CS-331 Introduction to Artificial Intelligence - *Spring 2023*

Programming Assignment 2

Made by: Muhammad Musa (23100004@lums.edu.pk)

Instructions:

- This project is on an group basis.
- Due date: **16 March 2023**. No submissions will be accepted after this date.
- You are required to submit **both** a .ipynb, and a .py file; both of them should be named: GroupNumber_PA2.
- Total Marks are 100. A breakdown of these marks is given with each part.
- **NO PLAGIARISM WILL BE TOLERATED** In case of finding any plagiarism, the case will be referred to the DC committee, and the student will be given a 0 in **all the projects**. Plagiarism is presenting someone else's work or ideas as your own, with or without their consent, by incorporating it into your work without full acknowledgment. All published and unpublished material, whether in manuscript, printed, or electronic form, is covered under this definition, along with ChatGPT. Plagiarism may be intentional or reckless, or unintentional. Under the regulations for examinations, intentional or reckless plagiarism is a disciplinary offense. If you are unsure about what counts as plagiarism and what doesn't, **PLEASE reach out** to the TAs so that they can guide you.

Part One [Regression]:

In this part we are going to work on different kinds of regressions and work on their implementation.

Dataset [5 marks]

The dataset we are going to use is *bike_hour.csv*. This dataset contains the hourly count of rental bikes between the years 2011 and 2012 with the corresponding weather and seasonal information and contains the following features:

- instant: record index
- dteday: date
- season: season (1:winter, 2:spring, 3:summer, 4:fall)
- mnth: month (1 to 12)
- hr: hour (0 to 23)
- holiday: weather day is a holiday or not
- weekday: day of the week
- workingday: if day is neither weekend nor holiday is 1, otherwise is 0.
- weathersit:
  - 1: Clear, Few clouds, Partly cloudy, Partly cloudy
  - 2: Mist + Cloudy, Mist + Broken clouds, Mist + Few clouds, Mist
  - 3: Light Snow, Light Rain + Thunderstorm + Scattered clouds, Light Rain + Scattered clouds
  - 4: Heavy Rain + Ice Pallets + Thunderstorm + Mist, Snow + Fog
- temp: Normalized temperature in Celsius. The values are derived via (t-t_min)/(t_max-t_min), t_min=-8, t_max=+39
- atemp: Normalized feeling temperature in Celsius. The values are derived via (t-t_min)/(t_max-t_min), t_min=-16, t_max=+50
- hum: Normalized humidity. The values are divided to 100 (max)
- windspeed: Normalized wind speed. The values are divided to 67 (max)
- casual: count of casual users
- registered: count of registered users
- cnt: count of total rental bikes including both casual and registered

You need to load this dataset and clean it to remove any N/A or null values.

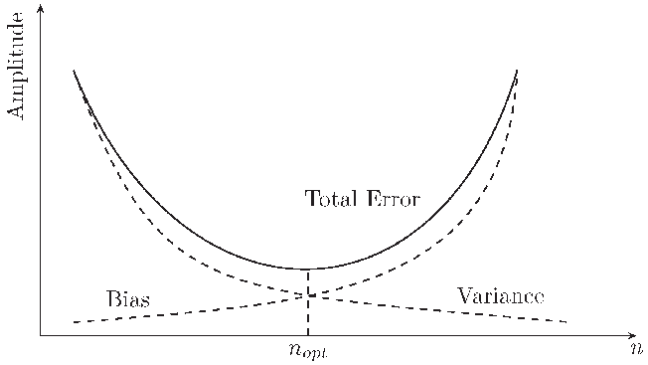

Figure 1: Bias and Variance Loss

$$\text{MSE} = \frac{1}{n}\sum_{i=1}^{n}(Y_i - \hat{Y}_i)^2$$

Figure 2: Mean Squared Error Loss Function

Polynomial Regression using Least Squares and Pseudo Inverse [20 marks]

In this part, we are going to use pseudo-inverse, bias error, and variance error, to find the best-fit equation for our dataset. We are going to make use of the equation:

$$w = (A^T A)^{-1} A^T y$$

We already have our dataset. For this question, we are going to focus on univariate polynomial regression, so we will select only one feature to use, which is going to be "atemp", the normalized feeling temperature. Your task is to perform polynomial regression on this variable for polynomials from 1 till 20 and make a variance loss (test loss) vs bias loss (training loss) graph similar to the one in **Figure 1** and answer the question asked in the notebook.

