Predicting the Diagnosis of Malignant Mesothelioma

Rob Pruette
Department of Statistical Science, Southern Methodist University
STAT 6302: Experimental Statistics II
Dr. Charles South
May 5, 2020

Introduction

From wicks in lamps and candles, to cloths preserving Egyptian pharaohs, people have long found use for asbestos. Asbestos fibers are malleable and resilient to heat, water, chemicals, and electricity, making asbestos a very useful resource in industries related to automobiles, construction, chemicals, and power (King, s.f.). In the late 1800s, countries all over world were beginning to manufacture asbestos on a large scale. The world demand for asbestos was at its peak in the early 1970s. It could be found in cement, insulation for electric wiring, roofing and flooring compounds, thermal insulation, caulking compounds, paints, and in a variety of other places (King, s.f.).

However, as early as 1897, an Austrian doctor claimed to believe that asbestos exposure was causing pulmonary failure in one of his patients (King, s.f.). Only one year later, London documented the first death caused by asbestos fibers in the lungs. In the late 1970s asbestos production began to decline, and by 2003 there were either full or partial bans in 17 countries across the globe. In 2005, asbestos was banned in all of the European Union. In the United States, there are currently limitations on asbestos exposure though, surprisingly, it is still not banned.

Studies in Germany, South Africa, the United States, and Britain concluded that there was indeed an association between asbestos exposure and cancer. This particular type of cancer is referred to as mesothelioma. Depending on where the tumor is located, mesothelioma can be categorized into four types: pleural, peritoneal, pericardial, and testicular. Pleural mesothelioma, in which the cancer is found in the lungs, is the most common type of mesothelioma. Unfortunately, there is still no cure, however, with treatment, patients are now able to survive mesothelioma longer than ever. In most cases, it could be anywhere from 15 to 70 years after a person has been exposed to asbestos before they are diagnosed with malignant mesothelioma. (Selby, s.f.)

Treatment for mesothelioma has included surgery, chemotherapy, and radiation therapy. However, with such a long latency period, mesothelioma is often difficult to treat by the time it is detected. (Selby, s.f.) Like many other illnesses, patients are often able to survive longer if the cancer is detected early. Early detection of mesothelioma can often be difficult because the symptoms surrounding this cancer mimic those of many other common illnesses. Therefore, finding new and efficient ways to detect this rare cancer is of great importance.

The purpose of this analysis is to use pre-existing data, to determine if a classification model, can better predict malignant mesothelioma.

Data

The data used for this analysis can be found in the UCI Machine Learning repository. The data represent information collected from patients' hospital records from the Dicle University Medical Faculty. There are 34 different features and the outcome variable is a binary variable that represents if the given patient was diagnosed with malignant mesothelioma. There are 324 observations, all representing individual patients. A data dictionary for this data is given below in *Table 1.*

Table 1: Data Dictionary

Variable	Description
age	Age of patient
gender	Male or female
city	Location of patient
asbestos.exposure	Whether patient has been exposed to asbestos
type.of.MM	Type of malignant mesothelioma
duration.of.asbestos.esposure	How long patient was exposed to asbestos (years)
diagnosis.method	Method used in diagnosis
keep.side	Side of lungs experiencing pleural plaques or mesothelioma traces
cytology	Was cytology exam conducted to test fluids samples for mesothelioma cells?
duration.of.symptoms	Time for which patients have shown symptoms (years)
dyspnea	Presence of dyspnea (Shortness of breath)
ache.on.chest	Presence of ache or pain on chest
weakness	Lack of strength
habit.of.ciagarette	Smoking habits
performance.status	Ability to perform normal tasks
white.blood	White blood cell count from blood test (cells/microliter)
cell.count..WBC.	White blood cell count from pleural fluid (cells/microliter)
hemoglobin..HGB.	Hemoglobin test conducted?
platelet.count..PLT.	Average number of platelets in blood (kilo platelets/microliter)
sedimentation	Test that measures how quickly erythrocytes settle (mm/hr)
blood.lactic.dehydrogenise..LDH.	Measure of lactate dehydrogenase in blood (international units/liter)
alkaline.phosphatise..ALP.	Amount of alkaline phosphatase in blood (international units/liter)
total.protein	Total amount of proteins in blood (grams/deciliter)
albumin	Albumin level in blood (grams/deciliter)
glucose	Glucose level in blood (milligrams/deciliter)
pleural.lactic.dehydrogenise	Measure of lactate dehydrogenase in pleural fluid (international units/liter)
pleural.protein	Total amount of proteins in pleural fluid (grams/deciliter)
pleural.albumin	Albumin level in pleural fluid (grams/deciliter)
pleural.glucose	Glucose level in pleural fluid (milligrams/deciliter)
dead.or.not	Is the patient alive?
pleural.effusion	Presence of pleural effusion
pleural.thickness.on.tomography	Presence of any form of thickening on lungs
pleural.level.of.acidity..pH.	Is the pleural fluid pH lower than the normal?
C.reactive.protein..CRP.	Measure of C reactive protein in blood
class.of.diagnosis	Patient diagnosed with malignant mesothelioma?

Statistical Methods

Predicting whether a patient will have mesothelioma is a problem of classification. In this analysis, models were fit using logistic regression, penalized logistic regression, principal component analysis, and clustering methods. All analyses were performed using R.

Before fitting any models, correlations between numeric variables were examined. Correlations greater than 0.6 are shown in *Table 3*.

Table 3: Correlations

Variable 1	Variable 2	Correlation
asbestos.exposure	duration.of.asbestos.exposure	0.7299
pleural.protein	pleural.albumin	0.9114
pleural.protein	pleural.effusion	0.6607
pleural.albumin	pleural.effusion	0.6031
diagnosis.method	class.of.diagnosis	-1.0000

The *diagnosis.method* variable was removed from the data due to its strong correlation with the outcome variable. It can be assumed that if the diagnosis method is known, then the outcome of the diagnosis is known as well.

From the *caret* package, the *nearZeroVar* function was used to identify variables that had few unique values relative to the number of observations and variables that had large ratios of the most common value to the second most common value. The cut off ratio was specified to be 95: 5. The variables *city* and *type.of.MM* were identified to have low variance. Specifically, the *city* levels *5*, *7*, and *8*. Since it is was not clear in the original data what each *city* level represented, cities *5*, *7*, and *8* were simply binned to represent one level. The *type.of.MM* variable was removed from the data.

Due to the relatively small sample size, the data were not separated into training and testing data, therefore, cross validation techniques were on each model. Models were trained using the *train* function in the *caret* package.

As seen in *Table 2*, the outcome variable contains roughly 30% "mesothelioma" diagnoses and 70% "healthy." In efforts to account for this imbalance, optimal cut points were implemented from the *OptimalCutpoints* package when making predictions. In the following section, however, results are reported with and without the use of these cut points.

The first model implemented was a logistic regression model using 10-fold cross validation with 10 repetitions. The second and third models were also a logistic regression models, but utilized penalization. The parameters *alpha* and *lambda* were tuned across the sequences given in *Table 4*.

Table 4: Tuning Parameters

	Begin	End	Length
alpha	0.0	1.0	11
lambda	0.01	0.2	10

One penalization model used the Kappa statistic as the metric, whereas the other model used ROC as the metric. Both models were tuned with the same parameter values and used 10-fold cross validation with 10 repetitions.

The fourth and fifth models trained implemented the use of Principal Component Analysis (PCA). Many variables in the data represented information obtained from blood and pleural fluid tests, and between some of these variables, there were strong correlations.

Therefore, it seemed appropriate to use PCA as a way to represent this information in a concise and uncorrelated way. The variables transformed are given explicitly in *Table 5*.

Table 5: PCA Variables

white.blood	cell.count..WBC.
platelet.count..PLT.	sedimentation
blood.lactic.dehydrogenise..LDH.	alkaline.phosphatise..ALP.
total.protein	albumin
glucose	pleural.lactic.dehydrogenise
pleural.protein	pleural.albumin
pleural.glucose	C.reactive.protein..CRP.

The PCA loadings were then used in logistic regression models. As before, both models used 10-fold cross validation with 10 repetitions and one model implemented the use of penalization. The tuning values for the penalized model are the same as those in *Table 4*.

Finally, the last model fit used PAM clustering. Gower's distance was used due to the large number of categorical variables. Once the ideal number of clusters to be used was determined, a new variable was created that assigned each observation to a cluster. With this the addition of this new variable, another logistic regression model was fit using 10-fold cross validation.

Results

Summary statistics are provided for all the variables in the dataset in *Table 2*. For skewed variables the medians are provided.

Without imposing any statistical model, by assuming every patient to be healthy, one would achieve a prediction accuracy of 70.37%. Although this would correctly predict all the patients diagnosed as healthy, it would incorrectly predict all the patients who have mesothelioma. None of the models achieve an accuracy that greatly surpasses the no-information rate, however, they do offer the advantage of being able to more accurately predict patients with mesothelioma. All models predicted the probability of a mesothelioma diagnosis.

