**American International University Bangladesh**
**Department of Computer Science**
**Project Report**

---

**Semester: Spring’21-22**
**Section: D**
**Course Name: Data Warehousing And Data Mining**

**Submitted To,**
*Dr. Akinul Islam Jony*
*Assistant Professor*
*Department of Computer Science*
*American International University-Bangladesh*

# TITLE: WEATHER FORECASTING USING DATA MINING APPLICATION

## GROUP MEMBERS INFORMATION

| NAME | ID |
|---|---|
| NAJMUL UDDIN | 18-38293-2 |
| SUMONA ISLAM | 18-37288-1 |
| MD. MILKAN ISLAM | 18-36958-1 |
| MAHZABIN MOSTARY | 18-38606-2 |

# Introduction

## Objective

Data mining is the process of uncovering patterns and finding anomalies and relationships in large datasets that can be used to make predictions about future trends. The main objective of data mining is to extract valuable information from available data. This generally involves the use of database techniques such as spatial indexes. Thus, these patterns can be seen as a kind of summary of the input data.In addition to being able to be used in additional analysis or, for example, in machine learning and predictive analysis. One of the examples we can give is data mining. This could identify several groups in the data, which can then be used to obtain more accurate results being able to predict problems through a decision support system. Neither data collection, data preparation,nor interpretation of results and information is part of the data mining stage.

## Description

Data mining involves the use of complicated data analysis tools to discover previously unknown, interesting patterns and relationships in large data set. These tools can include statistical models, mathematical algorithm and machine learning methods. The most significant analysis phase in the knowledge discovery in database (KDD) process is data mining. The basic purpose of data mining is to extract meaningful information from large amounts of raw data and convert it into a format that can be used effectively and efficiently. Data mining jobs are typically classified into two types: descriptive and predictive classification techniques. Data classification is the process of organizing data into categories/groups in such a way that data objects of same group are more similar and data objects from different groups are very dissimilar. The classification algorithm assigns a classification each instance to a certain class so that classification is possible. There will be the least amount of mistake. It's utilized to create models that are correct. Classification techniques can handle processing of large volume of data. It can predict categorical class labels and classifies data based on model built by using training set and associated class labels and then can be used for classifying newly available test data. Thus, it is outlined as an integral part of data analysis and is gaining more popularity.

## Outcome

1. To fully understand standard data mining methods and techniques such as association rules, data clustering and classification.

2. Learn new, advanced techniques for emerging applications (social network analysis, stream data mining).

3. Gain practical intuition about how to apply these techniques on datasets of realistic sizes using modern data analysis frameworks.

# Dataset URL

**https://github.com/dbnnomanbhai/WEATHER-DATASET-FOR-DATA-MINING/blob/main/weather.csv**

## Data processing Steps

The data processing and filtering consists of the following stages.

### Data Source

A data source is the location where data that is being used originates from. A data source may be the initial location where data is born or where physical information is first digitized, however even the most refined data may serve as a source, as long as another process accesses and utilizes it.

### Data Store

A data store is a repository for persistently storing collections of data, such as a database, a file system or a directory. In an information technology context, data stored can be of any type that can be rendered in digital format and placed in electronic media.

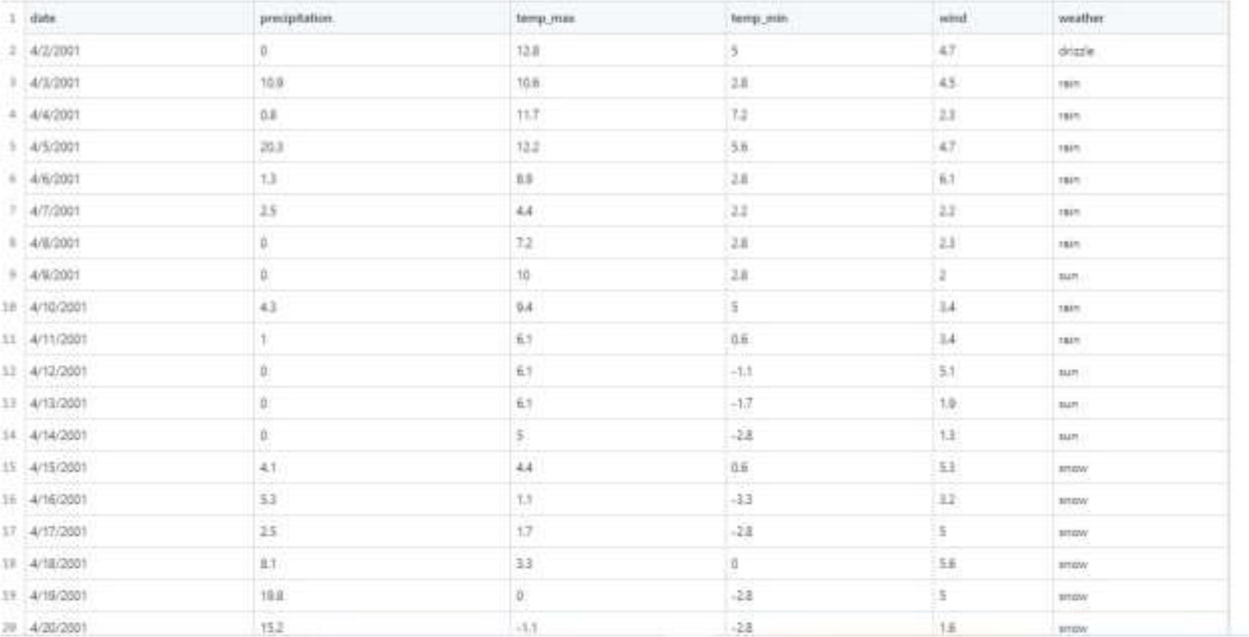
### Prepared Data

Once data store is completed then data prepared for selecting and preprocessing.

### Patterns

When data prepared for selecting and preprocessing after that data make  patterns for interpretation and assimilation.

### Dataset Description



| 1 | date | precipitation | temp_max | temp_min | wind | weather |
|---|---|---|---|---|---|---|
| 2 | 4/2/2001 | 0 | 12.8 | 5 | 4.7 | drizzle |
| 3 | 4/3/2001 | 10.9 | 10.6 | 2.8 | 4.5 | rain |
| 4 | 4/4/2001 | 0.8 | 11.7 | 7.2 | 2.3 | rain |
| 5 | 4/5/2001 | 20.3 | 12.2 | 5.6 | 4.7 | rain |
| 6 | 4/6/2001 | 1.3 | 8.9 | 2.8 | 6.1 | rain |
| 7 | 4/7/2001 | 2.5 | 4.4 | 2.2 | 2.2 | rain |
| 8 | 4/8/2001 | 0 | 7.2 | 2.8 | 2.3 | rain |
| 9 | 4/9/2001 | 0 | 10 | 2.8 | 2 | sun |
| 10 | 4/10/2001 | 4.3 | 9.4 | 5 | 3.4 | rain |
| 11 | 4/11/2001 | 1 | 6.1 | 0.6 | 3.4 | rain |
| 12 | 4/12/2001 | 0 | 6.1 | -1.1 | 5.1 | sun |
| 13 | 4/13/2001 | 0 | 6.1 | -1.7 | 1.9 | sun |
| 14 | 4/14/2001 | 0 | 5 | -2.8 | 1.3 | sun |
| 15 | 4/15/2001 | 4.1 | 4.4 | 0.6 | 5.3 | snow |
| 16 | 4/16/2001 | 5.3 | 1.1 | -3.3 | 3.2 | snow |
| 17 | 4/17/2001 | 2.5 | 1.7 | -2.8 | 5 | snow |
| 18 | 4/18/2001 | 8.1 | 3.3 | 0 | 5.6 | snow |
| 19 | 4/19/2001 | 19.8 | 0 | -2.8 | 5 | snow |
| 20 | 4/20/2001 | 15.2 | -1.1 | -2.8 | 1.6 | snow |

***FIGURE : SAMPLE OF DATASET***

To predict the weather, we need to measure the weather. If you want to know the weather is like tomorrow, it's pretty important to know what the weather was like today and yesterday. Knowing the average weather on a particular day of the year is also useful. Collecting data every day can show you patterns and trends, and help you figure out how our atmosphere works.

In here, it’s about 5236 instances in 5236 days and for each of these days we have 6 values of 6 attributes. We are going to be doing is predicting the weather attribute from the other attributes.

