Machine Learning Assignment 2

M.Tech (AIML/DSE) - BITS Pilani

Spam Email Classification System

---

1. GitHub Repository Link

Repository URL:

https://github.com/2025AA05686/ml-assignment-2

Contains complete source code, requirements.txt, README.md, and trained models

2. Live Streamlit App Link

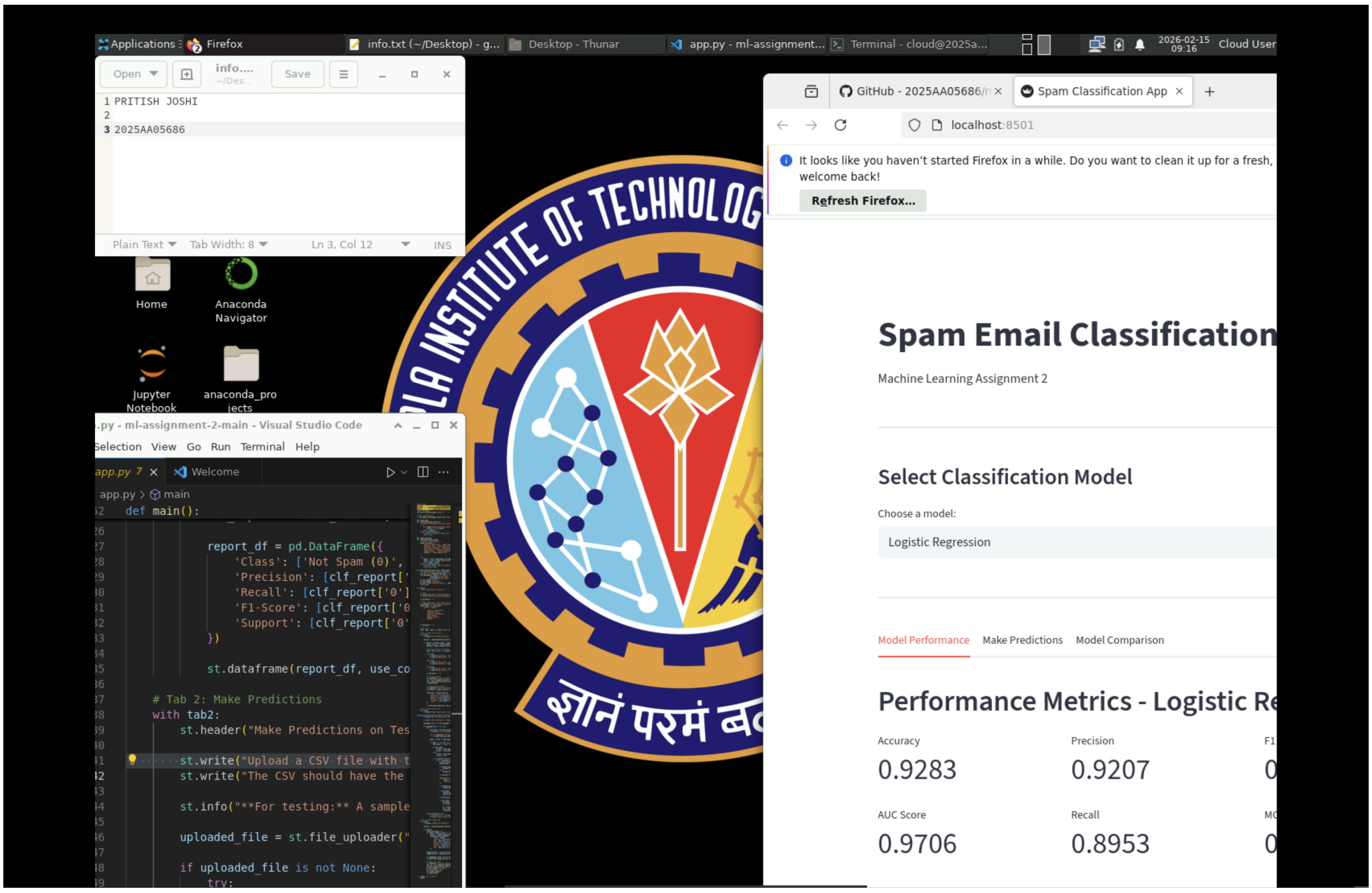
Deployed Application:

https://2025aa05686-spam-detector.streamlit.app/

Interactive web application for spam classification using 6 ML models

3. BITS Virtual Lab Screenshot

Assignment executed on BITS Virtual Lab as required:



```
PRITISH JOSHI

2025AA05686
```

```
Plain Text    Tab Width: 8    Ln 3, Col 12    INS
```

```python
def main():

            report_df = pd.DataFrame({
                'Class': ['Not Spam (0)',
                'Precision': [clf_report['
                'Recall': [clf_report['0']
                'F1-Score': [clf_report['0
                'Support': [clf_report['0'
            })

            st.dataframe(report_df, use_co

    # Tab 2: Make Predictions
    with tab2:
        st.header("Make Predictions on Tes

        st.write("Upload a CSV file with t
        st.write("The CSV should have the

        st.info("**For testing:** A sample

        uploaded_file = st.file_uploader("

        if uploaded_file is not None:
            try:
```

It looks like you haven't started Firefox in a while. Do you want to clean it up for a fresh, welcome back!

