# AMERICAN INTERNATIONAL UNIVERSITY-BANGLADESH

Choose an item.

# Project Cover Sheet

| | | | |
|---|---|---|---|
| Assignment Title: | Healthcare Dataset Stroke Prediction | | |
| Assignment No: | Final Project 1 | Date of Submission: | 25 December 2023 |
| Course Title: | Introduction To Data Science | | |
| Course Code: | CSC4180 | Section: | B |
| Semester: | Fall 2023-24 | Course Teacher: | Tohedul Islam |

**Declaration and Statement of Authorship:**

1. I/we hold a copy of this Assignment/Case-Study, which can be produced if the original is lost/damaged.
2. This Assignment/Case-Study is my/our original work and no part of it has been copied from any other student's work or from any other source except where due acknowledgement is made.
3. No part of this Assignment/Case-Study has been written for me/us by any other person except where such collaborationhas been authorized by the concerned teacher and is clearly acknowledged in the assignment.
4. I/we have not previously submitted or currently submitting this work for any other course/unit.
5. This work may be reproduced, communicated, compared and archived for the purpose of detecting plagiarism.
6. I/we give permission for a copy of my/our marked work to be retained by the Faculty for review and comparison, including review by external examiners.
7. I/we understand thatPlagiarism is the presentation of the work, idea or creation of another person as though it is your own. It is a formofcheatingandisaveryseriousacademicoffencethatmayleadtoexpulsionfromtheUniversity. Plagiarized material can be drawn from, and presented in, written, graphic and visual form, including electronic data, and oral presentations. Plagiarism occurs when the origin of them arterial used is not appropriately cited.
8. I/we also understand that enabling plagiarism is the act of assisting or allowing another person to plagiarize or to copy my/our work.

* *Student(s) must complete all details except the faculty use part.*
** Please submit all assignments to your course teacher or the office of the concerned teacher.

Group Name/No.: 19

| No | Name | ID | Program | Signature |
|---|---|---|---|---|
| 1 | MOHAMMAD BIN HARUN | 21-44583-1 | BSc [CSE] | |
| 2 | MD. HARUN OR RASHID | 21-44586-1 | BSc [CSE] | |

| *Faculty use only* | | |
|---|---|---|
| FACULTYCOMMENTS | **Marks Obtained** | |
| | | |
| | **Total Marks** | |
| | | |
| | | |

## Description of Dataset:

The Healthcare Dataset on stroke the dataset is used analysis from the specific method. On this dataset predicted stroke occurrence based on naïve bayes classification. Before using model classification first prepare the dataset to handle missing value and outliers. Using the Pearson's Chi-squared test find the significant relationship between attribute and class attribute (stroke). If there is no significant relationship found for the attribute remove the attribute from the dataset because there is no relationship with the class attribute (stroke). After all that using the naïve bayes classification model for the train data and predict the test data. Find the predictive accuracy of the naïve bayes classifier. And generate the confusion matrix calculate the recall, precision and f-measure value of classifier.

## Attributes:

The dataset contains several attributes that are various aspects related to stroke. The key attributes:

1. id: This id is unique identifier for each instance.
2. gender: gender of the individuals in the dataset.
3. age: age of individuals dataset
4. hypertension: Binary value of an individuals has hypertension (1) or not (0).
5. heart disease: Binary value of an individuals has a heart disease (1) or not (0).
6. ever married: ever married of individuals has ever been married (Yes) or not (No).
7. work type: The type of work individual is engaged in categorizing the occupation.
8. Residence type: Binary indicator of whether an individual resides in urban or rural area.
9. smoking status: The smoking habits of individuals categorizing as formerly smoked, never smoked or smokes.
10. BMI: Body Mass Index. Measure of body fat based on height and weight.
11. Average glucose level: The average glucose level in blood.
12. Stroke: Binary indicator of an individuals had a stroke (1) or not (0). And also, this is a class attribute.

## Data Preprocessing:

For Dataset are preprocessing clean data involved handle missing values, outliers. Imputation of missing values such as fill the mean for BMI. Outlier's removal based on the interquartile range (IQR) method.

**Library called:**

library(readxl), library(dplyr), library(e1071), library(caret)

**Description:**

The library "readxl" is called for reading the excel file. The library "dplyr" is called for manipulation the data. The library "e1071" is called that provide statistical learning, naïve bayes classifier. And the library "caret" called for provide training and evaluating model.

