Predicting the Onset of Diabetes Based on Diagnostic Measures

HarvardX Data Science: Capstone CYO Project

Ramone Jackson

6/20/2020

Contents

Introduction

Many persons from around the world are suffering from diabetes today. As such, I will be preparing a diagnostic system that will help to predict patients with diabetes.

For this project we will be using the diabetes dataset from www.kaggle.com. This dataset consist of 9 variables and 768 rows of records. Also note that each row consists of a patient's data and there diabetes status.Here 1 represents having diabetes while 0 represents not having diabetes.

The goal of this project is to find a good model hat could be use to predict if the patient has diabetes or not mased on the various measure. Since the doctors are focussing on Corona cases, it would help to diagnose patiemts easier to free up spaces for Corona patients in the hospital.

To find a really good model to use, we have chosen to use accuray, f1 test and sensitivity to depict which model we should use. The model will be accepted if the aerage of all three matrics is greater than 70%. Lastly, the 5-fold cross-validation as our cross-validation method for assessing.

The definition of our matrics are below:

- Accuracy - This indicates the number of outcomes that was currectly predicted by the model.
- F1 score - This indicates the harmonic mean of precision and sensitivity.
- Sensitivity - This indicates the proportion of people 'with diabetes' that were correctly classified.

Methods

Data Exploration

This dataset consists of 768 observations with 9 variables. Addionally, the dataset consist of 52 different ages were 268 patients have diabeeties and 500 does not. The top 6 of patients were: 22, 21, 25, 24, 23 and 28, which contributed to 39% of the overall patients. Therefore suggesting that there were many patients in their 20s. Lastly, The dataset also consited of 0 nulls

data dataset

```
## Loading the dataset from my github profile
data <- read.csv("https://raw.githubusercontent.com/RamoneRJackson/HarvardX_CYO/master/datasets_228_482

#####Evaluating the Data

#Finding the amount of observations and variables in the datset
dim(data)
```

```
## [1] 768   9
```

```
#Viewing the datatypes of the variables in the dataset
str(data)
```

```
## 'data.frame':    768 obs. of  9 variables:
##  $ Pregnancies             : int  6 1 8 1 0 5 3 10 2 8 ...
##  $ Glucose                 : int  148 85 183 89 137 116 78 115 197 125 ...
##  $ BloodPressure           : int  72 66 64 66 40 74 50 0 70 96 ...
##  $ SkinThickness           : int  35 29 0 23 35 0 32 0 45 0 ...
##  $ Insulin                 : int  0 0 0 94 168 0 88 0 543 0 ...
##  $ BMI                     : num  33.6 26.6 23.3 28.1 43.1 25.6 31 35.3 30.5 0 ...
##  $ DiabetesPedigreeFunction: num  0.627 0.351 0.672 0.167 2.288 ...
##  $ Age                     : int  50 31 32 21 33 30 26 29 53 54 ...
##  $ Outcome                 : int  1 0 1 0 1 0 1 0 1 1 ...
```

```
#Number of different ages in the dataset
n_distinct(data$Age)
```

```
## [1] 52
```

```
#Checking to see the number of diabetes patients
data%>%group_by(Outcome)%>%summarise(count=n())%>%
   kable() %>%
   kable_styling(bootstrap_options = c("striped", "hover", "condensed", "responsive"),
                 position = "center",
                 font_size = 10,
                 full_width = FALSE)
```

Outcome	count
0	500
1	268

```r
#Finding out if there any data with N/As
sapply(data, function(m){
  sum(is.na(m))
})%>%
   kable() %>%
   kable_styling(bootstrap_options = c("striped", "hover", "condensed", "responsive"),
                 position = "center",
                 font_size = 10,
                 full_width = FALSE)
```

	x
Pregnancies	0
Glucose	0
BloodPressure	0
SkinThickness	0
Insulin	0
BMI	0
DiabetesPedigreeFunction	0
Age	0
Outcome	0

```r
#Finding the top 6 ages represented in the dataset
data%>%group_by(Age)%>%summarise(count=n())%>%
  arrange(desc(count))%>%head(n=6)%>%
   kable() %>%
   kable_styling(bootstrap_options = c("striped", "hover", "condensed", "responsive"),
                 position = "center",
                 font_size = 10,
                 full_width = FALSE)
```

