Introduction to Machine Learning - CSC-532A

Understanding the Impact of Lifestyle Factors on Diabetes: A Machine Learning Approach

Submitted By:

Shashank M. Tripathi

Archita H. Patel

University Of Illinois at Springfield

Instructor:

Elham K. Buxton

Abstracts

The Diabetes Health Indicators Dataset is a valuable resource for forecasting the risk of developing diabetes and predicting its severity. The dataset includes various health indicators, such as blood pressure, BMI, and age, which can be used to create models that predict the likelihood of developing diabetes based on clinical data. These models can help healthcare professionals identify high-risk patients and create interventions to manage or treat the condition. By leveraging the information in the Diabetes Health Indicators Dataset, researchers and healthcare professionals can develop new strategies to prevent and manage diabetes, improving patient outcomes and reducing the burden of this disease on society. The data was collected through the Behavioral Risk Factor Surveillance System (BRFSS) survey conducted in 2015.

Problem Definition and Goals

The objective of this project is to predict the likelihood of diabetes based on various health indicators using machine learning algorithms. Accurately predicting diabetes status is essential in identifying at-risk individuals and developing effective intervention strategies. The dataset contains information from the BRFSS 2015 survey, including demographic and health information, such as age, sex, BMI, physical activity, and smoking status. The primary objective is to use the available features to predict the likelihood of an individual having diabetes or not.

Dataset Details

The Behavioral Risk Factor Surveillance System (BRFSS) survey from 2015 included data on health indicators and diabetes status in the file "diabetes_binary_BRFSS2015.csv"[1]. The dataset is split into two subsets, one for testing and the other for training machine learning models. Age, gender, and race/ethnicity demographic data, as well as health-related characteristics like smoking status, physical activity level, and body mass index (BMI), are all included in the dataset's variables. A binary indicator of diabetes status (having diabetes or not) serves as the goal variable. Using this dataset, machine learning models for predicting diabetes based on multiple health markers may be created and evaluated.

The variables are:

- **Diabetes:** a binary variable indicating whether the individual has been diagnosed with diabetes (1 for yes, 0 for no).
- **HighBP:** a binary variable indicating whether the individual has been diagnosed with high blood pressure (1 for yes, 0 for no).
- **HighChol:** a binary variable indicating whether the individual has been diagnosed with high cholesterol or not (1 for yes, 0 for no).
- **CholCheck**: a binary variable indicating whether the individual has had a cholesterol check in the past year or not (1 for yes, 0 for no).
- **BMI**: a numeric variable indicating the individual's body mass index.
- **Smoker**: a binary variable indicating whether the individual is a current smoker (1 for yes, 0 for no).
- **Stroke**: a binary variable indicating whether the individual has had a stroke (1 for yes, 0 for no).
- **HeartDiseaseorAttack**: a binary variable indicating whether the individual has had heart disease or a heart attack or not (1 for yes, 0 for no).
- **PhysActivity**: a binary variable indicating whether the individual engages in physical activity (1 for yes, 0 for no).
- **Fruits**: a binary variable indicating whether the individual consume fruits or not (1 for yes, 0 for no).
- **Veggies**: a binary variable indicating whether the individual consumes vegetables or not (1 for yes, 0 for no).
- **HvyAlcoholConsump**: a binary variable indicating whether the individual consume heavy amounts of alcohol or not (1 for yes, 0 for no).
- **AnyHealthcare**: a binary variable indicating whether the individual has received any healthcare in the past year (1 for yes, 0 for no).
- **NoDocbcCost**: a binary variable indicating whether the individual had to forego healthcare due to cost or not (1 for yes, 0 for no).
- **GenHlth**: a numeric variable indicating the individual's self-reported general health on a scale from 1 (poor) to 5 (excellent).
- **MentHlth**: a numeric variable indicating the number of days in the past month that the individual has felt mentally unwell.

- **PhysHlth**: a numeric variable indicating the number of days in the past month that the individual has felt physically unwell.
- **DiffWalk**: a binary variable indicating whether the individual has difficulty walking or not (1 for yes, 0 for no).
- **Sex**: a binary variable indicating the individual's sex (1 for male, 0 for female).
- **Age**: a numeric variable indicating the individual's age.
- **Education**: a numeric variable indicating the individual's level of education on a scale from 1 (less than high school) to 6 (graduate degree).
- **Income**: a numeric variable indicating the individual's income level on a scale from 1 (less than $10,000) to 8 (more than $75,000).

Related work

The goal of the article's analysis of a diabetes dataset is to determine a person's likelihood of having diabetes based on characteristics including age, blood glucose levels, blood pressure, and body mass index (BMI). The author does hypothesis testing to evaluate the significant differences between certain variables for diabetes and non-diabetic patients and then utilizes exploratory data analysis (EDA) to find patterns and outliers in the dataset. Several machine learning models, including logistic regression, decision trees, and random forests, are trained as part of the study to categorize a person as diabetic or non-diabetic based on their attributes. Based on evaluation measures like accuracy and F1 score, the highest-performing model is chosen for deployment. From the article, we learn about the process of analyzing a medical dataset using EDA, hypothesis testing, and machine learning techniques to predict whether a person has diabetes or not. Medical professionals and researchers interested in the study of diabetes can benefit from the author's thoughts and models.[2]

To determine if a patient has diabetes or not, the author is analyzing a dataset of diabetes patients. Exploratory data analysis (EDA) is used by the author to figure out the distribution of variables and correlations among them. To develop predictions based on the data, they then apply a variety of machine learning methods, such as decision trees and logistic regression. The analysis's purposes are to find which machine learning model performs the best and the most crucial features for predicting diabetes. The three most significant indicators of diabetes are age, body mass index, and blood glucose levels, according to this document. We also discover that the logistic regression model outperforms the other examined models. The author also makes some recommendations for future research, including gathering additional information on certain variables and investigating more advanced machine-learning strategies. The document offers information about how to perform an EDA and use machine learning algorithms for predictive analysis overall.[3]

Data Exploration and Preprocessing

The dataset "diabetes_binary_5050split_health_indicators_BRFSS2015.csv"[1] comprises of data from the 2015 Behavioral Risk Factor Surveillance System (BRFSS) survey on diabetes status and health markers. The dataset comprises 21 feature variables pertaining to demographic data, lifestyle characteristics, and medical history and has been preprocessed and cleaned for analysis. The goal variable is a binary diabetes status indicator equally represented in training and testing sets for positive and negative instances. Using this dataset, machine learning models for predicting diabetes based on multiple health markers may be created and evaluated. It is crucial to remember that there is no class imbalance in this dataset. Overall, this dataset offers a valuable resource for researching the link between health markers and diabetes and creating prediction models for detecting and treating diabetes.

Based on our analysis of the data, we have found that the distribution of the data is balanced, with each class representing 50% of the total data. This means that there is an equal representation of both classes in the data, and there is no class imbalance issue that may affect the performance of the machine learning algorithms. Having a balanced dataset is important as it ensures that the models are not biased towards any class and can make accurate predictions for both classes. Therefore, the balanced distribution of the data is a positive sign for any machine learning task that may be performed on it.

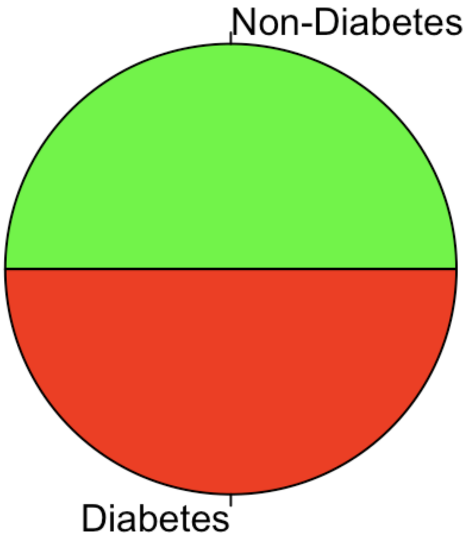


Fig 1.1

We performed a preprocessing step to ensure that our data is properly formatted and cleaned before training machine learning models. As a part of this step, we first checked the format of the dataset and ensured that it was in a suitable format for model training. We also converted all binary variables to categorical variables. his was done to make that the binary variables were treated as categorical variables by the machine learning algorithms and to make it easier to interpret the results.

By performing this preprocessing step, we have improved the quality of the data, which in turn can help to improve the performance of the machine learning algorithms. This step has also ensured that the data is in a suitable format for model training, and that the machine learning algorithms can handle the data appropriately. By using categorical variables, we can also avoid issues related to variable type mismatch or incorrect handling of binary variables during the model training process.

