American International University-Bangladesh

NAME: Sadia Islam Shafina

ID: 20-43539-1

COURSE: INTRODUCTION TO DATA SCIENCE

SECTION: C

SUBMISSION DATE: 15-08-2023

Description:

The weather dataset is well known dataset in the field of data science. In our daily lives and many different businesses, including transportation, agriculture, and emergency preparedness,
weather data is essential. Making educated decisions, enhancing safety and maximising operations may all be accomplished with the understanding and appropriate use of meteorological data. The data set contain many rows(8,776) and 8 column. They are: Date/Time, Temp_C(Temperature in degrees Celsius.), Dew Point Temp_C(Dew point temperature in degrees Celsius), Rel Hum_%(Relative humidity as a percentage), Wind Speed_km/h(Wind speed in kilometers per hour.), Visibility_km(Visibility in kilometers), Press_kPa(Atmospheric pressure in kilopascals.), Weather(Weather condition). There are different types of attributes in our titanic dataset and they are integer, numeric, character. For this project our goal is to obtain a clean preprocessed dataset.

Data set name : **Weather Data**

Data set Link: https://www.kaggle.com/datasets/bhanupratapbiswas/weather-data

Project solution design:

- Import the data set(Weather.Data)as a csv file.
- Structure of the dataset.
- The first view row of the dataset.

- Finding type of our column:
- Summary of dataset.
- Data preparation steps.(Conversion categorical to numeric, Delete one column,Check if any data is missing, Data Normalization)

- correlation technique
- Plot of Correlation Matrix
- Correlation Diagram :
- Split data into training and testing sets
- Accuracy
- 10 fold cross validation
- Confusion matrix

Code and the steps of the projects:

Import data:

Explanation: First of all, I selected my data set from Kaggle then I download the data set as a Excell file .Then change the format of the dataset file into CSV file. Then, I import my csv file in R Studio then I provide the following code.

Code:

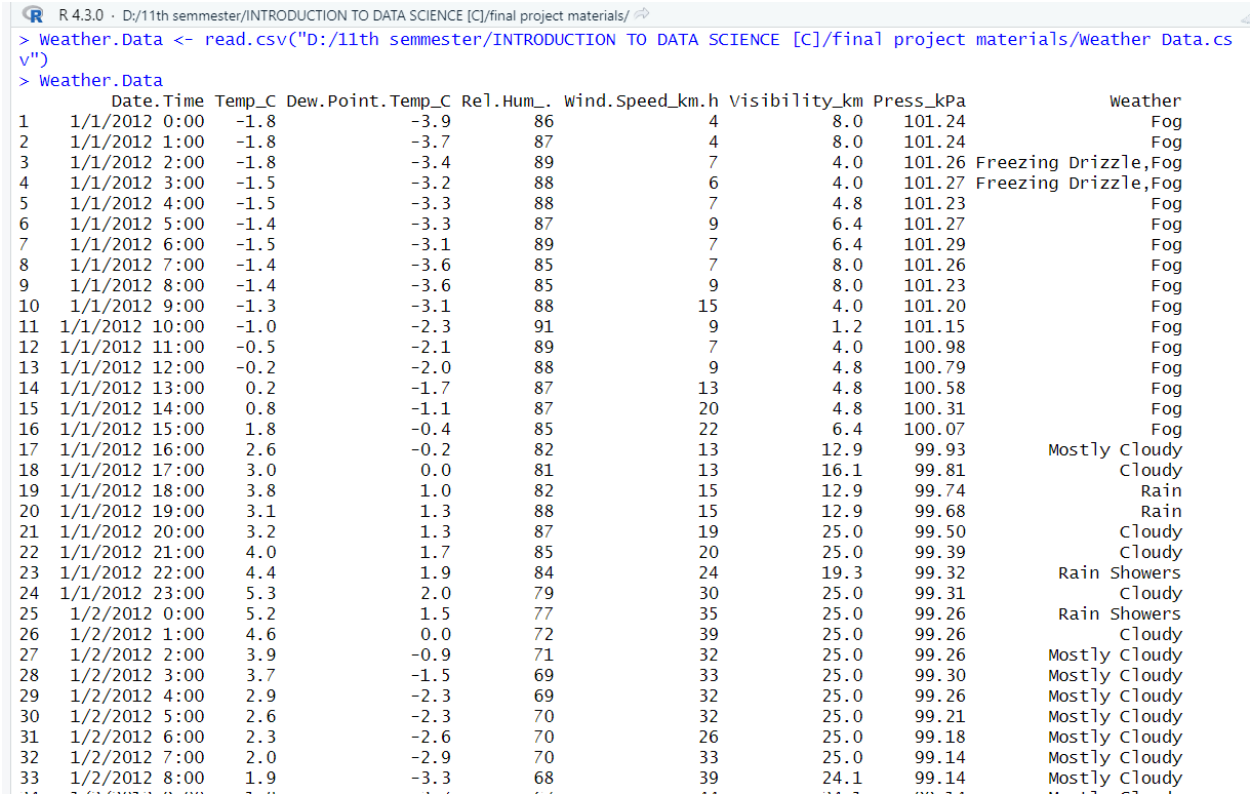
>setwd("D:/11th semmester/INTRODUCTION TO DATA SCIENCE [C]/final project materials")

>Weather.Data <- read.csv("D:/11th semmester/INTRODUCTION TO DATA SCIENCE [C]/final project materials/Weather Data.csv")

>View(Weather.Data)

>Weather.Data

Output:



R 4.3.0 · D:/11th semmester/INTRODUCTION TO DATA SCIENCE [C]/final project materials/

```
> Weather.Data <- read.csv("D:/11th semmester/INTRODUCTION TO DATA SCIENCE [C]/final project materials/Weather Data.csv")
> Weather.Data
         Date.Time Temp_C Dew.Point.Temp_C Rel.Hum_. Wind.Speed_km.h Visibility_km Press_kPa              Weather
1    1/1/2012 0:00   -1.8             -3.9        86               4           8.0    101.24                  Fog
2    1/1/2012 1:00   -1.8             -3.7        87               4           8.0    101.24                  Fog
3    1/1/2012 2:00   -1.8             -3.4        89               7           4.0    101.26 Freezing Drizzle,Fog
4    1/1/2012 3:00   -1.5             -3.2        88               6           4.0    101.27 Freezing Drizzle,Fog
5    1/1/2012 4:00   -1.5             -3.3        88               7           4.8    101.23                  Fog
6    1/1/2012 5:00   -1.4             -3.3        87               9           6.4    101.27                  Fog
7    1/1/2012 6:00   -1.5             -3.1        89               7           6.4    101.29                  Fog
8    1/1/2012 7:00   -1.4             -3.6        85               7           8.0    101.26                  Fog
9    1/1/2012 8:00   -1.4             -3.6        85               9           8.0    101.23                  Fog
10   1/1/2012 9:00   -1.3             -3.1        88              15           4.0    101.20                  Fog
11  1/1/2012 10:00   -1.0             -2.3        91               9           1.2    101.15                  Fog
12  1/1/2012 11:00   -0.5             -2.1        89               7           4.0    100.98                  Fog
13  1/1/2012 12:00   -0.2             -2.0        88               9           4.8    100.79                  Fog
14  1/1/2012 13:00    0.2             -1.7        87              13           4.8    100.58                  Fog
15  1/1/2012 14:00    0.8             -1.1        87              20           4.8    100.31                  Fog
16  1/1/2012 15:00    1.8             -0.4        85              22           6.4    100.07                  Fog
17  1/1/2012 16:00    2.6             -0.2        82              13          12.9     99.93         Mostly Cloudy
18  1/1/2012 17:00    3.0              0.0        81              13          16.1     99.81               Cloudy
19  1/1/2012 18:00    3.8              1.0        82              15          12.9     99.74                 Rain
20  1/1/2012 19:00    3.1              1.3        88              15          12.9     99.68                 Rain
21  1/1/2012 20:00    3.2              1.3        87              19          25.0     99.50               Cloudy
22  1/1/2012 21:00    4.0              1.7        85              20          25.0     99.39               Cloudy
23  1/1/2012 22:00    4.4              1.9        84              24          19.3     99.32          Rain Showers
24  1/1/2012 23:00    5.3              2.0        79              30          25.0     99.31               Cloudy
25   1/2/2012 0:00    5.2              1.5        77              35          25.0     99.26          Rain Showers
26   1/2/2012 1:00    4.6              0.0        72              39          25.0     99.26               Cloudy
27   1/2/2012 2:00    3.9             -0.9        71              32          25.0     99.26         Mostly Cloudy
28   1/2/2012 3:00    3.7             -1.5        69              33          25.0     99.30         Mostly Cloudy
29   1/2/2012 4:00    2.9             -2.3        69              32          25.0     99.26         Mostly Cloudy
30   1/2/2012 5:00    2.6             -2.3        70              32          25.0     99.21         Mostly Cloudy
31   1/2/2012 6:00    2.3             -2.6        70              26          25.0     99.18         Mostly Cloudy
32   1/2/2012 7:00    2.0             -2.9        70              33          25.0     99.14         Mostly Cloudy
33   1/2/2012 8:00    1.9             -3.3        68              39          24.1     99.14         Mostly Cloudy
```

