The Signal in the Noise with Machine Learning Algorithms and Spine Classification

Michael Streyle

December 1st, 2018

Table of Contents

Machine learning is commonly thought of as a tool primarily used in data science, however, machine learning is also very much a part of applied statistics. Most, if not all, machine learning algorithms were developed with contributions from statisticians. Inspired by the book "The Signal and the Noise: Why So Many Predictions Fail-But Some Don't", by Nate Silver, the main goal of this project is to use a relatively small data set and compare how various supervised machine learning algorithms find the signal in the data while incrementally adding random noise. Handling excess random noise is becoming increasingly important in the world of Big Data, especially considering the number of variables in machine learning applications such as genomics, recommendation algorithms, natural language processing, etc. The data set that I have chosen for the analysis consists of twelve spinal measurements and a classification variable consisting of two levels, normal and abnormal. The algorithms I am going to use to classify the data include logistic regression, support vector machines, random forests, and an algorithm called "Generalized, Unbiased, Interaction Detection and Estimation" (GUIDE), developed by Wei-Yin Loh at the University of Wisconsin, Madison. First, I compare the algorithms on the regular data before adding 10, 100, 500, and 1000 variables of random noise and comparing the performance of the various models.

The spinal measurements data I am going to use in this analysis is publicly available on Kaggle. The objective is to classify each observation as normal or abnormal using machine learning modeling. As seen in the pairs plot from Appendix I, none of the variables have strong correlations with each other, and the last 6 variables are essentially random noise. These random variables will be left out of some of the initial models so that I can compare the models with only meaningful variables with the models that include random variables in order to assess how the models handle excess noise in the data. The six non-random variables (pelvic_incidence,

pelvic_tilt, lumbar_lordosis_angle, sacral_slope, pelvic_radius, and degree_spondylolisthesis) all appear relatively normal and symmetric except degree_spondylolisthesis, which appears to be right skewed.

Logistic Regression

Logistic regression is one of the most popular and most powerful classification algorithms. Logistic regression models the probability of a yes or no outcome on the logit scale so that all the predictions are between 0 and 1 on the probability scale. The logit scale is the natural log of the odds that the response is one of the categories. Odds is related to probability through the equation odds = p / (1-p), where p is the probability of an observation belonging to a particular classification. Then, a cutoff, such as 0.5, is chosen and used to classify each observation as one of the levels of the response variable. In R, I use the glm function to create and train a logistic regression model with spine classification (Abnormal, Normal) as the response variable and the provided real spinal measurements as the predictor variables. By splitting the data into a training and test set, I train the model on a randomly selected 80% subset of the data and keep 20% of the data to test the performance of the model. Throughout the analysis I use several metrics to compare the performance of the various models. True positives represent observations classified as abnormal, which are truly abnormal. False positives represent observations classified as abnormal, but which are actually normal. True negatives represent observations which are classified as normal and are truly normal. False negatives are observations which are classified as normal, but which are actually abnormal. Additionally, the accuracy of a model is defined by Accuracy = (TP + TN) / (TP +FP + TN +FN), which can be interpreted as the percent of predictions which are correct.

After training the initial logistic regression model on the six real variables, the summary output shows that the most important variables in the model are pelvic_radius and degree_spondylolisthesis with respective p-values of 0.0028 and 3.52e-11, which are both well below any reasonable alpha value. The result of this model is an accuracy score of 0.8387 with 36 true positives, 6 false positives, 16 true negatives, and 4 false negatives, with positives representing abnormal classification.

Next, I added the two-way interaction between the two most significant variables: pelvic_radius and degree_spondylolisthesis. The resulting model is very similar to the initial first-order model. The summary of this interaction model indicated the interaction is not significant with a p-value of 0.131. This p-value is above most reasonable alpha cutoff values, although it is close to significant, so it could be included to determine if the interaction improves the performance of the model. I then computed the confusion matrix and the accuracy score for the model with the two-way interaction in order to compare with the first-order model. The interaction model has an accuracy score of 0.8225, with 36 true positives, 7 false positives, 15 true negatives, and 4 false negatives. The results show that the interaction made the model worse; the model with the interaction got one more prediction wrong than the original, first order model. Specifically, the interaction model produced one more false negative than the first order model, which decreased the accuracy score from 0.8387 to 0.8225.

A common variable selection technique for logistic regression is a stepwise procedure which models the data with a few variables at a time in order to filter out any unneeded variables. The result of the step() function in R is a model with only variables that are statistically significant. In this analysis, I only used the forward stepwise procedure which starts with a model with only an intercept and incrementally adds variables. Using this method, I am

still be able to use variable selection when I add random variables. The results of the forward stepwise procedure are an accuracy score of 0.8548, with 37 true positives, 6 false positives, 16 true negatives, and 3 false negatives. The variables included in this forward stepwise model are degree_spondylolisthesis (p-value = 2.98e-11), sacral_slope (p-value = 1.15e-05, pelvic_radius (p-value = 0.00012), pelvic_tilt (p-value = 0.01426), and Direct_tilt (p-value = 0.09473). All of these variables have very low p-values, except Direct_tilt which is only marginally significant.

All of the initial models use a cutoff value of 0.5, which means that fitted probabilities above 0.5 are categorized as normal and fitted probabilities below 0.5 are categorized as abnormal. Instead of using 0.5 as the cutoff value, testing various cutoff values can improve the model's performance. The code used to find the optimal cutoffs uses the fitted probabilities of the training set as possible cutoffs, and computes the accuracy score for the training set using each fitted probability as a cutoff. The cutoff with the highest accuracy score on the training set is then used as the optimal cutoff. Using this method, the optimal cutoff for the first order logistic regression model is 0.7252. With this cutoff and using the same forward stepwise method as above, the train/test split model’s accuracy score decreases from 0.8548 to 0.7419. This decrease in accuracy score highlights the main weakness of the train/test split methodology. The model is only tested on 20 percent of the data, which can lead to the model overfitting on the training set and performing poorly on the test set. In other words, the accuracy score optimizing method over-optimizes the model for the training set which then weakens the accuracy score of the test set predictions. An improved method to the train/test method is a procedure called cross-validation.

Cross-validation is a widely used method which enables the model to make predictions for the whole dataset. To do this, the data is split into n fold, or subgroups. Then, the model is

trained n times, each time leaving out one of the subgroups and using it as a test set. The result is predictions for every row in the data, yet still using separate data to train each model than what was used to test the model. Here, the confusion matrix and accuracy score is indicative of the model tested on every data value, because the values of the confusion matrix add up to 310 (the number of observations in the data) rather than 62 (20 percent of the observations in the data).

In this analysis, I implement cross-validation manually by making 5 subcategories of the data and training the model 5 times, each time keeping one fifth of the data out of training to be used as a test set. For the rest of the logistic regression models, I continue to use cross-validation, the forward stepwise variable selection method, and the tuning method to optimize the performance. All of these steps are performed while iterating through the code 5 times, once for each fold of the cross-validation. The logistic regression model trained only on the 6 real variables with cross-validation, stepwise procedure, and optimization has an accuracy score of 0.8581 with 189 true positives, 23 false positives, 77 true negatives, and 21 false negatives. Not only does this model perform slightly better than the previous models, but it is much more conclusive and less prone to overfitting because it utilizes all the data to make test predictions.

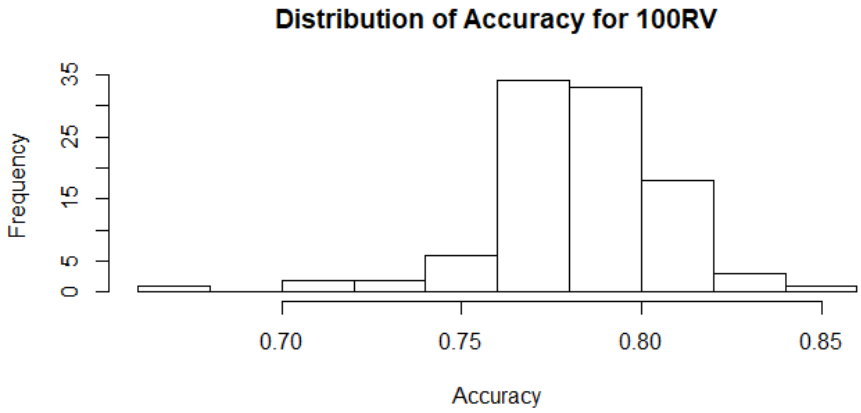
One of the main goals of this project is to assess how the chosen models perform as the amount of noise in the data increases. The first step towards this goal is to add in the random variables that were provided with the data. By adding the additional six random variables, and comparing to the previous logistic regression models, I can determine how logistic regression handles a small amount of increased random noise before continuing to additional random noise. The result of the logistic regression model using cross-validation, forward stepwise variable selection, and optimizing the results is an accuracy score of 0.8548 with 193 true positives, 28 false positives, 72 true negatives, and 17 false negatives. The significant variables in this model

are degree_spondylolisthesis, pelvic_radius, sacral_slope, and pelvic_tilt, all with p-values below 0.02. Considering the number of variables doubled without any additional meaningful data, this model maintains a relatively high performance.

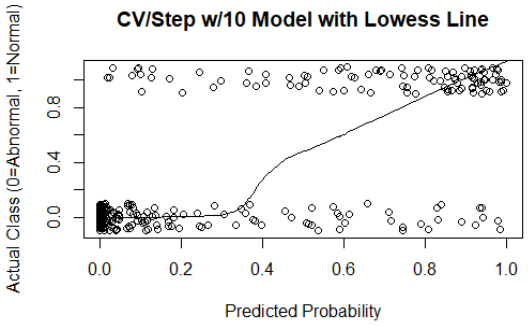
Next, I add an additional 10 random variables to the logistic regression model. These random variables were drawn from a standard normal distribution and are named X1, X2, … X10, respectively. After training a logistic regression model with cross-validation, forward stepwise variable selection, and performance optimizing, the resulting model still has degree_spondylolisthesis, sacral_slope, pelvic_radius, and pelvic_tilt as the most significant variables. Because cross-validation, stepwise variable selection, and performance optimizing all take place in the same loop, the only model summary I can easily access is the fifth iteration, however, this model can serve as an approximation of the overall model. In addition to the real variables, the model also has identified X1 as marginally significant. This indicates that logistic regression starts to break down with the addition of 16 random variables (6 provided and 10 created) to the 6 actual spine measurements. Despite the two marginally significant random variables, the model still performed reasonably well. The accuracy score is 0.8419 with 187 true positives, 26 false positives, 74 true negatives, and 23 false negatives.

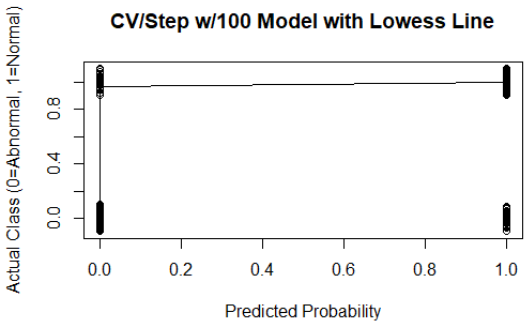
Next, I add 100 random variables to the 12 provided spinal measurement, again drawn from the standard normal distribution. The same process as above is used to train the model, which resulted in an accuracy of 0.7774, with 175 true positives, 34 false positives, 66 true negatives, and 35 false negatives. The additional 90 random variables has decreased the accuracy score from 0.8419 to 0.7774, which is the result of both fewer correct abnormal classification and fewer correct normal classifications. Below is a histogram of the accuracy score after training the model with the 100 random variables 100 times. The histogram gives insight into the

simulation variability of the model, which indicates how the model performs when repeated, and confirms that the model would continue to perform with an accuracy score between 0.75 and 0.82.



For each logistic regression model, I plotted the fitted probabilities against the actual classification with a lowess line fit to the data. The resulting plot reveals the relation between the predicted probability of the model and the actual class. Comparing two of these plots shows how logistic regression breaks down as the variables increase. Below is the plot for the model with 10 random variables and the plot for the model with 100 random variables. The plot for the 100 random variables shows quite clearly that the model has stopped producing meaningful fitted probabilities and is predicting 0's and 1's. This is concerning because it is impossible for the model to have predicted probabilities of complete certainty.

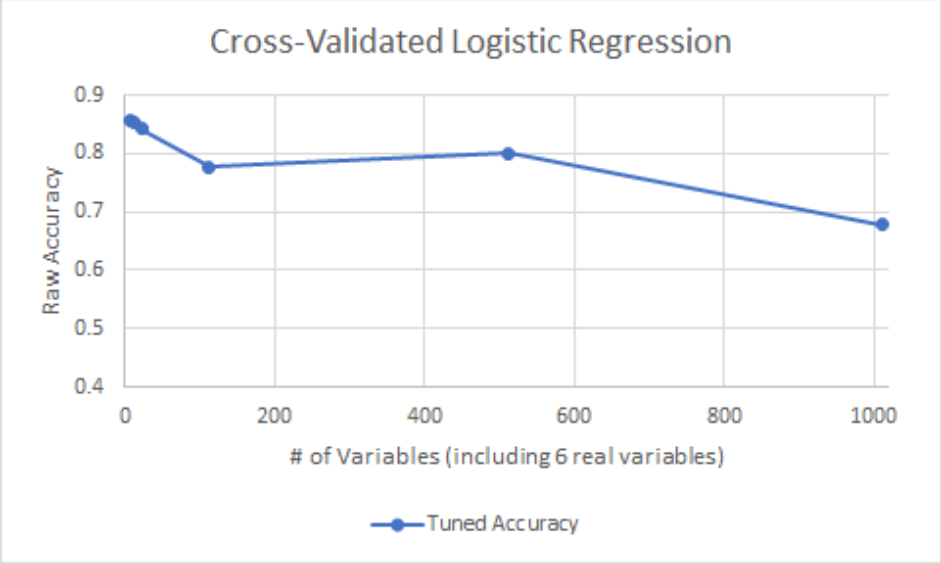




Another method for dealing with excess noise or too many extra variables is by implementing methods of dimension reduction. In this analysis, I implemented Principal Component Analysis (PCA) in an attempt to reduce the dimensionality of the data. The concept behind Principal Component Analysis is to transform the data in such a way that the maximum amount of variation is captured in the smallest number of variables by using linear combinations of the existing variables to create new, transformed variables called principal components. The results of the PCA analysis are in Appendix II and indicate that this data will not nicely reduce its dimensionality as was hoped. This is most likely due to the fact that 100 of the 113 predictor variables are independent random variables with no correlation to each other. Thus, the variation is not able to be captured in a linear combination of the variables in such a way that reduces the dimensionality. Therefore, I conclude that the original, cross-validated logistic regression model with forward stepwise variable selection is the best model for the data.

Similar to above, I implemented Principal Component Analysis with the addition of 500 random variables. The results can also be found in Appendix II, and are even worse than with the 100 random variables. Here, it would require including approximately 150 principal components in order to capture only 80 percent of the cumulative variance in the data and there is no clear drop off in variability. The results of the forward stepwise procedure results in an accuracy score of 0.80 with 175 true positives, 27 false positives, 73 true negatives, and 35 false negatives. Considering there are only 6 real variables out of 512 total variables, logistic regression is performing quite well. The stepwise variable selection is key to the model's success, however, because without it the model would result in complete separation of the data and become deprecated once the number of variables exceeds the number of observations (310).

Using the same process as above, I used Principal Component Analysis with the addition of 1000 random variables. The results are even worse than with the 500 random variables. Here, it would require including approximately 200 principal components in order to capture only 80 percent of the cumulative variance in the data. Therefore, again, the best model is simply with stepwise variable selection which results in an accuracy score of 0.6774, with 210 true positives and 100 false positives. This is the result of the model predicting every observation as abnormal, and the accuracy score converges to the ratio of abnormal and normal observations (210/310 = 0.6774). Thus, this model is essentially useless and logistic regression fails completely when there are over 1000 random variables and only 310 observations. Below is a plot which summarizes the performance of logistic regression when cross-validation, forward stepwise variable selection, and accuracy score performance optimizing are all used. Note that an accuracy of 0.6774 is a result of all abnormal predictions and is therefore useless.



Support Vector Machines

The next type of model, Support Vector Machines, are another very powerful and commonly used machine learning algorithm for classification. SVM's try to separate the data by drawing support vectors from each observation to a specified area of division, called a decision boundary, and tries to minimize the error. In two dimensions, this decision boundary would be a simple line, but as the dimensions increase, the dimensions of the decision boundary also increase. In R, I use the package e1071's svm() method to create Support Vector Machines. As with logistic regression, I first model only the real data before incrementally increasing the number of random variables to observe how Support Vector Machines are able to locate the signal in the noise.

In order to train a Support Vector Machine, I again created training and test sets with the same random seed that I used for my logistic regression models. The initial model is fit using the 6 real spinal measurement variables, with the provided random variables kept for later use. After training the model on the training set, and using the test set to make predictions, the model had an accuracy of 0.887, with 38 true positives, 5 false positives, 17 true negatives, and 2 false negatives, with positives representing abnormal classification and negatives representing normal classification. This model appears to perform very well. Using the tune.svm() function, which tunes the model parameters "gamma" and "cost" by trying variations of the parameters. The gamma parameter changes how far the influence of each training observation reaches, or how much weight is given to each training observation. The cost parameter defines how much the model should avoid misclassification during its optimization. After tuning, the accuracy score increases to 0.9032, with 38 true positives, 4 false positives, 18 true negatives, and 2 false

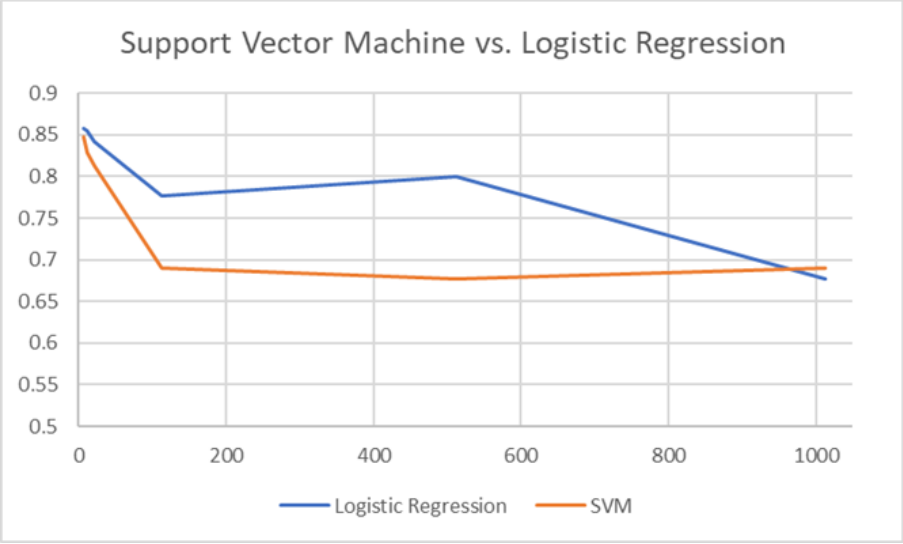
negatives. Both of these SVM models use the train/test split method. Next, I implement cross-validation with SVM's.

Cross-validation for Support Vector Machines uses the same process as cross-validation for logistic regression except without the stepwise variable selection and accuracy score tuning methods. Here, the Support Vector Machine tune.svm() function handles the tuning of the parameters and SVM's inherently optimize the performance measures. The SVM with cross-validation performs quite well. After tuning, the accuracy score is 0.8484, with 186 true positives, 23 false positives, 77 true negatives, and 24 false negatives. This is much more indicative of the true performance of the initial SVM model because through cross-validation, a prediction is made for every row in the dataset rather than only 20 percent of the data. Next, I evaluate how SVM's handle random noise by adding in the provided random variables, then incrementally adding additional random variables as I did with logistic regression.

After adding the provided random variables and refitting the model with cross-validation, the results are an accuracy of 0.8290, with 184 true positives, 27 false positives, 73 true negatives, and 26 false negatives. By adding the 6 random variables, the model had both more false positives and more false negatives. With the addition of 10 new random variables drawn from the standard normal distribution, the Support Vector Machine's performance worsened slightly. With cross-validation, the model had an accuracy of 0.8129, 181 true positives, 29 false positives, 71 true negatives, and 29 false negatives. Next, I add 100 random variables drawn from the standard normal distribution. After training the model with cross-validation and parameter tuning, the accuracy score is 0.6903, with 169 true positives, 55 false positives, 45 true negatives, and 41 false negatives. The main concern with this model is that there are nearly as many false negatives as true negatives, indicating it does a poor job of identifying normal

classifications. In terms of accuracy, this model performs only slightly better than a model which predicts all abnormal classifications.

With 500 random variables, the model breaks down completely and predicts every value to be abnormal, resulting in 210 true positives and 100 false negatives, with 0 true negatives and 0 false positives. Thus, the accuracy score of this model is simply the ratio of abnormal to normal spines in the dataset: 0.6774. This indicates that with 500 random variables the model is essentially useless. With 1000 random variables, the model actually performs slightly better than the model with 500 random variables. The accuracy score of the model is 0.6903, with 209 true positives, 95 false positives, 5 true negatives, and 1 false positive. Although still a poor performance, it is interesting that this model with 1000 random variables performed slightly better than the model with 500 random variables. This increase in performance is due to the simulation variability that results in fitting a model. Below is a plot which summarizes the accuracy score of the tuned, cross-validated models as the number of random variables increases. The accuracy score drops below 0.70 before 120 variables, which indicates SVM's do not handle excessive random noise very well.



Random Forests

Next, random forest models train multiple decision trees and average the results to produce a final model that is composed of elements from several decision trees. Each decision tree fits a model with a different set of predictor variables and the random forest model implements majority voting to make a prediction for each observation. For example, if five decision trees classify an observation as normal and 25 decision trees classify the same observation as abnormal, then the random forest will predict the observation is abnormal. The benefit of random forest models over a single decision tree is random forests tend to help reduce overfitting by basing the final prediction on the predictions of many different trees. First, I train a random forest model by using the train/test split method before implementing cross validation and adding random variables. The tuneRF function is a built-in method for tuning the "mtry" parameter of the random forest model in order to optimize the predictions. The mtry parameter tells the random forest model how many variables to randomly select at each split in a tree.

