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

- **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

```python
def split_and_visualize(X, y, dataset_name: str):
    splits = {}

    for ratio in [0.4, 0.6, 0.8, 0.9]:
        X_train, X_test, y_train, y_test = train_test_split(
            X, y, train_size=ratio, stratify=y, shuffle=True, random_state=42
        )

        splits[ratio] = (X_train, X_test, y_train, y_test)

    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

```python
def train_evaluate_decision_tree(
    X_train,
    y_train,
    X_test,
    y_test,
    dataset_name: str,
    split_ratio,
):
    # Dynamically set feature and class names
    feature_names = X_train.columns.tolist()
    class_names = [str(cls) for cls in np.unique(y_train)]

    # Train model
    clf = DecisionTreeClassifier(criterion="entropy", random_state=42)
    clf.fit(X_train, y_train)

    # Predictions
    y_pred = clf.predict(X_test)

    # Classification Report (with validation)
    print(f"\nClassification Report ({dataset_name}, {display_ratio(split_ratio)}):")
```

```
    print(
        classification_report(
            y_test,
            y_pred,
            target_names=class_names,
            labels=np.unique(y_test),  # Ensure alignment with actual classes
        )
    )

    # Confusion Matrix
    sns.heatmap(
        confusion_matrix(y_test, y_pred),
        annot=True,
        fmt="d",
        xticklabels=class_names,
        yticklabels=class_names,
    )
```

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.
- The classification report summarizes, for each class c:
  - **Precision** (Prec_c): measures the fraction of samples predicted as c that truly belong to c.
  - **Recall** (Rec_c): measures the fraction of true-c samples correctly identified.
  - **F1-score** ($\mathrm{F1}_c$): the harmonic mean of precision and recall
  - **Support**: the number of true samples of class c.
- For example, with a binary problem (classes "positive"/"negative"), the confusion matrix is:

$$\mathbf{CM} = \begin{pmatrix} \mathrm{TN} & \mathrm{FP} \\ \mathrm{FN} & \mathrm{TP} \end{pmatrix},$$

  where

  TN **(True Negative):** correctly predicted negatives.

  FP **(False Positive, Type I error):** negatives incorrectly predicted as positives.

  FN **(False Negative, Type II error):** positives incorrectly predicted as negatives.

TP **(True Positive):** correctly predicted positives.

From these entries we derive:

$$\begin{aligned}
\text{Accuracy} &= \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}}, \\
\text{False Positive Rate (FPR)} &= \frac{\text{FP}}{\text{FP} + \text{TN}}, \\
\text{False Negative Rate (FNR)} &= \frac{\text{FN}}{\text{FN} + \text{TP}}
\end{aligned}$$

$$\begin{aligned}
\text{Specificity (True Negative Rate)} &= \frac{\text{TN}}{\text{TN} + \text{FP}}, \\
\text{Precision} &= \frac{\text{TP}}{\text{TP} + \text{FP}}, \\
\text{Recall (Sensitivity)} &= \frac{\text{TP}}{\text{TP} + \text{FN}}
\end{aligned}$$

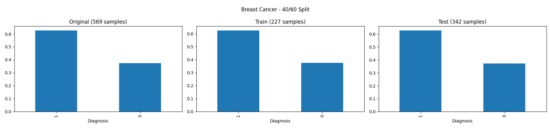
A well-performing classifier exhibits high TP and TN, and low FP and FN.

- *High FP* indicates many false alarms.
- *High FN* indicates many misses—critical in domains such as medical diagnosis.
- *High FP* indicates many false alarms ($\leq 15\%$).

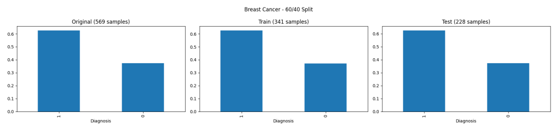
5.3 Breast Cancer Wisconsin Dataset

Dataset Description

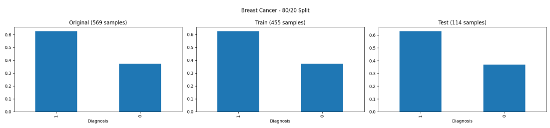
- **Description:** The UCI Breast Cancer Wisconsin (Diagnostic) dataset is used for classifying tumors as malignant or benign based on features derived from its imaging data.
- **Dataset Info:** 569 samples, binary labels (malignant vs. benign), 30 numeric features.
- **Preprocessing:** shuffle & stratified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



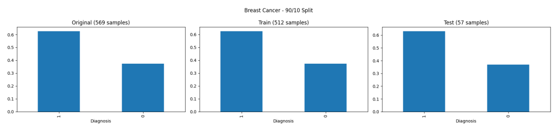
(a) Breast Cancer: class distribution (40/60 split).



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



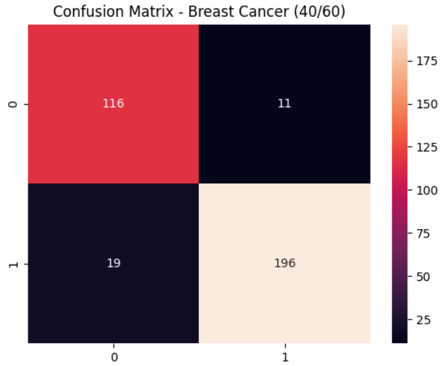
(c) Breast Cancer: class distribution (80/20 split).



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

Figure 1: Class distributions

Evaluating the decision tree classifiers



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

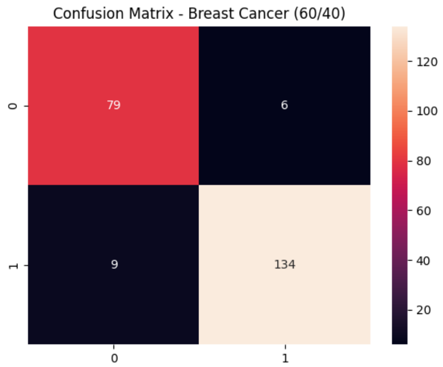
```
Classification Report (Breast Cancer, 40/60):
              precision    recall  f1-score   support

           0       0.86      0.91      0.89       127
           1       0.95      0.91      0.93       215

    accuracy                           0.91       342
   macro avg       0.90      0.91      0.91       342
weighted avg       0.91      0.91      0.91       342
```

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

Figure 2: Classification Report and Confusion Matrix (40/60 split)



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

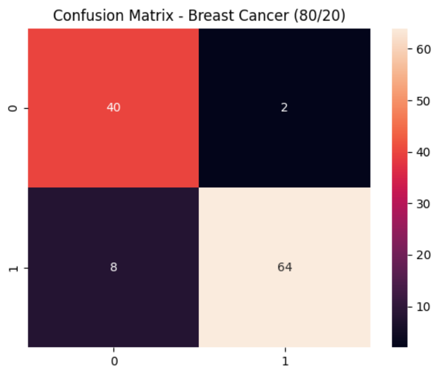
```
Classification Report (Breast Cancer, 60/40):
              precision    recall  f1-score   support

