ASSIGNMENT 07:

Question 1 (3 points):

Classifier	Briefly describe how a model is built (Enter "N/A" if the classifier does not build a model)	Briefly describe how the model is applied to a new data instance	"Ideal" Input Feature Type (discrete or continuous)
Naïve Bayes	N/A	Naïve Bayes calculates the probability of a new data instance belonging to each class based on its features. It uses Bayes' theorem assuming independence among features to compute the posterior probability. The instance is assigned to the class with the highest posterior probability.	Discrete
Support Vector Machine	SVM builds a model by finding the optimal hyperplane that best separates the classes in the feature space. It maximizes the margin between classes while minimizing classification errors.	For a new data instance, the model computes which side of the hyperplane the instance falls on. It assigns the instance to the class corresponding to the side of the hyperplane it lies on.	Both
Nearest Neighbor	N/A	Nearest Neighbor classifies a new data instance by comparing it to the labeled instances in the training set. It calculates the distance between the new instance and all other instances in the training set. The instance is assigned the label of the majority	Continuous

		class among its k-nearest neighbors, where k is a predefined parameter.	
Decision Trees	Decision Trees build a model by recursively partitioning the feature space based on feature values. They select the best feature at each node to split the data into homogeneous subsets with respect to the target variable. This process continues until a stopping criterion is met or the tree reaches its maximum depth.	For a new data Decision Tree traverses the tree from the root node to a leaf node, following the decision rules based on feature values. The instance is assigned to the class corresponding to the majority class in the leaf node reached.	Both

Question 2 (1 point): You have built a Naïve Bayes classifier model and it produces the following confusion matrix for a test set with 1000 data instances. Do you consider this model's performance to be acceptable? Why or why not?

		Predicted class	
		Class 1	Class 2
Actual class	Class 1	850	0
	Class 2	150	0

Answer:

From the confusion matrix, we can observe the following:

True Positives (TP): 850 instances are correctly classified as Class 1.
False Negatives (FN): 150 instances of Class 2 are incorrectly classified as Class 1.
True Negatives (TN): There are no instances correctly classified as Class 2.
False Positives (FP): There are no instances of Class 1 incorrectly classified as Class 2.
Now, let's calculate some metrics:

Accuracy= (TP + TN) / (TP + TN + FP + FN)
= (850 + 0) / (850 + 0 + 0 + 150) = 850 / 1000 = 0.85.

Precision= TP / (TP + FP).
= 850 / (850 + 0) = 1.

Recall = TP / (TP + FN).
= 850 / (850 + 150) = 0.85.

Based on these metrics:

The model achieves an accuracy of 85%
Precision is 1, which indicates that when the model predicts Class 1, it's always correct.
However, the recall for Class 2 is 0, meaning the model failed to correctly identify any instances of Class 2.
In conclusion, while the model performs well in predicting Class 1, it completely fails to predict Class 2.
Hence, the model's performance is not acceptable.

Question 3 (2 points): You have built a Decision Tree model with a max depth of 15. The following two confusion matrices have been generated using the model. The first confusion matrix denotes model performance using the training set, while the second confusion matrix is the performance using the test set. Why do you think the model has produced these results?

		Predicted class		
Actual class		High	Medium	Low
	High	750	5	10
	Medium	7	550	12
	Low	9	8	350

		Predicted class		
Actual class		High	Medium	Low
	High	140	150	175
	Medium	87	45	76
	Low	104	101	99

Answer:

Training Set Confusion Matrix:
- The model exhibits high accuracy on the training set, with correct predictions (high counts along the diagonal).
- There are instances of misclassification, particularly for Medium and Low classes.

Test Set Confusion Matrix:
- The model's performance on the test set is inferior compared to the training set, indicated by increased misclassifications across all classes.
- Particularly, the model struggles with accurate predictions for the Medium and Low classes.

Possible Reasons:
1. Overfitting:
   - The model may overfit the training data, capturing noise or outliers specific to the training set, hindering generalization.

2. Model Complexity:
   - The Decision Tree's complexity (max depth of 15) might be excessive for the dataset, leading to overfitting.

3. Insufficient Data
   - The model might lack representation of the dataset, resulting in biased learned patterns.

4. Class Imbalance:
   - The model could favor majority classes due to imbalances, causing poor performance for minority classes.

Question 4 (1 point): The age of patients in a medical data set ranges from 18 years old to 75 years old. There are 1000 patients in the data set. Describe how you would discretize the “Age” feature into 3 separate categories, such that there is a relatively even distribution of patients across the 3 categories.

Answer:
Determine the Range: Identify the minimum and maximum age in the dataset: 18 years old to 75 years old.

Calculate the Range for Each Category: Divide the total age range (75 - 18 = 57) by the number of desired categories (3) to determine the approximate range for each category.

Define Category Boundaries: Establish category boundaries based on the calculated range. Each category should have an approximately equal number of patients.

Discretize Age into Categories: Assign each patient to one of the three categories based on their age falling within the defined boundaries.

Category 1: Young patients
Age range: 18 to 35 (inclusive)

Category 2: Middle-Aged patients
Age range: 36 to 53 (inclusive)

Category 3: Older patients
Age range: 54 to 75 (inclusive)