AMERICAN INTERNATIONAL UNIVERSITY–BANGLADESH (AIUB)

Faculty of Science and Technology (FST)

Course Title: INTRODUCTION TO DATA SCIENCE

Fall 2023-2024

Section: (A), Group: 06

Project Title

To Predict Early Stage of Diabetes Using Naïve Bayes Classifier.

Supervised By

TOHEDUL ISLAM

Faculty of Science and Technology

American International University-Bangladesh

Submitted By:

NAME	ID
TONMOY DEY	20-44206-3
SHOWMITRA ROY	20-44208-3

Dataset Description:

This is an early–stage diabetes risk prediction dataset consisting of 520 samples. There are seventeen variables consisting of age, gender, polyuria, polydipsia, sudden weight loss, weakness, polyphagia, genital thrust, visual blurring, itching, irritability, delayed healing, partial paresis, muscle stiffness, Alopecia, obesity and class. In the dataset, all the instances are consisting of categorical value. Moreover, class variable is also divided in to two categories (Positive and Negative); if the outcome is positive then there is existence of diabetes. On the other hand, if it is negative then no diabetes.

Attributes:

Age: It gives us idea about how old the individuals are.

Gender: It gives us idea about the sex of the individual.

Polyuria: Excessive urination, a symptom often associated with diabetes due to increased glucose levels.

Polydipsia: Abnormally high thirst, another common symptom of diabetes caused by dehydration from frequent urination.

Sudden weight loss: A rapid and unexplained reduction in body weight, which can be indicative of underlying health issues, including diabetes.

Weakness: Generalized lack of physical strength or energy, which may be linked to diabetes-related fatigue.

Polyphagia: Excessive hunger, a symptom associated with diabetes due to the body's inability to utilize glucose properly.

Genital thrust: Increased urge or discomfort in the genital area, potentially linked to diabetes symptoms.

Visual blurring: Blurred vision, a common eye-related complication associated with diabetes.

Itching: Skin irritation or itching, which can be a result of diabetes-related skin conditions.

Irritability: Mood changes or heightened irritability, which may be influenced by fluctuations in blood sugar levels.

Delayed healing: Prolonged time for wounds or injuries to heal, a potential complication of diabetes.

Partial paresis: Partial paralysis or weakness in certain body parts, often associated with nerve damage in diabetes.

Muscle stiffness: Stiffness or rigidity in muscles, which may be a result of various diabetes-related factors.

Alopecia: Hair loss, which may be associated with diabetes or other health conditions.

Obesity: Excessive body weight, a risk factor for the development of type 2 diabetes.

Class: It let us know whether an individual is suffering from diabetes or not.

PURPOSE: Develop a predictive model (Naïve Bayes) using demographic data and diabetes symptoms to identify individuals at risk. The goal is to enhance early detection, understand symptom patterns, and contribute to predict diabetes management.

Project Overview:

The goal of this project is to develop a predictive model for identifying individuals at risk of diabetes and understanding associated symptoms. The dataset encompasses crucial demographic factors, including age and gender, along with a comprehensive set of symptoms such as polyuria, polydipsia, sudden weight loss, weakness, polyphagia, genital thrust, visual blurring, itching, irritability, delayed healing, partial paresis, muscle stiffness, Alopecia, and obesity. The target variable, "class," distinguishes individuals as either diagnosed with diabetes or not. Through rigorous analysis and modelling, this project seeks to contribute to the early detection and accurate prediction of diabetes, providing valuable insights into the manifestation of symptoms associated with this condition. The outcome aims to enhance medical intervention strategies and promote a proactive approach to managing diabetes-related health concerns. It is noticeable that the data set is not well formatted. The dataset has to be cleaned and pre-processed before using it.

Data pre-processing:

1. Importing the Dataset:

The dataset is located in a file called **"diabetes_data_upload.csv"** in the current working directory. To begin data pre-processing using R, the first step is to import the dataset. Once imported, the **diabetes_data_upload.csv** file is transformed into an R data frame and stored in a variable named "MyData". After printing the dataset, it looks like this-

R code:

```
MyData <- read.csv("F:/diabetes_data_upload.csv")

print(MyData)
```