**Read dataset:**

Read the dataset from the excel file using the read_excel(). And show the dataset on the terminal.

| Code | |
|---|---|
| stroke<- read_excel("E:/AIUB/9th semester/Data Science/Final/Dataset/healthcare-dataset-stroke-data.xlsx")<br><br>print(stroke) | |
| **Output** | **Description** |
| # A tibble: 5,110 x 12<br>id gender age hypertension heart_disease ever_married work_type Residence_type avg_glucose_level bmi<br><dbl> <chr> <dbl> <dbl> <dbl> <chr> <chr> <chr> <dbl> <chr><br>1 9046 Male 67 0 1 Yes Private Urban 229. 36.6<br>2 51676 Female 61 0 0 Yes Self-employ… Rural 202. N/A<br>3 31112 Male 80 0 1 Yes Private Rural 106. 32.5<br>4 60182 Female 49 0 0 Yes Private Urban 171. 34.4<br>5 1665 Female 79 1 0 Yes Self-employ… Rural 174. 24<br>6 56669 Male 81 0 0 Yes Private Urban 186. 29<br>7 53882 Male 74 1 1 Yes Private Rural 70.1 27.4<br>8 10434 Female 69 0 0 No Private Urban 94.4 22.8<br>9 27419 Female 59 0 0 Yes Private Rural 76.2 N/A<br>10 60491 Female 78 0 0 Yes Private Urban 58.6 24.2<br># i 5,100 more rows<br># i 2 more variables: smoking_status <chr>, stroke <dbl><br># i Use `print(n = ...)` to see more rows | Declare a stroke variable to the read dataset file from the directory folder and using the print function show the dataset. |

**Attribute names:**

To get the dataset attribute using the names () function which also be set the object.

| Code | |
|---|---|
| names(stroke) | |
| **Output** | **Description** |
| [1] "id" "gender" "age" "hypertension" "heart_disease"<br>[6] "ever_married" "work_type" "Residence_type" "avg_glucose_level" "bmi"<br>[11] "smoking_status" "stroke" | This returns the all-attributes names for the dataset. |

**Summary**

To obtain different types of attributes.

| Code | |
|---|---|
| summary(stroke) | |
| **Output** | **Description** |
| id<br>Min. : 67<br>1st Qu.:17741<br>Median :36932<br>Mean :36518<br>3rd Qu.:54682<br>Max. :72940<br>gender<br>Length:5110<br>Class :character<br>Mode :character<br>age<br>Min. : 0.08<br>1st Qu.:25.00<br>Median :45.00<br>Mean :43.23<br>3rd Qu.:61.00<br>Max. :82.00<br>hypertension<br>Min. :0.00000<br>1st Qu.:0.00000<br>Median :0.00000<br>Mean :0.09746<br>3rd Qu.:0.00000<br>Max. :1.00000<br>heart_disease<br>Min. :0.00000<br>1st Qu.:0.00000<br>Median :0.00000<br>Mean :0.05401<br>3rd Qu.:0.00000<br>Max. :1.00000<br>ever_married<br>Length:5110<br>Class :character<br>Mode :character<br>work_type<br>Length:5110<br>Class :character<br>Mode :character<br>Residence_type<br>Length:5110<br>Class :character<br>Mode :character<br>avg_glucose_level<br>Min. : 55.12<br>1st Qu.: 77.25<br>Median : 91.89<br>Mean :106.15<br>3rd Qu.:114.09<br>Max. :271.74<br>bmi<br>Length:5110<br>Class :character<br>Mode :character<br>smoking_status<br>Length:5110<br>Class :character<br>Mode :character<br>stroke<br>Min. :0.00000<br>1st Qu.:0.00000<br>Median :0.00000<br>Mean :0.04873<br>3rd Qu.:0.00000<br>Max. :1.00000 | This will provide a summary of each attribute of dataset stroke including measures minimum, 1st quantile, 3rd quantile, median, mean and maximum for numerical attributes as count for factors. |

**Dataset structure:**

To display the structure of an R objects, use the str(stroke).