Table 2: Summary Statistics

Variable	%	Mean (Median)
Age		54.74
Gender		
Male	58.64%	
Female	41.36%	
City		
0	30.86%	
1	12.96%	
2	15.74%	
3	7.72%	
4	7.41%	
5	0.62%	
6	20.37%	
7	4.01%	
8	0.31%	
Asbestos Exposure		
Yes	86.42%	
No	13.58%	
Type of MM		
0	95.68%	
1	3.40%	
2	0.93%	
Duration of Asbestos Exposure		30.19
Diagnosis Method		
0	29.63%	
1	70.37%	
Keep Side		
0	30.86%	
1	62.35%	
2	6.79%	
Cytology		
No	68.83%	
Yes	28.09%	
Duration of Symptoms		(5.00)
Dyspnea		
No	18.21%	
Yes	81.79%	
Ache on Chest		
No	31.79%	
Yes	68.21%	
Weakness		
No	38.89%	
Yes	61.11%	
Habit of Cigarette		
0	56.48%	
1	11.42%	
2	10.67%	
3	15.43%	
Performance Status		
No	47.84%	
Yes	52.16%	
White Blood Count		9457.45
White Blood Count (Pleural)		9.56
Hemoglobin Test		
No	57.72%	
Yes	42.28%	
Platelet Count		(345.00)
Sedimentation		70.69
Blood Lactic Dehydrogenase		(234.50)
Alkaline Phosphatase		66.16
Total Protein		6.58
Albumin		3.30
Glucose		112.41
Pleural Lactic Dehydrogenase		(510.00)
Pleural Protein		3.94
Pleural Albumin		2.08
Pleural Glucose		48.44
Dead or Not		
Dead	5.56%	
Alive	94.44%	
Pleural Effusion		
No	12.96%	
Yes	87.04%	
Pleural Thickness on Tomography		
No	40.43%	
Yes	59.57%	
Pleural Level of Acidity pH		
No	47.84%	
Yes	52.16%	
C Reactive Protein		64.19
Class of Diagnosis		
Healthy	70.37%	
Mesothelioma	29.63%	

The results of the first model that implemented the use of cross validated logistic regression are given in *Table 6* below.

Table 6: CV Logistic Regression Model Results

	Kappa	Accuracy	Sensitivity	Specificity	AUC	OCP[1]
CV Logistic Regression	0.2087	0.6945	0.7551	0.5024	0.8045	Default
	0.2191	0.6741	0.7684	0.4507		0.3995

[1] Optimal cut point

The 95% DeLong confidence interval for the AUC value is [0.7522, 0.8568].

Table 7 summarizes results from the two models that used penalized logistic regression. Since the ROC metric already employs the use of different cut points, there was not an optimal cut point used for that model.

Table 7: Penalized Logistic Regression Models Results

	Kappa	Accuracy	Sensitivity	Specificity	AUC	OCP
Penalized Logistic (ROC)	-	0.7115	0.9421	0.1629	0.7426	-
Penalized Logistic (Kappa)	0.1258	0.6930	0.8947	0.2138	0.8045	Default
	0.1414	0.6653	0.8082	0.3258		0.4172

The 95% DeLong confidence interval for the AUC for the ROC model is [0.6820, 0.8033] and for the Kappa model [0.6928, 0.8119]. The optimal tuning parameters for the ROC model were *alpha* = 1 and *lambda* = 0.01; for the Kappa model, *alpha* = 0.2 and *lambda* = 0.01. After penalization, some variables were reduced to zero. *Table 8* provides the coefficients resulting from these models.

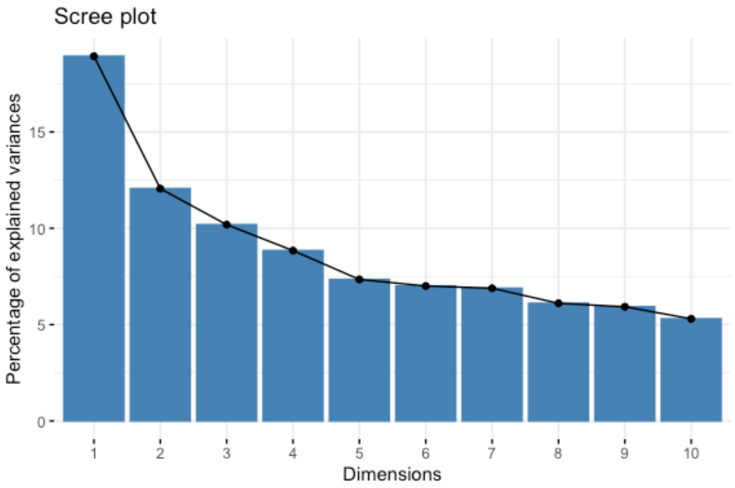
Table 8: Model Coefficients

Variable	ROC	Kappa
age	-0.3742	-0.4659
gender	-0.3472	-0.4193
city	0.1235	0.1802
asbestos.exposure	.	0.0741
duration.of.asbestos.esposure	0.3768	0.5415
keep.side	0.2982	0.3707
cytology	.	-0.0336
duration.of.symptoms	0.2109	0.2639
dyspnea	.	.
ache.on.chest	-0.0623	-0.1206
weakness	0.0610	0.1467
habit.of.ciagarette	.	0.0854
performance.status	0.0242	0.1192
white.blood	-0.0002	-0.0606
cell.count..WBC.	-0.1053	-0.1874
hemoglobin..HGB.	0.0614	0.1162
platelet.count..PLT.	-0.2714	-0.3705
sedimentation	0.0303	0.0867
blood.lactic.dehydrogenise..LDH.	.	0.0447
alkaline.phosphatise..ALP.	.	-0.0152

total.protein	0.0012	0.0356
albumin	0.0286	0.0608
glucose	0.0176	0.0952
pleural.lactic.dehydrogenise	.	-0.0367
pleural.protein	-0.1425	-0.1787
pleural.albumin	-0.0186	-0.0713
pleural.glucose	-0.0904	-0.1889
dead.or.not	.	0.0207
pleural.effusion	.	0.0587
pleural.thickness.on.tomography	.	0.0620
pleural.level.of.acidity..pH.	-0.1551	-0.2491
C.reactive.protein..CRP.	0.2369	0.2548

The two models that incorporated the use of Principal Component Analysis, were trained using all the loadings from the PCA results. *Figure 1* only shows the first ten dimensions, however, there were a total of 14 loadings.

Figure 1



The results from the logistic regression models with the PCA variables are given in *Table 9*.

Table 9: PCA Models

	Kappa	Accuracy	Sensitivity	Specificity	AUC	OCP
CV Logistic Reg. PCA	0.1893	0.6901	0.8385	0.3368	0.8045	Default
	0.1911	0.6622	0.7594	0.4309		0.3995
Penalized Logistic Reg. PCA	0.1383	0.6895	0.8789	0.2398	0.7544	Default
	0.1279	0.6533	0.7888	0.3310		0.4205

The 95% DeLong confidence interval for the AUC for the cross validated logistic regression model is [0.7522, 0.8568] and the penalized model [0.6952, 0.8136]. The best tuning parameters for the penalized model were *alpha* = 0 and *lambda* = 0.01. After penalization, all of the variables remained in the model.

Finally, the last model employed the use of clustering analysis. As seen in *Figure 2* below, the recommended number of clusters was two. Using the *Rtsne* package, *Figure 3* shows that by using dimension reduction techniques, separation of the observations can be clearly seen.

Figure 2

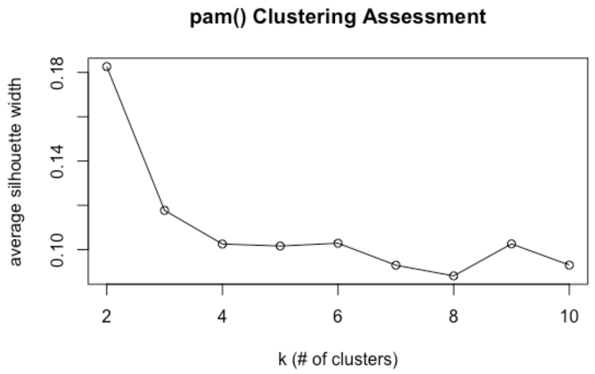
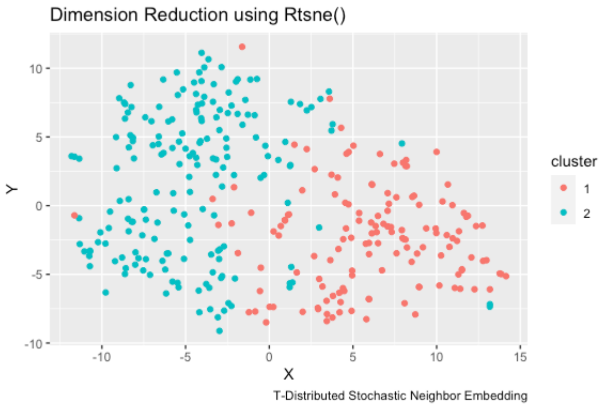


Figure 3



After cluster assignments were introduced as a new variable in the original data, the cross validated regression model provided the results in *Table 10*.

Table 10

	Kappa	Accuracy	Sensitivity	Specificity	AUC	OCP
CV Logistic Regression with Clustering	0.2511	0.7102	0.7660	0.5348	0.8166	Default
	0.2600	0.6913	0.7819	0.4780		0.4086

The 95% DeLong confidence interval for the AUC is [0.7655, 0.8677].

As the specified metric of model performance, Figure *4* provides a comparison of all the models using the Kappa statistic. The highest value was that of the logistic regression model that used clustering analysis (0.26), followed by logistic regression model only used the original data. *Figure 5*, similarly, compares the overall accuracies of each model. The model that achieved the highest accuracy was the penalized regression model that used ROC as the metric (0.7115).

