Machine Learning Repository

Diabetes
Abalone
Mushroom

UCI

FIVE 05 DATASET

Predict Students' Dropout and Academic Success

Default of Credit Card Clients

SHANAY

SHETH

PRANAV

SHINDE

FINAL ASSIGNMENT

Hyperparameter sensitivity trends across dataset types in UCI ML repository

CSIT332

Shape of the Data

Abalone

Data Scale & Dimensionality

Structure: 4,177 instances (Rows) × 8 features (Columns).
Status: Small / Low Dimensionality.

Class Distribution

Breakdown: Young (≤ 9 rings) vs. Old (> 9 rings).
Status: Perfectly Balanced (~50/50 split).

Complexity & Decision Boundaries

High noise with significant feature overlap.
Non-Linear.
Physical traits (weight/length) do not strictly correlate with age. Simple linear models will likely underperform; the problem requires a model capable of learning complex, non-linear boundaries to distinguish between Young and Old.

Shape of the Data

Diabetes

Data Scale & Dimensionality

Structure: 1 instances (Rows) × 20 features (Columns).
Status: Small / Low Dimensionality.

Class Distribution

Breakdown: Classification task, but class labels are not explicitly defined in the metadata.
Status: Not specified in dataset description.

Complexity & Decision Boundaries

Multivariate time-series data with irregular vs. fixed (fictitious) timestamps.
Features include categorical and integer codes representing insulin doses, blood glucose measurements, meals, exercise, and events.
Real timestamps (from electronic logs) mixed with artificial "logical time" entries (from paper logs), introducing temporal inconsistency.
Potentially non-linear relationships among patient events, insulin dosage patterns, and glucose levels.
Classification may require models that handle temporal sequences, irregular sampling, and coded event semantics.

Shape of the Data

Mushroom

Data Scale & Dimensionality

Structure: 8,124 instances (Rows) × 22 features (Columns)
Status: Medium-sized / Moderately High Dimensionality..

Class Distribution

Breakdown: Binary classification — edible (e) vs. poisonous (p).
Status: Known to be highly separable, though the dataset description does not specify imbalance. Historically, classes are close to balanced but not perfectly even

Complexity & Decision Boundaries

Entire feature space is categorical, many with multiple categories—requires models that handle high-cardinality categorical variables.
Class separation is often strong, due to odor, bruising, gill color, and spore-print color providing highly discriminative signals.
Non-linear interactions dominate; many edible vs. poisonous distinctions rely on combinations of categorical traits rather than single features.
Missing values exist (notably in stalk-root), adding mild preprocessing complexity.
Despite non-linearity, decision boundaries are relatively easy for tree-based models, leading to near-perfect classification performance.
Linear models without one-hot encoding or feature transformation would struggle due to the purely categorical and non-ordinal feature structure.

Shape of the Data

Predict Students' Dropout and Academic Success

Data Scale & Dimensionality

Structure: 4,424 instances (Rows) × 36 features (Columns).
Status: Medium-sized / Moderately High Dimensionality..

Class Distribution

Breakdown: Three-class classification — dropout, enrolled, graduate.
Status: Strongly imbalanced, with one class dominating the distribution (as stated in dataset description).

Complexity & Decision Boundaries

Heterogeneous feature space containing demographic, socio-economic, academic history, and performance attributes.
Mix of real, integer, and categorical features → requires models capable of handling diverse data types.
High likelihood of non-linear relationships among socioeconomic factors, previous education, and student outcomes.
Significant class imbalance introduces challenging decision boundaries, where majority-class bias may overshadow minority classes.
Predicting dropout vs. academic success involves interactions not linearly separable; models must capture complex, multi-factor dependencies.
Algorithms robust to imbalance and non-linearity (e.g., gradient boosting, tree ensembles, or cost-sensitive learning) are typically necessary.