The random forest model fit only on the 6 real variables and using the train/test split method has an accuracy score of 0.8065, 38 true positives, 2 false positives, 12 true negatives, and 10 false negatives. After running the tuneRF function on this model, the accuracy improved to 0.8306, with 152 true positives, 18 false positives, 53 true negatives, and 25 false negatives with positives representing abnormal classification and negatives representing normal classification. Notice that the confusion matrix for the tuned model does not add up to 310. This is because the tuneRF function produces a confusion matrix for the training set, which here is only 248 variables. Despite this weakness, the tune function does still give an idea of how much room for improvement there is in the model through the tuning of parameters. In the rest of the random forest models, I run the tune function first and then use the optimal mtry value to train

the model. The importance function for the tuned random forest model indicates that the most important variable is degree_spondylolisthesis with a MeanDecreaseGini value of 39.96. The next most important variables in the model are pelvic_radius, sacral_slope, and pelvic_incidence with MeanDecreaseGini values of 16.39, 13.11, and 13.60 respectively. This MeanDecreaseGini value is the total decrease in node impurities from splitting on the variable, averaged over all trees, measured by the Gini coefficient. The Gini coefficient is a metric for homogeneity from 0 (homogeneous) to 1 (heterogeneous) and the values listed above are summed for each variable and normalized at the end of the calculation.

In order to implement cross-validation, I use the same process as I used with the previous models. The cross-validated initial model used only the 6 real variables. After training the model five times and making predictions accordingly, the cross-validated initial model has an accuracy of 0.8419, with 187 true positives, 23 false positives, 74 true negatives, and 26 false negatives. Here, the accuracy score increased, but in general, the accuracy score of the cross-validated model should be relatively similar to the accuracy score of the train/test split method. If the cross-validated accuracy score decreases significantly, it is likely that the train/test split method is overfitting.

The remainder of the random forest models are trained using cross-validation. Next, I train a model with the provided random variables included. The cross-validated and tuned random forest model using all twelve provided variables has an accuracy of 0.8161 and 188 true positives, 22 false positives, 65 true negatives, and 35 false negatives, with positives representing abnormal classification and negatives representing normal classification. Considering the number of variables doubled, the model performed relatively well. For this model, the importance function indicates that the three most important variables were degree_spondylolisthesis,

pelvic_radius, and pelvic tilt. A key element of the varImpPlots in Appendix IV is that the meaningful variables are all more important than the random variables.

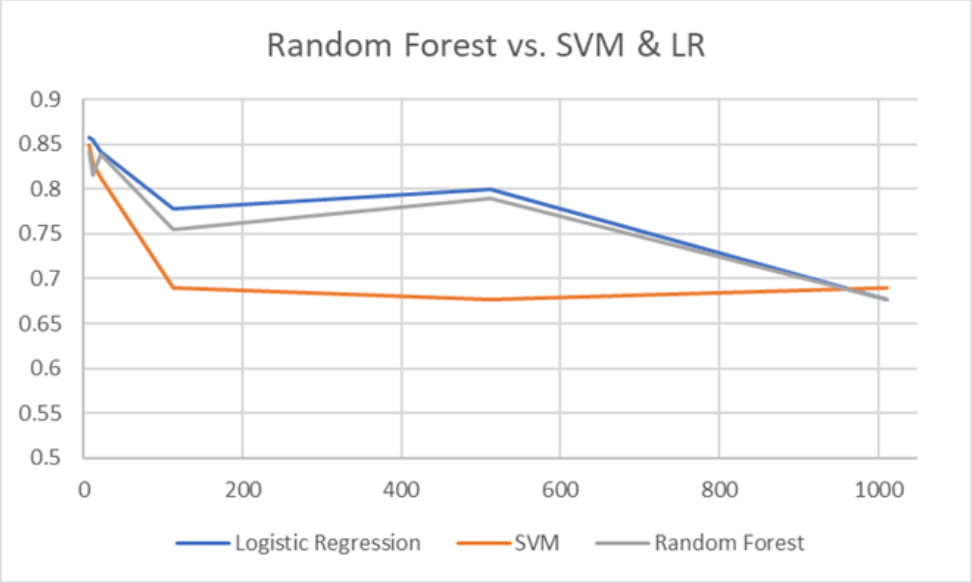
Next, I add 10 random variables consisting of randomly drawn values from a standard normal distribution. These new variables are labeled X1, X2, ... X10. The cross-validated and tuned model with the 12 provided variables plus the 10 newly created random variables resulted in an accuracy of 0.8387, with 185 true positives, 25 false positives, 75 true negatives, and 25 false negatives. The importance function output indicates that the six real variables were again the most important, with the provided random variables and the newly created random variables varying in low importance. Again, degree_spondylolisthesis is by far the most important variable.

Next, I trained the cross-validated and tuned random forest model with 100 variables drawn from the standard normal distribution in addition to the 12 provided variables. The newly added random variables are again named X1, X2, ... X100, respectively. This model produced an accuracy of 0.7548, 192 true positives, 18 false positives, 42 true negatives, and 58 false negatives, with positives representing abnormal classification and negatives representing normal classification. This model's main weakness is that there are more false negatives than true negatives. The importance function again shows the six real variables at the top, with the random variables scattered below.

Next, after adding 500 random variables from the standard normal distribution to the provided 12 variables, a cross-validated and tuned random forest model is trained. The results of the model are an accuracy score of 0.7903, 182 true positives, 28 false positives, 63 true negatives, and 37 false negatives, with positives representing abnormal classification and

negatives representing normal classification. Again, the variable importance indicates the six real variables are the most important, with the random variables scattered below.

Finally, I added 1000 random variables to the 12 provided variables, again drawn from a standard normal distribution. For this model, there are a total of 1012 variables and still only 310 observations. After training the cross-validated and tuned model, the accuracy score is 0.6774, 210 true positives, 100 false positives, 0 true negatives, and 0 false negatives, with positives representing abnormal classification and negatives representing normal classification. This model is simply classifying all of the observations as abnormal, thus the number of true positives is equal to the number of abnormal observations in the data. The plot below summarizes the performance of the tuned random forest models with respect to accuracy score.



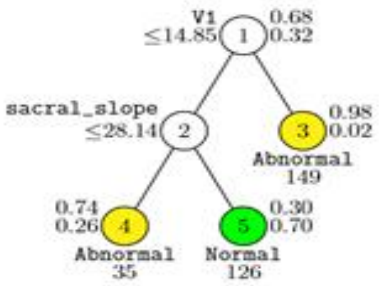
G.U.I.D.E

The last model use with the spine data is G.U.I.D.E, created by Professor Wei-Yin Loh at the University of Wisconsin, Madison. G.U.I.D.E stands for “Generalized, Unbiased, Interaction Detection and Estimation”, and the algorithm is based on decision trees. GUIDE has many

improvements over ordinary decision trees including built in cross-validation and a smarter algorithm for choosing variables and splits at each node. One of the options in GUIDE is to create an ensemble model similar to a random forest collection of trees. However, for this analysis, I decided to simply use the single tree model fitting methods of GUIDE. For simplicity, I opted to use all of the default settings that come with the GUIDE program, which include model fitting a single tree for classification, estimated prior probabilities, unit misclassification costs, and other default options. The GUIDE modeling process involves first creating a data description file that tells GUIDE how to handle missing values, variable types, whether or not the data includes headers, etc. Next, GUIDE creates a .IN file with all of the selected settings and type of classification. Finally, the program reads the data description file, the actual data file, and the .IN file, fits a tree to the data, and produces several files including a .OUT file which summarizes the output, a text file with the stored fitted classifications and actual classifications, and finally a .TEX file with the LaTex code for the decision tree diagram.

I first trained a GUIDE decision tree with just the 12 provided variables, including the six provided random variables. The results were an accuracy score of 0.839 with 172 true positives, 12 false positives, 88 true negatives, and 38 false negatives, with abnormal classification as positives, and normal classification as negatives. Each time a tree is fitted with GUIDE, a tree diagram is produced. Below is the tree diagram for the model with just the provided 12 variables. The tree is interpreted as follows: V1 is degree_spondylolisthesis and if its value is less than or equal to 14.85 and sacral_slope is less than 28.14, then the observation is classified as abnormal, if degree_spondylolisthesis is less than or equal to 14.85 and sacral_slope is greater than 28.14 then the observation is classified as normal, and if degree_spondylolisthesis is greater than 14.85,

then the observation is classified as abnormal. For each leaf node in the diagram, the proportions given indicate how many of the training observations are abnormal and normal.



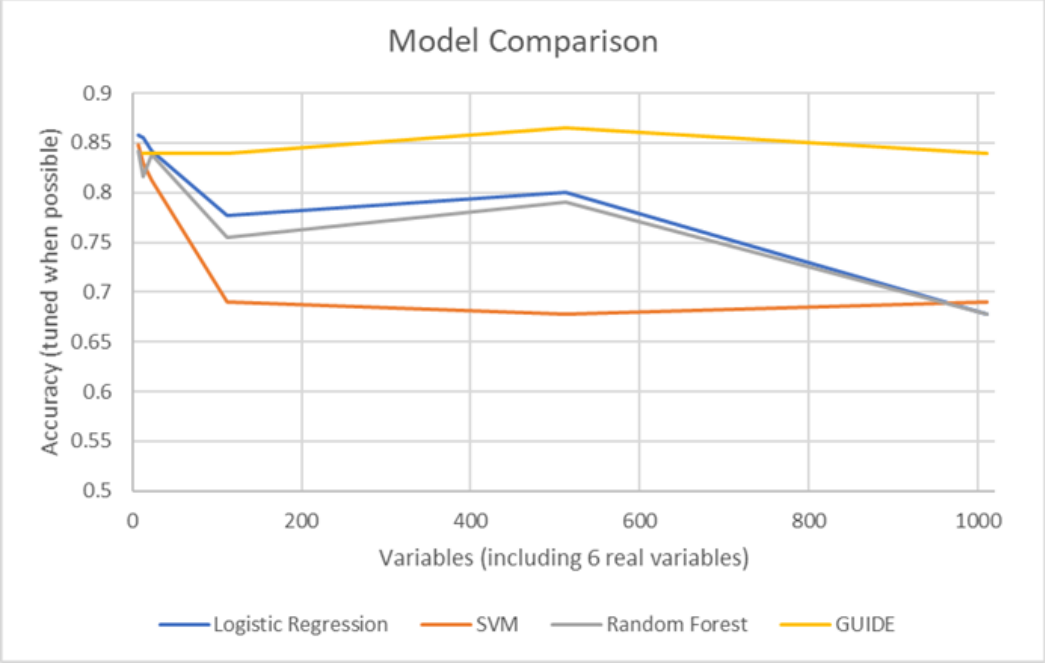
GUIDE v.29.0 0.50-SE pruned classification tree for predicting `classification` using estimated priors and unit misclassification costs. Maximum number of split levels is 10 and minimum node sample size is 2. At each split, an observation goes to the left branch if and only if the condition is satisfied. `V1` = `degree_spondylolisthesis`. Predicted classes and sample sizes printed below terminal nodes; class proportions for `classification` = `Abnormal` and `Normal` beside nodes. Second best split variable at root node is `pelvic_radius`.

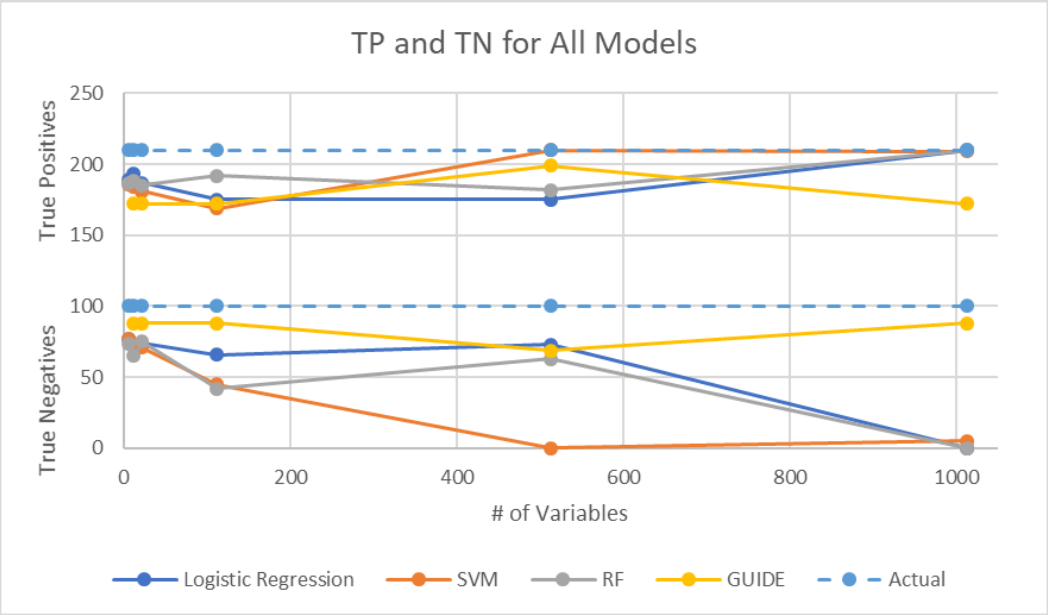
For example, node number three is composed of 98 percent abnormal observations and 2 percent normal observations, thus observations that fall into that leaf are classified as abnormal. The remaining trees for the other GUIDE models can be interpreted similarly and can be found in Appendix V.

Next, I added 10 random variables drawn from the standard normal distribution and repeated the tree fitting process. The results of this GUIDE tree were the exact same as the tree fit with just the 12 provided variables and the tree diagram is in Appendix V, tree 2. Next, I added 100 random variables to the data, and the results of the model were again the same as the original model, and the tree diagram can be found in Appendix V, tree 3. The GUIDE model fit with 500 random variables did change slightly from the previous models. The model resulted in 199 true positives, 31 false positives, 69 true negatives, and 11 false negatives, with an accuracy score of 0.865. The tree diagram can be found in Appendix V, tree 4. Finally, I fit the GUIDE

model with 1000 random variables. The results of this model were the same as the first 3 GUIDE models, and the tree diagram can be found in Appendix V, tree 5.

In order to calculate variable importance with the GUIDE, the program must be run again with "model importance" option selected instead of "model fitting". The most important variables throughout the GUIDE tree models were: degree_spondylolisthesis, pelvic_radius, pelvic_tilt, pelvic_incidence, lumbar_lordosis_angle, and sacral_slope, all of which are the real, meaningful spinal measurement variables. GUIDE's main advantage is its pruning process. The algorithm quickly prunes away the variables that are not significant. Below is a plot which summarizes the accuracy score of each of the models and a plot which shows the true positives and true negatives for each model. Both plots are useful for visually comparing the performances of the various models.





Overall, there is quite a large difference in performance between types of models with increasing levels of noise. GUIDE clearly performed the best, with logistic regression and random forest performing very similarly. One reason these models performed better than SVM's is because they had more effective variable selection methods. The models' performance tended to rely heavily on the tuning and variable selection process, especially when the excess random variables were introduced. Thus, the models with strong tuning and variable selection performed stronger than those that did not. Random forest and logistic regression both had a sound tuning process and an explicit variable selection procedure, resulting in rather strong performance until the excess data was overwhelming in comparison to the number of observations. Although GUIDE is new to me, the several improvements made by Professor Wei-Yin Loh appear to be very beneficial to its performance. Given time, I am confident that the performance of each of these models could be improved. However, the goal of this project was to compare how the selected models handled incrementally added random noise and subsequently kept track of the signal in the noise. In order to compare the performance of the models, I tried to keep the amount

of model tuning to a consistent level. GUIDE definitely appears to have outperformed each of the classic machine learning models, even with only its default settings and a single tree, which indicates the GUIDE program is very powerful. The process of comparing machine learning models has provided valuable insight into the importance of model tuning and variable selection processes.

Sources

Breiman, Leo, et al. “Package ‘RandomForest.’” r-Project, 25 Mar. 2018.

Hunt, Tyler. “Package ‘ModelMetrics.’” r-Project, 3 Nov. 2018.

Kuhn, Max. “Package ‘Caret.’” r-Project, 27 May 2018.

Loh, Wei-Yin. “GUIDE User Manual.” Department of Statistics, University of Wisconsin–Madison, 18 Oct. 2018.

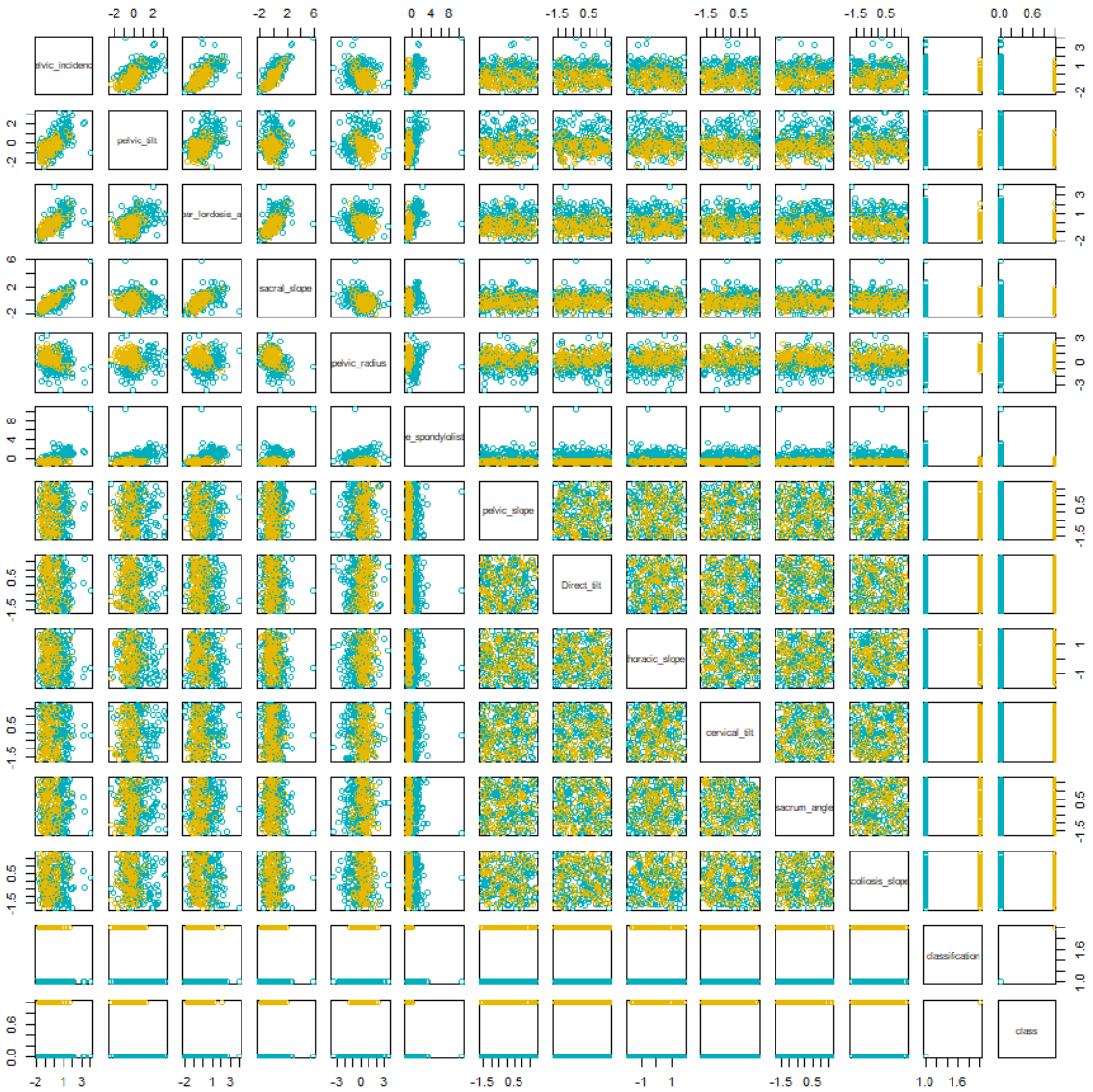
Meyer, David, et al. “Package ‘e1071.’” r-Project, 28 July 2018.

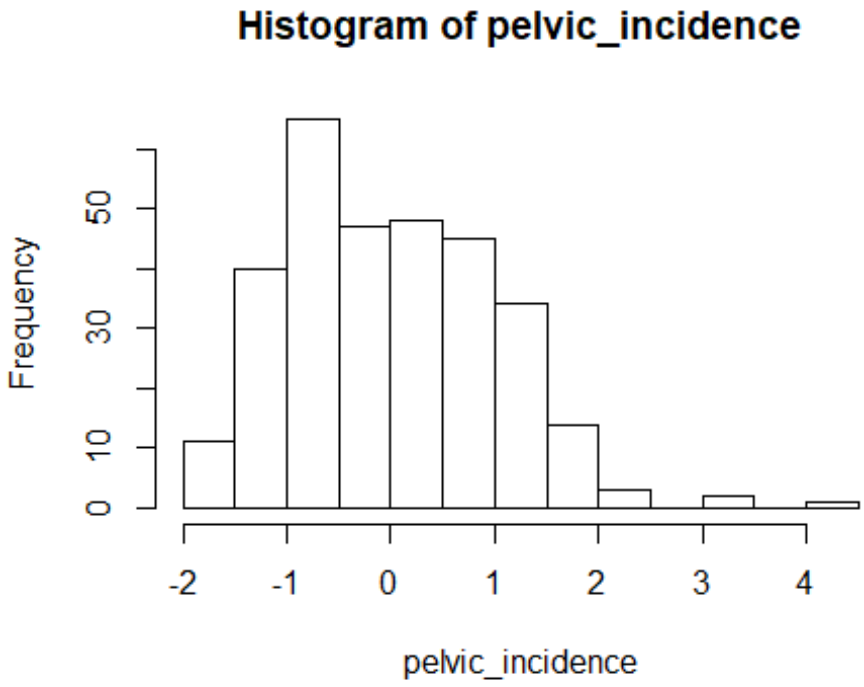
Ng, Andrew. “Support Vector Machines.” cs229, Stanford.edu.