| Code | |
|---|---|
| str(stroke) | |
| **Output** | **Description** |
| tibble [5,110 x 12] (S3: tbl_df/tbl/data.frame)<br>$ id : num [1:5110] 9046 51676 31112 60182 1665 ...<br>$ gender : chr [1:5110] "Male" "Female" "Male" "Female" ...<br>$ age : num [1:5110] 67 61 80 49 79 81 74 69 59 78 ...<br>$ hypertension : num [1:5110] 0 0 0 0 1 0 1 0 0 0 ...<br>$ heart_disease : num [1:5110] 1 0 1 0 0 0 1 0 0 0 ...<br>$ ever_married : chr [1:5110] "Yes" "Yes" "Yes" "Yes" ...<br>$ work_type : chr [1:5110] "Private" "Self-employed" "Private" "Private" ...<br>$ Residence_type : chr [1:5110] "Urban" "Rural" "Rural" "Urban" ...<br>$ avg_glucose_level: num [1:5110] 229 202 106 171 174 ...<br>$ bmi : chr [1:5110] "36.6" "N/A" "32.5" "34.4" ...<br>$ smoking_status : chr [1:5110] "formerly smoked" "never smoked" "never smoked" "smokes" ...<br>$ stroke : num [1:5110] 1 1 1 1 1 1 1 1 1 1 ...<br>> | This str will be show the data types and some values of the dataset attributes. |

**Conversion:**

BMI has some N/A which are contain in categorized. BMI is numeric value for Body Mass Index. Measure of body fat based on height and weight.

**Code:**

stroke$bmi<- as.integer(stroke$bmi) stroke$hypertension<-as.factor(stroke$hypertension) stroke$heart_disease<-as.factor(stroke$heart_disease) stroke$stroke<-as.factor(stroke$stroke)

**Description:**

Convert the BMI attribute char to integer. Convert the hypertension, heart_disease and stroke attribute numerical data to factor level.

**Missing value:**

Check is the dataset has any missing instance or not.

| Code | |
|---|---|
| colSums(is.na(stroke)) | |
| **Output** | **Description** |
| id 0<br>gender 0<br>age 0<br>hypertension 0<br>heart_disease 0<br>ever_married 0<br>work_type 0<br>Residence_type 0<br>avg_glucose_level 0<br>bmi 201<br>smoking_status 0<br>stroke 0 | On this dataset bmi attribute has 201 missing values found. |

**Handle missing values:**

For the bmi attribute missing values are handle using measure of central tendency: mean

| Code | |
|---|---|
| stroke$bmi[is.na(stroke$bmi)] <- mean(stroke$bmi, na.rm = TRUE)<br>colSums(is.na(stroke)) | |
| **Output** | **Description** |
| id 0<br>gender 0<br>age 0<br>hypertension 0<br>heart_disease 0<br>ever_married 0<br>work_type 0<br>Residence_type 0<br>avg_glucose_level 0<br>bmi 0<br>smoking_status 0<br>stroke 0 | Replace all missing value using the mean. |

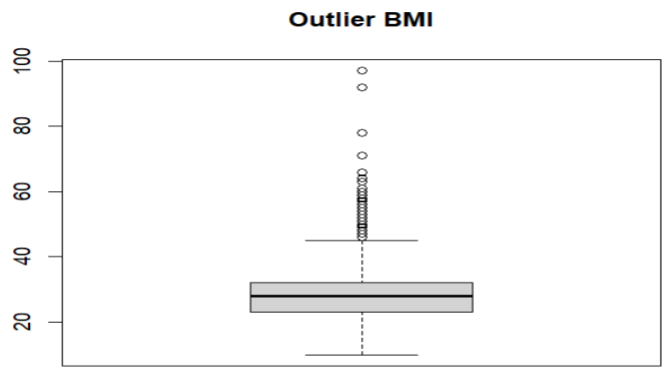
**Noisy value:**

Now since outliers are an issue terms of getting accurate results from this dataset, we remove them and replace them with the mean values of those attribute columns.

