

Vietnam National University,
Ho Chi Minh City

UNIVERSITY OF SCIENCE
FACULTY OF INFORMATION TECHNOLOGY

Project 03: Decision Tree

CS14003 – INTRODUCTION TO ARTIFICIAL INTELLIGENCE

Ngo Nguyen The Khoa 23127065
Bui Minh Duy 23127040
Nguyen Le Ho Anh Khoa 23127211

April 22, 2025

Contents

1	Group Information	2
2	Project Information	2
3	Work Assignment Table	3
4	Self-evaluation	3
5	Dataset Analysis and Experiments	4
5.1	Dataset Preparation and Preprocessing	4
5.2	Interpreting Classification Report and Confusion Matrix	4
5.3	Breast Cancer Wisconsin Dataset	6
5.4	Wine Quality Dataset	15
5.5	Car Evaluation Dataset	18
6	Comparative Analysis	23
7	References	24

1 Group Information

- **Subject:** Introduction to Artificial Intelligence.
- **Class:** 23CLC09.
- **Lecturer:** Bui Duy Dang, Le Nhut Nam.
- **Team members:**

No.	Fullname	Student ID	Email
1	Ngo Nguyen The Khoa	23127065	nntkhoa23@clc.fitus.edu.vn
2	Bui Minh Duy	23127040	bmduy23@clc.fitus.edu.vn
3	Nguyen Le Ho Anh Khoa	23127211	nlhakhoa23@clc.fitus.edu.vn

2 Project Information

- **Name:** Decision Tree Classifier.
- **Developing Environment:** Visual Studio Code (Windows).
- **Programming Language:** Python.
- **Libraries and Tools:**
 - **Libraries:**
 - * **scikit-learn:** Machine learning library for training and evaluating decision tree models.
 - * **pandas:** Data manipulation and analysis.
 - * **numpy:** Numerical operations.
 - * **matplotlib, seaborn:** Data visualization libraries.
 - * **graphviz:** Visualization of decision trees.
 - **Tools:**
 - * **Git, GitHub:** Source code version control.
 - * **Visual Studio Code:** Code editor for Python, LaTeX.
- **Datasets:**
 - **Breast Cancer Wisconsin (Diagnostic)**
 - **Wine Quality**
 - **Car Evaluation**

3 Work Assignment Table

No.	Task Description	Assigned to	Rate
1	Prepare all three datasets with proper preprocessing and stratified splits.	T.Khoa, M.Duy	100%
2	Implement and train decision tree models for each dataset with different train/test splits.	T.Khoa, A.Khoa	100%
3	Visualize decision trees using Graphviz.	Anh Khoa	100%
4	Evaluate classifiers with classification reports and confusion matrices.	A.Khoa, T.Khoa	100%
5	Analyze impact of tree depth on accuracy (80/20 split, varying max_depth values).	Minh Duy	100%
6	Research and integrate additional dataset.	Anh Khoa	100%
7	Conduct comparative analysis across the 3 datasets.	The Khoa	100%
8	Visualize and format results (accuracy tables, charts, dataset distributions, etc.).	Minh Duy	100%
9	Write and format final report with all results, insights, and figures.	Minh Duy	100%
10	Ensure overall cohesion, proofreading, and prepare final PDF submission.	All	100%

4 Self-evaluation

No.	Task Description	Rate
1	Prepare datasets with stratified splits and visualize class distributions.	100%
2	Train and visualize decision tree models on all datasets using multiple train/test splits.	100%
3	Evaluate decision trees using classification reports and confusion matrices.	100%
4	Analyze the impact of decision tree depth on model accuracy.	100%
5	Research and integrate an additional dataset for training and evaluation.	100%
6	Conduct comparative analysis across all datasets.	100%
7	Create charts, tables, and visualizations to support findings.	100%
8	Write and format the final report with insights and well-organized results.	100%
9	Team collaboration and adherence to project schedule.	100%

5 Dataset Analysis and Experiments

5.1 Dataset Preparation and Preprocessing

```

1 def split_and_visualize(X, y, dataset_name: str):
2     splits = {}
3
4     for ratio in [0.4, 0.6, 0.8, 0.9]:
5         X_train, X_test, y_train, y_test = train_test_split(
6             X, y, train_size=ratio, stratify=y, shuffle=True, random_state=42
7         )
8
9         splits[ratio] = (X_train, X_test, y_train, y_test)
10
11 return splits

```

To shuffle the dataset and ensure it is split in a stratified fashion, we use the `train_test_split` function from `sklearn.model_selection`. The function takes the dataset and the target variable as inputs, along with the desired train-test split ratio.

- The `shuffle` parameter randomizes the order of the samples before splitting.
- The `stratify` parameter ensures the dataset is split in a stratified fashion.

5.2 Interpreting Classification Report and Confusion Matrix

```

1 def train_evaluate_decision_tree(
2     X_train,
3     y_train,
4     X_test,
5     y_test,
6     dataset_name: str,
7     split_ratio,
8 ):
9     # Dynamically set feature and class names
10    feature_names = X_train.columns.tolist()
11    class_names = [str(cls) for cls in np.unique(y_train)]
12
13    # Train model
14    clf = DecisionTreeClassifier(criterion="entropy", random_state=42)
15    clf.fit(X_train, y_train)
16
17    # Predictions
18    y_pred = clf.predict(X_test)
19
20    # Classification Report (with validation)
21    print(f"\nClassification Report ({dataset_name}, {display_ratio(split_ratio)}):")

```

```
22     print(
23         classification_report(
24             y_test,
25             y_pred,
26             target_names=class_names,
27             labels=np.unique(y_test), # Ensure alignment with actual classes
28         )
29     )
30
31     # Confusion Matrix
32     sns.heatmap(
33         confusion_matrix(y_test, y_pred),
34         annot=True,
35         fmt="d",
36         xticklabels=class_names,
37         yticklabels=class_names,
38     )
```

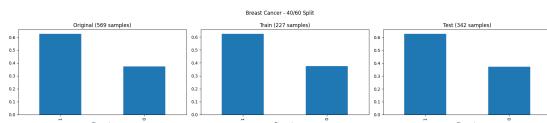
To generate the classification report and confusion matrix:

- The `classification_report` function provides a detailed report of the model's performance, including precision, recall, and F1-score for each class.
- The `confusion_matrix` function generates a matrix that shows the number of correct and incorrect predictions for each class.

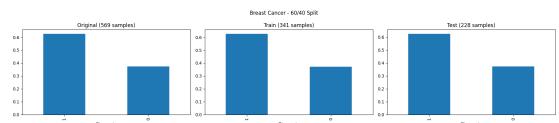
5.3 Breast Cancer Wisconsin Dataset

Dataset Description

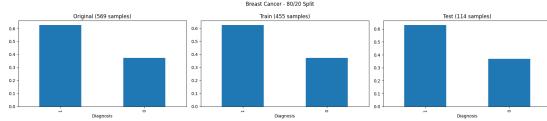
- **Description:** The UCI Breast Cancer Wisconsin (Diagnostic) dataset is used for classifying tumors as malignant or benign based on features derived from its imaging data.
- **Dataset Info:** 569 samples, binary labels (malignant vs. benign), 30 numeric features.
- **Preprocessing:** shuffle & stratified split at 40/60, 60/40, 80/20, 90/10.



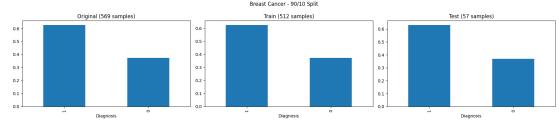
(a) Breast Cancer: class distribution (40/60 split).



(b) Breast Cancer: class distribution (60/40 split).



(c) Breast Cancer: class distribution (80/20 split).



(d) Breast Cancer: class distribution (90/10 split).