           0       0.90      0.93      0.91        85
           1       0.96      0.94      0.95       143

    accuracy                           0.93       228
   macro avg       0.93      0.93      0.93       228
weighted avg       0.93      0.93      0.93       228
```

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

Figure 3: Classification Report and Confusion Matrix (60/40 split)



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

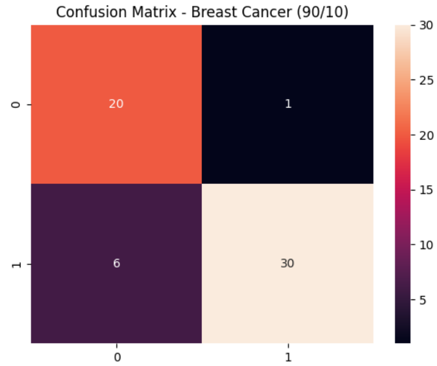
```
Classification Report (Breast Cancer, 80/20):
              precision    recall  f1-score   support

           0       0.83      0.95      0.89        42
           1       0.97      0.89      0.93        72

    accuracy                           0.91       114
   macro avg       0.90      0.92      0.91       114
weighted avg       0.92      0.91      0.91       114
```

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

Figure 4: Classification Report and Confusion Matrix (80/20 split)



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

Classification Report (Breast Cancer, 90/10):

	precision	recall	f1-score	support
0	0.77	0.95	0.85	21
1	0.97	0.83	0.90	36
accuracy			0.88	57
macro avg	0.87	0.89	0.87	57
weighted avg	0.89	0.88	0.88	57

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

Figure 5: Classification Report and Confusion Matrix (90/10 split)

Insights - Performance Evaluation

- **Accuracy by split ratio:**
  - 60/40 split achieved the highest test accuracy at **93%**.
  - 40/60 and 80/20 splits both reached **91%**.
  - 90/10 dropped to **88%**, showing increased variance with a very small test set.
- **Class-level performance:**
  - *Malignant (class 0):*
