UNIVERSITY OF LIVERPOOL

Data Mining and Machine Learning

(EBUS537)

Graded Assignment

Professor: Dongping Song & Eric Leung

Report By:

Ajitesh Jyotirmoy Kumar

Student ID- 201595383

Question 1

(1) The following dataset has been given in which the attributes have multiple values. So, we use the multi-way split and apply the Greedy strategy and Gini impurity measure to build a fully grown decision tree.

The dataset contains four attributes: Buying, Maintenance, Doors, and Safety. The class label is acceptance which has two values Acceptable(acc) and Unacceptable(unacc).

Firstly, to build a fully grown decision tree, we need to decide which attributes such as Buying, Maintenance, Doors, or Safety will be the parent node at level 1 of the decision tree.

Using the formulas for GINI Index and GINI split for each attribute, we decide the parent node at level 1.

GINI Index= 1-sum(p_j^2)

GINI(Split)= sum(n_j/n*GINI(j))

a) **Considering attribute- Buying**

GINI(Buying_high) = 1-$(46/204)^2$-$(158/204)^2$ = 1-0.0508-0.5998
= **0.3494**

GINI(Buying_med) = 1-$(47/107)^2$-$(60/107)^2$ = 1-0.1929- 0.3144
= **0.4927**

GINI(Buying_low) = 1-$(38/89)^2$-$(51/89)^2$ = 1-0.1823- 0.3283
= **0.4894**

GINI(Buying_split) = (204/400) *0.3494 + (107/400) *0.4927 + (89/400) *0.4894
= 0.1781 + 0.1317 + 0.1088= **0.4186**

b) **Considering attribute- Maintenance**

GINI(Maint_high) = 1- $(49/202)^2$-$153/202)^2$ = 1-0.0588- 0.5736
= **0.3676**

GINI(Maint_low) = 1-$(116/198)^2$-$(82/198)^2$ = 1-0.3432-0.1715
= **0.4853**

GINI(Maint_split) = (202/400) *0.3676 + (198/400) *0.4853
= 0.1856 + 0.2402 = **0.4258**

c) **Considering attribute- Doors**

GINI(Doors_3) = 1-$(61/202)^2$-$(141/202)^2$ = 1-0.0911-0.4872
= **0.4217**

GINI(Doors_5) = 1-$(70/198)^2$-$(128/198)^2$= 1- 0.1249-0.4179

= **0.4572**

GINI(Doors_split) = (202/400) *0.4217 + (198/400) *0.4572

= 0.2129 + 0.2263 = **0.4392**

d) Considering attribute- Safety

GINI(Safety_high) = 1-(75/134)^2-(59/134)^2= 1- 0.3132-0.1938

= **0.493**

GINI(Safety_med) = 1-(56/133)^2-(77/133)^2= 1- 0.1172- 0.3351

= **0.4877**

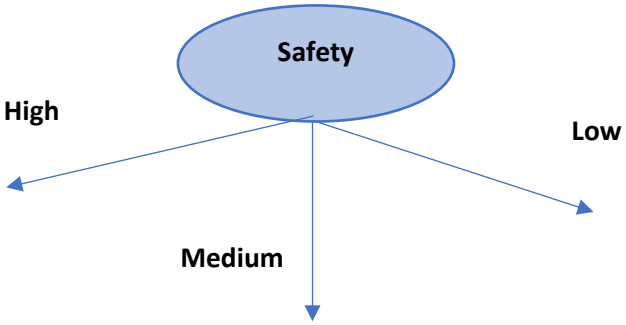
GINI(Safety_low) = 1-(133/133)^2- (0) = 1-1

= **0**

GINI(Safety_split) = (134/400) *0.493 + (133/400) *0.4877 + 0

= 0.1651+ 0.1621 = **0.3272**

According to Greedy Strategy, we choose the attribute "**Safety**" to split the tree at level 1.



Since the class labels under the High branch are not homogenous, we need to further split the node by selecting the attribute Buying, Maintenance, or Doors.

For Safety= High branch:

a) Considering attribute- Buying

GINI(Buying_high) = 1-(30/66)^2-(36/66)^2= 1-0.2066-0.2975

= **0.4959**

GINI(Buying_med) = 1-(27/39)^2-(12/39)^2= 1-0.4792-0.0946

= **0.4262**

GINI(Buying_low) = 1-(18/29)^2-(11/29)^2= 1-0.3852-0.1438

= **0.471**

GINI(Buying_split) = (66/134) *0.4959 +(39/134) *0.4262 +(29/134) *0.471

= 0.2442 +0.124 +0.1019= **0.4701**

b) Considering attribute- Maintenance

GINI(Maint_high) = 1- (29/63)^2-(34/63)^2= 1- 0.2118-0.2912

= **0.497**

GINI(Maint_low) = 1-(46/71)^2-(25/71)^2= 1-0.4197-0.1239

= **0.4564**

GINI(Maint_split) = (63/134) *0.497 + (71/134) *0.4564

= 0.2336+0.2418= **0.4754**

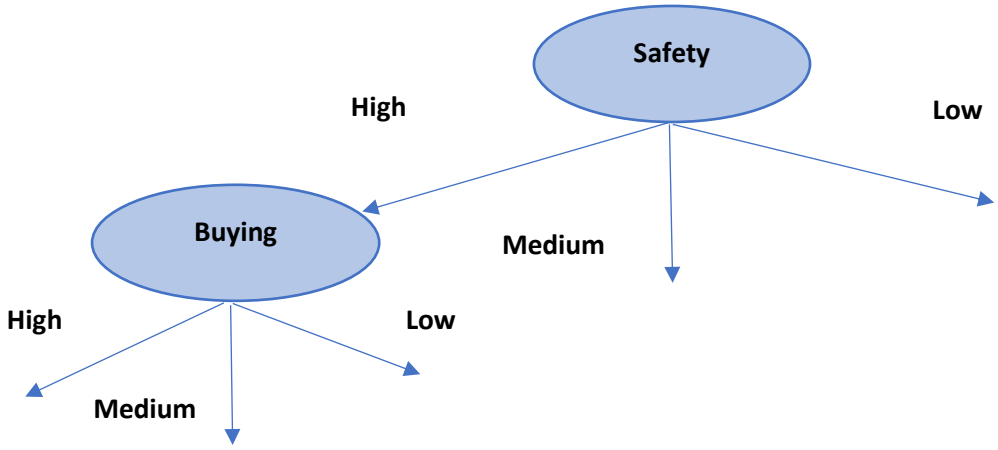
c) **Considering attribute -Doors**

GINI(Doors_3) = $1-(40/69)^2-(29/69)^2$= 1-0.336-0.1766
= **0.4874**

GINI(Doors_5) = $1-(35/65)^2-(30/65)^2$= 1-0.2899-0.213
= **0.4971**

GINI(Doors_split) = (69/134) *0.4874 + (65/134) *0.4971
= 0.2509 + 0.2411= **0.492**

As per Greedy Strategy, attribute **"Buying**" is chosen to split the tree at level 2 from the High branch.



Since no data objects have the same class label, we need to further split.

For Buying= High branch;

