There is a single task with four sub-tasks. It is not mandatory to solve all the sub-tasks. Solve as many as you can. If you are not comfortable with coding, just give it a try. That will also fetch you some points. We would like to see how you approach the problem and how organized your presentation of thoughts is. **For mathematical solution, you are required to upload a scanned pdf of your handwritten solution. You may also type the solution in MS Word or Latex whichever is preferred. For coding, you are required to submit the python files. Name the python file for subtask-2 as {your_roll_no_subtask2.py} and for subtask-4 as {your_roll_no_subtask4.py}. Make sure to write how you read the data at the beginning of your code. It will be better if you keep a path variable for loading the data which will make our evaluation easier.**

Task: Linear Classifiers using Logistic Regression and Gaussian Discriminant Analysis (100 points)

In this task, you are required to learn about two probabilistic linear classifiers. First, a discriminative linear classifier: logistic regression. Second, a generative linear classifier: Gaussian discriminant analysis (GDA). Both the algorithms find a linear decision boundary that separates the data into two classes, but make different assumptions. The goal of this task is to test your mathematical understanding of the algorithms and your coding skills. For the task, we will consider two datasets, provided in the following files:

- data/ds1_train,test.csv
- data/ds2_train,test.csv

Link to Dataset

Each file contains m examples, one example $(x^{(i)}, y^{(i)})$ per row. In particular, the i^{th} row contains columns $x_0^{(i)} \in \mathbb{R}$, $x_1^{(i)} \in \mathbb{R}$, and $y^{(i)} \in \{0, 1\}$. In the sub-tasks that follow, perform Logistic Regression and Gaussian Discriminant Analysis on these two datasets.

First Subtask: 20 points

The average empirical loss function for Logistic Regression is defined as:

$$J(\theta) = -\frac{1}{m}\sum_{i=1}^{m} y^{(i)}log(h_\theta(x^{(i)})) + (1 - y^{(i)})log(1 - h_\theta(x^{(i)}))$$

where $y^{(i)} \in \{0, 1\}$, $h_\theta(x) = g(\theta^T x)$ and $g(z) = \frac{1}{1+e^{-z}}$ and θ is an n dimensional parameter vector

Problem: Find the Hessian matrix H of the empirical loss function with respect to θ, and show that the hessian H is positive semi-definite in nature.

Second Subtask: 30 points (Coding)

In this sub-task, you need to fit a Logistic Regression model on both the datasets and report the accuracy for the training set and the test set. For classification, label a probability greater than or equal to 0.5 as 1, and a probability less than 0.5 as 0. Note, you have to implement Logistic Regression from scratch and optimize the weights using Gradient Descent algorithm. (use of Logistic Regression from sklearn won't fetch you any points). You may also plot the loss function with respect to the number of iterations for the training set. Print the respective prompts so as to make the output readable and add comments as necessary. Decide the value of the hyper-parameter learning rate by yourself. For tuning you may use Randomized Search Cross Validation, but your final code should not implement Randomized Search. It is only for your experiment. Choose the best stopping condition for gradient descent.

Note: Use of object-oriented programming is preferred, but not mandatory.

1 2

Third Subtask: 20 points

For Gaussian Discriminant Analysis, the joint probability distribution of (x, y) is given by the following equations:

$$p(y) = \begin{cases} \phi & if\, y = 1 \\ 1-\phi & if\, y = 0 \end{cases}$$

$$p(x|y=0) = \frac{1}{(2\pi)^{n/2}|\Sigma|^{1/2}} exp\Big(-\frac{1}{2}(x-\mu_0)^T\Sigma^{-1}(x-\mu_0)\Big)$$

$$p(x|y=1) = \frac{1}{(2\pi)^{n/2}|\Sigma|^{1/2}} exp\Big(-\frac{1}{2}(x-\mu_1)^T\Sigma^{-1}(x-\mu_1)\Big)$$

where ϕ, μ_0, μ_1, and Σ are the parameters of the model.

Let us assume that ϕ, μ_0, μ_1, and Σ are already found using some mathematical manipulation, and now we want to predict y given a new point x. In order to show that Gaussian Discriminant Analysis results in a classifier that has a linear decision boundary, show that the following expression is true.

$$p(y=1|x;\phi,\mu_0,\mu_1,\Sigma) = \frac{1}{1+exp(-(\theta^Tx+\theta_0))}$$

where $\theta \in \mathbb{R}^n$ and $\theta_0 \in R$ are appropriate functions of ϕ, μ_0, μ_1, and Σ.

Hint: Use Bayes theorem in order to get the above probability and with some mathematical manipulation on the expression obtained using Bayes Rule, try to express it in the form shown in the Right Hand Side, ultimately comparing the expressions, you will obtain the required result.

Fourth Subtask: 30 points (Coding)

By maximising the log likelihood of the probability distribution, we indeed obtain the values of the parameters i.e., the values of ϕ, μ_0, μ_1 and Σ. As you might not be aware about the likelihood of a probability distribution, the expressions for the optimal value of the parameters are already given below.

$$\phi = \frac{1}{m}\sum_{i=1}^{m} 1\{y^{(i)}=1\}$$

$$\mu_0 = \frac{\sum_{i=1}^{m} 1\{y^{(i)}=0\}x^{(i)}}{\sum_{i=1}^{m} 1\{y^{(i)}=0\}}$$

$$\mu_1 = \frac{\sum_{i=1}^{m} 1\{y^{(i)}=1\}x^{(i)}}{\sum_{i=1}^{m} 1\{y^{(i)}=1\}}$$

$$\Sigma = \frac{1}{m}\sum_{i=1}^{m}(x^{(i)}-\mu_{y^{(i)}})(x^{(i)}-\mu_{y^{(i)}})^T$$

Using the results mentioned above, write code to find the values of the parameters. Finally, find the probability of each of the training and test examples using the expression of probability that you had to prove given in the previous sub-task (expression resembling sigmoid function). Note that you have already obtained θ and θ_0 in terms of the parameters ϕ, μ_0, μ_1 and Σ. After you find the probabilities for the training and testing examples, mark a probability greater than or equal to 0.5 as 1 and a probability less than 0.5 as 0. Finally, print the accuracy of training set and test set for both the datasets.