AMERICAN INTERNATIONAL UNIVERSITY-BANGLADESH

Faculty of Science and Technology

Project Cover Page

Assignment Title:	FINALTERM PROJECT		
Assignment No:	02	Date of Submission:	25 December 2023
Course Title:	INTRODUCTION TO DATA SCIENCE		
Course Code:	CSC 4180	Section:	A
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 orfrom 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 collaboration has been authorized by the concerned teacher and is clearly acknowledged in the assignment.
4. I/we have not previously submitted or am 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 that Plagiarism 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 the 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 copymy/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.:03

No	Name	ID	Program	Signature
1	SHAGOR,MD RIHAN UDDIN	20-42909-1	BSc [CSE]	
2	THAJIB,NASIR	20-43247-1	BSc[CSE]	

Faculty use only		
FACULTYCOMMENTS	Marks Obtained	
	Total Marks	

Project Description :

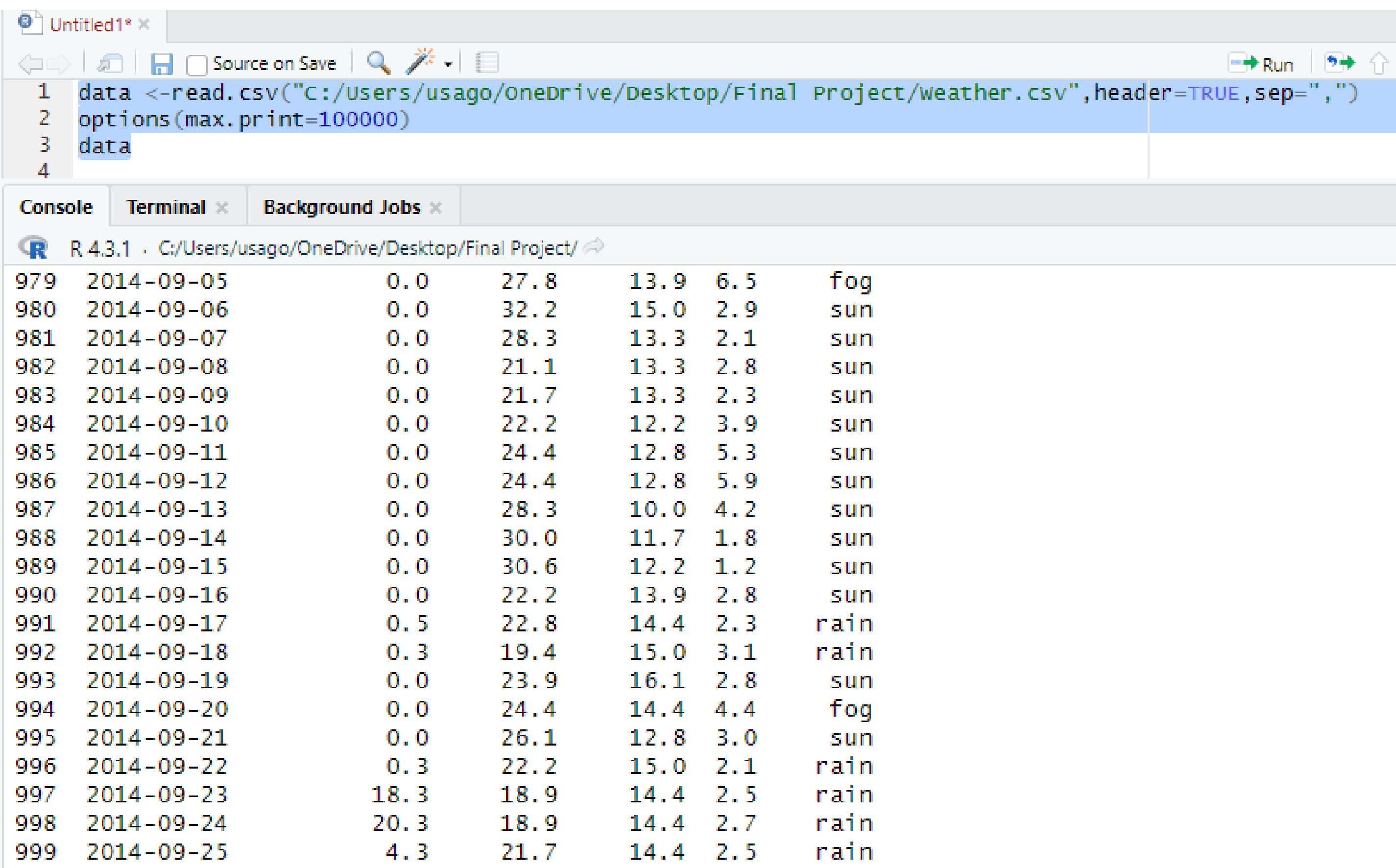
A through database of the elements that lead to predict whether it “drizzle" or "rain" or "sun" or "snow" or "fog . There are five fields for input fields and one field for an output field. date, precipitation ,temp_min (minimum temperature for each date), temp_max (maximum temperature for each date) and wind are representing the input fields, while the output field predict the weather, which is divided into five categories ("drizzle" "rain" "sun" "snow" "fog"); We will try to recover the missing value (if there any) and prepare dataset then apply naïve bayes algorithm for the dataset and calculate about the predictive accuracy univariate analysis .

