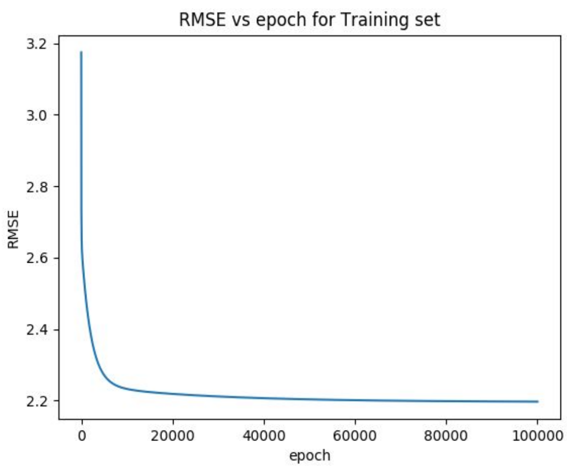
Machine Learning

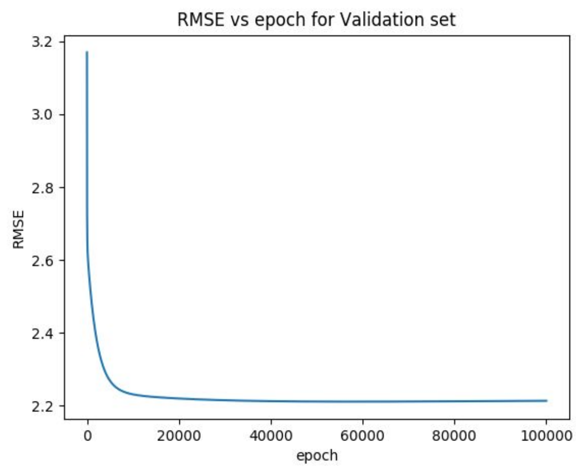
Assignment 1

Question1

For Q1(a)

I have used random.shuffle(f1) to shuffle the data first. Then I have encoded the label 'sex' with I = 0, F = 1 and M=1. Then I have normalized the data. After this, the data has been divided into 5 parts to implementing the 5-fold validation. The learning rate has been set to 0.1. For the five-folds, average RMSE has been calculated for each epoch and the mean of the values have been plotted as below:





For fold 1:
RMSE after training (gradient descent) = 2.2068534706447704
RMSE after running on validation (gradient descent) = 2.1739284571294846
final RMSE cost after training (Normal Eqn) = 2.206015227488462
final RMSE after running on the validation set (Normal Eqn)= 2.1680210454323308

For fold 2:
RMSE after training (gradient descent) = 2.199829236071652
RMSE after running on validation (gradient descent) = 2.2029706616084006
final RMSE cost after training (Normal Eqn) = 2.1988178006033197
final RMSE after running on the validation set (Normal Eqn)= 2.1979225048770332

For fold 3:
RMSE after training (gradient descent) = 2.2070751337933587
RMSE after running on validation (gradient descent) = 2.1658035817025656
final RMSE cost after training (Normal Eqn) = 2.2057209556815263
final RMSE after running on the validation set (Normal Eqn)= 2.1611871529033187

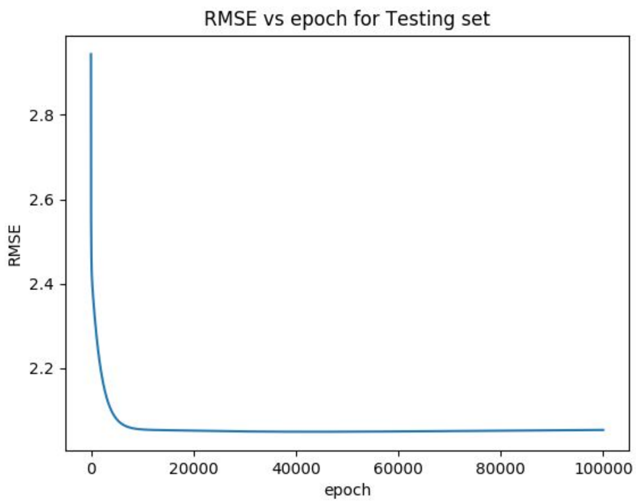
For fold 4:
RMSE after training (gradient descent) = 2.194998203590869
RMSE after running on validation (gradient descent) = 2.2141750129276034
final RMSE cost after training (Normal Eqn) = 2.1938133813964207
final RMSE after running on the validation set (Normal Eqn)= 2.214565158528943

