GE Case Study: Detecting Pneumoconiosis

Name withheld per instructions

April 2018

Contents

List of Tables

List of Figures

1 Introduction and Problem Statement

Pneumoconiosis, also known as “black lung disease”, is an occupational lung disease caused by dusts that are inhaled and deposited deep in the lungs causing damage (American Lung Association, 2018). It is often found in individuals working in the mining or agricultural industries. There is no cure for Pneumoconiosis, so early detection is vital for proper treatment.

The traditional method for detecting Pneumoconiosis requires trained doctors to review chest x-rays for abnormalities that may indicate the disease is present. However, due to the lack of trained personnel and the large number of patients waiting to be screened, a reliable and automated process for early detection is desired.

Figure 1 provides an example image of a patient with signs of Pneumoconiosis (Caruana, 2017). An algorithm has been used to segment the lung x-rays and gather data from the images. The images have been reviewed and categorized by trained professionals. The task in this project is to use the data from the images to develop a reliable and automated process that is able to correctly identify those patients having Pneumoconiosis while minimizing the occurences of false positives.

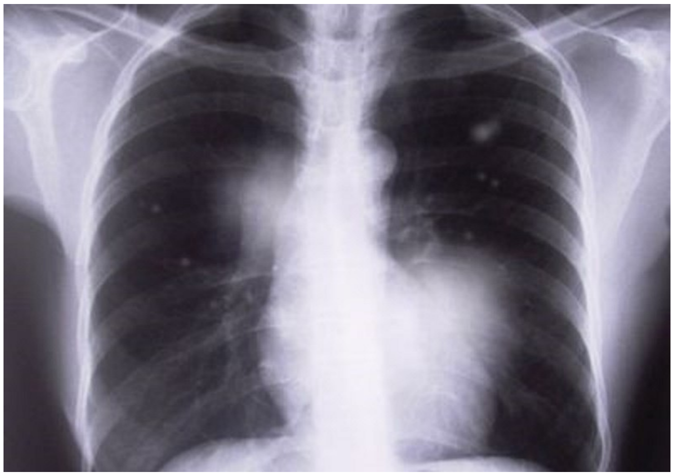


Figure 1: Signs of Pneumoconiosis

2 Data

The data consists of 2,606 observations from 473 patients and is split between six lung segments. The six segments represent the right upper, right middle, right lower, left upper, left middle, and left lower regions of the lung images. The data is provided in an MS Excel spreadsheet, with a separate tab for each segment.

Each segment contains 39 continuous variables and a categorical response variable labeled as ‘0’ for ‘Normal’, those without the disease, and ‘1’ for ‘Abnormal’ when Pneumoconiosis is present. The 39 continuous features fall into two categories, intensity based and co-occurrence matrix based. To obtain the intensity based features, a set of six features computed from the histogram of intensity values – mean, standard deviation, skewness, kurtosis, energy and entropy were extracted. Filters were then applied to the images. Of the 222 features that this method generated, only 34 were supplied for the case study.

Five additional features were extracted for the co-occurence matrix based variables. These features include energy, entropy, local homogeneity, correlation and inertia and are determined by the gray level co-occurrence matrix that was computed for the ROI.

3 Exploratory Data Analysis

Note in **Table 1** that the count of observations by segment differs. This indicates that not all of the 473 unique patients have observations for all lung segments. Summary statistics, including the mean, median, standard deviation, mimimum values, maximum values and any missing data were reviewed. They are not included here due to the size of the table, but are availabe in **Appendix A.1**.

Table 1: Lung Segment Observations

Position	Count
LeftLower	434
LeftMiddle	467
LeftUpper	392
RightLower	446
RightMiddle	470
RightUpper	397
Total	2606

Since the lung images from each patient are broken into six segments and each tab contains the patient number and the label, it is possible that the labels for a patient do not match across the six segments. This is not much of an issue at the segment level, but it is important that it is taken into consideration if the data is rolled up or combined into a single table. If the label for any segement is ‘Abnormal’, then it should be considered that disease is present in the patient, regardless of the value of the other segment labels. Verification was performed to ensure that no patients have inconsistent label values. The categorical variable, *Position*, indicating the which lung segment the observations belong to was created and added to the data set for use in EDA and modeling.

The prevalence rate for Pneumoconiosis in the data set is roughly 36%. The remaining 64% of patients are labeled as ‘Normal’, meaning no disease was detected during the initial analysis by the medical experts.

3.1 Correlation - Predictors

The correlation plot in **Figure 2** shows that some of the predictors are highly correlated with each other. This requires further investigation as high correlation among predictors can cause instability in the model. Generally, when two variables are highly correlated, it is best to remove one since the second is not providing new information.

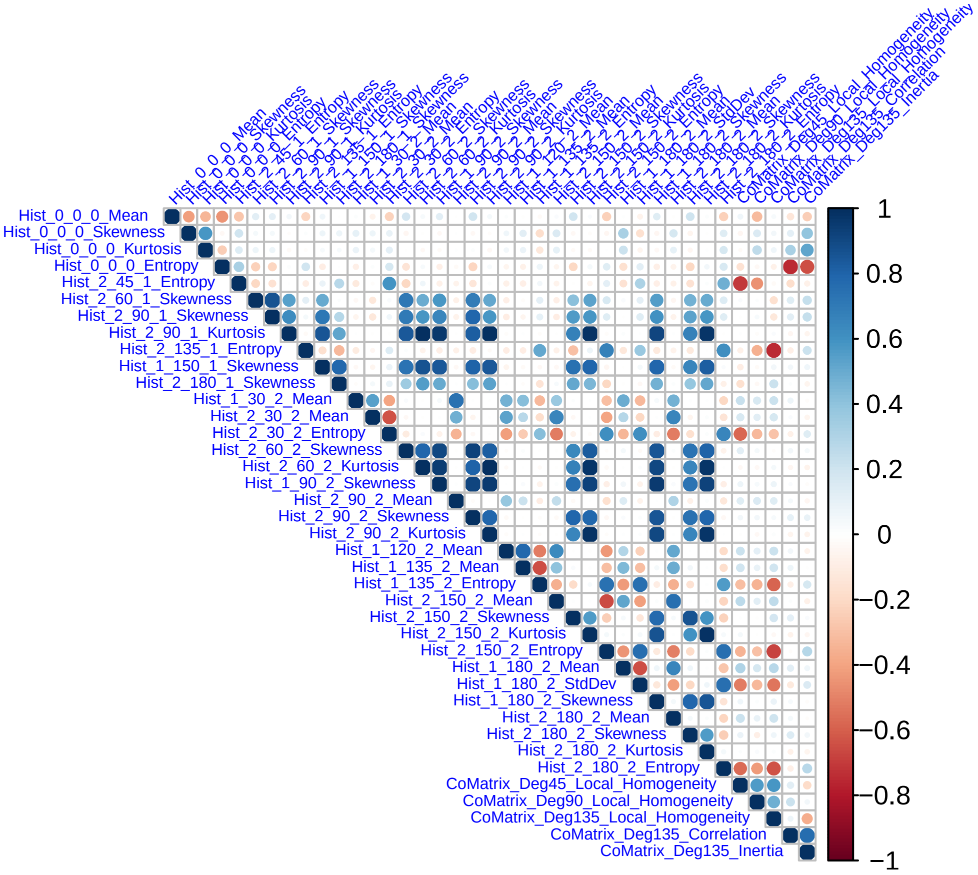


Figure 2: Correlation Matrix

Table 2: Highly Correlated Variables

Hist_1_90_2_Skewness	Hist_2_30_2_Entropy
Hist_1_180_2_Skewness	Hist_2_180_2_Entropy
Hist_2_90_2_Skewness	Hist_1_180_2_StdDev
Hist_1_150_1_Skewness	Hist_1_135_2_Entropy
Hist_2_60_2_Skewness	Hist_2_180_1_Skewness
Hist_2_60_2_Kurtosis	CoMatrix_Deg135_Local_Homogeneity
Hist_2_90_2_Kurtosis	CoMatrix_Deg45_Local_Homogeneity
Hist_2_180_2_Skewness	Hist_2_150_2_Mean
Hist_2_150_2_Kurtosis	Hist_2_180_2_Mean
Hist_2_180_2_Kurtosis	Hist_1_120_2_Mean
Hist_2_150_2_Skewness	Hist_1_30_2_Mean
Hist_2_90_1_Skewness	CoMatrix_Deg135_Inertia
Hist_2_150_2_Entropy	Hist_0_0_0_Entropy
Hist_2_60_1_Skewness	Hist_0_0_0_Skewness

In **Figure 3**, the variables having a pair-wise correlation greater than 0.5 or less than -0.5 have been removed, see **Table 2**. Since some modeling methods are more sensitive to multicollinearity than others, a new data set will be created from the remaining variables. This reduces the count of predictors from 40 to 12.

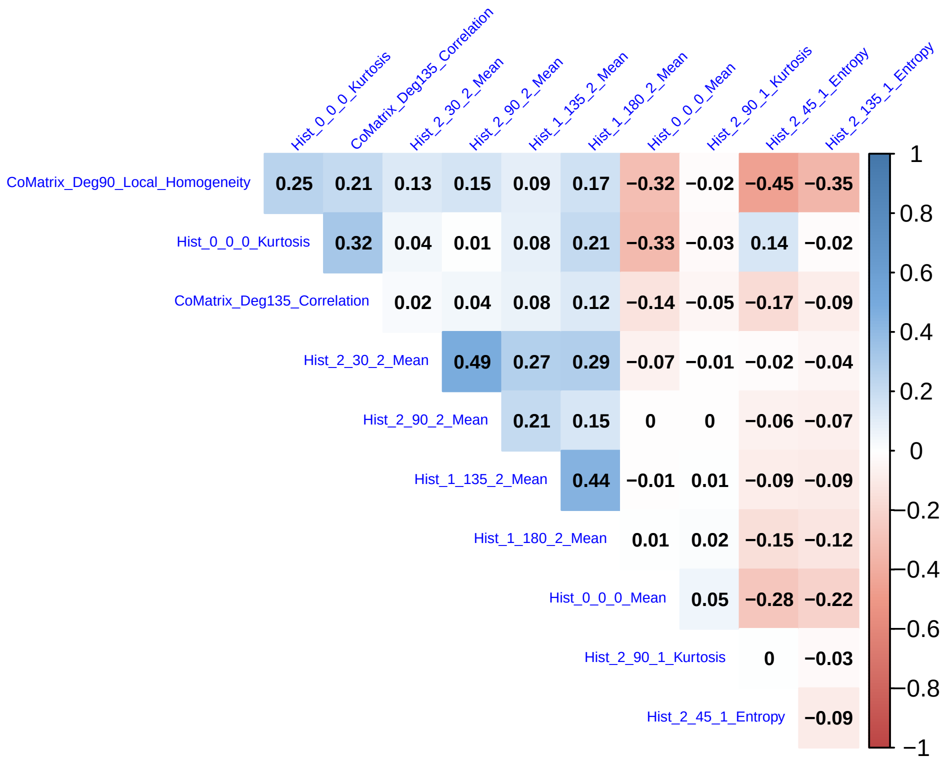


Figure 3: Correlation Matrix - Highly Correlated Variables Removed

3.2 Distribution - Predictors

Figure 4 provides examples of histograms for several of the predictors. Different modeling methods make assumptions about the distributions of the independent variables which could impact their relationship with the dependent variable, so it is good to review them. The distribution patterns seen here are common among the predictors in the dataset. Histograms for each of the predictors are available in **Appendix A.2**.

