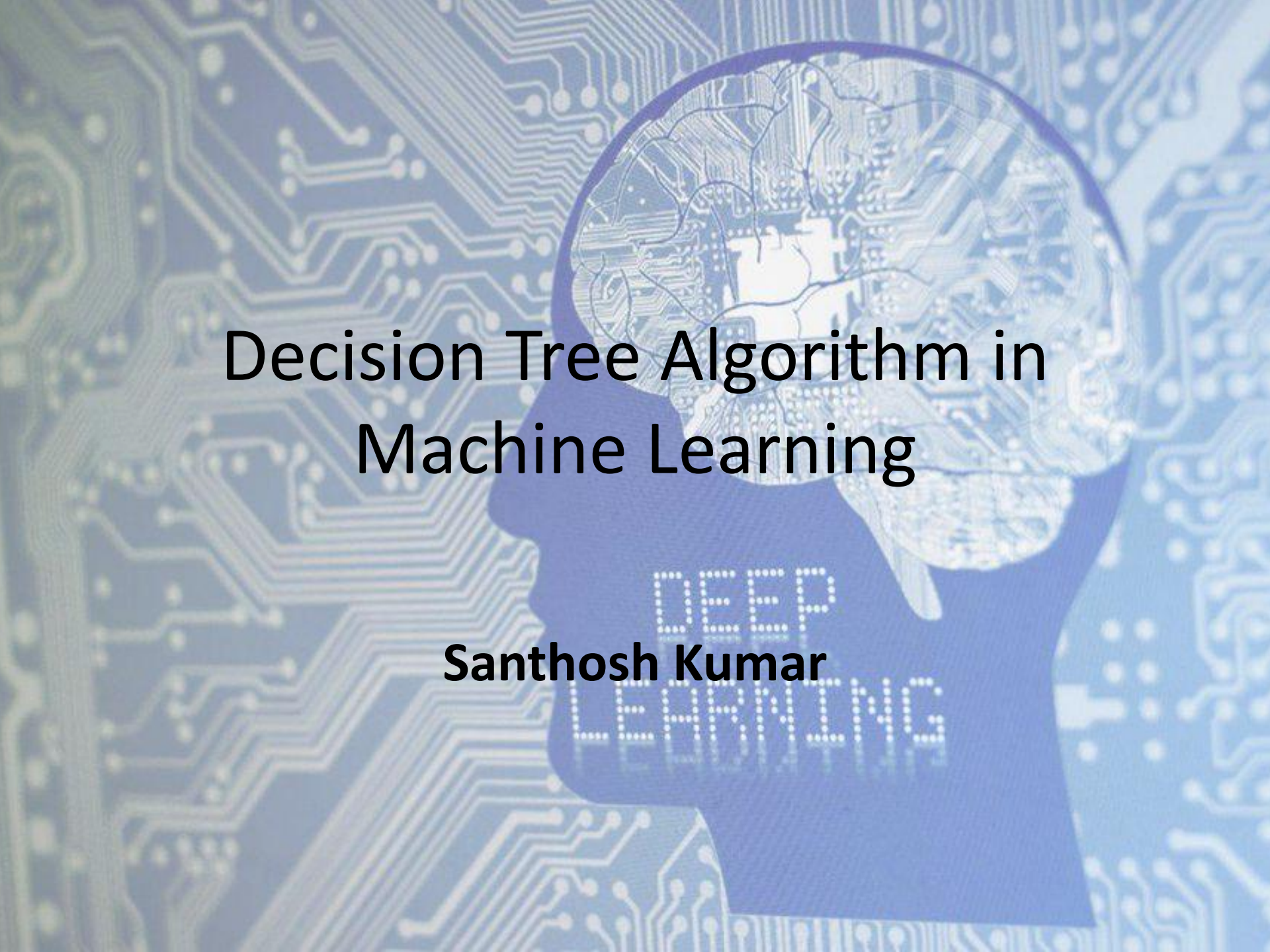
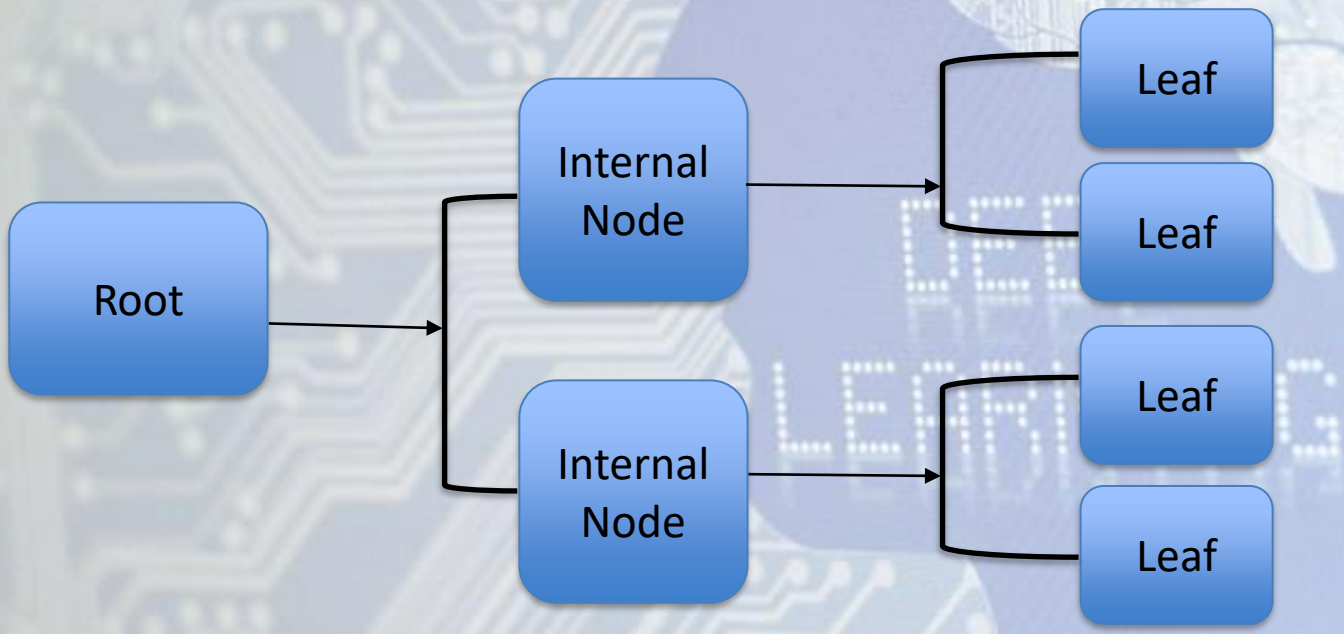
Decision Tree Algorithm in Machine Learning



