CSE343/ ECE363/ ECE563: Machine Learning W2022
Assignment: Linear / Logistic Regression
Max Marks: Programming: 100+5(BONUS), Theory: 18(UG/PG) + 12 (for PG only)
Due Date: 3/3/2022, 11:59 PM

Instructions

- This is a individual assignment.
- Try to attempt all questions.
- The theory questions should be your individual effort. Copying/Plagiarism will be dealt with strictly.
- Start early, solve the problems yourself.
- Late submission penalty: Refer policy on course website.
- Your submission would be a single .zip file (rollno_HW1.zip) file, that would contain two items (codes + .pdf file). You have to include all your plots, results, analysis, conclusion, and solutions for the theory questions in the pdf file. **Code should be well documented. Use comments for python scripts or markdown in Jupyter Notebooks.**
- Anything not written in the report will fetch 0 marks.
- It is preferred that you write LaTeX reports.
- Remember to **Turn in** after uploading on Google Classroom. No excuses or issues would be taken regarding this after the deadline.
- Start the assignment early. Resolve all your doubts from TAs in their office hours **at least two days before the deadline.**

PROGRAMMING QUESTIONS

1. (20 points) **Basic operations + Data visualization**
    1. (10 points) : Download the IRIS dataset.
        (a) Load the dataset using Pandas library in python.
        (b) Print column information (name, data types, value range or counts).
        (c) Plot histograms for continuous valued attributes and bar graphs for the discrete valued attributes and the target class.
    2. (10 points) : MNIST dataset.

(a) Load the dataset using "idx numpy" package.

(b) Visualize 2 random images from the dataset.

(c) Use TSNE (t-distributed stochastic neighbour embedding) algorithm to reduce data dimensions to 2, and plot the resulting data as a scatter plot. Comment on the separability of the data.

All model related tasks have to be done on a 80% train + 10% validation + 10% test split of the data. You can use scikit-learn for splitting the dataset.
The 90% (train+val) data needs to be split into 5-folds, treating 4 folds as train set and the fifth fold as your val set. **Scikit-learn KFold implementation may be used.**
For each fold performance has to be reported on their respective validation sets. Overall model performance should be reported on the 10% test set.
No KFold usage will lead to 0 marks in the assignment.

2. (35 points) **Linear Regression**

    1. (15 points) : Implement Linear Regression for the Abalone Dataset. The dataset contains 9 variables out of which the last column is the output variable and the other 8 are input attributes. Visualize some attributes via histograms. You need to implement gradient descent **from scratch** i.e. you cannot use any libraries for training the model (You may use numpy, but libraries like sklearn are not allowed). Choose an appropriate learning rate. You may need feature normalisation to achieve good performance and stable training.

        (a) (5 points) Plot the iteration vs RMSE graph for all 5 models. RMSE should be reported on the val set.

        (b) (8 points) Modify your Regression implementation by including L1 (LASSO) and L2 (Ridge Regression) regularization. Implement both regularization functions from scratch and train the model again. Try different values of the regularization parameter and report the best one. Plot similar iteration vs RMSE graph as before.

        (c) (2 points) Report test set accuracy of the best models, i.e. Only regression, Regression+L1, Regression + L2.

        (d) (3 points) Use Scikit-learn's implementation of Linear Regression to train the above models and report accuracy on the test set. Compare and analyse any differences

        (e) (2 points) Also implement the normal equation (closed form) for linear regression and get the optimal parameters directly for each fold. Report the accuracy on respective val sets.

3. (45 + 5(BONUS) points) **Logistic Regression**

    1. (10 points) Implement Binary Logistic Regression for the UCI Ionosphere dataset. Use scikit learn's implementation without any regularization. There are 34 features and one binary target label. Calculate mean and variance for all independent

variables in the train set. Plot histograms and box plots for the 5 features with the highest variance.
Choose an appropriate learning rate. You may need feature normalization.