Figure 4

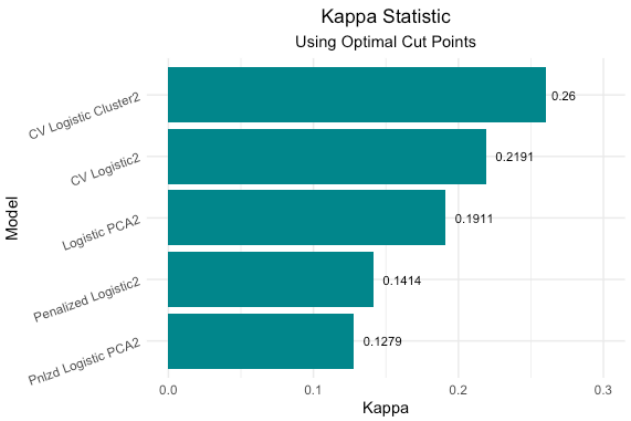
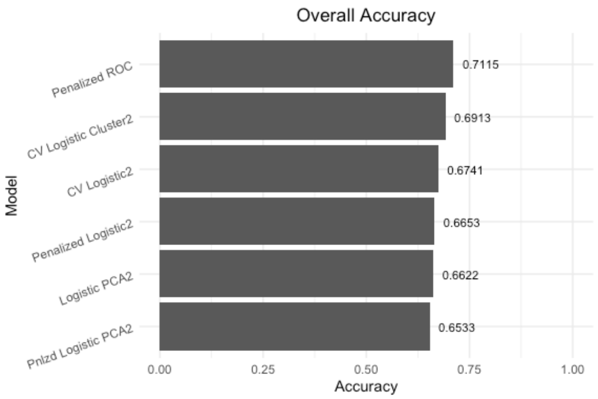
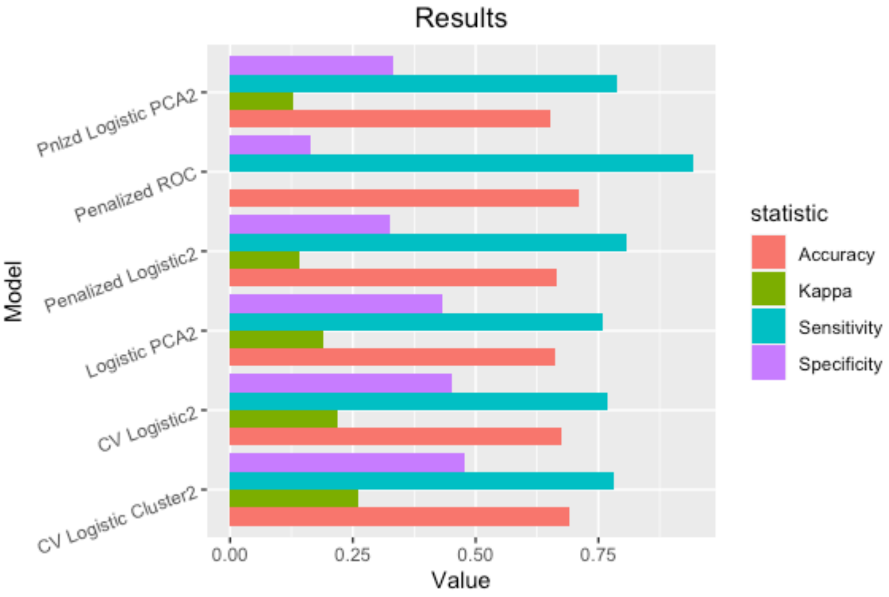


Figure 5



As an advantage to implementing a statistical model, it is also important to examine the sensitivity and specificity of these models. The results given in *Figure 6* are those after the use of optimal cut points. The penalized ROC model did result in the highest accuracy, however, its specificity is lower than all the other models.

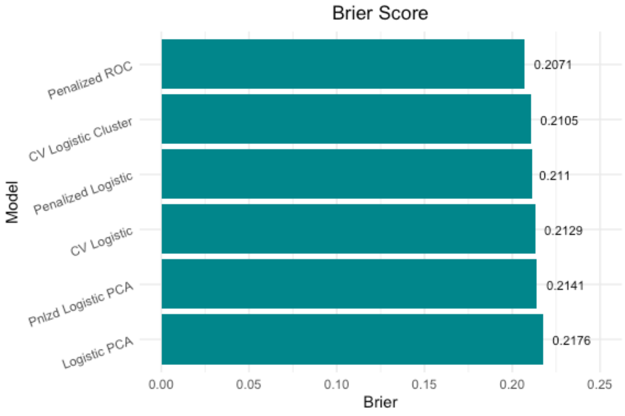
Figure 6

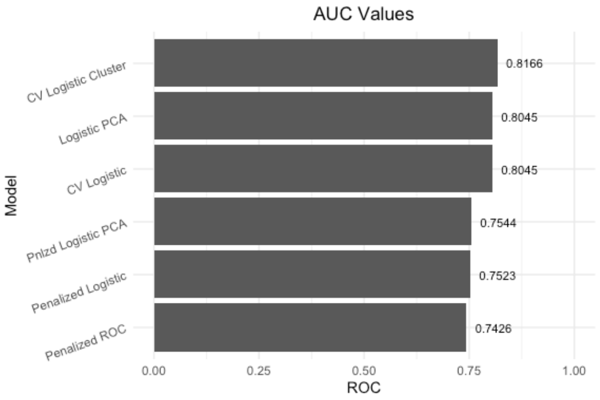


Finally, as the last ways to compare the models in this analysis, *Figure 7* and *Figure 8* show resulting Brier scores and AUC values from the models. These are helpful metrics to consider additionally, because they do not depend on the optimal cut points used in the analysis.

Figure 7

Figure 8





The penalized logistic regression model using ROC as the metric results in the best Brier score, but has the lowest AUC value. However, the logistic model that incorporates clustering, performs well by both metrics. The calibration plots provide additional evidence that the model which uses clustering, is perhaps, the best model. Although several models could be considered "well-calibrated," the clustering model most tightly hugs the diagonal line.

Figure 9

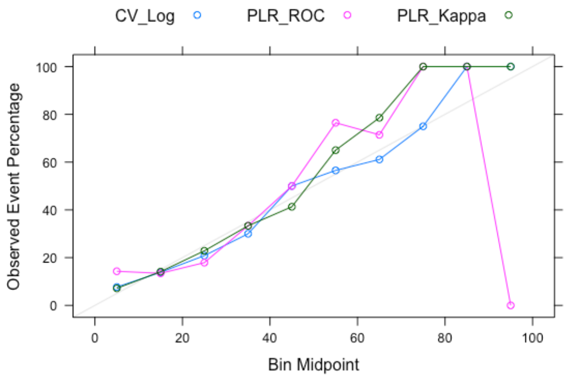
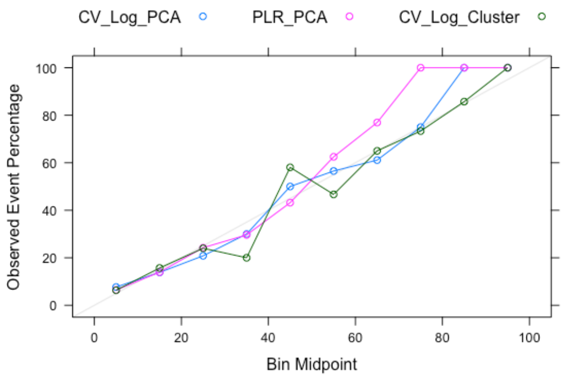


Figure 10



Conclusion and Future Research

Based on the results, the logistic regression model that incorporated the use of clustering analysis, performed best on this data. The benefits of this model are that it can better predict patients who have malignant mesothelioma, compared to the *no information* predictions, and it provides information to those in the medical field that is easily interpretable. *Table 11* provides the multiplicative effects each variable has on the odds of being diagnosed with mesothelioma. For example, for each year increase in asbestos exposure (*duration.of.asbestos.exposure*), the odds of being diagnosed with mesothelioma increase by 4.86%.

Table 11

Variable	*Coefficient*
asbestos.exposure	0.3019
gender	0.3746
pleural.level.of.acidity..pH.	0.4843
city7	0.5156
city1	0.5465
ache.on.chest	0.6784
dead.or.not	0.6879
dyspnoea	0.8424
cytology	0.8599
pleural.albumin	0.8690
pleural.protein	0.8759
cell.count..WBC.	0.9320
age	0.9486
pleural.glucose	0.9901
habit.of.cigarette3	0.9972
platelet.count..PLT.	0.9977
blood.lactic.dehydrogenise..LDH.	0.9999
white.blood	0.9999
pleural.lactic.dehydrogenise	1.0000
alkaline.phosphatise..ALP.	1.0009
glucose	1.0032
sedimentation	1.0036
C.reactive.protein..CRP.	1.0103
keep.side1	1.0193
albumin	1.0330
duration.of.asbestos.exposure	1.0486
habit.of.cigarette1	1.0501
total.protein	1.0524
duration.of.symptoms	1.0800
pleural.thickness.on.tomography	1.1082
performance.status	1.4802
city6	1.4944
weakness	1.6882
pleural.effusion	1.9252
hemoglobin..HGB.	1.9580
city4	2.5362
habit.of.cigarette2	2.5551
city2	3.5052
(Intercept)	6.3973
city3	6.4730
keep.side2	2.9849

As early detection of malignant mesothelioma is so difficult, the results of this analysis lead to insight as to how different predictors affect the odds of a positive diagnosis. This could give professionals a better idea of which symptoms most affect a diagnosis and could hopefully lead to quicker detection of cancer.

