PQRS Prediction from Physician database

Arun Chiriyankandath

22/04/2020

Introduction

The aim of the project is to predict the PQRS(Physician Quality Reporting System) participation of different physicians and try to understand whether we can predict this using the other variables in the physician compare data-set. In the project, I am trying to study a customized data-set having 10000 records which I have created from the physician compare data-set which I got from the CMS website. The original file is a huge file which is over 549 MB. It is having over 20 Million records and 40 variables. Since we have planned to use random forest, Support Vector Machine and Extreme Gradient Boost we have kept the size of the data-set to 10000. All these models are very computational intensive. The key steps which we are going to conduct are following.

1. Data Exploration and Visualization - In this section, we will try to understand more about the data and the variables.
2. Data Wrangling - In this section, we will try to reduce some variables and add some variables according to the existing data.
3. Data Modeling - In this section, we will model the data using the models which we explained above.
4. Results - In this section, we will showcase the results and compare the models and try to select a best model for the data in our hand.
5. Conclusion - In this section, we will gives a brief summary of the report, its potential impact, its limitations, and future work.

Installing Packages

The below code installs all the required packages.

```
if(!require(data.table)) install.packages("data.table", repos = "http://cran.us.r-project.org")
if(!require(dplyr)) install.packages("dplyr", repos = "http://cran.us.r-project.org")
if(!require(ggplot2)) install.packages("ggplot2", repos = "http://cran.us.r-project.org")
if(!require(randomForest)) install.packages("randomForest", repos = "http://cran.us.r-project.org")
if(!require(caret)) install.packages("caret", repos = "http://cran.us.r-project.org")
if(!require(corrplot)) install.packages("corrplot", repos = "http://cran.us.r-project.org")
if(!require(pROC)) install.packages("pROC", repos = "http://cran.us.r-project.org")
if(!require(e1071)) install.packages("e1071", repos = "http://cran.us.r-project.org")
if(!require(xgboost)) install.packages("xgboost", repos = "http://cran.us.r-project.org")

#opening the libraries
library(corrplot)
library(caret)
library(data.table)
library(dplyr)
```

```
library(ggplot2)
library(randomForest)
library(pROC)
library(e1071)
library(xgboost)
```

Following is the look-up table functionality created for categorizing the US states to different region.

```
NE.name <- c("Connecticut","Maine","Massachusetts","New Hampshire",
             "Rhode Island","Vermont","New Jersey","New York",
             "Pennsylvania")
NE.abrv <- c("CT","ME","MA","NH","RI","VT","NJ","NY","PA")
NE.ref <- c(NE.name,NE.abrv)

IS.name <- c("Puerto Rico","Marshall Islands","Guam","Virgin Islands")
IS.abrv <- c("PR","MH","GU","VI")
IS.ref <- c(IS.name,IS.abrv)

MW.name <- c("Indiana","Illinois","Michigan","Ohio","Wisconsin",
             "Iowa","Kansas","Minnesota","Missouri","Nebraska",
             "North Dakota","South Dakota")
MW.abrv <- c("IN","IL","MI","OH","WI","IA","KS","MN","MO","NE",
             "ND","SD")
MW.ref <- c(MW.name,MW.abrv)

S.name <- c("Delaware","District of Columbia","Florida","Georgia",
            "Maryland","North Carolina","South Carolina","Virginia",
            "West Virginia","Alabama","Kentucky","Mississippi",
            "Tennessee","Arkansas","Louisiana","Oklahoma","Texas")
S.abrv <- c("DE","DC","FL","GA","MD","NC","SC","VA","WV","AL",
            "KY","MS","TN","AR","LA","OK","TX")
S.ref <- c(S.name,S.abrv)

W.name <- c("Arizona","Colorado","Idaho","New Mexico","Montana",
            "Utah","Nevada","Wyoming","Alaska","California",
            "Hawaii","Oregon","Washington")
W.abrv <- c("AZ","CO","ID","NM","MT","UT","NV","WY","AK","CA",
            "HI","OR","WA")
W.ref <- c(W.name,W.abrv)

region.list <- list(
  Northeast=NE.ref,
  Midwest=MW.ref,
  South=S.ref,
  West=W.ref,
  Island=IS.ref)
```

Downloading the file

The following code downloads the data-set from the git-hub repository.

```
phys <- tempfile()
#You can either download the file to the "data" folder in your project directory and set to that direct
#phys <- read.csv("./data/phy_small.csv")
phys <- read.csv("https://raw.githubusercontent.com/arunchiri/CYO_project/master/data/phy_small.csv")
```

Dataset details

We can use the glimpse function for find the variables used in the data-set and the type of the variables used. So this gives us count of records, type of variables and the number of variables. The further study on the variables which will be done in the data exploration section.

```
#colnames shows the vaiables used in the datasets
colnames(phys)
```

```
##  [1] "NPI"
##  [2] "PAC.ID"
##  [3] "Professional.Enrollment.ID"
##  [4] "Last.Name"
##  [5] "First.Name"
##  [6] "Middle.Name"
##  [7] "Suffix"
##  [8] "Gender"
##  [9] "Credential"
## [10] "Medical.school.name"
## [11] "Graduation.year"
## [12] "Primary.specialty"
## [13] "Secondary.specialty.1"
## [14] "Secondary.specialty.2"
## [15] "Secondary.specialty.3"
## [16] "Secondary.specialty.4"
## [17] "All.secondary.specialties"
## [18] "Organization.legal.name"
## [19] "Group.Practice.PAC.ID"
## [20] "Number.of.Group.Practice.members"
## [21] "Line.1.Street.Address"
## [22] "Line.2.Street.Address"
## [23] "Marker.of.address.line.2.suppression"
## [24] "City"
## [25] "State"
## [26] "Zip.Code"
## [27] "Claims.based.hospital.affiliation.CCN.1"
## [28] "Claims.based.hospital.affiliation.LBN.1"
## [29] "Claims.based.hospital.affiliation.CCN.2"
## [30] "Claims.based.hospital.affiliation.LBN.2"
## [31] "Claims.based.hospital.affiliation.CCN.3"
## [32] "Claims.based.hospital.affiliation.LBN.3"
## [33] "Claims.based.hospital.affiliation.CCN.4"
## [34] "Claims.based.hospital.affiliation.LBN.4"
## [35] "Claims.based.hospital.affiliation.CCN.5"
## [36] "Claims.based.hospital.affiliation.LBN.5"
## [37] "Professional.accepts.Medicare.Assignment"
## [38] "Participating.in.eRx"
```

```
## [39] "Participating.in.PQRS"
## [40] "Participating.in.EHR"
```

Analysis

In this section, we are doing the following steps :- 1) data exploration and visualization, 2) data cleaning and Structuring 3) Modeling approaches

Data Exploration

Here we are further verifying the data using the glimpse function to understand the attributes and the types.

```
#glimpse shows the vaiables used in the datasets
glimpse(phys)
```

```
## Observations: 10,000
## Variables: 40
## $ NPI                                        <int> 1184828550, 1508965104, 15...
## $ PAC.ID                                     <dbl> 5890732267, 4082633680, 18...
## $ Professional.Enrollment.ID                 <fct> I20050418000159, I20051114...
## $ Last.Name                                  <fct> GILCHRIST, MEMOLI, ROMEO, ...
## $ First.Name                                 <fct> DONNA, KAREN, FRED, STEVEN...
## $ Middle.Name                                <fct> , M, P, A, , , , ALBERT, A...
## $ Suffix                                     <fct> , , , , , , , , , , , , , ...
## $ Gender                                     <fct> F, F, M, M, M, M, F, M, F,...
## $ Credential                                 <fct> CSW, OD, , MD, , MD, , DPM...
## $ Medical.school.name                        <fct> OTHER, UNIVERSITY ALABAMA ...
## $ Graduation.year                            <int> 1998, 1999, 1990, 1987, 19...
## $ Primary.specialty                          <fct> CLINICAL SOCIAL WORKER, OP...
## $ Secondary.specialty.1                      <fct> , , , , , , , , GENERAL SU...
## $ Secondary.specialty.2                      <fct> , , , , , , , , , , , , , ...
## $ Secondary.specialty.3                      <fct> , , , , , , , , , , , , , ...
## $ Secondary.specialty.4                      <fct> , , , , , , , , , , , , , ...
## $ All.secondary.specialties                  <fct> "", "", "", "", "", "", ""...
## $ Organization.legal.name                    <fct> "INTEGRATED THERAPEUTIC SE...
## $ Group.Practice.PAC.ID                      <dbl> 4789850256, 749187847, 660...
## $ Number.of.Group.Practice.members           <int> 1, 21, 74, 48, 2, 2, 1, 9,...
## $ Line.1.Street.Address                      <fct> 985 PATTON ST, 4101 EVANS ...
## $ Line.2.Street.Address                      <fct> , , 220 SURGICAL ONCOLOGY ...
## $ Marker.of.address.line.2.suppression       <fct> N, N, N, N, N, N, N, N, N,...
## $ City                                       <fct> NORTH BRUNSWICK, FORT MYER...
## $ State                                      <fct> NJ, FL, OH, VA, CA, FL, TX...
## $ Zip.Code                                   <int> 89022285, 339019310, 43213...
## $ Claims.based.hospital.affiliation.CCN.1    <int> NA, NA, 360035, 490043, NA...
## $ Claims.based.hospital.affiliation.LBN.1    <fct> "", "", "MOUNT CARMEL HEAL...
## $ Claims.based.hospital.affiliation.CCN.2    <int> NA, NA, 360012, 490063, NA...
## $ Claims.based.hospital.affiliation.LBN.2    <fct> "", "", "MOUNT CARMEL HEAL...
## $ Claims.based.hospital.affiliation.CCN.3    <int> NA, NA, NA, 490101, NA, 10...
## $ Claims.based.hospital.affiliation.LBN.3    <fct> , , , INOVA HEALTH CARE SE...
## $ Claims.based.hospital.affiliation.CCN.4    <int> NA, NA, NA, NA, NA, 100075...
## $ Claims.based.hospital.affiliation.LBN.4    <fct> "", "", "", "", "", "ST. J...
```

```
## $ Claims.based.hospital.affiliation.CCN.5  <int> NA, NA, NA, NA, NA, 100067...
## $ Claims.based.hospital.affiliation.LBN.5  <fct> "", "", "", "", "", "ST. A...
## $ Professional.accepts.Medicare.Assignment <fct> Y, Y, M, M, Y, Y, Y, Y, Y,...
## $ Participating.in.eRx                     <fct> N, N, N, N, N, Y, N, N, N,...
## $ Participating.in.PQRS                    <fct> N, N, N, Y, N, N, N, N, N,...
## $ Participating.in.EHR                     <fct> N, N, N, N, N, Y, N, Y, Y,...
```

From the results of glimpse function we could understand that the following variables can contribute little to the data Analysis. They are personal identifiers and we are going to remove them from our data-set which we will be used to model. 1. NPI 2. PAC.ID 3. Professional.Enrollment.ID 4. Last Name 5. First Name 6. Middle Name 7. Suffix 8. Line.1.Street.Address 9. Line.2.Street.Address

