**An Analysis of Missed Doctor Visits Due to Covid-19**

**Abstract**

Predictive models of missed appointments of non-related coronavirus symptoms during the coronavirus pandemic holds clinical importance since needed medical care appointments are being missed which can significantly affect someone’s health and well-being. The coronavirus has significantly changed the way people think, act, attend appointments, receive care, and more. It is important to indicate why people are missing needed medical appointments. To address this question, 5 different models (bagging, boosting, random forest, ridge, and lasso) were used on survey data from IPUMS to predict if people missed medical care appointments other than for coronavirus because of the coronavirus pandemic. The lasso model was chosen as the best model and chose 7 out of 26 coefficients as the most important indicators to determine if someone would miss a medical appointment that is not for the coronavirus during the coronavirus pandemic. These most important coefficients were if the person was male, if they were divorced, if they were of the races Aleut, Alaskan Native, or American Indian, if they received an associate degree from an academic program or a master’s degree or a doctoral degree, and if the person had health insurance coverage. This model was successful in indicating which variables were the most important to focus on as to why people missed their medical care appointments that were unrelated to the coronavirus during the coronavirus pandemic.

**Background and Introduction**

The data analyzed comes from the IPUMS Health Surveys. IPUMS provides worldwide census and survey data. The National Health Interview Survey (NHIS) has been covering data for over 50 years and averages an annual response feedback of 100,000 persons in 45,000 households [1]. The question being examined from this dataset is if people missed medical care appointments other than for coronavirus because of the coronavirus pandemic. This pandemic has taken millions of lives and fear of contracting the virus scares people and prevents them from leaving their house and going to an office where the coronavirus is being treated and is likely to be present. In turn, people will miss regular scheduled appointments because they are too scared or cannot make it to another appointment because of the pandemic. According to a study published in CDC in September 2020 [2], “As of June 30, 2020, an estimated 41% of U.S. adults reported having delayed or avoided medical care during the pandemic because of concerns about COVID-19, including 12% who reported having avoided urgent or emergency care. Medical care delay or avoidance might increase morbidity and mortality risk associated with treatable and preventable health conditions and might contribute to reported excess deaths directly or indirectly related to COVID-19.” This is still a growing problem and it’s affecting countless people from all around the world.

**Data Cleaning and Exploratory Analysis**

Our raw dataset contains 37358 observations and 14 variables. After cleaning the dataset, 15751 observations remain with 12 variables being observed. The response variable ‘CVDDNGCARE’ is a binary variable depicting if people missed medical care appointments other than for coronavirus because of the coronavirus pandemic or not. The predictor variables are explained below. Certain variable responses were combined into a single category.

Race is just the main racial background of the person. If the response did not fit into a category and was recorded as ‘other race’ it was deleted from the dataset. The category ‘American Indian or Alaskan Native and any other group’ was combined with the ‘Aleut, Alaskan Native, or American Indian’ group. The other categories chosen were ‘White’, ‘Black/African-American’, and ‘Asian or Pacific Islander’; now, there are 4 categories under the race variable.

Sex is just if then person is male or female. Any other genders were not included in this study. There were responses that were refused to answer and unknown genders, but they were deleted.

Marital status is the legal marital status of the person. If there was no response or if the person labeled their marital status as ‘separated’ or ‘unknown’, those categories were deleted. The categories studied were ‘married, ‘widowed’, ‘divorced’, and ‘never married’.

Hispanic ethnicity is if the person comes from Hispanic, Spanish, or Latino origin or ancestry. Responses as ‘Other Hispanic’ or ‘Origin unknown, refused or not reported’ were deleted and only responses of ‘Not Hispanic/Spanish origin’ or ‘Mexican’ were reported.

Education is the education that has been attained. If a person was in kindergarten or was in grades 1-12 but did not receive a high school diploma, then they were categorized into one category. If a person had a high school diploma, a GED or equivalent, some college but no degree, or an associate degree through a program that was technical, vocational, occupational, or an academic program then they were all combined into one category. If someone had a bachelor’s degree, master’s degree, professional degree, or doctoral degree then they each got their own category for those degrees. Nonresponses and unknown responses were deleted.

