Diabetes Model | Machine Learning Project

Elton Costa

3/2/2022

1. Introduction

Diabetes is a group of metabolic disorders characterized by a high blood sugar level over a prolonged period of time. Symptoms often include frequent urination, increased thirst and increased appetite. If left untreated, diabetes can cause many health complications.

As of 2019, an estimated 463 million people had diabetes worldwide (8.8% of the adult population). Rates are similar in women and men. Trends suggest that rates will continue to rise. Diabetes at least doubles a person's risk of early death. In 2019, diabetes resulted in approximately 4.2 million deaths. It is the 7th leading cause of death globally. The global economic cost of diabetes-related health expenditure in 2017 was estimated at US$727 billion. In the United States, diabetes cost nearly US$327 billion in 2017. Average medical expenditures among people with diabetes are about 2.3 times higher.[1]

The internet provides a range symptoms for people to watch out for, however using Google as a self diagnosis tool can be unreliable and, quite frankly, scary. With hospitals being extremely busy during the COVID-19 pandemic, it would impact the life quality if people could find out if they are at risk of being diabetic without having to visit a doctor.

This project aims to develop a machine learning model that predicts whether a patient is at risk of being diabetic. The data being worked with is the Early stage diabetes risk prediction dataset. It was created using questionnaires from the patients of Sylhet Diabetic Hospital (Bangladesh) and has been approved by a doctor [2].

Additionally, this document is structured as follows:

1. Introduction
2. Method and Data pre-processing
3. Exploratory Analysis
4. Model Evaluation
5. Final Validation
6. Conclusion

2. Method and Data pre-processing

For this report, the Early stage diabetes will be used and can be downloaded from the Machine Learning Repository.

```
#download database from UCI Machine Learning Repository
dl <- tempfile()
download.file(
```

```
 "https://archive.ics.uci.edu/ml/machine-learning-databases/00529/diabetes_data_upload.csv",
 dl)

#Assign csv database into the data
data <- read_csv(dl, col_types = "dfffffffffffffff")
colnames(data) <- make.names(colnames(data))
```

It is noticeable that the database is ordered by genre. This pre-ordering of the data will impact the outcome of the models. So first it is necessary to reorder the "database.

```
# this loop changes the order of the factors for which the first entry is "No"
no_ind <- which(data[1,]=="No")
for (i in no_ind) {
  data[,i] <- factor(data[[i]], levels = c("Yes","No"))
}
```

The Early stage diabetes risk prediction dataset is split into a training and a validation set (`diabetes` and `validation` respectively). Only the `diabetes` data set is used for model construction. The `validation` data set is used only for assessing the performance of the *final* model. `diabetes` is split into `train` and `test`. Various models are constructed using `train` and their performances are assessed using `test`. The best performing model is then retrained using `diabetes` and assessed using `validation`. This way, `validation` has no effect on which model is selected to be the final model.

`Validation` is 15% of the entire data set and `test` is 15% of `diabetes`. The reason 15% is used for testing and validating in this report is because the data set is quite small. Using 15% instead of 10% for example gives more data to assess the performance of the models.

```
# create a validation set - this is used to assess the final model
# diabetes data set is used for model training and selection
set.seed(4)
validation_index <- createDataPartition(data$class, times=1, p=0.15, list=FALSE)
validation <- data[validation_index,]
diabetes <- data[-validation_index,]
```

Before we start developing models, we will need to create train and test sets from diabetes. The train is used to construct various models and test is used to assess their performances. The best performing model will then be retrained using the diabetes data set and assessed using the validation data set.

```
set.seed(16)
test_index <- createDataPartition(diabetes$class, times=1, p=0.15, list=FALSE)
test <- diabetes[test_index,]
train <- diabetes[-test_index,]
```

3. Exploratory Analysis

Before start building the model, we need to understand the structure of the data, the distribution of ratings and the relationship of the predictors. This information will help build a better model.

The structure of the dataset `diabetes` is shown below. “Class” is the predictor variable - “positive” indicates the patient has diabetes. The features are made up of age, gender (biological sex) and a selection of conditions including obesity, alopecia and muscle stiffness. The data contains observations from 272 diabetic and 170 non-diabetic patients. Thus, the prevalence of the condition in the data set does not reflect true prevalence, since less than 10% people are estimated to be diabetic.

```
## tibble [442 x 17] (S3: tbl_df/tbl/data.frame)
##  $ Age               : num [1:442] 40 58 41 45 60 55 57 66 67 70 ...
##  $ Gender            : Factor w/ 2 levels "Male","Female": 1 1 1 1 1 1 1 1 1 1 ...
##  $ Polyuria          : Factor w/ 2 levels "Yes","No": 2 2 1 2 1 1 1 1 1 2 ...
##  $ Polydipsia        : Factor w/ 2 levels "Yes","No": 1 2 2 2 1 1 1 1 1 1 ...
##  $ sudden.weight.loss: Factor w/ 2 levels "Yes","No": 2 2 2 1 1 2 2 1 2 1 ...
##  $ weakness          : Factor w/ 2 levels "Yes","No": 1 1 1 1 1 1 1 1 1 1 ...
##  $ Polyphagia        : Factor w/ 2 levels "Yes","No": 2 2 1 1 1 1 1 2 1 1 ...
##  $ Genital.thrush    : Factor w/ 2 levels "Yes","No": 2 2 2 1 2 2 1 2 1 2 ...
##  $ visual.blurring   : Factor w/ 2 levels "Yes","No": 2 1 2 2 1 1 2 1 2 1 ...
##  $ Itching           : Factor w/ 2 levels "Yes","No": 1 2 1 1 1 1 2 1 1 1 ...
##  $ Irritability      : Factor w/ 2 levels "Yes","No": 2 2 2 2 1 2 2 1 1 1 ...
##  $ delayed.healing   : Factor w/ 2 levels "Yes","No": 1 2 1 1 1 1 1 2 2 2 ...
##  $ partial.paresis   : Factor w/ 2 levels "Yes","No": 2 1 2 2 1 2 1 1 1 2 ...
##  $ muscle.stiffness  : Factor w/ 2 levels "Yes","No": 1 2 1 2 1 1 2 1 1 2 ...
##  $ Alopecia          : Factor w/ 2 levels "Yes","No": 1 1 1 2 1 1 2 2 2 1 ...
##  $ Obesity           : Factor w/ 2 levels "Yes","No": 1 2 2 2 1 1 2 2 1 2 ...
##  $ class             : Factor w/ 2 levels "Positive","Negative": 1 1 1 1 1 1 1 1 1 1 ...
```

Figure 1 below shows a correlation plot of the features in the data set. Polydipsia and polyuria have the greatest correlation, which doesn't come as a surprise. The non-significant correlations are left blank. Although some of the features have high correlations, no dimensional reduction takes place in this report. This is primarily because the data set is quite small so it isn't necessary. Another reason is that [Approach 3: Decision Tree] is much more appealing with interpretable features.

