IMPLEMENTATION WEATHER PREDICTION USING MACHINE LEARNING ALGORITHM

AIM:

To predict the Weather Prediction using various Supervised Machine Learning Algorithm to determine which is the most accurate algorithm for that dataset.

weather forecasting has become the most challenging and important technique which helps us to predict the weather of any location. Weather prediction help in outdoor programming, crop cultivation, time management and other things that are concern for the mankind. From the last few decades, the advancement and development in science and technology enable scientists to make better and precise weather prediction. Another way to predict the weather using Machine Learning Algorithms which is used to help predict the weather. a process of collecting data on weather conditions, which records the **temperature, rainfall, evaporation, sunshine, wind direction, cloud, humidity wind speed and its direction**. Various Machine Learning Techniques are applied on weather data to predict climate parameters like temperature, wind speed, rainfall, meteorological pollution.

The four Supervised Machine Learning Algorithm which is used to predict the weather prediction dataset.

- KNN (K-Nearest Neighbours)
- Naive Bayes
- Decision Tree
- Random Forest

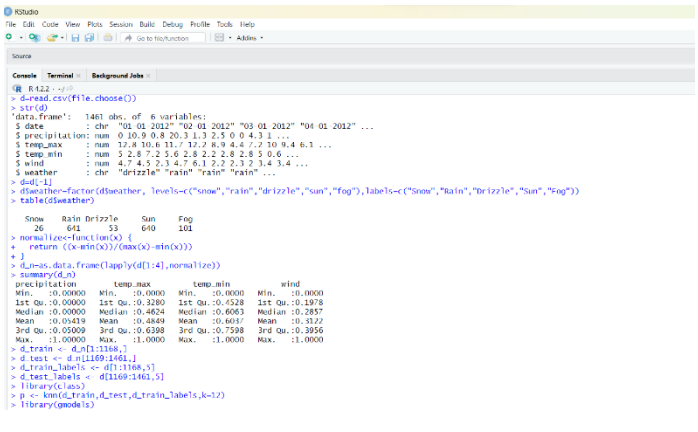
KNN ALGORITHM:

SOURCE CODE:

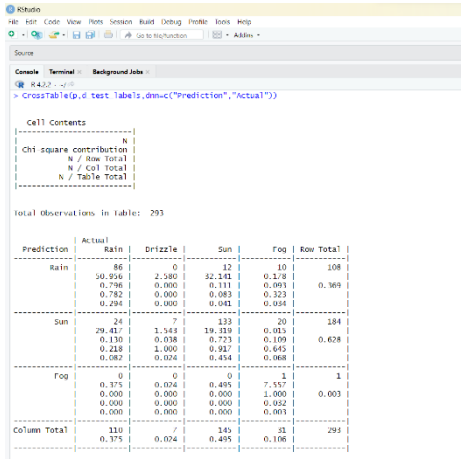
```
d=read.csv(file.choose())

str(d)

d=d[-1]

d$weather=factor(d$weather,

levels=c("snow","rain","drizzle","sun","fog"),labels=c("Snow","Rain","Drizzle","Sun","Fog"
))

table(d$weather)

normalize<-function(x) {

  return ((x-min(x))/(max(x)-min(x)))

}

d_n=as.data.frame(lapply(d[1:4],normalize))
```

```
summary(d_n)
d_train <- d_n[1:1168,]
d_test <- d_n[1169:1461,]
d_train_labels <- d[1:1168,5]
d_test_labels <- d[1169:1461,5]
library(class)
p <- knn(d_train,d_test,d_train_labels,k=12)
library(gmodels)
CrossTable(p,d_test_labels,dnn=c("Prediction","Actual"))
library(caret)
confusionMatrix(p,d_test_labels)
```

