Name: Chandramohan Natarajan

Student number: 190617866

Assignment number: 1

Module: ECS766

Exercise 0:

Which polynomial has the lowest `train` MSE? Which one has the lowest `test` MSE? [0.5 mark]

Polynomial with lowest train MSE → 9

Polynomial with lowest test MSE → 3

Exercise 1:

What trend do you observe when you analyse the dependence of train and test MSE on the polynomial order? First describe the observed trends, and then explain them. [1 mark]

Dependence of train MSE on Polynomial order:

There is a gradual decrease of MSE for orders 1 and 2. But drop in MSE is high as the polynomial order used is 3. Then MSE gradually decrease with increase in polynomial order and becomes zero at order 9.

Bias looks large for polynomial order 1 and 2. Underfitting can be observed.

Curves are overfitting for orders 8 and 9.

Data fit is good for polynomial order 3.

Dependence of test MSE on Polynomial order:

MSE slightly increase from polynomial order value 1 and 2. There is dip in MSE for order 3 till polynomial order 5. When order 6 is used , MSE increases gradually till polynomial order 8 . Then there is a steep increase from Polynomial order 9.

Polynomial curve looks terrible for higher polynomial order

Exercise 2:

Identify the models that are suffering from <u>under-fitting</u> and the ones suffering from <u>over-fitting</u>. Justify your choice based on your observations and use the theory that we have learnt to explain it. [0.5 mark]

Models suffering from underfitting: Polynomial order 1 and 2

Those models neither fit training data nor generalize test data. The possibility of making bad predictions is high.

Models suffering over fitting: 9

Because the model gives high variance, Bias looks too low. Possibiliyty of noise is imminent from the performance of model.

Exercise 3:

Which model would you pick as the best one amongst these 9 models? What are the parameter(s) and hyper-parameter(s) of your chosen model? [0.5 mark]

The best among the model is one with polynomial 5.

Train MSE - .01324

Test MSE – 0.1899

```
Learned model → 1.245 x^5 + 1.048 x^4 + 0.04529 x^3 - 0.9512 x^2 +
0.1219 x + 1.1
```

Exercise 5:

Focus on order=9. Describe AND explain the trend of each of the metrics below with respect to increasing values of λ (that is, first describe what the effect of increasing λ from zero upward is it on the parameter in question and then explain briefly and clearly the reasons behind it): [1 mark]

(a) **TRAIN MSE**
(b) **(b)** $\vec{w}^T.\vec{w}$
(c) **(c) TEST MSE**

When **λ is zero, there is zero MSE. And very high value for** $\vec{w}^T.\vec{w}$ **and TEST MSE. Once a value is assumed for Lambda, we see a value for Train MSE and Test MSE come down very heavily and overfitting phenomenon decreases. But when lambda is increased further , Test MSE starts increasing with increasing train MSE and decreasing** $\vec{w}^T.\vec{w}$**.**
With further increase in lambda value, Test MSE increase gradually and True model and learned model fit data well till lambda reaches .01.

When when lambda is further increase to .1, learnt model gets overfitting phenomenon after moving in X parameter.
The explanation behind these scenario is, we use regularization to fight overfitting by controlling the value of coefficients by adding a factor called lambda. We optimize the equation to get solution which gives us prediction model.
The model becomes very rigid when the coefficients are conrlled close to nill.
We may restrict the flexibility of model by controlling coefficients. We do trade off between MSE and coefficients, by managing lambda. When lambda is high the predictin is zero which is visible from straight line.
Overfitting phenomenon is controlled by varying lambda and visibly witness the model work with regularized bias and variance,

Exercise 6:

Now suppose that instead of 10 training instances, we had access to 100 train instances. Run the following script and inspect the change in the test error. Describe and explain the effect of having more training data on the test error (test MSE) and over-fitting. [1 mark]

With increasing training data, it has been observed that model has learnt the input . Data seems to fit the model. Bias ands variance are low. Prediction looks very good indeed.

Train MSE displays controlled pattern irrespective of increase in lambda.

Learnt model looks well regularized and the model is highly flexible with low MSE.

Exercise 7:

Which is the correct way to complete the script and calculate the variable SSE? Explain what each of the three options to calculate SSE would do and justify your choice. What is the resulting train and test errors? [0.5 mark]

Script

```
SSE = np.sum(np.power(yTest - np.matmul(phitest, w_map), 2))

print('Lambda: {0:1.5f}, CV SSE: {1:1.5f}.'.format(LAMBDA, SSE), end='')

ERROR_TRAIN = np.matmul(phiTrain, w_ml) - yTrain

ERROR_TEST = np.matmul(phiTest, w_ml) - yTest

print('MSE Train =
{:8.4f}'.format(np.dot(ERROR_TRAIN.T,ERROR_TRAIN)/len(ERROR_TRAIN)))

print('MSE Test =
{:8.4f}'.format(np.dot(ERROR_TEST.T,ERROR_TEST)/len(ERROR_TEST)))Lambda:
10.00000,
```

CV SSE: 0.96854.

MSE Train = 0.4828

MSE Test = 0.4934

Exercise 8:

Compare the training, validation and the test errors in both the linear model and the M5P model. First, explain the differences between the numerical values of each error separately for the linear model and for the M5P model. Then, bearing in mind that the M5P model is more complex than the linear model, explain why the numerical values of the errors seem to behave differently for the linear model and for the M5P model. [1 mark]

Linear regression Model with training set

Correlation coefficient	0.8923
Mean absolute error	4229.4683
Root mean squared error	7357.4334
Relative absolute error	43.9307 %
Root relative squared error	45.1391 %
Total Number of Instances	9080

Linear regression Model with training set – 50% split

Correlation coefficient	0.8734
Mean absolute error	4434.8591
Root mean squared error	8062.8445
Relative absolute error	45.8027 %
Root relative squared error	48.9467 %
Total Number of Instances	4540

M5P model with training set

Correlation coefficient	0.9351
Mean absolute error	2968.1157
Root mean squared error	5776.4447
Relative absolute error	30.8293 %
Root relative squared error	35.4395 %
Total Number of Instances	9080

M5P model with 50% split

Correlation coefficient	0.8929
Mean absolute error	3469.4293
Root mean squared error	7416.4842
Relative absolute error	35.8318 %
Root relative squared error	45.0229 %
Total Number of Instances	4540

Linear regression model takes note of all data in one go and gives us all parameters and values.. Whereas M6P model splits data into multiple regions as it comes under decision tree and process linear regression for all datas, which gives us more accurate error and flexibility.