Figure 1: Class distributions

Building Decision Tree Classifiers for each train/test proportions

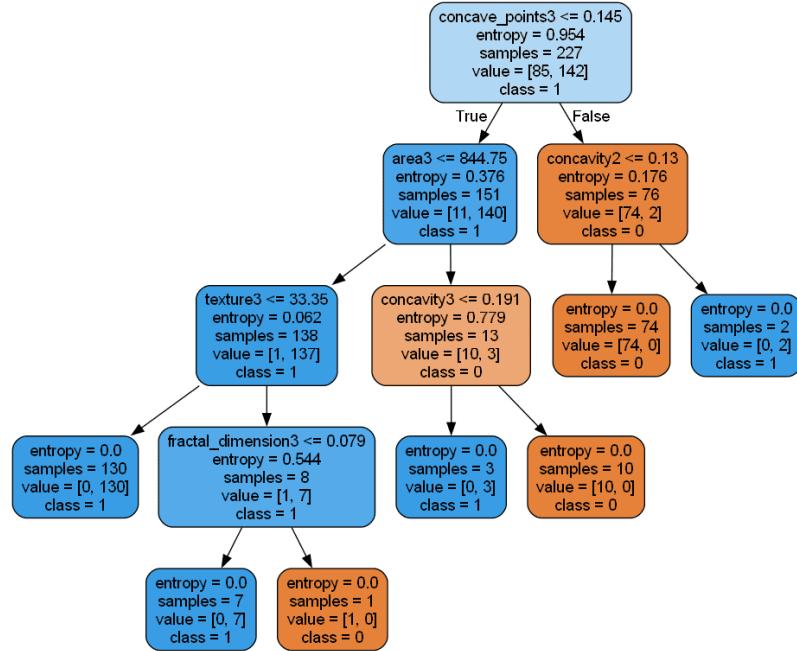


Figure 2: Breast Cancer: decision tree for 40/60 split.

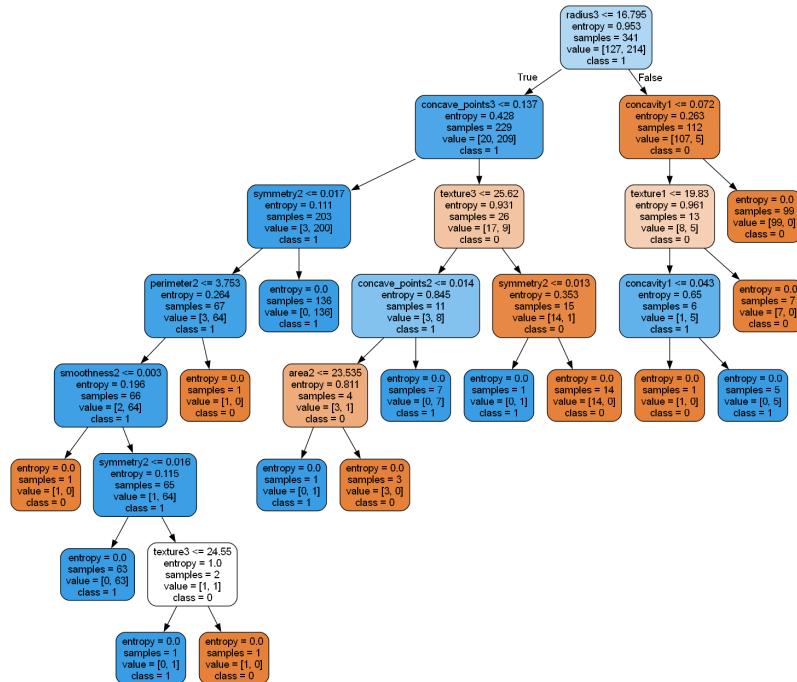


Figure 3: Breast Cancer: decision tree for 60/40 split.

5. Dataset Analysis and Experiments

Project 03: Decision Tree

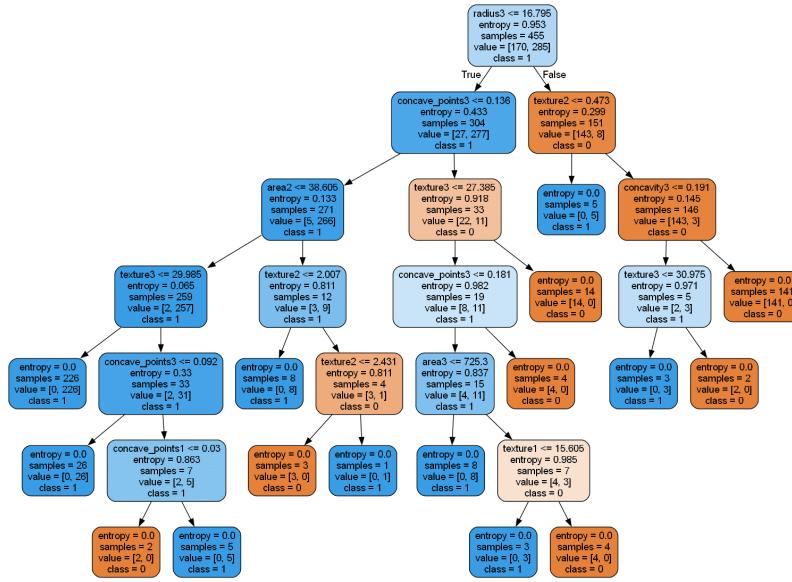


Figure 4: Breast Cancer: decision tree for 80/20 split.

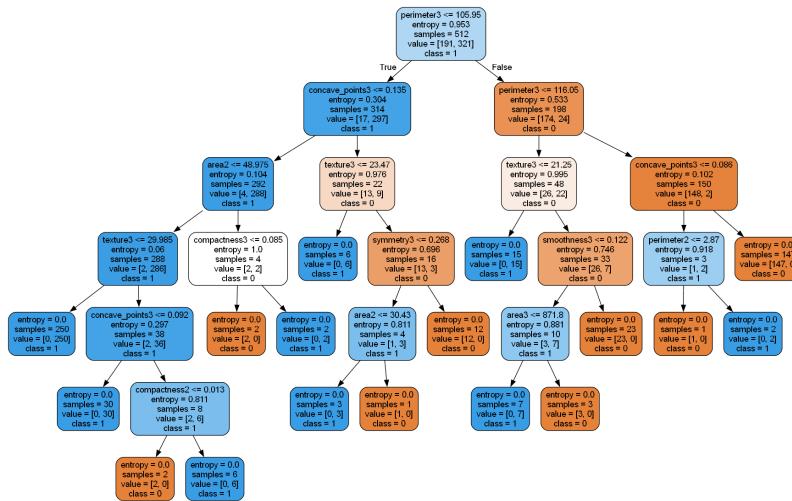
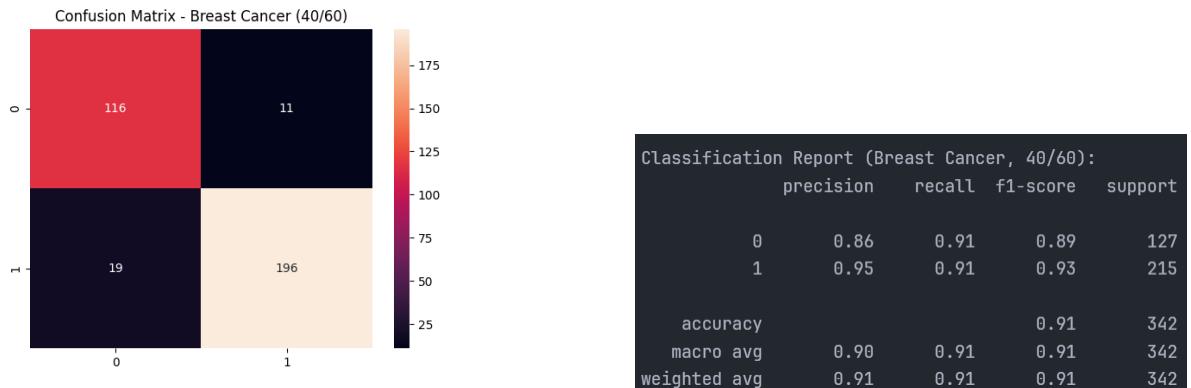


Figure 5: Breast Cancer: decision tree for 90/10 split.

