Assignment No	11
Title	Classification
Objective	Naïve Bayesian Algorithm , KNN , Id3 , C4.5
Roll No	MCA2565

1) Naïve Bayesian Algorithm

Source Code :-

```
install.packages("e1071")
install.packages("caTools")
install.packages("caret")

library(e1071)
library(caTools) library(caret)
View(iris)

ir<-iris train=ir[1:100,]
train

test=ir[101:150,] test
model=naiveBayes(Species~.,data=train)
model

test$Species model
test$Species

train$Species
pred=predict(model,test)
table(pred)
table(test$Species)
table(train$Species)

#shuffle iris file
ir1=ir[sample(nrow(ir)),]
View(ir1)
train=ir1[1:100,]
test=ir1[101:150,]
```

model=model=naiveBayes(Species~.,data=train)
pred=predict(model,test) table(pred)

table(train$Species)
table(test$Species)

Output :-

	Sepal.Length	Sepal.Width	Petal.Length	Petal.Width	Species
1	5.1	3.5	1.4	0.2	setosa
2	4.9	3.0	1.4	0.2	setosa
3	4.7	3.2	1.3	0.2	setosa
4	4.6	3.1	1.5	0.2	setosa
5	5.0	3.6	1.4	0.2	setosa
6	5.4	3.9	1.7	0.4	setosa
7	4.6	3.4	1.4	0.3	setosa
8	5.0	3.4	1.5	0.2	setosa
9	4.4	2.9	1.4	0.2	setosa
10	4.9	3.1	1.5	0.1	setosa
11	5.4	3.7	1.5	0.2	setosa
12	4.8	3.4	1.6	0.2	setosa
13	4.8	3.0	1.4	0.1	setosa
14	4.3	3.0	1.1	0.1	setosa
15	5.8	4.0	1.2	0.2	setosa
16	5.7	4.4	1.5	0.4	setosa
17	5.4	3.9	1.3	0.4	setosa
18	5.1	3.5	1.4	0.3	setosa
19	5.7	3.8	1.7	0.3	setosa

```
> View(iris)
> ir<-iris
> train=ir[1:100,]
> train
   Sepal.Length Sepal.Width Petal.Length Petal.Width  Species
1           5.1         3.5          1.4         0.2   setosa
2           4.9         3.0          1.4         0.2   setosa
3           4.7         3.2          1.3         0.2   setosa
4           4.6         3.1          1.5         0.2   setosa
5           5.0         3.6          1.4         0.2   setosa
6           5.4         3.9          1.7         0.4   setosa
7           4.6         3.4          1.4         0.3   setosa
8           5.0         3.4          1.5         0.2   setosa
9           4.4         2.9          1.4         0.2   setosa
10          4.9         3.1          1.5         0.1   setosa
11          5.4         3.7          1.5         0.2   setosa
12          4.8         3.4          1.6         0.2   setosa
13          4.8         3.0          1.4         0.1   setosa
14          4.3         3.0          1.1         0.1   setosa
15          5.8         4.0          1.2         0.2   setosa
16          5.7         4.4          1.5         0.4   setosa
17          5.4         3.9          1.3         0.4   setosa
18          5.1         3.5          1.4         0.3   setosa
19          5.7         3.8          1.7         0.3   setosa
20          5.1         3.8          1.5         0.3   setosa
21          5.4         3.4          1.7         0.2   setosa
22          5.1         3.7          1.5         0.4   setosa
> test=ir[101:150,]
> test
    Sepal.Length Sepal.Width Petal.Length Petal.Width   Species
101          6.3         3.3          6.0         2.5 virginica
102          5.8         2.7          5.1         1.9 virginica
103          7.1         3.0          5.9         2.1 virginica
104          6.3         2.9          5.6         1.8 virginica
105          6.5         3.0          5.8         2.2 virginica
106          7.6         3.0          6.6         2.1 virginica
107          4.9         2.5          4.5         1.7 virginica
108          7.3         2.9          6.3         1.8 virginica
109          6.7         2.5          5.8         1.8 virginica
110          7.2         3.6          6.1         2.5 virginica
111          6.5         3.2          5.1         2.0 virginica
```

```
> model=naiveBayes(Species~.,data=train)
> model

Naive Bayes Classifier for Discrete Predictors

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

A-priori probabilities:
Y
    setosa versicolor  virginica
       0.5        0.5        0.0

Conditional probabilities:
            Sepal.Length
Y             [,1]      [,2]
  setosa     5.006 0.3524897
  versicolor 5.936 0.5161711
