Introduction to Machine Learning

Assignment 1

Azure note book link –

https://omkarpandit-oap338.notebooks.azure.com/j/notebooks/Omkar_Pandit_Lab_Assignment1.ipynb

Q1. Synthetic Data: Comparative Analysis & Learning Curves [40 points]

a. Generate a dataset by sampling 1,200 values from a cubic function with noise added. Set values to range from 0 to 1. (Code)
b. Create a 70/30 train/test split of the dataset. (Code).
   ➔ Number samples in training set - 840
   Number samples in testing set - 360

c. Plot learning curves for the following two regression models: linear and polynomial (to the 4th degree). Vary the amount of training data by increments of at most 120 (i.e., at least 10 training sizes), evaluate using mean squared error, and remember to show curves for both the training and test datasets. (Code and Write-up).

   Train MSE vs Test MSE for linear model

No of Samples	Train MSE	Test MSE
84	1.964919	7.446621
168	2.062045	4.933331
252	2.069579	4.090214
336	2.068069	3.670778
420	2.060514	3.419923
504	2.050646	3.252407
588	2.043678	3.132521
672	2.036990	3.041971
756	2.030923	2.971699
840	2.028069	2.915411

   ➔ After applying linear model/ linear regression on different number of samples, it is observed that, for linear model, value of ‘Train MSE’ initially increased with addition of 84 samples to the training data. But after further additions of training samples it gradually decreased. In case of ‘Test MSE’ value keep on decreasing with more additional samples.

Train MSE vs Test MSE for polynomial model

No of Samples	Train MSE	Test MSE
84	0.802286	2.492431
168	0.832247	2.007762
252	0.855763	1.755440
336	0.877052	1.600208
420	0.893123	1.495221
504	0.904162	1.419685
588	0.911471	1.362844
672	0.916503	1.318645
756	0.920591	1.283329
840	0.923927	1.254341

➔ On applying polynomial regression model (with degree '4') it is observed that, value of 'Train MSE' keep on increasing with more and more addition of samples. But after certain point value of 'Train MSE' may stay constant or may decrease with addition of more samples to the training space. In case of 'Test MSE' value keep on decreasing with more additional samples.

d. Write a discussion analyzing and comparing the two models. Explain which model performs better and why. Also, address which models you think are under_tting versus overfitting and explain why. Final, discuss the impact of increasing the amount of training data on both models. Your discussion should consist of two to three paragraphs. (Write-up).

➔ In both the cases, MSE values keep on decreasing with addition of samples to training space. But if we compare both the models i.e. Linear model vs Polynomial model, For linear model, currently training and testing MSE's are decreasing with increase in sample space whereas in polynomial model even though test MSE is decreasing, train MSE is gradually increasing with increase in sample size. So, as per my observations Linear model is better than polynomial model.

As per current observations, for linear model if we keep on adding more samples to the training space then there might be a situation at which 'Test MSE' start increasing after a certain drop and 'Train MSE' keep on decreasing as it is doing right now. From this position, our linear model may fall into a case of overfitting.

Same thing with polynomial model as well, after certain point when 'Test MSE' starts increasing and 'Train MSE' keep on decreasing, model may fall into overfitting case.

Effect of increasing the amount of training data – Definitely, by providing more and more data points to the model helps in improving its understanding. So, currently in both the cases, with increasing amount of training data, both 'Test MSE' and 'Train MSE' values are decreasing.

Q2. Real Data: Comparative Analysis & Feature Analysis [40 points]

a. Load a real dataset not covered in class that is designed for the regression problem; e.g., from sklearn. datasets, Kaggle, your own data, etc. (Code)

➔ Dataset used – sklearn diabetes dataset

b. Create a 70/30 train/test split of the dataset. (Code)
➔ Number of samples in training set - 309
Number of samples in testing set - 133

c. Train and evaluate the predictive performance for each of the following regression models: linear, ridge, lasso, and polynomial (to the 4th degree). Evaluate using mean squared error. Report all values in a single table. (Code and Write-up)

Model	Test MSE
Linear	2821.738559584376
Polynomial	203103.00528653633
Ridge	3030.9743277657362
Lasso	3089.6371118751645

➔ So, after evaluating mean squared error for all four models, it is observed that linear model has a least mean squared error. So, in my case linear regression model is the one who is performing better than rest of the implemented models. But point to highlight is as ridge and lasso are more flexible models, if we set appropriate values for alpha, ridge and lasso may give better results than linear and polynomial models.

d. For the top-performing regression model, plot the learned feature coefficients. (Code and Write-up).

Feature name	Feature coefficients
Age	29.25034582
Sex	-261.70768053
BMI	546.29737263
Bp	388.40077257
S1	-901.95338706
S2	506.761149
S3	121.14845948
S4	288.02932495
S5	659.27133846
S6	41.37536901

e. Write a discussion analyzing and comparing the four models. Explain which model(s) perform the best/worst and why you think this occurs. Also discuss which features are most predictive for the top-performing model and why this may be. Your discussion should consist of two to three paragraphs. (Write-up).

➔ After comparing 'Test MSE' values all four models, in my case, best model with least 'Test MSE' score is a linear model and worst model with highest 'Test MSE' score is a polynomial model.

'Test MSE' defines mean squared error generated while using test data on your built model. So, 'Test MSE' is a very good indicator of defining a accuracy of a model. So, lower the test MSE value better the model is. So, I think in my case linear model is best while polynomial model is performing worst.

Most impacting features are the ones who have large feature coefficient values. So, for this best performing linear model as per the feature coefficient table (listed above), top 3 most predictive features are,

1. S5
2. BMI
3. S2

Q3. Real Data: Tuning Hyperparameters for Regularized Models [20 points].

a. Use the same training and test datasets from step 2c.
b. Plot performance curves for the following regression models when varying : ridge and lasso. Vary to have at least 10 values and evaluate using mean squared error. (Code and Write-up)
   ➔ I vary the value of alpha from 0.1 to 1 with an increment of 0.1.

Ridge regression model

Value of alpha	Mean squared error
0.1	2805.393845841174
0.2	2813.1519157286784
0.3	2837.3163558777396
0.4	2870.3051032099015
0.5	2908.035326830526
0.6	2948.2041735429375
0.7	2989.457424397802
0.8	3030.9743277657362
0.9	3072.2477071208145
1.0	3112.961424786802

Lasso regression model

Value of alpha	Mean squared error
0.1	2775.1600440020425
0.2	2806.1046829474594
0.3	2845.650183691481
0.4	2889.192438523361
0.5	2936.3282171871033
0.6	3003.147080461654
0.7	3089.6371118751645
0.8	3195.7989372117427
0.9	3316.3449484859043
1.0	3444.667115975281

c. Write a discussion analyzing and comparing the two types of regression models. Explain the impact of the _ parameter. Also, discuss which model performs the better and why you think this occurs. The discussion should consist of two to three paragraphs. (Write-up).

➔ In my case, for smallest value of alpha i.e. when alpha = 0.1, lasso model is performing better as mean squared error in that case is least (mse ~ 2775) while for ridge it is around 2805. But one point to notice, ridge regression results are more consistent i.e. not varying much as compare to the variation in MSE observed in cases of lasso regression.

Impact of alpha parameter – In my case alpha is working as a penalty factor. So, as we increase the value of alpha with that mean squared value is also increasing. So as per the facts when alpha is zero, it will work as linear regression model.

As I discussed earlier, ridge and lasso are more flexible models as compare to linear and polynomial models. So, in my case, when I set alpha = 0.1 or 0.2, for both ridge and lasso I am getting better results by achieving least mean squared values that what I got in case of linear and polynomial models.