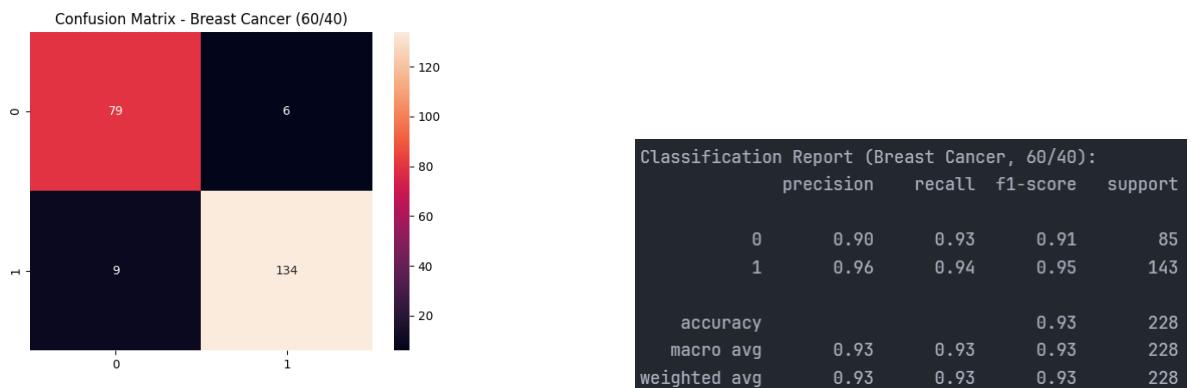
Evaluating the decision tree classifiers



(a) Breast Cancer: confusion matrix (40/60 split).

(b) Breast Cancer: Classification Report (40/60 split).

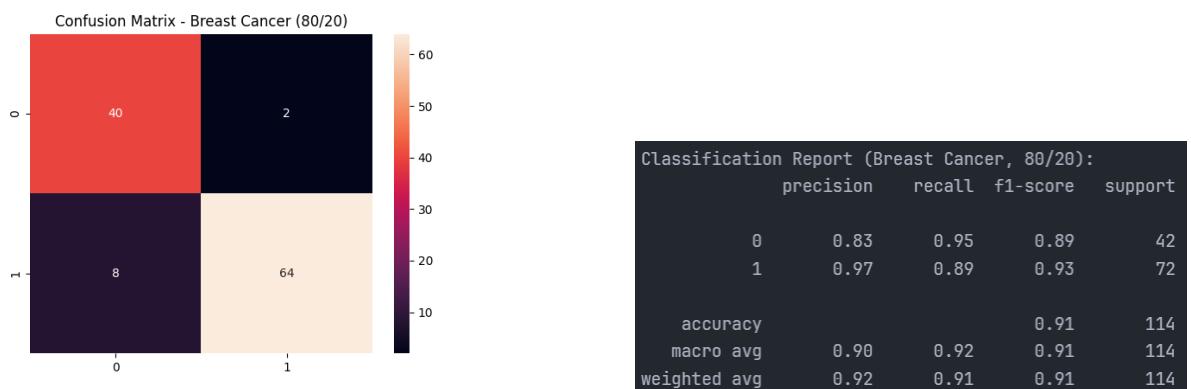
Figure 6: Classification Report and Confusion Matrix (40/60 split)



(a) Breast Cancer: confusion matrix (60/40 split).

(b) Breast Cancer: Classification Report (60/40 split).

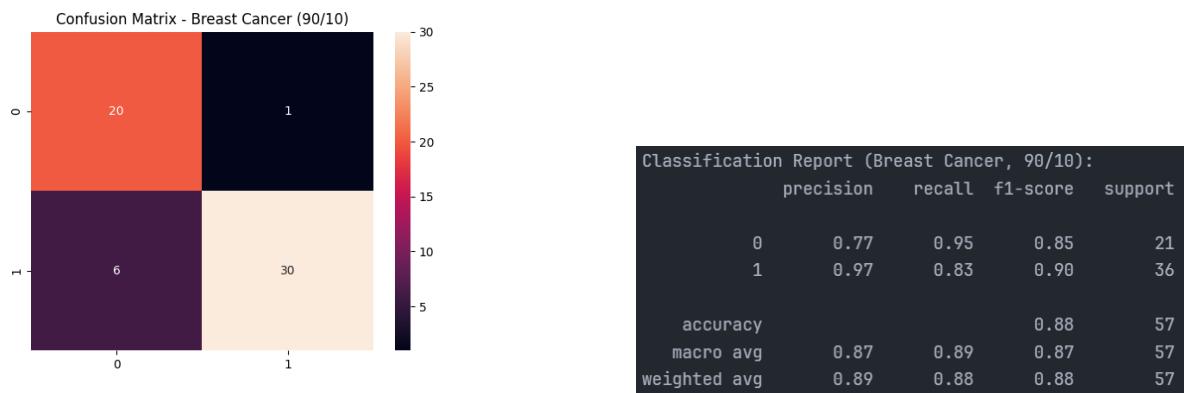
Figure 7: Classification Report and Confusion Matrix (60/40 split)



(a) Breast Cancer: confusion matrix (80/20 split).

(b) Breast Cancer: Classification Report (80/20 split).

Figure 8: Classification Report and Confusion Matrix (80/20 split)



(a) Breast Cancer: confusion matrix (90/10 split).

(b) Breast Cancer: Classification Report (90/10 split).

Figure 9: Classification Report and Confusion Matrix (90/10 split)

Insights into performance of these decision tree classifiers:

- sth
 - sth

Decision Tree Classifier with Different Depths

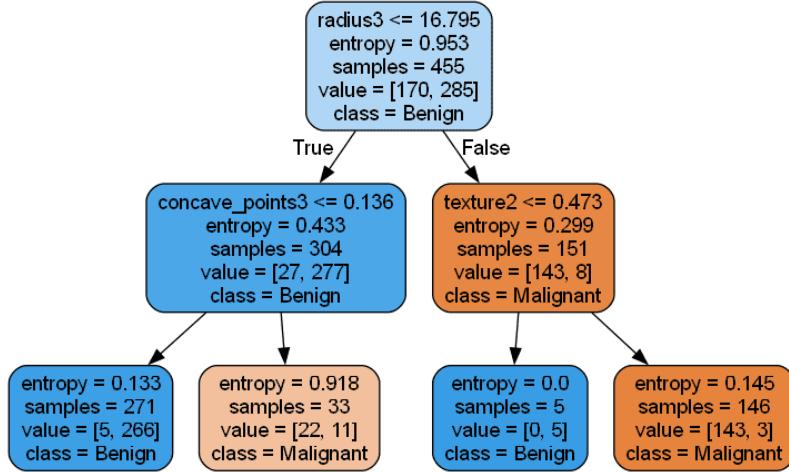


Figure 10: Breast Cancer: decision tree with `max_depth=2` (80/20 split).

