import numpy as np

import pandas as pd

import sklearn

from sklearn.model\_selection import train\_test\_split

from sklearn.ensemble import RandomForestClassifier

from sklearn.metrics import accuracy\_score

from sklearn.datasets import load\_iris

import sklearn.metrics as metrics

**01 Use the iris dataset**

from sklearn import datasets

iris = datasets.load\_iris()

import pandas as pd

data=pd.DataFrame({

    'sepal length':iris.data[:,0],

    'sepal width':iris.data[:,1],

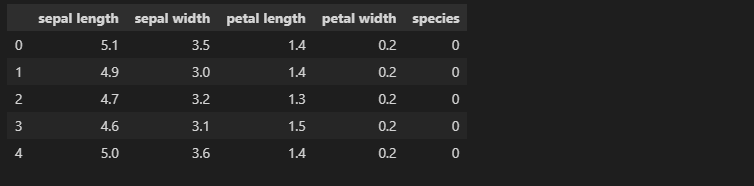
    'petal length':iris.data[:,2],

    'petal width':iris.data[:,3],

    'species':iris.target

})

data.head()



from sklearn.model\_selection import train\_test\_split

X= data[['sepal length', 'sepal width', 'petal length', 'petal width']]  # Features

y= data['species']  # Labels

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.3)

**02 Using SVM and Random Forest**

from sklearn.ensemble import RandomForestClassifier

from sklearn.svm import SVC

svm\_clf=SVC()

rf\_clf=RandomForestClassifier(n\_estimators=100)

svm\_clf.fit(X\_train,y\_train)

rf\_clf.fit(X\_train,y\_train)

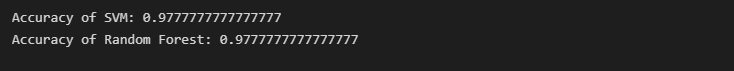
y\_pred\_svm=svm\_clf.predict(X\_test)

y\_pred\_rf=rf\_clf.predict(X\_test)

from sklearn import metrics

print("Accuracy of SVM:",metrics.accuracy\_score(y\_test, y\_pred\_svm))

print("Accuracy of Random Forest:",metrics.accuracy\_score(y\_test, y\_pred\_rf))



**03 Using grid search with cross-validation to select best parameters**

from sklearn.model\_selection import GridSearchCV

from sklearn.model\_selection import cross\_validate

from sklearn.metrics import recall\_score

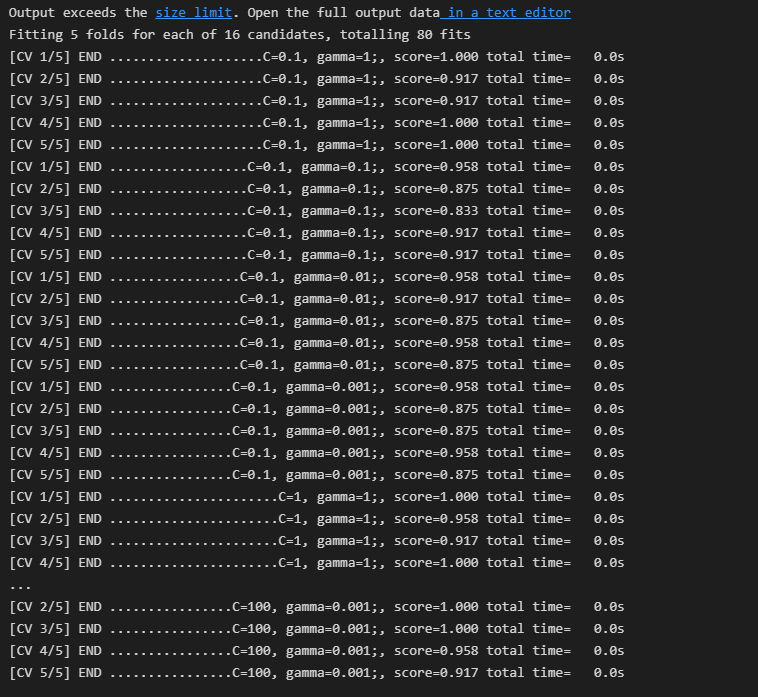
# Create a dictionary called param\_grid and fill out some parameters for C and Gamma

param\_grid = {'C':[0.1,1,10,100], 'gamma':[1,0.1,0.01,0.001]}

grid = GridSearchCV(SVC(), param\_grid, refit = True, verbose=3)

scoring = ['precision\_macro', 'recall\_macro']

scores = cross\_validate(grid, X, y, scoring=scoring)



sorted(scores.keys())



scores['test\_recall\_macro']



pred\_grid = grid.predict(X\_test)

print("Accuracy of Grid Search:",metrics.accuracy\_score(y\_test, pred\_grid))



**04 Using two matrix (Confusion\_matrix and F1-score)**

from sklearn.metrics import confusion\_matrix

from sklearn.metrics import f1\_score

# for SVM

print(confusion\_matrix(y\_test, y\_pred\_svm))

print(f1\_score(y\_test, y\_pred\_svm, average=None))



# for Random Forest

print(confusion\_matrix(y\_test, y\_pred\_rf))

print(f1\_score(y\_test, y\_pred\_rf, average=None))



# for Grid Search

print(confusion\_matrix(y\_test, pred\_grid))

print(f1\_score(y\_test, pred\_grid, average=None))

