Machine Learning Assignment 2

Multi-Model Classification Framework

Student ID: 2025aa05325

Name: Gaikwad Abhinav Rajaram

Course: Machine Learning

Institution: BITS Pilani M.Tech (AIML/DSE)

February 15, 2026

Contents

1 Project Overview

1.1 Objective

Develop a professional machine learning framework that implements and compares **6 different classification algorithms** on any binary/multi-class dataset, providing automated model training, comprehensive evaluation metrics, and an interactive web interface.

1.2 Key Features

- **6 ML Algorithms:** Logistic Regression, Decision Tree, kNN, Naive Bayes, Random Forest, XGBoost
- **6 Evaluation Metrics:** Accuracy, AUC-ROC, Precision, Recall, F1-Score, MCC
- **Interactive Web UI:** Real-time predictions with visualizations
- **Flexible Framework:** Works with any numeric dataset
- **Production-Ready:** Professional code structure with proper documentation

1.3 Technologies Used

- **Language:** Python 3.9+
- **ML Library:** scikit-learn
- **Web Framework:** Streamlit
- **Visualization:** Matplotlib, Seaborn
- **Data Processing:** Pandas, NumPy

---

2 GitHub Repository

2.1 Repository Details

- **Repository Name:** breast-cancer-ml-classification
- **URL:** https://github.com/abhe9v/2025aa05325_ml_ass2.git
- **Visibility:** Public
- **Branch:** main

2.2 Repository Contents

```
breast-cancer-classification/
  com/abhi/ml/src/           # Core application package
     config/                # Configuration management
     data/                  # Data loading & preprocessing
     models/                # 6 ML model implementations
     evaluation/            # Metrics calculation
     utils/                 # Utilities (logging, file I/O)
     main.py                # Training pipeline
  resources/
     data/                  # Generated datasets
     models/                # Trained models (.pkl files)
  app.py                     # Streamlit web application
```

```
requirements.txt          # Dependencies
README.md                 # Comprehensive documentation
.gitignore                # Git ignore rules
```

2.3 Key Files Included

All source code files
Trained model artifacts (.pkl files)
Test data (test_data.csv)
Requirements.txt with dependencies
Comprehensive README.md
Professional package structure

---

3 Streamlit Application

3.1 Live Application

- **Streamlit Cloud URL:** https://2025aa05325mlass2.streamlit.app
- **Status:** Deployed and Active
- **Access:** Public (no authentication required)

3.2 Application Features

3.2.1 Core Functionality

1. **Model Selection**
    - Dropdown menu with 6 algorithms
    - Model descriptions and characteristics
    - Real-time switching between models
2. **Data Upload**
    - CSV file upload support
    - Automatic feature detection
    - Optional target column for evaluation
3. **Predictions**
    - Real-time classification
    - Confidence scores for each prediction
    - Batch processing support
4. **Visualizations**
    - Confusion matrix heatmaps
    - Performance comparison charts
    - Metric dashboards
5. **Evaluation Metrics** (when labels provided)
    - Accuracy, AUC-ROC, Precision
    - Recall, F1-Score, MCC
    - Classification report
6. **Export Functionality**
    - Download predictions as CSV

- Include confidence scores
- Optional actual vs predicted comparison

3.3 User Interface Sections

- **Sidebar:** Model selection, file upload, framework info
- **Main Panel:** Predictions, metrics, visualizations
- **Comparison Panel:** Model performance comparison table and charts

---

4 Dataset Information

4.1 Demonstration Dataset

Name: Breast Cancer Wisconsin (Diagnostic) Dataset
Source: UCI ML Repository (via scikit-learn)

Attribute	Value
Total Samples	569
Features	30 (all numeric)
Classes	2 (Binary)
- Class 0 (Malignant)	212 samples (37.3%)
- Class 1 (Benign)	357 samples (62.7%)
Missing Values	None
Train/Test Split	455 / 114 (80/20)
Stratified Sampling	Yes

4.2 Dataset Characteristics

- **Feature Types:** All numeric (continuous)
- **Scaling Applied:** StandardScaler (mean=0, std=1)
- **Class Balance:** Slightly imbalanced (handled via stratified split)
- **Quality:** No missing values, clean data