The following secondary specialty information also have less importance compared to the primary specialty information. Most of the secondary specialty information is blank which we can see from the summary.

```
summary(phys)
```

```
##       NPI                PAC.ID            Professional.Enrollment.ID
##  Min.   :1.003e+09   Min.   :4.210e+07   I20031210000254:   3
##  1st Qu.:1.255e+09   1st Qu.:2.567e+09   I20050302000827:   3
##  Median :1.509e+09   Median :4.982e+09   I20070307000352:   3
##  Mean   :1.501e+09   Mean   :5.011e+09   I20110118001150:   3
##  3rd Qu.:1.750e+09   3rd Qu.:7.517e+09   I20120510000198:   3
##  Max.   :1.993e+09   Max.   :9.931e+09   I20031230000662:   2
##                                          (Other)        :9983
##     Last.Name      First.Name    Middle.Name   Suffix     Gender
##  SMITH   :  53   DAVID  : 220          :2519      :9755   F:3693
##  JOHNSON :  42   MICHAEL: 218   A      : 789   I  :   3   M:6307
##  MILLER  :  35   JOHN   : 217   M      : 694   II :  19
##  WILLIAMS:  33   JAMES  : 192   J      : 600   III:  47
##  PATEL   :  31   ROBERT : 182   L      : 534   IV :   6
##  LEE     :  30   WILLIAM: 126   E      : 370   JR.: 162
##  (Other) :9776   (Other):8845   (Other):4494   SR.:   8
##    Credential
##         :5764
##  MD     :2555
##  DC     : 311
##  OD     : 218
##  PT     : 213
##  NP     : 175
##  (Other): 764
##                                                 Medical.school.name
##  OTHER                                                         :4428
##  PALMER COLLEGE CHIROPRACTIC - DAVENPORT                       : 114
##  PENNSYLVANIA COLLEGE OF OPTOMETRY                             :  71
##  WAYNE STATE UNIVERSITY SCHOOL OF MEDICINE                     :  68
##  UNIVERSITY OF ILLINOIS AT CHICAGO HEALTH SCIENCE CENTER:  64
##  ILLINOIS COLLEGE OF OPTOMETRY AT CHICAGO                      :  61
##  (Other)                                                       :5194
##  Graduation.year          Primary.specialty
##  Min.   :1945    INTERNAL MEDICINE : 834
##  1st Qu.:1982    CHIROPRACTIC      : 790
##  Median :1992    FAMILY PRACTICE   : 766
##  Mean   :1991    PHYSICAL THERAPIST: 680
```

```
##  3rd Qu.:2001    NURSE PRACTITIONER: 575
##  Max.   :2013    OPTOMETRY         : 516
##                  (Other)           :5839
##                   Secondary.specialty.1       Secondary.specialty.2
##                                 :8926                       :9879
##  INTERNAL MEDICINE           : 320      INTERNAL MEDICINE:  29
##  CRITICAL CARE (INTENSIVISTS):  60      PAIN MANAGEMENT  :  13
##  PEDIATRIC MEDICINE          :  48      VASCULAR SURGERY :  13
##  EMERGENCY MEDICINE          :  41      PULMONARY DISEASE:   9
##  GERIATRIC MEDICINE          :  41      THORACIC SURGERY :   6
##  (Other)                     : 564      (Other)          :  51
##            Secondary.specialty.3  Secondary.specialty.4
##                          :9989                  :9999
##  GENERAL SURGERY      :   1      HAND SURGERY:   1
##  INTERNAL MEDICINE    :   1
##  MAXILLOFACIAL SURGERY:   1
##  MEDICAL ONCOLOGY     :   1
##  NEUROPSYCHIATRY      :   1
##  (Other)              :   6
##       All.secondary.specialties
##                      :8926
##  INTERNAL MEDICINE : 310
##  PEDIATRIC MEDICINE:  48
##  GERIATRIC MEDICINE:  38
##  GENERAL SURGERY   :  34
##  EMERGENCY MEDICINE:  33
##  (Other)           : 611
##                            Organization.legal.name Group.Practice.PAC.ID
##                                           :1536       Min.   :4.210e+07
##  BAYSTATE MEDICAL PRACTICES INC           :  66       1st Qu.:2.567e+09
##  PROVIDENCE HEALTH & SERVICES - OREGON:  28       Median :5.093e+09
##  ST JOHN HOSPITAL AND MEDICAL CENTER  :  19       Mean   :5.021e+09
##  ROCKINGHAM MEMORIAL HOSPITAL         :  17       3rd Qu.:7.517e+09
##  DARTMOUTH-HITCHCOCK CLINIC           :  16       Max.   :9.931e+09
##  (Other)                              :8318       NA's   :1521
##  Number.of.Group.Practice.members              Line.1.Street.Address
##  Min.   :  1.00                   3300 MAIN ST             :  13
##  1st Qu.:  1.00                   3400 MAIN ST             :  13
##  Median :  4.00                   5050 NE HOYT             :  13
##  Mean   : 28.19                   2 MEDICAL CTR DR         :  10
##  3rd Qu.: 17.00                   759 CHESTNUT ST WESSON G:   9
##  Max.   :924.00                   1000 S AVE               :   8
##  NA's   :22                       (Other)                  :9934
##  Line.2.Street.Address Marker.of.address.line.2.suppression          City
##         :7536          N:9224                                 NEW YORK   : 143
##  100    : 127          Y: 776                                 SPRINGFIELD: 114
##  200    : 102                                                 HOUSTON    :  88
##  101    :  66                                                 CHICAGO    :  80
##  1      :  64                                                 BROOKLYN   :  63
##  2      :  54                                                 COLUMBUS   :  59
##  (Other):2051                                                 (Other)    :9453
##      State          Zip.Code         Claims.based.hospital.affiliation.CCN.1
##  CA     : 914   Min.   :      603   Min.   : 10005
##  NY     : 810   1st Qu.:134354585   1st Qu.:110035
```

```
##  FL     : 665   Median :383013918   Median :230197
##  TX     : 602   Mean   :435822757   Mean   :251358
##  IL     : 464   3rd Qu.:731207904   3rd Qu.:380009
##  MA     : 401   Max.   :998018050   Max.   :670082
##  (Other):6144                       NA's   :4323
##                 Claims.based.hospital.affiliation.LBN.1
##                                     :4348
##  DIGNITY HEALTH                     :  64
##  BAYSTATE MEDICAL CENTER INC.       :  53
##  INOVA HEALTH CARE SERVICES         :  34
##  PROVIDENCE HEALTH & SERVICES-OREGON:  24
##  HCA HEALTHONE LLC                  :  22
##  (Other)                            :5455
##  Claims.based.hospital.affiliation.CCN.2
##  Min.   : 10001
##  1st Qu.:110163
##  Median :231301
##  Mean   :255390
##  3rd Qu.:380061
##  Max.   :800037
##  NA's   :6660
##                       Claims.based.hospital.affiliation.LBN.2
##                                                     :6689
##  DIGNITY HEALTH                                     :  34
##  INOVA HEALTH CARE SERVICES                         :  20
##  MEMORIAL HERMANN HEALTH SYSTEM                     :  18
##  ADVOCATE HEALTH AND HOSPITALS CORPORATION          :  16
##  PROVIDENCE HEALTH & SERVICES-OREGON                :  16
##  (Other)                                            :3207
##  Claims.based.hospital.affiliation.CCN.3
##  Min.   : 10005
##  1st Qu.:140010
##  Median :230273
##  Mean   :255407
##  3rd Qu.:370218
##  Max.   :670080
##  NA's   :8139
##             Claims.based.hospital.affiliation.LBN.3
##                                           :8153
##  INOVA HEALTH CARE SERVICES               :  17
##  DIGNITY HEALTH                           :  16
##  WILLIAM BEAUMONT HOSPITAL                :  15
##  MEMORIAL HERMANN HEALTH SYSTEM:  11
##  SENTARA PRINCESS ANNE HOSPITAL:  11
##  (Other)                                  :1777
##  Claims.based.hospital.affiliation.CCN.4
##  Min.   : 10023
##  1st Qu.:140174
##  Median :231334
##  Mean   :261341
##  3rd Qu.:390051
##  Max.   :670068
##  NA's   :8944
##             Claims.based.hospital.affiliation.LBN.4
```

```
##                                :8952
##  OAKWOOD HEALTHCARE INC        :  10
##  SENTARA PRINCESS ANNE HOSPITAL:  10
##  INOVA HEALTH CARE SERVICES    :   8
##  MAIN LINE HOSPITALS, INC.     :   7
##  TENET ST. MARY'S, INC.        :   7
##  (Other)                       :1006
##  Claims.based.hospital.affiliation.CCN.5
##  Min.   : 10029
##  1st Qu.:140223
##  Median :230303
##  Mean   :258435
##  3rd Qu.:370082
##  Max.   :800037
##  NA's   :9381
##                 Claims.based.hospital.affiliation.LBN.5
##                                      :9384
##  MEMORIAL HERMANN HEALTH SYSTEM      :   6
##  OAKWOOD HEALTHCARE INC              :   6
##  JACKSON COUNTY HEALTHCARE AUTHORITY:   5
##  ALEGENT CREIGHTON HEALTH            :   4
##  BOONE MEMORIAL HOSPITAL INC         :   4
##  (Other)                             : 591
##  Professional.accepts.Medicare.Assignment Participating.in.eRx
##  M:1743                                   N:8091
##  Y:8257                                   Y:1909
##
##
##
##
##
##  Participating.in.PQRS Participating.in.EHR
##  N:7849                N:7846
##  Y:2151                Y:2154
##
##
##
##
##
```

So based on the summary we can remove the following variables. 1. Claims.based.hospital.affiliation.LBN.1 2. Claims.based.hospital.affiliation.LBN.2 3. Claims.based.hospital.affiliation.LBN.3 4. Claims.based.hospital.affiliation.LBN.4 5. Claims.based.hospital.affiliation.LBN.5 6. Claims.based.hospital.affiliation.CCN.1 7. Claims.based.hospital.affiliation.CCN 8. Claims.based.hospital.affiliation.CCN.4

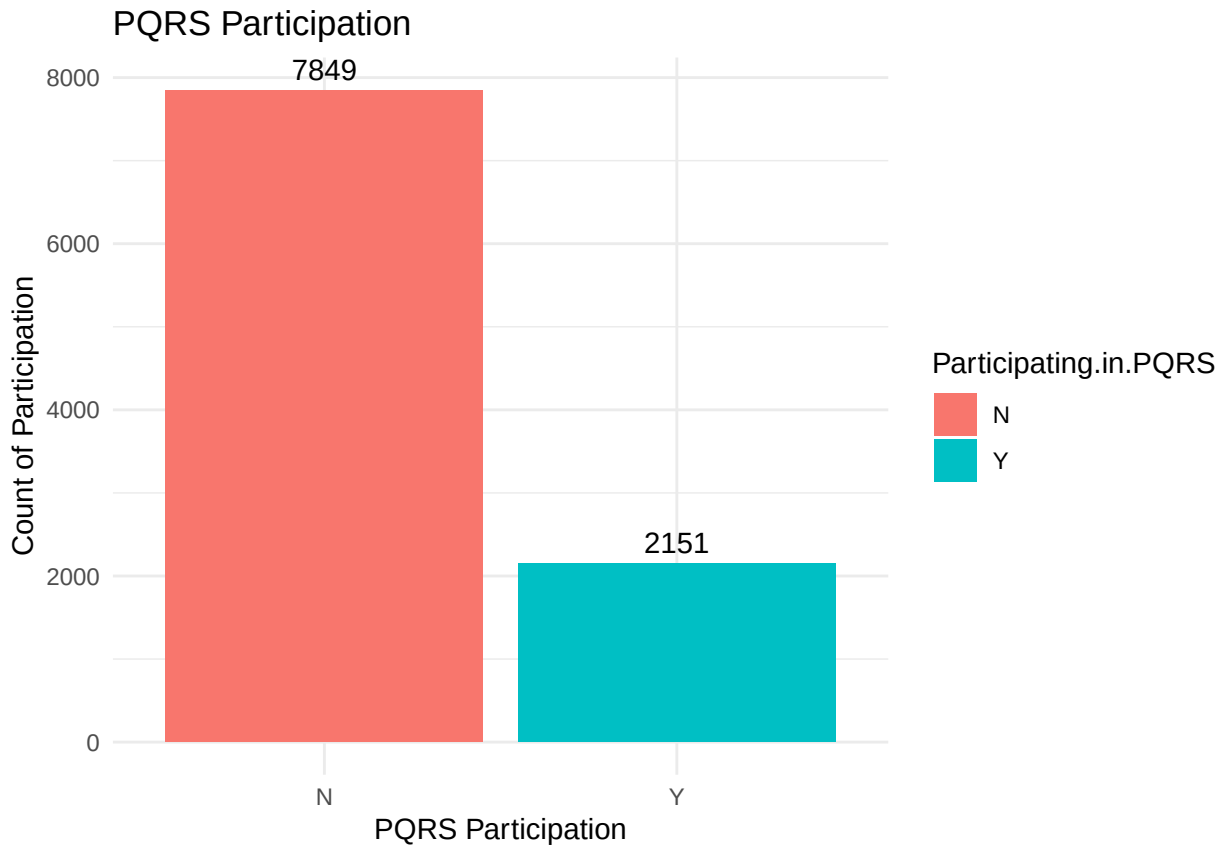
Lets see we can get more insights on the data Visualization.

Data Visualization

Plots of PQRS participation for different variables

The following plot will show the PQRS participation details from the data-set.

```
phys %>%
  group_by(Participating.in.PQRS) %>%
  tally() %>%
  ggplot(aes(x = Participating.in.PQRS, y = n,fill=Participating.in.PQRS)) +
  geom_bar(stat = "identity") +
  theme_minimal()+
  labs(x="PQRS Participation", y="Count of Participation")+
  ggtitle("PQRS Participation")+
  geom_text(aes(label = n), vjust = -0.5, position = position_dodge(0.9))
```



The visualization shows that majority of the providers are not participating in PQRS.

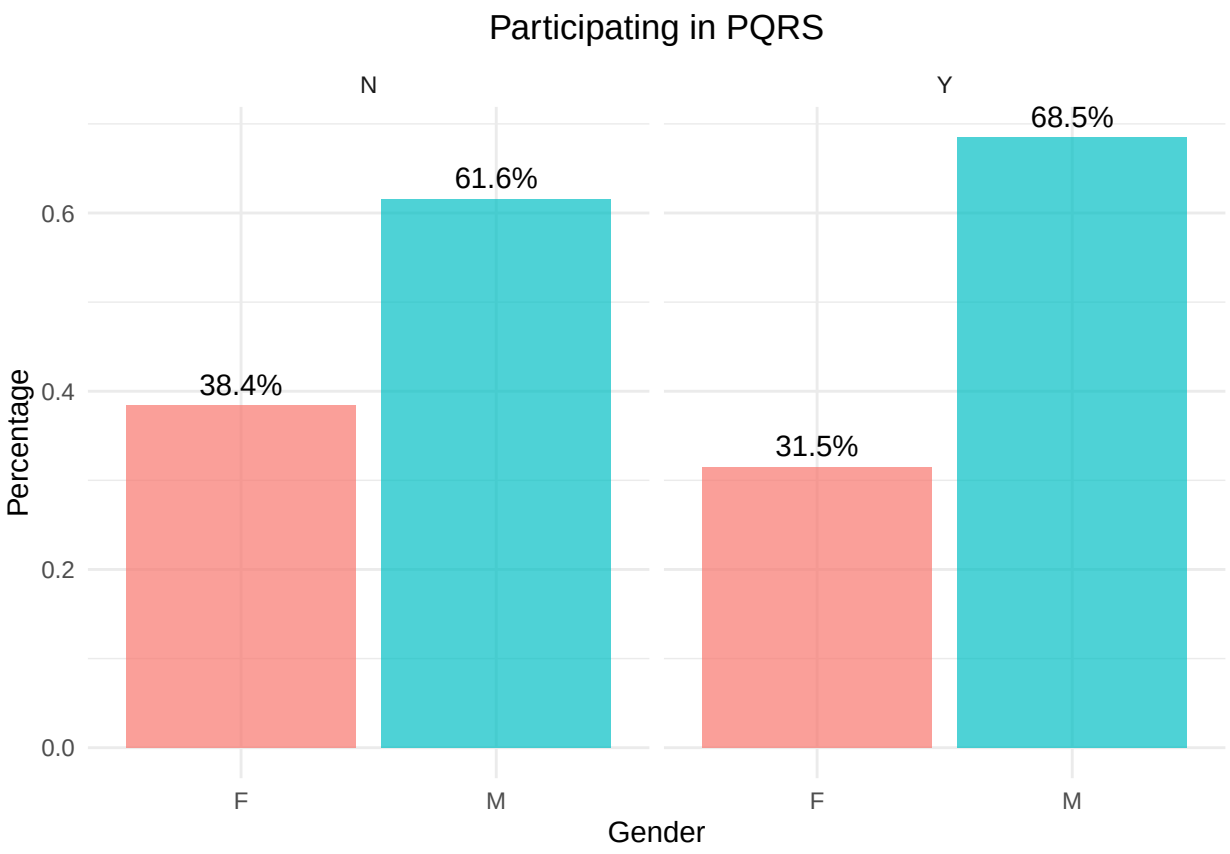
Lets see other attributes compared to the PQRS participation