	Age	Gender	Polyuria	Polydipsia	sudden.weight.loss	weakness	Polyphagia	Genital.thrush	visual.blurring	Itching	Irritability	delayed.healing
1	40	Male	No	Yes	No	Yes	No	No	No	Yes	No	Yes
2	58	Male	No	No	No	Yes	No	No	Yes	No	No	No
3	41	Male	Yes	No	No	Yes	Yes	No	No	Yes	No	Yes
4	45	Male	No	No	Yes	Yes	Yes	Yes	No	Yes	No	Yes
5	60	Male	Yes	Yes	Yes	Yes	Yes	No	Yes	Yes	Yes	Yes
6	55	Male	Yes	Yes	No	Yes	Yes	No	Yes	Yes	No	Yes
7	57	Male	Yes	Yes	No	Yes	Yes	Yes	No	No	No	Yes
8	66	Male	Yes	Yes	Yes	Yes	No	No	Yes	Yes	Yes	No
9	67	Male	Yes	Yes	No	Yes	Yes	Yes	No	Yes	Yes	No
10	70	Male	No	Yes	Yes	Yes	Yes	No	Yes	Yes	Yes	No
11	44	Male	Yes	Yes	No	Yes	No	Yes	No	No	Yes	Yes
12	38	Male	Yes	Yes	No	No	Yes	Yes	No	Yes	No	Yes
13	35	Male	Yes	No	No	No	Yes	Yes	No	No	Yes	Yes
14	61	Male	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	No	No
15	60	Male	Yes	Yes	No	Yes	Yes	No	Yes	Yes	No	Yes
16	58	Male	Yes	Yes	No	Yes	Yes	No	No	No	No	Yes
17	54	Male	Yes	Yes	Yes	Yes	No	Yes	No	No	No	Yes
18	67	Male	No	Yes	No	Yes	Yes	No	Yes	No	Yes	Yes
19	66	Male	Yes	Yes	No	Yes	Yes	No	Yes	No	No	No
20	43	Male	Yes	Yes	Yes	Yes	No	Yes	No	No	No	No

loss	weakness	Polyphagia	Genital.thrush	visual.blurring	Itching	Irritability	delayed.healing	partial.paresis	muscle.stiffness	Alopecia	Obesity	class
	Yes	No	No	No	Yes	No	Yes	No	Yes	Yes	Yes	Positive
	Yes	No	No	Yes	No	No	No	Yes	No	Yes	No	Positive
	Yes	Yes	No	No	Yes	No	Yes	No	Yes	Yes	No	Positive
	Yes	Yes	Yes	No	Yes	No	Yes	No	No	No	No	Positive
	Yes	Yes	No	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Positive
	Yes	Yes	No	Yes	Yes	No	Yes	No	Yes	Yes	Yes	Positive
	Yes	Yes	Yes	No	No	No	Yes	Yes	No	No	No	Positive
	Yes	No	No	Yes	Yes	Yes	No	Yes	Yes	No	No	Positive
	Yes	Yes	Yes	No	Yes	Yes	No	Yes	Yes	No	Yes	Positive
	Yes	Yes	No	Yes	Yes	Yes	No	No	No	Yes	No	Positive
	Yes	No	Yes	No	No	Yes	Yes	No	Yes	Yes	No	Positive
	No	Yes	Yes	No	Yes	No	Yes	No	Yes	No	No	Positive
	No	Yes	Yes	No	No	Yes	Yes	No	No	Yes	No	Positive
	Yes	Yes	Yes	Yes	Yes	No	No	No	No	Yes	Yes	Positive
	Yes	Yes	No	Yes	Yes	No	Yes	Yes	No	No	No	Positive
	Yes	Yes	No	No	No	No	Yes	Yes	Yes	No	No	Positive
	Yes	No	Yes	No	No	No	Yes	No	Yes	No	No	Positive
	Yes	Yes	No	Yes	No	Yes	Yes	Yes	Yes	Yes	Yes	Positive
	Yes	Yes	No	Yes	No	No	No	Yes	Yes	No	No	Positive
	Yes	No	Yes	No	No	No	No	No	No	No	No	Positive

2. Dealing with Missing Values:

For Numerical:

For checking, the missing value (NA) as our dataset is categorical expect (AGE). We will be using **colSums(is.na(Dataset))** for age attribute only as the value is numerical.

R code:

colSums(is.na(Dataset))

```
Console   Terminal ×   Background Jobs ×
R 4.3.1 · ~/
> colSums(is.na(MyData))
               Age            Gender           Polyuria         Polydipsia sudden.weight.loss           weakness
                 0                 0                  0                  0                 0                  0
        Polyphagia     Genital.thrush    visual.blurring            Itching       Irritability    delayed.healing
                 0                 0                  0                  0                 0                  0
   partial.paresis   muscle.stiffness           Alopecia            Obesity             class
                 0                 0                  0                  0                 0
>
```

For Categorical:

We have used **summary()** for the categorical attribute to find out the class, length and mode if it.

