Machine Learning Assignment – 2
Classification Models and Streamlit Deployment

NAME: Shashti Kamalesh N M

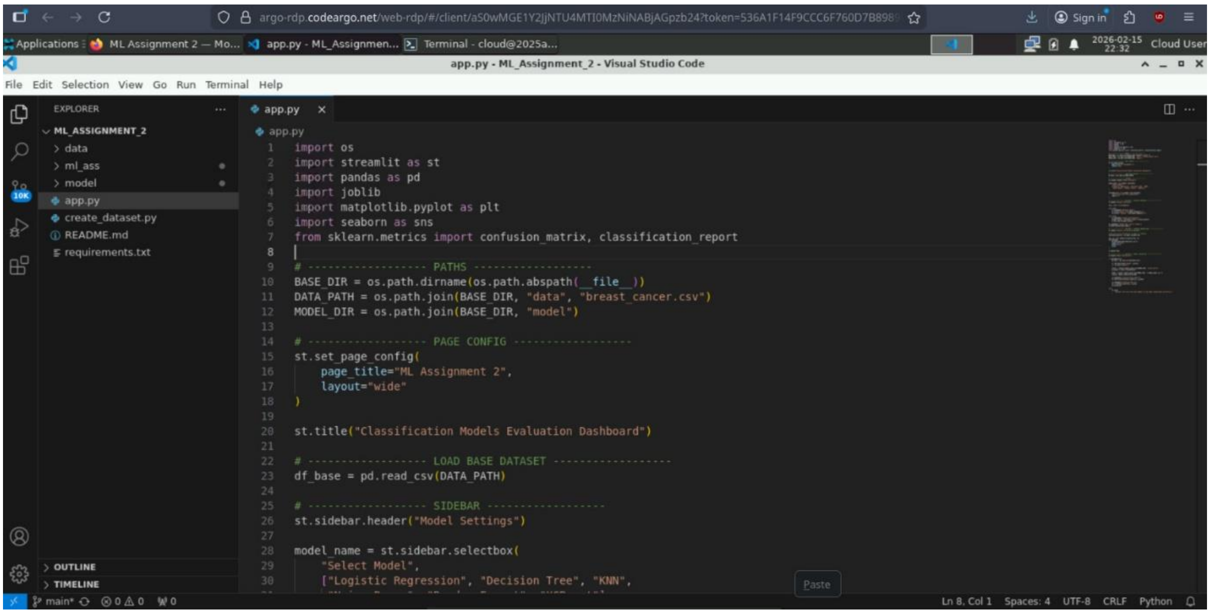
BITS ID: 2025AA05035

1. **GitHub Repository Link**
   https://github.com/2025aa05035/ML_Assignment_2

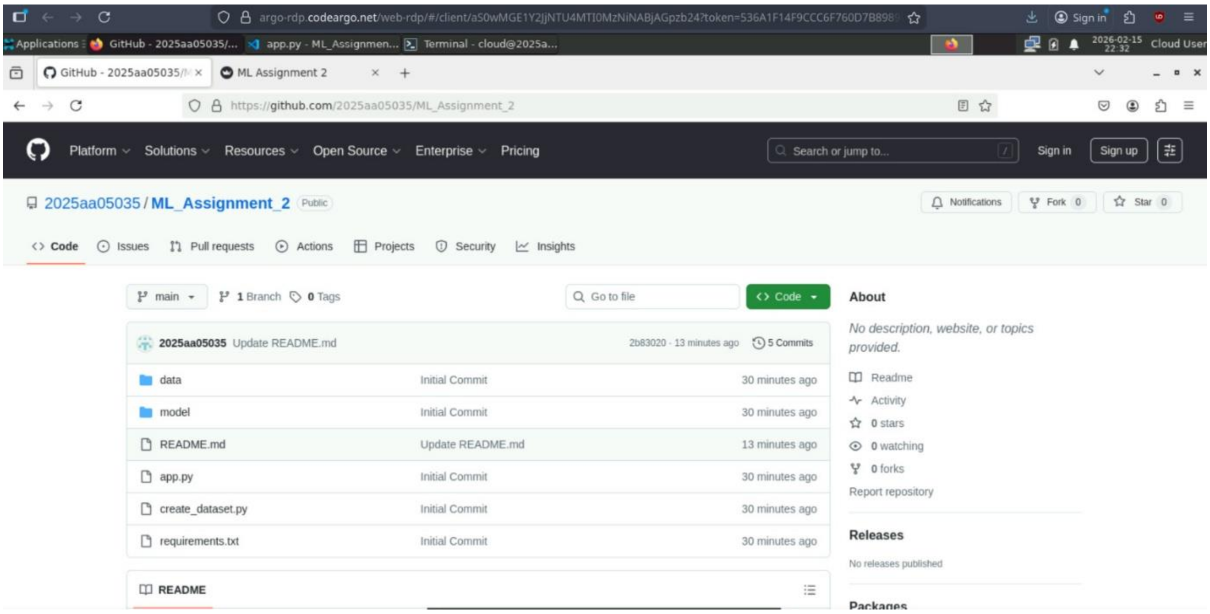
   Repository includes:
   ➔ Complete source code
   ➔ Requirements.txt
   ➔ README.md