To further improve the quality of our dataset before training machine learning models, we performed additional preprocessing steps. Firstly, we checked for and removed any missing or NA values in the dataset. This was done to ensure that the data is complete and there are no gaps in the data that could affect the model training process.

Next, we conducted statistical tests and plotted the relationships between the diabetes variable and the other variables in the dataset. This allowed us to identify which variables were most strongly correlated with diabetes, and which variables were unrelated. Based on this analysis, we removed any variables that were found to be unrelated to diabetes from the dataset. This was done to reduce the number of variables in the dataset and to ensure that only the most relevant variables were used in the machine learning models.

Overall, these preprocessing steps have helped to ensure that our dataset is of high quality and suitable for model training. By removing missing values, identifying relevant variables, and cleaning the data, we have improved the accuracy and effectiveness of any machine learning models that may be trained on this data.

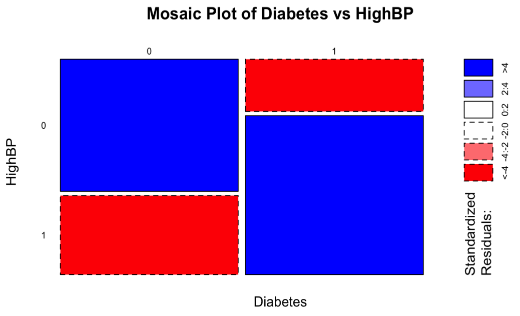


Fig 1.2 Diabetes/HighBP

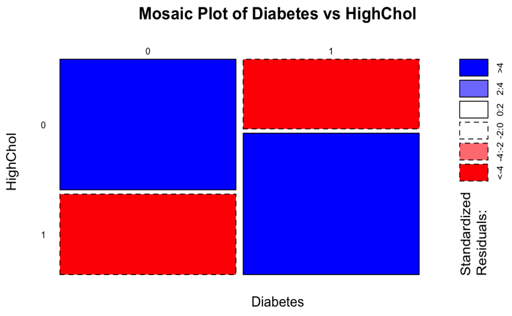


Fig 1.3 Diabetes/HighChol

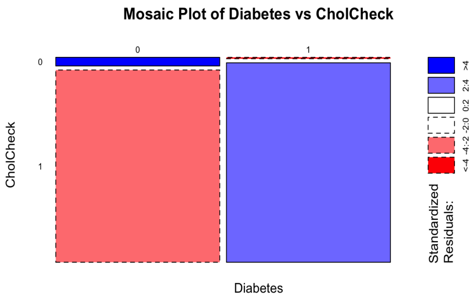


Fig 1.4 Diabetes/CholCheck

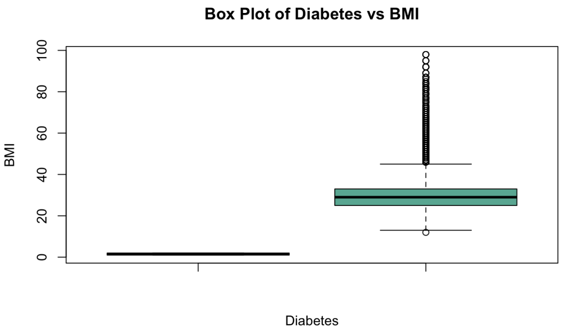


Fig 1.5 Diabetes/BMI

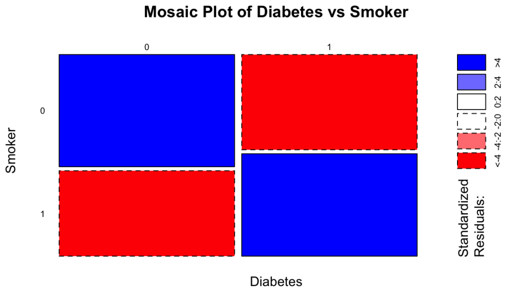


Fig 1.6 Diabetes/Smoker

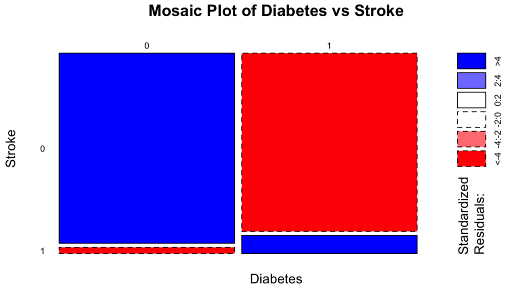


Fig 1.7 Diabetes/Stroke

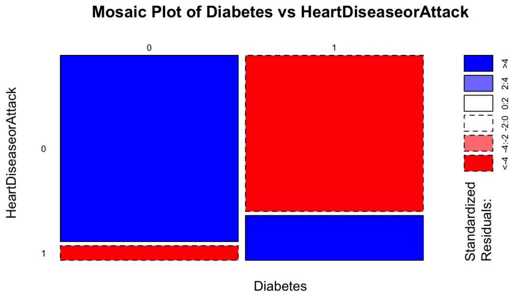


Fig 1.8 Diabetes/HeartDiseaseorAttack

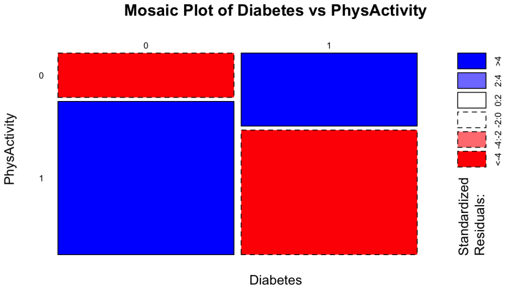


Fig 1.9 Diabetes/PhysActivity

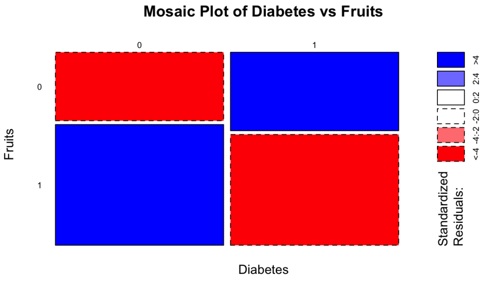


Fig 2.0 Diabetes/Fruits

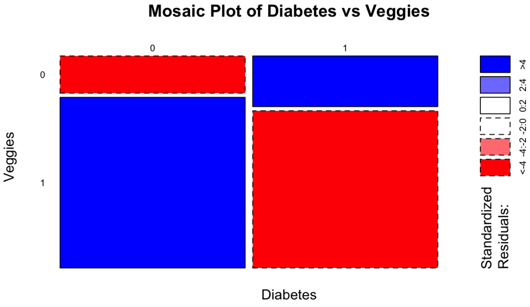


Fig 2.1 Diabetes/Veggies

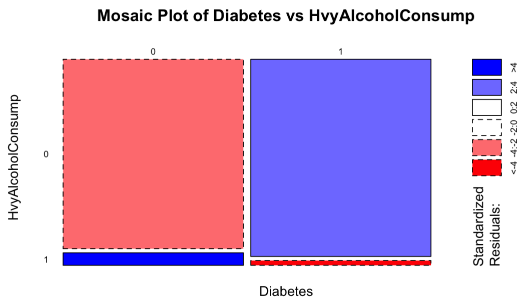


