**Logo

Description automatically generated**

**San Francisco Bay University**

**CS483 - Fundamentals of Artificial Intelligence**

**Homework Assignment #2**

**Due day: 6/26/2022**

**Instruction:**

1. **Push the source code to Github**
2. **Overdue homework submission could not be accepted.**
3. **Take academic honesty and integrity seriously (Zero Tolerance of Cheating & Plagiarism)**

1. From the following dataset with one feature *X*, please write python program to implement linear regression in **Normal Equation** to find **.** After that,plot (X,Y) points and fitting curve in **excel** or **matplotlib** python function

Table

Description automatically generated

2. Create hypothesis function/loss function/cost function for binary classification, and train your module by python program based on gradient descent algorithm from the following breast cancer dataset. After that, predict which class of last two records (highlighted in red color) is

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Clump Thickness  (Range:  1~10)  X1 | Uniformity of Cell Size  (Range:  1~10)  X2 | Uniformity of Cell Shape  (Range:  1~10)  X3 | Marginal Adhesion  (Range:  1~10)  X4 | Single Epithelial Cell Size  (Range:  1~10)  X5 | Bare Nuclei  (Range:  1~10)  X6 | Bland Chromatin  (Range:  1~10)  X7 | Normal Nucleoli  (Range:  1~10)  X8 | Mitoses  (Range:  1~10)  X9 | Class:  (2: benign,  4: malignant)  Y |
| 8 | **10** | **10** | **8** | **7** | **10** | **9** | **7** | **1** | **4** |
| 5 | **3** | **3** | **3** | **2** | **3** | **4** | **4** | **1** | **4** |
| 1 | **1** | **1** | **1** | **2** | **3** | **3** | **1** | **1** | **2** |
| 8 | **7** | **5** | **10** | **7** | **9** | **5** | **5** | **4** | **4** |
| 7 | **4** | **6** | **4** | **6** | **1** | **4** | **3** | **1** | **4** |
| 4 | **1** | **1** | **1** | **2** | **1** | **2** | **1** | **1** | **2** |
| 4 | **1** | **1** | **1** | **2** | **1** | **3** | **1** | **1** | **2** |
| 10 | **7** | **7** | **6** | **4** | **10** | **4** | **1** | **2** | **4** |
| 6 | **1** | **1** | **1** | **2** | **1** | **3** | **1** | **1** | **2** |
| 7 | **3** | **2** | **10** | **5** | **10** | **5** | **4** | **4** | **4** |
| 10 | **5** | **5** | **3** | **6** | **7** | **7** | **10** | **1** | **4** |
| 3 | **1** | **1** | **1** | **2** | **1** | **2** | **1** | **1** | **2** |
| 8 | **4** | **5** | **1** | **2** | **5** | **7** | **3** | **1** | **4** |
| 1 | **1** | **1** | **1** | **2** | **1** | **3** | **1** | **1** | **2** |
| 5 | **2** | **3** | **4** | **2** | **7** | **3** | **6** | **1** | **4** |
| 3 | **2** | **1** | **1** | **1** | **1** | **2** | **1** | **1** | **2** |
| 5 | **1** | **1** | **1** | **2** | **1** | **2** | **1** | **1** | **2** |
| 2 | **1** | **1** | **1** | **2** | **1** | **2** | **1** | **1** | **2** |
| 1 | **1** | **3** | **1** | **2** | **1** | **1** | **1** | **1** | **2** |
| 3 | **1** | **1** | **1** | **1** | **1** | **2** | **1** | **1** | **2** |
| 2 | **1** | **1** | **1** | **2** | **1** | **3** | **1** | **1** | **2** |
| 10 | **7** | **7** | **3** | **8** | **5** | **7** | **4** | **3** | **4** |
| 2 | **1** | **1** | **2** | **2** | **1** | **3** | **1** | **1** | **2** |
| 3 | **1** | **2** | **1** | **2** | **1** | **2** | **1** | **1** | **2** |
| 2 | **1** | **1** | **1** | **2** | **1** | **2** | **1** | **1** | **2** |
| 10 | **10** | **10** | **8** | **6** | **1** | **8** | **9** | **1** | **4** |
| 6 | **2** | **1** | **1** | **1** | **1** | **7** | **1** | **1** | **2** |
| 5 | **4** | **4** | **9** | **2** | **10** | **5** | **6** | **1** | **4** |
| 2 | **5** | **3** | **3** | **6** | **7** | **7** | **5** | **1** | **4** |
| 10 | **4** | **3** | **1** | **3** | **3** | **6** | **5** | **2** | **4** |
| 6 | **10** | **10** | **2** | **8** | **10** | **7** | **3** | **3** | **4** |
| 5 | **6** | **5** | **6** | **10** | **1** | **3** | **1** | **1** | **4** |
| 10 | **10** | **10** | **4** | **8** | **1** | **8** | **10** | **1** | **?** |
| 6 | **6** | **6** | **9** | **6** | **2** | **7** | **8** | **1** | **?** |