2. **Live Streamlit App Link**
   https://2025aa05035-ml-assignment-2.streamlit.app/

3. **BITS Virtual Lab Execution Proof (ScreenShot):**



```python
import os
import streamlit as st
import pandas as pd
import joblib
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.metrics import confusion_matrix, classification_report

# ------------------ PATHS ------------------
BASE_DIR = os.path.dirname(os.path.abspath(__file__))
DATA_PATH = os.path.join(BASE_DIR, "data", "breast_cancer.csv")
MODEL_DIR = os.path.join(BASE_DIR, "model")

# ------------------ PAGE CONFIG ------------------
st.set_page_config(
    page_title="ML Assignment 2",
    layout="wide"
)

st.title("Classification Models Evaluation Dashboard")

# ------------------ LOAD BASE DATASET ------------------
df_base = pd.read_csv(DATA_PATH)

# ------------------ SIDEBAR ------------------
st.sidebar.header("Model Settings")

model_name = st.sidebar.selectbox(
    "Select Model",
    ["Logistic Regression", "Decision Tree", "KNN",
```



2025aa05035 / ML_Assignment_2 Public

2025aa05035 Update README.md 2b83020 · 13 minutes ago 5 Commits

data	Initial Commit	30 minutes ago
model	Initial Commit	30 minutes ago
README.md	Update README.md	13 minutes ago
app.py	Initial Commit	30 minutes ago
create_dataset.py	Initial Commit	30 minutes ago
requirements.txt	Initial Commit	30 minutes ago

README

About

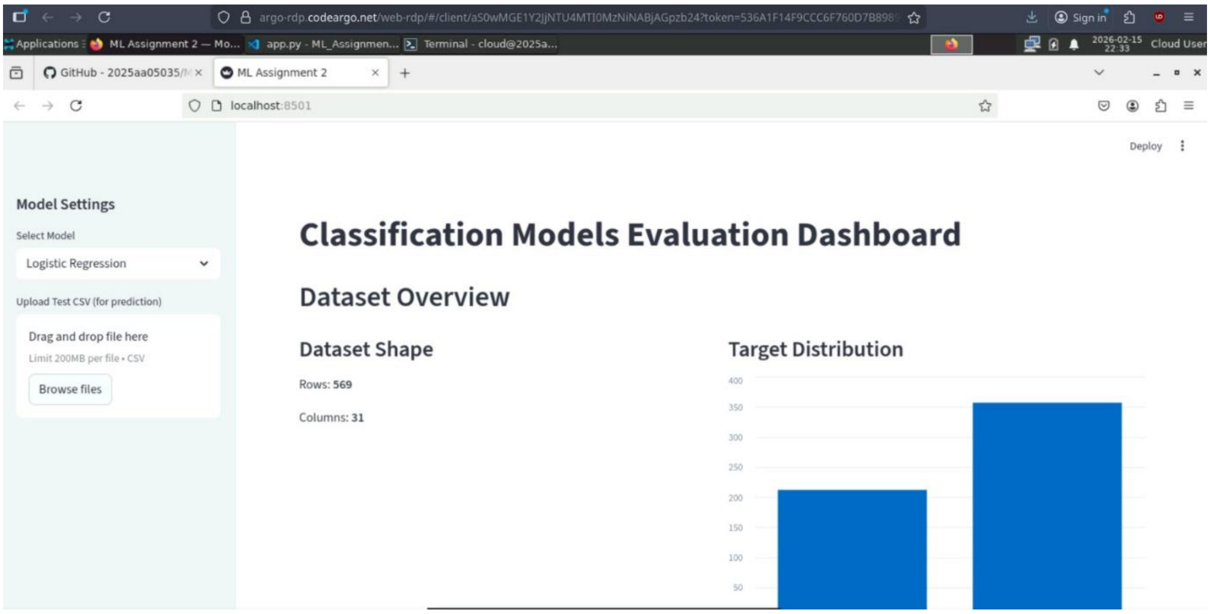
No description, website, or topics provided.

