ML - Assignment - I

Problem 4

a) We can split the data using K-Fold and analyse the data by getting MSE of training folds and MSE of test-folds.
The mean of MSE training folds and MSE mean of test-folds can be used to get learning curves.
We use validation_curve() function of sklearn to do the same.

validation_curve(Rdmodel,
trainingdata,
Testingdata,
paramname,
paramrange,
cross validation
folds,
scoringmethod,
jobs=value)

This method returns train_scores and test_scores which can be plotted using mean of test scores, mean of train scores along with param_range of penalty parameter 'α'.

The learning curves show the mean of MSE of training folds and Mean of MSE of test data increase as the penalty parameter increases.

Optimal value of penalty parameter can be found using RandomizedSearchCV method which gives best_estimator_.alpha for the model by iterating over given range of values.

We get α = 0.000258331735,
MSE = 5.680092364.
which is best value for ranges of α values.

b.

We standardize the input co-variates using StandardScalar() method in scikit learn library using

StandardScalar().fit(j).transform(j)

where 'j' is a reshaped array of column of co-variate values.

We once draw resulting learning curves for the standardized data.

The training data Mean Squared Error curve and testing data MSE curve is set to be closer than in non-standardized learning curve.

The curve after standardization is smoother.

Training score and test scores are set to lesser in standardized version than in the previous learning curve across the range from start to end.

c. Standardization of the data was done in order to decrease the Mean squared error. Hence we consider the

MSE error at beginning and end for lesser penalty parameter to higher penalty parameter for un-standardized Initial data based model and model based on Standardized data.

Non-Standardized based Model

Training data Curve (At beginning)

$x = 50$

$y = -15.98$

Test data Curve

$x = 50$

$y = -27.24$

Training data Curve (At End)

$x = 200$

$y = -25.66$

Test data Curve

$x = 200$

$y = -33.99$

Standardized data based Model

Training data Curve (At beginning)

$x = 50$
$y = -12.00$

Test data Curve

$x = 50$
$y = -20.67$

Training data Curve (At End)

$x = 200$
$y = -23.29$

Test data Curve

$x = 200$
$y = -29.71$

We consider ratios of both Model MSE (y) values

At beginning (approximate), $x = 50$.

$$\text{Ratio}_{\text{tr data MSE}} = \left(\frac{15.96}{12.0}\right) = 1.325$$

$$\text{Ratio}_{\text{test data MSE}} = \left(\frac{27.24}{20.67}\right) = 1.31785$$

At End, $x = 200$

$$\text{Ratio}_{\text{tr data MSE}} = \left(\frac{25.66}{23.29}\right) = 1.1017$$

$$\text{Ratio}_{\text{test data MSE}} = \frac{33.99}{29.71} = 1.1417$$

We can notice that Ratio at beginning point is higher than Ratio at a particular End Point. Hence, standardization has a better or bigger difference in decrease in MSE at begining of data that is for lower, smaller penalty parameter values.

d. No. We do not enough data.
Because both of the learning curves never converge across range of penalty parameters. Hence we are not able to get optimal value of penalty parameter 'α' from the graph.

e. The model performs best for α = 0.0002833735 penalty parameter for ridge regression.

MSE_{test} = 15.78987522 45

The model was trained on training data with MSE on training data MSE_{tr} = 5.68 00923

May be the model has got overfitting with training data hence performs lesser in case of test data.

MSE_{test} = 15.78987 5 not bad considering we have small input data which might lead to overfitting of model.

Problem-2

a.

In the problem 2) a), we use np.random.gamma(1, 1, 1000) to generate 1000 samples from Gamma(1, 1) function

Using the list of samples of function Gamma or Xi distribution of Gamma (1, 1) function; we get distribution Zi by simply substitute Xi values in the function or equation $Z = \frac{x^2}{1+\sin(x)}$.

After getting distribution Zi of 1000 samples of equation Z, we plot the histogram of 300 sample to get the pdf of 'z' plot.

b. $E[\omega]$ or Expectation of $E[\omega]$ can be found in the following way.

Expectation of a list of N samples is same as the mean of N samples.

In our case, we have a continous function. 'ω'. We have to derive a 'N' samples distribution of the function and take the mean of the 'N' samples so derived.

Here, we choose value of 'z' to be considered in the function to be 0 to infinity as for negative values 'z', $\log z$ has considered which does not exist as log of negative number is not defined.

Here, we have consider a random number from 0 to $1.79 \times e^{308}$ as the biggest number allowed in python is $1.8 \times e^{308}$.

We can consider a sample of W to be a sample 10000000 numbers

We get mean of 10000000 numbers of samples of distribution of W.

Mean W = 708.7778695

We get the mean of distribution which is equal to the Expectation of W.

Problem 3

To build the model, we have to replace NA values in training data, We replace the 'NA' values in training data using simple imputer from sklearn library.

Our strategy is to replace missing value with mean of the rest of the values.

The model is not the best model
Hence again, we use the Randomized
Search CV() function to get best
penalty parameter α value for the
model. We get $\alpha = 9.29330653033$

we use this best model on
test data after removing 'NA' values
in test data using Simple Imputer function of
sklearn. MSE training data = 0.16324625