Name - Yash Lakhtariya
Enrollment number - 21162101012
Branch - CBA Batch - 61
DMW Assignment 2

1. **What is classification? How does it differ from prediction?**

- **Classification Definition**: Classification involves categorizing data into predefined classes or categories. It's a technique used in machine learning and statistics, where the goal is to predict the class label of new observations based on past observations with known class labels.
- **Process of Classification**: In classification, algorithms analyze a training dataset to learn a mapping function from input variables (features) to the output variable (class label). This learned model is then applied to predict the class labels of unseen data.
- **Example**: For instance, in email spam detection, the aim is to categorize emails as either spam or not spam. The classification algorithm learns from labeled examples of emails (spam and non-spam) and then predicts the category of new, unseen emails based on features like keywords and sender information.

Difference from Prediction:

- **Classification Focus**: Classification specifically deals with assigning data points to predefined categories or classes.
- **Prediction Overview**: Prediction, however, is a broader term that encompasses various tasks, including classification. It aims to forecast future outcomes based on historical data but may involve predicting numerical values (regression) or identifying patterns (classification).

Name - Yash Lakhtariya
Enrollment number - 21162101012
Branch - CBA Batch - 61
DMW Assignment 2

2. **Differentiate supervised learning and unsupervised learning.**

Supervised Learning:

- **Learning from Labeled Data**: In supervised learning, the algorithm learns from a labeled dataset, where each training example contains input features along with corresponding output labels.
- **Mapping Function**: The objective is to learn a mapping function from input variables to output labels, enabling the algorithm to predict output labels for new, unseen data.
- **Examples**: Supervised learning tasks include classification (e.g., spam detection, image recognition) and regression (e.g., predicting house prices, stock prices).

Unsupervised Learning:

- **Learning from Unlabeled Data**: Unsupervised learning involves learning from an unlabeled dataset, where predefined output labels are absent.
- **Discovering Patterns**: The goal is to uncover hidden patterns or structures in the data, such as clusters or associations, without explicit guidance.
- **Exploration of Data**: Unsupervised learning algorithms explore the data to identify inherent relationships or groupings among the data points.

Key Differences:

- **Data Requirement**: Supervised learning requires labeled data for training, while unsupervised learning operates with unlabeled data.
- **Prediction vs. Discovery**: In supervised learning, the algorithm predicts output labels based on input features, whereas in

unsupervised learning, it discovers patterns or structures in the data without explicit output labels.

3. **Briefly outline the major steps of decision tree classification.**

Data Preprocessing:

- The initial step involves gathering and preparing the dataset. This includes handling missing values, encoding categorical variables, and dividing the dataset into training and testing subsets.

Feature Selection:

- Decision trees operate by selecting the most effective feature at each node to divide the data. Various criteria like Gini impurity or entropy are employed to gauge the quality of a split. The aim is to pinpoint the feature that maximizes information gain or minimizes impurity.

Building the Tree:

- Beginning with the root node, the decision tree algorithm recursively splits the dataset based on the chosen features. At each node, it picks the feature that best separates the data into distinct categories. This process continues until a stopping criterion is met, such as reaching a maximum tree depth, having a minimum number of samples at a node, or achieving pure leaf nodes (nodes containing only one category).

Pruning (Optional):

- Pruning serves to prevent overfitting by eliminating sections of the tree that don't significantly enhance classification accuracy. This

entails simplifying the tree by consolidating branches or removing nodes that contribute minimally to the accuracy on the validation set.

Prediction:

- Once the decision tree is constructed, it's ready to make predictions on new, unseen data. Starting from the root node, each feature's value is compared to the split criteria, and the corresponding branch is followed until a leaf node (containing the class label) is reached. The class label of the leaf node is assigned as the predicted class for the input data.

Evaluation:

- Lastly, the performance of the decision tree model is assessed using metrics like accuracy, precision, recall, and F1 score on the test dataset. These metrics provide insights into the model's generalization to unseen data and its ability to accurately classify instances into their respective categories.

