

1. GitHub Repository Link containing

- Complete source code
- requirements.txt
- README.md

<https://github.com/2025aa05376-himanshu/ml-assignment-2.git>

2. Live Streamlit App Link

- Deployed using Streamlit Community Cloud
- Must open an interactive frontend when clicked

<https://ml-assignment-2-n7je8bvqjgh7nevqr9f9yp.streamlit.app/>

3. Screenshot

The screenshot shows a Jupyter Notebook interface with the following content:

```
model.fit(X_train_p, y_train)
y_pred = model.predict(X_test_p)
y_prob = model.predict_proba(X_test_p)[:, 1]

Accuracy = accuracy_score(y_test, y_pred)
AUC = roc_auc_score(y_test, y_prob)
Precision = precision_score(y_test, y_pred)
Recall = recall_score(y_test, y_pred)
F1 = f1_score(y_test, y_pred)
MCC = matthews_corrcoef(y_test, y_pred)

results.append([name, Accuracy, AUC, Precision, Recall, F1, MCC])

joblib.dump(model, f"model/{name}.pkl")

# Show evaluation table
results_df = pd.DataFrame(results, columns=[
    "Model", "Accuracy", "AUC", "Precision", "Recall", "F1-Score", "MCC"
])
print("\nModel Performance:")
print(results_df)

Training logistic...
Training decision_tree...
Training knn...
Training naive_bayes...
Training random_forest...
Training xgboost...

Model Performance:
      Model  Accuracy      AUC  Precision  Recall  F1-Score      MCC
0   logistic  0.853473  0.908057  0.758814  0.618954  0.681785  0.593026
1  decision_tree  0.816178  0.761212  0.634333  0.649673  0.641912  0.518354
2        knn  0.829935  0.858420  0.686391  0.606536  0.643997  0.534599
3  naive_bayes  0.547655  0.763991  0.353816  0.948366  0.515361  0.335578
4  random_forest  0.849329  0.901982  0.733283  0.637908  0.682279  0.586589
5     xgboost  0.867562  0.929277  0.770941  0.679739  0.722473  0.638127
```

The screenshot shows a GitHub repository named 'ml-assignment-2'. The repository has 15 commits. The README file is updated, and the deployment_workflow is created. The repository is associated with the UCI Adult Income Dataset.

4. The Github README content

1. Problem Statement:

The objective of this project is to develop and compare multiple machine learning classification models to determine whether an individual's annual income exceeds \$50,000 using demographic and employment-related features from the U.S. Census dataset.

2. Dataset Description

- Dataset Name: Adult Income Dataset
- Source: UCI Machine Learning Repository
- Number of Instances: 48,842
- Number of Features: 14 input attributes
- Target Variable: Income (<=50K, >50K)
- Type of Problem: Binary Classification

The dataset contains both numerical and categorical attributes describing an individual's education level, occupation, working hours, marital status, and other socio-economic characteristics.

3. Data Preprocessing Steps

- Removed rows containing missing values
- Converted target variable into binary format (0 and 1)
- Applied One-Hot Encoding to categorical features
- Standardized numerical features
- Split dataset into 80% training and 20% testing sets

4. The following six classification Machine Learning Models are Implemented.

- Logistic Regression
- Decision Tree Classifier
- K-Nearest Neighbours
- Gaussian Naive Bayes
- Random Forest (Ensemble)
- XGBoost (Ensemble Boosting)

5. Evaluation Metrics

- Accuracy
- AUC Score
- Precision
- Recall
- F1 Score
- Matthews Correlation Coefficient (MCC)

6. Model Comparison Table

Model	Accuracy	AUC	Precision	Recall	F1	MCC
Logistic Regression	0.853473	0.908057	0.758814	0.618954	0.681785	0.593026
Decision Tree	0.816178	0.761212	0.634333	0.649673	0.641912	0.518354
K-Nearest Neighbours	0.829935	0.858420	0.686391	0.606536	0.643997	0.534599
Gaussian Naive Bayes	0.547655	0.763991	0.353816	0.948366	0.515361	0.335578
Random Forest (Ensemble)	0.849329	0.901982	0.733283	0.637908	0.682279	0.586589
XGBoost (Ensemble Boosting)	0.867562	0.929277	0.770941	0.679739	0.722473	0.638127

7. Observations

- Logistic Regression provided a strong baseline performance.
- Decision Tree showed moderate accuracy but potential overfitting.
- KNN performed well after feature scaling.
- Naive Bayes was computationally efficient but slightly lower in predictive power.
- Random Forest improved overall stability and generalization.
- XGBoost achieved the best performance across most evaluation metrics.