Data Preparation

Importing dataset from csv file:

Code :

```
data <-read.csv("C:/Users/usago/OneDrive/Desktop/Final Project/Weather.csv",header=TRUE,sep=",")
options(max.print=100000)
data
```



```
Untitled1*
Source on Save                                                   Run
1 data <-read.csv("C:/Users/usago/OneDrive/Desktop/Final Project/Weather.csv",header=TRUE,sep=",")
2 options(max.print=100000)
3 data
4
Console   Terminal   Background Jobs
R 4.3.1 · C:/Users/usago/OneDrive/Desktop/Final Project/
979   2014-09-05    0.0   27.8   13.9  6.5   fog
980   2014-09-06    0.0   32.2   15.0  2.9   sun
981   2014-09-07    0.0   28.3   13.3  2.1   sun
982   2014-09-08    0.0   21.1   13.3  2.8   sun
983   2014-09-09    0.0   21.7   13.3  2.3   sun
984   2014-09-10    0.0   22.2   12.2  3.9   sun
985   2014-09-11    0.0   24.4   12.8  5.3   sun
986   2014-09-12    0.0   24.4   12.8  5.9   sun
987   2014-09-13    0.0   28.3   10.0  4.2   sun
988   2014-09-14    0.0   30.0   11.7  1.8   sun
989   2014-09-15    0.0   30.6   12.2  1.2   sun
990   2014-09-16    0.0   22.2   13.9  2.8   sun
991   2014-09-17    0.5   22.8   14.4  2.3  rain
992   2014-09-18    0.3   19.4   15.0  3.1  rain
993   2014-09-19    0.0   23.9   16.1  2.8   sun
994   2014-09-20    0.0   24.4   14.4  4.4   fog
995   2014-09-21    0.0   26.1   12.8  3.0   sun
996   2014-09-22    0.3   22.2   15.0  2.1  rain
997   2014-09-23   18.3   18.9   14.4  2.5  rain
998   2014-09-24   20.3   18.9   14.4  2.7  rain
999   2014-09-25    4.3   21.7   14.4  2.5  rain
1000  2014-09-26    8.9   20.0   13.9  3.3  rain
```

Description: In order to get a . csv file into R, we can use read. csv, and as the only argument, put the path to the file we want to read in within quotation marks.