**Box plot:**

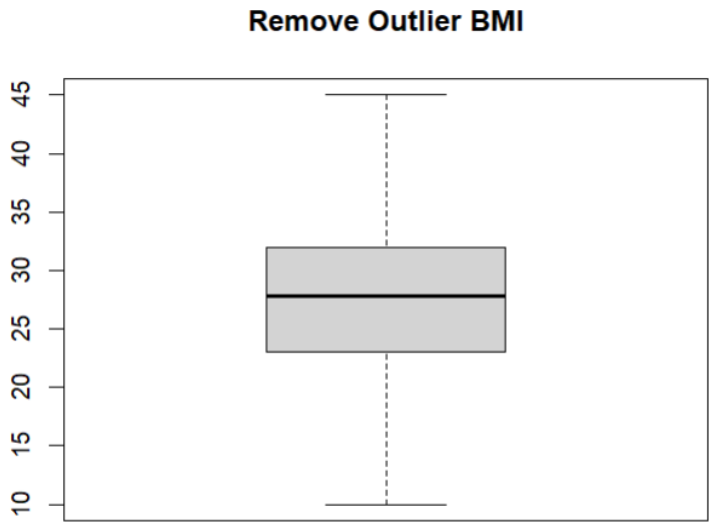
To Finding the outliers first using the box-plot graphical representation. After that remove the outliers.

**BMI**

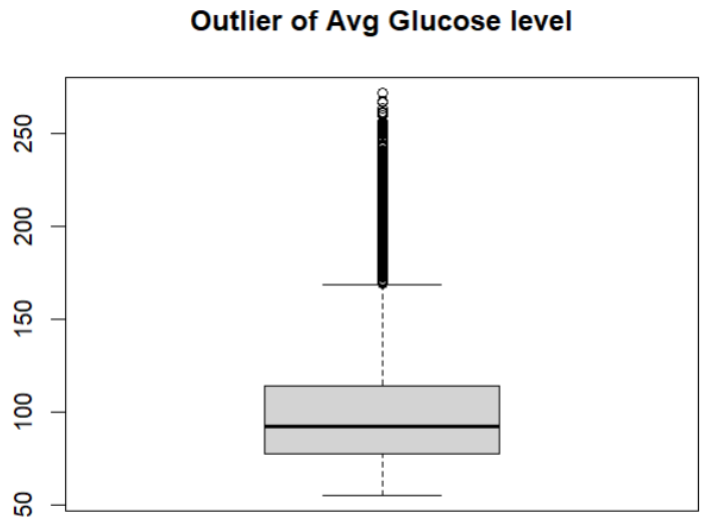
| Code | |
|---|---|
| boxplot(stroke$bmi,main="Outlier BMI") | |
| **Output** | **Description** |
|  | Generates a box plot for the 'bmi' attribute in the dataset, providing a visual representation of its distribution and identifying potential outliers. |

**Handle BMI outlier**

| Code |
|---|
| q1<-quantile(stroke$bmi,0.25)<br>q3<-quantile(stroke$bmi,0.75)<br>iqr<-q3-q1<br>lower<- q1-1.5*iqr<br>upper<- q3+1.5*iqr<br>outliers_bmi<- stroke$bmi < lower \| stroke$bmi > upper<br>stroke$bmi <- ifelse(outliers_bmi,NA,stroke$bmi)<br>colSums(is.na(stroke))<br>stroke$bmi[is.na(stroke$bmi)]<-mean(stroke$bmi,na.rm = TRUE)<br>boxplot(stroke$bmi, main='Remove Outlier BMI') |

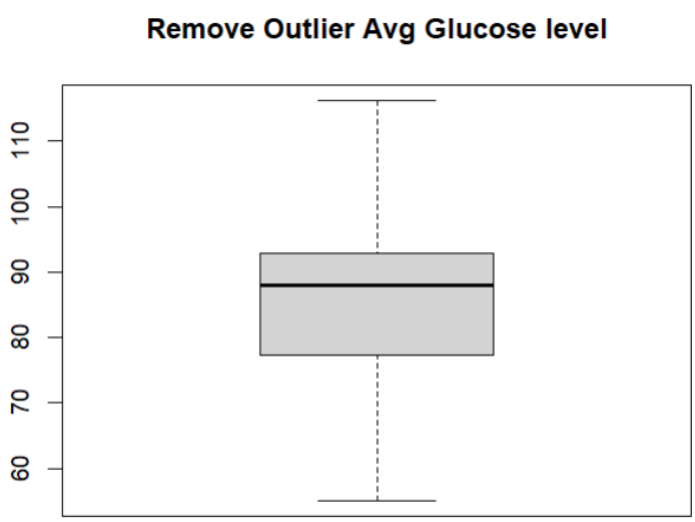
| Output | Description |
|---|---|
|  | To handle the outliers, computes the first quartile (q1), third quartile (q3), and interquartile range (iqr) for 'bmi', identifies outliers based on a iqr criterion, and replaces those outliers with NA, then replace the NA values with the mean of the 'bmi' attribute. |

