**Batch: H3-3 Roll No.: 16014022050**

**Experiment No.: 6**

|  |
| --- |
| **Title: Classification using decision tree algorithm** |

**Aim:** To build a decision tree classifier with C4.5 algorithm using R libraries

**Expected Outcome of Experiment:**

**CO3:** Understand the basic concept and techniques of Machine Learning regression and classification

**Books/ Journals/ Websites referred:**

1. Data Mining Concepts and Techniques Jiawei Han, Michelin Kamber, Jian Pie, 3rd edition

### Step 1: Install and load the packages

### Step 2: Create cross validation folds

### Step 3: Create the model

### Step 4: Interpret the model results

### Step 5: Visualize the tree

### Step 6: Interpret the classification rules from the tree

### Step 7: Predict the class for some random user input unseen data

**Procedure for Implementation in lab:**

1. Select a dataset suitable for classification from UCI data repository or Kaggle. Example:

Titanic dataset (<https://www.kaggle.com/datasets/yasserh/titanic-dataset>) , where “Survived” can be used as a class label.

1. **Students should provide the following details of the dataset:**
   1. Title:
   2. Source:
   3. Number of instances:
   4. Number of attributes:
   5. Attribute information:
2. Handle the missing values appropriately
3. Create the decision tree, interpret the tree to extract all the rules and perform prediction on unseen data.