    * Precision ranged from 0.86 (40/60) → 0.90 (60/40) → 0.83 (80/20) → 0.77 (90/10).
    * Recall stayed high (0.91, 0.93, 0.95, 0.95), ensuring most malignant tumors are detected.
  - *Benign (class 1):*
    * Precision consistently excellent (0.95-0.97), meaning very few benign cases are mislabeled malignant.
    * Recall varied from 0.91 (40/60), 0.94 (60/40), 0.89 (80/20) to 0.83 (90/10), indicating some benign samples get misclassified when test size shrinks.
- **Macro vs. weighted F1:** Both averaged around 0.91 for splits $\geq$40/60, dropping slightly for 90/10—indicating stable balanced performance except with very few test examples.
- **Clinical implication:**
  - Keeping malignant recall $\geq$90% is critical to minimize missed cancer diagnoses.

- A benign precision $\geq$85% keeps false alarms at a manageable level in screening programs.

Decision Tree Classifier with Different Depths

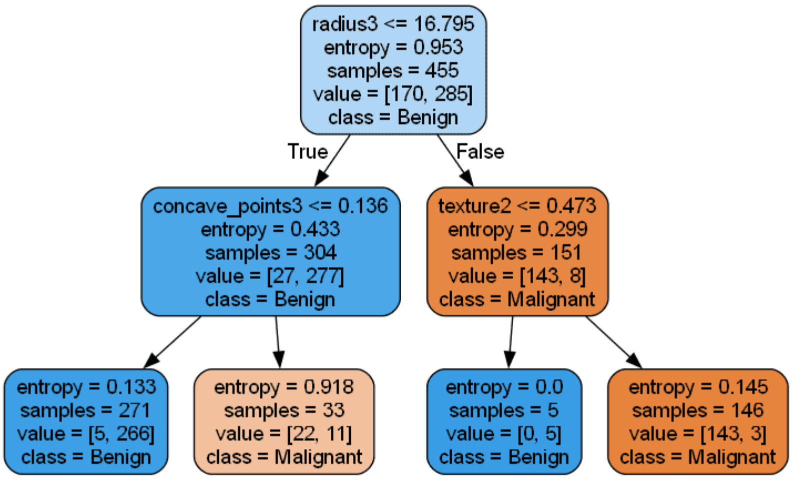


Figure 6: Breast Cancer: decision tree with `max_depth`=2 (80/20 split).

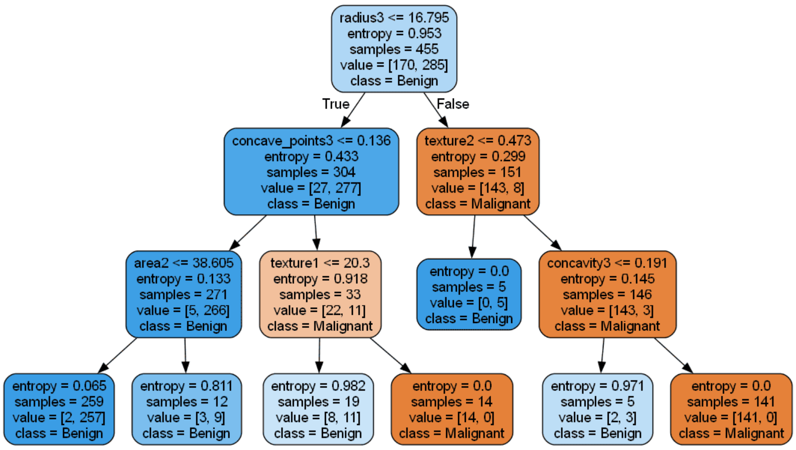


Figure 7: Breast Cancer: decision tree with `max_depth`=3 (80/20 split).

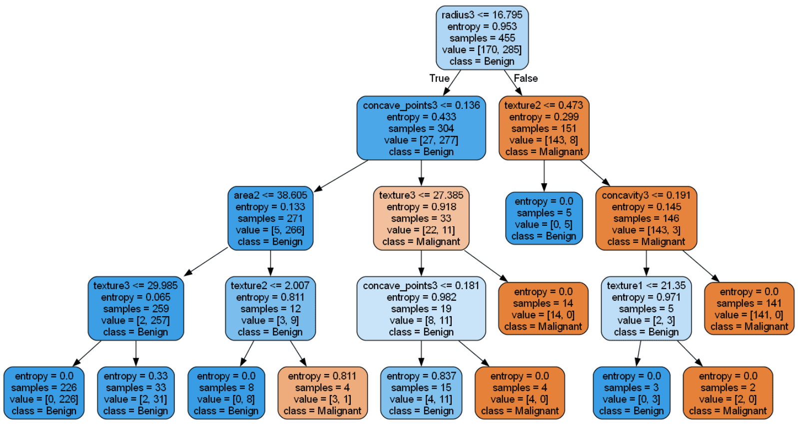


Figure 8: Breast Cancer: decision tree with `max_depth`=4 (80/20 split).

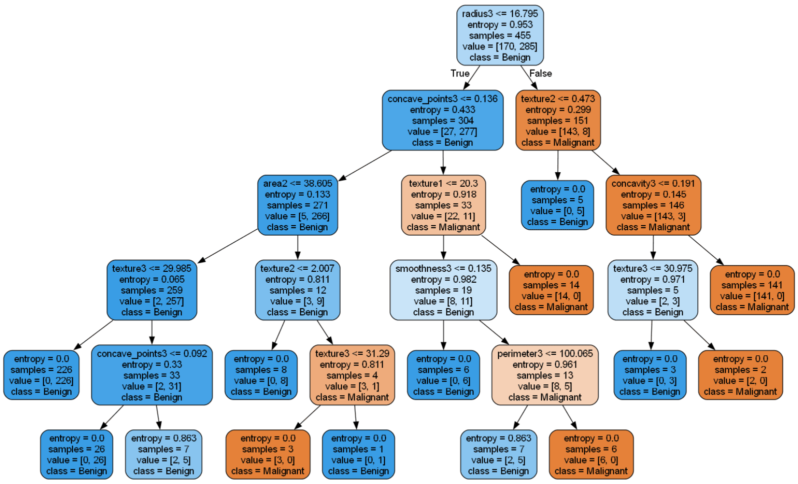


Figure 9: Breast Cancer: decision tree with `max_depth`=5 (80/20 split).

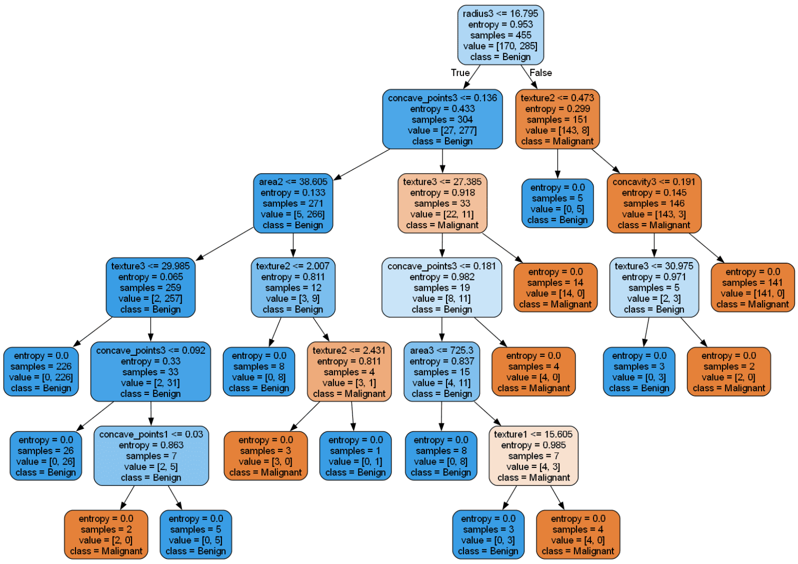


Figure 10: Breast Cancer: decision tree with `max_depth`=6 (80/20 split).

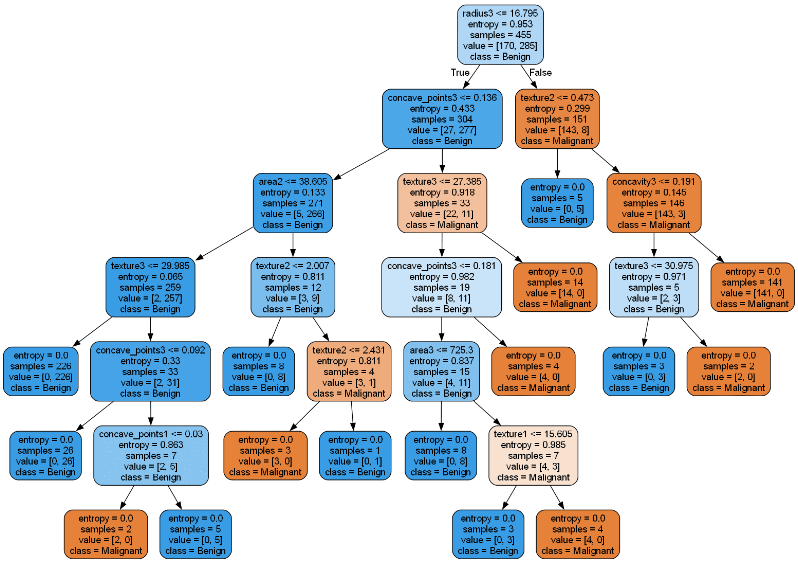


Figure 11: Breast Cancer: decision tree with `max_depth`=7 (80/20 split).

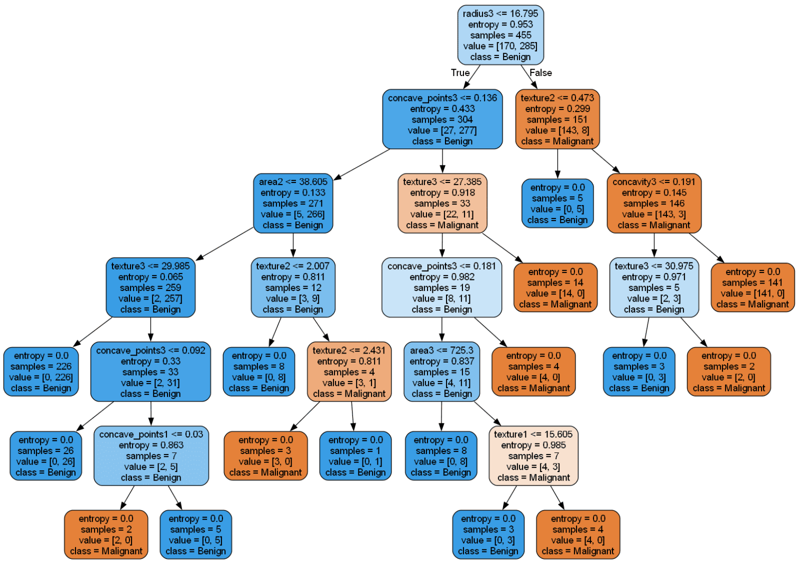
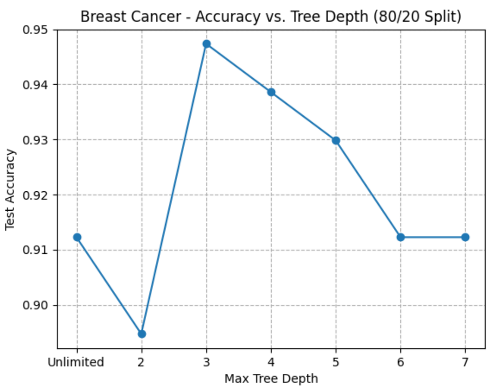


Figure 12: Breast Cancer: decision tree with `max_depth`=None (80/20 split).