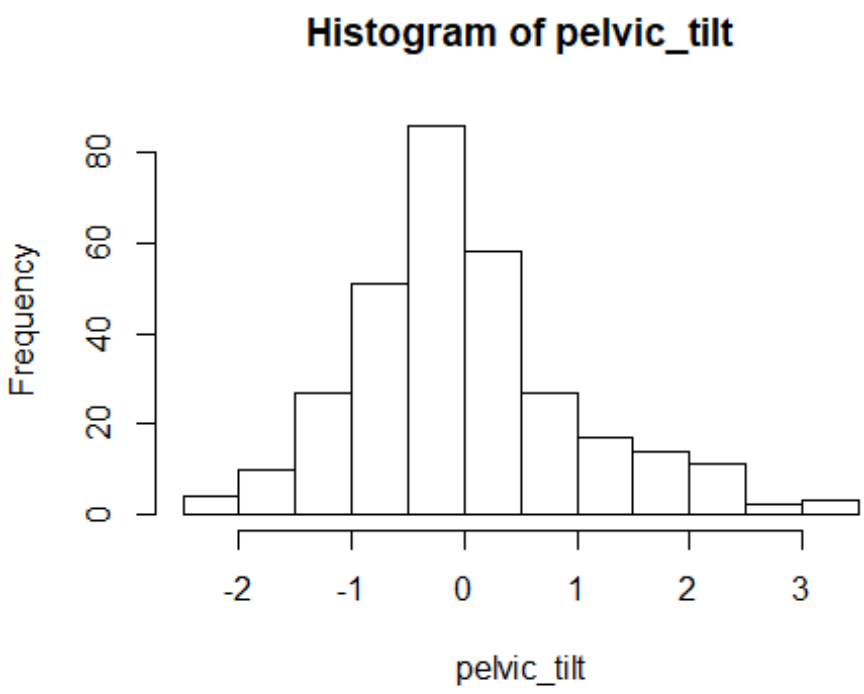
“Random Forests Leo Breiman and Adele Cutler.” Statistics at UC Berkeley | Department of Statistics.

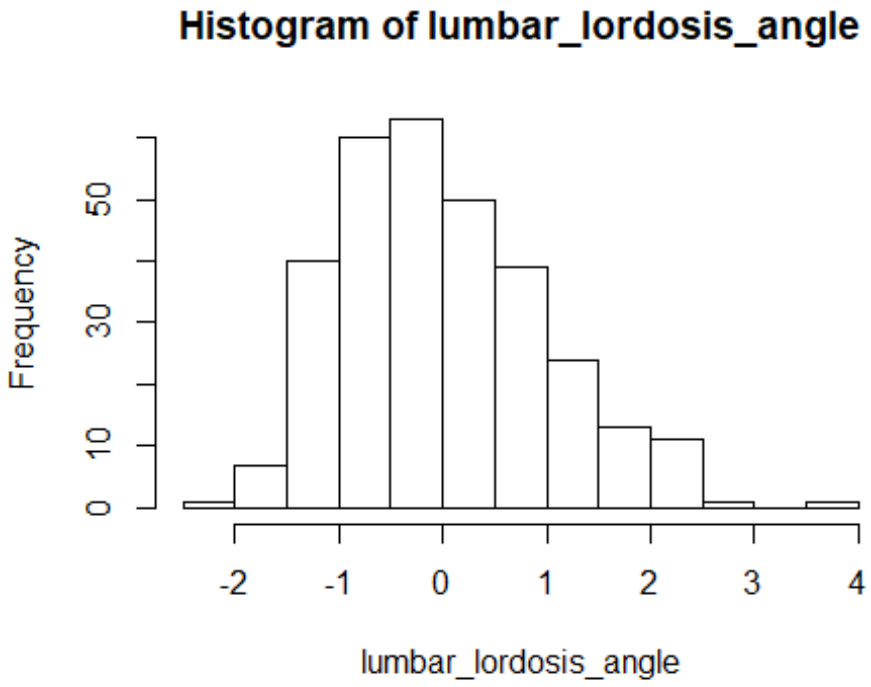
Appendix I: Initial Exploration

```
my_cols <- c("#00AFBB", "#E7B800", "#FC4E07")

pairs(data, col = my_cols[data$classification]) #pairs plot
```

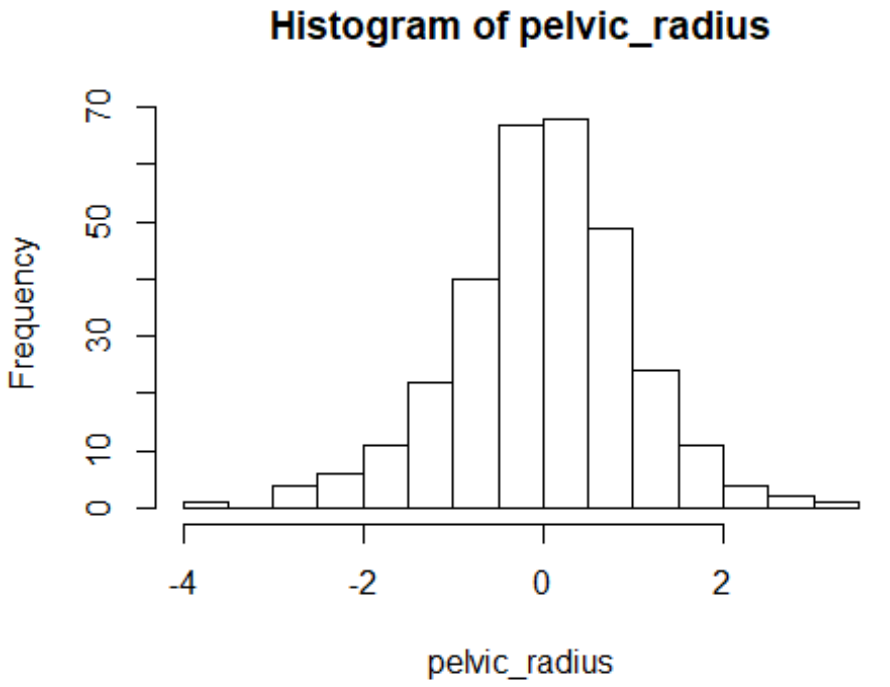


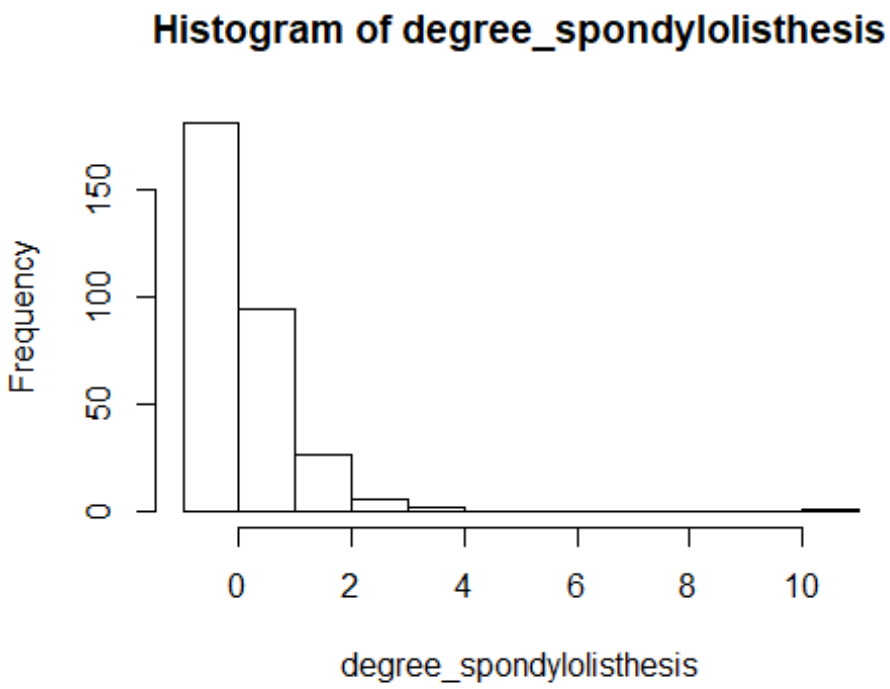

Histogram of pelvic_incidence
Frequency
0
10
30
50
-2
-1
0
1
2
3
4
pelvic_incidence

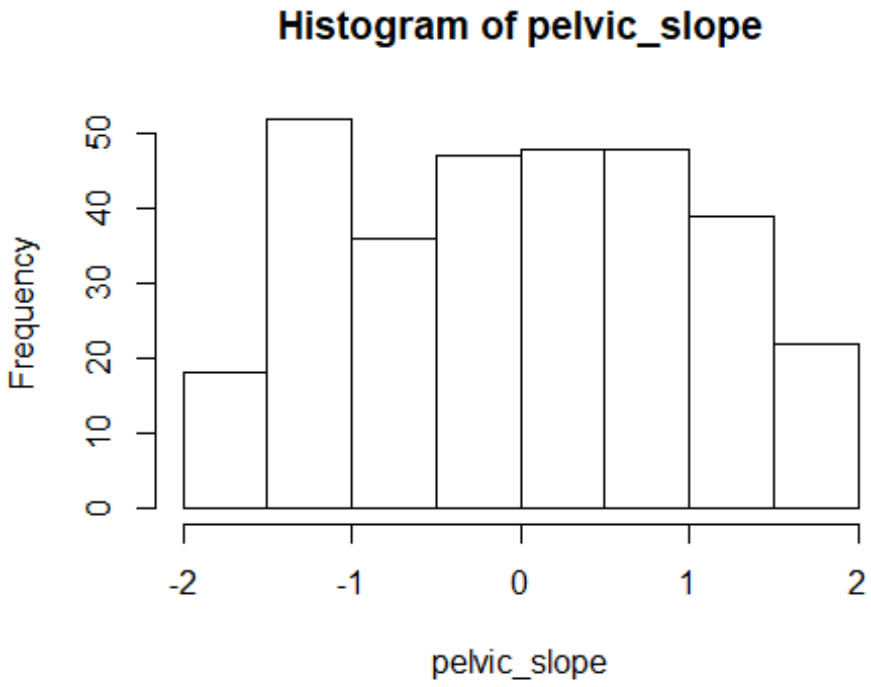

Histogram of pelvic_tilt
Frequency
0
20
40
60
80
-2
-1
0
1
2
3
pelvic_tilt

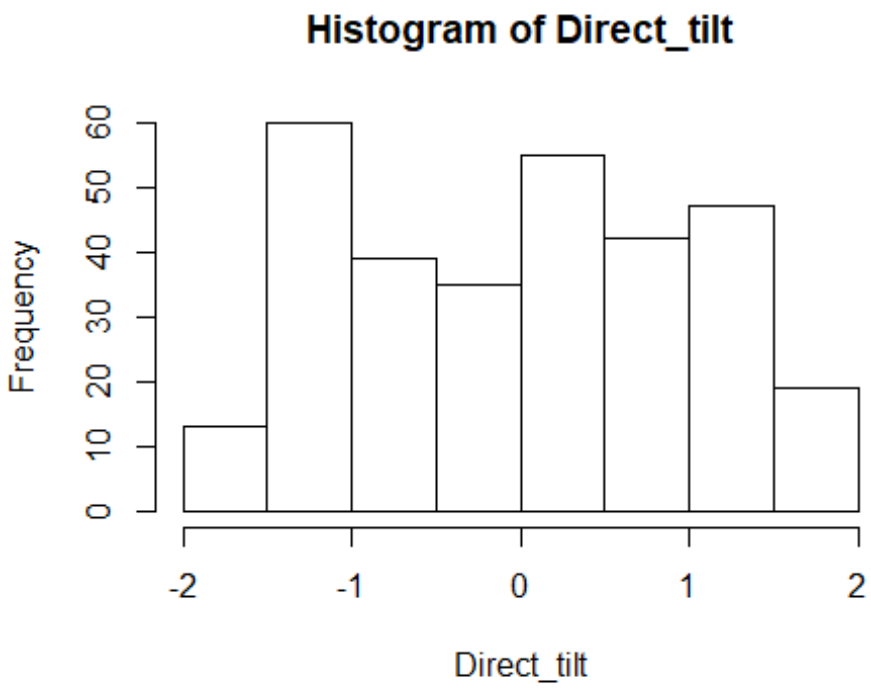

Histogram of lumbar_lordosis_angle
Frequency
0
10
30
50
-2
-1
0
1
2
3
4
lumbar_lordosis_angle


Histogram of sacral_slope
Frequency
0
20
40
60
80
120
-2
0
2
4
6
sacral_slope


Histogram of pelvic_radius
Frequency
0
10
30
50
70
-4
-2
0
2
pelvic_radius


Histogram of degree_spondylolisthesis
Frequency
0
50
100
150
0
2
4
6
8
10
degree_spondylolisthesis


Histogram of pelvic_slope
Frequency
0
10
20
30
40
50
-2
-1
0
1
2
pelvic_slope

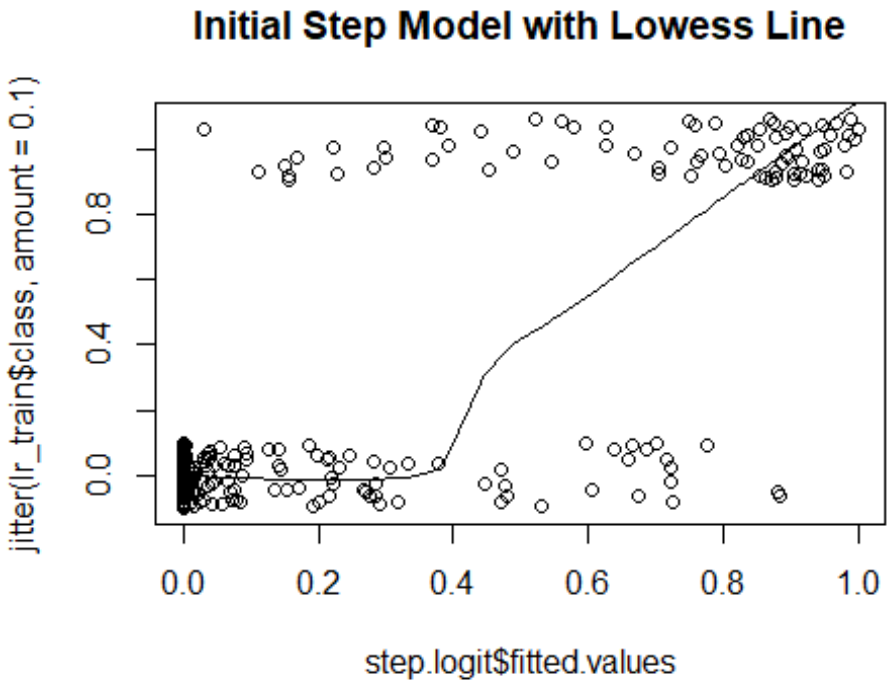

Histogram of Direct_tilt
Frequency
0
10
20
30
40
50
60
-2
-1
0
1
2
Direct_tilt

Appendix II: Logistic Regression

Initial Model

```
Call:
glm(formula = classification ~ pelvic_tilt + pelvic_incidence +
    lumbar_lordosis_angle + sacral_slope + pelvic_radius + degree_spon
dylolisthesis,
    family = binomial(link = "logit"), data = lr_train)

Deviance Residuals:
     Min        1Q    Median        3Q       Max
-2.40458  -0.39229  -0.04114   0.40079   2.80457

Coefficients:
                           Estimate Std. Error z value Pr(>|z|)
(Intercept)              -3.304e+00  5.267e-01  -6.274 3.52e-10 ***
pelvic_tilt              -5.314e+08  4.441e+08  -1.197  0.23146
pelvic_incidence          9.151e+08  7.648e+08   1.197  0.23146
lumbar_lordosis_angle     9.587e-02  4.788e-01   0.200  0.84132
sacral_slope             -7.127e+08  5.956e+08  -1.197  0.23146
pelvic_radius             1.147e+00  3.156e-01   3.633  0.00028 ***
degree_spondylolisthesis -6.375e+00  9.626e-01  -6.623 3.52e-11 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 308.84  on 247  degrees of freedom
Residual deviance: 141.83  on 241  degrees of freedom
AIC: 155.83

Number of Fisher Scoring iterations: 7


Call:
glm(formula = class ~ degree_spondylolisthesis + sacral_slope +
    pelvic_radius + pelvic_tilt + Direct_tilt, family = binomial(link
= "logit"),
    data = lr_train)
```

```
Deviance Residuals:
     Min        1Q    Median        3Q       Max
-2.07271  -0.37358  -0.04328   0.40737   2.63670

Coefficients:
                         Estimate Std. Error z value Pr(>|z|)
(Intercept)               -3.3643     0.5325  -6.318 2.64e-10 ***
degree_spondylolisthesis  -6.3018     0.9480  -6.648 2.98e-11 ***
sacral_slope               1.5558     0.3547   4.386 1.15e-05 ***
pelvic_radius              1.1881     0.3089   3.846  0.00012 ***
pelvic_tilt               -0.7932     0.3237  -2.451  0.01426 *
Direct_tilt               -0.3782     0.2264  -1.671  0.09473 .
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 308.84  on 247  degrees of freedom
Residual deviance: 140.48  on 242  degrees of freedom
AIC: 152.48

Number of Fisher Scoring iterations: 7

     [,1] [,2]
[1,]   36    6
[2,]    4   16

     [,1] [,2]
[1,]   36    7
[2,]    4   15

[1] 0.8387097

[1] 0.8225806

     [,1] [,2]
[1,]   37    6
[2,]    3   16

[1] 0.8548387
```

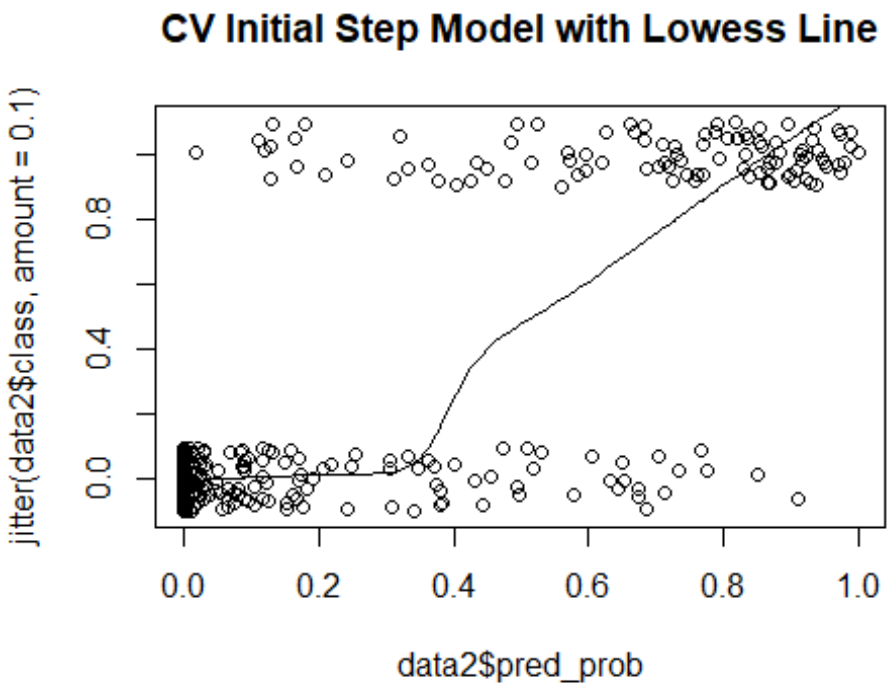


Find Optimal Cutoff for Train/Test Initial Model with Forward Stepwise

```
     [,1] [,2]
[1,]   40   16
[2,]    0    6

[1] 0.7419355
```

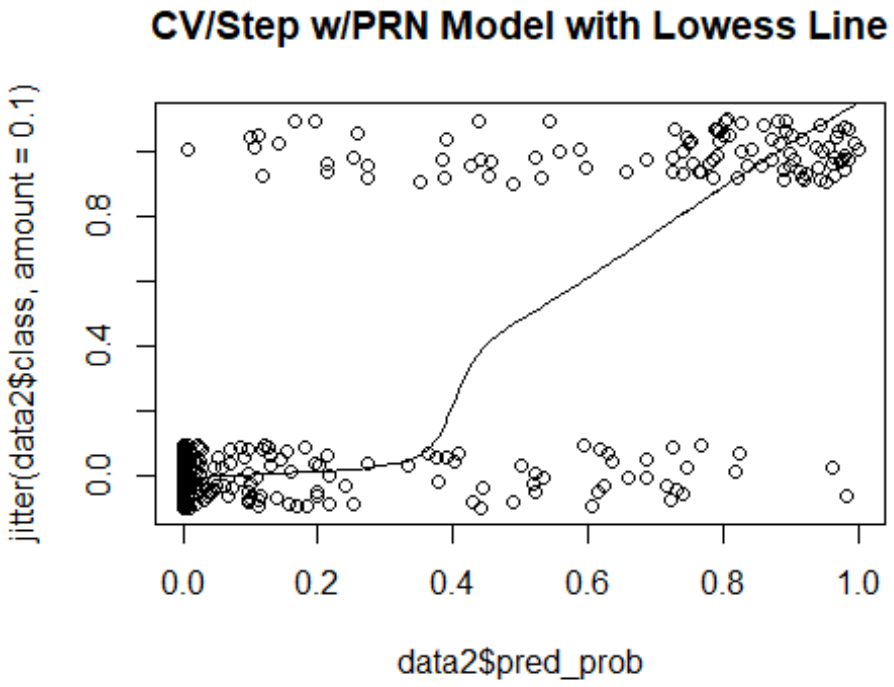
Cross-Validation with Initial Model with Optimizing with Set Seed

```
     [,1] [,2]
[1,]  189   23
[2,]   21   77

[1] 0.8580645
```