R Code:

summary(MyData)

```
Console   Terminal ×   Background Jobs ×
R 4.3.1 · ~/
> summary(MyData)
      Age           Gender            Polyuria          Polydipsia        sudden.weight.loss   weakness
 Min.   :16.00   Length:520         Length:520         Length:520         Length:520         Length:520
 1st Qu.:39.00   Class :character   Class :character   Class :character   Class :character   Class :character
 Median :47.50   Mode  :character   Mode  :character   Mode  :character   Mode  :character   Mode  :character
 Mean   :48.03
 3rd Qu.:57.00
 Max.   :90.00
  Polyphagia        Genital.thrush     visual.blurring      Itching          Irritability       delayed.healing
 Length:520         Length:520         Length:520         Length:520         Length:520         Length:520
 Class :character   Class :character   Class :character   Class :character   Class :character   Class :character
 Mode  :character   Mode  :character   Mode  :character   Mode  :character   Mode  :character   Mode  :character


 partial.paresis    muscle.stiffness     Alopecia           Obesity            class
 Length:520         Length:520         Length:520         Length:520         Length:520
 Class :character   Class :character   Class :character   Class :character   Class :character
 Mode  :character   Mode  :character   Mode  :character   Mode  :character   Mode  :character
```

The below R code creates tables for categorical variables in the dataset "MyData". It iterates through a list of categorical variable names ('Cat_Variables') and prints the frequency of each category, including any missing values (NA). The output consists of a series of tables, each labelled with the variable name. It is used to check the sum of the prevalence of different value of each variable. If it is less than 520 then it means, there is a missing value. This process is done to check the missing value for categorical data.

R Code:

```
Cat_Variables <- c('Gender', 'Polyuria', 'Polydipsia', 'sudden.weight.loss',
                   'weakness', 'Polyphagia', 'Genital.thrush', 'visual.blurring',
                   'Itching', 'Irritability', 'delayed.healing', 'partial.paresis',
                   'muscle.stiffness', 'Alopecia', 'Obesity', 'class')
for (variable in Cat_Variables) {
  cat("Table for", variable, ":\n")
  print(table(MyData[[variable]], useNA = "ifany"))
  cat("\n")
}
```

Console Terminal Background Jobs

R 4.3.1 · ~/

```
> Cat_Variables <- c('Gender', 'Polyuria', 'Polydipsia', 'sudden.weight.loss',
+                    'weakness', 'Polyphagia', 'Genital.thrush', 'visual.blurring',
+                    'Itching', 'Irritability', 'delayed.healing', 'partial.paresis',
+                    'muscle.stiffness', 'Alopecia', 'Obesity', 'class')
> for (variable in Cat_Variables) {
+   cat("Table for", variable, ":\n")
+   print(table(MyData[[variable]], useNA = "ifany"))
+   cat("\n")
+ }
Table for Gender :

Female   Male
   192    328

Table for Polyuria :

 No Yes
262 258

Table for Polydipsia :

 No Yes
287 233

Table for sudden.weight.loss :

 No Yes
303 217

Table for weakness :

 No Yes
215 305
```

Console Terminal Background Jobs

R 4.3.1 · ~/

```
Table for Polyphagia :

 No Yes
283 237

Table for Genital.thrush :

 No Yes
404 116

Table for visual.blurring :

 No Yes
287 233

Table for Itching :

 No Yes
267 253

Table for Irritability :

 No Yes
394 126

Table for delayed.healing :

 No Yes
281 239

Table for partial.paresis :

 No Yes
296 224
```

3. Finding the Correlation between class to other attributes:

The below R code defines a list of categorical variables (`Chi_Test`) to conduct chi-square tests later. These variables represent different features, such as age, gender, and symptoms. It will be tested for independence with another variable called "class." Moreover, below R code checks if there's any connection between each categorical variable in `Chi_Test` and the "class" variable in the dataset `MyData` using chi-square tests. It prints the results for each test.