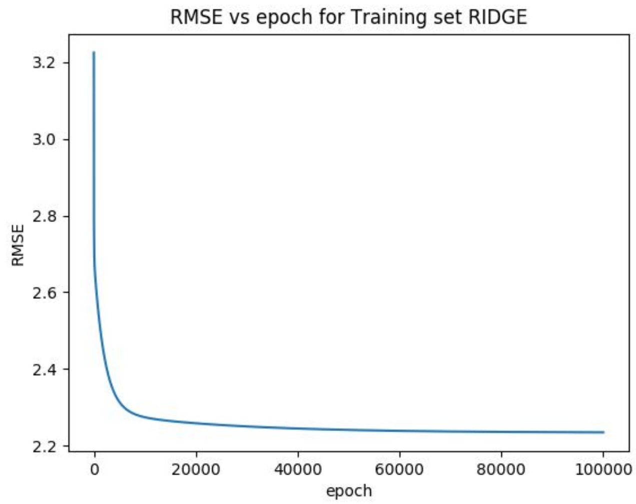
For fold 5:
RMSE after training (gradient descent) = 2.1705952524126007
RMSE after running on validation (gradient descent) = 2.3607399536997993
final RMSE cost after training (Normal Eqn) = 2.1685431664068915
final RMSE after running on the validation set (Normal Eqn)= 2.40526646507946

The final mean RMSE for the Training set was calculated to be 2.1958702593026502 and mean RMSE for the testing set was calculated to be 2.2235235334135703. The normal equation and its corresponding RMSE are also calculated for each of the folds and are printed while running the python file. The normal equation RMSEs are very close to the calculated ones. The RMSE values obtained with all the folds were almost equal for both the Gradient
Descent algorithm and the Normal Equation.

Question1 Part B

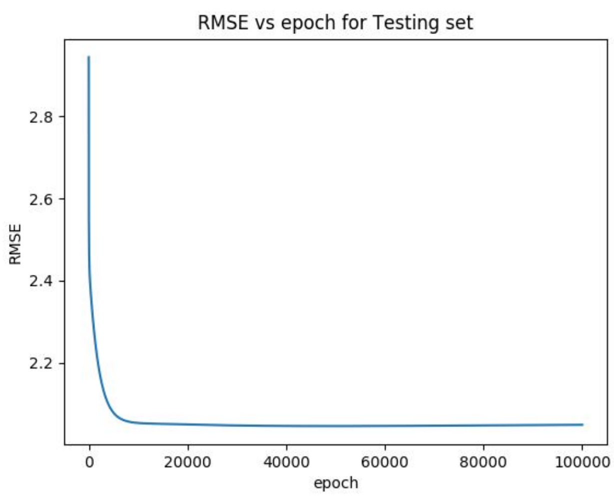
RIDGE
The hyperparameter for L2 Ridge is coming out to be 0.44019351852088745. Using Gradient descent with L2 regularization, 2.2078836650244607 is RMSE for the Training set and 2.1789249695444264 is RMSE for the Testing set Regression. (These might change as data is randomized)