```
Depth Analysis Results for Breast Cancer:
   Max Depth  Accuracy
0       None  0.912281
1          2  0.894737
2          3  0.947368
3          4  0.938596
4          5  0.929825
5          6  0.912281
6          7  0.912281
```

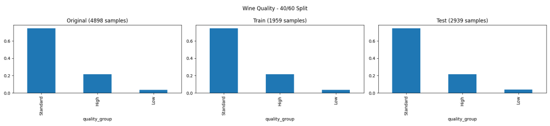
Insights - Depth and Accuracy

- **Underfitting at low depth:** `max_depth`=2 yields only **89.47%** accuracy—too shallow to capture key interactions.
- **Optimal depth = 3:**
  - Peaks at **94.74%** (∼5 pp gain over depth 2).
  - Balances bias/variance, capturing non-linear splits without overfitting.
- **Gradual overfitting beyond 3:**
  - Depth 4: 93.86%
  - Depth 5: 92.98%
  - Depth 6, 7, None: all 91.23%, matching the very deep tree but with far more complexity.
- **Interpretability trade-off:** A 3-level tree has under 10 nodes—easy to explain—while delivering maximum generalization.
- **Recommendation:** Limit `max_depth` to **3-4** for this dataset to sustain high accuracy, control overfitting, and preserve model simplicity.

5.4 Wine Quality Dataset

Dataset Description

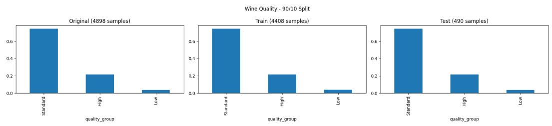
- **Description:** The UCI Wine Quality dataset is used for classifying wine samples into quality levels based on physicochemical properties such as acidity, alcohol content, etc.
- **Dataset Info:** 4898 samples, with labels from 0 (low quality) to 10 (high quality).
- **Preprocessing:** shuffle & stratified split at 40/60, 60/40, 80/20, 90/10.



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

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

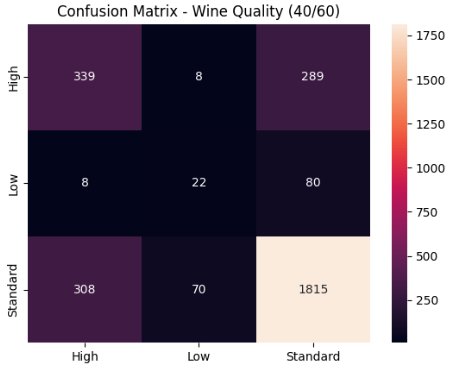
(c) Wine Quality: class distribution (80/20 split).



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

Figure 14: Class distributions

Evaluating the decision tree classifiers



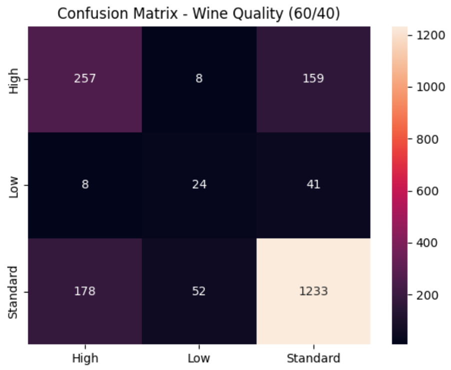
(a) Wine Quality: confusion matrix (40/60 split).

Classification Report (Wine Quality, 40/60):

	precision	recall	f1-score	support
High	0.52	0.53	0.53	636
Low	0.22	0.20	0.21	110
Standard	0.83	0.83	0.83	2193
accuracy			0.74	2939
macro avg	0.52	0.52	0.52	2939
weighted avg	0.74	0.74	0.74	2939

(b) Wine Quality: Classification Report (40/60 split).

Figure 15: Classification Report and Confusion Matrix (40/60 split)



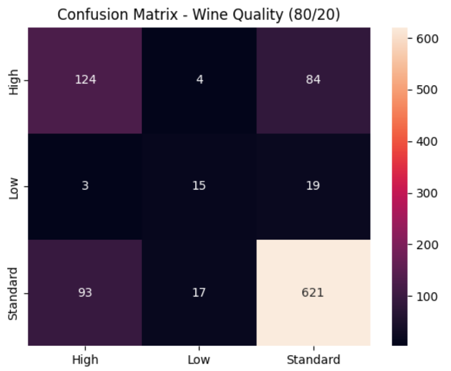
(a) Wine Quality: confusion matrix (60/40 split).

Classification Report (Wine Quality, 60/40):

	precision	recall	f1-score	support
High	0.58	0.61	0.59	424
Low	0.29	0.33	0.31	73
Standard	0.86	0.84	0.85	1463
accuracy			0.77	1960
macro avg	0.58	0.59	0.58	1960
weighted avg	0.78	0.77	0.78	1960

(b) Wine Quality: Classification Report (60/40 split).

Figure 16: Classification Report and Confusion Matrix (60/40 split)



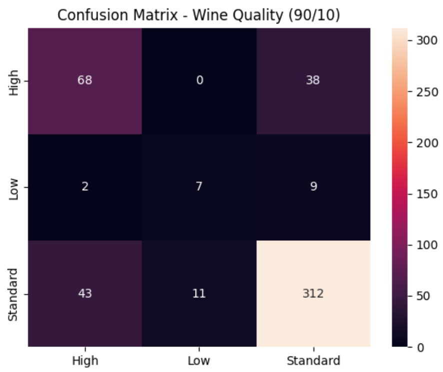
(a) Wine Quality: confusion matrix (80/20 split).

Classification Report (Wine Quality, 80/20):

	precision	recall	f1-score	support
High	0.56	0.58	0.57	212
Low	0.42	0.41	0.41	37
Standard	0.86	0.85	0.85	731
accuracy			0.78	980