Plot of PQRS Participation by gender

The below plot shows the PQRS participation based on gender

```
phys %>%
  ggplot(aes(x = Gender, group = Participating.in.PQRS)) +
  geom_bar(aes(y = ..prop.., fill = factor(..x..)),
           stat="count",
           alpha = 0.7) +
  geom_text(aes(label = scales::percent(..prop..), y = ..prop.. ),
            stat= "count",
            vjust = -.5) +
```

```
  labs(y = "Percentage", fill= "Gender") +
  facet_grid(~Participating.in.PQRS) +
  theme_minimal()+
  theme(legend.position = "none", plot.title = element_text(hjust = 0.5)) +
  ggtitle("Participating in PQRS")
```



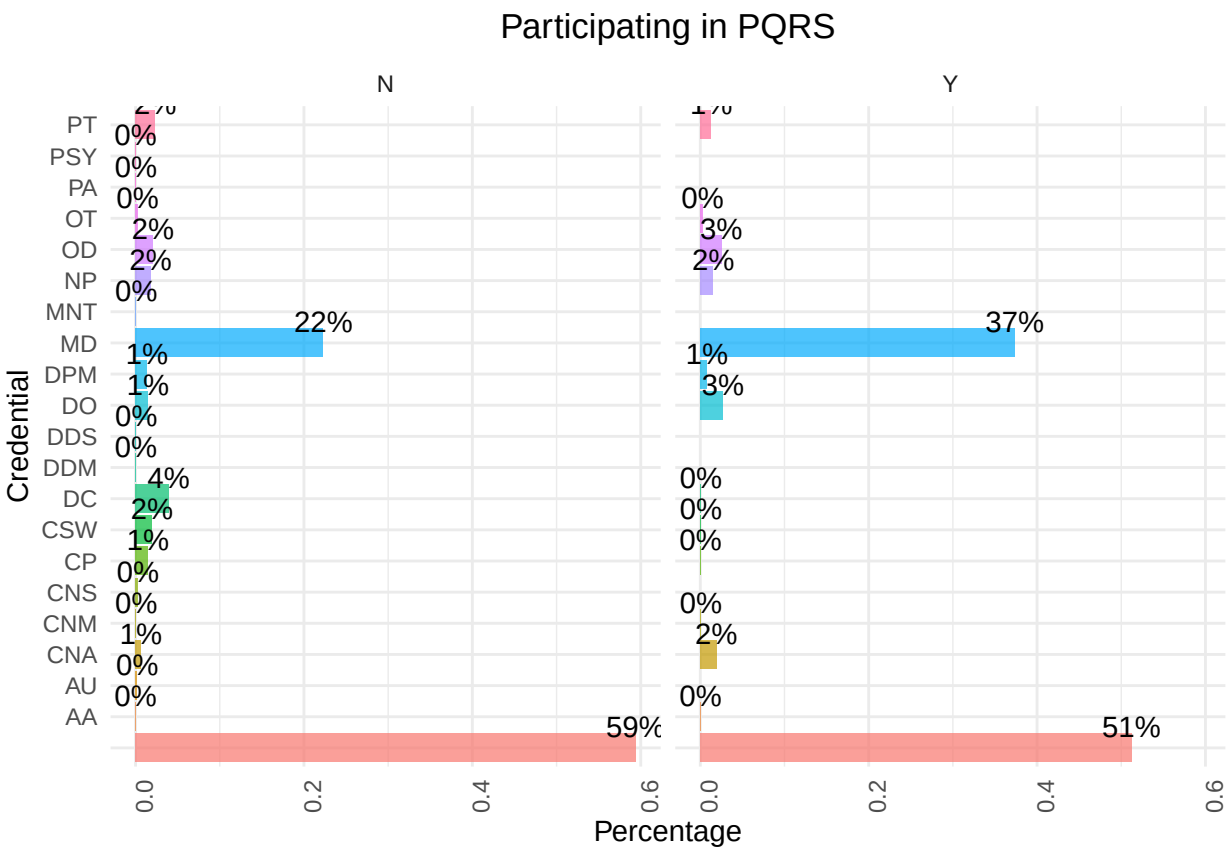
We can see based on the gender there is some variation in PQRS participation. Male Providers tend to participate more in PQRS. Female Providers shows a negative trend in participation. Most of them are not participating. So this data in male/female variations can be a driver in predicting the PQRS.

Plot of PQRS Participation by Credential

The below plot shows the PQRS participation based on Credential

```
phys %>%
  ggplot(aes(x = Credential, group = Participating.in.PQRS)) +
  geom_bar(aes(y = ..prop.., fill = factor(..x..)),
           stat="count",
           alpha = 0.7) +
  geom_text(aes(label = scales::percent(..prop..), y = ..prop.. ),
            stat= "count",
            vjust = -.5) +
  labs(y = "Percentage", fill= "Credential") +
  facet_grid(~Participating.in.PQRS) +
  theme_minimal()+
```

```
  theme(legend.position = "none", plot.title = element_text(hjust = 0.5)) +
  theme(axis.text.x = element_text(angle = 90, hjust = .5)) +
  ggtitle("Participating in PQRS") + coord_flip()
```

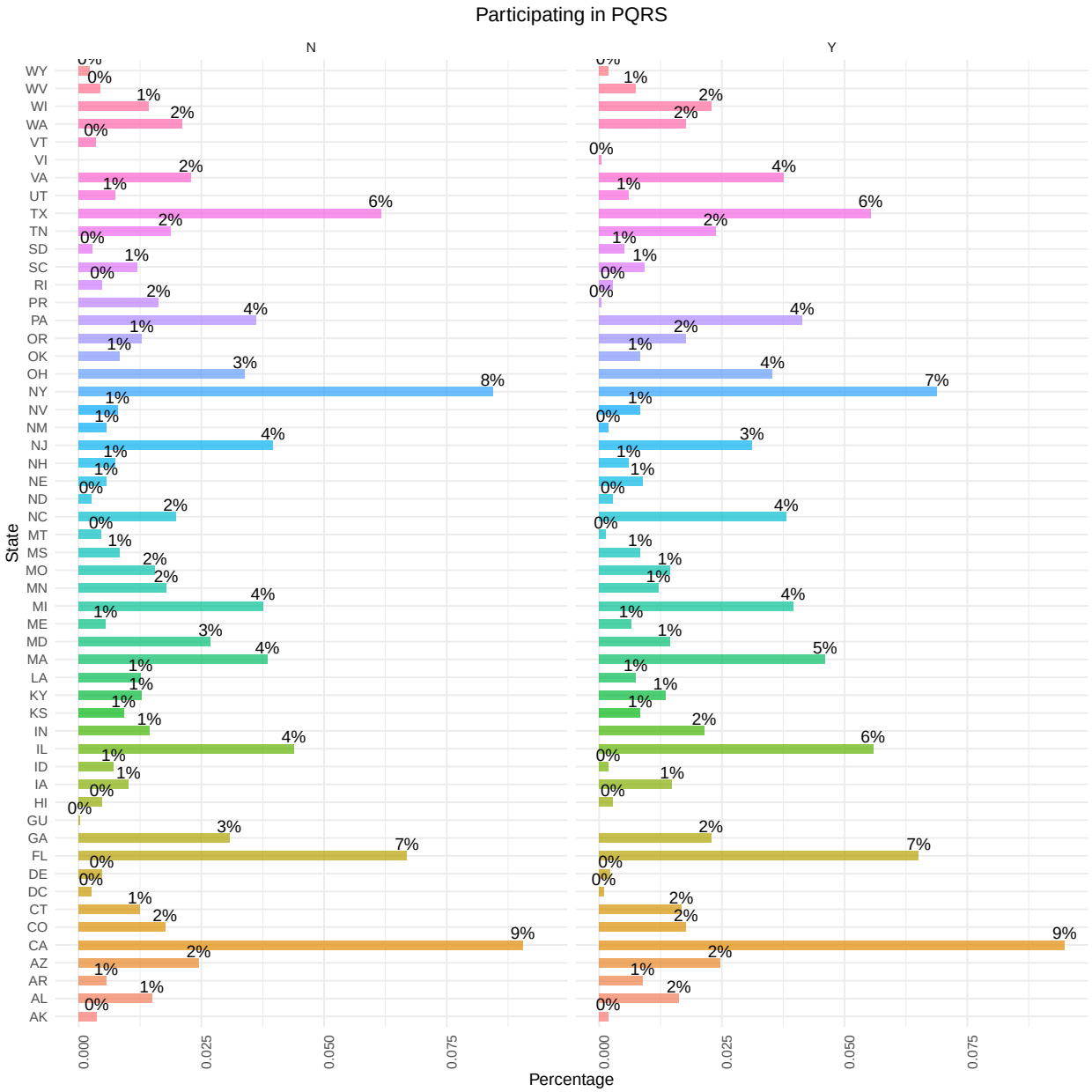


We can see from the plot that some credential holders are not at all participating. Some credential show more participation. So this data in credentials can be a driver in predicting the PQRS.

Plot of PQRS Participation by State

The below plot shows the PQRS participation by State

```
phys %>%
  ggplot(aes(x = State, group = Participating.in.PQRS )) +
  geom_bar(aes(y = ..prop.., fill = factor(..x..)),
           stat="count",
           alpha = 0.7, width = .5) +
  geom_text(aes(label = scales::percent(..prop..), y = ..prop.. ),
            stat= "count",
            vjust = -.5) +
  labs(y = "Percentage", fill= "State") +
  facet_grid(~Participating.in.PQRS) +
  theme_minimal()+
  theme(legend.position = "none", plot.title = element_text(hjust = .5)) +
  theme(axis.text.x = element_text(angle = 90, hjust = .5)) +
  ggtitle("Participating in PQRS") + coord_flip()
```

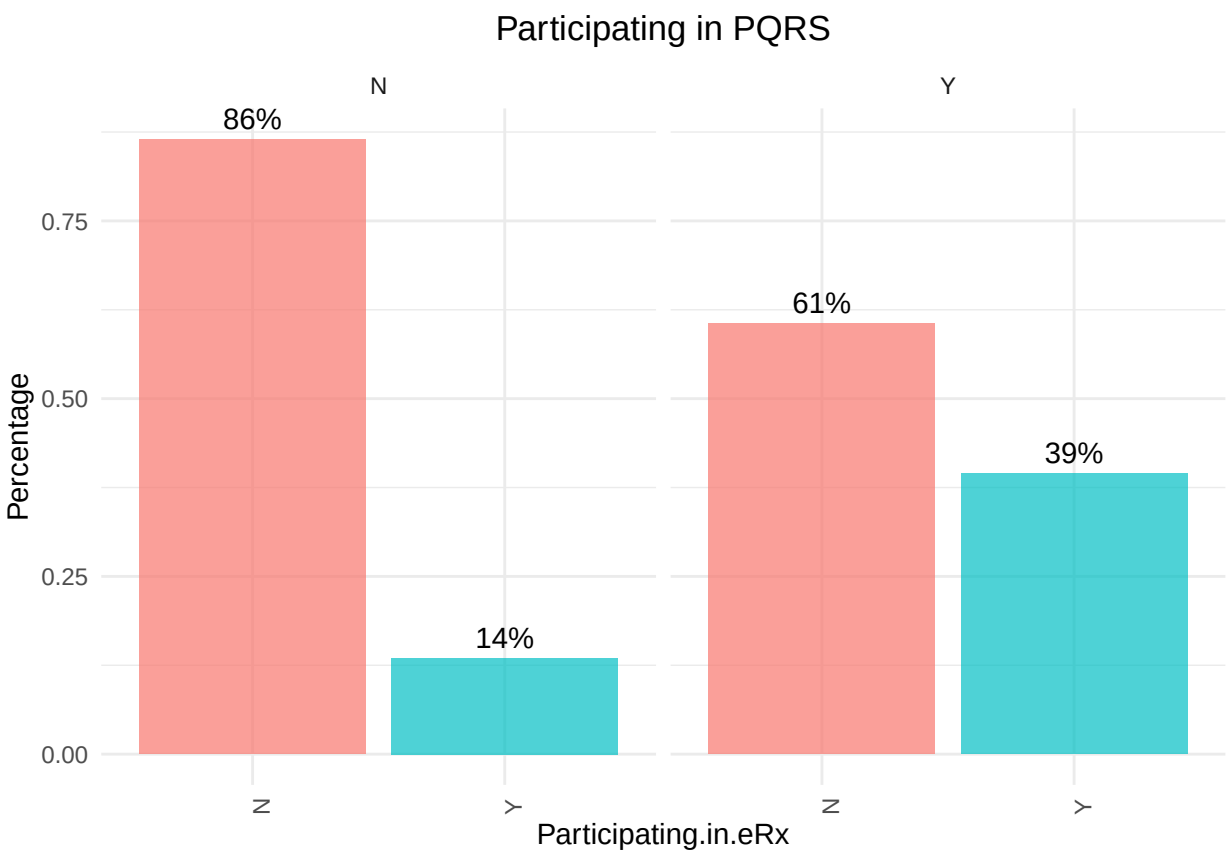


There are some states which show variability on the PQRS participation. Some states shows positive trend towards participation. Some states shows negative trend towards participation

Plot of PQRS Participation by eRX (Electronic Prescribing) participation

```
phys %>%
  ggplot(aes(x = Participating.in.eRx, group = Participating.in.PQRS)) +
  geom_bar(aes(y = ..prop.., fill = factor(..x..)),
           stat="count",
           alpha = 0.7) +
  geom_text(aes(label = scales::percent(..prop..), y = ..prop.. ),
            stat= "count",
            vjust = -.5) +
  labs(y = "Percentage", fill= "Participating.in.eRx") +
```

