

# QMSS Final Project

The Physician Compare website was created by the Centers for Medicare & Medicaid Services (CMS) in December 2010 as required by the Affordable Care Act (ACA) of 2010 to help patients assess and find doctors and hospitals. This dataset contains the information supplied to patients via that website, including patient satisfaction surveys and performance scores across over 100 metrics.

Looking at individual physician scores:

- MIPS
- Performance by measure category
- Organization MIPS

Possible project:

\* Prediction of whether to see a physician based on threshold \* Predict physician score based on all factors (predict MIPS?) + classification of whether you should go to a doctor based on MIPS score + train on several classification methods to analyze best approach for this data

**Question/Problem:** How can we better help patients assess and find doctors, where the scoring and rating come in a format not easily accessible or understandable by the average individual?

**Approach/Methods:** Supervised learning for binary classification utilizing the MIPS as a target with other physician scoring methods as predictors (which we know some of the metrics are direct factors of the individual MIPS scoring, such as the IA, ACI, and Quality category scorings). Potential methods outlined below, including generalized linear models and tree methods.

```
invisible(library(dplyr))
invisible(library(stringr))
invisible(library(tidyr))
invisible(library(dmm))
invisible(library(pcaPP))
invisible(library(caret))
```

```
set.seed(70856775)
```

```
# Professional.Enrollment.ID is non-predictive, same with PAC_id
dr_scores <- read.csv("full_doctor_scoring.csv", sep = ",", na = c("NA", "N/A"))
```

Potential Methods for Binary Classification:

Using overall MIPS for individuals where MIPS  $\geq 75$ , the positive payment adjustment threshold.

- could apply spline to other MIPS, ACI scorings since they're somewhat discrete in nature.
  - ACI  $\geq 0$  : clinician reported ACI category
  - ACI  $\geq 50$ : clinician achieved base score for ACI
  - MIPS  $< 30$ : Negative Payment Adjustment
- Predictive MIPS  $\geq 75$ , essentially.
- Methods to try:
  - glmnet for binary classification (elastic model/penalized logit)
  - glm logit model with polynomials?
  - tree model if we can make it work? (Single Tree, Random Forest, Boosting, Dbarts???)
  - PLSDA or LDA

- nnet or MARS
- PCA to look at similar variables?

```
# remove majority of missingness by removing each task type
no_tasks <- dr_scores[c(1:23, 57)]

# add a "UKN" for unknown category
no_tasks$Credential <- factor(no_tasks$Credential,
                             levels = c(levels(dr_scores$Credential), "UKN"))
no_tasks$Credential[is.na(no_tasks$Credential)] <- "UKN"

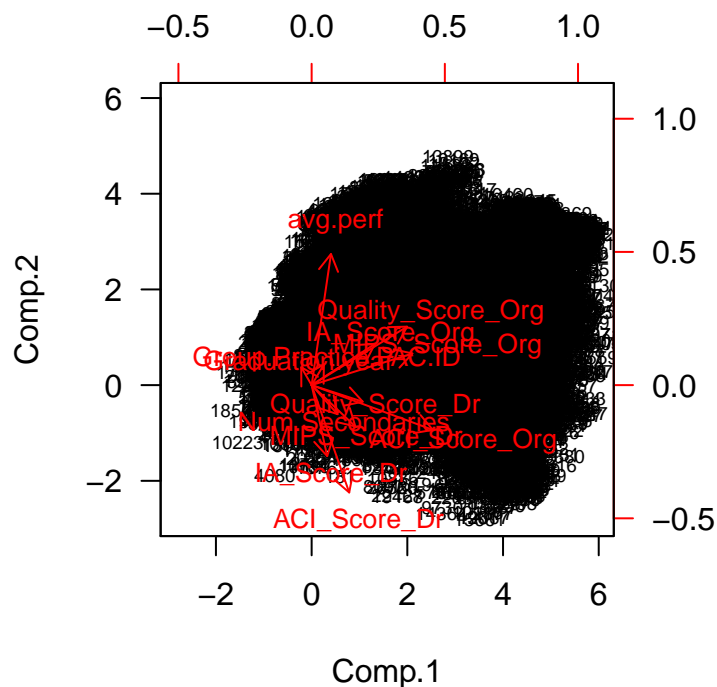
# -1 for not reporting individual ACI Score (NA value)
no_tasks$ACI_Score_Dr[is.na(no_tasks$ACI_Score_Dr)] <- -1
```