**Students should copy their R code and screenshots of output stepwise and paste them here.**

**Implementation:**

Title: Red Wine Quality

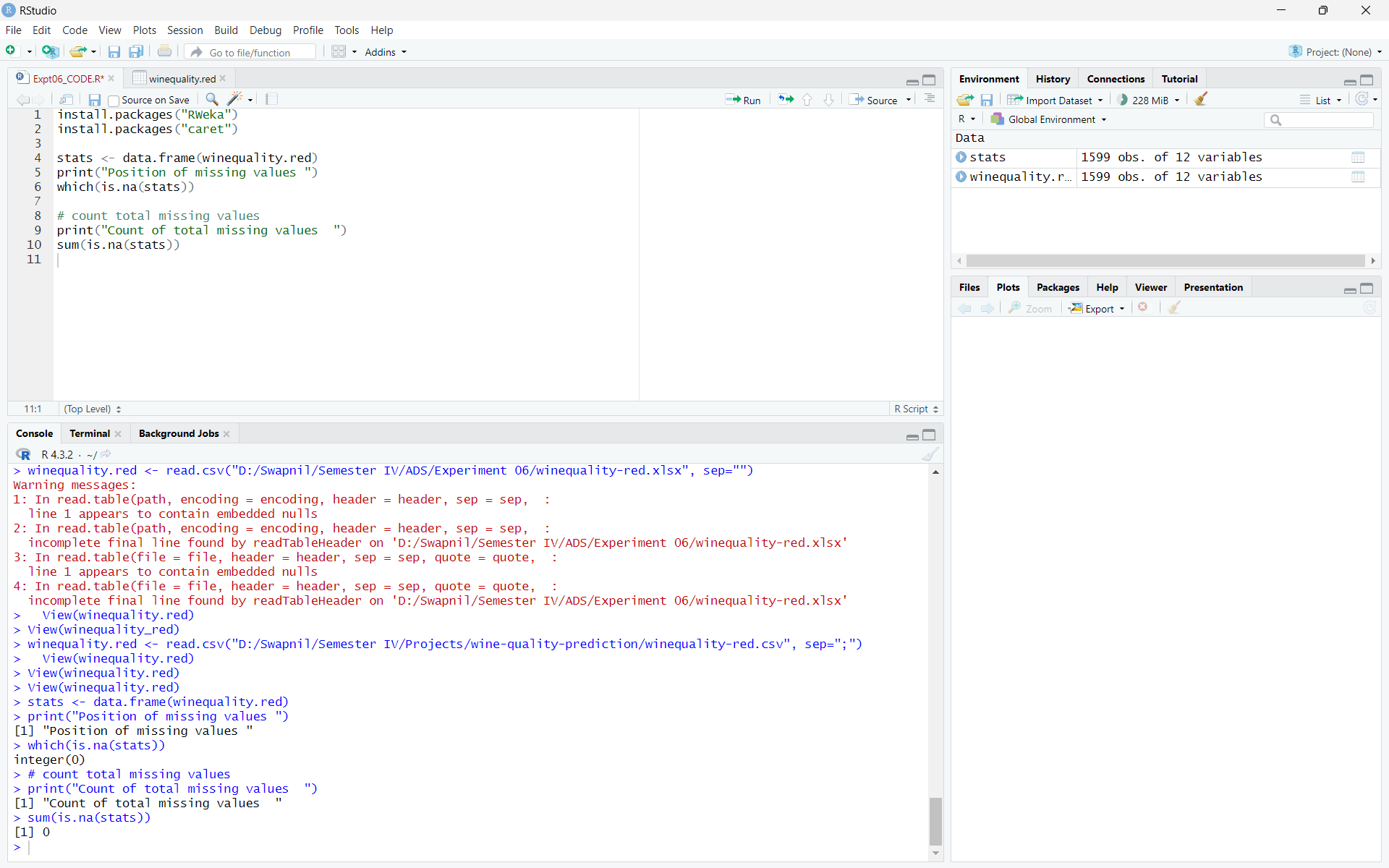
Source: Kaggle

Number of instances: 1599

Number of attributes: 12

Attribute information:

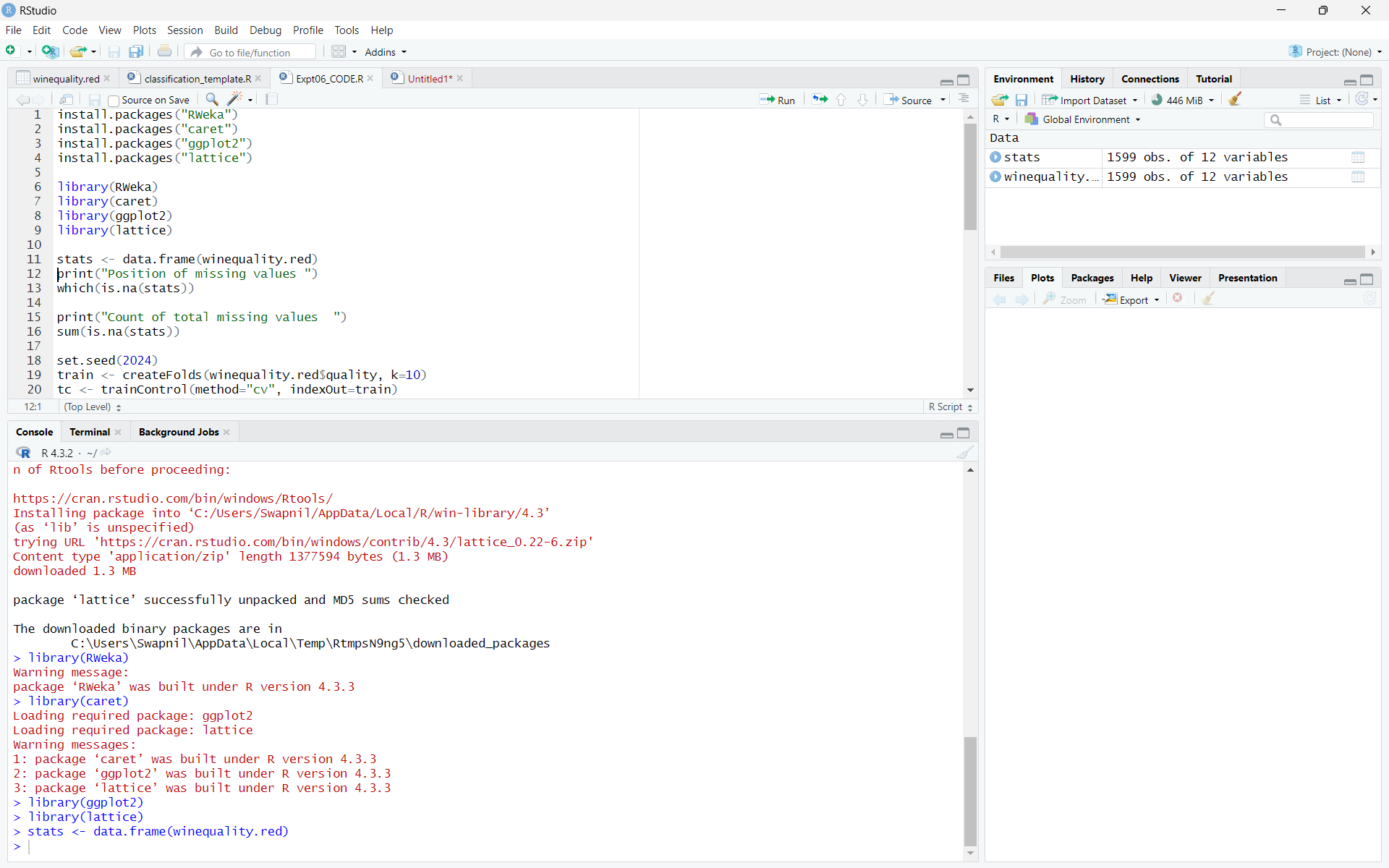
Handling missing data:



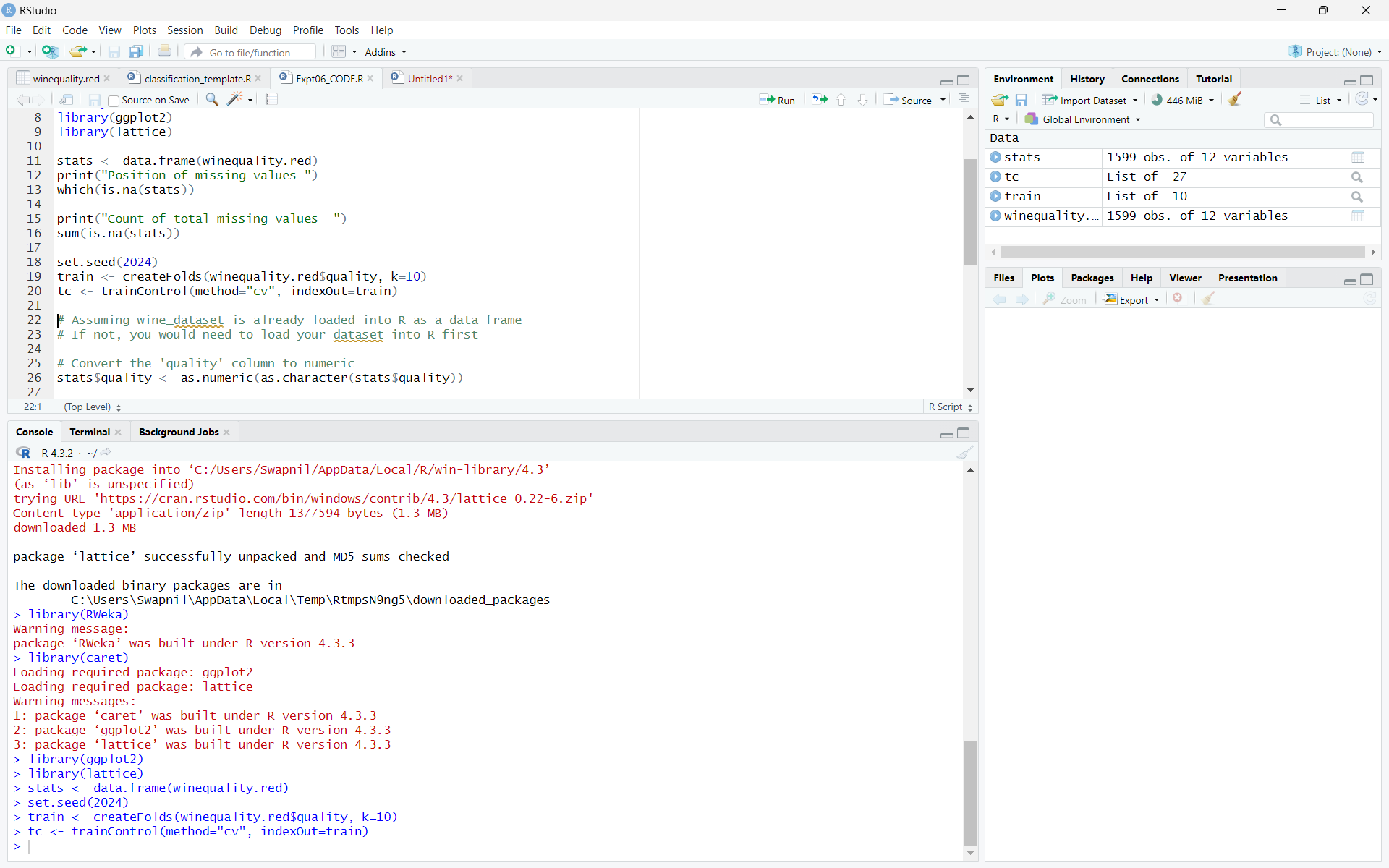
**No missing data was found in the chosen dataset.**

