bestmodel

November 26, 2023

[1]:
```
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression
from sklearn.tree import DecisionTreeClassifier
from sklearn.model_selection import RandomizedSearchCV
from sklearn.ensemble import RandomForestClassifier
from sklearn.naive_bayes import GaussianNB
from sklearn.metrics import classification_report, confusion_matrix,␣
↪accuracy_score

class Best_Model:
    def __init__(self, x_train, x_test, y_train, y_test):
        self.x_train = x_train
        self.x_test = x_test
        self.y_train = y_train
        self.y_test = y_test

    def Classification_report(self):
        models = {"Logistic Regression": LogisticRegression(),
                  "Decision Tree": DecisionTreeClassifier(),
                  "Random Forest": RandomForestClassifier(),
                  "Naive bayes": GaussianNB()}

        for model_name, model in models.items():
            print("For", model_name)
            model.fit(self.x_train, self.y_train)

            self.y_train_pred = model.predict(self.x_train)
            self.y_test_pred = model.predict(self.x_test)

            # train set classification report
            print("train set classification_report\n")
            print(classification_report(self.y_train, self.y_train_pred))
            # test set classification report
            print("test set classification_report\n")
            print(classification_report(self.y_test, self.y_test_pred))
          ␣
↪print("================================================================================")
```

```
  def Accuracy(self):
      models = {"Logistic Regression": LogisticRegression(),
                "Decision Tree": DecisionTreeClassifier(),
                "Random Forest": RandomForestClassifier(),
                "Naive bayes": GaussianNB()}

      for model_name, model in models.items():
          print("For", model_name)
          model.fit(self.x_train, self.y_train)

          self.y_train_pred = model.predict(self.x_train)
          self.y_test_pred = model.predict(self.x_test)

          # train set accuracy
          print("train set accuracy", accuracy_score(self.y_train, self.
↪y_train_pred) * 100)
          # test set accuracy
          print("test set accuracy", accuracy_score(self.y_test, self.
↪y_test_pred) * 100)
          ␣
↪print("================================================================================")

  def Confusion_matrix(self):
      models = {"Logistic Regression": LogisticRegression(),
                "Decision Tree": DecisionTreeClassifier(),
                "Random Forest": RandomForestClassifier(),
                "Naive bayes": GaussianNB()}

      for model_name, model in models.items():
          print("For", model_name)
          model.fit(self.x_train, self.y_train)

          self.y_train_pred = model.predict(self.x_train)
          self.y_test_pred = model.predict(self.x_test)

          # train set confusion matrix
          print("train set confusion_matrix\n")
          print(confusion_matrix(self.y_train, self.y_train_pred))
          # test set confusion matrix
          print("test set confusion_matrix")
          print(confusion_matrix(self.y_test, self.y_test_pred))
          ␣
↪print("================================================================================")
```