Early Diabetes Report

Daniel Alexander

6/6/2021

1 Introduction

Diabetes is a disease that occurs when a person's blood glucose is too high. Blood glucose is the body's main source of energy and comes from the food we eat. The pancreas produces a hormone called insulin that helps glucose get into our cells. When a person is not able to make enough, or any insulin they are then diabetic.

According to the "National Diabetes Statistics Report, 2020," 34.2 million Americans or, just over 10%, have diabetes. My father was one of the 34.2 million and sadly, this year, lost his battle with the disease. It is a ruthless, terrible disease that does not discriminate among race, age, gender, or social class.

A more frightening statistic is that 88 million American adults, nearly 1 in 3, have prediabetes. Fortunately for them, they still have an opportunity to change their future so early detection is very important.

The goal of this HarvardX PH125.9x Data Science Capstone Project is to use data science skills to develop a model that can predict diabetes diagnoses using early health predictors. The dataset for this project is the Early Stage Diabetes Risk Prediction Dataset consisting of 520 patient observations of the following 17 variables:

- Age
- Gender
- Polyuria
- Polydipsia
- Sudden Weight Loss
- Weakness
- Polyphagia
- Genital Thrush
- Visual Blurring
- Itching
- Irritability
- Delayed Healing
- Partial Paresis
- Muscle Stiffness
- Alopecia
- Obesity
- Class

The dataset was collected using direct questionnaires from the paitents of Sylhet Diabetes Hospital in Sylhet, Bangladesh and can be found at https://archive.ics.uci.edu/ml/machine-learning-databases/00529/diabetes_data_upload.csv (https://archive.ics.uci.edu/ml/machine-learning-databases/00529/diabetes_data_upload.csv).

I performed the key steps learned during the course to accomplish the goal. The steps included exploring the data to understand its structure, visualizing the data to uncover distributions and relationships, developing models using Logistic Regression and Random Forest Classification to satisfy the goal, then testing it to find which model was most accurate.

2 Methods and Analysis

My first step was to download the dataset from the ICS website and convert it to a data frame. I then performed exploratory data analysis to learn more about the data. Taking a glimpse of the data, it seems only the variable “Age” is numeric. The other 16 variables are character strings and appear to only consist of yes/no responses.

```
glimpse(data_set)
```

```
## Rows: 520
## Columns: 17
## $ Age                <int> 40, 58, 41, 45, 60, 55, 57, 66, 67, 70, 44, 38, ...
## $ Gender             <chr> "Male", "Male", "Male", "Male", "Male", "Male", ...
## $ Polyuria           <chr> "No", "No", "Yes", "No", "Yes", "Yes", "Yes", "Y...
## $ Polydipsia         <chr> "Yes", "No", "No", "No", "Yes", "Yes", "Yes", "Y...
## $ sudden.weight.loss <chr> "No", "No", "No", "Yes", "Yes", "No", "No", "Yes...
## $ weakness           <chr> "Yes", "Yes", "Yes", "Yes", "Yes", "Yes", "Yes",...
## $ Polyphagia         <chr> "No", "No", "Yes", "Yes", "Yes", "Yes", "Yes", "...
## $ Genital.thrush     <chr> "No", "No", "No", "Yes", "No", "No", "Yes", "No"...
## $ visual.blurring    <chr> "No", "Yes", "No", "No", "Yes", "Yes", "No", "Ye...
## $ Itching            <chr> "Yes", "No", "Yes", "Yes", "Yes", "Yes", "No", "...
## $ Irritability       <chr> "No", "No", "No", "No", "Yes", "No", "No", "Yes"...
## $ delayed.healing    <chr> "Yes", "No", "Yes", "Yes", "Yes", "Yes", "Yes", ...
## $ partial.paresis    <chr> "No", "Yes", "No", "No", "Yes", "No", "Yes", "Ye...
## $ muscle.stiffness   <chr> "Yes", "No", "Yes", "No", "Yes", "Yes", "No", "Y...
## $ Alopecia           <chr> "Yes", "Yes", "Yes", "No", "Yes", "Yes", "No", "...
## $ Obesity            <chr> "Yes", "No", "No", "No", "Yes", "Yes", "No", "No...
## $ class              <chr> "Positive", "Positive", "Positive", "Positive", ...
```

My next step was to confirm that the remaining variables only contained yes or no values by checking the number of unique values in the set.

```
sapply(data_set, function(x) length(unique(x)))
```

```
##                Age             Gender           Polyuria         Polydipsia
##                 51                  2                  2                  2
## sudden.weight.loss           weakness         Polyphagia     Genital.thrush
##                  2                  2                  2                  2
##    visual.blurring            Itching       Irritability    delayed.healing
##                  2                  2                  2                  2
##    partial.paresis   muscle.stiffness           Alopecia            Obesity
##                  2                  2                  2                  2
##              class
##                  2
```

The results reflected that all but “Age” contained only two unique values. I also checked to see if there were any “NA” or blank values in the data set and there were none. Because the dataset was complete I did not need to perform any data cleaning or imputation.