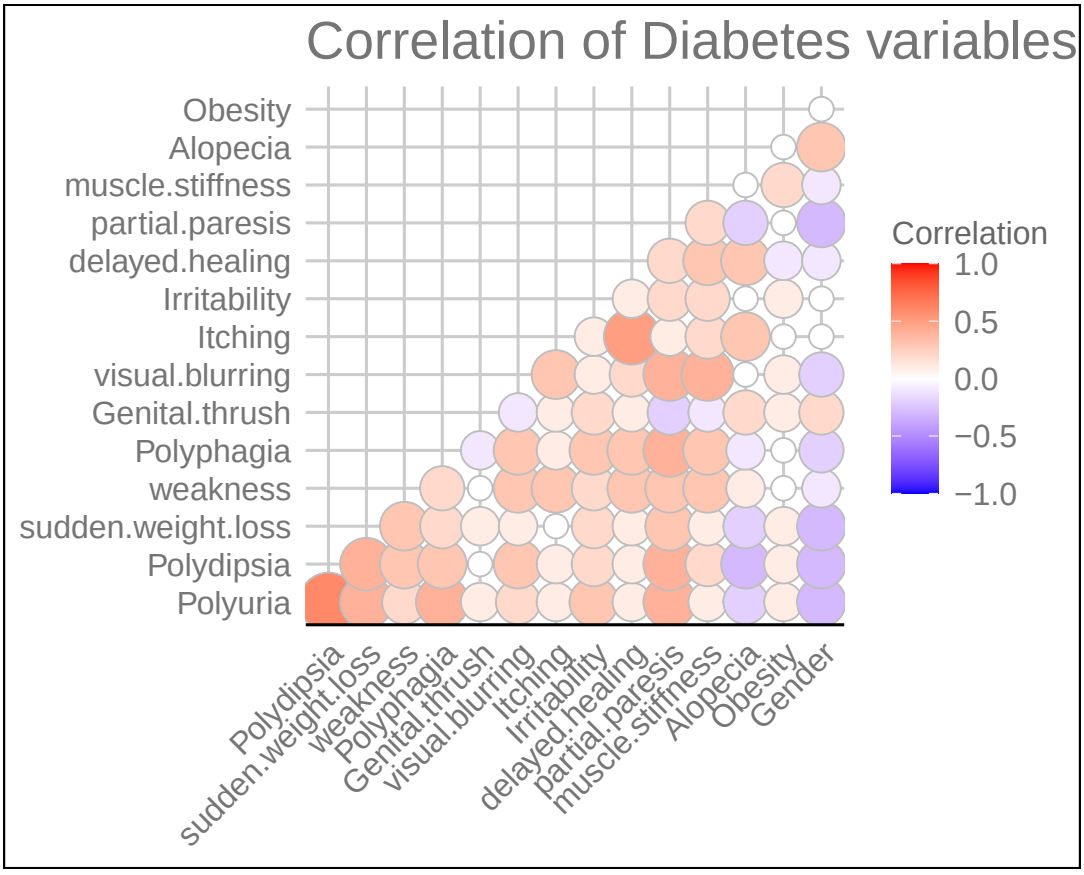

Figure 1 - Correlation plot of features in the diabetes data set.

This data set suggests that diabetes is more prevalent in females than is it in males, as illustrated in Figure 2. This is perhaps not in line with expectations, as research suggests that men are more likely to develop diabetes than women [3]. This is a reminder that data does not always accurately represent the population it was sampled from. The data set being worked with in this report only accounts for patients from one hospital in Bangladesh, so it would not be wise to make conclusions about diabetes on a world-wide scale.

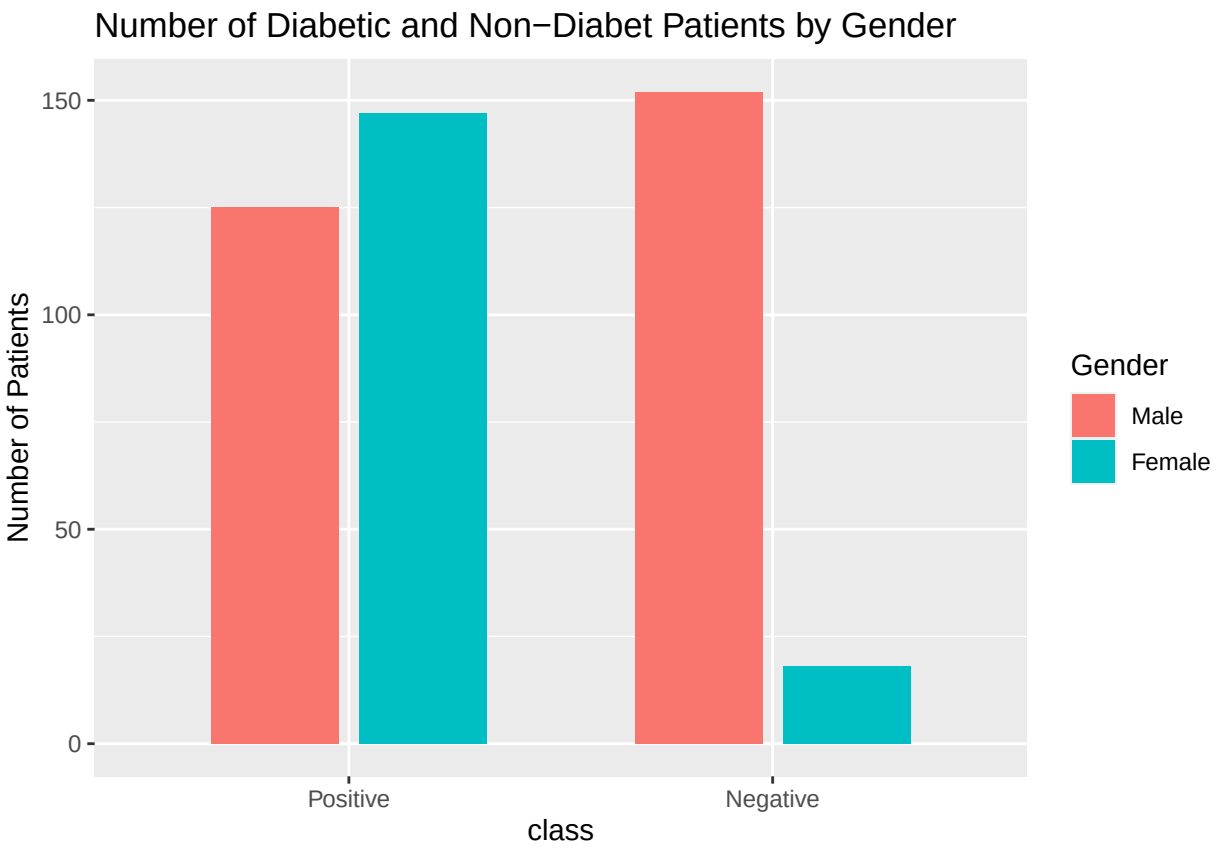

Figure 2 - Distribution of class by gender.

The distribution for age for positive and negative classes looks reasonably similar. Figure 3 below indicates that the spread for positive classes may be larger, however for the most part there doesn't appear to be a significant difference.

