Support Vector Machine

This Homework offers an opportunity to try out SVM using Python's sklearn library.

First you need to install sklearn using pip install.

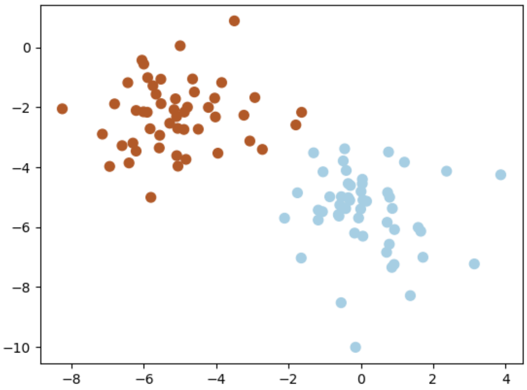
```
import numpy as np
import matplotlib.plot as plt
from sklearn import datasets
from sklearn.svm import SVC
```

Hand in all parts of this assignment in PLATO (both the code and report PDF file as specified). To get full marks, your functions (i.e., *.py files) must not only work correctly, but also must be clearly documented with sufficient comments for others to easily use and understand the code. You will lose marks for insufficient or unclear comments. In this assignment, you also need to hand in scripts showing tests of your functions on all the cases specified as well as the images and other answers requested. The scripts and results (as screenshots or otherwise) should be pasted into a single PDF file and clearly labeled. Note that lack of submission of either the code or the PDF will also result in loss of points.

Part 1. Linear SVM Classifiers

1. Create Datasets

The data to be used in this assignment is made and used by oneself. Use the *‘make_blobs’* function built into sklearn to generate data and show it as a scatterplot. Construct 100 samples of data into two categories: '*cluster_std*' =1.2, '*random_state*' =20, 30, and 40.

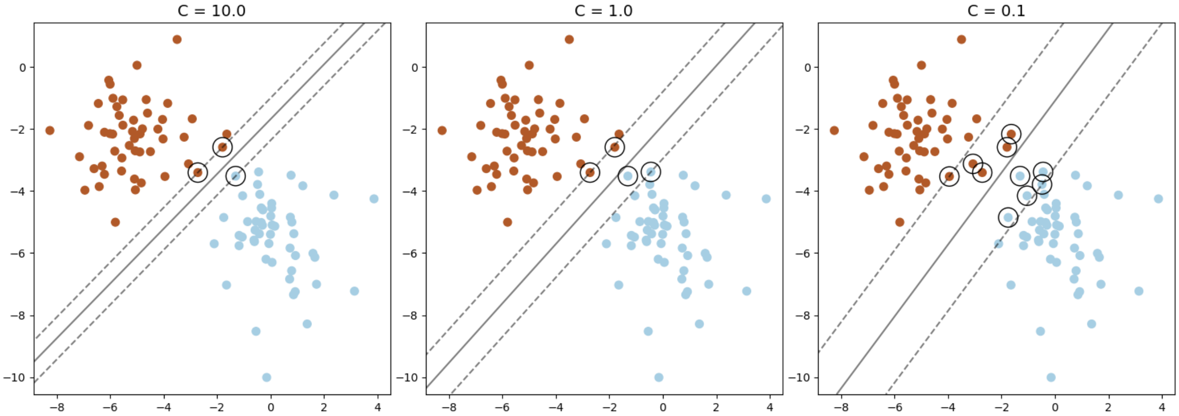


Example 1. random_state = 50

Note : In all three cases, the report must have a title attached. And the two groups should be marked in different colors.

2. Train SVM

Use the SVC (Support Vector Classifier) built into sklearn to find the decision boundary and margins. For all three data, kernel uses linear, and Misclassifications cost uses 10, 1, 0.1, and **write in the report comparing the results**.



Example 2. random_state = 50

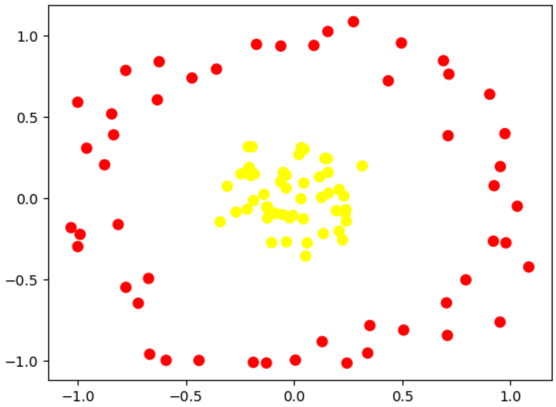
Note : Do any marking on the support vector.

Reference : https://scikit-learn.org/stable/modules/generated/sklearn.svm.SVC.html

Part 2. Nonlinear SVM

1. Create Datasets

Use the *'make_circles'* function built into sklearn to generate data and show it as a scatterplot. Construct 100 samples of data into two categories: '*factor*'=0.1, '*noise*'=0.1.

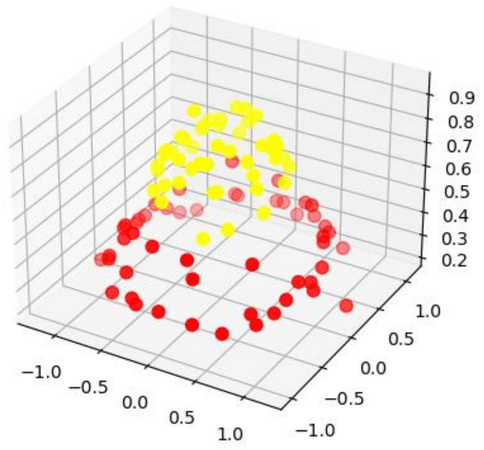


Example 3. factor = 0.2

2. Kernel function

Use the **Gaussian RBF kernel** function to project and display previously generated data into 3D space. Then plot the scatterplot with a three-dimensional graph.

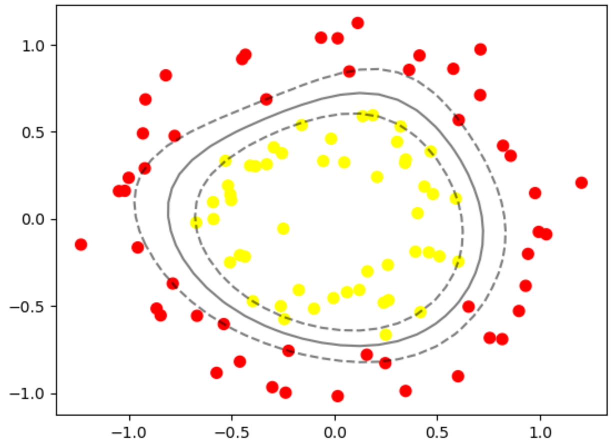
$$K_{RBF}(x, x_i) = e^{-\sum_i^n |x - x_i|^2}$$



Example 4. factor = 0.2

3. Train SVM

Use the SVC (Support Vector Classifier) built into sklearn to find the decision boundary and margins. Kernel uses '*rbf*', and Misclassifications cost uses 10, 0.1, and **write in the report comparing the results**.



Example 5. factor = 0.2

Deliverables

You will hand in your assignment in PLATO. You should hand in one zip file including two files, a file containing your code (i.e., *.py file). This file must have sufficient comments for others to easily use and understand the code. In addition, hand in a PDF document showing scripts (i.e., records of your interactions with the Python shell) showing the specified tests of your functions as well as the images and other answers requested. The PDF file has to be organized and easily readable / accessible. Assignments are to be handed in before 11:59pm on their due date