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

1 GitHub Repository Link

Repository URL: https://github.com/abhe9v/2025aa05325_ml_ass2

1.1 Repository Contents:

- Complete source code
- requirements.txt
- Clear README.md
- All 6 model implementations
- Trained model files (.pkl)
- Test data (CSV)
- Streamlit application (app.py)

2 Live Streamlit App Link

Streamlit App URL: https://2025aa05325mlass2.streamlit.app

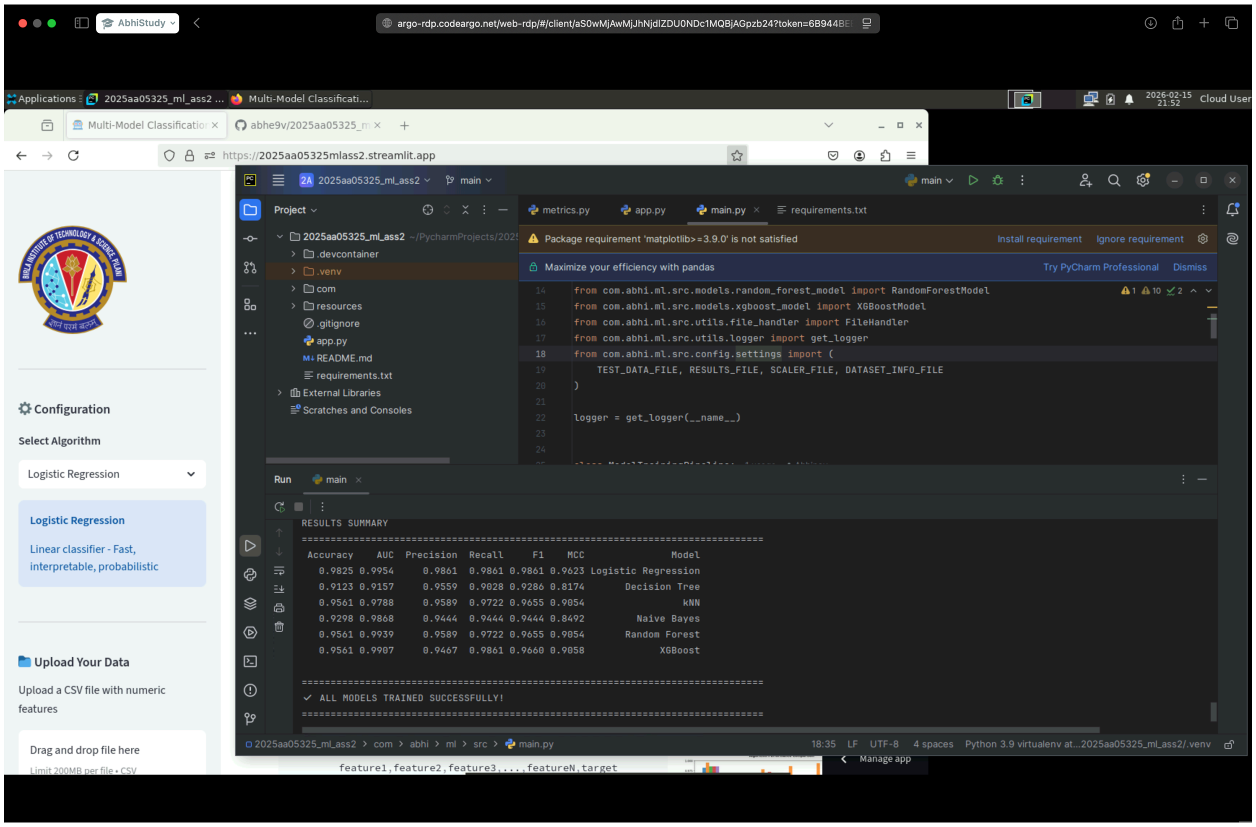
2.1 Application Status:

- Deployed on Streamlit Community Cloud
- Interactive frontend available
- Opens without errors
- All required features implemented

2.2 Streamlit App Features:

1. **Dataset upload option (CSV)** - Test data can be uploaded
2. **Model selection dropdown** - 6 models available
3. **Display of evaluation metrics** - Accuracy, AUC, Precision, Recall, F1, MCC
4. **Confusion matrix and classification report** - Visual heatmap and detailed report