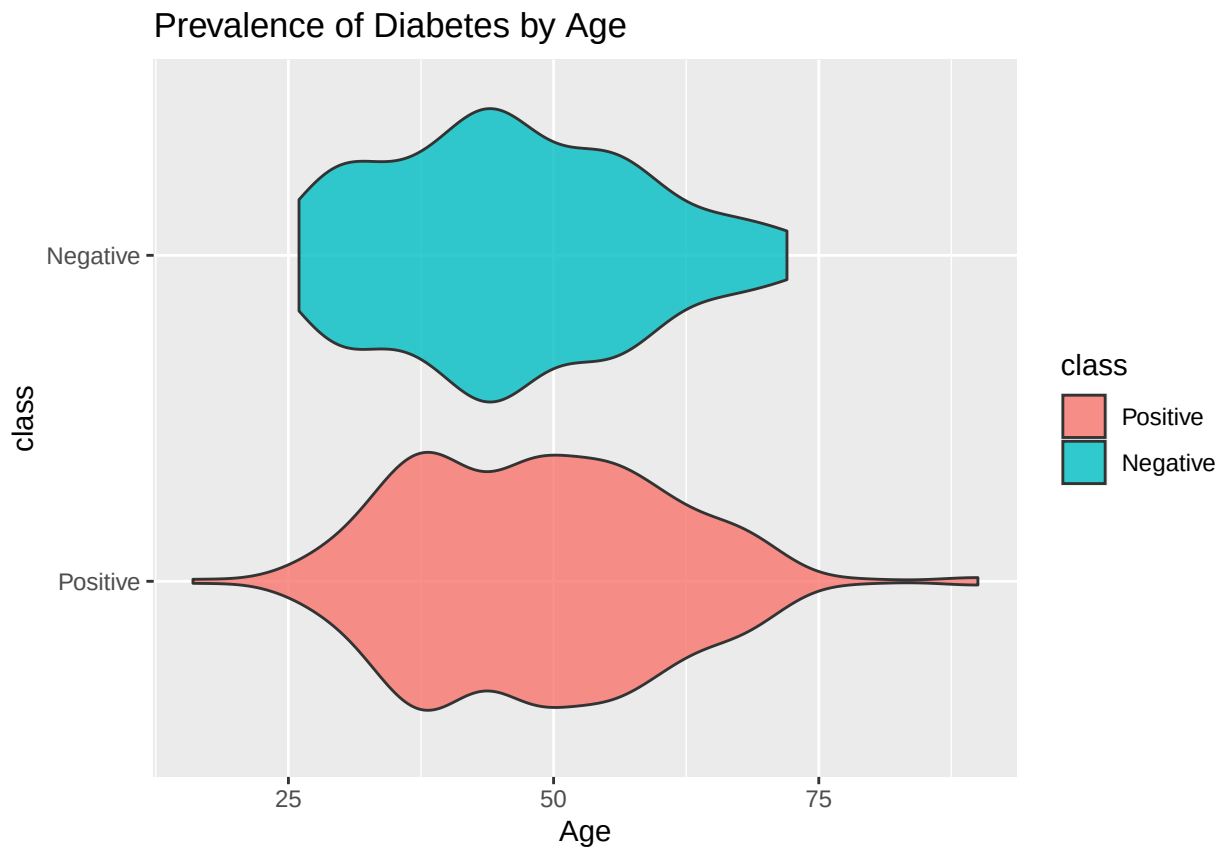

Figure 3: Distribution of age by class.

Further exploration can be carried out to discover properties of different features. Figure 4 shows the prevalence of diabetes by polydipsia and polyuria. It appears that if a patient has both polydipsia and polyuria then they are very likely to be diabetic. Otherwise, no confident conclusions could be drawn.

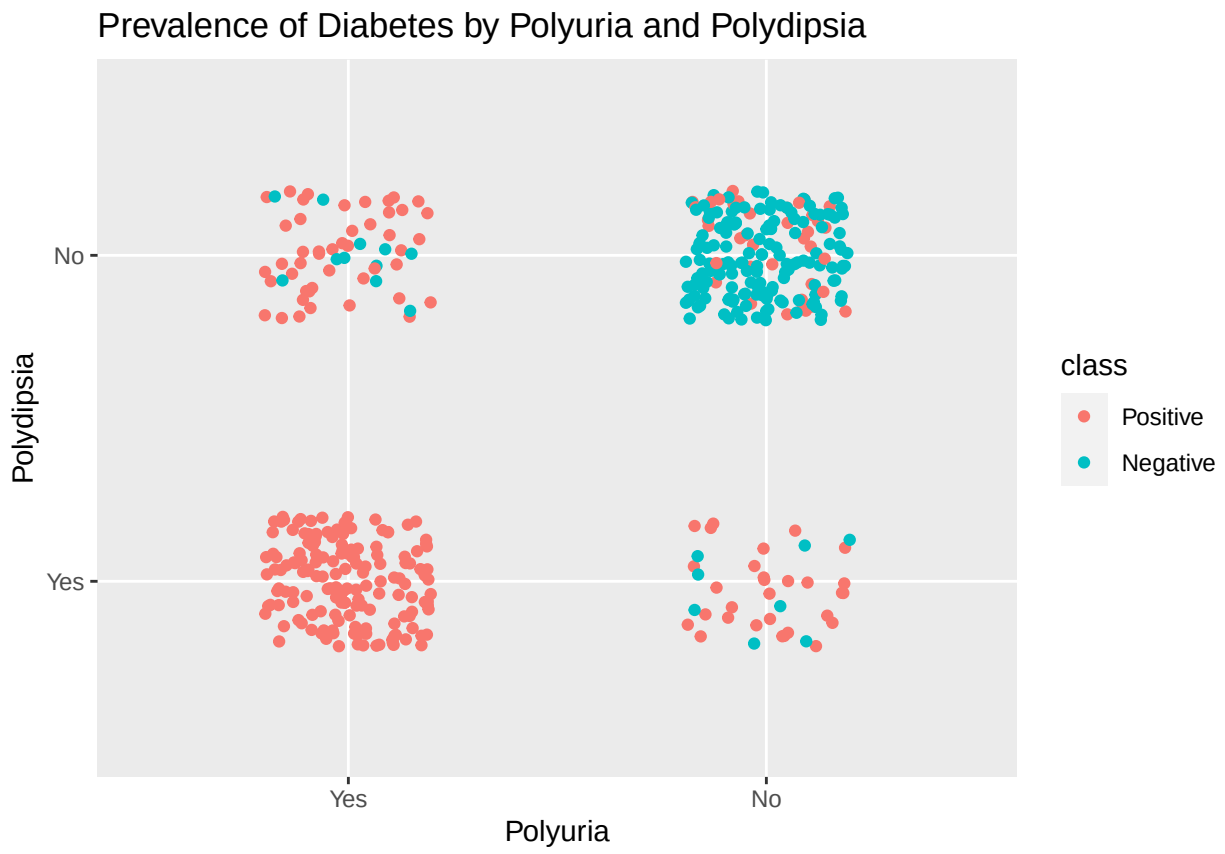

Figure 4 - Prevalence of diabetes by polydipsia and polyuria.