Employment status is if the person is working either for pay or not, or seasonal or contract work in the past 1 to 2 weeks. Only responses of employed or unemployed were kept and unknown and no response back were deleted from the column.

Total combined family income is the grouped family income, including children's income using an income bracket methodology which is adjusted for inflation. The categories are $0 - $34,999; the second is $35,000 - $49,999; the third is $50,000 - $74,999; the fourth is $75,000 - $99,999; and the fifth is $100,000 and over.

Health Insurance coverage status indicates whether the person lacks health insurance coverage or not. This variable just includes if they have coverage or not and any person who responded with a response of unknown was deleted from the column.

Region reports the region of the United States where the housing unit containing participants is located. There are 4 regions. The first is the Northeast: New England Division (Maine, New Hampshire, Vermont, Massachusetts, Rhode Island, and Connecticut) and Middle Atlantic Division (New York, New Jersey, and Pennsylvania). The second is the North Central/Midwest: East North Central Division (Michigan, Ohio, Indiana, Illinois, Wisconsin) and West North Central Division (Minnesota, Iowa, Missouri, North Dakota, South Dakota, Kansas, and Nebraska). The third is the South: South Atlantic Division (Delaware, Maryland, District of Columbia, Virginia, West Virginia, North Carolina, South Carolina, Georgia, and Florida), East South Central Division (Kentucky, Tennessee, Mississippi, and Alabama), and West South Central Division (Texas, Arkansas, Oklahoma, and Louisiana). The last region is the West: Pacific Division (Washington, Alaska, Oregon, California, and Hawaii) and Mountain Division (Montana, Idaho, Wyoming, Colorado, New Mexico, Arizona, Utah, and Nevada).

Urban-rural County classification depicts whether the household lives in a large central metro county, large fringe metro county, medium and small metro county, or nonmetropolitan county. These are the only 4 categories in this column.

Age is just the individual’s age. This is the only non-categorical variable in the dataset. Ages reported as unknown were deleted from the dataset and all the ages are 18 and above.

Year and the unique household identifier code was deleted because the year does not change for any of the variables and the unique household identifier code is basically just an index in the data and is not meaningful in predicting if people missed medical care appointments other than for coronavirus because of the coronavirus pandemic or not.

**Exploratory Data Analysis**

Chart, bar chart

Description automatically generated

**Fig 1**. This bar plot shows the explanatory variable and its responses. Over 83% of people did not miss non coronavirus related medical care appointments due to the coronavirus. This is imbalanced data which often results in overfitting in models. An approach to reduce the likelihood of overfitting this unbalanced data is resampling which was performed in the bagging, boosting, and random forest methods. Another approach to reduce the likelihood of overfitting is regularization which was done in the ridge and lasso models.

Table

Description automatically generated

**Fig 2**. This plot shows various relationships between the variables with different types of graphs.  The variable name URBRRL corresponds to their urban or rural county classification, MARSTAT is the marital status of the respondent, RACEA is the race, HISPETH corresponds to if the person comes from a Hispanic background, EDUC is the education level attained, EMPSTAT is the employment status, INCFAM07ON is the total combined family income, HINOTCOVE is the health insurance coverage status, and CVDDNGCARE if the person missed a medical care visit unrelated to the coronavirus during the coronavirus pandemic. Some basic statistics is that most of the responses are from region 3 (South) and more females responded to the survey, more people are unemployed at a higher age, and the majority or survey responses were people that are white.

**Models**

A total of 5 models were conducted on this dataset. Bagging, boosting, random forest, ridge, and lasso. These models were chosen because they all help to reduce variance and overfitting which is important because the response variable is very unbalanced which leads to overfitted models. Bagging helps to reduce overfitting because the bootstrap method can be implemented by taking repeated samples from the single training dataset. After these bootstrap training datasets are generated, decision trees are made to train each bootstrap sample to get a prediction. All these predictions are then averaged to reduce the variance of the model. The prediction estimates are all independent of each other in the bootstrap process. Each individual tree generally has low bias and high variance but taking the average of all the prediction estimates from the decision trees will result in a lower variance. For classification, bagging takes a majority vote on the classification tree in the bootstrap prediction process and the overall prediction is the one that is most occurring out of all the predictions from the trees.