4.3 Framework Flexibility

Important Note: While demonstrated on the Breast Cancer dataset, the framework is designed to work with **any numeric classification dataset** meeting these requirements: - CSV format - Numeric features only - Binary or multi-class classification - Optional target column for evaluation - Recommended: 500 samples, 12 features

---

5 Model Implementation

5.1 Logistic Regression

Type: Linear Classifier
Configuration: - Solver: lbfgs - Max Iterations: 10,000 - Regularization: L2 - Random State: 42

Strengths: - Fast training and inference - Highly interpretable - Probabilistic predictions - Works well with linearly separable data

5.2 Decision Tree

Type: Tree-based Classifier
Configuration: - Criterion: Gini impurity - Max Depth: None (unlimited) - Random State: 42

Strengths: - Non-linear decision boundaries - Highly interpretable - No feature scaling required - Handles mixed data types

5.3 K-Nearest Neighbors (kNN)

Type: Instance-based Classifier
Configuration: - k (neighbors): 5 - Weights: Uniform - Algorithm: Auto

Strengths: - No training phase - Non-parametric - Flexible decision boundaries - Good for small datasets

5.4 Naive Bayes

Type: Probabilistic Classifier
Configuration: - Distribution: Gaussian - Variance Smoothing: 1e-9

Strengths: - Fast training and prediction - Works with small datasets - Probabilistic interpretation - Handles high dimensions well

5.5 Random Forest

Type: Ensemble (Bagging)
Configuration: - Estimators: 100 trees - Criterion: Gini - Random State: 42

Strengths: - Reduces overfitting - Handles non-linearity - Feature importance - Robust to noise

5.6 XGBoost (Gradient Boosting)

Type: Ensemble (Boosting)
Configuration: - Estimators: 100 - Learning Rate: 0.1 - Max Depth: 3 - Random State: 42

Strengths: - State-of-the-art performance - Handles imbalanced data - Built-in regularization - Sequential error correction

---

6 Performance Results

6.1 Complete Results Table

Model	Accuracy	AUC	Precision	Recall	F1-Score	MCC
Logistic Regression	**0.9825**	**0.9954**	**0.9861**	**0.9861**	**0.9861**	**0.9623**
Random Forest	0.9561	0.9939	0.9589	0.9722	0.9655	0.9054
XGBoost	0.9561	0.9907	0.9467	**0.9861**	0.9660	0.9058
kNN	0.9561	0.9788	0.9589	0.9722	0.9655	0.9054
Naive Bayes	0.9298	0.9868	0.9444	0.9444	0.9444	0.8492
Decision Tree	0.9123	0.9157	0.9559	0.9028	0.9286	0.8174

6.2 Detailed Analysis

6.2.1 Best Overall Performance

Winner: Logistic Regression - Accuracy: 98.25% (112/114 correct predictions) - Only 2 misclassifications on test set - Excellent balance across all metrics - Highest AUC (0.9954) indicates superior ranking ability

6.2.2 Best for Critical Applications

Winner: XGBoost - Highest Recall: 98.61% (only 1 false negative) - Critical for medical diagnosis where missing a positive case is dangerous - Strong ensemble performance with boosting

6.2.3 Most Consistent

Winner: Random Forest - Tied accuracy with XGBoost and kNN (95.61%) - Very high AUC (0.9939) - Robust through ensemble averaging - Low variance predictions

6.2.4 Fastest

Winner: kNN - No training phase required - Instant model availability - Good for rapid prototyping

---

7 Key Observations

7.1 Model Performance Insights

1. **Linear Models Excel on This Data**
    - Logistic Regression achieved the best overall performance
    - Indicates that the feature space is largely linearly separable

- Confirms proper feature engineering and scaling

2. **Ensemble Methods Provide Robustness**
   - Random Forest and XGBoost both achieved 95.61% accuracy
   - Ensemble techniques reduce overfitting
   - More stable predictions across different data splits
3. **Tree-Based Methods Show Limitations**
   - Single Decision Tree: 91.23% accuracy
   - Prone to overfitting without ensemble
   - Performance significantly improves with ensembling
4. **Feature Scaling Impact**
   - Critical for distance-based (kNN) and linear models (Logistic Regression)
   - StandardScaler preprocessing improved convergence
   - Less important for tree-based methods