  virginica     NA        NA

            Sepal.Width
Y             [,1]      [,2]
  setosa     3.428 0.3790644
  versicolor 2.770 0.3137983
  virginica     NA        NA

            Petal.Length
Y             [,1]     [,2]
  setosa     1.462 0.173664
  versicolor 4.260 0.469911
  virginica     NA       NA

> test$Species
 [1] virginica virginica virginica virginica virginica virginica virginica virginica
 [9] virginica virginica virginica virginica virginica virginica virginica virginica
[17] virginica virginica virginica virginica virginica virginica virginica virginica
[25] virginica virginica virginica virginica virginica virginica virginica virginica
[33] virginica virginica virginica virginica virginica virginica virginica virginica
[41] virginica virginica virginica virginica virginica virginica virginica virginica
[49] virginica virginica
Levels: setosa versicolor virginica
> model

Naive Bayes Classifier for Discrete Predictors

Y             [,1]      [,2]
  setosa     0.246 0.1053856
  versicolor 1.326 0.1977527
  virginica     NA        NA

> test$Species
 [1] virginica virginica virginica virginica virginica virginica virginica virginica
 [9] virginica virginica virginica virginica virginica virginica virginica virginica
[17] virginica virginica virginica virginica virginica virginica virginica virginica
[25] virginica virginica virginica virginica virginica virginica virginica virginica
[33] virginica virginica virginica virginica virginica virginica virginica virginica
[41] virginica virginica virginica virginica virginica virginica virginica virginica
[49] virginica virginica
Levels: setosa versicolor virginica
>
```

```
> train$Species
  [1] setosa     setosa     setosa     setosa     setosa     setosa     setosa
  [8] setosa     setosa     setosa     setosa     setosa     setosa     setosa
 [15] setosa     setosa     setosa     setosa     setosa     setosa     setosa
 [22] setosa     setosa     setosa     setosa     setosa     setosa     setosa
 [29] setosa     setosa     setosa     setosa     setosa     setosa     setosa
 [36] setosa     setosa     setosa     setosa     setosa     setosa     setosa
 [43] setosa     setosa     setosa     setosa     setosa     setosa     setosa
 [50] setosa     versicolor versicolor versicolor versicolor versicolor versicolor
 [57] versicolor versicolor versicolor versicolor versicolor versicolor versicolor
 [64] versicolor versicolor versicolor versicolor versicolor versicolor versicolor
 [71] versicolor versicolor versicolor versicolor versicolor versicolor versicolor
 [78] versicolor versicolor versicolor versicolor versicolor versicolor versicolor
 [85] versicolor versicolor versicolor versicolor versicolor versicolor versicolor
 [92] versicolor versicolor versicolor versicolor versicolor versicolor versicolor
 [99] versicolor versicolor
Levels: setosa versicolor virginica
> pred=predict(model,test)
> table(pred)
pred
    setosa versicolor  virginica
         0         50          0
> table(test$Species)

    setosa versicolor  virginica
         0          0         50
> table(train$Species)

    setosa versicolor  virginica
        50         50          0
```

	Sepal.Length	Sepal.Width	Petal.Length	Petal.Width	Species
18	5.1	3.5	1.4	0.3	setosa
47	5.1	3.8	1.6	0.2	setosa
46	4.8	3.0	1.4	0.3	setosa
61	5.0	2.0	3.5	1.0	versicolor
125	6.7	3.3	5.7	2.1	virginica
115	5.8	2.8	5.1	2.4	virginica
72	6.1	2.8	4.0	1.3	versicolor
143	5.8	2.7	5.1	1.9	virginica
129	6.4	2.8	5.6	2.1	virginica

```
> #shuffle iris file
> ir1=ir[sample(nrow(ir)),]
> View(ir1)
> train=ir1[1:100,]
> test=ir1[101:150,]
> model=model=naiveBayes(Species~.,data=train)
> pred=predict(model,test)
> table(pred)
pred
    setosa versicolor  virginica
        17         15         18
> table(train$Species)

    setosa versicolor  virginica
        33         36         31
> table(test$Species)

    setosa versicolor  virginica
        17         14         19
>
> |
```