The last variable in the dataset was “class” and reflected whether or not the respondent had diabetes (Positive) or not (Negative). This was the variable the model needed to predict so I converted it from a character string to a factor.

```
data_set$class <- factor(data_set$class)
```

I then plotted the class values in a bar graph to see how the Positive and Negative values were distributed. As shown, there were approximately 50% more Positive outcomes than Negative with 320 of the 520 patients having diabetes.

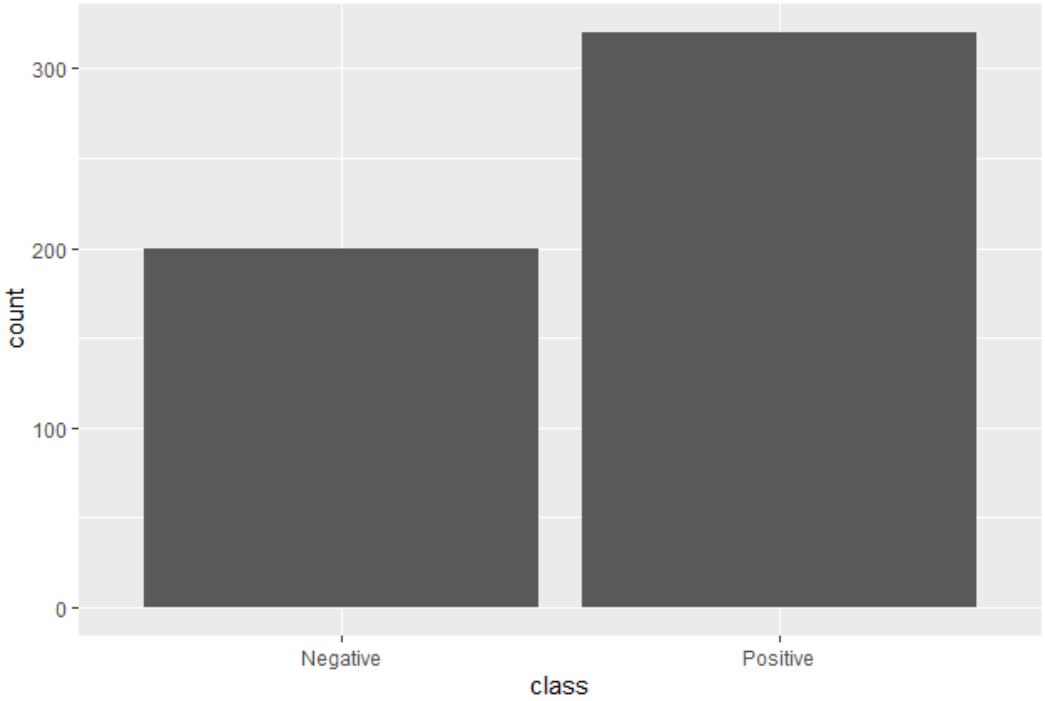


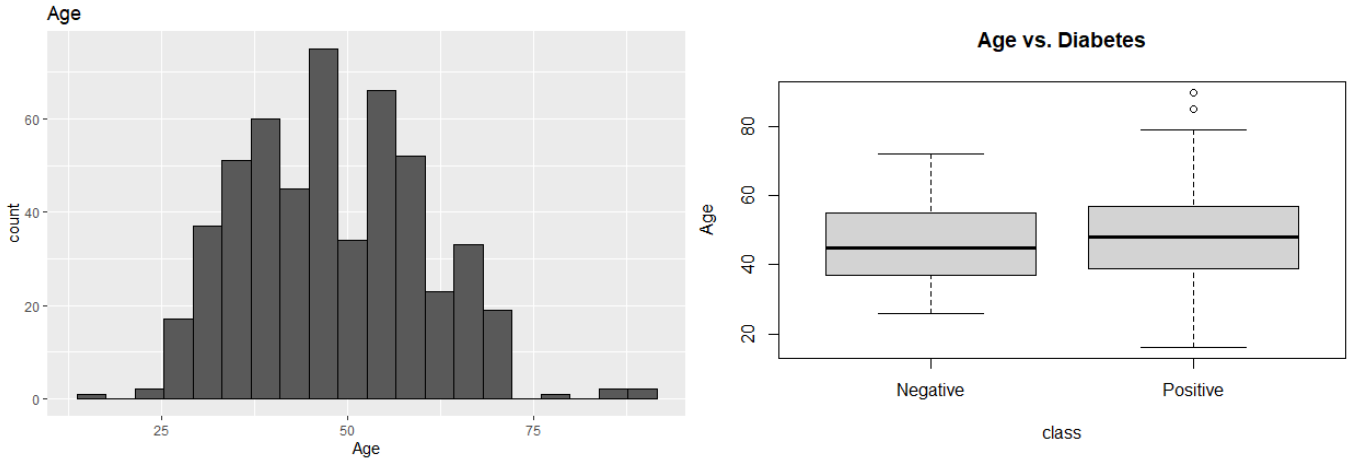
Chart reflects the distribution of diabetic respondents.