a) **Considering attribute- Maintenance**

GINI(Maint_High) = $1-(8/30)^2-(22/30)^2$= 1- 0.0711-0.5377
= **0.3912**

GINI(Maint_Low) = $1-(22/36)^2-(14/36)^2$
= **0.4753**

GINI(Maint_split) = (30/66) *0.3912 + (36/66) *0.4753
= **0.4370**

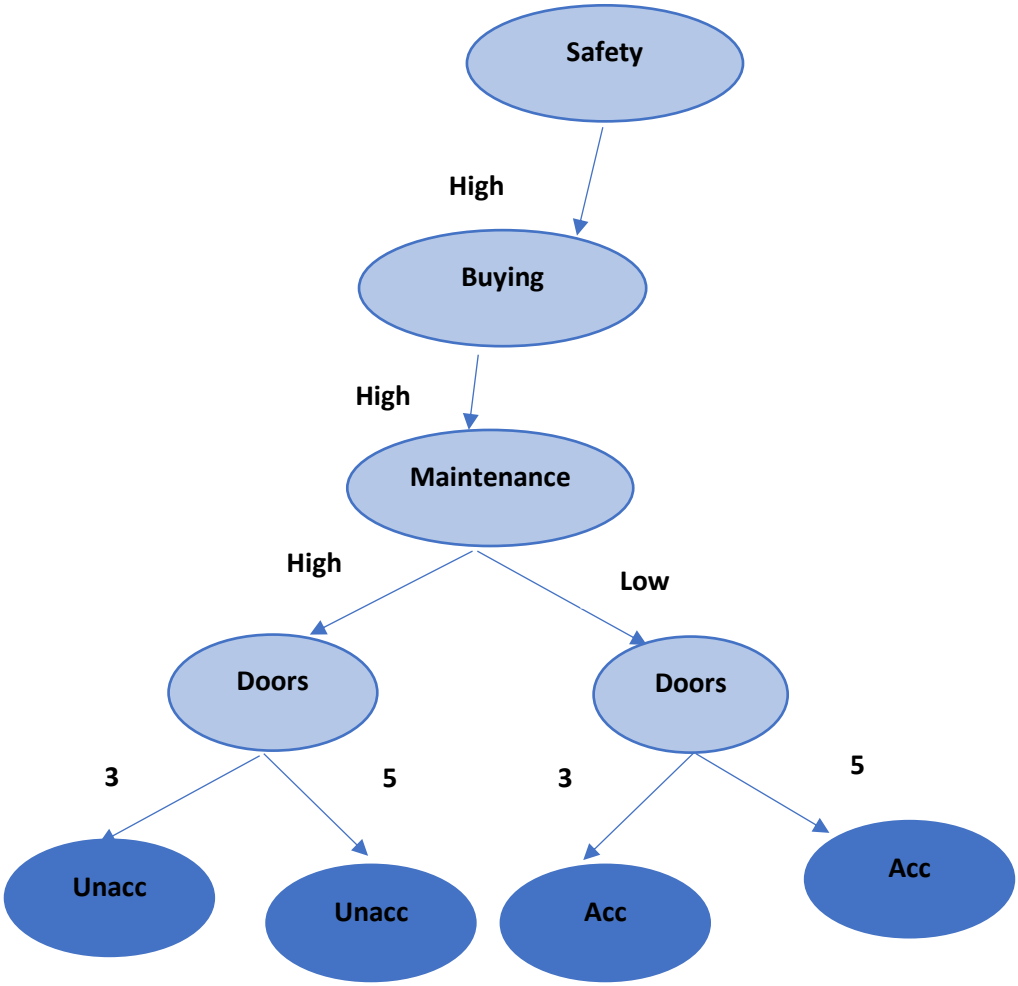
b) **Considering attribute- Doors**

GINI(Doors_3) = $1-(16/34)^2-(18/34)^2$
= **0.4982**

GINI(Doors_5) = $1-(14/32)^2-(18/32)^2$
= **0.4921**

GINI(Doors_split) = (34/66) *0.4982 +(32/66) *0.4921
= **0.4952**

As per Greedy Strategy, attribute **"Maintenance**" is chosen to split the tree at level 3.



For Buying=Medium branch:

a) **Considering attribute- Maintenance**

GINI(Maint_High) = $1-(5/17)^2-(12/17)^2$

= **0.4152**

GINI(Maint_Low) = $1-(15/22)^2-(7/22)^2$

= **0.4338**

GINI(Maint_split) = (17/39) *0.4152 +(22/39) *0.4338

= **0.4256**

b) **Considering attribute- Doors**

GINI(Doors_3) = $1-(13/18)^2-(5/18)^2$

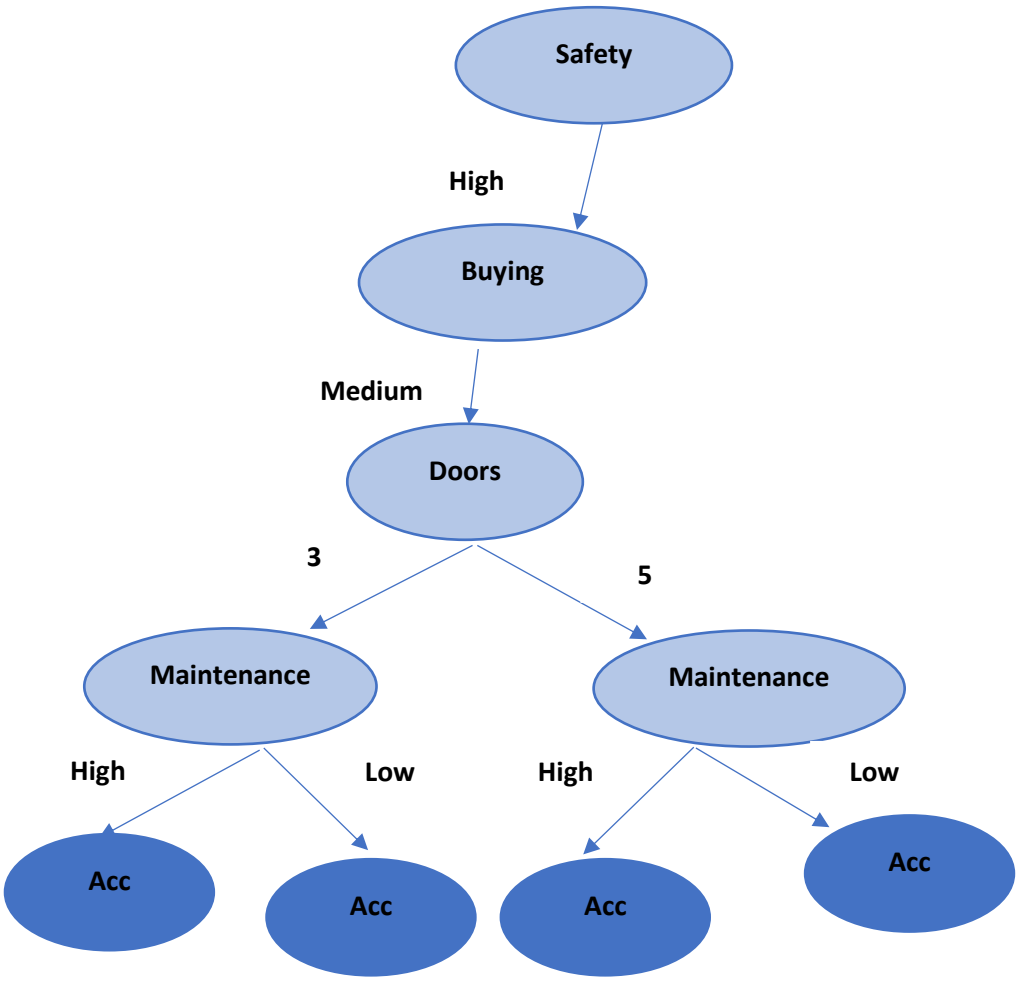
= **0.4012**

GINI(Doors_5) = $1-(14/21)^2-(7/21)^2$

= **0.4444**

GINI(Doors_split) = (18/39) *0.4012 + (21/39) *0.4444= **0.4244**

Hence, we choose the attribute "**Doors**" to split at level 3.



For Buying=Low branch;

a) **Considering attribute- Maintenance**

GINI(Maint_High) = $1-(9/16)^2-(7/16)^2$

=**0.4921**

GINI(Maint_Low) = $1-(9/13)^2-(4/13)^2$

=**0.4260**

GINI(Maint_split) = (16/29) *0.4921 + (13/29) *0.4260

= **0.4624**

b) **Considering attribute- Doors**

GINI(Doors_3) = $1-(11/17)^2-(6/17)^2$

= **0.4567**

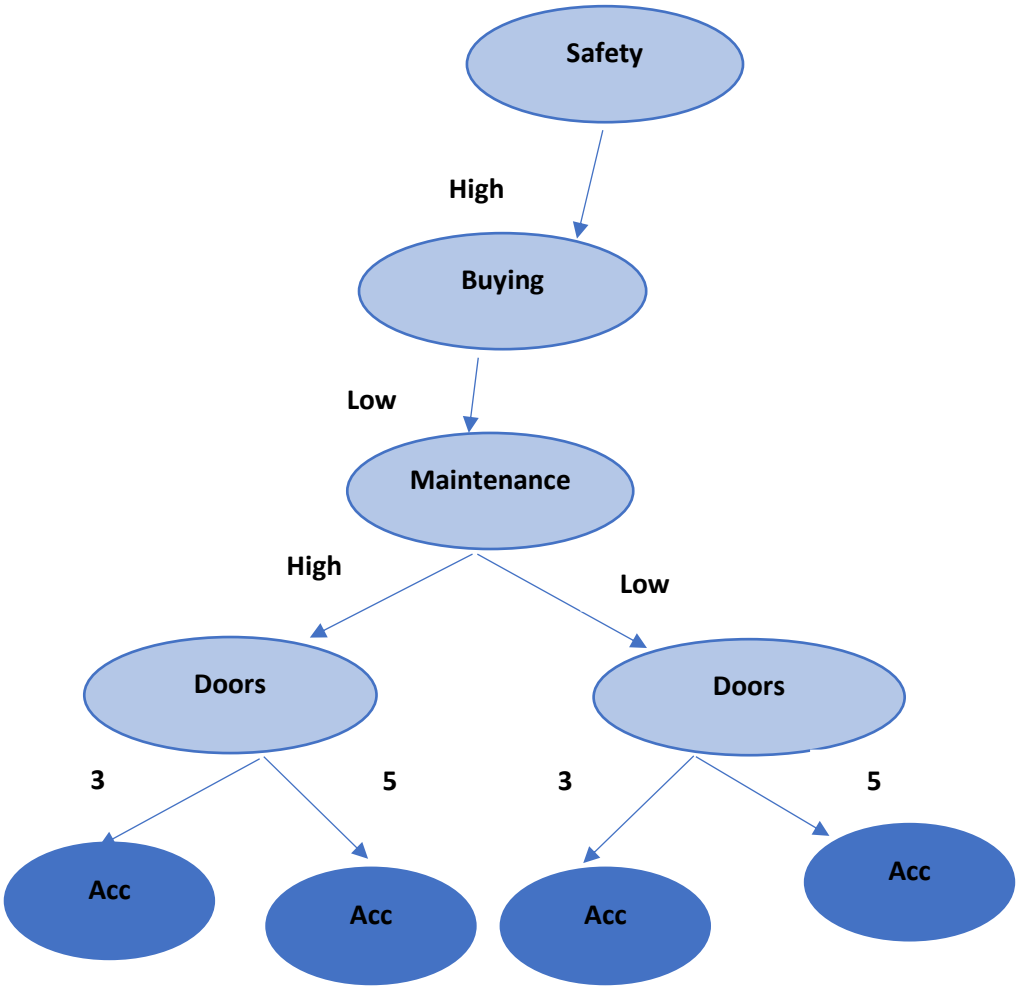
GINI(Doors_5) = $1-(7/12)^2-(5/12)^2$

= **0.4861**

GINI(Doors_split) = (17/29) *0.4567 + (12/29) *0.4861

= **0.4688**

Hence, we choose the attribute "**Maintenance**" to split at level 3.