R Code:

```
Chi_Test <- c("Age", "Gender", "Polyuria", "Polydipsia", "sudden.weight.loss",
"weakness",  "Polyphagia", "Genital.thrush", "visual.blurring", "Itching", "Irritability",

"delayed.healing", "partial.paresis", "muscle.stiffness", "Alopecia", "Obesity")

for (variable in Chi_Test) {

  cat("Chi-square test for", variable, "and class:\n")

  print(chisq.test(MyData[[variable]], MyData$class))

  cat("\n")

}
```

```
Console   Terminal ×   Background Jobs ×
R 4.3.1 · ~/
> Chi_Test <- c("Age", "Gender", "Polyuria", "Polydipsia", "sudden.weight.loss", "weakness",
+               "Polyphagia", "Genital.thrush", "visual.blurring", "Itching", "Irritability",
+               "delayed.healing", "partial.paresis", "muscle.stiffness", "Alopecia", "Obesity")
> for (variable in Chi_Test) {
+   cat("Chi-square test for", variable, "and class:\n")
+   print(chisq.test(MyData[[variable]], MyData$class))
+   cat("\n")
+ }
Chi-square test for Age and class:

        Pearson's Chi-squared test

data:  MyData[[variable]] and MyData$class
X-squared = 145.79, df = 50, p-value = 2.655e-11


Chi-square test for Gender and class:

        Pearson's Chi-squared test with Yates' continuity correction

data:  MyData[[variable]] and MyData$class
X-squared = 103.04, df = 1, p-value < 2.2e-16


Chi-square test for Polyuria and class:

        Pearson's Chi-squared test with Yates' continuity correction

data:  MyData[[variable]] and MyData$class
X-squared = 227.87, df = 1, p-value < 2.2e-16
```

```
Console   Terminal ×   Background Jobs ×
R 4.3.1 · ~/
Chi-square test for Polydipsia and class:

        Pearson's Chi-squared test with Yates' continuity correction

data:  MyData[[variable]] and MyData$class
X-squared = 216.17, df = 1, p-value < 2.2e-16


Chi-square test for sudden.weight.loss and class:

        Pearson's Chi-squared test with Yates' continuity correction

data:  MyData[[variable]] and MyData$class
X-squared = 97.296, df = 1, p-value < 2.2e-16


Chi-square test for weakness and class:

        Pearson's Chi-squared test with Yates' continuity correction

data:  MyData[[variable]] and MyData$class
X-squared = 29.768, df = 1, p-value = 4.87e-08


Chi-square test for Polyphagia and class:

        Pearson's Chi-squared test with Yates' continuity correction

data:  MyData[[variable]] and MyData$class
X-squared = 59.595, df = 1, p-value = 1.165e-14
```

Converting Age Attribute to Categorical:

As we have Age attribute in our dataset, moreover it is numeric in nature so, that's the reason we have converted it into categorical dataset by dividing it within four categories ("Young," "Adult," "Middle-aged," "Old") based on their age. The below code is used to examine the range of ages in the "Age" column of the dataset `MyData`, printing both the maximum and minimum values.

R Code:

MyData[,1]

print(max(MyData$Age))

print(min(MyData$Age))

```
Console   Terminal ×   Background Jobs ×
R 4.3.1 · ~/
> MyData[,1]
  [1] 40 58 41 45 60 55 57 66 67 70 44 38 35 61 60 58 54 67 66 43 62 54 39 48 58 32 42 52 38 53 57 41 37 54 49 48 60 63 35 30 53 50
 [43] 50 35 40 48 60 60 35 46 36 50 60 50 51 38 66 53 59 39 65 35 55 60 45 40 30 35 25 50 40 35 65 38 50 55 48 55 39 43 35 47 50 48
 [85] 35 49 38 28 68 35 45 48 40 40 36 56 30 31 35 39 48 85 90 72 70 69 58 47 25 39 53 52 68 79 55 45 30 45 65 34 48 35 40 47 38 55
[127] 66 57 32 48 47 43 30 16 35 66 54 58 51 40 47 62 49 53 68 61 39 38 44 45 50 42 55 57 62 33 55 48 56 38 28 68 35 45 48 40 57 41
[169] 37 54 49 48 60 63 35 30 53 50 50 35 40 31 35 39 48 85 90 72 70 69 58 54 64 36 43 31 66 61 58 69 40 28 37 34 30 67 60 58 54 43
[211] 39 40 43 49 47 45 57 72 30 27 38 43 40 55 68 29 37 30 45 47 35 32 56 50 52 26 60 65 72 30 45 65 70 35 54 30 46 53 42 55 48 55
[253] 39 43 35 47 50 48 35 62 33 55 48 56 38 28 68 35 45 48 40 57 47 45 57 72 30 27 38 43 40 47 45 57 72 30 27 38 43 40 54 30 46 53
[295] 42 55 48 55 39 43 35 47 61 58 69 40 28 37 34 30 67 60 58 54 43 33 55 36 28 34 65 34 64 44 36 43 53 47 58 56 51 59 50 30 46 53
[337] 42 55 48 55 39 43 35 47 61 58 69 40 28 37 34 30 67 60 58 54 43 33 55 48 56 38 28 68 35 45 48 40 57 47 45 57 72 30 27 38 43 40
[379] 47 62 49 53 68 61 39 38 44 36 43 53 47 58 56 51 59 50 30 46 53 64 44 36 43 53 47 58 56 51 59 50 30 46 53 42 55 48 55 39 43 35
[421] 47 61 67 66 43 62 54 39 48 58 32 42 52 38 53 57 41 37 54 49 48 60 63 35 30 53 50 50 35 40 48 60 38 28 68 35 45 48 40 57 47 45
[463] 57 72 30 27 38 43 40 47 45 57 72 30 27 38 43 40 54 30 46 53 42 55 48 55 39 43 50 30 46 53 64 44 36 43 53 47 68 64 66 67 70 44
[505] 38 35 61 60 58 54 67 66 43 62 54 39 48 58 32 42
> print(max(MyData$Age))
[1] 90
> print(min(MyData$Age))
[1] 16
> }
```