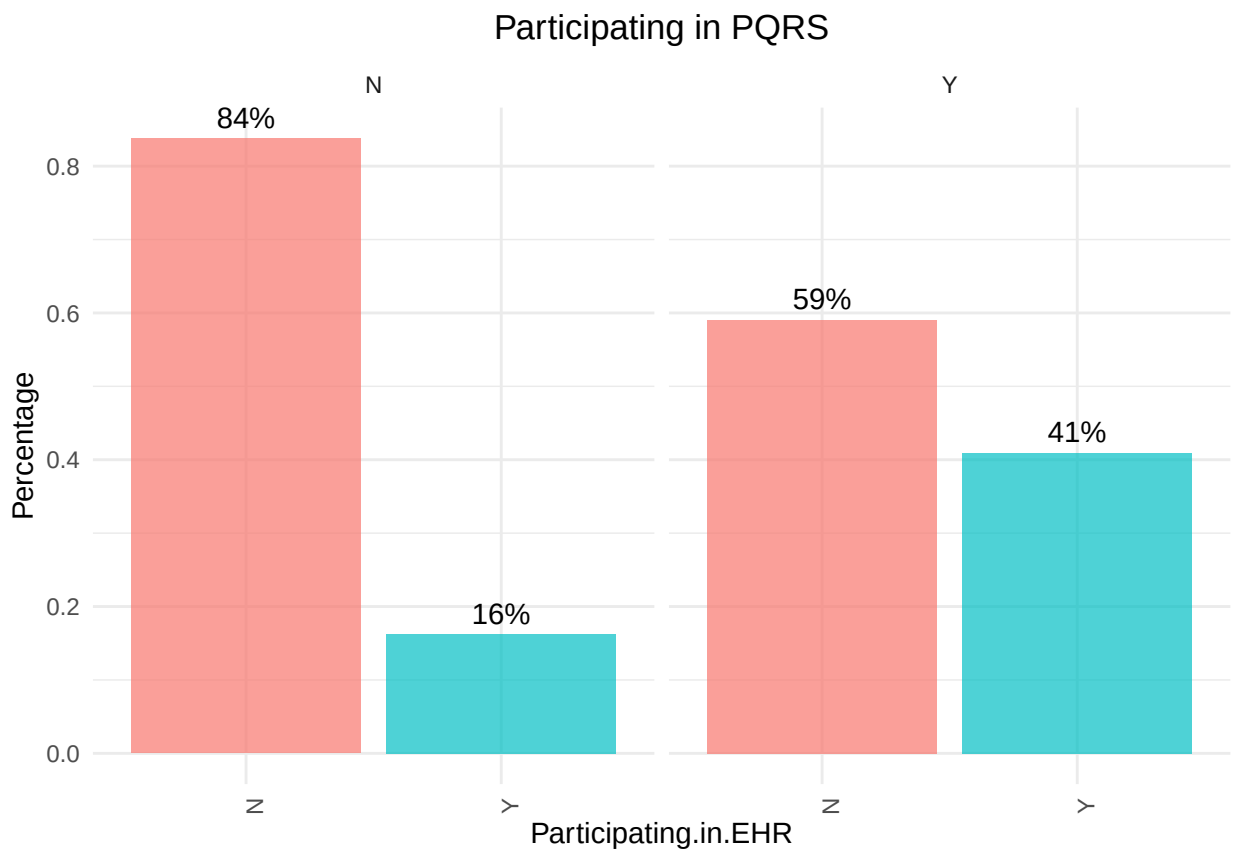
```
  facet_grid(~Participating.in.PQRS) +
  theme_minimal()+
  theme(legend.position = "none", plot.title = element_text(hjust = 0.5)) +
  theme(axis.text.x = element_text(angle = 90, hjust = .5)) +
  ggtitle("Participating in PQRS")
```



We can clearly see some correlation about eRx participation and PQRS participation from the above plot. There is a high chance that eRx participants, participate in PQRS.

Plot of PQRS Participation by EHR (Electronic Health Records) Participation

```
phys %>%
  ggplot(aes(x = Participating.in.EHR, group = Participating.in.PQRS)) +
  geom_bar(aes(y = ..prop.., fill = factor(..x..)),
           stat="count",
           alpha = 0.7) +
  geom_text(aes(label = scales::percent(..prop..), y = ..prop.. ),
            stat= "count",
            vjust = -.5) +
  labs(y = "Percentage", fill= "Participating.in.EHR") +
  facet_grid(~Participating.in.PQRS) +
  theme_minimal()+
  theme(legend.position = "none", plot.title = element_text(hjust = 0.5)) +
  theme(axis.text.x = element_text(angle = 90, hjust = .5)) +
  ggtitle("Participating in PQRS")
```

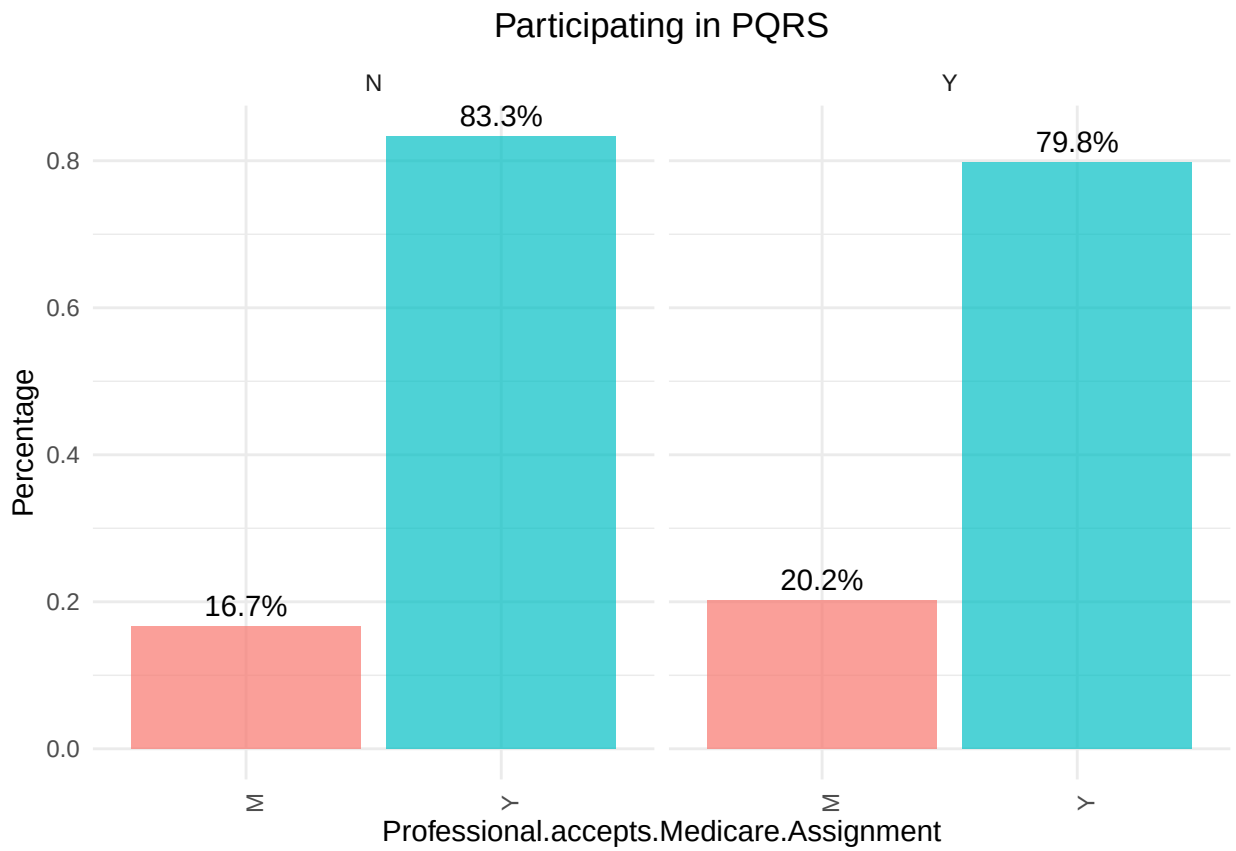


We can clearly see some correlation about EHR participation and PQRS participation from the above plot. There is a high chance that EHR participants, participate in PQRS.

Plot of PQRS Participation by PAMA (Professional.accepts.Medicare.Assignment) Participation

The below plot shows the PQRS participation by Professional.accepts.Medicare.Assignment participation

```
phys %>%
  ggplot(aes(x = Professional.accepts.Medicare.Assignment, group = Participating.in.PQRS)) +
  geom_bar(aes(y = ..prop.., fill = factor(..x..)),
           stat="count",
           alpha = 0.7) +
  geom_text(aes(label = scales::percent(..prop..), y = ..prop.. ),
            stat= "count",
            vjust = -.5) +
  labs(y = "Percentage", fill= "Professional.accepts.Medicare.Assignment") +
  facet_grid(~Participating.in.PQRS) +
  theme_minimal()+
  theme(legend.position = "none", plot.title = element_text(hjust = 0.5)) +
  theme(axis.text.x = element_text(angle = 90, hjust = .5)) +
  ggtitle("Participating in PQRS")
```

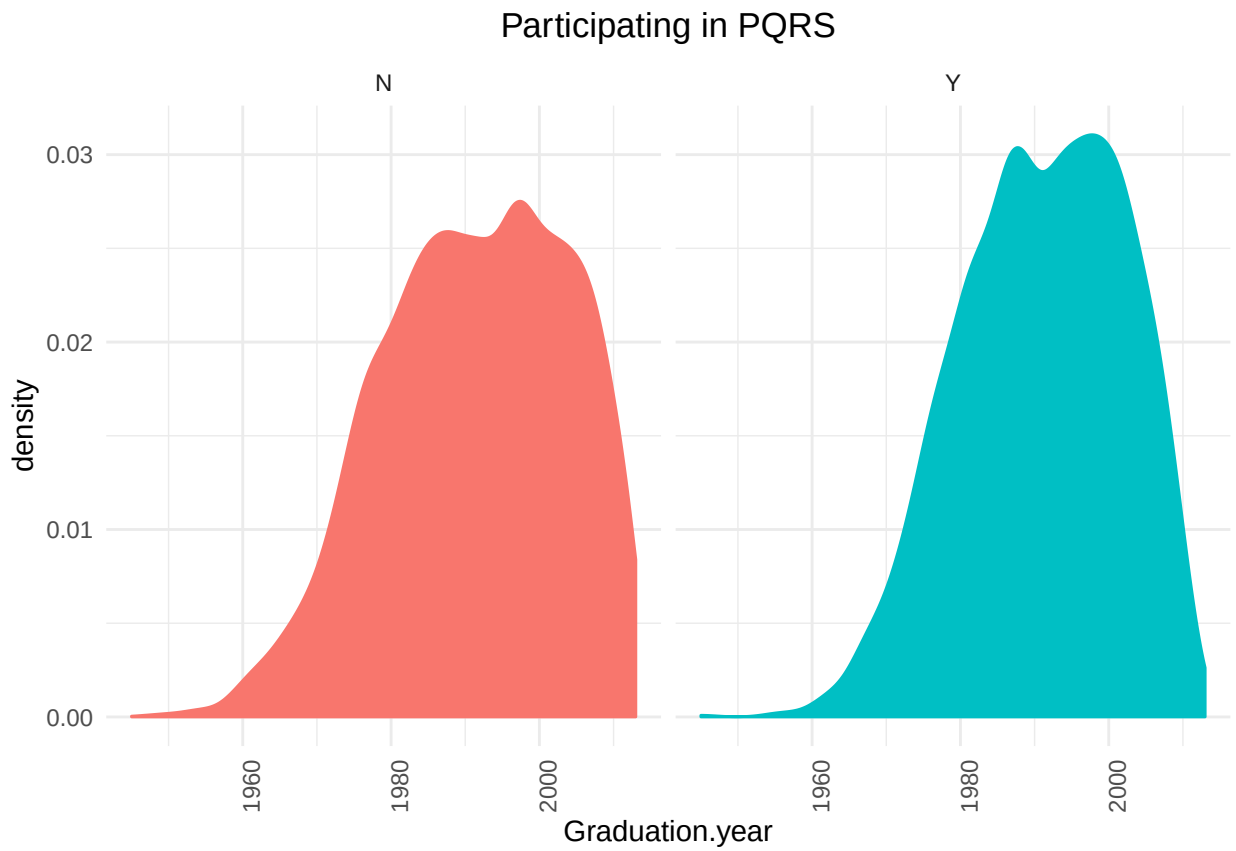


The PAMA participation shows very small correlation. Still we will use this as a variable in predicting the small changes. The plot seems equally distributed without much correlation.

Plot of PQRS Participation by Graduation year

The below plot shows the variations in the PQRS participation by Graduation year

```
phys %>%
  ggplot(aes(Graduation.year, fill = Participating.in.PQRS, colour = Participating.in.PQRS)) +
geom_density() +
  facet_grid(~Participating.in.PQRS) +
  theme_minimal()+
  theme(legend.position = "none", plot.title = element_text(hjust = 0.5)) +
  theme(axis.text.x = element_text(angle = 90, hjust = .5)) +
  ggtitle("Participating in PQRS")
```



We can see that the participation in PQRS has a increase depends on the graduation year. So the graduation year can be used as a predictor for predicting the PQRS participation.

Data Wrangling

The steps which we are following in data wrangling are the Structuring of the data for our analysis and cleaning the data

Structuring & Cleaning

First we are replacing all the integer NAs from the dataset with 0.

```
phys <- phys %>% mutate_if(is.integer, ~replace(., is.na(.), 0))
```

From Data analysis and visualization we conclude to use the following variables in our data modeling.

Following are their column names we selected based on the importance from the initial analysis. We selected states because, in US there are state level mandates in Healthcare. So we are not considering the city and ZIP. Even city level analysis can also be done. But there are lot of city information is there and categorizing the city and doing the analysis can be considered for future enhancement of this project. We are also not including any specialty information. This is done based on the thought that PQRS participation is not related to specialty.

