```
from sklearn import tree
import pydotplus
import collections
import numpy as np
import pandas
from sklearn.ensemble import RandomForestClassifier, BaggingClassifier,
AdaBoostClassifier, GradientBoostingClassifier, \
    VotingClassifier
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import accuracy_score
from sklearn import preprocessing as preproc
from sklearn import model_selection
from sklearn.metrics import accuracy_score
import xgboost as xgb
from sklearn.tree import DecisionTreeClassifier
from xgboost import XGBClassifier

# filename is for the dataset in use for the model. We have three datasets for
testing initially and we change the filename accordingly for each dataset we use
filename = "finalDataset3.xlsx"
df = pandas.read_excel(filename)

# For each of our datasets, the feature columns can be different. Based on the
dataset in use, we use the features accordingly.
#features1 = df[['Volume', 'unemp_rate', 'T5YIE', 'Close', 'Open', 'High', 'Low',
'GDP']]
#features2 = df[['Volume', 'unemp_rate', 'T5YIE', 'Close', 'Open', 'High', 'Low',
'GDP', 'prev_close']]
features3 = df[['Volume', 'unemp_rate', 'T5YIE', 'Close', 'Open', 'High', 'Low',
'GDP', 'prev_close', 'prev_open', 'prev_high', 'prev_low']]

# For the Ensemble Model, we will predict the Profit column for the entire dataset
targetColumn = df['Profit']

# Before test data split, we will perform scaling using MinMaxScaler for the entire
data of feature columns
scaler = preproc.MinMaxScaler()
scaler.fit(features3)
featureColumn = scaler.transform(features3)

# Using the seed value 42, we will split the dataset into test and train
accordingly. One third of the dataset is split into test set.
(X_train, X_test, Y_train, Y_test) =
model_selection.train_test_split(featureColumn, targetColumn, test_size=1/3,
random_state=42)

# When checking for VotingClassifier Model, we will use the following for-loop code
snippet to determine the accuracy score
# for estimator in range(10, 101, 10):
#     model1 = RandomForestClassifier(n_estimators=estimator)
#     model2 = XGBClassifier()
#     model3 = LogisticRegression()
#
#     model1.fit(X_train,Y_train)
#     model2.fit(X_train,Y_train)
#     model3.fit(X_train,Y_train)
#
#     model = VotingClassifier(estimators=[('dt', model1), ('knn', model2), ('lr',
model3)], voting='hard')
```

```
#     model.fit(X_train, Y_train)
#     print("Accuracy using %d as estimator value: " % estimator,
model.score(X_test, Y_test))
#     print()

# When checking for RandomForest, Bagging, AdaBoost, GradientBoosting or XGB
Classifier, we will use the following for-loop code snippet to determine the
accuracy score
for estimator in range(10, 101, 10):
    #model = RandomForestClassifier(n_estimators = estimator, criterion="entropy")
    model = BaggingClassifier(base_estimator=DecisionTreeClassifier(),
n_estimators=estimator)
    #model = AdaBoostClassifier(base_estimator=DecisionTreeClassifier(),
n_estimators=estimator)
    #model = GradientBoostingClassifier(n_estimators=estimator)
    #model = xgb.XGBClassifier(n_estimators=estimator)

    model.fit(X_train, Y_train)

    y_pred = model.predict(X_test)

    print("Accuracy using %d as estimator value: " %estimator,
accuracy_score(Y_test, y_pred))
```