Green University Of Bangladesh

Department Of Computer Science and Engineering (CSE)

Faculty of Sciences and Engineering

Semester: (Fall, Year: 2023), B.Sc. in CSE (DAY)

LAB REPORT NO - 09

Course Title: Data Mining Lab

Course Code: CSE-436 **Section:** D2

Lab Experiment Name: Implementation of stacking ensemble and AdaBoost ensemble

Student Details

	Name	ID
1	Shamim Ahmed	201902067

Lab Date : 29th December 2023
Submission Date : 5th January 2024
Course Teacher Name : Meherunnesa Tania

Lab Report Status

Mark:.......................... Signature:...............
Comments:.................. Date:......................

1 INTRODUCTION

In this lab report we are going to develop a Colab using Machine Learning Concept: stacking ensemble and AdaBoost ensemble. We will Choose an appropriate dataset Then we will implement the stacking ensemble and AdaBoost ensemble and analysis the result.

2 OBJECTIVE

The aim of this lab is to learn about Machine Learning Concept: stacking ensemble and AdaBoost ensemble and to evaluate its. Here we will have a clear idea about the stacking ensemble and AdaBoost ensemble and we are going to compare this two with a dataset and evalute the model.

3 THEORY

3.1 Stacking

Stacking is an ensemble learning technique where multiple diverse models, known as base models, are trained on the data. Their predictions serve as input features for a higher-level model, called the meta-model or blender, which combines these predictions to make final predictions. Stacking leverages the strengths of different models, potentially improving overall performance by capturing complex patterns in the data which we can see in figure 2. The base models can be of various types, and common choices for meta-models include linear models or more complex models. The process involves training base models, using them to predict, and then training a meta-model on those predictions.

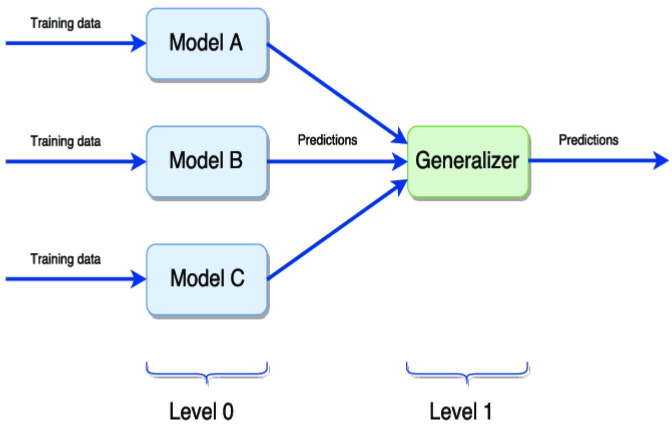

Figure 1: stacking ensemble learning

3.2 AdaBoost

AdaBoost, short for Adaptive Boosting, is an ensemble learning method that constructs a robust and accurate classifier by combining the outputs of multiple weak classifiers. In AdaBoost, weak classifiers—models slightly better than random chance, often simple decision trees—are trained iteratively. The algorithm assigns weights to training examples, emphasizing misclassified instances in subsequent iterations. The weights are updated to give more importance to incorrectly classified examples, guiding the weak classifiers to focus on challenging instances. By combining the predictions of these weighted weak classifiers, AdaBoost creates a strong, adaptive model that excels even in the presence of noise or outliers we can see in figure **??**. The final model is a weighted sum of the weak classifiers, providing a powerful and accurate predictive model.

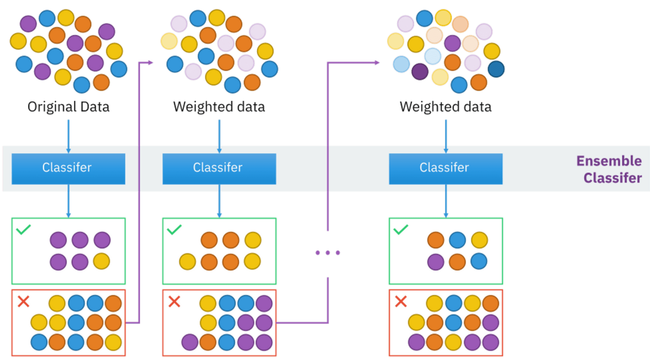

Figure 2: AdaBoost ensemble learning

4 IMPLEMENTATION

4.1 DATASET

For this experiment, I used the "diabetes.csv" dataset. The "diabetes.csv" dataset is typically used for binary classification tasks, meaning there are two classes for the target variable. The target variable in this dataset is often denoted as "Outcome," and it indicates the presence or absence of diabetes. The classes are typically represented as 1 for the presence of diabetes and 0 for the absence of diabetes.

