MACHINE LEARNING

PROGRAMMING ASSIGNMENT - 1

BY:

ASHIMA GARG

PhD19003

Folder Description:

- Solution 1
    - Solution i)
        - DATASET
            - Dataset.data/Dataset.csv
        - RESULT
        - SOURCE
            - config : directory and hyperparameter information
            - main : driver functions
            - data : data reader and pre-processor class
            - model : class for model operations
            - utils : utility functions
    - Solution ii)
        - DATASET
            - Train.csv : obtained from Lowest RMSE Value in Solution i)
            - Test.csv : obtained from Lowest RMSE Value in Solution i)
        - RESULT
        - SOURCE
            - config : directory and hyperparameter information
            - main : driver functions
            - data : data reader and pre-processor class
            - model : class for model operations
            - utils : utility functions
            - model_L1 : class for model using L1 Regularization
            - model_L2 : class for model using L1 Regularization
            - model_sklearn : class for model using sklearn library
    - Solution iii)
        - DATASET
            - data.csv
        - RESULT
        - SOURCE
            - config : directory and hyperparameter information
            - main : driver functions
            - model : class for model operations
            - data : data reader and pre-processor class

      - model_L1 : class for model using L1 Regularization
      - model_L2 : class for model using L2 Regularization

- Solution 2
  - Solution a
    - DATASET
      - train.csv
      - test.csv
      - dataset_description : contains the description of the features in the dataset.
    - RESULT
    - SOURCE
      - config : directory and hyperparameter information
      - main : driver functions
      - model : class for model operations
      - data : data reader and pre-processor class
      - model_L1 : class for model using L1 Regularization
      - model_L2 : class for model using L2 Regularization
  - Solution b
    - DATASET
      - train-images-idx3-ubyte
      - train-labels-idx1-ubyte
      - t10k-images-idx3-ubyte
      - t10k-labels-idx1-ubyte
    - RESULT
    - SOURCE
      - config : directory and hyperparameter information
      - data : data reader and pre-processor class
      - main : driver functions
      - model : class for model operations

Solution 1) Linear Regression

Part i) Linear Regression on Abalone Dataset

Hyperparameters Used for Gradient Descent Plots are:

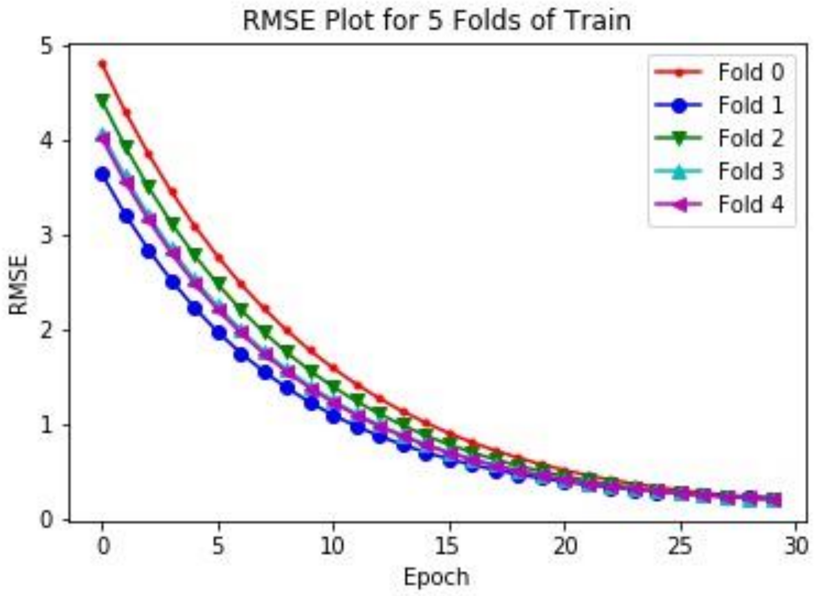
- Learning Rate: 0.001
- Number Of Epochs: 30

a) Training Set

- **RMSE Values For 5 Folds**

[0.4524278223932419, 0.3747874178412159, 0.4227582545441477, 0.40051982103426376, 0.39681849919133877]

