Name: P. Meghana

Reg No: 22BCE20209

Slot: G_1

ML ASSIGNMENT – 2

1) Implement a decision tree using CHAID Algorithm.

Day	Outlook	Temp	Humidity	Wind	Decision
1	Sunny	Hot	High	Weak	No
2	Sunny	Hot	High	Strong	No
3	Overcast	Hot	High	Weak	Yes
4	Rain	Mild	High	Weak	Yes
5	Rain	Cool	Normal	Weak	Yes
6	Rain	Cool	Normal	Strong	No
7	Overcast	Cool	Normal	Strong	Yes
8	Sunny	Mild	Normal	Weak	No
9	Sunny	Cool	High	Weak	Yes
10	Rain	Mild	Normal	Weak	Yes
11	Sunny	Mild	Normal	Strong	Yes
12	Overcast	Mild	High	Strong	Yes
13	Overcast	Hot	Normal	Weak	Yes
14	Rain	Mild	High	Strong	No

Chi-square $= \sqrt{(y-y')^2/y'}$

Outlook:

Sunny $\rightarrow$ Total = 5, Expected = 2.5

$\therefore$ Chi-square yes = 0.316, $\therefore$ Chi-square No = 0.316.

Overcast $\rightarrow$ Total = 4, Expected = 2.

$\therefore$ Chi-square Yes $= \sqrt{(4-2)^2/2} = 1.414$, Chi-Square No = 1.414

Rain → Total = 5, Expected = 2.5.

∴ chi-square Yes = 0.316, chi-square No = 0.316

chi-square value of outlook

→ $0.316 + 0.316 + 1.414 + 1.414 + 0.316 + 0.316 = 4.092$.

Temperature:

Hot → Total = 4, Expected = 2

∴ chi-square · Yes = 0, chi-square · No = 0.

Mild → Total = 6, Expected = 3.

∴ chi-square Yes = 0.577, Chi-square No = 0.577.

cool → Total = 4, Expected = 2.

∴ chi-square Yes = 0.707, chi-square No = 0.707.

∴ chi-square value of temp = $0 + 0 + 0.577 + 0.577 + 0.707 + 0.707$
$= 2.569$.

Humidity:

High → Total = 7, Expected = 3.5

∴ chi-square Yes = 0.267, chi-square No = 0.267

Normal → Total = 7, Expected = 3.5

∴ chi-square Yes = 1.336, chi-square No = 1.336.

∴ chi-square value of Humidity = $0.267 + 0.267 + 1.336 + 1.336$
$= 3.207$.

Wind:

weak → Total = 7, Expected = 3.5.

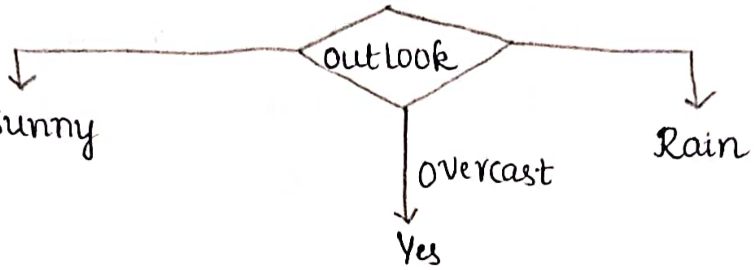
chi-square Yes = 0.802, chi-square No = 0.802.

strong → Total = 6, Expected = 3

chi-square Yes = 0, chi-square No = 0.

∴ chi-square value of wind = $0 + 0 + 0.802 + 0.802 = 1.604$

∴ From the following outlook has the Highest-Chisquare Value, the most significant feauture, the root Node.



Outlook = Sunny

→ 5 instances:

	Yes	No	Total	Expected	Chi-square	No
Hot	0	2	2	1	1	1
Mild	1	1	2	1	0	0
Cold	1	0	1	0.3	0.707	0.707

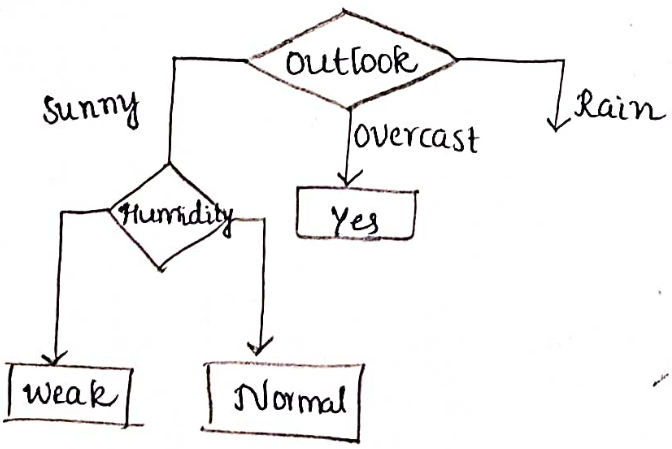
∴ for sunny $= 1+1+0.707+0.707+1+0 = 3.414$

Humidity:

	Yes	No	Total	Expected	Chi-square	No
Weak	1	2	3	1.5	0.408	0.408
Strong	1	1	2	1	0	0

Wind $= 0.408+0.408+0+0 = 0.816$.