Although all models in this analysis were fit using cross validation, which is robust to small samples, it would be beneficial to further test these models on more data. Since nearly all of the models are implementing the use of optimal cut points, in an ideal scenario, additional data would be held out to determine these values. However, with only 324 observations, setting aside data for training, testing, and cut points is not feasible. Further collection of data could also allow the implementation of machine learning methods which may have the ability to more accurately predict mesothelioma diagnoses.

Bibliography

King, D. (n.d.). *History of Asbestos*. Retrieved from Asbestos.com: www.asbestos.com/asbestos/history/

Selby, K. (n.d.). *The History of Mesothelioma*. Retrieved from Asbestos.com: www.asbestos.com/mesothelioma/history/

Appendix:

```
library(caret)
library(ggplot2)
library(corrplot)
library(pROC)
library(Hmisc)
library(tidyr)
library(OptimalCutpoints)
library(glmnet)
library(factoextra)
library(cluster)
library(dplyr)


# Read in Data
meso <- read.csv("/Users/rob.pruette/Documents/SMU Spring 2020/STAT 6302/Final
Project/MesotheliomaData.csv")
# 34 features, 1 outcome, and 324 observations
dim(meso)
summary(meso)

# There is a correlation of -1 between diagnosis method and the diagnosis outcome
cor(meso$diagnosis.method, meso$class.of.diagnosis)


# Histograms of Skewed Variables
########################################
ggplot(data = meso, aes(x = city)) +
  geom_histogram(bins = 10) +
  ggtitle("City")

ggplot(data = meso, aes(x = asbestos.exposure)) +
  geom_histogram(bins = 3) +
  ggtitle("Asbestos Exposure")

ggplot(data = meso, aes(x = type.of.MM)) +
  geom_histogram(bins = 4) + ggtitle("Type of MM")

ggplot(data = meso, aes(x = duration.of.asbestos.exposure)) +
  geom_histogram(bins = 20) + ggtitle("Duration of Asbestos Exposure")

ggplot(data = meso, aes(x = cytology)) +
  geom_histogram(bins = 3) + ggtitle("Cytology")

ggplot(data = meso, aes(x = duration.of.symptoms)) +
  geom_histogram(bins = 30) + ggtitle("Duration of Symptoms")

ggplot(data = meso, aes(x = dyspnoea)) +
  geom_histogram(bins = 3) + ggtitle("Dyspnoea")

ggplot(data = meso, aes(x = ache.on.chest)) +
  geom_histogram(bins = 3) + ggtitle("Ache on Chest")
```

```
ggplot(data = meso, aes(x = platelet.count..PLT.)) +
  geom_histogram(bins = 30) + ggtitle("Platelet Count PLT")

ggplot(data = meso, aes(x = blood.lactic.dehydrogenise..LDH.)) +
  geom_histogram(bins = 30) + ggtitle("Blood Lactic Dehydrogenise LDH")

ggplot(data = meso, aes(x = alkaline.phosphatise..ALP.)) +
  geom_histogram(bins = 30) + ggtitle("Alkaline Phosphatise ALP")

ggplot(data = meso, aes(x = pleural.lactic.dehydrogenise)) +
  geom_histogram(bins = 30) + ggtitle("Pleural Lactic Dehydrogenise")

ggplot(data = meso, aes(x = dead.or.not)) +
  geom_histogram(bins = 3) + ggtitle("Dead or Alive")

ggplot(data = meso, aes(x = pleural.effusion)) +
  geom_histogram(bins = 3) + ggtitle("Pleural Effusion")

#########################################

# Function that makes a nice matrix with all the correlations
flattenCorrMatrix <- function(cormat, pmat) {
  ut <- upper.tri(cormat)
  data.frame(
    row = rownames(cormat)[row(cormat)[ut]],
    column = rownames(cormat)[col(cormat)[ut]],
    cor  =(cormat)[ut],
    p = pmat[ut]
  )
}
# correlations between all variables
res2 <- rcorr(as.matrix(meso))
correlation.matrix <- flattenCorrMatrix(res2$r, res2$P)

# correlations greater than 0.5
correlation.matrix[which(abs(correlation.matrix$cor) > 0.6),]

# correlations with outcome variable
correlation.matrix[which(correlation.matrix$column == "class.of.diagnosis"),]

# Create new outcome variable that is binary 0, 1
meso$diagnosis_label <- ifelse(meso$class.of.diagnosis == 2, "Mesothelioma", "Healthy")

# The following variables are numeric, but they represent factors
# Change the variable to factors for analysis
meso$city <- as.factor(meso$city)
meso$keep.side <- as.factor(meso$keep.side)
meso$habit.of.cigarette <- as.factor(meso$habit.of.cigarette)

meso_dummy <- model.matrix(diagnosis_label ~ ., data = meso)[,-1]
nearZeroVar(meso_dummy, freqCut = 95/5, saveMetrics = FALSE, names = TRUE)
table(meso$city)
```

```
# New data set with variables removed
remove.indices <- which(colnames(meso) == "diagnosis.method" | colnames(meso) == "class.of.diagnosis" |
colnames(meso) == "type.of.MM")
meso2 <- meso[,-remove.indices]

# bin the city variable so that cities 5, 7, and 8 are one level
table(meso2$city)
meso2$city <- as.numeric(meso2$city) - 1
table(meso2$city)
meso2[which(meso2$city == 5 | meso2$city == 8),]$city <- 7
meso2$city <- factor(meso2$city)
table(meso2$city)

# Make the outcome variable a factor
meso2$diagnosis_label <- as.factor(meso2$diagnosis_label)

# Create a dataset with dummy variables to see what columns have low variance
# City 7 still has low variance, but combining it with another level seems questionable
meso2_dummy <- model.matrix(diagnosis_label ~ ., data = meso2)[,-1]
nearZeroVar(meso2_dummy, freqCut = 95/5, saveMetrics = FALSE, names = TRUE)

# No information rate
no_info_rate <- length(which(meso2$diagnosis_label == "Healthy")) / nrow(meso2)
no_info_rate

###########################################################################
# Cross Validated Logistic Regression (Model 1)
###########################################################################
set.seed(256)
logisticRegCV <- train(diagnosis_label ~ ., data = meso2,
                method = "glm", trControl = trainControl(method = "repeatedcv",
                                        number = 10,
                                        repeats = 10, savePredictions = TRUE,
                                        classProbs = TRUE))
as.data.frame(coef(logisticRegCV$finalModel))
# Results report an accuracy of 0.6945 and a Kappa statistics of 0.2087
model1_accuracy <- logisticRegCV$results$Accuracy
model1_kappa<- logisticRegCV$results$Kappa

# This loop using the confusion matrix from the model output to calculate sensitivity and specificity.
# Accuracy is also calculated and results in the same value as the model results ouput
logisticRegCV_accuracy <- array()
logisticRegCV_sens <- array()
logisticRegCV_spec <- array()
for (i in 1:100){
  logisticRegCV_accuracy[i] <- (logisticRegCV$resampledCM[i,1] + logisticRegCV$resampledCM[i,4]) /
sum(logisticRegCV$resampledCM[i,1:4])
  logisticRegCV_sens[i] <- logisticRegCV$resampledCM[i,1] / sum(logisticRegCV$resampledCM[i,c(1,3)])
  logisticRegCV_spec[i] <- logisticRegCV$resampledCM[i,4] / sum(logisticRegCV$resampledCM[i,c(2,4)])
}
# Accuracy
```

```
mean(logisticRegCV_accuracy)
# Sensitivity
model1_sensitivity <- mean(logisticRegCV_sens)
# Specificity
model1_specificity <- mean(logisticRegCV_spec)

# Identify an optimal cutpoint
m1_pred_df <- data.frame(logisticRegCV$pred)
head(m1_pred_df)

m1_preds <- data.frame(prob = predict(logisticRegCV, type = "prob")[,2])
m1_preds$pred <- predict(logisticRegCV)
m1_preds$obs <- meso2$diagnosis_label
head(m1_preds)
optcut0 <- summary(optimal.cutpoints(X = "prob", status = "obs", data = m1_preds,
                                     tag.healthy = "Healthy", methods = "MaxKappa"))
final_cut0 <- optcut0$MaxKappa$Global$optimal.cutoff$cutoff
final_cut0
m1_pred_df$new_pred_label <- as.factor(ifelse(m1_pred_df$Mesothelioma > final_cut0, "Mesothelioma",
"Healthy"))

# Create binary variables to use in the calculation of the Brier score
m1_pred_df$brier <- ifelse(m1_pred_df$obs == "Healthy", 0, 1)
head(m1_pred_df)

# Calculate the brier score for model 1
brier_empty <- array()
for (i in 1:nrow(m1_pred_df)){
  brier_empty[i] <- (m1_pred_df$Mesothelioma[i] - m1_pred_df$brier[i])**2
  }
model1_brier <- mean(brier_empty)
model1_brier


# Calculate the accuracy, kappa, sensitivity, and specificity again using the new cutpoint
rep_accuracy0 <- array()
