

AI Project Phase 2 Requirements

Load Dataset and Train/Test Split

1. **Load the Dataset:**
 - Choose a tabular classification dataset with 10 or more features.
2. **Split the Data:**
 - Separate the dataset into:
 - **Training Set:** For model training and validation.
 - **Test Set:** For final evaluation.
 - Use `train_test_split` from `sklearn.model_selection`.

Clean Missing Values and Remove Duplicates

1. **Handle Missing Values:**
 - Drop rows with missing values or use imputation techniques (e.g., mean, median, or mode).
2. **Verify Data Integrity:**
 - Check for duplicates and remove them if necessary.

Exploratory Plots (on the Training Set)

1. **Initial Exploration:**
 - Use `pandas` and `matplotlib / seaborn` to visualize distributions of features in the training set.
 - Identify potential outliers and trends.

Scale Numeric Features

1. **Fit the Scaler:**
 - Apply **Min–Max scaling** using `sklearn.preprocessing.MinMaxScaler` on the training set.
2. **Transform the Data:**
 - Use the fitted scaler to transform both the training and test sets.

Create Fuzzy Features (Based on Train-Set Statistics)

1. **Define Fuzzy Membership Functions:**
 - Select 2–3 numeric features from the training set.
 - Define simple fuzzy membership functions (e.g., triangular or trapezoidal) to represent categories such as “low,” “medium,” and “high.”
2. **Compute Membership Degrees:**
 - Use the defined membership functions to calculate the degree of membership for each data point in the selected features.
 - Append these membership degrees as new columns to the feature matrix `X`.

Train Decision Tree and Tune via Hill-Climbing

1. **Train the Model:**
 - Use `sklearn.tree.DecisionTreeClassifier` to train the model on the scaled and fuzzy-augmented training set.
2. **Hill-Climbing Search:**
 - Optimize a hyperparameter (e.g., `max_depth`) by iteratively evaluating neighboring values and moving to the one with the highest validation accuracy.
 - Stop when no further improvement is observed.

Compare to Brute-Force Grid Search

1. Define the Search Space:

Specify a range of values for the hyperparameter (e.g., `max_depth` from 1 to 10).
2. Grid Search:

Exhaustively evaluate all possible values within the range.
3. Comparison Metrics:

Search Path: Sequence of `max_depth` values evaluated.

Efficiency: Total number of model evaluations required.

Performance: Final validation accuracy achieved.

Visualize the Final Tree

- Use `sklearn.tree.plot_tree` or export to Graphviz to visualize the final, tuned decision tree.

Evaluate on Test Set and Report Metrics

1. Metrics to Report:

Accuracy

Precision & Recall (per class)

Confusion Matrix
2. Final Evaluation:

Evaluate the tuned model on the test set and report the results.

Example Datasets

Wine Quality Prediction , Breast Cancer Wisconsin , etc.