final\_project\_p8106

Michelle Lui

4/27/2021

## Import dataset

#import dataset  
data("NHANES")  
  
#clean dataset - limit to 2011/12 and only include biological predictors, dropped some biological predictors due to excess missing data  
nhanes = NHANES %>% janitor::clean\_names() %>% filter(survey\_yr == "2011\_12") %>% select(gender, age, race1, weight, height, bmi, pulse:bp\_dia3, direct\_chol:urine\_flow1, diabetes) %>% drop\_na()  
  
summary(nhanes)

## gender age race1 weight height   
## female:1735 Min. : 8.00 Black : 392 Min. : 23.00 Min. :124.5   
## male :1781 1st Qu.:25.00 Hispanic: 218 1st Qu.: 62.90 1st Qu.:160.3   
## Median :42.00 Mexican : 315 Median : 77.10 Median :167.8   
## Mean :41.81 White :2319 Mean : 78.25 Mean :167.5   
## 3rd Qu.:57.00 Other : 272 3rd Qu.: 91.10 3rd Qu.:175.4   
## Max. :80.00 Max. :188.50 Max. :199.9   
## bmi pulse bp\_sys\_ave bp\_dia\_ave   
## Min. :12.90 Min. : 40.00 Min. : 81 Min. : 0.00   
## 1st Qu.:22.90 1st Qu.: 64.00 1st Qu.:107 1st Qu.: 62.00   
## Median :26.60 Median : 72.00 Median :117 Median : 69.00   
## Mean :27.63 Mean : 73.41 Mean :119 Mean : 68.44   
## 3rd Qu.:31.30 3rd Qu.: 82.00 3rd Qu.:128 3rd Qu.: 77.00   
## Max. :69.00 Max. :136.00 Max. :209 Max. :116.00   
## bp\_sys1 bp\_dia1 bp\_sys2 bp\_dia2   
## Min. : 74.0 Min. : 0.0 Min. : 82.0 Min. : 0.00   
## 1st Qu.:108.0 1st Qu.: 62.0 1st Qu.:108.0 1st Qu.: 62.00   
## Median :118.0 Median : 70.0 Median :118.0 Median : 70.00   
## Mean :119.5 Mean : 69.2 Mean :119.2 Mean : 68.55   
## 3rd Qu.:128.0 3rd Qu.: 78.0 3rd Qu.:128.0 3rd Qu.: 78.00   
## Max. :212.0 Max. :110.0 Max. :208.0 Max. :116.00   
## bp\_sys3 bp\_dia3 direct\_chol tot\_chol   
## Min. : 78.0 Min. : 0.00 Min. :0.470 Min. : 1.530   
## 1st Qu.:106.0 1st Qu.: 62.00 1st Qu.:1.110 1st Qu.: 4.110   
## Median :116.0 Median : 70.00 Median :1.290 Median : 4.780   
## Mean :118.7 Mean : 68.33 Mean :1.365 Mean : 4.858   
## 3rd Qu.:128.0 3rd Qu.: 78.00 3rd Qu.:1.580 3rd Qu.: 5.530   
## Max. :210.0 Max. :116.00 Max. :4.030 Max. :10.290   
## urine\_vol1 urine\_flow1 diabetes   
## Min. : 1 Min. : 0.0110 No :3223   
## 1st Qu.: 47 1st Qu.: 0.3995 Yes: 293   
## Median : 88 Median : 0.6800   
## Mean :113 Mean : 0.9642   
## 3rd Qu.:156 3rd Qu.: 1.2140   
## Max. :446 Max. :10.1430

#create a feature plot to better visualize the data  
theme1 <- transparentTheme(trans = .4)   
trellis.par.set(theme1)  
#featurePlot(x = nhanes[, 1:19],  
# y = nhanes$diabetes,  
# scales = list(x = list(relation = "free"), y = list(relation = "free")),  
# plot = "density", pch = "|", auto.key = list(columns = 2))

## Create Data Partition

set.seed(2)  
  
train\_rows = createDataPartition(y = nhanes$diabetes,p = 0.7,list = FALSE)  
train = nhanes[train\_rows, ]