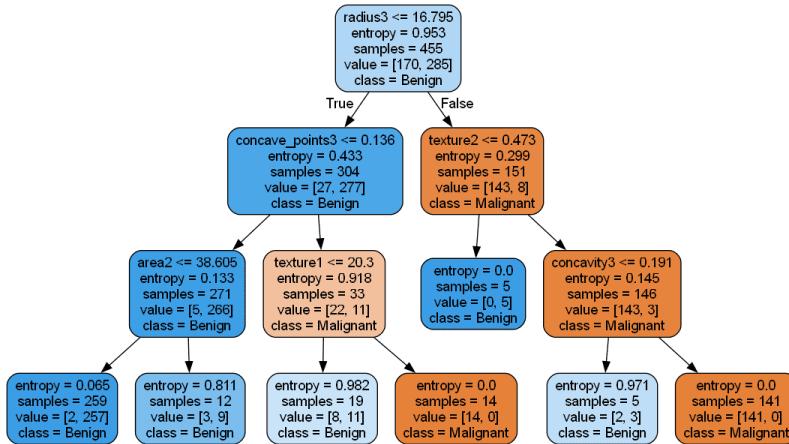


Figure 11: Breast Cancer: decision tree with `max_depth=3` (80/20 split).

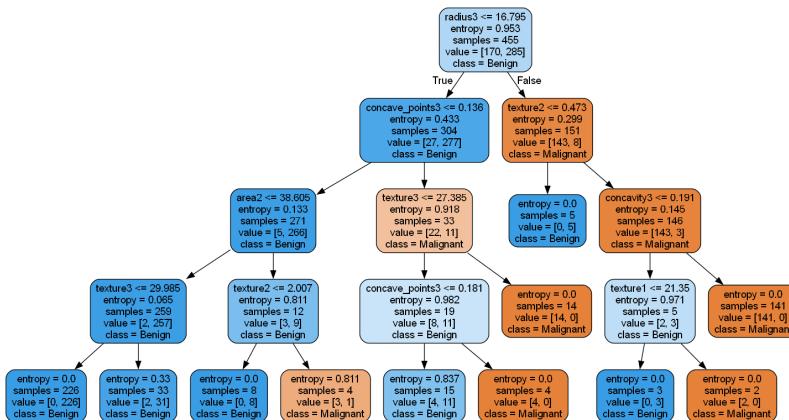


Figure 12: Breast Cancer: decision tree with `max_depth=4` (80/20 split).

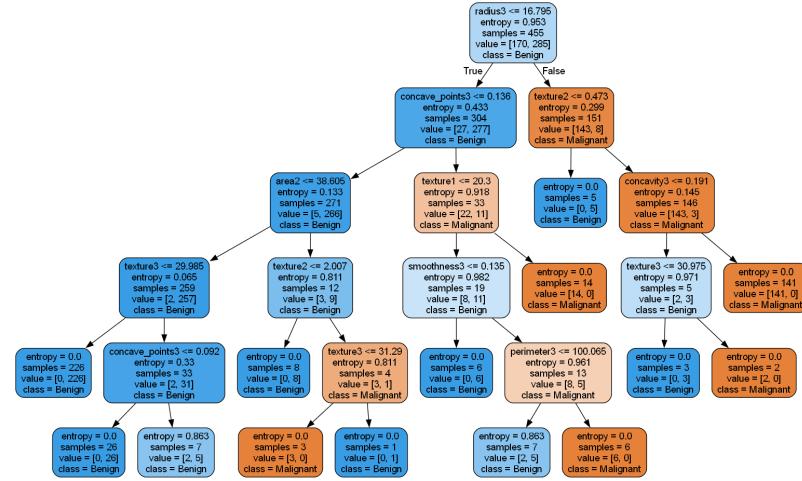


Figure 13: Breast Cancer: decision tree with `max_depth=5` (80/20 split).

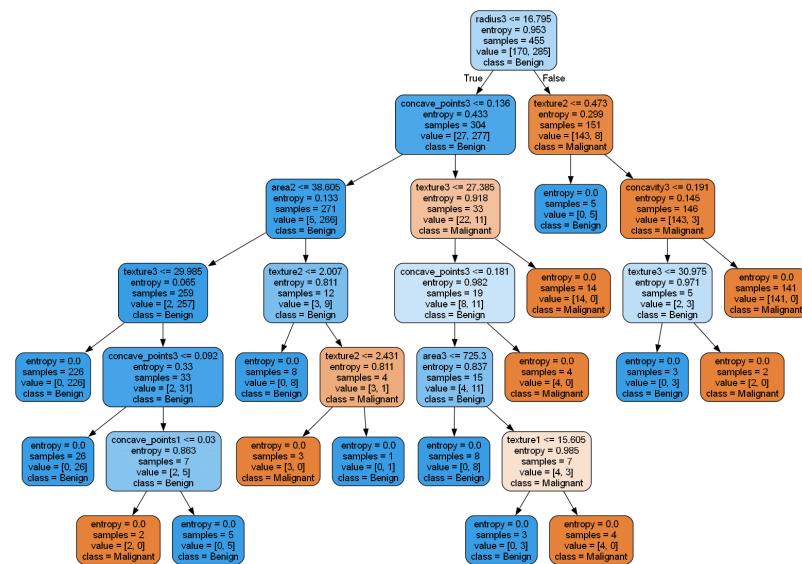
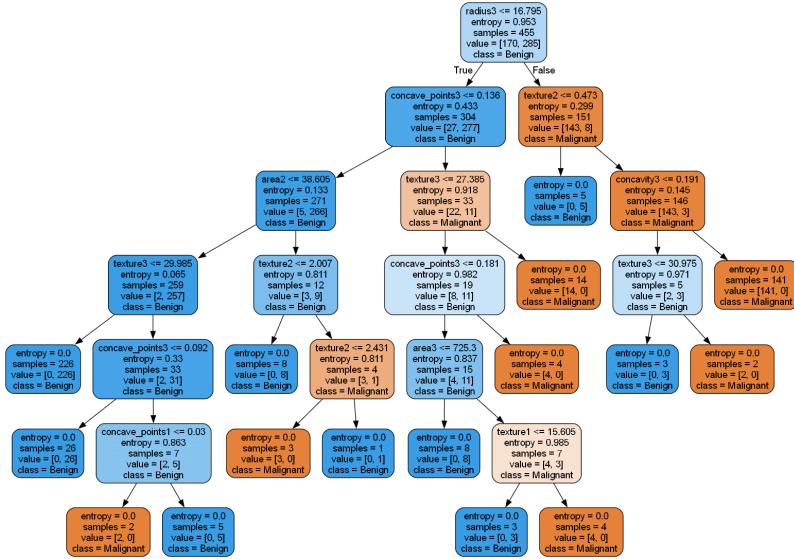
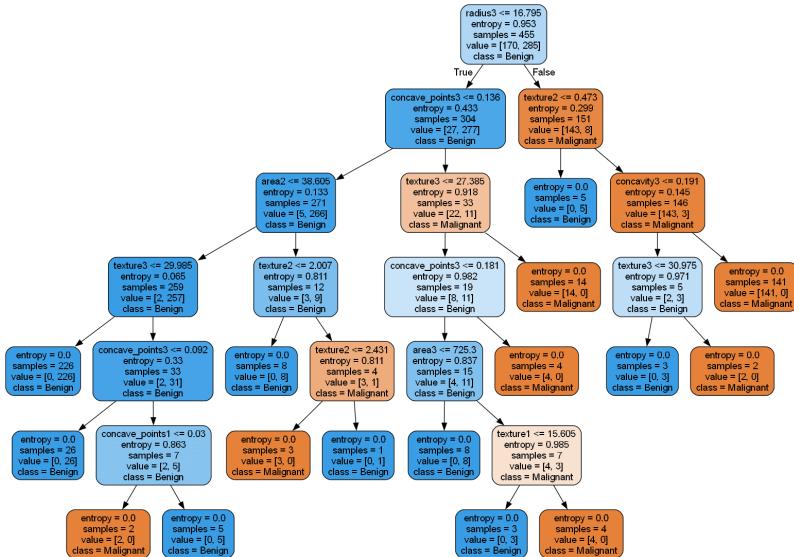
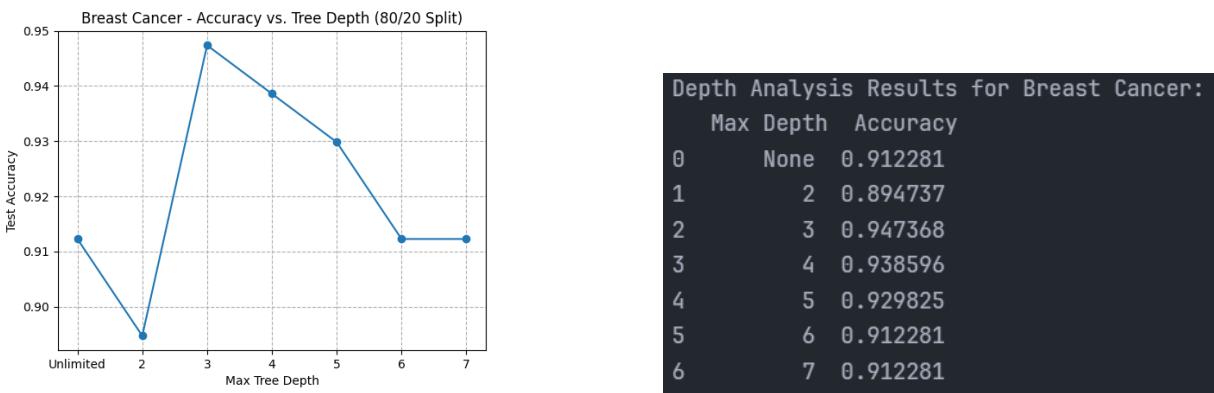