## Dataset Exploration

Data exploration refers to the initial step in data analysis in which data analysts use data visualization and statistical techniques to describe dataset characterizations, such as size, quantity, and accuracy, in order to better understand the nature of the data. Classification problem sometimes called supervised learning problem. We take as input a data set of classified examples. These examples are independent example with a class value attached and the idea is to produce automatically a model that is some kind of model that can classify your examples. That’s the classification problem. An instance with the different attribute values a fixed set of feature and then we add to that a class to get a classified example. That’s what we have in our training data set. These attributes can be discrete of continuous. What we looked at in the weather data was discrete or we call them normal attribute values where they belong to a certain fixed set well. And also the class can be discrete or continuous.

## How We Deal With Missing Value in Our Datasets

Incomplete data is an unavoidable problem in dealing with most of the real world data sources. There are two most used methods for dealing with missing values. In our project we used repeated by most frequent/average value. To estimate each of the missing values is a less cautious method, using the values is present in the datasets. For a categorical attribute, the most commonly occurring (non-missing) value is a simple yet effective technique to do this. This is easy to justify if the attribute values are very unbalanced. For example, if attribute X has possible values a, b and c which occur in proportions 80%, 15% and 5% respectively, it seems reasonable to estimate any missing values of attribute X by the value a. If the values are more evenly distributed, say in proportions 40%, 30% and 30%, the validity of this approach is much less clear

## 1. Implementation of Decision Tree

### About Decision Tree

Decision tree is the most powerful and popular tool for classification and prediction. A Decision tree is a flowchart like tree structure, where each internal node denotes a test on an attribute, each branch represents an outcome of the test, and each leaf node (terminal node) holds a class label. Decision Trees (DTs) are a non-parametric supervised learning method used for classification and regression. The goal is to create a model that predicts the value of a target variable by learning simple decision rules inferred from the data features. A tree can be seen as a piecewise constant approximation. A decision tree is a very specific type of probability tree that enables you to make a decision about some kind of process.

### Representation of Decision Tree

Decision trees classify instances by sorting them down the tree from the root to some leaf node, which provides the classification of the instance. An instance is classified by starting at the root node of the tree, testing the attribute specified by this node, then moving down the tree branch corresponding to the value of the attribute as shown in the above figure. This process is then repeated for the subtree rooted at the new node. The decision tree in above figure classifies a particular morning according to whether it is suitable for playing tennis and returning the classification associated with the particular leaf.(in this case Yes or No). For example, the instance (Outlook = Rain, Temperature = Hot, Humidity = High, Wind = Strong ) would be sorted down the leftmost branch of this decision tree and would therefore be classified as a negative instance. In other words we can say that decision tree represent a disjunction of conjunctions of constraints on the attribute values of instances.

## The Strengths of Decision Tree

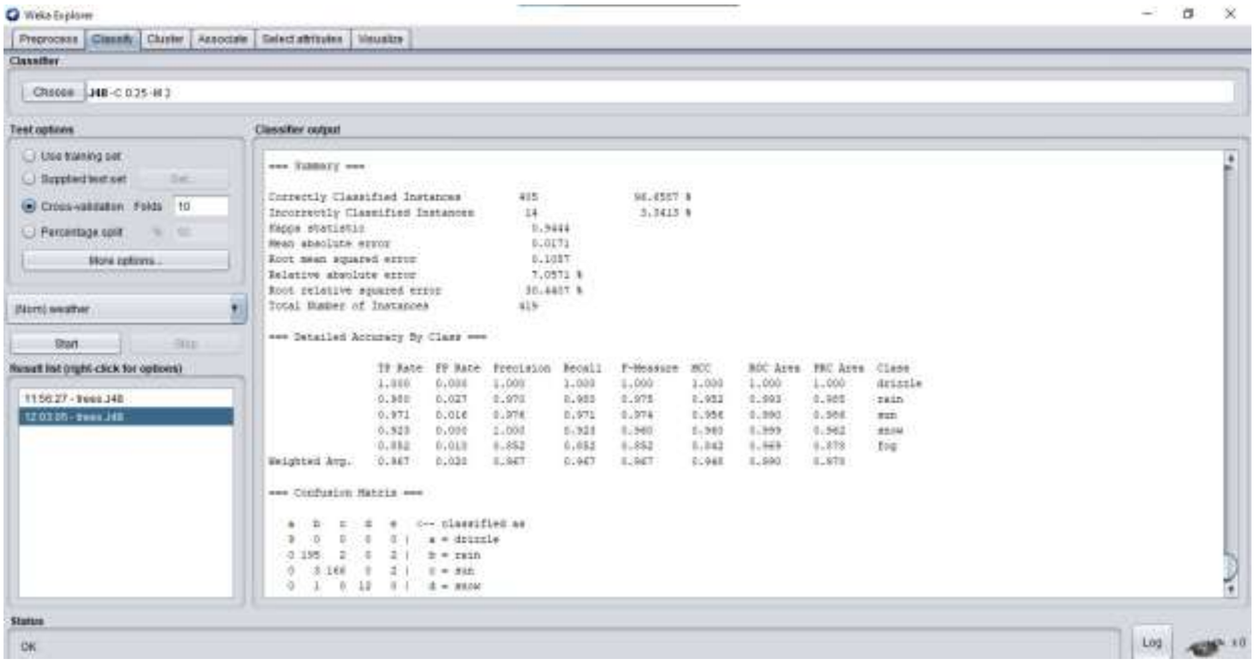
- Decision trees are able to generate understandable rules.
- Decision trees perform classification without requiring much computation.
- Decision trees are able to handle both continuous and categorical variables.
- Decision trees provide a clear indication of which fields are most important for prediction or classification.

## The weaknesses of Decision Tree Methods

- Decision trees are less appropriate for estimation tasks where the goal is to predict the value of a continuous attribute.
- Decision trees are prone to errors in classification problems with many class and relatively small number of training examples.
- Decision tree can be computationally expensive to train. The process of growing a decision tree is computationally expensive must be formed and compared.

## MODEL DEVELOPMENT BY WEKA TOOLS

Apply the weather data in Weka.  Explain the steps on how to implement   decision Tree in Weka Tool.



J48 -C 0.25 -M 2

○ Use training set
○ Supplied test set  Set...
● Cross-validation  Folds 10
○ Percentage split  % 66
More options...

(Nom) weather

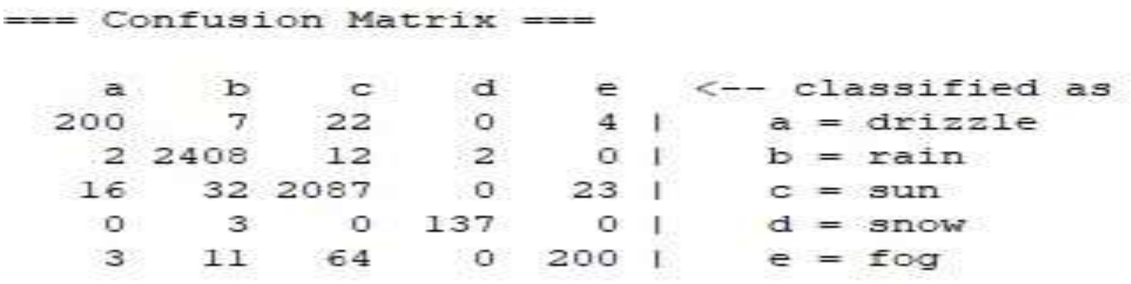
11:56:27 - trees.J48
12:03:05 - trees.J48

```
=== Summary ===

Correctly Classified Instances         405               96.6587 %
Incorrectly Classified Instances        14                3.3413 %
Kappa statistic                          0.9444
Mean absolute error                      0.0171
Root mean squared error                  0.1037
Relative absolute error                  7.0571 %
Root relative squared error             30.4407 %
Total Number of Instances              419

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 1.000    0.000    1.000      1.000    1.000      1.000    1.000     1.000     drizzle
                 0.980    0.027    0.970      0.980    0.975      0.952    0.993     0.985     rain
                 0.971    0.016    0.976      0.971    0.974      0.956    0.990     0.968     sun
                 0.923    0.000    1.000      0.923    0.960      0.960    0.999     0.962     snow
                 0.852    0.013    0.852      0.852    0.852      0.842    0.969     0.878     fog
Weighted Avg.    0.967    0.033    0.967      0.967    0.967      0.948    0.990     0.978

=== Confusion Matrix ===

   a   b   c   d   e   <-- classified as
   9   0   0   0   0 |   a = drizzle
   0 195   2   0   2 |   b = rain
   0   3 166   0   2 |   c = sun
   0   1   0  12   0 |   d = snow
```