Since the variable “Age” was the only numeric, continuous variable, it seemed reasonable to visualize and observe the distribution. The ages of the respondents varied from 16 to 90 years old and were summarized as:

```
summary(data_set$Age)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   16.00   39.00   47.50   48.03   57.00   90.00
```

The ages followed a near-normal distribution as shown in the graph on the left. Also notable was the minor impact age had on diabetic diagnoses as the boxplot on the right reflects.



Age distribution of respondents.

Age did not seem to be a relevant factor in diagnosing diabetes. Therefore, it seemed prudent to examine how the other variables would relate to a diabetic outcome. In order to accomplish this, I converted all variables other than “Age” and “class” to factors then built contingency tables to see if any symptoms stood out more than others.

```
x = c()
x = names(data_set)
x = x[-17] # Remove class
x = x[-1] # Remove Age
# Convert remaining variables to factors then print contingency tables
for (i in x){
  data_set[,i]=as.factor(data_set[,i])
}
```

It would be more apparent if the numbers were larger in opposing corners of the table. For example, if there were larger numbers in the “Negative/No” and “Positive/Yes” corners then it would appear that the condition impacted the diabetic outcome. The three tables that reflected this pattern most noticeably were the variables, “Polyuria,” “Polydipsia,” and “Alopecia” as shown below.

- Polyuria

```
table(data_set$Polyuria, data_set$class)
```

```
##
##       Negative Positive
##   No       185       77
##   Yes       15      243
```

- Polydipsia

```
table(data_set$Polydipsia, data_set$class)
```

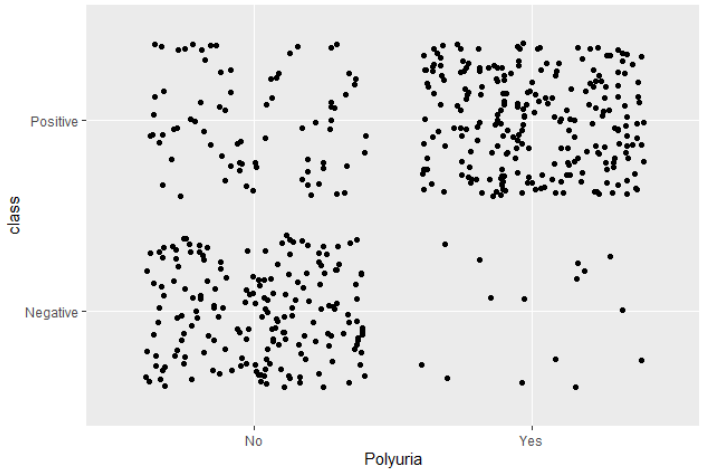
```
##
##       Negative Positive
##   No       192       95
##   Yes        8      225
```

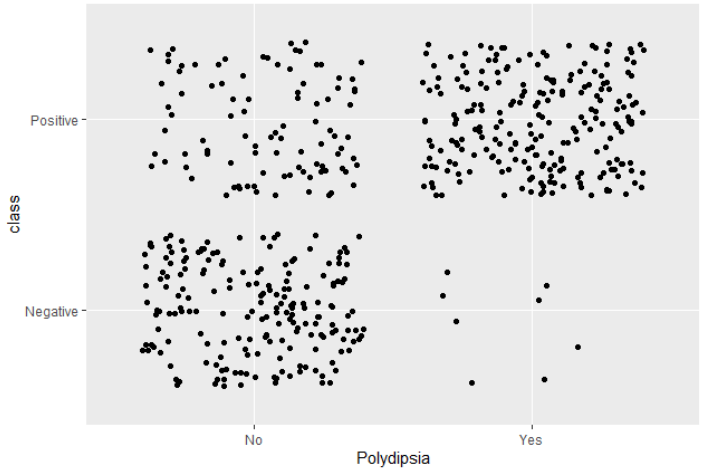
- Alopecia

```
table(data_set$Alopecia, data_set$class)
```

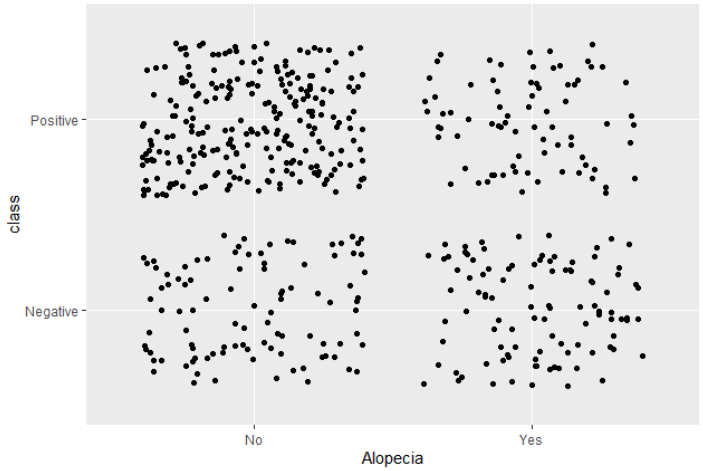
```
##
##       Negative Positive
##   No        99      242
##   Yes      101       78
```

The next step was to perform a visualization of the three variables based on the initial evaluation of the tables. The scatterplots clearly show correlation between a patient having Polyuria and Polydipsia symptoms and having diabetes.