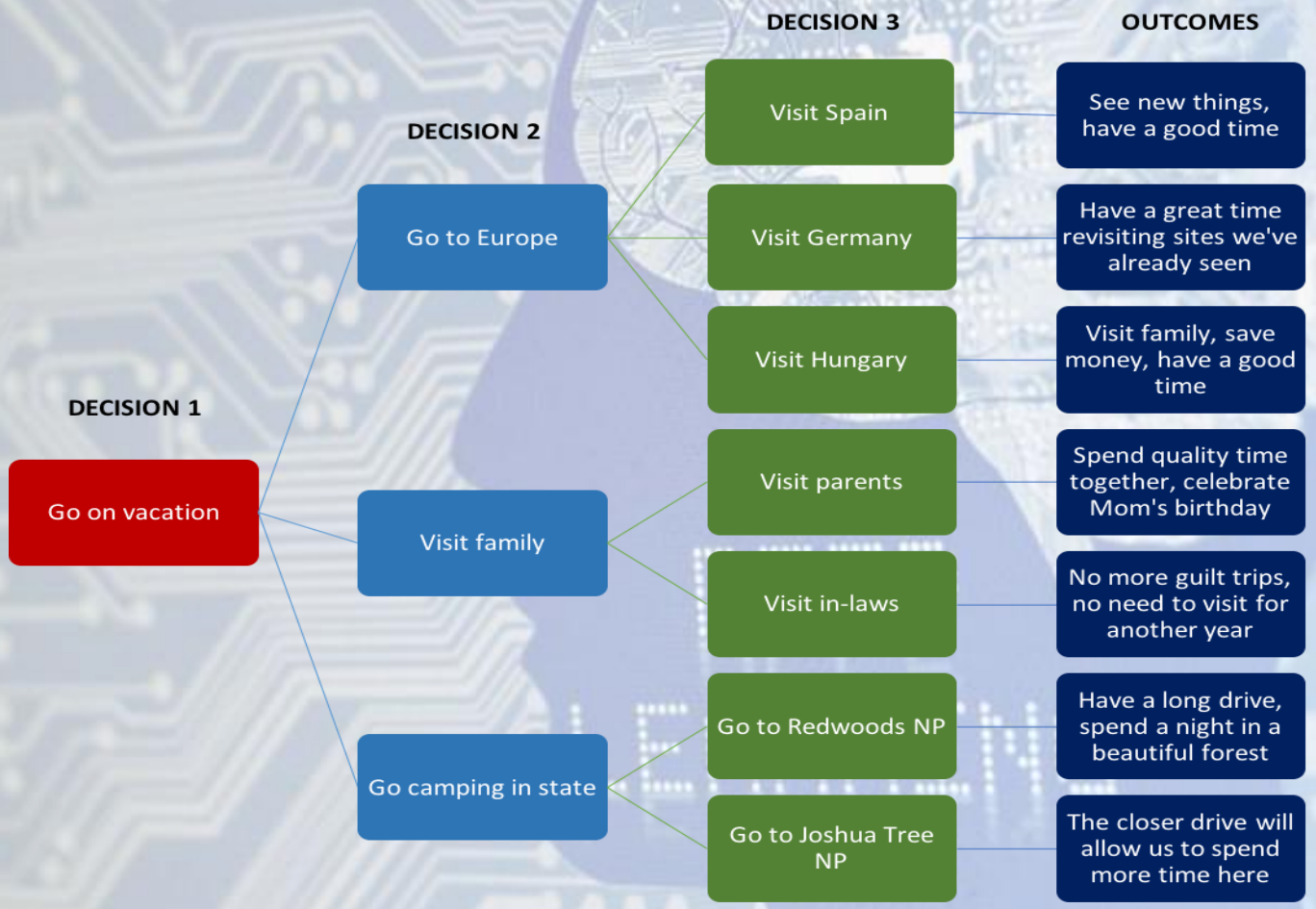
Santhosh Kumar

Introduction

- A Decision Tree is a supervised learning algorithm used for classification and regression.
- Works like a flowchart: root node (best feature to split)→ Internal Node→ Leaf nodes (final output class)



Example of a Decision Tree



How It Works

- A decision tree operates as a flowchart-like structure for making predictions or classifications based on a series of decisions inferred from data.

Workflow:

- 1. Start with all data at the root.
- 2. Split using the best attribute.
- 3. Repeat recursively.
- 4. Stop when node is pure(all the values falls under same category) or stopping condition met(as specified in parameters).

Decision Tree in Python

- Decision tree in Python is most commonly done using the ***scikit-learn*** library
- **Work-flow:**

Import necessary libraries Load and prepare the data

Train the model

Make predictions and evaluate

Parameters

- **Gini Impurity:** Probability of misclassification.
- **Entropy:** Measure of disorder/impurity.
- **Log-loss:** Reduction in entropy after a split.
- **Code:** class sklearn.tree.DecisionTreeClassifier(*, criterion='gini', splitter='best', max_depth=None, min_samples_split=2, min_samples_leaf=1, min_weight_fraction_leaf=0.0, max_features=None, random_state=None, max_leaf_nodes=None, min_impurity_decrease=0.0, class_weight=None, ccp_alpha=0.0, monotonic_cst=None)

Gini Impurity Example

- A measure of how often a randomly chosen sample would be **incorrectly classified** if randomly labeled.
- p_*i* is the probability of randomly picking an item of class *i*
- **Formula:** Gini = 1 - Σ(p_*i*^2).
- **Example:** 80% Yes, 20% No → Gini = 0.32.

Entropy Example

- A measure of **uncertainty/disorder** in a dataset.
- **Formula**: Entropy = - Σ(p_*i* log2(p_*i*)).
- **Example**: 80% Yes, 20% No → Entropy = 0.72.
- 50/50 split → Entropy = 1 (max disorder).
- 90/10 or 80/20 → Entropy = 0.47 (min disorder)
- 100/0 or 0/100 → Entropy = 0 (Pure node)

Log Loss (Cross-Entropy Loss)

- Log Loss measures the performance of a classification model that outputs probabilities.
- Used in probabilistic classifiers.
- Penalizes wrong predictions with high confidence.
- Example: True = 1, Predicted = 0.1 → High penalty.