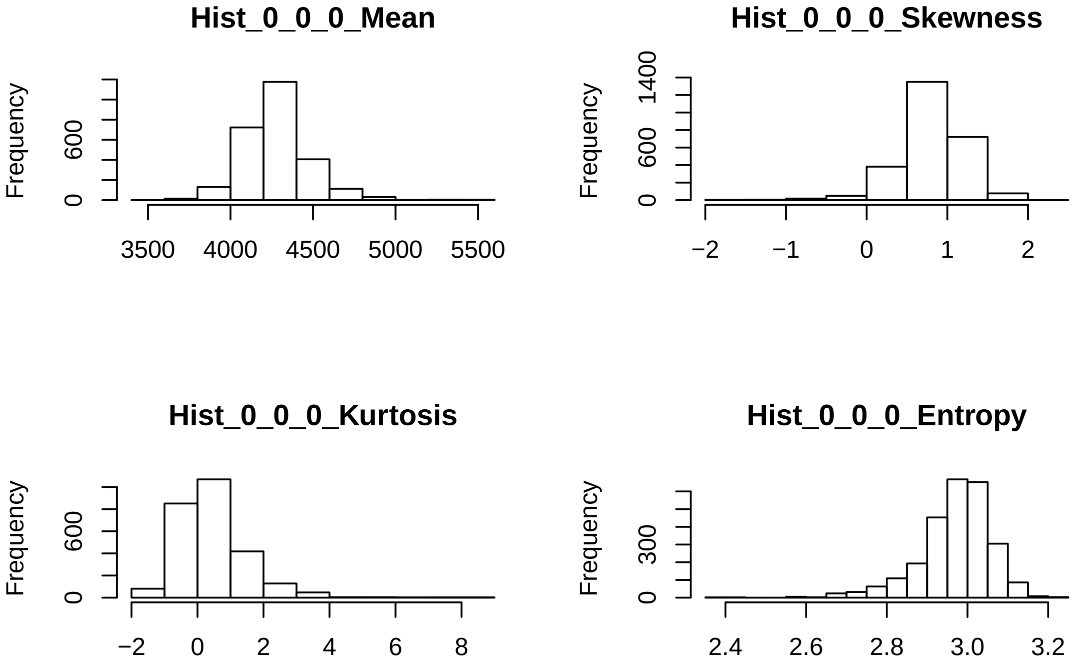


Figure 4: Histograms

3.3 Outliers

An outlier is an observation that is much smaller or larger than other values. It is important to identify potential outliers since they can skew or add bias to a model. The box plots **Figure 5** provide an easy way to determine if outliers are present for a variable. The box plots show the minimum and maximum values on the vertical line, and the box which are the boundaries for the 1st and 3rd quartiles. The horizontal line in the middle of the box is the median. Note that in the box plot for *Hist_0_0_0_Kurtosis*, the median is very close to the 1st and 3rd quartiles. Box plots for all predictors were created and reviewed, but were not provided to conserve space. Most variables display patterns similar to those seen in **Figure 5**. Box plots for the remaining variables are provided in **Appendix A.3**.

The potential outliers are the individual data points that fall above or below the box, or 1st and 3rd quartiles. While these points can be influential, their full impact is difficult to determine in advance. The traditional processes for handling outliers includes removing them or capping their values. Since no information was provided on the valid range of each variable, it will be assumed that all data points are valid. Additionally, some modeling methods are less sensitive to outliers than others, so the potential impact of these points will be consisdered for each model.

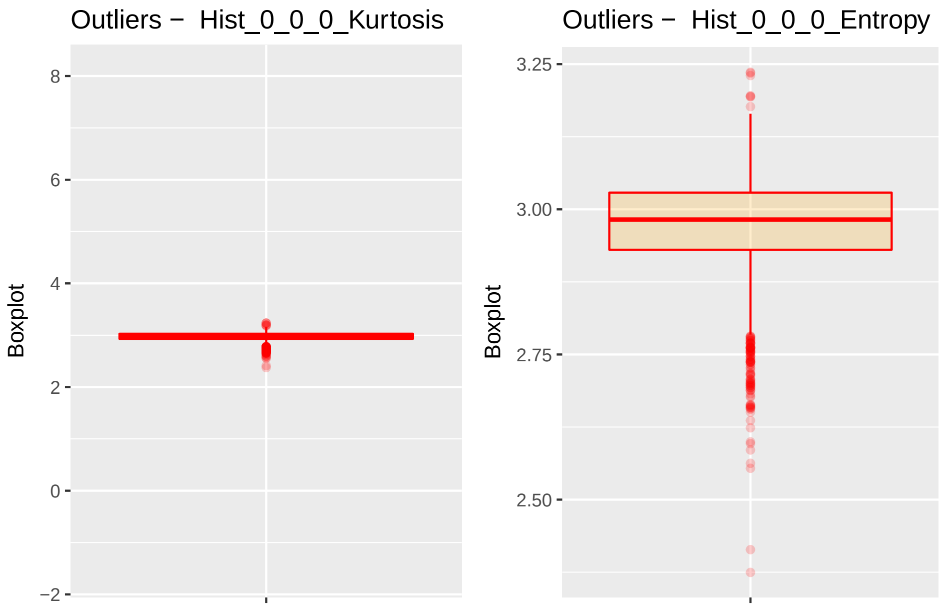


Figure 5: Box plots

3.4 Evaluation Metrics

In order to understand how well the models are performing, or detecting Pneumoconiosis, various evaluation metrics will be considered. The most common evaluation metrics for classification include recall (sensitivity), specificity, precision, F1 and accuracy and are based on the rates of True Positive [TP], True Negative [TN], False Positive [FN] and False Negative [FN].

Since the evaluation metrcis are defined in terms of TP, TN, FP and FN rates, definitions for each are provided. TP is when the model predicts 'Abnormal' and the patient actually does have Pneumoconiosis. TN is when the model predicts 'Normal' and the patient does not have Pneumoconiosis. FP is when the model predicts 'Abnormal, but the patient does not actually have Pneumoconiosis, and FN is when the model predicts 'Normal', but Pneumoconiosis is present. The following definitions may now be applied:

3.4.1 Evaluation metric definitions

- Accuracy - $(\frac{1}{N})|\{i|y_i = \hat{y}_i\}| = (\text{TN} + \text{TP})/(\text{TN} + \text{TP} + \text{FN} + \text{FP})$
- Recall (sensitivity) - $\frac{|\{i|y_i=\hat{y}_i,\hat{y}_i=1\}|}{|\{i|y_i=1\}|} = \text{TP}/(\text{TP} + \text{FN})$
- Specificity - true negative rate = TN/(TN + FP)
- Precision - $\frac{|\{i|y_i=\hat{y}_i,\hat{y}_i=1\}|}{|\{i|\hat{y}_i=1\}|} = \text{TP}/(\text{TP} + \text{FP})$
- F1 - weighted average of precision and recall = 2 * (TP/(TP + FN) * TP/(TP + FP))/(TP/(TP + FN) + TP/(TP + FP))
- Kappa - (observed accuracy - expected accuracy)/(1 - expected accuracy)
  - observed accuracy = (TN + TP) / (TN + TP + FN + FP)
  - expected accuracy = (((TN + FP) * (TN + FN)) / (TN + TP + FN + FP) + (TN * (FP + TP)) / (TN + TP + FN + FP)) / (TN + TP + FN + FP)

F1 takes both false positives and false negatives into account, so it can be more useful than accuracy alone, especially when there is an uneven class distribution (Zhu, Zeng, & Wang, 2010). The Kappa statistic will also be used for model evaluation and comparison. The Kappa statistic compares the observed accuracy with the expected accuracy and takes random chance into account (Cohen, n.d.).

3.5 Naive Model

A simple decision tree was used to fit a naive model. This model uses the full data set containing all precitors and with no transformations or scaling. The purpose of this model is to provide a baseline for the models that will be built since any improvements made to the data set or modeling method should result in a better performing model. **Table 3** provides the simple confusion matrix for the model. The model predicted 'Abnormal' for 750 observations when the true result actually was 'Abnormal', and predicted 'Normal' for 1,582 observations when the true result actually was 'Normal'. However, the model incorrectly classified 194 observations as 'Abnormal' when the true result was actually 'Normal', and 80 observations as 'Abnormal' when the true result was 'Normal'.

Table 3: Confusion Matrix

	Abnormal	Normal
Abnormal	750	80
Normal	194	1582

The evaluation metrics for the naive model are provided in **Table 4**. The performance of this simple model is much better than expected, with an accuracy of 89% and an F1 of 85%. With a specificity of 95% and a recall of 79%, the model is better at identifying true negatives than it is at identifying true positives. Note that precision, also known as the positive predictive value, is 90%.

Table 4: Evaluation Metrics

	Naive Model Metrics
Accuracy	0.8948580
Kappa	0.7663472
AccuracyLower	0.8824425
AccuracyUpper	0.9063781
AccuracyNull	0.6377590
AccuracyPValue	0.0000000
McnemarPValue	0.0000000
Sensitivity	0.7944915
Specificity	0.9518652
Pos Pred Value	0.9036145
Neg Pred Value	0.8907658
Precision	0.9036145
Recall	0.7944915
F1	0.8455468
Prevalence	0.3622410
Detection Rate	0.2877974
Detection Prevalence	0.3184958
Balanced Accuracy	0.8731784

The naive model is also useful for identifying the variables that are likely to be the best predictors. As seen in **Figure 6**, the variables *Hist_2_150_2_Entropy*, *Hist_2_30_2_Entropy*, and *Hist_1_180_2_Mean* play key roles in this model. Any observation having a *Hist_2_150_2_Entropy* greater than or equal to 2.7 are then evaluated against *Hist_2_30_2_Entropy*, while observations having a *Hist_2_150_2_Entropy* of less than 2.7 are evaluated against *Hist_1_180_2_Mean*.

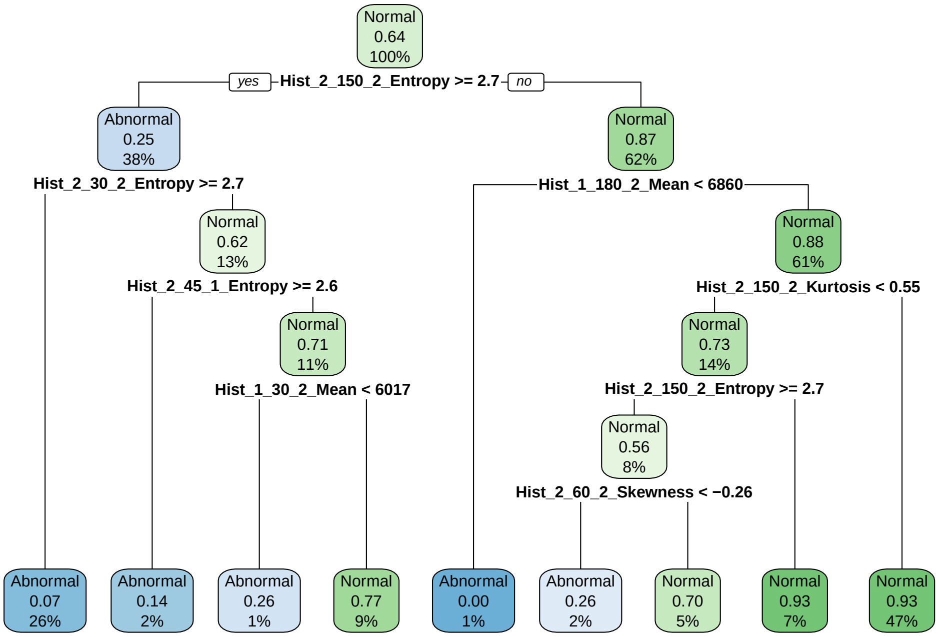


Figure 6: Naive Decision Tree

Figure 7 provides the Receiver Operating Characteristic curve (ROC). The ROC curve is a useful tool for evaluating predictive models and shows how well the model can distinguish between the true positives and the true negatives. The curve is the result of plotting the recall against 1 - specificty. The further the curve is from the diagonal line, the better the model is at discriminating between positives and negatives. A ROC curve that hugs the diagonal line means the model's predictive capabilities are not much better than chance.

Another useful statistic is the Area Under the Curve (AUC). This metric also measures how well the model predicts. An AUC of '1' indicates that the model perfectly separates the true positives from the true negatives, while an AUC of '0.5' means the model is not able to separate the true positives from the true negatives. ROC curves and AUC measures will also be used to evaluate the fitted models.

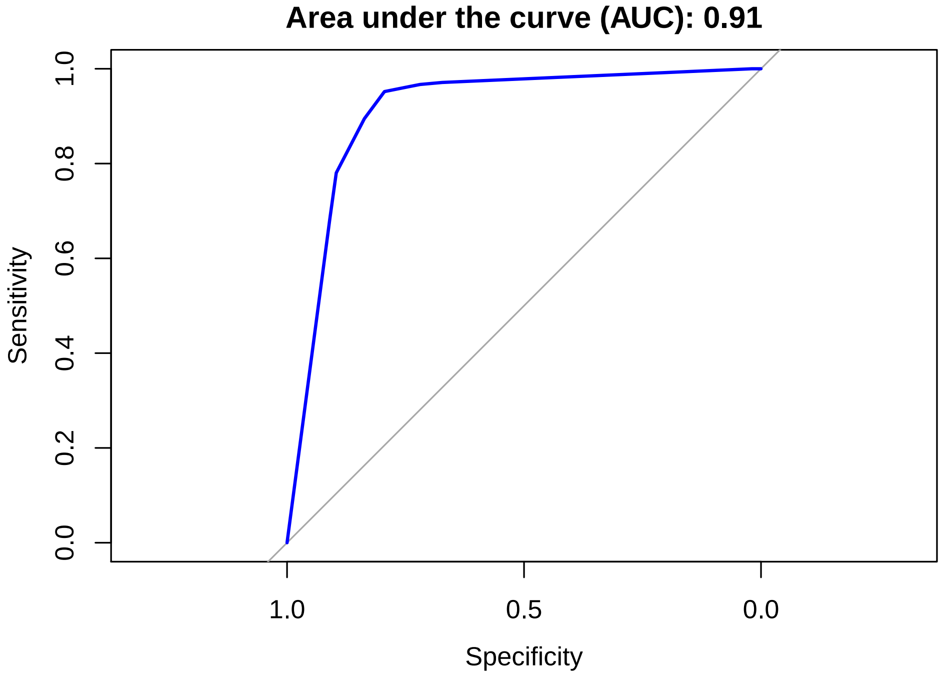


Figure 7: Naive ROC Curve

3.6 Variable Importance

It is always beneficial to reduce the complexity and dimensionality of a model whenever possible. One way to simplify a model is to reduce the number of predictors to include only those that actually provide value for the model. Including more predictors than needed increases the run-time for the model, and makes it prone to overfitting. An overfit model is one that fits the training data too well because it has learned the details and the noise. While the model may provide excellent results on the training data, it typically performs poorly on new data.