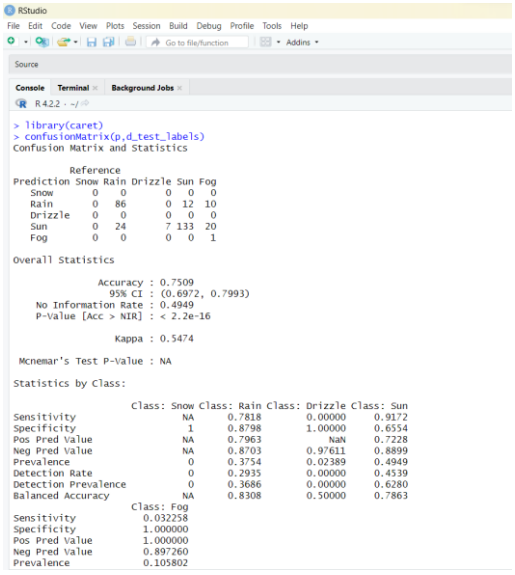
OUTPUT:



```
> d=read.csv(file.choose())
> str(d)
'data.frame':   1461 obs. of  6 variables:
 $ date         : chr  "01 01 2012" "02 01 2012" "03 01 2012" "04 01 2012" ...
 $ 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 3.4 3.4 ...
 $ weather      : chr  "drizzle" "rain" "rain" "rain" ...
> d=d[-1]
> dfweather=factor(dfweather, levels=c("snow","rain","drizzle","sun","fog"),labels=c("Snow","Rain","Drizzle","Sun","Fog"))
> table(dfweather)

   Snow    Rain Drizzle     Sun     Fog
     26     641      53     640     101
> normalizec<-function(x) {
+   return ((x-min(x))/(max(x)-min(x)))
+ }
> d_n=as.data.frame(lapply(d[1:4],normalize))
> summary(d_n)
 precipitation       temp_max         temp_min           wind
 Min.   :0.00000   Min.   :0.0000   Min.   :0.0000   Min.   :0.0000
 1st Qu.:0.00000   1st Qu.:0.3280   1st Qu.:0.4528   1st Qu.:0.1978
 Median :0.00000   Median :0.4624   Median :0.6063   Median :0.2857
 Mean   :0.05419   Mean   :0.4849   Mean   :0.6037   Mean   :0.3122
 3rd Qu.:0.05009   3rd Qu.:0.6398   3rd Qu.:0.7598   3rd Qu.:0.3956
 Max.   :1.00000   Max.   :1.0000   Max.   :1.0000   Max.   :1.0000
> d_train <- d_n[1:1168,]
> d_test <- d_n[1169:1461,]
> d_train_labels <- d[1:1168,5]
> d_test_labels <- d[1169:1461,5]
> library(class)
> p <- knn(d_train,d_test,d_train_labels,k=12)
> library(gmodels)
```



```
> CrossTable(p,d_test_labels,dnn=c("Prediction","Actual"))

   Cell Contents
|-------------------------|
|                       N |
| Chi square contribution |
|           N / Row Total |
|           N / Col Total |
|         N / Table Total |
|-------------------------|

Total Observations in Table:  293

             | Actual
  Prediction |      Rain |   Drizzle |       Sun |       Fog | Row Total |
-------------|-----------|-----------|-----------|-----------|-----------|
        Rain |        86 |         0 |        12 |        10 |       108 |
             |    50.956 |     2.580 |    32.141 |     0.178 |           |
             |     0.796 |     0.000 |     0.111 |     0.093 |     0.369 |
             |     0.782 |     0.000 |     0.083 |     0.323 |           |
             |     0.294 |     0.000 |     0.041 |     0.034 |           |
-------------|-----------|-----------|-----------|-----------|-----------|
         Sun |        24 |         7 |       133 |        20 |       184 |
             |    29.417 |     1.543 |    19.319 |     0.015 |           |
             |     0.130 |     0.038 |     0.723 |     0.109 |     0.628 |
             |     0.218 |     1.000 |     0.917 |     0.645 |           |
             |     0.082 |     0.024 |     0.454 |     0.068 |           |
-------------|-----------|-----------|-----------|-----------|-----------|
         Fog |         0 |         0 |         0 |         1 |         1 |
             |     0.375 |     0.024 |     0.495 |     7.557 |           |
             |     0.000 |     0.000 |     0.000 |     1.000 |     0.003 |
             |     0.000 |     0.000 |     0.000 |     0.032 |           |
             |     0.000 |     0.000 |     0.000 |     0.003 |           |
-------------|-----------|-----------|-----------|-----------|-----------|
Column Total |       110 |         7 |       145 |        31 |       293 |
             |     0.375 |     0.024 |     0.495 |     0.106 |           |
-------------|-----------|-----------|-----------|-----------|-----------|
```