Figure 5 suggests that weakness isn't a condition which is useful in predicting whether a patient is diabetic or not. However, sudden weight loss could be an indication that a patient is diabetic.

Figure 5 - Prevalence of diabetes by weakness and sudden wight loss.

Figure 6 is particularly unusual. It suggests that diabetes might not be dependent on obesity. There are numerous studies to suggest that obesity is a significant cause of diabetes, so it is likely that the data set doesn't represent the world population very well.

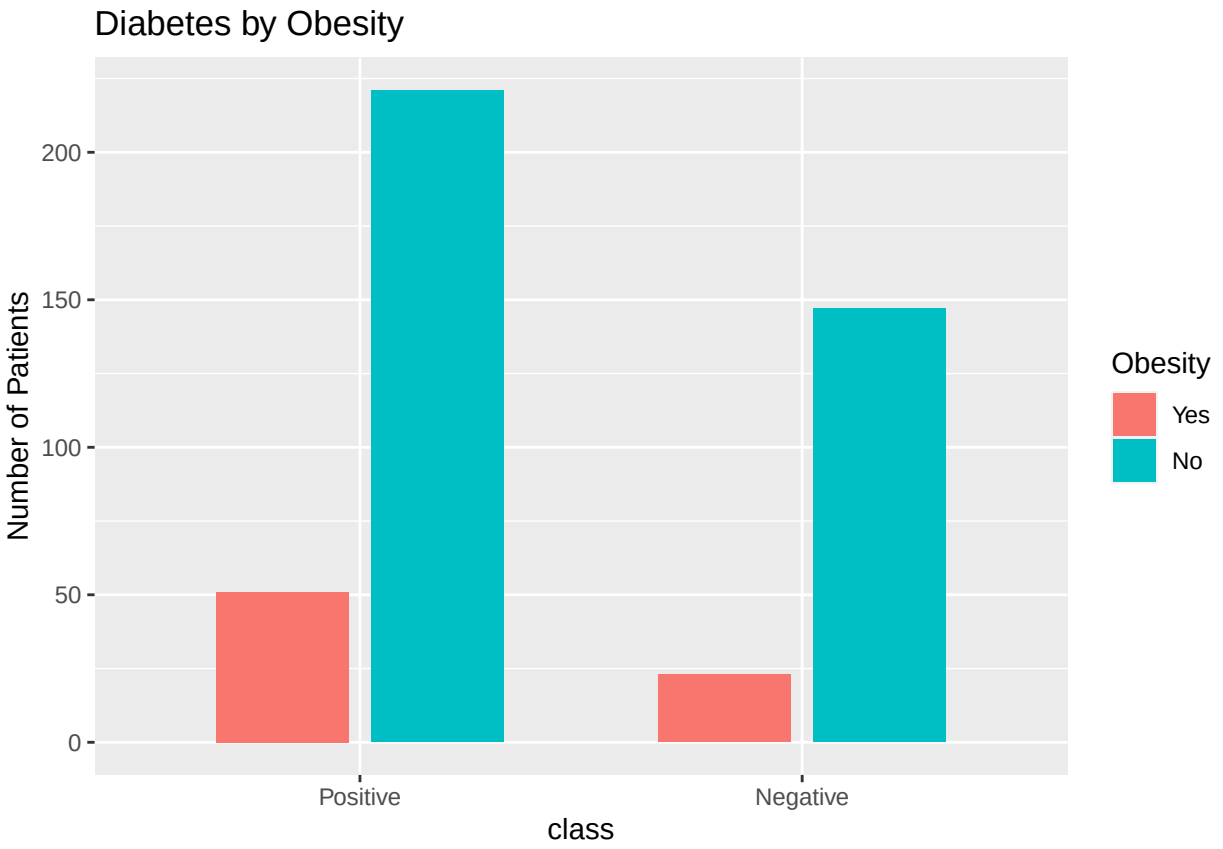

Figure 6 - Distribution of class by obesity.

4. Model Evaluation

We will construct a total of 5 models. [Model 1: Logistic Regression] constricts a logistic regression model. [Model 2: k-Nearest Neighbors] Constructs a k-nearest neighbours model. [Model 3: Decision Tree] constructs a decision tree. This method is in line with how people may expect doctors to make decisions in reality. [Model 4: Random Forest] is an extension on [Model 3: Decision Tree]. [Model 5: Ensemble] constructs an ensemble of the three best performing models. [Final Model (Results)] retrains the best performing model on a slightly larger data set and assesses its performance using a validation set which is not used for model construction or selection at any point in this report.

4.1: Logistic Regression

In this part of the model development, a logistic regression model will be built. The reason logistic regression is used instead of linear regression is that class is a binary variable. Therefore, it is appropriate for a model to predict the probability that the class of a patient is positive, for example.

The general form of a logistic regression model is

$$\log\left(\frac{\hat{\pi}_i}{1-\hat{\pi}_i}\right) = \mathbf{x}_i^T\beta \tag{1}$$

where $\hat{\pi}_i$ is the estimated probability that observation i is positive, $\mathbf{x}_i$ is the i^{th} vector in the design matrix and β is the vector of coefficients. In this case, the first element of $\mathbf{x}_i$ is 1 to activate the intercept in β, the second element of $\mathbf{x}_i$ is the age of observation i, and the rest of the elements are 1-0 dummy variables. For instance, the fourth element of $\mathbf{x}_i$ is 1 if observation i does **not** have polyuria, and is 0 otherwise. This is clear when looking at the summary of the final model towards the end of this section.

Classification models have various measures of performance. One is accuracy, which is the proportion of correctly classified patients. Sensitivity is the proportion of diabetic patients who are correctly classified. High sensitivity implies that a model is likely to correctly classify a diabetic patient. However, this can come at a cost of having low Specificity. Specificity is the proportion of non-diabetic patients that are correctly classified.

One choice that has to be made when constructing a logistic regression model is what cutoff to use. The cutoff p is such that $\hat{\pi}_i > p \Rightarrow$ observation i is classed as positive. A typical choice is 0.5 which will be used in this context.

The summary of the model, trained on `train`, indicates that around half of the features are statistically significant. Non-significant features include obesity, visual blurring and sudden weight loss. Statistically significant features include gender, polyuria and polydipsia.

Below the summary is the confusion matrix (tested on `test`).