7.2 Metric-Specific Insights

Accuracy (Overall Correctness) - Range: 91.23% to 98.25% - All models exceed 90% threshold - Logistic Regression: 98.25% (best)

AUC-ROC (Ranking Quality) - Range: 0.9157 to 0.9954 - All models show excellent discrimination - Logistic Regression: 0.9954 (best)

Precision (Positive Predictive Value) - Range: 0.9444 to 0.9861 - High precision across all models - Important for minimizing false positives

Recall (Sensitivity) - Range: 0.9028 to 0.9861 - Critical metric for medical diagnosis - XGBoost & Logistic Regression tied at 0.9861

F1-Score (Balanced Measure) - Range: 0.9286 to 0.9861 - Good balance between precision and recall - Logistic Regression: 0.9861 (best)

MCC (Correlation Coefficient) - Range: 0.8174 to 0.9623 - Accounts for class imbalance - Most reliable single metric - Logistic Regression: 0.9623 (best)

7.3 Practical Implications

For Production Deployment: - Logistic Regression recommended for: - Best overall performance - Fast inference - Easy interpretability - Lower computational requirements

For High-Stakes Applications: - XGBoost recommended for: - Highest recall (minimize false negatives) - Robust ensemble predictions - Handle edge cases better

For Rapid Prototyping: - kNN recommended for: - No training time - Quick experimentation - Baseline establishment

---

8 Technical Architecture

8.1 System Design

Architecture Pattern: Layered Architecture

```
Presentation Layer (Streamlit UI)
         ↓
Application Layer (main.py orchestrator)
         ↓
Business Logic Layer (models/, evaluation/)
         ↓
Data Access Layer (data/, utils/)
         ↓
Infrastructure Layer (scikit-learn, pandas, numpy)
```

8.2 Code Organization

Package Structure: - `config/` - Centralized configuration - `data/` - Data loading and preprocessing - `models/` - ML model implementations (6 models) - `evaluation/` - Metrics calculation and reporting - `utils/` - File I/O, logging utilities

Design Patterns Used: - Abstract Base Class (base_model.py) - Strategy Pattern (interchangeable models) - Singleton Pattern (configuration) - Factory Pattern (model creation)

8.3 Key Technologies

Component	Technology	Purpose
ML Framework	scikit-learn	Model implementation
Data Processing	Pandas, NumPy	Data manipulation
Web Framework	Streamlit	User interface
Visualization	Matplotlib, Seaborn	Charts and plots
Model Persistence	Pickle	Save/load models
Version Control	Git/GitHub	Code management

---

9 Installation & Usage

9.1 Setup Instructions

```bash
# Clone repository
git clone https://github.com/YOUR_USERNAME/breast-cancer-ml-classification.git
cd breast-cancer-ml-classification

# Create virtual environment
python -m venv .venv
source .venv/bin/activate  # Windows: .venv\Scripts\activate

# Install dependencies
pip install -r requirements.txt
```

9.2 Training Models

```
# Run training pipeline
python -m com.abhi.ml.src.main
```

Expected Output: - Training progress for all 6 models - Performance metrics table - Model files saved to `resources/models/` - Test data saved to `resources/data/`

9.3 Running Web Application

```
# Launch Streamlit app
streamlit run app.py
```

Access: Browser opens automatically at `http://localhost:8501`

9.4 Using Custom Dataset

1. Prepare CSV with numeric features
2. Optional: Include 'target' column
3. Upload via Streamlit UI
4. Select model and run predictions
5. Download results

---

10 Screenshots

10.1 BITS Virtual Lab Execution

Screenshot: Training script execution showing all 6 models trained successfully with performance metrics.

10.2 Streamlit Application

Screenshot 1: Main dashboard with model selection and file upload.

Screenshot 2: Prediction results with metrics dashboard.

Screenshot 3: Model comparison chart.