Taking 500 instance :

Code :

mydata <- head(data, 500)

mydata

```
mydata <- head(data, 500)
mydata
```

Console | Terminal | Background Jobs

R 4.3.1 · ~/

```
478 2013-04-22      0.0   16.1    5.0  4.3   sun
479 2013-04-23      0.0   17.8    3.9  2.8   sun
480 2013-04-24      0.0   21.1    6.1  3.0   sun
481 2013-04-25      0.0   21.7    6.7  1.1   sun
482 2013-04-26      0.0   20.6    8.3  2.2   fog
483 2013-04-27      0.0   13.9   10.6  5.9   sun
484 2013-04-28      1.0   15.0    9.4  5.2  rain
485 2013-04-29      3.8   13.9    6.7  4.2  rain
486 2013-04-30      0.0   12.8    4.4  2.4   sun
487 2013-05-01      0.0   18.3    3.3  3.1   sun
488 2013-05-02      0.0   20.6    6.7  4.0   sun
489 2013-05-03      0.0   21.7    9.4  4.9   sun
490 2013-05-04      0.0   25.0   11.1  6.5   sun
491 2013-05-05      0.0   28.9   11.7  5.3   sun
492 2013-05-06      0.0   30.6   12.2  2.0   sun
493 2013-05-07      0.0   20.6   11.1  3.3   sun
494 2013-05-08      0.0   19.4   11.1  1.9   sun
495 2013-05-09      0.0   22.8   10.0  1.3   sun
496 2013-05-10      0.0   26.1    9.4  1.0   sun
497 2013-05-11      0.0   27.2   12.2  2.6   sun
498 2013-05-12      6.6   21.7   13.9  3.9  rain
499 2013-05-13      3.3   18.9    9.4  5.0  rain
500 2013-05-14      0.0   18.3    7.8  2.4   sun
```

Description: We have taken 500 instance from our data using head(data,500) .

Attribute name of dataset :

Code :

names(mydata)

```

mydata <- head(data, 1000)
mydata
names(mydata)

```

```
996  2014-09-22          0.3     22.2     15.0  2.1    rain
997  2014-09-23         18.3     18.9     14.4  2.5    rain
998  2014-09-24         20.3     18.9     14.4  2.7    rain
999  2014-09-25          4.3     21.7     14.4  2.5    rain
1000 2014-09-26          8.9     20.0     13.9  3.3    rain
> names(mydata)
[1] "date"          "precipitation" "temp_max"      "temp_min"      "wind"          "weather"
>
```

Description: To know the names of each field in our data set we can use name() function .

Checking Missing value:

Code :

colsums(is.na(mydata))

```
names(mydata)

colsums(is.na(mydata))
```

```
> colsums(is.na(mydata))
         date precipitation      temp_max      temp_min          wind       weather
            0             0             0             0             0             0
>
```

Description: We checked all the missing value using colsums(is.na()) . There is no missing value in our data set .

Variable types in dataset:

Code :

str(mydata)

```
str(mydata)
```

```
> str(mydata)
'data.frame':   1000 obs. of  6 variables:
 $ date         : chr  "2012-01-01" "2012-01-02" "2012-01-03" "2012-01-04" ...
 $ precipitation: num  0 10.9 0.8 20.3 1.3 2.5 0 0 4.3 1 ...
 $ temp_max     : num  12.8 10.6 11.7 12.2 8.9 4.4 7.2 10 9.4 6.1 ...
 $ temp_min     : num  5 2.8 7.2 5.6 2.8 2.2 2.8 2.8 5 0.6 ...
 $ wind         : num  4.7 4.5 2.3 4.7 6.1 2.2 2.3 2 3.4 3.4 ...
 $ weather      : chr  "drizzle" "rain" "rain" "rain" ...
>
```

Description: By using str function we can get information about variable types in the dataset.

Checking unique values in target attribute(weather):

Code :

unique_values <- unique(mydata$weather)

unique_values

```
unique_values <- unique(mydata$weather)
unique_values
```

```
Console   Terminal   Background Jobs
R 4.3.1 · C:/Users/usago/OneDrive/Desktop/Final Project/
> unique_values <- unique(mydata$weather)
> unique_values
[1] "drizzle" "rain"    "sun"     "snow"    "fog"
>
```

Description: By using unique function, we can check unique values in dataset. We have five unique values in our target attribute.