## PCA to Explore Correlation of Variables

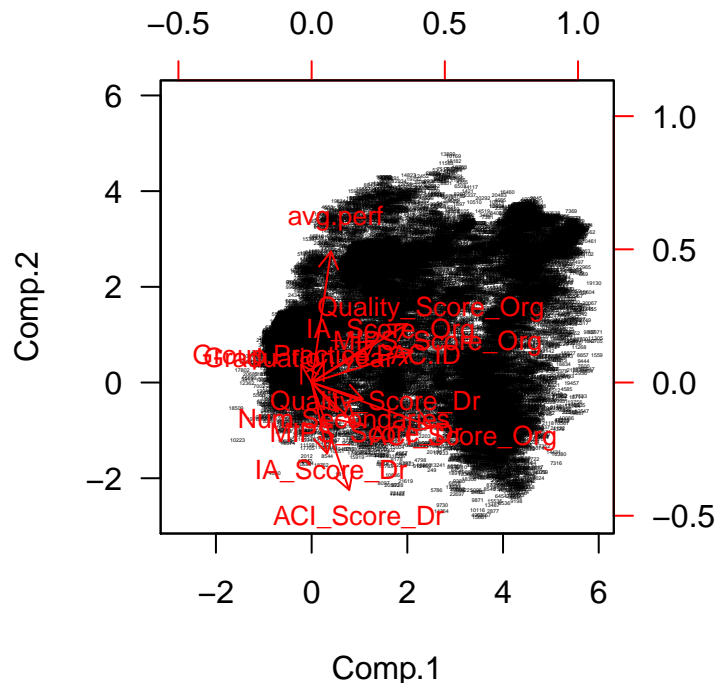
An initial PCA analysis to look understand how rows cluster based on the column variables. This is without considering the binary classifying aspect of MIPS  $\geq 75$ , and is just to understand some of the structure of the scores in the data.

```
# still minor missingness in Graduation Year and Quality Score
numerics <- no_tasks[, c(8, 10:11, 15:18, 20:24)] %>% na.omit(.)
pr_out <- PCAproj(numerics, scale = sd)
par(mar = c(5, 4, 3, 3) + 0.1, las = 1)
```

```
# visualization of any immediate outliers and the variables
biplot(pr_out, scale = 0, cex = c(0.6, 0.8))
```



```
# clusters within the observations visible
biplot(pr_out, scale = 0, cex = c(0.13, 0.8))
```



As mostly expected, the MIPS scores for the hospital organizations that each doctor works at are more correlated to each other than they are to the MIPS scores for each individual doctor. While the number of secondary specialties each doctor has is more correlated to the individual doctor MIPS scores, the group practice ID is more correlated to the organization scores. This is also fairly in line with our expectations that measures for the practice organization would cluster separately from the measures for the individual doctors. Interestingly, the average performance for an individual doctor across task categories seems to be more correlated to the organization scores though. Since performance is measured through individual patient reporting, their experience with the organization itself may be taken into consideration and bias their score, even if the doctor-patient interaction itself was positive.

In the second PCA plot, with the observation labels less cluttered from size, we see that there seem to be distinct clusters of observations. This would indicate groupings within the observations with distinguishing measurement characteristics. Many points do deviate from the groupings themselves. Still, overall, it doesn't appear that any observations seems like a major outlier, as seen in the first plot.