Training the model:

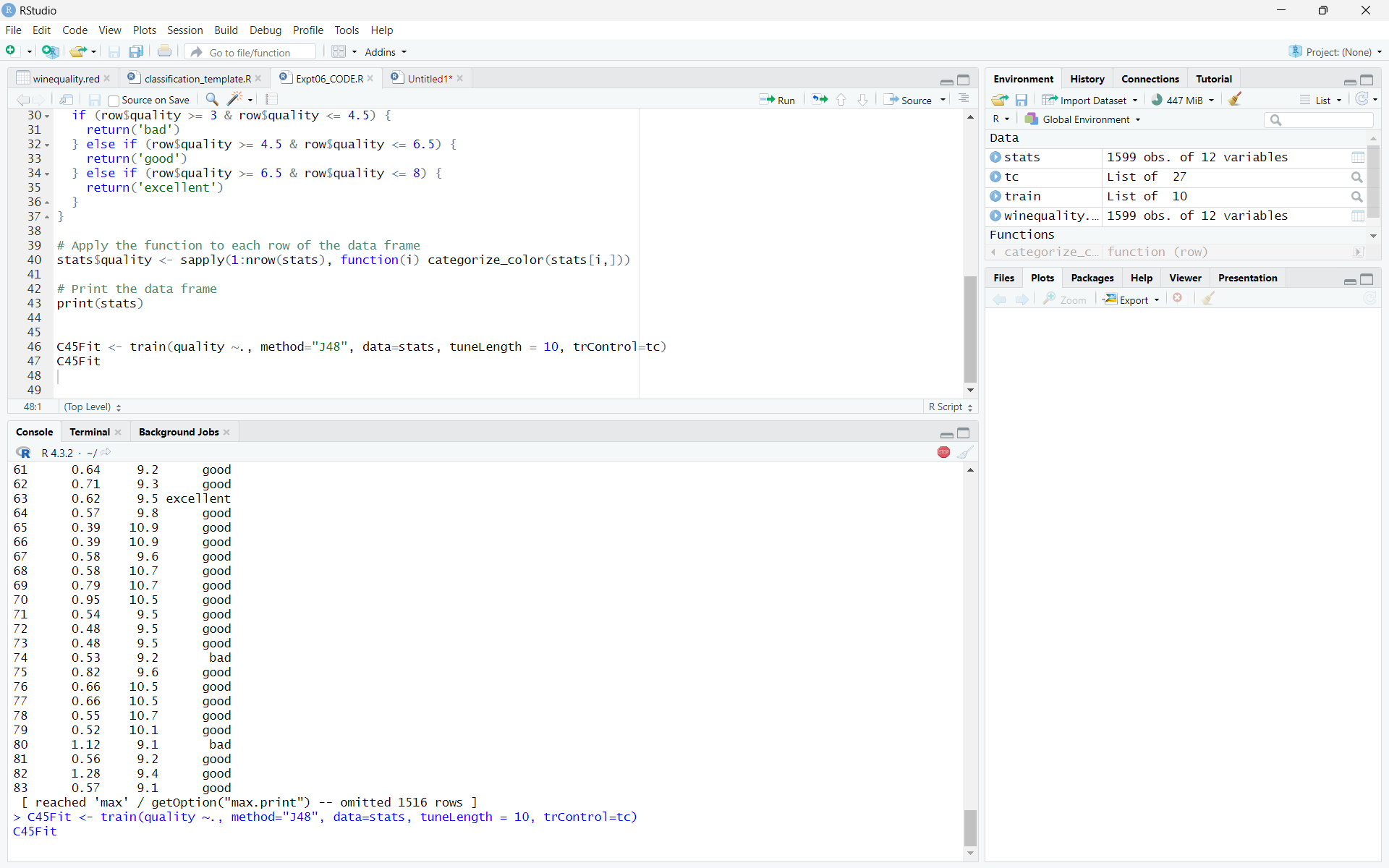
**Step 1: Install and Load the packages**