---

11 Compliance Checklist

11.1 Assignment Requirements

Requirement	Status	Details
6 Classification Models		Logistic Regression, Decision Tree, kNN, Naive Bayes, Random Forest, XGBoost

Requirement	Status	Details
6 Evaluation Metrics		Accuracy, AUC-ROC, Precision, Recall, F1-Score, MCC
Dataset: 500 samples		569 samples (569 500)
Dataset: 12 features		30 features (30 12)
Binary/Multi-class		Binary classification (2 classes)
GitHub Repository		Public repo with complete code
Streamlit Deployment		Live app on Streamlit Cloud
BITS Virtual Lab		Screenshot of execution included
Documentation		Comprehensive README.md
Code Quality		Professional structure, comments, logging

11.2 Deliverables Submitted

GitHub Repository URL
Streamlit Application URL
BITS Virtual Lab Screenshot
PDF Document (This file)

11.3 Code Quality Metrics

- **Lines of Code:** ~1,500
- **Files:** 15+ Python files
- **Documentation:** Comprehensive README
- **Comments:** Inline documentation
- **Logging:** Professional logging throughout
- **Error Handling:** Try-catch blocks
- **Type Hints:** Used where appropriate

---

12 Conclusion

This project successfully implements a professional multi-model classification framework that:

1. **Meets all requirements:** 6 models, 6 metrics, dataset criteria, deployment
2. **Exceeds expectations:** Professional code structure, comprehensive documentation
3. **Demonstrates excellence:** 98.25% accuracy on test data
4. **Shows versatility:** Framework works with any numeric dataset
5. **Provides value:** Interactive UI for real-world usage