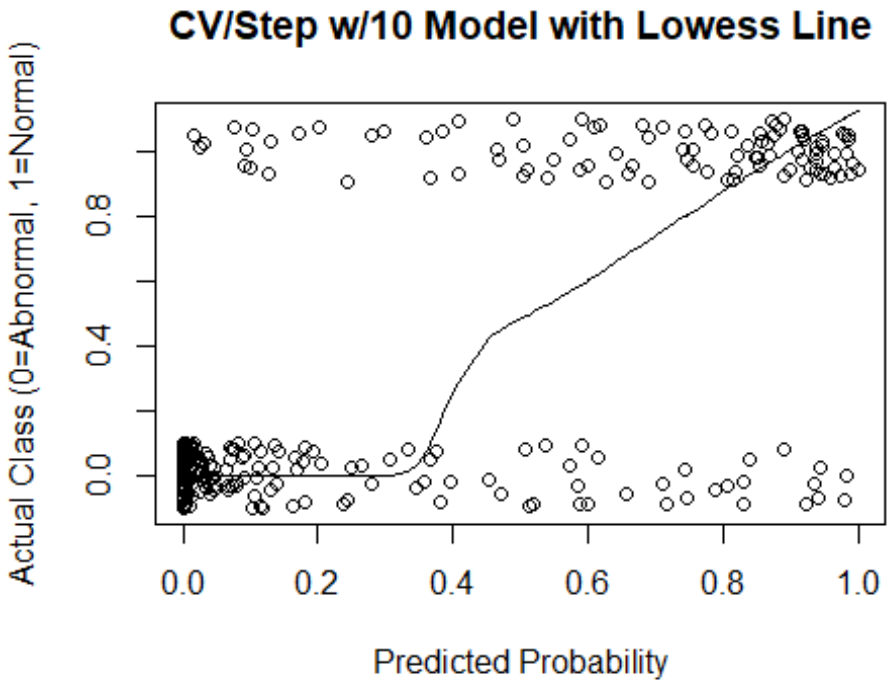
Accuracy Distribution of Cross-Validation with Initial Model with Optimizing
Add Provided Random Noise

```
     [,1] [,2]
[1,]  193   28
[2,]   17   72

[1] 0.8548387
```



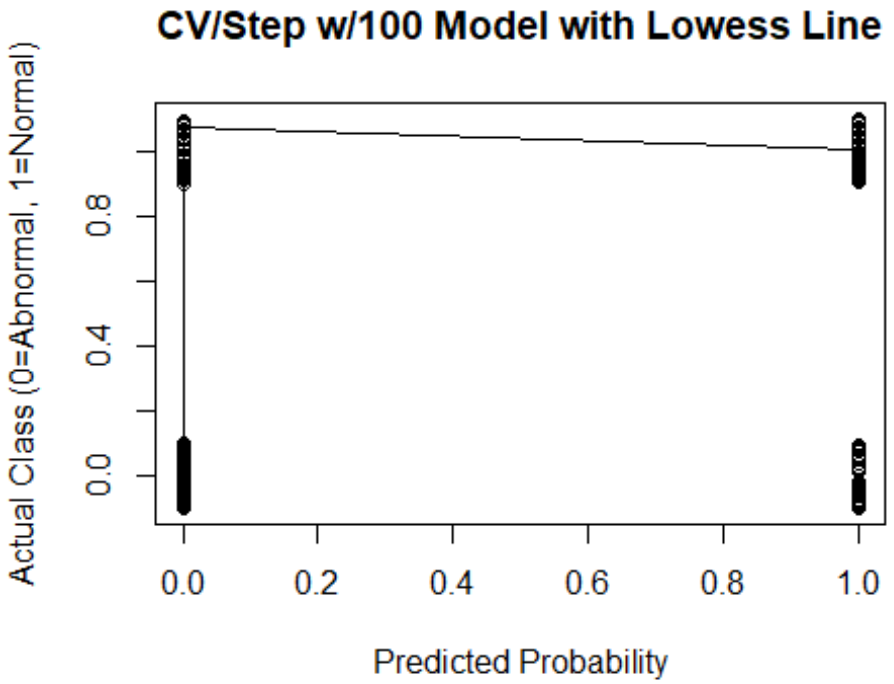
Add 10 Random Variables

```
     [,1] [,2]
[1,]  187   26
[2,]   23   74

[1] 0.8419355
```

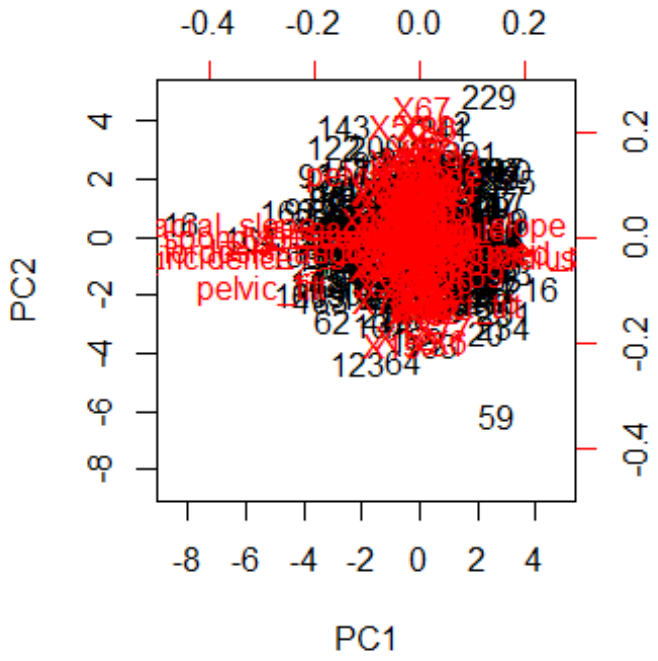


Add 100 Random Variables with set seed

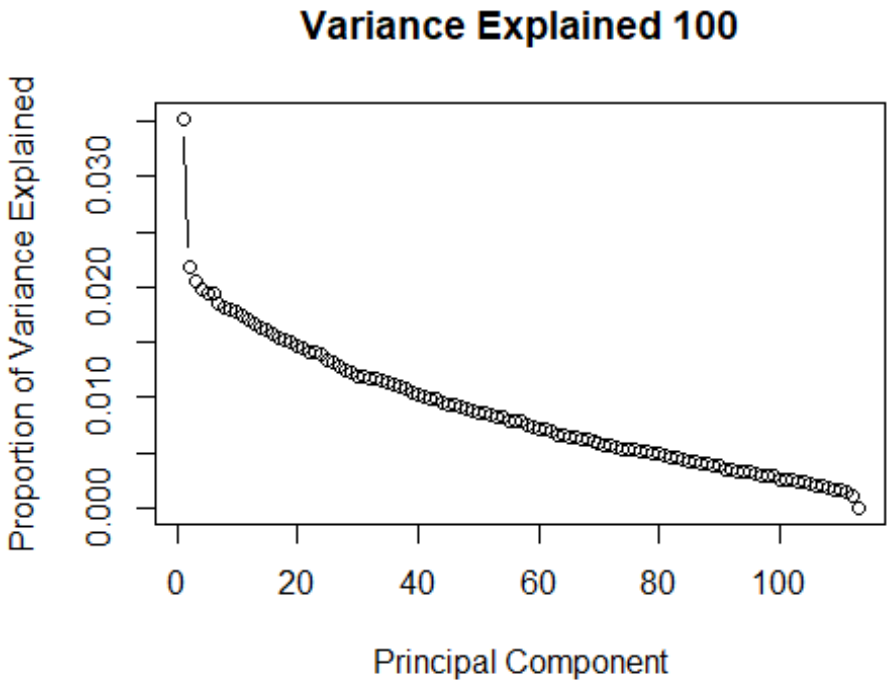
```
     [,1] [,2]
[1,]  175   34
[2,]   35   66

[1] 0.7774194
```

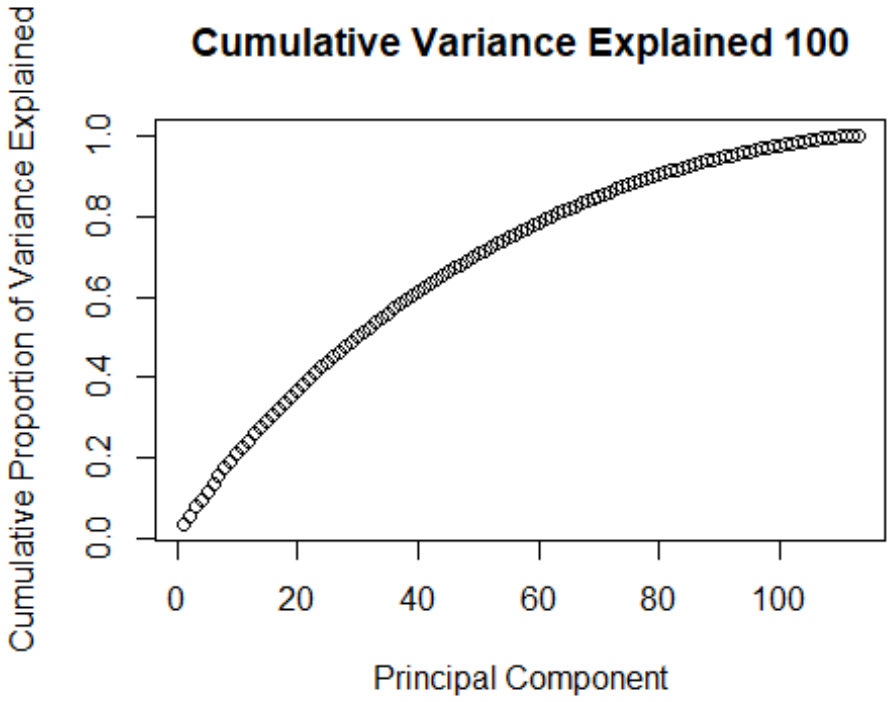



```
[1] "sdev"     "rotation" "center"   "scale"    "x"

                               PC1         PC2         PC3         PC4
pelvic_incidence      -0.4585396 -0.03986544  0.02559957 -0.04130906
pelvic_tilt           -0.2988447 -0.09654887  0.10297940  0.09719343
lumbar_lordosis_angle -0.3962508 -0.02873156 -0.03282033  0.01412432
sacral_slope          -0.3659877  0.02079636 -0.04390970 -0.12551259
pelvic_radius          0.1216077 -0.04026772 -0.04059324  0.06597922
```



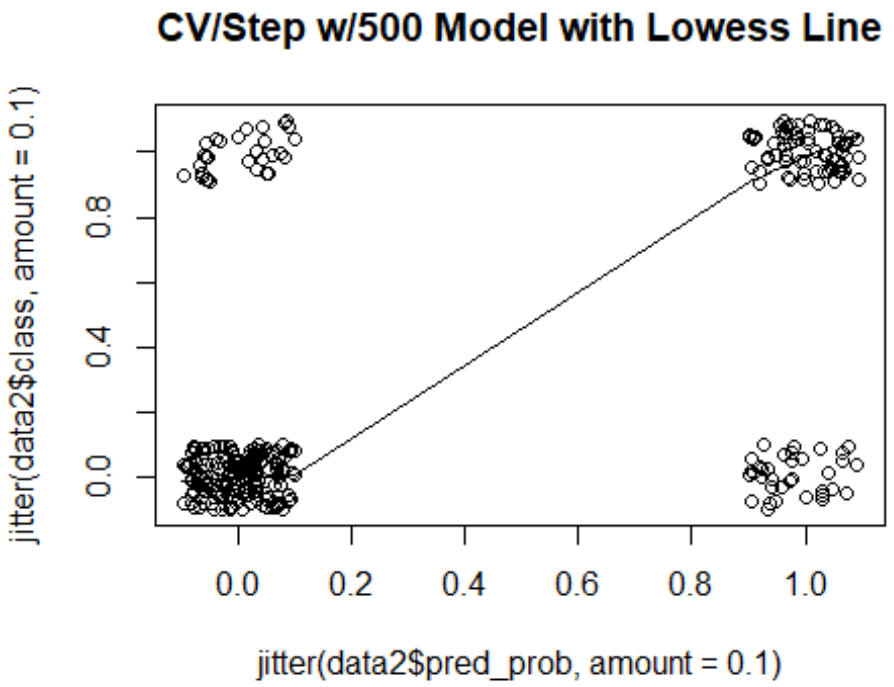

```
 [1] 3.980433 2.465193 2.320331 2.234025 2.204784 2.197591 2.101477
 [8] 2.045163 2.019319 1.999980

 [1] 0.03522507 0.02181587 0.02053390 0.01977013 0.01951136 0.01944771
 [7] 0.01859714 0.01809879 0.01787008 0.01769893 0.01732758 0.01708304
[13] 0.01661430 0.01631056 0.01610876 0.01578239 0.01542219 0.01515624
[19] 0.01498812 0.01468698
```


Variance Explained 100
Proportion of Variance Explained
0.000
0.010
0.020
0.030
0
20
40
60
80
100
Principal Component

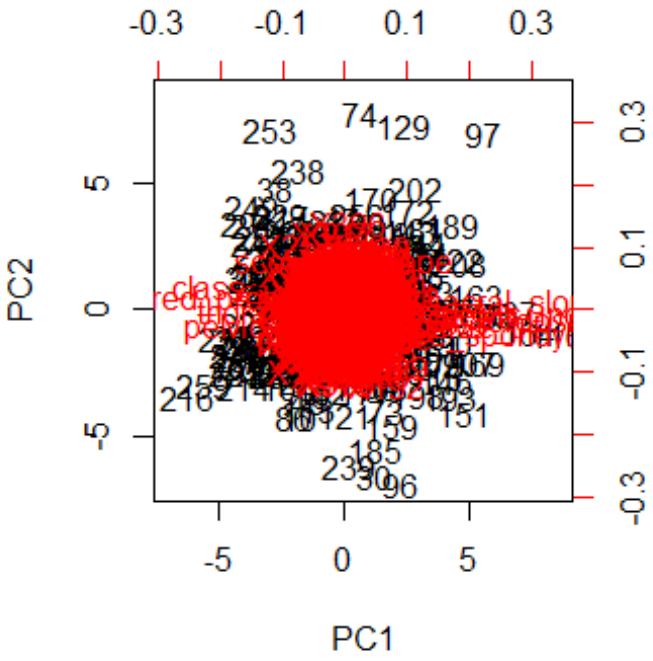

Cumulative Variance Explained 100
Cumulative Proportion of Variance Explained
0.0
0.2
0.4
0.6
0.8
1.0
0
20
40
60
80
100
Principal Component

Accuracy Distribution for 100 Random Variables
Add 500 Random Variables

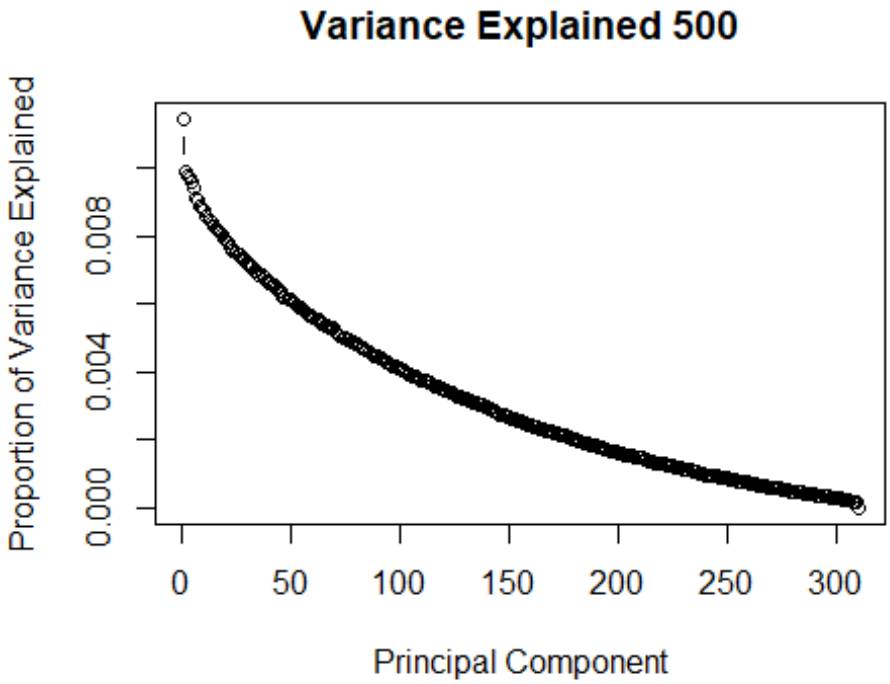
```
     [,1] [,2]
[1,]  175   27
[2,]   35   73

[1] 0.8
```

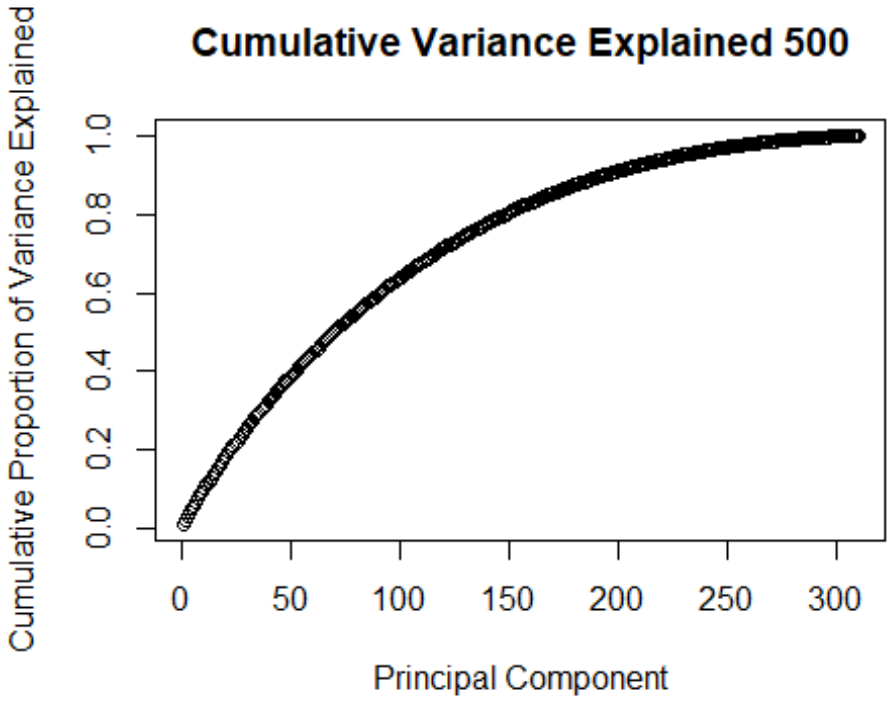



```
[1] "sdev"     "rotation" "center"   "scale"    "x"

                            PC1          PC2          PC3           PC
4
pelvic_incidence       0.3427986 -0.008366553  0.025014206  0.03062929
1
pelvic_tilt            0.2150461 -0.029647281 -0.026541067  0.01789278
7
lumbar_lordosis_angle  0.2936539 -0.017941039 -0.003384455  0.03059258
7
sacral_slope           0.2798461  0.011361719  0.051909732  0.02598992
9
pelvic_radius         -0.1085754 -0.028813317  0.027042366 -0.00163025
4
```



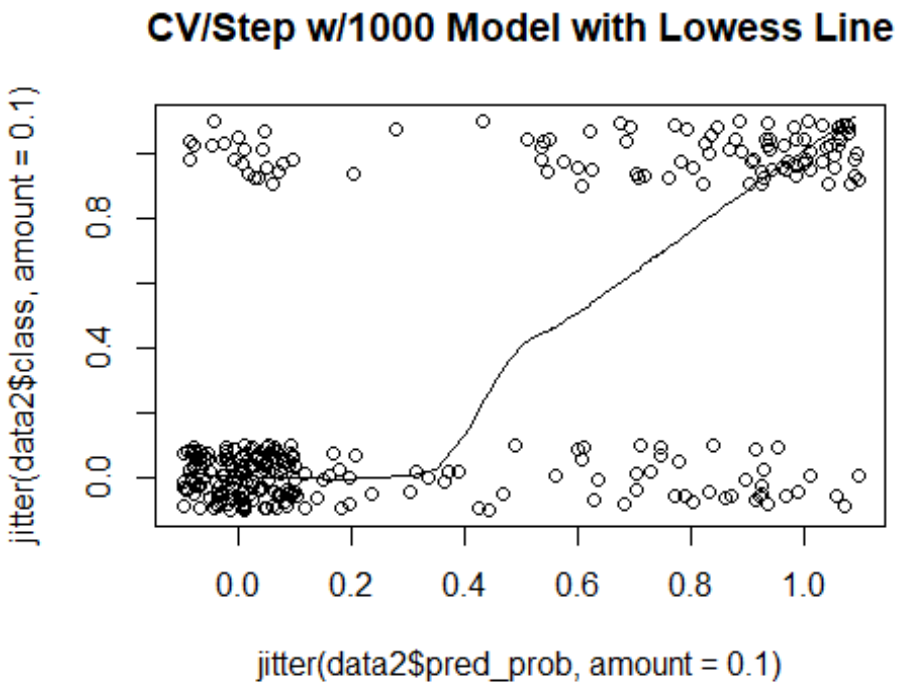

```
 [1] 5.874800 5.064426 5.022195 4.982542 4.934381 4.843773 4.679750
 [8] 4.662627 4.564496 4.523734

 [1] 0.011429572 0.009852969 0.009770806 0.009693662 0.009599963
 [6] 0.009423684 0.009104571 0.009071258 0.008880343 0.008801038
[11] 0.008727385 0.008548648 0.008463923 0.008390731 0.008292869
[16] 0.008238491 0.008174125 0.008094779 0.008002274 0.007914800
```


Variance Explained 500
Proportion of Variance Explained
0.000
0.004
0.008
0
50
100
150
200
250
300
Principal Component

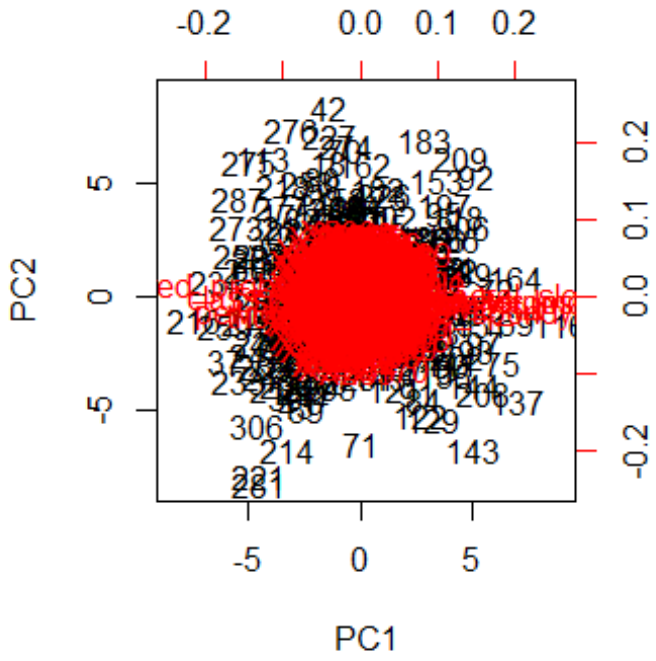

Cumulative Variance Explained 500
Cumulative Proportion of Variance Explained
0.0
0.2
0.4
0.6
0.8
1.0
0
50
100
150
200
250
300
Principal Component