OK

*FIGURE 01*

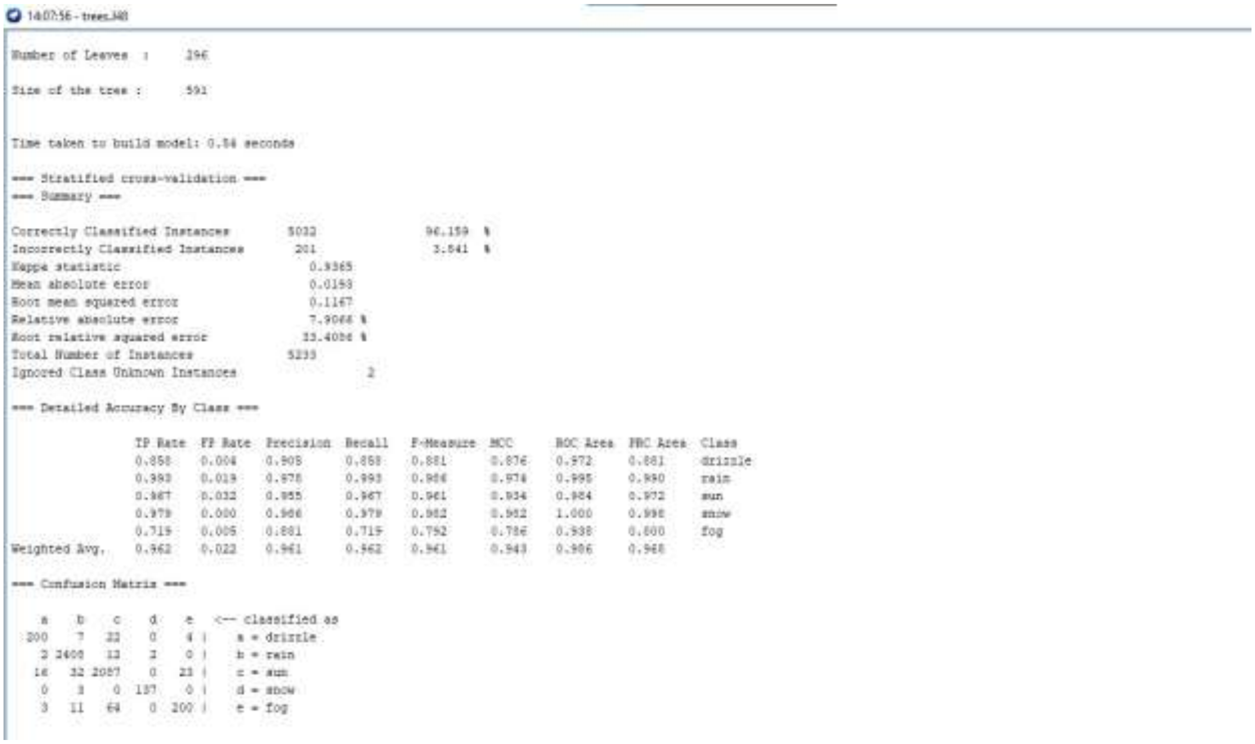
AFTER SELECTING DATA ,FOR DECISSION TREE WE SELECTED J48.AFTER MINING THE DATA OR DECISSION TREE WE GET 96.65% ACCURACY INSTEAD WE GET 3.35% INCORRECTLY CLASSFIED INSTANCE



```
=== Confusion Matrix ===

    a     b     c     d     e   <-- classified as
  200     7    22     0     4 |     a = drizzle
    2  2408    12     2     0 |     b = rain
   16    32  2087     0    23 |     c = sun
    0     3     0   137     0 |     d = snow
    3    11    64     0   200 |     e = fog
```

***FIGURE 02***

HERE IS OUR CONFUSION MATRIX. WHICH IS COMPARE ACTUAL VALUE WITH PREDICTED VALUE



```
Number of Leaves  :     296

Size of the tree :      591


Time taken to build model: 0.84 seconds

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances        5032               96.159  %
Incorrectly Classified Instances       201                3.841  %
Kappa statistic                          0.9365
Mean absolute error                      0.0158
Root mean squared error                  0.1167
Relative absolute error                  7.9066 %
Root relative squared error             33.4056 %
Total Number of Instances             5233
Ignored Class Unknown Instances                  2

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.858    0.004    0.905      0.858    0.881      0.876    0.972     0.881     drizzle
                 0.993    0.019    0.978      0.993    0.986      0.974    0.995     0.990     rain
                 0.967    0.032    0.955      0.967    0.961      0.934    0.984     0.972     sun
                 0.979    0.000    0.986      0.979    0.982      0.982    1.000     0.998     snow
                 0.719    0.005    0.881      0.719    0.792      0.786    0.938     0.800     fog
Weighted Avg.    0.962    0.022    0.961      0.962    0.961      0.943    0.986     0.968

=== Confusion Matrix ===

    a    b    c    d    e   <-- classified as
  200    7   22    0    4 |    a = drizzle
    2 2408   12    2    0 |    b = rain
   16   32 2087    0   23 |    c = sun
    0    3    0  137    0 |    d = snow
    3   11   64    0  200 |    e = fog
```

***FIGURE 03***

HERE HAS BEEN DESCRIBED, SIZE OF THE TREE IS 591 .WE GOT 0.9365 MEAN ABSOLUTE ERROR.HERE WE HAVE DISPLAY,ALL THE CLASSIFIED OF DECISSION TREE

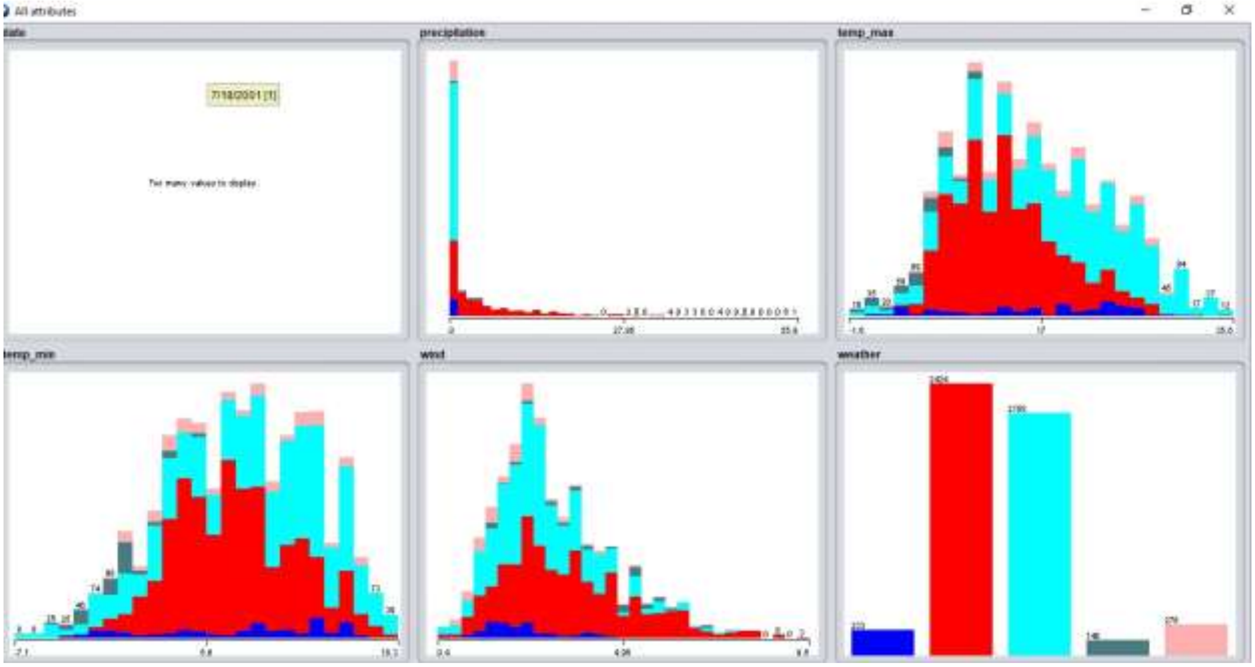
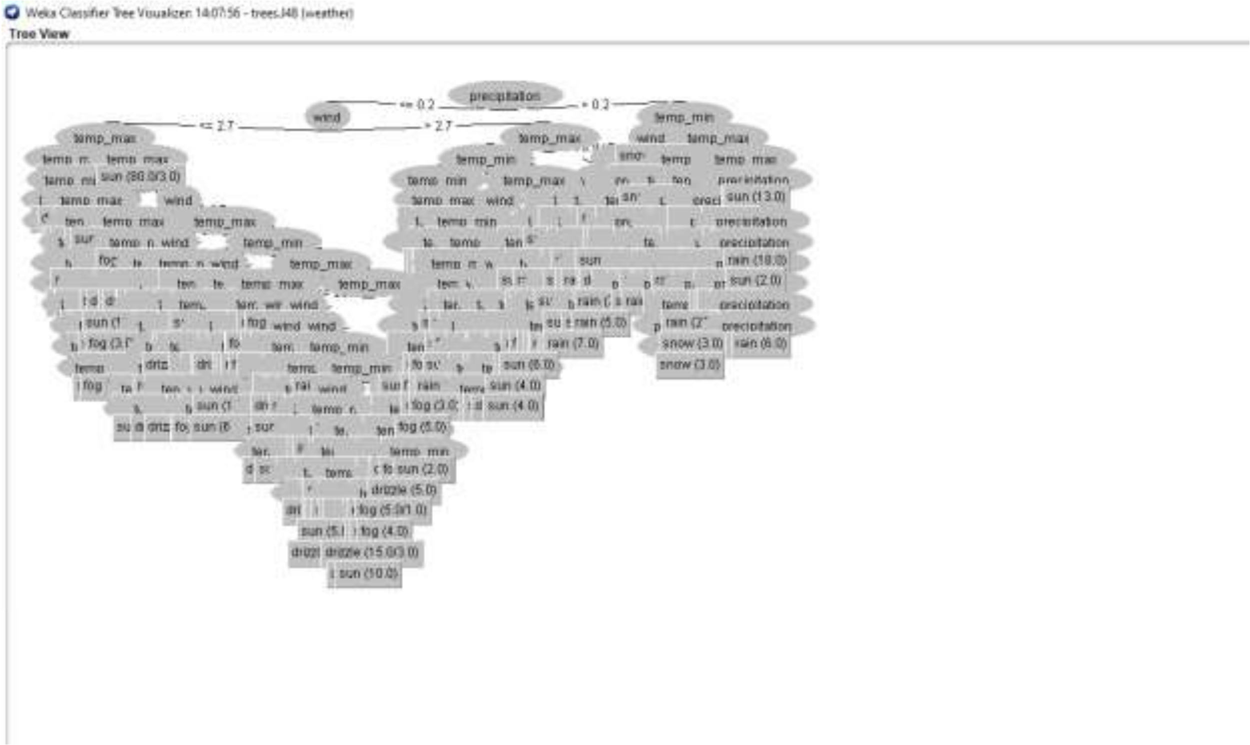