The below R code categorizes individuals in the dataset `MyData` into age groups ("Young," "Adult," "Middle-aged," "Old") based on their ages. It creates a new variable called "Age_Category" using nested `ifelse` statements. The resulting dataset is then displayed with the first few rows shown using `head(MyData)`.

R Code:

```
MyData$Age_Category <- ifelse(MyData$Age >= 15 & MyData$Age <= 25, "Young",
                              ifelse (MyData$Age >= 26 & MyData$Age <= 39, "Adult",
                                      ifelse (MyData$Age >= 40 & MyData$Age <= 59, "Middle-aged",
                                              ifelse (MyData$Age >= 60 & MyData$Age <= 99, "Old", NA))))
head(MyData)
```

```
Console   Terminal ×   Background Jobs ×
R 4.3.1 · ~/
> MyData$Age_Category <- ifelse(MyData$Age >= 15 & MyData$Age <= 25, "Young",
+                               ifelse (MyData$Age >= 26 & MyData$Age <= 39, "Adult",
+                                       ifelse (MyData$Age >= 40 & MyData$Age <= 59, "Middle-aged",
+                                               ifelse (MyData$Age >= 60 & MyData$Age <= 99, "Old", NA))))
> head(MyData)
  Age Gender Polyuria Polydipsia sudden.weight.loss weakness Polyphagia Genital.thrush visual.blurring Itching Irritability
1  40   Male       No        Yes                 No      Yes         No             No              No     Yes           No
2  58   Male       No         No                 No      Yes         No             No             Yes      No           No
3  41   Male      Yes         No                 No      Yes        Yes             No              No     Yes           No
4  45   Male       No         No                Yes      Yes        Yes            Yes              No     Yes           No
5  60   Male      Yes        Yes                Yes      Yes        Yes             No             Yes     Yes          Yes
6  55   Male      Yes        Yes                 No      Yes        Yes             No             Yes     Yes           No
  delayed.healing partial.paresis muscle.stiffness Alopecia Obesity    class Age_Category
1             Yes              No              Yes      Yes     Yes Positive  Middle-aged
2              No             Yes               No      Yes      No Positive  Middle-aged
3             Yes              No              Yes      Yes      No Positive  Middle-aged
4             Yes              No               No       No      No Positive  Middle-aged
5             Yes             Yes              Yes      Yes     Yes Positive          Old
6             Yes              No              Yes      Yes     Yes Positive  Middle-aged
> }
```

Feature Selection:

After using "Chi_square" test, we have found out the "p-value" and we have selected those attributes whose values are less than 0.05. Moreover, we have used Age_Category (converted to categorical dataset) instead of AGE. The missing attributes are Itching, delayed healing, weakness, and obesity.

