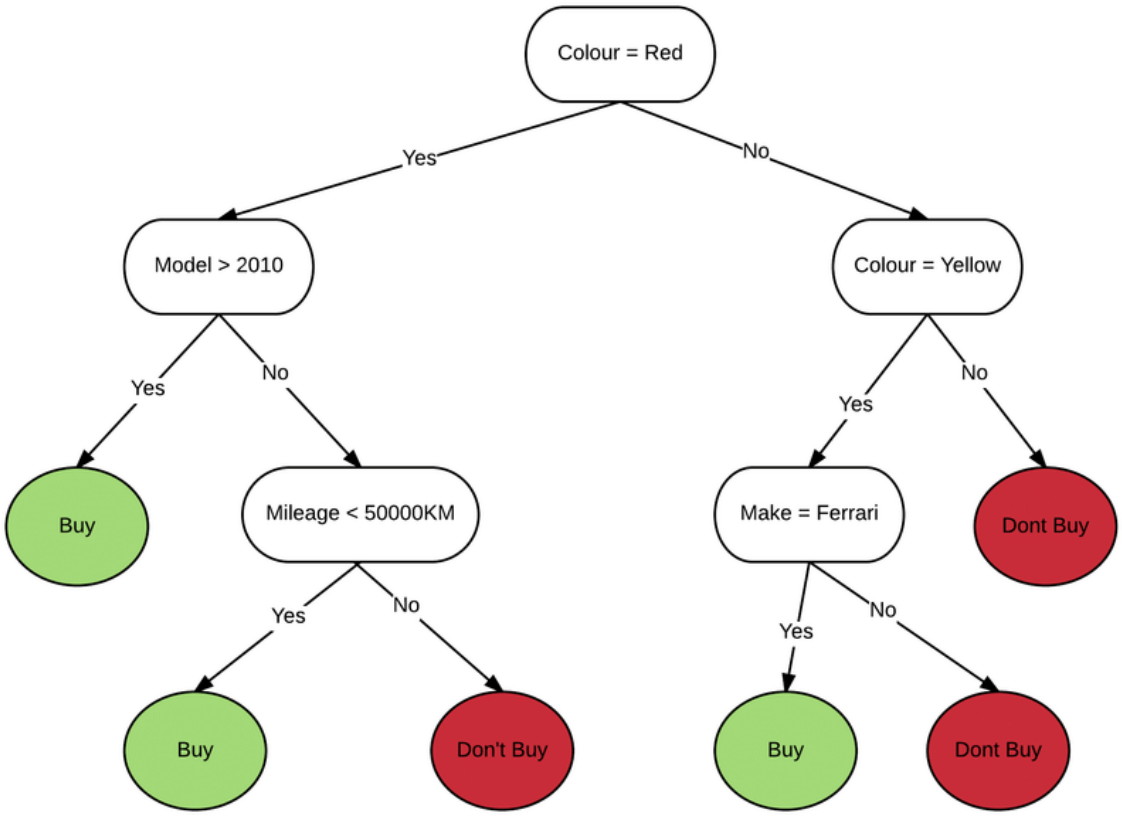
The **ID3 (Iterative Dichotomiser 3)** algorithm is a decision tree learning algorithm used in Machine Learning to create a classification model. It uses the concept of **Information Gain** to select the attribute that best splits the dataset into distinct classes.



Key Concepts

1. **Entropy**: A measure of impurity in the dataset.
2. **Information Gain**: Determines the best attribute to split the dataset.

   Information Gain=Entropy(Parent)−Weighted Entropy(Child)

3. **Decision Tree Construction**:
   - Choose the root node using the highest information gain.
   - Split the dataset based on the chosen attribute.
   - Repeat for each subset until no further split is possible.

Problem Description

The **ID3 Algorithm** uses entropy and information gain to construct a decision tree. This tree helps classify data points by finding the attribute that provides the best split. For example:

- **Dataset**:

Weather	Temperature	Play?
Sunny	Hot	No
Overcast	Hot	Yes
Rainy	Mild	Yes

- **Goal**: Predict the outcome (e.g., *Play?*) based on input conditions using a decision tree.

Steps

1. **Calculate Entropy**: For the dataset and individual attributes.
2. **Calculate Information Gain**: For each attribute.
3. **Build Tree**: Recursively choose the attribute with the highest gain and create branches.
4. **Classify Data**: Use the tree to classify new data.

Code Template

```
# Calculate entropy of a dataset
def calculate_entropy(data, target_col):
    # Compute the entropy of the target column
    pass

# Calculate information gain
def calculate_information_gain(data, attribute, target_col):
    # Find the information gain for a given attribute
    pass

# Build the decision tree
def build_tree(data, attributes, target_col):
    # Recursive function to construct the tree
    pass

# Predict the class for a given data point
def predict(tree, data_point):
    # Traverse the tree to predict the class
    pass
```

Task Description

Students are expected to:

1. Complete the logic for all functions in the template.
2. Test with different datasets.
3. Analyze the decision tree structure for various splits.
4. Experiment with new test data points and verify predictions.