Figure 14: Breast Cancer: decision tree with `max_depth=6` (80/20 split).

Figure 15: Breast Cancer: decision tree with `max_depth=7` (80/20 split).Figure 16: Breast Cancer: decision tree with `max_depth=None` (80/20 split).

Insights

- sth
 - sth

5.4 Wine Quality Dataset

- **Description:** 4,898 samples; original scores 0–10 grouped into Low (0–4), Standard (5–6), High (7–10).
- **Preprocessing:** label encoding, stratified splits at 40/60, 60/40, 80/20, 90/10.

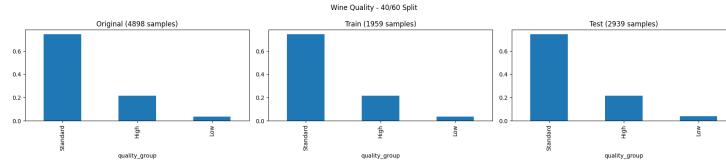


Figure 18: Wine Quality: class distribution (40/60 split).

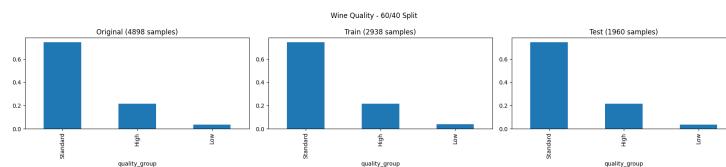


Figure 19: Wine Quality: class distribution (60/40 split).

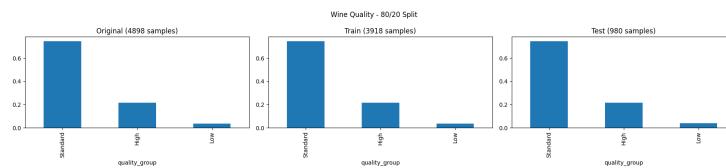


Figure 20: Wine Quality: class distribution (80/20 split).

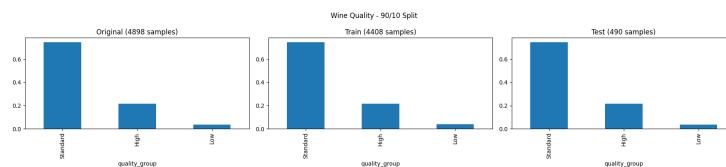


Figure 21: Wine Quality: class distribution (90/10 split).

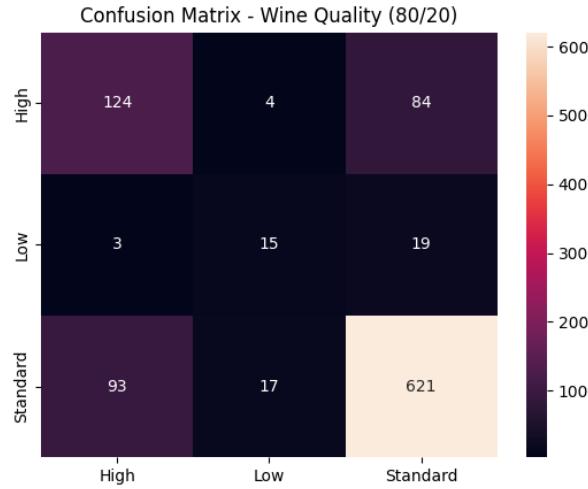
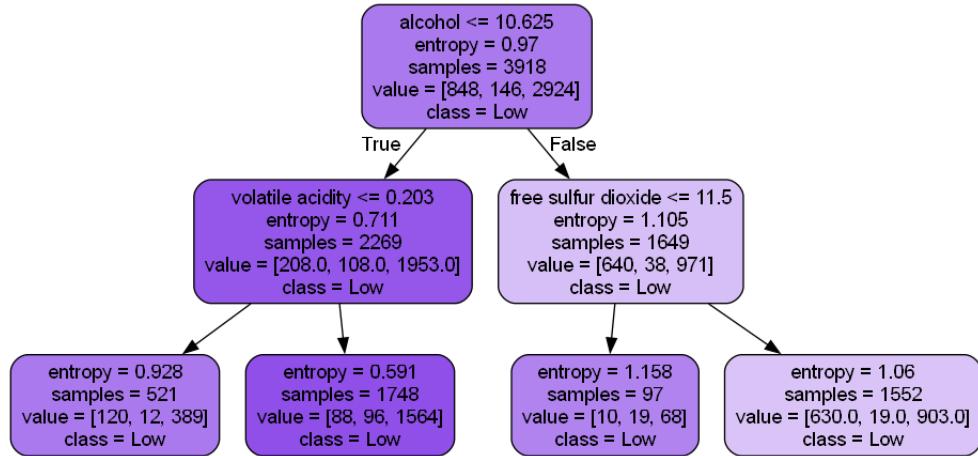
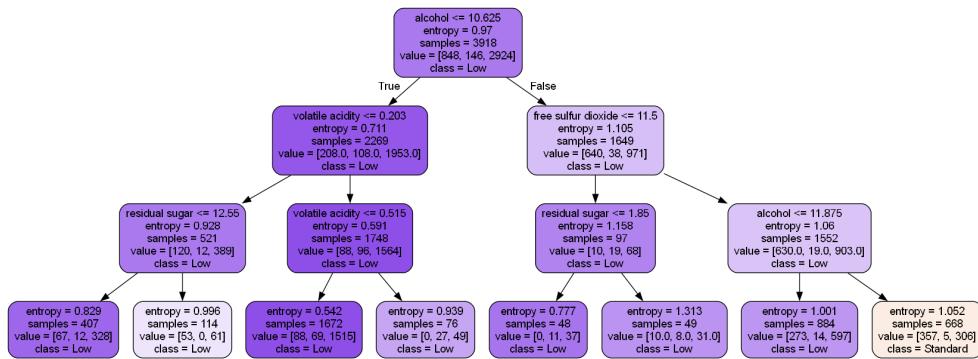


Figure 22: Wine Quality: confusion matrix (80/20 split).