Fig 2.2 Diabetes/HvyAlcoholConsump

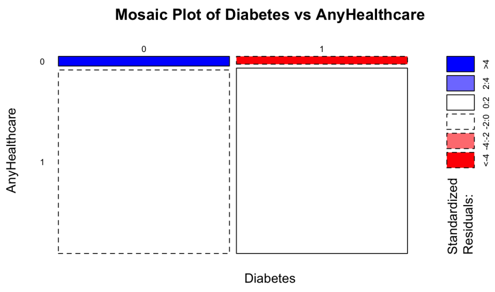


Fig 2.3 Diabetes/AnyHealthcare

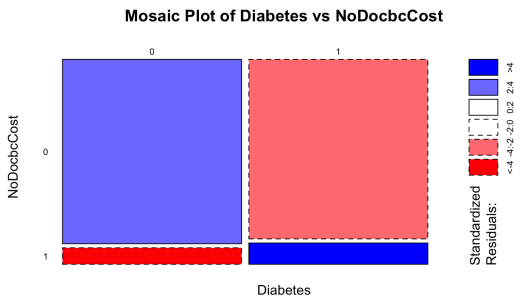


Fig 2.4 Diabetes/NoDocbcCost

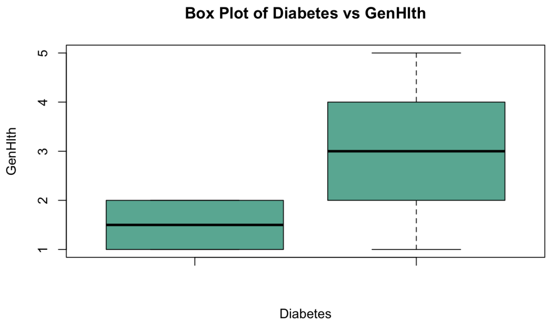


Fig 2.5 Diabetes/GenHlth

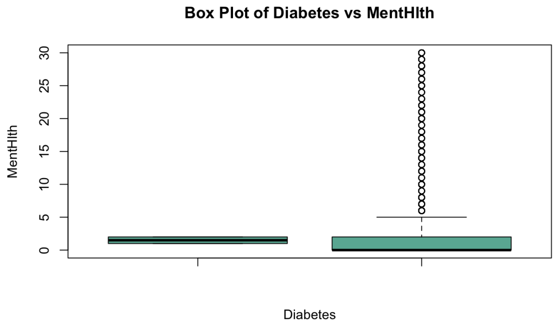


Fig 2.6 Diabetes/MentHlth

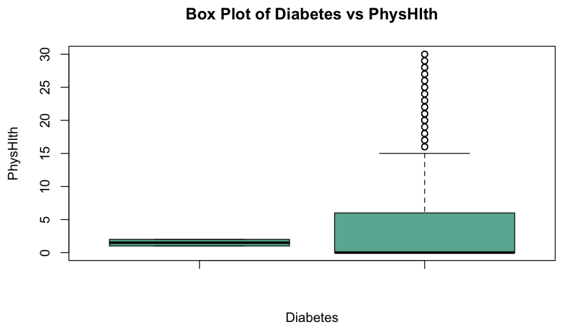


Fig 2.7 Diabetes/PhysHlth

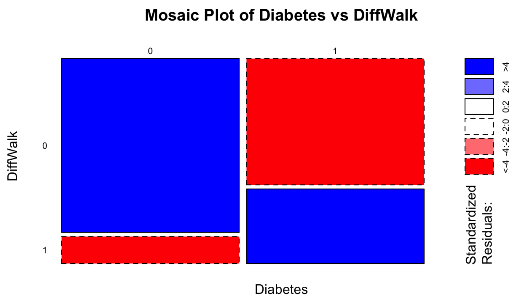


Fig 2.8 Diabetes/DiffWalk

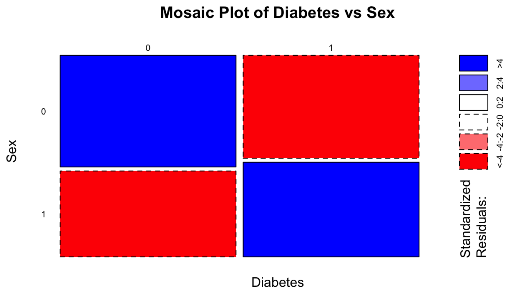


Fig 2.9 Diabetes/Sex

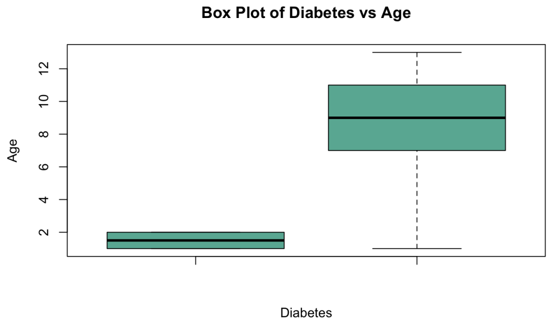


Fig 3.0 Diabetes/Age

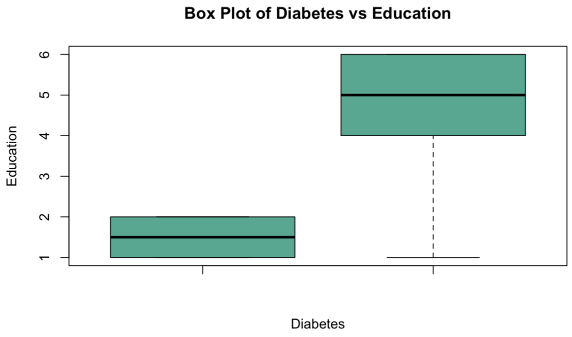


Fig 3.1 Diabetes/Education

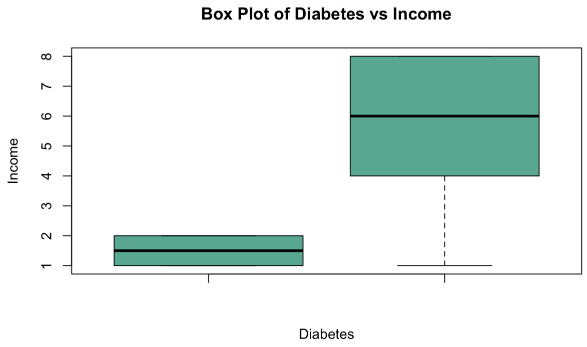


Fig 3.2 Diabetes/Income

The results of our statistical tests suggest strong correlations between the occurrence of diabetes and certain characteristics, such as high blood pressure, high blood sugar, heart disease or attack, physical activity, BMI, GenHlth, PhysHlth, DiffWalk, Age, Education, and Income. The p-values of these tests were all 0, indicating a highly significant association between these factors and the presence of diabetes.

In addition, we found significant associations between diabetes and other factors, such as MentHlth, CholCheck, Smoker, Fruits, Veggies, HvyAlcoholConsump, AnyHealthcare, and NoDocbcCost. The p-values for these tests were small, indicating a significant association between these factors and the presence of diabetes.

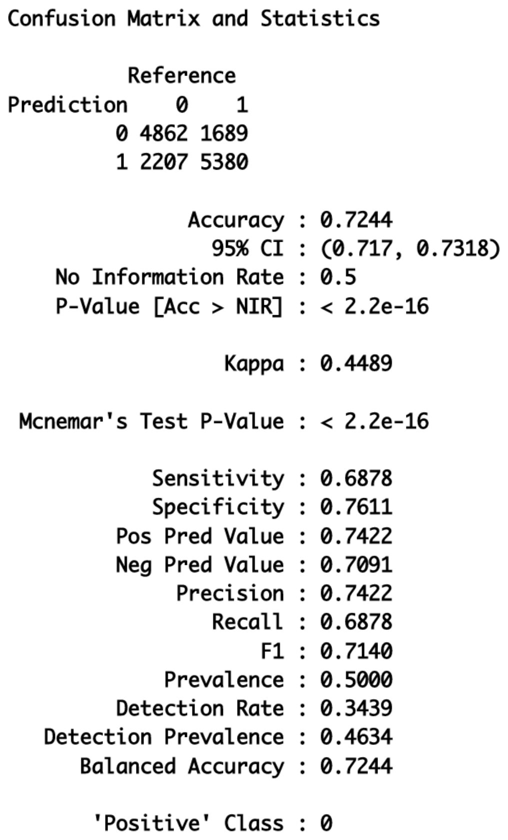
Data Analysis and Results

In this dataset analysis and prediction, we used the following models for data training and prediction.

- **KNN**
- **Lasso**
- **Ridge**
- **Elastic Net linear**
- **Random Forest**
- **Gradient Boosted Model (GBM)**
- **Support Vector Machine (SVM) Linear**
- **Support Vector Machine (SVM) Radial**
- **Neural Network**

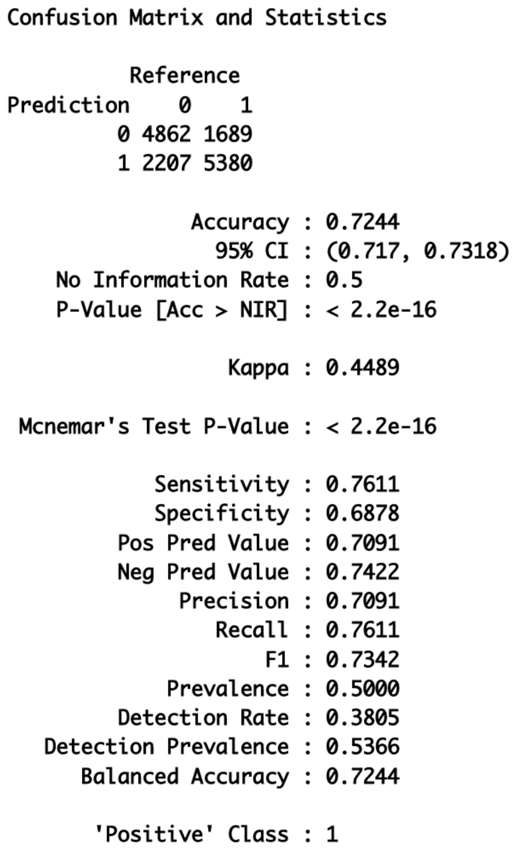
KNN Model:

To prepare our diabetes dataset for modeling, we partitioned it into training and test data. We then trained a KNN (K-Nearest Neighbors) model using the training data. The Calculated outcomes and confusion matrix is shown below.



```
Confusion Matrix and Statistics

          Reference
Prediction    0    1
         0 4862 1689
         1 2207 5380

               Accuracy : 0.7244
                 95% CI : (0.717, 0.7318)
    No Information Rate : 0.5
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.4489

 Mcnemar's Test P-Value : < 2.2e-16

            Sensitivity : 0.6878
            Specificity : 0.7611
         Pos Pred Value : 0.7422
         Neg Pred Value : 0.7091
              Precision : 0.7422
                 Recall : 0.6878
                     F1 : 0.7140
             Prevalence : 0.5000
         Detection Rate : 0.3439
   Detection Prevalence : 0.4634
      Balanced Accuracy : 0.7244

       'Positive' Class : 0
```

Fig 3.3



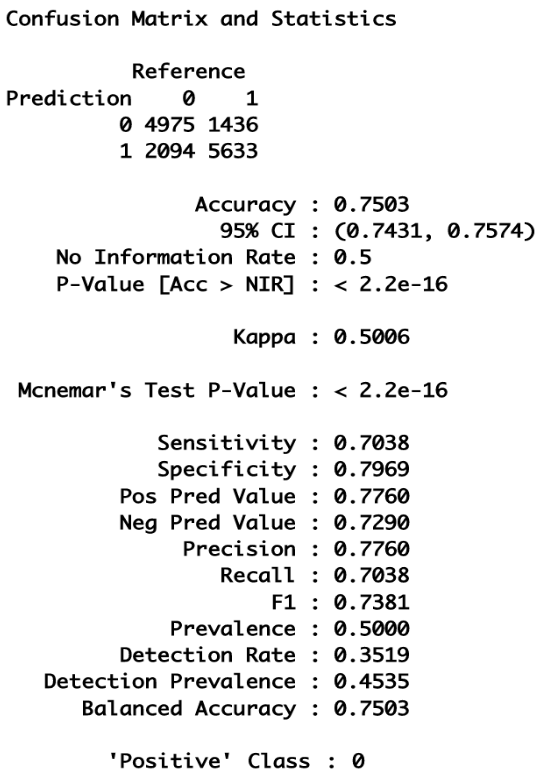
```
Confusion Matrix and Statistics

          Reference
Prediction    0    1
         0 4862 1689
         1 2207 5380

               Accuracy : 0.7244
                 95% CI : (0.717, 0.7318)
    No Information Rate : 0.5
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.4489

 Mcnemar's Test P-Value : < 2.2e-16

            Sensitivity : 0.7611
            Specificity : 0.6878
         Pos Pred Value : 0.7091
         Neg Pred Value : 0.7422
              Precision : 0.7091
                 Recall : 0.7611
                     F1 : 0.7342
             Prevalence : 0.5000
         Detection Rate : 0.3805
   Detection Prevalence : 0.5366
      Balanced Accuracy : 0.7244

       'Positive' Class : 1
```

Fig 3.4

The confusion matrix displays the outcomes of a two-class classification model (0 and 1). Class 0 was accurately predicted 4862 times by the model, but wrongly 2207 times. Similarly, the model successfully predicted class 1 5380 times while making wrong predictions 1689 times. With an accuracy of 0.7244, the model successfully predicted 72.44% of the observations. The actual and predicted classes exhibit a fair amount of agreement, as indicated by the Kappa value of 0.4489. With a sensitivity of 0.6878, the model successfully detected 68.78% of the positive cases. With a specificity of 0.7611, the model successfully detected 76.114% of the negative cases. The model's positive predictive value (PPV) is 0.7422, meaning that when it correctly predicts a positive case 74.22% of the time. The model's negative predictive value (NPV) is 0.7091, meaning that when it correctly predicts a negative situation 70.91% of the time. 50% of the observations fall into the positive class, according to the predominance of the positive class, which is 0.5. The detection rate is 0.3439, meaning that 34.39% of the positive cases were successfully detected by the model. The model correctly classified 46.34% of the data as belonging to the positive class, as indicated by the detection prevalence of 0.4634. Overall, the model's balanced accuracy, which is equal to the accuracy, is 0.7244. The accuracy that would be attained by consistently forecasting the most prevalent class or the no information rate, is higher than the model's performance.

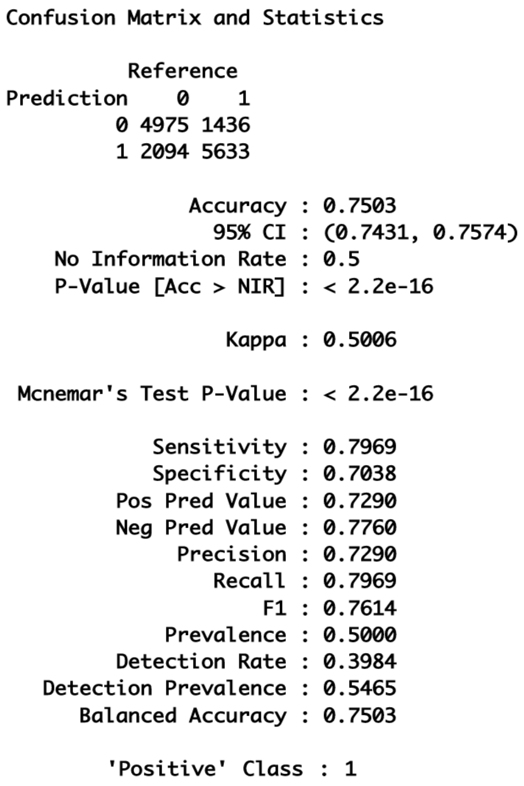
Random Forest Model:

The summary displays how well a random forest model performs for binary classification. A confusion matrix is used to compare the model's predictions to the actual class labels. The model's accuracy, which stands at 0.7503, means that it is accurate 75.03% of the time. For both classes (0 and 1), the precision, recall, and F1 scores are presented, showing how well the model performed for each class independently. Class 1 has higher recall than class 0, whereas class 0 has higher precision. The F1 results for both classes are generally respectable, with class 1 performing slightly better.



```
Confusion Matrix and Statistics

          Reference
Prediction    0    1
         0 4975 1436
         1 2094 5633

               Accuracy : 0.7503
                 95% CI : (0.7431, 0.7574)
    No Information Rate : 0.5
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.5006

 Mcnemar's Test P-Value : < 2.2e-16

            Sensitivity : 0.7038
            Specificity : 0.7969
         Pos Pred Value : 0.7760
         Neg Pred Value : 0.7290
              Precision : 0.7760
                 Recall : 0.7038
                     F1 : 0.7381
             Prevalence : 0.5000
         Detection Rate : 0.3519
   Detection Prevalence : 0.4535
      Balanced Accuracy : 0.7503

       'Positive' Class : 0
```

Fig 3.5

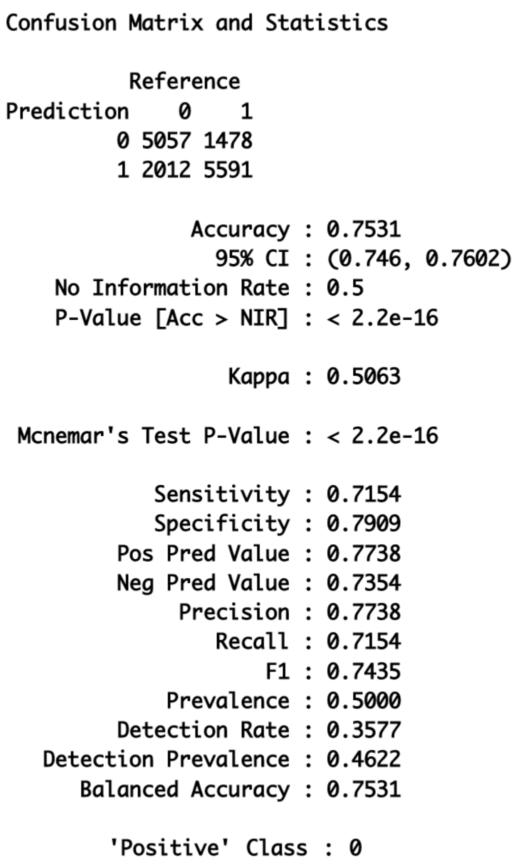


```
Confusion Matrix and Statistics

          Reference
Prediction    0    1
         0 4975 1436
         1 2094 5633

               Accuracy : 0.7503
                 95% CI : (0.7431, 0.7574)
    No Information Rate : 0.5
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.5006

 Mcnemar's Test P-Value : < 2.2e-16

            Sensitivity : 0.7969
            Specificity : 0.7038
         Pos Pred Value : 0.7290
         Neg Pred Value : 0.7760
              Precision : 0.7290
                 Recall : 0.7969
                     F1 : 0.7614
             Prevalence : 0.5000
         Detection Rate : 0.3984
   Detection Prevalence : 0.5465
      Balanced Accuracy : 0.7503

       'Positive' Class : 1
```