- **RMSE vs Epochs Plot using Gradient Descent**

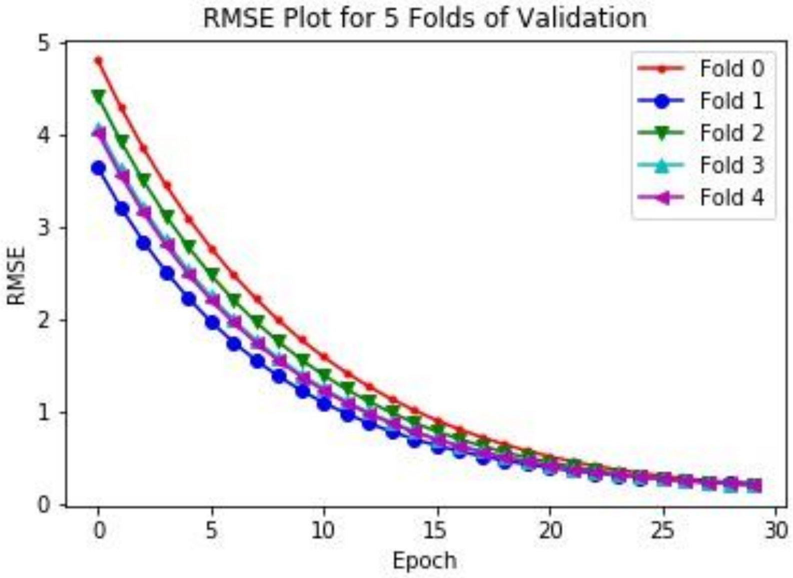


b) Validation Set

- **RMSE Values for 5 Folds**

[0.19819966178479356, 0.2023279098498869, 0.1859415452252507, 0.18873657999335897, 0.19144745475521174]

- **RMSE vs Epochs Plot using Gradient Descent**



b) RMSE using Normal Equation for 5 folds:

Training Set for 5 Folds

[0.015754244029368328, 0.012777735692760998, 0.013003133295745366, 0.01523872977219804, 0.015108727240727563]

Validation Set for 5 Folds

[0.00980025074660583, 0.0060083006427809225, 0.007159877140639773, 0.00676164065545948, 0.007031608846325517 4]

c) Comparison

- RMSE Values obtained from Normal Equation are lower than those obtained from Gradient Descent Algorithm.

Part ii) Regularization

Using Ridge, Lasso, GridSearchCV routines from sklearn library to perform 5-fold Cross Validation on train + validation test obtained from lowest RMSE using Normal Equation in Part i) which is **0.0060083006427809225**

Possible Values of Regularization Parameter (alpha) used are params = {'alpha':[0.000001, 0.00001, 0.0001, 0.001, 0.01, 0.1, 1, 10]}

a) Best Regularization Parameter for L2: 1e-06

b) Best Regularization Parameter for L1: 1e-05

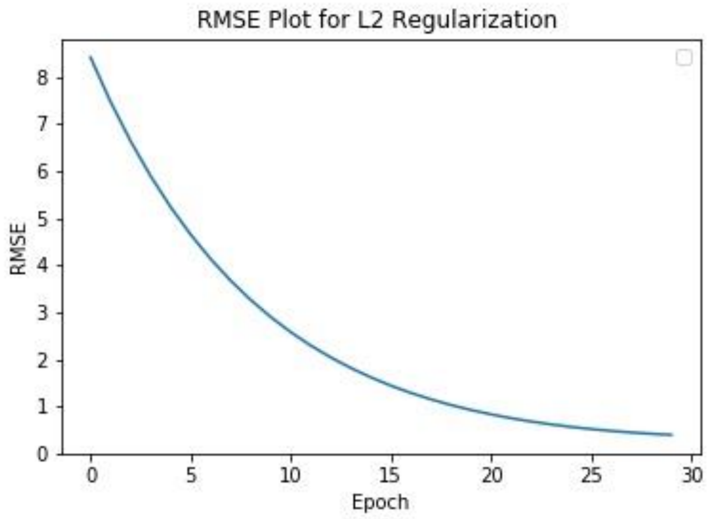
Using Gradient Descent Algorithm

Hyperparameters Used for Gradient Descent Plots are:

- Learning rate: 0.001
- Number of Epochs: 30

- ***L2 Regularization*** *with* Regularization Parameter for L2: ***1e-06***

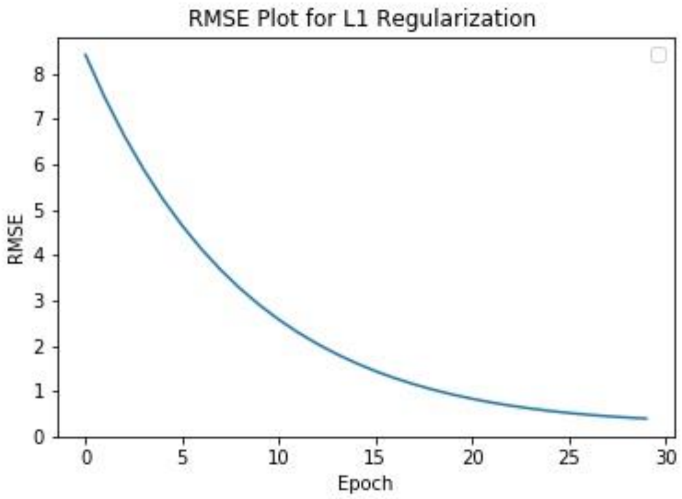
RMSE Plot for L2 Regularization



RMSE for Test Set using L2 Regularization: **0.3791853038179016**

- ***L1 Regularization*** *with* Regularization Parameter for L1: ***1e-05***

RMSE Plot for L1 Regularization



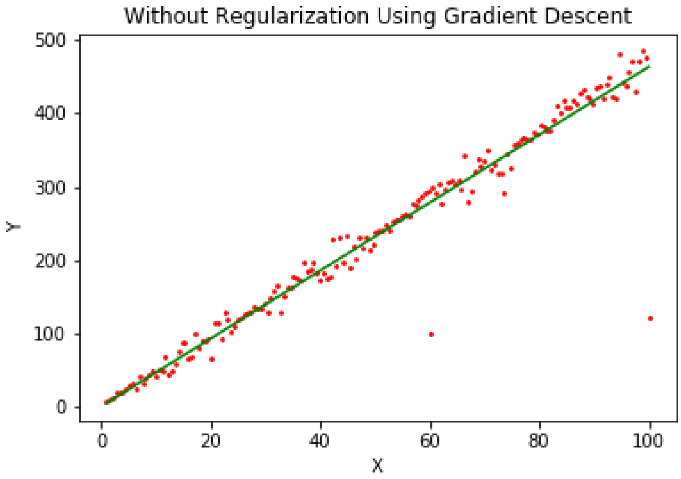
RMSE for Test Set using L1 Regularization: **0.3791853062673536**

Part iii) Best Fit Line Plots Using Gradient Descent Algorithm

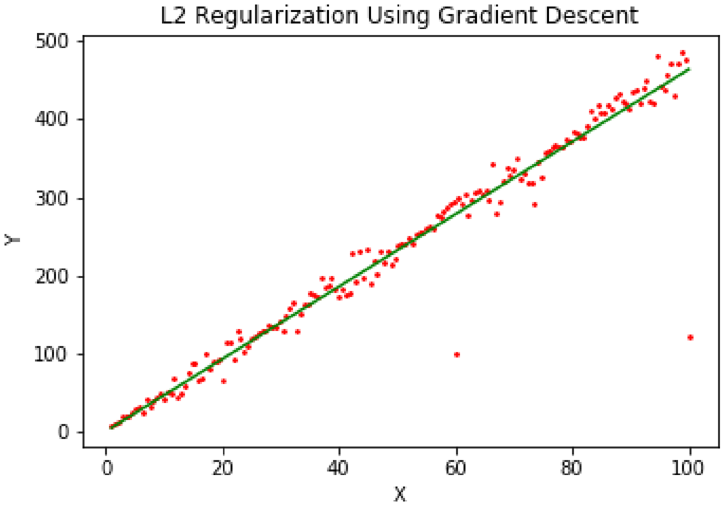
Hyperparameters Used for Gradient Descent Plots are:

- Learning rate = 0.0001
- Number of epochs = 10
- Regularization parameter(Lambda) = 0.01

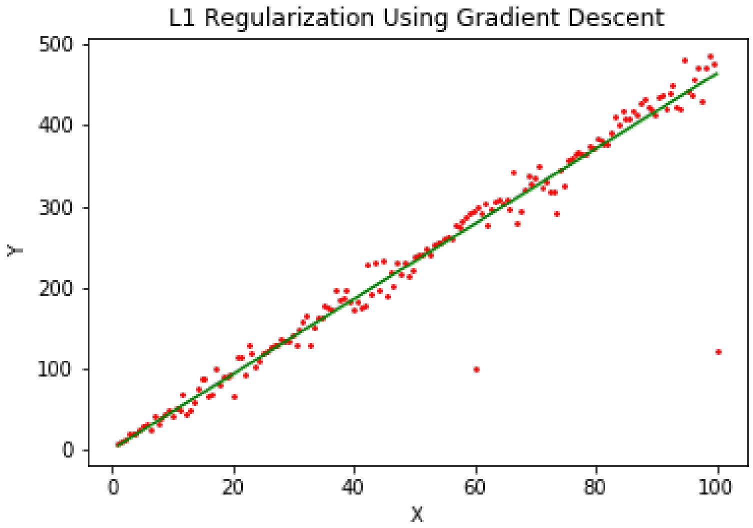
a) Without Regularization Plot



b) L2 Regularization Plot



c) L1 Regularization Plot



Observations Using Actual and Predicted Values

Case 1: Without Regularization

Actual	Predicted(With Gradient Descent)
8.4293375	4.63086793
10.51622485	7.36773623
12.33974404	10.10460453

Case 2: L2 Regularization

Actual	Predicted(With Gradient Descent)
8.4293375	4.63086383
10.51622485	7.36773217
12.33974404	10.1046005

Case 3: L1 Regularization

Actual	Predicted(With Gradient Descent)
8.4293375	4.63086793
10.51622485	7.36773623
12.33974404	10.10460453

Solution 2) Logistic Regression

Part i) Logistic Regression with L1 and L2 Regularization

All the features are not used to train the model. Only age, fnlwgt, education-num, capital-gain, capital-loss, hours-per-week are used as the features to model the data.

Data from Train.csv is divided into train and validation set with validation set having 30% of the randomly shuffled data. Data for Test Set is taken from Test.csv. Labels are encoded using sklearn library. Features are normalized before feeding into the model using:

Z =(x – min(x))/(max(x) – min(x))

Hyperparameters used:

- Number of Epochs: 2500
- Learning Rate: 0.1
- Regularization Parameter: 0.01

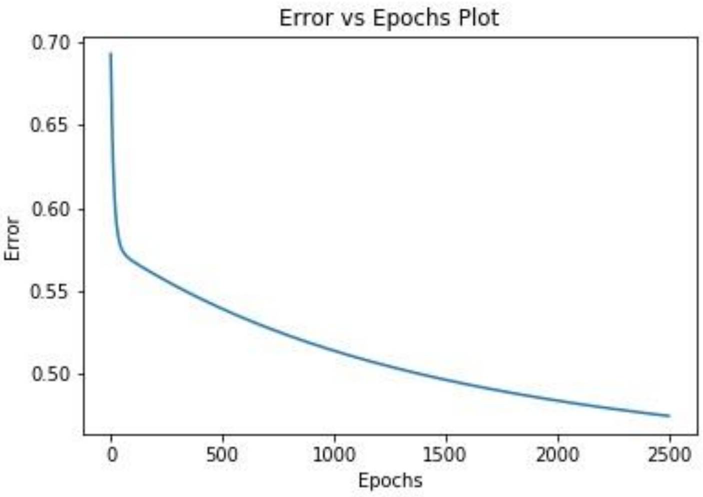
- **No Regularization**

Train Set accuracy: [76.91579047] without Regularization

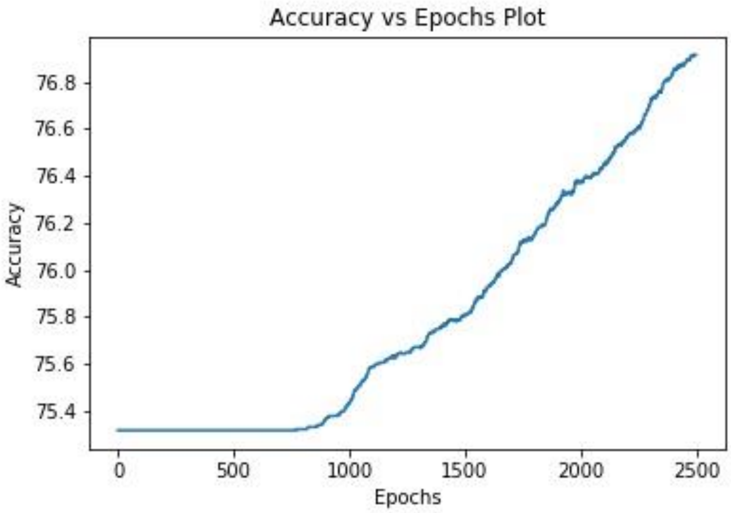
Validation Set Accuracy: [76.44783378] without Regularization

Test Set Accuracy: [77.13811421] without Regularization

- ***Error vs Iteration Graph***



- ***Accuracy vs Iteration Graph***



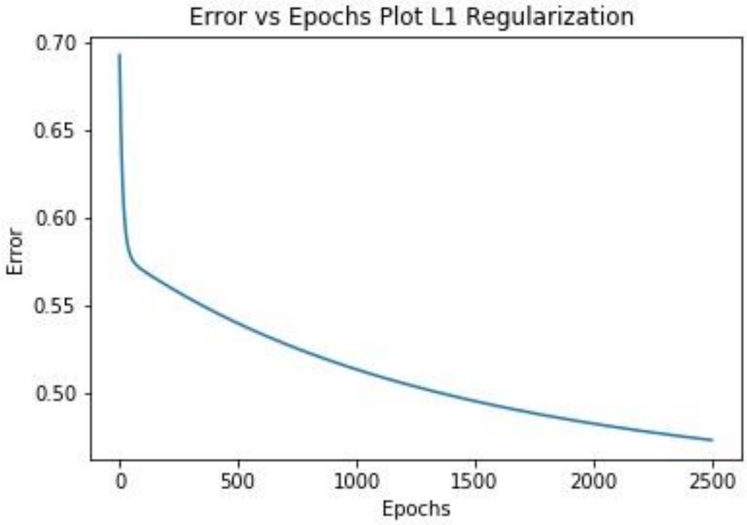
- **L1 Regularization:**

Train Set accuracy: [77.01051435] using L1 Regularization

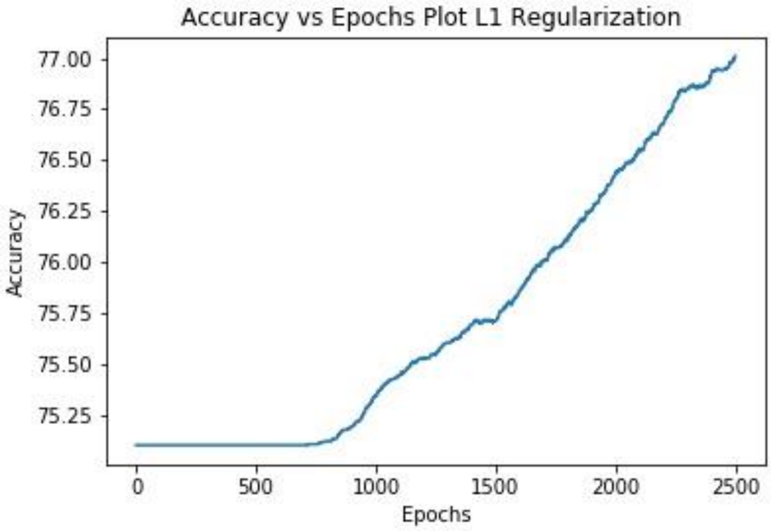
Validation Set Accuracy: [76.34836428] using L1 Regularization