****

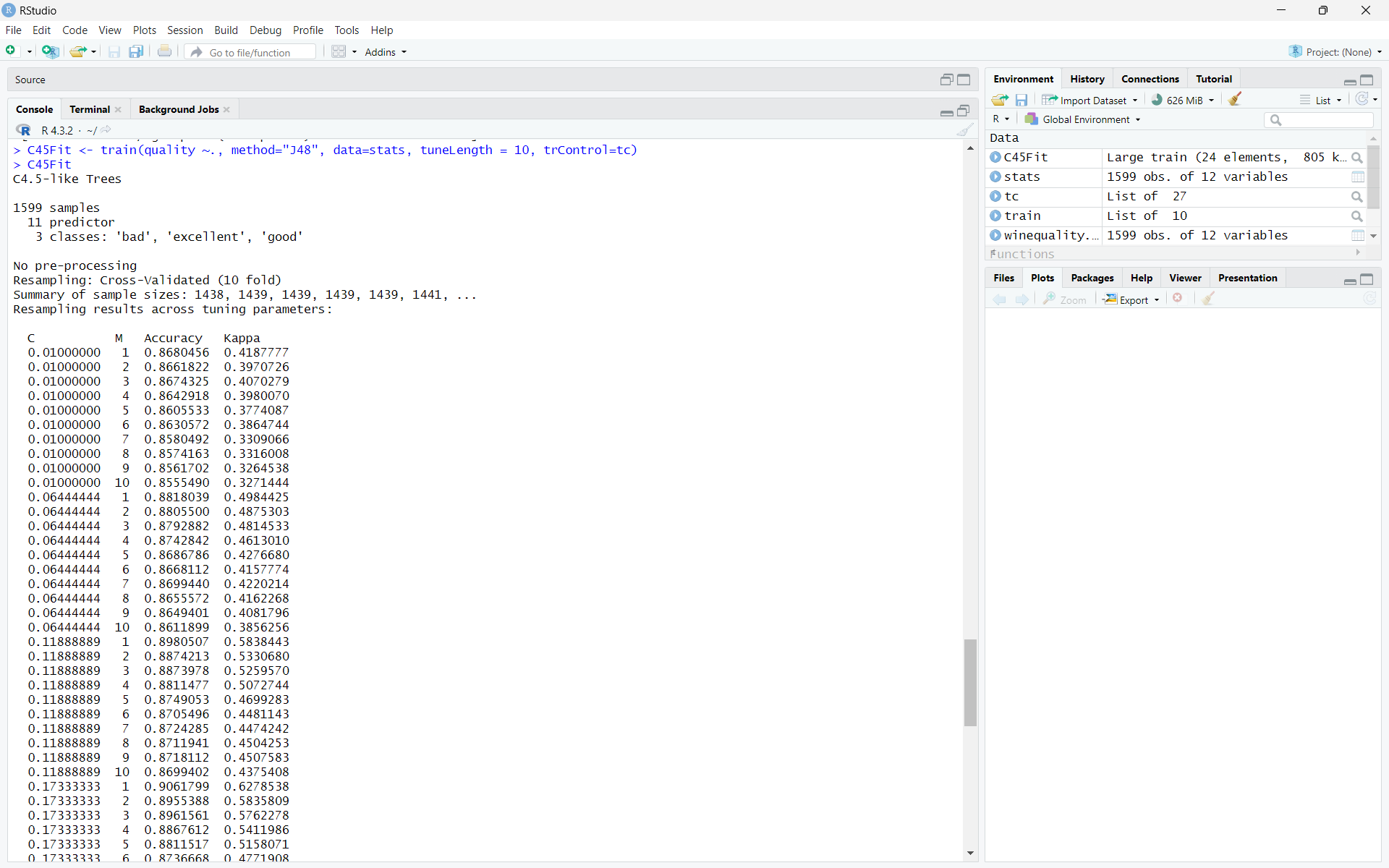
**Step 2: Create cross validation folds**