∴ Humidity has the Most dominant feauture.



Rain outlook branch:

→ Temperature for main outlook.

	Yes	No	Total	Expected	Chi-square Yes ~~Yes~~	No
Mild	2	1	3	1.5	0.408	0.408
Normal	1	1	2	1	0	0

∴ Chi-square temp $= 0.408 + 0.408 + 0 + 0 = 0.816$.

Humidity:

	Yes	No	Total	Expected	Chi-square Yes	No
High	1	1	2	1	0	0
Normal	2	1	3	1.5	0.408	0.408

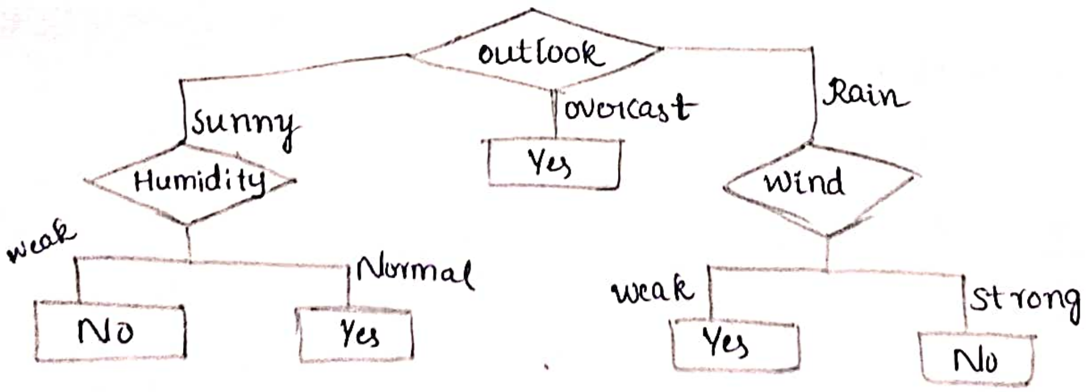
$= 0 + 0 + 0.408 + 0.408 = 0.816$.

Wind:

	Yes	No	Total	Expected	Chi-square Yes	Chi-square No
weak	3	0	3	1.5	1.225	1.225
strong	0	2	2	1	1	1

Chi-square wind $= 1.223 + 1.225 + 1 + 1 = 4.449$.

∴ wind feauture is significant. the decision tree is



2) Explain Multi-class classification techniques

→ Multi-class classification is a type of supervised learning used to classify instance into one of three or more classes. Unlike Binary classification, multi-class classification deals with a Broader Range of categories, Here are some common techniques used in multi-class classification.

1. One-vs-all (OVA) or One-vs-Rest (OVR):

→ Approach: This technique splits a multi-class problem into Multiple Binary classification problems. For each class, a seperate classifier is trained to distinguish between that class and all other classes. So, if there are N classes, N classifiers are created.

→ Prediction: During prediction, each classifier outputs a score or probability for its class, and the class with the highest score is choosen as the find prediction.

→ Pros: Simple and effective, works well for many linear models.

2. One-vs-one (OVO):

→ Approach: This technique creates a seperate classifier for each pair of classes. If there are N classes. $\left(\frac{N}{2}\right) = \left(\frac{N(N-1)}{2}\right)$ classifiers are created.

→ Pros: Often works well with complex, Non-linear Models and can handle imbalanced datasets.

→ Cons: It becomes computationally expensive for a large number of classes as it requires Numerous classifiers.

3. Decision Tree and Random Forests:

→ Approach: Decision trees can natively handle multiple classes by splitting the data based on criteria that optimize seperation among classes.

→ Pros: Works well with both linear and Non-linear relationships, robusts to overfitting.

→ Cons: It can become complex with deep trees, computationally expensive in random forests.

4. K-Nearest Neighbour (K-NN):

→ Approach: KNN is a distance-based method that assigns a class label based on the majority class of its K closest Neighbours.

→ Pros: Simple and interpretable; Non-parametric, so it makes no assumptions about data distribution.

→ Cons: It can be slow with large datasets and is sensitive to irrelevant feautures and Noisy data.

5. Naive Bayes Classifiers:

→ Approach: Naive bayes uses bayes theorem with a "Naive" assumpti-on that each feauture contributes independently to the probability of each class.

→ Pros: Computationally efficient, works with small dataset and next classification.

→ Cons: Assumes Independence among feautures, which is often not realistic, can struggle with continuous feautures.

3) Explain Bagging and Boosting with a suitable example.