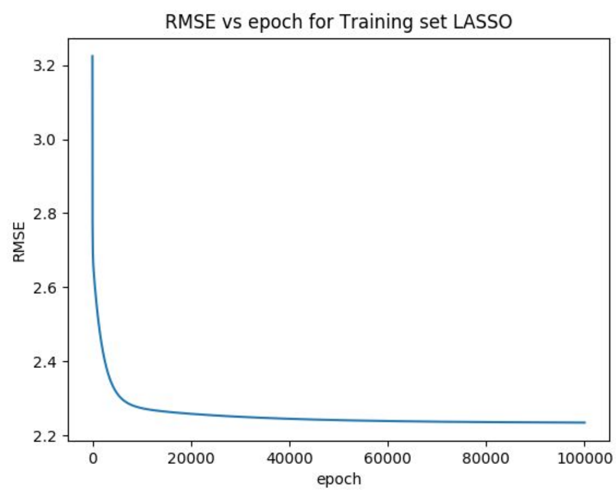




LASSO

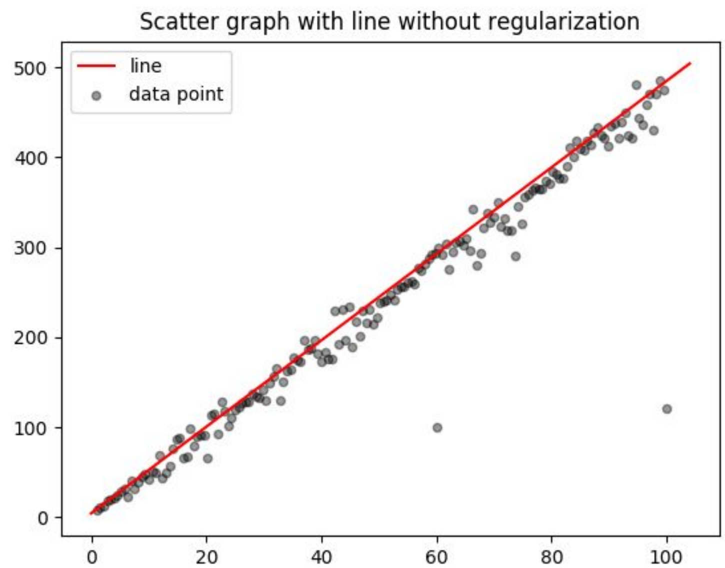
The hyperparameter for L1 Lasso is coming out to be 0.002245697995539774. Using Gradient descent with L1 regularization, 2.2068519161941795 is RMSE for the Training set and 2.1739361122427225 is RMSE for the Testing set Regression. (These might change as data is randomized)



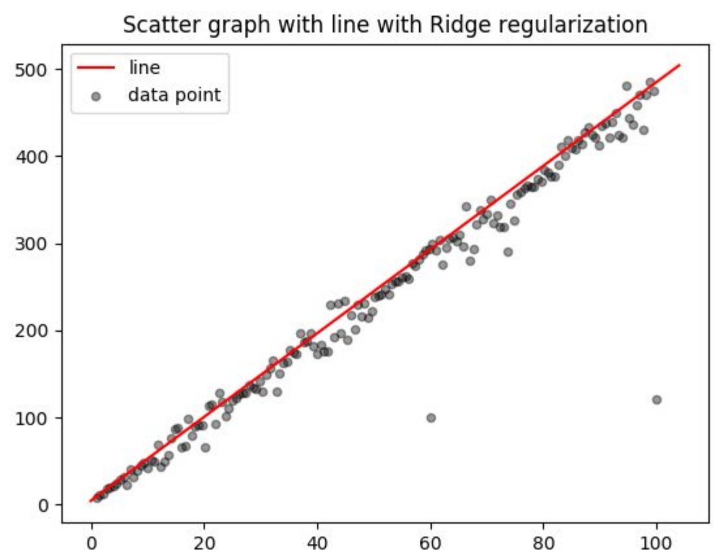


Question1 Part C

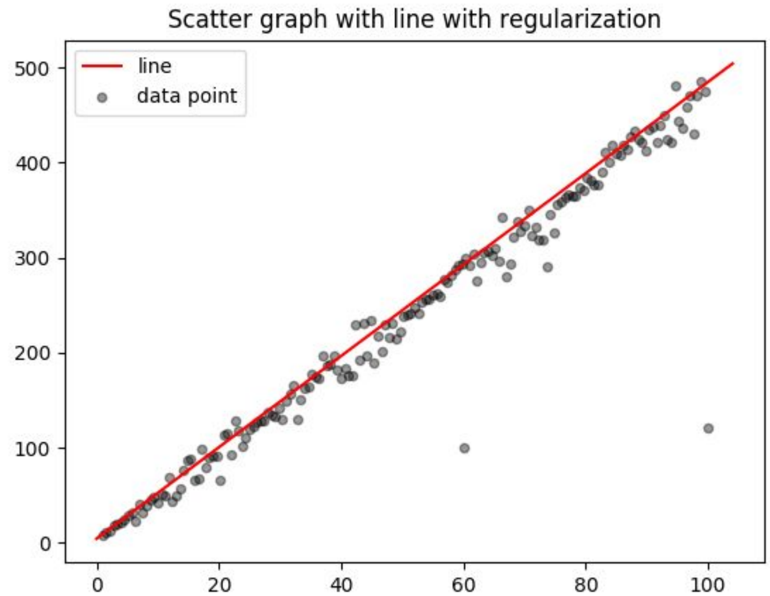
RMSE for normal gradient descent (best fit line Question) = 32.81660104055684. Scatter plot for no regularization:



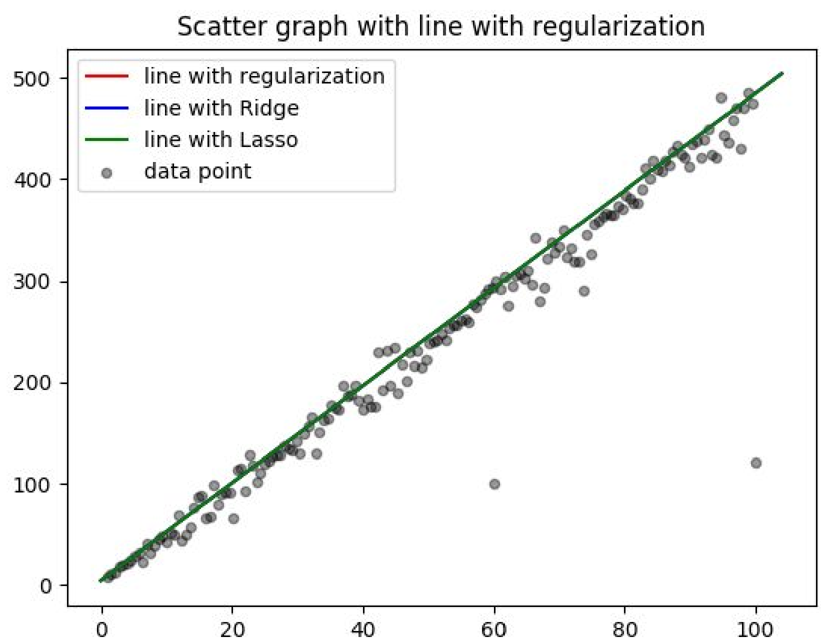
RMSE for L2 gradient descent (best fit line Question) = 32.81660104055683. Scatter plot for Ridge regularization:



RMSE for L1 gradient descent (best-fit line Question) =32.81660104055684. Scatter plot for Lasso regularization:



Scatter plot for with all three lines:



With regularization, there is a slight change in the theta which is not visually different but the RMSE on the training set increases a little but the testing RMSE reduces slightly. It is almost the same fit. This is because even without regularization the output line is a good fit because the data doesn’t contain noise nor has outliner points which increase the error for RMSE.

Question2

Question2 Part A

Complete train data is used.

Without regularization:

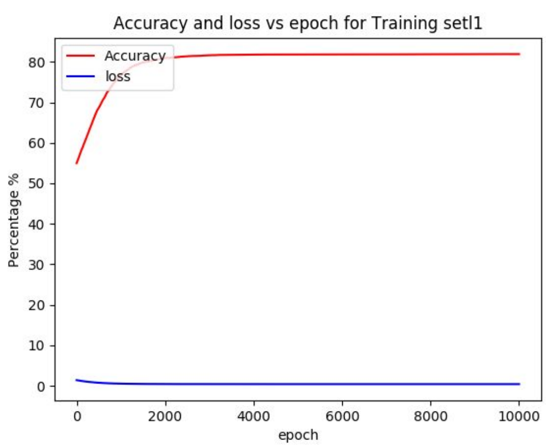
accuracy on training set = 81.91432928850872

