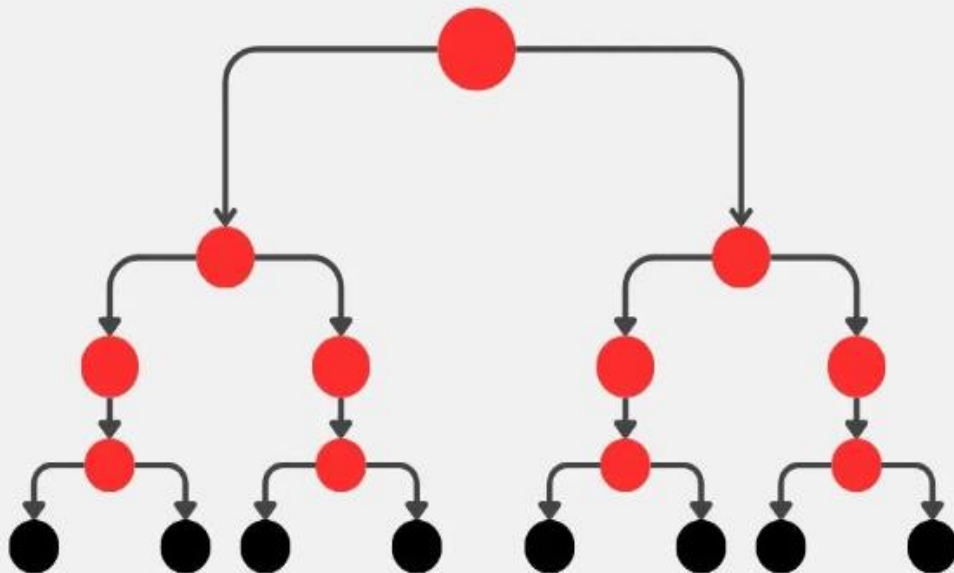# Decision
Trees
## Explained with Code

What is a Decision Tree

Decision Tree is a Supervised learning technique that can be used for both classification and Regression problems.

It is a tree-structured classifier, where, branches represent the decision rules, **Decision nodes** are used to make any decision and have multiple branches, whereas **Leaf nodes** are the output of those decisions and do not contain any further branches.
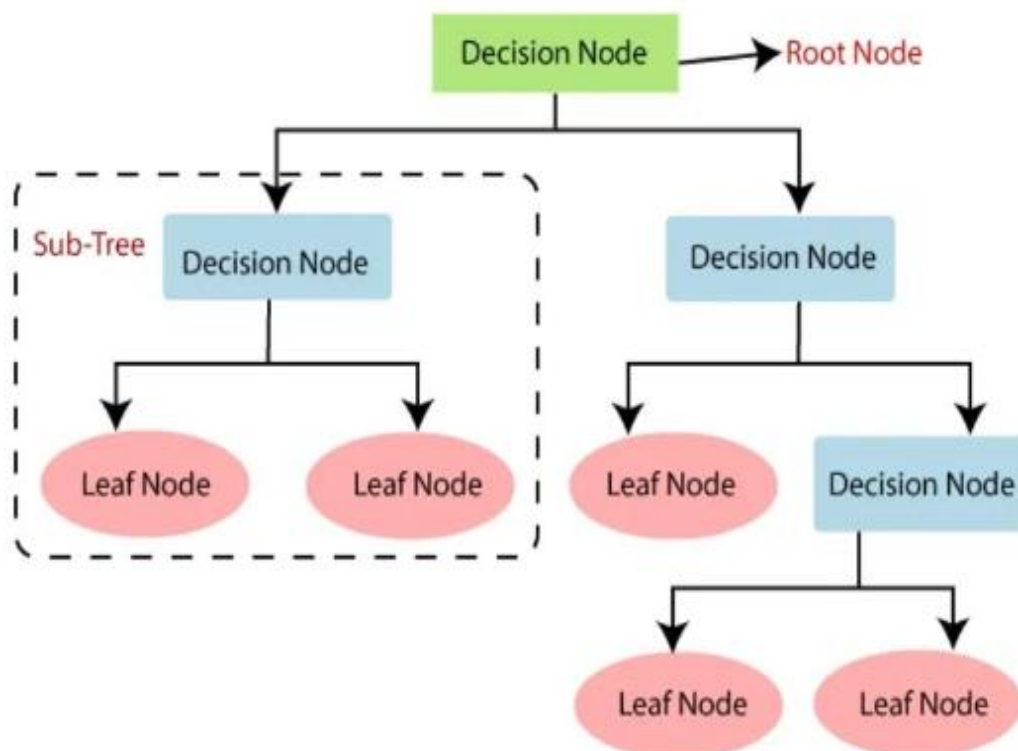
How its made
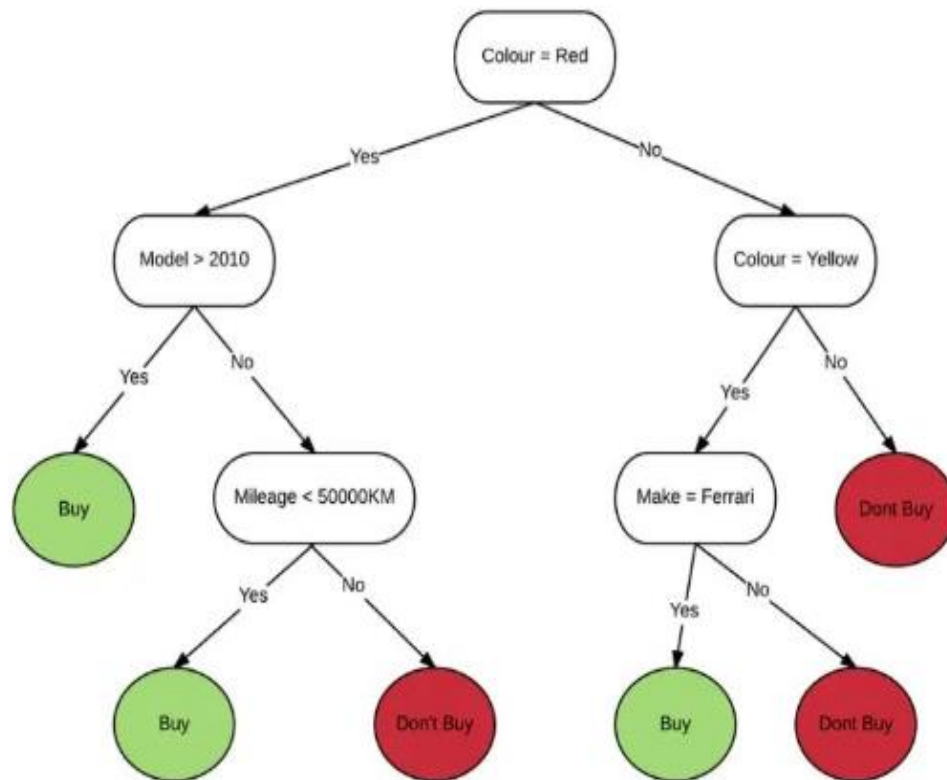
Starting Point:

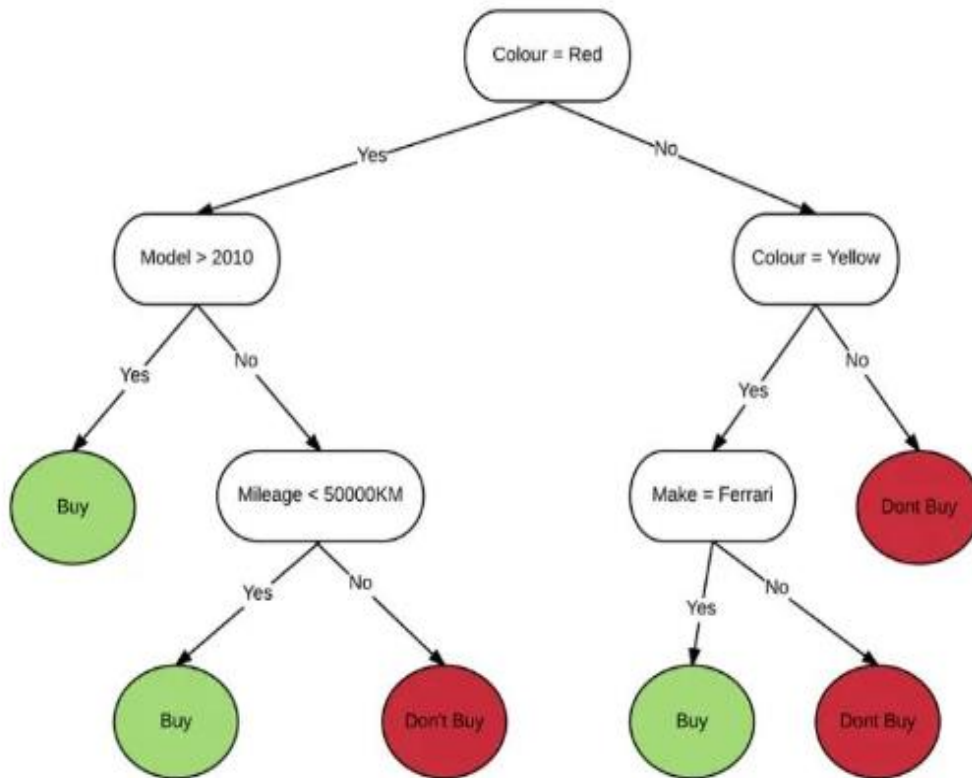- Begin with a question about the data at the top (root) of the tree.

Decision Nodes:

- Based on the answer (Yes/No) to the question, create branches to possible outcomes.

Repeat:

- For each outcome, ask a new question and create more branches.

Leaf Nodes:

- Continue this process until you reach final outcomes (leaf nodes) that provide a solution or decision.

Algorithm:

- Use an algorithm, like CART, to determine the best questions and order for splitting the data.

Code to Make and See Decision Tree

```python
# Import necessary libraries
from sklearn import datasets
from sklearn.model_selection import train_test_split
from sklearn.tree import DecisionTreeClassifier, export_text,
export_graphviz
from sklearn import tree
import graphviz

# Load a sample dataset (for example, the Iris dataset)
iris = datasets.load_iris()
X_train, X_test, y_train, y_test = train_test_split(iris.data,
iris.target, random_state=42)

# Create a Decision Tree model
dt_model = DecisionTreeClassifier()
dt_model.fit(X_train, y_train)

# Visualize the Decision Tree as text
tree_rules = export_text(dt_model, feature_names=iris.feature_names)
print("Decision Tree Rules:\n", tree_rules)

# Visualize the Decision Tree as a graph
dot_data = tree.export_graphviz(dt_model, out_file=None,
feature_names=iris.feature_names, class_names=iris.target_names,
filled=True, rounded=True, special_characters=True)
graph = graphviz.Source(dot_data)
graph.render("iris_decision_tree")  # This will save a file named
"iris_decision_tree.pdf"

# Display the Decision Tree graph (requires Graphviz)
graph.view("iris_decision_tree")
```

-----**THANKYOU**-----