

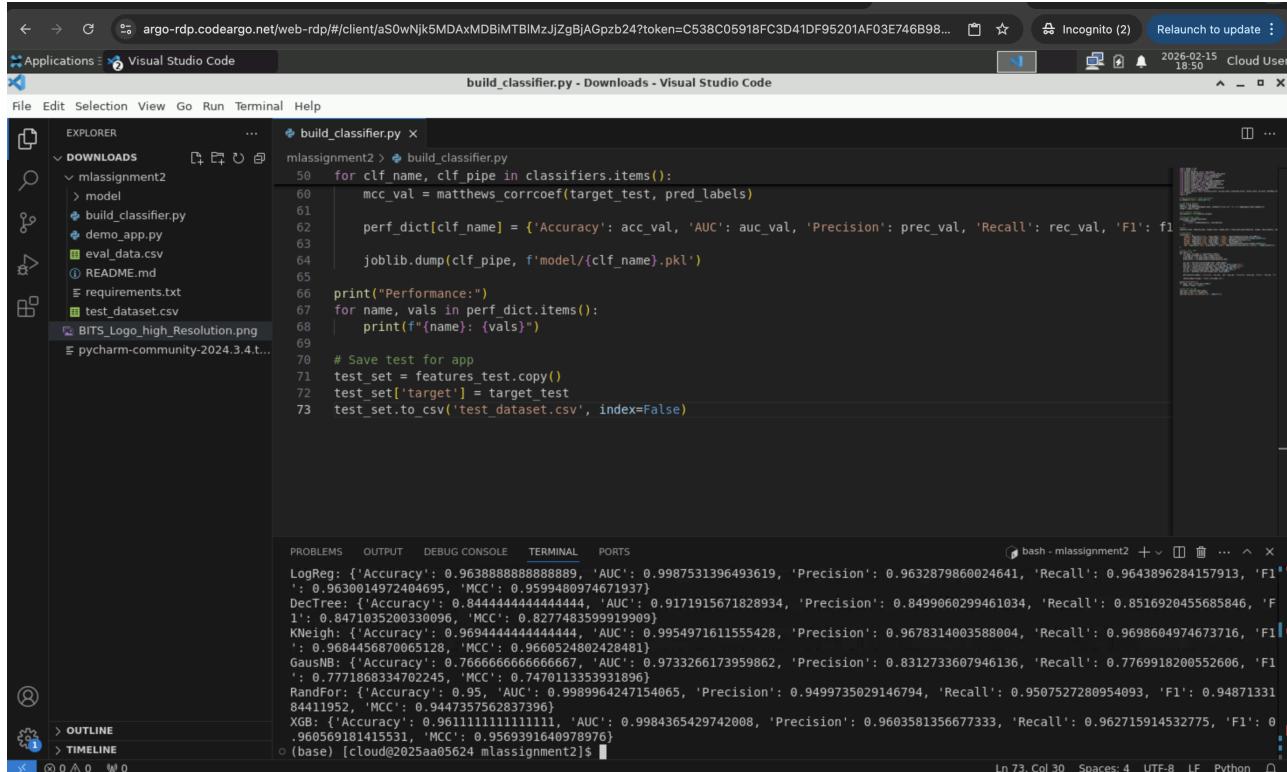
Assignment 2 Submission

Mutyala Sri Lakshmi Akhila (2025aa05624)

1. GitHub Repository: <https://github.com/2025aa05624-glitch/mlassignment2.git/>

2. Live Streamlit App: <https://mlassignment2-lqqfm7shbfurkdzqlcneks.streamlit.app/>

3. Screenshot



The screenshot shows a Visual Studio Code window with the following details:

- File Explorer:** Shows a project structure under 'DOWNLOADS' named 'mlassignment2'. It includes files like 'model', 'build_classifier.py', 'demo_app.py', 'eval_data.csv', 'README.md', 'requirements.txt', 'test_dataset.csv', and 'BITS_Logo_high_Resolution.png'.
- Code Editor:** Displays the content of 'build_classifier.py'.
- Terminal:** Shows the output of running the script, displaying various classifier metrics (Accuracy, AUC, Precision, Recall, F1) for different models (LogReg, DecTree, KNeigh, GausNB, RandFor, XGB).

```

build_classifier.py - Downloads - Visual Studio Code
File Edit Selection View Go Run Terminal Help
EXPLORER ... build_classifier.py x
 DOWNLOADS mlassignment2 > build_classifier.py
> model
+ build_classifier.py
+ demo_app.py
eval_data.csv
README.md
requirements.txt
test_dataset.csv
BITS_Logo_high_Resolution.png
pycharm-community-2024.3.4.t...
50     for clf_name, clf_pipe in classifiers.items():
51         mcc_val = matthews_corrcoef(target_test, pred_labels)
52
53         perf_dict[clf_name] = {'Accuracy': acc_val, 'AUC': auc_val, 'Precision': prec_val, 'Recall': rec_val, 'F1': f1}
54
55         joblib.dump(clf_pipe, f'model/{clf_name}.pkl')
56
57     print("Performance:")
58     for name, vals in perf_dict.items():
59         print(f'{name}: {vals}')
60
61     # Save test for app
62     test_set = features_test.copy()
63     test_set['target'] = target_test
64     test_set.to_csv('test_dataset.csv', index=False)
65
66
67
68
69
70
71
72
73
    PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
LogReg: {'Accuracy': 0.9638888888888889, 'AUC': 0.9987531396493619, 'Precision': 0.9632879860024641, 'Recall': 0.9643896284157913, 'F1': 0.9630014972404695, 'MCC': 0.9599480974671937}
DecTree: {'Accuracy': 0.8444444444444444, 'AUC': 0.9171915671828934, 'Precision': 0.8499660299461034, 'Recall': 0.8516920455685846, 'F1': 0.8471035200330096, 'MCC': 0.8277483599919999}
KNeigh: {'Accuracy': 0.9694444444444444, 'AUC': 0.9954971611555428, 'Precision': 0.9678314003588004, 'Recall': 0.9698604974673716, 'F1': 0.9684456870065126, 'MCC': 0.9660524802428481}
GausNB: {'Accuracy': 0.7666666666666667, 'AUC': 0.9733266173959862, 'Precision': 0.8312733607946136, 'Recall': 0.7769918200552606, 'F1': 0.777186834702245, 'MCC': 0.7470113353931896}
RandFor: {'Accuracy': 0.95, 'AUC': 0.998964247154065, 'Precision': 0.9499735029146794, 'Recall': 0.9507527280954093, 'F1': 0.9487133184411952, 'MCC': 0.9447357562837396}
XGB: {'Accuracy': 0.9611111111111111, 'AUC': 0.9984365429742008, 'Precision': 0.9603581356677333, 'Recall': 0.962715914532775, 'F1': 0.960569181415531, 'MCC': 0.9569391640978976}
(base) [cloud@2025aa05624 mlassignment2]$ 
  
```

Classification App for Digits

a. Problem Statement

Goal: Classify images of handwritten digits (0-9) with ML classifiers, evaluate them, and build a Streamlit app for interactive testing.

b. Dataset Description

Digits dataset: 1,797 entries, 64 numeric features (pixels). Classes: 10 digits. Balanced, no misses. From scikit-learn/UCI.

c. Models Used

Implemented:

1. Logistic Regression
2. Decision Tree
3. K-Nearest Neighbors

4. Gaussian Naive Bayes
5. Random Forest Ensemble
6. XGBoost Ensemble

Metrics Comparison

Classifier	Accuracy	AUC	Precision	Recall	F1	MCC
LogReg	0.96	1.00	0.96	0.96	0.96	0.96
DecTree	0.84	0.92	0.85	0.85	0.85	0.83
KNeigh	0.97	1.00	0.97	0.97	0.97	0.97
GausNB	0.77	0.97	0.83	0.78	0.78	0.75
RandFor	0.95	1.00	0.95	0.95	0.95	0.94
XGB	0.96	1.00	0.96	0.96	0.96	0.95

Performance Notes

Classifier	Notes on Performance
LogisticRegression	Reliable, near-perfect on simple patterns.
DecisionTree	Average, tends to overfit pixel noise.
KNN	Excellent, suits distance-based image tasks.
GaussianNaiveBias	Poorest, assumptions don't fit data well.
RandonForest	Good, ensembles help consistency.
XGBoost	Strong, boosting captures details.

Setup

1. Git clone <https://github.com/2025aa05624-glitch/mlassignment2.git>
2. pip install -r requirements.txt
3. python build_classifiers.py

How to Use

- streamlit run demo_app.py
- Upload CSV (with 'test_dataset' from repo), choose model, see results.

App Link

<https://mlassignment2-lgqfm7shbfurkdzqlcneks.streamlit.app/>