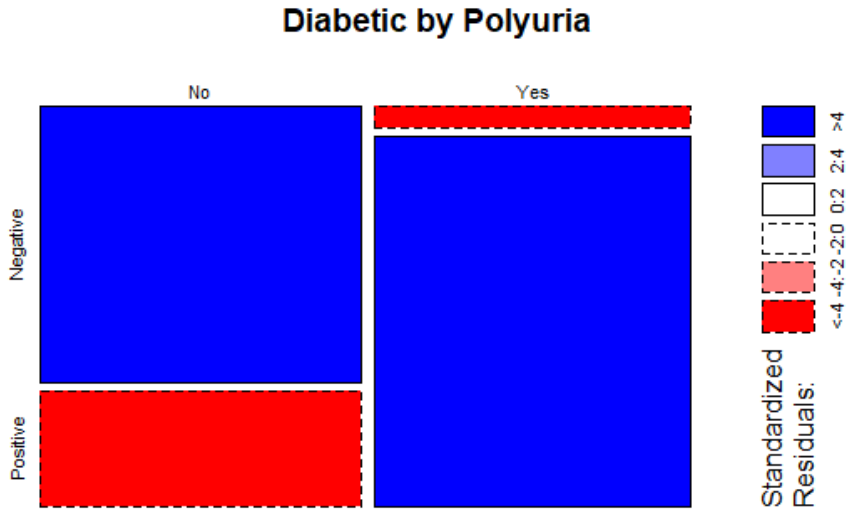


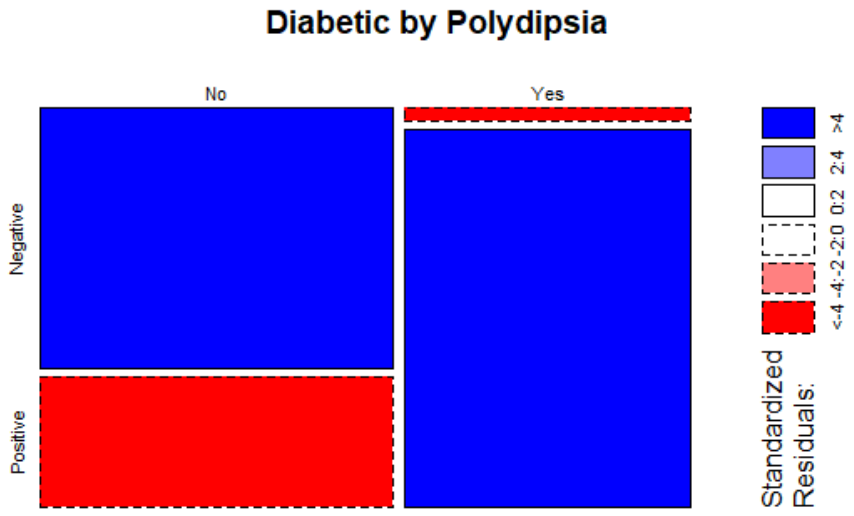


The scatterplot for Alopecia was not so distinct.



Scatterplots can be deceiving at times so I prepared two mosaic plots to further visualize the contingency tables in question. The plots included shading to reflect the level of the residual for the combination of levels. The darker shades represent opposite extremes for what would be the expected number of observations with independent variables.





As shown, Polyuria and Polydipsia are seemingly strong indicators of a person's diabetic outcome.

I used two modeling approaches to analyze the data set; logistic regression and random forest. In both cases, I used the same training and testing subsets of the dataset. The training set consisted of 80% of the observations while the testing set consisted of the remaining 20%.

Logistic Regression

This method is useful for predicting binary outcomes such as this. The Generalized Linear Model was fit using the glm() function. The summary of the model revealed "PolyuriaYes" and "PolydipsiaYes" had influenced class most positively with estimates of 4.05376 and 4.68610 respectively.

```
LR_model <- glm(class ~.,family=binomial(link='logit'),data=train_set)
summary(LR_model)
```

```
##
## Call:
## glm(formula = class ~ ., family = binomial(link = "logit"), data = train_set)
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -2.79113  -0.24844   0.00474   0.07844   2.92890
##
## Coefficients:
##                      Estimate Std. Error z value Pr(>|z|)
## (Intercept)           2.51791    1.14927   2.191 0.028460 *
## Age                  -0.03906    0.02629  -1.486 0.137368
## GenderMale           -4.46340    0.64814  -6.886 5.72e-12 ***
## PolyuriaYes           4.05376    0.73613   5.507 3.65e-08 ***
## PolydipsiaYes         4.68610    0.86924   5.391 7.01e-08 ***
## sudden.weight.lossYes 0.43089    0.58346   0.739 0.460202
## weaknessYes           0.45947    0.60351   0.761 0.446462
## PolyphagiaYes         1.21199    0.61472   1.972 0.048653 *
## Genital.thrushYes     1.71098    0.63764   2.683 0.007290 **
## visual.blurringYes    0.84018    0.70317   1.195 0.232151
## ItchingYes           -2.67874    0.71631  -3.740 0.000184 ***
## IrritabilityYes       2.30357    0.66245   3.477 0.000506 ***
## delayed.healingYes    0.04917    0.65837   0.075 0.940461
## partial.paresisYes    0.89849    0.57871   1.553 0.120526
## muscle.stiffnessYes  -0.96206    0.66604  -1.444 0.148611
## AlopeciaYes          -0.23749    0.68844  -0.345 0.730124
## ObesityYes           -0.35675    0.65376  -0.546 0.585282
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 552.40  on 413  degrees of freedom
## Residual deviance: 138.88  on 397  degrees of freedom
## AIC: 172.88
##
## Number of Fisher Scoring iterations: 8
```