Age	count
22	72
21	63
25	48
24	46
23	38
28	35

```r
#Top 6 ages of patients with diabetes
data%>%filter(Outcome==1)%>%group_by(Age)%>%summarise(count=n())%>%
  arrange(desc(count))%>%head(n=6)%>%
   kable() %>%
   kable_styling(bootstrap_options = c("striped", "hover", "condensed", "responsive"),
                 position = "center",
                 font_size = 10,
                 full_width = FALSE)
```

Age	count
25	14
29	13
31	13
41	13
22	11
43	11

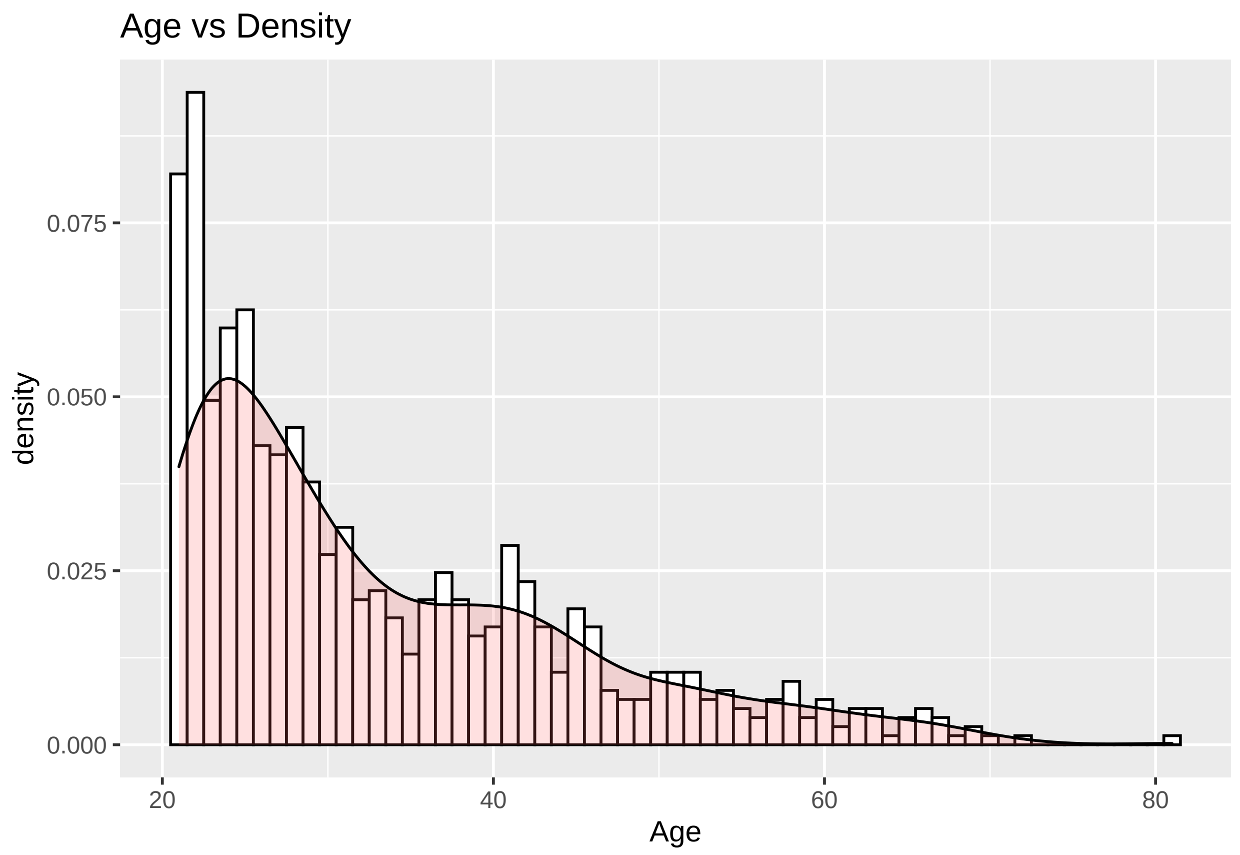
```r
#Viewing the first 6 rows of the dataset
data%>%head(n=6)%>%
  kable() %>%
  kable_styling(bootstrap_options = c("striped", "hover", "condensed", "responsive"),
                position = "center",
                font_size = 10,
                full_width = FALSE)
```

Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunction	Age	Outcome
6	148	72	35	0	33.6	0.627	50	1
1	85	66	29	0	26.6	0.351	31	0
8	183	64	0	0	23.3	0.672	32	1
1	89	66	23	94	28.1	0.167	21	0
0	137	40	35	168	43.1	2.288	33	1
5	116	74	0	0	25.6	0.201	30	0

Visualization

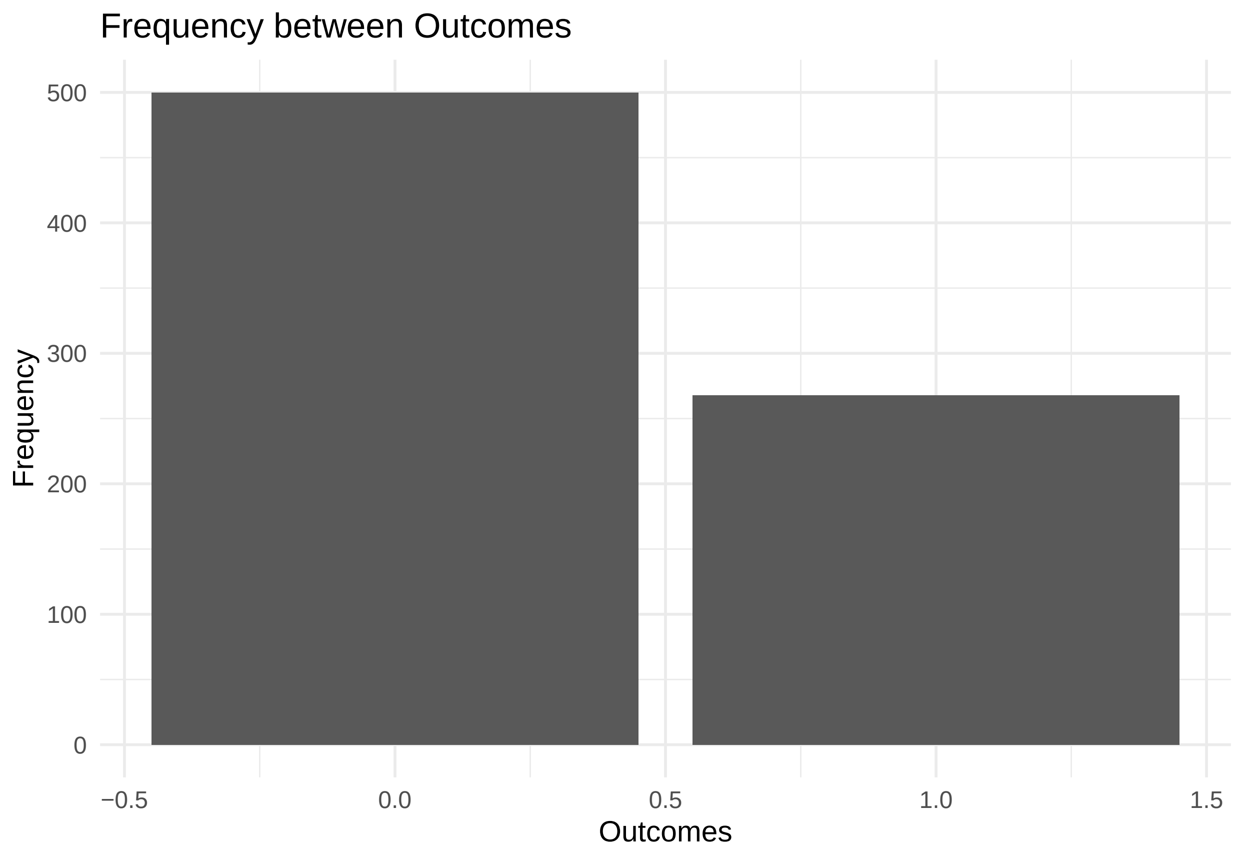
Age vs Density

This visualization shows that most of the persons chosen for the dataset aged between 20 and 40.