rep_sens0 <- array()
rep_spec0 <- array()
folds0 <- list()
kappa0 <- array()
count0 <- 1
for (j in 1:10){
  for(i in 1:10){
    if(j<10 & i<10){
      locator <- paste("Fold0", as.character(j), ".Rep0", as.character(i), sep = "")
    }else if(j >= 10 & i < 10){
      locator <- paste("Fold", as.character(j), ".Rep0", as.character(i), sep = "")
    }else if(j<10 & i >= 10){
      locator <- paste("Fold0", as.character(j), ".Rep", as.character(i), sep = "")
    }else{
      locator <- paste("Fold", as.character(j), ".Rep", as.character(i), sep = "")
    }
    df <- as.data.frame(m1_pred_df[which(m1_pred_df$Resample == locator), ])
    CM <- confusionMatrix(data = df$new_pred_label, reference = df$obs)
    rep_accuracy0[count0] <- CM$overall["Accuracy"]
    rep_sens0[count0] <- CM$byClass["Sensitivity"]
    rep_spec0[count0] <- CM$byClass["Specificity"]
    kappa0[count0] <- CM$overall["Kappa"]
    count0 = count0 +1
    folds0[count0] <- locator
  }
}
model1.1_accuracy <- mean(rep_accuracy0)
model1.1_sensitivity <- mean(rep_sens0)
model1.1_specificity <- mean(rep_spec0)
model1.1_kappa <- mean(kappa0)


# Using all the predicted data from the cross validation, examine calibration plot and ROC plot

# Calibration plot
calData <- calibration(obs ~ Mesothelioma, data = m1_pred_df, cuts = 10, class = "Mesothelioma")
xyplot(calData, auto.key = list(columns = 2))

# ROC Plot
mesoROC1 <- roc(m1_preds$obs, m1_preds$prob, class = "Mesothelioma")
model1_auc <- auc(mesoROC1)
model1_ci <- ci.auc(mesoROC1)
plot(mesoROC1, legacy.axes = TRUE)



##############################################################
# Penalized Logistic Regression, ROC Method (Model 2)
##############################################################

glmnGrid <- expand.grid(alpha = seq(0, 1, length = 11),
                        lambda = seq(0.01, 0.2, length = 10))
ctrl <- trainControl(method = "repeatedcv",
                     summaryFunction = twoClassSummary,
                     classProbs = TRUE,
                     repeats=10,
                     savePredictions = TRUE)
set.seed(546)
glmnFit <- train(x = data.matrix(meso2[, -c(which(colnames(meso2) == "diagnosis_label"))]),
                 y = meso2[, c(which(colnames(meso2) == "diagnosis_label"))],
                 method = "glmnet",
                 tuneGrid = glmnGrid,
                 metric = "ROC",
                 preProc = c("center", "scale"),
                 family = "binomial",
                 trControl = ctrl)
glmnFit$bestTune
# ROC value 0.6102
model2_roc <- mean(glmnFit$resample["ROC"][,1])
# Sensitivity 0.9421
```

```
model2_sensitivity <- mean(glmnFit$resample["Sens"][, 1])
# Specificity 0.1629
model2_specificity <- mean(glmnFit$resample["Spec"][, 1])


# The loop below allows me to get the accuracy for the model (even though it isn't necessary)
# the specificity and sensitivity match the model output
rep_accuracy <- array()
rep_sens <- array()
rep_spec <- array()
folds <- list()
count <- 1
for (j in 1:10){
  for(i in 1:10){
    if(j<10 & i<10){
      locator <- paste("Fold0", as.character(j), ".Rep0", as.character(i), sep = "")
    }else if(j >= 10 & i < 10){
      locator <- paste("Fold", as.character(j), ".Rep0", as.character(i), sep = "")
    }else if(j<10 & i >= 10){
      locator <- paste("Fold0", as.character(j), ".Rep", as.character(i), sep = "")
    }else{
      locator <- paste("Fold", as.character(j), ".Rep", as.character(i), sep = "")
    }

    df <- as.data.frame(glmnFit$pred[which(glmnFit$pred$alpha == glmnFit$bestTune$alpha & glmnFit$pred$lambda == glmnFit$bestTune$lambda & glmnFit$pred$Resample == locator),])
    CM <- confusionMatrix(data = df$pred, reference = df$obs)
    rep_accuracy[count] <- CM$overall["Accuracy"]
    rep_sens[count] <- CM$byClass["Sensitivity"]
    rep_spec[count] <- CM$byClass["Specificity"]
    count = count +1
    folds[count] <- locator
  }
}
coef(glmnFit$finalModel, s = glmnFit$bestTune$lambda)
model2_accuracy <- mean(rep_accuracy)

mean(rep_sens)
mean(rep_spec)

# Calculate the brier score
brier_data <- data.frame(glmnFit$pred)
brier_data$brier <- ifelse(brier_data$obs == "Healthy", 0, 1)
brier_empty3 <- array()
for (i in 1:nrow(brier_data)){
  brier_empty3[i] <- (brier_data$Mesothelioma[i] - brier_data$brier[i])**2
}
model2_brier <- mean(brier_empty3)

mesoROC2 <- roc(meso2$diagnosis_label, predict(glmnFit, type = "prob")[,2], class = "Mesothelioma")
model2_auc <- auc(mesoROC2)
model2_ci <- ci.auc(mesoROC2)
plot(mesoROC2, legacy.axes = TRUE)
```

```
############################################################
# Penalized Logistic Regression, Kappa Method (Model 3)
############################################################

ctrl2 <- trainControl(method = "repeatedcv",
                      classProbs = TRUE,
                      repeats=10,
                      savePredictions = TRUE)
set.seed(344)
glmnFit2 <- train(x = data.matrix(meso2[, -c(which(colnames(meso2) == "diagnosis_label"))]),
                  y = meso2[, c(which(colnames(meso2) == "diagnosis_label"))],
                  method = "glmnet",
                  tuneGrid = glmnGrid,
                  metric = "Kappa",
                  preProc = c("center", "scale"),
                  family = "binomial",
                  trControl = ctrl2)

glmnFit2$bestTune
mean(glmnFit2$resample$Accuracy)

coef(glmnFit2$finalModel, s=glmnFit2$bestTune$lambda)

# Determine the accuracy, sensitivity, specificity, and kappa of the model
rep_accuracy2 <- array()
rep_sens2 <- array()
rep_spec2 <- array()
rep_kappa2 <- array()
folds2 <- list()
count2 <- 1
for (j in 1:10){
  for(i in 1:10){
    if(j<10 & i<10){
      locator <- paste("Fold0", as.character(j), ".Rep0", as.character(i), sep = "")
    }else if(j >= 10 & i < 10){
      locator <- paste("Fold", as.character(j), ".Rep0", as.character(i), sep = "")
    }else if(j<10 & i >= 10){
      locator <- paste("Fold0", as.character(j), ".Rep", as.character(i), sep = "")
    }else{
      locator <- paste("Fold", as.character(j), ".Rep", as.character(i), sep = "")
    }

    df <- as.data.frame(glmnFit2$pred[which(glmnFit2$pred$alpha == glmnFit2$bestTune$alpha &
glmnFit2$pred$lambda == glmnFit2$bestTune$lambda & glmnFit2$pred$Resample == locator),])
    CM <- confusionMatrix(data = df$pred, reference = df$obs)
    rep_accuracy2[count2] <- CM$overall["Accuracy"]
    rep_kappa2[count2] <- CM$overall["Kappa"]
    rep_sens2[count2] <- CM$byClass["Sensitivity"]
    rep_spec2[count2] <- CM$byClass["Specificity"]
    count2 = count2 +1
    folds2[count2] <- locator
  }
}
# Accuracy (matches the output of the model)
model3_accuracy <- mean(rep_accuracy2)
# Sensitivity
model3_sensitivity <- mean(rep_sens2)
# Specificity
model3_specificity <- mean(rep_spec2)
# Kappa (matches the output of the model)
model3_kappa <- mean(rep_kappa2)


# Find optimal cutpoint
model3_preds <- data.frame(prob = predict(glmnFit2, type = "prob")[,2])
model3_preds$pred <- predict(glmnFit2)
model3_preds$obs <- meso2$diagnosis_label
model3_preds

cut_data <- data.frame(glmnFit2$pred[which(glmnFit2$pred$alpha == glmnFit2$bestTune$alpha &
glmnFit2$pred$lambda == glmnFit2$bestTune$lambda),])

optcut1 <- summary(optimal.cutpoints(X = "prob", status = "obs", data = model3_preds,
                                     tag.healthy = "Healthy", methods = "MaxKappa"))
final_cut1 <- optcut1$MaxKappa$Global$optimal.cutoff$cutoff
final_cut1

# Create new variables for the calculation of the brier statistic
cut_data$new_pred_label <- as.factor(ifelse(cut_data$Mesothelioma > final_cut1, "Mesothelioma",
"Healthy"))
cut_data$brier <- ifelse(cut_data$obs == "Healthy", 0, 1)
head(cut_data)

# Calculate brier statistics for both cutpoints
brier_empty4 <- array()
for (i in 1:nrow(cut_data)){
  brier_empty4[i] <- (cut_data$Mesothelioma[i] - cut_data$brier[i])**2
}