***FIGURE 04***

FLOW CHART TO ATTRIBUTE OF WEATHER. TEMPERATURE  VS WIND, WIND VS WEATHER.



***FIRGURE 05***

HERE IS THE DECISSION TREE OF MY  DATASET.NUMBER OF THE LEAVE  IS 296 AND SIZE OF THE TREE IS 591

## 2. Implementation Of Naïve Bayes Algorithm

### About Naïve Bayes Algorithm

The Naive Bayes algorithm is called "naive" because it makes the assumption that the occurrence of a certain feature is independent of the occurrence of other features. Bayes It refers to the statistician and philosopher Thomas Bayes and the theorem named after him, Bayes' theorem, which is the base for the Naive Bayes Algorithm

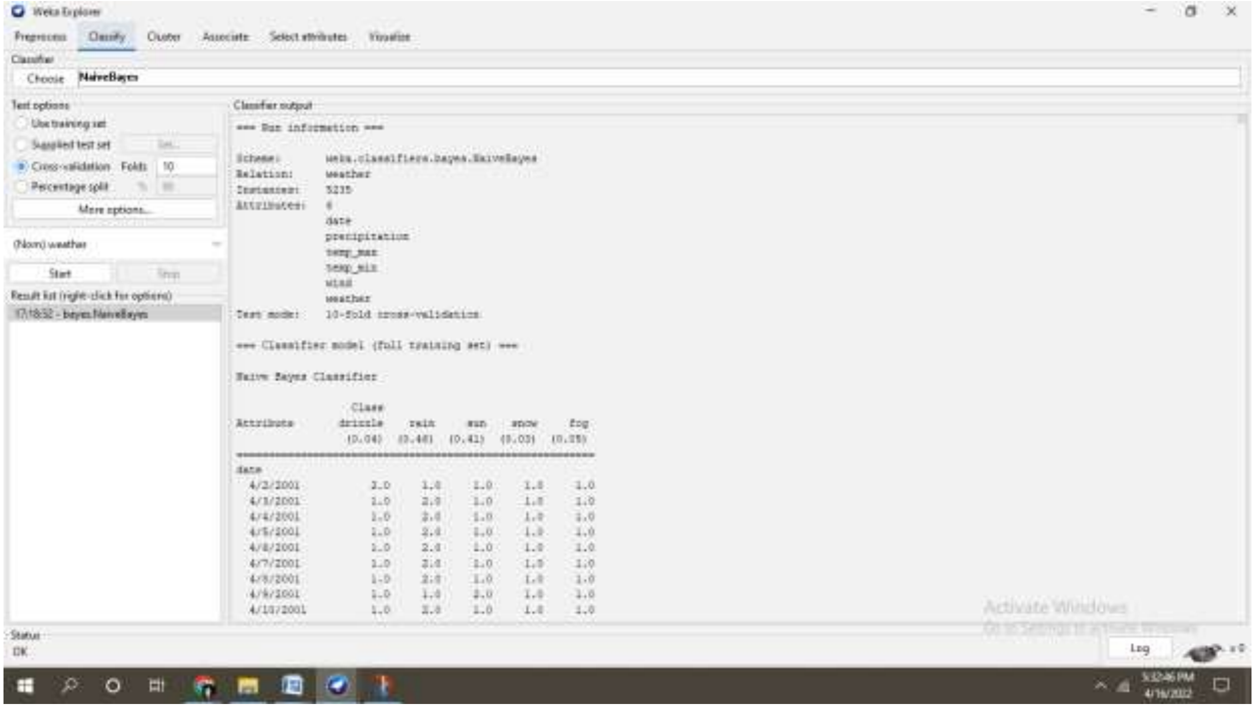
### Advantage of Naïve Bayes Algorithm

- It is simple and easy to implement
- It doesn't require as much training data
- It handles both continuous and discrete data
- It is highly scalable with the number of predictors and data points
- It is fast and can be used to make real-time predictions
- It is not sensitive to irrelevant features

### Disadvantage of Naïve Bayes Algorithm

- This limits the applicability of this algorithm in real-world use cases.
- This algorithm faces the 'zero-frequency problem' where it assigns zero probability to a categorical variable whose category in the test data set wasn't available in the training dataset
- Its estimations can be wrong in some cases, so you shouldn't take its probability outputs very seriously.
- Its estimations can be wrong in some cases, so you shouldn't take its probability outputs very seriously.

## MODEL DEVELOPMENT BY WEKA TOOLS



Classifier
Choose NaiveBayes

Test options
○ Use training set
○ Supplied test set
● Cross-validation Folds 10
○ Percentage split % 66

