Python Programming and Basic Data Science

Shuvro Pal
Data Scientist, Markopolo.ai
AI Team Lead, Zantrik
ML Facilitator, Crowdsource by Google

Decision Tree

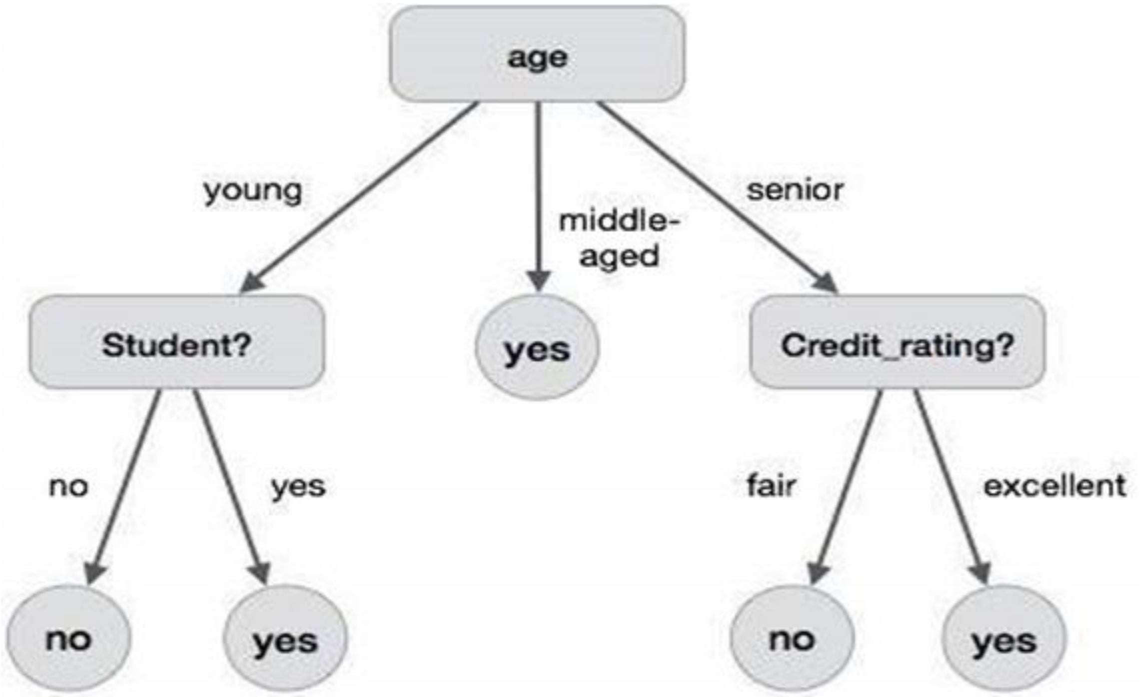
What is it?

A decision tree is like a flowchart that helps make decisions. It's a tree-like structure where:

- Nodes represent decisions or tests on a feature.
- Branches represent the outcome of the decision/test.
- Leaves represent the final decision or classification

Decision Tree

Textbook example?



Decision Tree

Use cases?

- Brainstorming Outcomes
- Presentation of Information
- Automatic Prioritization

Ahh..not quite getting it. More real world samples please..?

- Loan approval in banks
- Disease diagnosis in healthcare
- Spam detection in emails

Decision Tree

How it actually works?

Step-by-Step:

- Start at the root node (the first decision).
- Move down the tree by answering questions at each node.
- When you reach a leaf node, you get a final answer.

Graphical Representation:

```
          Is it raining?
              /   \
            Yes     No
           /          \
   Stay Home        Go Out
```

Decision Tree

Some math please?

Entropy (Uncertainty):

- Entropy measures uncertainty or impurity in data.
- Formula:

$$H(S) = -\sum p_i \log_2(p_i)$$

- Where *pi* is the proportion of each class.

Example:

- If you have 10 weather days, 6 rainy and 4 sunny, the entropy is..?

Decision Tree

One attribute

$$E(S) = \sum_{i=1}^{c} -p_i \log_2 p_i$$

Play Golf	
Yes	No
9	5

Entropy(PlayGolf) = Entropy (5,9)

= Entropy (0.36, 0.64)

= - (0.36 $\log_2$ 0.36) - (0.64 $\log_2$ 0.64)

= 0.94

Decision Tree

Multiple attributes

$$E(T,X) = \sum_{c \in X} P(c)E(c)$$

		Play Golf		
		Yes	No	
Outlook	Sunny	3	2	5
	Overcast	4	0	4
	Rainy	2	3	5
				14

E(PlayGolf, Outlook) = **P**(Sunny)*$\mathbf{E}$(3,2) + **P**(Overcast)*$\mathbf{E}$(4,0) + **P**(Rainy)*$\mathbf{E}$(2,3)

= (5/14)*0.971 + (4/14)*0.0 + (5/14)*0.971

= 0.693

Decision Tree

Pick the best split..

Information Gain (Choosing the Best Split):

- Information Gain tells us how much a feature reduces uncertainty.

		Play Golf	
		Yes	No
Outlook	Sunny	3	2
	Overcast	4	0
	Rainy	2	3
Gain = 0.247			

		Play Golf	
		Yes	No
Temp.	Hot	2	2
	Mild	4	2
	Cool	3	1
Gain = 0.029			

$$Gain(T,X) = Entropy(T) - Entropy(T,X)$$

		Play Golf	
		Yes	No
Humidity	High	3	4
	Normal	6	1
Gain = 0.152			

		Play Golf	
		Yes	No
Windy	False	6	2
	True	3	3
Gain = 0.048			

$$\mathbf{G}(\text{PlayGolf, Outlook}) = \mathbf{E}(\text{PlayGolf}) - \mathbf{E}(\text{PlayGolf, Outlook})$$

$$= 0.940 - 0.693 = 0.247$$

Decision Tree

Practical example?