****

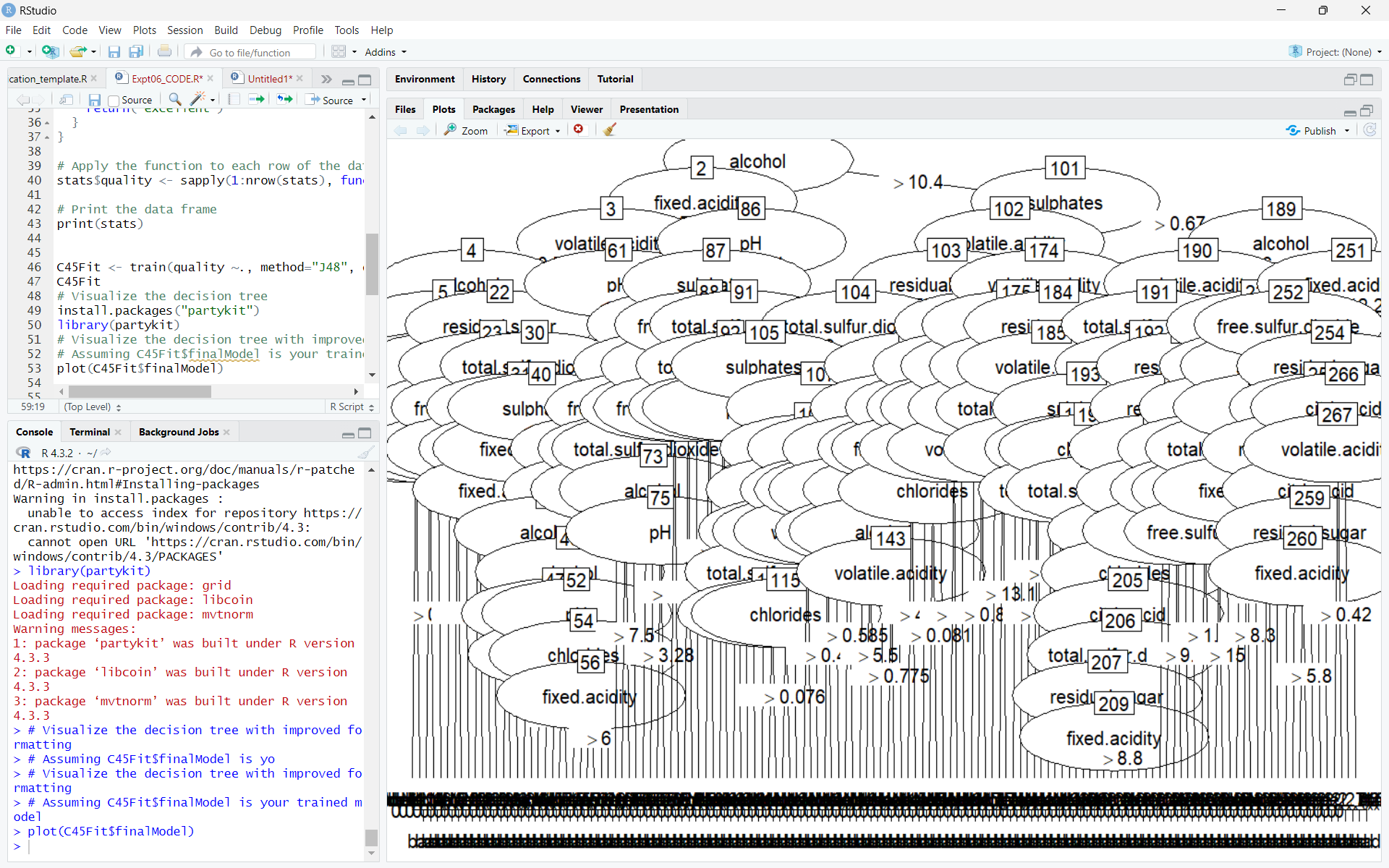
**Step 3: Create the model**

****

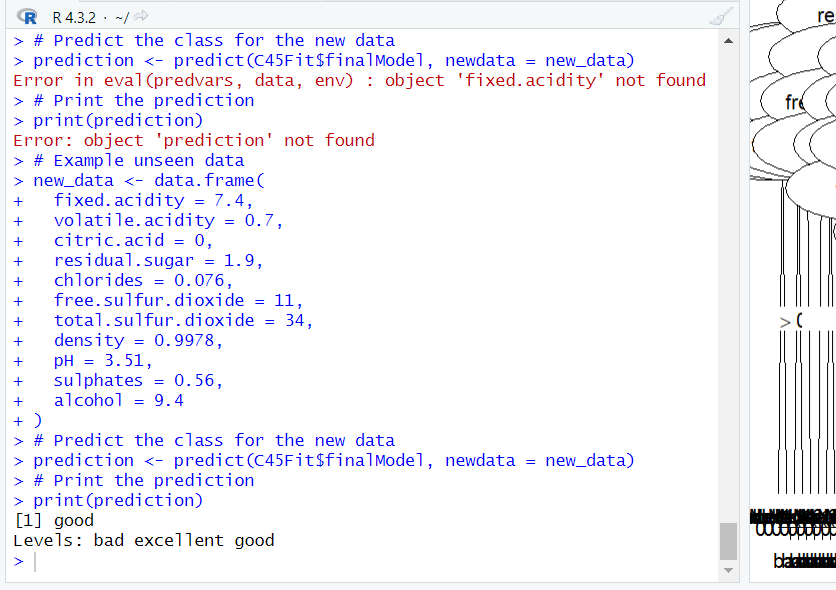
**Step 4: Interpret the model results**

****

**Step 5: Visualize the tree**

****

**Step 6: Predict the class for some random user input unseen data**



**Conclusion:**

In this experiment we built a decision tree classifier with C4.5 algorithm using R libraries which helped us understand the basic concept and techniques of Machine Learning regression and classification.

**Post lab Questions:**

**Q1. Explain in detail with mathematical formulae wherever necessary, the following decision tree algorithms with emphasis on the criteria they use, the type of data that they’re most suitable for and their pros and cons:**

* **CART**
* **C4.5**
* **C5.0**
* **CHAID**
* **ID3**

Criteria Used: **CART** uses Gini impurity for classification and variance for regression to decide the best split. Gini impurity is calculated using the formula:

**[ Gini(D) = 1 - pi2 ]**

where (pi) is the probability of an element being in class (i).

Suitable Data: CART is versatile and can handle both categorical and numerical data. It is particularly effective for classification tasks but can also be used for regression by minimizing the variance of the target variable.

Pros:

1. Can handle both classification and regression tasks.
2. Efficient for large datasets.
3. Prune the tree to avoid overfitting.

Cons:

1. Can be sensitive to small changes in the data.
2. May not perform well with continuous data if not properly discretized.

Criteria Used: **C4.5** uses information gain and gain ratio to decide the best split. Information gain is calculated as:

Type of Data: C4.5 works with both discrete and continuous data.

Pros:

1. Can handle both discrete and continuous data.
2. Handles missing values.
3. Can be used for both classification and regression tasks.

Cons:

1. Can be computationally expensive.
2. Prone to overfitting.

Criteria Used: **C5.0** uses a combination of information gain and gain ratio, similar to C4.5, but with improvements in efficiency and accuracy.

Type of Data: C5.0 works with both discrete and continuous data.

Pros:

1. Faster and more memory-efficient than C4.5.
2. Generates smaller rule sets.
3. More accurate than C4.5.

Cons:

1. Proprietary software, not freely available.
2. Can be computationally expensive for large datasets.

Criteria Used: **CHAID** uses Chi-square tests to decide the best split. It creates all possible cross-tabulations for each categorical predictor until the best outcome is achieved.

Type of Data: CHAID is suitable for datasets with lots of categorical variables. It can handle nominal, ordinal, and continuous data, where continuous predictors are split into categories with approximately equal numbers of observations.

Pros:

1. Effective for datasets with many categorical variables.
2. Prevents overfitting by only splitting a node if a significance criterion is fulfilled.

Cons:

1. Can be computationally expensive for large datasets.
2. Requires careful handling of continuous variables.

Criteria Used: **ID3** uses information gain to decide the best split. It is a greedy algorithm that grows the tree to its maximum size and then prunes it to improve generalization.

Type of Data: ID3 works with discrete or nominal data.

Pros:

1. Simple and easy to understand.
2. Efficient for datasets with discrete attributes.

Cons:

1. Can be sensitive to small changes in the data.
2. Prone to overfitting without pruning.

Q2. **Apply the C4.5 decision tree algorithm on the following dataset and build the decision tree. Extract all the classification rules from the resulting tree. Students are supposed to solve this in their notebook by hand and then upload the scan/photograph of the same here.**