## Warning: The `i` argument of ``[`()` can't be a matrix as of tibble 3.0.0.  
## Convert to a vector.  
## This warning is displayed once every 8 hours.  
## Call `lifecycle::last\_warnings()` to see where this warning was generated.

test = nhanes[-train\_rows, ]  
  
x = model.matrix(diabetes ~ ., train)[ ,-1]  
y = train$diabetes  
  
x2 = model.matrix(diabetes ~ ., test)[ ,-1]  
y2 = test$diabetes  
  
control1 = trainControl(method = "cv", selectionFunction = "best", sampling = "down")

## Lasso Model

set.seed(2)  
lasso\_fit = train(x, y,   
 method = "glmnet",  
 tuneGrid = expand.grid(alpha = 1, lambda = exp(seq(-20, 20,length=100))),  
 trControl = control1,  
 preProcess=c("center", "scale"),  
 family = "binomial")   
  
#Print the values of alpha and lambda that gave best prediction  
lasso\_fit$bestTune

## alpha lambda  
## 29 1 0.0001687877

#Print all of the options examined  
lasso\_fit$results

## alpha lambda Accuracy Kappa AccuracySD KappaSD  
## 1 1 2.061154e-09 0.7389215 0.2215589 0.03900975 0.05610940  
## 2 1 3.087329e-09 0.7389215 0.2215589 0.03900975 0.05610940  
## 3 1 4.624400e-09 0.7389215 0.2215589 0.03900975 0.05610940  
## 4 1 6.926725e-09 0.7389215 0.2215589 0.03900975 0.05610940  
## 5 1 1.037529e-08 0.7389215 0.2215589 0.03900975 0.05610940  
## 6 1 1.554078e-08 0.7389215 0.2215589 0.03900975 0.05610940  
## 7 1 2.327799e-08 0.7389215 0.2215589 0.03900975 0.05610940  
## 8 1 3.486727e-08 0.7389215 0.2215589 0.03900975 0.05610940  
## 9 1 5.222645e-08 0.7389215 0.2215589 0.03900975 0.05610940  
## 10 1 7.822814e-08 0.7389215 0.2215589 0.03900975 0.05610940  
## 11 1 1.171752e-07 0.7389215 0.2215589 0.03900975 0.05610940  
## 12 1 1.755125e-07 0.7389215 0.2215589 0.03900975 0.05610940  
## 13 1 2.628939e-07 0.7389215 0.2215589 0.03900975 0.05610940  
## 14 1 3.937795e-07 0.7389215 0.2215589 0.03900975 0.05610940  
## 15 1 5.898283e-07 0.7389215 0.2215589 0.03900975 0.05610940  
## 16 1 8.834829e-07 0.7389215 0.2215589 0.03900975 0.05610940  
## 17 1 1.323338e-06 0.7389215 0.2215589 0.03900975 0.05610940  
## 18 1 1.982180e-06 0.7389215 0.2215589 0.03900975 0.05610940  
## 19 1 2.969038e-06 0.7389215 0.2215589 0.03900975 0.05610940  
## 20 1 4.447216e-06 0.7389215 0.2215589 0.03900975 0.05610940  
## 21 1 6.661326e-06 0.7389215 0.2215589 0.03900975 0.05610940  
## 22 1 9.977764e-06 0.7389215 0.2215589 0.03900975 0.05610940  
## 23 1 1.494534e-05 0.7389215 0.2215589 0.03900975 0.05610940  
## 24 1 2.238609e-05 0.7389215 0.2215589 0.03900975 0.05610940  
## 25 1 3.353133e-05 0.7389215 0.2215589 0.03900975 0.05610940  
## 26 1 5.022539e-05 0.7389215 0.2215589 0.03900975 0.05610940  
## 27 1 7.523083e-05 0.7389215 0.2215589 0.03900975 0.05610940  
## 28 1 1.126856e-04 0.7389215 0.2215589 0.03900975 0.05610940  
## 29 1 1.687877e-04 0.7389215 0.2215589 0.03900975 0.05610940  
## 30 1 2.528211e-04 0.7377037 0.2203146 0.03911217 0.05667361  
## 31 1 3.786918e-04 0.7368907 0.2214648 0.03947130 0.05538297  
## 32 1 5.672290e-04 0.7372955 0.2218676 0.03930208 0.05525860  
## 33 1 8.496321e-04 0.7364825 0.2201904 0.04052799 0.05379048  
## 34 1 1.272634e-03 0.7372988 0.2187349 0.03961645 0.06097162  
## 35 1 1.906233e-03 0.7356728 0.2148977 0.03667617 0.05381881  
## 36 1 2.855279e-03 0.7360826 0.2150220 0.03435489 0.05360701  
## 37 1 4.276820e-03 0.7336535 0.2136671 0.03111187 0.04412642  
## 38 1 6.406097e-03 0.7291935 0.2119917 0.02693374 0.03629520  
## 39 1 9.595465e-03 0.7259497 0.2102839 0.02803904 0.04162855  
## 40 1 1.437271e-02 0.7235188 0.2206764 0.02956605 0.03674093  
## 41 1 2.152837e-02 0.7206667 0.2248849 0.02962718 0.04781001  
## 42 1 3.224658e-02 0.7166049 0.2175298 0.02420126 0.04170707  
## 43 1 4.830100e-02 0.7153854 