

🌳 Decision Trees Simplified! 🌳

If you've ever played "20 Questions," you've already experienced the logic of a **Decision Tree**! It's a machine learning algorithm that splits data into smaller groups based on decision rules. Here's why Decision Trees are a favorite among ML practitioners:

🚀 Key Features:

- 1 **Easy to Understand:** Interpretable and intuitive, even for non-technical stakeholders!
- 2 **Versatile:** Handles both numerical and categorical data seamlessly.
- 3 **No Scaling Required:** Unlike many ML models, Decision Trees don't require feature scaling or normalization.

💡 How it Works:

- Start at the **root node** (the entire dataset).
- Split data into branches based on features and thresholds.
- Each split tries to reduce **impurity** (measured by Gini Impurity or Entropy).
- The process continues until reaching a **leaf node** (final decision point).

🔥 Real-World Uses:

- **Credit Scoring** 📊
- **Healthcare Diagnosis** 🏥
- **Marketing Campaigns** 🎯

⚖️ Pros and Cons:

Pros:

- ☑ Simple, explainable, and requires minimal data prep.
- ☑ Can handle missing values.

Cons:

- ⚠ Prone to **overfitting** (but you can prune it or use Random Forests to mitigate!).
- ⚠ Splits can be biased towards features with more levels.

GitHub link: <https://github.com/NafisAnsari786/Machine-Learning-Algorithms/blob/main/8%20Decision%20Tree/Exercise/Titanic%20Survival%20prediction.ipynb>