Shape of the Data

Default of Credit Card Clients

Data Scale & Dimensionality

Structure: 0,000 instances (Rows) × 23 features (Columns).
Status: Large / Moderate Dimensionality

Class Distribution

Breakdown: Binary classification — default payment (Yes = 1) vs. no default (0).
Status: imbalanced, with non-defaulting clients forming the majority

Complexity & Decision Boundaries

Features include demographics (age, sex, education, marital status) and financial behavioral signals such as repayment history, bill amounts, and previous payments.
Contains sequential financial indicators (payment history and bill amounts over 6 months), introducing temporal patterns without explicit time-series structure.
Strongly non-linear relationships between repayment behavior and default likelihood.
Decision boundaries are influenced by multi-step interactions (e.g., credit limit × payment history × bill magnitude), creating a highly entangled feature space.
Models must account for scaling differences: some features are small integers (e.g., repayment status), while others are large monetary values (bill and payment amounts).
Class imbalance and overlapping patterns make simple linear models less effective; tree ensembles, gradient boosting, and neural networks typically outperform due to their ability to model complex, non-linear financial risk dynamics.

PROJECT WORKFLOW

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt.

01

Perform Preprocessing

02

For Each Classifier

03

For Each Hyperparameter Create a List of Possible Values

04

Perform GridSearch and report sensitivity (variance for each hyperparameter)

05

Analyze sensitivities across hyperparameters, across classifiers, across datasets