Bagging (Bootstrap Aggregration):

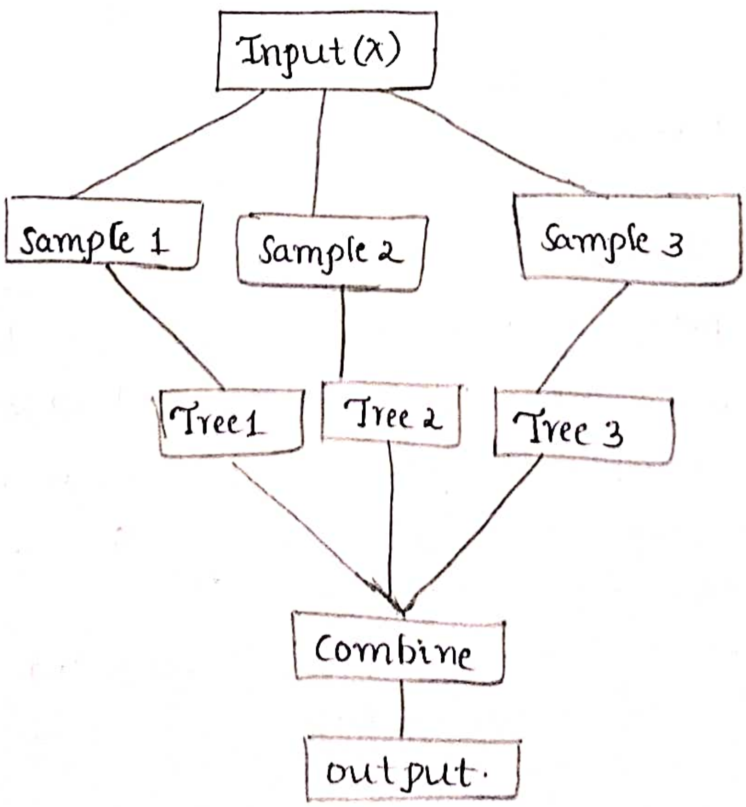
→Approach: Bagging trains multiple models independently on different random of subsets of the original training dataset. These subsets are created using Bootstrapping where each subset is created by sampling with Replacement, meaning that some samples may appear more that once in a subset, while others might be absent.

→Goal: Reduce variance and prevent overfitting.

Ex:

→ Suppose we want to predict whether a patient has a certain disease based on health data. We create 10 Subsets from the original dataset using bootstrapping and train a decision tree and each subset. Bagging helps because individual decision trees might overfit, but aggregating their predictions leads to a more generalised and accurate so result.

Bagging Ensemble:



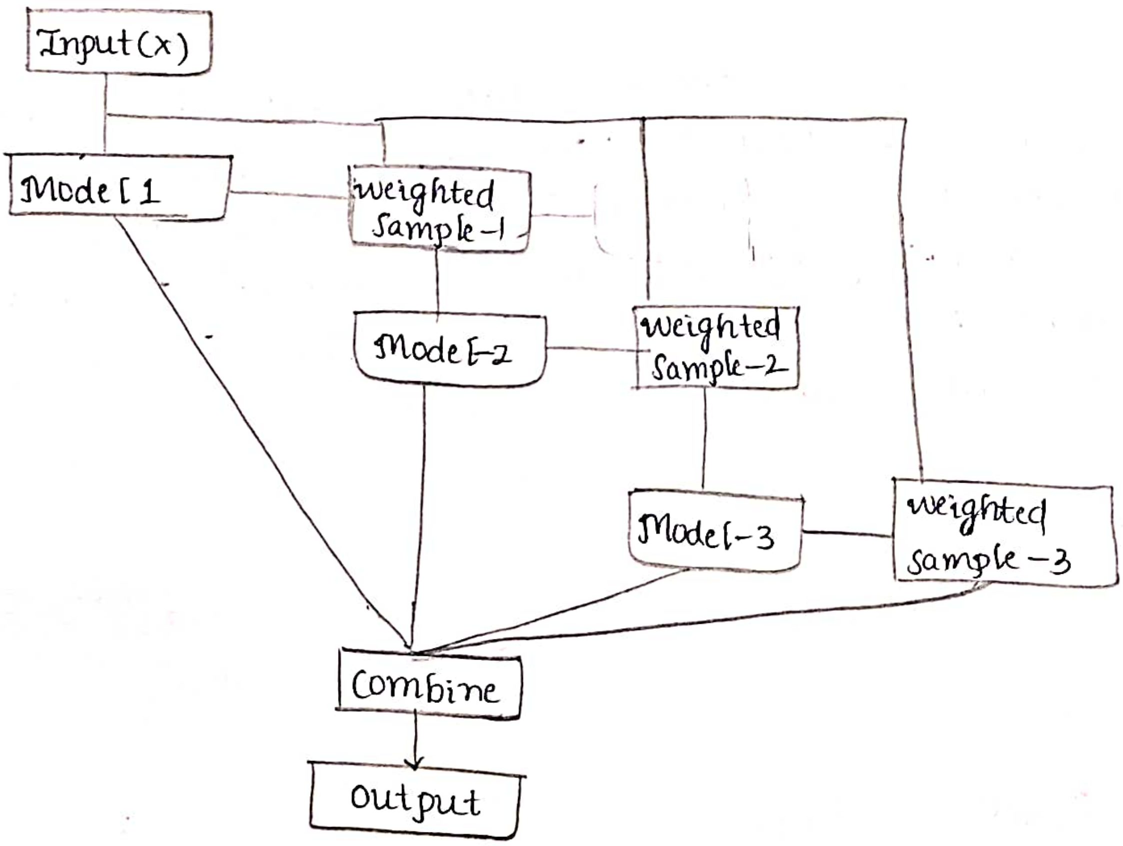
Boosting Ensemble Learning:

→ Boosting is a Machine learning Ensemble technique that by converting weak learners into strong learns reduces Bias and Variance.

→ The weak learners are applied to the dataset in a sequential training set. The first step is building an initial Model andfitting in into the training set.

→ A second model that tries to fix the errors generated by the first model is then fitted.

Boosting ensemble learning:



Ex: Suppose we are predicting whether customers will buy a product based on features like age income, and browsing history

→ The first weak classifier might predict with only 60% accuracy. The incorrectly classified samples are given higher weights in the next.

→ This process continuous for several sounds, and in the end, all models, predictions are combined, with a weighted approach.

4) Explain the basic Architecture of ANN.

The basic architecture of an Artifical Neural Network (ANN) mimics the structure and function of the human brain. ANNS consist of interconnected processing elements called neurons (or) nodes that are organized in layers.

1. Neurons (Nodes)

→ Each Neuron recieves one (or) more inputs, process. and produces an output.

→ Inputs are multiplied by weights that represent the strength (or) importance of each point input.

→ The Neuron them sums these weighted inputs and applies an active function.

→ 2. Layers of an ANN.

→ Input layer:

→ This is the first layer in the Network.

→ Neurons in this layer recieve the raw data directly from the dataset.

→ Hidden layer:

→ These are the intermediate layers between the input and output layers.

→ Hidden layers apply weights, Biases, and Activation functions to the inputs, creating Non-linear transform and learning complex patterns in the data.

→ output layer:

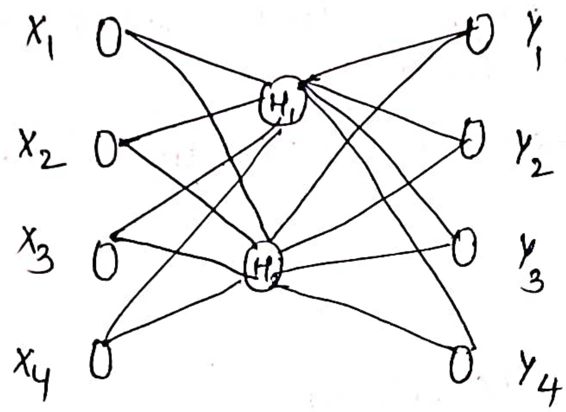
→ The final layer in the network which produces the ANNS prediction.

→ The Number of Neurons in this layer typically corresponds to the number of output classes or one node for regression tasts.

3. Connections (Activation functions):

- Activation function introduce non-linearity, which is crucial for learning complex patterns in data.

- Common activated function includes:
  1. Sigmoid: outputs values between 0 and 1, often used in a binary classification.
  2. ReLU (Rectified Linear unit): Outputs to represent probabilities for input directly if it's positive, Otherwise, it outputs zero



5) Construct a simple perception neural network using the AND TRUTH table. (learning rate = 0, bias = 0, $\omega_1 = \omega_2 = \omega_3 = \omega_4 = 0$), use binary activation function.

Input (X_1)	Input (X_2)	Target output (AND)
0	0	0
0	1	0
1	0	0
1	1	1

→ Initial parameters:

Learning table = 0

Bias table = 0

Initial weights $w_1 = 0$ and $w_2 = 0$

if $y_n > 0, 1$

$y_n < 0, 0$

$y = f(w_1 x_1 + w_2 x_2 + bias)$

$y = 1$ (By the binary activation function)

for $(x_1 = 0, x_2 = 0)$, $y = 1$

$x_2 = 0, x_2 = 1$, $y = 1$

$x_1 = 0, x_2 = 0$, $y = 1$

$x_1 = 1, x_2 = 1$, $y = 1$

Since the weights are Bias are zero and donot change, the perception will always output $y=1$ for every input.

Therefore, the perception fails to learn the AND function under these conditions.