Labeling weather column values & numeric conversation:

Code :

mydata$weather<-factor(mydata$weather,levels = c("rain", "drizzle","sun","snow","fog"),labels = c(1,2,3,4,5))

mydata$weather <- as.numeric(mydata$weather)

```
mydata$weather<-factor(mydata$weather,levels = c("rain", "drizzle","sun","snow","fog"),labels = c(1,2,3,4,5))
mydata$weather <- as.numeric(mydata$weather)
mydata$weather
```

```
Console   Terminal ×   Background Jobs ×
R 4.3.1 · C:/Users/usago/OneDrive/Desktop/Final Project/
[1] "drizzle" "rain"    "sun"     "snow"    "fog"
> mydata$weather<-factor(mydata$weather,levels = c("rain", "drizzle","sun","snow","fog"),labels = c(1,2,3,4,5))
> mydata$weather
  [1] 2 1 1 1 1 1 1 3 1 1 3 3 3 4 4 4 4 4 4 4 1 1 1 1 1 1 2 1 1 1 1 1 3 3 3 3 3 1 1 1 1 1 1 1 1 2 1 1 1 3 1 1 1 3
 [55] 1 1 4 3 4 4 3 1 3 1 1 4 3 3 1 1 1 4 4 1 4 1 4 1 1 1 1 1 3 3 1 2 1 1 1 1 1 1 3 1 3 4 1 3 3 3 1 1 1 2 3 1 1 1
[109] 1 1 1 3 1 3 1 1 1 1 2 1 1 1 1 1 1 3 3 3 3 1 3 3 3 3 3 2 3 1 1 3 1 1 1 1 1 1 3 3 1 3 1 1 1 1 3 1 1 3 1 1 1 3
[163] 1 1 3 3 3 1 3 1 1 3 3 1 1 2 1 1 3 1 1 1 1 1 1 3 2 3 3 1 1 2 5 2 1 1 1 1 3 3 3 1 3 1 1 3 3 2 2 2 3 3 3 2 3 3
[217] 3 3 1 2 3 2 3 3 3 3 3 3 3 3 2 2 3 1 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 1 1 3 3 3 3 3 3 5 3 2 2 2 1 5 5 3 2
[271] 2 1 3 3 3 3 3 3 3 3 3 3 2 2 2 1 1 1 1 3 3 1 1 1 1 1 1 1 3 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 3 1 1 1 1 2 1 1 1 1
[325] 1 1 1 1 1 2 5 3 1 1 1 1 1 1 1 1 1 1 3 1 1 1 1 1 1 4 4 1 4 4 1 1 1 1 1 4 1 1 1 1 2 2 3 3 1 1 1 1 1 1 1 4 2 3
[379] 3 3 3 2 2 2 2 2 2 2 1 1 1 1 1 1 1 1 1 1 2 1 1 1 1 1 3 1 2 1 1 1 1 2 1 1 1 3 1 1 1 1 1 1 1 1 1 1 1 3 3 1 1 1
[433] 2 5 1 1 1 1 1 1 1 3 1 1 1 4 3 3 3 3 3 1 1 1 2 3 3 3 3 1 1 1 1 1 3 1 1 1 1 1 5 1 2 1 1 3 1 3 3 3 3 5 3 1 1 3
[487] 3 3 3 3 3 3 3 3 3 3 3 1 1 3 1 5 1 3 3 3 1 1 1 1 3 1 1 1 1 3 3 3 1 3 3 3 3 3 3 3 3 3 1 3 3 3 3 3 1 3 1 1 3 1
[541] 1 1 1 1 3 3 3 3 3 3 5 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 5 3 3 3 3 3 5 3 3 3 3 1 5 3 3 3 3 3 3 1 3 3 3 1 1 5
[595] 3 3 3 3 3 3 3 3 1 1 1 1 1 3 3 3 3 1 1 1 1 3 5 3 3 3 3 3 5 1 1 3 3 3 1 3 1 1 3 1 3 1 1 1 1 1 1 1 3 3 1 1 1 3
```

Description: We have labeled weather column 1 for rain,2 for drizzle , 3 for sun, 4 for snow and 5 for fog . Then we have converted to numeric column for our future purpose.