For Safety= Medium branch;

a) **Considering attribute-Buying**

GINI(Buying_High) =1-$(52/68)^2$-$(16/68)^2$

=**0.36**

GINI(Buying_Med) =1-$(20/34)^2$-$(14/34)^2$

= **0.4845**

GINI(Buying_Low) = 1-$(20/31)^2$-$(11/31)^2$

= **0.4579**

GINI(Buying_split) = (68/133) *0.36 + (34/133) *0.4845 + (31/133) *0.4579

=**0.4145**

b) **Considering attribute- Maintenance**

GINI(Maint_High) = 1-$(20/63)^2$-$(43/63)^2$

= **0.4335**

GINI(Maint_Low) = 1-$(36/70)^2$-$(34/70)^2$= **0.4997**

GINI(Maint_split) = (63/133) *0.4335 +(70/133) *0.4997

= **0.4683**

c) **Considering attribute- Doors**

GINI(Doors_3) = $1-(44/65)^2-(21/65)^2$

= **0.4375**

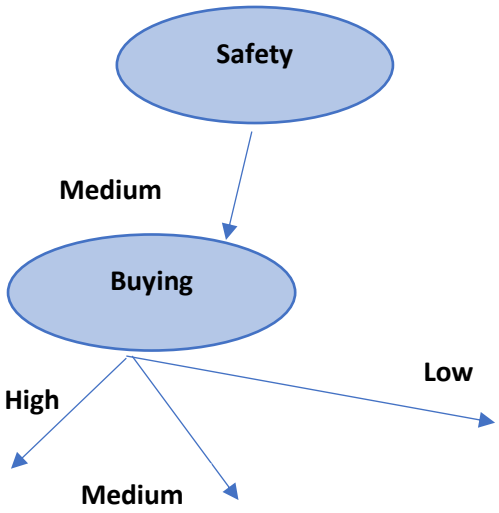
GINI(Doors_5) = $1-(35/68)^2-(33/68)^2$

= **0.4996**

GINI(Doors_split) = (65/133) *0.4375 + (68/133) *0.4996

= **0.4692**

Hence, as per Greedy Strategy, attribute "**Buying**" is chosen to split at level 2.



For Buying= High branch;

a) **Considering attribute-Maintenance**

GINI(Maint_High) = $1-(29/32)^2-(3/32)^2$

= **0.1695**

GINI(Maint_Low) = $1-(23/36)^2-(13/36)^2$

= **0.4615**

GINI(Maint_split) = (32/68) *0.1695 + (36/68) *0.4615

= **0.324**

b) **Considering attribute- Doors**

GINI(Doors_3) = $1-(7/35)^2-(28/35)^2$

= **0.32**

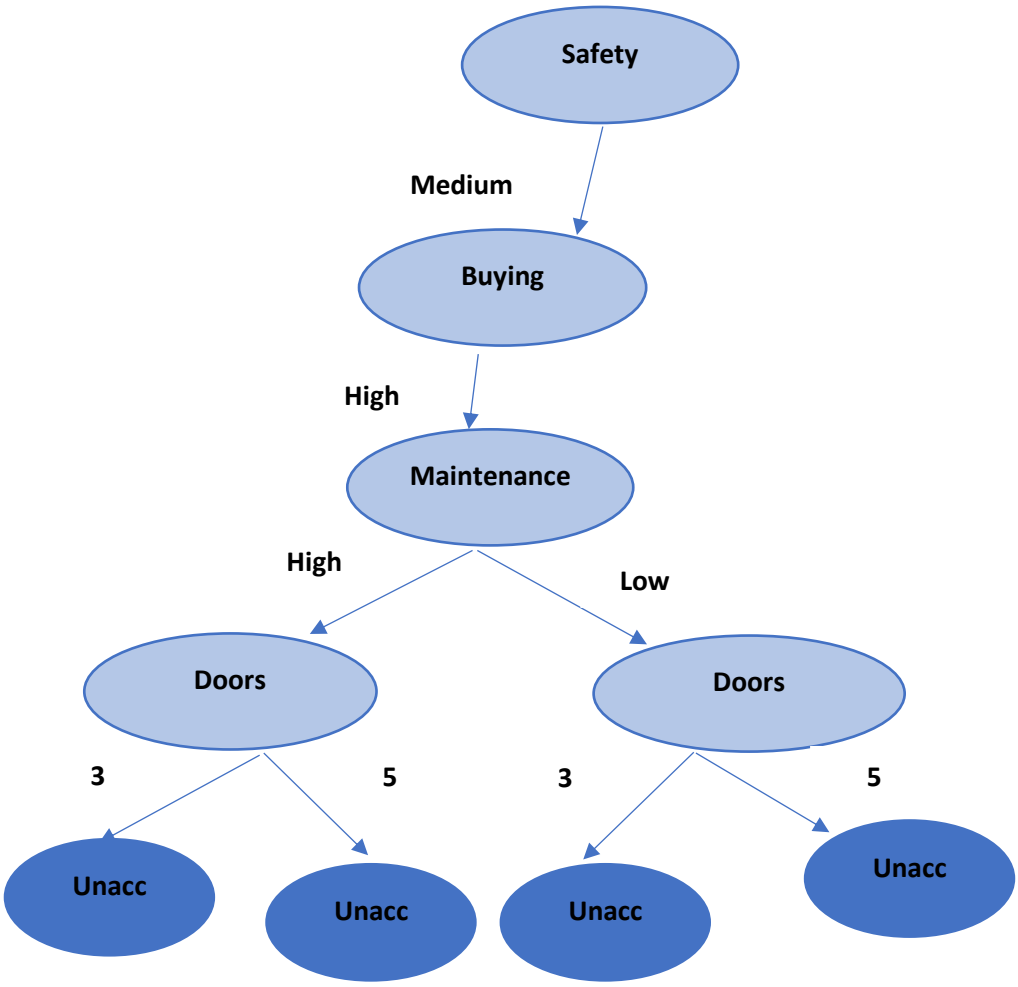
GINI(Doors_5) = $1-(24/33)^2-(9/33)^2$

= **0.3968**

GINI(Doors_split) = (35/68) *0.32 + (33/68) *0.3968

= **0.3572**

Hence, attribute "**Maintenance"** is chosen to split at level 3.



For Buying=Medium branch;