1. Gender - column 8
2. Credential - column 9

3. State - column 25
4. Professional.accepts.Medicare.Assignment - column 37
5. Participating.in.eRx - column 38
6. Participating.in.PQRS - column 39
7. Participating.in.EHR - column 40

Categorizing the Graduation Year

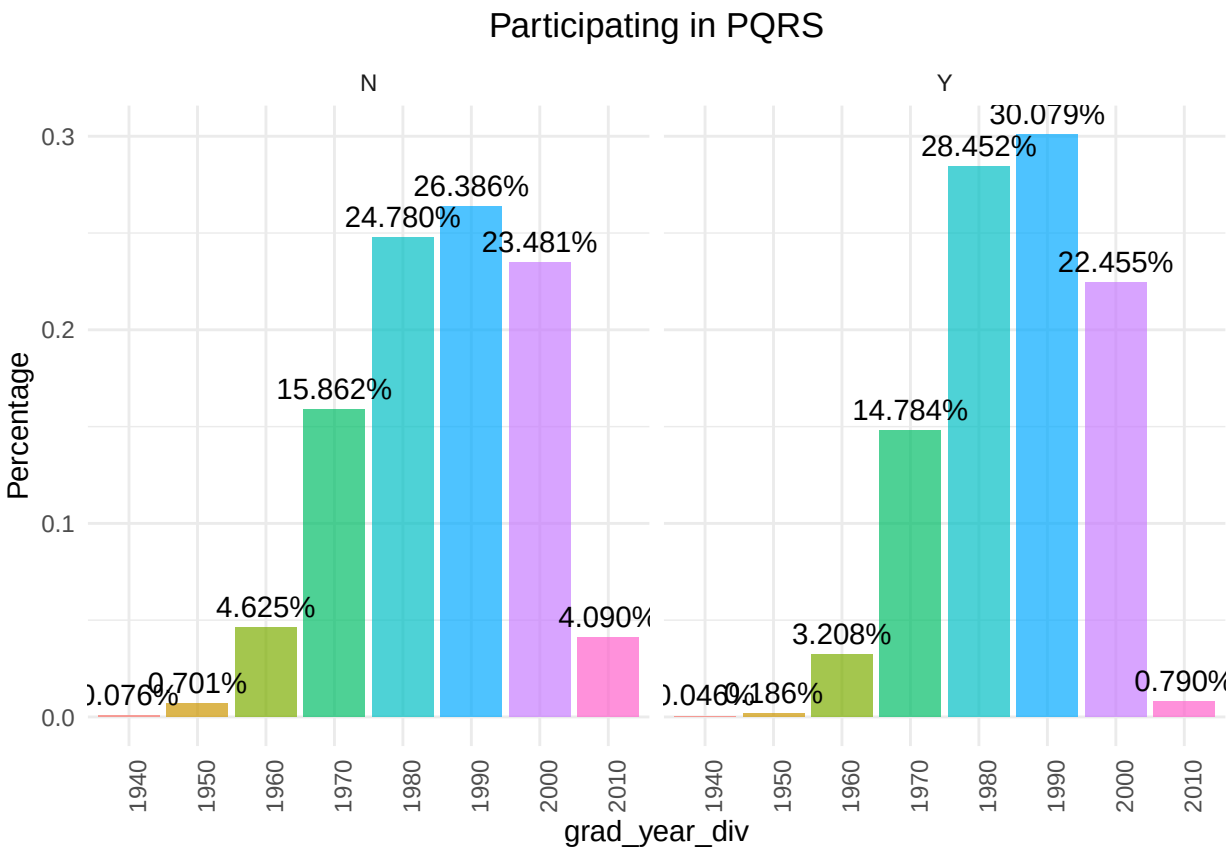
When we see the graduation year it's a continuous variable. So we are trying to factorize the years to different year divisions. Also we will visualize the participation in PQRS based on this year divisions.

Following code will do that.

```
phys_T <- phys
phys_T$grad_year_div <- as.factor(cut(phys_T$Graduation.year,breaks = c(0,1940,1950,1960,1970,1980,1990
                            labels = c("1930","1940","1950","1960","1970","1980","1990","2000","2010")))
```

The following plot will visualize the participation of PQRS based on these year divisions.

```
phys_T %>%
  ggplot(aes(x = grad_year_div, group = Participating.in.PQRS)) +
  geom_bar(aes(y = ..prop.., fill = factor(..x..)),
           stat="count",
           alpha = 0.7) +
  geom_text(aes(label = scales::percent(..prop..), y = ..prop.. ),
            stat= "count",
            vjust = -.5) +
  labs(y = "Percentage", fill= "grad_year_div") +
  facet_grid(~Participating.in.PQRS) +
  theme_minimal()+
  theme(legend.position = "none", plot.title = element_text(hjust = 0.5)) +
  theme(axis.text.x = element_text(angle = 90, hjust = .5)) +
  ggtitle("Participating in PQRS")
```



So we can understand that the factorized graduation year is giving more information and clarity of the participation in PQRS than continuous. In the 1980, 1990 data we could see more participation in PQRS. Since it's a factorized variable now it will be used as a categorical variable for models like random forest, SVM.

Categorizing the Group practice Member count

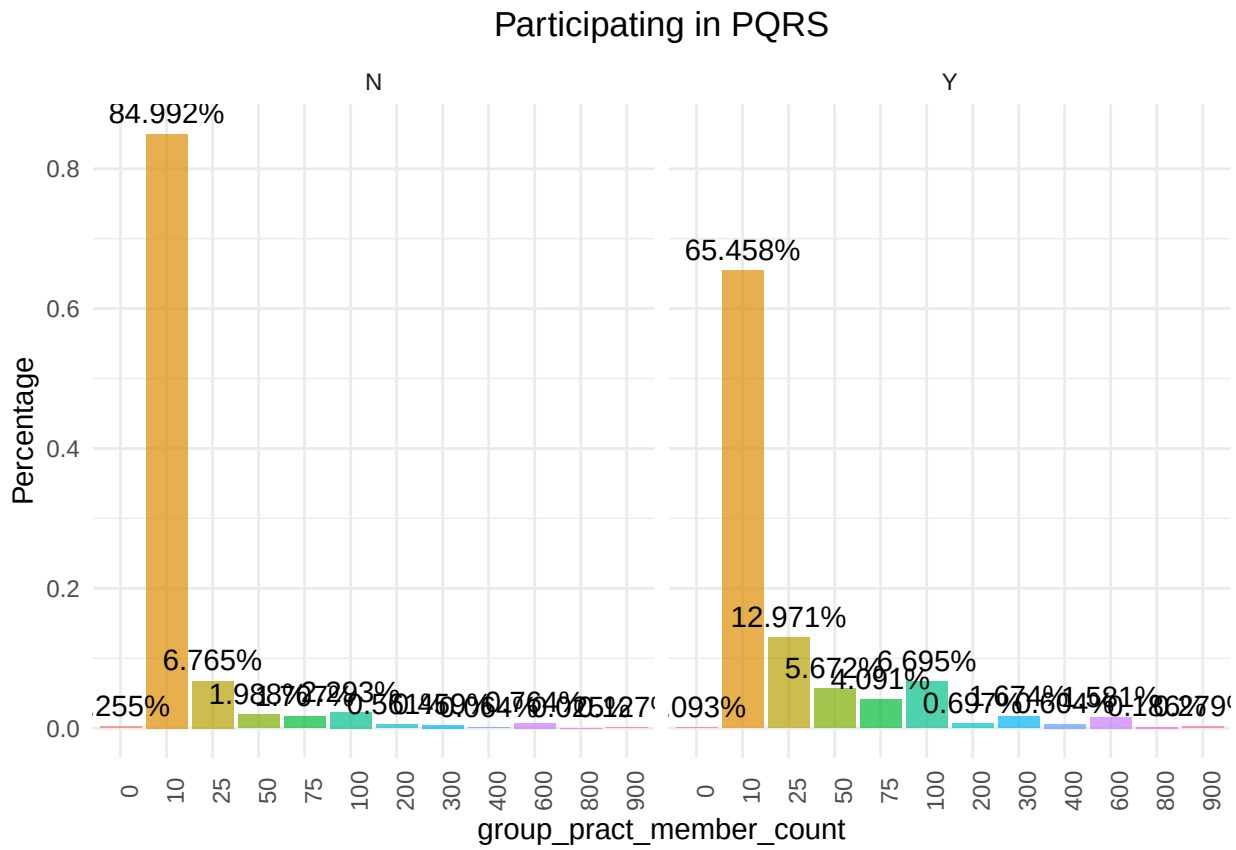
Here we are categorizing the Group practice Member count in different groups of member count to get more insights to the PQRS participation. The following code will group the member counts in different buckets.

```
phys_T$group_pract_member_count <- as.factor(cut(phys_T$Number.of.Group.Practice.members,breaks = c(-1,
                                   labels = c("0","10","25","50","75","100","200","300","400","500",
```

Visualizing the group practice member count buckets against the PQRS participation.

```
phys_T %>%
  ggplot(aes(x = group_pract_member_count, group = Participating.in.PQRS)) +
  geom_bar(aes(y = ..prop.., fill = factor(..x..)),
           stat="count",
           alpha = 0.7) +
  geom_text(aes(label = scales::percent(..prop..), y = ..prop.. ),
            stat= "count",
            vjust = -.5) +
  labs(y = "Percentage", fill= "group_pract_member_count") +
  facet_grid(~Participating.in.PQRS) +
  theme_minimal()+
```

```
  theme(legend.position = "none", plot.title = element_text(hjust = 0.5)) +
  theme(axis.text.x = element_text(angle = 90, hjust = .5)) +
  ggtitle("Participating in PQRS")
```



We could see that in the 25,50, 75, 100, 200, 600 member groups divisions shows more participation. So we can definitely use this data for predicting the PQRS participation.

Categorizing States to different regions for modeling

First we are changing state codes to different levels. Random Forest will take only 53 level for a variable state have 54 levels so we are dividing the levels to region wise. This below code does this. The look-up table *region.list* for the conversion to regions are created in the introduction section.

```
levels(phys_T$State)
```

```
##  [1] "AK" "AL" "AR" "AZ" "CA" "CO" "CT" "DC" "DE" "FL" "GA" "GU" "HI" "IA" "ID"
## [16] "IL" "IN" "KS" "KY" "LA" "MA" "MD" "ME" "MI" "MN" "MO" "MS" "MT" "NC" "ND"
## [31] "NE" "NH" "NJ" "NM" "NV" "NY" "OH" "OK" "OR" "PA" "PR" "RI" "SC" "SD" "TN"
## [46] "TX" "UT" "VA" "VI" "VT" "WA" "WI" "WV" "WY"
```

```
nlevels(phys_T$State)
```

```
## [1] 54
```

```
tst_regions <- sapply(phys_T$State,
                      function(x) names(region.list)[grep(x,region.list)])
tst_regions <- as.factor(unlist(tst_regions))
#adding the regions to the dataframe
phys_T$regions <- tst_regions
#summary of the regions
summary(phys_T$regions)
```

```
##    Island   Midwest Northeast     South      West
##       132      2149      2296      3348      2075
```

We can check the number of levels for credentials by using the following code. There are only 21 levels and it can be used for model since it's coming under the 53 levels limit for random forest and other models.

```
levels(phys_T$Credential)
```

```
##  [1] ""    "AA"  "AU"  "CNA" "CNM" "CNS" "CP"  "CSW" "DC"  "DDM" "DDS" "DO"
## [13] "DPM" "MD"  "MNT" "NP"  "OD"  "OT"  "PA"  "PSY" "PT"
```

```
nlevels(phys_T$Credential)
```

```
## [1] 21
```

Removing unnecessary variables & reducing the column names

```
colnames(phys_T)
```

```
##  [1] "NPI"
##  [2] "PAC.ID"
##  [3] "Professional.Enrollment.ID"
##  [4] "Last.Name"
##  [5] "First.Name"
##  [6] "Middle.Name"
##  [7] "Suffix"
##  [8] "Gender"
##  [9] "Credential"
## [10] "Medical.school.name"
## [11] "Graduation.year"
## [12] "Primary.specialty"
## [13] "Secondary.specialty.1"
## [14] "Secondary.specialty.2"
## [15] "Secondary.specialty.3"
## [16] "Secondary.specialty.4"
## [17] "All.secondary.specialties"
## [18] "Organization.legal.name"
## [19] "Group.Practice.PAC.ID"
## [20] "Number.of.Group.Practice.members"
## [21] "Line.1.Street.Address"
## [22] "Line.2.Street.Address"
## [23] "Marker.of.address.line.2.suppression"
```

```
## [24] "City"
## [25] "State"
## [26] "Zip.Code"
## [27] "Claims.based.hospital.affiliation.CCN.1"
## [28] "Claims.based.hospital.affiliation.LBN.1"
## [29] "Claims.based.hospital.affiliation.CCN.2"
## [30] "Claims.based.hospital.affiliation.LBN.2"
## [31] "Claims.based.hospital.affiliation.CCN.3"
## [32] "Claims.based.hospital.affiliation.LBN.3"
## [33] "Claims.based.hospital.affiliation.CCN.4"
## [34] "Claims.based.hospital.affiliation.LBN.4"
## [35] "Claims.based.hospital.affiliation.CCN.5"
## [36] "Claims.based.hospital.affiliation.LBN.5"
## [37] "Professional.accepts.Medicare.Assignment"
## [38] "Participating.in.eRx"
## [39] "Participating.in.PQRS"
## [40] "Participating.in.EHR"
## [41] "grad_year_div"
## [42] "group_pract_member_count"
## [43] "regions"
```

```
phys_F <- phys_T[c(8,9,37,38,39,40,41,42,43)]
#Data summary after triming the columns
summary(phys_F)
```

```
##  Gender     Credential   Professional.accepts.Medicare.Assignment
##  F:3693          :5764   M:1743
##  M:6307   MD     :2555   Y:8257
##           DC     : 311
##           OD     : 218
##           PT     : 213
##           NP     : 175
##           (Other): 764
##  Participating.in.eRx Participating.in.PQRS Participating.in.EHR grad_year_div
##  N:8091               N:7849                N:7846               1990   :2718
##  Y:1909               Y:2151                Y:2154               1980   :2557
##                                                                  2000   :2326
##                                                                  1970   :1563
##                                                                  1960   : 432
##                                                                  2010   : 338
##                                                                  (Other):  66
##  group_pract_member_count      regions
##  10     :8079             Island   : 132
##  25     : 810             Midwest  :2149
##  100    : 324             Northeast:2296
##  50     : 278             South    :3348
##  75     : 222             West     :2075
##  600    :  94
##  (Other): 193
```

```
dim(phys_F)
```

```
## [1] 10000     9
```

```
colnames(phys_F)
```

```
## [1] "Gender"
## [2] "Credential"
## [3] "Professional.accepts.Medicare.Assignment"
## [4] "Participating.in.eRx"
## [5] "Participating.in.PQRS"
## [6] "Participating.in.EHR"
## [7] "grad_year_div"
## [8] "group_pract_member_count"
## [9] "regions"
```

The below code will make the column variable names to short form for easy usage.

```
#converting the column names to short forms.
names(phys_F)[3] <- "PAMA"
names(phys_F)[4] <- "eRx"
names(phys_F)[5] <- "PQRS"
names(phys_F)[6] <- "EHR"
names(phys_F)[7] <- "grad_year"
names(phys_F)[8] <- "GMPC" #group member practice count divisions
#verifying the column names after conversion
colnames(phys_F)
```

```
## [1] "Gender"     "Credential" "PAMA"       "eRx"        "PQRS"
## [6] "EHR"        "grad_year"  "GMPC"       "regions"
```