4.2 Data Splitting

4.2.1 Training Set (80% of the data)

80% of the dataset is used for training the model. The training set contains pairs of 'X' and corresponding target values 'y'.

4.2.2 Testing Set (20% of the data)

20% of the dataset is reserved for evaluating the performance of the trained model. The testing set also contains pairs of 'X' and corresponding target values 'y'.

4.3 Implementation of stacking ensemble and AdaBoost ensemble

```python
# Import necessary libraries
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.ensemble import StackingClassifier, AdaBoostClassifier
from sklearn.tree import DecisionTreeClassifier
from sklearn.ensemble import RandomForestClassifier
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import accuracy_score
from sklearn.preprocessing import LabelEncoder

# Load the diabetes dataset from a local file
file_path = "/content/diabetes.csv"
dataset = pd.read_csv(file_path)

# Print data types of each column
print(dataset.dtypes)

# Handle non-numeric values by encoding categorical columns
label_encoder = LabelEncoder()
for column in dataset.columns:
    if dataset[column].dtype == 'object':
        dataset[column] = label_encoder.fit_transform(dataset[column])

# Split the dataset into features (X) and target variable (y)
X = dataset.iloc[:, :-1]
y = dataset.iloc[:, -1]

# Split the data into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2,
    random_state=42)

# Base models for stacking ensemble
base_models = [
    ('dt', DecisionTreeClassifier(random_state=42)),
    ('rf', RandomForestClassifier(random_state=42))
]

# Stacking ensemble using Logistic Regression as meta-classifier
stacking_model = StackingClassifier(estimators=base_models,
    final_estimator=LogisticRegression())

# AdaBoost ensemble using Decision Tree as the base estimator
```

```python
adaboost_model = AdaBoostClassifier(base_estimator=DecisionTreeClassifier(
    random_state=42))

# Train the models
stacking_model.fit(X_train, y_train)
adaboost_model.fit(X_train, y_train)

# Make predictions
stacking_predictions = stacking_model.predict(X_test)
adaboost_predictions = adaboost_model.predict(X_test)

# Evaluate accuracy
stacking_accuracy = accuracy_score(y_test, stacking_predictions)
adaboost_accuracy = accuracy_score(y_test, adaboost_predictions)

print(f"Stacking Ensemble Accuracy: {stacking_accuracy:.4f}")
print(f"AdaBoost Ensemble Accuracy: {adaboost_accuracy:.4f}")
```

Listing 1: User Implementation of stacking ensemble and AdaBoost ensemble on the diabetes.csv dataset

5 RESULT

Stacking Ensemble Accuracy : The stacking ensemble achieved an accuracy of approximately 73.38%. This indicates that the combination of base models, where predictions are aggregated by a meta-model (in this case, Logistic Regression), resulted in a model that performed reasonably well on the test data. An accuracy of 73.38% suggests that the stacking ensemble correctly predicted the outcome for that percentage of instances in the test set.

```
Pregnancies                   int64
Glucose                       int64
BloodPressure                 int64
SkinThickness                 int64
Insulin                       int64
BMI                         float64
DiabetesPedigreeFunction    float64
Age                           int64
Outcome                       int64
dtype: object
Stacking Ensemble Accuracy: 0.7338
AdaBoost Ensemble Accuracy: 0.7143
```

Figure 3: Stacking Ensemble And

AdaBoost Ensemble Accuracy : The AdaBoost ensemble achieved an accuracy of around 71.43%. This indicates the performance of AdaBoost, which sequentially combines weak learners (typically decision trees) to form a strong model. An accuracy of 71.43% implies that AdaBoost accurately predicted the class labels for that percentage of instances in the test set.

6 CONCLUSION

In this lab report I implement stacking ensemble and AdaBoost ensembl with "diabetes.csv" dataset.The implementation of stacking and AdaBoost ensembles on the diabetes dataset resulted in reasonable accuracy levels of approximately 73.38% and 71.43%, respectively. The most difficult part was for me to combine different model but with help from my course teacher I overcome from that. Overall it was a greate experimant to done.