**Average glucose level**

| Code | |
|---|---|
| boxplot(stroke$avg_glucose_level,main="Outlier of Avg Glucose level") | |
| **Output** | **Description** |
|  | Generates a box plot for the ' avg_glucose_level ' attribute in the dataset, providing a visual representation of its distribution and identifying potential outliers. |

**Handle average glucose level**

| Code |
|---|
| q1 <- quantile(stroke$avg_glucose_level,0.25)<br>q3 <- quantile(stroke$avg_glucose_level,0.75)<br>iqr <- q3 - q1<br>lower <- q1-1.5*iqr<br>upper<- q3+1.5*iqr<br>outliers_avg_glucose_level<- stroke$avg_glucose_level < lower \| stroke$avg_glucose_level > upper<br>stroke$avg_glucose_level <- ifelse(outliers_avg_glucose_level,NA,stroke$avg_glucose_level)<br>colSums(is.na(stroke))<br>stroke$avg_glucose_level[is.na(stroke$avg_glucose_level)]<-mean(stroke$avg_glucose_level,na.rm = TRUE)<br>colSums(is.na(stroke))<br>boxplot(stroke$avg_glucose_level,main="Remove Outlier Avg Glucose level") |

| Output | Description |
|---|---|
|  | To handle the outliers, computes the first quartile (q1), third quartile (q3), and interquartile range (iqr) for 'avg_glucose_level', identifies outliers based on a iqr criterion, and replaces those outliers with NA, then replace the NA values with the mean of the 'avg_glucose_level' attribute. |

## Analysis Technique

**Statistical correlation:**

Using correlation technique Pearson correlation test for find the significant attribute along with target attribute. If it didn't find any significant there will be removed the attribute from the dataset.

**Code**

```
Pearsons_Chi_squared<- function(attribute){
  count_instances<- table(stroke[[attribute]],stroke$stroke)
  p_value<- chisq.test(count_instances)
  return(p_value)
}
pearson_corelation_id <- Pearsons_Chi_squared('id')
pearson_corelation_gender <- Pearsons_Chi_squared('gender')
pearson_corelation_age <- Pearsons_Chi_squared('age')
pearson_corelation_hypertension <- Pearsons_Chi_squared('hypertension')
pearson_corelation_heart_disease <- Pearsons_Chi_squared('heart_disease')
pearson_corelation_ever_married <- Pearsons_Chi_squared('ever_married')
pearson_corelation_work_type <- Pearsons_Chi_squared('work_type')
pearson_corelation_Residence_type <- Pearsons_Chi_squared('Residence_type')
pearson_corelation_smoking_status <- Pearsons_Chi_squared('smoking_status')
pearson_corelation_bmi <- Pearsons_Chi_squared('bmi')
pearson_corelation_avg_glucose_level <- Pearsons_Chi_squared('avg_glucose_level')
pearson_corelation_stroke <- Pearsons_Chi_squared('stroke')
if(pearson_corelation_id$p.value<0.05){
  print("There is a significant relationship between id and stroke.")
}else{
  stroke<- subset(stroke,select = -id)
  print("No significant relationship found.")
}


if(pearson_corelation_gender$p.value<0.05){
  print("There is a significant relationship between gender and stroke.")
```

```
}else{
  stroke<- subset(stroke,select = -gender)
  print("No significant relationship found.")
}

if(pearson_corelation_age$p.value<0.05){
  print("There is a significant relationship between age and stroke.")
}else{
  stroke<- subset(stroke,select = -age)
  print("No significant relationship found.")
}

if(pearson_corelation_hypertension$p.value<0.05){
  print("There is a significant relationship between hypertension and stroke.")
}else{
  stroke<- subset(stroke,select = -hypertension)
  print("No significant relationship found.")
}

if(pearson_corelation_heart_disease$p.value<0.05){
  print("There is a significant relationship between heart_disease and stroke.")
}else{
  stroke<- subset(stroke,select = -heart_disease)
  print("No significant relationship found.")
}

if(pearson_corelation_ever_married$p.value<0.05){
  print("There is a significant relationship between ever married and stroke.")
}else{
  stroke<- subset(stroke,select = -ever_married)
  print("No significant relationship found.")
```

```
}
if(pearson_corelation_work_type$p.value<0.05){
  print("There is a significant relationship between work type and stroke.")
}else{
  stroke<- subset(stroke,select = -work_type)
  print("No significant relationship found.")
}
if(pearson_corelation_Residence_type$p.value<0.05){
  print("There is a significant relationship between residence type and stroke.")
}else{
  stroke<- subset(stroke,select = -Residence_type)
  print("No significant relationship found.")
}
if(pearson_corelation_smoking_status$p.value<0.05){
  print("There is a significant relationship between smoking status and stroke.")
}else{
  stroke<- subset(stroke,select = -smoking_status)
  print("No significant relationship found.")
}
if(pearson_corelation_bmi$p.value<0.05){
  print("There is a significant relationship between bmi and stroke.")
}else{
  stroke<- subset(stroke,select = -bmi)
  print("No significant relationship found.")
}
if(pearson_corelation_avg_glucose_level$p.value<0.05){
  print("There is a significant relationship between avg glucose level and stroke.")
}else{
  stroke<- subset(stroke,select = -avg_glucose_level)
  print("No significant relationship found.")
}
```

```
if(pearson_corelation_stroke$p.value<0.05){
  print("There is a significant relationship between stroke  and stroke.")
}
else{
  stroke<- subset(stroke,select = -stroke)
  print("No significant relationship found.")
}
```

