

```
from sklearn.datasets import load_wine
from sklearn.cluster import DBSCAN
from sklearn.ensemble import BaggingClassifier, AdaBoostClassifier
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score
import time
from sklearn.tree import DecisionTreeClassifier

# Загрузка набора данных о вине
wine = load_wine()

# Получение признаков (X) и целевой переменной (y)
X = wine.data
y = wine.target

# Разделение данных на обучающую и тестовую выборки
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

# Задача 1: Реализация бэггинга

# Создание экземпляра базового классификатора
base_classifier = DecisionTreeClassifier()

# Создание экземпляра BaggingClassifier
bagging = BaggingClassifier(base_classifier, n_estimators=10)

# Измерение времени, затраченного на бэггинг
start_time = time.time()
bagging.fit(X_train, y_train)
bagging_predictions = bagging.predict(X_test)
bagging_time = time.time() - start_time

# Задача 2: Реализация бустинга на тех же данных, что и бэггинг

# Создание экземпляра AdaBoostClassifier
boosting = AdaBoostClassifier(base_classifier, n_estimators=10)

# Измерение времени, затраченного на бустинг
start_time = time.time()
boosting.fit(X_train, y_train)
boosting_predictions = boosting.predict(X_test)
boosting_time = time.time() - start_time

# Задача 3: Сравнение результатов бэггинга и бустинга

# Вычисление точности бэггинга и бустинга
bagging_accuracy = accuracy_score(y_test, bagging_predictions)
boosting_accuracy = accuracy_score(y_test, boosting_predictions)

print("Точность бэггинга:", bagging_accuracy)
print("Точность бустинга:", boosting_accuracy)
print("Время бэггинга:", bagging_time)
print("Время бустинга:", boosting_time)

import matplotlib.pyplot as plt

# Данные для графика
точность = [bagging_accuracy, boosting_accuracy]
время = [bagging_time, boosting_time]
метки = ['Бэггинг', 'Бустинг']

# Создание графика с полосами
fig, ax1 = plt.subplots()

# График точности на левой оси y
ax1.bar(метки, точность, color='blue')
ax1.set_ylabel('Точность')
ax1.set_ylim([0, 1])

# Создание второй оси y для времени
ax2 = ax1.twinx()
ax2.plot(метки, время, color='red', marker='o')
ax2.set_ylabel('Время (секунды)')

# Добавление подписей и заголовка
plt.xlabel('Метод ансамбля')
plt.title('Сравнение Бэггинга и Бустинга')

# Отображение графика
plt.show()
```



Точность бэггинга: 0.9722222222222222  
Точность бустинга: 0.9444444444444444  
Время бэггинга: 0.02495551109313965  
Время бустинга: 0.0034530162811279297