model3_brier <- mean(brier_empty4)

# Calculate accuracy, kappa, sensitivity, and specificity using new cutpoint
rep_accuracy3 <- array()
rep_sens3 <- array()
rep_spec3 <- array()
kappa3 <- array()
folds3 <- list()
count3 <- 1
for (j in 1:10){
  for(i in 1:10){
    if(j<10 & i<10){
      locator <- paste("Fold0", as.character(j), ".Rep0", as.character(i), sep = "")
    }else if(j >= 10 & i < 10){
      locator <- paste("Fold", as.character(j), ".Rep0", as.character(i), sep = "")
    }else if(j<10 & i >= 10){
      locator <- paste("Fold0", as.character(j), ".Rep", as.character(i), sep = "")
    }else{
      locator <- paste("Fold", as.character(j), ".Rep", as.character(i), sep = "")
    }
    df <- as.data.frame(cut_data[which(cut_data$Resample == locator), ])
    CM <- confusionMatrix(data = df$new_pred_label, reference = df$obs)
    rep_accuracy3[count3] <- CM$overall["Accuracy"]
    rep_sens3[count3] <- CM$byClass["Sensitivity"]
    rep_spec3[count3] <- CM$byClass["Specificity"]
    kappa3[count3] <- CM$overall["Kappa"]
    count3 <- count3 +1
    folds3[count3] <- locator
  }
}
# Accuracy
model3.1_accuracy <- mean(rep_accuracy3)
# Sensitivity
model3.1_sensitivity <- mean(rep_sens3)
# Specificity
model3.1_specificity <- mean(rep_spec3)
# Kappa
model3.1_kappa <- mean(kappa3)

mesoROC3 <- roc(model3_preds$obs, model3_preds$prob, class = "Mesothelioma")
model3_auc <- auc(mesoROC3)
model3_ci <- ci.auc(mesoROC3)
plot(mesoROC3, legacy.axes = TRUE)

##############################################################
# Principal Component Analysis
##############################################################

# Numeric variables to be used in PCA
testvars <- meso2[, c(14:15, 17:27, 32)]
colnames(testvars)
testsPCA <- prcomp(testvars, center = T, scale = T)
summary(testsPCA)

# Scree plot
fviz_eig(testsPCA)

# Loadings
testsPCA$rotation

# Graphical representation of components in the first two dimensions
# Not seeing any separation
fviz_pca_ind(testsPCA, label = "none", habillage = meso2$diagnosis_label,
        addEllipses = TRUE, ellipse.level = 0.95, palette = "Dark1", axes = c(2,6))
# Join original variables and the variables from PCA
pca_data <- data.frame(meso2[, -c(14:15, 17:27, 32)], testsPCA$x[, 1:14])


##############################################################
```

```
# Cross Validated Logistic Regression, with PCA variables (Model 4)
##############################################################
set.seed(221)
logisticRegCV_PCA <- train(diagnosis_label ~ ., data = pca_data,
                        method = "glm", trControl = trainControl(method = "repeatedcv",
                                                                number = 10,
                                                                classProbs = TRUE,
                                                                savePredictions = TRUE,
                                                                repeats = 10))
logisticRegCV_PCA$resample
logisticRegCV_PCA$results

cv_pca_df <- data.frame(logisticRegCV_PCA$pred)
head(cv_pca_df)
dim(cv_pca_df)
# Find optimal cutpoint

m4_preds <- data.frame(prob = predict(logisticRegCV_PCA, type = "prob")[,2])
m4_preds$pred <- predict(logisticRegCV_PCA)
m4_preds$obs <- meso2$diagnosis_label
head(m4_preds)


optcut_pca <- summary(optimal.cutpoints(X = "prob", status = "obs", data = m4_preds,
                                      tag.healthy = "Healthy", methods = "MaxKappa"))
final_cut_pca <- optcut_pca$MaxKappa$Global$optimal.cutoff$cutoff
final_cut_pca
cv_pca_df$new_pred_label <- as.factor(ifelse(cv_pca_df$Mesothelioma > final_cut_pca, "Mesothelioma",
"Healthy"))

# Create new variables for brier score calculation
cv_pca_df$brier <- ifelse(cv_pca_df$obs == "Healthy", 0, 1)
head(cv_pca_df)


rep_accuracy_cvPCA <- array()
rep_sens_cvPCA <- array()
rep_spec_cvPCA <- array()
kappa_cvPCA <- array()
count_cvPCA <- 1
for (j in 1:10){
  for(i in 1:10){
    if(j<10 & i<10){
      locator <- paste("Fold0", as.character(j), ".Rep0", as.character(i), sep = "")
    }else if(j >= 10 & i < 10){
      locator <- paste("Fold", as.character(j), ".Rep0", as.character(i), sep = "")
    }else if(j<10 & i >= 10){
      locator <- paste("Fold0", as.character(j), ".Rep", as.character(i), sep = "")
    }else{
      locator <- paste("Fold", as.character(j), ".Rep", as.character(i), sep = "")
    }

    df <- as.data.frame(cv_pca_df[which(cv_pca_df$Resample == locator),])
    CM <- confusionMatrix(data = df$pred, reference = df$obs)
    rep_accuracy_cvPCA[count_cvPCA] <- CM$overall["Accuracy"]
    rep_sens_cvPCA[count_cvPCA] <- CM$byClass["Sensitivity"]
    rep_spec_cvPCA[count_cvPCA] <- CM$byClass["Specificity"]
    kappa_cvPCA[count_cvPCA] <- CM$overall["Kappa"]
    count_cvPCA = count_cvPCA +1
  }
}
model4_accuracy <- mean(rep_accuracy_cvPCA)
model4_sensitivity <- mean(rep_sens_cvPCA)
model4_specificity <- mean(rep_spec_cvPCA)
model4_kappa <- mean(kappa_cvPCA)


brier_empty_pca_oldcut <- array()
for (i in 1:nrow(cv_pca_df)){
  brier_empty_pca_oldcut[i] <- (cv_pca_df$Mesothelioma[i] - cv_pca_df$brier[i])**2
}
model4_brier <- mean(brier_empty_pca_oldcut)




rep_accuracy_cvPCA_cut <- array()
rep_sens_cvPCA_cut <- array()
rep_spec_cvPCA_cut <- array()
kappa_cvPCA_cut <- array()
count_cvPCA_cut <- 1
for (j in 1:10){
  for(i in 1:10){
    if(j<10 & i<10){
      locator <- paste("Fold0", as.character(j), ".Rep0", as.character(i), sep = "")
    }else if(j >= 10 & i < 10){
      locator <- paste("Fold", as.character(j), ".Rep0", as.character(i), sep = "")
    }else if(j<10 & i >= 10){
      locator <- paste("Fold0", as.character(j), ".Rep", as.character(i), sep = "")
    }else{
      locator <- paste("Fold", as.character(j), ".Rep", as.character(i), sep = "")
    }

    df <- as.data.frame(cv_pca_df[which(cv_pca_df$Resample == locator),])
    CM <- confusionMatrix(data = df$new_pred_label, reference = df$obs)
    rep_accuracy_cvPCA_cut[count_cvPCA_cut] <- CM$overall["Accuracy"]
    rep_sens_cvPCA_cut[count_cvPCA_cut] <- CM$byClass["Sensitivity"]
    rep_spec_cvPCA_cut[count_cvPCA_cut] <- CM$byClass["Specificity"]
    kappa_cvPCA_cut[count_cvPCA_cut] <- CM$overall["Kappa"]
    count_cvPCA_cut = count_cvPCA_cut +1
    }
}
model4.1_accuracy <- mean(rep_accuracy_cvPCA_cut)
model4.1_sensitivity <- mean(rep_sens_cvPCA_cut)
model4.1_specificity <- mean(rep_spec_cvPCA_cut)
model4.1_kappa <- mean(kappa_cvPCA_cut)
```

```
mesoROC4 <- roc(m4_preds$obs, m4_preds$prob, class = "Mesothelioma")
model4_auc <- auc(mesoROC4)
model4_ci <- ci.auc(mesoROC4)
plot(mesoROC4, legacy.axes = TRUE)

##############################################################
# Penalized Regression, Kappa Metric, with PCA variables (Model 5)
##############################################################
set.seed(843)
glmnFit2_PCA <- train(x = data.matrix(pca_data[,-which(colnames(pca_data)== "diagnosis_label"),]),
          y = pca_data[, which(colnames(pca_data) == "diagnosis_label")],
          method = "glmnet",
          tuneGrid = glmnGrid,
          metric = "Kappa",
          preProc = c("center", "scale"),
          family = "binomial",
          trControl = ctrl2)


glmnFit2_PCA$results
coef(glmnFit2_PCA$finalModel, s=glmnFit2_PCA$bestTune$lambda)
glmnFit2_PCA$finalModel$tuneValue
glmnFit2_PCA$bestTune

# Create data set that has cv predictions with best tuning parameters
pen_pca_data <- data.frame(glmnFit2_PCA$pred[which(glmnFit2_PCA$pred$alpha ==
glmnFit2_PCA$bestTune$alpha & glmnFit2_PCA$pred$lambda == glmnFit2_PCA$bestTune$lambda),])
head(pen_pca_data)

# Find the accuracy, kappa, sensitivity, and specificity of model
rep_accuracy_pen_pca <- array()
rep_sens_pen_pca <- array()
rep_spec_pen_pca <- array()
kappa_pen_pca <- array()
count_pen_pca <- 1
for (j in 1:10){
 for(i in 1:10){
  if(j<10 & i<10){
   locator <- paste("Fold0", as.character(j), ".Rep0", as.character(i), sep = "")