(Nom) weather

Result list (right-click for options)
17:18:32 - bayes.NaiveBayes

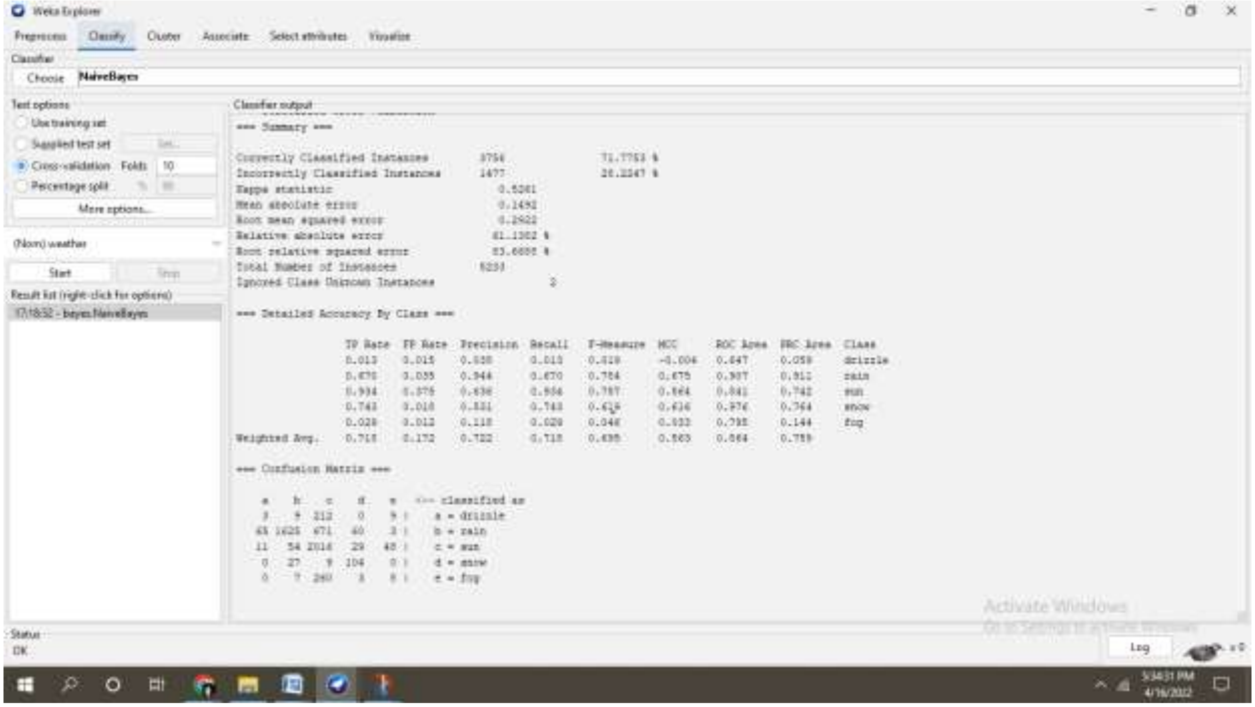
Status
OK

Classifier output

```
=== Run information ===

Scheme:       weka.classifiers.bayes.NaiveBayes
Relation:     weather
Instances:    5235
Attributes:   6
              date
              precipitation
              temp_max
              temp_min
              wind
              weather
Test mode:    10-fold cross-validation

=== Classifier model (full training set) ===

Naive Bayes Classifier

                   Class
Attribute        drizzle     rain      sun     snow      fog
                  (0.04)   (0.46)   (0.41)   (0.03)   (0.05)
==============================================================
date
  4/2/2001            2.0      1.0      1.0      1.0      1.0
  4/3/2001            1.0      2.0      1.0      1.0      1.0
  4/4/2001            1.0      2.0      1.0      1.0      1.0
  4/5/2001            1.0      2.0      1.0      1.0      1.0
  4/6/2001            1.0      2.0      1.0      1.0      1.0
  4/7/2001            1.0      2.0      1.0      1.0      1.0
  4/8/2001            1.0      2.0      1.0      1.0      1.0
  4/9/2001            1.0      1.0      2.0      1.0      1.0
  4/10/2001           1.0      2.0      1.0      1.0      1.0
```

***FIGURE : 06***

HERE IS MY DATASET DESCRIPTION



Classifier
Choose NaiveBayes

Test options
○ Use training set
○ Supplied test set
● Cross-validation Folds 10
○ Percentage split % 66

(Nom) weather

Result list (right-click for options)
17:18:32 - bayes.NaiveBayes

Status
OK

Classifier output

```
=== Summary ===

Correctly Classified Instances        3756               71.7753 %
Incorrectly Classified Instances      1477               28.2247 %
Kappa statistic                          0.5261
Mean absolute error                      0.1492
Root mean squared error                  0.2922
Relative absolute error                 61.1302 %
Root relative squared error             83.6888 %
Total Number of Instances             5233
Ignored Class Unknown Instances                  2

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.013    0.015    0.038      0.013    0.019      -0.004   0.647     0.058     drizzle
                 0.670    0.035    0.944      0.670    0.784      0.675    0.907     0.912     rain
                 0.934    0.375    0.636      0.934    0.757      0.564    0.841     0.742     sun
                 0.743    0.010    0.531      0.743    0.619      0.616    0.976     0.764     snow
                 0.028    0.012    0.118      0.028    0.046      0.033    0.735     0.144     fog
Weighted Avg.    0.718    0.172    0.722      0.718    0.695      0.563    0.864     0.759

=== Confusion Matrix ===

    a    b    c    d    e   <-- classified as
    3    9  212    0    9 |    a = drizzle
   65 1625  671   40    3 |    b = rain
   11   54 2016   29   40 |    c = sun
    0   27    9  104    0 |    d = snow
    0    7  260    3    8 |    e = fog
```

***FIGURE : 07***

AFTER APPLIFED 10 FOLD CROSS VALIDATION I GOT 71.77% ACCURACY AND 29.21% INCORRECT CLASSFIED INSTANCE

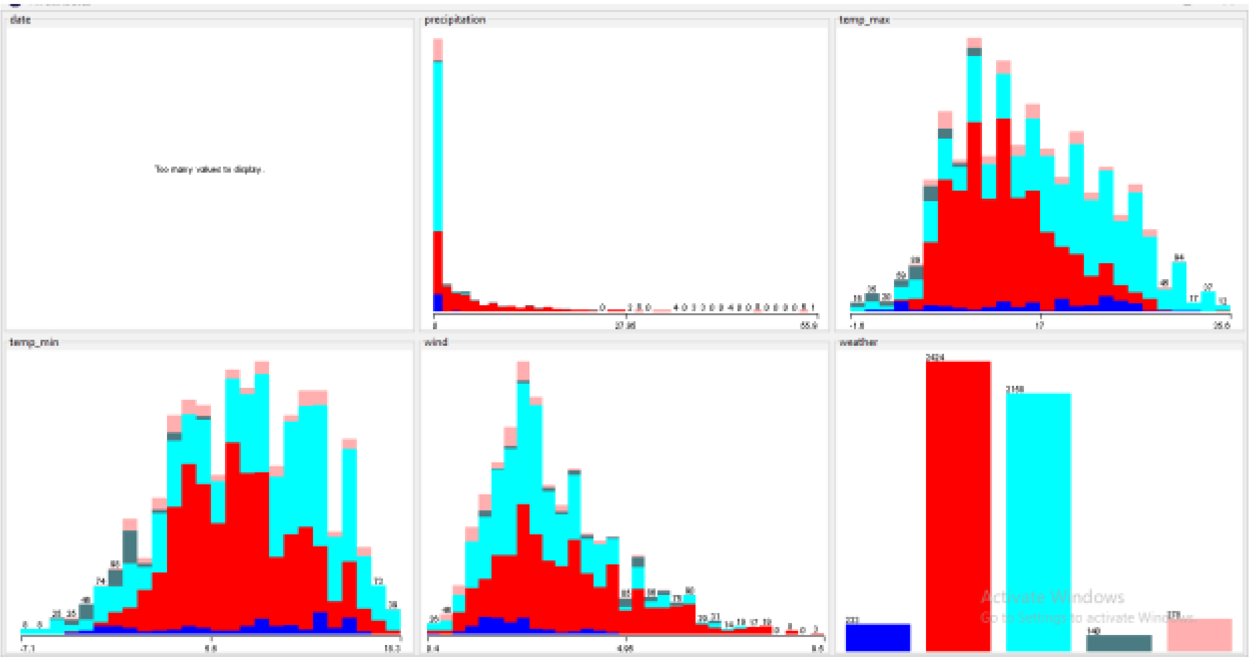


***FIGURE : 08***

FLOW CHART OF NAÏVE BAYES ALGORITHM APPLIED ON DATASET

## 3. Implementation of Nearest Neighbor Classification.

### About Nearest Neighbor Classification

K-NN Classification is used to develop this model. The k-nearest neighbor (k-NN) method is one of the data mining techniques considered to be among the top 10 techniques for data mining. The k-NN method tries to classify an unknown sample based on the known classification of its neighbors. In our dataset, the set of samples with known classification is available, which is the training set. Intuitively, each sample should be classified similarly to its surrounding samples. Therefore, if the classification of a sample is unknown, then it could be predicted by considering the classification of its nearest neighbor samples**.**