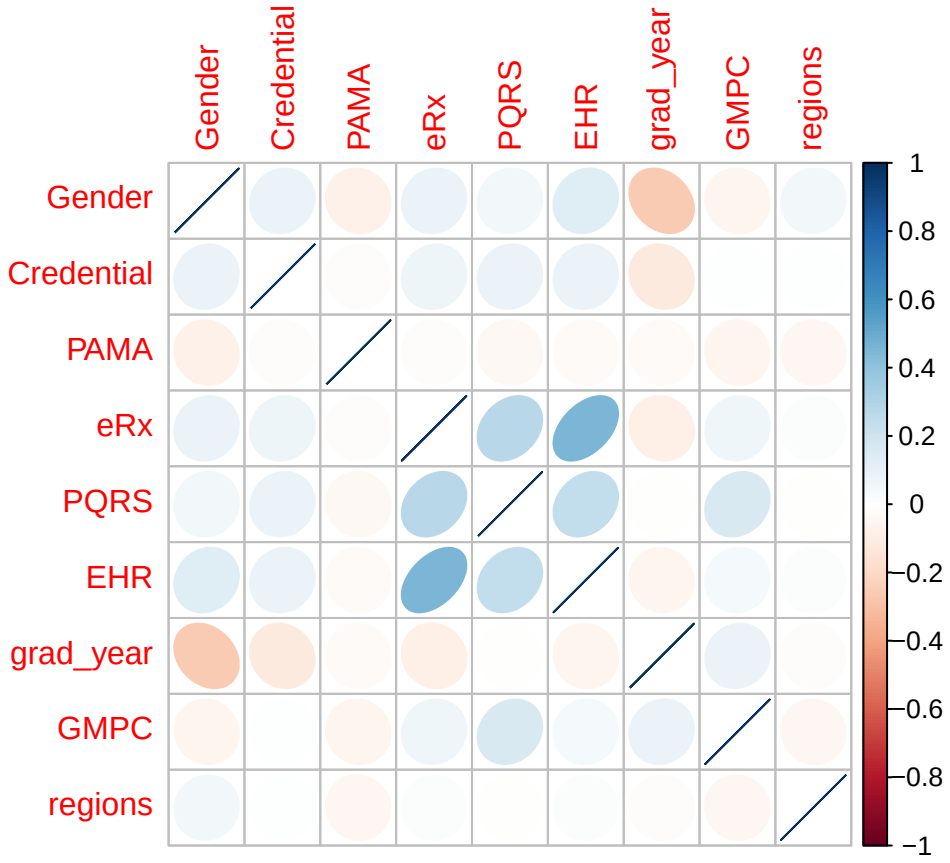
```
#Checking the data for issues after data wrangling
glimpse(phys_F)
```

```
## Observations: 10,000
## Variables: 9
## $ Gender     <fct> F, F, M, M, M, M, F, M, F, M, F, M, F, F, M, M, M, F, M,...
## $ Credential <fct> CSW, OD, , MD, , MD, , DPM, MD, , , MD, DC, , , CP, MD, ...
## $ PAMA       <fct> Y, Y, M, M, Y, Y, Y, Y, Y, Y, Y, Y, Y, Y, Y, Y, Y, Y, Y,...
## $ eRx        <fct> N, N, N, N, N, Y, N, N, N, N, Y, Y, N, N, N, N, N, N, N,...
## $ PQRS       <fct> N, N, N, Y, N, N, N, N, N, N, N, N, N, N, N, N, Y, N, N,...
## $ EHR        <fct> N, N, N, N, N, Y, N, Y, Y, N, N, N, N, N, N, N, Y, N, N,...
## $ grad_year  <fct> 1990, 1990, 1980, 1980, 1990, 2000, 2000, 2000, 1990, 19...
## $ GMPC       <fct> 10, 10, 50, 25, 10, 10, 10, 10, 10, 10, 10, 10, 10, 10, ...
## $ regions    <fct> Northeast, South, Midwest, South, West, South, South, We...
```

Correlation of different variables

The below plot shows the correlation between different variables in the physician dataset

```
# LEts make a correlation matrix for the data
corrplot(cor(sapply(phys_F,as.integer)),method = "ellipse")
```



Most of the variables which we are using are having positive correlation with PQRS participation. We are using PAMA which is negatively correlated. We are hoping to negate any over-fitting effect by selecting PAMA which is having negative correlation.

Modeling Approaches.

We are planing to use the following models. 1. Random Forest 2. Support Vector Machine
3. Extreme Gradient Boost

Random Forest

Random forests or random decision forests are an ensemble learning method for classification, regression and other tasks that operate by constructing a multitude of decision trees at training time and outputting the class that is the mode of the classes (classification) or mean prediction (regression) of the individual trees.Random decision forests correct for decision trees' habit of over fitting to their training set. Here we are going to use this method to model our train dataset and for predicting using the model.

Splitting into train & test

For modeling we need to have a train dataset and test dataset. So we are split them with the following code.

```
set.seed(30, sample.kind="Rounding")
indexes = sample(1:nrow(phys_F), size=0.8*nrow(phys_F))
RF_train <- phys_F[indexes,]
RF_test <- phys_F[-indexes,]
```

Random Forest Model Execution

Training (Random Forest)

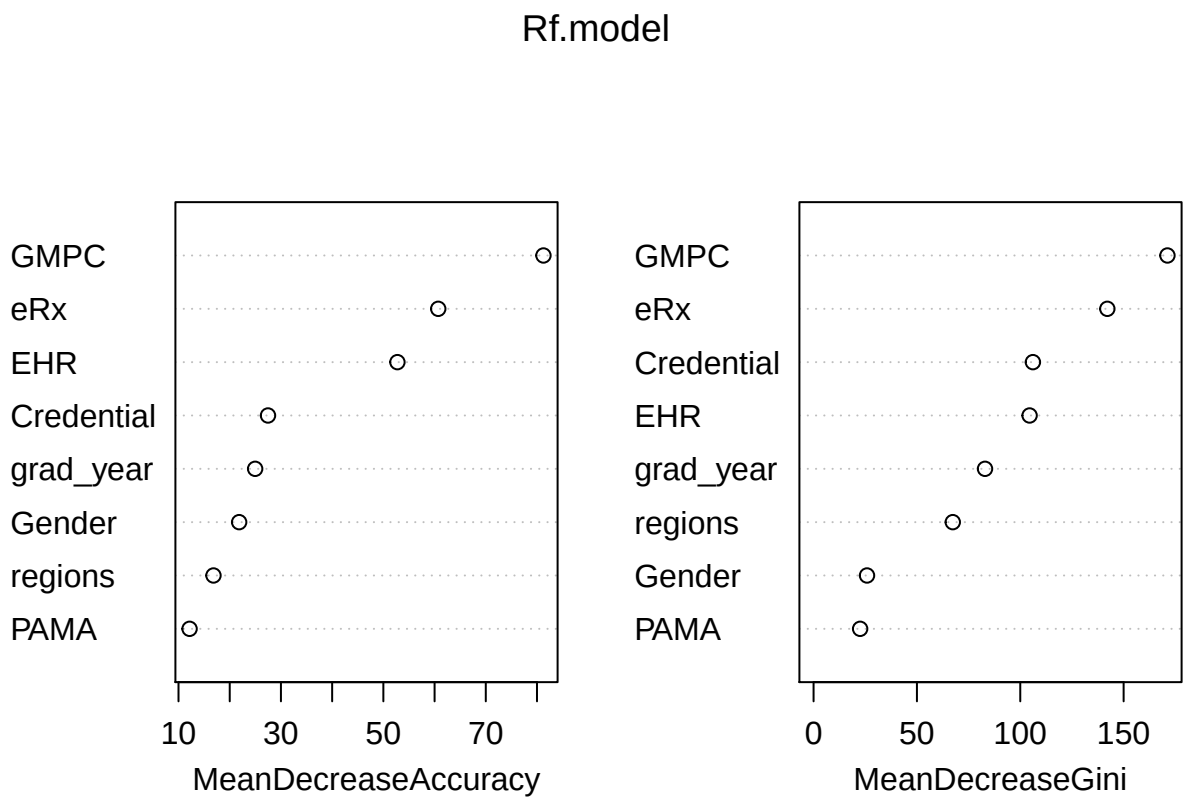
The following code will train the Random Forest Model using ntree = 1000

```
Rf.model <- randomForest(PQRS~.,RF_train, importance=TRUE,ntree=1000)
```

Varplot of different parameters (Random Forest)

The following code will show the plot of different variables used for modeling and how important they are for the modeling. We can see that the GMPC, eRx, EHR are the most important variables in the RF model.

```
varImpPlot(Rf.model)
```



Predict (Random Forest)

The following code will predict the random Forest Model using the test dataset. The code also print the confusion matrix to understand the accuracy and other parameters.

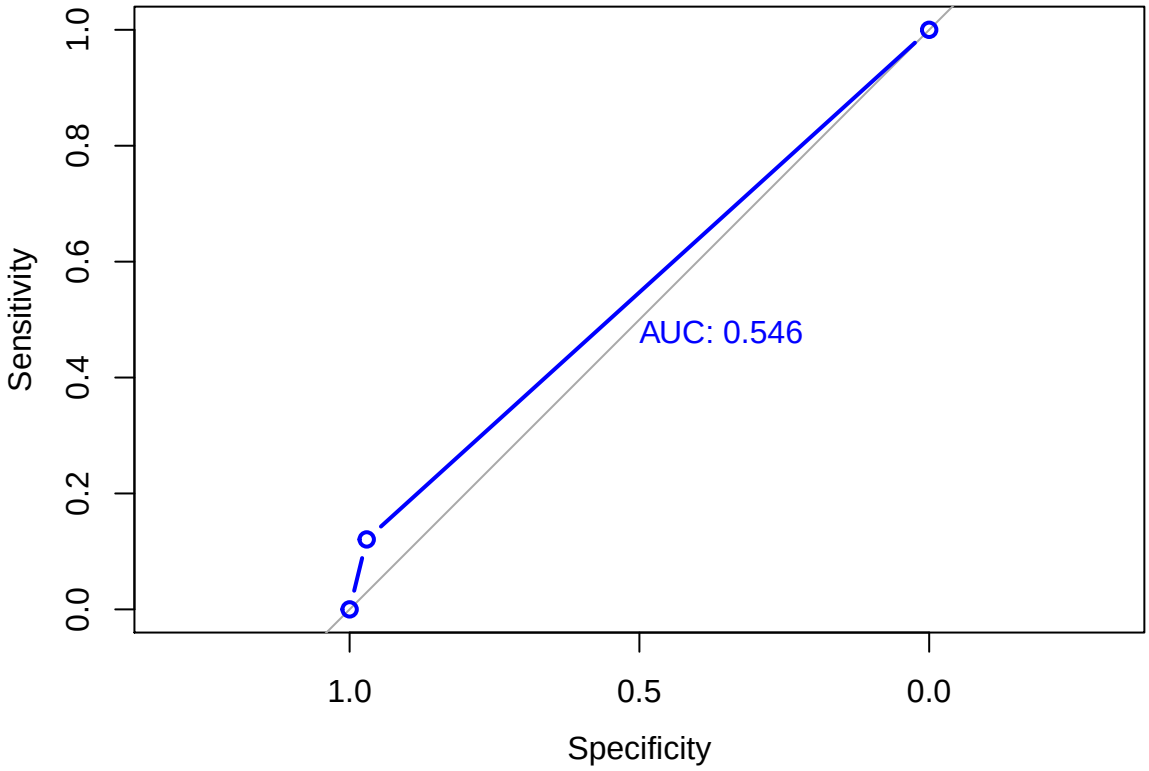
```
Rf.prd <- predict(Rf.model, newdata = RF_test)
confusionMatrix(RF_test$PQRS, Rf.prd)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    N    Y
##          N 1547   47
##          Y  357   49
##
##                Accuracy : 0.798
##                  95% CI : (0.7797, 0.8154)
##     No Information Rate : 0.952
##     P-Value [Acc > NIR] : 1
##
##                   Kappa : 0.1275
##
##  Mcnemar's Test P-Value : <2e-16
##
##             Sensitivity : 0.8125
##             Specificity : 0.5104
##          Pos Pred Value : 0.9705
##          Neg Pred Value : 0.1207
##              Prevalence : 0.9520
##          Detection Rate : 0.7735
##    Detection Prevalence : 0.7970
##       Balanced Accuracy : 0.6615
##
##        'Positive' Class : N
##
```

ROC plot (Random Forest)

The ROC curve shows the trade-off between sensitivity (or TPR) and specificity (1 – FPR). Classifiers that give curves closer to the top-left corner indicate a better performance. As a baseline, a random classifier is expected to give points lying along the diagonal (FPR = TPR). The closer the curve comes to the 45-degree diagonal of the ROC space, the less accurate the test. To compare different classifiers, it can be useful to summarize the performance of each classifier into a single measure. One common approach is to calculate the area under the ROC curve, which is abbreviated to AUC. It is equivalent to the probability that a randomly chosen positive instance is ranked higher than a randomly chosen negative instance. We are expecting to get a high AUC.

```
plot.roc(as.numeric(RF_test$PQRS), as.numeric(Rf.prd),lwd=2, type="b",print.auc=TRUE,col ="blue")
```



Support Vector Machine

support-vector machines (SVMs, also support-vector networks) are supervised learning models with associated learning algorithms that analyze data used for classification and regression analysis. Given a set of training examples, each marked as belonging to one or the other of two categories, an SVM training algorithm builds a model that assigns new examples to one category or the other, making it a non-probabilistic binary linear classifier. An SVM model is a representation of the examples as points in space, mapped so that the examples of the separate categories are divided by a clear gap that is as wide as possible. New examples are then mapped into that same space and predicted to belong to a category based on the side of the gap on which they fall. Here we are going to use this method to model our train dataset and for predicting using the model.

Splitting into train & test

For modeling we need to have a train dataset and test dataset. So we are split them with the following code.

```
set.seed(30, sample.kind="Rounding")
indexes = sample(1:nrow(phys_F), size=0.8*nrow(phys_F))
SVM_train <- phys_F[indexes,]
SVM_test <- phys_F[-indexes,]
```

Support Vector Machine Model Execution

Tuning (SVM)

The following code will tune the SVM model parameters.

```
tune_prm <- tune(svm,factor(PQRS)~.,data = SVM_train)
```

Training (SVM)

The following code will train the SVM model using the best tuned parameters from the training dataset

```
SVM_model <- svm(SVM_train$PQRS~., data=SVM_train
                 ,type="C-classification", gamma=tune_prm$best.model$gamma
                 ,cost=tune_prm$best.model$cost
                 ,kernel="radial")
```

Predict (SVM)

The following code will predict the SVM Model using the test dataset. The code also print the confusion matrix to understand the accuracy and other parameters.