The ‘caret’ package in R provides the ‘rfe’ function for recursive feature elimination which is a simple backwards selection process to identify the best predictors. The function implements backwards selection of predictors based on predictor importance ranking. The predictors are ranked and the less important ones are sequentially eliminated prior to modeling (Kuhn, n.d.). The goal is to find a subset of predictors that can be used to produce an accurate model. **Figure 8** shows that the best accuracy - roughly 93%, is achieved with 16 variables. Note that the accuracy degrades as more predictors are added to the model.

It is also worth noting that the ‘rfe’ function allows the selection process to be performed using numerous cross validation techniques, including leave one out cross validation. However, the function was implemented using 10-fold cross validation, repeated five times. The results were almost identical to those of LOOCV but completed in seconds while the LOOCV method took over an hour to run.

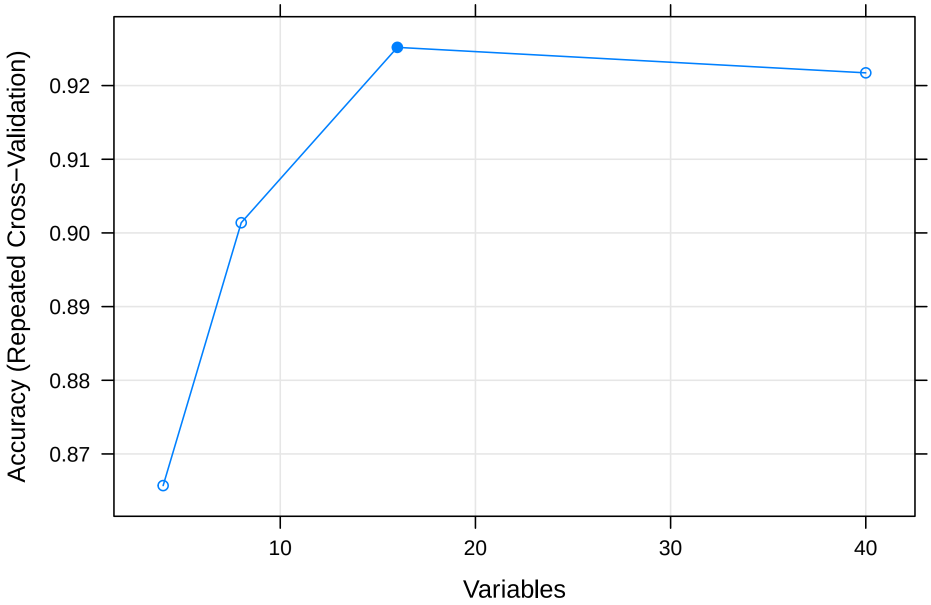


Figure 8: Variable Selection

The predictors selected through the 'rfe' function are listed in **Table 5**. This list includes all but three of the variables selected in the naive model. The predictors selected by removing the highly correlated varaiables are listed in **Table 6**. This list of predictors is quite different from those selected in the naive model.

Table 5: Backwards Selection Variables

Backwards Selection
Hist_2_150_2_Entropy
Hist_2_30_2_Entropy
Hist_2_180_2_Entropy
Hist_1_120_2_Mean
Hist_1_180_2_StdDev
Hist_1_90_2_Skewness
Hist_2_90_2_Mean
Hist_2_90_1_Kurtosis
Hist_1_135_2_Entropy
Hist_2_150_2_Mean
Hist_1_180_2_Mean
CoMatrix_Deg90_Local_Homogeneity
Hist_2_30_2_Mean
Hist_0_0_0_Mean
Hist_2_180_2_Kurtosis
Hist_1_30_2_Mean

Table 6: Low Correlation Predictor Variables

Low Correlation
Hist_0_0_0_Mean
Hist_0_0_0_Kurtosis
Hist_2_45_1_Entropy
Hist_2_90_1_Kurtosis
Hist_2_135_1_Entropy
Hist_2_30_2_Mean
Hist_2_90_2_Mean
Hist_1_135_2_Mean
Hist_1_180_2_Mean
CoMatrix_Deg90_Local_Homogeneity
CoMatrix_Deg135_Correlation
Position

Table 7 lists the six variables that are common for both selection sets. It would preferrable if more variables were common between the two data sets, but it is not too surprising that the lists differ. Simply removing the variables that are highly correlated does not necessarily mean that the remaining varaibles are good predictors. Models will be fitted using the full data set, the backwards selection data set and the low correlation data set so that the results may be compared.

Table 7: Common Predictor Variables

Common Predictors
Hist_2_90_2_Mean
Hist_2_90_1_Kurtosis
Hist_1_180_2_Mean
CoMatrix_Deg90_Local_Homogeneity
Hist_2_30_2_Mean
Hist_0_0_0_Mean

Since only the low correlation data set included the *Position* variable, the six lung segments will be modeled as one data set. This prevents the need of having to run mulitiple models of each type for each lung segment. While it is possible that the data from a particular segment may be more predictive than others, this could vary by model type. If performance from the combined data sets is less than adequate then further analysis may be necessary on the individual segments to understand if there is a statistically significant difference between them.

3.7 Leave One Out Cross Validation

In leave one out cross validation [LOOCV] each observation is used as a validation set and the remaining n - 1 observations are used as the training set. The model is fit to the training set and validated against the individual response from the validation set. This process is repeated until each observation has been used as the validation set.

LOOCV offers some advantages and disadvantages. Some advantages of the LOOCV approach include the lack of randomness in the data, and reduced bias. Since all observations are used to both train and test the model, there is no chance of data being excluded from the process. For this reason, LOOCV will always provide the same results, no matter how many times the model is run, provided the data set does not change. Additionally, using n - 1 observations to train the model reduces the bias. The benefit is that there is a reduction in the over-estimation of the test error that is seen with other cross validation methods.

As for the disadvantages, LOOCV can be computationally expense since the model needs to be fit and run once for each row in the data set. More complex models will require additional time to run with LOOCV. Lastly, even though the individual iteration's test error is unbiased, LOOCV has high variability since only one observation is used in the validation set for prediction.

4 Modeling

Models will now be fitted using the three data sets discussed earlier. The task is to find the best model, using leave one out cross validation, that maximizes the identification of true positives and true negatives while minimizing the false positives and false negatives. The 'caret' package in R will be used for all model building. This package acts as a wrapper for other R packages. While using the 'caret' package does incur some overhead, it offers many benefits. These benefits include a consistent method for building models of various types, a well formatted and thorough confusion matrix, variable importance functionality and the package itself is very well documented. Another benefit of using the 'caret' package is that with the LOOCV method, both the observed value and the prediction are saved within the resulting model object and can be easily retrieved.

Evaluation metrics, which are averaged across the cross-validated samples are included for each model, but ROC and Variable Importance plots are included for only the be best performing model of each type.

4.1 Generalized Linear Model

Generalized Linear Models [GLM] are an extension of linear regression that allow for the response variable to have a non-normal distribution. Having a binary response variable, the relationship between the predictors and the response variable is not linear, so the logit link function is used to provide a transformation and ensure that the response is constrained between 0 and 1. The logit function is defined as:

$$logit(p) = log\frac{p}{1-p}$$

The evaluation metrics for the GLM models are available in **Table 8**. The full data set has a slight edge here with an accuracy of 90.64%. All other metrics agree that the model performs slightly better as well, which is not always the case.

Table 8: GLM Model Results

Model	Data	Accuracy	Sensitivity	Specificity	Precision	Kappa	F1	ROC
GLM	Full	0.9063699	0.8527542	0.9368231	0.8846154	0.7957683	0.8683927	0.9597630
GLM	RFE Selection	0.8963929	0.8305085	0.9338147	0.8769575	0.7731689	0.8531012	0.9518270
GLM	Low Correlation	0.8564850	0.7595339	0.9115523	0.8298611	0.6835999	0.7931416	0.9130591

The confusion matrix for this model is available in **Table 9**. While the model is not performing poorly, it is only slightly better than the naive model and predicts Pneumoconiosis in 105 patients when no disease is actually present; this is known as a type I error. The model also fails to predict the presence of Pneumoconiosis in 139 patients; this is a type II error.

Table 9: GLM Model Confusion Matrix

	Abnormal	Normal
Abnormal	805	105
Normal	139	1557

The AUC in **Figure 9** of 0.959763 is slightly misleading given a model accuracy of 90.64%. For classification, no single evaluation metric tells the whole story, so it is always best to review several. The 'confusionMatrix' function in the 'caret' package simplifies the process of generating all necessary evaluation metrics.

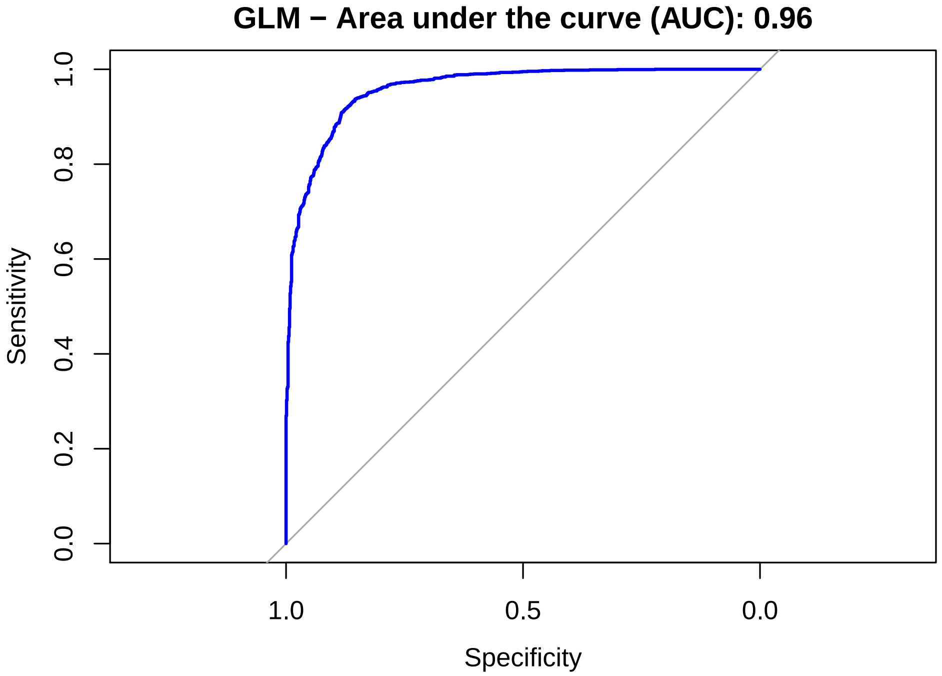


Figure 9: GLM ROC Curve

Lastly, since GLM is extension of linear regression, the regression coefficients are available for the model in **Table 10**. To conserve space, only the first 10 are listed here. The full list of coefficients may be providedd for further review if requested. Note the rather large negative intercept and the small role that variables like *Hist_0_0_0_Mean*, and *Hist_0_0_0_Kurtosis* play.

Table 10: GLM Model Coefficients

Variable	Coefficient
(Intercept)	-430.5202100
Hist_0_0_0_Mean	0.0121352
Hist_0_0_0_Skewness	1.7659236
Hist_0_0_0_Kurtosis	0.0221390
Hist_0_0_0_Entropy	12.9470078
Hist_2_45_1_Entropy	54.4027012
Hist_2_60_1_Skewness	1.4798851
Hist_2_90_1_Skewness	-2.0924020
Hist_2_90_1_Kurtosis	0.2586639
Hist_2_135_1_Entropy	-11.4864399

4.2 Support Vector Machine

A Support Vector Machine [SVM] is a supervised machine learning algorithm defined by a separating hyperplane. In this algorithm, each data element is plotted in n - dimensional space, where n is the number of features. Then classification is performed by finding the hyperplane that best differentiates the classes (Ray, 2017b). SVM is a very popular modeling techhinque that often provides great results with very little tuning.

A radial basis function kernel [RBF] was selected as the classifier since the data is not linearly separable. The RBF is defined as:

$$f(x) = \sum_{i}^{N} \alpha_i y_i exp(-\left\| x - x_i \right\|^2 / 2\sigma^2) + b$$

The RBF has two tuning parameters available through the 'caret' package. A wider or softer margin is created when C (cost) is decreased and that for larger values of σ the decision boundary tends to be smoother and more flexible. It also tends to misclassify more often, but reduces the likelihood of overfitting the model (H. Wang, 2014).