Add 1000 Random Variables

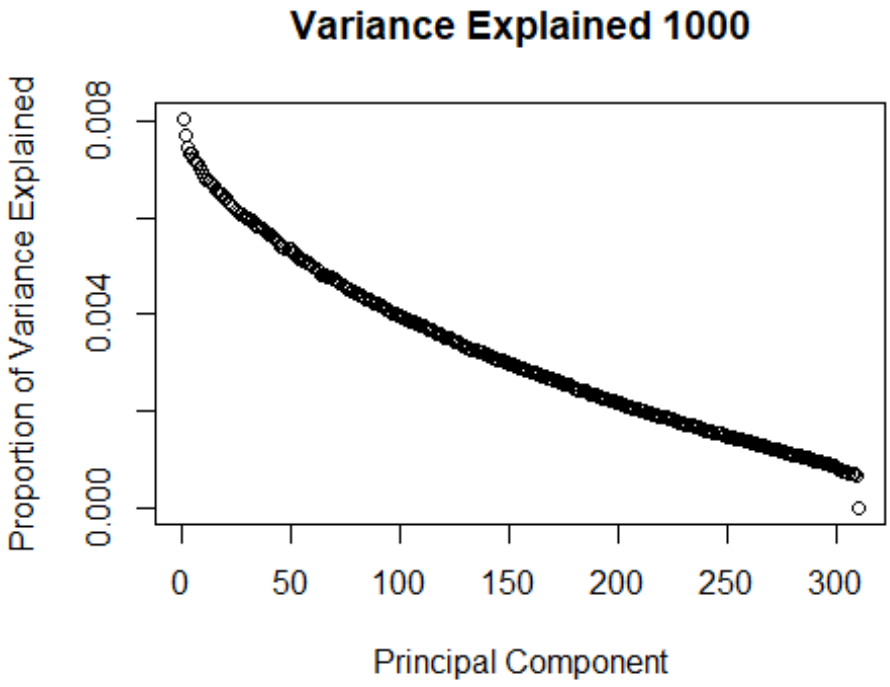
```
NULL

     [,1] [,2]
[1,]  171   25
[2,]   39   75

[1] 0.7935484
```

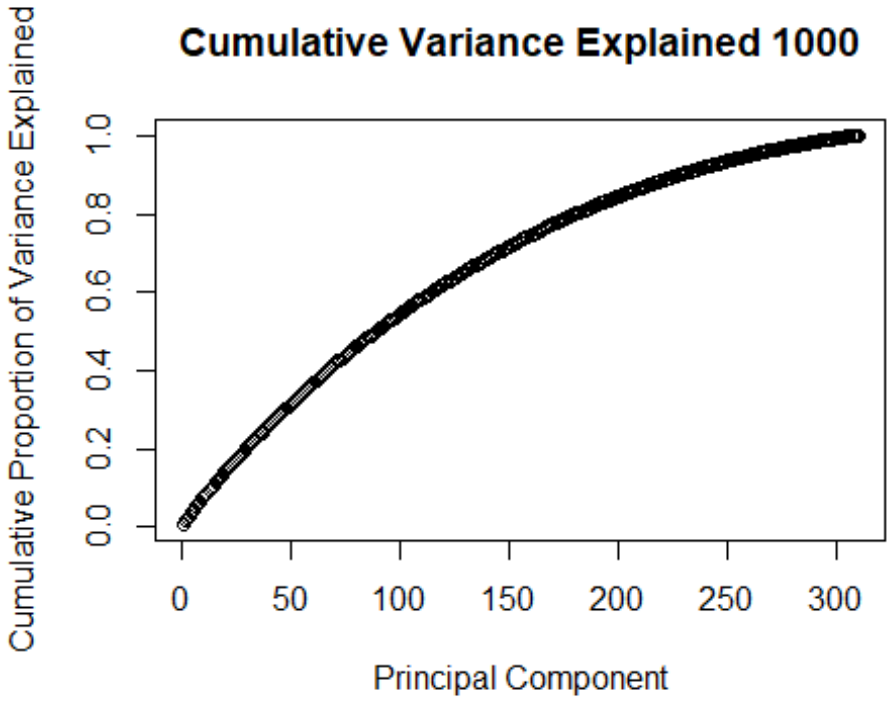



```
[1] "sdev"     "rotation" "center"   "scale"    "x"

                             PC1          PC2          PC3          PC
4
pelvic_incidence       0.26039490 -0.008145069  0.018005619  0.0438685
2
pelvic_tilt            0.15651824 -0.015057780 -0.042867289  0.0175702
4
lumbar_lordosis_angle  0.22794649 -0.006828880  0.005227278  0.0113035
9
sacral_slope           0.21767067  0.000768123  0.055082960  0.0432308
4
pelvic_radius         -0.09339356 -0.020479532  0.028754927 -0.0189513
3
```



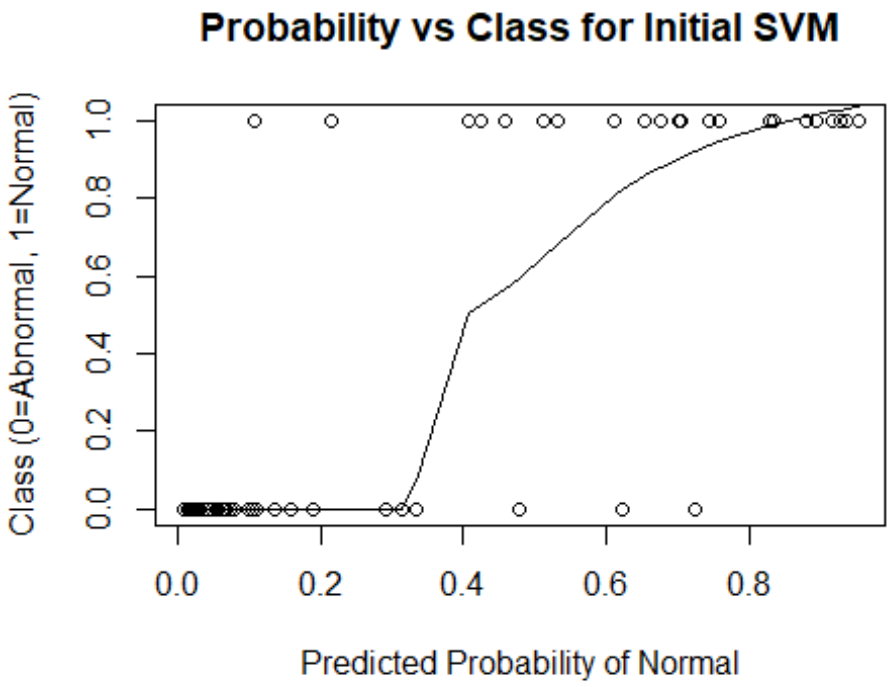

```
 [1] 8.159674 7.822566 7.566269 7.419606 7.416358 7.311467 7.268683
 [8] 7.240749 7.114971 7.056890

 [1] 0.008047015 0.007714562 0.007461803 0.007317166 0.007313962
 [6] 0.007210520 0.007168326 0.007140778 0.007016737 0.006959457
[11] 0.006859799 0.006795145 0.006750020 0.006689229 0.006664719
[16] 0.006590777 0.006537566 0.006509526 0.006471214 0.006419935
```


Variance Explained 1000
Proportion of Variance Explained
0.000
0.004
0.008
0
50
100
150
200
250
300
Principal Component


Cumulative Variance Explained 1000
Cumulative Proportion of Variance Explained
0.0
0.2
0.4
0.6
0.8
1.0
0
50
100
150
200
250
300
Principal Component

Appendix III: Support Vector Machines

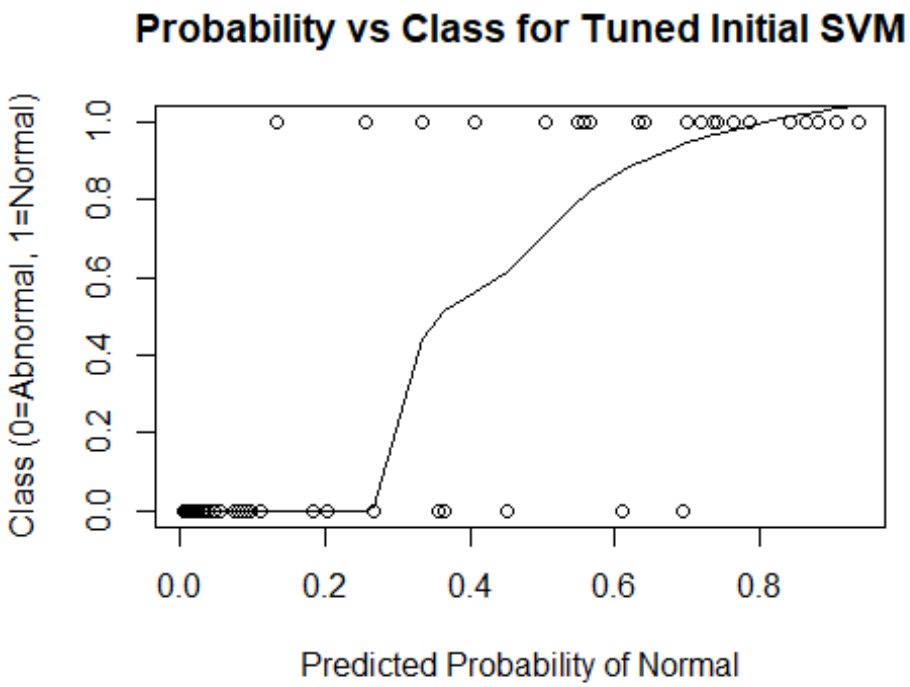
Initial Model




```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction Abnormal Normal
##   Abnormal       38      5
##   Normal          2     17
##
##                Accuracy : 0.8871
##                  95% CI : (0.7811, 0.9534)
##     No Information Rate : 0.6452
##     P-Value [Acc > NIR] : 1.515e-05
##
##                   Kappa : 0.7456
##  Mcnemar's Test P-Value : 0.4497
##
##             Sensitivity : 0.9500
##             Specificity : 0.7727
```

```
##          Pos Pred Value : 0.8837
##          Neg Pred Value : 0.8947
##              Prevalence : 0.6452
##          Detection Rate : 0.6129
##    Detection Prevalence : 0.6935
##       Balanced Accuracy : 0.8614
##
##        'Positive' Class : Abnormal
##

##
## Parameter tuning of 'svm':
##
## - sampling method: 10-fold cross validation
##
## - best parameters:
##  gamma cost
##    0.1    1
##
## - best performance: 0.1323333
##
## - Detailed performance results:
##    gamma  cost     error dispersion
## 1  1e-05 1e-03 0.3148333 0.06779804
## 2  1e-04 1e-03 0.3148333 0.06779804
## 3  1e-03 1e-03 0.3148333 0.06779804
## 4  1e-02 1e-03 0.3148333 0.06779804
## 5  1e-01 1e-03 0.3148333 0.06779804
## 6  1e-05 1e-02 0.3148333 0.06779804
## 7  1e-04 1e-02 0.3148333 0.06779804
## 8  1e-03 1e-02 0.3148333 0.06779804
## 9  1e-02 1e-02 0.3148333 0.06779804
## 10 1e-01 1e-02 0.3148333 0.06779804
## 11 1e-05 1e-01 0.3148333 0.06779804
## 12 1e-04 1e-01 0.3148333 0.06779804
## 13 1e-03 1e-01 0.3148333 0.06779804
## 14 1e-02 1e-01 0.3148333 0.06779804
## 15 1e-01 1e-01 0.2708333 0.07764771
## 16 1e-05 1e+00 0.3148333 0.06779804
## 17 1e-04 1e+00 0.3148333 0.06779804
## 18 1e-03 1e+00 0.3148333 0.06779804
## 19 1e-02 1e+00 0.2266667 0.13152712
## 20 1e-01 1e+00 0.1323333 0.05612596
## 21 1e-05 1e+01 0.3148333 0.06779804
## 22 1e-04 1e+01 0.3148333 0.06779804
## 23 1e-03 1e+01 0.2265000 0.12364438
```

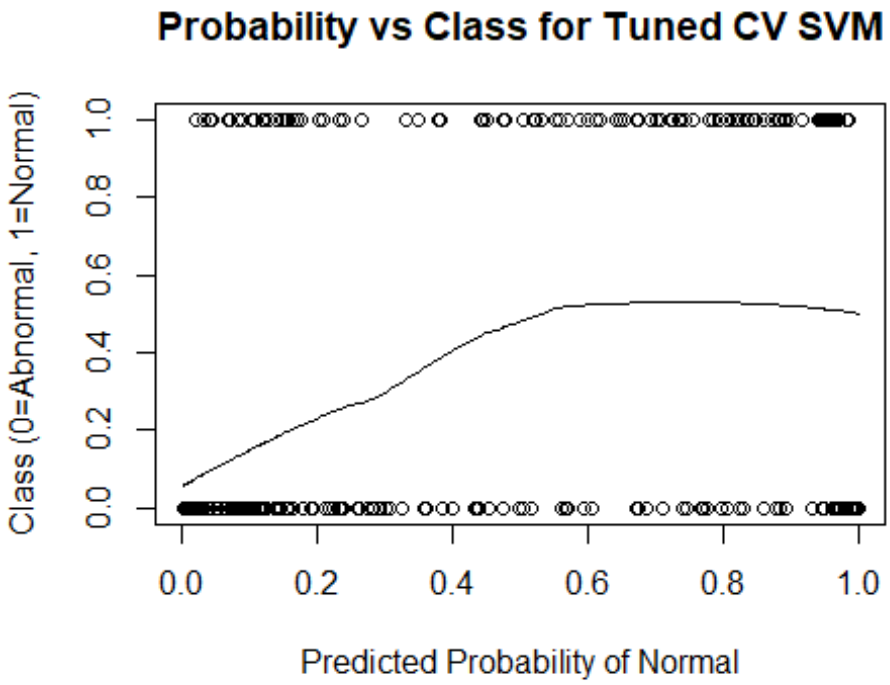
```
## 24 1e-02 1e+01 0.1405000 0.06538608
## 25 1e-01 1e+01 0.1363333 0.06254282

## Confusion Matrix and Statistics
##
##           Reference
## Prediction Abnormal Normal
##   Abnormal       38      4
##   Normal          2     18
##
##                Accuracy : 0.9032
##                  95% CI : (0.8012, 0.9637)
##     No Information Rate : 0.6452
##     P-Value [Acc > NIR] : 3.301e-06
##
##                   Kappa : 0.7842
##  Mcnemar's Test P-Value : 0.6831
##
##             Sensitivity : 0.9500
##             Specificity : 0.8182
##          Pos Pred Value : 0.9048
##          Neg Pred Value : 0.9000
##              Prevalence : 0.6452
##          Detection Rate : 0.6129
##    Detection Prevalence : 0.6774
##       Balanced Accuracy : 0.8841
##
##        'Positive' Class : Abnormal
##
```



Initial Model with Cross-Validation

```
## The following objects are masked from data:
##
##     cervical_tilt, class, classification,
##     degree_spondylolisthesis, Direct_tilt, lumbar_lordosis_angle,
##     pelvic_incidence, pelvic_radius, pelvic_slope, pelvic_tilt,
##     sacral_slope, sacrum_angle, scoliosis_slope, thoracic_slope

##    gamma cost
## 25   0.1   10

## Confusion Matrix and Statistics
##
##           Reference
## Prediction Abnormal Normal
##   Abnormal      186     23
##   Normal         24     77
##
##                Accuracy : 0.8484
##                  95% CI : (0.8035, 0.8864)
##     No Information Rate : 0.6774
##     P-Value [Acc > NIR] : 5.083e-12
##
```

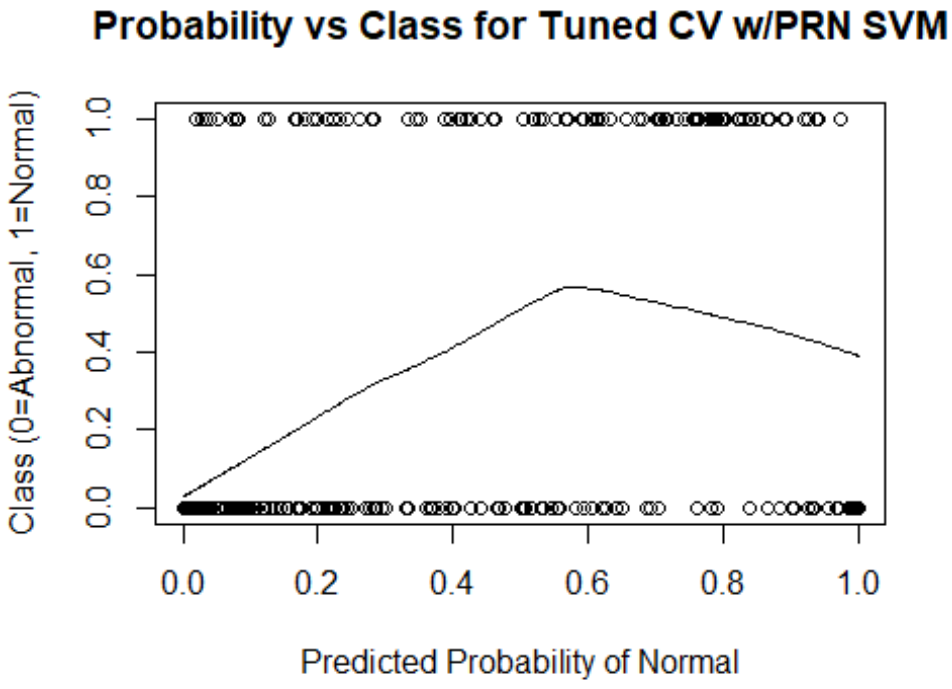
```
##                   Kappa : 0.654
##  Mcnemar's Test P-Value : 1
##
##             Sensitivity : 0.8857
##             Specificity : 0.7700
##          Pos Pred Value : 0.8900
##          Neg Pred Value : 0.7624
##              Prevalence : 0.6774
##          Detection Rate : 0.6000
##    Detection Prevalence : 0.6742
##       Balanced Accuracy : 0.8279
##
##        'Positive' Class : Abnormal
##
```



Add Provided Random Noise

```
##    gamma cost
## 24  0.01   10

## Confusion Matrix and Statistics
##
##           Reference
## Prediction Abnormal Normal
```

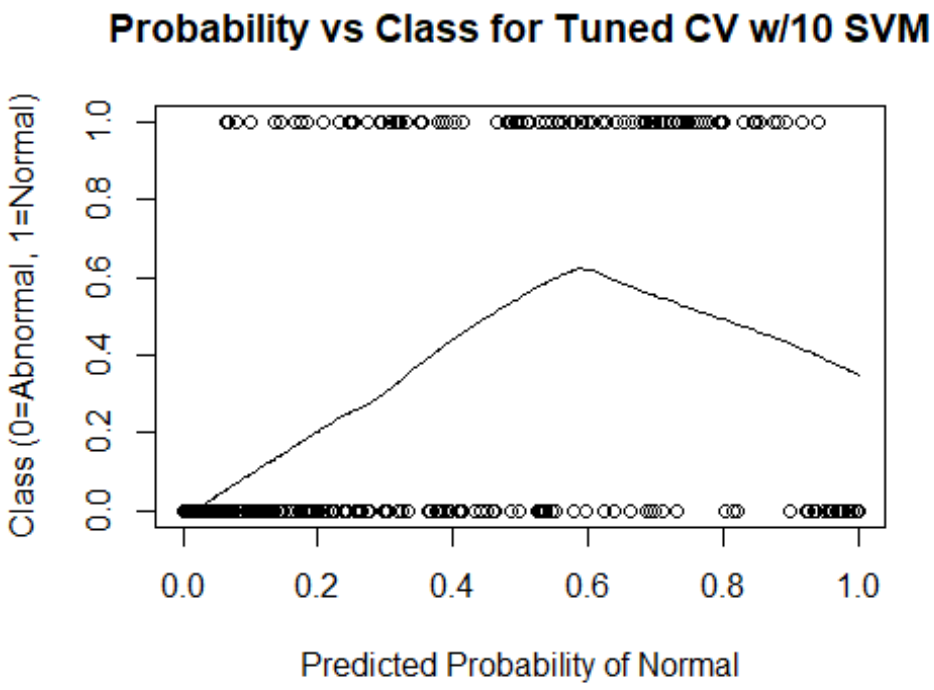
```
##   Abnormal      184     27
##   Normal         26     73
##
##                Accuracy : 0.829
##                  95% CI : (0.7824, 0.8692)
##     No Information Rate : 0.6774
##     P-Value [Acc > NIR] : 1.229e-09
##
##                   Kappa : 0.6078
##  Mcnemar's Test P-Value : 1
##
##             Sensitivity : 0.8762
##             Specificity : 0.7300
##          Pos Pred Value : 0.8720
##          Neg Pred Value : 0.7374
##              Prevalence : 0.6774
##          Detection Rate : 0.5935
##    Detection Prevalence : 0.6806
##       Balanced Accuracy : 0.8031
##
##        'Positive' Class : Abnormal
##
```