a) **Considering attribute- Maintenance**

GINI(Maint_High) = $1-(9/18)^2-(9/18)^2$

= **0.5**

GINI(Maint_Low) = $1-(11/16)^2-(5/16)^2$

= **0.4298**

GINI(Maint_split) = (18/34) *0.5 + (16/34) *0.4298

= **0.4669**

b) **Considering attribute- Doors**

GINI(Doors_3) = $1-(8/13)^2-(5/13)^2$

= **0.4735**

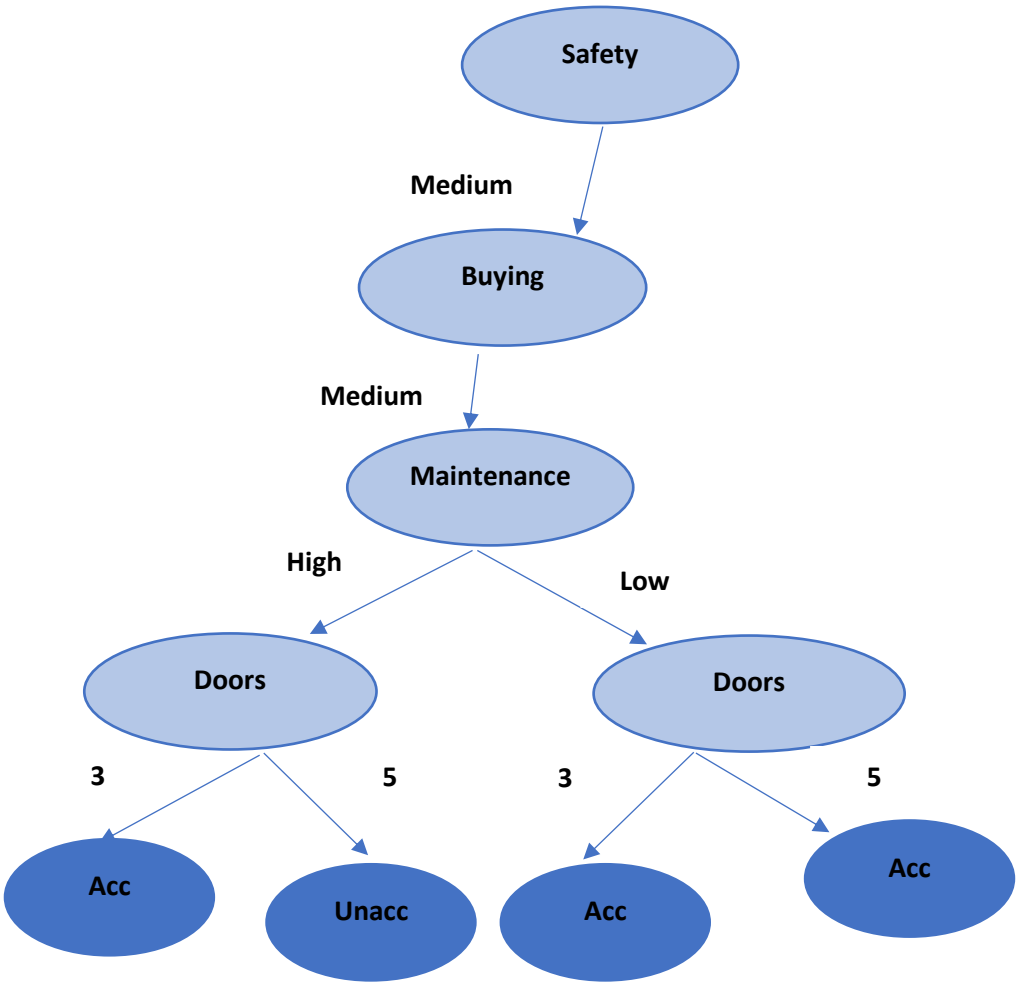
GINI(Doors_5) = $1-(12/21)^2-(9/21)^2$

= **0.4899**

GINI(Doors_split) = (13/34) *0.4735 + (21/34) *0.4899

= **0.4835**

Hence, attribute "**Maintenance**" is chosen to split at level 3.



For Buying=Low branch;

a) **Considering attribute- Maintenance**

GINI(Maint_High) = $1-(8/13)^2-(5/13)^2$
= **0.4735**

GINI(Maint_Low) = $1-(12/18)^2-(6/18)^2$
= **0.4445**

GINI(Maint_split) = (13/31) *0.4735 + (18/31) *0.4445
= **0.4565**

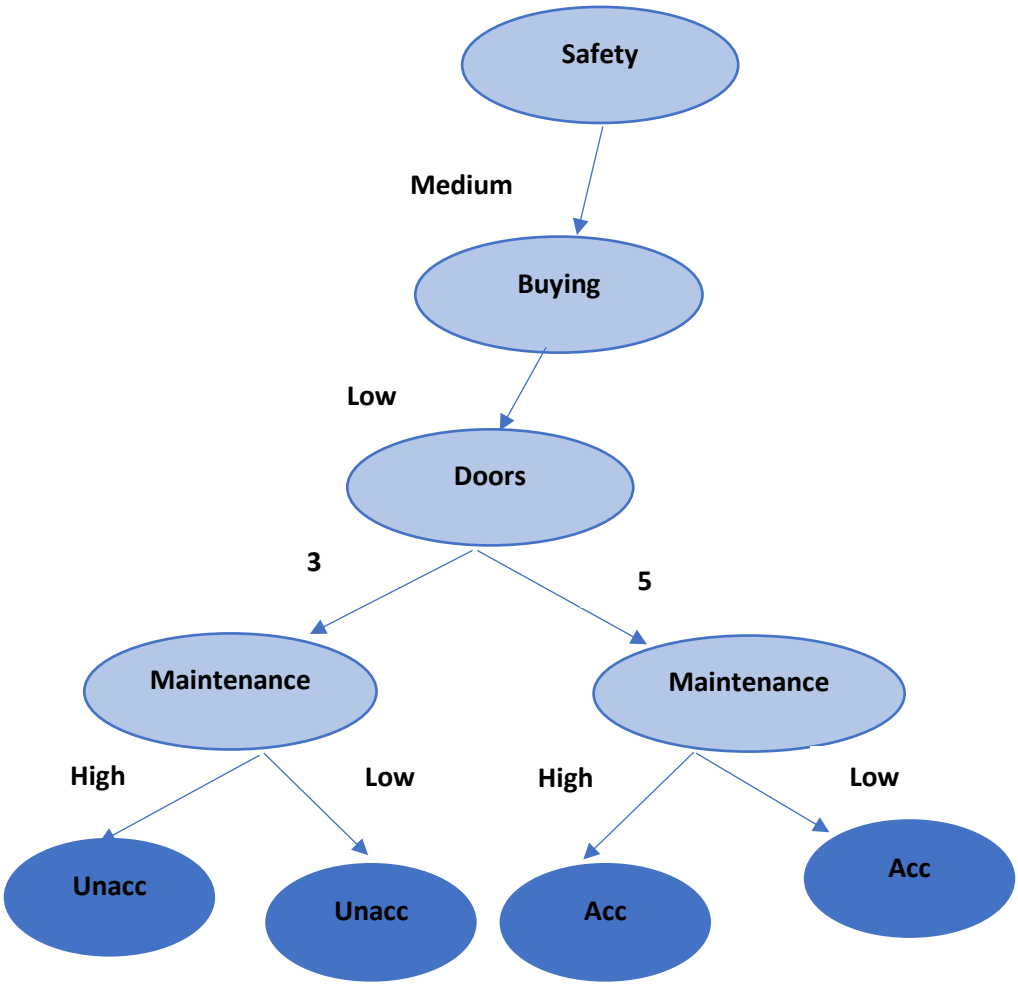
b) **Considering attribute- Doors**

GINI(Doors_3) = $1-(6/17)^2-(11/17)^2$
= **0.4569**

GINI(Doors_5) = $1-(14/14)^2$
= **0**

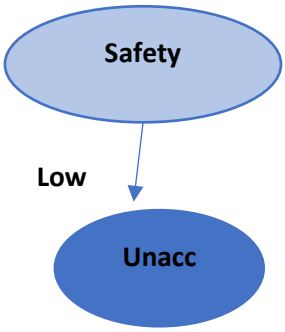
GINI(Doors_split) = (17/31) *0.4569 + 0
= **0.2505**

Hence, the attribute "**Doors**" is chosen to split at level 3.

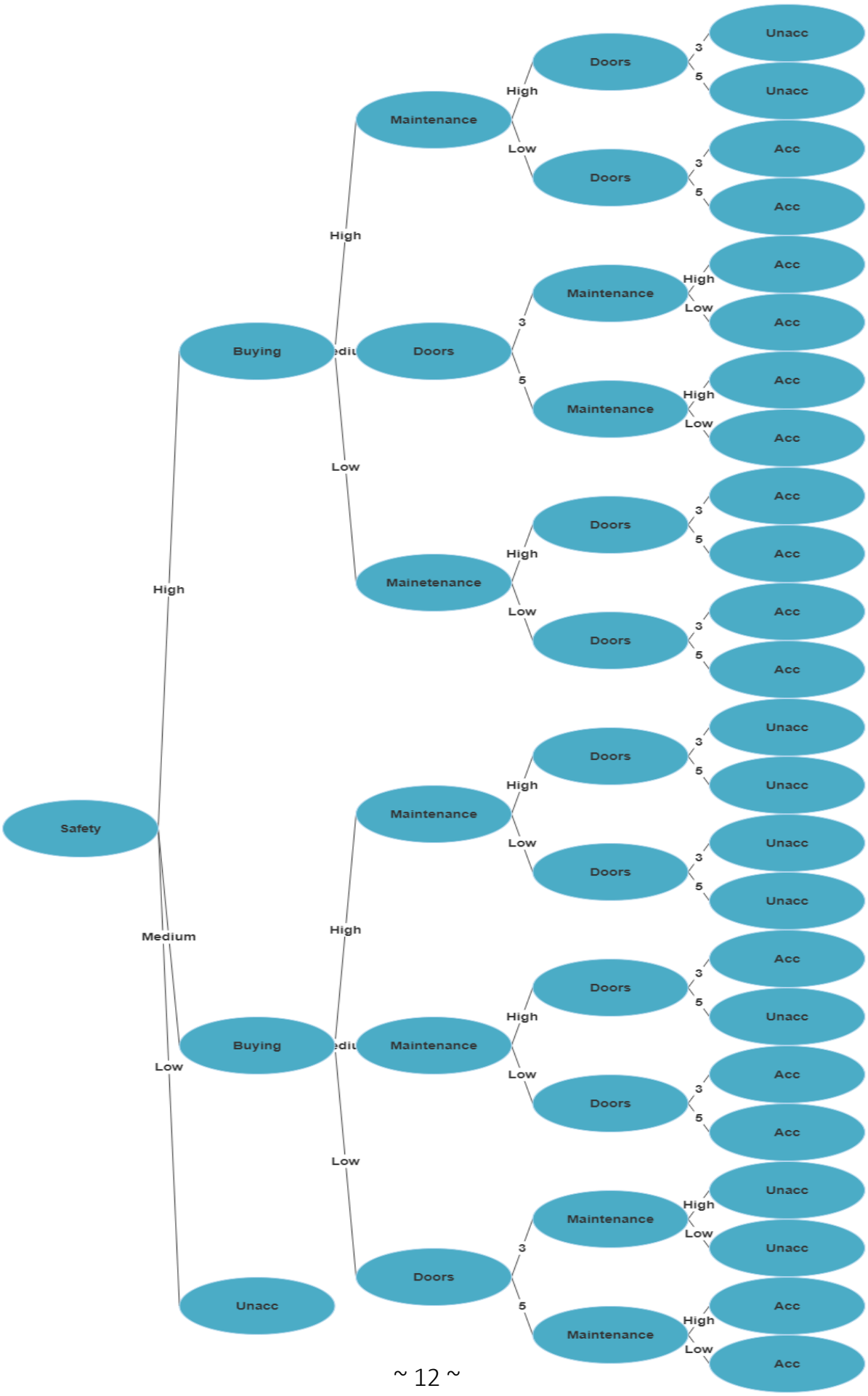


For Safety= Low branch;

