Assignment III
Calculus, Optimization, Probability
6th August 2023 11:55 PM

	You should submit a pdf. It should contain each question, then code you have written, result obtained followed by your interpretation/inference/comments of the result. Use paper and pen only if it's very much necessary. Try to do it in computer (Matlab/Octave) to the extent possible
0.	Write answers to all questions asked in the Mid-Term examination. Then upload the scanned pdf along with the Assignment 3 submission.
1.	Use pen and paper to find the critical points for the function x^2y^2 - x^2 - y^2 . Which of these are minima, maxima, and saddle points (Refer)? Plot the function and comment on the behaviour of the function at the critical point using https://www.wolframalpha.com/
2.	Generate a toy Dataset: (-1,-1),(-2,-2),(-2,-1),(-1,2), (1,2),(2,1),(1,1),(2,2) . a. Fit a regression line (using pseudo inverse and using GD), what is the MSE on the training set?
3.	Take a linear regression dataset from the website and experiment (You can use all forms of regression, including LASSO and Ridge, and you can use library function) https://www.kaggle.com/rtatman/datasets-for-regression-analysis
4.	Use Newton's (using paper & pen, own code, and built-in function) method to find the optimal line for the question in Q2.
5.	Generate 1000 random numbers from standard uniform (U) and normal distribution (X). a. Now transform the random variable U[0, 1] to V[a = 5, b = 10] b. Transform X(0, 1) to Y(5, 10) (i.e., mean 5 and std 10) c. What are the values of first, second, third, and 4th moments and central moments (as per theory) for the random variables V and Y? d. What are the values of first, second, third, and 4th moments and central moments for the random variables V and Y as per the given sample?

```
function LinRegression()
    clear all; close all;
    DataPoints = [-10 -10;-8 -2; -6 -12; -4 -4; 10 10;8 2; 6 12; 4 4; ];
    DataPoints = DataPoints';
    DataPoints = [ones(1, size(DataPoints, 2)); DataPoints];
    RegMdl = [0;-1]; %y = -X
    W = findRegModel(DataPoints, RegMdl);
end

function W = findRegModel(DataPts, W)
    xMin = min(DataPts(2, :)) - 1;
    xMax = max(DataPts(2, :)) + 1;
    xrange = xMin:0.01:xMax; alpha = 0.05;
    Data = DataPts(1:end-1, :); predY = DataPts(end, :);
    for i = 1:30
        y = (W(1)+W(2)*xrange);
        clf; plot (DataPts(2, :), DataPts(3, :), 'bo');  hold on;
        plot (xrange, y,'-.r');    pause(1);

        hThetaX = W'*Data;
        gradJ = mean(repmat((hThetaX - predY), size(Data, 1), 1).*Data, 2);
        W = W - alpha*gradJ; %apply gradient descent

        if (max(abs(alpha*gradJ)) < 0.01)
            %if gradient is almost zero
            return;
        end
    end
end
```

6.	Load Fisher Iris Data and understand the dataset. Divide 70% of the sample at random and keep it as training set. Plot the values (histogram) of each feature for all training sample and pick two features (out of 4) that you feel can be modelled using Gaussian distribution. 1. Now assume that the samples (with the above selected features) follows normal distribution. Design a Bayes classifier for the 30% test sample and report the accuracy. 2. Repeat step 1, but this time build a naïve Bayes classifier.