Fig 3.6

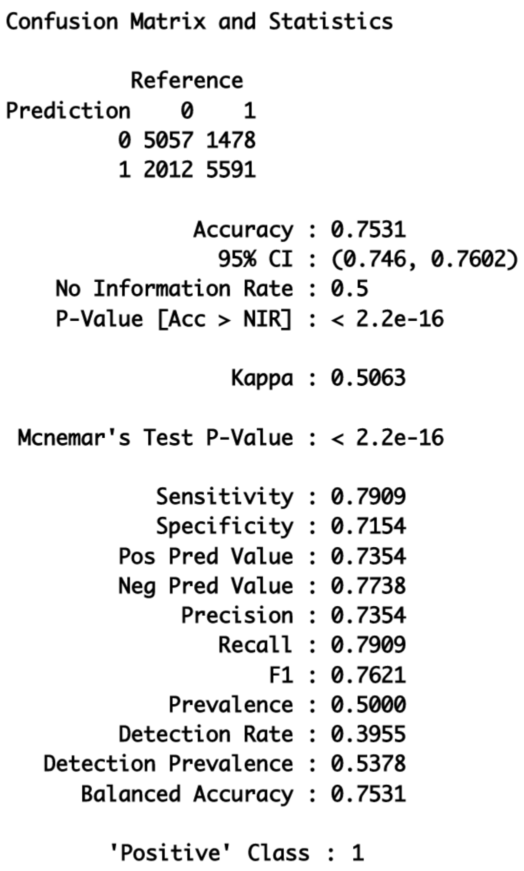
Gradient Boosted Model (GBM):

For a binary classification task, the given table displays the true class values and GBM predictions. The model's accuracy was 0.7531, meaning it accurately predicted 75.31% of the cases. Both classes' precision, recall, and F1 scores are given, with Class 1's somewhat greater recall and lower precision than class 0's.



```
Confusion Matrix and Statistics

          Reference
Prediction    0    1
         0 5057 1478
         1 2012 5591

               Accuracy : 0.7531
                 95% CI : (0.746, 0.7602)
    No Information Rate : 0.5
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.5063

 Mcnemar's Test P-Value : < 2.2e-16

            Sensitivity : 0.7154
            Specificity : 0.7909
         Pos Pred Value : 0.7738
         Neg Pred Value : 0.7354
              Precision : 0.7738
                 Recall : 0.7154
                     F1 : 0.7435
             Prevalence : 0.5000
         Detection Rate : 0.3577
   Detection Prevalence : 0.4622
      Balanced Accuracy : 0.7531

       'Positive' Class : 0
```

Fig 3.7

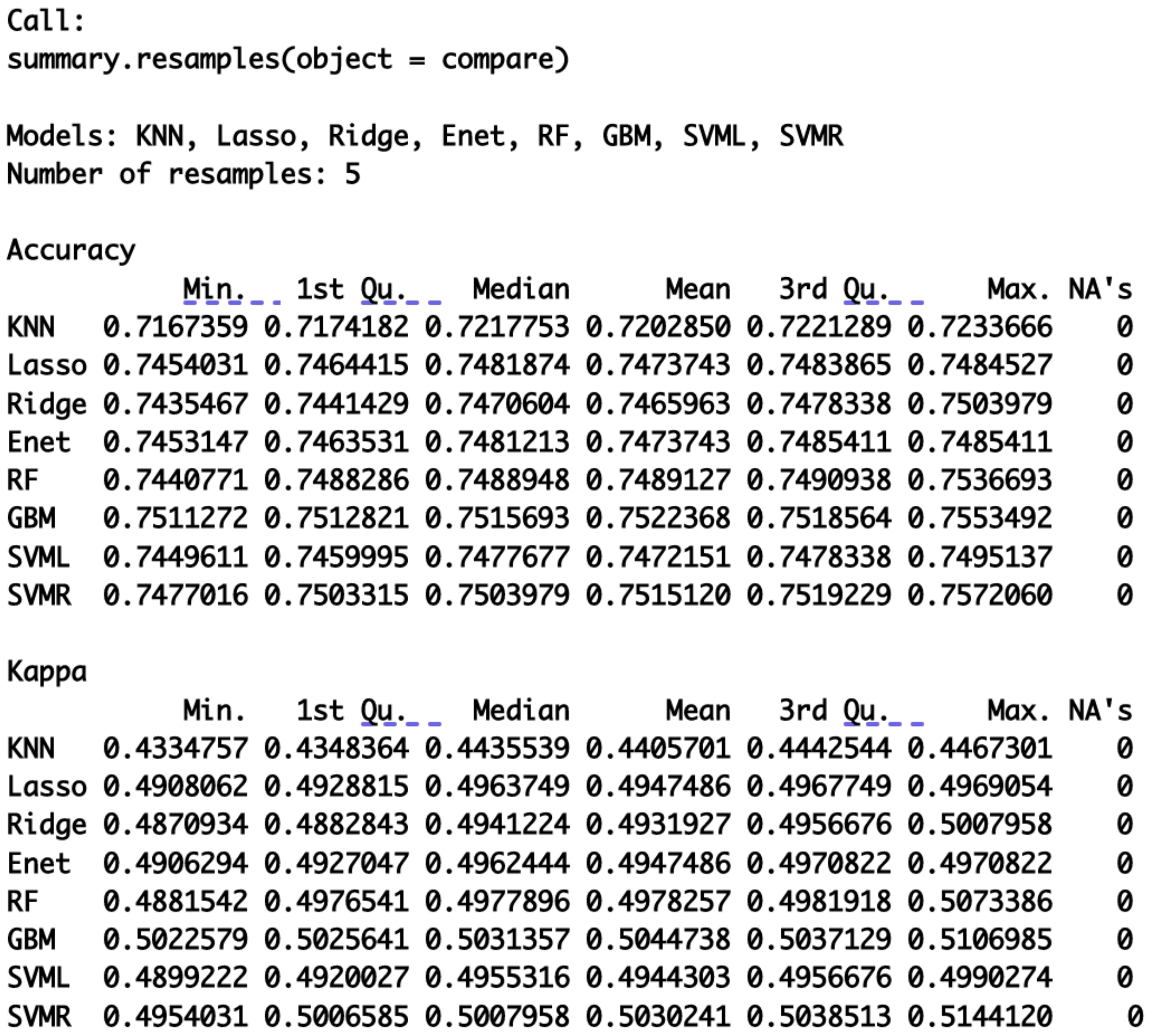


```
Confusion Matrix and Statistics

          Reference
Prediction    0    1
         0 5057 1478
         1 2012 5591

               Accuracy : 0.7531
                 95% CI : (0.746, 0.7602)
    No Information Rate : 0.5
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.5063

 Mcnemar's Test P-Value : < 2.2e-16

            Sensitivity : 0.7909
            Specificity : 0.7154
         Pos Pred Value : 0.7354
         Neg Pred Value : 0.7738
              Precision : 0.7354
                 Recall : 0.7909
                     F1 : 0.7621
             Prevalence : 0.5000
         Detection Rate : 0.3955
   Detection Prevalence : 0.5378
      Balanced Accuracy : 0.7531

       'Positive' Class : 1
```

Fig 3.8

Compare models after prediction and performance evaluation:

Based on the accuracy and kappa values, this comparison evaluates the performance of eight distinct models (KNN, Lasso, Ridge, Enet, RF, GBM, SVML, and SVMR) on a specific dataset. Five resamples were used for cross-validation training and testing of the models.



```
Call:
summary.resamples(object = compare)

Models: KNN, Lasso, Ridge, Enet, RF, GBM, SVML, SVMR
Number of resamples: 5

Accuracy
           Min.   1st Qu.    Median      Mean   3rd Qu.      Max. NA's
KNN   0.7167359 0.7174182 0.7217753 0.7202850 0.7221289 0.7233666    0
Lasso 0.7454031 0.7464415 0.7481874 0.7473743 0.7483865 0.7484527    0
Ridge 0.7435467 0.7441429 0.7470604 0.7465963 0.7478338 0.7503979    0
Enet  0.7453147 0.7463531 0.7481213 0.7473743 0.7485411 0.7485411    0
RF    0.7440771 0.7488286 0.7488948 0.7489127 0.7490938 0.7536693    0
GBM   0.7511272 0.7512821 0.7515693 0.7522368 0.7518564 0.7553492    0
SVML  0.7449611 0.7459995 0.7477677 0.7472151 0.7478338 0.7495137    0
SVMR  0.7477016 0.7503315 0.7503979 0.7515120 0.7519229 0.7572060    0

Kappa
           Min.   1st Qu.    Median      Mean   3rd Qu.      Max. NA's
KNN   0.4334757 0.4348364 0.4435539 0.4405701 0.4442544 0.4467301    0
Lasso 0.4908062 0.4928815 0.4963749 0.4947486 0.4967749 0.4969054    0
Ridge 0.4870934 0.4882843 0.4941224 0.4931927 0.4956676 0.5007958    0
Enet  0.4906294 0.4927047 0.4962444 0.4947486 0.4970822 0.4970822    0
RF    0.4881542 0.4976541 0.4977896 0.4978257 0.4981918 0.5073386    0
GBM   0.5022579 0.5025641 0.5031357 0.5044738 0.5037129 0.5106985    0
SVML  0.4899222 0.4920027 0.4955316 0.4944303 0.4956676 0.4990274    0
SVMR  0.4954031 0.5006585 0.5007958 0.5030241 0.5038513 0.5144120     0
```

Fig 3.9

This is a summary of the results from 5-fold cross-validation for several models: KNN, Lasso, Ridge, Elastic Net, Random Forest (RF), Gradient Boosting Machine (GBM), Support Vector Machine with linear kernel (SVML), and Support Vector Machine with radial kernel (SVMR). The evaluation metric used is accuracy and Kappa.

- For accuracy, the models with the highest mean accuracy are GBM (0.752) and SVMR (0.751), followed by Lasso (0.747), Ridge (0.747), Enet (0.747), RF (0.748), SVML (0.747), and KNN (0.720).

- For Kappa, the models with the highest mean Kappa are SVMR (0.503), followed by GBM (0.504), RF (0.498), Lasso (0.495), Enet (0.495), Ridge (0.493), SVML (0.494), and KNN (0.441).

It is worth noting that the optimal model selection should not be based solely on these results, and it is recommended to perform additional analyses, such as feature importance and model interpretation, to select the best model.

Neural Network:

We have used a neural network with 2 input layers that have 'ReLU' activation functions and an output layer that uses the SoftMax activation function. After running the network, we obtained the graph as follows and the confusion matrix below

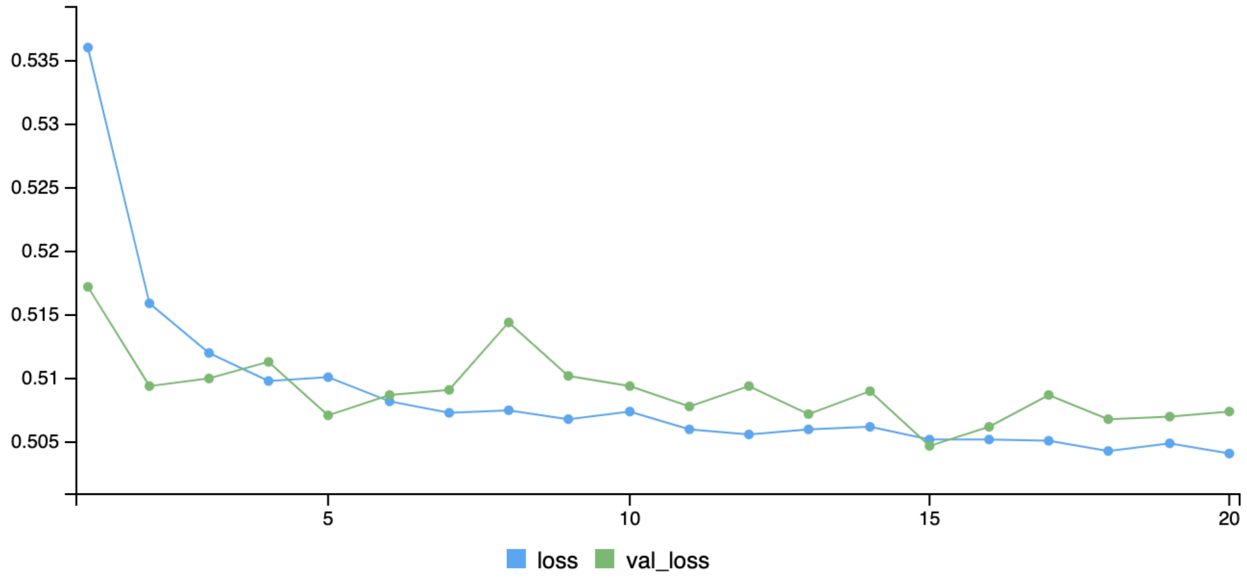
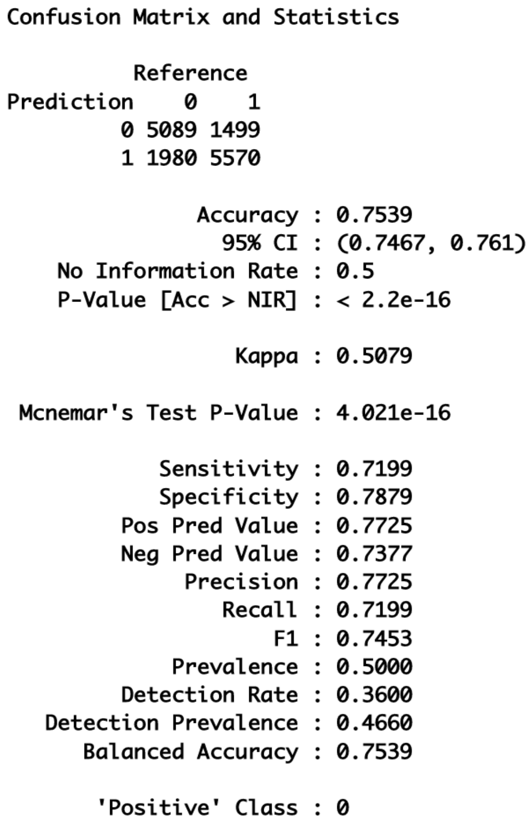
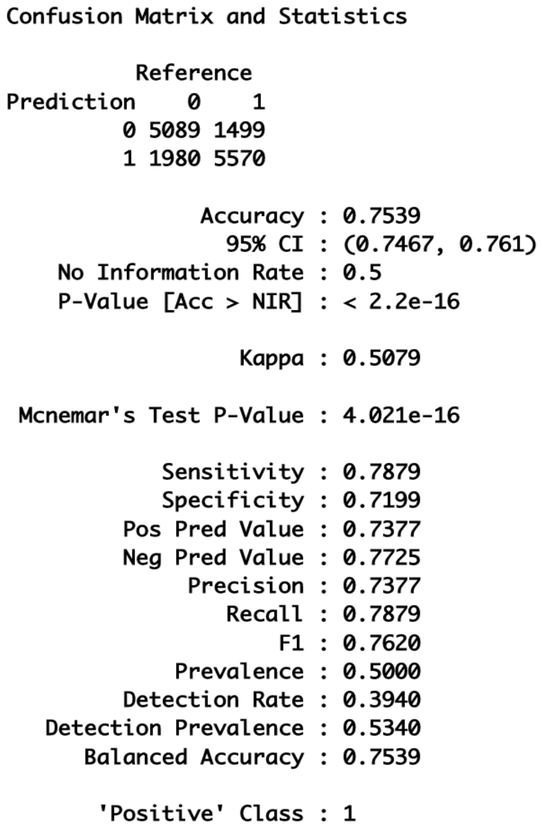


Fig 4.0



```
Confusion Matrix and Statistics

          Reference
Prediction    0    1
         0 5089 1499
         1 1980 5570

               Accuracy : 0.7539
                 95% CI : (0.7467, 0.761)
    No Information Rate : 0.5
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.5079

 Mcnemar's Test P-Value : 4.021e-16

            Sensitivity : 0.7199
            Specificity : 0.7879
         Pos Pred Value : 0.7725
         Neg Pred Value : 0.7377
              Precision : 0.7725
                 Recall : 0.7199
                     F1 : 0.7453
             Prevalence : 0.5000
         Detection Rate : 0.3600
   Detection Prevalence : 0.4660
      Balanced Accuracy : 0.7539

       'Positive' Class : 0
```