Imagine you want to predict if a person will play tennis based on the weather:

Weather	Outlook	Temperature	Play Tennis
Sunny	High	Hot	No
Sunny	Low	Hot	Yes
Rainy	High	Cold	Yes
Rainy	Low	Mild	Yes
Overcast	High	Hot	Yes

First Decision (Outlook):

- Split the data on the feature with the highest information gain (e.g., outlook).
- Create a tree based on which feature helps the most in predicting "Play Tennis."

Decision Tree

Practical example..

Resulting Decision Tree:

- After calculations, you may get a tree like:

```
          Outlook?
        /    |    \
  Sunny   Rainy   Overcast
    /          \       \
   No      Yes      Yes
```

Decision Tree

Why it is powerful?

- **Easy to Understand:** Just like asking questions to make decisions in real life.
- **No Need for Data Scaling:** Unlike algorithms like linear regression, decision trees don’t require data normalization.
- **Flexible:** Works for both classification (yes/no decisions) and regression (numerical predictions).

Decision Tree

Fun Fact!

Decision trees are part of some of the most powerful models (like Random Forests)

Random Forest

What is it?

- A random forest is like a team of decision trees working together to make better predictions. Instead of relying on just one tree, we create multiple decision trees and combine their predictions.
- **Key idea:** A forest is more reliable than a single tree!

Random Forest

Use cases?

- Customer segmentation
- Fraud detection in banking
- Image classification

Random Forest

How does it work?

Step-by-Step:

- **Create multiple Decision Trees:** Each tree is trained on a slightly different random sample of the data.
- **Voting:** For classification tasks, each tree "votes" for a class, and the majority class wins.
- **Averaging:** For regression tasks, the average of all trees' outputs is used.

Random Forest

How does it work?

Visual:

- Imagine each tree asks slightly different questions, and the majority of trees give the final answer.
- Show a diagram of multiple trees with arrows pointing to a final decision node.

Tree 1   Tree 2   Tree 3   ...   Tree N

| | | |

Yes/No   Yes/No   Yes/No   Yes/No

\ | / |

Final Decision (Voting/Average)

Random Forest

Relation with DTs?

Random Forest = Multiple Decision Trees + Randomness

- A single decision tree can sometimes overfit (learn too much from noise in the data), but random forests reduce overfitting by:
  - Training each tree on a random subset of the data.
  - Randomly selecting a subset of features to split at each node.
- Boosting Decision Trees: By building multiple trees, a random forest makes a more reliable and robust model than using a single decision tree.

Random Forest

Bring the maths..

Why Multiple Trees?

- If we only have one decision tree, it might make mistakes (called overfitting) by memorizing the training data.
- By using many trees with different samples of data and features, the random forest averages out the errors of individual trees, making better predictions overall.

They do the **Bagging**!

Random Forest

Bring the maths..

Why Multiple Trees?

- If we only have one decision tree, it might make mistakes (called overfitting) by memorizing the training data.
- By using many trees with different samples of data and features, the random forest averages out the errors of individual trees, making better predictions overall.

They do the **Bagging**!

Random Forest

Bring the maths..

Random forests use a technique called **bagging** (bootstrap aggregating).

Bagging: Each tree is trained on a random sample (with replacement) of the data.

Averaging the Trees: For regression, we take the average of all tree predictions; for classification, the majority vote wins.

Random Forest

Bring the maths..

Classification (Majority Voting):

$$\hat{y} = \text{mode}(T_1(x), T_2(x), \ldots, T_n(x))$$

where T1,T2,...,Tn are the decision trees in the forest, and xxx is the input.

Regression (Averaging):

$$\hat{y} = \frac{1}{n}\sum_{i=1}^{n} T_i(x)$$

where *Ti(x)* is the prediction from the *i-th* tree, and *n* is the total number of trees.

Random Forest

Practical example?

Let’s say we want to predict whether a customer will get a loan. We have several features like income, credit score, and employment status.

Income	Credit Score	Employment Status	Loan Approved
50K	700	Employed	Yes
30K	650	Self-Employed	No
70K	750	Employed	Yes
...	...	...	...

Create a Random Forest:

- Each tree will be trained on a different subset of the data (e.g., some trees might focus more on credit score, others on income).
- Each tree will make a prediction (Yes/No for loan approval).

Majority Vote:

- After each tree votes, the most common decision will be the final prediction.

Random Forest

Advantages?

- **More Accurate:** By using many trees, random forests generally provide more accurate predictions than a single decision tree.
- **Handles Missing Data:** Random forests can handle missing data better because they use multiple decision trees.
- **Reduces Overfitting:** Single trees can memorize training data, but random forests reduce this risk by averaging many trees.
- **Feature Importance:** Random forests can help identify the most important features (e.g., income or credit score) by measuring how much each feature reduces uncertainty.

Future of AI


সবার জন্য কৃত্তিম বুদ্ধিমত্তা
The Matrix
has you...
Java
আপনার সন্তানকে
মেশিন লার্নিং শিক্ষা দিন
SUFIAN SIDDIQUI