Since all the attributes have the same class label, "**Unacc**," that is, all are homogenous; hence no further split is required.



Fully grown Decision Tree (Training Dataset):


Safety
High
Medium
Low
Buying
Buying
Unacc
High
Low
High
Low
Maintenance
Doors
Mainetenance
Maintenance
Doors
Doors
Maintenance
Doors
3
5
High
Low
Unacc
Unacc
Acc
Acc
Acc
Acc
Acc
Acc
Acc
Acc
Unacc
Unacc
Unacc
Unacc
Acc
Unacc
Acc
Acc
Unacc
Unacc
Acc
Acc

(2) Pruning:

When Safety= High branch;

Pruning can be done on the nodes Safety > High>Buying >High> Maintenance> High as the majority of the class labels (leaf nodes) under the attribute Doors is "**Unacc**."

Pruning can be done on the nodes Safety > High>Buying >High> Maintenance> Low as the majority of the class labels (leaf nodes) under the attribute Doors is "**Acc**."

Pruning can be done on the nodes Safety > High>Buying >Medium> Doors as the majority of the class labels (leaf nodes) under the attribute Doors is "**Acc**."

Pruning can be done on the nodes Safety > High>Buying >Low> Maintenance as the majority of the class labels (leaf nodes) under the attribute Maintenance is "**Acc**."

When Safety=Medium branch;

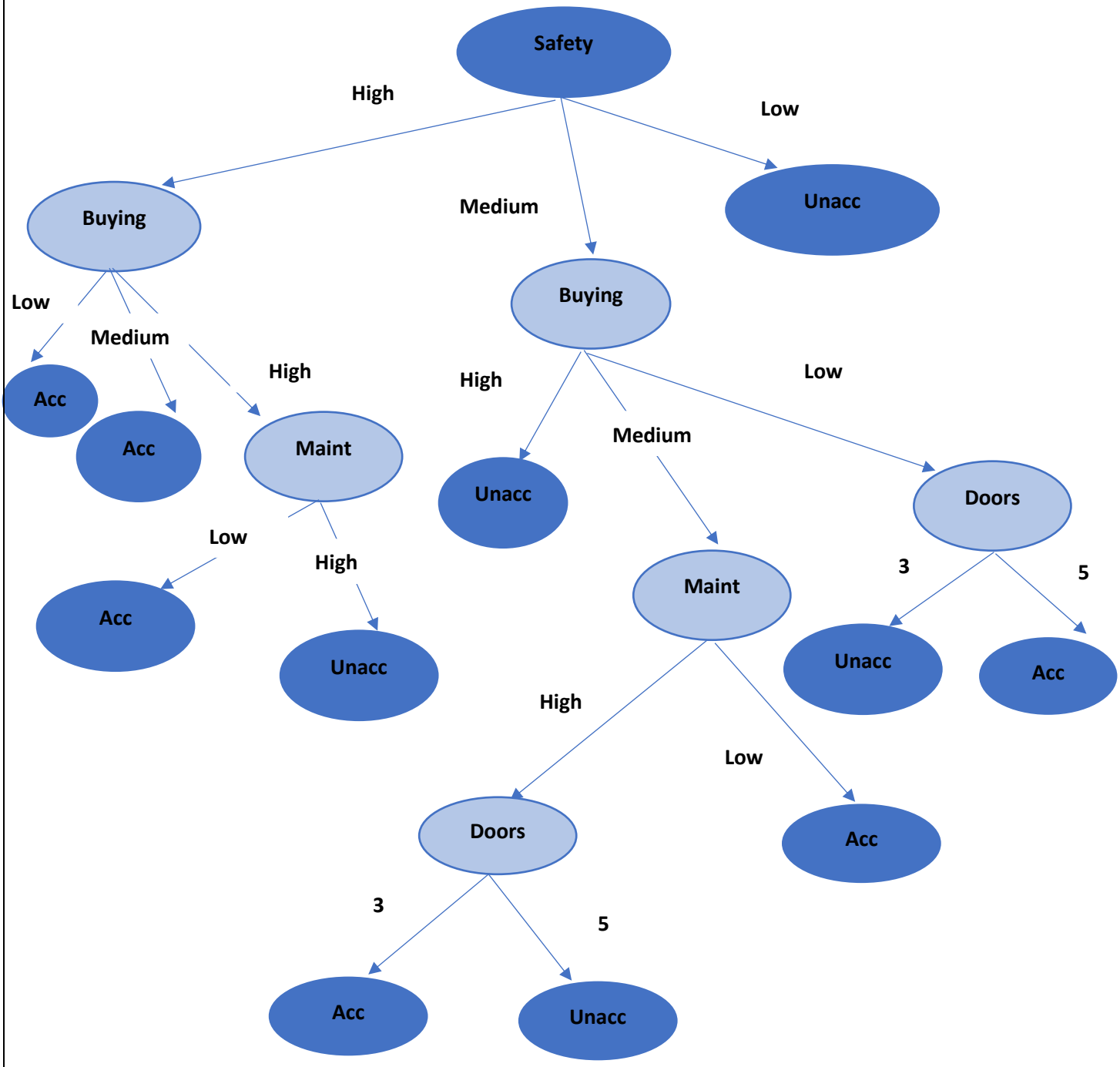
Pruning can be done on the nodes Safety > Medium>Buying >High> Maintenance as the majority of the class labels (leaf nodes) under the attribute Maintenance is "**Unacc**."

Pruning can be done on the nodes Safety > Medium>Buying >Medium> Maintenance> Low as the majority of the class labels (leaf nodes) under the attribute Doors is "**Acc**."

Pruning can be done on the nodes Safety > Medium>Buying >Low> Doors > 3 > Maintenance as the majority of the class labels (leaf nodes) under the attribute Maintenance is "**Unacc**."

Pruning can be done on the nodes Safety > Medium>Buying >Low> Doors > 5 > Maintenance as the majority of the class labels (leaf nodes) under the attribute Maintenance is "**Acc**."

So, **the decision tree after Pruning** would be as:



Result (Comparison of Training Dataset and Testing Dataset):

The training dataset is used to train the model to predict a given outcome which can be seen from the pruned decision tree of the Training Dataset.

We can check the Testing dataset (Actual results) with post-pruned Training dataset (Predicted results) and apply the outcome in the Confusion Matrix by calculating True Positive, True Negative, False Positive, and False Negative effects.

The result can be applied as shown in the table below:

buying	maint	doors	safety	Actual	Predicted
high	low	3	med	acc	unacc
low	high	5	low	unacc	unacc
high	high	5	high	unacc	unacc
low	high	5	high	unacc	acc
med	low	5	med	acc	acc
high	low	3	med	unacc	unacc
med	low	5	med	acc	acc
high	high	3	low	unacc	unacc
high	low	5	high	unacc	acc
low	high	5	high	unacc	acc
high	low	5	med	acc	unacc
high	high	3	med	unacc	unacc
low	low	5	low	unacc	unacc
high	high	3	med	unacc	unacc
high	high	5	high	unacc	unacc
high	high	3	med	acc	unacc
med	high	5	med	unacc	unacc
low	low	3	low	unacc	unacc
high	low	3	high	unacc	acc
med	high	3	med	acc	acc

The Confusion Matrix generated from the above comparison is:

Confusion Matrix			
		Predicted	
		Accepted	Unaccepted
Actual	Accepted	3	3
	Unaccepted	4	10

And, the results obtained are as follows:

Accuracy	**0.65**
Error Rate	**0.35**
Precision	**0.429**
Recall	**0.5**

It suggests that the model is making a correct prediction of only **65%** and is not that robust when this particular test data is applied to it.

A r**ecal**l value of **50%** suggests that the model is half correct when the actual value is positive, and the predicted value is correct.