Random forest provides an improvement over bagged trees because random forest has a method to decorrelate the trees. The beginning of the process is the same as bagging with the bootstrap samples and decision trees, but random forest further reduces the variance after the tree predictions are averaged by a distinct technique. This technique is deployed when each split in a tree is considered in the prediction process for the bootstrap method, a random selection of predictor variables is chosen as opposed to all the predictors. A fresh selection of predictors is taken at each split. The number of predictor variables chosen at each split is commonly the square root of the total number of predictor variables which was done for this model.

Boosting also involves a bootstrap tree building procedure, but each tree is grown dependent on the previous. Boosting also has a fixed number of spits in its tree, 4 was chosen in the boosted model. The algorithm starts with the training data and the sample average of all the predictor variable outputs which is the residual which is the initial estimate. Then a tree is fitted to the training data with that initial residual, the updated predicted value gets updated when multiplied by lambda, a shrinkage parameter, 0.001 is the default value and was chosen for the model. This now gets added to the initial prediction. The residual can now be updated. This process is repeated over and over until all the bootstrap samples are used. Every single time a new tree is built, the residual gets updated. This process improves the tree iteration after iteration and reduces the variation and bias of the model. The last tree is kept and that is used as the model’s predicted estimate.

Ridge regression aims to minimize the coefficient estimates by using a penalty term called L2 regularization that shrinks the estimates towards zero, but all the coefficients are being shrunk by the same factor, so no coefficients are eliminated (Appendix E). There is a tuning parameter lambda that controls the strength of the penalty term. If lambda is zero, then ridge regression just equals least squares regression, and when lambda is infinity, all the coefficients have shrunk to zero. Selecting an optimal lambda can be done through cross validation (Appendix B). The lambda chosen in the final ridge model was ​​0.002782008 and the variables were standardized previously.

Lasso regression is just like ridge regression but has the ability to shrink coefficients to zero instead of just close to zero like ridge regression. Lasso uses an L1 regularization penalty which adds an absolute value to the magnitude of the coefficients. This method can result in the loss of coefficients through variable selection and create a sparse model, a model that involves a subset of the variables. Like ridge regression, choosing an optimal lambda is critical and done through cross validation. The optimal lambda in the lasso model was 0.0002732291.

The technique that ridge and lasso uses to choose the best lambda is cross validation. This is done by making a grid of many lambda values, and then cross validation error rate is computed for each lambda. The tuning parameter is picked that results in the smallest cross validation error. Then the model gets refitted on the training data with the selected value for the tuning parameter.

**Models and Results**

The data was split 70% training and 30% testing. The ridge regression model had the lowest misclassification rate; the error rate was 0.1641981. Lasso regression was the second best model with a misclassification rate of 0.1669488 and was chosen as the final model. The ridge and lasso models had 26 coefficients into consideration for the model and lasso only kept 7 of them (Appendix A). Random forest had the next lowest misclassification rate of 0.16825397 and used 3 random variables in the model’s training process in the tree building process. The next model with the lowest misclassification rate was boosting with a value of 0.18662717. Age had the highest relative influence for the model with age being the second highest. Age was a variable that shrunk to zero in the lasso model, but education was a more influential variable in the lasso model. Health insurance coverage status had the least relative influence in the boosting model (Appendix C). The boosting model had an interaction depth of 4 and both boosting and bagging had 5000 trees to fit in the training stage. Bagging was the model with the lowest misclassification rate of 0.19410431 (Appendix D). 

**Discussion**

I chose the lasso regression model as the best model. Even though it had a slightly higher misclassification rate than ridge, it was only by less than 3 tenths of a percent and still had an accuracy rate above 83.3%. The difference between the two accuracy rates is so small. Lasso also only used 7 coefficients to predict if someone would miss a medical care appointment that is not treatment for the coronavirus during the coronavirus pandemic. Ridge used 26 coefficients to predict this. Sacrificing accuracy to get a model that is far simpler is justified since the difference between the accuracy rates are so small. This sparse model is very important when interpreting data because it shows the most important variables in predictions. It lets the focus be on fewer variables that have the biggest impact instead of distracting, less important variables.

