

DECISION TREE CLASSIFICATION – HEART DISEASE DATASET

Assignment 13

OBJECTIVE

The objective of this assignment is to apply Decision Tree Classification on the Heart Disease dataset, perform exploratory data analysis, preprocess the data, implement the model, perform hyperparameter tuning, evaluate its performance, and interpret the results.

DATA PREPARATION

The dataset was loaded into Python using the Pandas library.

Important Correction:

The Excel workbook contains metadata in the first sheet and the actual dataset in the second sheet. By default, `pandas.read_excel()` loads the first sheet. Therefore, to correctly load the dataset, the `sheet_name` parameter was used.

```
df = pd.read_excel("heart_disease.xlsx", sheet_name=1)
```

Using `sheet_name=1` ensures that the second sheet (which contains the actual dataset) is loaded properly.

The dataset contains the following medical attributes:

- age
- sex
- cp (chest pain type)
- trestbps (resting blood pressure)
- chol (cholesterol)
- fbs (fasting blood sugar)
- restecg (resting ECG results)
- thalach (maximum heart rate achieved)
- exang (exercise induced angina)
- oldpeak
- slope
- thal
- num (target variable – presence of heart disease)

The following preprocessing steps were performed:

- Loaded dataset correctly from second sheet
- Checked structure using `head()`, `info()`, `describe()`
- Verified missing values using `isnull().sum()`

- Removed missing records using dropna()
- Encoded categorical variables using Label Encoding
- Feature scaling was not applied because Decision Trees are not sensitive to scaling

Target Variable:

num → Indicates presence or absence of heart disease.

EXPLORATORY DATA ANALYSIS (EDA)

EDA was performed to understand data structure and relationships.

The following steps were conducted:

- Checked data types of all features
- Verified missing values
- Observed statistical summary using describe()
- Visualized feature distributions using histograms
- Generated correlation matrix

Observations:

- Age, cholesterol, and maximum heart rate show variation among patients.
 - Some features show moderate correlation with heart disease presence.
 - The dataset is suitable for classification modeling.
-

FEATURE ENGINEERING

The following preprocessing steps were applied:

- Boolean columns converted to integer values
- Categorical variables encoded using Label Encoding
- No feature scaling was applied (Decision Trees are scale-invariant)

Features (X) → All columns except "num"

Target (y) → "num"

DECISION TREE CLASSIFICATION

The dataset was split using 80-20 train-test split:

```
X_train, X_test, y_train, y_test = train_test_split(
    X, y, test_size=0.2, random_state=42
)
```

A Decision Tree model was implemented:

```
dt = DecisionTreeClassifier(random_state=42)
dt.fit(X_train, y_train)
```

Predictions were made on the test set.

MODEL EVALUATION

Evaluation Metrics Used:

- Accuracy
- Precision
- Recall
- F1-score
- ROC-AUC Score

For multi-class classification, ROC-AUC requires probability scores and specification of `multi_class` parameter.

Correct implementation:

```
y_pred = dt.predict(X_test)
y_prob = dt.predict_proba(X_test)

roc_auc_score(y_test, y_prob, multi_class='ovr')
```

The `multi_class='ovr'` (One-vs-Rest) method was used to correctly compute ROC-AUC for multi-class classification.

Results Interpretation:

Accuracy → Overall correct predictions

Precision → Correctness of positive predictions

Recall → Ability to identify actual positive cases

F1-score → Balance between precision and recall

ROC-AUC → Model's ability to distinguish between classes

HYPERPARAMETER TUNING

To reduce overfitting and improve generalization, GridSearchCV was used.

Parameters tuned:

- `max_depth`
- `min_samples_split`
- `criterion` (gini, entropy)

GridSearchCV with 5-fold cross-validation was applied to select optimal parameters.

After tuning:

- Model generalization improved
 - Overfitting reduced
 - Performance metrics improved
-

MODEL ANALYSIS

The Decision Tree structure was visualized using `plot_tree()`.

Important features contributing to prediction include:

- Age
- Chest pain type
- Cholesterol
- Maximum heart rate

Decision Trees provide clear rule-based splits, making them interpretable for medical decision-making.

INTERVIEW QUESTIONS

1. What are common hyperparameters of Decision Tree models?

`max_depth`

Controls tree depth. Large depth may cause overfitting.

`min_samples_split`

Minimum samples required to split a node.

`min_samples_leaf`

Minimum samples required at leaf node.

`criterion`

Split quality measure (gini or entropy).

These parameters help balance bias and variance.

2. Difference between Label Encoding and One-Hot Encoding?

Label Encoding:

- Assigns integer values
- Suitable for ordinal variables
- May introduce false order

One-Hot Encoding:

- Creates binary columns
 - Suitable for nominal variables
 - Prevents ordinal issues
-

CONCLUSION

In this assignment:

- The dataset was correctly loaded from the second sheet using `sheet_name=1`.
- Data preprocessing and encoding were performed properly.
- Decision Tree model was implemented successfully.
- Hyperparameter tuning improved performance.
- ROC-AUC was correctly calculated using `multi_class='ovr'`.

The Decision Tree model provides interpretable and effective classification for predicting heart disease.