Figure 10 provides an example of the non-linear decision boundary. Note that the separation is less than ideal, but that is to be expected since these variables were selected at random and represent only part of the models separation capabilities. Unforunately, it is not possible to plot the separation planes in 16 - dimensions.

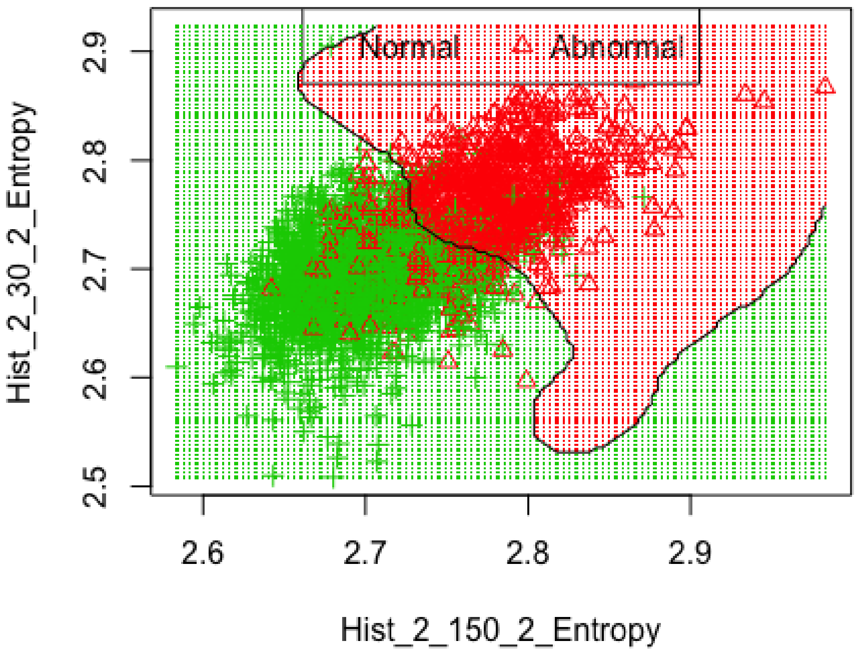


Figure 10: Decision Boundaries

The evaluation metrics for the SVM models are available in **Table 11**. As with the GLM models, the full data set has a slight edge here with an accuracy of 91.86%.

Table 11: SVM Model Results

Model	Data	Accuracy	Sensitivity	Specificity	Precision	Kappa	F1	ROC
SVM	Full	0.9186493	0.8633475	0.9500602	0.9075724	0.8220604	0.8849077	0.9700834
SVM	RFE Selection	0.9128933	0.8516949	0.9476534	0.9023569	0.8091614	0.8762943	0.9716405
SVM	Low Correlation	0.8752878	0.7754237	0.9320096	0.8662722	0.7238303	0.8183343	0.9342194

The confusion matrix for this model is available in **Table 12**. The model predicts Pneumoconiosis in 83 patients when no disease is actually present, and fails to predict the presence of Pneumoconiosis in 129 patients.

Table 12: SVM Model Confusion Matrix

	Abnormal	Normal
Abnormal	815	83
Normal	129	1579

The AUC in **Figure 11** of 0.9700834 is a slight improvement over the best GLM model.

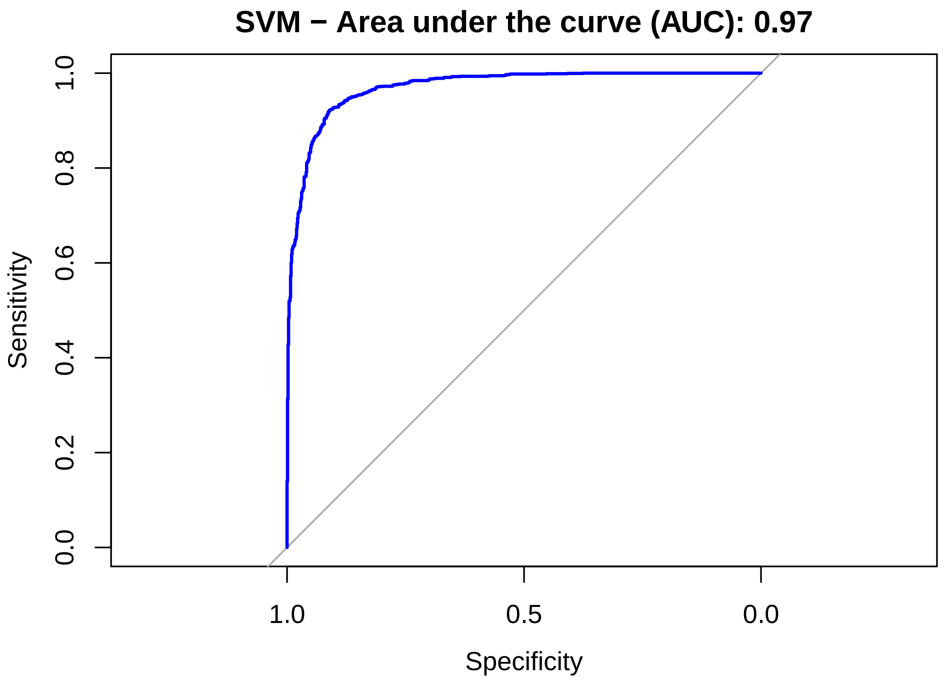


Figure 11: SVM ROC Curve

The SVM models do show an improvement, but not as much as hoped or expected. This is partially due to the use of LOOCV. Manually tuning this model would likely significantly improve the model's performance.

4.3 k - Nearest Neighbor

The k - Nearest Neighbor [kNN] algorithm is one of the simplest, yet most popular classifiers. It is non-parametric, meaning it makes no assumptions about the distribution of the data. It is also a lazy algorithm, meaning the training phase is very minimal. The algorithm performs classification through a majority vote of its k- nearest neighbors, hence the name (Bronshtein, 2017)

Some of the advantages of kNN include insensitivity to outliers, high accuracy, and useful for classification and regression. However, kNN has high memory requirements and the data should be centered and scaled prior to fitting the model.

The evaluation metrics for the kNN models are available in **Table 13**. The trimmed data set from the 'rfe' method performed best with an accuracy of 93.44%. This is the best performing model so far.

Table 13: kNN Model Results

Model	Data	Accuracy	Sensitivity	Specificity	Precision	Kappa	F1	ROC
kNN	Full	0.8910207	0.7891949	0.9488568	0.8975904	0.7578197	0.8399098	0.9523933
kNN	RFE Selection	0.9343822	0.8993644	0.9542720	0.9178378	0.8573638	0.9085072	0.9762134
kNN	Low Correlation	0.8568688	0.6684322	0.9638989	0.9131693	0.6711869	0.7718654	0.9193306

The confusion matrix for this model is available in **Table 14**. The model predicts Pneumoconiosis in 76 patients when no disease is actually present, but fails to predict the presence of Pneumoconiosis in 95 patients.

Table 14: kNN Model Confusion Matrix

	Abnormal	Normal
Abnormal	849	76
Normal	95	1586

The AUC in **Figure 12** of 0.9762134 shows that the model is performing quite well. Recall that the AUC of a perfect model is 1.00.

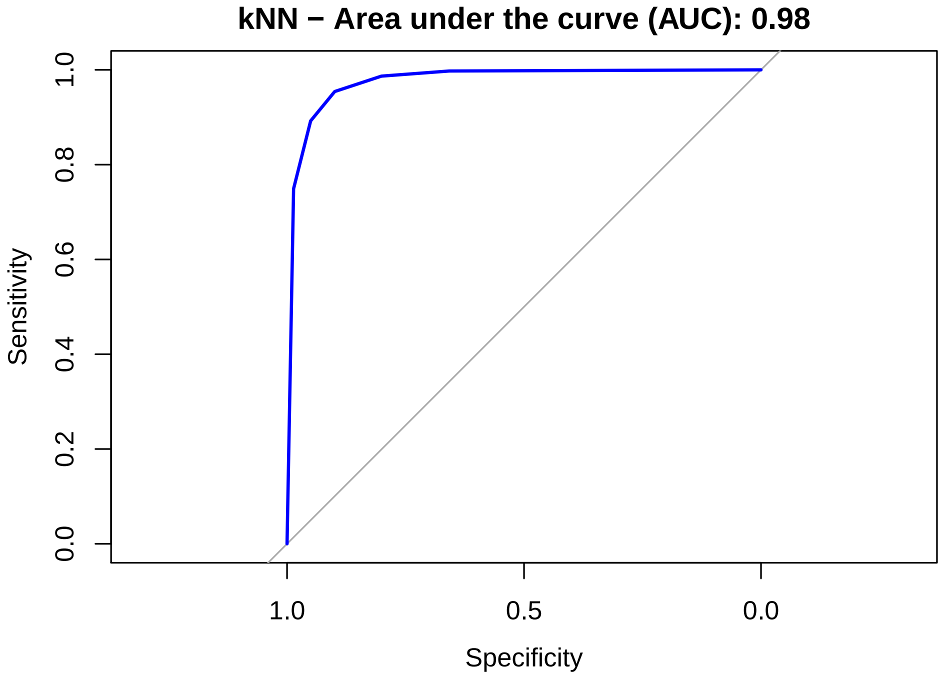


Figure 12: kNN ROC Curve

The plot in **Figure 13** shows how the AUC changes as more neighbors are selected. The model performs best with 5 neighbors and the performance degrades as more neighbors are added.

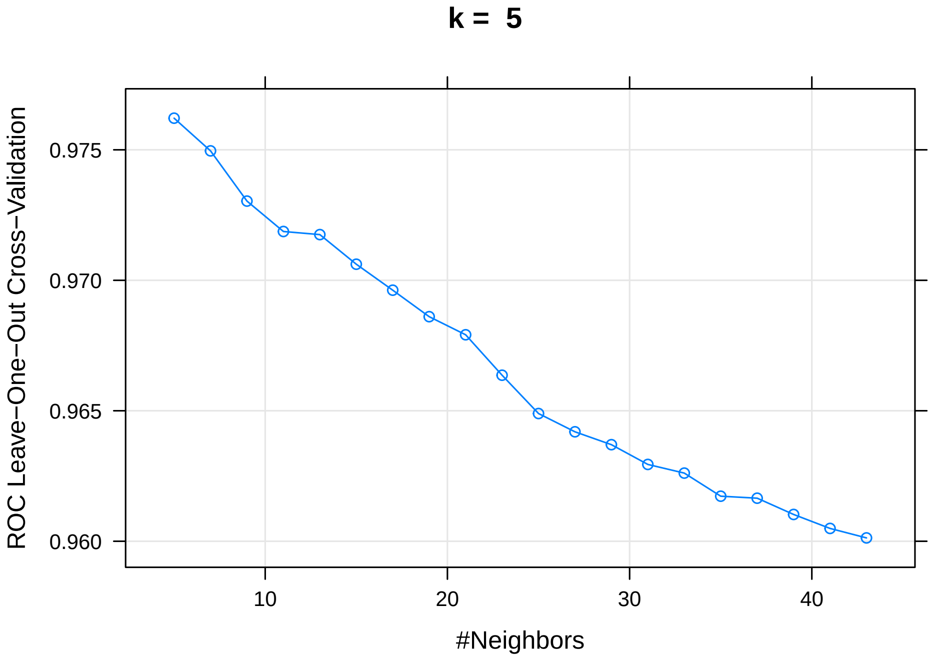


Figure 13: Values of k

4.4 Naive Bayes Models

The Naive Bayes [NB] classifier is a simple and fast learner that is known to provide excellent results. It is based on Bayes' Theorem which provides a way of calculating posterior probability P(c|x) from P(c), P(x) and P(x|c). The model works by first creating a frequency table from the data, then using that frequency table to calculate the probabilities. The Naive Bayesian equation (below) is then used to calcuate the posterior probability for each class, selecting the class with the highest probability (Ray, 2017a).

$$P(c|x) = \frac{P(x|c)P(c)}{P(x)}$$

Unfortunately, Naive Bayes did not perform as well as expected, as seen in **Table 15**. The trimmed data set from the 'rfe' method provides the best accuracy at 88.1%. The confusion matrix for this model is provided in **Table 16** and the ROC plot is available in **Figure 14**.

The recall (sensitivity) for these models is the lowest that have been observed so far. While the type I errors of 112 are in line with other models, the type II (false negative) errors of 198 are quite high in comparison. It was suspected that the bimodal distribution of several variables was causing an issue, but removing them did not help the models' performance.