### Prepare dataset for training/testing

```
# we do not expect PAC_id or Professional.Enrollment.ID to be predictive
# create factor for classification prediction (Individual Doctor MIPS Score >= 75)
# Score_source_org has no variance -- they're all group

no_tasks <- na.omit(no_tasks)[, c(3:13, 15:ncol(no_tasks))]
no_tasks$MIPS75_Dr <- factor(no_tasks$MIPS_Score_Dr >= 75.0,
                             levels = c(TRUE, FALSE),
                             labels = c("yes", "no"))

head(no_tasks)
```

##	last_name	first_name	Gender	Credential	
## 1	KANTER	DAVID	F	UKN	
## 2	GOTESMAN	ALEXANDER	M	UKN	
## 3	HARTMAN	MATTHEW	M	UKN	
## 4	NASAJPOUR	HOSSEIN	M	UKN	
## 5	MCCOPPIN	HOLLY	F	UKN	
## 6	JOHN	JAYASHREE	F	UKN	
##	Medical.school.name	Graduation.year			
## 1	OTHER	1998			
## 2	OTHER	1999			
## 3	UNIVERSITY OF PITTSBURGH SCHOOL OF MEDICINE	2002			
## 4	UNIVERSITY OF FLORIDA COLLEGE OF MEDICINE	2003			
## 5	UNIVERSITY OF MISSOURI, COLUMBIA SCHOOL OF MEDICINE	2006			
## 6	OTHER	1984			
##	Primary.specialty	Num.Secondaries			
## 1	PHYSICAL MEDICINE AND REHABILITATION	0			
## 2	UROLOGY	0			
## 3	DIAGNOSTIC RADIOLOGY	0			
## 4	PLASTIC AND RECONSTRUCTIVE SURGERY	0			
## 5	DERMATOLOGY	0			
## 6	PSYCHIATRY	0			
##	Group.Practice.PAC.ID	State			
## 1	8123911500	NY			
## 2	9032293709	NJ			
## 3	8426364738	PA			
## 4	4385876655	MS			
## 5	1456329655	CO			
## 6	8729381033	GA			
##	Organization.legal.name	Quality_Score_Org			
## 1	PHYSICAL MEDICINE and REHAB MEDICAL SERVICE GROUP	100			
## 2	LAKEWOOD UROLOGY LIMITED LIABILITY COMPANY	0			
## 3	ALLEGHENY CLINIC RADIOLOGY	100			
## 4	SOUTH CENTRAL CLINICS, INC	0			
## 5	LAKE LOVELAND DERMATOLOGY PC	0			
## 6	ST FRANCIS PHYSICIAN PRACTICES LLC	0			
##	ACI_Score_Org	IA_Score_Org	MIPS_Score_Org	Score_Source_Dr	
## 1	100	40	100.00	group	
## 2	0	0	0.00	individual	
## 3	0	40	100.00	group	
## 4	0	20	7.50	individual	
## 5	0	0	0.00	individual	
## 6	0	10	3.75	individual	
##	Quality_Score_Dr	ACI_Score_Dr	IA_Score_Dr	MIPS_Score_Dr	avg.perf
## 1	100.0	100	40	100.0	74.0
## 2	67.0	0	0	40.2	87.6
## 3	100.0	-1	40	100.0	100.0
## 4	69.2	100	40	81.5	47.0
## 5	72.6	100	40	83.6	54.4
## 6	22.9	-1	0	19.5	99.0
##	MIPS75_Dr				
## 1	yes				
## 2	no				
## 3	yes				
## 4	yes				

```