2) KNN

Source Code :-

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

table(iris$Species)
str(iris$Species)
head(iris)
ir=iris
train=ir[1:100,]
 ir1=ir[sample(nrow(ir)),]
head(ir1)
normalize<-function(x){
 return((x-min(x)/(max(x)-min(x))))
}
iris_n<-as.data.frame(lapply(ir1[,c(1,2,3,4)],normalize))
str(iris_n)
 iris_train<-iris_n[1:129,]
iris_test<-iris_n[130:150,]
iris_train_target<-iris[1:129,5]
iris_test_target<-iris[130:150,5]
iris_train_target
dim(iris_train)
dim(iris_test)
model<-knn(iris_train,iris_test,cl=iris_train_target,k=13)
```

model
table(iris_test_target,model)

Output:-

```
> library(class)

Warning message:
package ‘class’ was built under R version 4.5.2

> table(iris$Species)

    setosa versicolor  virginica
        50         50         50
> str(iris$Species)
 Factor w/ 3 levels "setosa","versicolor",..: 1 1 1 1 1 1 1 1 1 1 ...
> head(iris)
  Sepal.Length Sepal.Width Petal.Length Petal.Width Species
1          5.1         3.5          1.4         0.2  setosa
2          4.9         3.0          1.4         0.2  setosa
3          4.7         3.2          1.3         0.2  setosa
4          4.6         3.1          1.5         0.2  setosa
5          5.0         3.6          1.4         0.2  setosa
6          5.4         3.9          1.7         0.4  setosa
> ir=iris
> train=ir[1:100,]
>   ir1=ir[sample(nrow(ir)),]
> head(ir1)
    Sepal.Length Sepal.Width Petal.Length Petal.Width    Species
127          6.2         2.8          4.8         1.8  virginica
150          5.9         3.0          5.1         1.8  virginica
85           5.4         3.0          4.5         1.5 versicolor
109          6.7         2.5          5.8         1.8  virginica
21           5.4         3.4          1.7         0.2     setosa
87           6.7         3.1          4.7         1.5 versicolor
> normalize<-function(x){
+   return((x-min(x)/(max(x)-min(x)))
+ }
```

