Q1. (8 points)

(1) (3 points)

- C4.5 (gain ratio) vs. ID3 (information gain)
  The information gain measure is biased towards attributes with a large number of values. The gain ratio measure is proposed to overcome this bias, which is normalization to information gain using SplitInfo.

- Cart (gini-index) vs. MDL (minimum description length)
  The gini-index is biased toward multivalued attributes and has difficulty when the number of classes is large. The MDL principle has the least bias toward multivalued attributes.

(2) (2 points)

RainForest maintains an $AVC$-set for each attribute, at each tree node, describing the training tuples at the node. The $AVC$-set of an attribute $A$ at a node $N$ gives the class label counts for each attribute $A$ for the tuples at $N$. The set of all $AVC$-sets at a node $N$ is the $AVC$-group of $N$. The size of an $AVC$-set for an attribute $A$ at a node $N$ depends only on the number of distinct values of $A$ and the number of classes in the set of tuples at $N$. Typically, this size should fit in memory, even for real-world data. Thus, the use of data structures to hold aggregate information regarding the training data is the major factor of improving the scalability of decision tree induction.

(3) (3 points)

Bagging and boosting each generate a set of classification or prediction models. Voting strategies are used to combine the predictions for a given unknown tuple. The bagged classifier often has significantly greater accuracy than a single classifier derived from the original training data. The increased accuracy occurs because the composite model reduces the variance of the individual classifiers. Boosting can further improve accuracy compared with bagging. Unlike bagging, where each classifier was assigned an equal vote, boosting assigns a weight to each classifier’s vote, based on how well the classifier performed. The lower a classifier’s error rate, the more accurate it is, and thus, the higher its weight for voting should be.
Q2. (8 points)

(1) (4 points)

Let $D$ be a training set of tuples and their associated class labels, and each tuple is represented by an $n$-dimensional attribute vector $X = (x_1, x_2, ..., x_n)$. Suppose there are $m$ classes $C_1, C_2, ..., C_m$. Classification is to derive the maximum posteriori, i.e., the maximal $P(C_i | X)$. This can be derived from Bayes’ theorem $P(C_i | X) = P(X | C_i) P(C_i) / P(X)$. Since $P(X)$ is constant for all classes, only $P(C_i | X) = P(X | C_i)$ needs to be maximized.

(2) (4 points)

In a data stream environment, a system cannot store the complete set of data in main memory. As a result, only part of the raw data and some summarized data may be stored. Most of the raw data are processed only once and discarded. Thus, one needs to know the count of the number of records that the value $a_i$ and the class label $v$ occurred together.

A data stream is partitioned into a set of disjoint windows, each of which consists of a portion of the stream. The coming data are continuously used to test the classifier to see whether the classifier is still sufficiently accurate. Once the data in a window is full, the counts for the occurrences of all distinct $a_i \cap v$ are computed. After computing the counts, the raw data of the stream can be discarded. These counts are used to train the classifier, i.e., update the probability distributions.

Reference:
Q3. (8 points)

(1) (4 points)

The training tuples are described by \( n \) attributes. Each tuple represents a point in an \( n \)-dimensional space. In this way, all of the training tuples are stored in an \( n \)-dimensional space. When given an unknown tuple, a \( k \)-nearest neighbor classifier searches the pattern space for the \( k \) training tuples that are closest to the unknown tuple. These \( k \) training tuples are the \( k \)-nearest neighbors of the unknown tuple. Closeness is defined in terms of a distance metric, such as Euclidean distance. For \( k \)-nearest neighbor classification, the unknown tuple is assigned the most common class among its \( k \)-nearest neighbors. When \( k=1 \), the unknown tuple is assigned the class of the training tuple that is closest to it in the pattern space.

(2) (4 points)

A \( k \)-nearest neighbor classifier is more appropriate than a decision tree classifier when the training tuples are updated frequently. For \( k \)-nearest neighbor classification, we only need to store the updated points in the pattern space; in contrast, for decision tree induction, we have to update a classification model.

Compared with a decision tree classifier, a \( k \)-nearest neighbor classifier can be more robust to noisy data by averaging \( k \)-nearest neighbors.

Note: There could be many answers for Q3.2. You can get full marks if your answer is reasonable.
Q4. (8 points)

(1) (3 points)

One challenge of classification using microarray data is that the number of genes is significantly greater than the number of samples. Thus, it is essential to reduce the dimensionality before applying traditional classification techniques. Selection of relevant genes for classification is known as feature selection. Many feature selection techniques (e.g., partial least squares (PLS)) have been reported in the literature.

Cross validation is a well-established technique used to optimize the features chosen in a classifier. In $m$-fold cross-validation, the training set is randomly divided into $m$ disjoint subsets with roughly equal size. Each of these $m$ subsets is left out in turn for evaluation, and the other $(m-1)$ subsets are used as inputs to the classification algorithm. The most popular form of cross-validation is leave-one-out cross-validation.

Reference:

(2) (2 points)

The classification accuracy is often improved using a subset instead of the entire set of genes. This result is quite natural since the dimensionality is significantly reduced due to feature selection. In addition, a small set of relevant genes is convenient for developing diagnostic tests.

(3) (3 points)

- **Decision tree induction**
  This method evaluates an attribute selection measure (e.g., information gain) for every attribute to select the splitting attribute. This computation cost is overwhelming in a high-dimensional space.

- **$k$-nearest neighbor classification**
  This method adopts a distance metric (e.g., Euclidean distance) to represent closeness between tuples. Dimensionality curse occurs in a high-dimensional space. The distance between neighbors could be dominated by irrelevant attributes.

**Note:** There could be many answers for Q4. You can get full marks if your answer is reasonable.