Assignment 2: kNN, Linear Regression, Perceptrons, Kernels
Due Date: 11.59 PM 10/13/2023 Maximum Points: 150

Please follow the instructions. Failure to adhere to instructions will result in a penalty.

1 Question 1 (25 pts)

In this question, we are going to design non-linear feature maps for the dataset in Table 1. Note that your input x has two features (or dimensions) $x1$ and $x2$. The feature map $\phi(x)$, where $x = (x1, x2)$ creates one or more additional features and transforms this into a higher-dimensional space. An example of this transform $\phi(x)$ can be $\phi(x) = (x1, x2, x1^2)$.

- 1 a. How would the dataset look like if you applied the feature map $\phi(x) = (x1, x2, x1^2)$? (2 pts)
- 1 b. The goal of feature maps is to ensure that the dataset is linearly separable. How can you tell if a feature map works? [Hint: It has something to do with being able to draw a line and when you can draw a line](2 pts)
- 1 c. Design a feature map $\phi(x)$ that linearly separates the dataset that you see in Table 1(6 pts)
- 1 d. Notice that you have 4 samples in the dataset. Let us say you create a Kernel function $K(x, z) = (1 + x1z1)(1 + x2z2)$. How would the values of the kernel value look like for all pairs of points in the dataset? (4 pts)
- 1 e. Recall that $K(x, z) = \langle \phi(x), \phi(z) \rangle$ (dot-product), where $\phi(x)$ is the feature-map. For the kernel function defined above, what is the feature-map $\phi(x)$ (6 pts)
- 1 f. You are designing a malware classifier. It is a linear classifier where you plan to use kernels. The dataset of malware and software has several properties P_i. An example of this property is: P_{10}=*Does this software spawn a known malicious process?*. You design a kernel function $K(x, z)$ that corresponds to the number of properties that are true for both samples x and z. For example, if both samples have 5 same properties that are true, $K(x, z) = 5$. Describe the feature map $\phi(x)$ that this kernel uses. (5 pts)
- Instruction: Submit q1.pdf or q1.png/jpg file. Do not submit a doc file.

Table 1: Table with features X1, X2 and Output Label

X1	X2	Output
1	1	Class 1
1	0	Class 2
0	1	Class 2
0	0	Class 1

2 Question 2: k-Nearest Neighbors (30 pts)

Inside the lab2_dataset/q2/ folder, you will find the following file:

- The Iris flower dataset: Iris.csv

The above iris flower dataset includes four features: sepal length, sepal width, petal length, and petal width, and the target variable is the species of the iris flower (setosa, versicolor, or virginica). Given the dataset, your task is to build a k-Nearest Neighbors (k-NN) model to predict the species of iris flowers.

Instructions:

- Complete the implementation of k-NN algorithm (k=3) to build a predictive model in the given **q2_starter.py** file.
- If needed you can change the path of the input file, but change it back to what is given in the **q2_starter.py** before submission.
- Complete and submit the **q2_starter.py** file.

3 Question 3: Linear Regression (30 pts)

Inside the lab2_dataset/q3/ folder, you will find the following files:

- 1. The input training data is : X_train_q3.csv
- 2. The input training label is : y_train_q3.csv
- 3. The test data is: X_test_q3.csv

Given the above dataset of graduate admissions, which includes various features like GRE score, TOEFL score, university rating, statement of purpose strength, letter of recommendation strength, undergraduate GPA, research experience, and the corresponding labels indicating 'Chance of Admit':

Instructions:

- Complete the **q3_starter.py** file.
- linear_regression_equation(): Use the the equation $y = w_0 + w_1x_1 + w_2x_2 + ... + w_nx_n$ to calculate the 'Chance of Admit' . Write the code to calculate the Mean Square Error value for the training set for $w_0 = w_1 = w_2 = .. = 1$.
- linear_regression_library(): Train a linear regression model (using a library) on the training data. Predict the 'Chance of Admit' for the provided test dataset. Note that the test labels are not provided, and you have to make predictions based on the linear regression model trained.
- Save and submit your predictions for X_test_q3.csv in a file named **y_predict_q3.csv** and submit your code file **q3_starter.py**.