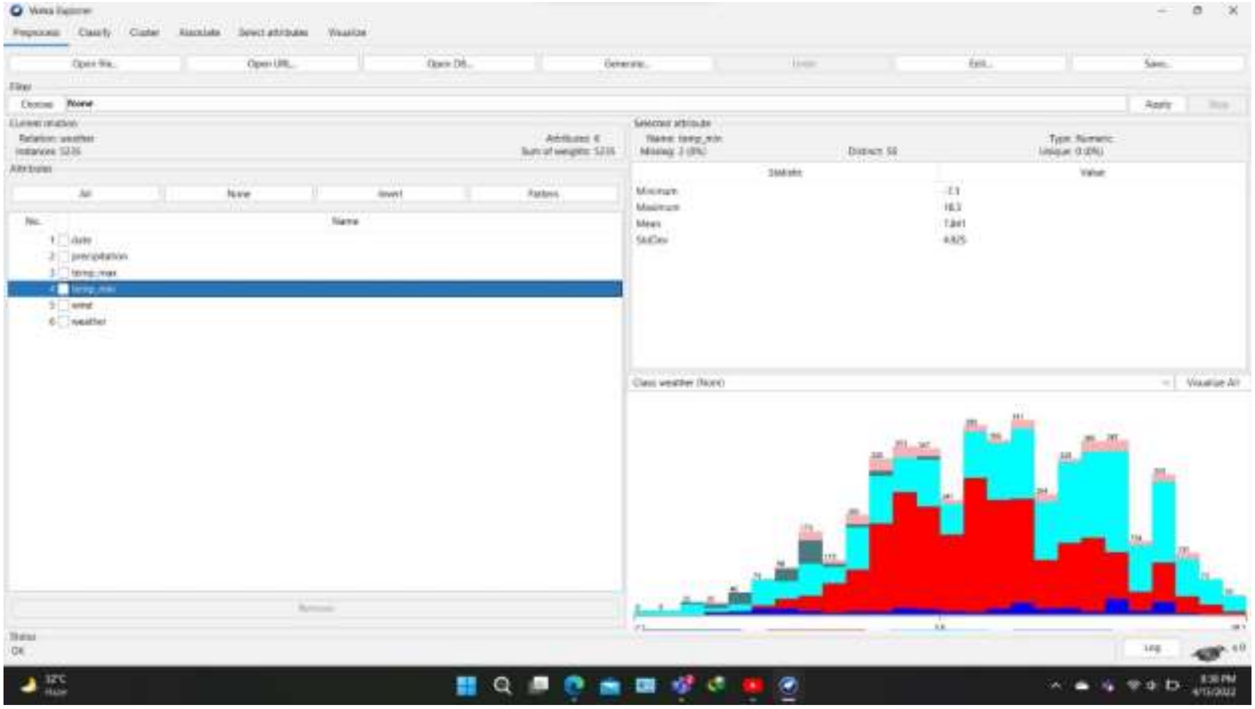
## ADVANTAGES

- **K-NN is pretty intuitive and simple**: K-NN algorithm is very simple to understand and equally easy to implement.
- **K-NN has no assumptions**: K-NN is a non-parametric algorithm which means there are assumptions to be met to implement K-NN.

- **No Training Step:** K-NN does not explicitly build any model, it simply tags the new data entry based learning from historical data. New data entry would be tagged with majority class in the nearest neighbor.

## DISADVANTAGES

- **K-NN slow algorithm**: K-NN might be very easy to implement but as dataset grows efficiency or speed of algorithm declines very fast.
- **Curse of Dimensionality:** KNN works well with small number of input variables but as the numbers of variables grow K-NN algorithm struggles to predict the output of new data point.
- **K-NN needs homogeneous features**: Deciding to build k-NN using a common distance, like Euclidean or Manhattan distances, it is completely necessary that features have the same scale, since absolute differences in features weight the same.
-
- **MODEL DEVELOPMENT BY WEKA TOOLS**



Filter: None

Current relation
Relation: weather
Instances: [illegible]
Attributes: 6
Sum of weights: [illegible]

Attributes

| No. | Name |
|---|---|
| 1 | date |
| 2 | precipitation |
| 3 | temp_max |
| 4 | temp_min |
| 5 | wind |
| 6 | weather |

Selected attribute
Name: temp_min
Type: Numeric
Missing: 2 (0%)
Distinct: 58
Unique: 0 (0%)

| Statistic | Value |
|---|---|
| Minimum | -7.1 |
| Maximum | 18.3 |
| Mean | 7.841 |
| StdDev | 4.925 |

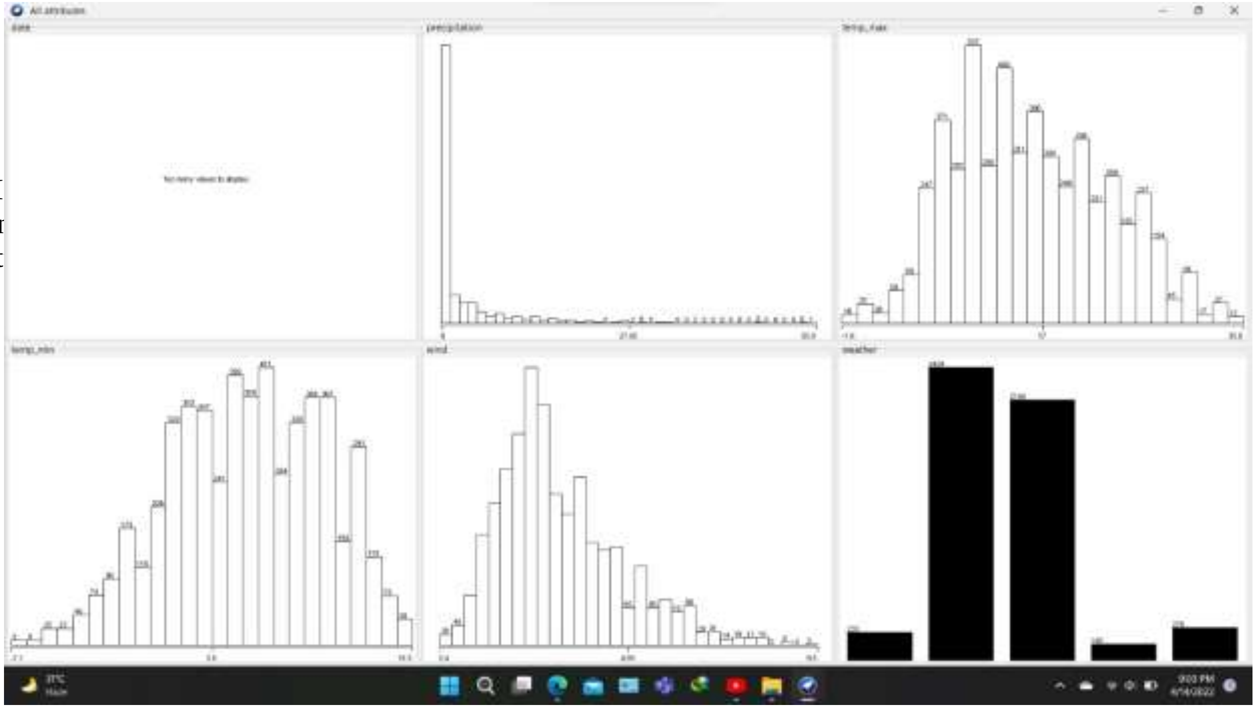
Class: weather (Nom)

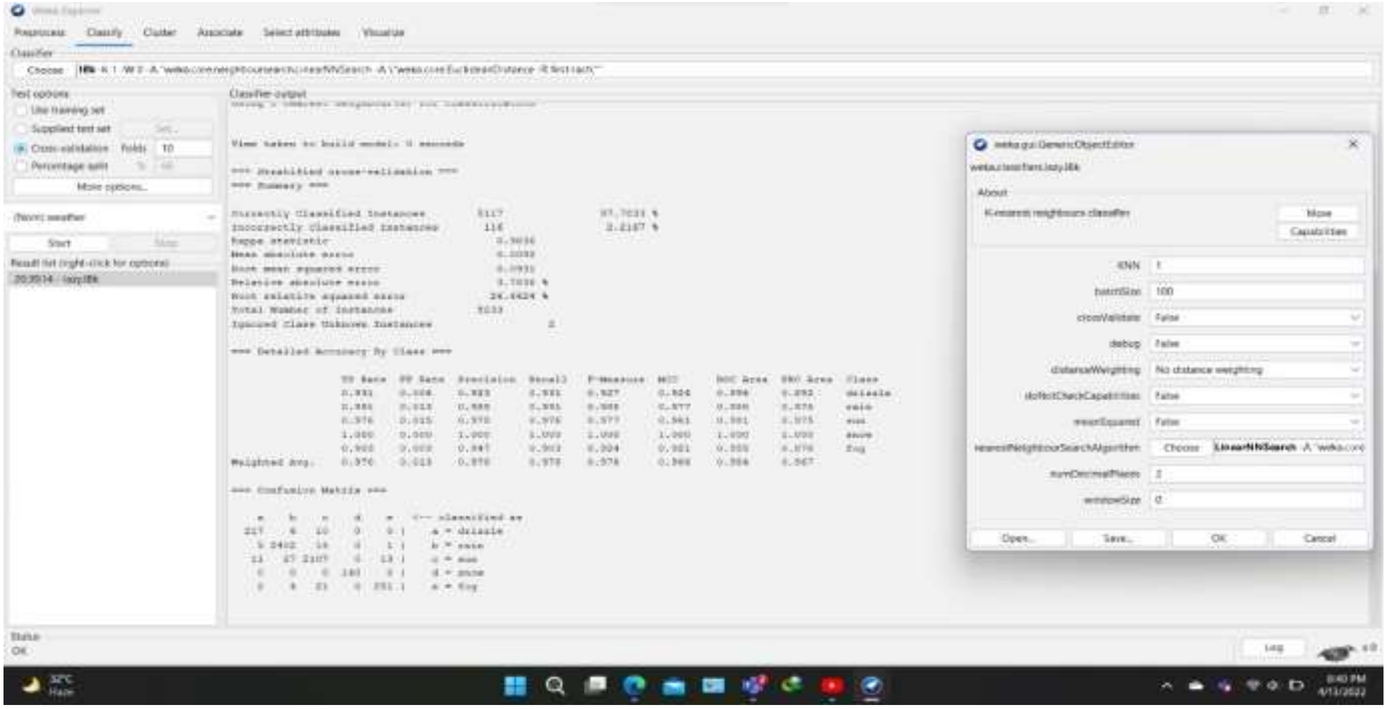
Status
OK

*FIGURE : 09*

In this picture we can see that the data set is loaded on the Weka tool to perform the model development process. In the current relation section of this software, it is visible that, the name of the relation is weather. Also there are 6 attributes, instances of 5235 and the sum of weights are the same as instances

