###### SFWR TECH 4DA3 / 6DA3 – Data Analytics and Big Data

**Midterm Exam**

EVENING CLASS Dr. J. Fortuna

DURATION OF EXAMINATION: 2 Hours October 2020

McMaster University Midterm Examination

This examination paper includes 5 pages and 4 questions. You are responsible for ensuring that your copy of the paper is complete.

**SPECIAL INSTRUCTIONS:**

All work is to be completed using Python. Please submit a .py file (i.e. your code) as a solution for each question to the course dropbox. The marks allocated for each question are shown on this exam paper. This is an “open book” examination. All resources – including the course website – are available for your use.

The distribution of marks is as follows (do not write in this box !):

|  |  |  |
| --- | --- | --- |
| Question | Marks  Available | Mark  Obtained |
| 1 | **5** |  |
| 2 | **5** |  |
| 3 | **10** |  |
| 4 | **10** |  |
| Total | **30** |  |

**Question 1 (5 marks). Multivariate Data**

Use Python to implement the following:

1. Create 1000 30-dimensional datapoints drawn from a **multivariate uniform distribution** where each dimension’s values range from **-10 to 10 inclusive**. These correspond to the **X** values
2. Create an error vector of length 1000 of values drawn from a **normal distribution** with a standard deviation of 3 and a mean of zero.
3. Randomly generate 31  values drawn from a **uniform distribution** where the values range from **-10 to 10 inclusive**.
4. Calculate an output **y** representing a multiple linear regression model including error using the datapoints, beta values, and error vector that you created.
5. Save both the **X** values and the **y** values as a space delimited text file called **data1.txt**. Your file must contain a single matrix with 31 columns – the first 30 are for the **X** values and the 31st column is for the **y** values
6. Additionally, save the randomly generated values to a space delimited text file called **betas.txt**.

The easiest way to load and save space delimited files is to use numpy.loadtxt(…) and numpy.savetxt(…) respectively.

**Question 2 (5 marks). Multivariate Data**

Use Python to implement the following:

1. Create 1000 2 dimensional datapoints (class 1) drawn from a **normal distribution** with 2 random variance values drawn from a **uniform distribution** with a range between **2 and 5 inclusive** and 2 random means drawn from a **uniform distribution** with a range between **-3 and 0 inclusive**. There is to be no covariance in this data.
2. Create another 1000 2 dimensional datapoints (class 2) drawn from a normal distribution with 2 random variance values drawn from a **uniform distribution** with a range between **1 and 3 inclusive** and 2 random means drawn from a **uniform distribution** with a range between **0 and 3 inclusive**. There is to be no covariance in this data.
3. Concatenate these datapoints to create a single matrix of 2000 datapoints where the first 1000 will be considered to belong to class 1 and the second 1000 will be considered to belong to class 2 (this data will be used in question 4).
4. Save the data as a space delimited text file called **data2.txt**.

The easiest way to load and save space delimited files is to use numpy.loadtxt(…) and numpy.savetxt(…) respectively.

**Question 3 (10 marks): Multivariate Regression**

Use Python to implement the following:

1. Calculate all of the regression parameters (**beta** values) for the data in data1.txt (**X** values and **y** values).
2. Create another 500 30-dimensional X values drawn from a **uniform distribution** with a range of **-10 to 10 inclusive**. These will be called **Xtest**. Additionally, create an error vector of length 500 with a mean of 0 and a standard deviation of 3. Calculate output **y** values using **Xtest** and the **ORIGINAL beta** values from the **betas.txt** file you calculated in Question 1 and the error vector. These will be called **ytest**. In other words, what we have done is create a test set using the SAME parameters as the data we generated in Question 1.
3. Using the beta values you calculated in step 1, determine the outputs **y** for the 500 30 dimensional **Xtest** values. These y values will be called **ypred**. This corresponds to your predicted **y** values for the test data **Xtest**
4. Find the **mean square error** on the test data. To calculate the mean square error, simply take the difference between **ytest** and **ypred**, square the difference, sum all of the squared differences and then divide by the number of samples.

**Question 4 (10 marks): Fishers Linear Discriminant (FLD)**

From the data in the file data2.txt, train a FLD classifier and determine the value of the threshold such that when the **training data** is classified, there are “few” errors. In other words, try some different values of the threshold and find the one that produces the smallest number of errors over both classes. You do not need to use a for loop to try all possible values of the threshold – simply try a few of them…

Provide a scatter plot of the training data along with the **discriminant line** (i.e. the line perpendicular to the vector **w** as found from FLD). To determine the discriminant line, you must use the threshold that you found above when you were trying to minimize the error. In other words, the intercept of the discriminant line may not be zero.