4 Question 4: Perceptron 65 pts

4.1 Part A: 25 pts

You have a dataset D described below

i	X_i	Y_i
1	(-1,1)	1
2	(-1,-1)	-1
3	(0.5,0.5)	1
4	(1,-1)	1
5	(0.5,-1)	-1

Run the base perception algorithm **by hand** and fill out the values for w_t, $\langle w_t, X_i \rangle$, $Y_i.X_i$, w_{t+1} for time $t = 1, 2, 3, 4, 5, 6$ in a table looking like

time (t)	X_i	Y_i	w_t	$\langle w_t, X_i \rangle$	$Y_i.X_i$	w_{t+1}
1	(-1,1)	1	(0,0)	...	...	..
...						

4.2 Part B: 40 pts

Consider the iris dataset. Start with the classes 'Iris-setosa' and 'Iris-versicolor'. You are going to implement the two variants of perceptions described in Algorithms 1,2. You can also find

Algorithm 1 BaselinePerceptronTrain

```
procedure BaselinePerceptronTrain($D, MaxIter$)
    $w \leftarrow 0, b \leftarrow o$                                   ▷ initialize weights and bias
    for iter =1...MaxIter and $Mistakes > 0$ do
        for all $(x, y) \in D$ do
            $a \leftarrow w.x + b$                               ▷ compute activation for this example
            if $ya \leq o$ then
                $Mistakes \leftarrow Mistakes + 1$                        ▷ made a mistake
                $w \leftarrow w + y.x$                                    ▷ update weights
                $b \leftarrow b + y$                                      ▷ update bias
            end if
        end for
    end for
    return $w, b, iter$
end procedure
```

Algorithm 2 KernelizedPerceptronTrain

```
procedure BaselinePerceptronTrain($D, MaxIter$)
    $\alpha \leftarrow 0, b \leftarrow 0$                              ▷ initialize weights and bias
    for iter =1...MaxIter and $Mistakes > 0$ do
        for all $(x_n, y_n) \in D$ do
            $a \leftarrow \sum_m \alpha_m K(x_m, x_n) + b$       ▷ compute activation for this example
            if $y_n a \leq 0$ then
                $Mistakes \leftarrow Mistakes + 1$                        ▷ made a mistake
                $\alpha_n \leftarrow \alpha_n + y_n$                      ▷ update weights
                $b \leftarrow b + y_n$                                    ▷ update bias
            end if
        end for
    end for
    return $w, b, iter$
end procedure
```

the starter code as **q4_perceptron_starter.py** or **q4_perceptron_starter.ipynb**. I recommend using **google colab** for this problem.

You need to follow the steps for this problem (20 pts):

- Extract the dataset, and replace the labels of classes 'Iris-setosa' and 'Iris-versicolor' with '+1' and '-1'.
- Set MaxIter = 100,000
- Count the number of mistakes you make for every iteration across the training set
- Exit the loop if the number of mistakes for an iteration = 0, or you cross max iterations

You need to print the number of iterations it took for you to exit the loop.

Repeat this experiment with classes 'Iris-versicolor' (replace with -1) and 'Iris-virginica' (replace with +1).

Your experiment can take about 2 to 10 minutes to execute in the worst case.

In the next part, you are going to implement Kernelized Perceptron. The Kernel that you are going to use is *sklearn.metrics.pairwise.polynomial_kernel(X,X,degree=25)* You again need to follow these steps: (20 pts)

- Extract the dataset, and replace the labels of classes 'Iris-setosa' and 'Iris-versicolor' with '+1' and '-1'.
- Set MaxIter = 100,000
- Count the number of mistakes you make for every iteration across the training set
- Exit the loop if the number of mistakes for an iteration = 0, or you cross max iterations

Instructions:

- Complete and submit the **q4_perceptron_starter.py** or **q4_perceptron_starter.ipynb** file.

Extra Instructions:

- Write your name, email, student id as comment inside each code and pdf/image files.
- For question 1 submit q1.pdf or q1.png/jpg file. Do not submit a doc file.
- Do not submit files with different names. Use the mentioned name for the code and output file.
- Do not submit any given dataset files.
- Do not submit multiple files of the same code or the same output CSV.
- Please check your submitted CSV files. prediction CSV files should have only one column with a header.
- If you use google colab, upload the dataset folder with the same name. Do not change the names of folders or files. Example code:

  ```
  from google.colab import drive
  drive.mount(’/content/drive’)
  import pandas as pd
  data = pd.read_csv(’/content/drive/My Drive/lab2_dataset/q2/Iris.csv’)
  # Before submission change back the path to ’lab2_dataset/q2/Iris.csv’
  # data = pd.read_csv(’lab2_dataset/q2/Iris.csv’)
  ```