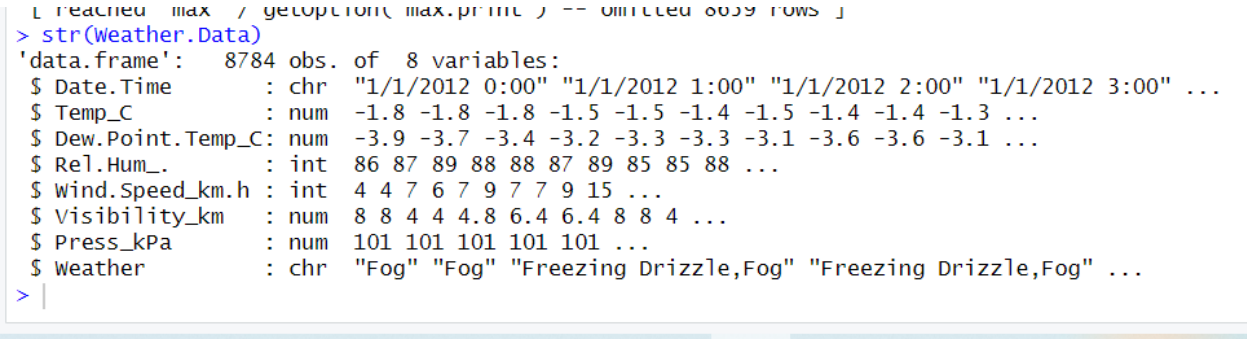
Structure of the dataset:

Explain: The str() function displays the structure of the dataset, including the variables, their data types and the first few values. This will give us an overview of the dataset.

Code:

```
> str(Weather.Data)
```

Output:



```
[ reached 'max' / getOption("max.print") -- omitted 8659 rows ]
> str(Weather.Data)
'data.frame':   8784 obs. of  8 variables:
 $ Date.Time       : chr  "1/1/2012 0:00" "1/1/2012 1:00" "1/1/2012 2:00" "1/1/2012 3:00" ...
 $ Temp_C          : num  -1.8 -1.8 -1.8 -1.5 -1.5 -1.4 -1.5 -1.4 -1.4 -1.3 ...
 $ Dew.Point.Temp_C: num  -3.9 -3.7 -3.4 -3.2 -3.3 -3.3 -3.1 -3.6 -3.6 -3.1 ...
 $ Rel.Hum_.       : int  86 87 89 88 88 87 89 85 85 88 ...
 $ Wind.Speed_km.h : int  4 4 7 6 7 9 7 7 9 15 ...
 $ Visibility_km   : num  8 8 4 4 4.8 6.4 6.4 8 8 4 ...
 $ Press_kPa       : num  101 101 101 101 101 ...
 $ Weather         : chr  "Fog" "Fog" "Freezing Drizzle,Fog" "Freezing Drizzle,Fog" ...
>
```

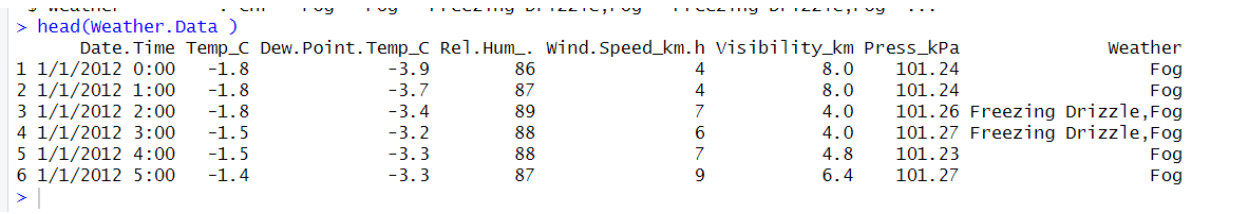
The first view row of the dataset:

Explanation: The head() function displays the first few rows of the dataset. This will help us get a sense of the data and verify that it has been imported correctly.

Code:

```
> head(Weather.Data)
```

Output:



```
> head(Weather.Data )
      Date.Time Temp_C Dew.Point.Temp_C Rel.Hum_. Wind.Speed_km.h Visibility_km Press_kPa              Weather
1 1/1/2012 0:00   -1.8             -3.9        86               4           8.0    101.24                  Fog
2 1/1/2012 1:00   -1.8             -3.7        87               4           8.0    101.24                  Fog
3 1/1/2012 2:00   -1.8             -3.4        89               7           4.0    101.26 Freezing Drizzle,Fog
4 1/1/2012 3:00   -1.5             -3.2        88               6           4.0    101.27 Freezing Drizzle,Fog
5 1/1/2012 4:00   -1.5             -3.3        88               7           4.8    101.23                  Fog
6 1/1/2012 5:00   -1.4             -3.3        87               9           6.4    101.27                  Fog
>
```

Finding type of our column:

Explanation: Using sapply, we can know which column has which type.

code

```
> sapply(Weather.Data, class)
```

Output:

```
> sapply(Weather.Data, class)
       Date.Time           Temp_C Dew.Point.Temp_C        Rel.Hum_.  Wind.Speed_km.h    Visibility_km
     "character"        "numeric"        "numeric"        "integer"        "integer"        "numeric"
       Press_kPa          Weather
       "numeric"      "character"
> |
```

Summary of full data set:

Explanation: The summary() function provides summary statistics (count, mean, median, etc.) for numeric variables in the dataset. This will give us insights into the distribution and central tendencies of the variables.

code

```
> summary(Weather.Data)
```

Output:

```
> summary(Weather.Data)
  Date.Time            Temp_C        Dew.Point.Temp_C    Rel.Hum_.      Wind.Speed_km.h Visibility_km
 Length:8784        Min.   :-23.300   Min.   :-28.500   Min.   : 18.00   Min.   : 0.00   Min.   : 0.20
 Class :character   1st Qu.:  0.100   1st Qu.: -5.900   1st Qu.: 56.00   1st Qu.: 9.00   1st Qu.:24.10
 Mode  :character   Median :  9.300   Median :  3.300   Median : 68.00   Median :13.00   Median :25.00
                    Mean   :  8.798   Mean   :  2.555   Mean   : 67.43   Mean   :14.95   Mean   :27.66
                    3rd Qu.: 18.800   3rd Qu.: 11.800   3rd Qu.: 81.00   3rd Qu.:20.00   3rd Qu.:25.00
                    Max.   : 33.000   Max.   : 24.400   Max.   :100.00   Max.   :83.00   Max.   :48.30
   Press_kPa         Weather
 Min.   : 97.52   Length:8784
 1st Qu.:100.56   Class :character
 Median :101.07   Mode  :character
 Mean   :101.05
 3rd Qu.:101.59
 Max.   :103.65
> |
```

Data preparation steps:

From our instruction, I need to apply KNN. Before apply KNN I need to prepare my data set. According to my weather data set first of all, I convert categorical data to numeric data in weather column then I delete one (date.time) column for prepare my data set. In below I provide those code.

Conversion categorical to numeric(weather column):

Explaination: The factor() function is applied to the "Weather" column, specifying the original categories and their corresponding numeric labels. The levels argument defines the original weather categories, while the labels argument assigns numeric labels to each category. This transformation is need for KNN .

