Bahria University,

Karachi Campus

LAB EXPERIMENT NO.

_10_

LIST OF TASKS

TASK NO	OBJECTIVE
1.	**Optimization of Ensemble Models using Decision Trees** Enhance the performance of existing Adaboost and Bagging algorithms which utilize Decision Trees as base learners. The objective is to optimize these models to achieve 100% efficiency, signifying perfect accuracy on the dataset. This will involve analyzing the current model configurations, identifying inefficiencies, and adjusting the Decision Tree parameters and training process to maximize predictive accuracy.
2.	**Implementation of Adaboost and Bagging Algorithms with Support Vector Machine** Implement Adaboost and Bagging ensemble techniques using Support Vector Machines (SVM) as the base learner. This task will explore how the robust features of SVM can be leveraged within ensemble frameworks to potentially improve model performance. The focus will be on configuring the SVM settings appropriately within each ensemble method, evaluating their effectiveness, and comparing the results to those obtained with Decision Trees.

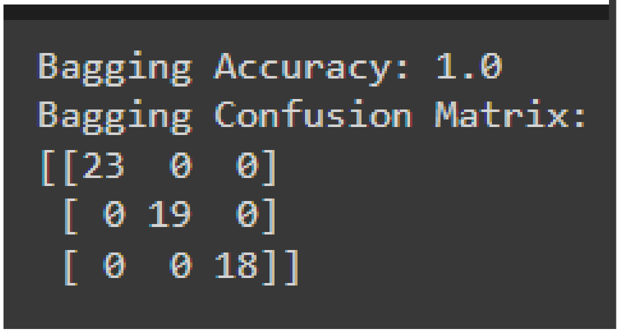
Submitted On:

6/7/2024

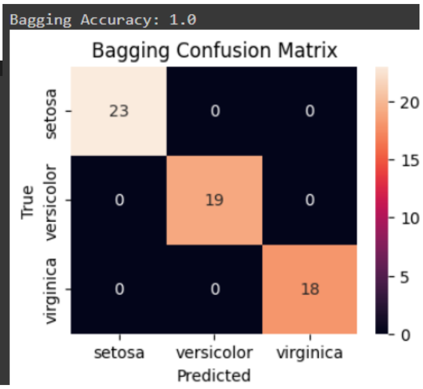
TASK NO 1: Optimization of Ensemble Models using Decision Trees.

```
import numpy as np
import pandas as pd
from sklearn.ensemble import BaggingClassifier
from sklearn.tree import DecisionTreeClassifier
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score, confusion_matrix
from sklearn.datasets import load_iris
iris = load_iris()
X = iris.data
y = iris.target
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.4,
random_state=42)
base_classifier = DecisionTreeClassifier(max_depth=1)
bagging_clf = BaggingClassifier(base_classifier, n_estimators=100,
random_state=42)
bagging_clf.fit(X_train, y_train)
y_pred = bagging_clf.predict(X_test)
accuracy = accuracy_score(y_test, y_pred)
conf_matrix = confusion_matrix(y_test, y_pred)
print(f'Bagging Accuracy: {accuracy}')
import matplotlib.pyplot as plt
import seaborn as sns
plt.figure(figsize=(4,3))
sns.heatmap(conf_matrix, annot=True, fmt='d',
xticklabels=iris.target_names, yticklabels=iris.target_names)
plt.xlabel('Predicted')
plt.ylabel('True')
plt.title('Bagging Confusion Matrix')
plt.show()
```

OUTPUT:



```
Bagging Accuracy: 1.0
Bagging Confusion Matrix:
[[23  0  0]
 [ 0 19  0]
 [ 0  0 18]]
```

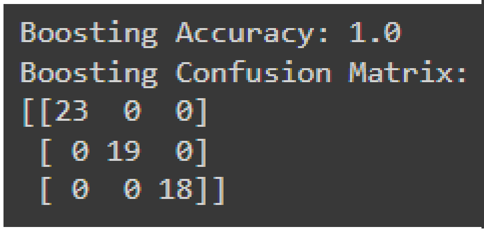


```
Bagging Accuracy: 1.0
```

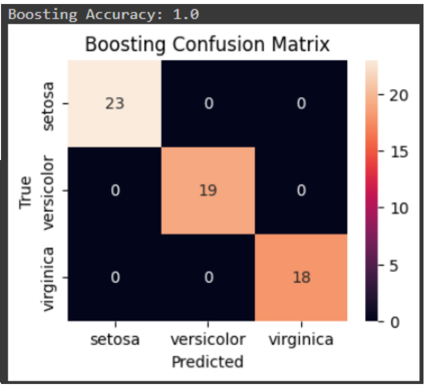
TASK NO 1: Implementation of Adaboost and Bagging Algorithms with Support Vector Machine.

```
import numpy as np
import pandas as pd
from sklearn.ensemble import AdaBoostClassifier
from sklearn.svm import SVC
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score, confusion_matrix
from sklearn.datasets import load_iris
iris = load_iris()
X = iris.data
y = iris.target
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.4,
random_state=42)
base_classifier = SVC(kernel="linear")
boosting_clf = AdaBoostClassifier(base_classifier,
n_estimators=100, random_state=42,algorithm='SAMME',
learning_rate=0.06)
boosting_clf.fit(X_train, y_train)
y_pred = boosting_clf.predict(X_test)
accuracy = accuracy_score(y_test, y_pred)
conf_matrix = confusion_matrix(y_test, y_pred)
print(f'Boosting Accuracy: {accuracy}')
print('Boosting Confusion Matrix:')
print(conf_matrix)
import matplotlib.pyplot as plt
import seaborn as sns
plt.figure(figsize=(4,3))
sns.heatmap(conf_matrix, annot=True, fmt='d',
xticklabels=iris.target_names, yticklabels=iris.target_names)
plt.xlabel('Predicted')
plt.ylabel('True')
plt.title('Boosting Confusion Matrix')
print(f'Boosting Accuracy: {accuracy}')
plt.show()
```

OUTPUT:



```
Boosting Accuracy: 1.0
Boosting Confusion Matrix:
[[23  0  0]
 [ 0 19  0]
 [ 0  0 18]]
```



```
Boosting Accuracy: 1.0
```