R Code:

```
Selected_Features <- MyData[, c('Age_Category', 'Gender', 'Polyuria', 'Polydipsia',
'sudden.weight.loss', 'Polyphagia', 'Genital.thrush', 'visual.blurring' , 'Irritability',

'partial.paresis', 'muscle.stiffness', 'Alopecia', 'class')]

head(Selected_Features)
```

```
Console   Terminal ×   Background Jobs ×
R 4.3.1 · ~/
> Selected_Features <- MyData[, c('Age_Category', 'Gender', 'Polyuria', 'Polydipsia', 'sudden.weight.loss',
+                                 'Polyphagia', 'Genital.thrush', 'visual.blurring' , 'Irritability',
+                                 'partial.paresis', 'muscle.stiffness', 'Alopecia', 'class')]
> head(Selected_Features)
  Age_Category Gender Polyuria Polydipsia sudden.weight.loss Polyphagia Genital.thrush visual.blurring Irritability partial.paresis
1  Middle-aged   Male       No        Yes                 No         No             No              No           No              No
2  Middle-aged   Male       No         No                 No         No             No             Yes           No             Yes
3  Middle-aged   Male      Yes         No                 No        Yes             No              No           No              No
4  Middle-aged   Male       No         No                Yes        Yes            Yes              No           No              No
5          Old   Male      Yes        Yes                Yes        Yes             No             Yes          Yes             Yes
6  Middle-aged   Male      Yes        Yes                 No        Yes             No             Yes           No              No
  muscle.stiffness Alopecia    class
1              Yes      Yes Positive
2               No      Yes Positive
3              Yes      Yes Positive
4               No       No Positive
5              Yes      Yes Positive
6              Yes      Yes Positive
> }
```

After selecting the correlated attributes, we have find out the summary(), nrow() (to know the number of instances) and also the names() (to know the names of the attributes those are selected) of those attributes.

R Code:

summary(Selected_Features)

names(Selected_Features)

nrow(Selected_Features)

```
Console   Terminal ×   Background Jobs ×
R 4.3.1 · ~/
> summary(Selected_Features)
 Age_Category          Gender            Polyuria          Polydipsia        sudden.weight.loss  Polyphagia
 Length:520         Length:520         Length:520         Length:520         Length:520         Length:520
 Class :character   Class :character   Class :character   Class :character   Class :character   Class :character
 Mode  :character   Mode  :character   Mode  :character   Mode  :character   Mode  :character   Mode  :character
 Genital.thrush     visual.blurring    Irritability       partial.paresis    muscle.stiffness     Alopecia
 Length:520         Length:520         Length:520         Length:520         Length:520         Length:520
 Class :character   Class :character   Class :character   Class :character   Class :character   Class :character
 Mode  :character   Mode  :character   Mode  :character   Mode  :character   Mode  :character   Mode  :character
    class
 Length:520
 Class :character
 Mode  :character
> names(Selected_Features)
 [1] "Age_Category"       "Gender"             "Polyuria"           "Polydipsia"         "sudden.weight.loss" "Polyphagia"
 [7] "Genital.thrush"     "visual.blurring"    "Irritability"       "partial.paresis"    "muscle.stiffness"   "Alopecia"
[13] "class"
> nrow(Selected_Features)
[1] 520
> }
```

Building the Model:

1. Building Model With 10-Fold Cross Validation:

We have used **" library(e1071)"** (for building the naïve Bayes model) and **"ibrary(caret)"** (To find the predictive accuracy using 10-fold cross validation and dividing the data into training and test set) libraries to perform 10-fold cross-validation for a Naive Bayes classifier on the dataset "Selected_Features,". For calculating accuracy for each fold, we have stored the results in "accuracy_results." The final accuracy summary is the mean of the accuracy results across folds.

R Code:

```
library(e1071)
library(caret)
set.seed(123)
folds <- createFolds(Selected_Features$class, k = 10, list = TRUE, returnTrain = TRUE)
accuracy_results <- numeric(length(folds))
for (i in 1:10) {
  Training1 <- Selected_Features[folds[[i]], ]
  Testing1 <- Selected_Features[-folds[[i]], ]
  NB_model1 <- naiveBayes(class ~ ., data = Training1, laplace = 1)
  predictions <- predict(NB_model1, Testing1[, -13])
  conf_matrix <- table(predictions, Testing1$class)
  accuracy_results[i] <- sum(diag(conf_matrix)) / sum(conf_matrix)
}
```

Console Terminal Background Jobs

R 4.3.1 · ~/

```
> library(e1071)
> library(caret)
> set.seed(123)
> folds <- createFolds(Selected_Features$class, k = 10, list = TRUE, returnTrain = TRUE)
> accuracy_results <- numeric(length(folds))
>
> for (i in 1:10) {
+   Training1 <- Selected_Features[folds[[i]], ]
+   Testing1 <- Selected_Features[-folds[[i]], ]
+
+   NB_model1 <- naiveBayes(class ~ ., data = Training1, laplace = 1)
+   predictions <- predict(NB_model1, Testing1[, -13])
+
+   conf_matrix <- table(predictions, Testing1$class)
+   accuracy_results[i] <- sum(diag(conf_matrix)) / sum(conf_matrix)
+ }
> }
```

We have printed the (NB_model1) to see the prior and conditional probabilities for Naïve Bayes classifier.

