SVM Project

In this project you are asked to run experiments on the Wisconsin Breast Cancer dataset. There are 569 examples, each labeled as 0 or 1. Classical approaches achieve accuracy of over 98%. You are asked to train SVM classifiers for this problem using **scikit-learn**. The challenge is to select the free parameters to maximize the accuracy. You are asked to produce a total of 4 classifiers:

1. A classifier trained with 8% of the data using a polynomial kernel. Name it **SVM-p8.py**.
2. A classifier trained with 8% of the data using an exponential (rbf) kernel. Name it **SVM-e8.py**.
3. A classifier trained with 16% of the data using a polynomial kernel. Name it **SVM-p16.py**.
2. A classifier trained with 16% of the data using an exponential (rbf) kernel. Name it **SVM-e16.py**.

- Your programs must set the random seed of python to 1 to make sure that your results are reproducible.
- The "p" programs must use a polynomial kernel, and the "e" programs must use an exponential kernel,
- Your programs will be tested by training them on randomly selected fractions of the dataset. The testing data will be the entire dataset.
- The training and testing of each program should not take more than 3 minutes.

Installing scikit-learn

```
pip install -U scikit-learn
```

Provided programs and data

1. The dataset is given in the files **x_test.csv** and **y_test.csv**
2. A random subset of 46 training examples in **x_train8.csv** and **y_train8.csv**
3. A random subset of 91 training examples in **x_train16.csv** and **y_train16.csv**
4. An example program **SVM-16.py**.
5. A program that can extract a random fraction from the training data is available as **fraction_xy.py**.

What you need to do

Determine the parameters for the SVM to maximize the accuracy.

Grading

We will generate random subsets of training examples by running the program **fraction_xy.py** with a seed that is kept secret. If, for example, the seed is 7, generating a fraction of 8% can be done as follows:

```
python3 fraction_xy.py  x_test.csv y_test.csv 0.08 7

This creates the files x_test_7_8.csv and y_test_7_8.csv
that should be renamed to x_train.csv and y_train.csv
```

Your grade will be based on the accuracy of your models trained with the generated examples and tested on the entire testing data.

What you need to submit

Your submission should be a single zip archive named **netid.zip**, where **netid** is your net id. The zip archive should contain the following:

1. Source code of the python scripts. They should be named as follows:

```
SVM-p8.py, SVM-e8.py, SVM-p16.py, SVM-e16.py,
```

2. Documentation describing the results of experiments/accuracy that your programs achieve on the provided data.

SCIKIT-LEARN

Scikit-learn is a popular free software machine learning library for the Python programming language. Their description of SVM can be found in the following link:
https:// scikit −learn.org/stable/modules/svm.html
The method that corresponds to what was covered in class is **SVC** (Support Vector Classification). The description of its parameters can be found in:
https:// scikit −learn.org/stable/modules/generated/sklearn.svm.SVC.html

Running SVC with a polynomial kernel (and soft margins) requires the following parameters to be set:

```
C = positive float value
Kernel = 'poly'
degree = nonnegative integer value
gamma = positive float value. You cannot use 'scale' or 'auto'. (1.0 in class.)
coef0 = float value. (1.0 in class.)
```

Running SVC with a exponential kernel (and soft margins) requires the following parameters to be set:

```
C = positive float value
Kernel = 'rbf'
gamma = positive float value. You cannot use 'scale' or 'auto'.
```