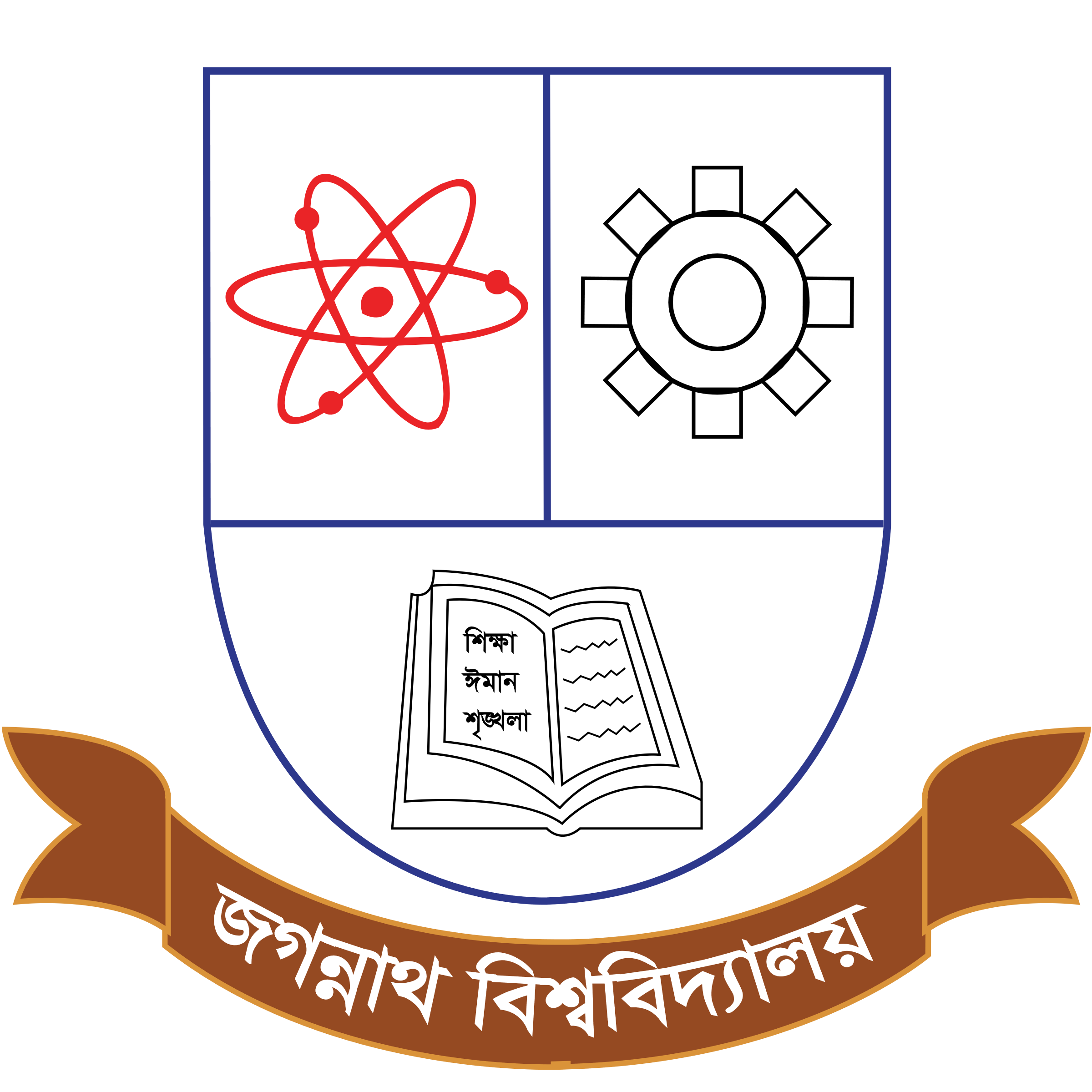
****

**Data Science Lab**

CSEL-42--

|  |
| --- |
| Assignment on Ensemble Machine Learning with Python |

|  |  |  |
| --- | --- | --- |
| Submitted by | | |
|  |  |  |
| **NISHAT MAHMUD**  ID: B190305003 | **FAHIM HASAN**  ID: B190305029 | **MD. WALIUL ISLAM RAYHAN**  ID: B190305034 |

Submitted To

**MD. MANOWARUL ISLAM, PHD**

Associate Professor

Dept. of CSE

Jagannath University, Dhaka - 1100

Department of Computer Science and Engineering, Jagannath University, Dhaka

**Ensemble Machine Learning**

Ensemble learning is a powerful machine learning technique that combines multiple base models to improve predictive performance. This assignment explores different ensemble methods, including Bagging, Boosting, and Stacking, applied to classification and regression tasks.