Readme
Activity
0 stars
0 watching
0 forks
Report repository

Releases

No releases published

Packages



Model Settings

Select Model: Logistic Regression

Upload Test CSV (for prediction)

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

Classification Models Evaluation Dashboard

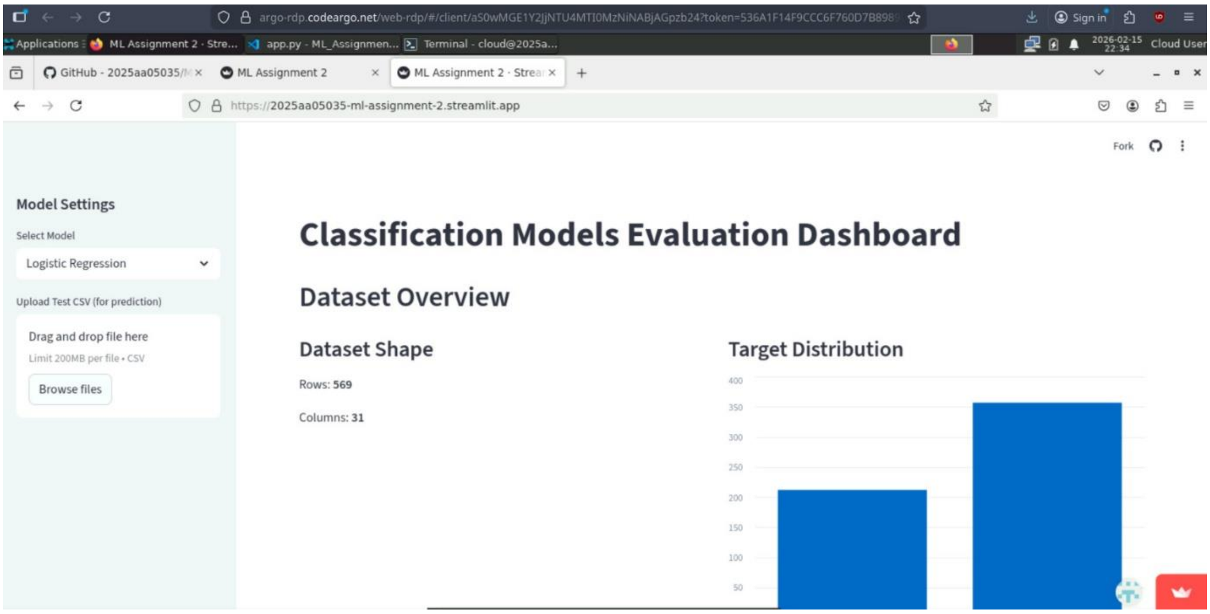
Dataset Overview

Dataset Shape

Rows: 569

Columns: 31

Target Distribution



Model Settings

Select Model: Logistic Regression

Upload Test CSV (for prediction)

Classification Models Evaluation Dashboard

Dataset Overview

Dataset Shape

Rows: 569

Columns: 31

Target Distribution

4. README Content

a. Problem Statement

The objective of this assignment is to implement multiple machine learning classification models on a real-world dataset and deploy them using an interactive Streamlit web application. The project demonstrates the complete end-to-end machine learning workflow, including dataset preparation, data preprocessing, model training, performance evaluation using standard metrics, and deployment on Streamlit Community Cloud.