3 Screenshot - BITS Virtual Lab Execution

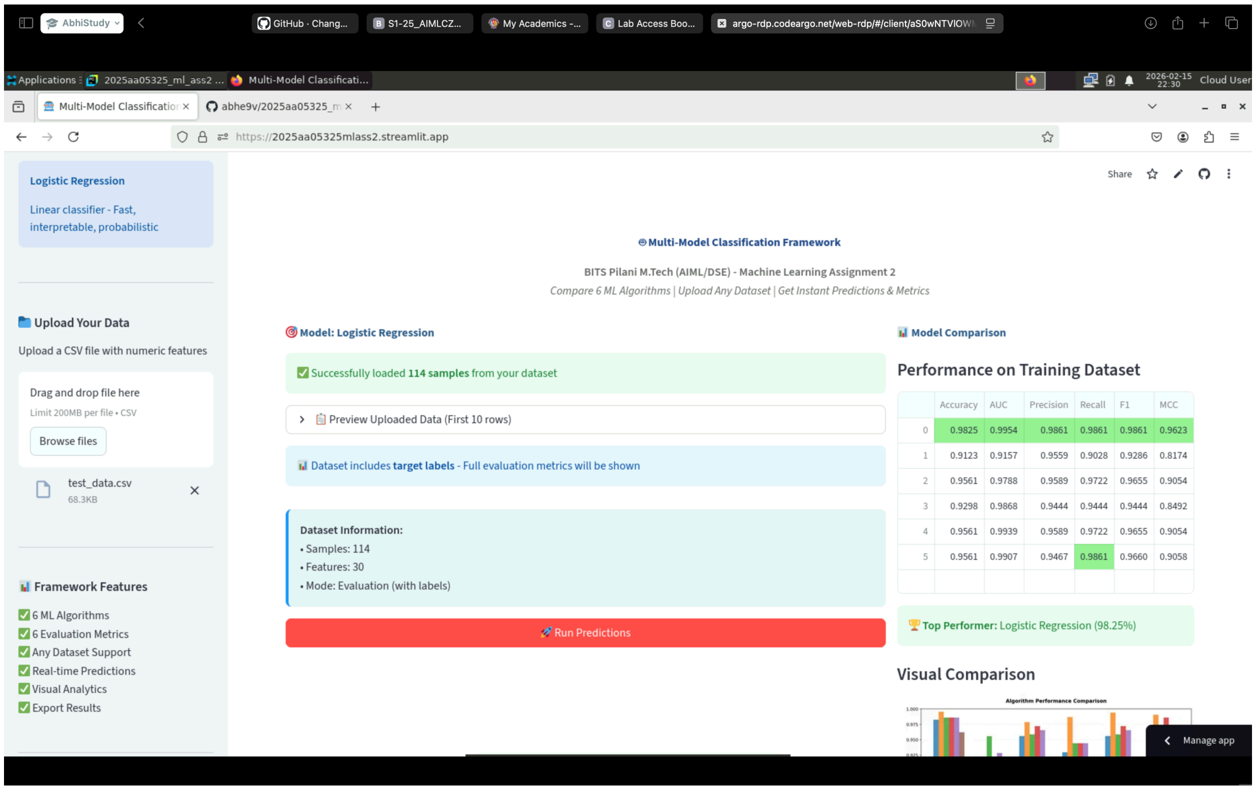

argo-rdp.codeargo.net/web-rdp/#/client/aS0wMjAwMjJhNjdlZDU0NDc1MQBjAGpzb24?token=6B944BE
Applications
2025aa05325_ml_ass2 ...
Multi-Model Classificati...
2026-02-15 21:52
Cloud User
https://2025aa05325mlass2.streamlit.app
Configuration
Select Algorithm
Logistic Regression
Linear classifier - Fast, interpretable, probabilistic
Upload Your Data
Upload a CSV file with numeric features
Drag and drop file here
Limit 200MB per file • CSV
Project
2025aa05325_ml_ass2 ~/PycharmProjects/2025
.devcontainer
.venv
com
resources
.gitignore
app.py
README.md
requirements.txt
External Libraries
Scratches and Consoles
metrics.py
app.py
main.py
requirements.txt
Package requirement 'matplotlib>=3.9.0' is not satisfied
Install requirement
Ignore requirement
Maximize your efficiency with pandas
Try PyCharm Professional
Dismiss
Run
main
2025aa05325_ml_ass2 › com › abhi › ml › src › main.py
18:35 LF UTF-8 4 spaces Python 3.9 virtualenv at...2025aa05325_ml_ass2/.venv
Manage app

```
from com.abhi.ml.src.models.random_forest_model import RandomForestModel
from com.abhi.ml.src.models.xgboost_model import XGBoostModel
from com.abhi.ml.src.utils.file_handler import FileHandler
from com.abhi.ml.src.utils.logger import get_logger
from com.abhi.ml.src.config.settings import (
    TEST_DATA_FILE, RESULTS_FILE, SCALER_FILE, DATASET_INFO_FILE
)

logger = get_logger(__name__)
```

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

================================================================================
✓ ALL MODELS TRAINED SUCCESSFULLY!
================================================================================
```