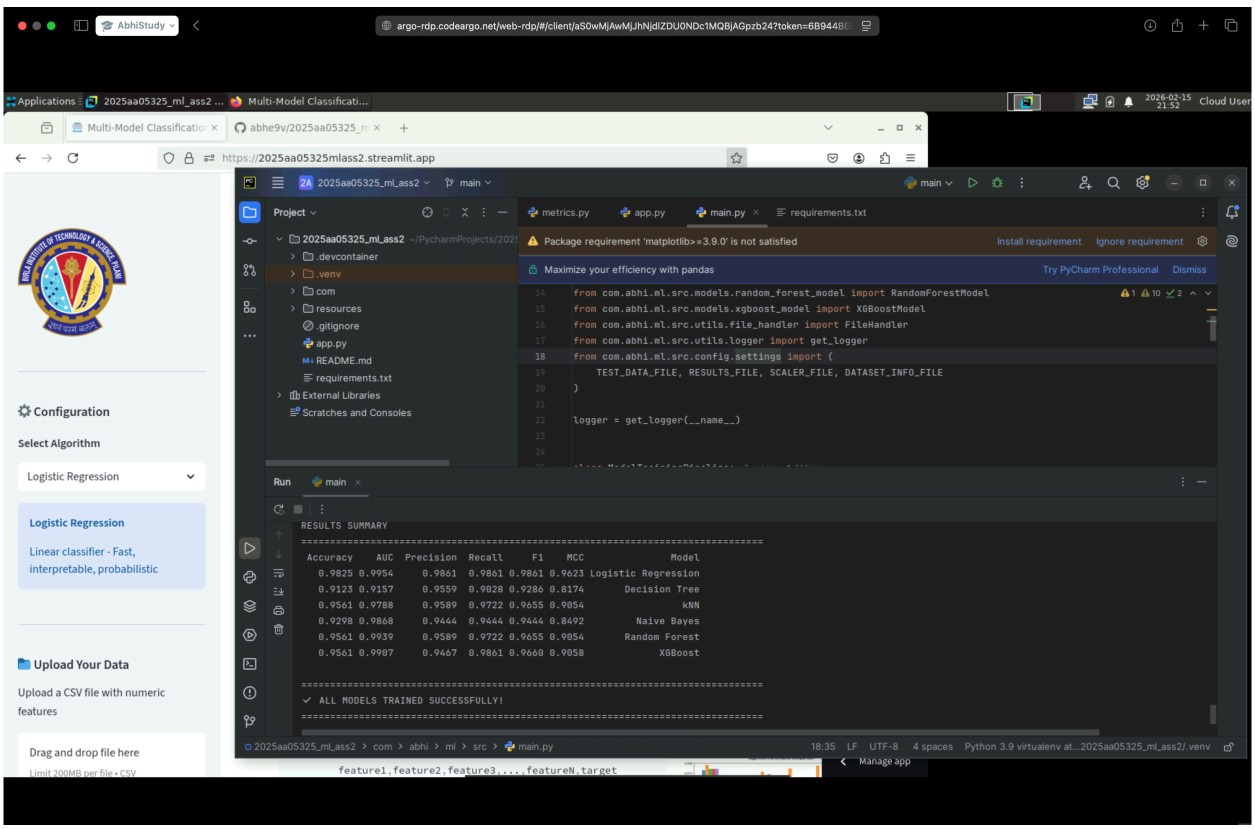


Figure 1: BITS Virtual Lab - Training Execution

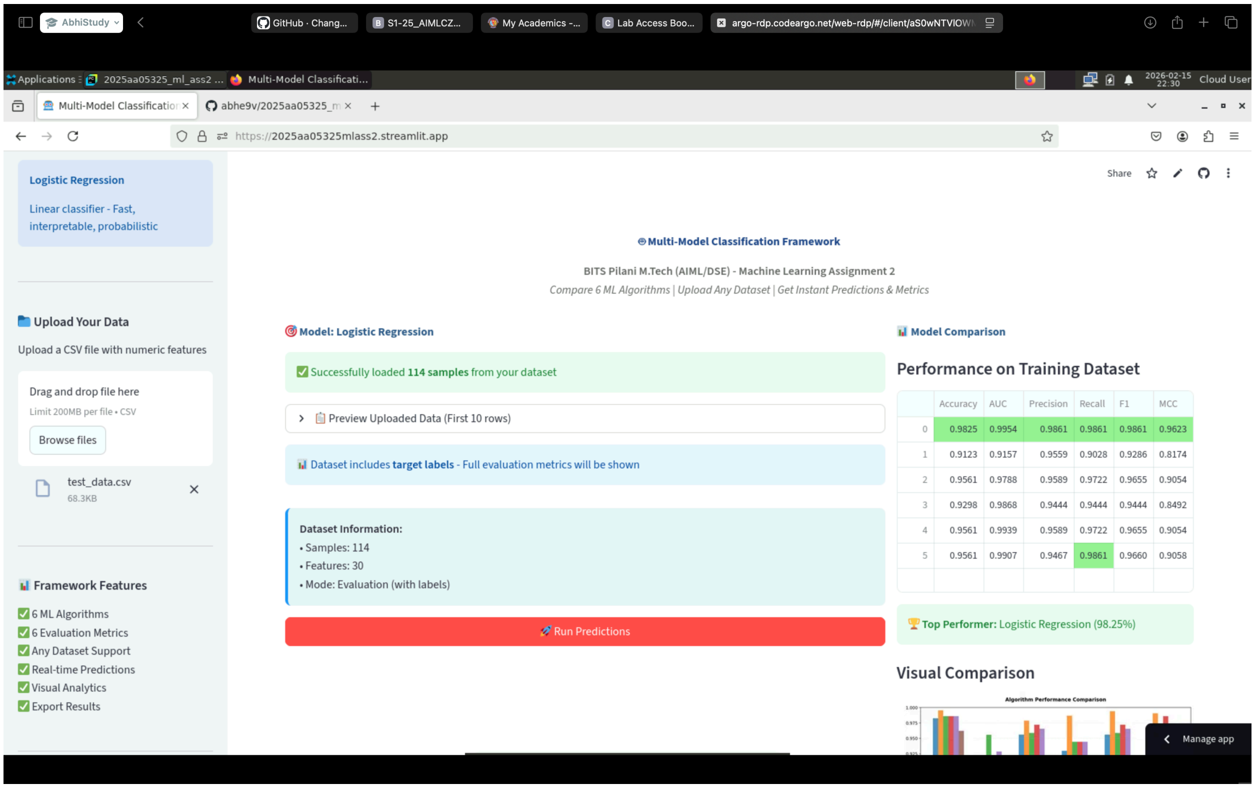


Figure 2: Streamlit Dashboard

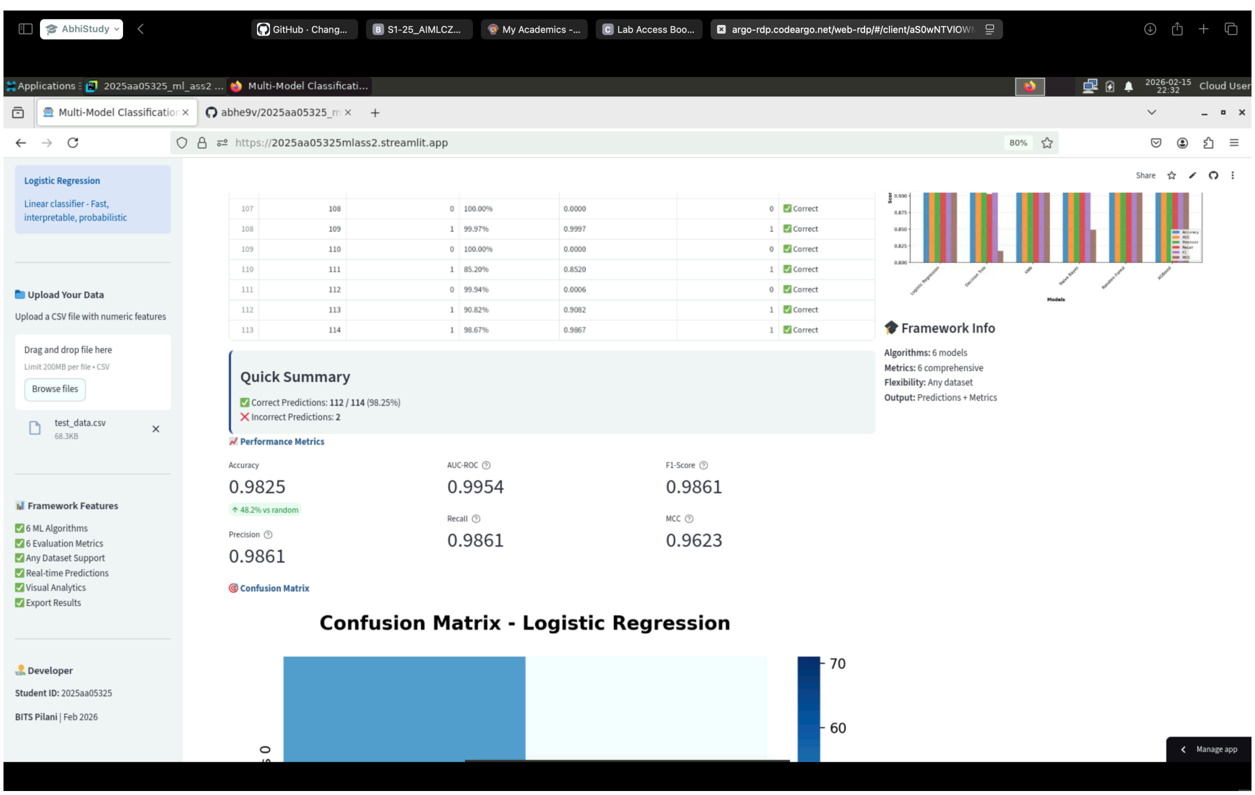
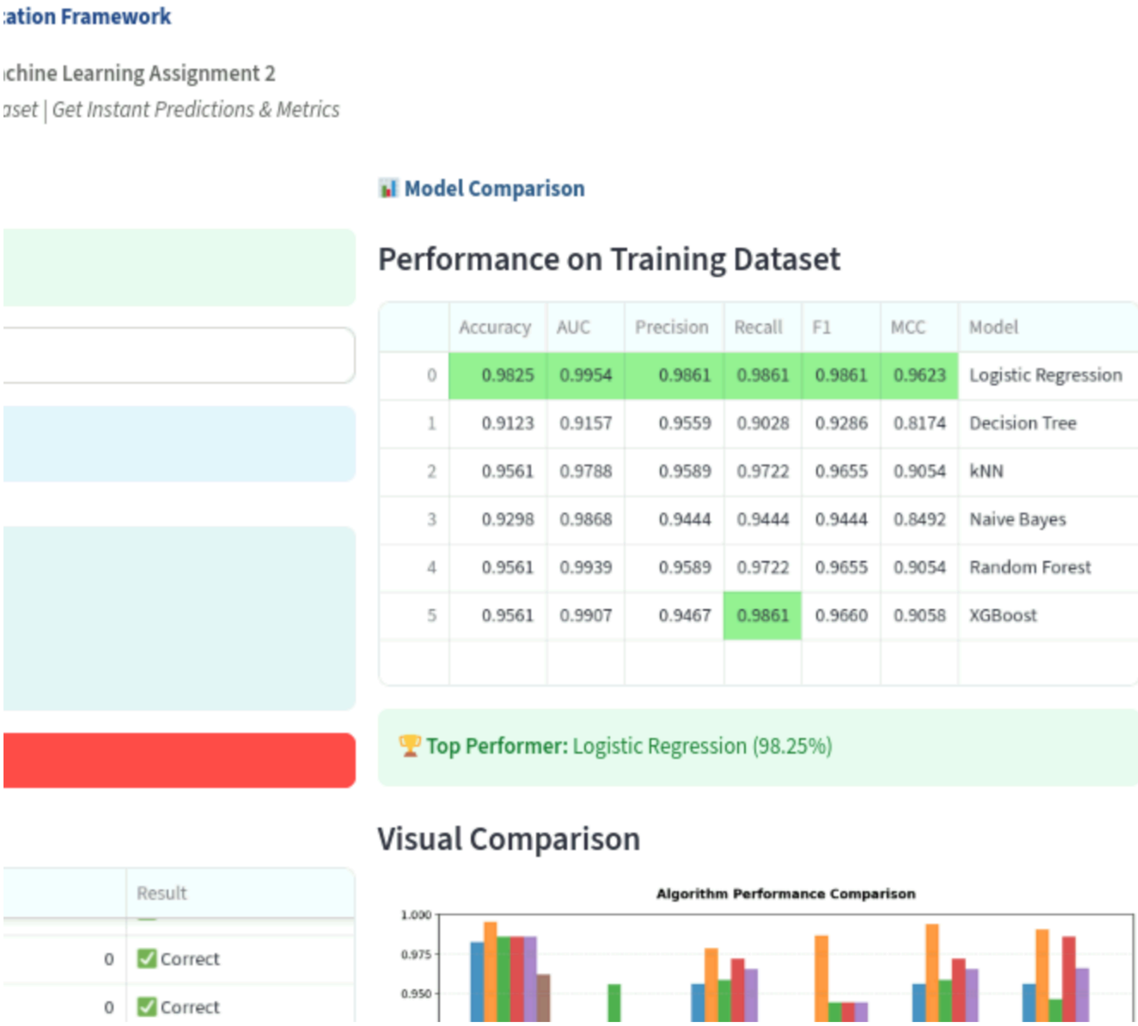


Figure 3: Prediction Results



	Accuracy	AUC	Precision	Recall	F1	MCC	Model
0	0.9825	0.9954	0.9861	0.9861	0.9861	0.9623	Logistic Regression
1	0.9123	0.9157	0.9559	0.9028	0.9286	0.8174	Decision Tree
2	0.9561	0.9788	0.9589	0.9722	0.9655	0.9054	kNN
3	0.9298	0.9868	0.9444	0.9444	0.9444	0.8492	Naive Bayes
4	0.9561	0.9939	0.9589	0.9722	0.9655	0.9054	Random Forest
5	0.9561	0.9907	0.9467	0.9861	0.9660	0.9058	XGBoost

Figure 4: Model Comparison

The framework is production-ready, well-documented, and serves as a solid foundation for future machine learning projects.

---

13 Appendix

13.1 Dependencies (requirements.txt)

```
streamlit>=1.28.0
scikit-learn>=1.3.0
numpy>=1.24.0,<2.0.0
pandas>=2.0.0
matplotlib>=3.9.0
seaborn>=0.12.0
scipy>=1.10.0
```

Submitted By:
Student ID: 2025aa05325
Name: Gaikwad Abhinav Rajaram
Course: Machine Learning Date: February 15, 2026