shape=class, color=class)) +
    scale_color_manual(name = "class", breaks = c("0","1"), labels = c("Normal", "Abnormal"), values=c("#56b4e9", "#f08080")) +
    geom_point() +
    guides(shape = FALSE) +

```



```

theme_classic()
s4 = ggplot(training, aes(x=degree_spondylolisthesis, y=lumbar_lordosis_angle, shape=class, color=class,
scale_color_manual(name = "class", breaks = c("0","1"), labels = c("Normal", "Abnormal"), values=c("#
geom_point() +
guides(shape = FALSE) +
theme_classic()
grid.arrange(s1, s2, s3, s4, ncol=4)

## ----fig.width=12,fig.height=4-----
# Feature Selection using Lasso
par(mfrow=c(1,2))
set.seed(123)
lassolam = glmnet(as.matrix(training_fix.balance[, -which(names(training_fix.balance)=="class")] ),
training_fix.balance$class, family = "binomial", standardize=T, alpha=1)
lassolamcv = cv.glmnet(as.matrix(training_fix.balance[, -which(names(training_fix.balance)=="class")] ),
training_fix.balance$class, family = "binomial", type.measure = 'class', standardize=T)
plot_glmnet(lassolam, label=12, xlab="Log Lambda (Lasso Trace - all variables)")
plot(lassolamcv, xlab="log(Lambda) - Lasso Cross Validation (nfold=10)")

## ----fig.width=11,fig.height=5.5-----
par(mfrow=c(2,2))
set.seed(123)
# Feature Selection using Ridge
ridgelam = glmnet(as.matrix(training_fix.balance[, -which(names(training_fix.balance)=="class")] ),
training_fix.balance$class, family = "binomial", standardize=T, alpha=0)
ridgelamcv = cv.glmnet(as.matrix(training_fix.balance[, -which(names(training_fix.balance)=="class")] ),
training_fix.balance$class, family = "binomial", type.measure = 'class', standardize=T)
plot_glmnet(ridgelam, label=12, xlab="Log Lambda (Ridge Trace - all variables)")
plot(ridgelamcv, xlab="log(Lambda) - Ridge Cross Validation (nfold=10)")

# Feature Selection using ElasticNet
set.seed(123)
enlam = glmnet(as.matrix(training_fix.balance[, -which(names(training_fix.balance)=="class")] ),
training_fix.balance$class, family = "binomial", standardize=T, alpha=0.2)
enlamcv = cv.glmnet(as.matrix(training_fix.balance[, -which(names(training_fix.balance)=="class")] ),
training_fix.balance$class, family = "binomial", type.measure = 'class', standardize=T)
plot_glmnet(enlam, label=12, xlab="Log Lambda (Elastic Net Trace - all variables)")
plot(enlamcv, xlab="log(Lambda) - Elastic Net Cross Validation (nfold=10)")

## ----echo=FALSE, message=FALSE, warning=FALSE, results='hide'-----
# Lasso Model
set.seed(123)
best_lambda = lassolamcv$lambda.min
x = as.matrix(training_fix.balance[, -which(names(training_fix.balance)=="class")])
y = training_fix.balance$class
lasso_model = glmnet(x, y, alpha = 1, lambda = best_lambda, family = "binomial", standardize = T)
lasso_coef = coef(lasso_model)
varimp = varImp(lasso_model, lambda = best_lambda)
varimp$varnames = rownames(varimp)
varimp = varimp[order(-varimp$Overall),] %>% select(Overall)

```

```

varimp %>% xtable(caption = "Lasso Model - Coefficients", digits=digit) %>% print(caption.placement = "t")

# Ridge Model
set.seed(123)
rbest_lambda = ridgelamcv$lambda.min
rx = as.matrix(training_fix.balance[, -which(names(training_fix.balance)=="class")])
ry = training_fix.balance$class
ridge_model = glmnet(rx, ry, alpha = 0, lambda = rbest_lambda, family = "binomial", standardize = T)
ridge_coef = coef(ridge_model)

varimpr = varImp(ridge_model, lambda = rbest_lambda)
varimpr$varnames = rownames(varimpr)

varimpr = varimpr[order(-varimpr$Overall),] %>% select(Overall)
varimpr %>% xtable(caption = "Ridge Model - Coefficients", digits=digit) %>% print(caption.placement = "t")