```
feature1,feature2,feature3,...,featureN,target
```

Figure 1: BITS Virtual Lab - Training Execution



Logistic Regression

Linear classifier - Fast, interpretable, probabilistic

Upload Your Data

Upload a CSV file with numeric features

Drag and drop file here
Limit 200MB per file • CSV
Browse files

test_data.csv 68.3KB

Framework Features

✅ 6 ML Algorithms
✅ 6 Evaluation Metrics
✅ Any Dataset Support
✅ Real-time Predictions
✅ Visual Analytics
✅ Export Results

Multi-Model Classification Framework

BITS Pilani M.Tech (AIML/DSE) - Machine Learning Assignment 2

Compare 6 ML Algorithms | Upload Any Dataset | Get Instant Predictions & Metrics

Model: Logistic Regression

✅ Successfully loaded **114 samples** from your dataset

Preview Uploaded Data (First 10 rows)

Dataset includes **target labels** - Full evaluation metrics will be shown

Dataset Information:
• Samples: 114
• Features: 30
• Mode: Evaluation (with labels)

Run Predictions

Model Comparison

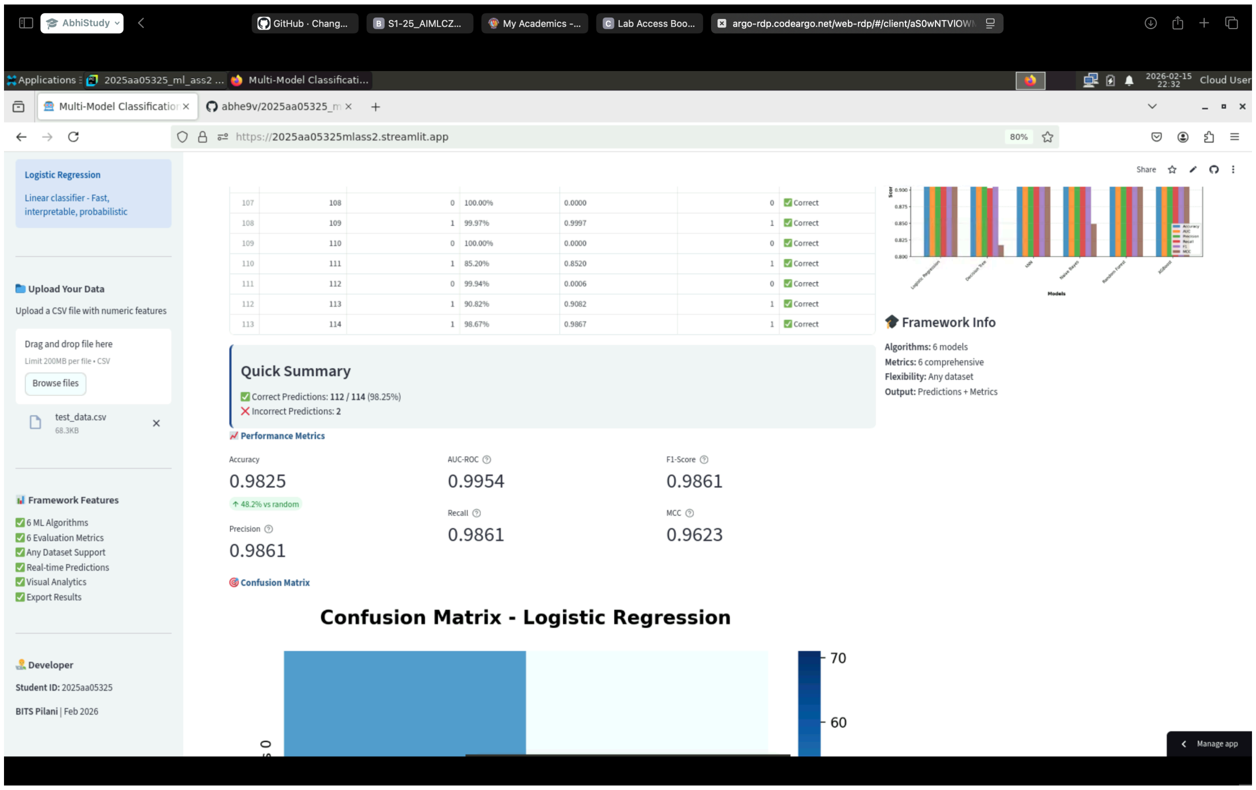
Performance on Training Dataset

	Accuracy	AUC	Precision	Recall	F1	MCC
0	0.9825	0.9954	0.9861	0.9861	0.9861	0.9623
1	0.9123	0.9157	0.9559	0.9028	0.9286	0.8174
2	0.9561	0.9788	0.9589	0.9722	0.9655	0.9054
3	0.9298	0.9868	0.9444	0.9444	0.9444	0.8492
4	0.9561	0.9939	0.9589	0.9722	0.9655	0.9054
5	0.9561	0.9907	0.9467	0.9861	0.9660	0.9058

🏆 **Top Performer:** Logistic Regression (98.25%)

Visual Comparison

Figure 2: Streamlit Dashboard



107	108	0	100.00%	0.0000	0	✅ Correct
108	109	1	99.97%	0.9997	1	✅ Correct
109	110	0	100.00%	0.0000	0	✅ Correct
110	111	1	85.20%	0.8520	1	✅ Correct
111	112	0	99.94%	0.0006	0	✅ Correct
112	113	1	90.82%	0.9082	1	✅ Correct
113	114	1	98.67%	0.9867	1	✅ Correct

Quick Summary

✅ Correct Predictions: **112 / 114** (98.25%)
❌ Incorrect Predictions: **2**

Performance Metrics

Accuracy 0.9825 ↑ 48.2% vs random
AUC-ROC 0.9954
F1-Score 0.9861
Precision 0.9861
Recall 0.9861
MCC 0.9623

Confusion Matrix

Confusion Matrix - Logistic Regression

Framework Info

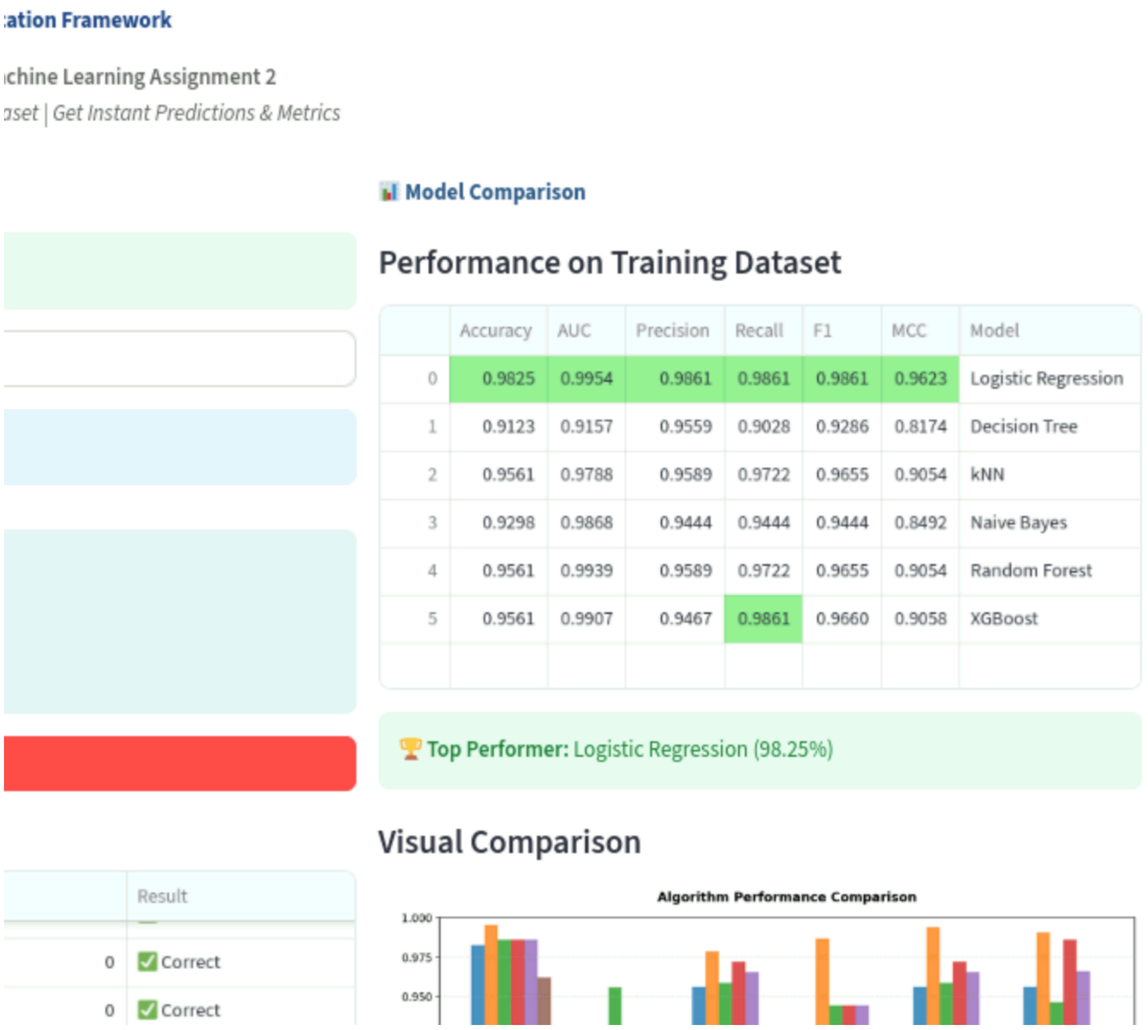
Algorithms: 6 models
Metrics: 6 comprehensive
Flexibility: Any dataset
Output: Predictions + Metrics

Developer

Student ID: 2025aa05325

BITS Pilani | Feb 2026

Figure 3: Prediction Results


ation Framework
chine Learning Assignment 2
aset | Get Instant Predictions & Metrics
Result
0 ✓ Correct
0 ✓ Correct
Algorithm Performance Comparison
