Assignment 02

Total marks: 30

Instructions:

1. Collate your codes (as jupyter-notebooks (.ipynb) or Python files (.py)) and PDF into a zip folder and rename it with your entry number.
2. Submit three separate code files each for Nonlinear regression, Multivariate regression and Logistic regression.
3. Compile into a single PDF (Figures.pdf) the relevant plots and their inferences.
4. Plots should be labeled clearly.
5. Do not use any library until it is specified. You may use `numpy` and `matplotlib`.

1 Nonlinear Regression [14 Marks]

Consider a classic example of throwing up a tennis ball in the air. We can predict the ball's height (h) at any instance of time (t) using Newton's laws of motion (ignoring air resistance). A data set (prob1data.txt) which follows similar trajectory is provided for training your model.

1. Plot the training data. Write a code in Python to perform nonlinear regression on the given data. Implement batch gradient descent algorithm for optimization. (Choose $\alpha = 0.01$, number of iterations = 50000) [5 Marks]
2. Implement stochastic gradient descent for optimization of weights. Plot cost history (J) vs number of iterations for both cases batch gradient descent and stochastic gradient descent. Comment on the difference, if any. [2 Marks]
3. Plot the cost history (J) vs number of iterations for different learning rates ($\alpha = 0.1, 0.5, 0.01, 0.05$). Write your inferences from the plot. [1 Mark]
4. Implement line search method (Secant method) to find learning rate (α). Optimize the weights using batch gradient descent and plot the cost history (J) vs number of iterations for variable learning rate. Comment on the difference between implementing line search method and choosing arbitrary α. [6 Marks]

2 Multivariate Regression [6 Marks]

Housing Price Prediction Problem. Suppose 'Mr. X' is planning to buy a house in Delhi and wants to predict the price of the house given some features like number of bedrooms, number of bathrooms, area of the house, etc.
The file 'prob2data.csv' contains a training set of housing prices in Delhi.

1. Read the excel file using `pandas` and perform data cleaning. Remove 1^{st} column 'id' which may not be necessary here. Perform mean normalization of features. [1 Mark]
2. Write a Python code to perform multivariate regression to predict the house price. Consider all 5 columns ('bedrooms',...,'yr built') as features. Implement batch gradient descent for optimization of weights. [4 Marks]
3. Predict the house price using the model, for 4 bedrooms, 2.5 bathrooms, 2570 sq. feet area, 2 floors, 2005 yr. built, and state the difference between the model prediction and actual value (Rs. 719000). Show in % error. [1 Mark]

3 Logistic Regression - Linear Classifier [10 Marks]

Apply the logistic regression (linear classifier) algorithm discussed in the lab session to predict next-day rain based on the 10 years of daily weather observations from many locations within a country. The dataset contains many factors taken into consideration to specify whether it rained or not on that particular day. The training and testing dataset is provided in the files titled 'weather_train.csv' and 'weather_test.csv', respectively. Carry out the following tasks as assignment problems:

1. Inspect and plot some portion of the training data using `pandas`. Segregate the training and testing data into two separate variables consisting of 'feature values' and corresponding 'predictions' (the prediction column is titled '`RainTomorrow`' in the dataset). To simplify the problem a bit, clean the whole data by carrying out the following sub-tasks:

   (a) Convert the predictions in the binary format by using '`1`' for '`YES`' and '`0`' for '`NO`'.

   (b) Identify and drop the feature columns having datatype 'object'.

   (c) Identify cells having '`NaN`' or '`NA`' values and replace them with mean values of their respective columns.

   (d) Normalize all the feature values by scaling them between `0` and `1`. The values in feature matrix '$\mathbf{X}$' can be normalized as:

   $$\mathbf{X}_{norm} = \frac{\mathbf{X} - min(\mathbf{X})}{max(\mathbf{X}) - min(\mathbf{X})}$$

   Execute the above sub-tasks and display some portion of the data and its head after each data cleaning step. [3 Marks]

2. Classify the cleaned dataset using binary classification algorithm discussed in the class and calculate the optimized weights and training set accuracy for the model (use Truncated Newton's Method in `SciPy` for optimization). [3 Marks]

3. Plot the cost history (J) vs. the number of iterations. [Hint: You can make use of 'callback function' in `Optimize.minimize` to store the cost history] [3 Marks]

4. Apply the trained model on the cleaned test dataset to predict the testing accuracy of the model. [1 Marks]