## 5      yes
## 6      no
```

```
splitter <- createDataPartition(no_tasks$MIPS75_Dr,
                                p = 0.5, list = F)
train_split <- no_tasks[splitter, ]
test_split <- no_tasks[-splitter, ]

in_train <- createDataPartition(train_split$MIPS75_Dr,
                                p = 0.6, list = F)
training <- train_split[in_train, ]
testing <- train_split[-in_train, ]
```

```
in_train <- createDataPartition(no_tasks$MIPS75_Dr,
                                p = 0.8, list = F)
training <- no_tasks[in_train, ]
testing <- no_tasks[-in_train, ]

training <- training[, c(3:4, 8, 12:15, 17:22)]
testing <- testing[, c(3:4, 8, 12:15, 17:22)]
```

```
# even splits of factors with several levels
# returns padded resampled version though!
```

```
schools <- as.character(sort(unique(no_tasks$Medical.school.name)))
primary <- as.character(sort(unique(no_tasks$Primary.specialty)))
creds <- as.character(sort(unique(no_tasks$Credential)))
group_fctr <- c(schools, primary, creds)

in_train <- sapply(group_fctr, FUN = function(st) {
  task_st <- filter(no_tasks, (
    (Primary.specialty == st) | (Medical.school.name == st) | (Credential == st)))
  if (nrow(task_st) == 1) return(1L)
  return(createDataPartition(task_st$MIPS75_Dr,
                             p = 0.8, list = FALSE))
})
names(in_train) <- group_fctr
train_fctr <- bind_rows(lapply(group_fctr, FUN = function(st) {
  tasks_st <- filter(no_tasks, (
    (Primary.specialty == st) | (Medical.school.name == st) | (Credential == st)))
  x <- in_train[[st]]
  return(tasks_st[x, , drop = FALSE])
})))
test_fctr <- bind_rows(lapply(group_fctr, FUN = function(st) {
  task_st <- filter(no_tasks, (
    (Primary.specialty == st) | (Medical.school.name == st) | (Credential == st)))
  x <- in_train[[st]]
  return(task_st[-x, , drop = FALSE])
})))

train_fctr <- train_fctr[, c(3:5, 7:8, 12:15, 17:22)]
test_fctr <- test_fctr[, c(3:5, 7:8, 12:15, 17:22)]
```