  }else if(j >= 10 & i < 10){
   locator <- paste("Fold", as.character(j), ".Rep0", as.character(i), sep = "")
  }else if(j<10 & i >= 10){
   locator <- paste("Fold0", as.character(j), ".Rep", as.character(i), sep = "")
  }else{
   locator <- paste("Fold", as.character(j), ".Rep", as.character(i), sep = "")
  }

  df <- as.data.frame(pen_pca_data[which(pen_pca_data$Resample == locator),])
  CM <- confusionMatrix(data = df$pred, reference = df$obs)
  rep_accuracy_pen_pca[count_pen_pca] <- CM$overall["Accuracy"]
  rep_sens_pen_pca[count_pen_pca] <- CM$byClass["Sensitivity"]
  rep_spec_pen_pca[count_pen_pca] <- CM$byClass["Specificity"]
```

```
    kappa_pen_pca[count_pen_pca] <- CM$overall["Kappa"]
    count_pen_pca = count_pen_pca +1
  }
}
model5_accuracy <- mean(rep_accuracy_pen_pca)
model5_sensitivity <- mean(rep_sens_pen_pca)
model5_specificity <- mean(rep_spec_pen_pca)
model5_kappa <- mean(kappa_pen_pca)


# Find optimal cutpoint
model5_preds <- data.frame(prob = predict(glmnFit2_PCA, type = "prob")[,2])
model5_preds$pred <- predict(glmnFit2_PCA)
model5_preds$obs <- meso2$diagnosis_label
model5_preds


optcut_pen_pca <- summary(optimal.cutpoints(X = "prob", status = "obs", data = model5_preds,
                                    tag.healthy = "Healthy", methods = "MaxKappa"))
final_cut_pen_pca <- optcut_pen_pca$MaxKappa$Global$optimal.cutoff$cutoff
final_cut_pen_pca

# Create new prediction outcome
pen_pca_data$new_pred_label <- as.factor(ifelse(pen_pca_data$Mesothelioma > final_cut_pen_pca,
"Mesothelioma", "Healthy"))

# Create new variables for Brier calculation
pen_pca_data$brier <- ifelse(pen_pca_data$obs == "Healthy", 0, 1)

# Calculate brier scores
brier_empty_pen_pca_old <- array()
for (i in 1:nrow(pen_pca_data)){
  brier_empty_pen_pca_old[i] <- (pen_pca_data$Mesothelioma[i] - pen_pca_data$brier[i])**2
}
model5_brier <- mean(brier_empty_pen_pca_old)

# Accuracy, sensitivity, specificity and kappa for new cutpoint
rep_accuracy_pen_pca_cut <- array()
rep_sens_pen_pca_cut <- array()
rep_spec_pen_pca_cut <- array()
kappa_pen_pca_cut <- array()
count_pen_pca_cut <- 1
for (j in 1:10){
  for(i in 1:10){
    if(j<10 & i<10){
      locator <- paste("Fold0", as.character(j), ".Rep0", as.character(i), sep = "")
    }else if(j >= 10 & i < 10){
      locator <- paste("Fold", as.character(j), ".Rep0", as.character(i), sep = "")
    }else if(j<10 & i >= 10){
      locator <- paste("Fold0", as.character(j), ".Rep", as.character(i), sep = "")
    }else{
      locator <- paste("Fold", as.character(j), ".Rep", as.character(i), sep = "")
    }
    df <- as.data.frame(pen_pca_data[which(pen_pca_data$Resample == locator),])
    CM <- confusionMatrix(data = df$new_pred_label, reference = df$obs)
    rep_accuracy_pen_pca_cut[count_pen_pca_cut] <- CM$overall["Accuracy"]
    rep_sens_pen_pca_cut[count_pen_pca_cut] <- CM$byClass["Sensitivity"]
    rep_spec_pen_pca_cut[count_pen_pca_cut] <- CM$byClass["Specificity"]
    kappa_pen_pca_cut[count_pen_pca_cut] <- CM$overall["Kappa"]
    count_pen_pca_cut = count_pen_pca_cut +1
  }
}
model5.1_accuracy <- mean(rep_accuracy_pen_pca_cut)
model5.1_sensitivity <- mean(rep_sens_pen_pca_cut)
model5.1_specificity <- mean(rep_spec_pen_pca_cut)
model5.1_kappa <- mean(kappa_pen_pca_cut)

mesoROC5 <- roc(model5_preds$obs, model5_preds$prob, class = "Mesothelioma")
model5_auc <- auc(mesoROC5)
model5_ci <- ci.auc(mesoROC5)
plot(mesoROC5, legacy.axes = TRUE)


#################################################################
# Clustering
#################################################################

# PAM
gower.meso <- daisy(meso2[,-33], metric = "gower")
gower.matrix <- as.matrix(gower.meso)

# Most similar patients
meso2[which(gower.matrix == min(gower.matrix[gower.matrix != min(gower.matrix)]), arr.ind = TRUE)[1,],]

# Most dissimilar clients
meso2[which(gower.matrix == max(gower.matrix[gower.matrix != min(gower.matrix)]), arr.ind =
TRUE)[1,],]

asw <- numeric(0)
for (k in 1:9){
  asw[k] <- pam(gower.meso, k+1)$silinfo$avg.width
}
k.best <- which.max(asw)
cat("silhouette-optimal number of clusters:", k.best +1, "\n")
plot(2:10, asw, type = "o", main = "pam() Clustering Assessment",
     xlab = "k (# of clusters)", ylab = "average silhouette width")
axis(1, k.best, paste("best", k.best, sep = "\n"), col = "red", col.axis = "red")


k <- 2
pam_fit <- pam(gower.meso, diss = TRUE, k)
pam_results <- meso2 %>%
  mutate(cluster = pam_fit$clustering) %>%
  group_by(cluster) %>%
  do(the_summary = summary(.))
```

```
pam_results$the_summary




pam.meso2 <- pam(meso2[, -33], k=2)

##Cluster visualization
fviz_cluster(object = pam.meso2,
            data=meso2.cluster,
            ellipse.type = "convex",
            palette = "jco",
            geom = "point",
            repel = TRUE,
            ggtheme = theme_bw(),
            axis = c(2,3) )

library(Rtsne)
tsne_obj <- Rtsne(gower.meso, is_distance = TRUE)
tsne_data <- tsne_obj$Y %>%
  data.frame() %>%
  setNames(c("X", "Y")) %>%
  mutate(cluster = factor(pam_fit$clustering))

ggplot(aes(x = X, y = Y), data = tsne_data) +
  geom_point(aes(color = cluster)) +
  labs(title = "Dimension Reduction using Rtsne()", caption = "T-Distributed Stochastic Neighbor
Embedding")


meso2_cluster <- data.frame(meso2)
colnames(meso2_cluster)
meso2_cluster$cluster <- pam_fit$clustering

############################################################
# CV Logistic Regression with Cluster Variable (Model 6)
############################################################
set.seed(256)
logisticCV.cluster <- train(diagnosis_label ~ ., data = meso2_cluster,
                      method = "glm", trControl = trainControl(method = "repeatedcv",
                                                          number = 10,
                                                          repeats = 10,
                                                          savePredictions = TRUE,
                                                          classProbs = TRUE))


model6_kappa <- logisticCV.cluster$results$Kappa



logistic_cluster_accuracy <- array()
logistic_cluster_sens <- array()
```

```
logistic_cluster_spec <- array()
for (i in 1:100){
  logistic_cluster_accuracy[i] <- (logisticCV.cluster$resampledCM[i,1] +
logisticCV.cluster$resampledCM[i,4]) / sum(logisticCV.cluster$resampledCM[i,1:4])
  logistic_cluster_sens[i] <- logisticCV.cluster$resampledCM[i,1] /
sum(logisticCV.cluster$resampledCM[i,c(1,3)])
  logistic_cluster_spec[i] <- logisticCV.cluster$resampledCM[i,4] /
sum(logisticCV.cluster$resampledCM[i,c(2,4)])
}
model6_accuracy <- mean(logistic_cluster_accuracy)
model6_sensitivity <- mean(logistic_cluster_sens)
model6_specificity <- mean(logistic_cluster_spec)


cluster.df <- data.frame(logisticCV.cluster$pred)
head(cluster.df)

m6_preds <- data.frame(prob = predict(logisticCV.cluster, type = "prob")[,2])
m6_preds$pred <- predict(logisticCV.cluster)
m6_preds$obs <- meso2$diagnosis_label
head(m6_preds)
optcut.clust <- summary(optimal.cutpoints(X = "prob", status = "obs", data = m6_preds,
                                    tag.healthy = "Healthy", methods = "MaxKappa"))
final_cut_cluster <- optcut.clust$MaxKappa$Global$optimal.cutoff$cutoff
final_cut_cluster
cluster.df$new_pred_label <- as.factor(ifelse(cluster.df$Mesothelioma > final_cut_cluster, "Mesothelioma",
"Healthy"))
cluster.df$brier <- ifelse(cluster.df$obs == "Healthy", 0, 1)
head(cluster.df)

brier_empty_cluster <- array()
for (i in 1:nrow(cluster.df)){
  brier_empty_cluster[i] <- (cluster.df$Mesothelioma[i] - cluster.df$brier[i])**2
}
model6_brier <- mean(brier_empty_cluster)




rep_accuracy_clust <- array()
rep_sens_clust <- array()
rep_spec_clust <- array()
folds_clust <- list()
kappa_clust <- array()
count_clust <- 1
for (j in 1:10){
  for(i in 1:10){
    if(j<10 & i<10){