The coefficients in the final model that were chosen were sex if the person was male

(SEXmale); marital status if the person was divorced (MARSTAT30); race if the person was Aleut, Alaskan Native, or American Indian (RACEA300); education if the person had an associate degree from an academic program (EDUC303), or the person had a master’s degree (MA, MS, Med, MBA) (RACEA501), or a doctoral degree (PhD, EdD) (EDUC 503); and if the person has health insurance coverage (HINOTCOVE2). The table is shown below (Table 1).     
Table

Description automatically generated  
Table 1: Coefficients for the lasso regression predictions.

The closer a variable’s correlation estimate is to 1 indicates that that variable is more strongly correlated to the response variable; the further away from 1 a variable’s correlation estimate is, the less correlated that variable is. In terms of odds ratio, a variable’s correlation estimate that is closer to 1 and above 0 means the odds of a success in the response variable is more likely, and if the variable’s correlation estimate is further from 1 and below 0, the odds of a success in the response variable is less likely.

In the output in table 1, if the person is male then the odds of them not receiving needed medical care other than for coronavirus because of the coronavirus pandemic is e-0.22246472 ≈ 0.8005 times more likely than if the person was female, holding all other variables constant. So, a female is less likely to miss a medical care appointment that is unrelated to the coronavirus during the coronavirus pandemic. If a person is divorced, then the odds of them not receiving needed medical care other than for coronavirus because of the coronavirus pandemic is e0.01503367 ≈ 1.015 times more likely than if the person was either widowed or never married, holding all other variables constant.  If a person is of Aleut, Alaskan Native, or American Indian race, then the odds of them not receiving needed medical care other than for coronavirus because of the coronavirus pandemic is e0.0873996 ≈ 1.09133 times more likely than if the person’s race is Black/African American or an Asian or Pacific Islander race, holding all other variables constant. If a person has an associate degree from an academic program, then the odds of them not receiving needed medical care other than for coronavirus because of the coronavirus pandemic is e-0.08065601 ≈ 0.9225 times more likely than if the person has a bachelor’s degree, master’s degree, professional degree, or doctoral degree, holding all other variables constant.  If a person has a master’s degree, then the odds of them not receiving needed medical care other than for coronavirus because of the coronavirus pandemic is e0.08258054 ≈ 1.086 times more likely than if the person has an associate degree from an academic program, bachelor’s degree, professional degree, or doctoral degree, holding all other variables constant. If a person has a doctoral degree, then the odds of them not receiving needed medical care other than for coronavirus because of the coronavirus pandemic is e0.09168455 ≈ 1.096 times more likely than if the person has an associate degree from an academic program, bachelor’s degree, master’s degree, or professional degree, holding all other variables constant.  If the person has medical care coverage, then the odds of them not receiving needed medical care other than for coronavirus because of the coronavirus pandemic is e-0.12716773 ≈ 0.880586 times more likely than if the person does not have health insurance coverage, holding all other variables constant.

Based on these findings, it is clear which variables are more significant than others and this sparse model is the reason for this. The variables with a correlation coefficient above 0 means it is even more correlated to the likelihood of someone not receiving medical care unrelated to the coronavirus during the coronavirus pandemic. It is important to note that the interpretation of a variable is when all other variables are held fixed. The lowest indicator out of these coefficients is if the person is male and the best indicator is if the person had a doctoral degree. The higher the degree, the more likely a person is to miss a needed medical care appointment that is not because of the coronavirus during the coronavirus pandemic. Different speculations can be made as to why this is but this information is helpful in a way that higher educated people need to be targeted more to make sure they get to their appointments even during a pandemic. From these findings, energy can be concentrated to the more important indicator variables to see why people are missing their medical care appointments unrelated to the coronavirus during the coronavirus pandemic.

The limitations of this study includes an unbalanced response variable, and surveys always have the possibility of false input from the respondents. The sample size is fairly large, however. There can also be some correlation among the variables which might skew some results. The region variable is only split into 4 segments, including more splits in segments, even split state by state would be most helpful to more accurately target which areas are more likely to miss a medical care appointment unrelated to the coronavirus during the coronavirus pandemic. More data can also help and that would likely help to further balance the responses for each variable. Missing needed medical appointments that are not due to coronavirus during the coronavirus pandemic is a serious health risk and identifying the factors that are causing people to miss these appointments can have a significant impact on their lives.