I used the anova() function to test for the significance of the model using the chi-square test. This tests the model based on the change in deviance when the predictors of my model are added to the null model.

```
anova(LR_model, test='Chisq')
```

```
## Analysis of Deviance Table
##
## Model: binomial, link: logit
##
## Response: class
##
## Terms added sequentially (first to last)
##
##
##                   Df Deviance Resid. Df Resid. Dev  Pr(>Chi)
## NULL                                413     552.40
## Age                1    3.566       412     548.83  0.058976 .
## Gender             1  112.201       411     436.63 < 2.2e-16 ***
## Polyuria           1  166.195       410     270.43 < 2.2e-16 ***
## Polydipsia         1   74.437       409     196.00 < 2.2e-16 ***
## sudden.weight.loss 1    3.090       408     192.91  0.078777 .
## weakness           1    0.051       407     192.86  0.821237
## Polyphagia         1    6.545       406     186.31  0.010518 *
## Genital.thrush     1    6.203       405     180.11  0.012755 *
## visual.blurring    1    0.267       404     179.84  0.605050
## Itching            1   21.481       403     158.36 3.574e-06 ***
## Irritability       1   13.512       402     144.85  0.000237 ***
## delayed.healing    1    0.074       401     144.77  0.785646
## partial.paresis    1    3.050       400     141.72  0.080722 .
## muscle.stiffness   1    2.466       399     139.26  0.116339
## Alopecia           1    0.075       398     139.18  0.784850
## Obesity            1    0.301       397     138.88  0.583385
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

The next step was to predict the outcomes in the test set. Once I performed the predict() function, I needed to convert the results into a binary “Positive” or “Negative” so that they could used as factors. I then created a confusion matrix to assess the results.