Advantages

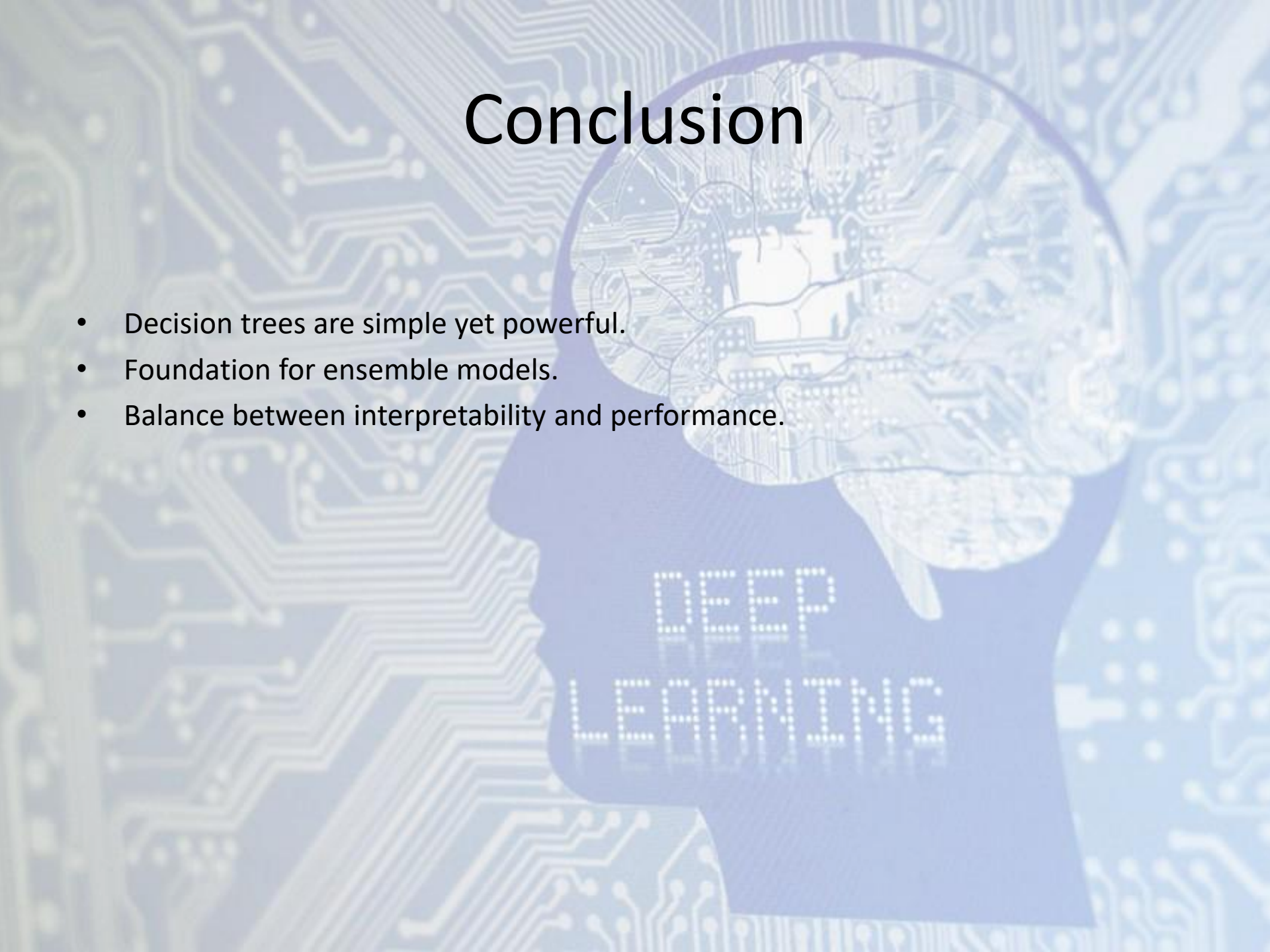
- Easy to interpret & visualize.
- Works with numerical and categorical data.
- Requires little preprocessing.
- "gini" → faster, works well in practice.
- "entropy" / "log_loss" → use when you want information-gain-based splitting.

Limitations

- Overfitting with deep trees.
- Sensitive to small data changes.
- Biased towards features with many categories.

Conclusion

- Decision trees are simple yet powerful.
- Foundation for ensemble models.
- Balance between interpretability and performance.



Links to refer:

- Python_web_page
- Web_page_2
- Parameters
- Link_4
- YouTube Links

Thank you !!!