macro avg	0.61	0.61	0.61	980
weighted avg	0.78	0.78	0.78	980

(b) Wine Quality: Classification Report (80/20 split).

Figure 17: Classification Report and Confusion Matrix (80/20 split)



(a) Wine Quality: confusion matrix (90/10 split).

Classification Report (Wine Quality, 90/10):

	precision	recall	f1-score	support
High	0.60	0.64	0.62	106
Low	0.39	0.39	0.39	18
Standard	0.87	0.85	0.86	366
accuracy			0.79	490
macro avg	0.62	0.63	0.62	490
weighted avg	0.79	0.79	0.79	490

(b) Wine Quality: Classification Report (90/10 split).

Figure 18: Classification Report and Confusion Matrix (90/10 split)

Insights - Performance Evaluation

- **Overall accuracy trend:**
  - Rises steadily from **74%** (40/60) → **77%** (60/40) → **78%** (80/20) → **79%** (90/10).
  - Larger training sets consistently improve generalization.
- **Class-level performance:**
  - *Standard (majority) class:*
    - Precision/recall ≈ 0.83 at 40/60, rising to *approx*0.87 at 90/10.
    - High support (2,193→366) yields consistently strong F1-scores (0.83 → 0.86).
  - *High quality:*
    - Precision improves from 0.52 → 0.60; recall from 0.53 → 0.64 as training size grows.
    - Indicates better detection of top-tier wines with more data.
  - *Low quality:*
    - Lowest metrics: precision 0.22 → 0.39, recall 0.20 → 0.39 across splits.
    - Small support (110→18) makes "Low" wines hardest to classify.
- **Macro vs. weighted averages:**
  - Macro-avg F1 climbs from 0.52 → 0.62, reflecting improvement on minority classes.
  - Weighted-avg F1 follows overall accuracy closely (0.74 → 0.79).

- **Class imbalance impact:** The dominant "Standard" category ($\approx 70\%$ of samples) drives overall accuracy; minority classes require targeted strategies (e.g. class weighting) for balanced performance.

Decision Tree Classifier with Different Depths

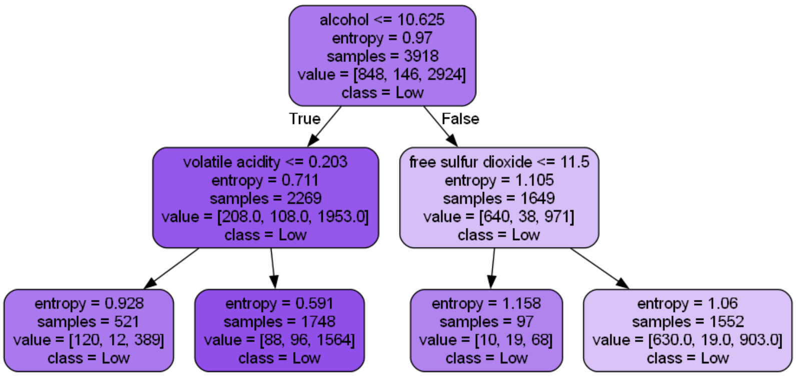


Figure 19: Wine Quality: decision tree with `max_depth`=2 (80/20 split).

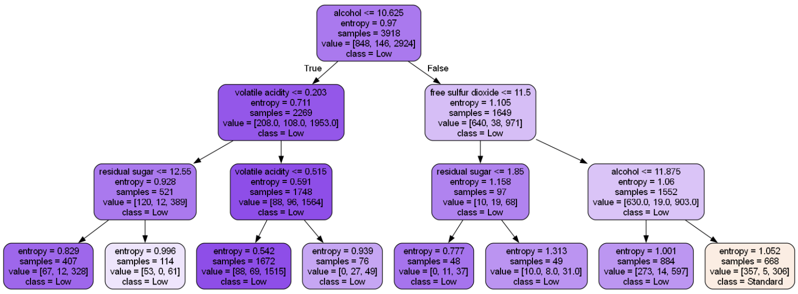


Figure 20: Wine Quality: decision tree with `max_depth`=3 (80/20 split).

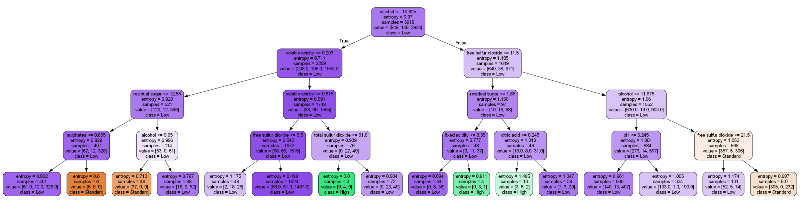


Figure 21: Wine Quality: decision tree with `max_depth`=4 (80/20 split).

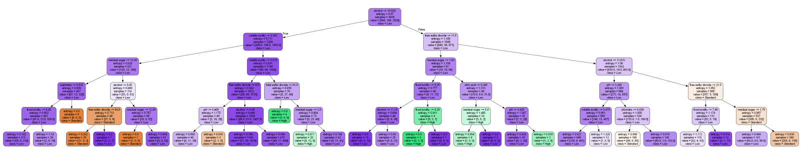


Figure 22: Wine Quality: decision tree with `max_depth`=5 (80/20 split).

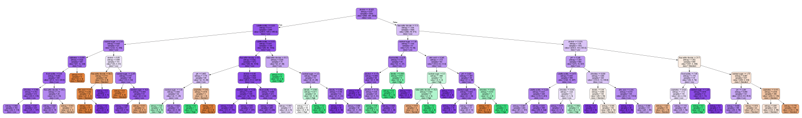


Figure 23: Wine Quality: decision tree with `max_depth`=6 (80/20 split).

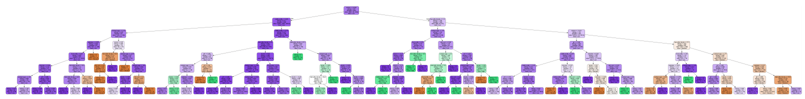
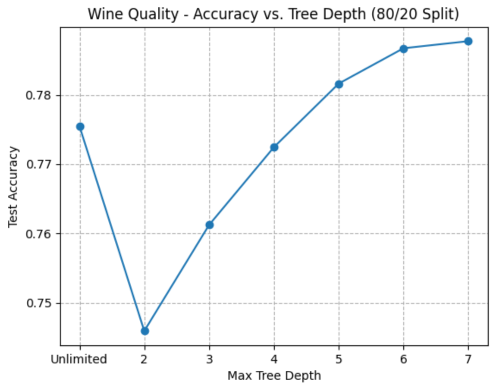


Figure 24: Wine Quality: decision tree with `max_depth`=7 (80/20 split).



Depth Analysis Results for Wine Quality:

	Max Depth	Accuracy
0	None	0.775510
1	2	0.745918
2	3	0.761224
3	4	0.772449
4	5	0.781633
5	6	0.786735
6	7	0.787755

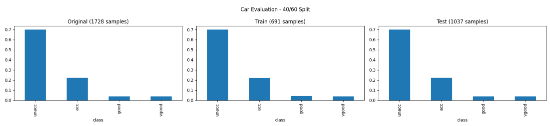
Insights - Depth and Accuracy

- **Underfitting at low depth:** `max_depth`=2 yields only **74.6%** accuracy, failing to capture interactions among acidity, alcohol, and sulphates.