Also, the **precision** value shows that the model's odds have made a correct prediction when the model predicts a positive value is **42.9%.**

(3) Application of Decision Tree-based Classification Methods:

A decision tree is a type of supervised learning. Subsets of data are trained to create the decision tree. A segmentation attribute is assigned to each node in the decision tree. Category ownership is represented by leaf nodes. Through the process of training the subset repeatedly and finally forming a tree, we can identify the correct decision set. The expected outcome is the root of the tree. (Hui, Jing, and Tao, 2011)

One such application of decision tree is made in understanding the criminal behavior pattern in a particular region that would enable the police to make decisions rapidly, legitimately, and scientifically. Here, the most common decision tree algorithm ID3 based on the calculation of the Information gain of each node is used. The top-down approach is used to ensure a simple tree is obtained and attributes are allocated according to information gain. A decision tree for a multi-attribute, multi-object data table would be enormous using the decision tree algorithm, and hence the attributes are divided into condition attributes(non-essential) and decision attributes(essential). This is done to improve the efficiency and accuracy of the

model. The local optimization problem is avoided by appropriately reducing the condition attribute, thereby maintaining the classification ability of the condition attribute as compared to the decision attribute. The nodes are selected as per the descending values of the information gain, and the rules obtained are in order with the actual situation. (Hui, Jing, and Tao, 2011).

Another application of decision tree can be seen in the fault diagnosis of rolling bearing based on the permutation Entropy. Rolling bearing is one of the most important parts of rotating machinery and the faultiest part that often leads to delay in production and endangers personal safety. Presently, the vibration signal of rolling bearing is extracted through short-time Fourier transform, Wigner-Ville distribution, and Wavelet transform. Still, the vibration signal does not show the desired output because of working conditions and a complex transmission system. Variational Mode Decomposition (VMD) is another method, irrespective of time-frequency interference, to calculate vibration signals. Here, the vibration signal is broken down into band-limited intrinsic mode functions (BIMF), from which the permutation entropy of the same can be extracted and used as the fault feature information. Decision tree usually adopts attribute selection method based on information gain rate, which is further based on entropy. VMD is used to analyze the vibration signal of rolling bearings, and permutation entropy is calculated for each BIMF as a quantitative feature. The decision tree algorithm is then used to predict four kinds of rolling bearing states, and the results suggest that the recognition rate of the decision tree model is effective in fault diagnosis of rolling bearing. (Chen, Yang, and Lou, 2019).

Question 2:

(1.a) The process of data mining is a creative one requiring many different skills and knowledge. In order to transform business problems into data mining tasks, we need a standard approach that will suggest appropriate data transformations and data mining techniques, as well as a way to analyze the results and document the experience. The CRISP-DM (CRoss Industry Standard Process for Data Mining) model was developed to address some aspects of the problems by providing a framework for carrying out data mining projects that is independent of both the industry sector and the technology. This process model is designed to reduce the cost, improve the reliability, repeatability, quality, and manageability of large data mining projects.

The CRISP-DM methodology can be described as a hierarchical process model, which comprises four levels of abstraction (from general to particular): phases, generic tasks, specialized tasks, and process instances as shown in Figure 1.

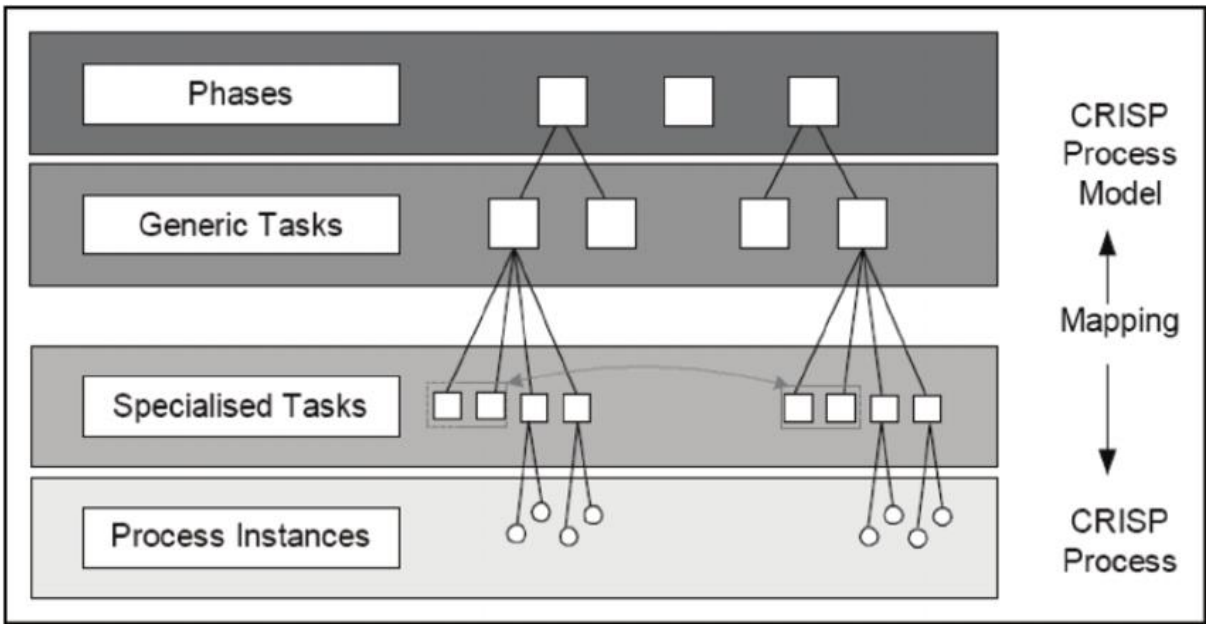


Figure 1: Four level breakdown of the CRISP-DM model for Data Mining (Gersten, Wirth and Arndt Predictive Modeling in Automotive Direct Marketing: Tools, Experiences and Open Issues, 2000)

There are only a few **phases** to the data mining process at the top. The second level is called **generic** since it is designed to cover all scenarios that may be encountered during data mining. In the third level, the **specialized tasks** are described as to how they should be carried out in specific circumstances. Fourth, the **process instance** level consists of a record of data mining actions, decisions, and outcomes. (Wirth and Hipp, 2000).

An overview of the lifecycle of a data mining project is provided by the CRISP-DM reference model. It is composed of the phases of a project, their respective tasks, and their outputs. The six phases of a data mining project are outlined as shown in Figure 2. Symbolically, the outer circle in Figure 2 depicts the cyclical nature of data mining.

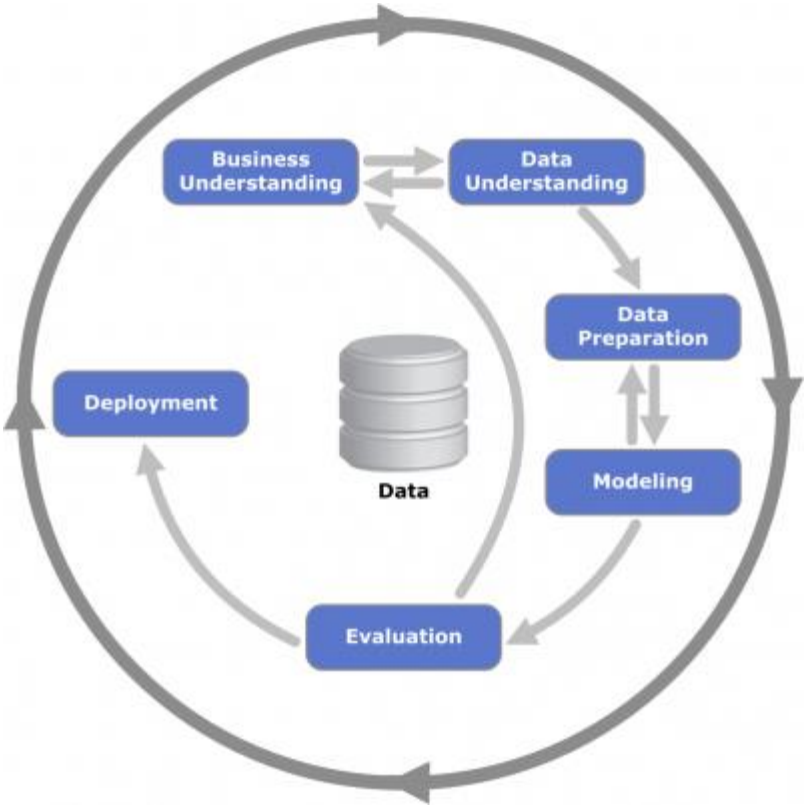


Figure 2: CRISP-DM model for Data Mining (Sridharan, 2018)

The model is explained as follows:

- **Business Understanding-** Business-oriented approach to understanding the project objectives and requirements. This knowledge needs to be formulated as a data mining problem, and a preliminary plan is developed.
- **Data Understanding-** This phase begins with an analysis of the data, followed by activities to gain familiarity with the data and to identify potential data quality issues.
- **Data Preparation-** Preparation of the data covers all activities associated with building the final dataset from raw data (data that will be fed into the modeling tool(s)).
- **Modeling-** At this stage, you select various models and apply them, calculating the parameters to obtain optimal values. Different data mining techniques such as K-means Clustering, Association Rule Mining or Fuzzy Logic, etc., may be used for one type of data mining problem. However, specific data formats are required for some techniques.
- **Evaluation-** In the process of identifying whether any important business issues have been overlooked, one of the key objectives is to determine if they need further consideration. An end-of-phase decision should be made on how to use the data mining results.
- **Deployment-** In the deployment phase, the process can be simple or complex, such as the generation of a report or the implementation of a repeatable data mining method. It is usually necessary to organize and present the knowledge so that the client can make use of it. (Wirth and Hipp, 2000)

A real-life scenario where the K-means Clustering technique can be applied is **Weather forecasting or weather pattern prediction** by considering attributes such as maximum

temperature, minimum temperature, humidity, precipitation, etc over a given period of time. Data mining is concerned with the discovery of hidden patterns within a data set. In this dataset, we apply K-means and partitional clustering techniques to predict patterns in the dataset.