(a) (**BONUS** 5 points) Reduce the number of features via Principal Component Analysis (PCA) and use the reduced data set for model training. Retain different amounts of variance values, ranging from 0.9 to 1 in steps of 0.1. Compare results (Accuracy, Precision, Recall and F-1 score) of the best PCA model and without PCA model. A tutorial for PCA is given here. You can use scikit-learn for this.

(b) (10 points) Use Logistic Regression from scikit-learn with L1 and L2 regularization. Compare model performance on test set with and without it. Use Precision, Recall and F-1 metrics for comparison.

(c) (10 points) Plot ROC-AUC curve for various threshold values, ranging from 0 to 1 in steps of 0.1 (total 11 plots). Plot all ROC curves on the same graph. Calculate the values to be plotted. **Using prebuilt implementations for ROC-AUC is not allowed.** A tutorial on ROC curves is given here and here

(d) (5 points)Use Scikit-learn to plot the ROC-AUC curves and comment on any observed differences.

2. Implement Multiclass Logistic Regression for the MNIST dataset. Follow below methodologies. No need to use KFold for OVO and OVR implementations.

(a) (5 points) Train models in a One-vs-One (OVO) fashion. This splits data into classwise pairs and trains a model for only one classwise pair at a time. Then the results of each model are combined via a heuristic like majority voting for the final classification. **Use Scikit-Learn's implementation of OVO**. Report Accuracy, Precision, Recall and F-1 score for simple and L2 regularized Logistic regression on the test dataset.

(b) (5 points) Train models in a One-vs-Rest (OVR) fashion. This method treats a single class as 1 and others as 0. This results in 10 models, 1 for each class. **Use Scikit-Learn's OVR implementation**. Classify the test dataset. Do the same analysis as OVO.

THEORY QUESTIONS (for everyone)

4. (10+2x4=18 points) Linear Regression

1. Derive the closed form solution to the linear regression problem for the dataset $\mathcal{D} = \{(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \ldots, (\mathbf{x}_N, y_N)\}$, where $\mathbf{x}_i \in \mathbb{R}^d$, and $y_i \in \mathbb{R}$, for $i = 1, 2, \ldots, N$. Let $\mathbf{y} = \mathbf{X}\theta + \epsilon$ be the regression model, where $\mathbf{y}$ is an $N \times 1$ vector constructed by concatenating the target variables y_i, $i = 1, \ldots, N$, and the matrix $\mathbf{X}_{N \times d} = [\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N]^\top$ contains the input data vectors. The regression parameters θ needs to be estimated.

2. Write the conditions under which the closed form solution to Linear Regression exists.
3. If we have a closed form solution for Linear Regression, why do we use Gradient Descent? Give an example of a situation where Gradient Descent is a better option than closed form calculations.
4. Prove that for simple linear regression, the least square fit line always passes through the point $(\bar{X}, \bar{Y})$, where $\bar{X}$ and $\bar{Y}$ represent the arithmetic mean of the independent variables and dependent variables respectively.
5. Can we use Linear regression for classification? If yes, how?

THEORY QUESTIONS (For PG Students only)

5. (2x4=8 points) Which of the below expressions are linear regression models? Justify.
    1. $\theta_0 x_0 + \theta_1 x_1 + ... + \theta_n x_n = 0$
    2. $\theta_0 \sin(x_0) + \theta_1 \sin(x_1) + ... + \theta_n \sin(x_n) = 0$
    3. $\sin(\theta_0 x_0) + \sin(\theta_1 x_1) + ... + \sin(\theta_n x_n) = 0$
    4. $y_i = w_0 + \sum_{j=1}^{N} w_j sinh(x_{i,j})$
6. (2x2=4 points) Logistic Regression
    1. What is the loss function used to train a Logistic Regression model. Write the expression for the probabilistic model used and the mathematical expression for the loss function.
    2. Modify the above expression to include (a) Gaussian and (b) Laplacian (doubly exponential) regularization.