- **Steady gains with depth:**
  - Depth 3 → **76.0%** (+1.5 pp),
  - Depth 4 → **77.0%** (+1.1 pp),
  - Depth 5 → **78.2%** (+0.9 pp).
  - Depth 6 → **78.7%** (+0.6 pp),
  - Depth 7 → **78.8%** (+0.1 pp).
- **Unrestricted tree underperforms:** The fully grown tree (None) reaches **77.6%**, below depth 7, indicating pruning aids generalization.
- **Optimal depth range:** Depth 5-7 balances complexity and predictive power; marginal gains beyond depth 6 suggest diminishing returns.
- **Practical recommendation:** For multi-class wine quality prediction, set `max_depth` around **6** to maximize test accuracy while keeping the tree interpretable.

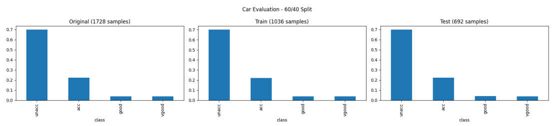
5.5 Car Evaluation Dataset

Dataset Description

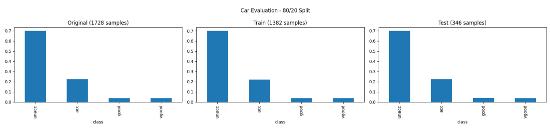
- **Description:** Car Evaluation Database was derived from a simple hierarchical decision model originally developed for the demonstration of DEX, M. Bohanec, V. Rajkovic: Expert system for decision making.
- **Dataset Info:** 1728 samples, 4 classes (unacc, acc, good, vgood), 6 categorical attributes (buying price, maintenance cost, doors, etc.).
- **Preprocessing:** shuffle & stratified split at 40/60, 60/40, 80/20, 90/10.



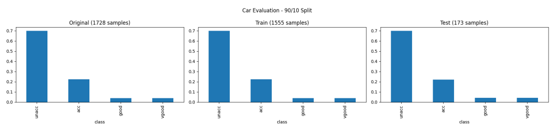
(a) Car Evaluation: class distribution (40/60 split).



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



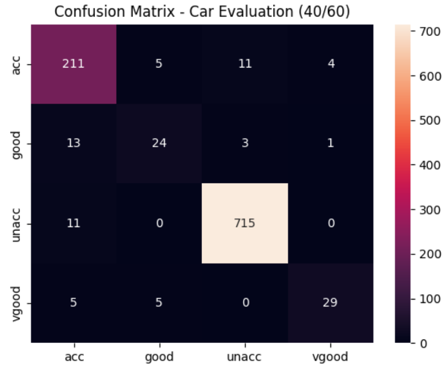
(c) Car Evaluation: class distribution (80/20 split).



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

Figure 26: Class distributions

Evaluating the decision tree classifiers



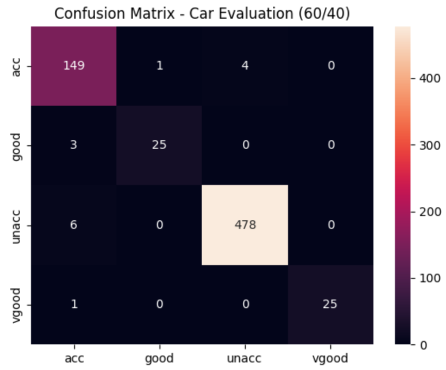
(a) Car Evaluation: confusion matrix (40/60 split).

Classification Report (Car Evaluation, 40/60):

	precision	recall	f1-score	support
acc	0.88	0.91	0.90	231
good	0.71	0.59	0.64	41
unacc	0.98	0.98	0.98	726
vgood	0.85	0.74	0.79	39
accuracy			0.94	1037
macro avg	0.85	0.81	0.83	1037
weighted avg	0.94	0.94	0.94	1037

(b) Car Evaluation: Classification Report (40/60 split).

Figure 27: Classification Report and Confusion Matrix (40/60 split)



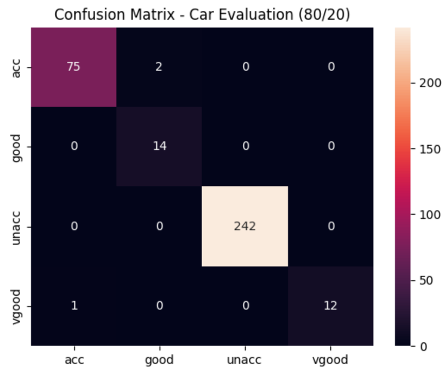
(a) Car Evaluation: confusion matrix (60/40 split).

Classification Report (Car Evaluation, 60/40):

	precision	recall	f1-score	support
acc	0.94	0.97	0.95	154
good	0.96	0.89	0.93	28
unacc	0.99	0.99	0.99	484
vgood	1.00	0.96	0.98	26
accuracy			0.98	692
macro avg	0.97	0.95	0.96	692
weighted avg	0.98	0.98	0.98	692

(b) Car Evaluation: Classification Report (60/40 split).

Figure 28: Classification Report and Confusion Matrix (60/40 split)



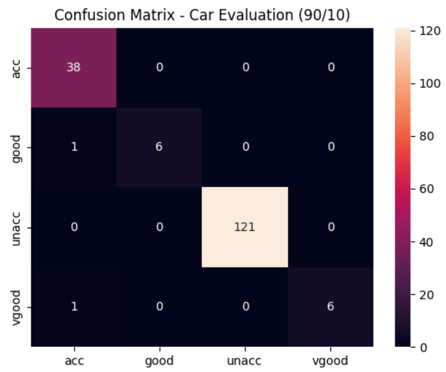
(a) Car Evaluation: confusion matrix (80/20 split).

Classification Report (Car Evaluation, 80/20):

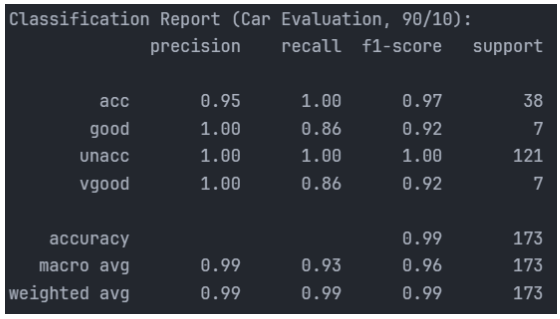
	precision	recall	f1-score	support
acc	0.99	0.97	0.98	77
good	0.88	1.00	0.93	14
unacc	1.00	1.00	1.00	242
vgood	1.00	0.92	0.96	13
accuracy			0.99	346
macro avg	0.97	0.97	0.97	346
weighted avg	0.99	0.99	0.99	346