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction Positive Negative
##   Positive       39        2
##   Negative        2       24
##
##                Accuracy : 0.9403
##                  95% CI : (0.8541, 0.9835)
##     No Information Rate : 0.6119
##     P-Value [Acc > NIR] : 7.023e-10
##
##                   Kappa : 0.8743
##
##  Mcnemar's Test P-Value : 1
##
##             Sensitivity : 0.9512
##             Specificity : 0.9231
##          Pos Pred Value : 0.9512
##          Neg Pred Value : 0.9231
##              Prevalence : 0.6119
##          Detection Rate : 0.5821
##    Detection Prevalence : 0.6119
##       Balanced Accuracy : 0.9371
##
##        'Positive' Class : Positive
##
```

We are going to save and present the results of this and the next models on the table below.We chose sensitivity in addition to accuracy given the importance of sensitivity i.e. performance of the model to correctly classify a diabetic patient.

Method	Accuracy	Sensitivity
Model (1) Logistical Regression	0.9402985	0.9512195

Table 1: Results after construction of the Logistical Regression model.

4.2: k-Nearest Neighbours

The second approach is to construct a k-nearest neighbours model.In principle we want to pick the k that maximizes accuracy. The goal of cross validation is to estimate these quantities for any given algorithm and set of tuning parameters such as k.

In the model development, a 8-fold cross-validation is used to select the k that will generate the optimal accuracy. The code will use fold from 2 to 8 meaning a total of 7 folds tested. Additionally, we will use the tuneGrid parameter in order to try out 17 values between 2 and 19 neighbours. To do this with caret, we needed to define a column named k, so we use this: data.frame(k = seq(2, 19, 1)). It is known that the value of k as a predictor can be defined as the square root of the number of records in the dataset. In this case, the train dataset has 375 records, and thus k was defined as 19.

That said, the k-nearest neighbors model will be trained 7*18 = 126 times. Given our train dataset is small, this does not require more than a few seconds of processing. However in the case of larger datasets, it would be prudent to run the simulations in a smaller part of the dataset first in order to determine the parameters.

The results are shown in Figure 7 below, highlighting the optimal value of $k = 2$.

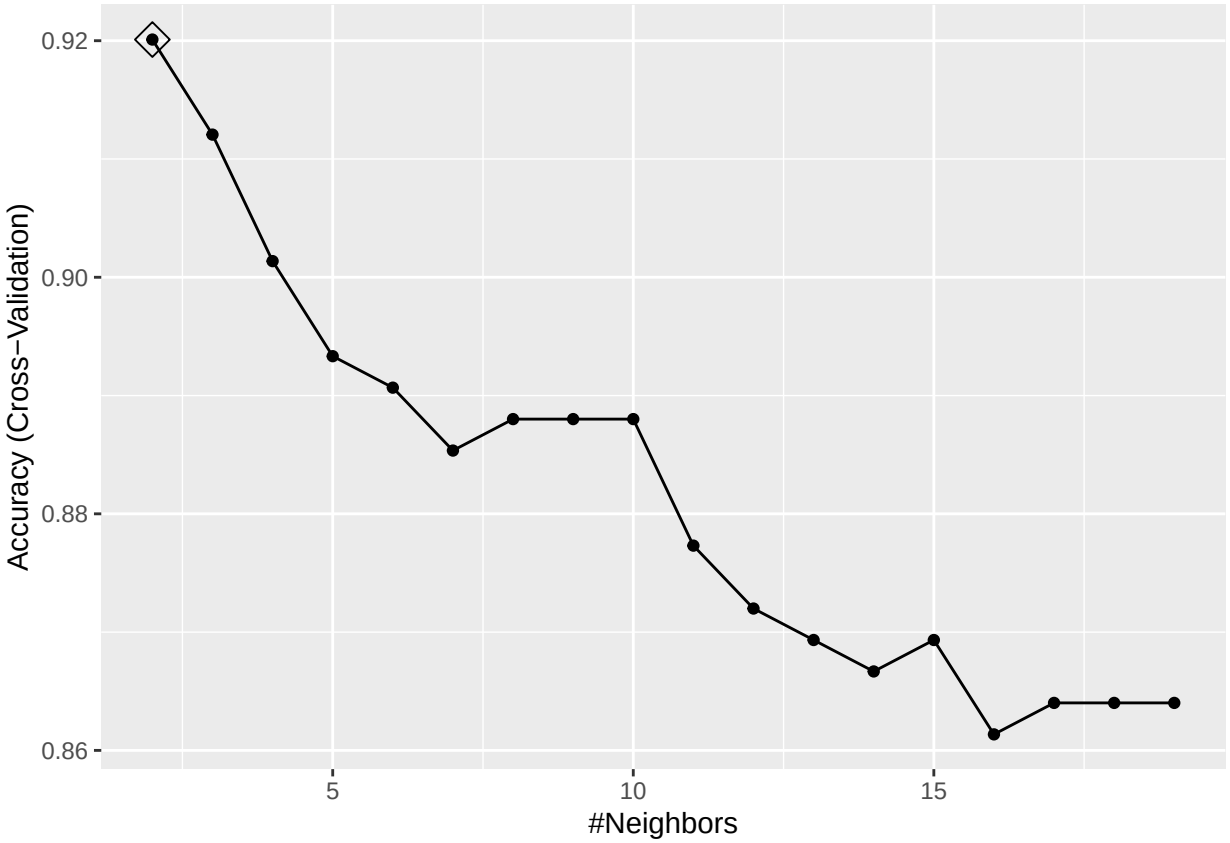

Figure 7 - Cross-validation results for kNN model. Optimal k is 1.

Following cross-validation, the `train` data set is used to construct a kNN model using $k = 2$. The confusion matrix is shown below.

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction Positive Negative
##   Positive       38        1
##   Negative        3       25
##
##                Accuracy : 0.9403
##                  95% CI : (0.8541, 0.9835)
##     No Information Rate : 0.6119
##     P-Value [Acc > NIR] : 7.023e-10
##
##                   Kappa : 0.876
##
##  Mcnemar's Test P-Value : 0.6171
##
##             Sensitivity : 0.9268
##             Specificity : 0.9615
##          Pos Pred Value : 0.9744
##          Neg Pred Value : 0.8929
##              Prevalence : 0.6119
##          Detection Rate : 0.5672
##    Detection Prevalence : 0.5821
##       Balanced Accuracy : 0.9442
##
##        'Positive' Class : Positive
##
```

Table 2 indicates that the accuracy is the same as the logistic regression model, however the sensitivity is slightly lower. That said, sensitivity is important when predicting diabetes as we want to reduce false negatives i.e. people with diabetes testing negative. With that consideration, logistic is a better model because it presented higher sensitivity.

Method	Accuracy	Sensitivity
Model (1) Logistical Regression	0.9402985	0.9512195
Model (2) Knn neighrbours	0.9402985	0.9268293

Table 2: Results after construction of the Knn model.

4.3: Decision Tree

This section constructs a decision tree. One advantage of decision trees is that they are highly interpretable. Even more so than linear models. The way in which decision trees make classifications is in line with how many people would expect physicians to predict the class of a potentially diabetic patient.

The rpart package is used to construct the decision tree. However, before the model is constructed, an optimal complexity parameter is chosen (the factor by which the models performance needs to improve by to warrant another split). Bootstrap (25 samples of 25% of the data set) is used to select the optimal complexity parameter. This is the default approach taken by the train function in the caret package. The default minsplit of 20 and minbucket of 7 are used.

Like the logistic regression model, this decision tree returns probabilities, not classes. Again, some cutoff p is chosen such that $\hat{\pi}_i > p \Rightarrow$ observation i is classed as being diabetic. Figure 8 shows the results from 5-fold cross-validation, highlighting the optimal value of 0.046.

