**Assignment 4**

Show **all** your work for each question

This assignment is to be completed individually.

Total: 37 points

Due: Dec. 6

|  |  |
| --- | --- |
| 1. Given the following weights W and b of a logistic regression model, calculate the prediction of the model on the following test instances and . | 2 points |
| 2 . Given that you have a logistic regression model for binary classification (true labels are 0 or +1) with the initial weights and and a bias weight b=0.5, perform one update of the weights after the model is the given the training instance [2,3] which has the true label (+1). The model uses the sigmoid function, Cross Entropy Loss, and a learning rate of 0.25. Show all calculations to determine the new weights after training on this one training instance, which includes the calculations for your gradient descent and the output of your model (probability of being the positive class) for this given instance before you perform the updates. | 4 points |
| 3. Use the following training set to train a perceptron. Show all steps by filling in the table.  Assume a learning rate of **0.25** and initial weights of [**1, 1, 0]** for [], with a bias of **-1**.  Use the sign activation function that we discussed in class   |  |  |  |  | | --- | --- | --- | --- | |  |  |  | Y | | 1 | 0 | 0 | +1 | | 0 | 0 | 1 | -1 | | 1 | 0 | 1 | +1 | | 1 | 1 | 1 | -1 | | 4 points |
| 4. AdaBoost  Given the following dataset, we selected 10 samples for our first boosting round. Original dataset:   |  |  |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | | X | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | | Y | -1 | +1 | +1 | +1 | -1 | -1 | -1 | -1 | +1 | -1 |   Selected for first Boosting round   |  |  |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | | X | 2 | 2 | 3 | 4 | 7 | 7 | 7 | 8 | 9 | 10 | | Y | +1 | +1 | +1 | +1 | -1 | -1 | -1 | -1 | +1 | -1 |   The decision stump will use 5.5 as the split point (if x predict +1, otherwise predict -1).  Perform one iteration of AdaBoost starting with the selected samples for the first boosting round. Calculate the new weights for each instance within the dataset. | 4 points |

## Code (2 parts, but only one python script) 11 points

Complete and submit the python script **a4.py** that reads in the file **full\_netflix\_data.csv**. Your code will create a Logistic regression classifier. Your model does not train on the **full\_netflix\_data.csv**, but rather on a simpler dataset that is given in the starter code. The **full\_netflix\_data.csv**will be used to perform visualization techniques and is originally from https://www.kaggle.com/datasets/octopusteam/full-netflix-dataset. Do not look up the dataset prior to completing this assignment.   
The methods and functions descriptions are given in the starter code and below:

|  |  |
| --- | --- |
| **Read\_data**(file\_path) | Reads in the file given from command line and returns the dataset as a list of lists where each internal is an instance with its class |

**Class LogisticReg**

|  |  |
| --- | --- |
| **dot\_product(self, X, Y)** | Calculates the dot product between vectors X and Y. Returns the dot product. |
| **sigmoid(self, Z)** | Calculates the sigmoid of the input Z. Returns the output of the sigmoid function. |
| **gradient(self, X, Y)** | Calculates the gradient when the model is given the input vector X and true label Y. Returns the gradient. |
| **update\_weights(self, gradient)** | Updates the weights (self.weights) of the model given the calculated gradient. Does not return a value. |
| **cross\_entropy\_loss(self, X,  Y)** | Calculate the cross\_entropy\_loss when the model is given the input vector X with true label Y. Returns the loss value. |
| **train\_and\_plot\_loss(self, dataset, num\_epochs)** | This is given the dataset and number of epochs to train over the dataset as input. The code should iterate over the dataset *n\_epochs* times and update the weights of the model after each single instance.  This code should also create two different scatter plots using the plotly library. The first graph shows the cross entropy loss after each weight update which occurs after each time the model trains on a single instance. The second graph sows the mean cross entropy loss after each epoch (one full iteration through the dataset).  Does not return a value.  Have a look at your plots and see if the values and trends make sense. |

|  |  |
| --- | --- |
| **visualize\_data(dataset)** | Given a dataset, create three visual representations of some data from the dataset using three different visualization styles. You can choose what data is being visualized and how you visualize it, but it needs to be three different techniques. |

# IMPORTANT for your code!

* Keep in mind that your code should generate 5 figures/plots/visualizations.
* To run the code, you will need to run from command line:
  + python a4.py full\_netflix\_data.csv
* I will run the code in the same manner and therefore, it must work this way.
* The code submission must be a .py file and you don’t need to submit the data csv file.
* You may **NOT** import any additional libraries or packages that are not already imported with the starter code.
* The only usage from the math library that you can use are the math.e and math.log.
* If you import additional libraries, your code will automatically be marked out of 50% and all code that uses the unapproved library will be marked as incorrect.
* Any attempt to modify the declaration of functions, such as the parameters that it uses, will be marked as incorrect.
* Any modification to the test code (all code below the line ###TESTS) will result in a **0** on the entire coding assignment.
* Code that can’t be run due to a syntax error will be marked out of 50%. If you are unable to get a function working, have it return the default values that are in the starter code. This should make your code not give a syntax error and not be marked out of 50% because of it.
* Your submitted code must **NOT** contain any additional print statements than what was given in the starter code.
* Have fun with the assignment and the data!