R Code:

print(NB_model1)

```
Console   Terminal ×   Background Jobs ×
R 4.3.1 · ~/
> print(NB_model1)

Naive Bayes Classifier for Discrete Predictors

Call:
naiveBayes.default(x = X, y = Y, laplace = laplace)

A-priori probabilities:
Y
 Negative  Positive
0.3846154 0.6153846

Conditional probabilities:
          Age_Category
Y               Adult Middle-aged         Old       Young
  Negative 0.311111111 0.555555556 0.150000000 0.005555556
  Positive 0.260416667 0.534722222 0.204861111 0.013888889

          Gender
Y             Female      Male
  Negative 0.1055556 0.9055556
  Positive 0.5416667 0.4652778

          Polyuria
Y                 No        Yes
  Negative 0.93333333 0.07777778
  Positive 0.23611111 0.77083333

          Polydipsia
Y                 No       Yes
  Negative 0.9611111 0.0500000
  Positive 0.3159722 0.6909722
```

```
Console   Terminal ×   Background Jobs ×
R 4.3.1 · ~/
          sudden.weight.loss
Y                  No       Yes
  Negative 0.8555556 0.1555556
  Positive 0.4166667 0.5902778

          Polyphagia
Y                  No       Yes
  Negative 0.7777778 0.2333333
  Positive 0.4027778 0.6041667

          Genital.thrush
Y                  No       Yes
  Negative 0.8333333 0.1777778
  Positive 0.7395833 0.2673611

          visual.blurring
Y                  No       Yes
  Negative 0.7166667 0.2944444
  Positive 0.4652778 0.5416667

          Irritability
Y                   No        Yes
  Negative 0.93333333 0.07777778
  Positive 0.66319444 0.34375000

          partial.paresis
Y                  No       Yes
  Negative 0.8444444 0.1666667
  Positive 0.4062500 0.6006944

          muscle.stiffness
Y                  No       Yes
  Negative 0.7166667 0.2944444
  Positive 0.5972222 0.4097222
```

We have printed (conf_matrix) to see the Confusion Matrix.

R Code:

print(conf_matrix)

```
Console   Terminal ×   Background Jobs ×
R 4.3.1 · ~/
> print(conf_matrix)

predictions Negative Positive
   Negative       18        6
   Positive        2       26
> }
```

We have used the below code to calculate the mean accuracy, precision, recall, and F1 score for a Naive Bayes classifier's performance using 10-fold cross-validation. The results are stored in a data frame named "Results1" and we have printed it to see the model evaluation.

R Code:

```
mean_accuracy <- mean(accuracy_results)

rec <- sensitivity(conf_matrix)

prec <- posPredValue(conf_matrix)

f_measure1 <- 2 * (prec * rec) / (prec + rec)

Results1 <- data.frame(Model = 'Naive Bayes', Accuracy = mean_accuracy, Precision = prec, Recall = rec, F1Score = f_measure1)

print(Results1)
```

```
Console   Terminal ×   Background Jobs ×
R 4.3.1 · ~/
> mean_accuracy <- mean(accuracy_results)
> rec <- sensitivity(conf_matrix)
> prec <- posPredValue(conf_matrix)
> f_measure1 <- 2 * (prec * rec) / (prec + rec)
>
> Results1 <- data.frame(Model = 'Naive Bayes', Accuracy = mean_accuracy, Precision = prec, Recall = rec, F1Score = f_measure1)
> print(Results1)
       Model  Accuracy Precision Recall   F1Score
1 Naive Bayes 0.8884615      0.75    0.9 0.8181818
> }
```

2. Building Model with Dividing the Data into Training and Test Set:

We split the "Selected_Features" dataset into training (80%) and testing (20%) sets, building a Naive Bayes classifier ("NB_model2") on the training set, and evaluating its accuracy on the testing set, reporting the results in "conf_matrix_test" and "accuracy_test."