```
> iris_n<-as.data.frame(lapply(ir1[,c(1,2,3,4)],normalize))
> str(iris_n)
'data.frame':   150 obs. of  4 variables:
 $ Sepal.Length: num  5.01 4.71 4.21 5.51 4.21 ...
 $ Sepal.Width : num  1.97 2.17 2.17 1.67 2.57 ...
 $ Petal.Length: num  4.63 4.93 4.33 5.63 1.53 ...
 $ Petal.Width : num  1.758 1.758 1.458 1.758 0.158 ...
>   iris_train<-iris_n[1:129,]
> iris_test<-iris_n[130:150,]
> iris_train_target<-iris[1:129,5]
> iris_test_target<-iris[130:150,5]
> iris_train_target
  [1] setosa     setosa     setosa     setosa     setosa     setosa     setosa
  [8] setosa     setosa     setosa     setosa     setosa     setosa     setosa
 [15] setosa     setosa     setosa     setosa     setosa     setosa     setosa
 [22] setosa     setosa     setosa     setosa     setosa     setosa     setosa
 [29] setosa     setosa     setosa     setosa     setosa     setosa     setosa
 [36] setosa     setosa     setosa     setosa     setosa     setosa     setosa
 [43] setosa     setosa     setosa     setosa     setosa     setosa     setosa
 [50] setosa     versicolor versicolor versicolor versicolor versicolor versicolor
 [57] versicolor versicolor versicolor versicolor versicolor versicolor versicolor
 [64] versicolor versicolor versicolor versicolor versicolor versicolor versicolor
 [71] versicolor versicolor versicolor versicolor versicolor versicolor versicolor
 [78] versicolor versicolor versicolor versicolor versicolor versicolor versicolor
 [85] versicolor versicolor versicolor versicolor versicolor versicolor versicolor
 [92] versicolor versicolor versicolor versicolor versicolor versicolor versicolor
 [99] versicolor versicolor virginica  virginica  virginica  virginica  virginica
[106] virginica  virginica  virginica  virginica  virginica  virginica  virginica
[113] virginica  virginica  virginica  virginica  virginica  virginica  virginica
[120] virginica  virginica  virginica  virginica  virginica  virginica  virginica
[127] virginica  virginica  virginica
Levels: setosa versicolor virginica
Levels: setosa versicolor virginica
> dim(iris_train)
[1] 129   4
> dim(iris_test)
[1] 21  4
> model<-knn(iris_train,iris_test,cl=iris_train_target,k=13)
> model
 [1] setosa     setosa     setosa     versicolor virginica  setosa     virginica
 [8] setosa     setosa     setosa     setosa     virginica  setosa     setosa
[15] setosa     setosa     versicolor setosa     versicolor versicolor versicolor
Levels: setosa versicolor virginica
> table(iris_test_target,model)
                model
iris_test_target setosa versicolor virginica
      setosa          0          0         0
      versicolor      0          0         0
      virginica      13          5         3
>
> |
```

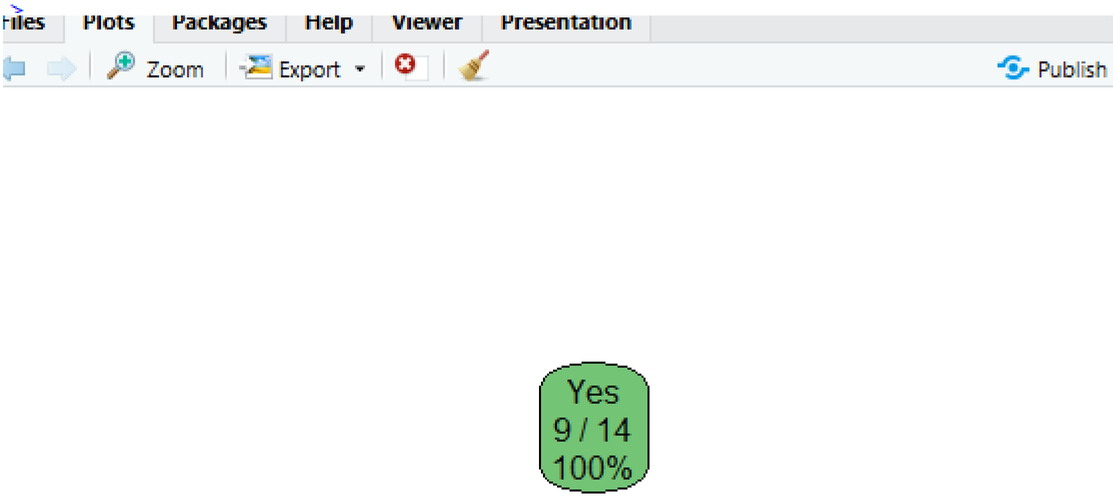
3)ID3

Source Code :-

```
#Install and load required packages
install.packages("rpart")
install.packages("rpart.plot")
library(rpart)
library(rpart.plot)

#Create a sample dataset
data <- data.frame(
  Outlook = c("Sunny","Sunny","Overcast","Rain","Rain","Rain","Overcast",
            "Sunny","Sunny","Rain","Sunny","Overcast","Overcast","Rain"),
  Temperature = c("Hot","Hot","Hot","Mild","Cool","Cool","Cool",
               "Mild","Cool","Mild","Mild","Mild","Hot","Mild"),
  Wind = c("Weak","Strong","Weak","Weak","Weak","Strong","Strong",
         "Weak","Weak","Weak","Strong","Strong","Weak","Strong"),
  Humidity = c("High","High","High","High","Normal","Normal","High",
            "High","Normal","Normal","Normal","High","Normal","High"),
  PlayTennis = c("NO","NO","Yes","Yes","Yes","NO","Yes",
              "No","Yes","Yes","Yes","Yes","Yes","No")
)