Test Set Accuracy: [77.21779548] using L1 Regularization

- *Error vs Iteration Graph*



- *Accuracy vs Iteration Graph*



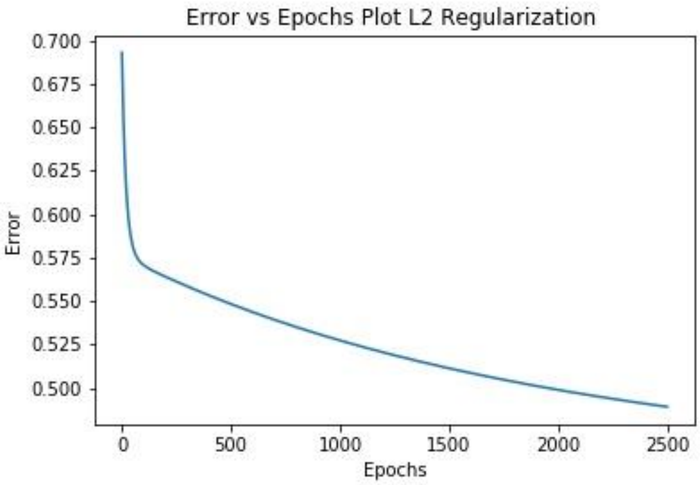
- **L2 Regularization:**

Train Set accuracy: [76.20062518] using L2 Regularization

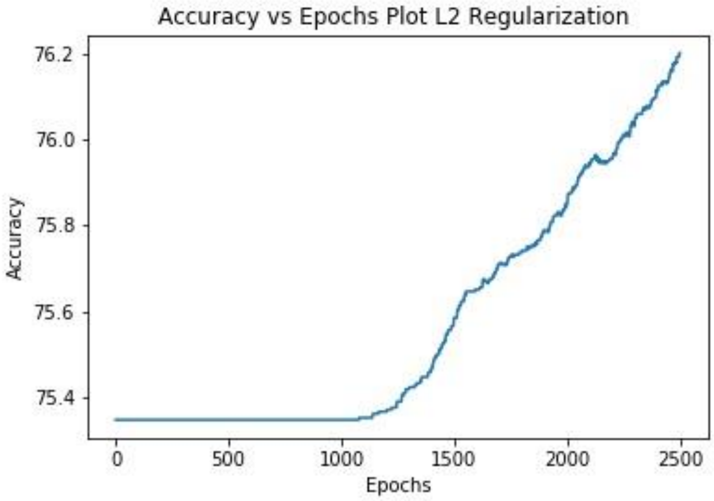
Validation Set Accuracy: [75.30946065] using L2 Regularization

Test Set Accuracy: [76.43426295] using L2 Regularization

- *Error vs Iteration Graph*



- *Accuracy vs Iteration Graph*



Part ii) Regularization on MNIST Dataset with One V/S Rest Approach

Note: Observations are taken on first 10000 samples of both training and testing because of CPU Limitations.

- **L1 Regularization Accuracy of 10 Classes**
- *Train Set Accuracy Score-L1 Regularized:* ***0.9849***
- Accuracy of each class in Train Set in table listed below:

Class Label	Accuracy
Label 0	1.0
Label 1	1.0
Label 2	0.9767911200807265
Label 3	0.9641472868217055
Label 4	0.9979591836734694
Label 5	0.9733487833140209
Label 6	0.9990138067061144
Label 7	0.9981308411214953
Label 8	0.9597457627118644
Label 9	0.9723926380368099

- *Test Set Accuracy Score-L1 Regularized:* ***0.8696***
- Accuracy of each class in Test Set in table listed below:

Class Label	Accuracy
Label 0	0.9551020408163265
Label 1	0.9647577092511013
Label 2	0.812015503875969
Label 3	0.8425742574257425
Label 4	0.8808553971486762
Label 5	0.8038116591928252
Label 6	0.906054279749478
Label 7	0.882295719844358
Label 8	0.8223819301848049
Label 9	0.8146679881070367

- **L2 Regularization Accuracy of 10 Classes**
  - *Train Set Accuracy Score-L2 Regularized****: 0.9852***
  - Accuracy of each class in Train Set