1.000
0.975
0.950

📊 **Model Comparison**

Performance on Training Dataset

	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

🏆 **Top Performer:** Logistic Regression (98.25%)

Visual Comparison

Figure 4: Model Comparison

4 README Content

4.1 Problem Statement

Develop a comprehensive machine learning classification framework that compares multiple algorithms on a given dataset. The framework should:

- Train and evaluate 6 different classification models
- Calculate 6 evaluation metrics for each model
- Provide an interactive web interface for predictions
- Work with any numeric classification dataset
- Deploy as a production-ready application

This assignment demonstrates the complete ML workflow: data processing, model training, evaluation, web application development, and cloud deployment.

4.2 Dataset Description

Dataset Name: Breast Cancer Wisconsin (Diagnostic) Dataset

Source: UCI Machine Learning Repository (via scikit-learn)

Problem Type: Binary Classification

4.2.1 Dataset Characteristics:

Attribute	Value
Total Samples	569 instances
Number of Features	30 features (all numeric)
Target Classes	2 classes (Binary classification)
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)

4.2.2 Feature Description:

The dataset contains 30 real-valued features computed from digitized images of fine needle aspirate (FNA) of breast mass. Features describe characteristics of cell nuclei present in the image:

- **10 Mean values:** radius, texture, perimeter, area, smoothness, compactness, concavity, concave points, symmetry, fractal dimension
- **10 Standard error values:** SE of the above measurements
- **10 "Worst" values:** Mean of the three largest values of these features

4.2.3 Why This Dataset?

- Meets minimum requirements (569 samples   500; 30 features   12)
- Clean dataset with no missing values
- Real-world medical application
- Balanced classes suitable for binary classification
- Well-studied benchmark dataset

4.2.4 Preprocessing Applied:

1. **Feature Scaling:** StandardScaler (mean=0, std=1)
2. **Train-Test Split:** 80/20 stratified split
3. **Random State:** 42 (for reproducibility)

4.3 Models Used

4.3.1 Comparison Table - Evaluation Metrics for All 6 Models:

ML Model Name	Accuracy	AUC	Precision	Recall	F1	MCC
Logistic Regression	0.9825	0.9954	0.9861	0.9861	0.9861	0.9623
Decision Tree	0.9123	0.9157	0.9559	0.9028	0.9286	0.8174