#Build ID3 Decision Tree
model <- rpart(PlayTennis ~ Outlook + Temperature + Humidity + Wind,
            data = data,
            method = "class",
            parms = list(split = "information")) #ID3 uses information gain

#Display model summary
print(model)

#plot decision tree
rpart.plot(model, type = 3, extra = 102,fallen.leaves = TRUE,box.palette = "Green")
```

Output :-

```
> library(rpart)
> library(rpart.plot)
> #Create a sample dataset
> data <- data.frame(
+   Outlook = c("Sunny","Sunny","Overcast","Rain","Rain","Rain","Overcast",
+               "Sunny","Sunny","Rain","Sunny","Overcast","Overcast","Rain"),
+   Temperature = c("Hot","Hot","Hot","Mild","Cool","Cool","Cool",
+                   "Mild","Cool","Mild","Mild","Mild","Hot","Mild"),
+   Wind = c("Weak","Strong","Weak","Weak","Weak","Strong","Strong",
+            "Weak","Weak","Weak","Strong","Strong","Weak","Strong"),
+   Humidity = c("High","High","High","High","Normal","Normal","High",
+                "High","Normal","Normal","Normal","High","Normal","High"),
+   PlayTennis = c("No","No","Yes","Yes","Yes","No","Yes",
+                  "No","Yes","Yes","Yes","Yes","Yes","No")
+ )
> #Build ID3 Decision Tree
> model <- rpart(PlayTennis ~ Outlook + Temperature + Humidity + Wind,
+                data = data,
+                method = "class",
+                parms = list(split = "information")) #ID3 uses information gain
> #Display model summary
> print(model)
n= 14

node), split, n, loss, yval, (yprob)
      * denotes terminal node

1) root 14 5 Yes (0.1428571 0.2142857 0.6428571) *
> #plot decision tree
> rpart.plot(model, type = 3, extra = 102,fallen.leaves = TRUE,box.palette = "Green")
>
>
```



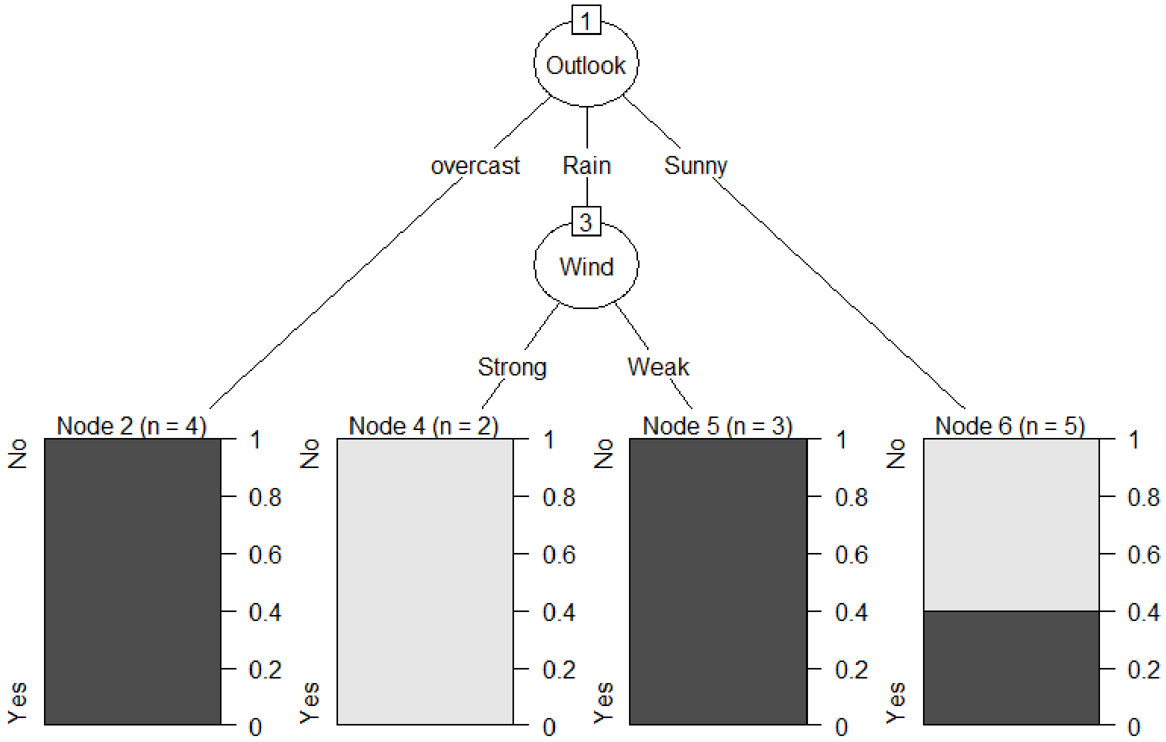
4) C4.5

Source Code :-