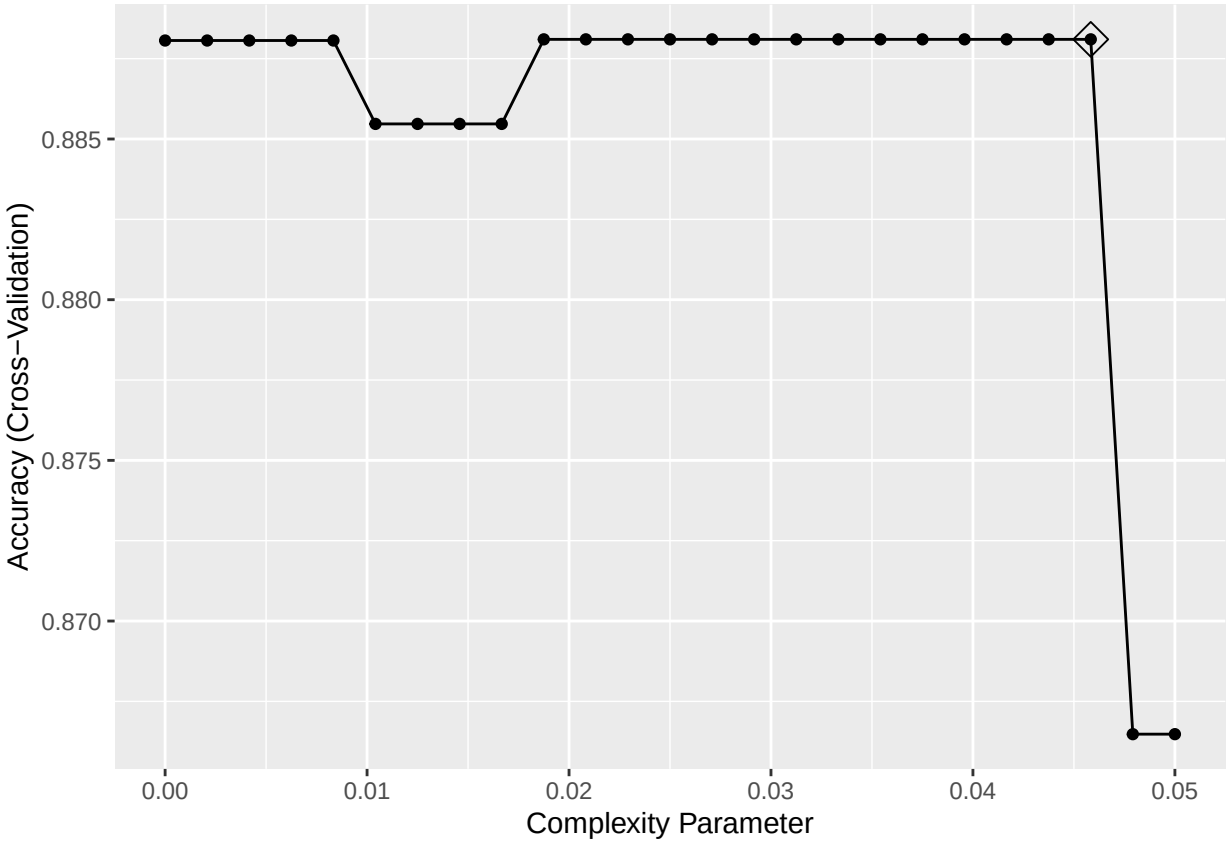

Figure 8 - Bootstrap (25 samples of 25% of the data) results. Optimal cp is 0.046

Figure 9 illustrates exactly how the tree makes decisions. The root node makes the first split based on whether the patient has polyuria or not. If they do, they are classed as being diabetic. If not, a further split is made based on the patient's gender, and so on. The percentage at the bottom of each leaf is the proportion of observations in train that lie in that leaf. The decimal above the percentage is the proportion of observations in that leaf that are non-diabetic.

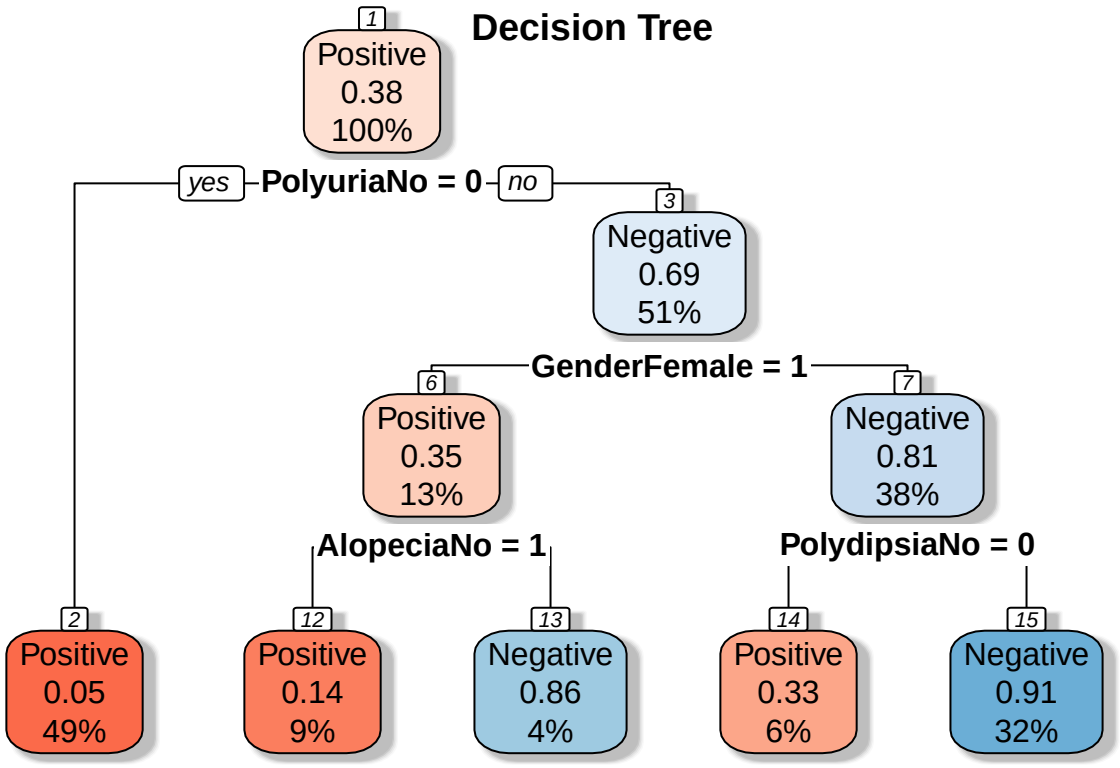

Figure 9 - Decision tree.

The decision tree confusion matrix is shown below.

