306281422_stats101c_hw3

November 10, 2024

[ ]:

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from sklearn import datasets
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score
from sklearn.tree import DecisionTreeClassifier
from sklearn.ensemble import BaggingClassifier, RandomForestClassifier,␣
 ↪GradientBoostingClassifier


data = pd.read_csv("banknote.csv", header=None)
dataset = np.array(data)

X = dataset[:, 0:4]
#X_1 = (X[:, 0] - X[:, 0].mean()) / X[:, 0].std()
#X_2 = (X[:, 1] - X[:, 1].mean()) / X[:, 1].std()
y = dataset[:, 4]
```

[ ]:

```
# Split the data into train and test sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3,␣
 ↪random_state=22)
```

[ ]:

```
# Step 1: Evaluate the base classifier (Decision Tree)
dtree = DecisionTreeClassifier(max_depth=5, random_state=22)
dtree.fit(X_train, y_train)

# Predict and evaluate the Decision Tree model
y_pred_dt = dtree.predict(X_test)
print("Decision Tree")
print("Train data accuracy:", accuracy_score(y_true=y_train, y_pred=dtree.
 ↪predict(X_train)))
print("Test data accuracy:", accuracy_score(y_true=y_test, y_pred=y_pred_dt))
print("Test Error Rate:", 1 - accuracy_score(y_true=y_test, y_pred=y_pred_dt))
```

```
Decision Tree
Train data accuracy: 0.9895833333333334
Test data accuracy: 0.9635922330097088
Test Error Rate: 0.036407766990291246
```

```
[ ]: random_forest = RandomForestClassifier(n_estimators=51, max_depth=5,␣
      ↪random_state=22)
     random_forest.fit(X_train, y_train)

     # Predict and evaluate the Random Forest model
     y_pred_rf = random_forest.predict(X_test)
     print("Random Forest")
     print("Train data accuracy:", accuracy_score(y_true=y_train,␣
      ↪y_pred=random_forest.predict(X_train)))
     print("Test data accuracy:", accuracy_score(y_true=y_test, y_pred=y_pred_rf))
     print("Test Error Rate:", 1 - accuracy_score(y_true=y_test, y_pred=y_pred_rf))
```

```
Random Forest
Train data accuracy: 0.990625
Test data accuracy: 0.9951456310679612
Test Error Rate: 0.004854368932038833
```

```
[ ]: boosting = GradientBoostingClassifier(n_estimators=51, max_depth=5,␣
      ↪random_state=22)
     boosting.fit(X_train, y_train)
     y_pred_boost = boosting.predict(X_test)
     print("Gradient Boosting")
     print("Train data accuracy:", accuracy_score(y_true=y_train, y_pred=boosting.
      ↪predict(X_train)))
     print("Test data accuracy:", accuracy_score(y_true=y_test, y_pred=y_pred_boost))
     print("Test Error Rate:", 1 - accuracy_score(y_true=y_test,␣
      ↪y_pred=y_pred_boost))
```

```
Gradient Boosting
Train data accuracy: 1.0
Test data accuracy: 0.9927184466019418
Test Error Rate: 0.007281553398058249
```

Both Random Forest and Boosting models have significantly low test error compared to Decision Tree model. This suggests that Random Forest and Boosting generalize better on the dataset.