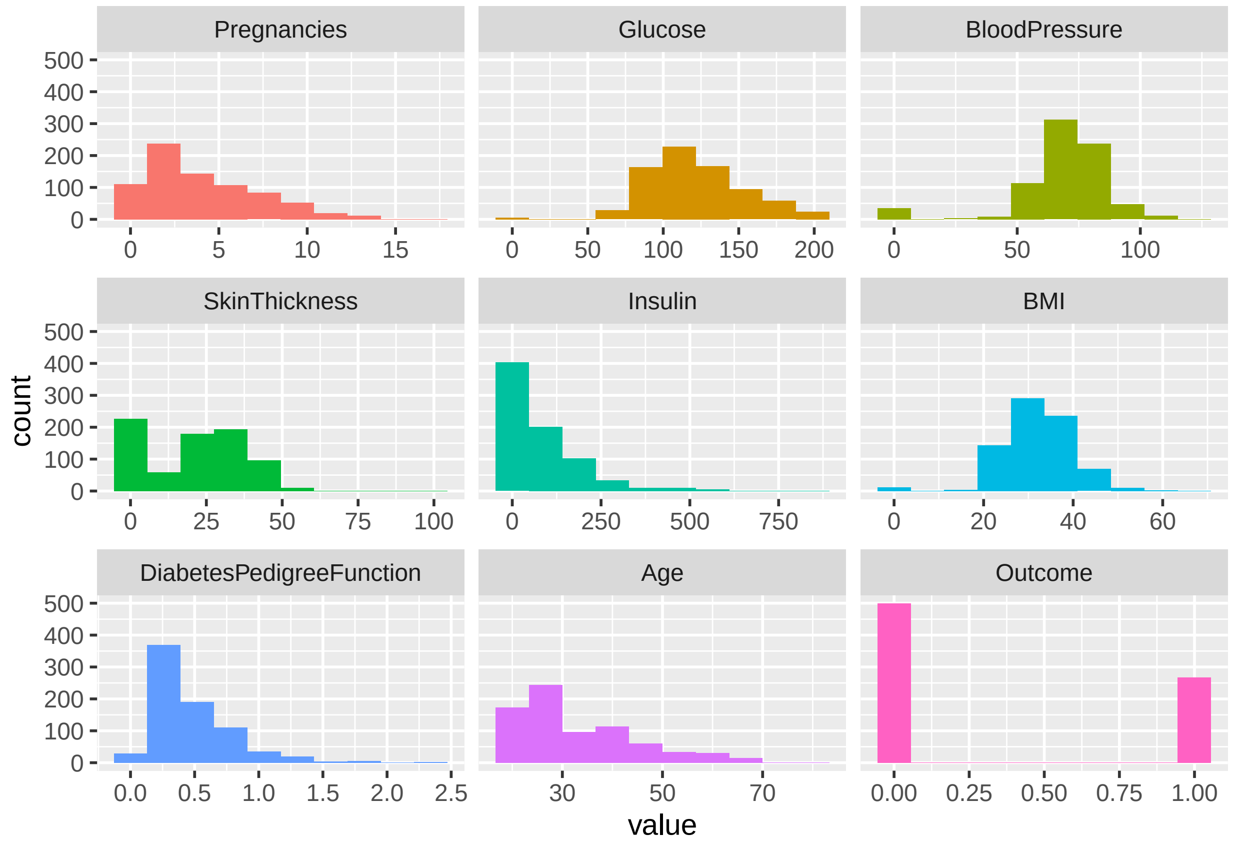


Frequency between Outcomes

The number of persons without diabetes in the dataset almost doubles the amount of person wth the disease.

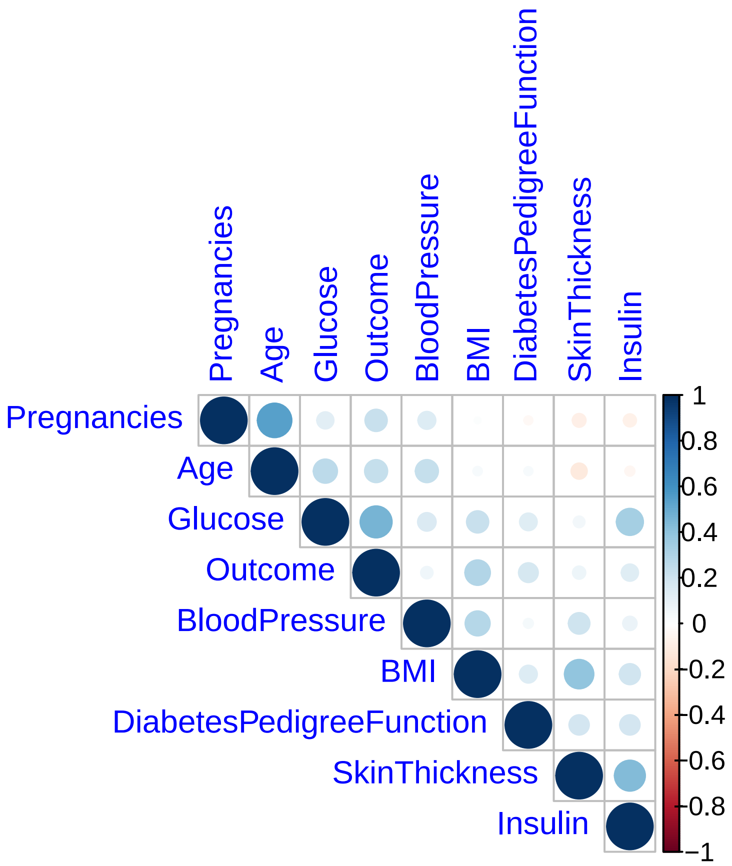

Frequency between Outcomes
Frequency
500
400
300
200
100
0
−0.5
0.0
0.5
1.0
1.5
Outcomes

Plot on the distribution in variables



Correlation between variables

There is no high level of correlation. Therefore all variables should be kept.



Data Cleaning

1. Setting outcome to “yes” or “no”

```
# Setting outcome as a character variable
data<- data%>%mutate(Outcome=
                          ifelse(Outcome ==1,"yes","no"))
```

2. Setting outcome as a factor

```
data$Outcome <- as.factor(data$Outcome)
```

The data after cleaning

```
#Viewing the first 10 rows of the dataset
data%>%head(n=10)%>%
   kable() %>%
   kable_styling(bootstrap_options = c("striped", "hover", "condensed", "responsive"),
                 position = "center",
                 font_size = 10,
                 full_width = FALSE)
```

Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunction	Age	Outcome
6	148	72	35	0	33.6	0.627	50	yes
1	85	66	29	0	26.6	0.351	31	no
8	183	64	0	0	23.3	0.672	32	yes
1	89	66	23	94	28.1	0.167	21	no
0	137	40	35	168	43.1	2.288	33	yes
5	116	74	0	0	25.6	0.201	30	no
3	78	50	32	88	31.0	0.248	26	yes
10	115	0	0	0	35.3	0.134	29	no
2	197	70	45	543	30.5	0.158	53	yes
8	125	96	0	0	0.0	0.232	54	yes

Modeling

Creation of Training set and Validation Set

Firstly, we have chosen the 5-fold cross-validation as our cross-validation method for assessing the model's performance. Because there are not many records in the dataset, we ave chosen to split the data 25:75 with 25% going to the validation set and 75% going to the training set. This will allow us to have a good portion of our datst to train while allowing the a large enough valiation set to give reliable estimates.

Before the split is done, we will set the seed to 1, then use the createDataPartition functin to help split the dataset for us. Additionally, we will create a function *"train.control"* to will be used to set the train controll as 5-fold cross-validation. Lastly, we will train our algorithm on the training set and use the validation set to predict and compute our apparent errors.

```
# Split the dataset into train and test set
# Set seed as a starting point

set.seed(1, sample.kind='Rounding')
train_index <- createDataPartition(y = data$Outcome, p = 0.25, list = F)

training_set <- data[-train_index,]
validation_set <- data[train_index,]


#Showing the dimension of the training set
dim(training_set)
```

```
## [1] 576   9
```

```
#Showing the dimension of the validation set
dim(validation_set)
```

```
## [1] 192   9
```

```
# Set seed as a starting point
set.seed(1, sample.kind='Rounding')

# Defining the training control that will be used for the model

train.control <- trainControl(method = "cv", #Cross Validation method
                              classProbs = TRUE,
                              number = 5,  #The number of folds
                              summaryFunction = twoClassSummary)

#Function to determine the avg of the average of the matrics
Metrics_Avg <- function(f1_score, accuracy, sensitivity)
  {
     (f1_score + accuracy + sensitivity)/3
    }
```

```
 head(validation_set)%>%
   kable() %>%
   kable_styling(bootstrap_options = c("striped", "hover", "condensed", "responsive"),
                 position = "center",
                 font_size = 10,
                 full_width = FALSE)
```

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunction	Age	Outco
13	10	139	80	0	0	27.1	1.441	57	no
16	7	100	0	0	0	30.0	0.484	32	yes
19	1	103	30	38	83	43.3	0.183	33	no
24	9	119	80	35	0	29.0	0.263	29	yes
27	7	147	76	0	0	39.4	0.257	43	yes
29	13	145	82	19	110	22.2	0.245	57	no

```
head(training_set)%>%
  kable() %>%
  kable_styling(bootstrap_options = c("striped", "hover", "condensed", "responsive"),
                 position = "center",
                 font_size = 10,
                 full_width = FALSE)
```

Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunction	Age	Outcome
6	148	72	35	0	33.6	0.627	50	yes
1	85	66	29	0	26.6	0.351	31	no
8	183	64	0	0	23.3	0.672	32	yes
1	89	66	23	94	28.1	0.167	21	no
0	137	40	35	168	43.1	2.288	33	yes
5	116	74	0	0	25.6	0.201	30	no

Naive Guessing Mode

This is a base case method that was chosen by assuming that if the Diabetes Pedigree Function is larger than 0.5 then the patient would have diabetes.

```
#Printing the results of our Model
print(Accuracy_guess)
```

```
## [1] 0.6354167
```

```
print(F1_Score_guess)
```

```
## [1] 0.4852941
```

```
print(Sensitivity_guess)
```

```
## [1] 0.4925373
```

As you can see, that ths model does not meet the standard as all three measures are pretty low.

Naive Bayes QDA Model

Naive Bayes classifiers are a family of probabilistic classifiers that make a very strong independence assumption about the data. In particular, naive Bayes classifiers assume that all X variables are independent. This strong assumption is rarely true, however, frequently leads to simple and effective classifiers.