Class Label	Accuracy
Label 0	1.0
Label 1	1.0
Label 2	0.9778002018163471
Label 3	0.9631782945736435
Label 4	1.0
Label 5	0.9698725376593279
Label 6	1.0
Label 7	1.0
Label 8	0.9608050847457628
Label 9	0.9744376278118609

- *Test Set Accuracy Score-L2 Regularized:* ***0.8606***
- Accuracy of each class in Test Set

Class Label	Accuracy
Label 0	0.9459183673469388
Label 1	0.9621145374449339
Label 2	0.7945736434108527
Label 3	0.8445544554455445
Label 4	0.8645621181262729
Label 5	0.797085201793722
Label 6	0.8966597077244259

Label 7	0.8696498054474708
Label 8	0.8172484599589322
Label 9	0.7978196233894945

Since, the difference between Train and Test set accuracies is not very much, it is a good fit.

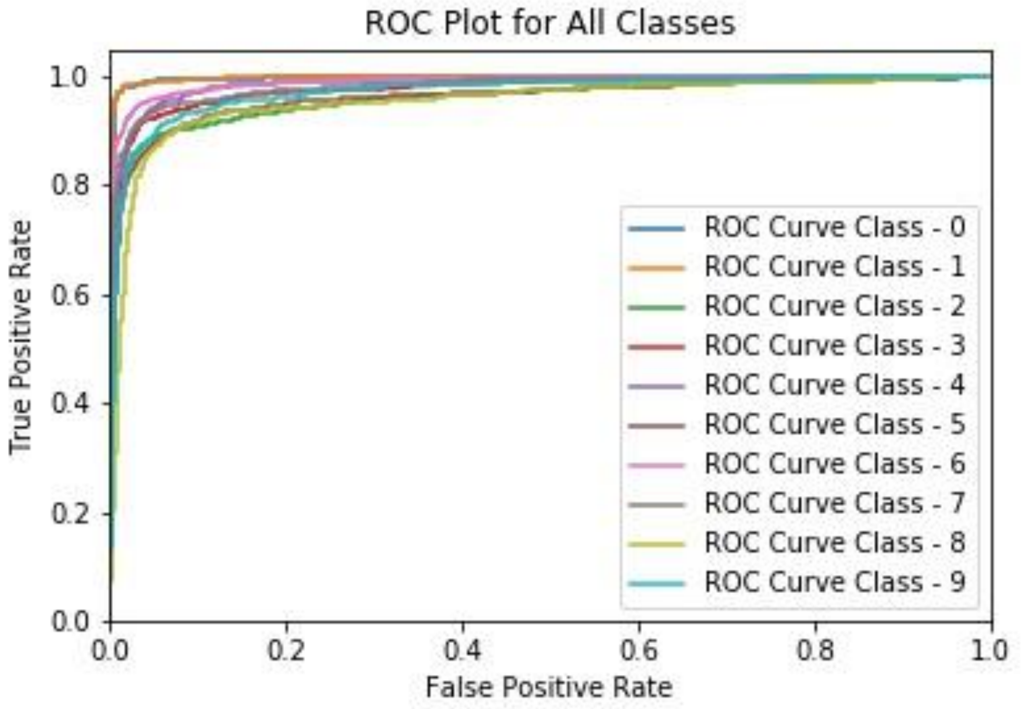
Part iii) ROC Plot for All 10 Classes.

Note: Sklearn library used to plot ROC curve.

True Positive rate: (TP)/(TP + FN)

False Positive rate: (FP)/(TN + FP)

- The lines get smoother as the number of training and test examples are increased. Plots were more distorted when trained using 100 samples. It gets smoother as the number of samples increased to 1000(As shown below).
- In Test Set, Accuracy of Class Label 1, i.e. digits that are 1 is the highest in both L1 and L2 Regularization. This can be verified in the ROC Curve also as the curve for Class Label 1 is the best as compared to other classes.



References:

1) Normal Equation: http://cs229.stanford.edu/notes/cs229-notes1.pdf
2) Ridge, Lasso, GridSearchCV: https://scikit-learn.org/stable/
3) Normalization of Features: https://medium.com/@rrfd/standardize-or-normalize-examples-in-python-e3f174b65dfc