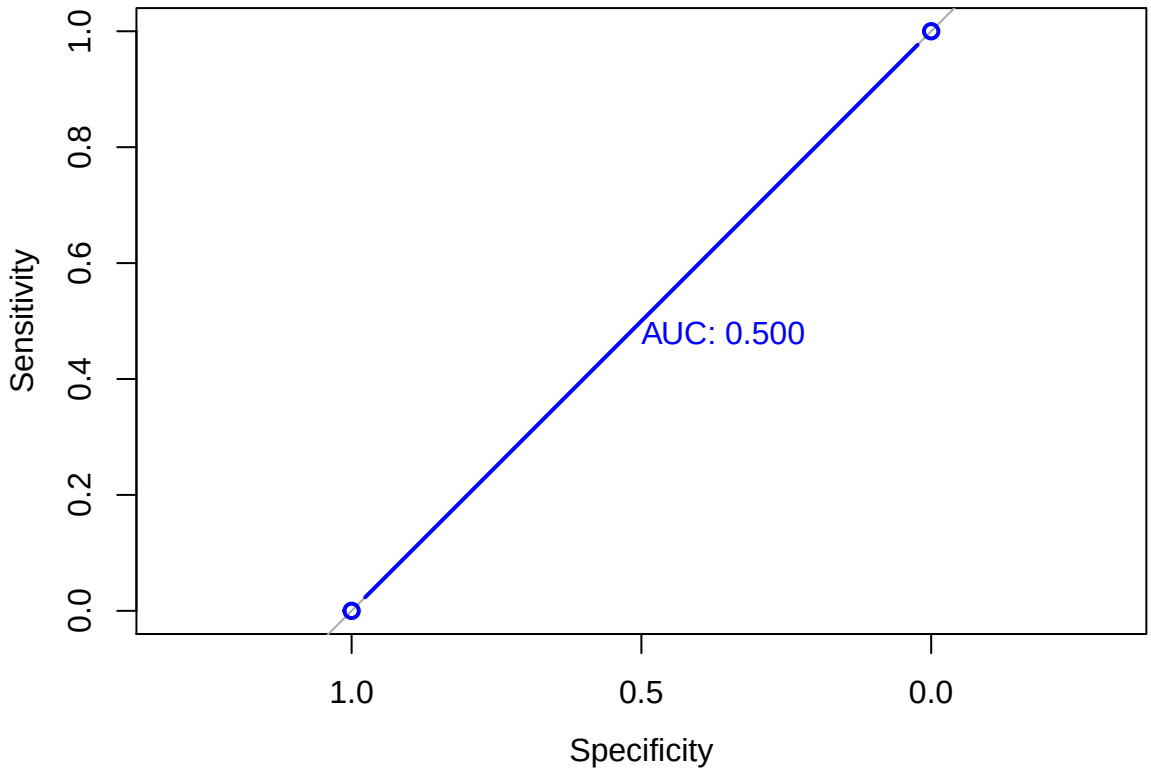
```
SVM_prd <- predict(SVM_model,newdata=SVM_test)
confusionMatrix(SVM_prd,SVM_test$PQRS)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    N    Y
##          N 1594  406
##          Y    0    0
##
##                Accuracy : 0.797
##                  95% CI : (0.7787, 0.8144)
##     No Information Rate : 0.797
##     P-Value [Acc > NIR] : 0.5133
##
##                   Kappa : 0
##
##  Mcnemar's Test P-Value : <2e-16
##
##             Sensitivity : 1.000
##             Specificity : 0.000
##          Pos Pred Value : 0.797
##          Neg Pred Value :   NaN
##              Prevalence : 0.797
##          Detection Rate : 0.797
##    Detection Prevalence : 1.000
##       Balanced Accuracy : 0.500
##
##        'Positive' Class : N
##
```

ROC plot (SVM)

We have explained the ROC plot in Random forest. Here we are plotting the ROC for SVM to understand the area under the ROC curve, which is abbreviated to AUC. The AUC is less compared to Random Forest.

```
SVM_plot <- plot.roc (as.numeric(SVM_test$PQRS), as.numeric(SVM_prd),lwd=2, type="b", print.auc=TRUE,col
```



Clearing the objects from memory for XGBoost execution

```
rm(phys,phys_T,RF_train,SVM_train)
```

XGBoost

XGBoost is an implementation of gradient boosted decision trees designed for speed and performance. The implementation of the model supports the features of the scikit-learn and R implementations, with new additions like regularization. Three main forms of gradient boosting are supported:

1. Gradient Boosting algorithm also called gradient boosting machine including the learning rate.
2. Stochastic Gradient Boosting with sub-sampling at the row, column and column per split levels.
3. Regularized Gradient Boosting with both L1 and L2 regularization.

The library provides a system for use in a range of computing environments, not least:

Parallelization of tree construction using all of your CPU cores during training. Distributed Computing for training very large models using a cluster of machines. Out-of-Core Computing for very large datasets that don't fit into memory. Cache Optimization of data structures and algorithm to make best use of hardware.

Here we are going to use this method to model our train dataset and for predicting using the model.

Splitting into train & test (SVM)

For modeling we need to have a train dataset and test dataset. So we are split them with the following code.

```
set.seed(30, sample.kind="Rounding")
xgbData <- phys_F
indexes <- sample(1:nrow(xgbData), size=0.8*nrow(xgbData))
XGBtrain <- xgbData[indexes,]
XGBtest <- xgbData[-indexes,]
```

XGBoost Model Execution

Tuning (XGB)

The following code for tuning the XGBoost is taking about 1 hr(in my machine having 8 logical cpus and 8 gm ram) for execution. We are using the EVAL parameter = 'FALSE' in the code chunk to avoid execution of these tuning process. We will use the best tune parameters to train and predict in the coming sections, which we got as result of this tuning process.

```
#Tuning XBGTree using Caret Package
#Hyperparameters to tune:
#nrounds: Number of trees, default: 100
#max_depth: Maximum tree depth, default: 6
#eta: Learning rate, default: 0.3
#  gamma: Used for tuning of Regularization, default: 0
#  colsample_bytree: Column sampling, default: 1
#  min_child_weight: Minimum leaf weight, default: 1
#  subsample: Row sampling, default: 1
#  We'll break down the tuning of these into five sections:
#
#   Fixing learning rate eta and number of iterations nrounds
#  Maximum depth max_depth and child weight min_child_weight
#  Setting column colsample_bytree and row sampling subsample
#  Experimenting with different gamma values
#  Reducing the learning rate eta

# set seed
set.seed(30, sample.kind="Rounding")
xgbData <- phys_F
indexes <- sample(1:nrow(xgbData), size=0.8*nrow(xgbData))
XGBtrain <- xgbData[indexes,]
XGBtest <- xgbData[-indexes,]
# note to start nrounds from 200, as smaller learning rates result in errors so
# big with lower starting points that they'll mess the scales
formula = PQRS~.
nrounds <- 1000

tune_grid <- expand.grid(
  nrounds = seq(from = 200, to = nrounds, by = 50),
  eta = c(0.025, 0.05, 0.1, 0.3),
  max_depth = c(2, 3, 4, 5, 6,8,10,15,20,25,30,35,40,50),
  gamma = 0,
  colsample_bytree = 1,
```

```
  min_child_weight = 1,
  subsample = 1
)

tune_control <- caret::trainControl(
  method = "cv", # cross-validation
  number = 3, # with n folds
  classProbs = TRUE
)

xgb_tune <- caret::train(
  formula,
  data = XGBtrain,
  trControl = tune_control,
  tuneGrid = tune_grid,
  method = "xgbTree"
)
predictions<-predict(xgb_tune,XGBtest)
confusionMatrix(predictions,XGBtest$PQRS) #0.799

#output of best Tune is nrounds = 200 eta = 0.05, max_dept = 2
xgb_tune$bestTune

ggplot(xgb_tune)

xgb_tune$bestTune$nrounds
xgb_tune$bestTune$max_depth
xgb_tune$bestTune$eta
xgb_tune$bestTune$min_child_weight
xgb_tune$bestTune$subsample
tune_grid2 <- expand.grid(
  nrounds = seq(from = 50, to = nrounds, by = 50), #700
  eta = xgb_tune$bestTune$eta,
  max_depth = c(1,2,3,5,10,15,20,25,30,35), #15
  gamma = 0,
  colsample_bytree = 1,
  min_child_weight = c(1, 2, 3), #3
  subsample = 1
)

xgb_tune2 <- caret::train(
  formula,
  data = XGBtrain,
  trControl = tune_control,
  tuneGrid = tune_grid2,
  method = "xgbTree",
  verbose = TRUE
)

ggplot(xgb_tune2)
predictions2<-predict(xgb_tune2,XGBtest)
confusionMatrix(predictions2,XGBtest$PQRS) #0.792
xgb_tune$bestTune
```

```
xgb_tune2$bestTune
tune_grid3 <- expand.grid(
  nrounds = seq(from = 50, to = nrounds, by = 50), #150
  eta = xgb_tune$bestTune$eta,
  max_depth = xgb_tune2$bestTune$max_depth,
  gamma = 0,
  colsample_bytree = c(0.4, 0.6, 0.8, 1.0), #0.8
  min_child_weight = xgb_tune2$bestTune$min_child_weight,
  subsample = c(0.5, 0.75, 1.0) #0.75
)

xgb_tune3 <- caret::train(
  formula,
  data = XGBtrain,
  trControl = tune_control,
  tuneGrid = tune_grid3,
  method = "xgbTree",
  verbose = TRUE
)

ggplot(xgb_tune3)
predictions3<-predict(xgb_tune3,XGBtest)
confusionMatrix(predictions3,XGBtest$PQRS) #.7925

xgb_tune$bestTune
xgb_tune2$bestTune
xgb_tune3$bestTune
tune_grid4 <- expand.grid(
  nrounds = seq(from = 50, to = nrounds, by = 50), #500
  eta = xgb_tune$bestTune$eta,
  max_depth = xgb_tune2$bestTune$max_depth,
  gamma = c(0, 0.05, 0.1, 0.5, 0.7, 0.9, 1.0), #0.1
  colsample_bytree = xgb_tune3$bestTune$colsample_bytree,
  min_child_weight = xgb_tune2$bestTune$min_child_weight,
  subsample = xgb_tune3$bestTune$subsample
)

xgb_tune4 <- caret::train(
  formula,
  data = XGBtrain,
  trControl = tune_control,
  tuneGrid = tune_grid4,
  method = "xgbTree",
  verbose = TRUE
)
ggplot(xgb_tune4)
predictions4<-predict(xgb_tune4,XGBtest)
confusionMatrix(predictions4,XGBtest$PQRS) #0.8015
xgb_tune$bestTune
xgb_tune2$bestTune
xgb_tune3$bestTune
xgb_tune4$bestTune
```

```
tune_grid5 <- expand.grid(
  nrounds = seq(from = 100, to = 500, by = 25), #300
  eta = c(0.01, 0.015, 0.025, 0.05, 0.1), #0.025
  max_depth = xgb_tune2$bestTune$max_depth,
  gamma = xgb_tune4$bestTune$gamma,
  colsample_bytree = xgb_tune3$bestTune$colsample_bytree,
  min_child_weight = xgb_tune2$bestTune$min_child_weight,
  subsample = xgb_tune3$bestTune$subsample
)
xgb_tune5 <- caret::train(
  formula,
  data = XGBtrain,
  trControl = tune_control,
  tuneGrid = tune_grid5,
  method = "xgbTree",
  verbose = TRUE
)

xgb_tune5$bestTune
xgb_tune4$bestTune
xgb_tune3$bestTune
xgb_tune2$bestTune
xgb_tune$bestTune

ggplot(xgb_tune5)
predictions5<-predict(xgb_tune5,XGBtest)
confusionMatrix(predictions5,XGBtest$PQRS) #0.802
```

Training (SVM)

