Module 10: Decision Trees: Part Two

Learning outcomes

1. Identify a tree depth that balances model complexity and accuracy.
2. Determine how bagging, random forests and boosting improve prediction accuracy.
3. Evaluate decision tree and ensemble classifiers using feature selection.
4. Apply tree ensemble methods to real-world problems.
5. Differentiate between decision trees, KNN and ensemble methods.

Why use tree ensembles?

- Single decision trees can overfit the training data.
- Ensemble methods combine multiple trees to reduce variance and improve generalisation.
- The main trade-off is between interpretability and performance.

Tree depth

- Shallow trees may underfit the data.
- Deep trees can model complex relationships but risk overfitting.
- Use training and validation accuracy to determine the optimal depth.

Bagging (bootstrap aggregating)

- Bagging trains multiple models on different bootstrap samples drawn with replacement.
- Bagging aggregates predictions – majority voting for classification, averaging for regression.
- Bagging is the most effective when individual trees are unstable and their errors are uncorrelated.
- Bagging helps reduce variance and improve prediction stability.

Random forests

- A random forest is an extension of bagging that introduces additional randomness.
- At each split, the model selects a random subset of features.
- This decorrelation leads to better generalisation and reduces the dominance of strong predictors.

Boosting

- Boosting builds an ensemble of models sequentially.

- Boosting corrects the errors of previous models with the use of each new model.
- Boosting improves accuracy by reducing both bias and variance.
- Boosting can be sensitive to noisy data but often yields strong performance.

Model building workflow

1. Import the data set and conduct exploratory data analysis (EDA).
2. Translate categorical predictors into numerical format.
3. Prepare the target variable and split the data into training, validation and test sets.
4. Train ensemble models: decision tree, random forest, boosting and bagging.
5. Compare validation scores to select the best-performing classifier.
6. Perform feature selection to identify the most relevant predictors.

Feature selection methods

- Impurity-based importance
  - It measures how well a feature splits the data.
  - Features that reduce impurity the most are considered more important.
  - This method reflects importance within the training data only.

- Permutation importance
  - It evaluates feature importance by measuring the drop in model performance after shuffling a feature's values.
  - More reliable for identifying features that contribute to generalisable predictions.
  - It can be applied to both training and validation data.

Comparing KNN and decision trees

KNN vs decision trees

Aspect	KNN	Decision tress
Learning type	Lazy learner	Eager learner
Interpretability	Low	High (especially for shallow trees)
Sensitivity to scaling	High	Low
Performance on high-dim	Poor (due to distance metrics)	Better (selects relevant splits)

Training time	Minimal (no training phase)	Requires model fitting
Memory usage	High	More efficient once trained

Model fairness and interpretability

- Interpretability helps users understand and trust model decisions.
- Fairness requires ensuring models do not reinforce existing societal biases.
- Bias can enter through training data, feature selection or model assumptions.
- Transparent models, such as decision trees, aid in identifying unfair decision patterns.
- In high-stakes domains, it is essential to involve diverse stakeholders in model development.