4. **What are the various termination conditions for decision trees?**

Decision trees halt their splitting process based on specific termination conditions to avoid overfitting and maintain an optimal level of complexity. Here are some common termination conditions:

- **Maximum Tree Depth**: Setting a limit on the tree's depth prevents it from growing excessively deep, which could lead to overfitting the training data.
- **Minimum Samples per Leaf**: Ensuring each leaf node comprises a minimum number of samples prevents the formation of small, isolated nodes prone to capturing noise in the data.

Name - Yash Lakhtariya
Enrollment number - 21162101012
Branch - CBA Batch - 61
DMW Assignment 2

- **Minimum Samples per Split**: Requiring a minimum number of samples to split a node ensures that nodes are divided only when there's sufficient data to make reliable decisions.
- **Maximum Number of Nodes or Leaves**: Constraining the total number of nodes or leaves in the tree prevents it from becoming overly intricate, enhancing interpretability.
- **Impurity Threshold**: Halting the splitting process when the impurity (e.g., Gini impurity, entropy) drops below a specific threshold ensures that further splits don't notably enhance classification purity.
- **Cross-Validation**: Utilizing cross-validation techniques to evaluate the tree's performance on validation data and terminating the growth process when additional splitting fails to enhance performance.

These termination conditions regulate the size and complexity of the decision tree, preventing it from closely fitting the training data and thereby enhancing its ability to generalize to unseen data.

5. **List various attribute selection measures. What are the different issues of these measures?**

Attribute selection measures, also called splitting criteria, are essential for determining the optimal attribute to split the data at each node of the decision tree. Here are some common attribute selection measures along with their associated issues:

1. **Gini Impurity**: Gini impurity tends to favor attributes with numerous distinct values, potentially leading to biased splits.

Name - Yash Lakhtariya
Enrollment number - 21162101012
Branch - CBA Batch - 61
DMW Assignment 2

2. **Entropy**: Similar to Gini impurity, entropy also tends to favor attributes with a large number of distinct values.

3. **Information Gain**: Information gain may lean towards attributes with many distinct values, possibly resulting in biased splits.

4. **Gain Ratio**: Calculating gain ratio may be computationally intensive, particularly for datasets containing numerous attributes.

5. **Chi-square Statistic**: Chi-square statistic may not perform effectively with attributes having a large number of distinct values or sparse data.

Each attribute selection measure possesses its own strengths and limitations. The choice of measure should align with the dataset's characteristics and the specific requirements of the classification task.

6. What is overfitting? Explain the methods to avoid it.

Overfitting arises when a machine learning model captures noise or random fluctuations in the training data, learning patterns that don't generalize effectively to unseen data. Essentially, the model fits too closely to the training data, performing well on it but failing to perform adequately on new, unseen data.

Methods to avoid overfitting include:

1. **Cross-Validation**: Divide the dataset into multiple subsets for

training and testing. By assessing the model's performance on different subsets, you can gauge its ability to generalize to unseen data.

2. **Regularization**: Add a penalty term to the model's cost function to discourage overly complex models. Techniques like L1 (Lasso) and L2 (Ridge) regularization penalize large coefficients, mitigating overfitting.

3. **Feature Selection**: Choose only the most relevant features that contribute significantly to the model's predictive ability. Removing irrelevant or redundant features reduces model complexity, aiding in the prevention of overfitting.

4. **Early Stopping**: Monitor the model's performance on a validation set during training and halt training when performance begins to deteriorate. This prevents the model from excessively learning noise in the training data.

5. **Ensemble Methods**: Combine multiple weak learners (models) to create a stronger, more robust model. Techniques such as bagging, boosting, and random forests aggregate predictions from multiple models, reducing overfitting.

6. **Data Augmentation**: Expand the training dataset by generating synthetic data points or perturbing existing ones. This exposes the model to a broader range of examples, diminishing the risk of overfitting.