0.2150464 0.02318906 0.04365378  
## 44 1 7.234835e-02 0.7019937 0.2022887 0.01964916 0.04414558  
## 45 1 1.083680e-01 0.6865894 0.1853342 0.02465944 0.04301146  
## 46 1 1.623206e-01 0.6715618 0.1752595 0.03056515 0.04497615  
## 47 1 2.431343e-01 0.4185278 0.0000000 0.43040766 0.00000000  
## 48 1 3.641822e-01 0.4185278 0.0000000 0.43040766 0.00000000  
## 49 1 5.454956e-01 0.4185278 0.0000000 0.43040766 0.00000000  
## 50 1 8.170784e-01 0.4185278 0.0000000 0.43040766 0.00000000  
## 51 1 1.223873e+00 0.4185278 0.0000000 0.43040766 0.00000000  
## 52 1 1.833195e+00 0.4185278 0.0000000 0.43040766 0.00000000  
## 53 1 2.745878e+00 0.4185278 0.0000000 0.43040766 0.00000000  
## 54 1 4.112954e+00 0.4185278 0.0000000 0.43040766 0.00000000  
## 55 1 6.160647e+00 0.4185278 0.0000000 0.43040766 0.00000000  
## 56 1 9.227814e+00 0.4185278 0.0000000 0.43040766 0.00000000  
## 57 1 1.382202e+01 0.4185278 0.0000000 0.43040766 0.00000000  
## 58 1 2.070351e+01 0.4185278 0.0000000 0.43040766 0.00000000  
## 59 1 3.101105e+01 0.4185278 0.0000000 0.43040766 0.00000000  
## 60 1 4.645034e+01 0.4185278 0.0000000 0.43040766 0.00000000  
## 61 1 6.957632e+01 0.4185278 0.0000000 0.43040766 0.00000000  
## 62 1 1.042159e+02 0.4185278 0.0000000 0.43040766 0.00000000  
## 63 1 1.561013e+02 0.4185278 0.0000000 0.43040766 0.00000000  
## 64 1 2.338186e+02 0.4185278 0.0000000 0.43040766 0.00000000  
## 65 1 3.502285e+02 0.4185278 0.0000000 0.43040766 0.00000000  
## 66 1 5.245949e+02 0.4185278 0.0000000 0.43040766 0.00000000  
## 67 1 7.857720e+02 0.4185278 0.0000000 0.43040766 0.00000000  
## 68 1 1.176980e+03 0.4185278 0.0000000 0.43040766 0.00000000  
## 69 1 1.762956e+03 0.4185278 0.0000000 0.43040766 0.00000000  
## 70 1 2.640670e+03 0.4185278 0.0000000 0.43040766 0.00000000  
## 71 1 3.955365e+03 0.4185278 0.0000000 0.43040766 0.00000000  
## 72 1 5.924601e+03 0.4185278 0.0000000 0.43040766 0.00000000  
## 73 1 8.874250e+03 0.4185278 0.0000000 0.43040766 0.00000000  
## 74 1 1.329242e+04 0.4185278 0.0000000 0.43040766 0.00000000  
## 75 1 1.991025e+04 0.4185278 0.0000000 0.43040766 0.00000000  
## 76 1 2.982285e+04 0.4185278 0.0000000 0.43040766 0.00000000  
## 77 1 4.467059e+04 0.4185278 0.0000000 0.43040766 0.00000000  
## 78 1 6.691050e+04 0.4185278 0.0000000 0.43040766 0.00000000  
## 79 1 1.002229e+05 0.4185278 0.0000000 0.43040766 0.00000000  
## 80 1 1.501203e+05 0.4185278 0.0000000 0.43040766 0.00000000  
## 81 1 2.248598e+05 0.4185278 0.0000000 0.43040766 0.00000000  
## 82 1 3.368095e+05 0.4185278 0.0000000 0.43040766 0.00000000  
## 83 1 5.044950e+05 0.4185278 0.0000000 0.43040766 0.00000000  
## 84 1 7.556651e+05 0.4185278 0.0000000 0.43040766 0.00000000  
## 85 1 1.131884e+06 0.4185278 0.0000000 0.43040766 0.00000000  
## 86 1 1.695409e+06 0.4185278 0.0000000 0.43040766 0.00000000  
## 87 1 2.539492e+06 0.4185278 0.0000000 0.43040766 0.00000000  
## 88 1 3.803815e+06 0.4185278 0.0000000 0.43040766 0.00000000  
## 89 1 5.697600e+06 0.4185278 0.0000000 0.43040766 0.00000000  
## 90 1 8.534232e+06 0.4185278 0.0000000 0.43040766 0.00000000  
## 91 1 1.278312e+07 0.4185278 0.0000000 0.43040766 0.00000000  
## 92 1 1.914739e+07 0.4185278 0.0000000 0.43040766 0.00000000  
## 93 1 2.868019e+07 0.4185278 0.0000000 0.43040766 0.00000000  
## 94 1 4.295904e+07 0.4185278 0.0000000 0.43040766 0.00000000  
## 95 1 6.434681e+07 0.4185278 0.0000000 0.43040766 0.00000000  
## 96 1 9.638281e+07 0.4185278 0.0000000 0.43040766 0.00000000  
## 97 1 1.443684e+08 0.4185278 0.0000000 0.43040766 0.00000000  
## 98 1 2.162443e+08 0.4185278 0.0000000 0.43040766 0.00000000  
## 99 1 3.239046e+08 0.4185278 0.0000000 0.43040766 0.00000000  
## 100 1 4.851652e+08 0.4185278 0.0000000 0.43040766 0.00000000