R Code:

```
s <- sample(nrow(Selected_Features), nrow(Selected_Features) * 0.8)

Training2 <- Selected_Features[s, ]

Testing2 <- Selected_Features[-s, ]

NB_model2 <- naiveBayes(class ~ ., data = Training2, laplace = 1)

predictions_test <- predict(NB_model2, Testing2[, -13])

conf_matrix_test <- table(predictions_test, Testing2$class)

accuracy_test <- sum(diag(conf_matrix_test)) / sum(conf_matrix_test)
```

```
Console   Terminal ×   Background Jobs ×
R 4.3.1 · ~/
> s <- sample(nrow(Selected_Features), nrow(Selected_Features) * 0.8)
> Training2 <- Selected_Features[s, ]
> Testing2 <- Selected_Features[-s, ]
> NB_model2 <- naiveBayes(class ~ ., data = Training2, laplace = 1)
> predictions_test <- predict(NB_model2, Testing2[, -13])
> conf_matrix_test <- table(predictions_test, Testing2$class)
> accuracy_test <- sum(diag(conf_matrix_test)) / sum(conf_matrix_test)
> }
```

We have printed the (NB_model2) to see the prior and conditional probabilities for Naïve Bayes classifier.

R Code:

print(NB_model2)

```
Console   Terminal ×   Background Jobs ×
R 4.3.1 · ~/
> print(NB_model2)

Naive Bayes Classifier for Discrete Predictors

Call:
naiveBayes.default(x = X, y = Y, laplace = laplace)

A-priori probabilities:
Y
 Negative  Positive
0.3677885 0.6322115

Conditional probabilities:
          Age_Category
Y                Adult Middle-aged         Old       Young
  Negative 0.267973856 0.575163399 0.176470588 0.006535948
  Positive 0.269961977 0.528517110 0.205323194 0.011406844

          Gender
Y            Female      Male
  Negative 0.1045752 0.9084967
  Positive 0.5513308 0.4562738

          Polyuria
Y                 No        Yes
  Negative 0.92810458 0.08496732
  Positive 0.25095057 0.75665399

          Polydipsia
Y                 No        Yes
  Negative 0.97385621 0.03921569
  Positive 0.31558935 0.69201521
```

```
Console   Terminal ×   Background Jobs ×
R 4.3.1 · ~/
          sudden.weight.loss
Y                 No       Yes
  Negative 0.8496732 0.1633987
  Positive 0.4068441 0.6007605

          Polyphagia
Y                 No       Yes
  Negative 0.7581699 0.2549020
  Positive 0.4334601 0.5741445

          Genital.thrush
Y                 No       Yes
  Negative 0.8496732 0.1633987
  Positive 0.7680608 0.2395437

          visual.blurring
Y                 No       Yes
  Negative 0.6993464 0.3137255
  Positive 0.4524715 0.5551331

          Irritability
Y                  No        Yes
  Negative 0.91503268 0.09803922
  Positive 0.63498099 0.37262357

          partial.paresis
Y                 No       Yes
  Negative 0.8235294 0.1895425
  Positive 0.3764259 0.6311787

          muscle.stiffness
Y                 No       Yes
  Negative 0.6862745 0.3267974
  Positive 0.5855513 0.4220532
```

We have printed (conf_matrix_test) to see the Confusion Matrix.

R Code:

print(conf_matrix_test)

```
Console   Terminal ×   Background Jobs ×
R 4.3.1 · ~/
> print(conf_matrix_test)

predictions_test Negative Positive
        Negative       42        7
        Positive        5       50
> }
```

We have calculated recall, precision, and F1 score for a Naive Bayes classifier on a separate testing set, storing the results in a data frame named "Results2" and we have printed it to see the model evaluation.

R-Code:

recall <- sensitivity(conf_matrix_test)

precision <- posPredValue(conf_matrix_test)

f_measure <- 2 * (precision * recall) / (precision + recall)

Results2 <- data.frame(Model = 'Naive Bayes', Accuracy = accuracy_test, Precision = precision, Recall = recall, F1Score = f_measure)

print(Results2)

```
Console   Terminal ×   Background Jobs ×
R 4.3.1 · ~/
> recall <- sensitivity(conf_matrix_test)
> precision <- posPredValue(conf_matrix_test)
> f_measure <- 2 * (precision * recall) / (precision + recall)
>
> Results2 <- data.frame(Model = 'Naive Bayes', Accuracy = accuracy_test, Precision = precision, Recall = recall, F1Score = f_measure)
> print(Results2)
       Model  Accuracy Precision   Recall F1Score
1 Naive Bayes 0.8846154 0.8571429 0.893617   0.875
> }
```

Conclusion:

In summary, this diabetes prediction project is important for spotting early signs and risks of diabetes. Using this model, we can take steps about how to overcome it. Moreover, we have calculated metrics like accuracy, precision, recall, and F1 score. These provide us with a quick overview of the model's performance and its ability to identify diabetes cases accurately. Analyzing factors like age, gender, and symptoms gives us a better understanding of how diabetes shows up. The goal is to contribute to better healthcare strategies for timely detection and management of diabetes.