A clustering algorithm is an unsupervised learning algorithm. Data objects are grouped by cluster analysis based on information found within the data that describes the objects and their relationships. Data set instances are characterized by the characteristics they possess, such as features or attributes that define the different aspects of the instance. K-Means clustering algorithm divides a set of n objects into k clusters according to input parameter k.

The measure of cluster similarity is based on the mean of the attributes in the cluster. (Shobha and Asha, 2017).

(1.b) One of the most straightforward unsupervised learning calculations is K-means that takes care of the various attributes in the weather prediction dataset. Through a specific number of attributes, this technique goes for a basic and straightforward approach to group a given information collection. It is essential to characterize k focal points, one for each group. A sensible way of thinking ought to be taken based on various types of outcomes in various attributes. (Raghavendran and R, 2019)

Also, K-Means Clustering is being used because of the following reasons:

(i) A fast, robust, simple to understand, and comparatively Efficient technique.

(ii) In the case of distinct datasets, it gives the best results.

(iii) It is flexible and easy to understand.

(iv) Enhances accuracy and reduces the computational cost. (Priya Patro, 2021)

(2.a) The dataset has been used to predict the weather pattern of a town called Jaipur located in the western part of India. The data consists of 13 variables and 115 data points that determine the weather pattern of a particular area and has been taken over a period of 3 years. Two types of variables can be seen in the dataset: **Ordinal variable**, which is the date on which the other attributes have been recorded, and the rest are **Continuous variables** such as Mean temperature, mean pressure, Mean dewpoint, Maximum humidity, Minimum humidity, Maximum temperature, Minimum temperature, Maximum dewpoint, Minimum dewpoint, Maximum pressure, Minimum pressure, precipitation.

The dataset is obtained from the website **Kaggle**: https://www.kaggle.com/rajatdey/jaipur-weather-forecasting.

(2.b) The potential insights that can be generated from the dataset would be the optimal number of clusters(k) through the Elbow method and visualizing different clusters and their attributes such as 'means' and 'within the sum of squares' and through histogram of specific attributes that determines weather forecast over a given period of time.

(3.a) (i) The Elbow method is applied to obtain the optimal number of clusters that would be used in the dataset for further predictions. The code used here is:

```
fviz_nbclust(J1.subset, kmeans, method = "wss") +
  labs(subtitle = "Elbow method")

km <- kmeans(J1.subset, centers=3, nstart=100)
```

And, the graph obtained through the Elbow method is:

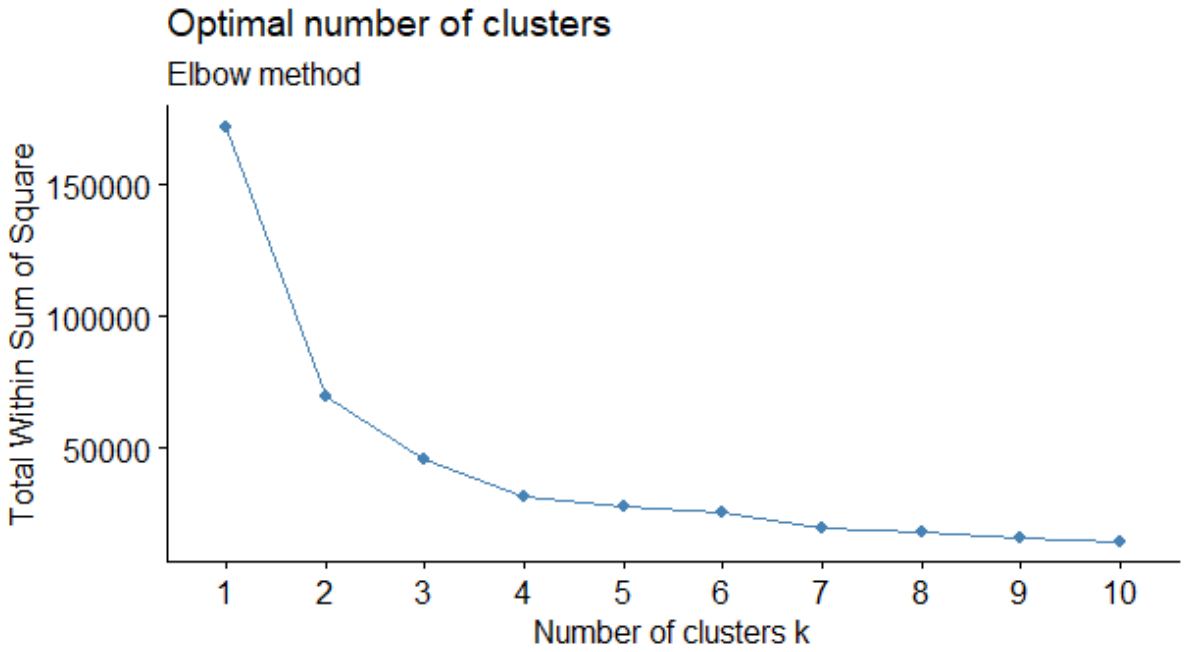


Figure 3: Optimal Number of Clusters

(ii) As the optimal number of clusters is three where the Sum of Squares curve tends to decrease, three distinct cluster plots are created with sizes of 42, 27, and 46, respectively, which suggests that there are three different groups of similar data or data tending to three separate seasons of a region over a given period. The code used here is:

```
fviz_cluster(list(data=J1.subset, cluster=km.clusters))
```

The cluster plot obtained is as follows:

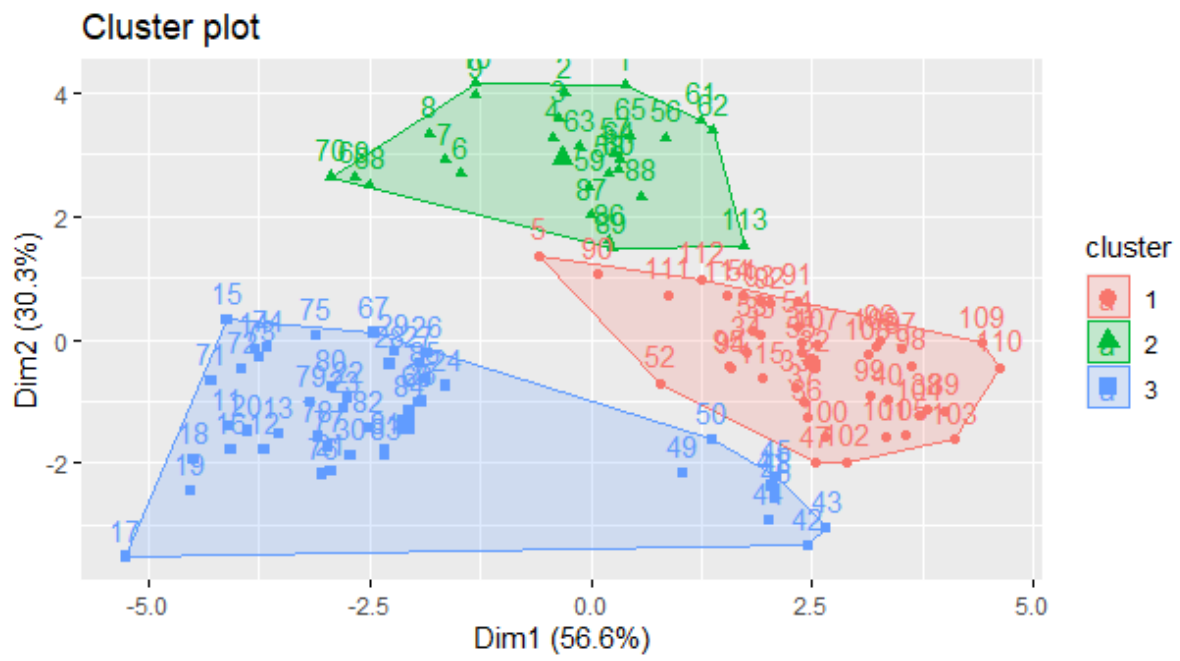


Figure 3: Cluster Plot

(iii) The Histogram is obtained for the attribute Mean Temperature across the dataset. This shows that Jaipur city has a temperature of 25^{0}-28^{0}C throughout the year, tending towards the warm weather. The code used here is:

```
hist(J1.subset$'meantempm', xlab="meantempm", main= "Histogram of meantempm", breaks = sqrt(nrow(J1.subset)))
```