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction Positive Negative
##   Positive       40        2
##   Negative        1       24
##
##                Accuracy : 0.9552
##                  95% CI : (0.8747, 0.9907)
##     No Information Rate : 0.6119
##     P-Value [Acc > NIR] : 6.736e-11
##
##                   Kappa : 0.9051
##
##  Mcnemar's Test P-Value : 1
##
##             Sensitivity : 0.9756
##             Specificity : 0.9231
##          Pos Pred Value : 0.9524
##          Neg Pred Value : 0.9600
##              Prevalence : 0.6119
##          Detection Rate : 0.5970
##    Detection Prevalence : 0.6269
```

```
##       Balanced Accuracy : 0.9493
##
##        'Positive' Class : Positive
##
```

Besides its high level of interpretability, Table 3 shows that the decision tree is the best performing model so far with higher accuracy and sensitivity. The next section, [Approach 4: Random Forest], expands on the idea of decision trees.

Method	Accuracy	Sensitivity
Model (1) Logistical Regression	0.9402985	0.9512195
Model (2) Knn neighrbours	0.9402985	0.9268293
Model (3) Decision tree	0.9552239	0.9756098

Table 3: Results after construction of the decision tree model.

4.4: Random Forest

This model is an extension of the decision tree - a random forest is a collection of decision trees. The way the random forest makes predictions is by some form of majority vote among all of the trees. Trees are constructed in a similar way as the previous section, however at each node a random subset of features is chosen to make the split.

This increases the independence between the trees, this parameter is `mtry` in the randomForest package [4]. Again, bootstrap (25 samples of 25%) is used to choose an optimal mtry. The results are shown below in Figure 10. The optimal value is 5. The randomForest package takes the default `nodesize` (minimum size of terminal nodes) to be 1 and the default `ntree` (number of decision trees in the forest) to be 500.

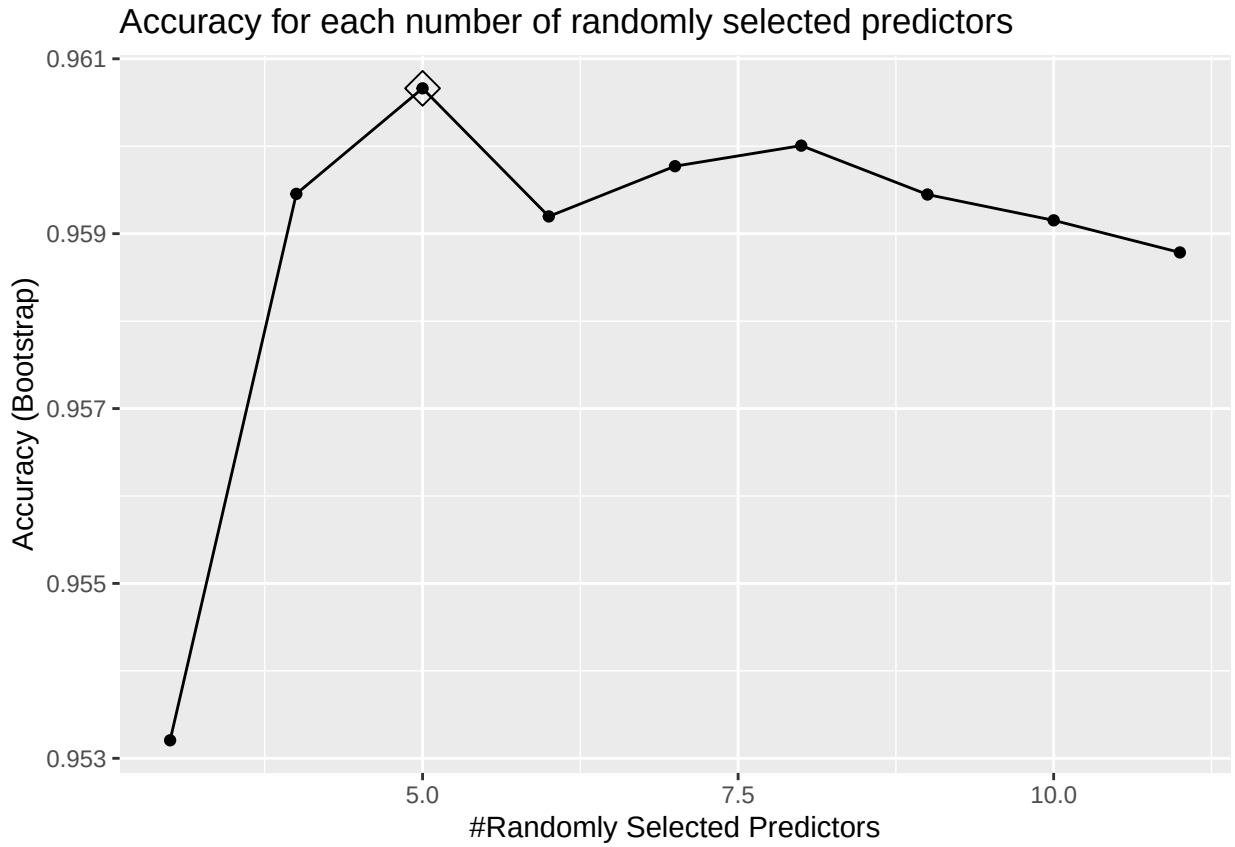

Figure 10: Bootstrap results for various values of mtry.

The confusion matrix below indicates that the random forest performs very well in comparison to the previous models.

The table below shows the importance of each variable is also accessible via the importance function. The attribute polyuria is a clear winner, meaning it is likely to be the root note in most of the decision trees in the forest.