H
or
at





IBk -K 1 -W 0 -A "weka.core.neighboursearch.LinearNNSearch -A \"weka.core.EuclideanDistance -R first-last\""

● Cross-validation Folds 10

(Nom) weather

20:30:14 - lazy.IBk

```
Time taken to build model: 0 seconds

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances        5117               97.7833 %
Incorrectly Classified Instances       116                2.2167 %
Kappa statistic                          0.9636
Mean absolute error                      0.0089
Root mean squared error                  0.0931
Relative absolute error                  3.7838 %
Root relative squared error             26.4424 %
Total Number of Instances             5233
Ignored Class Unknown Instances                  2

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.931    0.006    0.923      0.931    0.927      0.924    0.996     0.892     drizzle
                 0.988    0.013    0.988      0.988    0.988      0.977    0.988     0.978     rain
                 0.976    0.015    0.978      0.976    0.977      0.961    0.981     0.975     sun
                 1.000    0.000    1.000      1.000    1.000      1.000    1.000     1.000     snow
                 0.903    0.003    0.947      0.903    0.924      0.921    0.955     0.878     fog
Weighted Avg.    0.978    0.013    0.978      0.978    0.978      0.966    0.984     0.967

=== Confusion Matrix ===

    a    b    c    d    e   <-- classified as
  217    6   10    0    0 |    a = drizzle
    5 2402   16    0    1 |    b = rain
   11   27 2107    0   13 |    c = sun
    0    0    0  180    0 |    d = snow
    0    4   21    0  251 |    e = fog
```

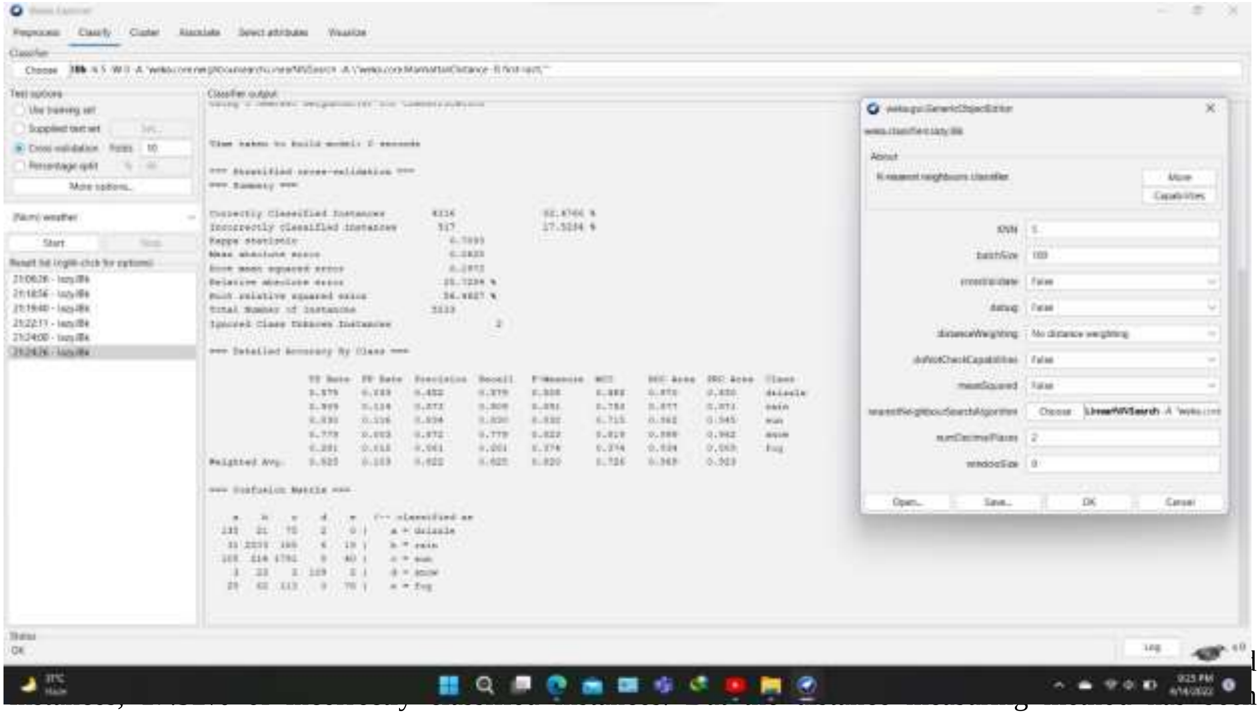
weka.classifiers.lazy.IBk

K-nearest neighbours classifier

| | |
|---|---|
| KNN | 1 |
| batchSize | 100 |
| crossValidate | False |
| debug | False |
| distanceWeighting | No distance weighting |
| doNotCheckCapabilities | False |
| meanSquared | False |
| nearestNeighbourSearchAlgorithm | Choose LinearNNSearch -A "weka.core |
| numDecimalPlaces | 2 |
| windowSize | 0 |

***FIGURE : 11***

Here on the classifier section, we have chosen lazy -> IBK on the classifier and it is using the Euclidian distance method to determine the accuracy of the k nearest neighbor, by using a  10 folds cross validation technique. The value of the K is 1, which finally it shows that the model has 97.78% of correctly classified instances and 2.216% of incorrectly classified instances.



Classifier: IBk -K 5 -W 0 -A "weka.core.neighboursearch.LinearNNSearch -A \"weka.core.ManhattanDistance -R first-last\""

Test options: ○ Use training set ○ Supplied test set ● Cross-validation Folds 10 ○ Percentage split % 66

(Nom) weather

```
21:06:28 - lazy.IBk
21:18:56 - lazy.IBk
21:19:40 - lazy.IBk
21:22:11 - lazy.IBk
21:24:00 - lazy.IBk
21:24:26 - lazy.IBk
```