```
LR_result <- predict(LR_model, newdata = test_set, type='response')
LR_result <- ifelse(LR_result > 0.5,"Positive","Negative")
LR_result <- as.factor(LR_result)
confusionMatrix(data=LR_result, reference=test_set$class)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction Negative Positive
##   Negative       39        6
##   Positive        1       60
##
##                Accuracy : 0.934
##                  95% CI : (0.8687, 0.973)
##     No Information Rate : 0.6226
##     P-Value [Acc > NIR] : 1.276e-13
##
##                   Kappa : 0.8628
##
##  Mcnemar's Test P-Value : 0.1306
##
##             Sensitivity : 0.9750
##             Specificity : 0.9091
##          Pos Pred Value : 0.8667
##          Neg Pred Value : 0.9836
##              Prevalence : 0.3774
##          Detection Rate : 0.3679
##    Detection Prevalence : 0.4245
##       Balanced Accuracy : 0.9420
##
##        'Positive' Class : Negative
##
```

This resulted in a 93.4% accuracy with 97.5% sensitivity and 90.9% specificity.

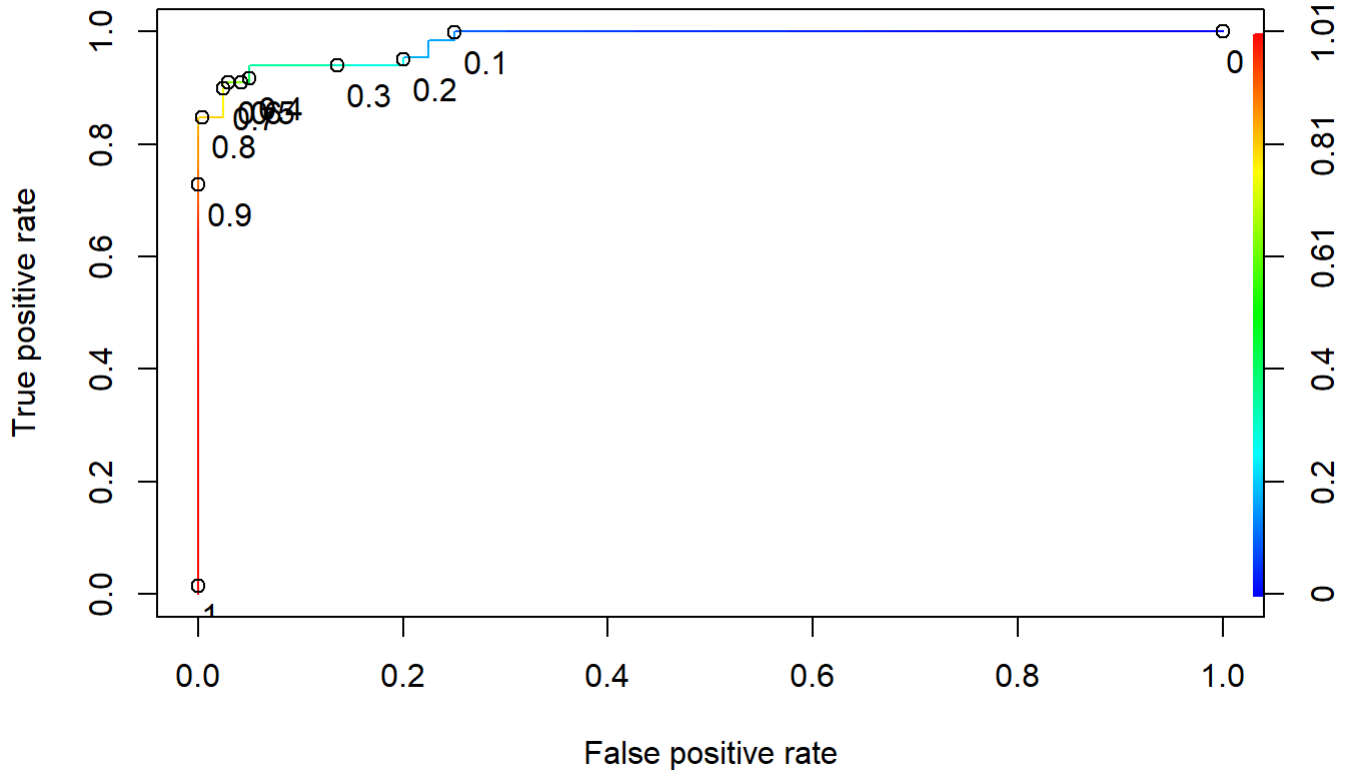
Accuracy reflects whether the test actually measured what it was supposed to and gives a general understanding of the performance of a classifier. Sensitivity is the measure of the ability for a prediction model to select instances of a certain combination of variables and corresponds to the True Positive rate. Conversely, specificity corresponds to the True Negative rate or the probability that the diagnosis is non-diabetic when the patient is in fact, non-diabetic.

These are good results, however; I wanted to better evaluate the performance of the model. I learned the Area Under Curve - Receiver Operating Characteristics (AUC-ROC) curve is “one of the most important evaluation metrics for checking any classification model’s performance.” More can be found on this topic at https://towardsdatascience.com/understanding-auc-roc-curve-68b2303cc9c5 (https://towardsdatascience.com/understanding-auc-roc-curve-68b2303cc9c5).

The graph below reflects the ROC curve or, the performance of the model. It is a comparison between the sensitivity and specificity. If a model’s performance is more accurate, the line will curve more toward the upper left corner. If the model’s performance is less accurate, it will follow a linear, 45-degree line from bottom left to upper right more closely.

```
prediction1 <- predict(LR_model, newdata=test_set, type="response")
ROC_pred <- prediction(prediction1, test_set$class)
ROC_perf <- performance(ROC_pred, measure = "tpr", x.measure = "fpr")

plot(ROC_perf, colorize = TRUE, text.adj = c(-0.2,1.7), print.cutoffs.at = seq(0,1,0.1))
```



Finally, I produced the AUC score to see precisely how much the model is able to predict a diabetic outcome or not. The higher the AUC, the better it is able to predict an outcome.

```
auc <- performance(ROC_pred, measure = "auc")
auc <- auc@y.values[[1]]
auc
```