**References**

[1] Lynn A. Blewett, Julia A. Rivera Drew, Miriam L. King and Kari C.W. Williams. IPUMS

Health Surveys: National Health Interview Survey, Version 6.4 [dataset]. Minneapolis,

MN: IPUMS, 2019. https://doi.org/10.18128/D070.V6.4

[2] Czeisler, M. É., Marynack, K., & Clarke, K. (2020, September 11). *Delay or avoidance of*

*medical care because of COVID-19–related concerns - United States, June 2020*.

Centers for Disease Control and Prevention. Retrieved December 10, 2021, from

https://www.cdc.gov/mmwr/volumes/69/wr/mm6936a4.htm.

Appendix A

Chart

Description automatically generated

Above is a plot of the variable selection for the lasso regression model. It is clear that the coefficients can shrink to exactly zero. After this step, cross validation will be performed to choose the optimal lambda.

Appendix B

Chart, histogram

Description automatically generated

Above is a plot that chooses the optimal lambda for a ridge and lasso model that will result in the lowest cross-validation error.

Appendix C

Table

Description automatically generated with low confidence

Chart

Description automatically generated

The above plots show a visual representation of the relative frequency for the variable’s importance in the boosting model. The order or rank for the variables is shown in the above table. Age has the highest influence and health insurance coverage has the lowest influence.

Appendix D

Text

Description automatically generated

Above is the output for the bagging model, the confusion matrix is used to calculate the accuracy and misclassification rate. Misclassification rate is calculated by adding the false positives and false negatives and dividing by the total number of points. The accuracy rate is calculated by 1 – the misclassification rate. A confusion matrix like this was produced for each method and the same calculations were made the same for each model. A demonstration of how the calculation is shown below on the bagging confusion matrix.

Misclassification rate = (1761 + 379) / 11025 = 0.19410431

Accuracy rate = 1 – misclassification rate = 1 – 0. 19410431 = 0.80589569

Appendix E

Table

Description automatically generated with medium confidence

The above plot shows the coefficients for the ridge regression model. Notice how none of the coefficients got eliminated, only shrunk. EDUC503 is a doctoral degree and is highly correlated to a person missing a needed medical care appointment that is not to the coronavirus during the coronavirus pandemic.

**R Code**

***Step 1.*** *Importing necessary libraries*

```{r}

library(ISLR)

library(caret)

library(tidyverse)

library(glmnet)

library(Rcpp)

library(boot)

library(MASS)

library(tree)

library(randomForest)

library(gbm)

library(plyr)

```

***Step 2.*** *Reading in the data*

```{r}

covid = read.csv("/Users/lukemoynihan/Downloads/Covid data\_raw.csv")

```

***Step 3.*** *Cleaning the data*

```{r}

#combining races

covid$RACEA[covid$RACEA==350] <- 300

covid$RACEA[covid$RACEA>549] <- NA

covid=subset(covid, RACEA == 100 | RACEA == 200| RACEA == 300| RACEA == 400)

#SEX

# 0 for male and 1 for female, deleting 7,8,9 unkowns

covid$SEX[covid$SEX == '1'] <- "male"

covid$SEX[covid$SEX == '2'] <- "female"

covid<-subset(covid, SEX!="7" & SEX!="9")

# Marital Status

covid<-subset(covid, MARSTAT!= 99 & MARSTAT!=0 & MARSTAT!= 40)

#Hispanic ethnicity

covid<-subset(covid, HISPETH!= 93)

covid$HISPETH[covid$HISPETH==60] <- 20

covid=subset(covid, HISPETH == 20 | HISPETH == 10)

#Education

covid$EDUC[covid$EDUC==102] <- 103

covid$EDUC[covid$EDUC==103] <- 116

covid$EDUC[covid$EDUC==201] <- 202

covid$EDUC[covid$EDUC==202] <- 301

covid$EDUC[covid$EDUC==301] <- 302

covid$EDUC[covid$EDUC==302] <- 303

covid=subset(covid, EDUC == 116 | EDUC == 303| EDUC == 501| EDUC == 502|

EDUC == 400 | EDUC == 503)

#employee status

covid=subset(covid, EMPSTAT == 200 | EMPSTAT == 100)

#health care

covid=subset(covid, HINOTCOVE == 1 | HINOTCOVE == 2)

#response variable, missed apointment

covid=subset(covid, CVDDNGCARE == 1 | CVDDNGCARE == 2)

#including ages above 18 and below 105

covid=subset(covid, AGE >17)

covid=subset(covid,AGE<105)