| Output | Description |
|---|---|
| `> pearson_corelation_stroke <- Pearsons_Chi_squared('stroke')`<br>`> if(pearson_corelation_id$p.value<0.05){`<br>`+  print("There is a significant relationship between id and stroke.")`<br>`+ }else{`<br>`+  stroke<- subset(stroke,select = -id)`<br>`+  print("No significant relationship found.")`<br>`+ }`<br>`[1] "No significant relationship found."`<br>`> if(pearson_corelation_gender$p.value<0.05){`<br>`+  print("There is a significant relationship between gender and stroke.")`<br>`+ }else{`<br>`+  stroke<- subset(stroke,select = -gender)`<br>`+  print("No significant relationship found.")`<br>`+ }`<br>`[1] "No significant relationship found."`<br>`> if(pearson_corelation_age$p.value<0.05){`<br>`+  print("There is a significant relationship between age and stroke.")`<br>`+ }else{`<br>`+  stroke<- subset(stroke,select = -age)`<br>`+  print("No significant relationship found.")`<br>`+ }`<br>`[1] "There is a significant relationship between age and stroke."`<br>`> if(pearson_corelation_hypertension$p.value<0.05){`<br>`+  print("There is a significant relationship between hypertension and stroke.")`<br>`+ }else{`<br>`+  stroke<- subset(stroke,select = -hypertension)`<br>`+  print("No significant relationship found.")`<br>`+ }`<br>`[1] "There is a significant relationship between hypertension and stroke."`<br>`> if(pearson_corelation_heart_disease$p.value<0.05){`<br>`+  print("There is a significant relationship between heart_disease and stroke.")`<br>`+ }else{`<br>`+  stroke<- subset(stroke,select = -heart_disease)`<br>`+  print("No significant relationship found.")`<br>`+ }`<br>`[1] "There is a significant relationship between heart_disease and stroke."` | Here create a function pearson_corelation_stroke function using the Pearson correlation for significant relationship. Check the condition with p value if the p value less 0.05 then there will be significant relationship otherwise no significant found. |