Name - Yash Lakhtariya
Enrollment number - 21162101012
Branch - CBA Batch - 61
DMW Assignment 2

By implementing these methods, models can be developed that generalize well to unseen data and are less susceptible to overfitting.

7. **Why is naive Bayesian classification called "naive"? Briefly outline the major ideas of naive Bayesian classification.**

Naive Bayesian classification earns its "naive" moniker due to its strong assumption of feature independence, which is often unrealistic in real-world data. Despite this simplification, naive Bayes classifiers can surprisingly excel in practice, particularly for text classification tasks.

The core concepts of naive Bayesian classification include:

➔ **Bayes' Theorem**: The classification is grounded in Bayes' theorem, which quantifies the probability of a hypothesis given certain evidence.

➔ **Feature Independence**: The "naive" aspect involves assuming that all features are conditionally independent given the class label. This implies that the occurrence of one feature does not influence the probability of another feature given the class label.

➔ **Calculating Class Probabilities**: Naive Bayesian classifiers compute the probability of each class label given the input features using Bayes' theorem. The class label with the highest probability is assigned to the input instance.

➔ **Estimating Feature Probabilities**: Training data is utilized to estimate the probabilities of each feature occurring given each class

label. These probabilities are crucial for calculating the likelihood term in Bayes' theorem.

➔ **Classification Rule**: Leveraging the estimated feature and class probabilities, naive Bayesian classifiers employ a classification rule to assign the most probable class label to new instances.

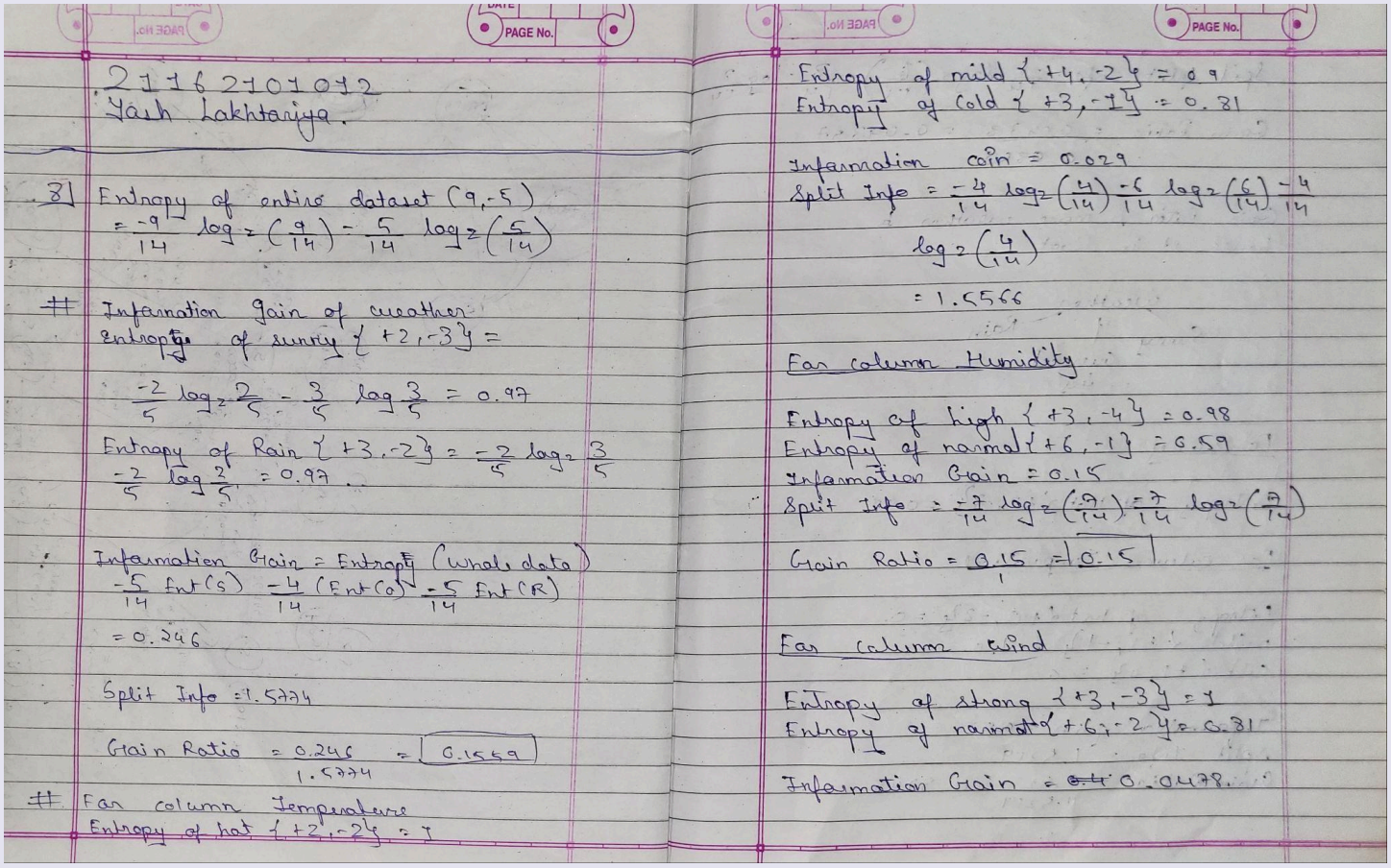
Despite its simplicity and the unrealistic feature independence assumption, naive Bayesian classification often showcases robust performance in real-world scenarios, particularly for text-related tasks like spam detection and document categorization.