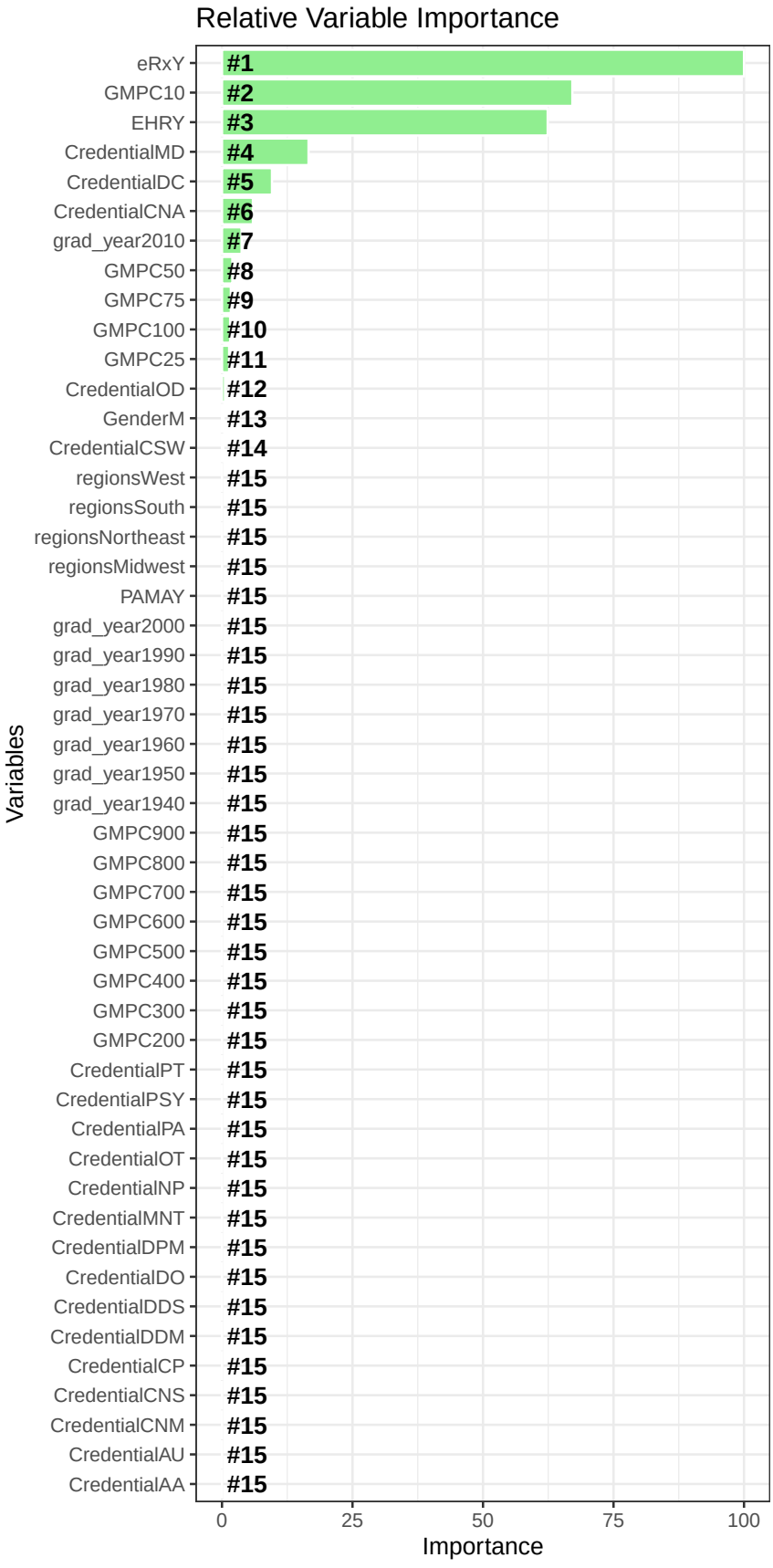
The following code will train the SVM model using the best tuned parameters from the training dataset

```
formula = PQRS~.
fitControl <- trainControl(method="cv", number = 3,classProbs = TRUE )
#We are using the best tunned hyperparameters (mentioned in above section) in below model.
xgbGrid <- expand.grid(nrounds = 250,
                       max_depth = 1,
                       eta = .025,
                       gamma = 0.05,
                       colsample_bytree = .4,
                       min_child_weight = 1,
                       subsample = 0.75
)
XGB.model <- train(formula, data = XGBtrain,
                   method = "xgbTree"
                   ,trControl = fitControl
                   , verbose=0
                   , maximize=FALSE
                   ,tuneGrid = xgbGrid
)
importance <- varImp(XGB.model)
varImportance <- data.frame(Variables = row.names(importance[[1]]),
                            Importance = round(importance[[1]]$Overall,2))
```

Varplot of different parameters (XGBoost)

The following code will show the plot of different variables used for modeling and how important they are for the modeling. We can see eRX, group count with member strength 10 plus, EHR and credentials are the main important parameters in the XGBoost model we created.

```
rankImportance <- varImportance %>%
  mutate(Rank = paste0('#',dense_rank(desc(Importance))))
ggplot(rankImportance, aes(x = reorder(Variables, Importance),
                           y = Importance)) +
  geom_bar(stat='identity',colour="white", fill = "lightgreen") +
  geom_text(aes(x = Variables, y = 1, label = Rank),
            hjust=0, vjust=.5, size = 4, colour = 'black',
            fontface = 'bold') +
  labs(x = 'Variables', title = 'Relative Variable Importance') +
  coord_flip() +
  theme_bw()
```


Relative Variable Importance
eRxY #1
GMPC10 #2
EHRY #3
CredentialMD #4
CredentialDC #5
CredentialCNA #6
grad_year2010 #7
GMPC50 #8
GMPC75 #9
GMPC100 #10
GMPC25 #11
CredentialOD #12
GenderM #13
CredentialCSW #14
regionsWest #15
regionsSouth #15
regionsNortheast #15
regionsMidwest #15
PAMAY #15
grad_year2000 #15
grad_year1990 #15
grad_year1980 #15
grad_year1970 #15
grad_year1960 #15
grad_year1950 #15
grad_year1940 #15
GMPC900 #15
GMPC800 #15
GMPC700 #15
GMPC600 #15
GMPC500 #15
GMPC400 #15
GMPC300 #15
GMPC200 #15
CredentialPT #15
CredentialPSY #15
CredentialPA #15
CredentialOT #15
CredentialNP #15
CredentialMNT #15
CredentialDPM #15
CredentialDO #15
CredentialDDS #15
CredentialDDM #15
CredentialCP #15
CredentialCNS #15
CredentialCNM #15
CredentialAU #15
CredentialAA #15
Variables
0
25
50
75
100
Importance

Predict (XGBoost)

The following code will predict the random Forest Model using the test dataset. The code also print the confusion matrix to understand the accuracy and other parameters.

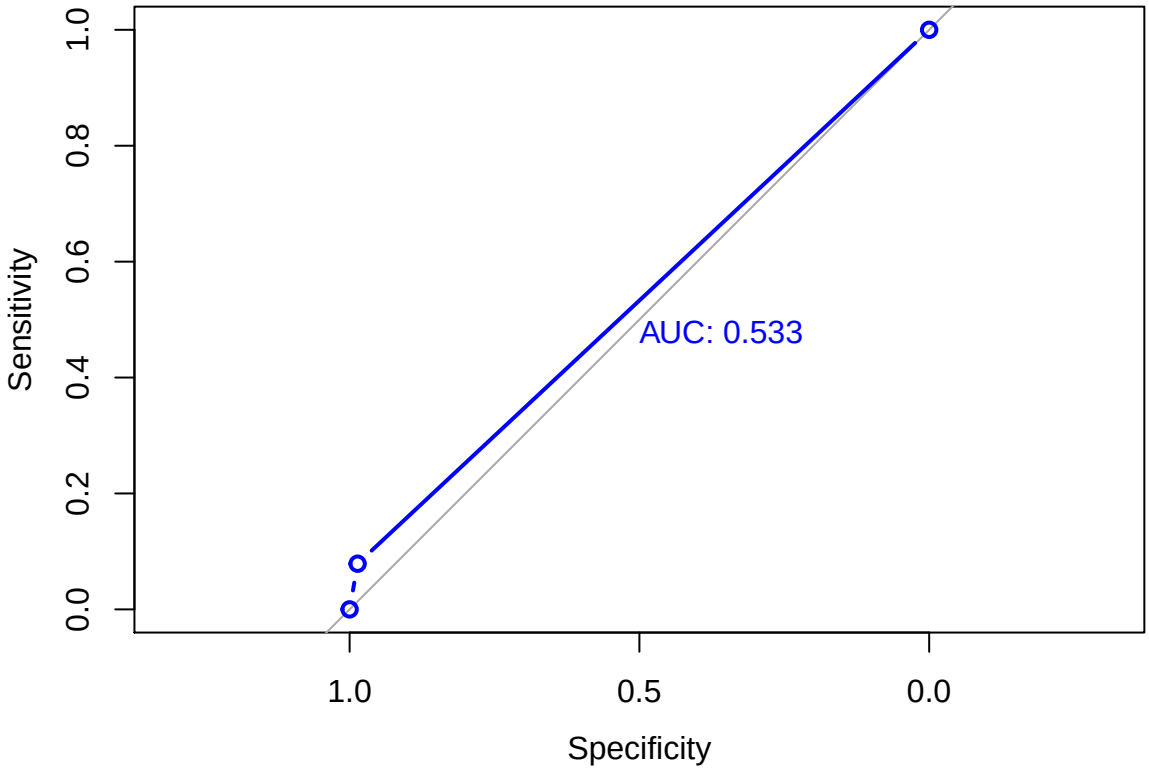
```
XGB.prd <- predict(XGB.model,XGBtest)
confusionMatrix(XGB.prd, XGBtest$PQRS)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    N    Y
##          N 1572  374
##          Y   22   32
##
##                Accuracy : 0.802
##                  95% CI : (0.7838, 0.8193)
##     No Information Rate : 0.797
##     P-Value [Acc > NIR] : 0.3001
##
##                   Kappa : 0.096
##
##  Mcnemar's Test P-Value : <2e-16
##
##             Sensitivity : 0.98620
##             Specificity : 0.07882
##          Pos Pred Value : 0.80781
##          Neg Pred Value : 0.59259
##              Prevalence : 0.79700
##          Detection Rate : 0.78600
##    Detection Prevalence : 0.97300
##       Balanced Accuracy : 0.53251
##
##        'Positive' Class : N
##
```

ROC plot (XGBoost)

We have explained the ROC plot in Random forest. Here we are plotting the ROC for XGBoost to understand the area under the ROC curve, which is abbreviated to AUC. The AUC is less compared to Random Forest.

```
XGB.plot <- plot.roc (as.numeric(XGBtest$PQRS), as.numeric(XGB.prd),lwd=2, type="b", print.auc=TRUE,col
```



Results

In this section, we will showcase the results and compare the models and try to select a best model for the data in our hand.

Comparison of confusion Matrix

In this section we will compare the confusion matrix from different models.

Random Forest Confusion Matrix

Following code will print the confusion matrix for Random Forest model.

```
confusionMatrix(RF_test$PQRS, Rf.prd)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    N    Y
##          N 1547   47
##          Y  357   49
##
```

```
##                Accuracy : 0.798
##                  95% CI : (0.7797, 0.8154)
##     No Information Rate : 0.952
##     P-Value [Acc > NIR] : 1
##
##                   Kappa : 0.1275
##
##  Mcnemar's Test P-Value : <2e-16
##
##             Sensitivity : 0.8125
##             Specificity : 0.5104
##          Pos Pred Value : 0.9705
##          Neg Pred Value : 0.1207
##              Prevalence : 0.9520
##          Detection Rate : 0.7735
##    Detection Prevalence : 0.7970
##       Balanced Accuracy : 0.6615
##
##        'Positive' Class : N
##
```

SVM Confusion Matrix

Following code will print the confusion matrix for SVM model.

```
confusionMatrix(SVM_prd,SVM_test$PQRS)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    N    Y
##          N 1594  406
##          Y    0    0
##
##                Accuracy : 0.797
##                  95% CI : (0.7787, 0.8144)
##     No Information Rate : 0.797
##     P-Value [Acc > NIR] : 0.5133
##
##                   Kappa : 0
##
##  Mcnemar's Test P-Value : <2e-16
##
##             Sensitivity : 1.000
##             Specificity : 0.000
##          Pos Pred Value : 0.797
##          Neg Pred Value :   NaN
##              Prevalence : 0.797
##          Detection Rate : 0.797
##    Detection Prevalence : 1.000
##       Balanced Accuracy : 0.500
##
##        'Positive' Class : N
```

##

XGBoost Confusion Matrix

Following code will print the confusion matrix for SVM model.

```
confusionMatrix(XGB.prd, XGBtest$PQRS)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    N    Y
##          N 1572  374
##          Y   22   32
##
##                Accuracy : 0.802
##                  95% CI : (0.7838, 0.8193)
##     No Information Rate : 0.797
##     P-Value [Acc > NIR] : 0.3001
##
##                   Kappa : 0.096
##
##  Mcnemar's Test P-Value : <2e-16
##
##             Sensitivity : 0.98620
##             Specificity : 0.07882
##          Pos Pred Value : 0.80781
##          Neg Pred Value : 0.59259
##              Prevalence : 0.79700
##          Detection Rate : 0.78600
##    Detection Prevalence : 0.97300
##       Balanced Accuracy : 0.53251
##
##        'Positive' Class : N
##
```

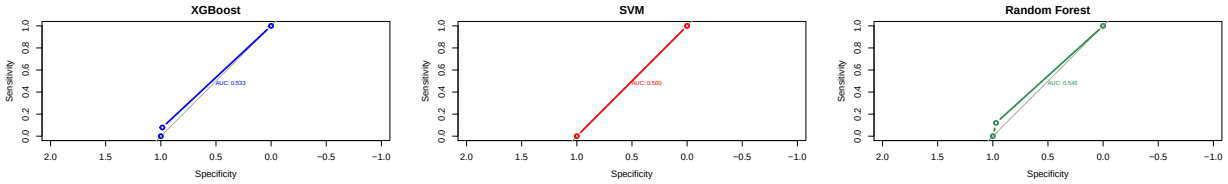
Comparison on Confusion Matrix

From the comparison pf confusion matrix for different models, we could see the XGBoost is giving high accuracy. So if the project need high accuracy we can select the XGBoost model. If we need high sensitivity, then we can select the SVM model which gives high sensitivity. If we need a balanced model from these 3 models with uniform values for sensitivity, specificity, Prevalence, we can select the random forest which gives balanced values for all these parameters. So each model has it on merits and selecting a model is as per the project requirement.

Comparison on ROC plot

From the ROC plots, we could see that AUC for Random Forest come out as the highest. Since it have balanced values for all other parameters from the confusion matrix, for this study i am recommending the random forest for further predicting.

```
par(mfrow=c(2,3))
plot.roc (as.numeric(XGBtest$PQRS), as.numeric(XGB.prd),main="XGBoost",lwd=2, type="b", print.auc=TRUE,
plot.roc (as.numeric(SVM_test$PQRS), as.numeric(SVM_prd),main="SVM",lwd=2, type="b", print.auc=TRUE, co
plot.roc (as.numeric(RF_test$PQRS), as.numeric(Rf.prd), main="Random Forest",lwd=2, type="b", print.auc=
```



Conclusion

In this section, we will gives a brief summary of the report, its potential impact, its limitations, and future work.

Brief Summary

In this project we tried to predict the PQRS(Physician Quality Reporting System) participation among physicians who are present in the physician dataset, which we got from the CMS website. We used 3 different models, 1. Random Forest, 2. SVM model 3. XGBoost Model. As per our study, Random forest model which got good results in terms of balanced parameters from confusion matrix and highest AUC.

Potential Impact

The potential Impact which we can take from this project is that, we found out there is a possibility of predicting PQRS participation using advanced models using the parameters from the physician data set.

Limitations

Following are the limitations. 1. The AUC which we got is very less from all models ~ .5 2. Accuracy which we got is very less. ~ .8 3. We have limitations on the processing power on a laptop.

Future work

Following are the future work which we can do on this project 1. Our dataset is very unbalanced. We can use the SMOTE method to make the dataset balanced and can use the XGBoost method to model an improved balanced model which can deliver more good results. 2. We can neural networks for future improvement in modeling 3. We can use Deep Learning Tensor flow, Keras, MXNet for better modeling and predicting.