8. Streamlit Application Features

The application has been deployed using Streamlit Community Cloud. The deployed web application includes:

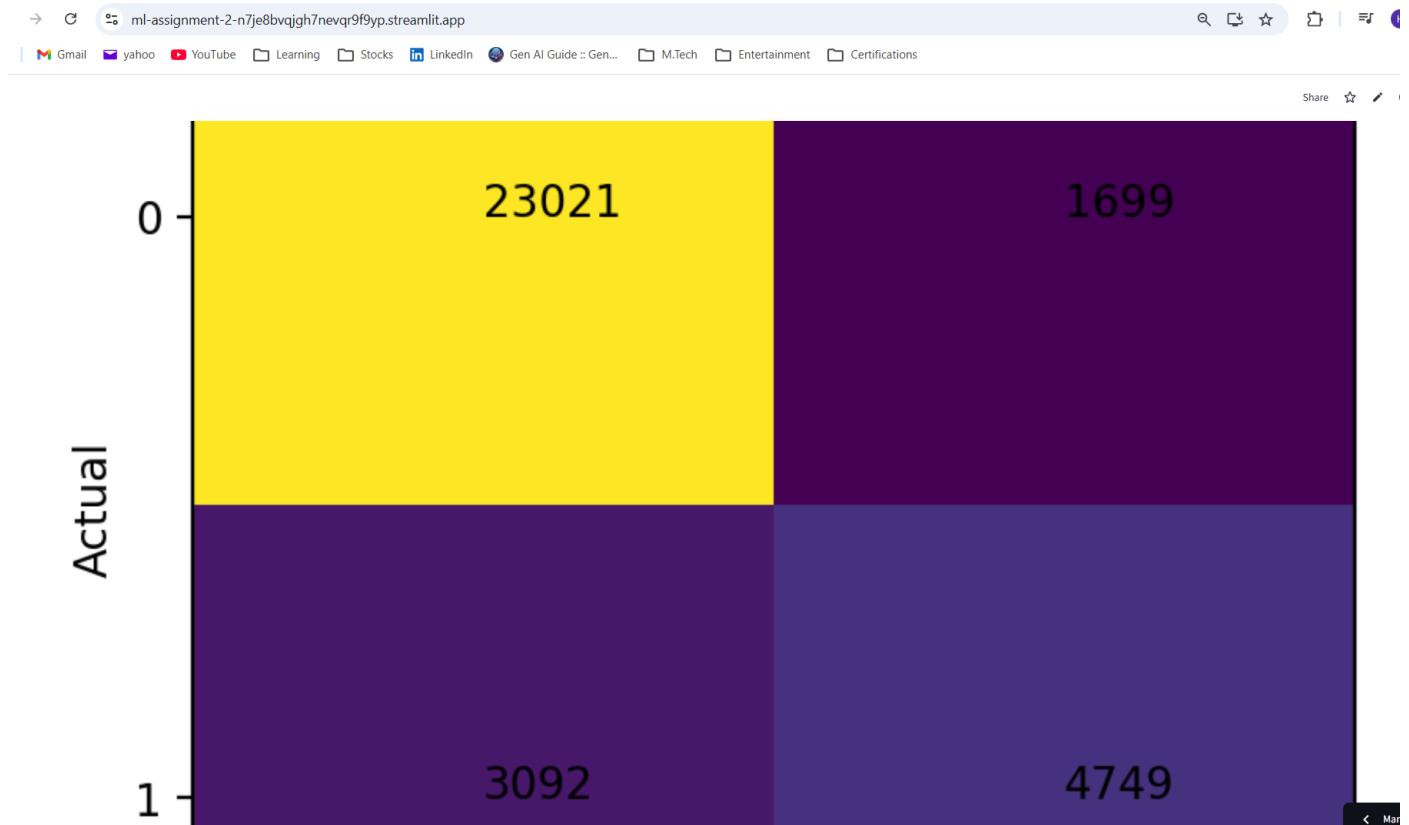
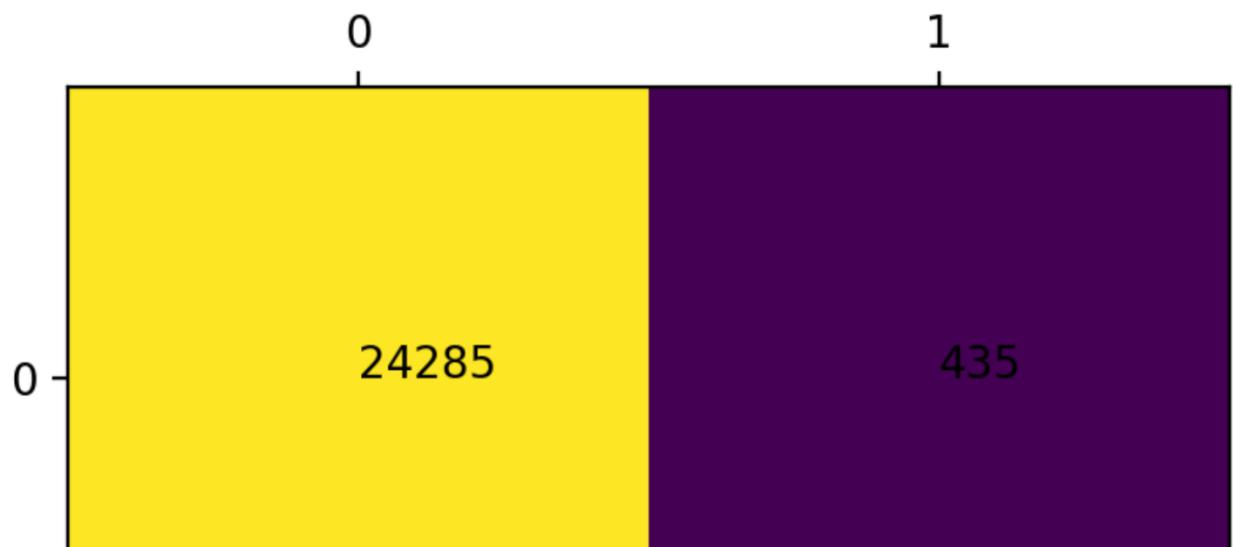
- CSV test dataset upload
- Model selection dropdown
- Display of evaluation metrics
- Confusion matrix visualization
- Classification report