Code:

```
>Weather.Data$Weather <- factor(Weather.Data$Weather, levels=c("Fog","Freezing Drizzle,Fog","Mostly Cloudy","Cloudy","Rain","Rain Showers","Mainly Clear","Snow Showers","Clear","Snow","Freezing Rain,Fog","Freezing Rain","Freezing Drizzle","Rain,Snow","Moderate Snow","Freezing Drizzle,Snow","Freezing Rain,Snow Grains","Snow,Blowing Snow","Freezing Fog","Haze","Rain,Fog","Drizzle,Fog","Rain,Snow","Freezing Drizzle,Haze","Freezing Rain,Haze","Snow,Haze","Drizzle","Snow,Fog","Snow,Ice Pellets","Thunderstorms,Rain Showers","Thunderstorms,Rain","Rain,Haze","Thunderstorms,Rain Showers,Fog","Thunderstorms","Thunderstorms,Rain,Fog","Thunderstorms,Moderate Rain Showers,Fog","Thunderstorms,Heavy Rain Showers","Rain Showers,Fog","Rain Showers,Snow Showers","Snow Pellets","Rain,Snow,Fog","Moderate Rain,Fog","Freezing Rain,Ice Pellets,Fog","Drizzle,Ice Pellets,Fog","Drizzle,Snow","Rain,Ice Pellets","Drizzle,Snow,Fog","Rain,Snow Grains","Snow Showers,Fog","Moderate Snow,Blowing Snow","Rain,Snow,Ice Pellets"), labels = c(1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29,30,31,32,33,34,35,36,37,38,39,40,41,42,43,44,45,46,47,48,49,50,51))
>Weather.Data
```

Output:

```
> Weather.Data$Weather <- factor(Weather.Data$Weather, levels=c("Fog","Freezing Drizzle,Fog","Mostly Cloudy","Cloud
y","Rain","Rain Showers","Mainly Clear","Snow Showers","Clear","Snow","Freezing Rain,Fog","Freezing Rain","Freezing Dri
zzle","Rain,Snow","Moderate Snow","Freezing Drizzle,Snow","Freezing Rain,Snow Grains","Snow,Blowing Snow","Freezing Fo
g","Haze","Rain,Fog","Drizzle,Fog","Rain,Snow","Freezing Drizzle,Haze","Freezing Rain,Haze","Snow,Haze","Drizzle" ), la
bels = c(1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27))
> Weather.Data
        Date.Time Temp_C Dew.Point.Temp_C Rel.Hum_. Wind.Speed_km.h Visibility_km Press_kPa Weather
1     1/1/2012 0:00   -1.8            -3.9        86              4           8.0    101.24       1
2     1/1/2012 1:00   -1.8            -3.7        87              4           8.0    101.24       1
3     1/1/2012 2:00   -1.8            -3.4        89              7           4.0    101.26       2
4     1/1/2012 3:00   -1.5            -3.2        88              6           4.0    101.27       2
5     1/1/2012 4:00   -1.5            -3.3        88              7           4.8    101.23       1
6     1/1/2012 5:00   -1.4            -3.3        87              9           6.4    101.27       1
7     1/1/2012 6:00   -1.5            -3.1        89              7           6.4    101.29       1
8     1/1/2012 7:00   -1.4            -3.6        85              7           8.0    101.26       1
9     1/1/2012 8:00   -1.4            -3.6        85              9           8.0    101.23       1
10    1/1/2012 9:00   -1.3            -3.1        88             15           4.0    101.20       1
11   1/1/2012 10:00   -1.0            -2.3        91              9           1.2    101.15       1
12   1/1/2012 11:00   -0.5            -2.1        89              7           4.0    100.98       1
13   1/1/2012 12:00   -0.2            -2.0        88              9           4.8    100.79       1
14   1/1/2012 13:00    0.2            -1.7        87             13           4.8    100.58       1
15   1/1/2012 14:00    0.8            -1.1        87             20           4.8    100.31       1
16   1/1/2012 15:00    1.8            -0.4        85             22           6.4    100.07       1
17   1/1/2012 16:00    2.6            -0.2        82             13          12.9     99.93       3
18   1/1/2012 17:00    3.0             0.0        81             13          16.1     99.81       4
19   1/1/2012 18:00    3.8             1.0        82             15          12.9     99.74       5
20   1/1/2012 19:00    3.1             1.3        88             15          12.9     99.68       5
21   1/1/2012 20:00    3.2             1.3        87             19          25.0     99.50       4
22   1/1/2012 21:00    4.0             1.7        85             20          25.0     99.39       4
23   1/1/2012 22:00    4.4             1.9        84             24          19.3     99.32       6
24   1/1/2012 23:00    5.3             2.0        79             30          25.0     99.31       4
25    1/2/2012 0:00    5.2             1.5        77             35          25.0     99.26       6
26    1/2/2012 1:00    4.6             0.0        72             39          25.0     99.26       4
27    1/2/2012 2:00    3.9            -0.9        71             32          25.0     99.26       3
28    1/2/2012 3:00    3.7            -1.5        69             33          25.0     99.30       3
29    1/2/2012 4:00    2.9            -2.3        69             32          25.0     99.26       3
30    1/2/2012 5:00    2.6            -2.3        70             32          25.0     99.21       3
31    1/2/2012 6:00    2.3            -2.6        70             26          25.0     99.18       3
```

Delete one column:

Code:

```
>Weather.Data <- Weather.Data[, -which(names(Weather.Data) == "Date.Time")]
>print(Weather.Data)
```

Output:

```
> Weather.Data <- Weather.Data[, -which(names(Weather.Data) == "Date.Time")]
> print(Weather.Data)
     Temp_C Dew.Point.Temp_C Rel.Hum_. Wind.Speed_km.h Visibility_km Press_kPa Weather
1      -1.8             -3.9        86              4           8.0     101.24       1
2      -1.8             -3.7        87              4           8.0     101.24       1
3      -1.8             -3.4        89              7           4.0     101.26       2
4      -1.5             -3.2        88              6           4.0     101.27       2
5      -1.5             -3.3        88              7           4.8     101.23       1
6      -1.4             -3.3        87              9           6.4     101.27       1
7      -1.5             -3.1        89              7           6.4     101.29       1
8      -1.4             -3.6        85              7           8.0     101.26       1
9      -1.4             -3.6        85              9           8.0     101.23       1
10     -1.3             -3.1        88             15           4.0     101.20       1
11     -1.0             -2.3        91              9           1.2     101.15       1
12     -0.5             -2.1        89              7           4.0     100.98       1
13     -0.2             -2.0        88              9           4.8     100.79       1
14      0.2             -1.7        87             13           4.8     100.58       1
15      0.8             -1.1        87             20           4.8     100.31       1
16      1.8             -0.4        85             22           6.4     100.07       1
17      2.6             -0.2        82             13          12.9      99.93       3
18      3.0              0.0        81             13          16.1      99.81       4
19      3.8              1.0        82             15          12.9      99.74       5
20      3.1              1.3        88             15          12.9      99.68       5
21      3.2              1.3        87             19          25.0      99.50       4
22      4.0              1.7        85             20          25.0      99.39       4
23      4.4              1.9        84             24          19.3      99.32       6
24      5.3              2.0        79             30          25.0      99.31       4
25      5.2              1.5        77             35          25.0      99.26       6
26      4.6              0.0        72             39          25.0      99.26       4
27      3.9             -0.9        71             32          25.0      99.26       3
28      3.7             -1.5        69             33          25.0      99.30       3
29      2.9             -2.3        69             32          25.0      99.26       3
30      2.6             -2.3        70             32          25.0      99.21       3
31      2.3             -2.6        70             26          25.0      99.18       3
32      2.0             -2.9        70             33          25.0      99.14       3
```

Check if any data is missing:

Explanation: Finding and handling missing values in a dataset is important because missing data can introduce bias and affect the accuracy of our analysis and results. Using this bellow code I can know the missing value in every column.

code:

```
number_of_missing_value=colSums(is.na(Weather.Data))
number_of_missing_value
```

output:

```
[ reached 'max' / getOption("max.print") -- omitted 8642 rows ]
> number_of_missing_value=colSums(is.na(Weather.Data))
> number_of_missing_value
          Temp_C Dew.Point.Temp_C        Rel.Hum_.   Wind.Speed_km.h    Visibility_km
               0                0                0                0                0
       Press_kPa          Weather
               0                0
```

In my weather data set there is no missing value.

