CSCI316: Week 4 - Classification by Splitting Data (Decision Trees)

📑Topic Overview

This week's focus is on understanding **Decision Trees** as a supervised classification model. You'll explore how models are trained (induction) and used (deduction), the theory behind splitting criteria, and practical implementation.

🔍1. The Classification Problem

- **Goal**: Use attributes/features to classify an instance into a target class.
- **Training Data**: Labeled examples used to learn a model.
- **Test Data**: Unlabeled examples used to evaluate the model.

Example dataset:

Tid	Attrib1	Attrib2	Attrib3	Class
1	Yes	Large	125K	No
...	...	...	...	...
10	No	Small	90K	Yes

The objective is to induce a **Decision Tree (DT)** that generalizes the pattern in training data and applies it to unseen data.

🗜️2. What is a Decision Tree?

- A **flowchart-like tree structure**:
- Internal node: test on an attribute.
- Branch: outcome of the test.
- Leaf node: class label.

Advantages:

- Mimics human decision-making.
- Easy to interpret.
- Good accuracy and efficiency.

---

3. Tree Induction Process

Recursive greedy strategy to build the tree:

1. If all records belong to the same class → create a leaf node.
2. If dataset is empty → create leaf node using majority class of parent.
3. If no attributes left → majority vote.
4. Otherwise:
5. Find the best attribute to split.
6. Split data.
7. Recursively apply on child nodes.

🕐4. Splitting Criteria

🔗Attribute Types:

- **Nominal/Categorical**: e.g., marital status, car type
- **Ordinal**: e.g., small < medium < large
- **Continuous**: e.g., income, age

🕐Splitting Strategies:

- **Multi-way split**: one branch per unique value
- **Binary split**: partition values into two subsets

🪣Continuous Attributes:

- Discretization or thresholding: e.g., (A < 80K), (A ≥ 80K)
- Try all possible splits and choose the best

⚖️5. Measuring Impurity & Best Split

Common Impurity Measures:

Measure	Idea
Entropy	Uncertainty/information content (Shannon)
Information Gain	Difference in entropy before and after split
Gain Ratio	Normalize InfoGain to avoid multivalued bias
Gini Index	Measure of likelihood of incorrect classification
Variance	Used for binary classification

Example - Entropy: $H(D) = -\sum p(x) \log_2 p(x)$

Information Gain: $IG(D, A) = H(D) - \sum_{i=1}^{v} \frac{|D_i|}{|D|} H(D_i)$

💡6. Stopping Criteria

- No attributes left.
- All records in node are same class.
- Dataset is empty.
- Pre-pruning: based on thresholds like min dataset size, impurity level, or max depth.

📝7. Tree Pruning

Overfitting:

- Tree fits training data too closely, poor generalization.

🗓️Solutions:

- **Pre-pruning**: Stop tree early if conditions not met.
- **Post-pruning**: Build full tree, then remove branches using validation set.

📈8. Random Forest: Ensemble of Trees

- Combines predictions from multiple decision trees.
- **Bagging**: Bootstrapped aggregation (samples with replacement)
- **Forest-RI**: Random feature selection at each split
- **Forest-RC**: Random combinations of features

Advantages:

- Improves accuracy
- Reduces variance
- Handles large/high-dimensional data

🛠️9. Python Implementation Summary

- Use Python `dict` to represent tree

```
# Example: A tree with one feature split
{
```

```
 0: {"Yes": "No", "No": "Yes"}
}
```

- Use recursive functions to build and traverse the tree
- `calcShannonEnt()`: Computes entropy
- `chooseBestMultiSplit()`: Picks best split attribute
- Handle categorical and binned continuous features

🎓Practice Idea

Build a decision tree from scratch in Python using:

- Manual entropy calculations
- Dictionary-based tree structure
- Recursion for training and classification

Let me know if you'd like quizzes, coding labs, or cheat sheets for this topic!