**Data splitting:**

| Code |
|---|
| set.seed(123)<br>split_Index<- createDataPartition(stroke$stroke, p = 0.8, list = FALSE)<br>train_data<- stroke[split_Index, ]<br>test_data <- stroke[-split_Index, ] |

| Output | Description |
|---|---|
| train data:<br><br>```# A tibble: 4,089 × 8<br>age hypertension heart_disease ever_married work_type bmi smoking_status stroke<br><dbl> <fct> <fct> <chr> <chr> <dbl> <chr> <fct><br>1 67 0 1 Yes Private 36 formerly smoked 1<br>2 80 0 1 Yes Private 32 never smoked 1<br>3 49 0 0 Yes Private 34 smokes 1<br>4 79 1 0 Yes Self-employed 24 never smoked 1<br>5 81 0 0 Yes Private 29 formerly smoked 1<br>6 74 1 1 Yes Private 27 never smoked 1<br>7 69 0 0 No Private 22 never smoked 1<br>8 59 0 0 Yes Private 28.4 Unknown 1<br>9 78 0 0 Yes Private 24 Unknown 1<br>10 81 1 0 Yes Private 29 never smoked 1<br># ℹ 4,079 more rows<br># ℹ Use `print(n = ...)` to see more rows```<br><br>test data:<br><br>```# A tibble: 1,021 × 8<br>age hypertension heart_disease ever_married work_type bmi smoking_status stroke<br><dbl> <fct> <fct> <chr> <chr> <dbl> <chr> <fct><br>1 61 0 0 Yes Self-employed 28.4 never smoked 1<br>2 79 0 1 Yes Private 28 never smoked 1<br>3 50 1 0 Yes Self-employed 30 never smoked 1<br>4 60 0 0 No Private 37 never smoked 1<br>5 57 0 1 No Govt_job 28.4 Unknown 1<br>6 52 1 0 Yes Self-employed 27.8 never smoked 1<br>7 82 0 1 Yes Private 32 Unknown 1<br>8 69 0 1 Yes Self-employed 28 smokes 1<br>9 72 1 0 Yes Private 23 formerly smoked 1<br>10 76 1 0 Yes Private 33 never smoked 1<br># ℹ 1,011 more rows<br># ℹ Use `print(n = ...)` to see more rows```<br><br>This process helps in creating a random and representative split of the dataset for training and testing a predictive model. The list = FALSE argument ensures that the output is a vector of indices rather than a list. |

**Naïve bayes model:**

Naïve bayes model using for applied a predicting target data from the test data.

| Code |
|---|
| nb_model <- naiveBayes(stroke ~ ., data = train_data)<br><br>predictions <- predict(nb_model, test_data)<br><br>table(predictions) |

| Output | Description |
|---|---|
| ```predictions<br>0 1<br>974 47```<br><br>Here predict the test data using the naïve bayes classification among the train data. |

**Accuracy:** calculate the naïve bayes predictive accuracy using the divided the data into training and test set.

| Code |
|---|
| accuracy <- sum(predictions == test_data$stroke) / nrow(test_data)<br><br>cat("Naive Bayes Accuracy:", round(accuracy,2), "\n") |

| Output | Description |
|---|---|
| Naive Bayes Accuracy: 0.93 | The naïve bayes predictive accuracy of training ang test set is 0.93. the round operation can configure the value in 2 digits. |

**Confusion Matrix:**

To generate a confusion matrix for dataset using naïve bayes classification and calculate the recall, precision and f-measure of the classifier.

| Code |
|---|
| confusion_matrix<- table(predictions,test_data$stroke)<br>recall_matrix<- confusion_matrix[2,2]/sum(confusion_matrix[2,])<br>precision_matrix<- confusion_matrix[2,2]/sum(confusion_matrix[,2])<br>f_measure <- 2 * (precision_matrix * recall_matrix) / (precision_matrix + recall_matrix)<br>cat("Confusion Matrix: ",confusion_matrix,"\n")<br>cat("Recall: ",recall_matrix_second_class,"\n")<br>cat("Precision: ",precision_matrix_second_class,"\n")<br>cat("f-measure: ",f_measure,"\n") |

| Output | Description |
|---|---|
| > cat("Confusion Matrix: ",confusion_matrix,"\n")<br>Confusion Matrix:  936 36 38 11<br>> cat("Recall: ",recall_matrix_second_class,"\n")<br>Recall:  0.2340426<br>> cat("Precision: ",precision_matrix_second_class,"\n")<br>Precision:  0.962963<br>> cat("f-measure: ",f_measure,"\n")<br>f-measure:  0.2291667 | The output shown the calculated confusion matrix, recall, precision and f-measure value of classifier. |