```
##                       MeanDecreaseGini
## Age                          16.086266
## GenderFemale                 16.505554
## PolyuriaNo                   43.475844
## PolydipsiaNo                 30.504503
## sudden.weight.lossNo          7.907834
## weaknessNo                    3.312321
## PolyphagiaNo                  4.607518
## Genital.thrushNo              3.470212
## visual.blurringNo             4.337292
## ItchingNo                     5.455314
## IrritabilityNo                7.893192
## delayed.healingNo             5.942976
## partial.paresisNo             9.907556
## muscle.stiffnessNo            3.957003
## AlopeciaNo                    7.259748
## ObesityNo                     3.142203
```

Table 4 - Final random forest model attributes

The random forest model achieves an accuracy of 0.97 which is better than all prior models. the same is not observed for sensitivity which actually is the same as the decision tree model. Table 5 shows the performances of the first four models combined.

Method	Accuracy	Sensitivity
Model (1) Logistical Regression	0.9402985	0.9512195
Model (2) Knn neighrbours	0.9402985	0.9268293
Model (3) Decision tree	0.9552239	0.9756098
Model (4) Random Forest	0.9701493	0.9756098

`Table 5 - Results after construction of the Random forest model.`

4.5: Ensemble

The final model is an ensemble of the three best performing models.The decision tree is not considered as part of the ensemble model given the random forest is supposed to be a better version of a decision tree.

The ensemble takes a majority vote for each observation from the three models (logistic regression, kNN and random forest) and uses that as its prediction. By dropping one of the four models ties are avoided. Ensembling machine learning models is a great strategy to improve accuracy on test sets - it reduces the reliability on the performance of only one algorithm.

The confusion matrix below shows the accuracy of the model which is below than expected.

The result table below surprisly shows that the ensemble performs worse than the random forest. More on why this may be the case is discussed in the Conclusion.

Method	Accuracy	Sensitivity
Model (1) Logistical Regression	0.9402985	0.9512195
Model (2) Knn neighrbours	0.9402985	0.9268293
Model (3) Decision tree	0.9552239	0.9756098
Model (4) Random Forest	0.9701493	0.9756098
Model (5) Ensemble	0.9552239	0.9512195

`Table 6 - Results after construction of the Ensemble model.`

5. Final Validation

In the model development, the random forest achieves the best accuracy and sensitivity. Therefore, it is selected to be the final model.

The entire `diabetes` data set is now used to construct a random forest. Like before, bootstrap is used to select an optimal mtry. All other parameters remain unchanged. The results from the bootstrap are shown in Figure 14. The optimal mtry value is 7.

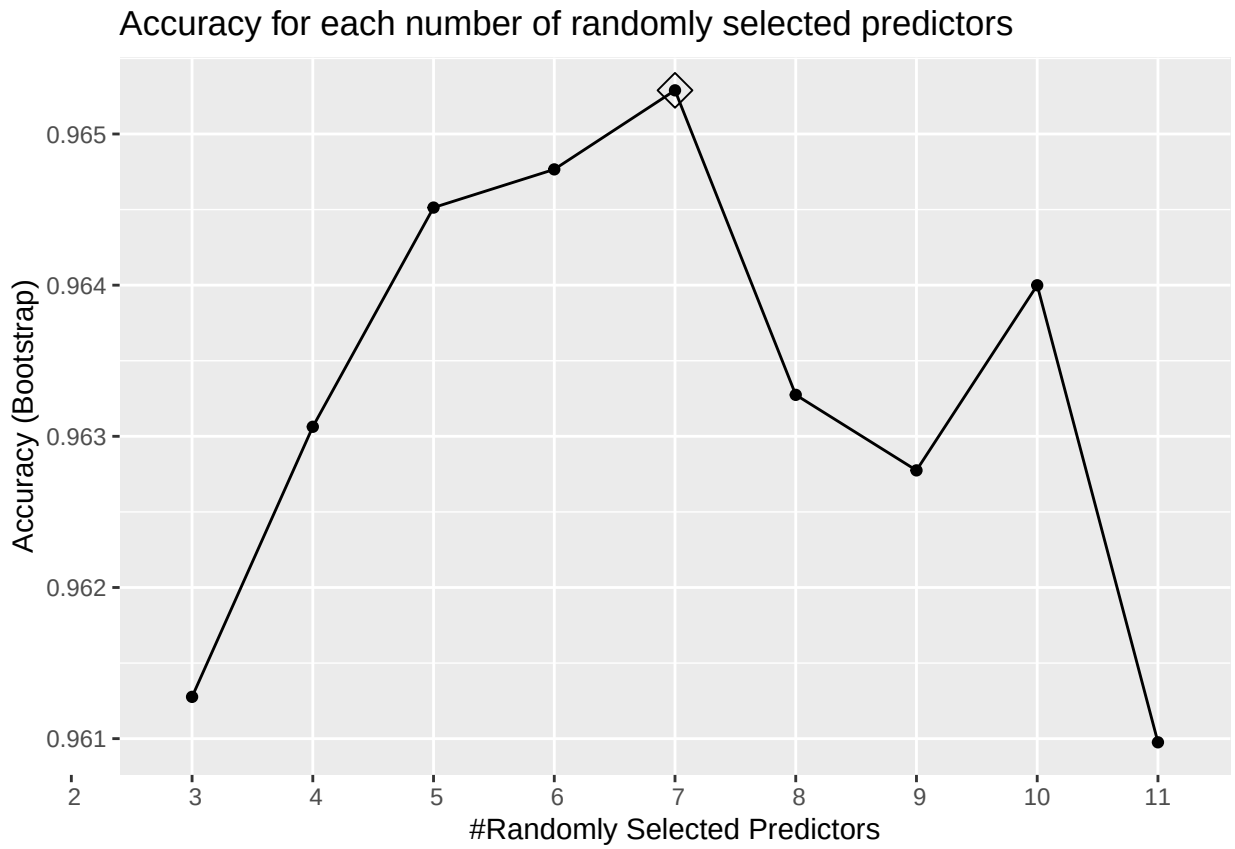

Figure 11: Bootstrap results for various values of mtry.

The confusion matrix indicates that the random forest achieves perfect accuracy, correctly identifying all patients in the `validation` data set. Although the algorithm couldn't have performed any better, the results should be interpreted with caution. More on this is discussed in the Conclusion.

Method	Accuracy	Sensitivity
Model (1) Logistical Regression	0.9402985	0.9512195
Model (2) Knn neighrbours	0.9402985	0.9268293
Model (3) Decision tree	0.9552239	0.9756098
Model (4) Random Forest	0.9701493	0.9756098
Model (5) Ensemble	0.9552239	0.9512195
Final Validation	0.9871795	0.9791667

`Table 7 - Results after Final validation using Diabetes.`

6. Conclusion

This report constructs a model using `diabetes` and predicts the class of each patient in `validation`. Though the model performs well with accuracy of 0.9871795 and sensitivity of 0.9791667, it is important to note conclusions regarding the size and source of the data utilized.

Future work

The `diabetes` data set contains 442 observations. The final model is only tested on 78 observations. The final model would be much more reliable if it was trained and tested on a larger data set. A project like this would benefit from having access to a larger number of observations.

The data is sampled from one hospital, Sylhet Diabetic Hospital (SDH). That said, it is necessary to utilize this model to predict the class of patients in other hospitals from other countries. A significant improvement on this report would be if the data set was sampled from various hospitals across the world. Thus, the final model would be useful on a global scale.

Using a larger data set taken from a global sample would give the model much more credibility, however it is almost certain that the estimated accuracy of the model would change and perhaps ensemble could prove to be a better model.

Additional considerations

The model is proven to be a great tool to predict diabetes. The diagnosis for diabetes could be largely aided with a simple questionnaire. A model trained on a more appropriate data set with a solid excellent performance, it would be a valuable diagnostic tool to be shared and utilized by doctors around the world. Granted that patients answer the questions accurately, the test could even be made available online. The results could indicate a percentage of a person being at risk, and if the risk was reasonably high (above 10% or so) the person could be advised to seek proper medical advice.

Diabetes can have a severe impact on many bodily functions. A machine learning model such as the final model in this report could help to detect diabetes at an early stage. This could prevent strokes, blindness and even amputations in many patients.

References

[1] UCI Machine Learning Repository *Early stage diabetes risk prediction dataset.* https://archive.ics.uci.edu/ml/datasets/Early+stage+diabetes+risk+prediction+dataset. (date last accessed - March 2020)

[2] Diabetes.co.uk *Diabetes Prevalence.* https://www.diabetes.co.uk/diabetes-prevalence.html (date last accessed - March 2020)

[3] Bazain/NHS *Men 'develop diabetes more easily'.* https://www.nhs.uk/news/diabetes/men-develop-diabetes-more-easily/ (date last accessed - March 2020)