Name - Yash Lakhtariya
Enrollment number - 21162101012
Branch - CBA Batch - 61
DMW Assignment 2

8. **Generate the decision tree with height 2 for given below data. Use gain ratio method for attribute splitting. Also classify the given data: X= (Rain, Cool, High, Strong).**

Day	Outlook	Temperature	Humidity	Wind	Play Tennis
Day1	Sunny	Hot	High	Weak	No
Day2	Sunny	Hot	High	Strong	No
Day3	Overcast	Hot	High	Weak	Yes
Day4	Rain	Mild	High	Weak	Yes
Day5	Rain	Cool	Normal	Weak	Yes
Day6	Rain	Cool	Normal	Strong	No
Day7	Overcast	Cool	Normal	Strong	Yes
Day8	Sunny	Mild	High	Weak	No
Day9	Sunny	Cool	Normal	Weak	Yes
Day10	Rain	Mild	Normal	Weak	Yes
Day11	Sunny	Mild	Normal	Strong	Yes
Day12	Overcast	Mild	High	Strong	Yes
Day13	Overcast	Hot	Normal	Weak	Yes
Day14	Rain	Mild	High	Strong	No

Name - Yash Lakhtariya
Enrollment number - 21162101012
Branch - CBA Batch - 61
DMW Assignment 2



21162101012
Yash Lakhtariya.

3] Entropy of entire dataset (9,-5)
$= \frac{-9}{14}\log_2\left(\frac{9}{14}\right) - \frac{5}{14}\log_2\left(\frac{5}{14}\right)$

Information gain of weather
Entropy of sunny {+2,-3} =

$\frac{-2}{5}\log_2\frac{2}{5} - \frac{3}{5}\log\frac{3}{5} = 0.97$

Entropy of Rain {+3,-2} = $\frac{-2}{5}\log_2\frac{3}{5}$
$\frac{-2}{5}\log\frac{2}{5} = 0.97$

Information Gain = Entropy (whole data)
$\frac{-5}{14}$ Ent(S) $\frac{-4}{14}$ (Ent(O) $\frac{-5}{14}$ Ent(R)
$= 0.246$