## -----
t1 <- kable(varimp, format = "latex", booktabs = TRUE)
t2 <- kable(varimpr, format = "latex", booktabs = TRUE)

cat(c("\\begin{table}[!htb]
\\begin{minipage}{.5\\linewidth}
\\caption{Lasso Model - Coefficients}
\\centering",
t1,
"\\end{minipage}%
\\begin{minipage}{.5\\linewidth}
\\centering
\\caption{Ridge Model - Coefficients}",
t2,
"\\end{minipage}
\\end{table}"
))

## ----echo=FALSE, message=FALSE, warning=FALSE, results='hide',fig.show='hide'----
# Lasso Prediction
testingx = as.matrix(testing[, -which(names(testing)=="class")])
prob <- predict(lasso_model, newx= testingx, s=best_lambda, type = 'response')
pred <- prediction(prob, testing$class)
perf <- performance(pred, "tpr", "fpr")
performance(pred, "auc") # shows calculated AUC for model
max(performance(pred, "acc")@y.values %>% unlist())
plot(perf, colorize=FALSE, col="black") # plot ROC curve
lines(c(0,1),c(0,1), col = "gray", lty = 4 )

## ----echo=FALSE, message=FALSE, warning=FALSE, results='hide',fig.show='hide'----
# Ridge Prediction
rprob <- predict(ridge_model, newx= testingx, s=rbest_lambda, type = 'response')
rpred <- prediction(rprob, testing$class)
rperf <- performance(rpred, "tpr", "fpr")

```

```

performance(rpred,"auc") # shows calculated AUC for model
max(performance(rpred, "acc")@y.values %>% unlist())
plot(rperf,colorize=FALSE, col="black") # plot ROC curve
lines(c(0,1),c(0,1),col = "gray", lty = 4 )

## ----echo=FALSE, message=FALSE, warning=FALSE, results='hide',fig.show='hide'----
#ElasticNet Model
set.seed(123)
best_lambda = enlamcv$lambda.min
enx = as.matrix(training_fix.balance[, -which(names(training_fix.balance)=="class")])
eny = training_fix.balance$class
en_model = glmnet(enx, eny, alpha = 0.2, lambda = best_lambda, family = "binomial", standardize = T)
en_coef = coef(en_model)

## ----echo=FALSE, message=FALSE, warning=FALSE, results='hide',fig.show='hide'----
#ElasticNet Prediction
enprob <- predict(en_model, newx= testingx, s=rbest_lambda, type = 'response')
enpred <- prediction(enprob, testing$class)
enperf <- performance(enpred,"tpr","fpr")
performance(enpred,"auc") # shows calculated AUC for model
max(performance(enpred, "acc")@y.values %>% unlist())
plot(enperf,colorize=FALSE, col="black") # plot ROC curve
lines(c(0,1),c(0,1),col = "gray", lty = 4 )

## ----echo=FALSE, message=FALSE, warning=FALSE, results='hide',fig.show='hide'----
# Logistic Regression Model Using BIC Stepwise Without Interaction Terms
set.seed(123)
nullmodel <- glm(class~1, family = 'binomial', data = training_fix.balance)
fullmodel <- glm(class~., family = 'binomial', data = training_fix.balance)
n = nrow(training_fix.balance)
model1 <- step(nullmodel, scope = formula(fullmodel), direction="both",trace=0)
summary(model1)
roc(training_fix.balance$class,fitted(model1),plot=T,print.thres="best",legacy.axes=T,
     print.auc =T,col="red3")

prob = predict(model1, type="response", testing)
pred = ifelse(prob>0.5,1,0)
roc(testing$class, prob,plot=T,print.thres="best",legacy.axes=T,
     print.auc =T,col="red3")
confusionMatrix(pred %>% as.factor(), testing$class)

## ----fig.width=5,fig.height=3-----
# Models Combined Prediction Results
perf.data = data.frame(fpr = unlist(perf@x.values), tpr=unlist(perf@y.values), Model="LASSO")
rperf.data = data.frame(fpr = unlist(rperf@x.values), tpr=unlist(rperf@y.values), Model="Ridge")
enperf.data = data.frame(fpr = unlist(enperf@x.values), tpr=unlist(enperf@y.values), Model="ElasticNet")
rocs.data = rbind(perf.data, rperf.data, enperf.data)
ggplot(rocs.data, aes(x=fpr, y=tpr, colour=Model)) +
  geom_line(lwd=1) +

```

```
theme_classic() +  
labs(y="True Positice Rate",  
      x="False Positive Rate") +  
ggtitle("ROC Curves") +  
theme(plot.title = element_text(hjust = 0.5))  
  
## ----code = readLines(knitr::purl("Lower Back Pain.rmd", documentation = 1)), echo = T, eval = F----  
## NA
```