Add 10 Random Variables

```
##    gamma cost
## 24  0.01   10
```

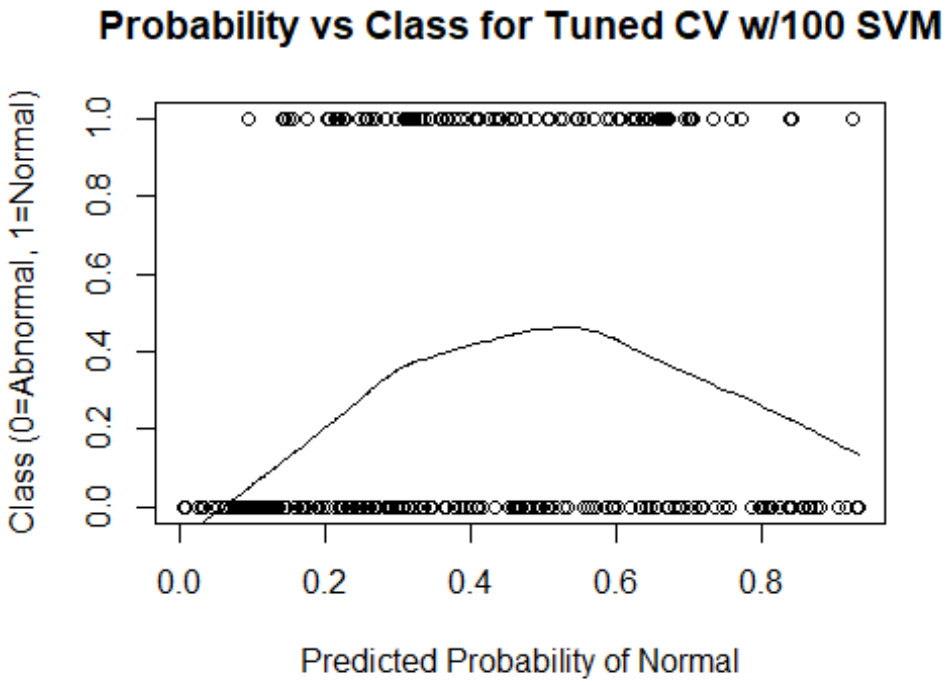
```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction Abnormal Normal
##   Abnormal      181     29
##   Normal         29     71
##
##                Accuracy : 0.8129
##                  95% CI : (0.765, 0.8548)
##     No Information Rate : 0.6774
##     P-Value [Acc > NIR] : 6.441e-08
##
##                   Kappa : 0.5719
##  Mcnemar's Test P-Value : 1
##
##             Sensitivity : 0.8619
##             Specificity : 0.7100
##          Pos Pred Value : 0.8619
##          Neg Pred Value : 0.7100
##              Prevalence : 0.6774
##          Detection Rate : 0.5839
##    Detection Prevalence : 0.6774
##       Balanced Accuracy : 0.7860
##
##        'Positive' Class : Abnormal
##
```



Add 100 Random Variables

```
##    gamma cost
## 23 0.001   10

## Confusion Matrix and Statistics
##
##           Reference
## Prediction Abnormal Normal
##   Abnormal      169     55
##   Normal         41     45
##
##                Accuracy : 0.6903
##                  95% CI : (0.6356, 0.7414)
##     No Information Rate : 0.6774
##     P-Value [Acc > NIR] : 0.3375
##
##                   Kappa : 0.2645
##  Mcnemar's Test P-Value : 0.1846
##
##             Sensitivity : 0.8048
##             Specificity : 0.4500
##          Pos Pred Value : 0.7545
```

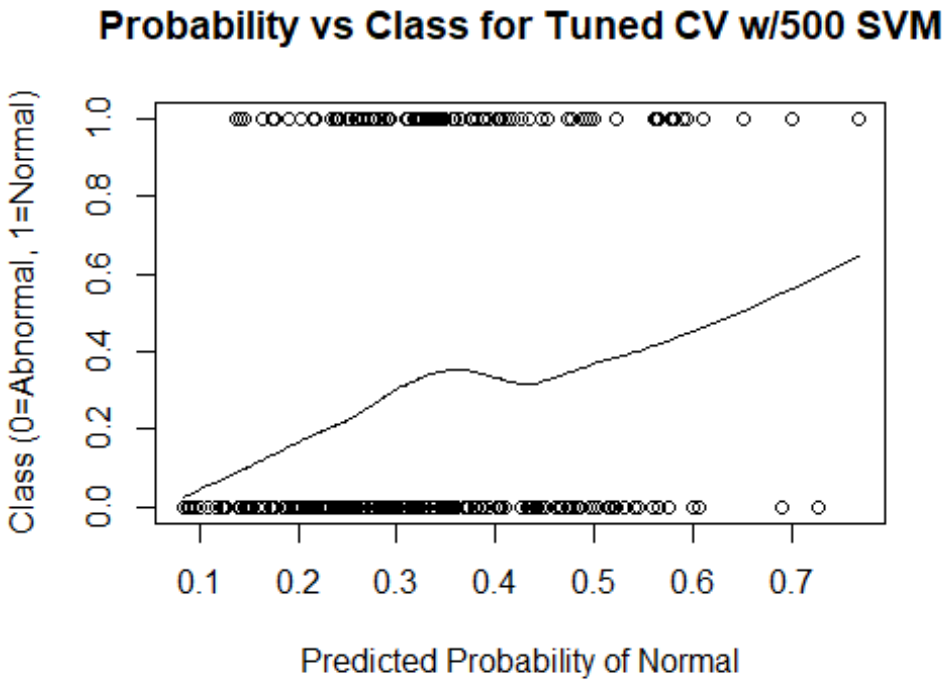
```
##          Neg Pred Value : 0.5233
##              Prevalence : 0.6774
##          Detection Rate : 0.5452
##    Detection Prevalence : 0.7226
##       Balanced Accuracy : 0.6274
##
##        'Positive' Class : Abnormal
##
```



Add 500 Random Variables

```
##   gamma  cost
## 1 1e-05 0.001

## Confusion Matrix and Statistics
##
##           Reference
## Prediction Abnormal Normal
##   Abnormal      210    100
##   Normal          0      0
##
##                Accuracy : 0.6774
##                  95% CI : (0.6223, 0.7292)
##     No Information Rate : 0.6774
```

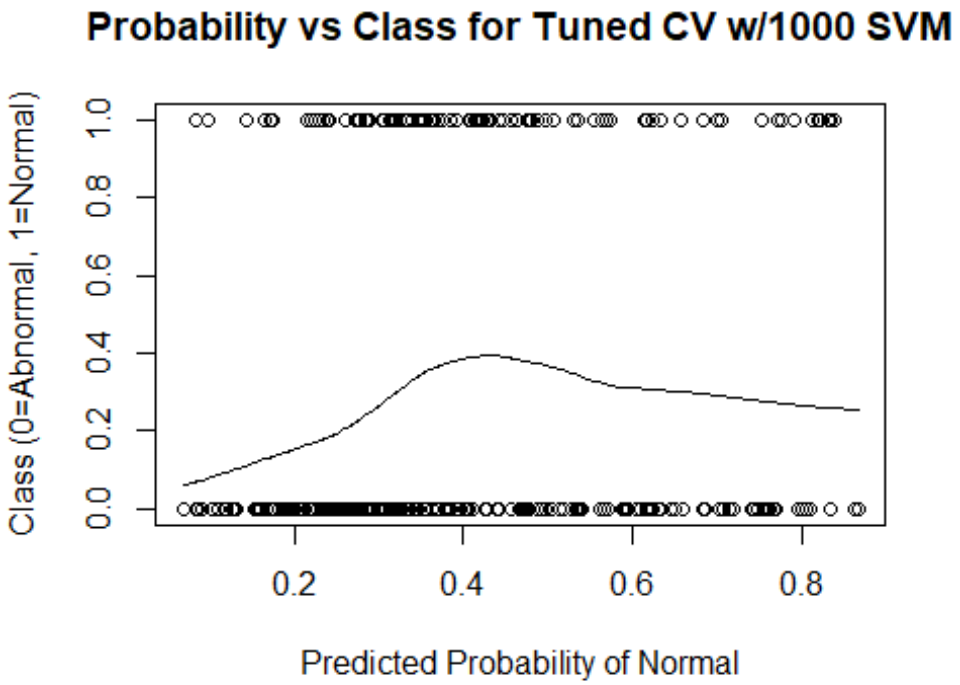
```
##     P-Value [Acc > NIR] : 0.5271
##
##                   Kappa : 0
##  Mcnemar's Test P-Value : <2e-16
##
##             Sensitivity : 1.0000
##             Specificity : 0.0000
##          Pos Pred Value : 0.6774
##          Neg Pred Value :    NaN
##              Prevalence : 0.6774
##          Detection Rate : 0.6774
##    Detection Prevalence : 1.0000
##       Balanced Accuracy : 0.5000
##
##        'Positive' Class : Abnormal
##
```



Add 1000 Random Variables

```
##    gamma cost
## 23 0.001   10

## Confusion Matrix and Statistics
##
```

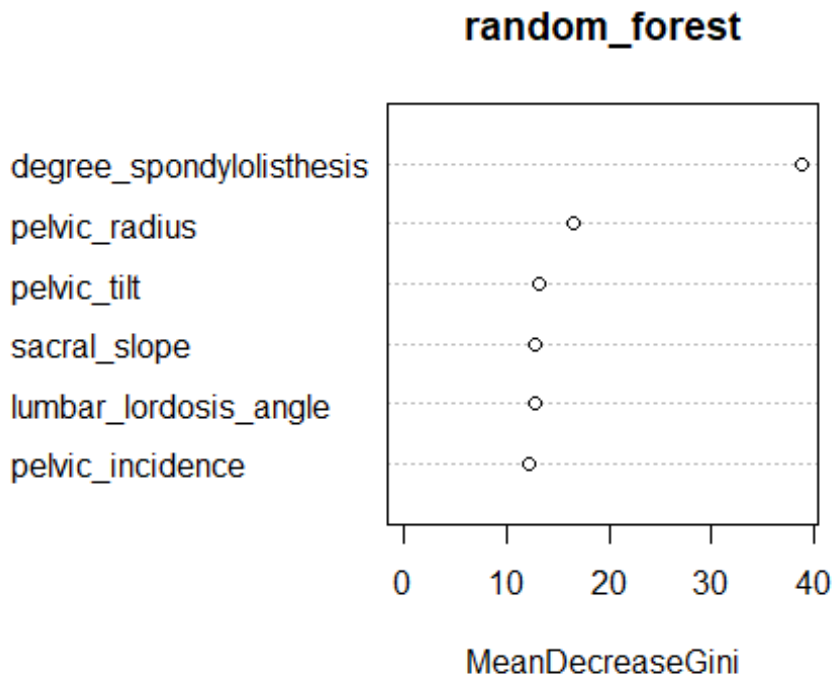
```
##           Reference
## Prediction Abnormal Normal
##   Abnormal      209     95
##   Normal          1      5
##
##                Accuracy : 0.6903
##                  95% CI : (0.6356, 0.7414)
##     No Information Rate : 0.6774
##     P-Value [Acc > NIR] : 0.3375
##
##                   Kappa : 0.06
##  Mcnemar's Test P-Value : <2e-16
##
##             Sensitivity : 0.9952
##             Specificity : 0.0500
##          Pos Pred Value : 0.6875
##          Neg Pred Value : 0.8333
##              Prevalence : 0.6774
##          Detection Rate : 0.6742
##    Detection Prevalence : 0.9806
##       Balanced Accuracy : 0.5226
##
##        'Positive' Class : Abnormal
##
```



Appendix IV: Random Forests

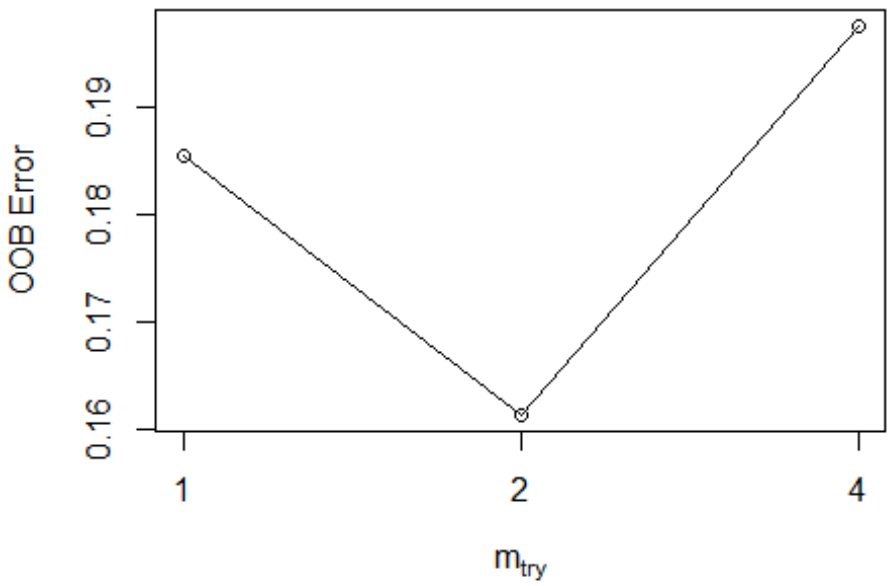
Initial Model

```
                          MeanDecreaseGini
pelvic_tilt                      13.15211
pelvic_incidence                 12.14187
lumbar_lordosis_angle            12.68480
sacral_slope                     12.80926
pelvic_radius                    16.58574
degree_spondylolisthesis         38.84536
```

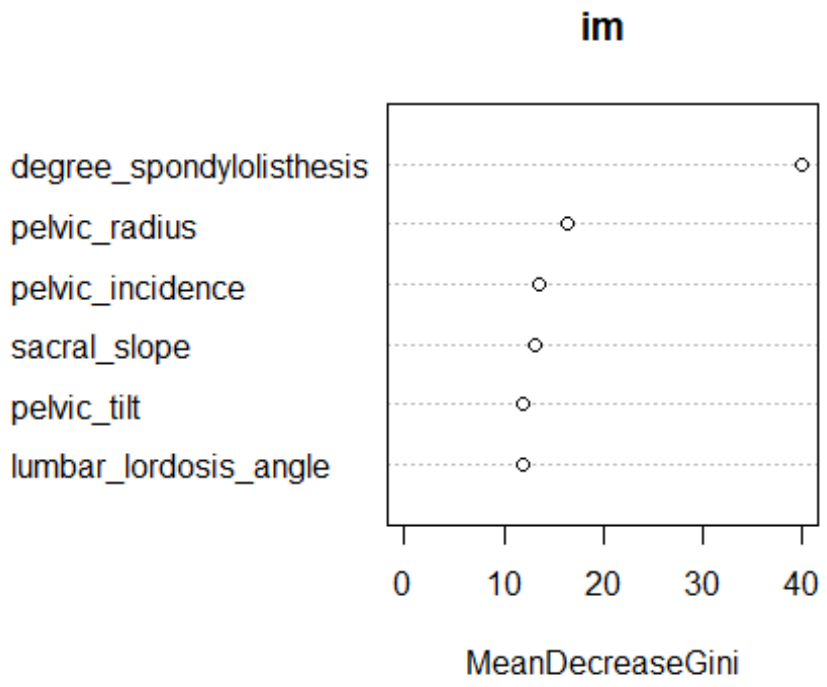



```
Confusion Matrix and Statistics

          Reference
Prediction Abnormal Normal
  Abnormal       38      2
  Normal         10     12

               Accuracy : 0.8065
                 95% CI : (0.6863, 0.8958)
    No Information Rate : 0.7742
```

```
    P-Value [Acc > NIR] : 0.33265

                  Kappa : 0.5396
 Mcnemar's Test P-Value : 0.04331

            Sensitivity : 0.7917
            Specificity : 0.8571
         Pos Pred Value : 0.9500
         Neg Pred Value : 0.5455
             Prevalence : 0.7742
         Detection Rate : 0.6129
   Detection Prevalence : 0.6452
      Balanced Accuracy : 0.8244

       'Positive' Class : Abnormal


mtry = 2  OOB error = 16.13%
Searching left ...
mtry = 1    OOB error = 18.55%
-0.15 0.05
Searching right ...
mtry = 4    OOB error = 19.76%
-0.225 0.05
```

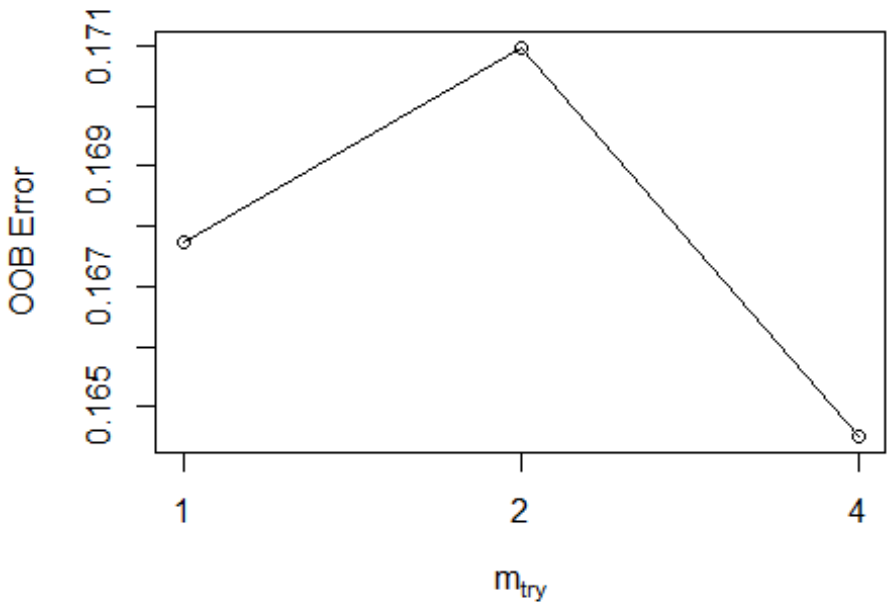


```
Call:
 randomForest(x = x, y = y, mtry = res[which.min(res[, 2]), 1])
               Type of random forest: classification
                     Number of trees: 500
No. of variables tried at each split: 2

        OOB estimate of  error rate: 16.94%
Confusion matrix:
         Abnormal Normal class.error
Abnormal      152     18   0.1058824
Normal         24     54   0.3076923

                           MeanDecreaseGini
pelvic_incidence                  13.60084
pelvic_tilt                       11.98641
lumbar_lordosis_angle             11.80908
sacral_slope                      13.10558
pelvic_radius                     16.39389
degree_spondylolisthesis          39.96065
```

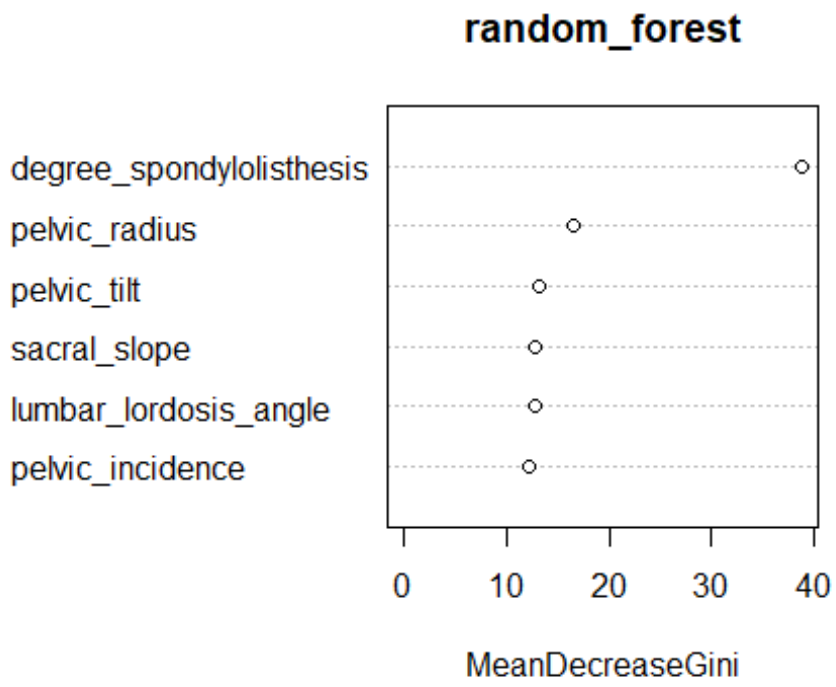


Cross-Validation Model

```
mtry = 2  OOB error = 17.1%
Searching left ...
mtry = 1    OOB error = 16.77%
0.01886792 0.05
Searching right ...
mtry = 4    OOB error = 16.45%
0.03773585 0.05
```

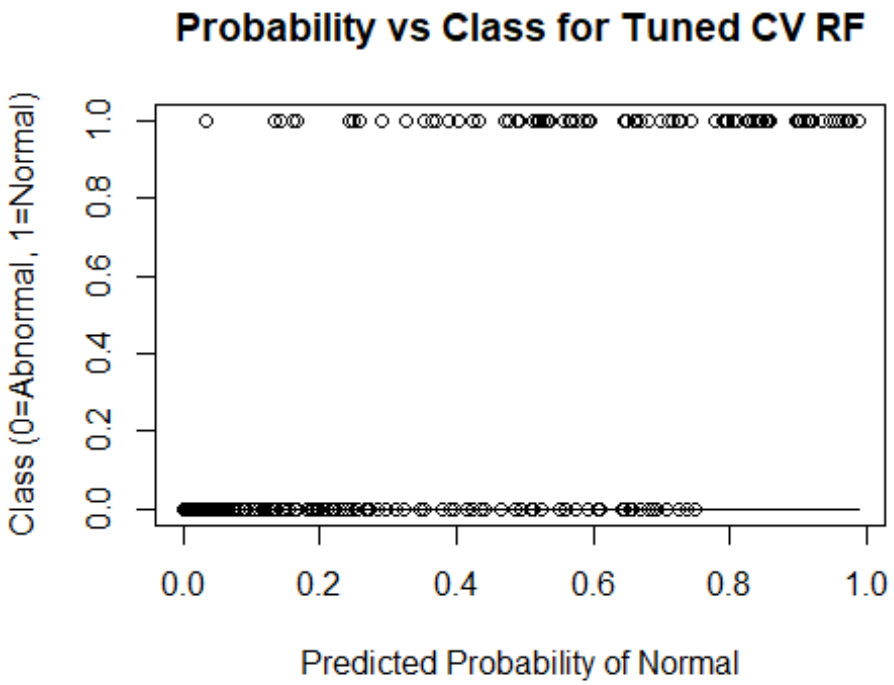



```
      mtry  OOBError
1.OOB    1 0.1677419
2.OOB    2 0.1709677
4.OOB    4 0.1645161

                         MeanDecreaseGini
pelvic_tilt                       13.15211
pelvic_incidence                  12.14187
lumbar_lordosis_angle             12.68480
sacral_slope                      12.80926
pelvic_radius                     16.58574
degree_spondylolisthesis          38.84536
```