Convert date column values sequent and numeric type :

Code :

mydata$date <- seq_along(mydata$date)

mydata$date <- as.numeric(mydata$date)

mydata$date

```
mydata$date <- seq_along(mydata$date)
mydata$date <- as.numeric(mydata$date)
mydata$date
```

```
Console   Terminal ×   Background Jobs ×
R 4.3.1 · C:/Users/usago/OneDrive/Desktop/Final Project/
> mydata$date <- seq_along(mydata$date)
> mydata$date <- as.numeric(mydata$date)
> mydata$date
  [1]   1   2   3   4   5   6   7   8   9  10  11  12  13  14  15  16  17  18  19  20  21
 [22]  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40  41  42
 [43]  43  44  45  46  47  48  49  50  51  52  53  54  55  56  57  58  59  60  61  62  63
 [64]  64  65  66  67  68  69  70  71  72  73  74  75  76  77  78  79  80  81  82  83  84
 [85]  85  86  87  88  89  90  91  92  93  94  95  96  97  98  99 100 101 102 103 104 105
[106] 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126
[127] 127 128 129 130 131 132 133 134 135 136 137 138 139 140 141 142 143 144 145 146 147
[148] 148 149 150 151 152 153 154 155 156 157 158 159 160 161 162 163 164 165 166 167 168
[169] 169 170 171 172 173 174 175 176 177 178 179 180 181 182 183 184 185 186 187 188 189
[190] 190 191 192 193 194 195 196 197 198 199 200 201 202 203 204 205 206 207 208 209 210
[211] 211 212 213 214 215 216 217 218 219 220 221 222 223 224 225 226 227 228 229 230 231
[232] 232 233 234 235 236 237 238 239 240 241 242 243 244 245 246 247 248 249 250 251 252
```

Description: In our dataset date column is in date format which create difficulty for applying co relation technic so for our better outcome we have replaced the date values date format to each date sequentially using seq_along() function . Then we converted it to numeric values.

<u>Pearson's Correlation :</u>

Code :

co_relation<-(cor(mydata[,c('date','precipitation','temp_min','temp_max','wind','weather')]))

co_relation

```
co_relation<-(cor(mydata[,c('date','precipitation','temp_min','temp_max','wind','weather')]))
co_relation
```

```
Console   Terminal ×   Background Jobs ×
R 4.3.1 · ~/
> co_relation
                       date precipitation     temp_min      temp_max        wind      weather
date           1.0000000000   -0.01135756  0.025122367  0.0004422434 -0.06543744 -0.033175444
precipitation -0.0113575618    1.00000000 -0.102307172 -0.2546402073  0.25768206 -0.249862903
temp_min       0.0251223673   -0.10230717  1.000000000  0.8529552351 -0.04998485  0.002183174
temp_max       0.0004422434   -0.25464021  0.852955235  1.0000000000 -0.15234978  0.270371165
wind          -0.0654374374    0.25768206 -0.049984849 -0.1523497768  1.00000000 -0.145108330
weather       -0.0331754442   -0.24986290  0.002183174  0.2703711645 -0.14510833  1.000000000
> 
```

Description: Pearson's co relation is a technic of measure the continues value of relation between two column. By using **cor()** function we can get the co relation value to check whether it is significant or not.

From our co relation value of the data set we have identified that **date** column have no relation with any of the attribute & all of the co relation value compare to **date** attribute is close to 0 . **date** attribute is non-significant attribute for the dataset , except that all the attribute is **significant attribute** for the column .

<u>Removing non-significant attribute:</u>

Code :