Let's test with a basic linear model:

```
# Using the factor stratified dataset
ols_strat <- lm(MIPS75_Dr == "yes" ~ ., data = train_fctr)
yhat_strat <- predict(ols_strat, newdata = test_fctr)
z_ols_strat <- factor(yhat_strat > 0.5, levels = c(TRUE, FALSE),
                      labels = c("yes", "no"))
confusionMatrix(z_ols_strat, reference = test_fctr$MIPS75_Dr)
```

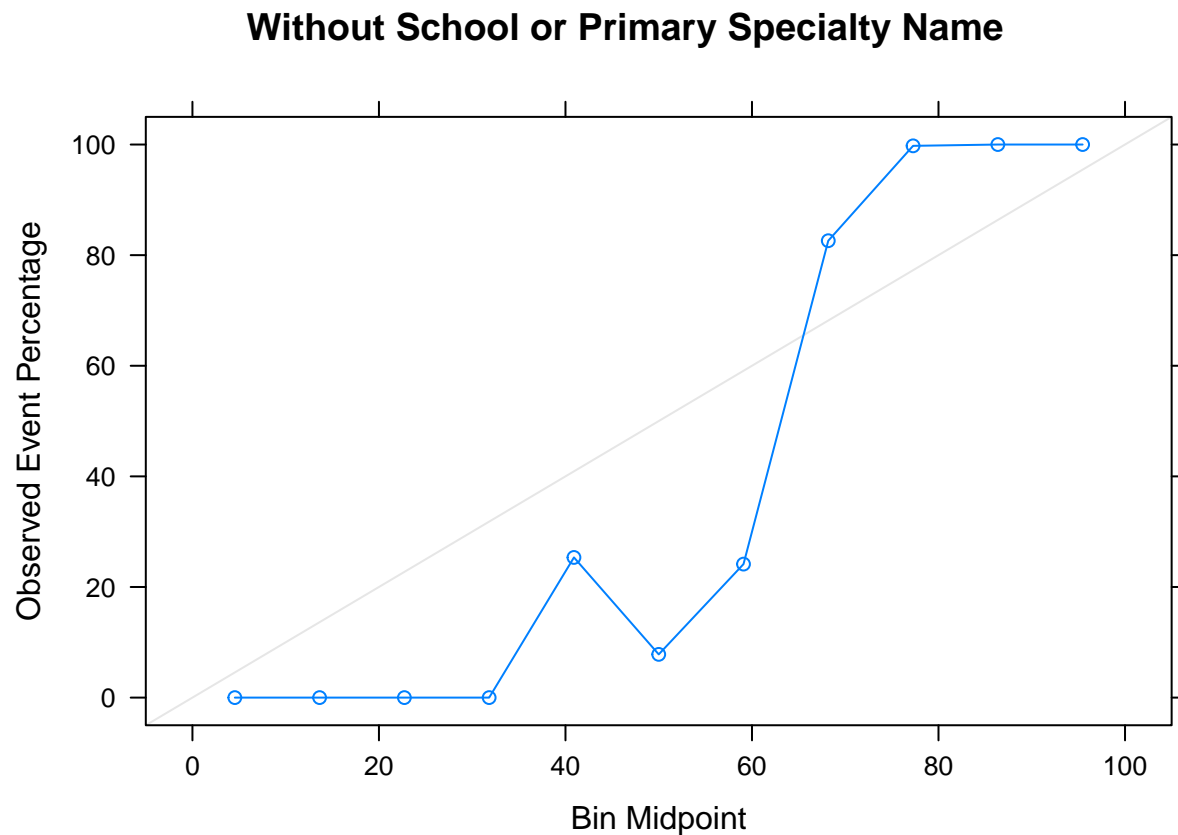
```
## Confusion Matrix and Statistics
##
##              Reference
## Prediction  yes   no
##          yes 9363  621
##          no  152 3419
##
##              Accuracy : 0.943
##              95% CI : (0.9389, 0.9468)
##      No Information Rate : 0.702
##      P-Value [Acc > NIR] : < 2.2e-16
##
##              Kappa : 0.859
##
##  Mcnemar's Test P-Value : < 2.2e-16
##
##              Sensitivity : 0.9840
##              Specificity : 0.8463
##              Pos Pred Value : 0.9378
##              Neg Pred Value : 0.9574
##              Prevalence : 0.7020
##              Detection Rate : 0.6907
##      Detection Prevalence : 0.7366
##              Balanced Accuracy : 0.9152
##
##      'Positive' Class : yes
##
```

```
# testing without stratification, med school name, primary speciality
ols <- lm(MIPS75_Dr == "yes" ~ ., data = training)
yhat <- predict(ols, newdata = testing)
z_ols <- factor(yhat > 0.5, levels = c(TRUE, FALSE),
                labels = c("yes", "no"))
confusionMatrix(z_ols, reference = testing$MIPS75_Dr)
```

```
## Confusion Matrix and Statistics
##
##              Reference
## Prediction  yes   no
##          yes 3148  209
##          no   65 1176
##
##              Accuracy : 0.9404
```