Split Info = 1.5774

Gain Ratio $= \frac{0.246}{1.5774} =$ 0.1559

For column Temperature
Entropy of hot {+2,-2} = 1
Entropy of mild {+4,-2} = 0.9
Entropy of cold {+3,-1} = 0.81

Information coin = 0.029
Split Info $= \frac{-4}{14}\log_2\left(\frac{4}{14}\right) \frac{-6}{14}\log_2\left(\frac{6}{14}\right) \frac{-4}{14}$
$\log_2\left(\frac{4}{14}\right)$
$= 1.5566$

For column Humidity

Entropy of high {+3,-4} = 0.98
Entropy of normal {+6,-1} = 0.59
Information Gain = 0.15
Split Info $= \frac{-7}{14}\log_2\left(\frac{7}{14}\right) \frac{7}{14}\log_2\left(\frac{7}{14}\right)$

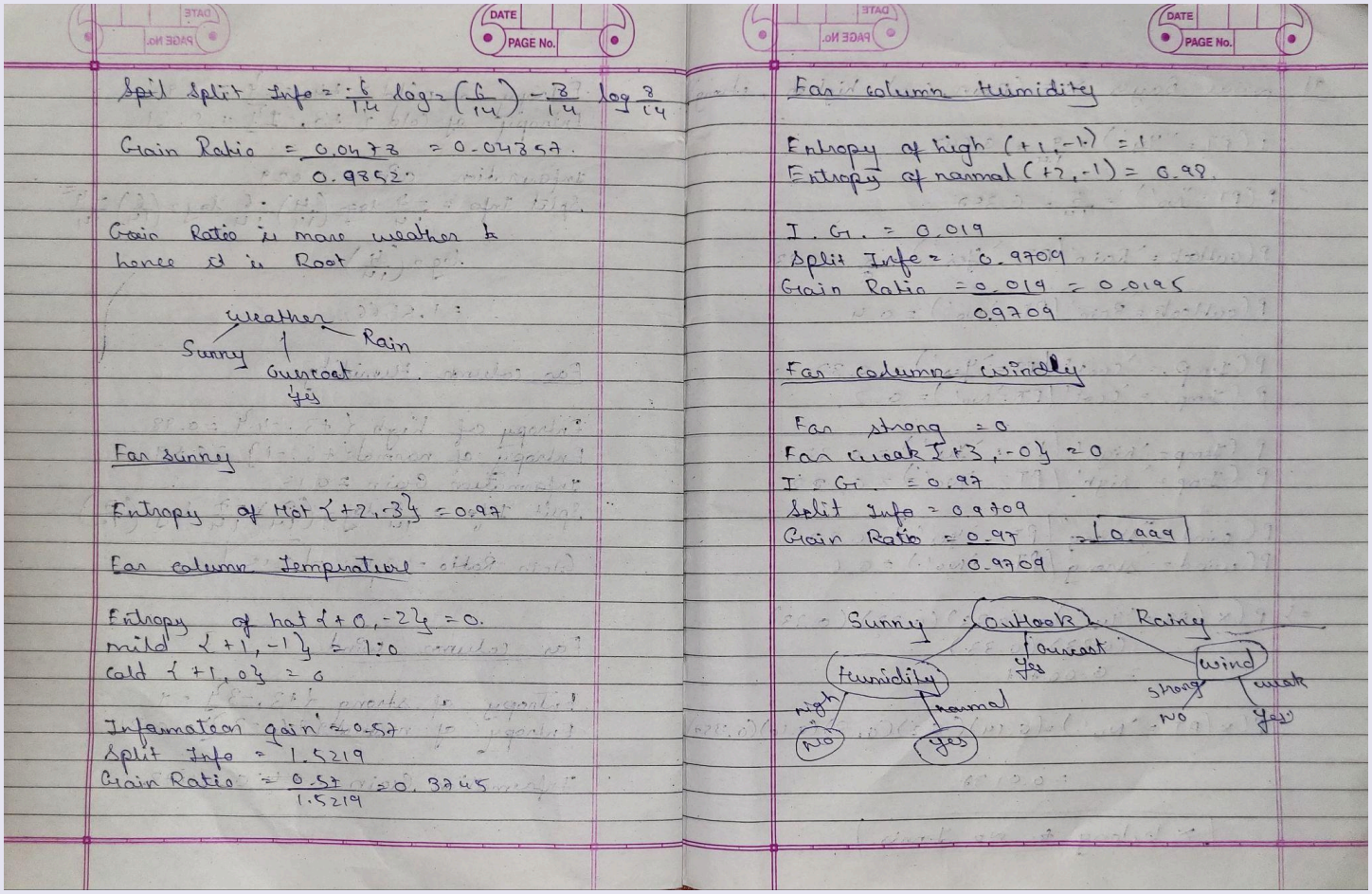
Gain Ratio $= \frac{0.15}{1} =$ 0.15

For column wind

Entropy of strong {+3,-3} = 1
Entropy of normal {+6,-2} = 0.81

Information Gain = ~~0.4~~ 0.0478.

Name - Yash Lakhtariya
Enrollment number - 21162101012
Branch - CBA Batch - 61
DMW Assignment 2



Spil Split Info = $-\frac{6}{14}\log_2\left(\frac{6}{14}\right) - \frac{8}{14}\log\frac{8}{14}$

Gain Ratio = $\frac{0.0473}{0.9852} = 0.048057$.

Gain Ratio is more weather is hence it is Root.

Weather
Sunny | Rain
Overcast
Yes

For sunny

Entropy of Hot {+2,-3} = 0.97

For column Temprature

Entropy of hot {+0,-2} = 0.
mild {+1,-1} = 1.0