mydata1<-mydata

mydata1$date<- NULL

mydata1

```
mydata1<-mydata
mydata1$date<- NULL
mydata1
```

```
Console   Terminal ×   Background Jobs ×
R 4.3.1 · ~/
> mydata<-mydata1
Error: object 'mydata1' not found
> mydata1<-mydata
> mydata1$date<- NULL
> mydata1
   precipitation temp_max temp_min wind weather
1            0.0     12.8      5.0  4.7       2
2           10.9     10.6      2.8  4.5       1
3            0.8     11.7      7.2  2.3       1
4           20.3     12.2      5.6  4.7       1
5            1.3      8.9      2.8  6.1       1
6            2.5      4.4      2.2  2.2       1
7            0.0      7.2      2.8  2.3       1
8            0.0     10.0      2.8  2.0       3
9            4.3      9.4      5.0  3.4       1
10           1.0      6.1      0.6  3.4       1
11           0.0      6.1     -1.1  5.1       3
12           0.0      6.1     -1.7  1.9       3
13           0.0      5.0     -2.8  1.3       3
14           4.1      4.4      0.6  5.3       4
15           5.3      1.1     -3.3  3.2       4
16           2.5      1.7     -2.8  5.0       4
17           8.1      3.3      0.0  5.6       4
18          19.8      0.0     -2.8  5.0       4
19          15.2     -1.1     -2.8  1.6       4
20          13.5      7.2     -1.1  2.3       4
21           3.0      8.3      3.3  8.2       1
```

Description: Removed non-significant attribute **date.**

Labeling target attribute to categorical :

Code:

mydata2<-mydata1
mydata2$weather<-factor(mydata2$weather,levels = c(1,2,3,4,5),labels = c("rain", "drizzle","sun","snow","fog"))

```
mydata2<-mydata1
mydata2$weather<-factor(mydata2$weather,levels = c(1,2,3,4,5),labels = c("rain", "drizzle","sun","snow","fog"))
```

```
Console   Terminal ×   Background Jobs ×
R 4.3.1 · ~/
> mydata2
  precipitation temp_max temp_min wind weather
1           0.0     12.8      5.0  4.7 drizzle
2          10.9     10.6      2.8  4.5    rain
3           0.8     11.7      7.2  2.3    rain
4          20.3     12.2      5.6  4.7    rain
5           1.3      8.9      2.8  6.1    rain
6           2.5      4.4      2.2  2.2    rain
7           0.0      7.2      2.8  2.3    rain
```

Description:
labeled weather attribute to categorical again for understanding properly in naïve bayes

Applying Naïve Bayes classification

Install and load package for naivebayes:

Code:

install.packages("e1071")

library(e1071)

```
install.packages("e1071")
library(e1071)
```

Description:
Installed package “e1071” for using Naïve Bayes model and loaded this package using library() function

1.Dividing the data into training and test set:

Code:

set.seed(123)

sample_index <- sample(1:nrow(mydata2), 0.7 * nrow(mydata2))

train_data <- mydata[sample_index,]

test_data <- mydata[-sample_index,]

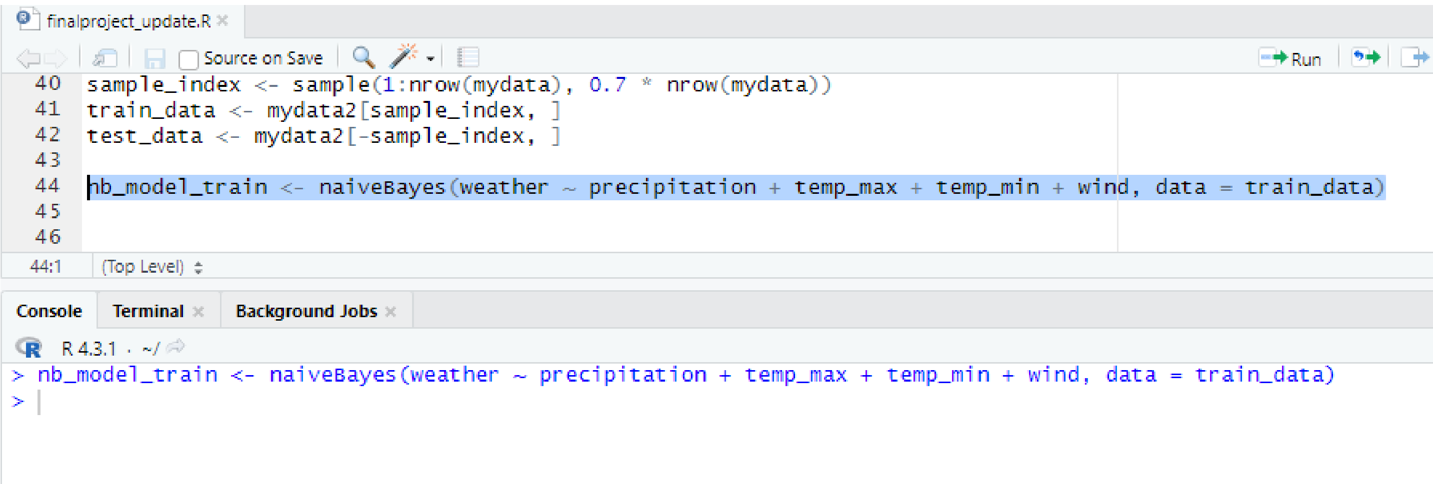
```
set.seed(123)
sample_index <- sample(1:nrow(mydata2), 0.7 * nrow(mydata2))
train_data <- mydata2[sample_index, ]
test_data <- mydata2[-sample_index, ]
```

