

1. **Library Loading:** The script begins by loading the necessary libraries, including `caret`, `rpart`, and `rpart.plot`. These libraries are essential for data manipulation, decision tree construction, and visualization.
2. **Working Directory Setting:** The working directory is set to the location where the data file (`oulad-assessments.csv`) is stored, ensuring that R can locate and read the data file accurately.
3. **Data Reading:** The data from the CSV file (`oulad-assessments.csv`) is read into R and stored in the data variable.
4. **Missing Values Handling:** The script handles missing values in the dataset using the `na.omit()` function, which eliminates any rows with missing values. Depending on the dataset's nature and the extent of missingness, alternative methods for handling missing data could be considered.
5. **Target Variable Conversion:** The target variable score is converted into a factor with appropriate levels. This step is crucial for classification tasks as it ensures that R treats the target variable as a categorical variable.
6. **Data Splitting:** The dataset is divided into training and testing sets using the `createDataPartition()` function from the `caret` package. The training set (`training_set`) receives 80% of the data, and the remaining 20% goes to the testing set (`testing_set`).
7. **Decision Tree Model Training:** A decision tree model is trained using the `rpart()` function. The model predicts the score based on all other variables in the dataset.
8. **Predictions Making:** The trained decision tree model is used to make predictions on the testing set. The prediction type (`type = "class"`) indicates that the model predicts class labels (levels of score).
9. **Model Performance Evaluation:** The model's performance is evaluated using a confusion matrix generated by the `confusionMatrix()` function from the `caret` package. This matrix provides insights into the model's prediction accuracy compared to the actual values.
10. **Decision Tree Visualization:** The decision tree model is visualized using the `rpart.plot()` function. This function generates a graphical representation of the tree, with various parameters adjusted to customize the plot's appearance, including title, color scheme, branch style, and leaf distribution.

## Reflection

The journey through the R script walkthrough offered an intriguing exploration into predictive modeling, particularly in the context of predicting student assessment scores. Beginning with essential tasks like loading libraries and handling missing data, it underscored the critical role of data preprocessing in machine learning. Transforming the target variable into a factor and splitting the data for validation showcased the importance of thoughtful data preparation. The iterative process of model training and evaluation, culminating in the visualization of the decision tree, provided a comprehensive understanding of the model's construction and interpretability. This experience not only deepened my grasp of predictive modeling steps but also highlighted its tangible impact on real-world scenarios, reinforcing the significance of each meticulous step in model building.