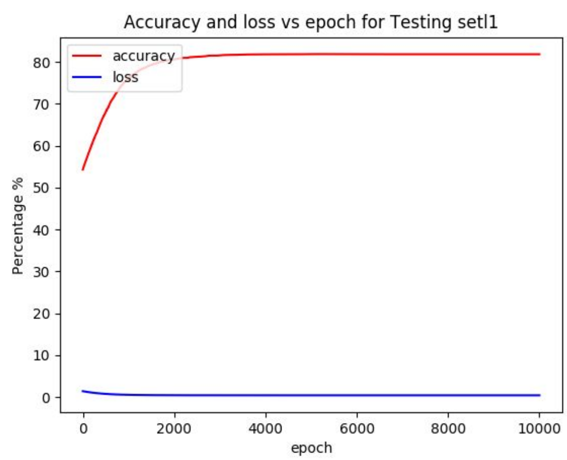
accuracy on test set = 81.81274900398407

With L1 regularization:

accuracy on training set = 81.91432928850872

accuracy on test set = 81.81274900398407

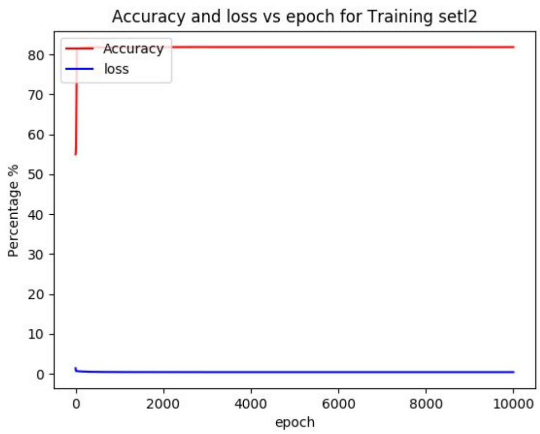


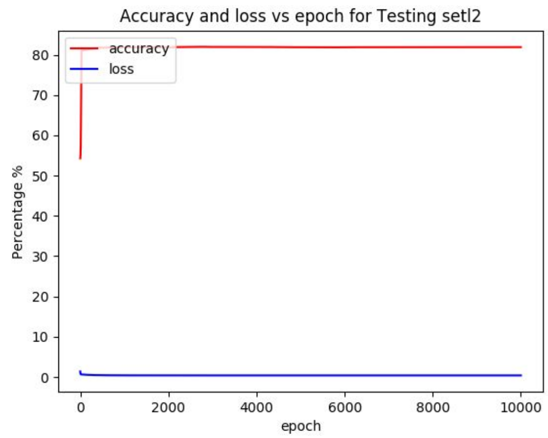


With L2 regularization:

accuracy on training set = 81.91432928850872

accuracy on test set = 81.81274900398407





Question2 Part B

For one vs Rest

for class 0:

Accuracy for class 0 with L1 regularization (on training set) = 99.23666666666666

Accuracy for class 0 with L1 regularization (on test set) = 99.2

Accuracy for class 0 with L2 regularization (on training set) = 99.17833333333334

Accuracy for class 0 with L2 regularization (on test set) = 99.16

for class 1:

Accuracy for class 1 with L1 regularization (on training set) = 99.19833333333334

Accuracy for class 1 with L1 regularization (on test set) = 99.39

Accuracy for class 1 with L2 regularization (on training set) = 99.14333333333335

Accuracy for class 1 with L2 regularization (on test set) = 99.33999999999999

for class 2:

Accuracy for class 2 with L1 regularization (on training set) = 98.08833333333334

Accuracy for class 2 with L1 regularization (on test set) = 98.06

Accuracy for class 2 with L2 regularization (on training set) = 97.91333333333333

Accuracy for class 2 with L2 regularization (on test set) = 98.00999999999999

for class 3:

Accuracy for class 3 with L1 regularization (on training set) = 97.715

Accuracy for class 3 with L1 regularization (on test set) = 97.98

Accuracy for class 3 with L2 regularization (on training set) = 97.53166666666667

Accuracy for class 3 with L2 regularization (on test set) = 97.89999999999999

for class 4:

Accuracy for class 4 with L1 regularization (on training set) = 98.37333333333333

Accuracy for class 4 with L1 regularization (on test set) = 98.24000000000001

Accuracy for class 4 with L2 regularization (on training set) = 98.24833333333333

Accuracy for class 4 with L2 regularization (on test set) = 98.2

for class 5:

Accuracy for class 5 with L1 regularization (on training set) = 97.59

Accuracy for class 5 with L1 regularization (on test set) = 97.75

Accuracy for class 5 with L2 regularization (on training set) = 97.02666666666667

Accuracy for class 5 with L2 regularization (on test set) = 97.31

for class 6:

Accuracy for class 6 with L1 regularization (on training set) = 98.82166666666666

Accuracy for class 6 with L1 regularization (on test set) = 98.7

Accuracy for class 6 with L2 regularization (on training set) = 98.73166666666665

Accuracy for class 6 with L2 regularization (on test set) = 98.7

for class 7:

Accuracy for class 7 with L1 regularization (on training set) = 98.53166666666667

Accuracy for class 7 with L1 regularization (on test set) = 98.52

Accuracy for class 7 with L2 regularization (on training set) = 98.41833333333334

Accuracy for class 7 with L2 regularization (on test set) = 98.5

for class 8:

Accuracy for class 8 with L1 regularization (on training set) = 96.32833333333333

Accuracy for class 8 with L1 regularization (on test set) = 96.35000000000001

Accuracy for class 8 with L2 regularization (on training set) = 96.22166666666668

Accuracy for class 8 with L2 regularization (on test set) = 96.41999999999999

for class 9:

Accuracy for class 9 with L1 regularization (on training set) = 96.67666666666666

Accuracy for class 9 with L1 regularization (on test set) = 96.74000000000001

Accuracy for class 9 with L2 regularization (on training set) = 96.60833333333333

Accuracy for class 9 with L2 regularization (on test set) = 96.63000000000001

It is a good fit as the accuracy on the test set is also as good as training set.

For Train set:

L1 Regularization score of normal logistic regression = (without OneVsRest method) : 92.06166666666667

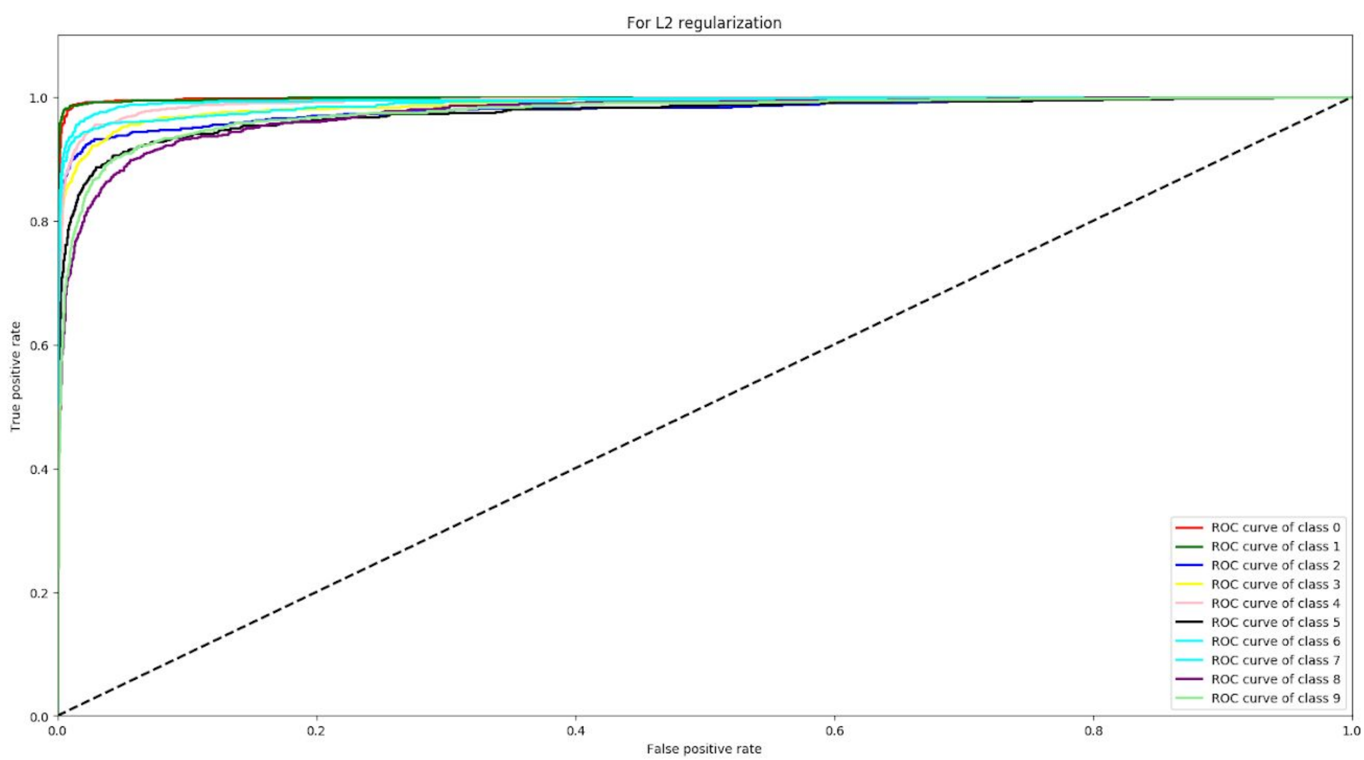
L2 Regularization score of normal logistic regression = (without OneVsRest method): 91.31833333333333