WorkFlow

Importing Data

Setting Up Hyper parameter for all of the classification model used

Setting up all model being used: decision_tree, knn, logistic_regression, svm

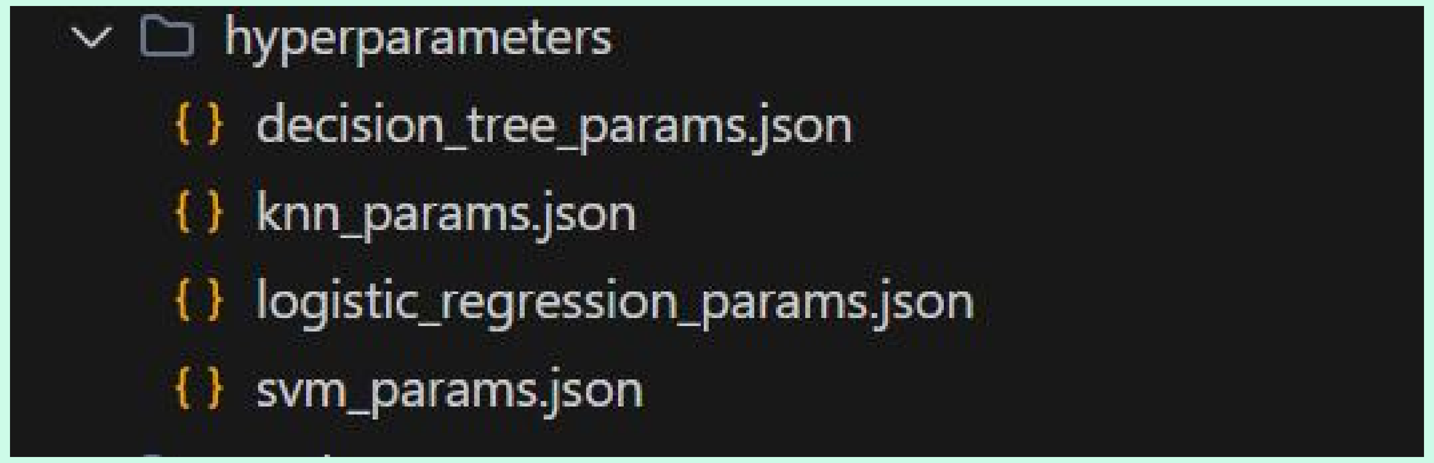
Preprocessing of the data

Sensitivity Check Run the modelsget AUC ROC, F1 Score and Accuracy

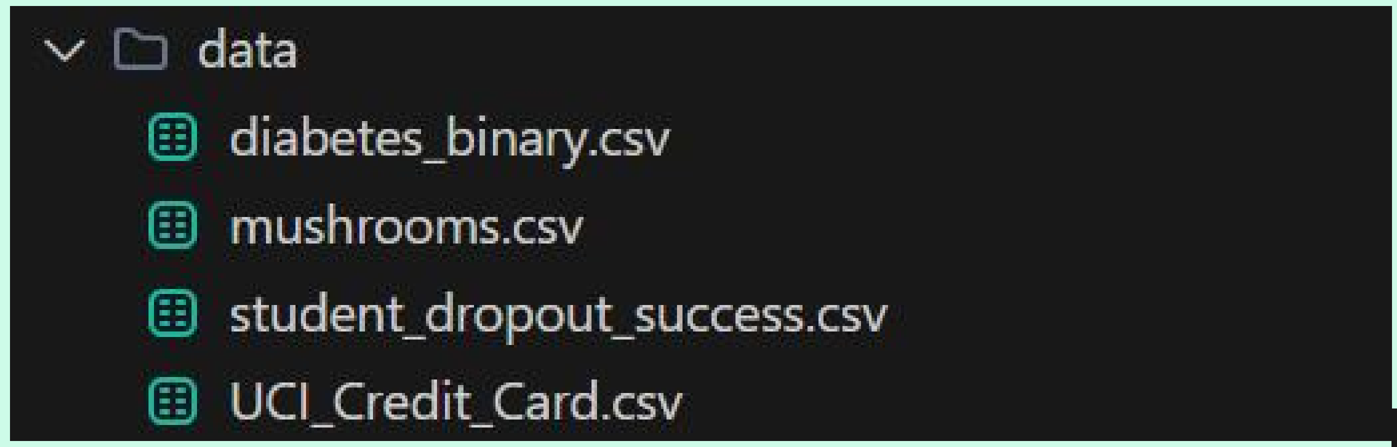
Find the variance for each of the model and plot them