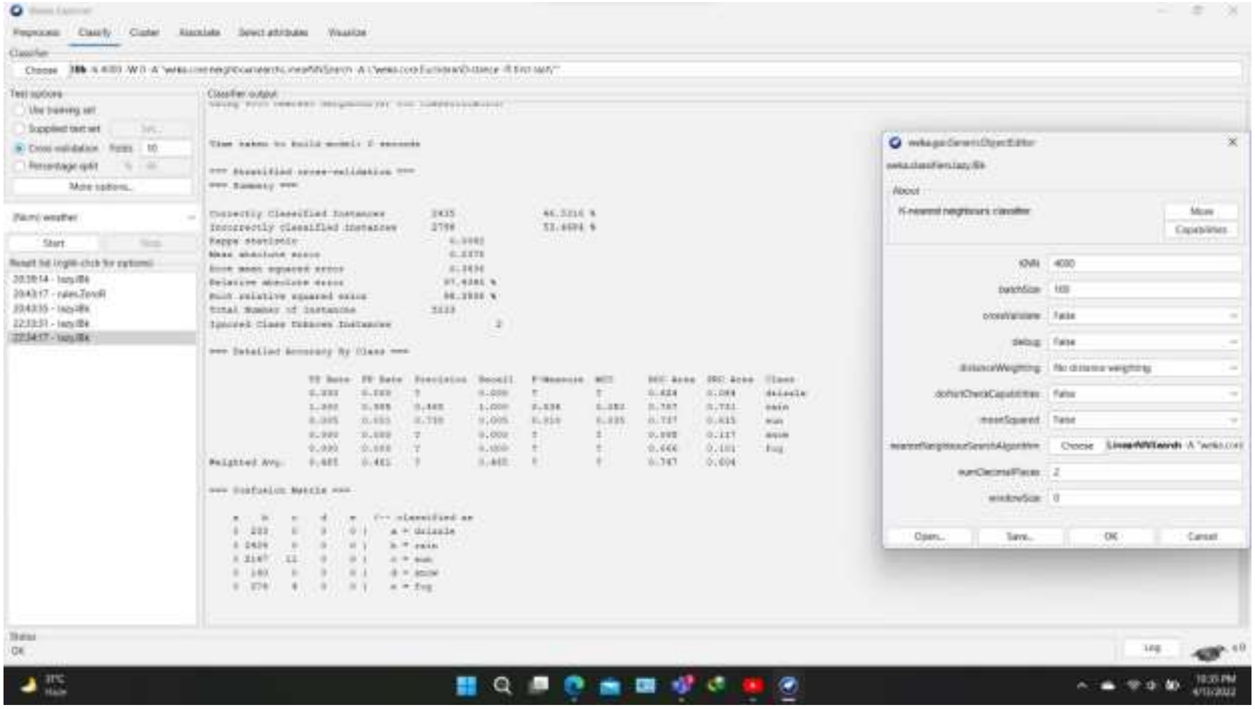
```
Time taken to build model: 0 seconds

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances        4316               82.4766 %
Incorrectly Classified Instances       917               17.5234 %
Kappa statistic                          0.7593
Mean absolute error                      0.0835
Root mean squared error                  0.2872
Relative absolute error                 25.7234 %
Root relative squared error             56.4827 %
Total Number of Instances             5233
Ignored Class Unknown Instances                  2

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.575    0.023    0.452      0.575    0.506      0.493    0.870     0.430     drizzle
                 0.909    0.124    0.872      0.909    0.891      0.784    0.877     0.871     rain
                 0.830    0.126    0.836      0.830    0.832      0.715    0.862     0.845     sun
                 0.779    0.003    0.872      0.779    0.822      0.819    0.888     0.962     snow
                 0.281    0.012    0.561      0.281    0.374      0.374    0.924     0.569     fog
Weighted Avg.    0.825    0.103    0.822      0.825    0.820      0.726    0.869     0.823

=== Confusion Matrix ===

    a    b    c    d    e   <-- classified as
  133   21   75    2    0 |    a = drizzle
   31 2233  165    6   19 |    b = rain
  105  214 1792    9   40 |    c = sun
    3   22    3  109    2 |    d = snow
   25   62  113    3   78 |    e = fog
```

weka.classifiers.lazy.IBk

K-nearest neighbours classifier.

| | |
|---|---|
| KNN | 1 |
| batchSize | 100 |
| crossValidate | False |
| debug | False |
| distanceWeighting | No distance weighting |
| doNotCheckCapabilities | False |
| meanSquared | False |
| nearestNeighbourSearchAlgorithm | Choose LinearNNSearch -A "weka.core |
| numDecimalPlaces | 2 |
| windowSize | 0 |

changed, now it's Manhattan distance here.



Classifier: IBk -K 4000 -W 0 -A "weka.core.neighboursearch.LinearNNSearch -A \"weka.core.EuclideanDistance -R first-last\""

Test options: ○ Use training set ○ Supplied test set ● Cross-validation Folds 10 ○ Percentage split % 66

(Nom) weather

```
20:38:14 - lazy.IBk
20:43:17 - rules.ZeroR
20:43:35 - lazy.IBk
22:33:31 - lazy.IBk
22:34:17 - lazy.IBk
```

```
Time taken to build model: 0 seconds

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances        2435               46.5316 %
Incorrectly Classified Instances      2798               53.4684 %
Kappa statistic                          0.0082
Mean absolute error                      0.2375
Root mean squared error                  0.3436
Relative absolute error                 97.4381 %
Root relative squared error             98.3988 %
Total Number of Instances             5233
Ignored Class Unknown Instances                  2

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.000    0.000    ?          0.000    ?          ?        0.424     0.084     drizzle
                 1.000    0.985    0.465      1.000    0.634      0.082    0.787     0.731     rain
                 0.005    0.001    0.733      0.005    0.010      0.035    0.727     0.615     sun
                 0.000    0.000    ?          0.000    ?          ?        0.598     0.117     snow
                 0.000    0.000    ?          0.000    ?          ?        0.666     0.101     fog
Weighted Avg.    0.465    0.461    ?          0.465    ?          ?        0.747     0.604

=== Confusion Matrix ===

    a    b    c    d    e   <-- classified as
    0  233    0    0    0 |    a = drizzle
    0 2424    0    0    0 |    b = rain
    0 2167   11    0    0 |    c = sun
    0  183    0    0    0 |    d = snow
    0  274    4    0    0 |    e = fog
```

weka.classifiers.lazy.IBk

K-nearest neighbours classifier.

| | |
|---|---|
| KNN | 4000 |
| batchSize | 100 |
| crossValidate | False |
| debug | False |
| distanceWeighting | No distance weighting |
| doNotCheckCapabilities | False |
| meanSquared | False |
| nearestNeighbourSearchAlgorithm | Choose LinearNNSearch -A "weka.core |
| numDecimalPlaces | 2 |
| windowSize | 0 |

***FIGURE : 13***

Here the value of K has been significantly increased to 4000. After that it’s visible, the accuracy level has been massively reduced to a 46.53% and the percentage of error has been increased. The incorrectly classified instances now 53.47%

## MODEL COMPARISON:

Comparison of used classification methods in this project; Decision Tree (DT), k-Nearest Neighbor (k-NN), Naïve Bayes (NB), has been compared. The effects of parameters including size of the dataset, kind of the independent attributes, and the number of the discrete and continuous attributes are the key parameters of this comparison. Based on the results, it can be concluded that in the datasets with few numbers of records, the comparison between classifiers may not do correctly. When the number of the records and the number of the attributes in each record are increased, the results become more stable. Decision Tree, k-NN obtain higher AUC than Naïve Bayes.

Nearest neighbor classifiers are based on learning by analogy, that is by comparing a given test tuple with training tuples which are similar to it. The training tuples are described by n attributes. Each tuple represents a point in an n-dimensional space. In this way, all of the training tuples are stored in an n-dimensional pattern space. When given an unknown tuple, a k-nearest neighbor (k-NN) classifier searches the pattern space for the k training tuples which are closest to the unknown tuple. These k training tuples are the k-nearest neighbors of the unknown tuple.

We have developed KNN model and got pretty good accuracy. But the K value plays a significant role to differ the error and incorrect instances percentage. Yet KNN classification is more optimizable and easier to use in weka tools. Considering as many as instances to develop the model leads to some errors and less correct results. On that account Decision tree is more accurate on the results.

A decision tree (DT) is a flowchart-like tree structure, where each internal node denotes a test on an attribute, each branch represents an outcome of the test, and each leaf node (or terminal node) holds a class label . The topmost node in a tree is the root node. During tree construction, attribute selection measures are used to select the attribute which best partitions the tuples into distinct classes. Three popular attribute selection measures are Information Gain, Gain Ratio, and Gini Index. When DTs are built, many of the branches may reflect noise or outliers in the training data. Tree pruning attempts to identify and remove such branches, with the goal of improving classification accuracy on unseen data. By using decision tree we have achieved the best possible outcome comparing with the other two models.

Naïve Bayes has given the lowest accuracy in comparison with other models. As the basic idea in NB approaches is to use the joint probabilities of words and categories to estimate the probabilities of categories given a document. The naïve part of NB methods is the assumption of word independence, i.e., the conditional probability of a word given a category is assumed to be independent from the conditional probabilities of other words given that category. This assumption makes the computation of the NB classifiers far more efficient but in case of our dataset this model was not efficient enough.

## DISCUSSION AND ANALYISIS

From this project, we have gained knowledge about the efficacy of DT, k-NN, NB has been investigated and these three methods is compared to each other with respect to different circumstances of attributes deployment. In this model development and comparison process, the size of datasets, types of the attributes and the number of discrete and continuous attributes have been considered. From our analysis we have shown that for small datasets, in all cases deviation is such high that the comparison between classifiers may not do correctly. For larger datasets results become more stable and the comparisons can be done. That is the reason for choosing our large dataset to perform the model development and comparison with maximum accuracy. Decision Tree as an implementation of that, show an efficient performance in all datasets. Two classifiers including k-NN and Naïve Bayes are also high efficacy in the current work. Following the used process of evaluation of our project, we can choose the best classifier according to data type and continuous or discrete attributes existing in datasets. Therefore, using the above conclusion our most efficient classifier with respect to dataset characteristics is Decision Tree. Lastly, it should be mentioned that our current project is based on simulation data and the generated datasets that are not dependent to a special problem. Consequently, the above results can be extended to a wide range of problems and real datasets made from data collected from actual real-world problems are suitable for model developing and comparing the mentioned methods. In future studies, we will be using the experience gained from this project, taking larger real datasets and using other evaluation criteria and applying new classifiers on datasets with more variables could be as open problems in this field.