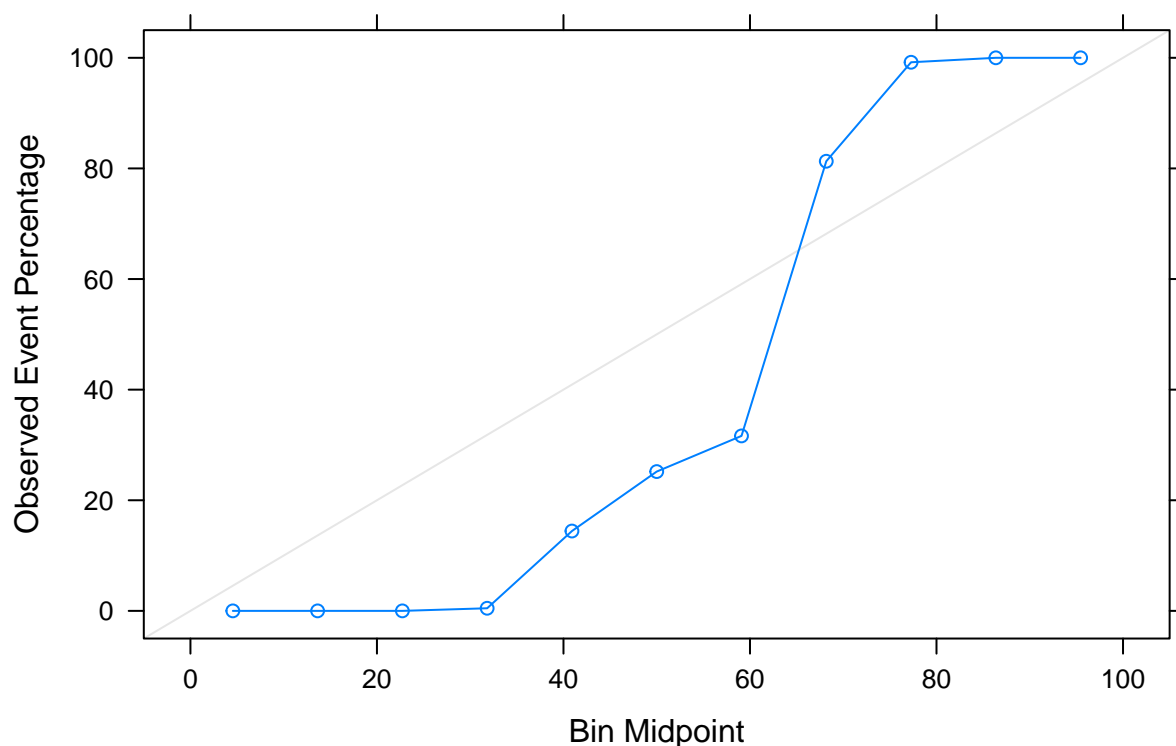
```
##          95% CI : (0.9332, 0.9471)
##    No Information Rate : 0.6988
##    P-Value [Acc > NIR] : < 2.2e-16
##
##          Kappa : 0.8541
##
##    Mcnemar's Test P-Value : < 2.2e-16
##
##          Sensitivity : 0.9798
##          Specificity : 0.8491
##          Pos Pred Value : 0.9377
##          Neg Pred Value : 0.9476
##          Prevalence : 0.6988
##          Detection Rate : 0.6846
##          Detection Prevalence : 0.7301
##          Balanced Accuracy : 0.9144
##
##          'Positive' Class : yes
##
```

```
calibration(MIPS75_Dr ~ yhat, data = testing) %>%
  plot(., main = "Without School or Primary Specialty Name")
```



```
calibration(MIPS75_Dr ~ yhat_strat, data = test_fctr) %>%
  plot(., main = "With Stratification by Factor")
```

## With Stratification by Factor



The stratified model seems to fit a bit better compared to the model trained and tested without medical school name or primary specialty. This is especially for bins of lower probabilities, such as 0.5, which is fairly expected, as a larger number of observations from stratification and including medical school name and primary specialty would likely provide much more information. These factors, medical school name and primary specialty, probably strongly influence scores. Both the doctor's medical school name and the types of diseases and treatments could bias how scorers evaluate. For example, a scorer may be predisposed to giving a higher score if the doctor being evaluated attended a more prestigious school. Certain fields of medicine, such as surgery or oncology, could incur much higher costs or have higher mortality rates compared to fields such as family medicine, skewing factors of the final MIPS score, such as cost and quality scoring. Still, even with fairly good accuracies, we can see that the model isn't calibrated in an ideal manner.