Figure 23: Wine Quality: decision tree with `max_depth=2` (80/20 split).Figure 24: Wine Quality: decision tree with `max_depth=3` (80/20 split).

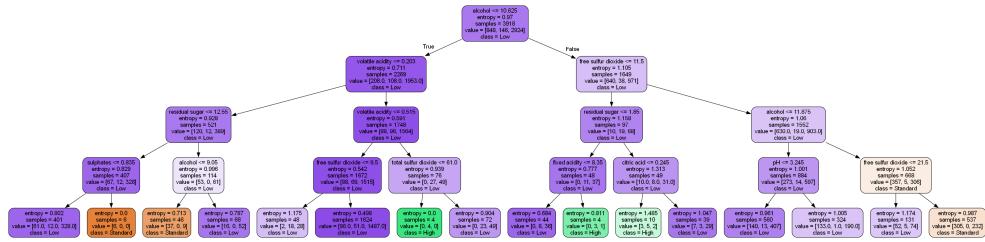


Figure 25: Wine Quality: decision tree with `max_depth=4` (80/20 split).

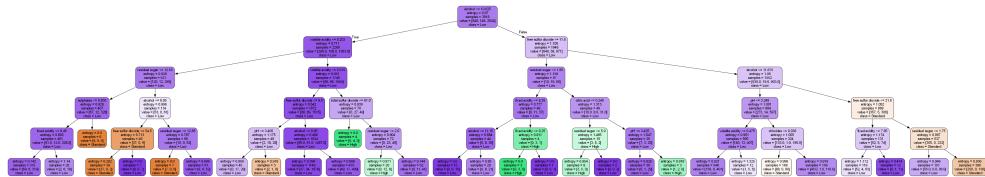


Figure 26: Wine Quality: decision tree with `max_depth=5` (80/20 split).

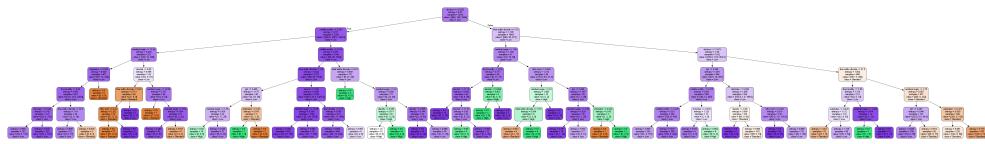


Figure 27: Wine Quality: decision tree with `max_depth=6` (80/20 split).



Figure 28: Wine Quality: decision tree with `max_depth=7` (80/20 split).

5.5 Car Evaluation Dataset

- **Description:** 1,728 samples; 4 classes (unacc, acc, good, vgood), 6 categorical features.
- **Preprocessing:** label encoding, stratified splits at 40/60, 60/40, 80/20, 90/10.

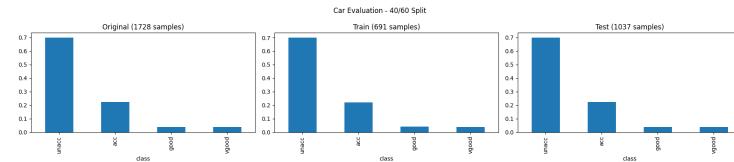


Figure 29: Car Evaluation: class distribution (40/60 split).

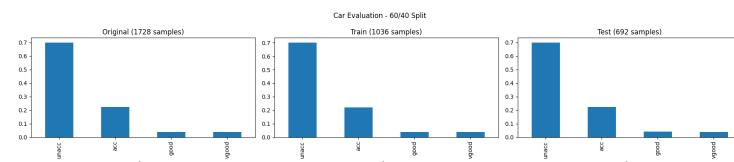


Figure 30: Car Evaluation: class distribution (60/40 split).

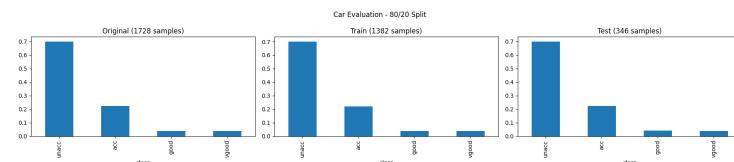


Figure 31: Car Evaluation: class distribution (80/20 split).

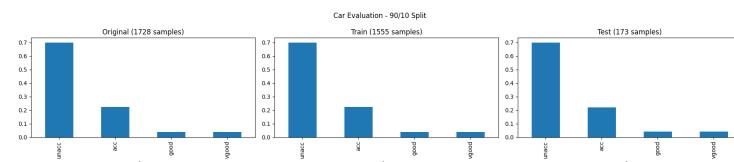


Figure 32: Car Evaluation: class distribution (90/10 split).

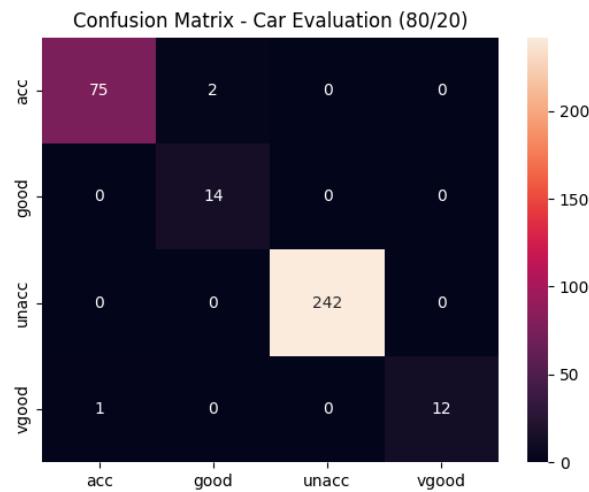


Figure 33: Car Evaluation: confusion matrix (80/20 split).

