Python Programming for Engineers
Assignment # 5
Due date: Sunday, Jan. 16th, 2022
IMPORTANT!

1. Submit your HWs **ONLINE** before the due date
2. HW reports should contain:
    a. The description of the problem and proposed solution
    b. The program code
    c. Any program outputs
3. Submitted codes should be well-commented.

Classification in Python

This assignment will require you to implement and interpret some of the classification concepts in Python.

Using Classification Methods on the Breast Cancer Dataset

In this assignment you will be using the Breast Cancer Wisconsin dataset. For more information about this dataset, see the following website:

https://archive.ics.uci.edu/ml/datasets/Breast+Cancer+Wisconsin+(Diagnostic)

The dataset consists of 569 patients with either 'Malignant' or 'Benign' breast lesions. 30 features are computed from a digitized image of a fine needle aspirate (FNA) of the breast mass.

Your objective here will be to perform classification on the dataset to predict the diagnosis of each sample from its features (i.e. binary classification). You will evaluate the classification performance of two well-known classifiers: bayes classifier and support vector machines (SVM).

First, load the cancer dataset. This can be done using the following snippet of code:

```
from sklearn.datasets import load_breast_cancer
data = load_breast_cancer()
X = data.data # Input features
y = data.target # Class label (0: Malignant, 1: Benign)
```

Then, separate your data into 70% training and 30% test set using train_test_split:

```
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split( ??? )
```

Scikit-learn library has built-in methods for Naive Bayes and SVM classification:

```
from sklearn.naive_bayes import GaussianNB
from sklearn.svm import SVC
svm_cl   = SVC(C=1, gamma='scale', probability=True)
bayes_cl = GaussianNB(var_smoothing=1e-7)
```

You need to modify the following code piece to train and test your classifiers:

```
CL.fit(X_train, y_train) # CL is the classifier model to train
y_pred = CL.predict(X_test) # Cancer prediction on the test set
# You need the probability of being cancer for ROC plot and AUC computation
y_proba = CL.predict_proba(X_test)[:,1]
```

Train your classifier with different parameter settings:

- SVM: Try at least 3 different values for C
- Bayes: Try at least 3 different values for *var_smoothing*

Given the trained classifiers, the performance of the model could be tested using the following metrics, accuracy_score, roc_curve, auc:

```
from sklearn.metrics import accuracy_score, roc_curve, auc
# The prediction accuracy
print ( "Prediction accuracy : %.2f"  % accuracy_score( ??? ))
# Receiver Operating Cracateristic Curve (ROC)
fpr, tpr, thr = roc_curve( ??? )
plt.plot(fpr, tpr)
# Area Under ROC curve (AUC)
print ( "AUC score : %.2f"  % auc( ??? ))
```

You can use these scores to measure the efficacy of a particular classifier. Your output should contain the following:

- Classification accuracy of the two classifiers (Bayes and SVM) with different parameter settings.
- ROC curves and AUC scores of the two classifiers (Bayes and SVM) with different parameter settings.

Given this output, respond to the following questions:

- What is the highest classification accuracy you achieve?
- How does the classifier parameters affect the classification performance?
- Which classifier performs better? Any thoughts why?
- What does the AUC score represent? Why is it an important metric?