Interpretation Analysis:

The code implements a binary classification task that uses a decision tree model to predict pass/fail outcomes based on student data.

A brief overview of each step is as follows:

Libraries and data preparation: Important libraries (such as caret, rpart, and rpart.plot) are used for data manipulation, decision tree building, and visualization.

will be loaded for.

The data is read from her CSV file named "oulad-assessments.csv" and processed to handle missing values.

Partitioning the data: The dataset is split into a training set (80%) and a test set (20%) using createDataPartition to ensure model training and evaluation of different data subsets.

Model training: The decision tree model (rpart) is trained using the expression ``score ~

This means predicting the target variable (probability of pass/fail) based on all available features.

Make a prediction: The trained model predicts the outcome of the unseen test data and creates predictions that are stored in the Predictions variable.

Transformation of predictor and target variables: Both the predicted and actual results of the test set are transformed into coefficients (0 or 1) assuming a binary outcome (pass/fail).

For accurate evaluation, the predicted result level is adjusted to match the actual result level.

Ratings: Confusion matrices are generated by confusionMatrix and provide detailed performance metrics including true positives, false positives, true negatives, and false negatives.

Visualization: This code includes rpart.plot(model) which visualizes the decision tree structure and is useful for understanding the logic of the model for prediction.

This comprehensive workflow shows how to build, evaluate, and interpret decision tree models for binary classification tasks.

Interpretation of decision tree:

**Root Node:**

The tree starts by splitting data based on the feature "code\_module," indicating the importance of the type of course module in predicting student scores.

**Branches from Root:**

Left Branch (code\_module = AAA, BBB, DDD, GGG):

Further split based on "assessment\_type."

If assessment\_type is "Exam" or "TMA," the model predicts "pass," suggesting these assessment types correlate with student success.

If not, the model predicts "fail," implying alternative assessment types may not predict success effectively.

Right Branch (code\_module = CCC, DDD, FFF):

Split based on "final\_result."

If final\_result > 76, the model predicts "pass," indicating high final results are predictive of passing.

If final\_result <= 76, split again based on "assessment\_type."

If assessment\_type is "Exam," the model predicts "pass," suggesting exams are predictive even with lower final results.

If not, the model predicts "fail," implying alternative assessment types may not be as indicative of success.

The decision tree highlights the importance of code module type (code module) and assessment type (assessment type) in predicting student scores.

Students taking Exams or TMAs in code modules AAA, BBB, DDD, and GGG seem to have better chances of passing.

A high final­\_result is a strong predictor of passing for code modules CCC, DDD, and FFF.

Exams might be a better indicator of passing than other assessment types for students with lower final\_results in code modules CCC, DDD, and FFF..