# Model coefficients  
coef(lasso\_fit$finalModel, s = lasso\_fit$bestTune$lambda)

## 23 x 1 sparse Matrix of class "dgCMatrix"  
## 1  
## (Intercept) -0.017221332  
## gendermale 0.262833864  
## age 1.269545424  
## race1Hispanic -0.066709306  
## race1Mexican -0.301804930  
## race1White -0.477947921  
## race1Other -0.141978372  
## weight -1.144275818  
## height 0.278937097  
## bmi 1.835112291  
## pulse 0.534089535  
## bp\_sys\_ave 0.013520485  
## bp\_dia\_ave .   
## bp\_sys1 -1.539988504  
## bp\_dia1 -0.190755978  
## bp\_sys2 0.003237414  
## bp\_dia2 -0.336137258  
## bp\_sys3 1.807495779  
## bp\_dia3 0.466746153  
## direct\_chol -0.407105827  
## tot\_chol -0.163464405  
## urine\_vol1 0.093481640  
## urine\_flow1 0.226351573

# Make predictions  
lasso\_pred = lasso\_fit %>% predict(x2) %>% as.numeric()  
lasso\_pred\_p = ifelse(lasso\_pred-1 > 0.5,1,0)  
  
test\_outcome\_lasso = (as.numeric(y2)-1)  
  
misclasserror\_lasso = mean(lasso\_pred\_p != test\_outcome\_lasso, na.rm=T)  
print(paste('Accuracy Model 1', 1-misclasserror\_lasso))