For Test set:

L1 Regularization score of normal logistic regression = (without OneVsRest method): 92.12

L2 Regularization score of normal logistic regression = (without OneVsRest method): 91.56

Question2 Part B



Aayush Gupta 2017125

Page no.

Machine Learning Assignment 1

(3) (i) Given, $P(y=1|x,w) = g(w_0 + w_1 x)$
where $g(z)$ is a logistic function.
Also, it is an increasing function
i.e. $g(z) = \dfrac{1}{1+e^{-z}}$

As it is a function of x
$w_0 + w_1 x$ acts as a linear equation, so it extends from $-\infty$ to $+\infty$ when ~~x~~ extends from $-\infty$ to $+\infty$ and $w_1 \neq 0$

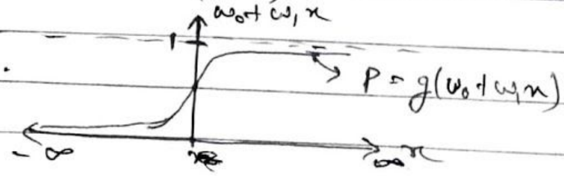
So $g(z) = \dfrac{1}{1+e^{-z}}$

$-\infty < z = w_0 + w_1 x < \infty$

$g(z)$ at $z = -\infty = \dfrac{1}{1+e^{\infty}} \Rightarrow 0$

$g(z)$ at $z = \infty = \dfrac{1}{1+e^{-\infty}} \rightarrow 1$

So, for ~~the~~ range of $P(y=1|x,w)$ is $(0,1)$



(ii) logit function : $l(x) = \log\left(\frac{x}{1-x}\right)$

$$\frac{d}{dx} l(x) = \left(\frac{1-x}{x}\right) \frac{(1-x) - x(-1)}{(1-x)^2}$$

$$= \frac{1}{x(1-x)}$$

which is always positive for $0<x<1$
Hence, it is an increasing function.
For 0^+ & 1^-, logit tends to $-\infty$ and $+\infty$ respectively.

So, logit goes from $(-\infty$ to $\infty)$ for $0<x<1$

(4) (a) As, RMSE takes the square of error, hence the higher errors are given more weights and are ~~easier to~~ easier to reduce using the gradient descent as higher errors are undesirable and reduced significantly due to higher weights.

(b) In a data, where noise is very far away from the actual data, RMSE would give higher weight to reduce the loss, which is infact not good for the performance of the actual data

RMSE would increase the error in the actual data. In such cases MAE is considered as a better option, as it is more robust to the noise and focuses more on actual data.

(C) Quantile loss function is more useful over MAE when we need to predict intervals rather than discrete points which have variable variance. It is basically a modification of MAE

$$L_\gamma(y, y^p) = \sum_{i=y_i<y_i^p} (\gamma-1).|y_i - y_i^p| + \sum_{i=y_i \geq y_i^p} \gamma |y_i - y_i^p|$$

where γ is a parameter which is called quantile. By changing it we can adjust the weight of errors which can help us to adjust for amount of overestimation or underestimation. For $\gamma = 0.5$, it turns into MAE.