Description :
Dataset divided into training data and test data . set.seed(123) is used to set the seed for the random number generator . sample() used to create a random sample of indices for splitting our dataset into a training set (70%) and a test set(rest 30%) .

Creating a Naive bayes model:

Code:

nb_model_train <- naiveBayes(weather ~ precipitation + temp_max + temp_min + wind, data = train_data)



Description :

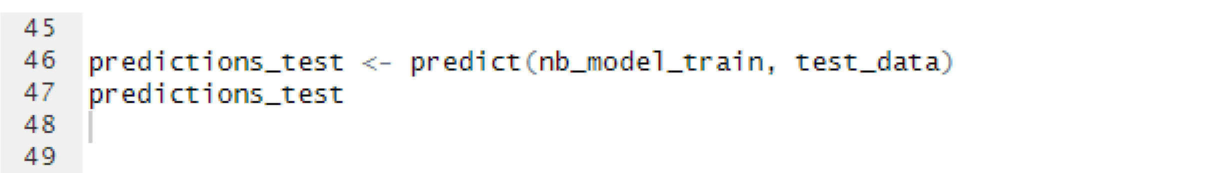
Created a Naive Bayes model using training data by using naiveBayes() function from “e1071” packages.

Prediction Test:

Code:

predictions_test <- predict(nb_model_train, test_data)

predictions_test



```
Console   Terminal ×   Background Jobs ×
R 4.3.1 · ~/
> nb_model_train <- naiveBayes(weather ~ precipitation + temp_max + temp_min + wind, data = train_data)
> predictions_test <- predict(nb_model_train, test_data)
> predictions_test
  [1] sun     rain    rain    drizzle rain    drizzle snow    snow    snow    rain    drizzle rain    sun    
 [14] rain    rain    drizzle rain    rain    sun     sun     drizzle snow    snow    rain    snow    sun    
 [27] rain    rain    rain    snow    rain    drizzle rain    sun     sun     sun     rain    rain    sun    
 [40] rain    rain    rain    sun     sun     sun     sun     sun     sun     rain    rain    sun     sun    
 [53] sun     rain    rain    sun     sun     sun     rain    sun     rain    rain    sun     sun     sun    
 [66] sun     sun     sun     rain    sun     sun     sun     sun     sun     sun     sun     sun     sun    
 [79] sun     sun     sun     sun     sun     sun     sun     sun     sun     rain    rain    rain    rain   
 [92] sun     rain    rain    sun     rain    rain    rain    rain    rain    rain    rain    rain    snow   
[105] snow    rain    rain    drizzle drizzle rain    drizzle drizzle drizzle drizzle drizzle rain    rain   
[118] rain    rain    sun     sun     rain    rain    rain    rain    rain    rain    rain    sun     rain   
[131] drizzle drizzle sun     rain    rain    rain    rain    sun     rain    rain    sun     sun     sun    
[144] rain    sun     sun     sun     rain    rain    sun    
Levels: rain drizzle sun snow fog
> 
```

Description :

By using predict(nb_model_train, test_data) we get the predicted values for each observation in our test set from the dataset. So we can compare them with the actual values to evaluate the performance of our model.