## [1] "Accuracy Model 1 0.679962013295347"

## Random Forest

mtry\_vals = c(ncol(train)-1, sqrt(ncol(train)-1), 0.5\*ncol(train)-1)  
  
mtry\_grid = expand.grid(.mtry=mtry\_vals)  
  
set.seed(2)  
rf\_fit = train(diabetes ~.,   
 data = train,   
 method="rf",   
 trControl = control1,   
 metric="Accuracy",   
 tuneGrid=mtry\_grid,   
 ntree=100)  
  
rf\_fit$results

## mtry Accuracy Kappa AccuracySD KappaSD  
## 1 4.358899 0.7893020 0.3067161 0.03897397 0.07573702  
## 2 9.000000 0.8022507 0.3364938 0.01955935 0.06052209  
## 3 19.000000 0.7901265 0.3190312 0.02824320 0.04245803

rf\_fit$bestTune

## mtry  
## 2 9

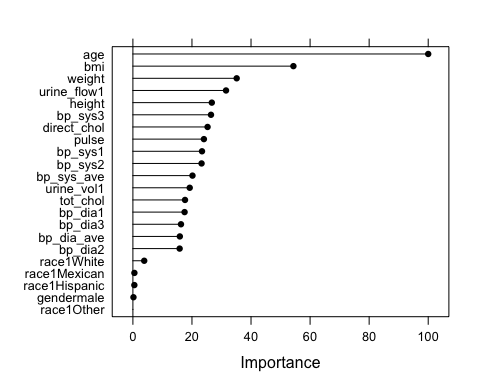
rf\_fit$finalModel

##   
## Call:  
## randomForest(x = x, y = y, ntree = 100, mtry = param$mtry)   
## Type of random forest: classification  
## Number of trees: 100  
## No. of variables tried at each split: 9  
##   
## OOB estimate of error rate: 21.6%  
## Confusion matrix:  
## No Yes class.error  
## No 153 53 0.2572816  
## Yes 36 170 0.1747573

varImp(rf\_fit)

## rf variable importance  
##   
## only 20 most important variables shown (out of 22)  
##   
## Overall  
## age 100.0000  
## bmi 54.3528  
## weight 35.1443  
## urine\_flow1 31.5354  
## height 26.7215  
## bp\_sys3 26.4446  
## direct\_chol 25.2856  
## pulse 24.0449  
## bp\_sys1 23.4010  
## bp\_sys2 23.2512  
## bp\_sys\_ave 20.1751  
## urine\_vol1 19.2435  
## tot\_chol 17.6468  
## bp\_dia1 17.5009  
## bp\_dia3 16.2830  
## bp\_dia\_ave 15.8940  
## bp\_dia2 15.8291  
## race1White 3.8331  
## race1Mexican 0.5097  
## race1Hispanic 0.4914

plot(varImp(rf\_fit))



varImpPlot(rf\_fit$finalModel)



rf\_pred = predict(rf\_fit, test) %>% as.numeric()  
rf\_pred\_p = ifelse(rf\_pred-1 > 0.5,1,0)  
  
test\_outcome\_rf = (as.numeric(test$diabetes)-1)  
  
misclasserror\_rf = mean(rf\_pred\_p != test\_outcome\_rf, na.rm=T)  
print(paste('Accuracy Model 2', 1-misclasserror\_rf))

## [1] "Accuracy Model 2 0.784425451092118"

## Model Comparisons

resamp = resamples(list(lasso = lasso\_fit, rf = rf\_fit))   
summary(resamp)

##   
## Call:  
## summary.resamples(object = resamp)  
##   
## Models: lasso, rf   
## Number of resamples: 10   
##   
## Accuracy   
## Min. 1st Qu. Median Mean 3rd Qu. Max. NA's  
## lasso 0.6747967 0.7163235 0.7479964 0.7389215 0.7689839 0.7845528 0  
## rf 0.7723577 0.7943287 0.8004065 0.8022507 0.8085070 0.8461538 0  
##   
## Kappa   
## Min. 1st Qu. Median Mean 3rd Qu. Max. NA's  
## lasso 0.1385273 0.1857883 0.2129418 0.2215589 0.2350735 0.3151592 0  
## rf 0.2363828 0.2979727 0.3372955 0.3364938 0.3677098 0.4568916 0