Structure of the Code

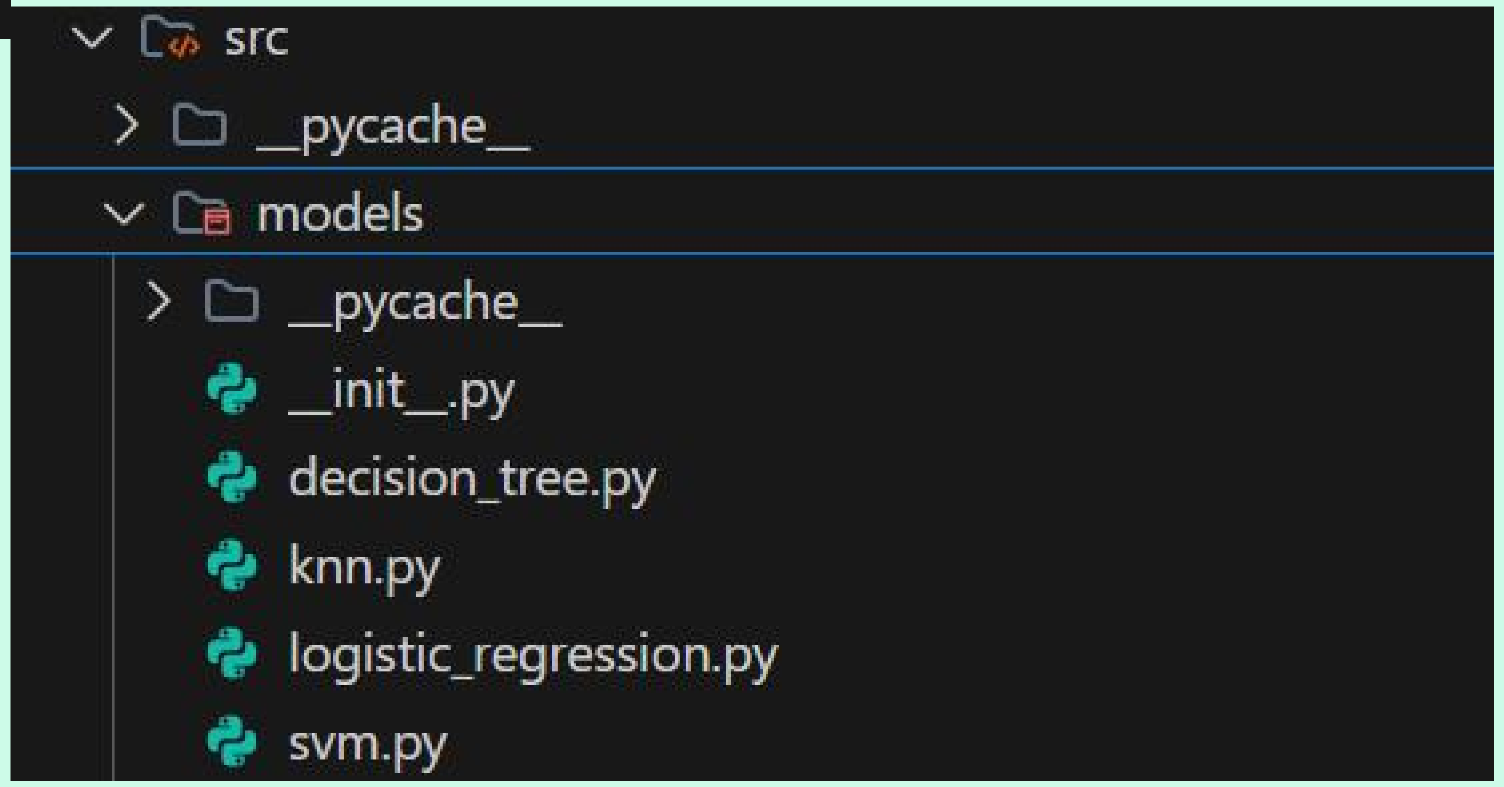
Code



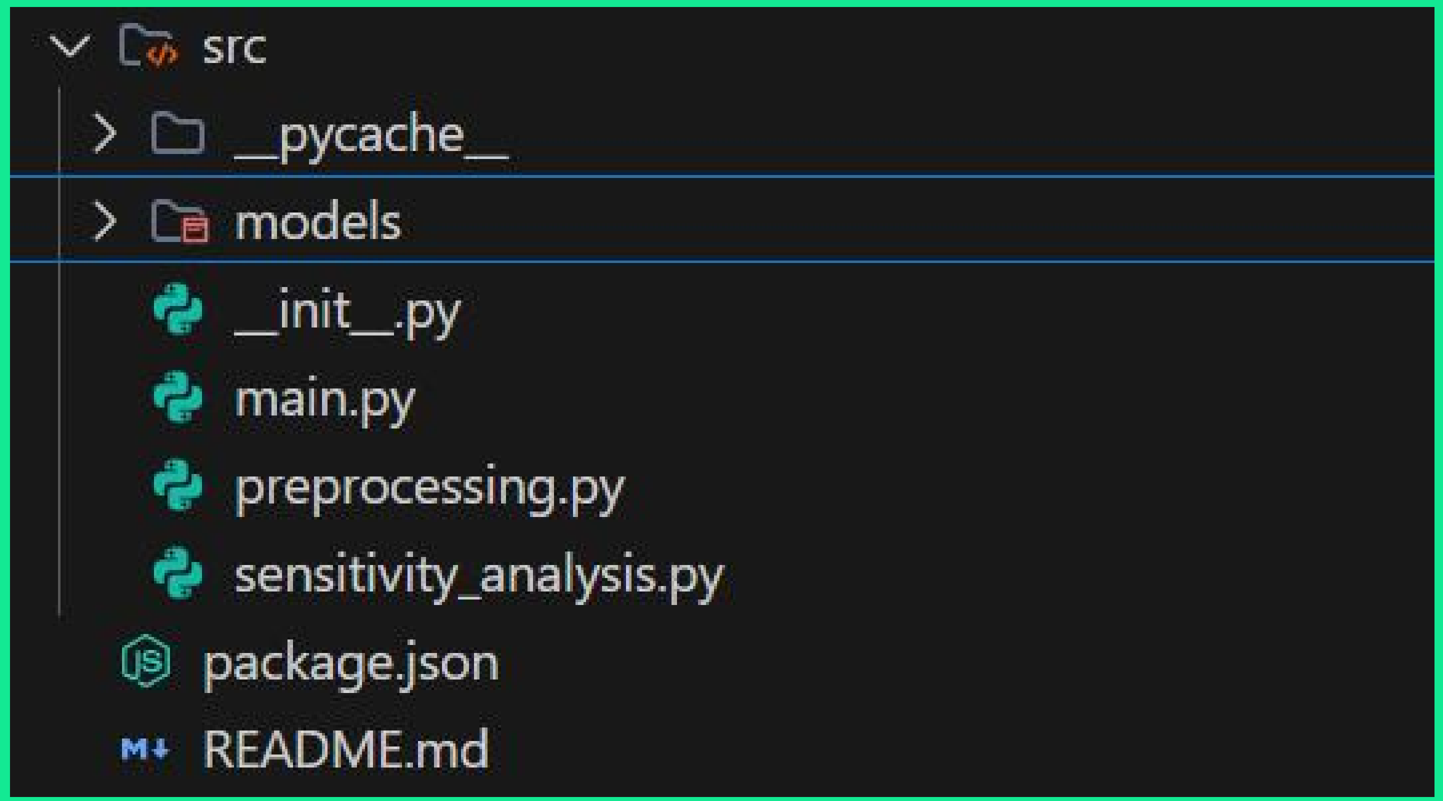
```
hyperparameters
    decision_tree_params.json
    knn_params.json
    logistic_regression_params.json
    svm_params.json
```