Accuracy Test:

Code:

accuracy_test <- sum(predictions_test == test_data$weather) / nrow(test_data)

accuracy_test

```
finalproject_update.R* ×
Source on Save                                                         Run
nb_model_train <- naiveBayes(weather ~ precipitation + temp_max + temp_min + wind, data = train

predictions_test <- predict(nb_model_train, test_data)
predictions_test

accuracy_test <- sum(predictions_test == test_data$weather) / nrow(test_data)
accuracy_test

49:1   (Top Level) ÷
```

```
Console   Terminal ×   Background Jobs ×
R 4.3.1 · ~/
> accuracy_test <- sum(predictions_test == test_data$weather) / nrow(test_data)
> accuracy_test
[1] 0.7466667
> accuracy_test <- sum(predictions_test == test_data$weather) / nrow(test_data)
```

Description:

We know, Accuracy=$\left(Number\ of\ Correct\ Prediction/\ Total\ Number\ of\ Predictions\right)$ .Here we calculate the sum of total correct prediction test which compare to actual data and divide by total number of test_data . We get accuracy 0.74 which means our model accuracy is **74%** .

Confusion Matrix for Naïve Bayes:

Code:

conf_matrix <- table(predictions_test, test_data$weather)

conf_matrix

```
accuracy_test

conf_matrix <- table(predictions_test, test_data$weather)
conf_matrix

```

Console Terminal Background Jobs

R 4.3.1 · ~/

```
> conf_matrix

predictions_test rain drizzle sun snow fog
         rain      63       0   0    3   0
         drizzle    1       5   9    0   0
         sun       12       7  38    0   3
         snow       3       0   0    6   0
         fog        0       0   0    0   0
>
```

Description:

By using confusion matrix we can get the result of actual and prediction result quantity. Example for predicted rain: 63 instance of the data set correctly predicted as rain 1, instance incorrectly predicted as drizzle, 12 instances incorrectly predicted as sun, 0 instances incorrectly predict for fog. So same for all the target value in the test_data$weather attribute .

TP, FP , FN and TN VALUES :

Code:

TP <- conf_matrix[2, 2]

FP <- conf_matrix[1, 2]

FN <- conf_matrix[2, 1]

TN <- conf_matrix[1, 1]

TP

FP

FN

TN

```r
TP <- conf_matrix[2, 2]
FP <- conf_matrix[1, 2]
FN <- conf_matrix[2, 1]
TN <- conf_matrix[1, 1]
TP
FP
FN
TN
```

```
Console   Terminal ×   Background Jobs ×
R 4.3.1 · ~/
> TP <- conf_matrix[2, 2]
> FP <- conf_matrix[1, 2]
> FN <- conf_matrix[2, 1]
> TN <- conf_matrix[1, 1]
> TP
[1] 5
> FP
[1] 0
> FN
[1] 1
> TN
[1] 63
>
```

Description:

Extracting TP(True Positive),FP(False Positive),FN(False Negative),TN(True Negative) values from confusion matrix.

Recall, FP rate , Precision and F- measure:

Code:

recall <- TP / (TP + FN)

recall

FP_rate <- FP / (FP + TN)

FP_rate

precision <- TP / (TP + FP)

precision

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

f_measure

```r
recall <- TP / (TP + FN)
recall
FP_rate <- FP / (FP + TN)
FP_rate
precision <- TP / (TP + FP)
precision
f_measure <- 2 * (precision * recall) / (precision + recall)
f_measure
```

Console Terminal Background Jobs

R 4.3.1 · ~/

```
> recall <- TP / (TP + FN)
> recall
[1] 0.8333333
> FP_rate <- FP / (FP + TN)
> FP_rate
[1] 0
> precision <- TP / (TP + FP)
> precision
[1] 1
> f_measure <- 2 * (precision * recall) / (precision + recall)
> f_measure
[1] 0.9090909
> 
```

Description:

In our naïve Bayes model we get the values from confusion matrix

- Recall: **83.33%** This means that our Naive Bayes model correctly identified approximately 83.33% of the instances that actually belonged to the positive class
- False Positive Rate (FP Rate): **0%**.
- Precision: **100% .**
- F-measure: **90.91%** It means our Naive Bayes model achieved a good balance between precision and recall.