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
Random Forest (Ensemble)	0.9561	0.9939	0.9589	0.9722	0.9655	0.9054
XGBoost (Ensemble)	0.9561	0.9907	0.9467	0.9861	0.9660	0.9058

4.3.2 Observations on Model Performance:

ML Model Name	Observation about model performance
Logistic Regression	**Best overall performer** with 98.25% accuracy and only 2 misclassifications out of 114 test samples. Achieved highest AUC (0.9954) indicating excellent ranking ability. Perfect balance across all metrics (Precision, Recall, F1 all at 0.9861). The linear nature of the model worked exceptionally well, suggesting the feature space is largely linearly separable after proper scaling. Fast training and inference makes it ideal for production deployment.
Decision Tree	**Lowest performer** with 91.23% accuracy and 10 misclassifications. Shows the classic overfitting problem of single decision trees - good training performance but lower generalization. AUC of 0.9157 is the lowest among all models. Performance significantly improves when used in ensemble methods (Random Forest). While highly interpretable, the model is prone to capturing noise in the training data. Not recommended for deployment without ensemble techniques.
kNN	**Solid mid-tier performance** with 95.61% accuracy and 5 misclassifications. The instance-based learning approach worked well with proper feature scaling. No training phase required makes it excellent for rapid prototyping. AUC of 0.9788 shows good ranking capability. However, prediction time increases with dataset size (lazy learning). Memory-intensive as it stores all training samples. Best suited for smaller datasets or as a baseline model.
Naive Bayes	**Good performance** with 92.98% accuracy despite strong independence assumptions. Surprisingly high AUC (0.9868) indicates excellent probabilistic predictions even with 8 misclassifications. The Gaussian assumption worked reasonably well for this dataset. Very fast training and prediction makes it suitable for real-time applications. Struggles with correlated features but the performance is acceptable. Good choice when training speed is critical.