Table 15: Naive Bayes Model Results

Model	Data	Accuracy	Sensitivity	Specificity	Precision	Kappa	F1	ROC
Naive Bayes	Full	0.8775902	0.7521186	0.9488568	0.8930818	0.7257181	0.8165612	0.9471744
Naive Bayes	RFE Selection	0.8810437	0.7902542	0.9326113	0.8694639	0.7373759	0.8279689	0.9463245
Naive Bayes	Low Correlation	0.8284728	0.5847458	0.9669073	0.9093904	0.5977465	0.7117988	0.9157463

Table 16: NB Model Confusion Matrix

	Abnormal	Normal
Abnormal	746	112
Normal	198	1550

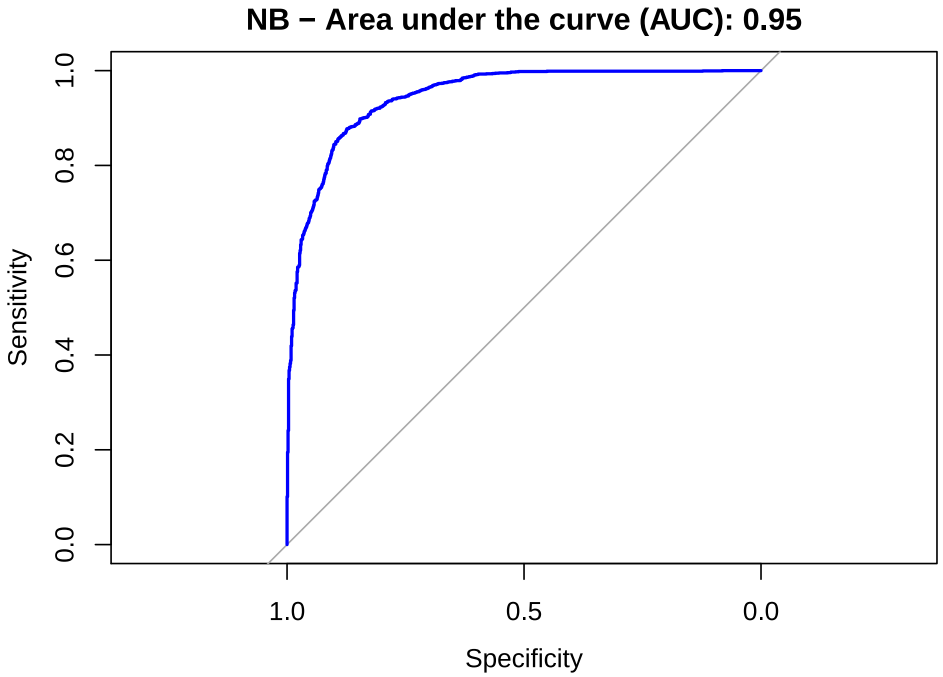


Figure 14: Naive Bayes ROC Curve

4.5 Random Forest Models

The Random Forest [RF] algoritm is also quite popular and can be used for regression or classification. They are known for above average results since they encompass all the benefits of decision trees, and are not prone to over-fitting.

For classification, a Random Forest will grow many decision trees. Each tree provides a classification which is considered a vote. The alogrithm then chooses the classification that has the most votes from all the trees in the forest.

There is some debate about the need for cross validation with Random Forests since each tree is grown with a list of randomly selected samples. This process leaves out approximately $\frac{1}{3}$ of the samples for testing the grown tree. Because the samples are randomly selected, each tree has its own learning and out-of-bag sample set, making LOOCV somewhat redundant. However, it is technically not incorrect to use cross validation with Random Forests since cross validation is one way to ensure that all data is used during the training or testing phase.

As seen in **Table 18**, the model using the trimmed data set from the ‘rfe’ function is the best of the three, and the best model of all so far. The model has an accuracy of 93.86%.

Table 17: Random Forest Model Results

Model	Data	Accuracy	Sensitivity	Specificity	Precision	Kappa	F1	ROC
Random Forest	Full	0.9259401	0.8527542	0.9675090	0.9371362	0.8365335	0.8929562	0.9780787
Random Forest	RFE Selection	0.9386032	0.8697034	0.9777377	0.9568765	0.8644521	0.9112098	0.9853100
Random Forest	Low Correlation	0.9263239	0.8548729	0.9669073	0.9361949	0.8374942	0.8936877	0.9735638

Figure 15 provides a Variable Importance plot. This plot shows the variables that were most impactful in the model. One disadvantage of Random Forests, as well as many machine learning alogrithms, is that they operate as a 'black box' so it is difficult to understand how the model is making predictions.

The variables are listed from the top down in order of importance as determined by the Mean Decrease Gini. Variables that result in nodes with higher purity have a higher decrease in the Gini coefficient. The Variable Importance plot can also be used for feature selection. Note the break after *Hist_1_180_2_StdDev*. *Hist_2_150_2_Mean* and the variables below it are not contributing as much to the model, so it is possible that removing them would not have a huge impact on the model's performance.

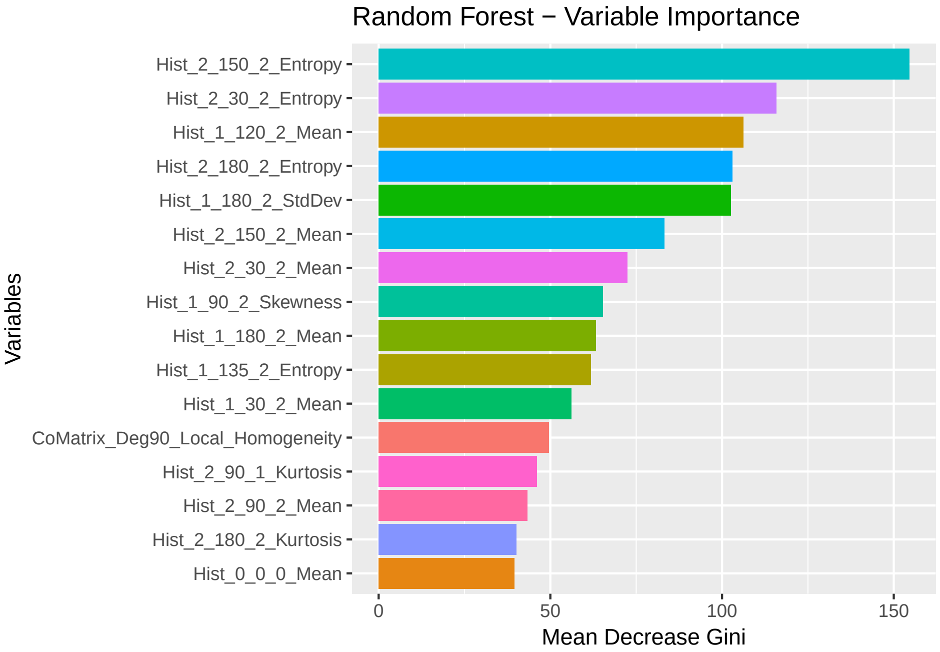


Figure 15: RF Variable Importance

The confusion matrix for the model is available in **Table 19**. The model misclassified 123 patients as disease free when Pneumoconiosis actually was present. However, the model was able to mimimize the number of false positives and classified 37 patients as having Pneumoconiosis, when the disease was not actually present.

Table 18: RF Model Confusion Matrix

	Abnormal	Normal
Abnormal	821	37
Normal	123	1625

As previously noted, the AUC in **Figure 16** of 0.98531 is slightly misleading. While the specificity is quite good, the recall still shows some issues. If one were to evaluate the model based soley on the AUC, they might believe the model is performing much better than it actually is.

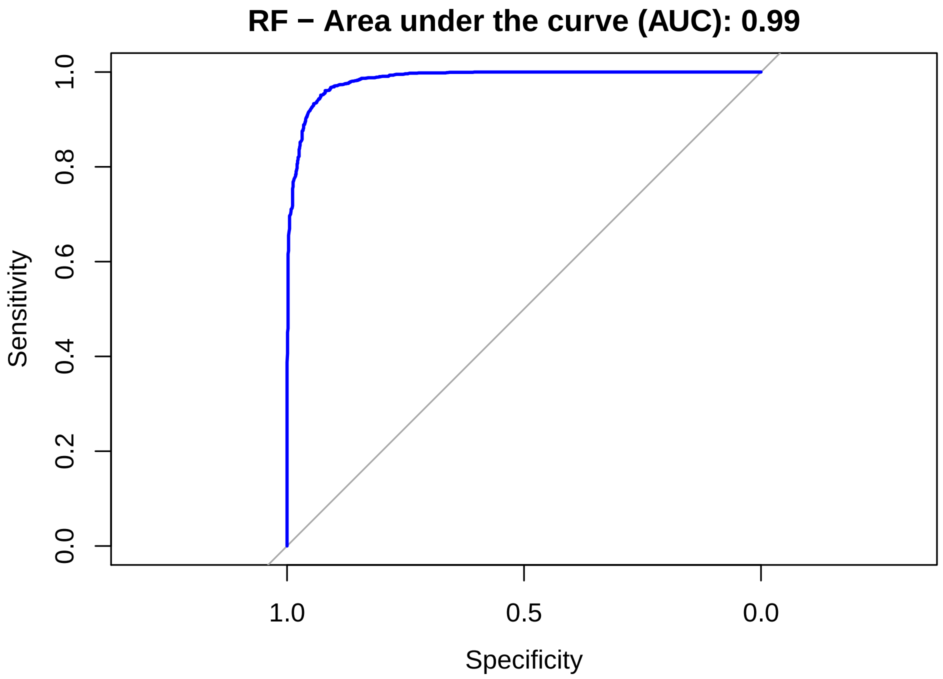


Figure 16: RF ROC Curve

4.6 XGradient Boosting

This final model should not be included as a formal submission since it does not use LOOCV. It is included to illustrate the predictive capabilities that are possible.

Gradient boosting is a machine learning technique for regression and classification problems, which produces a prediction model in the form of an ensemble of weak prediction models, typically decision trees. It builds the model in a stage-wise fashion like other boosting methods do, and it generalizes them by allowing optimization of an arbitrary differentiable loss function (Brownlee, 2016).

The evaluation metrics for the model is available in **Table 19**. The model was run against a 70% - 30% training and test set, then the selected model was run against the full data set. The test set accuracy was 92.96%, while the accuracy on the full data set was 97.89%.

Table 19: XGBoost Model Results

	Model	Data	Accuracy	Sensitivity	Specificity	Presicion	Kappa	F1	ROC
Accuracy	X-Gradient Boost	Training Set	0.9295775	0.8939929	0.9497992	0.9100719	0.8470225	0.9019608	NA
Accuracy1	X-Gradient Boost	Training Set	0.9788949	0.9682203	0.9849579	0.9733759	0.9542700	0.9707913	NA

The Variable Importance plot is available in **Figure 17**. Note how the variable *Hist_2_150_2_Entropy* plays a very significant role, followed by *Hist_2_30_2_Entropy*.

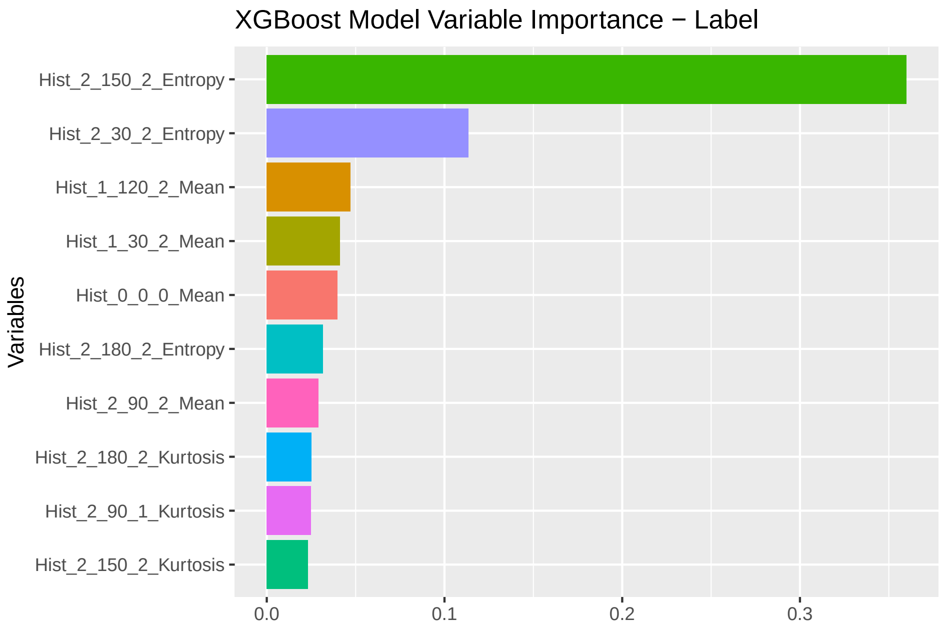


Figure 17: XGBoost Variable Importance

The AUC in **Figure 18** of 0.9765891 further illustrates how individual metrics can be deceiving. Note how this AUC is less than that seen from the Random Forest model, even though the model is performing much better.

The confusion matrix in **Table 20** shows that the model incorrectly predicts Pneumoconiosis in 25 patients when the disease is not actually present and fails to predict Pneumoconiosis in only 30 patients when the disease actaully is present.