Linear Regression by Gradient Descent [15 marks]

For this part, you have to implement multivariate linear regression for the *bike_hours.csv* dataset. Since this is multivariate, we no longer need to restrict ourselves to one feature. However, not all features are helpful in regression, so you have to filter them a little and explain your choice in the notebook, where asked. After that, you have to split the dataset to test and train, and perform gradient descent using Mean Squared Error Loss (MSE) (**Figure 2**). Finally, you have to check your accuracy on the dataset and report it.

Part Two [Logistic Regression]:

Dataset [5 marks]

The dataset we are going to use for this part is the **iris** dataset provided by the sklearn library. The dataset contains information on three types of iris plants, where one of them (**iris-setosa**) is linearly separable from the other two. The features of this dataset are:

- sepal length in cm
- sepal width in cm
- petal length in cm
- petal width in cm

$$L_{BCE} = -\frac{1}{n}\sum_{i=1}^{n}(Y_i \cdot \log \hat{Y}_i + (1 - Y_i) \cdot \log (1 - \hat{Y}_i))$$

Figure 3: Binary Cross Entropy Loss Function

- class:
  - Iris Setosa
  - Iris Versicolour
  - Iris Virginica

We have not provided this dataset as it is easily accessible through the library itself. Therefore, you will load it yourself. However, as you will notice when you load the dataset from sklearn, the classes are not stored in strings but as 0s, 1s, and 2s, where a number represents each type. However, since we are performing simple logistic regression to differentiate between linearly separable data, you have to change the class labels to only 0s and 1s, where 0 would represent the Iris Setosa class while 1 would represent both of the other two. After this, perform a 70-30 test-train split. However, **please note** that you have to ensure that no class imbalances occur in them, which means that both the test and train dataset contain roughly equal numbers of both instances (1s and 0s).

Logistic Regression from Scratch [15 marks]

Now that you have the dataset, you will perform logistic regression using gradient descent and binary cross-entropy loss (**Figure 3**) and finally find the test loss. *Hint:*This part is similar to implementing linear regression from scratch, so refer to that.

K-Fold Cross-Validation [15 marks]

As you might have noticed, our iris dataset is quite small with only 150 instances, and those too lessen after doing the test-train split. Therfore, to make our model perform better, you are now going to implement k-fold cross-validation. In k cross-validation, you divide your training set into k parts and use k-1 of them for training, while one part is used for validation. We then rerun this k times, where a different part from the total of k parts is used for validation each time. Note that the test data set still remains completely separate. This will result in you having k different weights (one for each fold). In the end, take an average of all the weights and those will be your final weights.
In this part, you are going to implement k-fold cross-validation, keeping in mind these points and then use the weights obtained to run the model on the test dataset, reporting the loss (Entropy Loss) and answering the question asked in the notebook.

Logistic Regression using Libraries [5 marks]

This part is pretty easy and straightforward. All you have to do is use sklearn libraries to run logistic regression on your iris training dataset and use that model to test the iris test dataset and report the cross entropy loss on the test dataset, and you're done.

Part Three [SVMs]:

SVM Training and Visualisation [14 marks]

In this part, you will again use the iris dataset and train SVMs on it for three different kernels: linear, polynomial, and rbf. You are then to visualize the decision functions of all three SVMs, similar to how they are visualized in **Figure 4**.

To help with the plotting, please look into the following functions:

- np.meshgrid

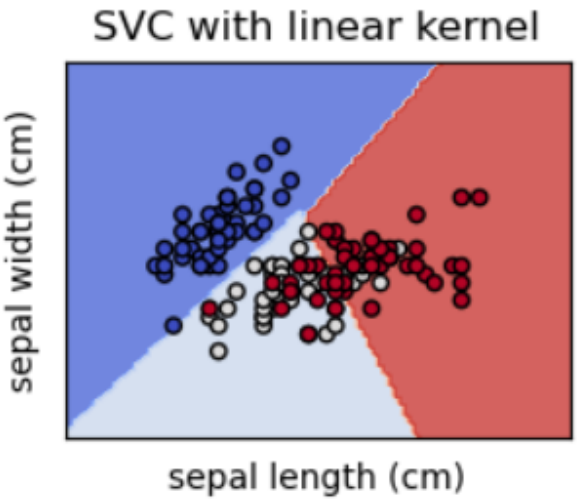

Figure 4: SVM with Linear Kernel

- plt.scatter
- plt.contourf

Please note however that if you want to plot these decision boundaries using some other functions, you are completely allowed to do so.

Testing the SVMs [6 marks]

For this part, find the accuracies for all three SVMs that you trained. For finding accuracies use the same train test data split of 70-30 that you used in logistic regression. Finally, answer the question asked in the jupyter notebook.

BEST OF LUCK FOR YOUR ASSIGNMENT !!!