3. Given the following dataset, design python function as **binary** classifier for the following two classes.

- Plot all points in two different classes first in Excel or python matplotlib functions

- Observe what boundary decision **function** is good to separate two classes

- Build up hypothesis function/loss function/cost function based on your selected decision function

- Write python program to train your model

- After model training, plot decision boundary function in Excel or python matplotlib functions

|  |  |  |
| --- | --- | --- |
| X1 | X2 | Y |
| -3.98 | **-0.12** | **1** |
| -3.464 | **-2.11** | **1** |
| -3.461 | **1.89** | **1** |
| -2.22 | **-3.474** | **1** |
| -2.02 | **0.03** | **0** |
| -2.01 | **3.459** | **1** |
| -1.42 | **-1.409** | **0** |
| -1.416 | **1.419** | **0** |
| -1.09 | **0.08** | **0** |
| -0.19 | **-4.13** | **1** |
| 0.01 | **1.02** | **0** |
| 0.03 | **-2.12** | **0** |
| 0.04 | **2.06** | **0** |
| 0.06 | **3.97** | **1** |
| 0.07 | **0.1** | **0** |
| 0.12 | **-1.12** | **0** |
| 1.11 | **0.09** | **0** |
| 1.411 | **1.419** | **0** |
| 1.414 | **-1.415** | **0** |
| 1.86 | **3.47** | **1** |
| 1.96 | **-0.12** | **0** |
| 2.11 | **-3.472** | **1** |
| 3.461 | **-1.87** | **1** |
| 3.464 | **2.07** | **1** |
| 4.12 | **0.09** | **1** |

4. Train your hypothesis **functions** for multiclass classification from the following given dataset by python program. And then predict what Y’s value is in the last 4 samples (highlighted in red color). Verify your classifier model by plotting all points and decision lines

|  |  |  |
| --- | --- | --- |
| X1 | X2 | Y |
| 3.25 | **7.956** | **2** |
| 3.3 | **2.2** | **0** |
| 3.32 | **3.41** | **0** |
| 3.35 | **10.272** | **2** |
| 4.01 | **1.65** | **0** |
| 4.03 | **2.51** | **0** |
| 4.05 | **4.21** | **0** |
| 4.05 | **7.38** | **2** |
| 4.06 | **11.412** | **2** |
| 4.07 | **9.198** | **2** |
| 5.22 | **2.15** | **0** |
| 5.24 | **3.41** | **0** |
| 5.25 | **7.866** | **2** |
| 5.28 | **10.008** | **2** |
| 8.15 | **6.3** | **1** |
| 8.23 | **7.95** | **1** |
| 9.38 | **7.34** | **1** |
| 9.4 | **8.21** | **1** |
| 10.2 | **6.52** | **1** |
| 10.8 | **7.72** | **1** |
| 4.01 | **3.02** | **?** |
| 9.1 | **6.5** | **?** |
| 3.50 | **9.50** | **?** |
| 6.01 | **6.01** | **?** |