Data Normalization:

Code:

```
> library(dplyr)
> Weather.Data <- as.data.frame(sapply(Weather.Data, as.numeric))
> min_max_norm <- function(x) {
      (x - min(x, na.rm = TRUE)) / (max(x, na.rm = TRUE) - min(x, na.rm = TRUE))
      }

>normalized_data <- Weather.Data %>%
        mutate(across(everything(), min_max_norm))
>print(normalized_data)
```

Output:

```
> print(normalized_data)
        Temp_C Dew.Point.Temp_C Rel.Hum_. Wind.Speed_km.h Visibility_km Press_kPa Weather
1   0.38188277       0.46502836 0.8292683      0.04819277    0.16216216 0.6068515    0.00
2   0.38188277       0.46880907 0.8414634      0.04819277    0.16216216 0.6068515    0.00
3   0.38188277       0.47448015 0.8658537      0.08433735    0.07900208 0.6101142    0.02
4   0.38721137       0.47826087 0.8536585      0.07228916    0.07900208 0.6117455    0.02
5   0.38721137       0.47637051 0.8536585      0.08433735    0.09563410 0.6052202    0.00
6   0.38898757       0.47637051 0.8414634      0.10843373    0.12889813 0.6117455    0.00
7   0.38721137       0.48015123 0.8658537      0.08433735    0.12889813 0.6150082    0.00
8   0.38898757       0.47069943 0.8170732      0.08433735    0.16216216 0.6101142    0.00
9   0.38898757       0.47069943 0.8170732      0.10843373    0.16216216 0.6052202    0.00
10  0.39076377       0.48015123 0.8536585      0.18072289    0.07900208 0.6003263    0.00
11  0.39609236       0.49527410 0.8902439      0.10843373    0.02079002 0.5921697    0.00
12  0.40497336       0.49905482 0.8658537      0.08433735    0.07900208 0.5644372    0.00
13  0.41030195       0.50094518 0.8536585      0.10843373    0.09563410 0.5334421    0.00
14  0.41740675       0.50661626 0.8414634      0.15662651    0.09563410 0.4991843    0.00
15  0.42806394       0.51795841 0.8414634      0.24096386    0.09563410 0.4551387    0.00
16  0.44582593       0.53119093 0.8170732      0.26506024    0.12889813 0.4159869    0.00
17  0.46003552       0.53497164 0.7804878      0.15662651    0.26403326 0.3931485    0.04
18  0.46714032       0.53875236 0.7682927      0.15662651    0.33056133 0.3735726    0.06
19  0.48134991       0.55765595 0.7804878      0.18072289    0.26403326 0.3621533    0.08
20  0.46891652       0.56332703 0.8536585      0.18072289    0.26403326 0.3523654    0.08
21  0.47069272       0.56332703 0.8414634      0.22891566    0.51559252 0.3230016    0.06
22  0.48490231       0.57088847 0.8170732      0.24096386    0.51559252 0.3050571    0.06
23  0.49200710       0.57466919 0.8048780      0.28915663    0.39708940 0.2936378    0.10
24  0.50799290       0.57655955 0.7439024      0.36144578    0.51559252 0.2920065    0.06
25  0.50621670       0.56710775 0.7195122      0.42168675    0.51559252 0.2838499    0.10
26  0.49555950       0.53875236 0.6585366      0.46987952    0.51559252 0.2838499    0.06
27  0.48312611       0.52173913 0.6463415      0.38554217    0.51559252 0.2838499    0.04
28  0.47957371       0.51039698 0.6219512      0.39759036    0.51559252 0.2903752    0.04
29  0.46536412       0.49527410 0.6219512      0.38554217    0.51559252 0.2838499    0.04
30  0.46003552       0.49527410 0.6341463      0.38554217    0.51559252 0.2756933    0.04
31  0.45470693       0.48960302 0.6341463      0.31325301    0.51559252 0.2707993    0.04
32  0.44937833       0.48393195 0.6341463      0.39759036    0.51559252 0.2642741    0.04
33  0.44760213       0.47637051 0.6097561      0.46987952    0.49688150 0.2642741    0.04
```

correlation technique

Explanation: The dplyr library and the cor() function to calculate the correlation matrix between the "Weather" column and a subset of selected numerical attributes from the "Weather.Data" dataset. The selected_attributes vector lists the names of the attributes to be included in the correlation analysis. The cor() function computes the Pearson correlation coefficients between the "Weather" column (categorical) and each of the selected numerical attributes. In a dataset, correlation is used to analyse the relationship between related pairs of variables. It enables us to determine whether changes in one variable may be related to changes in another variable.

Code:

```
>library(dplyr)
>selected_attributes <- c("Temp_C", "Dew.Point.Temp_C", "Rel.Hum_.", "Wind.Speed_km.h", "Visibility_km", "Press_kPa")

>correlation_matrix <- cor(Weather.Data$Weather, Weather.Data[, selected_attributes])

>print(correlation_matrix)
```

Output:

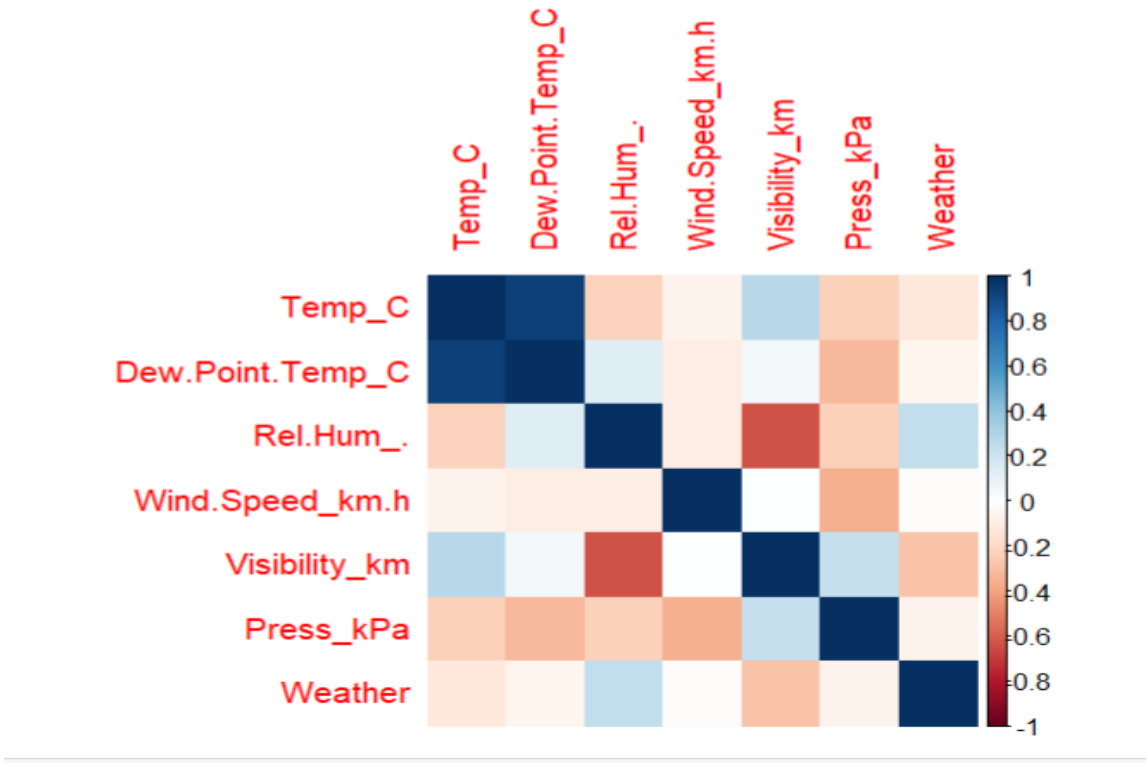
```
> print(correlation_matrix)
       Temp_C Dew.Point.Temp_C Rel.Hum_. Wind.Speed_km.h Visibility_km   Press_kPa
[1,] -0.121535      -0.05065331  0.241203     -0.01084815    -0.2892672 -0.06432801
>
```

Plot of Correlation Matrix:

Code:

```
library(corrplot)
a<-cor(normalized_data)
corrplot(a,method="color")
```

Output:



Correlation Diagram :

Explanation: First install the corrplot package then select the attributes and label attributes . Then the correlation heatmap generated successfully.

Code:

```
>library(dplyr)
>library(corrplot)
>selected_attributes <- c("Temp_C", "Dew.Point.Temp_C", "Rel.Hum_.", "Wind.Speed_km.h", "Visibility_km", "Press_kPa")
>correlation_matrix <- cor(Weather.Data$Weather, Weather.Data[, selected_attributes])

>corrplot(correlation_matrix, method = "color", type = "full", tl.col = "black")
```

Output:

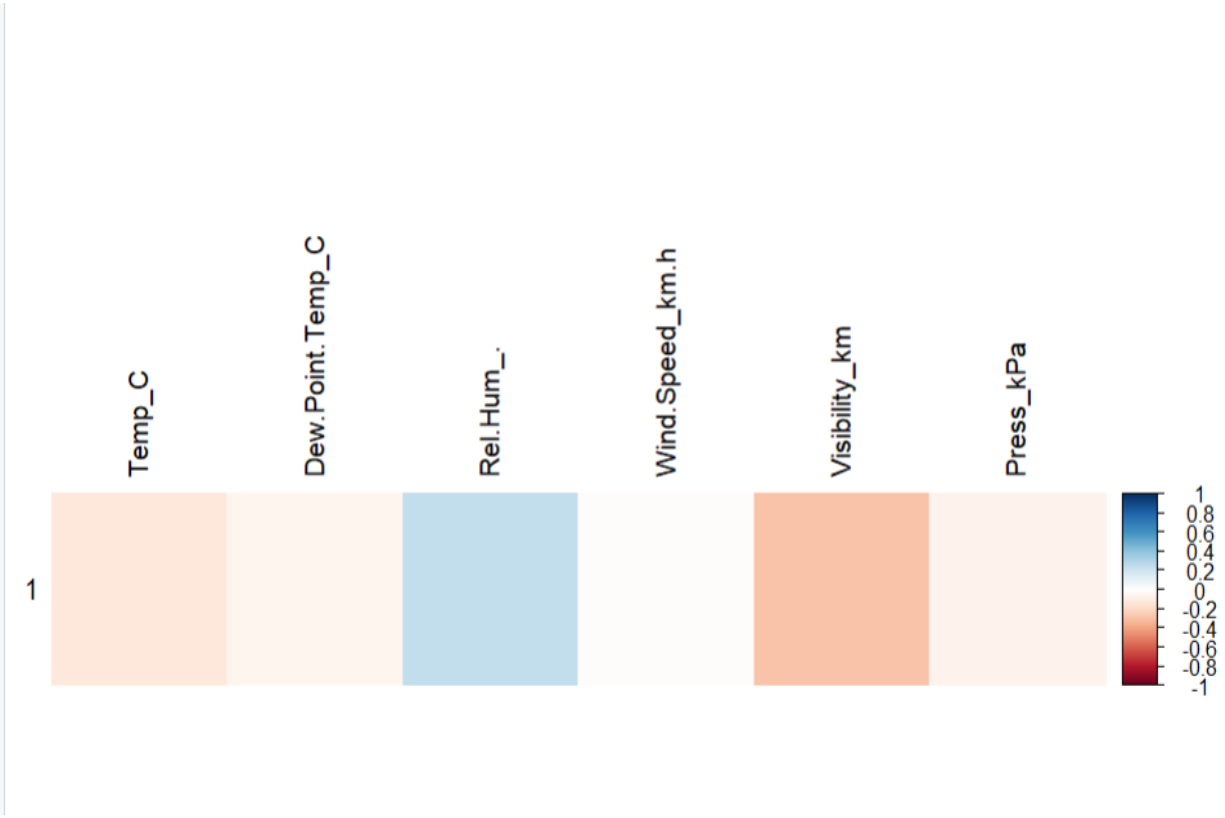


Diagram between weather and Temperature attribute

Code:

```
> library(ggplot2)

> ggplot(Weather.Data, aes(x = Weather, y = Temp_C)) +
+   geom_point() +
+   geom_smooth(method = "lm", se = FALSE, color = "blue") +
+   labs(x = "Weather", y = "Temp_C") +
+   theme_minimal()
```

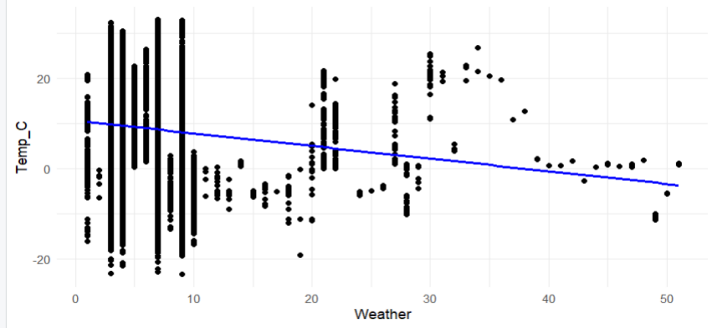
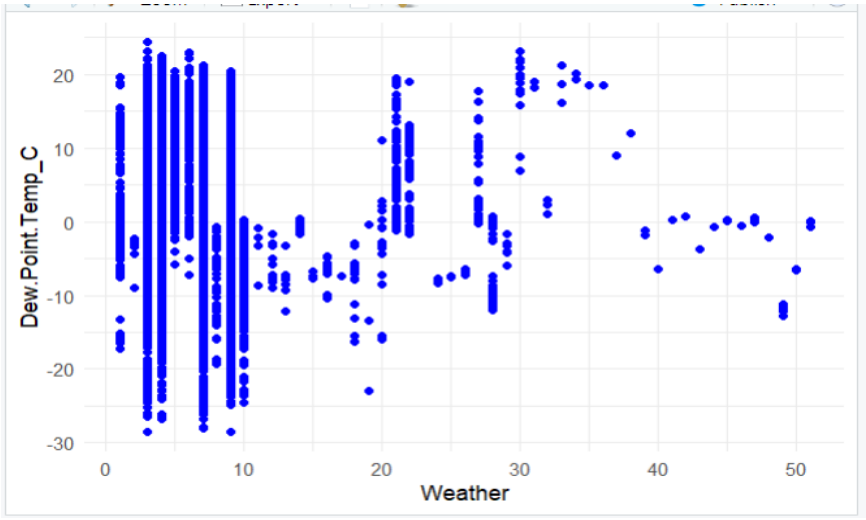


Diagram between weather and Dew.Point.Temp_Cattribute

Code:

```
> library(ggplot2)
>
>
> ggplot(Weather.Data, aes(x = Weather, y = Dew.Point.Temp_C)) +
+   geom_point(color = "blue") +
+   labs(x = "Weather", y = "Dew.Point.Temp_C") +
+   theme_minimal()
```

Output:



Split data into training and testing sets

Explanation:

After normalizing the data set, the next step is data splitting. Data splitting basically involves splitting the data set into training and testing data set. According my data set , I train data 70% and test data 30% . The sample() function generates a random sample of row indices from the normalized_data dataset, with a size determined by multiplying 70% by nrow(normalized_data). The "Weather" column is assigned to variables, and the dataset is printed, displaying training data and attributes.

Code :

```
random <- sample(1:nrow(normalized_data), 0.7 * nrow(normalized_data))

weather_train <- normalized_data[random, ]
weather_test <- normalized_data[-random, ]

weather_train_labels <- weather_train$Weather
weather_test_labels <- weather_test$Weather

weather_train

weather_test
```

