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	Uses training data to estimate the probabilities of each class and feature given each class. Given the class, it is assumed that features are conditionally independent.	Uses the Bayes theorem to determine the probability of each class for a new instance based on the feature values. identifies the class whose calculated probability is highest.	Discrete: Discrete features are ideal since Naïve Bayes relies on probability distributions. Continuous features need to be discretized.
Support Vector Machine	Transforms data into a high-dimensional space and determines which hyperplane with the maximum margin is most effective at dividing the classes. optimizes a quadratic function while taking constraints into account.	Determines which side of the learned hyperplane the new instance falls on in the high-dimensional space. The side corresponds to the predicted class.	Continuous: Continuous features are ideal since SVMs learn a geometrical separator between classes. Discrete features can be used but may not be as effective.
Nearest Neighbor	N/A (lazy learner - no model is built)	determines the separation between every training instance that has been stored and the new instance. Finds the k closest instances and predicts the majority class among them.	Continuous: Continuous features are ideal to calculate meaningful distances. Discrete features can be used with specialized

			distance functions.
Decision Trees	Recursively splits the training instances into subsets based on feature values, in order to maximize separation between classes. Splitting criteria is based on metrics like information gain or Gini impurity.	To classify a new instance, the decision tree model starts at the root node and traverses down the tree. At each internal node, it checks the corresponding feature value of the instance and follows the appropriate branch based on the learned split conditions. This process continues until a leaf node is reached. The model then predicts the majority class of the training instances that ended up in that leaf during training.	Discrete: Creating distinct splits is best accomplished with discrete features. Continuous feature's function, but they must be discretized according to predetermined thresholds.

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?

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

<your answer goes here>

The model's performance is not acceptable. It predicts all 850 Class 1 instances correctly, but incorrectly labels all 150 Class 2 instances as Class 1. This indicates the model has overfit to Class 1 and has not learned to identify Class 2 at all. The model needs to be improved to reduce this high false negative rate for Class 2. This suggests that the model failed to learn the patterns differentiating Class 2 and has overfit to Class 1. Class 2's high false negative rate raises questions about possible problems such as class imbalance or inadequate model parameters. To more accurately identify both classes, the model needs to be improved.

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?

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

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

<your answer goes here>
The model has overfit to the training data. It achieves very high accuracy on the training set, but performs much worse on the test set, indicating it has not generalized well. The tree has most likely become extremely complex with a maximum depth of 15, and it is splitting very precisely depending on noise in the training set. Because of this, the model is able to "memorize" the training examples and achieve high accuracy on the training set. But much lower accuracy results from these learned patterns not being able to generalize to the unseen test instances. Simplifying the tree can help the model perform better by lowering overfitting. Reducing the maximum depth hyperparameter or utilizing other regularization strategies, such as minimum samples per leaf or pruning, could accomplish this. The model might be able to learn more broadly applicable patterns if additional training data is gathered. Furthermore, features that are noisy or redundant could be eliminated using feature selection or dimensionality reduction techniques.

Question 4 (1 point): The age of patients in a medical data set range 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.

<your answer goes here>
To discretize the "Age" feature into 3 relatively even categories:

1. Sort the ages of the 1000 patients in ascending order.
2. Assign the first 333 patients to category "Young.”
3. Assign the next 333 patients to category "Middle-aged.”

4. Assign the last 334 patients to category "Old.”

This would yield roughly 333 patients per age category, providing an even distribution. The exact age cutoffs between categories would depend on the distribution of ages in the dataset, but this approach should produce a relatively balanced discretization. This discretization approach aims to balance information gain with simplicity, turning the continuous "Age" feature into a categorical one with a meaningful distribution for further analysis. However, if the ages are skewed, with more younger patients, the cutoffs would shift accordingly to maintain an even split. The key is to ensure that each category contains roughly one-third of the total patients, regardless of the actual age ranges covered.