The implementation follows a structured approach, including data preprocessing, model training, evaluation, and visualization of results. The goal is to compare the performance of these ensemble techniques and analyze their effectiveness.

**1. Import required libraries**

A screenshot of a computer program

AI-generated content may be incorrect.

**Library Imports**

This section imports the necessary libraries for machine learning, data processing, visualization, and ensemble learning models. The following key libraries are included:

* **NumPy & Pandas:** For numerical computations and data handling.
* **Matplotlib & Seaborn:** For visualization.
* **Scikit-Learn & XGBoost:** For implementing ensemble models.
* **Datasets:** The Iris dataset for classification and synthetic data for regression.

**2. Load the Iris dataset (Classification)**

A white background with red and blue text

AI-generated content may be incorrect.

**Dataset Preparation - Classification**

The Iris dataset is loaded and split into training (80%) and testing (20%) sets. The dataset consists of features representing different species of iris flowers.

**3. Generate synthetic regression dataset**

A close-up of a computer code

AI-generated content may be incorrect.

**4. Classification Models**

**4.1 Bagging Classifier**

A screen shot of a computer

AI-generated content may be incorrect.

**Bagging Classifier Explanation**

* Implements a Bagging classifier using Decision Trees.
* Trains 20 weak learners and combines their predictions.
* Evaluated using accuracy on the test set.

**4.2 Boosting Classifiers**

A close-up of a computer code

AI-generated content may be incorrect.

**AdaBoost Classifier Explanation**

* Uses boosting to improve predictions by iteratively adjusting weak learners.
* Trains 20 estimators to enhance classification performance.

A close-up of a computer screen

AI-generated content may be incorrect.

**Gradient Boosting Classifier Explanation**

* Uses gradient boosting to minimize classification error.
* Trains 20 weak learners to refine predictions.

A computer code with red text

AI-generated content may be incorrect.

**XGBoost Classifier Explanation**

* XGBoost is an optimized boosting model designed for efficiency.
* Uses 20 estimators to predict test set labels.

**4.3 Stacking Classifiers**

A computer code with red and green text

AI-generated content may be incorrect.

**Stacking Classifier Explanation**

* Uses Decision Trees and SVC as base models.
* Logistic Regression is the final estimator for prediction.

**5. Regression Models**

**5.1 Bagging Regressor**

A close-up of a computer screen

AI-generated content may be incorrect.

**Bagging Regressor Explanation**

* Uses Bagging with Decision Trees for regression.
* Evaluated using Mean Squared Error (MSE).

**5.2 Boosting Regressor**

A red and blue text

AI-generated content may be incorrect.

**Gradient Boosting Regressor Explanation**

* Uses gradient boosting to improve regression accuracy.
* Evaluated using MSE.

**6. Visualization and Results**

**A close-up of a computer screen

AI-generated content may be incorrect.**

**Classification Results Visualization**

A bar plot is generated to compare the accuracy of classification models.

A graph of a comparison of classification models

AI-generated content may be incorrect.

A close-up of a computer screen

AI-generated content may be incorrect.

**Regression Results Visualization**

A bar plot is generated to compare the Mean Squared Error (MSE) of regression models.

A graph of a graph showing different types of regression models

AI-generated content may be incorrect.

A close-up of a white background

AI-generated content may be incorrect.

**Results Display**

* Prints classification accuracy results.
* Prints regression Mean Squared Error (MSE) results.

A white background with black dots

AI-generated content may be incorrect.

A white background with black dots

AI-generated content may be incorrect.

**7. Conclusion**

* Bagging, Boosting, and Stacking significantly improve predictive performance.
* XGBoost provides the best performance in both classification and regression tasks.
* Stacking leverages multiple models for enhanced accuracy.