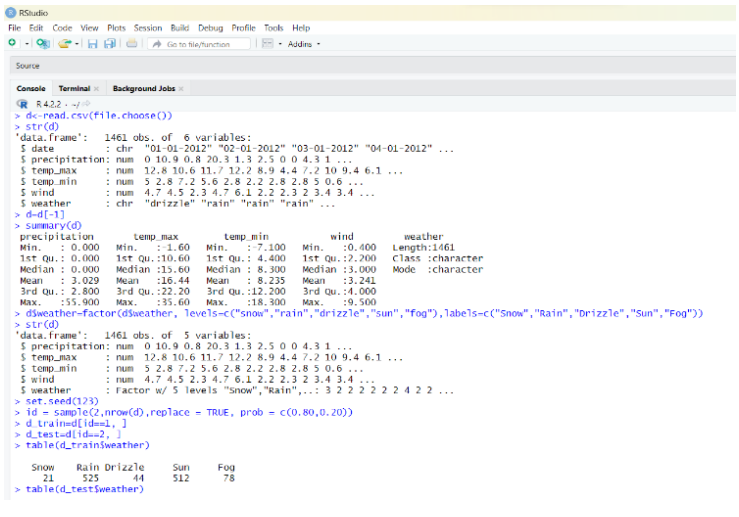
```
> library(caret)
> confusionMatrix(p,d_test_labels)
Confusion Matrix and Statistics

          Reference
Prediction Snow Rain Drizzle Sun Fog
   Snow       0    0       0   0   0
   Rain       0   86       0  12  10
   Drizzle    0    0       0   0   0
   Sun        0   24       7 133  20
   Fog        0    0       0   0   1

Overall Statistics

               Accuracy : 0.7509
                 95% CI : (0.6972, 0.7993)
    No Information Rate : 0.4949
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.5474

 Mcnemar's Test P-Value : NA

Statistics by Class:

                     Class: Snow Class: Rain Class: Drizzle Class: Sun
Sensitivity                   NA      0.7818        0.00000      0.9172
Specificity                    1      0.8798        1.00000      0.6554
Pos Pred Value                NA      0.7963            NaN      0.7228
Neg Pred Value                NA      0.8703        0.97611      0.8899
Prevalence                     0      0.3754        0.02389      0.4949
Detection Rate                 0      0.2935        0.00000      0.4539
Detection Prevalence           0      0.3686        0.00000      0.6280
Balanced Accuracy             NA      0.8308        0.50000      0.7863
                     Class: Fog
Sensitivity            0.032258
Specificity            1.000000
Pos Pred Value         1.000000
Neg Pred Value         0.897260
Prevalence             0.105802
```

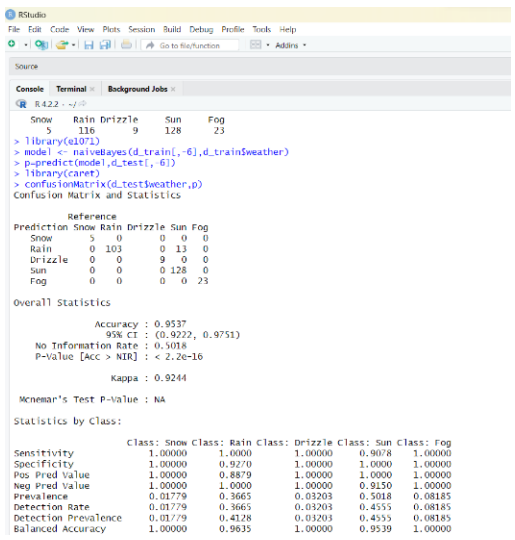
NAÏVE BAYES ALGORITHM:

SOURCE CODE:

```
d<-read.csv(file.choose())

str(d)

summary(d)

d$weather=factor(d$weather,
levels=c("snow","rain","drizzle","sun","fog"),labels=c("Snow","Rain","Drizzle","Sun","Fog"
))

str(d)

set.seed(123)

id = sample(2,nrow(d),replace = TRUE, prob = c(0.80,0.20))

d_train=d[id==1, ]

d_test=d[id==2, ]

table(d_train$weather)

table(d_test$weather)

library(e1071)

model <- naiveBayes(d_train[,-6],d_train$weather)

p=predict(model,d_test[,-6])

library(caret)

confusionMatrix(d_test$weather,p)
```

OUTPUT:



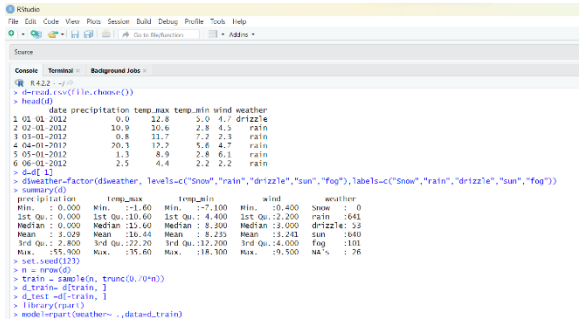


DECISION TREE ALGORITHM:

SOURCE CODE:

```
d=read.csv(file.choose())

str(d)

d=d[-1]

d$weather=factor(d$weather,
levels=c("Snow","rain","drizzle","sun","fog"),labels=c("Snow","rain","drizzle","sun","fog"))

summary(d)

set.seed(123)

n = nrow(d)

train = sample(n, trunc(0.70*n))

d_train= d[train, ]

d_test=d[-train, ]

install.packages("rpart")

library(rpart)

model=rpart(weather~ .,data=d_train)

plot(model, margin = 0.01)

text(model,use.n = TRUE, pretty = TRUE, cex=0.8)

P=predict(model,d_test,type="class")

library(corrplot)

corrplot(cor(d[,-5]),type="lower",method="number")

library(caret)

confusionMatrix(P,d_test$weather)
```

OUTPUT:



RStudio

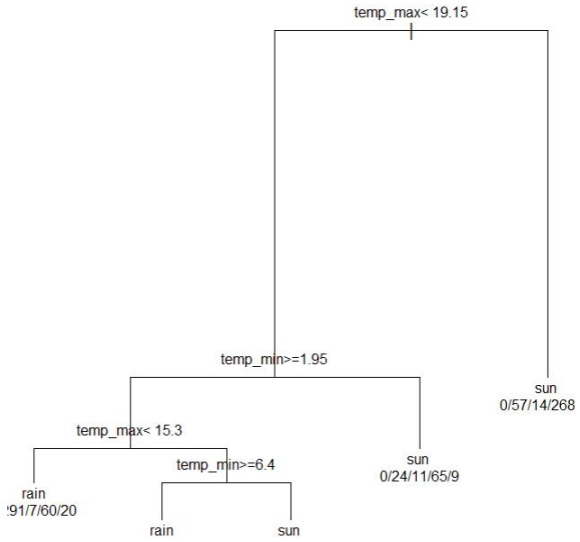
File Edit Code View Plots Session Build Debug Profile Tools Help