Output:

```
> weather_train
         Temp_C Dew.Point.Temp_C  Rel.Hum_. Wind.Speed_km.h Visibility_km  Press_kPa Weather
7619 0.49733570       0.51417769 0.57317073      0.44578313    0.49688150 0.74225122    0.04
5922 0.88632327       0.78071834 0.29268293      0.15662651    1.00000000 0.60848287    0.04
4608 0.72824156       0.68620038 0.41463415      0.10843373    0.51559252 0.63784666    0.16
8201 0.47602131       0.46691871 0.50000000      0.10843373    0.49688150 0.58890701    0.06
6404 0.58792185       0.63894140 0.68292683      0.04819277    0.51559252 0.60522023    0.12
7451 0.42095915       0.42344045 0.53658537      0.08433735    1.00000000 0.71941272    0.04
1782 0.40852575       0.51039698 0.90243902      0.18072289    0.26403326 0.78629690    0.06
2089 0.43161634       0.33459357 0.28048780      0.07228916    0.33056133 0.66231648    0.18
4992 0.74422735       0.86956522 0.91463415      0.00000000    0.51559252 0.40293638    0.04
6055 0.60923623       0.72022684 0.89024390      0.07228916    0.49688150 0.48613377    0.12
7379 0.50444050       0.56899811 0.73170732      0.26506024    0.49688150 0.47797716    0.10
1832 0.42273535       0.54631380 0.98780488      0.00000000    0.00000000 0.78629690    0.00
5962 0.81172291       0.88846881 0.74390244      0.18072289    0.49688150 0.49265905    0.04
3552 0.72646536       0.70699433 0.47560976      0.07228916    0.51559252 0.61500816    0.04
4774 0.82238011       0.88846881 0.70731707      0.10843373    0.51559252 0.41924959    0.04
7899 0.35523979       0.36105860 0.54878049      0.39759036    0.51559252 0.42903752    0.16
4587 0.71580817       0.72400756 0.54878049      0.08433735    0.51559252 0.57748777    0.16
8485 0.42806394       0.53875236 0.92682927      0.20481928    0.39708940 0.49102773    0.06
619  0.33037300       0.27977316 0.37804878      0.28915663    0.51559252 0.61337684    0.06
6232 0.71047957       0.60680529 0.29268293      0.22891566    1.00000000 0.60848287    0.16
2647 0.55239787       0.53119093 0.46341463      0.33734940    0.49688150 0.58564437    0.04
7864 0.59857904       0.58601134 0.48780488      0.20481928    0.33056133 0.37030995    0.08
1200 0.30195382       0.31568998 0.57317073      0.08433735    0.51559252 0.65252855    0.16
6866 0.39786856       0.46691871 0.76829268      0.08433735    0.51559252 0.81402936    0.04
3108 0.69982238       0.79017013 0.79268293      0.10843373    0.49688150 0.40946166    0.06
7924 0.33392540       0.36294896 0.62195122      0.08433735    0.51559252 0.50570962    0.06
5188 0.80106572       0.81852552 0.56097561      0.07228916    0.51559252 0.66721044    0.04
5425 0.79040853       0.86767486 0.74390244      0.08433735    0.51559252 0.54323002    0.12
7957 0.40142096       0.39886578 0.51219512      0.04819277    0.49688150 0.75693312    0.12
4875 0.81172291       0.83931947 0.59756098      0.18072289    0.51559252 0.58890701    0.04
8241 0.40497336       0.48393195 0.80487805      0.13253012    0.49688150 0.75693312    0.06
7564 0.46891652       0.39508507 0.32926829      0.20481928    0.51559252 0.76998369    0.06
341  0.04085258       0.03591682 0.52439024      0.00000000    0.51559252 0.75040783    0.06
7414 0.44937833       0.44234405 0.50000000      0.15662651    0.51559252 0.61663948    0.06
4953 0.74777975       0.78071834 0.60975610      0.15662651    0.49688150 0.53507341    0.16
3384 0.83303730       0.75236295 0.34146341      0.18072289    0.51559252 0.66394780    0.12
6513 0.57904085       0.64839319 0.74390244      0.10843373    1.00000000 0.69983687    0.12
7562 0.46181172       0.40075614 0.36585366      0.15662651    0.51559252 0.78629690    0.04
6239 0.64653641       0.64083176 0.51219512      0.07228916    0.51559252 0.60032626    0.16
```

```
> weather_test
        Temp_C Dew.Point.Temp_C Rel.Hum_. Wind.Speed_km.h Visibility_km Press_kPa Weather
2   0.38188277       0.46880907 0.8414634      0.04819277    0.16216216 0.6068515    0.00
3   0.38188277       0.47448015 0.8658537      0.08433735    0.07900208 0.6101142    0.02
14  0.41740675       0.50661626 0.8414634      0.15662651    0.09563410 0.4991843    0.00
17  0.46003552       0.53497164 0.7804878      0.15662651    0.26403326 0.3931485    0.04
18  0.46714032       0.53875236 0.7682927      0.15662651    0.33056133 0.3735726    0.06
20  0.46891652       0.56332703 0.8536585      0.18072289    0.26403326 0.3523654    0.08
23  0.49200710       0.57466919 0.8048780      0.28915663    0.39708940 0.2936378    0.10
31  0.45470693       0.48960302 0.6341463      0.31325301    0.51559252 0.2707993    0.04
32  0.44937833       0.48393195 0.6341463      0.39759036    0.51559252 0.2642741    0.04
34  0.44582593       0.46880907 0.5975610      0.53012048    0.49688150 0.2642741    0.04
36  0.45293073       0.47258979 0.5853659      0.36144578    0.49688150 0.2724307    0.04
39  0.43339254       0.41020794 0.4634146      0.39759036    0.49688150 0.2952692    0.04
40  0.41385435       0.40642722 0.5000000      0.39759036    0.49688150 0.3083197    0.04
44  0.32859680       0.31001890 0.4756098      0.28915663    0.51559252 0.4045677    0.04
50  0.24156306       0.21361059 0.4390244      0.24096386    0.51559252 0.4747145    0.06
58  0.14742451       0.12476371 0.4634146      0.22891566    0.49688150 0.6003263    0.04
64  0.15097691       0.11909263 0.4268293      0.22891566    0.49688150 0.6394780    0.12
72  0.10657194       0.08128544 0.4512195      0.20481928    0.51559252 0.7096248    0.04
73  0.10301954       0.08128544 0.4634146      0.15662651    0.51559252 0.7112561    0.04
78  0.08348135       0.08506616 0.5365854      0.10843373    0.51559252 0.6933116    0.12
82  0.11545293       0.11342155 0.5243902      0.10843373    0.49688150 0.6786297    0.04
83  0.14387211       0.14933837 0.5487805      0.08433735    0.49688150 0.6557912    0.04
85  0.17051510       0.12854442 0.4024390      0.13253012    0.49688150 0.6084829    0.18
87  0.21314387       0.17958412 0.4268293      0.08433735    0.39708940 0.5628059    0.18
92  0.26465364       0.29867675 0.6463415      0.10843373    0.33056133 0.5106036    0.18
97  0.25754885       0.31758034 0.7439024      0.04819277    0.19750520 0.4567700    0.18
100 0.32504440       0.39319471 0.7682927      0.13253012    0.19750520 0.4437194    0.18
102 0.28952043       0.35916824 0.7804878      0.04819277    0.07900208 0.4355628    0.18
104 0.28774423       0.35538752 0.7804878      0.10843373    0.19750520 0.4551387    0.18
107 0.30728242       0.34971645 0.6707317      0.20481928    1.00000000 0.4779772    0.12
113 0.32504440       0.29867675 0.4512195      0.22891566    0.49688150 0.5024470    0.12
115 0.28774423       0.26654064 0.4634146      0.13253012    0.51559252 0.5203915    0.16
120 0.24689165       0.24574669 0.5243902      0.10843373    0.51559252 0.5399674    0.06
```

Accuracy

Explanation: Accuracy is the proportion of correctly predicted instances (or observations) in the testing dataset out of the total number of instances. It gives you an overall sense of the model's performance in terms of correctly classifying different classes.

Code:

```
>install.packages("class")
>library(class)
>set.seed(123)
>random <- sample(1:nrow(normalized_data), 0.7 * nrow(normalized_data))
>weather_train <- normalized_data[random, ]
```

```
>weather_test <- normalized_data[-random, ]
>weather_train_labels <- weather_train$Weather
>weather_test_labels <- weather_test$Weather
>
>k <- 3
>predicted_labels <- knn(train = weather_train[, -which(names(weather_train) == "Weather")],
                        test = weather_test[, -which(names(weather_test) == "Weather")],
                        cl = weather_train_labels,
                        k = k)

>accuracy <- sum(predicted_labels == weather_test_labels) / length(weather_test_labels)

>cat("Accuracy:", accuracy, "\n")
```

Output:

```
> cat("Accuracy:", accuracy, "\n")
Accuracy: 0.4362671
>
```

10 fold cross validation:

Explanation: The code evaluates K-nearest neighbors classifier performance using 10-fold cross-validation, training and testing on dataset subsets, and calculating mean accuracy. It estimates generalization to new data and assesses predictive accuracy.

Code:

```
> install.packages("class")
library(class)
install.packages("caret")
library(caret)
set.seed(123)
num_folds <- 10
fold_indices <- createFolds(normalized_data$Weather, k = num_folds)
accuracies <- numeric(num_folds)
for (i in 1:num_folds) {

  test_indices <- fold_indices[[i]]
```

```
  train_indices <- setdiff(1:nrow(normalized_data), test_indices)

  weather_train <- normalized_data[train_indices, ]
  weather_test <- normalized_data[test_indices, ]


  input_features_train <- weather_train[, c("Temp_C", "Dew.Point.Temp_C", "Rel.Hum_.",
                                            "Wind.Speed_km.h", "Visibility_km", "Press_kPa")]
  input_features_test <- weather_test[, c("Temp_C", "Dew.Point.Temp_C", "Rel.Hum_.",
                                          "Wind.Speed_km.h", "Visibility_km", "Press_kPa")]
  weather_train_labels <- weather_train$Weather
  weather_test_labels <- weather_test$Weather


  k <- 3
  predicted_labels <- knn(train = input_features_train,
                          test = input_features_test,
                          cl = weather_train_labels,
                          k = k)


  accuracies[i] <- sum(predicted_labels == weather_test_labels) / length(weather_test_labels)
}

mean_accuracy <- mean(accuracies)
cat("Mean Accuracy (10-Fold Cross-Validation):", mean_accuracy, "\n")
```

Output:

```
> mean_accuracy <- mean(accuracies)
> cat("Mean Accuracy (10-Fold Cross-Validation):", mean_accuracy, "\n")
Mean Accuracy (10-Fold Cross-Validation): 0.4650588
```

Confusion matrix:

Explanation: Generating confusion matrix and reporting recall and precision values for KNN classifier helps assess performance, understand behavior, and make informed decisions for model adjustments and improvements.