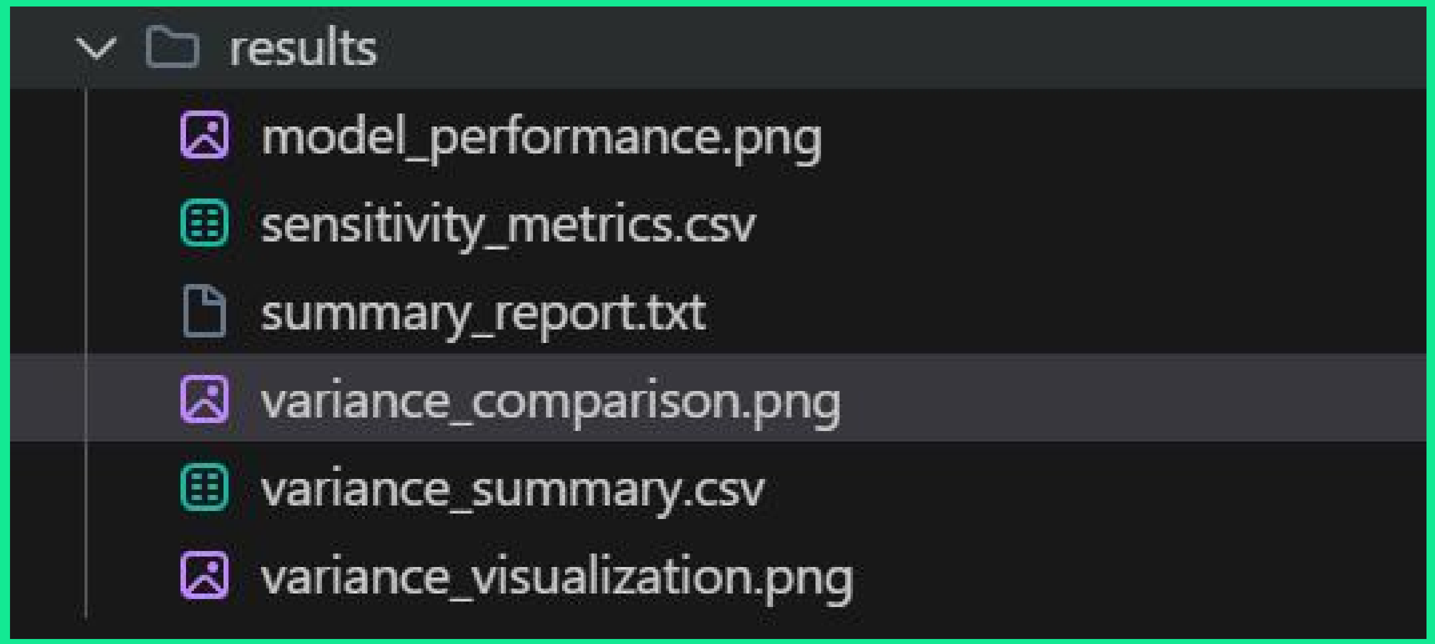
```
data
    diabetes_binary.csv
    mushrooms.csv
    student_dropout_success.csv
    UCI_Credit_Card.csv
```



```
src
    __pycache__
    models
        __pycache__
        __init__.py
        decision_tree.py
        knn.py
        logistic_regression.py
        svm.py
```



```
src
    __pycache__
    models
    __init__.py
    main.py
    preprocessing.py
    sensitivity_analysis.py
package.json
README.md
```



```
results
    model_performance.png
    sensitivity_metrics.csv
    summary_report.txt
    variance_comparison.png
    variance_summary.csv
    variance_visualization.png
```