Table 20: XGB Model Confusion Matrix

	0	1
0	914	25
1	30	1637

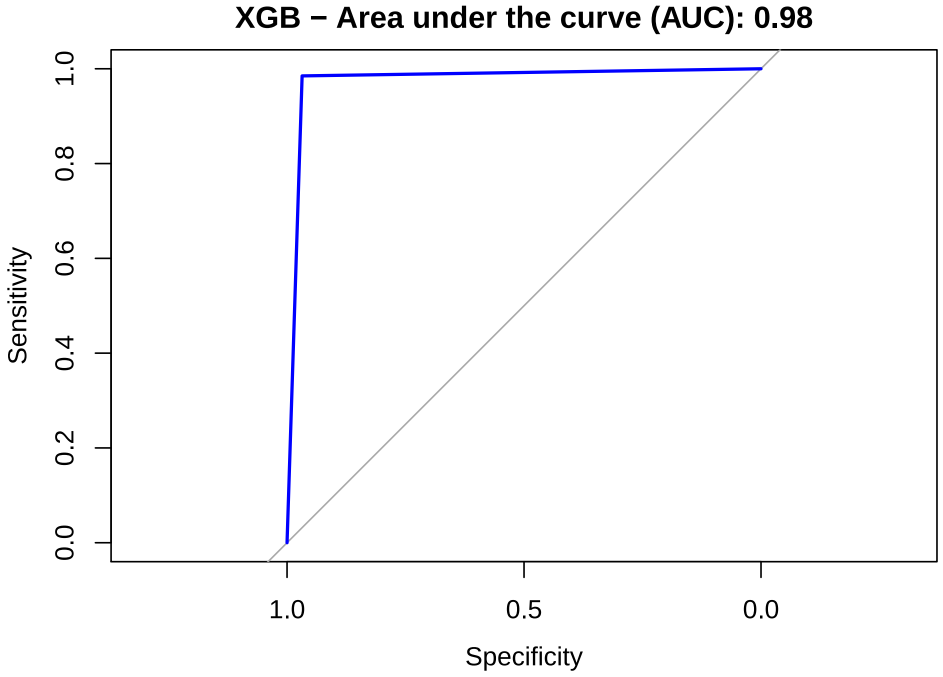


Figure 18: XGBoost ROC Curve

5 Conclusion

Several models have been presented with varying results. While most models performed reasonably well, there is much that could be done to tune individual models for better performance. Adhering to the requirements of LOOCV limited such peformance improvements, but LOOCV did prove itself to be a useful tool in determining model performance.

The XGradiant Boosting model provided an example of the performance that is possible with the data. The kNN and the Random Forest models both performed quite well, with the Random Forest winning by a small margin. However, if a preferred model were to be selected, the kNN might be a better choice since it is much easier to explain to a business user.

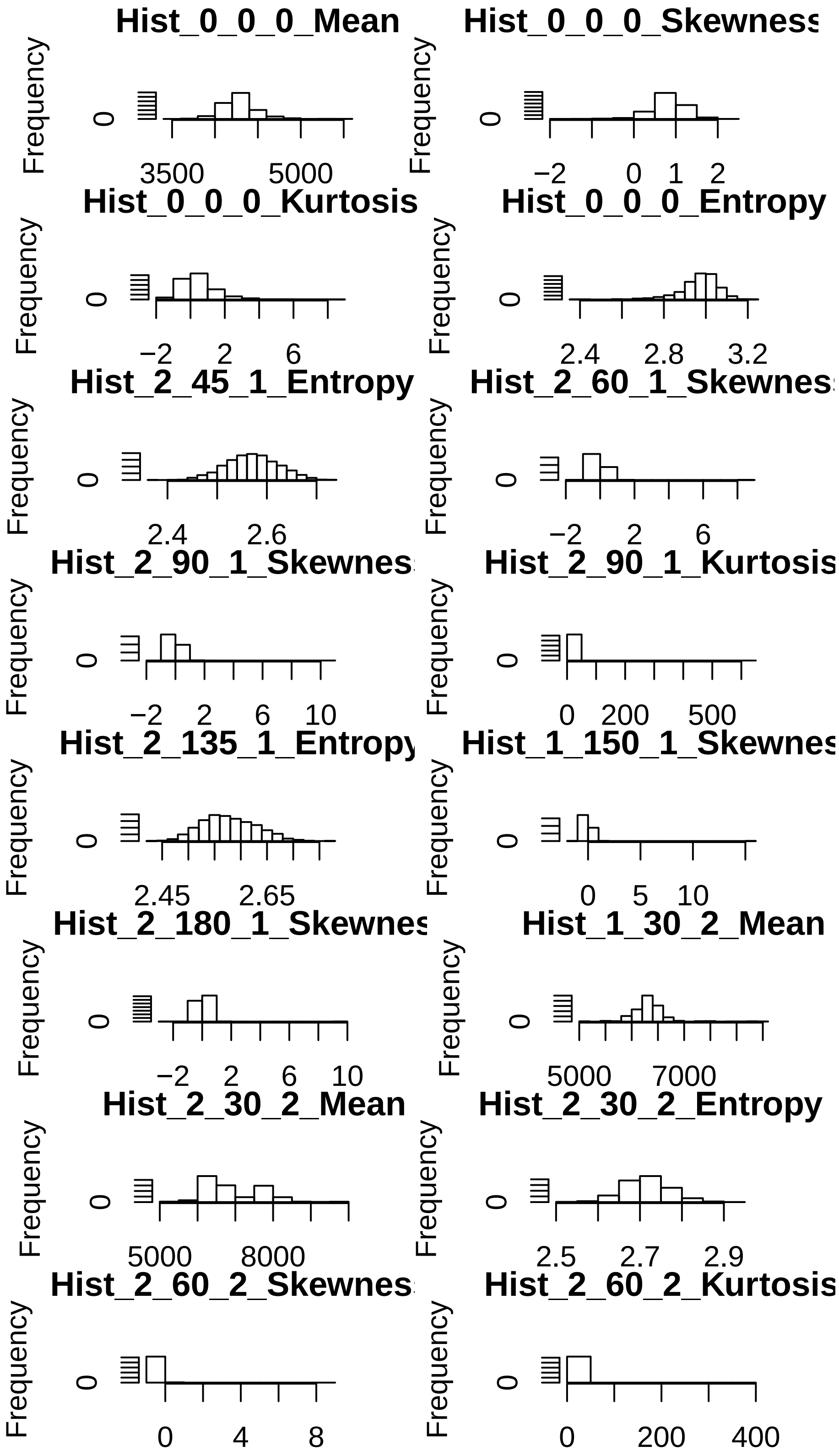
As previously discussed, it is possible that the data from one or more lung segments may yeild better predictions than data from other segments. This would be especially true if Pneumoconiosis was more common in particular areas of the lungs. Further research on how Pneumoconiosis starts or spreads may lead to better, more targeted models that provide improved results.

