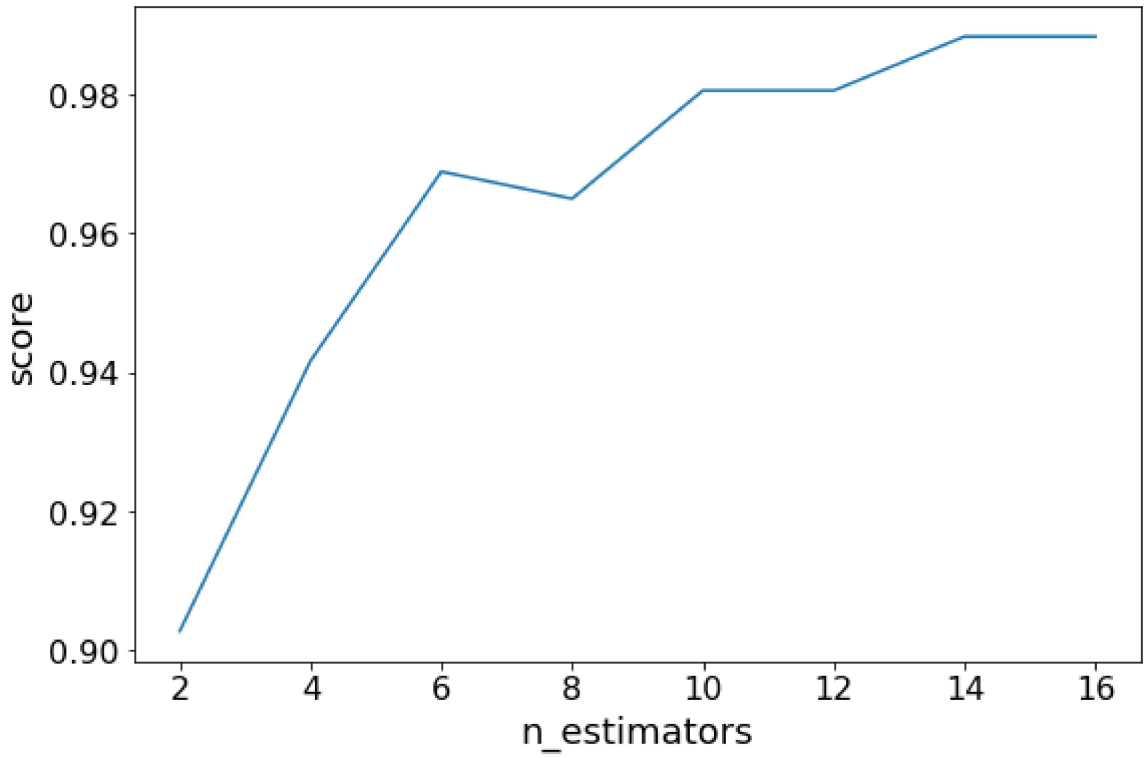
```
# Bagging

from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score
from sklearn.tree import DecisionTreeClassifier
import pandas as pd
from sklearn.ensemble import BaggingClassifier, AdaBoostClassifier
import matplotlib.pyplot as plt


df = pd.read_csv("heart.csv")

df.head()
```

	age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	ca	thal
0	52	1	0	125	212	0	1	168	0	1.0	2	2	3
1	53	1	0	140	203	1	0	155	1	3.1	0	0	3
2	70	1	0	145	174	0	1	125	1	2.6	0	0	3
3	61	1	0	148	203	0	1	161	0	0.0	2	1	3
4	62	0	0	138	294	1	1	106	0	1.9	1	3	2

```
y = df["target"]
x = df.drop(columns=["target"])

# for i in range(len(y)):
#   x.append([df["trestbps"][i],df["chol"][i]])

X_train, X_test, y_train, y_test = train_test_split(x, y, test_size = 0.25, random_state = 22

estimator_range = [2,4,6,8,10,12,14,16]

models = []
scores = []

for n_estimators in estimator_range:

    # Create bagging classifier
    clf = BaggingClassifier(n_estimators = n_estimators, random_state=22)

    # Fit the model
    clf.fit(X_train, y_train)

    # Append the model and score to their respective list
    models.append(clf)
    scores.append(accuracy_score(y_true = y_test, y_pred = clf.predict(X_test)))
```

```
# Generate the plot of scores against number of estimators
plt.figure(figsize=(9,6))
plt.plot(estimator_range, scores)

# Adjust labels and font (to make visable)
plt.xlabel("n_estimators", fontsize = 18)
plt.ylabel("score", fontsize = 18)
plt.tick_params(labelsize = 16)

print(scores)
# Visualize plot
plt.show()
```

```
[0.9027237354085603, 0.9416342412451362, 0.9688715953307393, 0.9649805447470817, 0.9805
```




```
estimator_range = [2,4,6,8,10,12,14,16]

models = []
scores = []

for n_estimators in estimator_range:

    # Create bagging classifier
    clf = AdaBoostClassifier(n_estimators = n_estimators)

    # Fit the model
    clf.fit(X_train, y_train)

    # Append the model and score to their respective list
```

```
    models.append(clf)
    scores.append(accuracy_score(y_true = y_test, y_pred = clf.predict(X_test)))

# Generate the plot of scores against number of estimators
plt.figure(figsize=(9,6))
plt.plot(estimator_range, scores)

# Adjust labels and font (to make visable)
plt.xlabel("n_estimators", fontsize = 18)
plt.ylabel("score", fontsize = 18)
plt.tick_params(labelsize = 16)


print(scores)

# Visualize plot
plt.show()
```

```
[0.7315175097276264, 0.7821011673151751, 0.8132295719844358, 0.8093385214007782, 0.8249
```