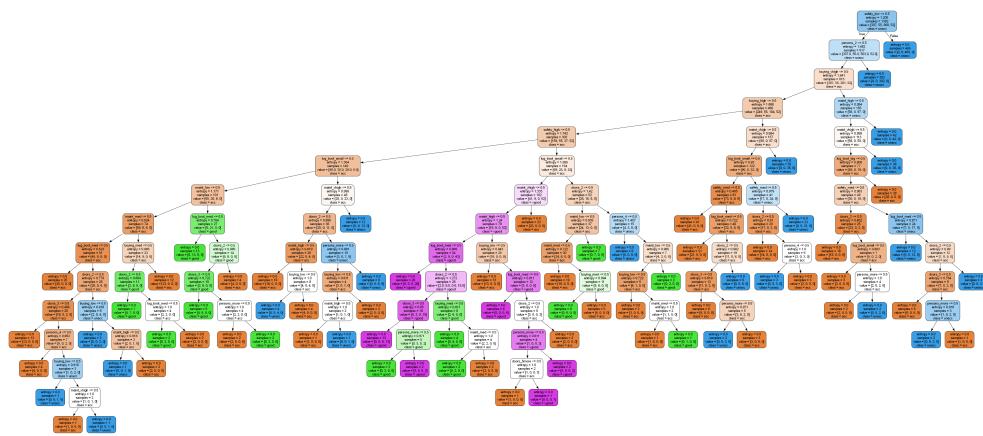
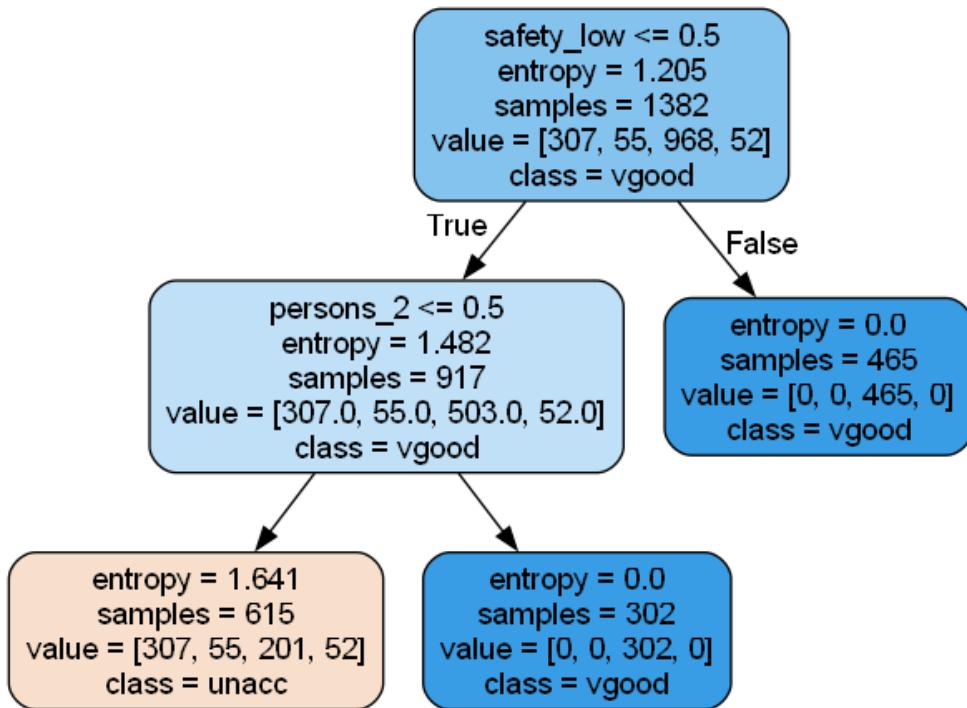
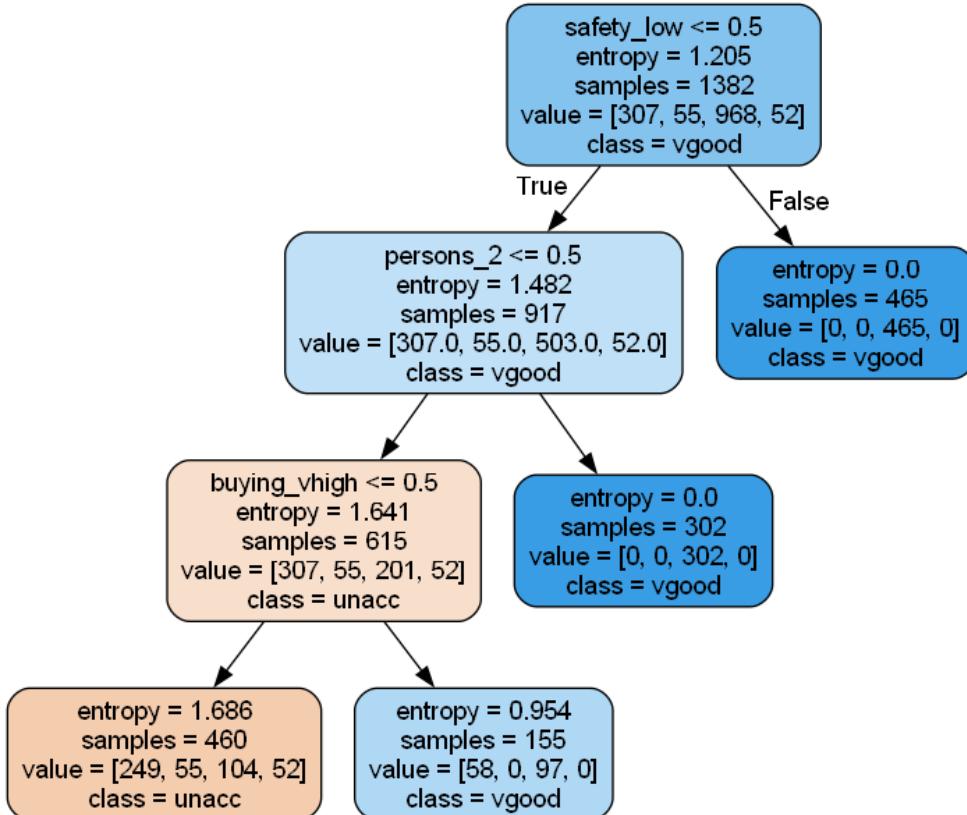
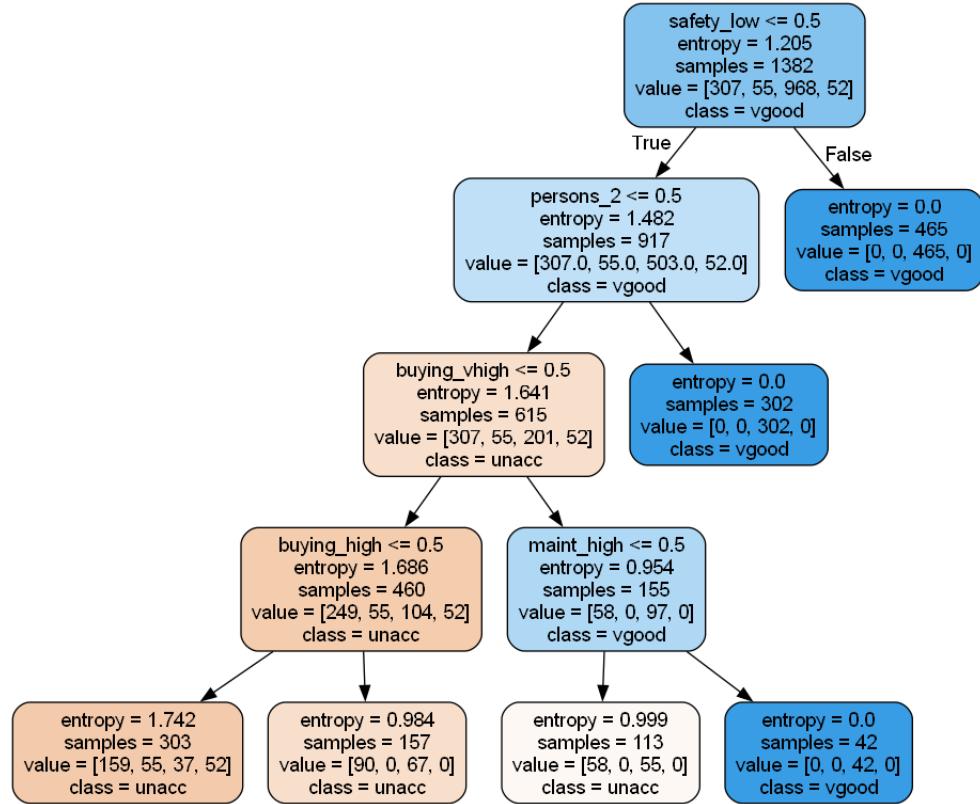
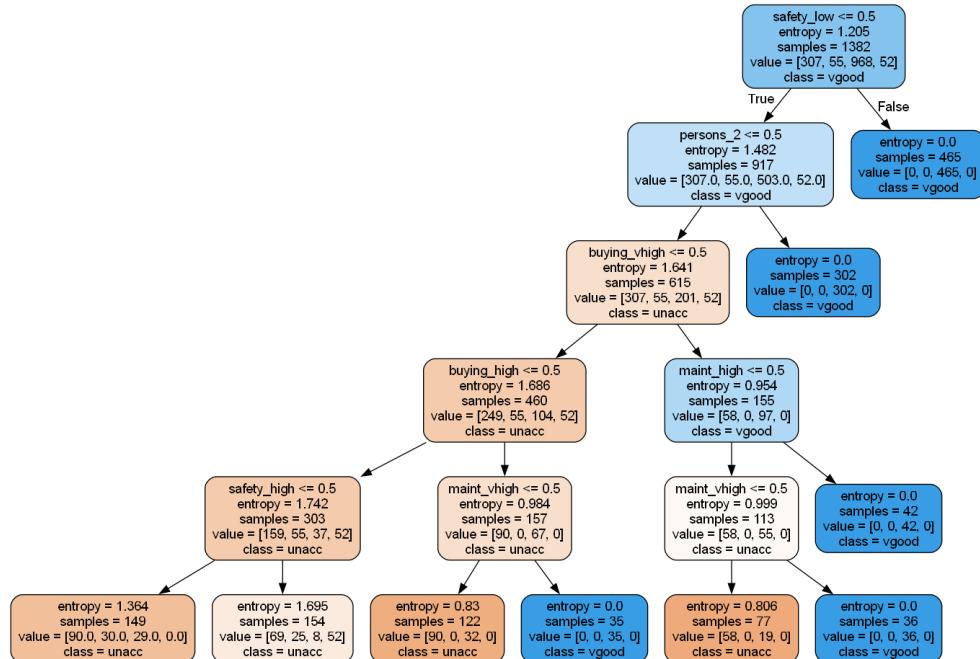


Figure 34: Car Evaluation: decision tree (base) for 80/20 split.