      locator <- paste("Fold0", as.character(j), ".Rep0", as.character(i), sep = "")
    }else if(j >= 10 & i < 10){
      locator <- paste("Fold", as.character(j), ".Rep0", as.character(i), sep = "")
    }else if(j<10 & i >= 10){
      locator <- paste("Fold0", as.character(j), ".Rep", as.character(i), sep = "")
    }else{
      locator <- paste("Fold", as.character(j), ".Rep", as.character(i), sep = "")
    }
    df <- as.data.frame(cluster.df[which(cluster.df$Resample == locator), ])
    CM <- confusionMatrix(data = df$new_pred_label, reference = df$obs)
    rep_accuracy_clust[count_clust] <- CM$overall["Accuracy"]
    rep_sens_clust[count_clust] <- CM$byClass["Sensitivity"]
    rep_spec_clust[count_clust] <- CM$byClass["Specificity"]
    kappa_clust[count_clust] <- CM$overall["Kappa"]
    count_clust = count_clust +1
    folds_clust[count_clust] <- locator
  }
}
model6.1_accuracy <- mean(rep_accuracy_clust)
model6.1_sensitivity <- mean(rep_sens_clust)
model6.1_specificity <- mean(rep_spec_clust)
model6.1_kappa <- mean(kappa_clust)

mesoROC6 <- roc(m6_preds$obs, m6_preds$prob, class = "Mesothelioma")
model6_auc <- auc(mesoROC6)
model6_ci <- ci.auc(mesoROC6)
plot(mesoROC6, legacy.axes = TRUE)



# Predictions to create calibration plots
model1_pred <- data.frame(predict(logisticRegCV, newdata = meso2[,-33], type = "prob"))
model2_pred <- data.frame(predict(glmnFit, testX = meso2[, -33], type = "prob"))
model3_pred <- data.frame(predict(glmnFit2, testX = meso2[, -33], type = "prob"))
model4_pred <- data.frame(predict(logisticRegCV_PCA, newdata = pca_data[,-which(colnames(pca_data)==
"diagnosis_label"),], type = "prob"))
model5_pred <- data.frame(predict(glmnFit2_PCA, testX = pca_data[,-which(colnames(pca_data)==
"diagnosis_label"),], type = "prob"))
model6_pred <- data.frame(predict(logisticCV.cluster, newdata = meso2_cluster, type = "prob"))

test_models <- data.frame("CV_Log" = model1_pred$Mesothelioma, "PLR_ROC"=
model2_pred$Mesothelioma, "PLR_Kappa"=model3_pred$Mesothelioma, "Diagnosis" =
meso2$diagnosis_label)
test_models2 <- data.frame("CV_Log_PCA" = model4_pred$Mesothelioma, "PLR_PCA" =
model5_pred$Mesothelioma, "CV_Log_Cluster" = model6_pred$Mesothelioma, "Diagnosis" =
meso2$diagnosis_label)

cal.models1 <- calibration(Diagnosis ~ CV_Log + PLR_ROC + PLR_Kappa, data = test_models, cuts = 10,
class = "Mesothelioma")
cal.models2 <- calibration(Diagnosis ~ CV_Log_PCA + PLR_PCA + CV_Log_Cluster, data = test_models2,
cuts = 10, class = "Mesothelioma")
xyplot(cal.models1, auto.key = list(columns = 3))
xyplot(cal.models2, auto.key = list(columns = 3))



model1_vals <- c(model1_accuracy, model1_sensitivity, model1_specificity, model1_kappa, model1_brier)
```

```
model1.1_vals <- c(model1.1_accuracy, model1.1_sensitivity, model1.1_specificity, model1.1_kappa,
model1_brier)
model2_vals <- c(model2_accuracy, model2_sensitivity, model2_specificity, NA, model2_brier)
model3_vals <- c(model3_accuracy, model3_sensitivity, model3_specificity, model3_kappa, model3_brier)
model3.1_vals <- c(model3.1_accuracy, model3.1_sensitivity, model3.1_specificity, model3.1_kappa,
model3_brier)
model4_vals <- c(model4_accuracy, model4_sensitivity, model4_specificity, model4_kappa, model4_brier)
model4.1_vals <- c(model4.1_accuracy, model4.1_sensitivity, model4.1_specificity, model4.1_kappa,
model4_brier)
model5_vals <- c(model5_accuracy, model5_sensitivity, model5_specificity, model5_kappa, model5_brier)
model5.1_vals <- c(model5.1_accuracy, model5.1_sensitivity, model5.1_specificity, model5.1_kappa,
model5_brier)
model6_vals <- c(model6_accuracy, model6_sensitivity, model6_specificity, model6_kappa, model6_brier)
model6.1_vals <- c(model6.1_accuracy, model6.1_sensitivity, model6.1_specificity, model6.1_kappa,
model6_brier)

final.matrix <- matrix(c(model1_vals, model1.1_vals,  model2_vals, model3_vals, model3.1_vals,
model4_vals, model4.1_vals, model5_vals, model5.1_vals, model6_vals, model6.1_vals), ncol = 5, byrow =
T)
colnames(final.matrix) <- c("Accuracy", "Sensitivity", "Specificity", "Kappa", "Brier")
rownames(final.matrix) <- c("CV Logistic", "CV Logistic2", "Penalized ROC", "Penalized Logistic",
                      "Penalized Logistic2", "Logistic PCA", "Logistic PCA2", "Pnlzd Logistic PCA",
                      "Pnlzd Logistic PCA2", "CV Logistic Cluster", "CV Logistic Cluster2")
final.df <- as.data.frame(final.matrix)
final.df$Model <- rownames(final.matrix)
final.df$OptCutPoint <- c("No", "Yes", "No", "No", "Yes", "No", "Yes", "No", "Yes", "No", "Yes")
final.df

final.df_OCP <- final.df[c(2,5,7,9,11),]

ggplot(data = final.df[which(final.df$OptCutPoint == "Yes"),], aes(x = reorder(Model, Kappa), y = Kappa))+
  geom_bar(stat = "identity", position = "dodge", fill = "turquoise4") +
  geom_text(aes(label = round(Kappa, 4)), position = position_dodge(width = 0.2), hjust = -0.25, size = 3) +
  coord_flip() +
  theme_minimal() +
  ylim(0,0.3) +
  xlab("Model") +
  theme(axis.text.y = element_text(angle = 20, hjust = 1)) +
  theme(plot.title = element_text(hjust = 0.5), plot.subtitle = element_text(hjust = 0.5)) +
  labs(title = "Kappa Statistic", subtitle = "Using Optimal Cut Points")

final.df
ggplot(data = final.df[which(final.df$OptCutPoint == "Yes" | final.df$Model == "Penalized ROC"),], aes(x =
reorder(Model, Accuracy), y = Accuracy))+
  geom_bar(stat = "identity", position = "dodge") +
  geom_text(aes(label = round(Accuracy, 4)), position = position_dodge(width = 0.2), hjust = -0.25, size = 3) +
  coord_flip() +
  theme_minimal() +
  ylim(0,1) +
  xlab("Model") +
  theme(axis.text.y = element_text(angle = 20, hjust = 1)) +
  theme(plot.title = element_text(hjust = 0.5)) +
  labs(title = "Overall Accuracy")
```

```
gather.df <- gather(final.df, key = "statistic", value = "value", Accuracy, Sensitivity, Specificity, Kappa, Brier,
OptCutPoint)
gather.df2 <- gather.df[-which(gather.df$statistic == "OptCutPoint" | gather.df$statistic == "Brier"),]
gather.df2 <- gather.df2[-which(gather.df2$Model == "Pnlzd Logistic PCA" | gather.df2$Model == "Penalized
Logistic" | gather.df2$Model == "Logistic PCA" | gather.df2$Model == "CV Logistic Cluster" |
gather.df2$Model == "CV Logistic"),]

ggplot(gather.df2, aes(fill = statistic, x = Model, y = as.numeric(value)))+
  geom_bar(position = "dodge", stat = "identity")+
  scale_y_continuous() +
  theme(axis.text.y = element_text(angle = 20, hjust = 1)) +
  coord_flip() +
  labs(title = "Results", y = "Value")+
  theme(plot.title = element_text(hjust = 0.5))

ggplot(data = final.df[which(final.df$OptCutPoint == "No" | final.df$Model == "Penalized ROC"),], aes(x =
reorder(Model, -Brier), y = Brier))+
  geom_bar(stat = "identity", position = "dodge", fill = "turquoise4") +
  geom_text(aes(label = round(Brier, 4)), position = position_dodge(width = 0.2), hjust = -0.25, size = 3) +
  coord_flip() +
  theme_minimal() +
  ylim(0,0.25) +
  xlab("Model") +
  theme(axis.text.y = element_text(angle = 20, hjust = 1)) +
  theme(plot.title = element_text(hjust = 0.5)) +
  labs(title = "Brier Score")


roc_df <- data.frame(ROC = c(model1_auc, model2_auc, model3_auc, model4_auc, model5_auc,
model6_auc))
names <- c("CV Logistic","Penalized ROC", "Penalized Logistic", "Logistic PCA","Pnlzd Logistic PCA",
"CV Logistic Cluster")
roc_df$Model <- names
roc_df

ggplot(data = roc_df, aes(x = reorder(Model, ROC), y = ROC))+
  geom_bar(stat = "identity", position = "dodge") +
  geom_text(aes(label = round(ROC, 4)), position = position_dodge(width = 0.2), hjust = -0.25, size = 3) +
  coord_flip() +
  theme_minimal() +
  ylim(0,1) +
  xlab("Model") +
  theme(axis.text.y = element_text(angle = 20, hjust = 1)) +
  theme(plot.title = element_text(hjust = 0.5)) +
  labs(title = "AUC Values")

varImp(logisticCV.cluster)
cluster.impo <- varImp(logisticCV.cluster, scale = FALSE)
plot(cluster.impo, top = 10, main = "CV Logistic Regression with Clustering")
```