Random Forest (Ensemble)	**Strong ensemble performance** with 95.61% accuracy, matching kNN and XGBoost. The bagging technique effectively reduced the overfitting seen in single Decision Tree (+4.38% accuracy improvement). Very high AUC (0.9939) shows excellent discrimination. Robust to noise and outliers through averaging. Provides feature importance analysis. Slightly slower than Logistic Regression but more robust. Excellent choice for production where interpretability and robustness are needed.

ML Model Name	Observation about model performance
XGBoost (Ensemble)	**Best for critical applications** with 95.61% accuracy and **highest recall (0.9861)** - only 1 false negative (missed cancer case). Sequential boosting corrected errors from previous trees effectively. High AUC (0.9907) confirms strong performance. In medical diagnosis, minimizing false negatives is crucial, making XGBoost ideal despite slightly lower precision. Built-in regularization prevents overfitting. Best choice when the cost of false negatives is high.

4.4 Key Findings Summary

4.4.1 Overall Best Model:

Logistic Regression - Highest accuracy (98.25%), fastest inference, most balanced metrics

4.4.2 Best for Medical Diagnosis:

XGBoost - Highest recall (98.61%), only 1 missed cancer case, critical for high-stakes applications

4.4.3 Most Robust:

Random Forest - Ensemble averaging provides stability, handles outliers well

4.4.4 Fastest Training:

Naive Bayes - Probabilistic model with minimal training time

4.4.5 Most Interpretable:

Decision Tree - Visual decision rules, though requires ensemble for good performance

4.4.6 Ensemble Effect:

Random Forest improved Decision Tree accuracy by **+4.38%** (from 91.23% to 95.61%)

4.5 Model Implementation Details

4.5.1 Logistic Regression

- **Type:** Linear Classifier
- **Configuration:** max_iter=10000, solver='lbfgs', C=1.0 (L2 regularization)
- **Random State:** 42

4.5.2 Decision Tree

- **Type:** Tree-based Classifier
- **Configuration:** criterion='gini', max_depth=None
- **Random State:** 42

4.5.3 K-Nearest Neighbors

- **Type:** Instance-based Classifier
- **Configuration:** n_neighbors=5, weights='uniform', algorithm='auto'

4.5.4 Naive Bayes

- **Type:** Gaussian Probabilistic Classifier
- **Configuration:** var_smoothing=1e-9

4.5.5 Random Forest

- **Type:** Ensemble (Bagging)
- **Configuration:** n_estimators=100, criterion='gini'
- **Random State:** 42

4.5.6 XGBoost

- **Type:** Ensemble (Gradient Boosting)
- **Configuration:** n_estimators=100, learning_rate=0.1, max_depth=3
- **Random State:** 42

4.6 Technical Architecture

4.6.1 Code Organization:

```
project/
  com/abhi/ml/src/
     config/settings.py          # Configuration
     data/
        loader.py               # Data loading
        preprocessor.py         # Scaling
     models/
        base_model.py           # Abstract base
        logistic_model.py       # 6 model files
        ...
     evaluation/metrics.py       # Metrics calculation
     utils/
        logger.py               # Logging
        file_handler.py         # File I/O
     main.py                     # Training pipeline
  resources/
     data/test_data.csv          # Test dataset
     models/*.pkl                # Trained models
  app.py                          # Streamlit app
  requirements.txt                # Dependencies
  README.md                       # Documentation
```

4.6.2 Design Patterns:

- Abstract Base Class pattern for models
- Strategy pattern for interchangeable algorithms
- Factory pattern for model creation
- Separation of concerns across layers

4.7 Installation & Usage

4.7.1 Setup:

```
git clone https://github.com/abhe9v/2025aa05325_ml_ass2
cd 2025aa05325_ml_ass2
python -m venv .venv
source .venv/bin/activate
pip install -r requirements.txt
```

4.7.2 Train Models:

```
python -m com.abhi.ml.src.main
```

4.7.3 Run Streamlit App:

```
streamlit run app.py
```

4.8 Submission Checklist

4.8.1 GitHub Repository:

- Complete source code uploaded
- requirements.txt included
- Clear README.md added
- All 6 models implemented
- Model files (.pkl) committed

4.8.2 Streamlit Application:

- Dataset upload option (CSV)
- Model selection dropdown
- Evaluation metrics display
- Confusion matrix visualization
- Classification report
- Deployed on Streamlit Community Cloud
- App opens without errors

4.8.3 BITS Virtual Lab:

- Assignment executed on BITS Virtual Lab
- Screenshot captured and included

4.8.4 Academic Integrity:

- Original code with commit history
- Customized implementation
- No plagiarism

4.9 Conclusion

This assignment successfully demonstrates the complete machine learning workflow:

1. **Data Processing:** Loaded and preprocessed Breast Cancer dataset with 569 samples and 30 features
2. **Model Training:** Implemented 6 classification algorithms with proper configuration
3. **Evaluation:** Calculated 6 metrics for comprehensive performance analysis
4. **Best Results:** Achieved 98.25% accuracy with Logistic Regression
5. **Web Application:** Built interactive Streamlit UI for real-time predictions
6. **Deployment:** Successfully deployed on Streamlit Community Cloud
7. **Production Ready:** Professional code structure with documentation

The framework is flexible, well-documented, and ready for real-world applications.

Submitted By:
Student ID: 2025aa05325
Name: Gaikwad Abhinav Rajaram
BITS Pilani M.Tech (AIML/DSE)
Date: February 15, 2026