```
#Install and load
install.packages("RWeka")
library(RWeka)

#Step l : Create Dataset
data <- data.frame(
        Outlook=c("Sunny","Sunny","overcast","Rain","Rain","Rain","Sunny",
              "Sunny","Rain","Sunny","overcast","overcast","overcast","Rain"),
        Temperature=c("Hot","Hot","Hot","Mild","Cool","Cool","Cool",
             "Mild","Cool","Mild","Mild","Mild","Hot","Mild"),
        Humidity = c("High","High","High","Normal","Normal","Normal","High",
              "Normal","Normal","Normal","High","Normal","High","Normal"),
        Wind=c("Weak","Strong","Weak","Weak","Weak","Strong","Strong",
             "Weak","Weak","Weak","Strong","Strong","Weak","Strong"),
        playTennis = c("No","No","Yes","Yes","Yes","No", "Yes",
              "No","Yes","Yes","Yes","Yes","Yes","No")
)
#Step 2 : Convert all string columns to factors
data[] <- lapply(data, as.factor)

#step 3 : Train c4.5 model (J48)
model <- J48(playTennis ~ Outlook + Temperature + Humidity + Wind, data = data)

#Step 4: View model Summary
summary(model)

install.packages("partykit")
library(partykit)

plot(as.party(model))
```

Output :-

```
> library(RWeka)
> #Step 1 : Create Dataset
> data <- data.frame(
+             Outlook=c("Sunny","Sunny","Overcast","Rain","Rain","Rain","Sunny",
+                    "Sunny","Rain","Sunny","Overcast","Overcast","Overcast","Rain"),
+             Temperature=c("Hot","Hot","Hot","Mild","Cool","Cool","Cool",
+                   "Mild","Cool","Mild","Mild","Mild","Hot","Mild"),
+             Humidity = c("High","High","High","Normal","Normal","Normal","High",
+                    "Normal","Normal","Normal","High","Normal","High","Normal"),
+             Wind=c("Weak","Strong","Weak","Weak","Weak","Strong","Strong",
+                   "Weak","Weak","Weak","Strong","Strong","Weak","Strong"),
+             playTennis = c("No","No","Yes","Yes","Yes","No", "Yes",
+                    "No","Yes","Yes","Yes","Yes","Yes","No")
+ )
> #Step 2 : Convert all string columns to factors
> data[] <- lapply(data, as.factor)
> #step 3 : Train c4.5 model (J48)
> model <- J48(playTennis ~ Outlook + Temperature + Humidity + Wind, data = data)
> #Step 4: View model Summary
> summary(model)

=== Summary ===

Correctly Classified Instances          12               85.7143 %
Incorrectly Classified Instances         2               14.2857 %
Kappa statistic                          0.7143
Mean absolute error                      0.1714
Root mean squared error                  0.2928
Relative absolute error                 36.9231 %
Root relative squared error             61.0586 %
Total Number of Instances               14

=== Confusion Matrix ===

 a b   <-- classified as
 5 0 | a = No
 2 7 | b = Yes
> install.packages("partykit")
WARNING: Rtools is required to build R packages but is not currently installed. Please
download and install the appropriate version of Rtools before proceeding:

https://cran.rstudio.com/bin/windows/Rtools/
Warning: package 'partykit' is in use and will not be installed

> library(partykit)
> plot(as.party(model))
>
```


1
Outlook
overcast
Rain
Sunny
3
Wind
Strong
Weak
Node 2 (n = 4)
Node 4 (n = 2)
Node 5 (n = 3)
Node 6 (n = 5)
No
Yes
1
0.8
0.6
0.4
0.2
0