For this project, e will be using the QDA version of the Naive Bayes as

```r
#Printing the results of our Model
print(Accuracy_qda)
```

```
## [1] 0.7291667
```

```r
print(F1_Score_qda)
```

```
## [1] 0.59375
```

```r
print(Sensitivity_qda)
```

```
## [1] 0.5671642
```

This model is still not as good as we would like it to be.

K-Nearest Neighbors Model

K-nearest neighbors (kNN) estimates the conditional probabilities in a similar way to bin smoothing. However, kNN is easier to adapt to multiple dimensions.

```r
#Printing the results of our Model
print(Accuracy_knn)
```

```
## [1] 0.7916667
```

```r
print(F1_Score_knn)
```

```
## [1] 0.6363636
```

```r
print(Sensitivity_knn)
```

```
## [1] 0.5223881
```

The accurcy for this modl was good but we could increase both the F1 Score and the Sensitivity some more.

Logistic Regression Model

Logistic regression is an extension of linear regression that assures that the estimate of conditional probability is between 0 and 1. Note that with this model, we can no longer use least squares. Instead we compute the maximum likelihood estimate (MLE).

In R, we can fit the logistic regression model with the function glm() (generalized linear models). In order to use this function, we would have to use the family function to specify the binomial version of this model.

```r
#Printing the results of our GLM Model
print(Accuracy_glm)
```

```
## [1] 0.78125
```

```r
print(F1_Score_glm)
```

```
## [1] 0.6557377
```

```r
print(Sensitivity_glm)
```

```
## [1] 0.5970149
```

Even though this model is not that bad, we can still get an even better model.

Random Forrest Model

Ths is a well known machine learning approach that solves the errors from the decision tree. In rain forest, the whole dataset is not required for making a splitting decision. Only some aggregated information is required and to increase the effeciency of both the F1 Score and the sensitivity, we will tune the model to 7.

```r
#Printing the results of our  Model
print(Accuracy_rf)
```

```
## [1] 0.7864583
```

```r
print(F1_Score_rf)
```

```
## [1] 0.6771654
```

```r
print(Sensitivity_rf)
```

```
## [1] 0.641791
```

This is a really good model to use to fit the data, as such, we will stop here.

Result

This is the summary results for all the model built, that were trained on training set and validation on the test set.

```r
results%>%
  kable() %>%
  kable_styling(bootstrap_options = c("striped", "hover", "condensed", "responsive"),
                position = "center",
                font_size = 10,
                full_width = FALSE)
```

Model_Name	Accuracy	F1_Score	Sensitivity	Matric_Avg
Naive Guessing Model	0.6354167	0.4852941	0.4925373	0.5377494
Naive Bayes QDA Model	0.7291667	0.5937500	0.5671642	0.6300269
K-nearest neighbors Model	0.7916667	0.6363636	0.5223881	0.6501395
Logistic Regression Model	0.7812500	0.6557377	0.5970149	0.6780009
Random Forrest Model	0.7864583	0.6771654	0.6417910	0.7018049

As expectd, the worst overall model was our Naive Guessing Model, followed by the Naive Bayes Model. Even though the Naive Bayes Model had a higher the K-Nearest Neighbors Model, its overall average of the three metrics was still lower than the K-Nearest Neighbors Model. The Logistic Regression Model was the second best model however, the avg of the matric did not reach the 70% acceptance level.

From our results, it is safe to say that the Random Forrest Model is a good model to predict if the patient has diabeties as it gives a good passing grade for all three of our metrics.

Conclusion

In conclusion, after fitting various models on the data, the best model would have been Random Forrest Model.

There were minimal limitation since the data was partially clean. However, the distribution of ages could have been sbreaded out more since most of the patients were in their 20s.

In the future, I would recoment havng a datset with a better mix in ages.