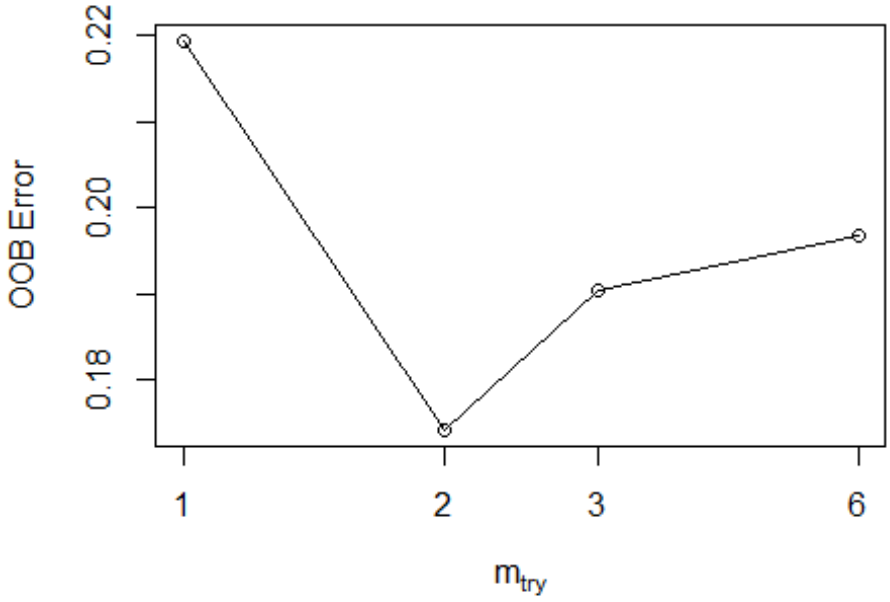

```
Confusion Matrix and Statistics

          Reference
Prediction Abnormal Normal
  Abnormal      187     23
  Normal         26     74

               Accuracy : 0.8419
                 95% CI : (0.7965, 0.8807)
    No Information Rate : 0.6871
    P-Value [Acc > NIR] : 3.261e-10

                  Kappa : 0.6355
 Mcnemar's Test P-Value : 0.7751

            Sensitivity : 0.8779
            Specificity : 0.7629
         Pos Pred Value : 0.8905
         Neg Pred Value : 0.7400
             Prevalence : 0.6871
         Detection Rate : 0.6032
   Detection Prevalence : 0.6774
      Balanced Accuracy : 0.8204
```

```
       'Positive' Class : Abnormal
```

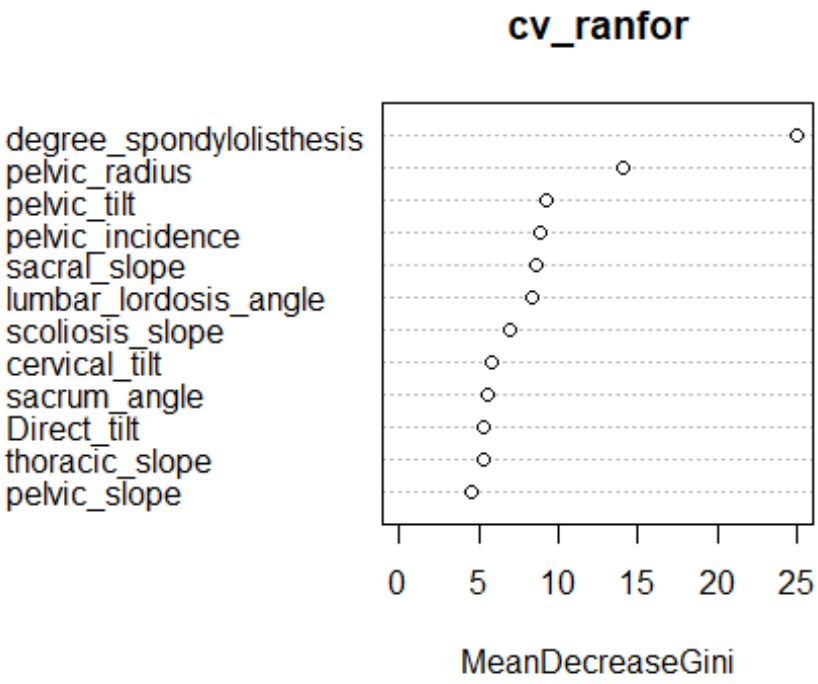


Add Provided Random Noise

```
mtry = 3  OOB error = 19.03%
Searching left ...
mtry = 2    OOB error = 17.42%
0.08474576 0.05
mtry = 1    OOB error = 21.94%
-0.2592593 0.05
Searching right ...
mtry = 6    OOB error = 19.68%
-0.1296296 0.05
```

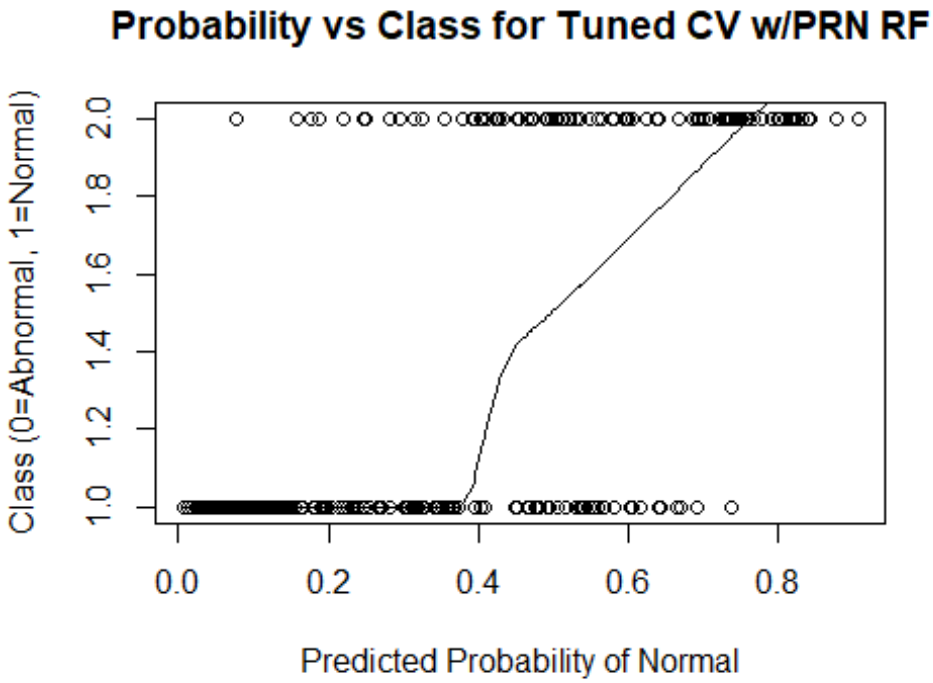



```
      mtry  OOBError
1.OOB    1 0.2193548
2.OOB    2 0.1741935
3.OOB    3 0.1903226
6.OOB    6 0.1967742

                         MeanDecreaseGini
pelvic_incidence                 8.859955
pelvic_tilt                      9.250360
lumbar_lordosis_angle            8.412710
sacral_slope                     8.561810
pelvic_radius                   14.124541
degree_spondylolisthesis        25.057467
pelvic_slope                     4.515013
Direct_tilt                      5.361990
thoracic_slope                   5.314960
cervical_tilt                    5.856255
sacrum_angle                     5.578070
scoliosis_slope                  6.937531
```



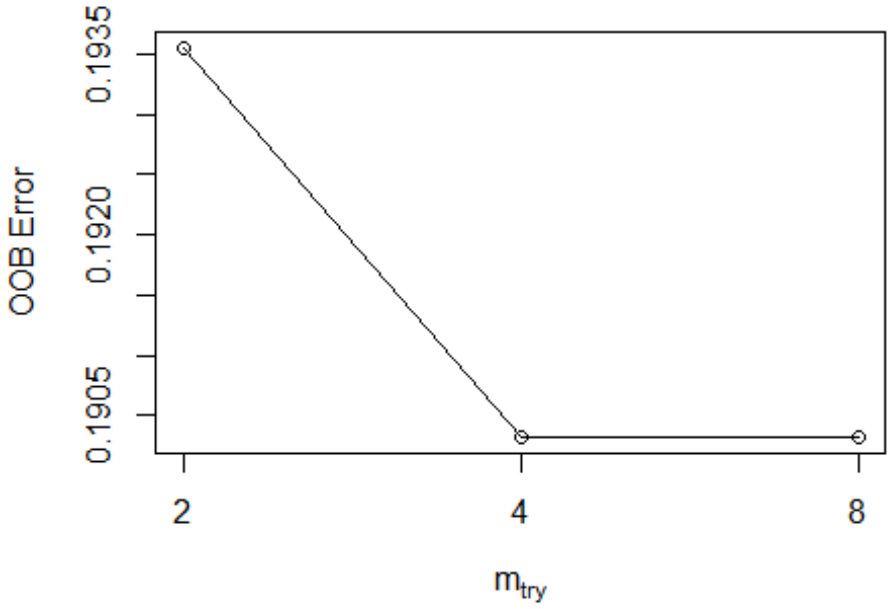

```
Confusion Matrix and Statistics

          Reference
Prediction Abnormal Normal
  Abnormal      188     22
  Normal         35     65

               Accuracy : 0.8161
                 95% CI : (0.7684, 0.8577)
    No Information Rate : 0.7194
    P-Value [Acc > NIR] : 5.282e-05

                  Kappa : 0.5645
 Mcnemar's Test P-Value : 0.112

            Sensitivity : 0.8430
            Specificity : 0.7471
         Pos Pred Value : 0.8952
         Neg Pred Value : 0.6500
             Prevalence : 0.7194
         Detection Rate : 0.6065
   Detection Prevalence : 0.6774
      Balanced Accuracy : 0.7951
```

```
       'Positive' Class : Abnormal
```



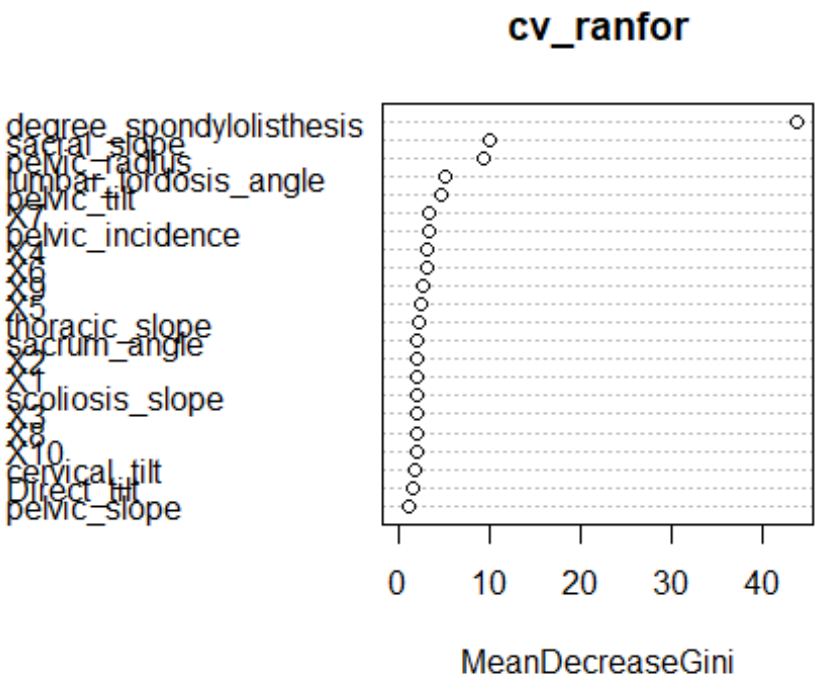
Add 10 Random Variables

```
mtry = 4  OOB error = 19.03%
Searching left ...
mtry = 2    OOB error = 19.35%
-0.01694915 0.05
Searching right ...
mtry = 8    OOB error = 19.03%
0 0.05
```



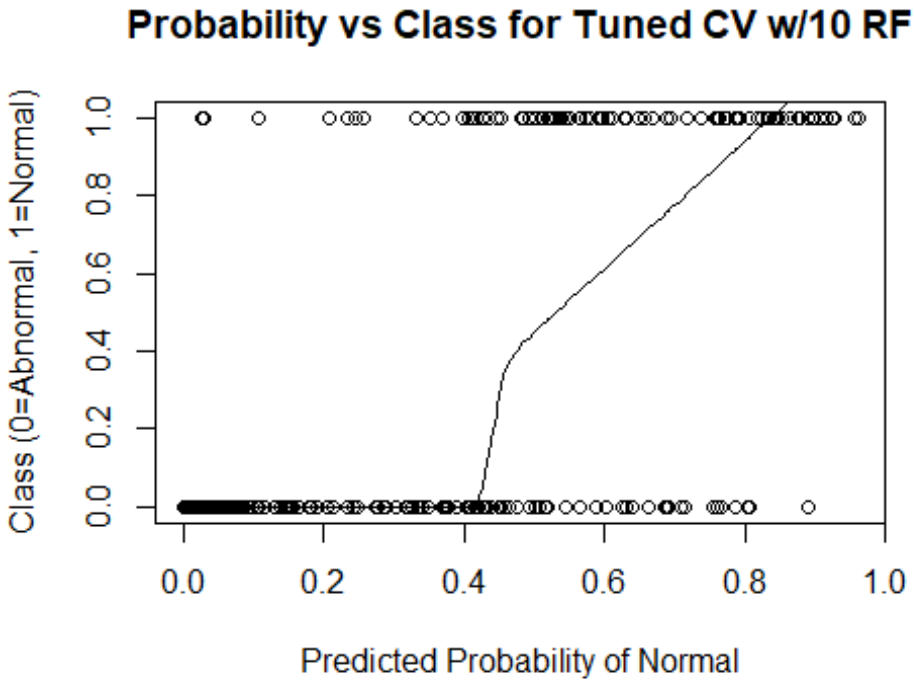

```
      mtry  OOBError
2.OOB    2 0.1935484
4.OOB    4 0.1903226
8.OOB    8 0.1903226

                         MeanDecreaseGini
pelvic_incidence                3.2022659
pelvic_tilt                     4.5670598
lumbar_lordosis_angle           5.0248268
sacral_slope                    9.9702033
pelvic_radius                   9.1985254
degree_spondylolisthesis       43.9117143
pelvic_slope                    0.9751303
Direct_tilt                     1.4061935
thoracic_slope                  2.0743864
cervical_tilt                   1.8274781
sacrum_angle                    1.9724421
scoliosis_slope                 1.8839118
X1                              1.9651829
X2                              1.9712072
X3                              1.8640025
X4                              3.1620878
X5                              2.3345345
X6                              3.1114970
X7                              3.3487048
```

```
X8                                1.8452842
X9                                2.6035675
X10                               1.8380684
```



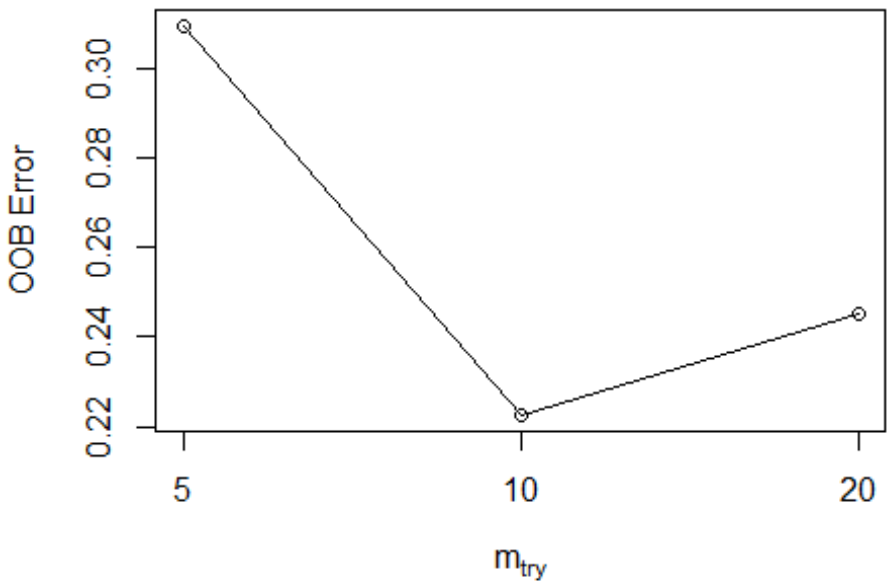

```
Confusion Matrix and Statistics

          Reference
Prediction Abnormal Normal
  Abnormal      185     25
  Normal         25     75

               Accuracy : 0.8387
                 95% CI : (0.7929, 0.8779)
    No Information Rate : 0.6774
    P-Value [Acc > NIR] : 8.779e-11

                  Kappa : 0.631
 Mcnemar's Test P-Value : 1

            Sensitivity : 0.8810
            Specificity : 0.7500
         Pos Pred Value : 0.8810
         Neg Pred Value : 0.7500
             Prevalence : 0.6774
         Detection Rate : 0.5968
```

```
   Detection Prevalence : 0.6774
      Balanced Accuracy : 0.8155

       'Positive' Class : Abnormal
```



Add 100 Random Variables

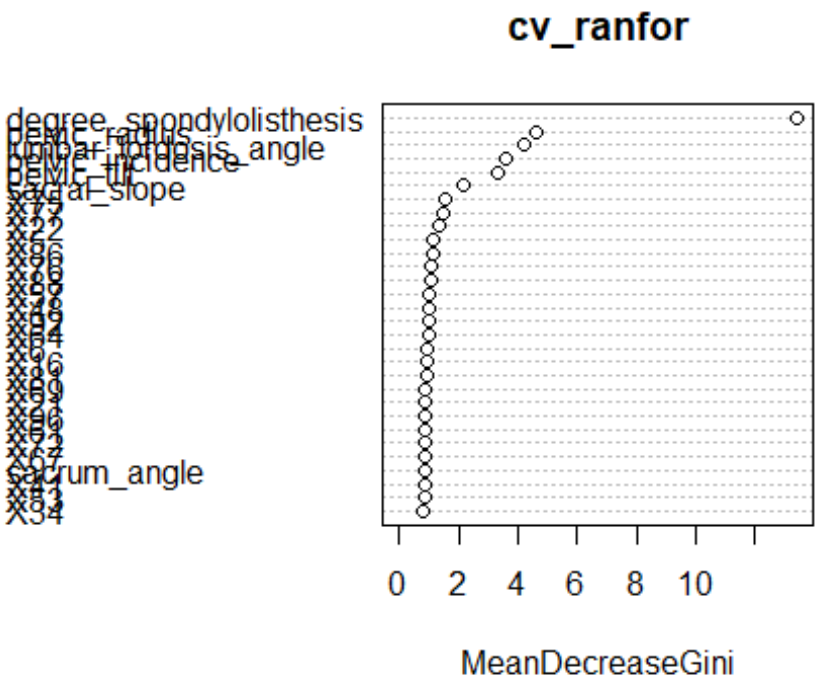
```
mtry = 10  OOB error = 22.26%
Searching left ...
mtry = 5    OOB error = 30.97%
-0.3913043 0.05
Searching right ...
mtry = 20   OOB error = 24.52%
-0.1014493 0.05
```




```
       mtry  OOBError
5.OOB     5 0.3096774
10.OOB   10 0.2225806
20.OOB   20 0.2451613

                         MeanDecreaseGini
pelvic_incidence                3.5641762
pelvic_tilt                     3.3364984
lumbar_lordosis_angle           4.1795320
sacral_slope                    2.1758695
pelvic_radius                   4.6062258
degree_spondylolisthesis       13.4208132
pelvic_slope                    0.5021879
Direct_tilt                     0.4562610
thoracic_slope                  0.6121050
cervical_tilt                   0.6490038
sacrum_angle                    0.8462769
scoliosis_slope                 0.8053812
X1                              0.6778439
X2                              0.7681018
X3                              0.5986865
X4                              0.6395936
X5                              1.1574296
X6                              0.9578584
X7                              0.8195534
```

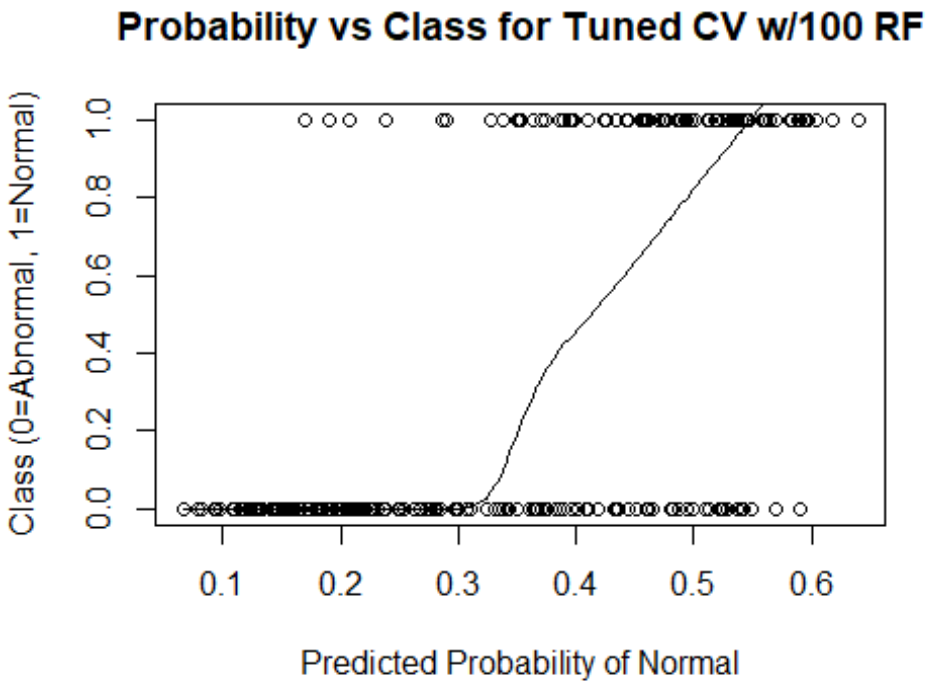
X8	0.5693077
X9	0.8036344
X10	0.5131597
X11	0.6718749
X12	0.8218499
X13	0.6936367
X14	0.8080790
X15	0.7415135
X16	0.9315695
X17	0.8227422
X18	0.7675297
X19	0.6834504
X20	0.5525062
X21	0.8873626
X22	1.3218316
X23	0.4555300
X24	0.7460673
X25	0.8135725
X26	0.8221727
X27	0.8228608
X28	0.6093707
X29	0.8005218
X30	0.4597668
X31	0.4503310
X32	0.5027736
X33	0.4704129
X34	0.8271870
X35	0.6919351
X36	0.4129119
X37	0.4176781
X38	0.4859588
X39	0.6571532
X40	0.5607707
X41	0.8422408
X42	0.6237343
X43	0.5391351
X44	0.6084440
X45	0.5481742
X46	0.6294841
X47	0.5116424
X48	1.0152811
X49	0.4335624
X50	0.4320520
X51	0.6124192
X52	0.6786963
X53	0.8362618

```
X54                                0.7361601
X55                                0.7630174
X56                                0.5427745
X57                                1.0309441
X58                                0.6285753
X59                                0.4628716
```




```
Confusion Matrix and Statistics

          Reference
Prediction Abnormal Normal
  Abnormal      192     18
  Normal         58     42

               Accuracy : 0.7548
                 95% CI : (0.703, 0.8017)
    No Information Rate : 0.8065
    P-Value [Acc > NIR] : 0.9896

                  Kappa : 0.3734
 Mcnemar's Test P-Value : 7.691e-06

            Sensitivity : 0.7680
            Specificity : 0.7000
```