```

***Step 4.*** *Converting categorical variables to categorical and deleting any N/A values in the data*

```{r}

covid =na.omit(covid)

covid$REGION = as.factor(covid$REGION)

covid$URBRRL = as.factor(covid$URBRRL)

covid$SEX = as.factor(covid$SEX)

covid$MARSTAT = as.factor(covid$MARSTAT)

covid$RACEA = as.factor(covid$RACEA)

covid$EDUC = as.factor(covid$EDUC)

covid$EMPSTAT = as.factor(covid$EMPSTAT)

covid$INCFAM07ON = as.factor(covid$INCFAM07ON)

covid$HINOTCOVE = as.factor(covid$HINOTCOVE)

covid$HISPETH = as.factor(covid$HISPETH)

covid$CVDDNGCARE = as.factor(covid$CVDDNGCARE)

```

*Step 5. Changing response variable to 0 and 1 and dropping YEAR and NHISHID column*

```{r}

covid$CVDDNGCARE <- revalue(covid$CVDDNGCARE, c('1'=0))

covid$CVDDNGCARE <- revalue(covid$CVDDNGCARE, c('2'=1))

drops <- c("YEAR","NHISHID")

covid=covid[ , !(names(covid) %in% drops)]

```

***Bagging model***

*Training the data, 70% train, 30% test, using 11 variables and 5000 trees*

``{r}

train = sample(1: nrow(covid ), nrow(covid )\*0.7)

bag.covid =randomForest(CVDDNGCARE~.,data=covid ,subset =train,

mtry=11,ntree=5000,importance =TRUE)

bag.covid

```

Text

Description automatically generated

***Boosting Model***

*‘ The same libraries and reading in and cleaning the data portion ‘ (Steps 1 – 5)*

*Creating y as the response variable and the y.test variable, 70% training and 30% testing*

```{r}

y=covid$CVDDNGCARE

train = sample(1: nrow(covid ), nrow(covid )\*0.7)

test=(-train )

y.test=y[test]

```

*Boosting function, use distribution = “Bernoulli” because it is a classification problem*

```{r}

attach(covid)

boost.covid =gbm((unclass(CVDDNGCARE)-1)~.,data=covid[train,],distribution="bernoulli", n.trees =5000,interaction.depth=4)

```

*Use boosted model to predict CVDDNGCARE on the test set*

```{r}

yhat.boost=predict(boost.covid,newdata =covid[-train ,],n.trees =5000)

```

A picture containing table

Description automatically generated

***Random Forest Model***

*‘ The same libraries and reading in and cleaning the data portion ‘ (Steps 1 – 5)*

*Creating y as the response variable and the y.test variable, 70% training and 30% testing*

```{r}

y=covid$CVDDNGCARE

train = sample(1: nrow(covid ), nrow(covid )\*0.7)

test=(-train )

y.test=y[test]

```

*Random forest model*

```{r}

rf.covid =randomForest(CVDDNGCARE~.,data=covid ,subset =train,

mtry=3,importance =TRUE,type="classification")

yhat.rf = predict (rf.covid,newdata =covid[-train ,])

rf.covid

importance(rf.covid)

```

Table

Description automatically generated

***Ridge Regression Model***

*‘ The same libraries and reading in and cleaning the data portion ‘ (Steps 1 – 5)*

*Have to pass an x matrix along with a y vector*

```{r}

x=model.matrix(CVDDNGCARE~.,covid)[,-1]

y=covid$CVDDNGCARE

```

```{r}

grid =10^seq(10,-2, length =100) #length implies the length, number of values in the sequence. Here we have 100 lambdas.

```

*Creating y as the response variable and the y.test variable, 70% training and 30% testing*