```
## [1] 0.9833333
```

Here, the AUC is an impressive .98333.

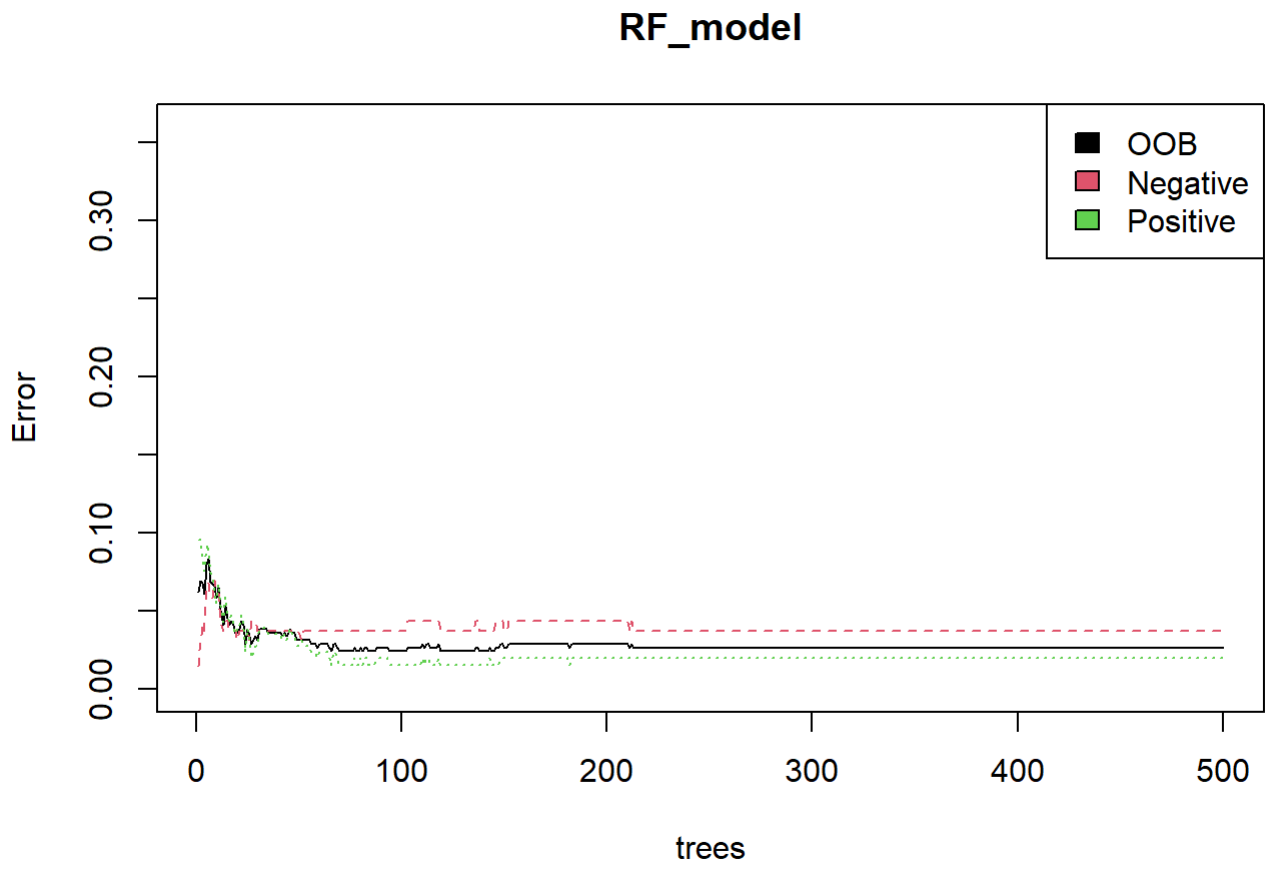
Random Forest Classifier

The second modeling approach I used was the Random Forest Classifier. This is a classification algorithm that uses several individual decision trees that each generate a class prediction. In the end, the class that is produced the most often becomes the model's prediction. Like the logistic regression method, I began by creating the model.

```
RF_model <- randomForest(factor(class) ~ Age + Gender + Polyuria + Polydipsia + sudden.weight.lo
ss + weakness + Polyphagia + Genital.thrush + visual.blurring + Itching + Irritability + delaye
d.healing + partial.paresis + muscle.stiffness + Alopecia + Obesity, data = train_set)
```

Then, I plotted the error rate of the model.