(b) Car Evaluation: Classification Report (80/20 split).

Figure 29: Classification Report and Confusion Matrix (80/20 split)



(a) Car Evaluation: confusion matrix (90/10 split).



Classification Report (Car Evaluation, 90/10):

	precision	recall	f1-score	support
acc	0.95	1.00	0.97	38
good	1.00	0.86	0.92	7
unacc	1.00	1.00	1.00	121
vgood	1.00	0.86	0.92	7
accuracy			0.99	173
macro avg	0.99	0.93	0.96	173
weighted avg	0.99	0.99	0.99	173

(b) Car Evaluation: Classification Report (90/10 split).

Figure 30: Classification Report and Confusion Matrix (90/10 split)

Insights - Performance Evaluation

- **Accuracy by split ratio:**
  - 40/60 split: **94%**
  - 60/40 split: **98%**
  - 80/20 split: **99%**
  - 90/10 split: **99%**

  Accuracy rises sharply as the training set grows, peaking at 99% when $\geq 80\%$ of data is used for training.

- **Class-level performance:**
  - *unacc (majority):* Precision and recall $\geq 98\%$ across all splits, reflecting the model's ease in identifying unacceptable cars.
  - *acc:* Precision climbs from 88% $\rightarrow$ 99%, recall from 91% $\rightarrow$ 100% as training size increases, showing strong learning of the "acceptable" class.
  - *good:*
    * 40/60 split: Precision 71%, Recall 59% (support 41)
    * 60/40 split: Precision 96%, Recall 89% (support 28)
    * 80/20 split: Precision 88%, Recall 100% (support 14)

    Smaller support for "good" leads to greater variance in its metrics.
  - *vgood (minority):*
    * Precision: 85-100%
    * Recall by split:

  - · 40/60: 74%
  - · 60/40: 96%
  - · 80/20: 92%
  - · 90/10: 86%

  The recall swings reflect the model's difficulty detecting very-good cars when examples are scarce.

- **Macro vs. weighted F1:**
  - Macro-avg F1 improves from 83% (40/60) → 96% (60/40) → 97% (80/20) → 96% (90/10).
  - Weighted-avg F1 mirrors accuracy (94-99%), driven by the dominant "unacc" class.

- **Class imbalance impact:** The "unacc" class (≈ 70% of samples) drives most of the overall accuracy. Minority classes ("good", "vgood") benefit greatly from more training data.

Decision Tree Classifier with Different Depths

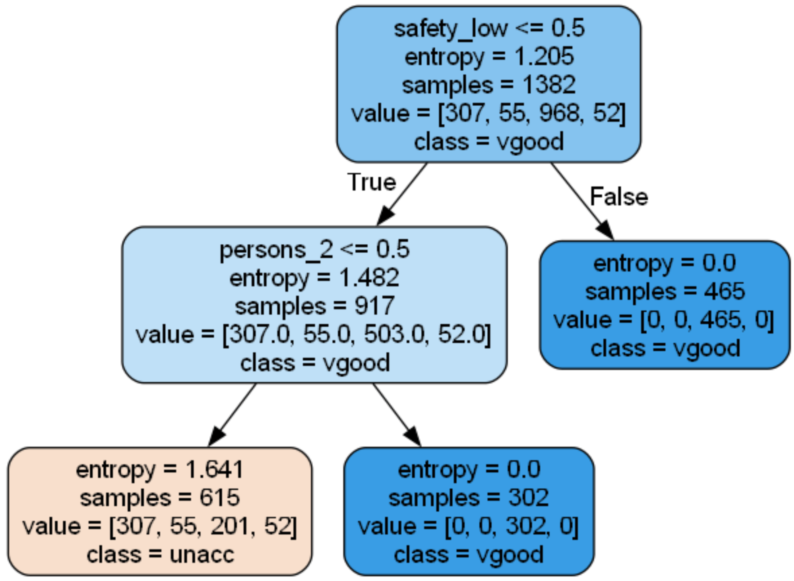


Figure 31: Car Evaluation: decision tree with `max_depth`=2 (80/20 split).

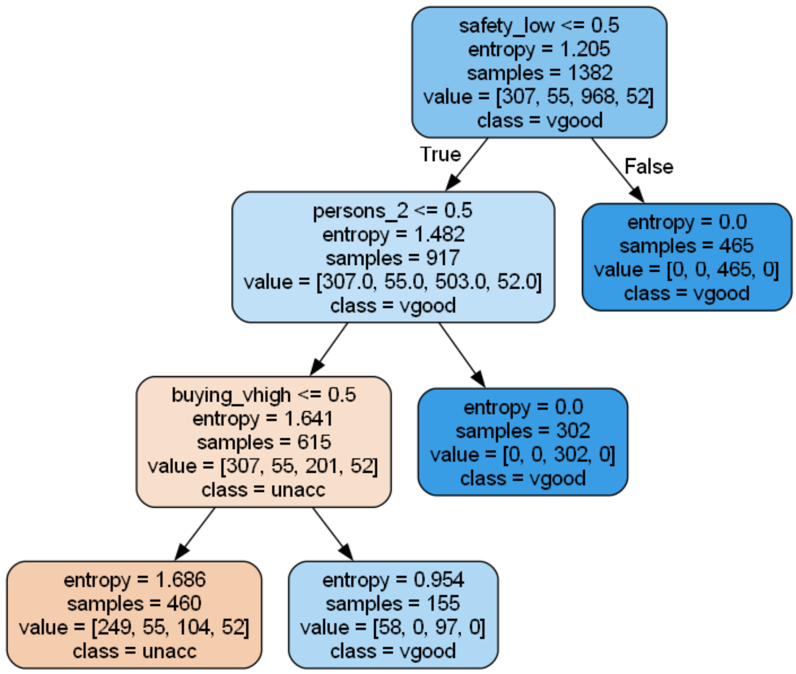


Figure 32: Car Evaluation: decision tree with `max_depth`=3 (80/20 split).

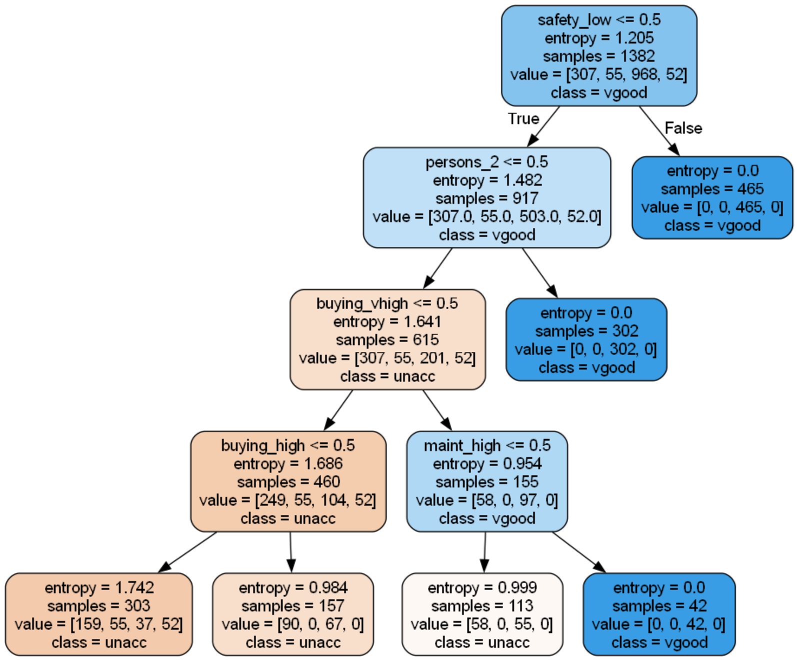


Figure 33: Car Evaluation: decision tree with `max_depth`=4 (80/20 split).

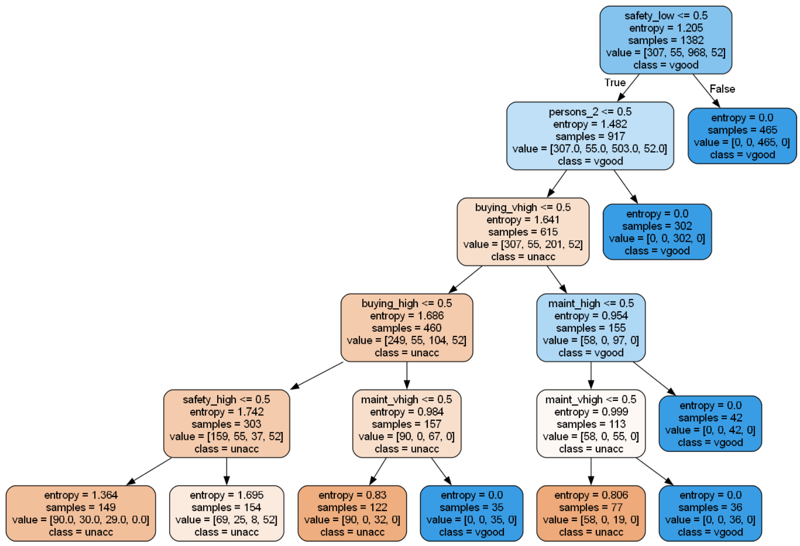


Figure 34: Car Evaluation: decision tree with `max_depth`=5 (80/20 split).

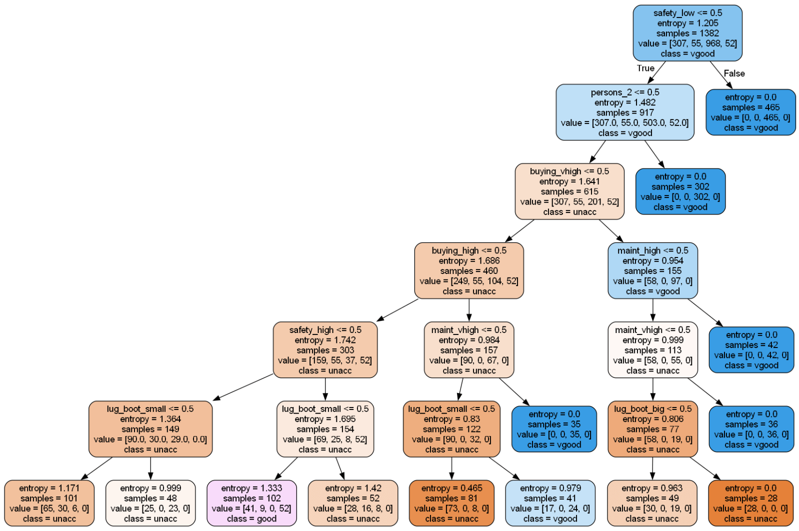


Figure 35: Car Evaluation: decision tree with `max_depth`=6 (80/20 split).

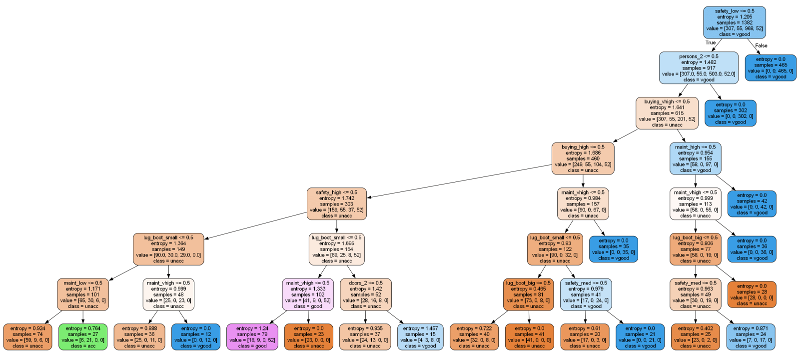


Figure 36: Car Evaluation: decision tree with `max_depth`=7 (80/20 split).

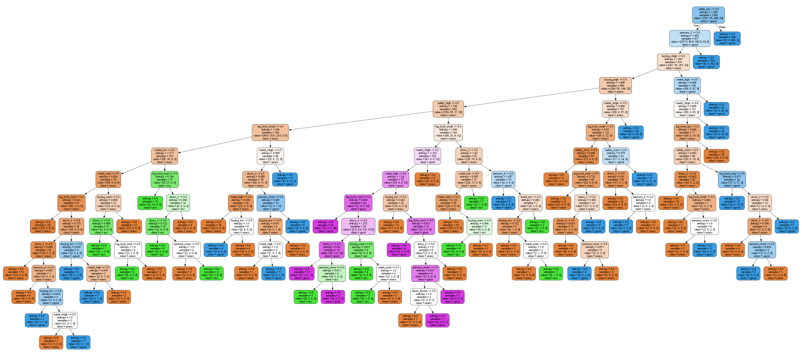
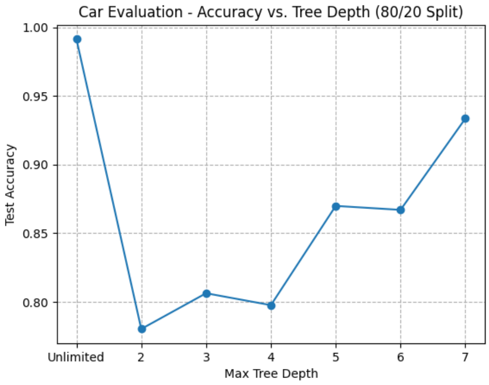


Figure 37: Car Evaluation: decision tree with `max_depth`=None (80/20 split).