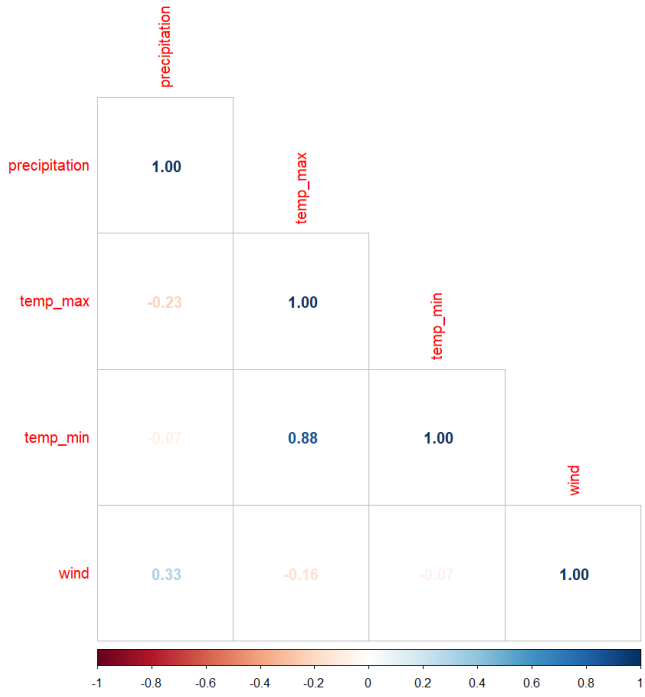
Source

Console Terminal Background Jobs

R 4.2.2 · ~/

```
> plot(model, margin = 0.04)
> text(model,use.n = TRUE, pretty = TRUE, cex=0.8)
> P=predict(model,d_test,type="class")
> library(corrplot)
> corrplot(cor(d[,-5]),type="lower",method="number")
> library(caret)
> confusionMatrix(P,d_test$weather)
Confusion Matrix and Statistics

          Reference
Prediction Snow rain drizzle sun fog
   Snow       0    0       0   0   0
   rain       0  176       0   0   0
   drizzle    0    0       0   0   0
   sun        0   17      18 180  35
   fog        0    0       0   0   0

Overall Statistics

               Accuracy : 0.8357
                 95% CI : (0.797, 0.8696)
    No Information Rate : 0.4531
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.7091

 Mcnemar's Test P-Value : NA

Statistics by Class:

                     Class: Snow Class: rain Class: drizzle Class: sun Class: fog
Sensitivity                   NA      0.9119        0.00000     1.0000     0.00000
Specificity                    1      1.0000        1.00000     0.7154     1.00000
Pos Pred Value                NA      1.0000            NaN     0.7200         NaN
Neg Pred Value                NA      0.9320        0.95775     1.0000     0.91784
Prevalence                     0      0.4531        0.04225     0.4225     0.08216
Detection Rate                 0      0.4131        0.00000     0.4225     0.00000
Detection Prevalence           0      0.4131        0.00000     0.5869     0.00000
Balanced Accuracy             NA      0.9560        0.50000     0.8577     0.50000
```



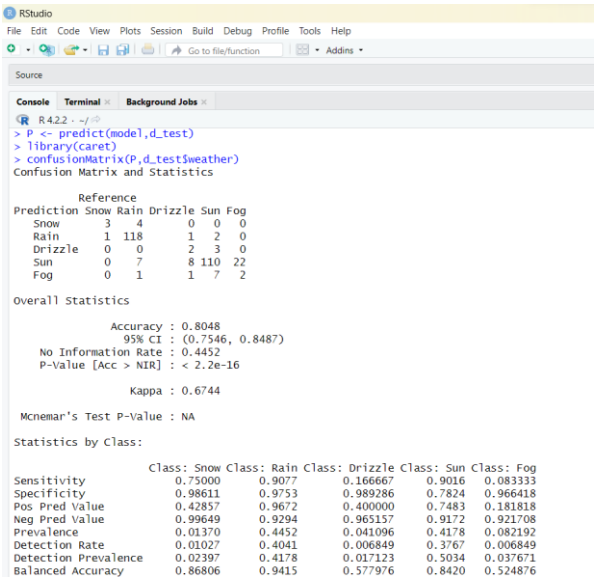


RANDOM FOREST ALGORITHM:

SOURCE CODE:

```
d=read.csv(file.choose())

str(d)

d=d[-1]

d$weather=factor(d$weather,
levels=c("snow","rain","drizzle","sun","fog"),labels=c("Snow","Rain","Drizzle","Sun","Fog"
))

table(d$weather)

set.seed(1234)

id<-sample(2,nrow(d),prob = c(0.8,0.2),replace = TRUE)

d_train<-d[id==1, ]

d_test<-d[id==2, ]

library(randomForest)

model = randomForest(weather~ ., data=d_train, ntree=50,mtry=3)

model

P <- predict(model,d_test)

library(caret)

confusionMatrix(P,d_test$weather)
```

OUTPUT:



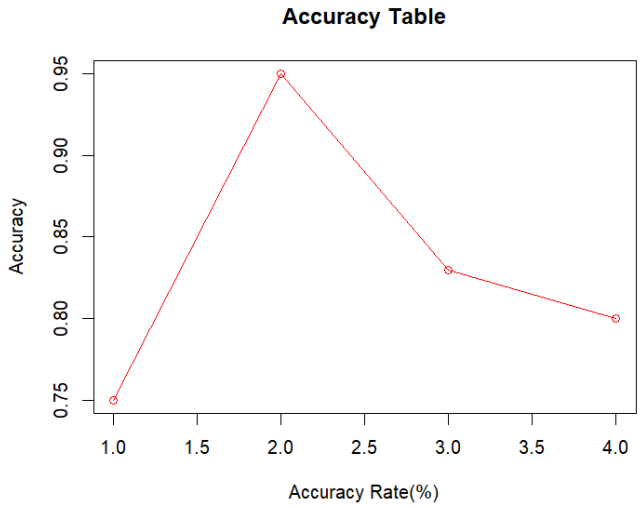
```
> P <- predict(model,d_test)
> library(caret)
> confusionMatrix(P,d_test$weather)
Confusion Matrix and Statistics

          Reference
Prediction Snow Rain Drizzle Sun Fog
   Snow       3    4       0   0   0
   Rain       1  118       1   2   0
   Drizzle    0    0       2   3   0
   Sun        0    7       8 110  22
   Fog        0    1       1   7   2

Overall Statistics

               Accuracy : 0.8048
                 95% CI : (0.7546, 0.8487)
    No Information Rate : 0.4452
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.6744

 Mcnemar's Test P-Value : NA

Statistics by Class:

                     Class: Snow Class: Rain Class: Drizzle Class: Sun Class: Fog
Sensitivity              0.75000      0.9077       0.166667      0.9016   0.083333
Specificity              0.98611      0.9753       0.989286      0.7824   0.966418
Pos Pred Value           0.42857      0.9672       0.400000      0.7483   0.181818
Neg Pred Value           0.99649      0.9294       0.965157      0.9172   0.921708
Prevalence               0.01370      0.4452       0.041096      0.4178   0.082192
Detection Rate           0.01027      0.4041       0.006849      0.3767   0.006849
Detection Prevalence     0.02397      0.4178       0.017123      0.5034   0.037671
Balanced Accuracy        0.86806      0.9415       0.577976      0.8420   0.524876
```

RESULT:

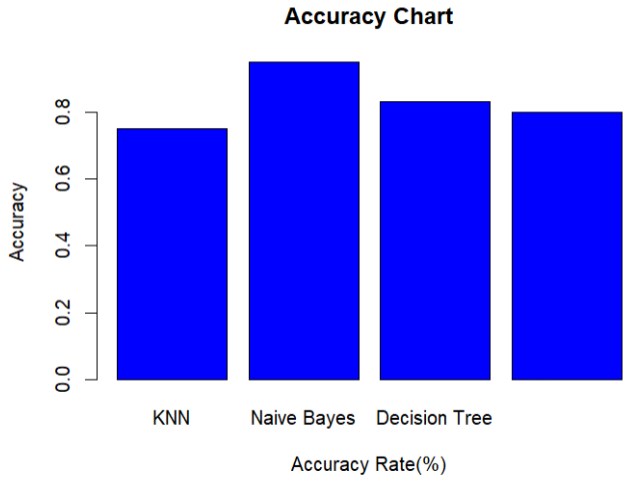
Accuracy Table

Algorithm	KNN	Naïve Bayes	Decision Tree	Random Forest
Accuracy	0.7509	0.9537	0.8357	0.8048

Line Graph:



Bar Graph:



CONCLUSION:

The weather prediction using machine learning algorithm project has been successful in predicting weather patterns with a high level of accuracy using the **Naïve Bayes algorithm**. The Naïve Bayes algorithm is a simple, yet powerful probabilistic classifier that is particularly effective in dealing with high-dimensional and sparse data, such as weather data. The success of the Naïve Bayes algorithm in predicting weather patterns shows the potential for machine learning to improve weather forecasting accuracy and provide more precise weather information to people worldwide. The findings from this project can be used to develop more sophisticated machine learning models for weather prediction, and ultimately, help mitigate the impact of severe weather events on people and the environment.