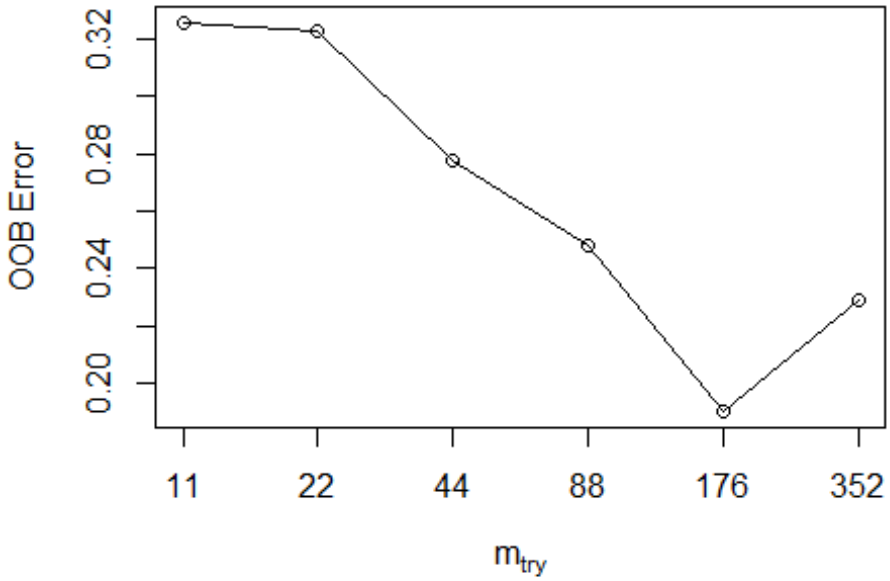
```
         Pos Pred Value : 0.9143
         Neg Pred Value : 0.4200
             Prevalence : 0.8065
         Detection Rate : 0.6194
   Detection Prevalence : 0.6774
      Balanced Accuracy : 0.7340

       'Positive' Class : Abnormal
```



Add 500 Random Variables

```
mtry = 22  OOB error = 32.26%
Searching left ...
mtry = 11   OOB error = 32.58%
-0.01 0.05
Searching right ...
mtry = 44   OOB error = 27.74%
0.14 0.05
mtry = 88   OOB error = 24.84%
0.1046512 0.05
mtry = 176  OOB error = 19.03%
0.2337662 0.05
```

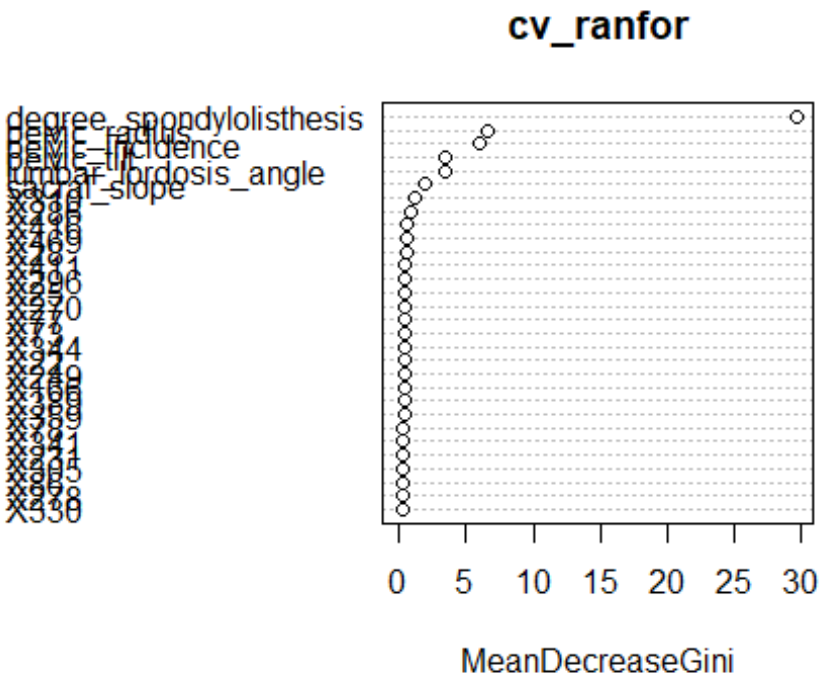
```
mtry = 352  OOB error = 22.9%
-0.2033898 0.05
```




```
        mtry  OOBError
11.OOB    11 0.3258065
22.OOB    22 0.3225806
44.OOB    44 0.2774194
88.OOB    88 0.2483871
176.OOB  176 0.1903226
352.OOB  352 0.2290323

                         MeanDecreaseGini
pelvic_incidence               6.01100462
pelvic_tilt                    3.38621535
lumbar_lordosis_angle          3.38204323
sacral_slope                   1.84095368
pelvic_radius                  6.61375200
degree_spondylolisthesis      29.69157311
pelvic_slope                   0.04516613
Direct_tilt                    0.05921861
thoracic_slope                 0.28618526
cervical_tilt                  0.10897747
sacrum_angle                   0.05223586
scoliosis_slope                0.06663832
X1                             0.02960129
```

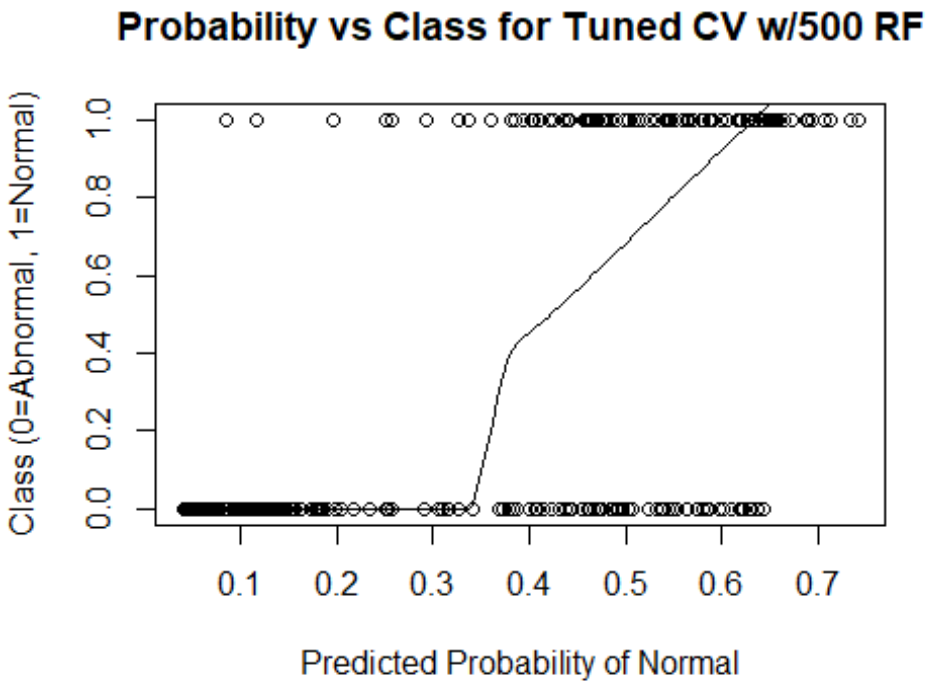
X2	0.07636687
X3	0.16661184
X4	0.04863451
X5	0.10742465
X6	0.26578690
X7	0.17616101
X8	0.08328187
X9	0.19831939
X10	0.04033133
X11	0.11273203
X12	0.06154676
X13	0.12715521
X14	0.21590919
X15	0.01675745
X16	0.07298429
X17	0.09978005
X18	0.07644006
X19	0.04767568
X20	0.01928116
X21	0.22156802
X22	0.36132039
X23	0.03678497
X24	0.07818254
X25	0.40941571
X26	0.19638778
X27	0.09877176
X28	0.55740077
X29	0.04900979
X30	0.09326776
X31	0.05102179
X32	0.06179487
X33	0.10394346
X34	0.19932217
X35	0.04293341
X36	0.05409918
X37	0.08539927
X38	0.11589262
X39	0.03693409
X40	0.07416488
X41	0.17868996
X42	0.05328176
X43	0.04976898
X44	0.05743946
X45	0.04320375
X46	0.06791816
X47	0.04224700

```
X48                                    0.22214187
X49                                    0.10783932
X50                                    0.06129321
X51                                    0.11258447
X52                                    0.07751939
X53                                    0.07971532
X54                                    0.09749348
X55                                    0.05284229
X56                                    0.06923572
```



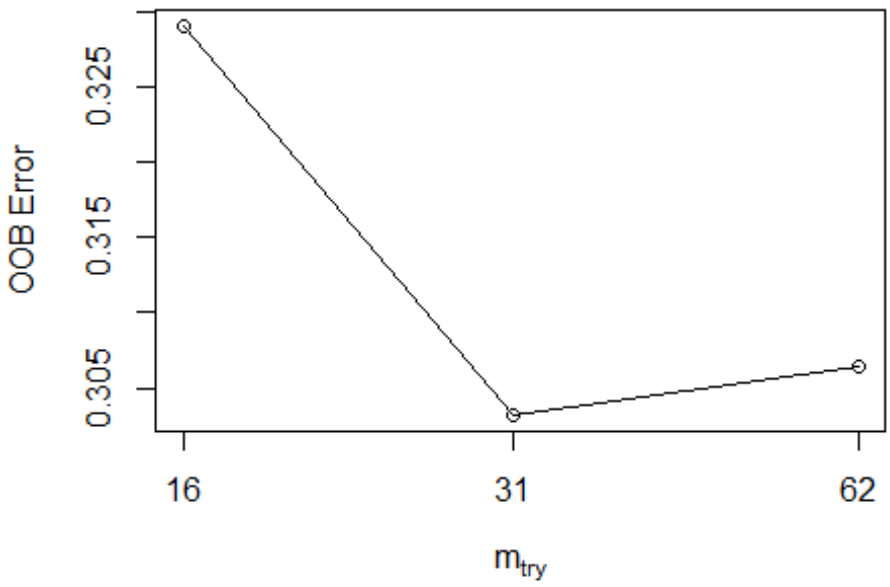

```
Confusion Matrix and Statistics

          Reference
Prediction Abnormal Normal
  Abnormal      182     28
  Normal         37     63

               Accuracy : 0.7903
                 95% CI : (0.7407, 0.8343)
    No Information Rate : 0.7065
    P-Value [Acc > NIR] : 0.0005294

                  Kappa : 0.5087
 Mcnemar's Test P-Value : 0.3210620
```

```
            Sensitivity : 0.8311
            Specificity : 0.6923
         Pos Pred Value : 0.8667
         Neg Pred Value : 0.6300
             Prevalence : 0.7065
         Detection Rate : 0.5871
   Detection Prevalence : 0.6774
      Balanced Accuracy : 0.7617

       'Positive' Class : Abnormal
```



Add 1000 Random Variables

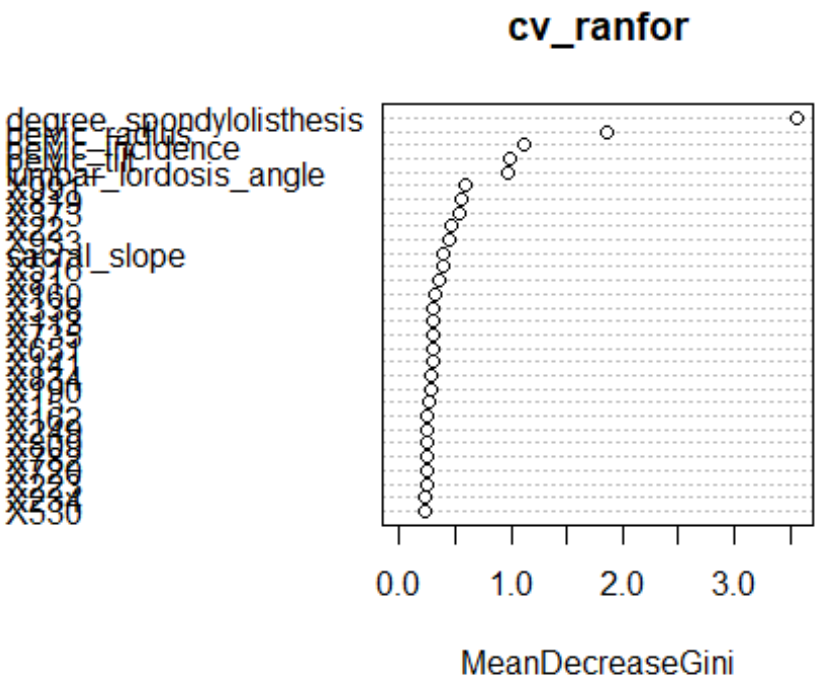
```
mtry = 31  OOB error = 30.32%
Searching left ...
mtry = 16   OOB error = 32.9%
-0.08510638 0.05
Searching right ...
mtry = 62   OOB error = 30.65%
-0.0106383 0.05
```




```
       mtry  OOBError
16.OOB   16 0.3290323
31.OOB   31 0.3032258
62.OOB   62 0.3064516

                          MeanDecreaseGini
pelvic_incidence                1.115865888
pelvic_tilt                     0.981450644
lumbar_lordosis_angle           0.975049651
sacral_slope                    0.395121701
pelvic_radius                   1.849761856
degree_spondylolisthesis        3.565151873
pelvic_slope                    0.043054130
Direct_tilt                     0.112771258
thoracic_slope                  0.036546526
cervical_tilt                   0.048519792
sacrum_angle                    0.084515479
scoliosis_slope                 0.098642966
X1                              0.110725066
X2                              0.095204713
X3                              0.084426512
X4                              0.133966330
X5                              0.196215418
X6                              0.066426525
X7                              0.101167765
```

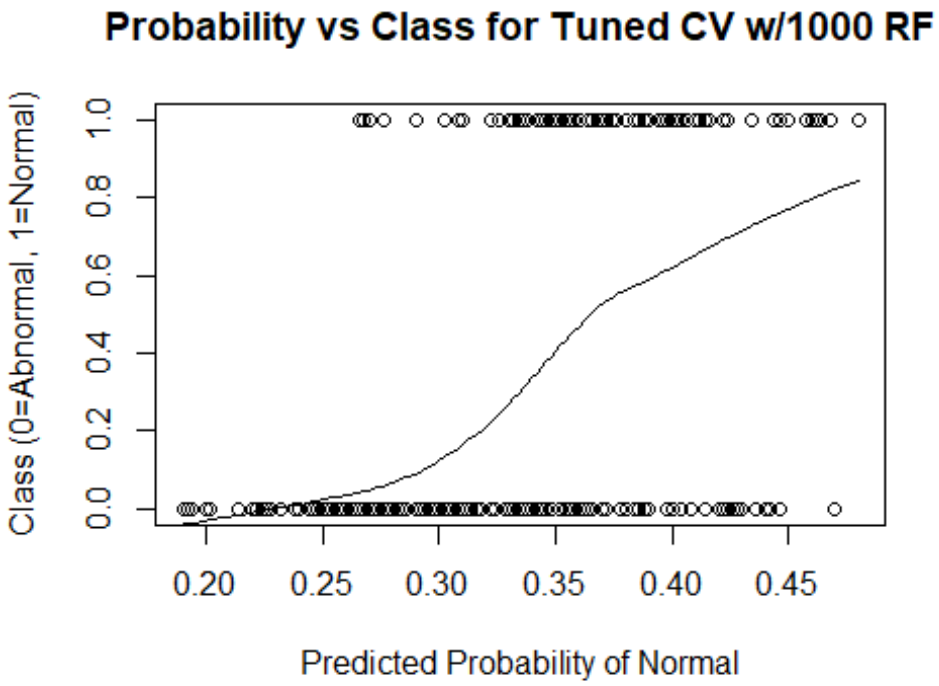
X8	0.102328764
X9	0.086409782
X10	0.060620587
X11	0.119566181
X12	0.138858466
X13	0.045334508
X14	0.071295411
X15	0.259547404
X16	0.065672330
X17	0.069879710
X18	0.112287686
X19	0.088829552
X20	0.066341833
X21	0.081837614
X22	0.473760239
X23	0.070109449
X24	0.050637532
X25	0.086569566
X26	0.117118955
X27	0.106494432
X28	0.140308903
X29	0.062346060
X30	0.080721929
X31	0.062293997
X32	0.102054121
X33	0.043153305
X34	0.130603745
X35	0.095472731
X36	0.118973340
X37	0.093745077
X38	0.075535333
X39	0.037310434
X40	0.078961066
X41	0.072955051
X42	0.139290860
X43	0.024685360
X44	0.112154248
X45	0.110484666
X46	0.087231615
X47	0.027096028
X48	0.161762101
X49	0.051950633
X50	0.092888960
X51	0.068466958
X52	0.103834530
X53	0.083579779

```
X54                                0.113287144
X55                                0.073942322
X56                                0.063352996
X57                                0.230502271
X58                                0.103424052
X59                                0.031666818
```



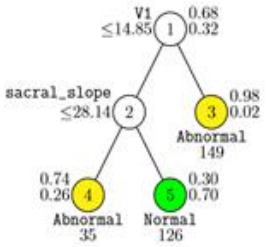

```
Confusion Matrix and Statistics

          Reference
Prediction Abnormal Normal
  Abnormal      210      0
  Normal        100      0

               Accuracy : 0.6774
                 95% CI : (0.6223, 0.7292)
    No Information Rate : 1
    P-Value [Acc > NIR] : 1

                  Kappa : 0
 Mcnemar's Test P-Value : <2e-16

            Sensitivity : 0.6774
            Specificity :     NA
```

```
        Pos Pred Value :     NA
        Neg Pred Value :     NA
            Prevalence : 1.0000
        Detection Rate : 0.6774
  Detection Prevalence : 0.6774
     Balanced Accuracy :     NA

      'Positive' Class : Abnormal
```



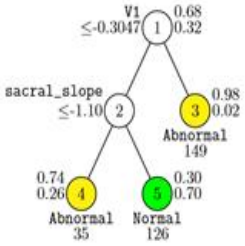
Appendix V: G.U.I.D.E Tree Diagrams

1) Tree for original data



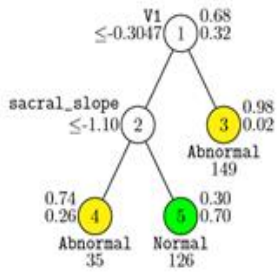
GUIDE v.29.0 0.50-SE pruned classification tree for predicting `classification` using estimated priors and unit misclassification costs. Maximum number of split levels is 10 and minimum node sample size is 2. At each split, an observation goes to the left branch if and only if the condition is satisfied. `V1` = `degree_spondylolisthesis`. Predicted classes and sample sizes printed below terminal nodes; class proportions for `classification` = `Abnormal` and `Normal` beside nodes. Second best split variable at root node is `pelvic_radius`.

2) Tree for data with 10 Random Variables



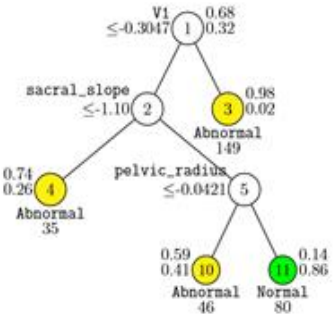
GUIDE v.29.0 0.50-SE pruned classification tree for predicting `classification` using estimated priors and unit misclassification costs. Maximum number of split levels is 10 and minimum node sample size is 2. At each split, an observation goes to the left branch if and only if the condition is satisfied. `V1` = `degree_spondylolisthesis`. Predicted classes and sample sizes printed below terminal nodes; class proportions for `classification` = `Abnormal` and `Normal` beside nodes. Second best split variable at root node is `pelvic_radius`.

3) Tree with 100 Random Variables



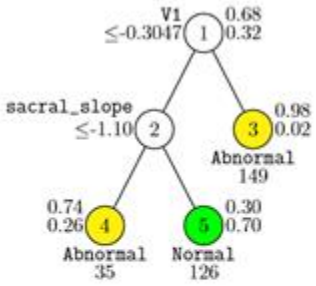
GUIDE v.29.0 0.50-SE pruned classification tree for predicting `classification` using estimated priors and unit misclassification costs. Maximum number of split levels is 10 and minimum node sample size is 2. At each split, an observation goes to the left branch if and only if the condition is satisfied. `V1` = `degree_spondylolisthesis`. Predicted classes and sample sizes printed below terminal nodes; class proportions for `classification` = `Abnormal` and `Normal` beside nodes. Second best split variable at root node is `pelvic_radius`.

4) Tree with 500 Random Variables



GUIDE v.29.0 0.50-SE pruned classification tree for predicting `classification` using estimated priors and unit misclassification costs. Maximum number of split levels is 10 and minimum node sample size is 2. At each split, an observation goes to the left branch if and only if the condition is satisfied. `V1` = `degree_spondylolisthesis`. Predicted classes and sample sizes printed below terminal nodes; class proportions for `classification` = `Abnormal` and `Normal` beside nodes. Second best split variable at root node is `pelvic_radius`.

5) Tree diagram with 1000 Random Variables



GUIDE v.29.0 0.50-SE pruned classification tree for predicting `classification` using estimated priors and unit misclassification costs. Maximum number of split levels is 10 and minimum node sample size is 2. At each split, an observation goes to the left branch if and only if the condition is satisfied. `V1` = `degree_spondylolisthesis`. Predicted classes and sample sizes printed below terminal nodes; class proportions for `classification` = `Abnormal` and `Normal` beside nodes. Second best split variable at root node is `pelvic_radius`.