```
plot(RF_model, ylim = c(0, 0.36))
legend("topright", colnames(RF_model$err.rate), col = 1:3, fill = 1:3)
```



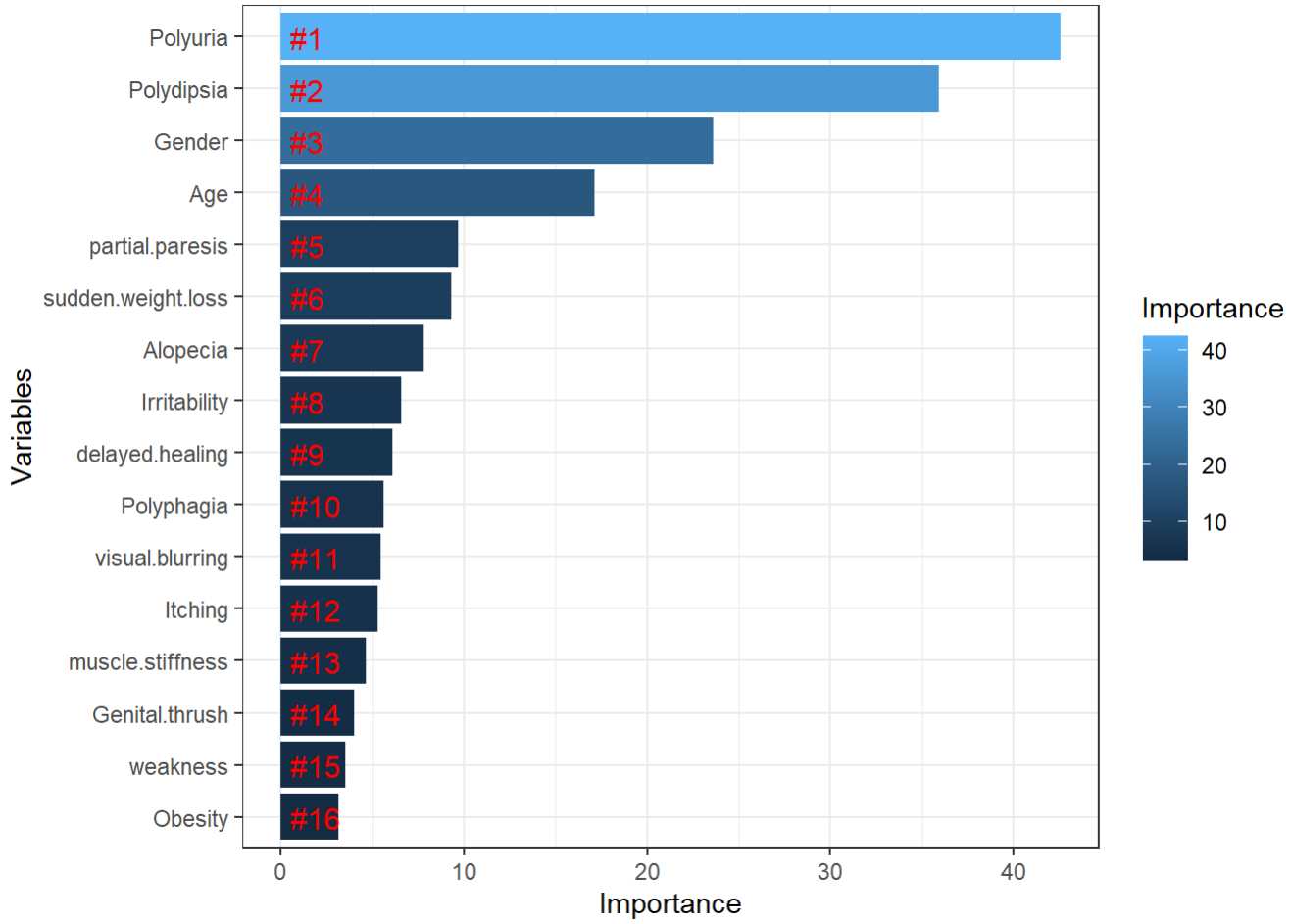
The OOB line is the overall error rate and appears to be very low at around 5%. My next effort was to visualize the importance of the variables that the model produced. A very useful way to see this is in a bar graph.

```
# Rank the variables and plot on bar graph
importance <- importance(RF_model)

# Lower Gini means stronger factor in partitioning data into classes
varImportance <- data.frame(Variables = row.names(importance), Importance = round(importance[ ,
'MeanDecreaseGini'],2))

rankImportance <- varImportance %>%
  mutate(Rank = paste0('#',dense_rank(desc(Importance))))

ggplot(rankImportance, aes(x = reorder(Variables, Importance), y = Importance, fill = Importanc
e)) +
  geom_bar(stat='identity') +
  geom_text(aes(x = Variables, y = 0.5, label = Rank), hjust=0, vjust=0.55, size = 4, colour =
'red') +
  labs(x = 'Variables') +
  coord_flip() +
  theme_bw()
```



Interestingly, the top three most important variables were Polyuria, Polydipsia, and Gender; the same three variables that influenced the outcome most significantly in the logistic regression method as well.

The final step to test this method was to see how it performed on the test set of data.

```
prediction2 <- predict(RF_model, test_set)
RF_predictions <- data.frame(test_set, class=prediction2)
table(RF_predictions$class)
```

```
##
## Negative Positive
##       40       66
```

```
table(test_set$class)
```

```
##
## Negative Positive
##       40       66
```

The results were identical between the predicted outcomes in the test set and the actual outcomes in the full dataset. The model predicted 66 of 106, or 62% of the patients would be diagnosed with diabetes. In comparison, the actual number of patients with diabetes in the entire data set was 320 of 520, also 62%.

3 Results

I analyzed this dataset using both linear regression and random forest classifier and found the latter to be a better predictor of a diabetic diagnosis. The linear regression method proved to be very accurate with a .9833 AUC meaning, it is nearly 100% accurate. Random forest, however; proved to be 100% accurate on this dataset.

4 Conclusion and Future Work

The goal of this HarvardX PH125.9x Data Science Capstone Project was to use data science skills to develop a model that could predict diabetes diagnoses using early health predictors from the Early Stage Diabetes Risk Prediction Dataset. It was motivated by personal experience witnessing the devastating effects of the disease and a desire to help others detect the warning signs early before it is too late.

Certain variables proved to be more significant than others in this study including polyuria and polydipsia. While the variables were very impactful in this dataset, their significance may be overstated due to the relatively small size of the dataset when you consider just how many people suffer from this disease. With only 520 patient observations based on simple yes or no responses, it is difficult to state that the findings are completely reliable. It would be very beneficial to use a larger dataset with precise medical measurements in future work.