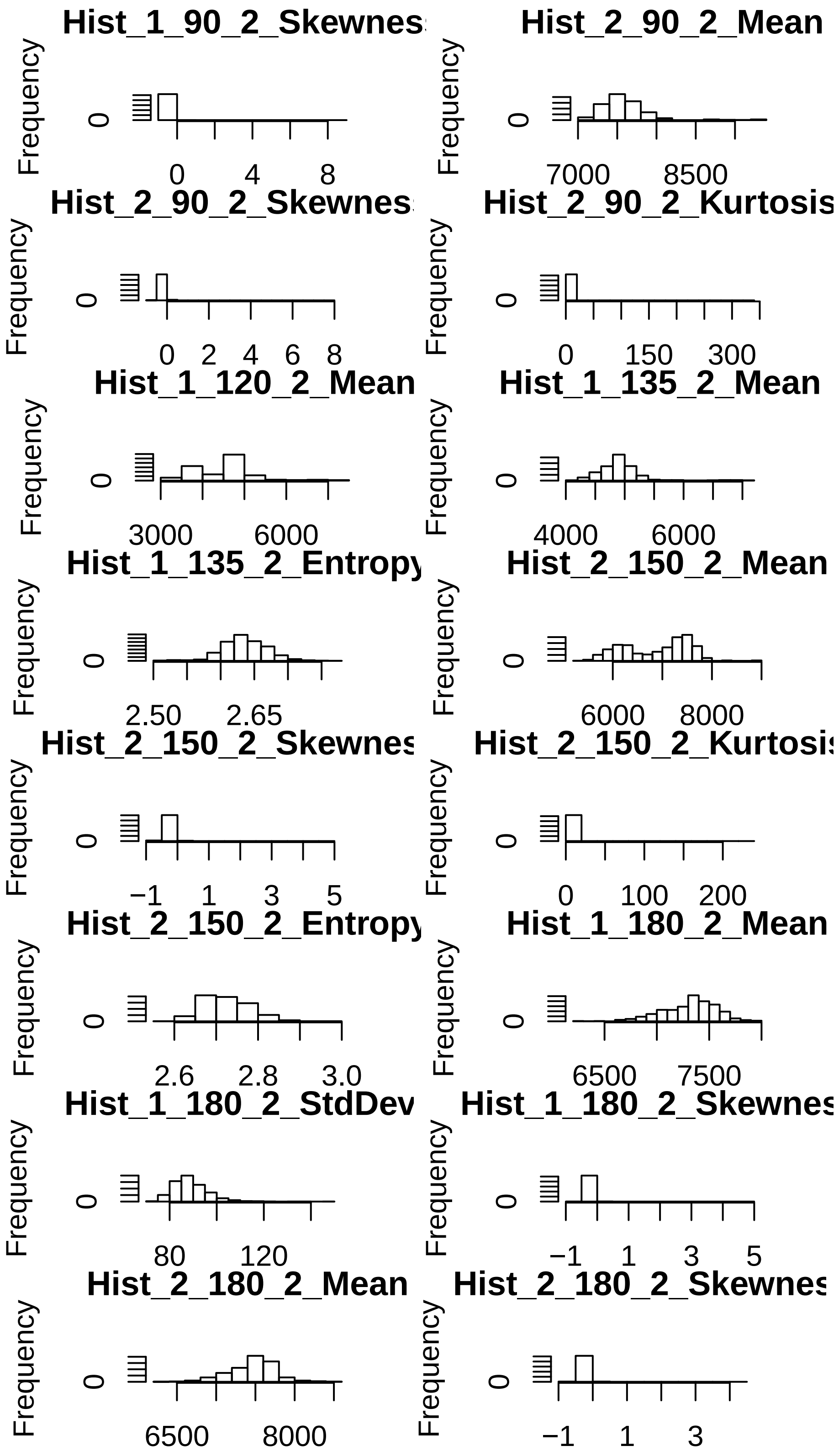
A Appendices

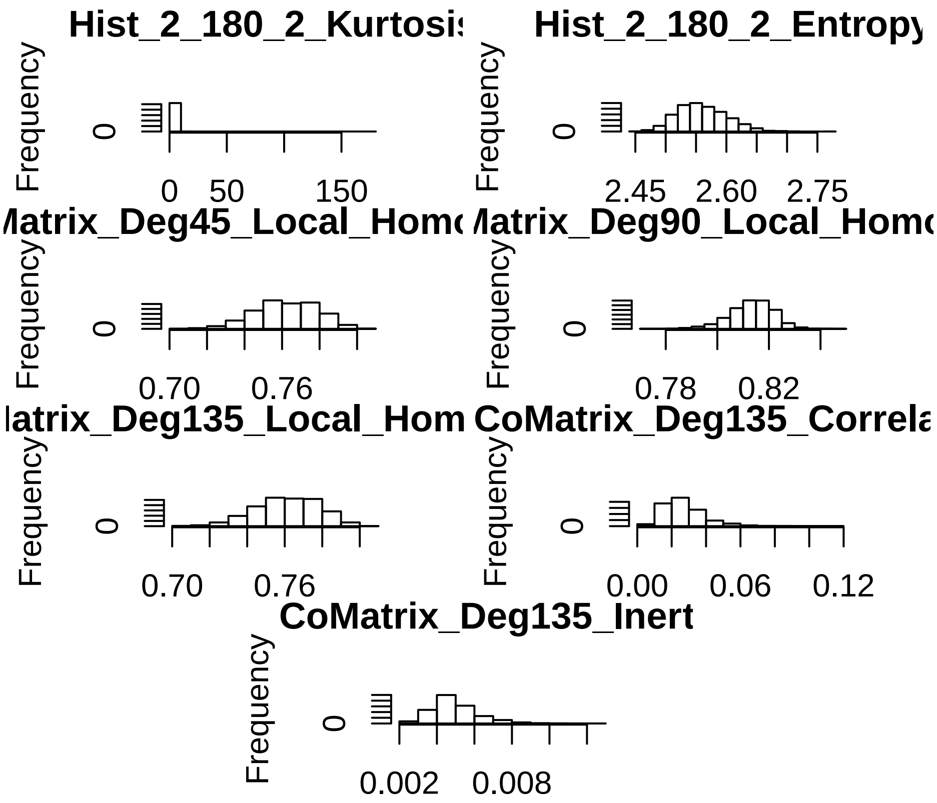
A.1 Summary Statistics for Predictors

	Mean	Median	Stdev	Minimum	Maximum	NAs
Hist__0__0__0__Mean	4280.46	4268.85	201.17	3598.77	5568.12	0
Hist__0__0__0__Skewness	0.81	0.82	0.40	-1.89	2.15	0
Hist__0__0__0__Kurtosis	0.45	0.30	1.01	-1.57	8.11	0
Hist__0__0__0__Entropy	2.97	2.98	0.09	2.37	3.24	0
Hist__2__45__1__Entropy	2.57	2.57	0.05	2.38	2.74	0
Hist__2__60__1__Skewness	-0.10	-0.10	0.32	-1.58	8.61	0
Hist__2__90__1__Skewness	-0.03	-0.07	0.35	-1.15	10.83	0
Hist__2__90__1__Kurtosis	2.34	1.99	11.91	0.46	608.27	0
Hist__2__135__1__Entropy	2.58	2.57	0.05	2.42	2.76	0
Hist__1__150__1__Skewness	-0.05	-0.05	0.35	-1.66	15.42	0
Hist__2__180__1__Skewness	0.04	0.03	0.36	-2.50	9.58	0
Hist__1__30__2__Mean	6334.04	6317.53	333.99	5077.50	8595.73	0
Hist__2__30__2__Mean	6929.42	6669.07	756.05	5313.35	9734.08	0
Hist__2__30__2__Entropy	2.72	2.71	0.05	2.51	2.92	0
Hist__2__60__2__Skewness	-0.21	-0.19	0.21	-0.86	8.86	0
Hist__2__60__2__Kurtosis	1.37	0.97	7.69	0.15	391.45	0
Hist__1__90__2__Skewness	-0.09	-0.09	0.18	-0.60	8.96	0
Hist__2__90__2__Mean	7584.56	7540.59	309.69	7001.85	9280.66	0
Hist__2__90__2__Skewness	-0.21	-0.20	0.19	-0.63	7.95	0
Hist__2__90__2__Kurtosis	1.32	1.11	6.58	0.24	335.87	0
Hist__1__120__2__Mean	4476.96	4604.67	704.44	3276.49	7385.27	0
Hist__1__135__2__Mean	4925.07	4899.61	402.36	4174.22	7112.95	0
Hist__1__135__2__Entropy	2.64	2.64	0.03	2.50	2.76	0
Hist__2__150__2__Mean	6879.67	7107.17	660.16	5314.03	8922.39	0
Hist__2__150__2__Skewness	-0.20	-0.17	0.15	-0.86	4.87	0
Hist__2__150__2__Kurtosis	1.31	0.89	4.55	0.18	228.15	0
Hist__2__150__2__Entropy	2.72	2.72	0.05	2.58	2.98	0
Hist__1__180__2__Mean	7310.77	7344.22	260.36	6221.45	7934.95	0
Hist__1__180__2__StdDev	88.81	87.84	7.25	73.31	148.98	0
Hist__1__180__2__Skewness	-0.08	-0.07	0.10	-0.91	4.69	0
Hist__2__180__2__Mean	7460.84	7506.90	295.08	6377.89	8551.48	0
Hist__2__180__2__Skewness	-0.18	-0.16	0.12	-0.73	4.07	0
Hist__2__180__2__Kurtosis	1.18	0.89	3.59	0.15	178.97	0
Hist__2__180__2__Entropy	2.56	2.56	0.04	2.44	2.77	0
CoMatrix__Deg45__Local__Homogeneity	0.76	0.76	0.02	0.71	0.80	0
CoMatrix__Deg90__Local__Homogeneity	0.81	0.81	0.01	0.77	0.85	0
CoMatrix__Deg135__Local__Homogeneity	0.76	0.76	0.02	0.70	0.80	0
CoMatrix__Deg135__Correlation	0.03	0.02	0.01	0.01	0.12	0
CoMatrix__Deg135__Inertia	0.00	0.00	0.00	0.00	0.01	0

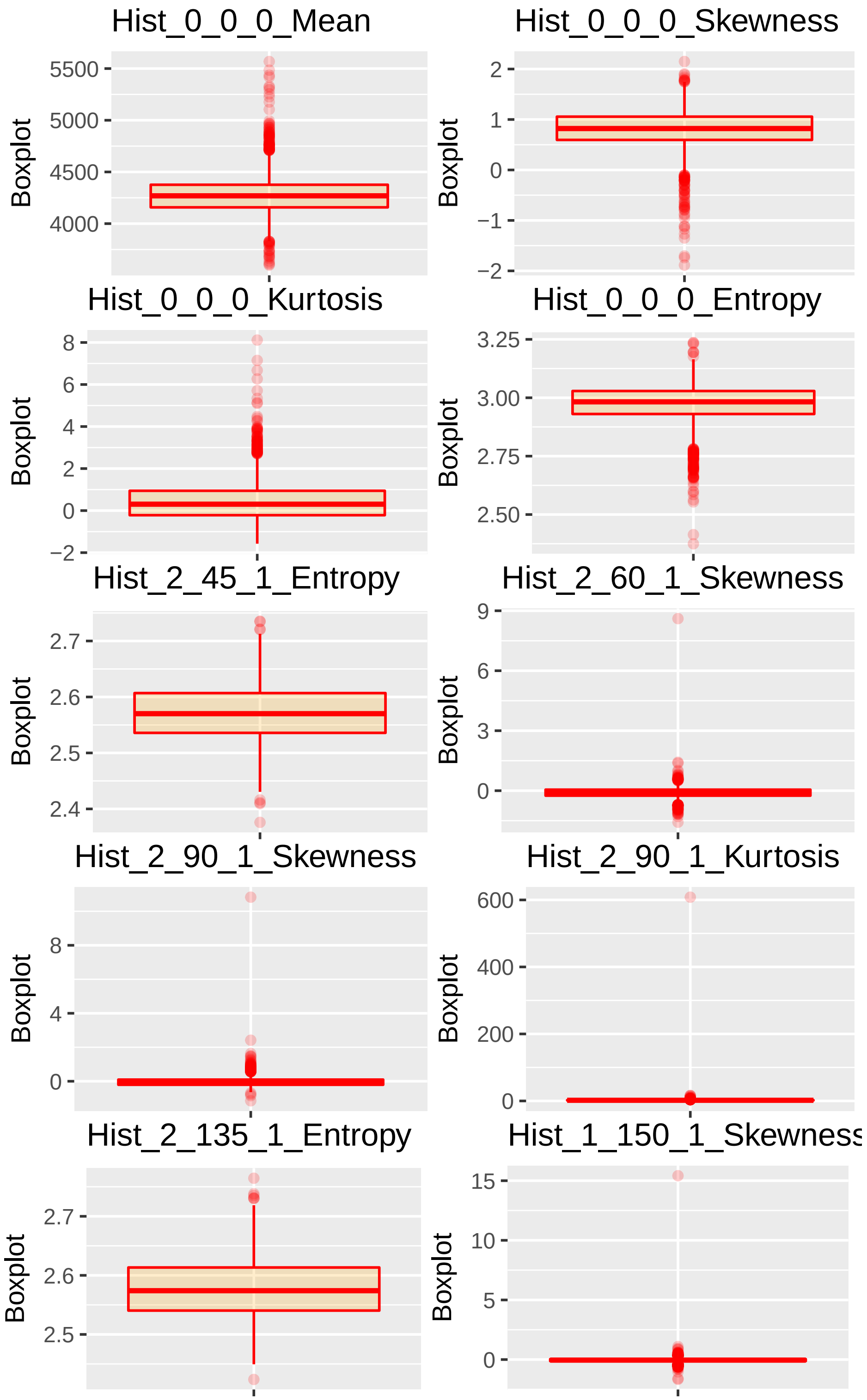
A.2 Histograms for Predictors

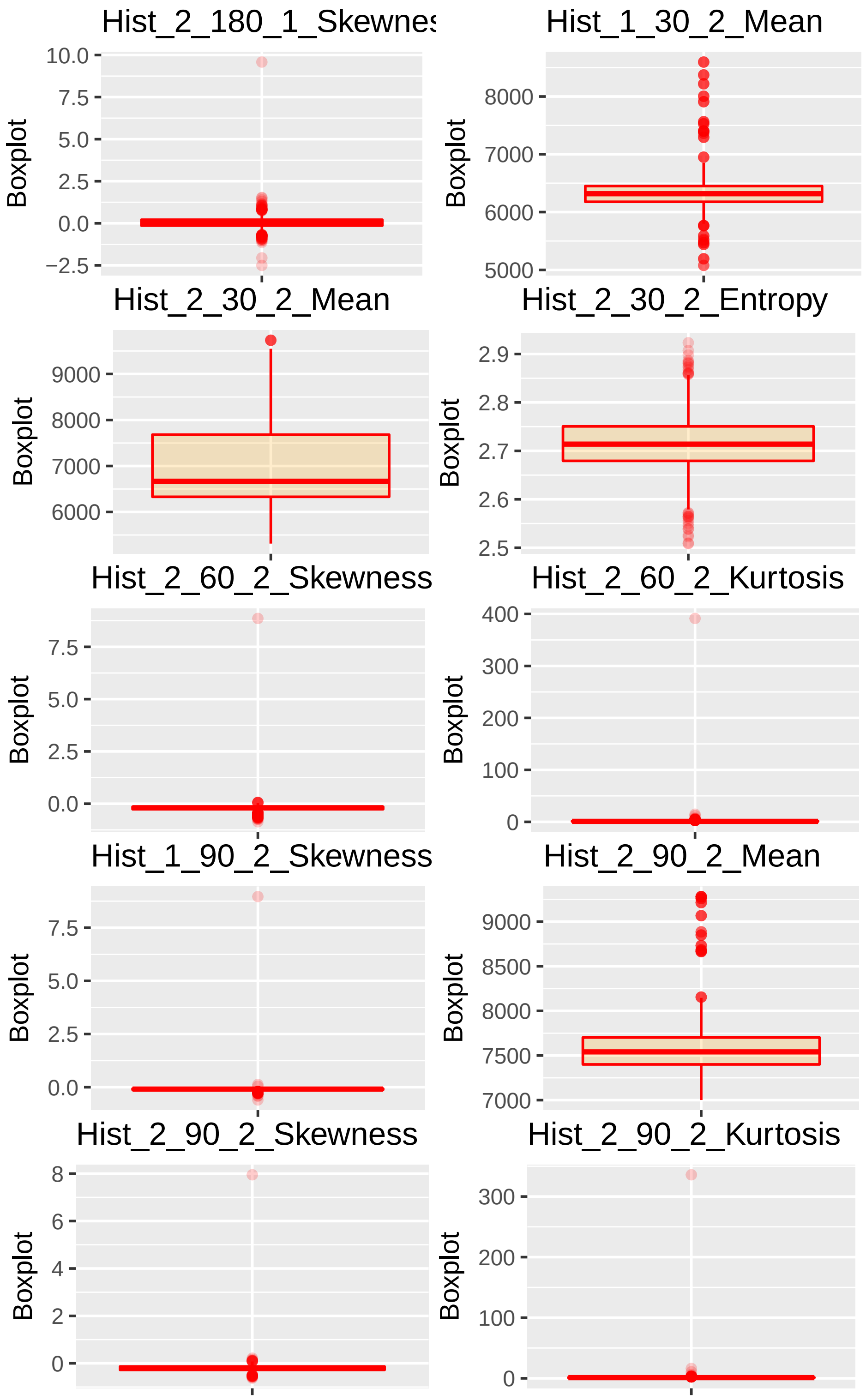



Hist_1_90_2_Skewnes
Frequency
0
0 4 8
Hist_2_90_2_Mean
Frequency
0
7000 8500
Hist_2_90_2_Skewnes
Frequency
0
0 2 4 6 8
Hist_2_90_2_Kurtosis
Frequency
0
0 150 300
Hist_1_120_2_Mean
Frequency
0
3000 6000
Hist_1_135_2_Mean
Frequency
0
4000 6000
Hist_1_135_2_Entropy
Frequency
0
2.50 2.65
Hist_2_150_2_Mean
Frequency
0
6000 8000
Hist_2_150_2_Skewnes
Frequency
0
−1 1 3 5
Hist_2_150_2_Kurtosis
Frequency
0
0 100 200
Hist_2_150_2_Entropy
Frequency
0
2.6 2.8 3.0
Hist_1_180_2_Mean
Frequency
0
6500 7500
Hist_1_180_2_StdDev
Frequency
0
80 120
Hist_1_180_2_Skewnes
Frequency
0
−1 1 3 5
Hist_2_180_2_Mean
Frequency
0
6500 8000
Hist_2_180_2_Skewnes
Frequency
0
−1 1 3