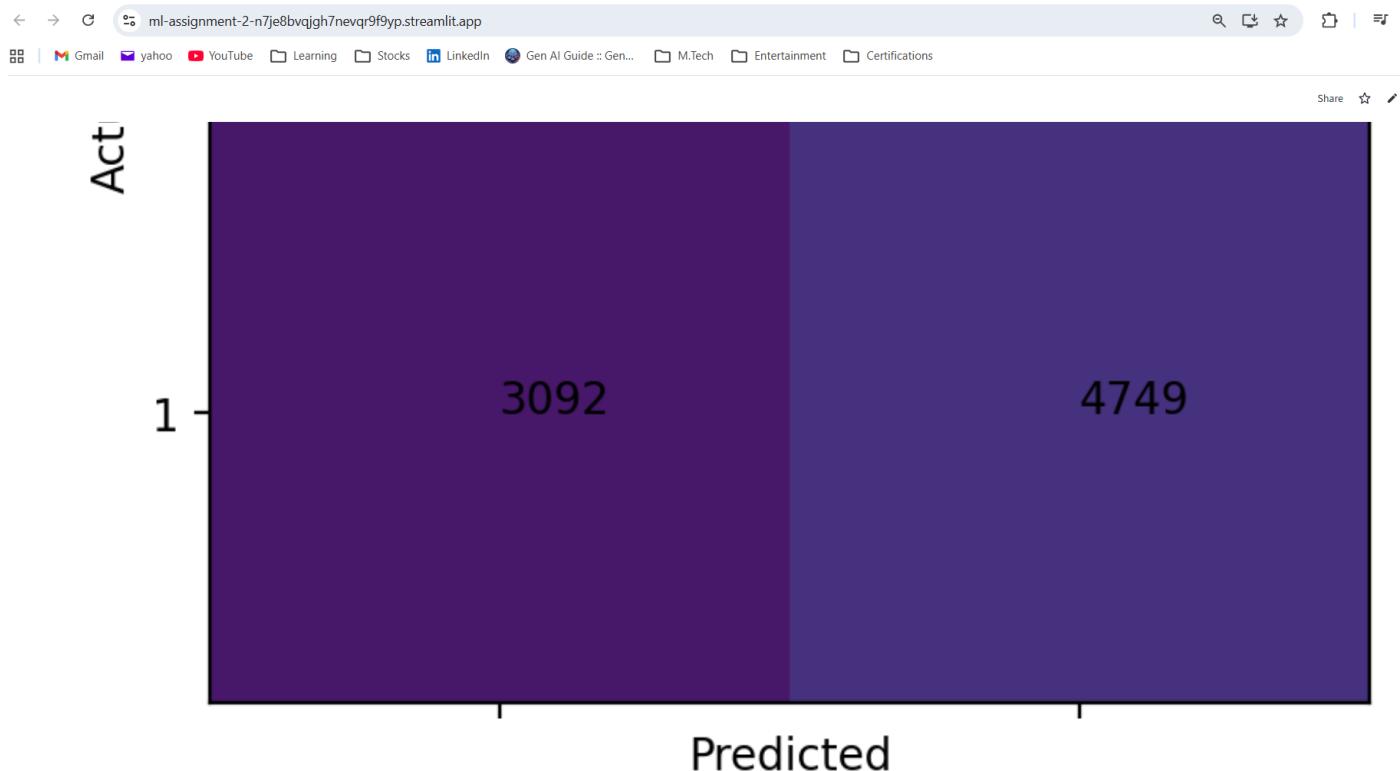
The screenshot shows a Streamlit application titled "Machine Learning Classification Models". At the top, there's a header "UCI Adult Income Dataset". Below it, a "Select Model" dropdown is set to "Logistic Regression". Underneath, there's a "Upload CSV File" section with a "Drag and drop file here" button and a "Browse files" button. A file named "adult.csv" (3.4MB) is listed. The main content area is titled "Evaluation Metrics" and contains a table:

	Metric	Score
0	Accuracy	0.8529
1	Precision	0.7365
2	Recall	0.6057
3	F1 Score	0.6647
4	MCC	0.5761

At the bottom right of the table is a "Manage app" button.

Confusion Matrix





Classification Report

precision recall f1-score support

0	0.97	0.98	0.98	24720
1	0.94	0.91	0.93	7841

accuracy		0.97	32561	
macro avg	0.96	0.95	0.95	32561
weighted avg	0.97	0.97	0.97	32561