Figure 35: Car Evaluation: decision tree with `max_depth=2` (80/20 split).Figure 36: Car Evaluation: decision tree with `max_depth=3` (80/20 split).

Figure 37: Car Evaluation: decision tree with `max_depth=4` (80/20 split).Figure 38: Car Evaluation: decision tree with `max_depth=5` (80/20 split).

5. Dataset Analysis and Experiments

Project 03: Decision Tree

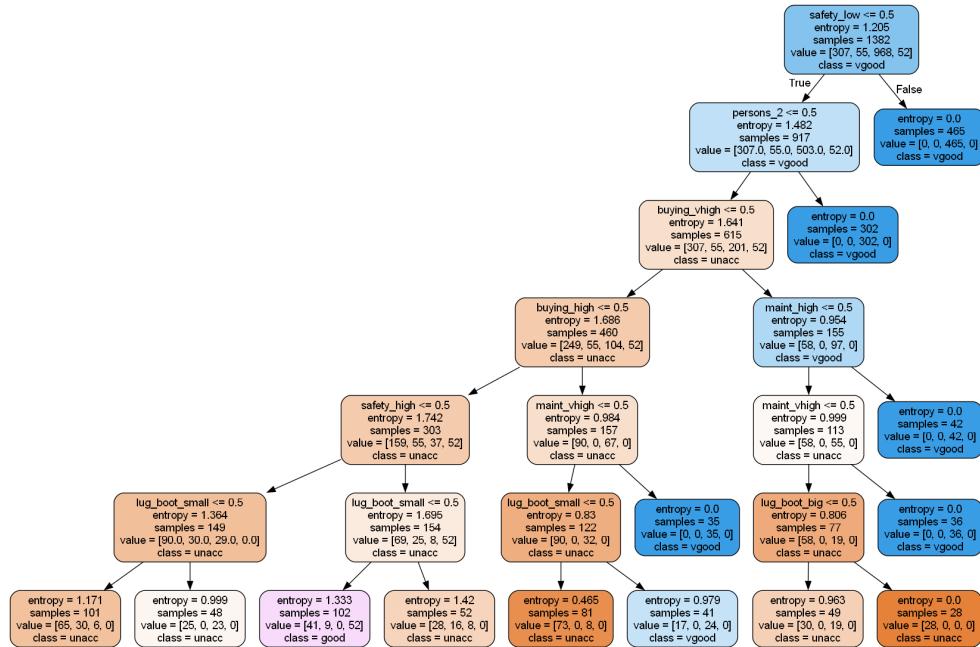


Figure 39: Car Evaluation: decision tree with `max_depth=6` (80/20 split).

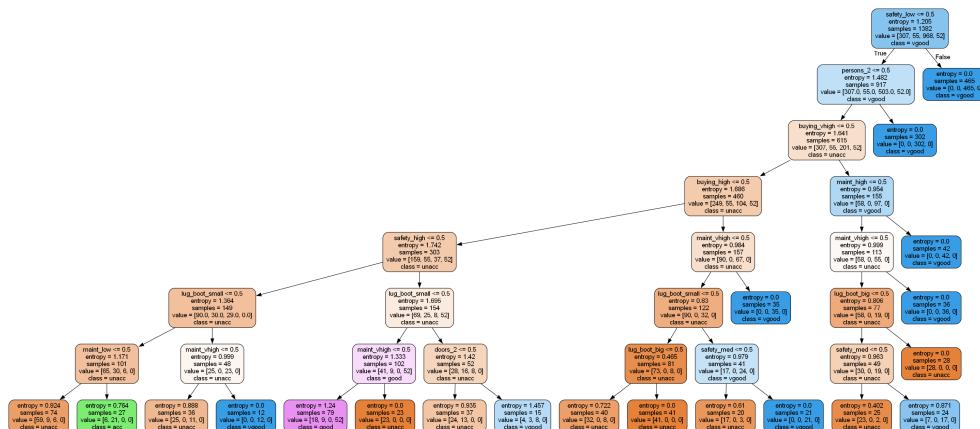


Figure 40: Car Evaluation: decision tree with `max_depth=7` (80/20 split).

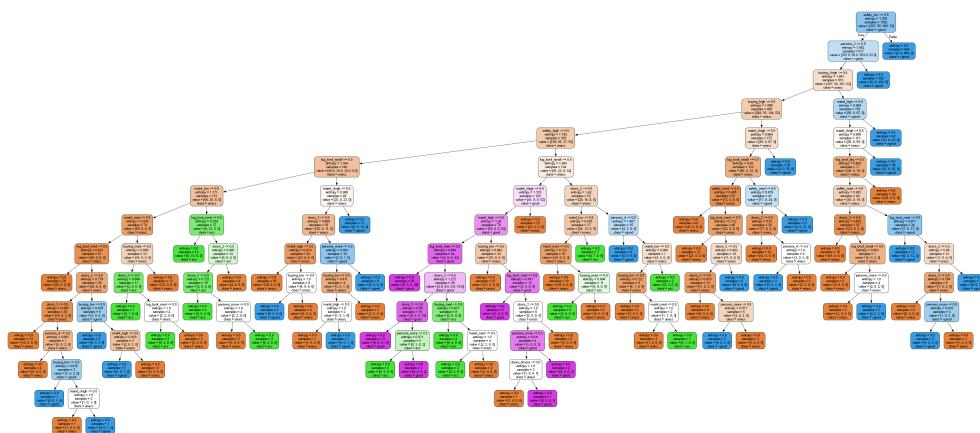


Figure 41: Car Evaluation: decision tree with `max_depth=None` (80/20 split).

6 Comparative Analysis

- **Objective:** Compare Decision Tree performance across:

1. Breast Cancer (binary classification, continuous features)
2. Wine Quality (multi-class, numerical features)
3. Car Evaluation (multi-class, categorical features)

- **Comparison Criteria:**

- Accuracy, Precision, Recall, F1-Score
- Effect of feature type and count
- Class distribution and balance
- Impact of `max_depth` on overfitting

- **Observations:**

- **Breast Cancer:** Highest accuracy; binary labels and well-separated numeric features helped model performance.
- **Wine Quality:** Lower precision for middle-quality wines; overlapping features across quality groups reduced clarity.
- **Car Evaluation:** Performed well despite 4 classes; decision tree easily handled categorical data. Slight overfitting observed at deep trees.

- **Conclusion:**

- Decision Trees adapt well to both categorical and numerical data, but class imbalance and feature overlap affect performance.
- Simpler datasets with clear boundaries (like Breast Cancer or Car Evaluation) yield higher accuracy.
- Proper depth tuning is essential to maintain generalization.

7 References

1. [DecisionTreeClassifier Documentation](#)