```{r}

train=sample(1:nrow(x),nrow(x)\*0.7) #training set will have half the data here

test=(-train )

y.test=y[test]

```

*Cross validation function, default is 10 fold validation*

```{r}

cv.out =cv.glmnet(x[train,],y[train],family='binomial', alpha =0)

```

*Assigning the lambda with the lowest cross validation error rate*

```{r}

bestlam =cv.out$lambda.min

```

*Use predict function to get predicted values for the test data using the fitted model from the training data. Then we need the newx argument inside the predict function to specify the test data we want to fit the model to, the s argument is used to specify the optimal lambda*

```{r}

ridge.mod =glmnet(x[train,],y[train],family="binomial",alpha =0,lambda=grid)

ridge.pred=predict(ridge.mod,s=bestlam ,newx=x[test,])

```

*Refit model to the full data set*

```{r}

out=glmnet(x,y,family="binomial",alpha =0)

predict(out ,type="coefficients",s=bestlam)

```

*Predicting the ridge model*

```{r}

ridge.pred=predict(ridge.mod,s=bestlam ,newx=x[test,],type="response")

```

*Using a loop to get the optimal cut-off for our ridge prediction*

```{r}

for(i in seq(from = 0.005, to = 4, by = 0.005)) { # Running for-loop

ridge.prediction = rep('0', 4726)

ridge.prediction[ridge.pred > i] = '1'

print(i)

print(table(ridge.prediction, y.test))

print((table(ridge.prediction, y.test)[2]+table(ridge.prediction, y.test)[3])/4726)

}

```

Table

Description automatically generated with low confidence

*This output is excluding other estimates, the bottom estimate is the best prediction with a misclassification rate of 0.1641981*

***Lasso Regression***

*‘ The same libraries and reading in and cleaning the data portion ‘ (Step 1 – 5)*

*Have to pass an x matrix along with a y vector*

```{r}

x=model.matrix(CVDDNGCARE~.,covid)[,-1]

y=covid$CVDDNGCARE

```

```{r}

grid =10^seq(10,-2, length =100) #length implies the length, number of values in the sequence. Here we have 100 lambdas.

```

*Creating y as the response variable and the y.test variable, 70% training and 30% testing*

```{r}

train=sample(1:nrow(x),nrow(x)\*0.7) #training set will have half the data here

test=(-train )

y.test=y[test]

```

*Alpha = 1 for lasso*

```{r}

lasso.mod =glmnet(x[train,],y[train],family="binomial",alpha =1,lambda=grid)

plot(lasso.mod,xvar ="lambda",)

```

*Cross validation function, default is 10 fold validation*

```{r}

cv.out =cv.glmnet(x[train,],y[train],family='binomial', alpha =1)

```

*Assigning the lambda with the lowest cross validation error rate*

```{r}

bestlam =cv.out$lambda.min

```

*Use predict function to get predicted values for the test data using the fitted model from the training data. Then we need the newx argument inside the predict function to specify the test data we want to fit the model to, the s argument is used to specify the optimal lambda*

```{r}

lasso.pred=predict(lasso.mod,s=bestlam ,newx=x[test,])

```

```{r}

out=glmnet(x,y,family="binomial",alpha =1,lambda=grid)

lasso.coef=predict (out,type ="coefficients",s=bestlam )

lasso.coef

```

(output is table 1 which is printed earlier in this report)

```{r}

lasso.pred=predict(lasso.mod,s=bestlam ,newx=x[test,],type="response")

```

*Using a loop to get the optimal cut-off for our lasso prediction*

```{r}

for(i in seq(from = 0.005, to = 4, by = 0.005)) { # Running for-loop

lasso.prediction = rep('0', 4726)

lasso.prediction[lasso.pred > i] = '1'

print(i)

print(table(lasso.prediction, y.test))

print((table(lasso.prediction, y.test)[2]+table(lasso.prediction, y.test)[3])/4726)

}

```

Table

Description automatically generated

*This output is excluding other estimates, the bottom estimate is the best prediction with a misclassification rate of 0.1669488*