Fig 4.1



```
Confusion Matrix and Statistics

          Reference
Prediction    0    1
         0 5089 1499
         1 1980 5570

               Accuracy : 0.7539
                 95% CI : (0.7467, 0.761)
    No Information Rate : 0.5
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.5079

 Mcnemar's Test P-Value : 4.021e-16

            Sensitivity : 0.7879
            Specificity : 0.7199
         Pos Pred Value : 0.7377
         Neg Pred Value : 0.7725
              Precision : 0.7377
                 Recall : 0.7879
                     F1 : 0.7620
             Prevalence : 0.5000
         Detection Rate : 0.3940
   Detection Prevalence : 0.5340
      Balanced Accuracy : 0.7539

       'Positive' Class : 1
```

Fig 4.2

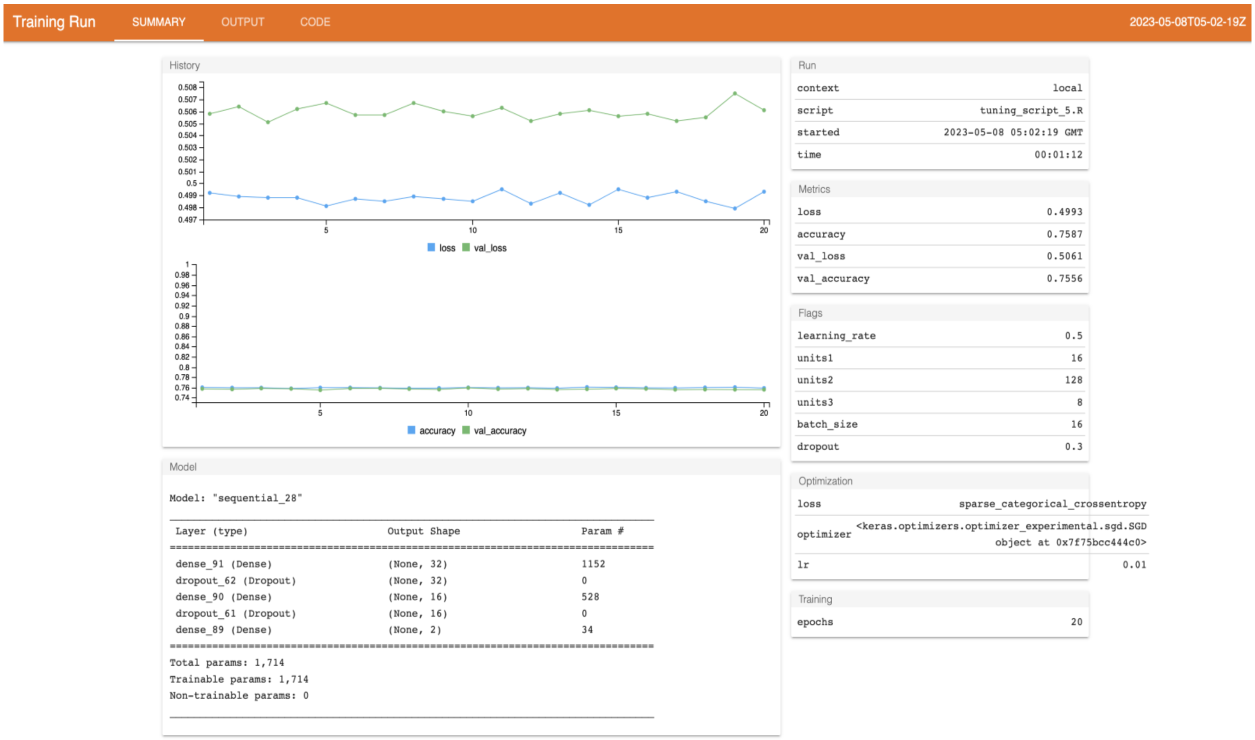
This confusion matrix and statistics represent the performance evaluation of a binary classification model. The confusion matrix shows the number of true positive (5089), false positive (1499), false negative (1980), and true negative (5570) predictions made by the model.

The accuracy of the model is 0.7539, which means that the model correctly predicted 75.39% of the instances. The kappa coefficient, which measures the agreement between predicted and actual classes, is 0.5079.

The sensitivity of the model is 0.7879, which means that the model correctly identified 78.79% of the positive instances. The specificity of the model is 0.7199, which means that the model correctly identified 71.99% of the negative instances. The positive predictive value (precision) is 0.7377, which means that when the model predicted the positive class, it was correct 73.77% of the time. The negative predictive value is 0.7725, which means that when the model predicted the negative class, it was correct 77.25% of the time.

The prevalence of the positive class is 0.5, which means that 50% of the instances belong to the positive class. The detection rate is 0.3940, which means that the model detected 39.40% of the positive instances. The detection prevalence is 0.5340, which means that 53.40% of the predicted positive instances were actually positive.

The balanced accuracy is equal to the accuracy in this case, as the prevalence of the two classes is the same. Finally, the p-value for the Mcnemar's test is < 2.2e-16, which means that there is a significant difference between the performance of the model and the null hypothesis.



Run

context	local
script	tuning_script_5.R
started	2023-05-08 05:02:19 GMT
time	00:01:12

Metrics

loss	0.4993
accuracy	0.7587
val_loss	0.5061
val_accuracy	0.7556

Flags

learning_rate	0.5
units1	16
units2	128
units3	8
batch_size	16
dropout	0.3

Optimization

loss	sparse_categorical_crossentropy
optimizer	<keras.optimizers.optimizer_experimental.sgd.SGD object at 0x7f75bcc444c0>
lr	0.01

Training

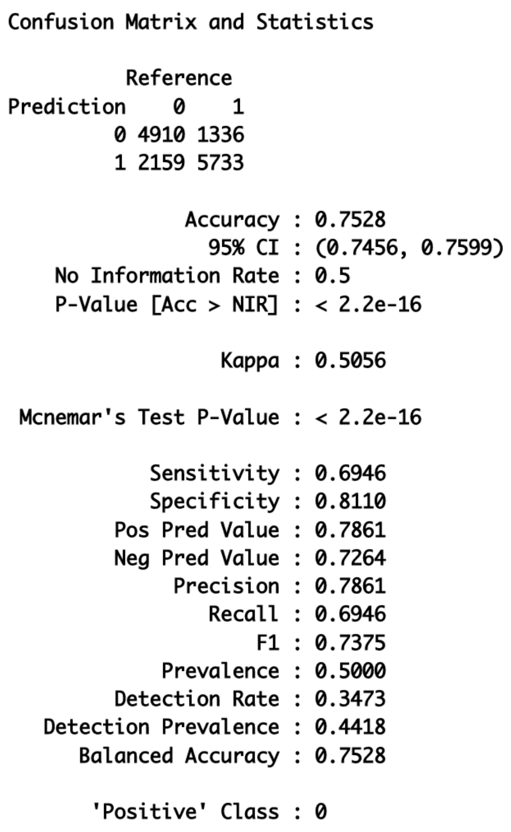
epochs	20

Model

```
Model: "sequential_28"
________________________________________________________________________________
 Layer (type)                    Output Shape                  Param #
================================================================================
 dense_91 (Dense)                (None, 32)                    1152
 dropout_62 (Dropout)            (None, 32)                    0
 dense_90 (Dense)                (None, 16)                    528
 dropout_61 (Dropout)            (None, 16)                    0
 dense_89 (Dense)                (None, 2)                     34
================================================================================
Total params: 1,714
Trainable params: 1,714
Non-trainable params: 0
________________________________________________________________________________
```