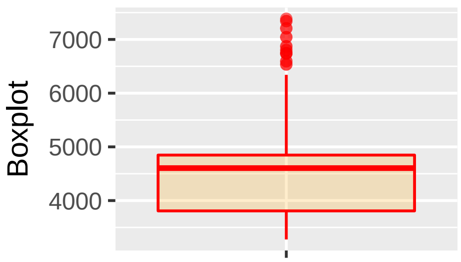

Hist_2_180_2_Kurtosis
Frequency
0
0
50
150
Hist_2_180_2_Entropy
Frequency
0
2.45
2.60
2.75
Matrix_Deg45_Local_Homo
Frequency
0
0.70
0.76
Matrix_Deg90_Local_Homo
Frequency
0
0.78
0.82
latrix_Deg135_Local_Hom
Frequency
0
0.70
0.76
CoMatrix_Deg135_Correla
Frequency
0
0.00
0.06
0.12
CoMatrix_Deg135_Inert
Frequency
0
0.002
0.008

A.3 Box Plots for Predictors

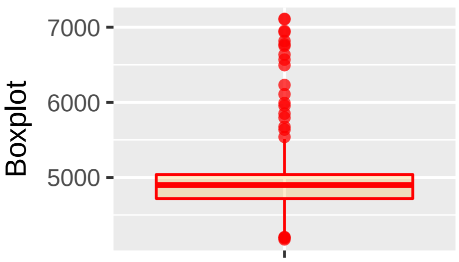



Hist_2_180_1_Skewnes
Boxplot
10.0
7.5
5.0
2.5
0.0
−2.5
Hist_1_30_2_Mean
Boxplot
8000
7000
6000
5000
Hist_2_30_2_Mean
Boxplot
9000
8000
7000
6000
Hist_2_30_2_Entropy
Boxplot
2.9
2.8
2.7
2.6
2.5
Hist_2_60_2_Skewness
Boxplot
7.5
5.0
2.5
0.0
Hist_2_60_2_Kurtosis
Boxplot
400
300
200
100
0
Hist_1_90_2_Skewness
Boxplot
7.5
5.0
2.5
0.0
Hist_2_90_2_Mean
Boxplot
9000
8500
8000
7500
7000
Hist_2_90_2_Skewness
Boxplot
8
6
4
2
0
Hist_2_90_2_Kurtosis
Boxplot
300
200
100
0

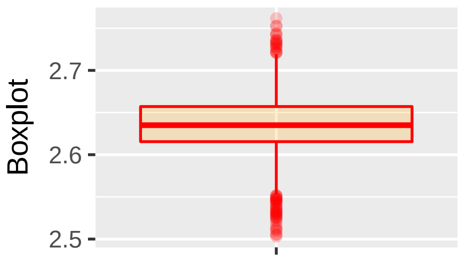
Hist_1_120_2_Mean



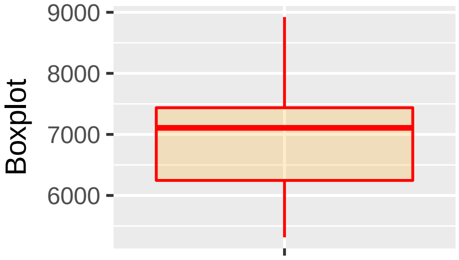
Hist_1_135_2_Mean



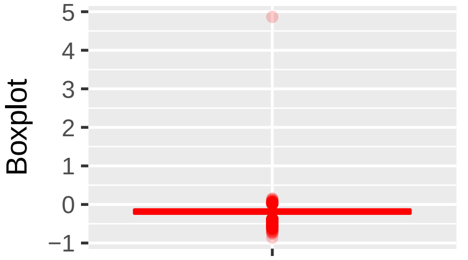
Hist_1_135_2_Entropy



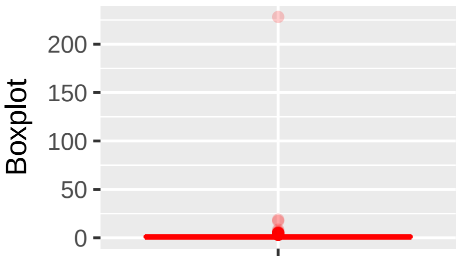
Hist_2_150_2_Mean



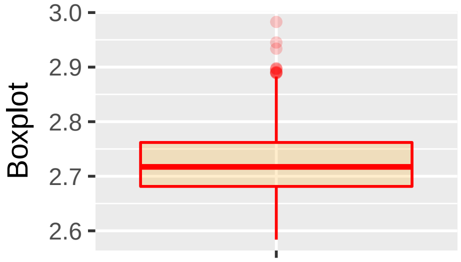
Hist_2_150_2_Skewness



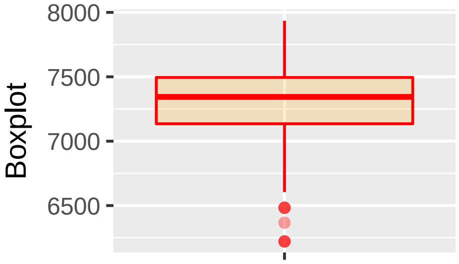
Hist_2_150_2_Kurtosis



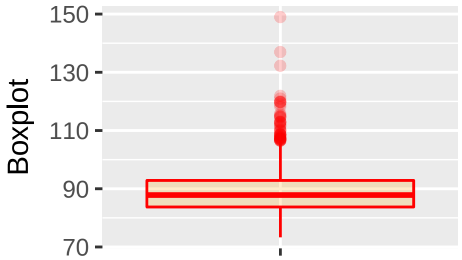
Hist_2_150_2_Entropy



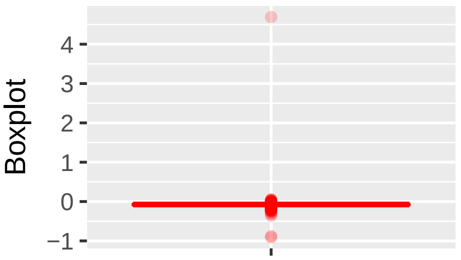
Hist_1_180_2_Mean

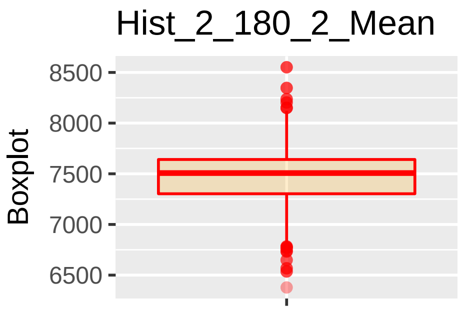


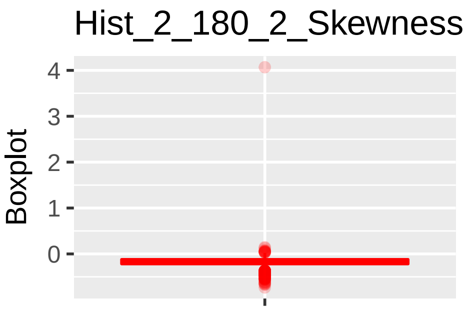
Hist_1_180_2_StdDev

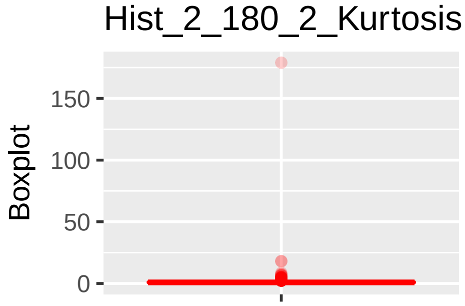


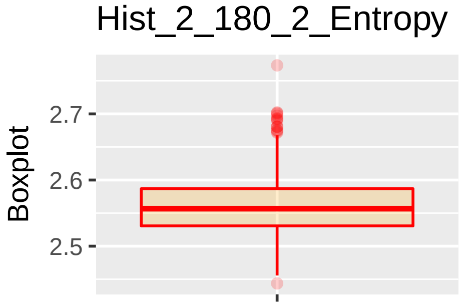
Hist_1_180_2_Skewness

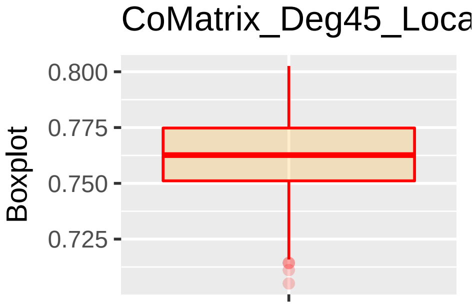


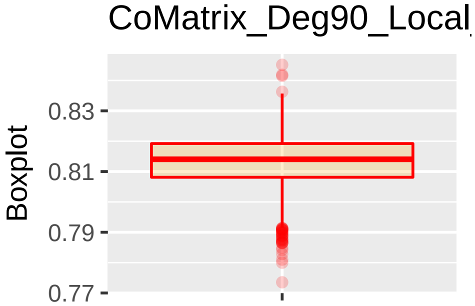

Hist_2_180_2_Mean
Boxplot
8500
8000
7500
7000
6500

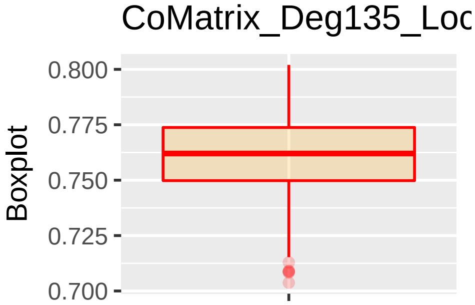

Hist_2_180_2_Skewness
Boxplot
4
3
2
1
0

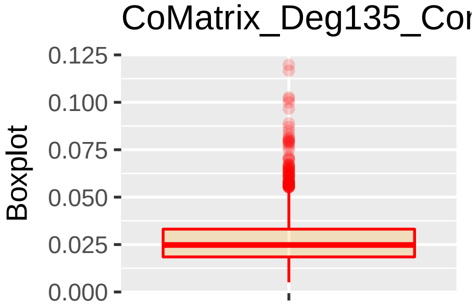

Hist_2_180_2_Kurtosis
Boxplot
150
100
50
0

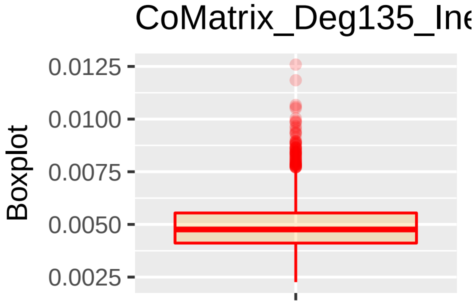

Hist_2_180_2_Entropy
Boxplot
2.7
2.6
2.5


CoMatrix_Deg45_Loca
Boxplot
0.800
0.775
0.750
0.725


CoMatrix_Deg90_Local
Boxplot
0.83
0.81
0.79
0.77


CoMatrix_Deg135_Loc
Boxplot
0.800
0.775
0.750
0.725
0.700


CoMatrix_Deg135_Cor
Boxplot
0.125
0.100
0.075
0.050
0.025
0.000


CoMatrix_Deg135_Ine
Boxplot
0.0125
0.0100
0.0075
0.0050
0.0025

References

American Lung Association. (2018). Pneumoconiosis. Retrieved from http://www.lung.org/lung-health-and-diseases/lung-disease-lookup/pneumoconiosis/

Bronshtein, A. (2017, April). A quick introduction to k-nearest neighbors algorithm. *Medium*. Medium. Retrieved from https://medium.com/@adi.bronshtein/a-quick-introduction-to-k-nearest-neighbors-algorithm-62214cea29c7

Brownlee, J. (2016, September). A gentle introduction to the gradient boosting algorithm for machine learning. *Machine Learning Mastery*. Retrieved from https://machinelearningmastery.com/gentle-introduction-gradient-boosting-algorithm-machine-learning/

Caruana, L. (2017). Black lung case detected in nsw. Retrieved from http://www.miningmonthly.com/coal/safety-and-health/black-lung-case-detected-in-nsw/

Cohen. (n.d.). Kappa in plain english. Cross Validated. Retrieved from https://stats.stackexchange.com/q/82187

Kuhn, M. (n.d.). RFE backwards feature selection. *RDocumentation.org*. Retrieved from https://www.rdocumentation.org/packages/caret/versions/6.0-79/topics/rfe

Ray, S. (2017a, September). 6 easy steps to learn naive bayes algorithm (with code in python). *Analytics Vidhya*. Retrieved from https://www.analyticsvidhya.com/blog/2017/09/naive-bayes-explained/

Ray, S. (2017b, September). Understanding support vector machine algorithm from examples (along with code). *Analytics Vidhya*. Retrieved from https://www.analyticsvidhya.com/blog/2017/09/understaing-support-vector-machine-example-code/

Wang, H. (2014, March). [ML] how sigma matters in svm rbf kernel. *Haohan's Computational Biology Paradise*. Retrieved from http://haohanw.blogspot.com/2014/03/ml-how-sigma-matters-in-svm-rbf-kernel.html

Zhu, W., Zeng, N., & Wang, N. (2010). Sensitivity, specificity, accuracy, associated confidence interval and roc analysis with practical sas® implementations. Retrieved from https://www.lexjansen.com/nesug/nesug10/hl/hl07.pdf