

```

import matplotlib.pyplot as plt
from sklearn.ensemble import RandomForestClassifier
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import LabelEncoder
from sklearn.metrics import accuracy_score
import pandas as pd

dataset = "breast-cancer.csv"

# setup dataframe
names = ['class', 'age', 'menopause', 'tumor-size', 'inv-nodes', 'node-caps', 'deg-malig', 'breast', 'breast-quad', 'irradiat']
df = pd.read_csv(dataset, names=names)

# label encode for processing
label_encoders = {}
for column in df.columns:
    le = LabelEncoder()
    df[column] = le.fit_transform(df[column])
    label_encoders[column] = le

# show dataframe
print(df.head())

# drop target for X
X = df.drop('class', axis=1)

# set y to target
y = df['class']

# n_estimators hyperparameter for random forest
n_estimators = []

# corresponding accuracies for the n_estimators
accuracies = []

# use a range of 1-200 n_estimators
for i in range(1, 200):

    # split data for training/testing
    X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=42)

    # train the model with the current iteration of n_estimators
    clf = RandomForestClassifier(n_estimators=i, random_state=43)
    clf.fit(X_train, y_train)

    # predict and score
    y_pred = clf.predict(X_test)
    accuracy = accuracy_score(y_test, y_pred)

    # add information to corresponding lists
    n_estimators.append(i)
    accuracies.append(accuracy)

# print the max accuracy along with the n_estimator that yielded it
print(max(accuracies))
print(n_estimators[accuracies.index(max(accuracies))])

# plot the comparison
plt.plot(n_estimators, accuracies)
plt.xlabel('N_estimators')
plt.ylabel('Accuracy')
plt.title('Random Forest n_estimators compared to accuracy')
plt.show()

```