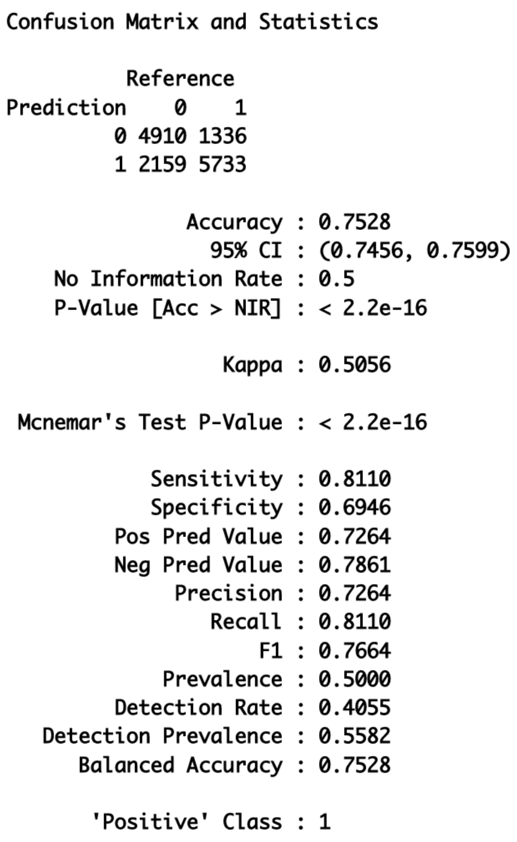
Fig 4.3

After fine-tuning the hyperparameters, we selected the best set of hyperparameters and trained the neural network model using these parameters. Additionally, we merged the training and validation data together to train the model. After running the neural network model with updated values, we found a confusion matrix and plot as below.



```
Confusion Matrix and Statistics

          Reference
Prediction    0    1
         0 4910 1336
         1 2159 5733

               Accuracy : 0.7528
                 95% CI : (0.7456, 0.7599)
    No Information Rate : 0.5
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.5056

 Mcnemar's Test P-Value : < 2.2e-16

            Sensitivity : 0.6946
            Specificity : 0.8110
         Pos Pred Value : 0.7861
         Neg Pred Value : 0.7264
              Precision : 0.7861
                 Recall : 0.6946
                     F1 : 0.7375
             Prevalence : 0.5000
         Detection Rate : 0.3473
   Detection Prevalence : 0.4418
      Balanced Accuracy : 0.7528

       'Positive' Class : 0
```

Fig 4.4



```
Confusion Matrix and Statistics

          Reference
Prediction    0    1
         0 4910 1336
         1 2159 5733

               Accuracy : 0.7528
                 95% CI : (0.7456, 0.7599)
    No Information Rate : 0.5
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.5056

 Mcnemar's Test P-Value : < 2.2e-16

            Sensitivity : 0.8110
            Specificity : 0.6946
         Pos Pred Value : 0.7264
         Neg Pred Value : 0.7861
              Precision : 0.7264
                 Recall : 0.8110
                     F1 : 0.7664
             Prevalence : 0.5000
         Detection Rate : 0.4055
   Detection Prevalence : 0.5582
      Balanced Accuracy : 0.7528

       'Positive' Class : 1
```

Fig 4.5

The given confusion matrix and statistics represent the performance of a binary classification model. The model has made predictions for two classes, 0 and 1, and the confusion matrix shows the number of true positives, false positives, false negatives, and true negatives. The model has predicted 4910 true negatives and 5733 true positives, but it has also made 2159 false positives and 1336 false negatives.

The accuracy of the model is 0.7528, which indicates that it has correctly predicted the class labels for 75.28% of the samples. The kappa score of 0.5056 indicates that the model's performance is slightly better than random chance. The sensitivity of the model is 0.8110, which means that it correctly identifies 81.10% of the positive class samples, while the specificity of the model is 0.6946, which means that it correctly identifies 69.46% of the negative class samples. The positive predictive value (PPV) of the model is 0.7264, which indicates that when the model predicts a positive class label, it is correct 72.64% of the time. The negative predictive value (NPV) of the model is 0.7861, which indicates that when the model predicts a negative class label, it is correct 78.61% of the time. The precision and recall of the model are also 0.7264 and 0.8110, respectively.

The prevalence of the positive class is 0.5, and the detection rate and detection prevalence are 0.4055 and 0.5582, respectively. The balanced accuracy of the model is 0.7528, which is the average of sensitivity and specificity. Overall, the model's performance is decent, but there is still room for improvement.

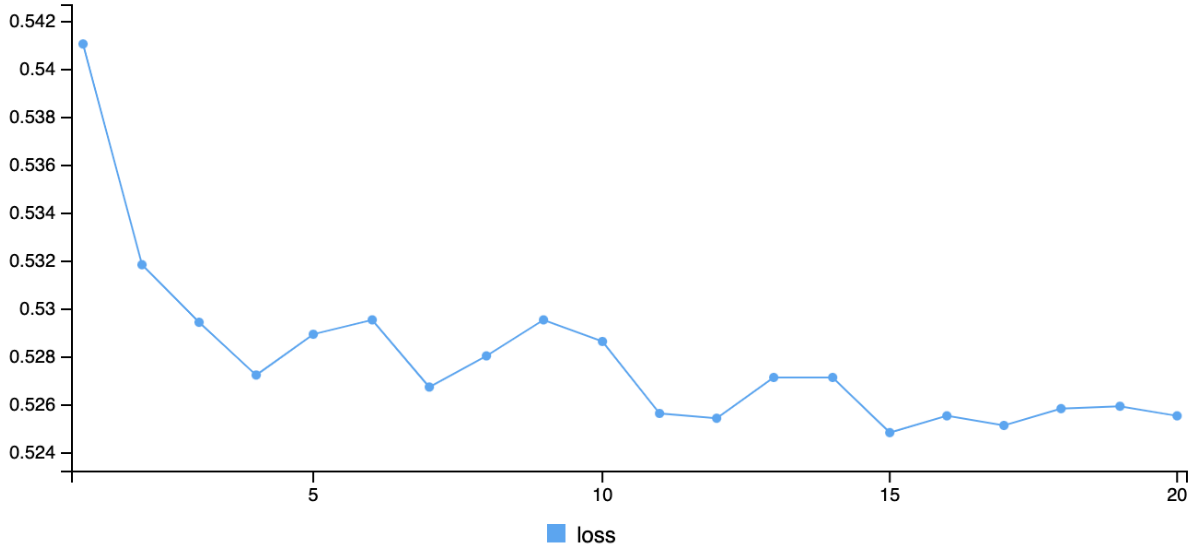


Fig 4.6

A neural network model had the highest accuracy, precision, recall, and F1 score for a classification task. However, other models like Random Forest and Gradient Boosting Model also performed well and may be considered depending on other factors.

Conclusion

The dataset includes data on people's health statuses, including their diabetes condition. Equal amounts of the data were divided into two groups, one representing people with diabetes and the other representing those without diabetes. The dataset also includes several characteristics, such as demographic data like age, gender, and Income as well as health markers like body mass index, physical activity level, and smoking status.

The split of the data is balanced, with each class accounting for 50% of the total data, according to the data analysis. This guarantees that there are no issues with the class imbalance that can impair the effectiveness of the machine learning methods.

The dataset has undergone various kinds of preprocessing methods, such as missing value removal, relevant variable identification, and data cleaning, to enhance its quality. As a result, any machine learning models that may be trained on this data will now be more accurate and efficient. This has helped to ensure that the dataset is of good quality and suitable for model training.

The results of statistical analyses showed a strong association between the occurrence of diabetes and several factors, including high blood pressure, high blood sugar, heart disease or attack, physical activity, BMI, general health, physical health, DiffWalk, age, education, and income. A number of additional characteristics, including MentHlth, CholCheck, Smoker, Fruits, Veggies, Heavy Alcohol Consumption, AnyHealthcare, and NoDocbcCost, were also significantly associated with diabetes.

These data collectively imply that several traits and circumstances are significantly linked to the development of diabetes. Machine learning models can be created using this data to predict the likelihood of diabetes based on these parameters. These results can also be used to guide public health initiatives and regulations meant to lower diabetes incidence.

From the evaluation results, KNN had the lowest minimum accuracy of 0.7167359 and the highest maximum accuracy of 0.7233666. GBM had the highest median accuracy of 0.7515693, while the median accuracies of KNN, Lasso, Ridge, Enet, RF, SVML, and SVMR were in the range of 0.7174182 to 0.7503979. The mean accuracy of all models was in the range of 0.7202850 to 0.7522368. There were no missing values in the accuracy summary.

For kappa, KNN had the lowest minimum kappa of 0.4334757, and SVMR had the highest maximum kappa of 0.5144120. GBM had the highest median kappa of 0.5031357, while the median kappas of KNN, Lasso, Ridge, Enet, RF, and SVML were in the range of 0.4348364 to 0.5007958. The mean kappa of all models was in the range of 0.4405701 to 0.5044738. There were no missing values in the kappa summary.

The neural network had an accuracy of 0.7528 and a kappa of 0.5056. The confusion matrix shows that the model correctly predicted 4910 true negatives and 5733 true positives, but misclassified 1336 false negatives and 2159 false positives.

In conclusion, the GBM model had the highest median accuracy and kappa, while SVMR had the highest maximum kappa. However, the neural network had a comparable accuracy and kappa to the other models.

Reference

[1]https://www.kaggle.com/datasets/alexteboul/diabetes-health-indicators-dataset?select=diabetes_binary_5050split_health_indicators_BRFSS2015.csv

[2]https://www.kaggle.com/code/anastasiyaigonina/diabetes-eda-hypothesis-testing-predictions

[3] https://www.kaggle.com/code/gabrielsober/diabetes-eda-prediction