Machine Learning (PG) M2021
Assignment-2

Max Marks: 200 **Due Date**: 14/10/2021, 11:59 PM

Instructions

- This assignment should be attempted individually. Keep collaborations at high level discussions. Copying/Plagiarism will be dealt with strictly.
- You can only use python as the programming language.
- Create a '.pdf' report that contains your approach, pre-processing, assumptions etc. Add all the analysis related to the question in the written format in the report, anything not in the report will not be marked. Use plots where ever required. **Submit code, readme and report files in ZIP format with naming convention** `A2_rollno_name.zip`. This nomenclature has to be followed strictly
- Remember to **turn in** after uploading on Google Classroom. No justifications would be taken regarding this after the deadline.
- Start the assignment early. Resolve all your doubts from TAs in their office hours **two days before the deadline.**
- You should be able to replicate your results during the demo, failing which will fetch zero marks.

Important Instructions (applicable to all the questions)

- All the experiments are to be done with 5-fold splits with one fold used as a validation set and remaining four folds as the training sets at a particular instance. In this way, you will have a total of five models. You have to implement a generic function that can split the dataset in n-folds. You can not use any inbuilt function for this.
- Save your models using 'joblib'. During demo, you must be able to load your saved models and replicate the reported results.

1. (50 points) For this question you have to use Abalone Dataset. The dataset contains 9 variables out of which the last column is the output variable and the other 8 are input attributes. You may need feature normalisation.
(a) A file named 'Regression.py' containing a 'Regression' class is attached with the assignment. You need to fill the suitable code in this class. In this class you can use '.fit()' of 'LinearRegression' from the sklearn. However, you have to write '.predict()' from scratch using the outcomes of '.fit()'.
(b) Use the 'Regression' class to prepare a table containing training and validation mean square (MSE) error for each fold. Also, report the mean training and validation MSE. Implement your own MSE function. Also, compare the output of your function with MSE from the sklearn. **20 Points**

(c) Now, instead of using the ‘Regression’ class, use normal equations1 to make the predictions, and repeat the table in part (b) above. **20 Points**
(d) Finally, use the ‘LinearRegression’ from the sklearn to make the predictions, and prepare a similar table as in (a) and (b). Is there any deviation between the performance of the three approaches? If yes, why? **10 points**

2. (75 points) For this question you have to use the Diabetes dataset.
(a) Visualize and analyze the dataset. **5 Points**
(b) A file with the name ‘LogRegression.py’ containing a ‘LogRegression’ class is attached with the assignment. You have to write this class from the scratch without using sklearn.
(c) Using ‘LogRegression’ class, report the performance over 5-folds in terms of accuracy. Prepare a table similar to question (1) above. Also, plot the training curves with each fold as the validation set. For each fold there should be two plots, one for accuracy, and other for loss. Each plot should contain two curves, one representing training statistics, and other validation statistics. **30 Points**
(d) Modify the ‘LogRegression’ class to include the l2 regularization. Perform a grid search over the regularization constant (λ) to obtain its optimal value. With the optimal value of λ, repeat the tables and curves of part (c) above. Explain any difference in the performance. **30 Points**
(e) Now, use the logistic regression from the sklearn to obtain the performance over the 5-folds in (c) and (d). Is the performance similar to (c) and (d)? **10 Points**

3. (75 points) For this question you have to use this library to load MNIST dataset (60K train + 10k test).
(a) Visualize the dataset using 5 instances of each class. **5 Points**
(b) Logistic Regression is a binary classifier, i.e. it can be used to classify the datasets into two classes. However, it can be extended to multi class problem using One-vs-one (OVO) and One-vs-Rest (OVR) (Refer to ”Pattern Recognition and Machine Learning. Springer, Christopher M. Bishop, page 182”.) approaches. Extend the ‘LogRegression’ class in part 2 (d) to include the One-vs-One (OVO) approach. Prepare a performance table similar to question (2) above. Apart from this, prepare a table containing class-wise accuracy for each fold. Apart from provided reference for OVO and OVR, you can search for the other different sources also. **30 Points**
(c) Further extend ‘LogRegression’ class in (b) above to include the One-vs-Rest (OVR) approach. Repeat the results of part (b) above. **30 Points**
By now, you must have a generic ‘LogRegression’ capable of handling l2 regularization, OVO, and OVR for any number of classes (and not just the class numbers’ in the question).
(d) Finally, use logistic regression from the sklearn to repeat the above parts. Is there any performance difference? **10 Points**