## GLM Models – Logit and GLMnet Penalized

Resplit to only put half in each to make functions run a bit faster

```
in_train <- createDataPartition(no_tasks$MIPS75_Dr,  
                                p = 0.6, list = F)  
training <- no_tasks[in_train, ]  
testing <- no_tasks[-in_train, ]  
  
training <- training[, c(3:4, 8, 12:15, 17:22)]  
testing <- testing[, c(3:4, 8, 12:15, 17:22)]
```

```
# even splits of factors with several levels  
# returns padded resampled version though!
```



```

schools <- as.character(sort(unique(no_tasks$Medical.school.name)))
primary <- as.character(sort(unique(no_tasks$Primary.specialty)))
creds <- as.character(sort(unique(no_tasks$Credential)))
group_fctr <- c(schools, primary, creds)

in_train <- sapply(group_fctr, FUN = function(st) {
  task_st <- filter(no_tasks, (
    (Primary.specialty == st) | (Medical.school.name == st) | (Credential == st)))
  if (nrow(task_st) == 1) return(1L)
  return(createDataPartition(task_st$MIPS75_Dr,
                             p = 0.5, list = FALSE))
})
names(in_train) <- group_fctr
train_fctr <- bind_rows(lapply(group_fctr, FUN = function(st) {
  tasks_st <- filter(no_tasks, (
    (Primary.specialty == st) | (Medical.school.name == st) | (Credential == st)))
  x <- in_train[[st]]
  return(tasks_st[x, , drop = FALSE])
})))
test_fctr <- bind_rows(lapply(group_fctr, FUN = function(st) {
  task_st <- filter(no_tasks, (
    (Primary.specialty == st) | (Medical.school.name == st) | (Credential == st)))
  x <- in_train[[st]]
  return(task_st[-x, , drop = FALSE])
})))

train_fctr <- train_fctr[, c(3:5, 7:8, 12:15, 17:22)] %>% distinct(.)
test_fctr <- test_fctr[, c(3:5, 7:8, 12:15, 17:22)] %>% distinct(.)

logit_strata <- glm(MIPS75_Dr ~ ., data = train_fctr, family = binomial(link = "logit"))
z_strata <- predict(logit_strata, newdata = test_fctr, type = "response") > 0.5
z_strata <- factor(z_strata, levels = c(TRUE, FALSE), labels = c("no", "yes"), order = T)
confusionMatrix(z_strata, test_fctr$MIPS75_Dr)

## Confusion Matrix and Statistics
##
##              Reference
## Prediction   yes    no
##          yes 13122   63
##          no   217  5782
##
##              Accuracy : 0.9854
##              95% CI : (0.9836, 0.9871)
##      No Information Rate : 0.6953
##      P-Value [Acc > NIR] : < 2.2e-16
##
##              Kappa : 0.9658
##
##  Mcnemar's Test P-Value : < 2.2e-16
##
##              Sensitivity : 0.9837
##              Specificity : 0.9892
##              Pos Pred Value : 0.9952

```

```
##          Neg Pred Value : 0.9638
##          Prevalence : 0.6953
##          Detection Rate : 0.6840
##          Detection Prevalence : 0.6873
##          Balanced Accuracy : 0.9865
##
##          'Positive' Class : yes
##
```

```
# too many factors just not in both training and testing, level-wise

#logit <- glm(MIPS75_Dr ~ ., data = training, family = binomial(link = "logit"))
#z <- predict(logit, newdata = testing, type = "response") > 0.5
#z <- factor(z, levels = c(TRUE, FALSE), labels = c("no", "yes"), order = T)
#confusionMatrix(z, testing$MIPS75_Dr)
```

Wow glm fit way better than I thought so quickly

```
# glmnet
ctrl <- trainControl(method = "cv", number = 3)

enet_strata <- train(formula(logit_strata), data = train_fctr, method = "glmnet",
                     trControl = ctrl, tuneLength = 10)
enet_hat_strata <- predict(enet_strata, newdata = test_fctr)
confusionMatrix(enet_hat_strata, reference = test_fctr$MIPS75_Dr)
```

```
## Confusion Matrix and Statistics
##
##          Reference
## Prediction  yes    no
##          yes 13335   22
##          no    4   5823
##
##          Accuracy : 0.9986
##          95% CI : (0.998, 0.9991)
##          No Information Rate : 0.6953
##          P-Value [Acc > NIR] : < 2.2e-16
##
##          Kappa : 0.9968
##
##          Mcnemar's Test P-Value : 0.0008561
##
##          Sensitivity : 0.9997
##          Specificity : 0.9962
##          Pos Pred Value : 0.9984
##          Neg Pred Value : 0.9993
##          Prevalence : 0.6953
##          Detection Rate : 0.6951
##          Detection Prevalence : 0.6963
##          Balanced Accuracy : 0.9980
##
```

```
##          'Positive' Class : yes  
##
```

```
#enet <- train(formula(logit), data = training, method = "glmnet",  
#              trControl = ctrl, tuneLength = 10)  
#enet_hat <- predict(enet, newdata = testing)  
#confusionMatrix(enet_hat, reference = testing$MIPS75_Dr)
```