Report: Decision Trees and Random Forests

Report: Decision Trees and Random Forests

Objective : The objective of this task is to explore and implement two popular tree-based machine learning models - Decision Trees and Random Forests - using the heart.csv dataset. The task aims to build classification models, visualize decision-making processes, interpret feature significance, and evaluate performance using standard metrics.

Dataset Overview

Name: heart.csv

Description: The dataset contains 303 rows and 14 features including age, cholesterol level, resting blood pressure, chest pain type, etc. The target variable indicates whether the patient has a heart disease (1) or not (0).

Tools and Libraries Used

- Python 3.x

- Pandas for data handling

- Scikit-learn for model building and evaluation

- Graphviz for decision tree visualization

- Matplotlib for plotting

- Seaborn (optional) for advanced visualizations

**Methodology**

1. Data Preprocessing

- Loaded the dataset using Pandas

- Verified data types and missing values (none found)

- Split data into features (X) and target (y)

- Used train\_test\_split() to divide data (80% training, 20% testing)

2. Decision Tree Classifier

- Trained a basic Decision Tree using DecisionTreeClassifier()

- Set maximum depth to avoid overfitting

- Visualized the tree using export\_graphviz() + graphviz.Source()

- Measured training and testing accuracy

3. Overfitting Analysis

# - Varying the max\_depth parameter showed:

- Very shallow trees underfit (low train & test accuracy)

- Very deep trees overfit (high train accuracy but low test accuracy)

- Optimal accuracy was found at moderate depth (e.g., depth = 4-5)

4. Random Forest Classifier

- Trained using RandomForestClassifier(n\_estimators=100)

- Aggregated results from multiple decision trees (bagging)

- Compared accuracy and generalization vs. single decision tree

5. Feature Importance

- Extracted feature\_importances\_ from the trained random forest

- Visualized using a horizontal bar chart

- Found that features like cp (chest pain), thalach (max heart rate), and exang (exercise-induced angina) were most predictive

6. Cross-Validation

- Used cross\_val\_score() with 5 folds

- Applied only on Random Forest for better generalization analysis

- Average CV accuracy confirmed robustness of the model

Results Summary

Model | Train Accuracy | Test Accuracy | Cross-Validation (Avg)

Decision Tree | ~0.98 | ~0.79 | -

Random Forest | ~0.99 | ~0.84 | ~0.85

Key Takeaways

- Decision Trees are easy to interpret but prone to overfitting without regularization (e.g., max depth).

- Random Forests significantly improve generalization by combining multiple trees and using randomness (bagging).

- Feature importance analysis can provide valuable insights into which variables most influence the prediction.

- Cross-validation is essential to assess model reliability beyond a single test split.

Project Files

Task-5-DecisionTrees

- decision\_tree\_random\_forest.ipynb

- heart.csv

- README.md

Conclusion

This task provided a comprehensive understanding of decision tree-based models and ensemble techniques. By comparing single tree performance with that of a random forest, we observed the power of ensemble methods in reducing overfitting and improving accuracy.