Code:

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

set.seed(123)
num_folds <- 10

fold_indices <- createFolds(normalized_data$Weather, k = num_folds)

confusion_matrices <- list()
recalls <- numeric(num_folds)
precisions <- numeric(num_folds)
calculate_metrics <- function(cm) {
  recall <- cm[1, 1] / sum(cm[1, ])
  precision <- cm[1, 1] / sum(cm[, 1])
  return(list(recall = recall, precision = precision))
}

for (i in 1:num_folds) {
  test_indices <- fold_indices[[i]]
  train_indices <- setdiff(1:nrow(normalized_data), test_indices)

  weather_train <- normalized_data[train_indices, ]
  weather_test <- normalized_data[test_indices, ]

  input_features_train <- weather_train[, c("Temp_C", "Dew.Point.Temp_C", "Rel.Hum_.",
                                            "Wind.Speed_km.h", "Visibility_km", "Press_kPa")]
  input_features_test <- weather_test[, c("Temp_C", "Dew.Point.Temp_C", "Rel.Hum_.",
                                          "Wind.Speed_km.h", "Visibility_km", "Press_kPa")]
  weather_train_labels <- weather_train$Weather
  weather_test_labels <- weather_test$Weather

  k <- 3
  predicted_labels <- knn(train = input_features_train,
                          test = input_features_test,
```

```
                cl = weather_train_labels,
                k = k)

  confusion_matrices[[i]] <- table(predicted = predicted_labels, actual = weather_test_labels)

  metrics <- calculate_metrics(confusion_matrices[[i]])
  recalls[i] <- metrics$recall
  precisions[i] <- metrics$precision
}

mean_recall <- mean(recalls)
mean_precision <- mean(precisions)

cat("Mean Recall:", mean_recall, "\n")
cat("Mean Precision:", mean_precision, "\n")

for (i in 1:num_folds) {
  cat("Confusion Matrix (Fold", i, "):\n")
  print(confusion_matrices[[i]])
  cat("\n")
}
```

Output:

```
> cat("Mean Recall:", mean_recall, "\n")
Mean Recall: 0.6001523
> cat("Mean Precision:", mean_precision, "\n")
Mean Precision: 0.7025136
>
> # Print individual confusion matrices for eac
```

Confusion Matrix (Fold 1):

predicted \ actual	0	0.04	0.06	0.08	0.1	0.12	0.14	0.16	0.18	0.22	0.24	0.26	0.28	0.3	0.34	0.4	0.42	0.46	0.48	0.5	0.52	0.54	0.58	0.62	0.84	0.92
0	9	0	1	4	0	0	0	0	1	0	0	0	0	0	0	1	2	0	0	0	0	0	0	0	0	0
0.02	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0.04	0	89	41	3	4	49	2	17	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0.06	0	46	81	3	4	28	0	10	5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0.08	1	4	2	13	5	4	0	1	1	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0
0.1	0	5	9	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0
0.12	0	47	22	2	1	92	0	37	3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0.14	0	3	1	1	0	0	0	0	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0.16	0	18	13	0	2	41	0	73	3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0.18	0	0	4	0	0	0	0	2	16	0	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0
0.2	0	0	0	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0.22	0	0	0	0	0	0	0	0	0	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0.24	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0.26	0	0	0	1	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0.28	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0
0.3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0.32	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0.34	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0
0.36	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0.38	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0
0.4	0	0	1	3	0	0	0	0	0	0	0	0	0	0	0	3	2	0	0	0	0	0	0	0	0	0
0.42	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	1	7	0	0	0	0	0	0	0	0	0
0.46	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0
0.48	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0.5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0

Confusion Matrix (Fold 2):

predicted \ actual	0	0.02	0.04	0.06	0.08	0.1	0.12	0.14	0.16	0.18	0.22	0.32	0.34	0.36	0.38	0.4	0.42	0.46	0.52	0.54	0.58	0.6	0.62	0.72	0.74	0.88	0.92
0	9	0	0	0	0	1	0	0	0	0	0	0	0	0	2	1	1	0	1	0	0	0	1	0	0	1	0
0.02	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0.04	0	0	88	30	3	5	59	1	21	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0.06	0	0	42	95	13	8	23	1	7	3	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0
0.08	1	0	6	4	12	0	1	0	0	0	1	0	0	0	0	2	0	0	0	0	0	1	0	0	0	0	0
0.1	0	0	4	5	2	2	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0
0.12	0	0	50	18	0	1	88	0	31	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0
0.14	0	0	1	1	0	0	3	1	1	3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0.16	0	0	16	11	0	4	38	0	78	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0.18	0	0	1	3	0	0	0	0	1	19	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0.2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0.22	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0
0.24	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0.26	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0.28	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0.3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0.34	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0.36	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0.38	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2	0	0	0	0	0	0	0	0	0	0	0	0
0.4	1	0	0	0	0	1	0	0	0	0	0	0	0	0	10	0	0	0	0	0	0	0	0	0	1	0	0
0.42	1	0	0	0	0	0	0	0	0	0	0	0	0	0	2	6	0	0	0	1	0	0	1	0	0	0	0
0.46	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0
0.48	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0.5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Confusion Matrix (Fold 3):

predicted \ actual	0	0.04	0.06	0.08	0.1	0.12	0.14	0.16	0.18	0.2	0.24	0.26	0.3	0.34	0.38	0.4	0.42	0.52	0.54	0.56	0.78	0.92
0	13	0	0	0	0	0	1	0	1	0	0	0	0	0	0	1	1	0	0	0	0	0
0.02	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0.04	0	75	38	0	2	58	0	20	2	0	0	0	0	0	0	0	0	0	0	0	0	0
0.06	1	34	92	6	5	28	0	21	7	0	0	0	0	0	0	0	1	0	0	0	0	0
0.08	0	4	6	20	0	0	0	2	1	0	0	1	0	0	0	1	0	0	0	0	0	0
0.1	0	3	3	2	1	1	0	3	0	0	0	0	0	0	0	0	1	0	0	0	0	0
0.12	0	66	11	3	5	90	0	33	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0.14	0	1	1	0	0	0	1	0	2	0	0	0	0	0	0	0	0	0	0	0	0	0
0.16	0	19	12	1	0	36	1	58	2	0	0	0	0	0	0	0	0	0	0	0	0	0
0.18	0	1	4	2	0	0	1	0	27	0	0	0	1	0	0	2	0	0	1	1	1	0
0.2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0.22	0	0	1	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0
0.24	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0.26	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0
0.28	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0.3	1	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0
0.32	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0.34	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
0.36	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0.38	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0
0.4	2	0	0	5	1	0	0	0	0	0	0	0	0	0	0	5	2	0	0	0	0	0
0.42	3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2	0	0	0	0	0
0.46	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0
0.48	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0.5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0.52	0	0	0	2	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0
0.54	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	5	0	0	0
0.56	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	1	0	0

Confusion Matrix (Fold 4):

predicted \ actual	0	0.04	0.06	0.08	0.1	0.12	0.14	0.16	0.18	0.22	0.24	0.26	0.3	0.34	0.36	0.38	0.4	0.42	0.48	0.5	0.52	0.54	0.58	0.66	0.76	0.9	0.92	0.94	0.96	1
0	12	0	1	1	0	0	0	0	0	0	0	0	0	0	1	0	4	2	0	0	0	0	0	0	0	0	0	0	0	0
0.02	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0.04	0	91	38	1	5	46	0	14	2	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0
0.06	0	41	85	6	9	18	0	11	6	0	0	1	0	0	0	0	0	0	0	0	4	0	0	0	0	0	1	0	0	0
0.08	2	4	6	14	2	0	0	0	2	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	1	0	0	0	0
0.1	1	1	3	1	2	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0
0.12	0	43	18	2	1	102	0	34	4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0
0.14	0	0	2	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0.16	0	23	15	0	1	46	0	68	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0.18	1	0	2	1	0	0	4	1	19	0	0	1	2	2	0	0	0	1	0	0	0	0	0	0	0	0	0	0	1	1
0.2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0.22	0	0	0	1	0	0	0	0	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0.24	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0.26	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0.28	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0.3	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0.32	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0.34	0	0	0	0	0	0	0	0	1	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0.36	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0.38	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0.4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	3	4	0	0	0	0	0	0	0	0	0	0	0	0
0.42	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	2	0	0	0	0	0	0	0	0	0	0	0	0
0.46	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0
0.48	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0.5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	2	0	0	0	0	0	0	0	0	0	0
0.52	0	0	2	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Confusion Matrix (Fold 5):

predicted \ actual	0	0.04	0.06	0.08	0.1	0.12	0.14	0.16	0.18	0.22	0.24	0.26	0.3	0.34	0.38	0.4	0.42	0.52	0.54	0.56	0.58	0.7	0.82	0.98
0	11	0	0	1	0	0	0	0	1	0	0	0	0	0	0	2	2	0	0	0	0	0	0	0
0.02	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0.04	0	80	43	2	3	62	1	18	2	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0
0.06	0	43	82	4	3	20	4	13	4	1	0	0	0	0	0	0	0	2	0	0	0	0	0	0
0.08	0	4	6	11	5	2	0	0	1	0	0	1	0	0	0	5	0	2	0	1	0	0	0	0
0.1	1	4	2	0	3	2	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0.12	0	57	33	0	2	98	0	36	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0.14	0	2	1	1	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0.16	0	15	11	1	2	24	1	55	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0.18	2	2	2	0	0	0	2	0	29	1	1	0	3	2	0	0	0	0	1	0	0	0	0	0
0.2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0.22	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0.24	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0.26	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0.28	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0.3	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0.32	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0.34	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	1
0.36	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0.38	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
0.4	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	8	0	0	1	0	0	0	1	0
0.42	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	3	0	0	0	0	0	0	0
0.46	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0.48	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0.5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0.52	0	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0

Confusion Matrix (Fold 6):

predicted \ actual	0	0.04	0.06	0.08	0.1	0.12	0.14	0.16	0.18	0.22	0.24	0.26	0.3	0.34	0.38	0.4	0.42	0.5	0.52	0.54	0.62	0.64	0.68	0.8	0.88	0.92
0	14	0	0	1	0	0	0	0	0	0	0	0	0	0	2	1	1	0	0	0	0	0	0	0	0	0
0.02	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0.04	0	104	35	1	3	48	1	16	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0.06	0	33	84	12	7	18	1	14	3	0	1	0	1	0	0	0	0	0	3	0	0	0	0	0	0	0
0.08	1	2	7	15	2	1	0	0	0	0	0	1	0	0	0	2	0	0	2	0	0	0	0	0	1	0
0.1	0	2	4	1	2	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0
0.12	0	43	26	0	5	101	0	31	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0.14	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0.16	0	19	7	0	2	42	0	63	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0.18	0	1	2	0	0	1	1	1	28	0	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0
0.2	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0.22	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0.24	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0.26	0	0	0	1	0	0	0	0	0	0	0	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0.28	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0.3	0	0	0	0	0	0	0	0	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0.32	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0.34	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0
0.36	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0.38	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0.4	1	0	0	1	1	0	0	0	0	0	0	0	0	0	0	5	1	0	0	1	0	0	1	1	0	0
0.42	1	0	0	1	0	0	0	0	1	0	0	0	0	0	0	2	1	0	1	0	1	0	0	0	0	1
0.46	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Confusion Matrix (Fold 7):

predicted \ actual	0	0.02	0.04	0.06	0.08	0.1	0.12	0.14	0.16	0.18	0.2	0.22	0.26	0.3	0.34	0.4	0.42	0.52	0.54	0.56	0.58	0.92	0.96	1
0	5	0	0	1	1	0	0	0	1	2	0	0	0	0	0	0	0	0	0	0	0	0	2	1
0.02	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0.04	0	0	85	36	4	2	64	4	28	4	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0.06	0	0	45	85	11	4	28	2	7	5	0	0	0	0	0	0	0	0	0	0	1	0	0	0
0.08	0	0	3	8	9	2	0	0	0	0	0	0	1	0	0	1	0	2	0	0	2	0	0	0
0.1	0	0	6	8	0	4	4	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0.12	0	0	47	24	0	0	87	0	28	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0
0.14	0	0	1	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0.16	0	0	21	16	0	0	27	0	64	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0.18	0	1	4	3	0	1	1	0	2	24	1	1	1	0	0	0	0	0	0	1	0	0	0	0
0.2	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0.22	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0.24	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0.26	1	0	0	0	0	0	0	0	0	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0
0.28	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0.3	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0.32	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0.34	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0.36	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0.38	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0
0.4	2	0	0	0	1	0	0	0	0	0	0	0	0	0	0	8	2	0	2	0	0	0	0	0
0.42	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	3	2	0	0	0	0	0	0
0.46	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0.48	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0.5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0.52	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1	2	0	0	0	0	0
0.54	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2	0	0	0	1	0

Confusion Matrix (Fold 8):

predicted \ actual	0	0.02	0.04	0.06	0.08	0.1	0.12	0.14	0.16	0.18	0.22	0.24	0.26	0.28	0.3	0.34	0.36	0.4	0.42	0.46	0.5	0.52	0.54	0.58	0.64	0.76	0.92
0	16	0	0	1	0	0	0	0	1	1	0	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0
0.02	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
0.04	0	0	92	39	3	7	54	2	11	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0.06	0	0	33	88	10	6	21	3	16	3	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0
0.08	1	0	1	11	15	1	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	2	0	0	0	0	0
0.1	0	0	0	1	2	0	5	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0
0.12	0	0	54	14	1	1	98	1	34	1	0	0	0	0	0	0	0	1	0	0	0	0	0	1	0	0	0
0.14	0	0	2	1	0	0	0	2	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0.16	0	0	15	16	0	1	28	0	72	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0.18	0	1	1	2	0	0	0	0	1	24	0	1	1	2	1	2	0	0	0	0	1	1	1	0	0	1	0
0.2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0.22	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0.24	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0.26	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
0.28	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0.3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0.32	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0.34	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0
0.36	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0.38	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0.4	4	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	5	1	0	0	0	0	0	0	0	0
0.42	1	0	0	0	1	0	0	0	0	1	0	0	0	0	0	0	0	1	8	0	0	0	0	0	0	0	1
0.46	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
0.48	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0.5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0.52	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Confusion Matrix (Fold 9):

predicted \ actual	0	0.02	0.04	0.06	0.08	0.1	0.12	0.14	0.16	0.18	0.24	0.26	0.28	0.3	0.34	0.36	0.4	0.42	0.52	0.54	0.58	0.86	0.92	1
0	5	0	0	0	1	1	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0
0.02	0	1	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0
0.04	0	0	92	39	2	3	39	1	18	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0.06	0	0	52	88	6	5	20	5	8	8	0	0	0	0	0	0	0	0	2	0	0	0	0	0
0.08	0	0	0	8	13	2	0	0	0	1	0	0	0	0	0	0	2	0	0	0	0	0	0	0
0.1	0	0	3	7	1	4	3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0.12	0	0	52	20	1	3	109	0	37	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0.14	0	0	0	2	0	0	0	2	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0.16	0	0	16	8	0	1	32	3	77	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0.18	0	0	0	3	0	0	2	1	0	20	0	0	0	0	2	0	0	0	0	1	0	0	0	0
0.2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0.22	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0
0.24	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0.26	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	1
0.28	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0
0.3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0.32	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0.34	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0.36	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0.38	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0
0.4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	10	4	1	0	0	0	0
0.42	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	1	4	0	0	0	0	0
0.46	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0.48	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0.5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0.52	0	0	0	0	2	1	0	0	0	0	0	0	0	0	0	0	0	0	2	0	0	0	0	0

Confusion Matrix (Fold 10):

predicted \ actual	0	0.02	0.04	0.06	0.08	0.1	0.12	0.14	0.16	0.18	0.2	0.22	0.26	0.38	0.4	0.42	0.52	0.54	0.56	0.58	0.6	0.64	0.66	0.92	0.98	1
0	11	0	0	0	1	0	0	0	1	1	0	0	0	1	1	3	0	1	0	0	0	0	0	0	0	0
0.02	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0.04	0	0	94	49	6	3	48	2	15	2	0	0	0	1	0	0	1	0	0	0	0	0	1	0	0	0
0.06	0	0	43	75	9	11	17	0	14	7	0	0	0	0	0	0	1	0	0	2	1	0	0	0	0	0
0.08	1	0	1	7	12	2	0	0	1	2	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0
0.1	0	0	1	3	0	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0.12	0	0	46	19	2	4	114	1	30	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0.14	0	0	1	1	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0.16	0	0	14	9	0	1	30	0	61	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0.18	1	0	3	4	0	1	1	3	0	29	0	1	0	0	0	0	1	0	0	0	0	0	0	0	0	0
0.2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0.22	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0.24	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0.26	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0
0.28	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0.3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0.32	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0.34	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0
0.36	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0.38	0	0	0	0	0	0	0	0	0	0	0	0	0	2	2	0	0	0	0	0	0	0	0	0	0	0
0.4	2	0	0	0	1	0	0	0	0	0	0	0	0	0	7	0	0	0	0	0	0	0	0	0	0	0
0.42	3	1	0	0	0	0	0	0	0	0	0	0	0	0	2	7	0	0	0	0	0	0	0	0	0	0
0.46	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0.48	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0.5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0.52	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0