cold {+1,0} = 0

Infomation gain = 0.57
Split Info = 1.5219
Gain Ratio = $\frac{0.57}{1.5219} = 0.3745$

For column Humidity

Entropy of high (+1,-1) = 1
Entropy of normal (+2,-1) = 0.98

I.G. = 0.019
Split Info = 0.9709
Gain Ratio = $\frac{0.019}{0.9709} = 0.0195$

For column windy

For strong = 0
For weak {+3,-0} = 0
I.G. = 0.97
Split Info = 0.9709
Gain Ratio = $\frac{0.97}{0.9709} = 0.999$

Name - Yash Lakhtariya
Enrollment number - 21162101012
Branch - CBA Batch - 61
DMW Assignment 2

9. **Classify the given data using naïve bayes classifier with reference to below training data.**

X= (Rain, Cool, High, Strong).

Day	Outlook	Temperature	Humidity	Wind	Play Tennis
Day1	Sunny	Hot	High	Weak	No
Day2	Sunny	Hot	High	Strong	No
Day3	Overcast	Hot	High	Weak	Yes
Day4	Rain	Mild	High	Weak	Yes
Day5	Rain	Cool	Normal	Weak	Yes
Day6	Rain	Cool	Normal	Strong	No
Day7	Overcast	Cool	Normal	Strong	Yes
Day8	Sunny	Mild	High	Weak	No
Day9	Sunny	Cool	Normal	Weak	Yes
Day10	Rain	Mild	Normal	Weak	Yes
Day11	Sunny	Mild	Normal	Strong	Yes
Day12	Overcast	Mild	High	Strong	Yes
Day13	Overcast	Hot	Normal	Weak	Yes
Day14	Rain	Mild	High	Strong	No

Name - Yash Lakhtariya
Enrollment number - 21162101012
Branch - CBA Batch - 61
DMW Assignment 2