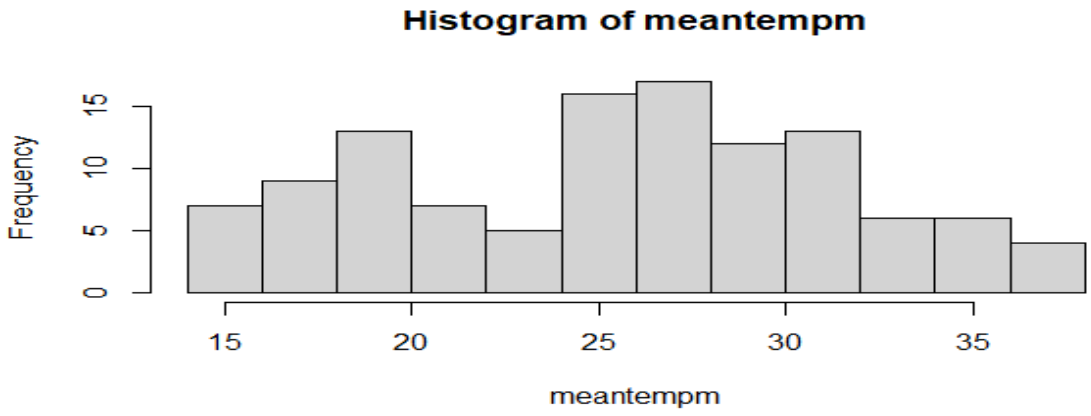


Figure 4: Histogram of Mean Temperature

The Histogram of Mean dew Temperature is obtained across the dataset, which shows that the mean dew point over a given period of time is 22^0C Td – 23^0C Td, meaning the humidity in the region is on the higher side with a very little amount of rainfall. The code used here is:

```
hist(J1.subset$'meandewptm', xlab="meandewptm", main= "Histogram of meandewptm", breaks = sqrt(nrow(J1.subset)))
```

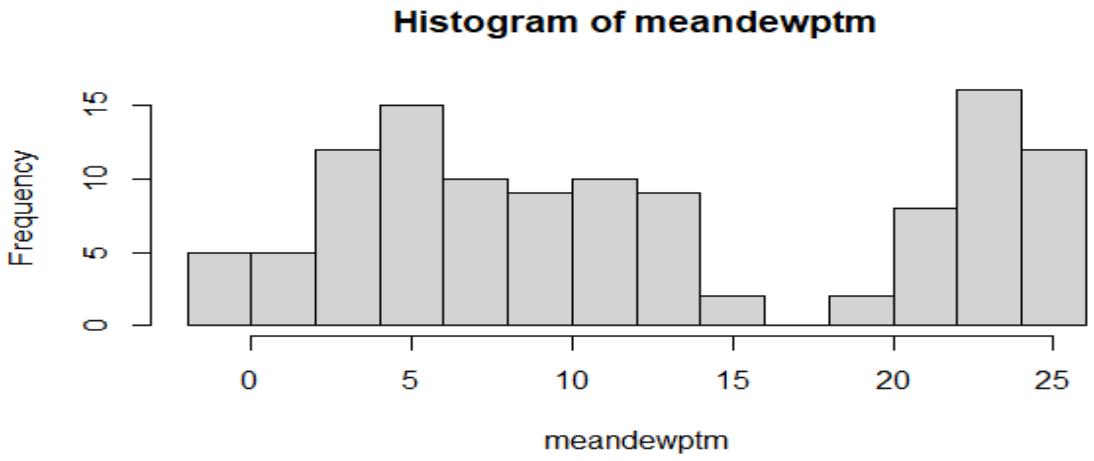


Figure 5: Histogram of Mean dew Temperature

The Histogram for Mean pressure is obtained from the dataset, which shows that Jaipur city has a mean pressure of around 1015 hPa, which tends to be on the higher side throughout the year, tending towards fine and dry weather with very little amount of rainfall. The code used here is:

```
hist(J1.subset$'meanpressurem', xlab="meanpressurem", main= "Histogram of meanpressurem", breaks = sqrt(nrow(J1.subset)))
```

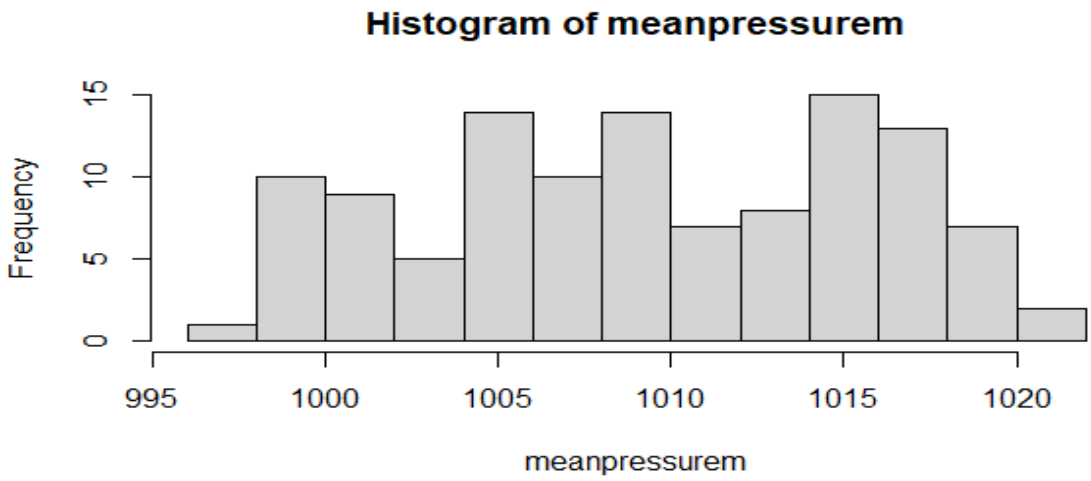


Figure 6: Histogram of Mean Pressure

The Mean temperature, Mean pressure and Mean dew temperature is mentioned because it forms one of the most important parameters for the prediction of rainfall in a particular area apart from weather forecast as well.

(3.b) The novelty of application of K-means clustering in this dataset helps in understanding the weather pattern of a given area over a period of time and segmenting three different seasons in a year based on the grouping of similar datasets.

The significance of this application is that it represents a clear picture of the nature and relationship of data in the given dataset, and the bigger the size of the cluster, the more prolonged the season is in that particular area.

Conclusion:

K-means clustering is a data modeling technique helpful in separating a group of data into clusters depending upon the similarity of nature of the data and thus helping in interpreting the dataset logically.

References:

Chen, X., Yang, Z. and Lou, W., 2019. Fault Diagnosis of Rolling Bearing Based on the Permutation Entropy of VMD and Decision Tree. In: *2019 3rd International Conference on Electronic Information Technology and Computer Engineering (EITCE)*. [online] IEEE. Available at: https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=9095187[Accessed 9 January 2022].

Gersten, W., Wirth, R. and ArndtPredictive Modeling in Automotive Direct Marketing: Tools, Experiences and Open Issues, D., 2000. Predictive Modeling in Automotive Direct Marketing: Tools, Experiences and Open Issues. In: *Proceedings of the sixth ACM SIGKDD international conference on Knowledge discovery and data mining*. [online] Researchgate. Available at: https://www.researchgate.net/publication/221653889_Predictive_modeling_in_automotive_direct_marketing_tools_experiences_and_open_issues[Accessed 14 January 2022].

Hui, W., Jing, W. and Tao, Z., 2011. Analysis of decision tree classification algorithm based on attribute reduction and application in criminal behavior. In: *2011 3rd International Conference on Computer Research and Development*. [online] IEEE. Available at: https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=5763966[Accessed 9 January 2022].

Priya Patro, L., 2021. Understanding K-means Clustering. [Blog] *Medium*, Available at: https://priya231299.medium.com/understanding-k-means-deeply-8a6a7ff56051[Accessed 10 January 2022].

Raghavendran, R. and R, J., 2019. WEATHER PREDICTION USING K-MEANS CLUSTERING AND NAIVE BAYES ALGORITHMS. *Journal of Emerging Technologies and Innovative Research (JETIR)*, [online] 6(6). Available at: https://www.researchgate.net/publication/333917116_WEATHER_PREDICTION_USING_K-MEANS_CLUSTERING_AND_NAIVE_BAYES_ALGORITHMS[Accessed 10 January 2022].

Shobha, N. and Asha, T., 2017. Monitoring weather based meteorological data: Clustering approach for analysis. In: *2017 International Conference on Innovative Mechanisms for Industry Applications (ICIMIA)*. [online] IEEE. Available at: https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=7975575[Accessed 10 January 2022].

Sridharan, M., 2018. CRISP-DM: A framework for Data Mining & Analysis. [Blog] *Think Insights*, Available at: https://thinkinsights.net/digital/crisp-dm/[Accessed 10 January 2022].

Wirth, R. and Hipp, J., 2000. *CRISP-DM: Towards a Standard Process Model for Data Mining*. [ebook] Available at: http://www.cs.unibo.it/~montesi/CBD/Beatriz/10.1.1.198.5133.pdf[Accessed 10 January 2022].