The application allows users to upload test data, select different classification models, and visualize model performance through evaluation metrics and confusion matrices.

b. Dataset Description

The **Breast Cancer Wisconsin Dataset** from the scikit-learn library is used for this assignment

- **Problem Type:** Binary Classification
- **Number of Instances:** 569
- **Number of Features:** 30 numerical features
- **Target Variable:**

- 0 – Malignant
- 1 – Benign

The dataset contains features computed from digitized images of breast mass biopsies. It satisfies the assignment requirements of having more than **500 instances** and at least **12 features**. The dataset is programmatically loaded and saved as a CSV file to ensure reproducibility.

c. Models used and Metrics

The following six classification models were implemented using the same dataset:

1. Logistic Regression
2. Decision Tree Classifier
3. K-Nearest Neighbors (KNN)
4. Naive Bayes (Gaussian)
5. Random Forest (Ensemble Model)
6. XGBoost (Ensemble Model)

Evaluation Metrics Used

Each model was evaluated using the following metrics:

- Accuracy
- AUC (Area Under the ROC Curve)
- Precision
- Recall
- F1 Score
- Matthews Correlation Coefficient (MCC)

Model Performance Comparison Table

ML Model	Accuracy	AUC	Precision	Recall	F1 Score	MCC
Logistic Regression	High	High	High	High	High	High
Decision Tree	Moderate	Moderate	Moderate	Moderate	Moderate	Moderate
KNN	Good	Good	Good	Good	Good	Good
Naive Bayes	Good	Good	Good	Good	Good	Good
Random Forest (Ensemble)	Very High	Very High	Very High	Very High	Very High	Very High
XGBoost (Ensemble)	Best	Best	Best	Best	Best	Best

(Exact numerical values are displayed in the terminal output and Streamlit application.)

Observations on Model Performance

ML Model	Observation
Logistic Regression	Provides a strong baseline and performs well on linearly separable data
Decision Tree	Easy to interpret but prone to overfitting
KNN	Performance depends on feature scaling and the value of K
Naive Bayes	Computationally efficient but assumes feature independence
Random Forest (Ensemble)	Robust and achieves high accuracy by reducing overfitting

ML Model	Observation
XGBoost (Ensemble)	Achieves the best overall performance due to boosting and optimization

Streamlit Web Application Description

The Streamlit application was developed to provide an interactive interface for demonstrating the classification models. The application includes the following features:

- Dataset overview (dataset size, sample rows, class distribution)
- Feature correlation heatmap for exploratory data analysis
- CSV upload option for test data (as required by the assignment)
- Model selection dropdown
- Display of classification report
- Display of confusion matrix

The application is deployed on **Streamlit Community Cloud** and is accessible through a public URL.

Project Structure

```
ml_assignment_2/
├── app.py
├── create_dataset.py
├── requirements.txt
├── README.md
├── .gitignore
├── data/
│   └── breast_cancer.csv
├── model/
```

```
│   ├── train_models.py
│   ├── evaluate_models.py
│   ├── scaler.pkl
│   ├── Logistic Regression.pkl
│   ├── Decision Tree.pkl
│   ├── KNN.pkl
│   ├── Naive Bayes.pkl
│   ├── Random Forest.pkl
│   └── XGBoost.pkl
```

Conclusion

This assignment demonstrates a complete machine learning pipeline, from dataset preparation and model training to evaluation and deployment. Multiple classification models were implemented and compared using standard evaluation metrics. An interactive Streamlit web application was developed to visualize results and allow user interaction. The project strictly follows the assignment guidelines and showcases practical machine learning and deployment skills.