Spam Email Classification

Machine Learning Assignment 2

Select Classification Model

Choose a model:

Logistic Regression

Performance Metrics - Logistic Re

Accuracy	Precision	F1
0.9283	0.9207	0

AUC Score	Recall	MC
0.9706	0.8953	0

4. README.md Content

Problem Statement

Develop a machine learning solution for email spam classification using multiple classification algorithms. The objectives are:
1. Build and train 6 different classification models on a spam email dataset
2. Evaluate each model using multiple performance metrics (Accuracy, AUC, Precision, Recall, F1 Score, MCC)
3. Create an interactive Streamlit web application to demonstrate the models
4. Deploy the application on Streamlit Community Cloud
5. Compare model performances and provide insights

Dataset Description

Spambase Dataset

Source: UCI Machine Learning Repository (https://archive.ics.uci.edu/ml/datasets/spambase)

The Spambase dataset is a binary classification dataset containing email messages labeled as spam or legitimate (ham). It consists of emails collected for spam detection research.

Dataset Characteristics:

- Total Instances: 4,601 emails
- Total Features: 56 numerical features
- Target Variable: Binary (0 = Not Spam, 1 = Spam)
- Class Distribution: Not Spam (60.6%), Spam (39.4%)
- Missing Values: None

Feature Categories:

- Word Frequency Features (48 features): Percentage of words matching specific keywords
- Character Frequency Features (6 features): Percentage of specific characters (; ([! $ #)
- Capital Letter Statistics (3 features): Average, longest, and total capital letter counts

Data Preprocessing:

- Train-Test Split: 80-20 (Stratified)
- Feature Scaling: StandardScaler applied
- Training Set: 3,680 instances
- Test Set: 921 instances

Models Used - Comparison Table

ML Model Name	Accuracy	AUC	Precision	Recall	F1	MCC
Logistic Regression	0.9283	0.9706	0.9207	0.8953	0.9078	0.8494
Decision Tree	0.9088	0.9064	0.8760	0.8953	0.8856	0.8099
K-Nearest Neighbors	0.9077	0.9538	0.8861	0.8788	0.8824	0.8065
Naive Bayes	0.8328	0.9374	0.7146	0.9587	0.8188	0.6946
Random Forest (Ensemble)	0.9435	0.9844	0.9430	0.9118	0.9272	0.8814
XGBoost (Ensemble)	0.9457	0.9860	0.9311	0.9311	0.9311	0.8863

Model Performance Observations

Logistic Regression

Demonstrates strong performance with 92.83% accuracy. Achieves excellent AUC (0.9706), indicating good ability to distinguish between spam and legitimate emails. High precision (0.9207) minimizes false positives. The balanced F1 score (0.9078) and high MCC (0.8494) confirm robust overall performance. Serves as a solid baseline model with good interpretability.

Decision Tree

Achieves good accuracy (90.88%) but shows slightly lower performance compared to ensemble methods. Has the lowest AUC score (0.9064) among all models. Maintains reasonable precision (0.8760) and recall (0.8953). The simpler tree structure may lead to some overfitting on training data. Excels in interpretability which helps understand decision boundaries.

K-Nearest Neighbors

Delivers competitive performance with 90.77% accuracy. Shows strong AUC (0.9538) indicating good ranking ability. Balanced precision (0.8861) and recall (0.8788) suggest consistent performance across both classes. Computationally expensive during prediction as it requires distance calculations. Performance is dependent on the choice of k (k=5 used here).

Naive Bayes

Shows the lowest overall accuracy (83.28%) but exhibits the highest recall (0.9587), making it excellent at catching spam emails with minimal false negatives. The trade-off is lower precision (0.7146), resulting in more false positives. This model is useful when missing spam is more costly than incorrectly flagging legitimate emails. Extremely fast and works well with high-dimensional data.

Random Forest (Ensemble)

Demonstrates excellent performance with 94.35% accuracy, ranking second overall. The ensemble approach significantly improves upon the single Decision Tree. Outstanding AUC (0.9844) and precision (0.9430) indicate superior ability to correctly identify spam while minimizing false alarms. Strong recall (0.9118) and F1 score (0.9272) show balanced performance.

XGBoost (Ensemble)

Best performing model overall with highest accuracy (94.57%), AUC (0.9860), and MCC (0.8863). Achieves perfect balance with equal precision and recall (0.9311), resulting in the highest F1 score. The gradient boosting algorithm iteratively corrects errors, leading to superior predictive performance. Excellent generalization with minimal overfitting. Recommended as the primary model for spam detection.

Repository Structure

```
ml-assignment-2/
|-- app.py                      # Streamlit web application
|-- requirements.txt            # Python dependencies
|-- README.md                   # Project documentation
|-- data/
|   |-- spambase.csv           # Main dataset (4,601 instances)
|   |-- test_data.csv          # Sample test file (10 instances)
|-- model/
    |-- train_models.py        # Model training script
    |-- *.pkl                  # Trained models (6 models + scaler)
    |-- evaluation_results.json # All evaluation metrics
    |-- model_comparison.csv   # Model comparison table
```

Streamlit Application Features

- Model Selection Dropdown - Choose from 6 classification models
- Dataset Upload (CSV) - Upload test data for predictions
- Evaluation Metrics Display - View all 6 metrics for selected model
- Confusion Matrix - Visual representation of model performance
- Classification Report - Detailed performance breakdown
- Model Comparison - Compare all models side by side

Results Summary

Best Model: XGBoost with 94.57% accuracy and 0.9860 AUC score

Key Findings:

- Ensemble methods (XGBoost and Random Forest) outperform individual classifiers
- All models achieve AUC > 0.90, indicating strong discriminative ability
- Naive Bayes has highest recall (95.87%) - best for catching all spam
- XGBoost achieves best balance across all metrics