9] naive Bayes $x = (Rain, cool, high, strong)$

$P(PT = \text{'yes'}) = \frac{9}{14} = 0.642$

$P(PT = \text{'No'}) = \frac{5}{14} = 0.357$

$P(\text{outlook} = \text{'Rain'} / PT = \text{'yes'}) = \frac{3}{9} = 0.33$

$P(\text{outlook} = \text{'Rain'} / PT = \text{'No'}) = 0.4$

$P(\text{Temp.} = \text{'cool'} / PT = \text{'yes'}) = 0.33$
$P(\text{Temp.} = \text{'cool'} / PT = \text{'No'}) = 0.2$

$P(\text{Temp} = \text{'high'} / PT = \text{'yes'}) = 0.33$
$P(\text{Temp} = \text{'high'} / PT = \text{'No'}) = 0.8$

$P(\text{wind} = \text{'strong'} / PT = \text{'yes'}) = 0.33$
$P(\text{wind} = \text{'strong'} / PT = \text{'No'}) = 0.6$

$\Rightarrow P(x / PT = \text{'yes'}) = (0.33)(0.33)(0.33)(0.33)(0.33)$
$= 0.00761$

$P(x / PT = \text{'No'}) = (0.14)(0.2)(0.8)(0.6)(0.357)$

$= 0.0137$

x belong to No Tennis

Name - Yash Lakhtariya
Enrollment number - 21162101012
Branch - CBA Batch - 61
DMW Assignment 2

10. How is clustering different from classification?

Clustering:

- Clustering is an unsupervised learning technique that groups similar data points together based on their features or characteristics.
- The objective of clustering is to identify inherent patterns or structures in the data without predefined classes or labels.
- Clustering algorithms segment the data into clusters where data points within the same cluster are more alike than those in other clusters.
- Clustering doesn't require labeled data, and the number of clusters may not be predetermined.

Classification:

- Classification is a supervised learning technique used to forecast the class label of input data based on past observations with known class labels.
- The aim of classification is to learn a mapping function from input features to output labels, enabling the algorithm to classify new, unseen instances into predefined classes.
- Classification algorithms are trained on labeled data, where each instance is associated with a class label.
- The outcome of classification is a decision boundary that delineates different classes in the feature space.

In essence, clustering centers on grouping similar data points based on intrinsic properties, whereas classification focuses on predicting the class label of input data using labeled training examples.

Name - Yash Lakhtariya
Enrollment number - 21162101012
Branch - CBA Batch - 61
DMW Assignment 2

11. Algorithm for K-Means & K-Medoids Clustering

K-Means Algorithm:

1. **Initialization**: Randomly select K cluster centroids (points) in the feature space.
2. **Assignment**: Assign each data point to the nearest centroid based on Euclidean distance.
3. **Update Centroids**: Recalculate the centroids of each cluster by computing the mean of all data points assigned to that cluster.
4. **Repeat Steps 2-3**: Iterate the assignment and centroid update steps until convergence, which occurs when the centroids no longer change significantly or a maximum number of iterations is reached.
5. **Output**: The final centroids represent the centers of the clusters, and each data point is assigned to one of the K clusters based on its proximity to the centroids.

K-Medoids Algorithm:

1. **Initialization**: Randomly choose K data points as initial medoids.
2. **Assignment**: Assign each data point to the nearest medoid based on a chosen distance metric (e.g., Euclidean distance).
3. **Update Medoids**: For each cluster, select the data point that minimizes the total distance to other points in the cluster and designate it as the new medoid.
4. **Repeat Steps 2-3**: Iterate the assignment and medoid update steps until convergence, which occurs when the medoids no longer change significantly or a maximum number of iterations is reached.
5. **Output**: The final medoids represent the centers of the clusters, and

Name - Yash Lakhtariya
Enrollment number - 21162101012
Branch - CBA Batch - 61
DMW Assignment 2

each data point is assigned to one of the K clusters based on its proximity to the medoids.

Both K-Means and K-Medoids aim to minimize intra-cluster distance and maximize inter-cluster distance, but they differ in how they define and update the cluster centers (centroids in K-Means, medoids in K-Medoids).