```
Depth Analysis Results for Car Evaluation:
   Max Depth  Accuracy
0       None  0.991329
1          2  0.780347
2          3  0.806358
3          4  0.797688
4          5  0.869942
5          6  0.867052
6          7  0.933526
```

Insights - Depth and Accuracy

- **Underfitting at shallow depths:**
  - `max_depth`=2: **78.0%**
  - `max_depth`=3: **80.6%**
  - `max_depth`=4: **79.8%**

  Very low depths cannot capture the multi-attribute categorical splits.

- **Rapid gains with moderate depth:**
  - Depth 5: **87.0%** (+8.2 pp over depth 4)
  - Depth 6: **86.7%**
  - Depth 7: **93.4%** (+6.7 pp over depth 6)

  Indicates that deeper trees are needed to model complex combinations of the six categorical features.

- **Unrestricted tree (None): 99.1%**—the highest accuracy, achieving nearly perfect classification by fully expanding on all splits.

- **Interpretability vs. performance:**
  - Limiting to depth 7 yields 93.4% accuracy with a still-manageable tree size.
  - Allowing no limit pushes accuracy to 99.1% at the cost of a very large, less interpretable tree.

- **Recommendation:**
  - If maximum accuracy is required, use unrestricted depth.
  - For a balance of interpretability and high performance, cap `max_depth` at **7**.

6 Comparative Analysis

After evaluating decision tree performance on all three datasets, we examine how key dataset characteristics—number of classes, number of features, and sample size—influence model metrics (accuracy, precision, recall, F_1-score).

6.1 Summary of Best Test Accuracies

Dataset	Classes	Features	Best Accuracy (%)
Breast Cancer Wisconsin	2	30	94.74 (depth=3)
Wine Quality	3	11	78.78 (depth=7)
Car Evaluation	4	6	99.13 (depth=None)

6.2 Impact of Number of Classes

- **Binary vs. multi-class:** The binary Breast Cancer dataset achieved the highest accuracy (94.74%); only two labels result in simpler decision boundaries.
- **Three classes (Wine Quality):** Introducing "Low," "Standard," and "High" quality classes reduced peak accuracy by ≈ 16 pp compared to binary, due to overlapping physicochemical profiles.
- **Four classes (Car Evaluation):** Despite four target categories, Car Evaluation attained 99.13% because its categorical attributes produce very clear splits for each class.
- **Lesson:** Accuracy generally decreases as class count increases, but well-separable features can offset this effect.

6.3 Impact of Number of Features

- **High dimensionality (30 features):** Breast Cancer's 30 numeric features provided rich discriminative power; however, deeper trees began to overfit on less informative dimensions.
- **Medium dimensionality (11 features):** Wine Quality's 11 continuous features sufficed to reach ~79% accuracy but required deeper trees (depth 7) to capture complex interactions (e.g. acidity vs. alcohol).
- **Low dimensionality (6 features):** Car Evaluation's six categorical attributes yielded $> 99\%$ accuracy, showing that a small number of highly informative categorical features can outperform larger numeric feature sets.
- **Lesson:** More features improve performance only if they carry meaningful signal; high-level categorical features may be more effective than many noisy numeric ones.

6.4 Impact of Sample Size

- **Small sample (569):** Breast Cancer's smaller size produced high variance at extreme splits (e.g. 90/10), but stratified sampling kept accuracy within 2-3 pp across ratios.

- **Large sample (4,898):** Wine Quality's larger sample stabilized performance: accuracy varied by only ∼5 pp between smallest and largest training sets.
- **Medium sample (1,728):** Car Evaluation sat between the two: accuracy varied < 5 pp across splits, indicating diminishing returns once a "critical mass" of examples is reached.
- **Lesson:** Larger sample sizes increase metric stability and reduce sensitivity to train/test ratio, but only until class overlap—not data scarcity—becomes the limiting factor.

6.5 Overall Recommendations

- For **binary problems** with many numeric features, limit tree depth to 3-4 to maximize generalization and interpretability.
- For **multi-class problems**, ensure features are well separated or apply feature engineering before using deep trees.
- When using **categorical features**, even low-dimensional datasets can achieve high accuracy with moderate depth.
- Always **tune `max_depth`** via cross-validation: our experiments found optimal depths of 3 (Breast Cancer), 7 (Wine Quality), and unlimited (Car Evaluation), balancing bias-variance and model complexity.

7 References

1. DecisionTreeClassifier Documentation