MACHINE LEARNING

REGRESSION

ACADEMIC YEAR 2023/2024

QUEEN MARY UNIVERSITY OF LONDON

SOLUTIONS

EXERCISE ♯1 (SOL): Using the following definitions:

$$\begin{aligned} \hat{y}_i &= f(x_i) \\ e_i &= y_i - \hat{y}_i \end{aligned}$$

we can obtain for the absolute $|e|$ and squared errors e^2 of the predictions made by $f_1(x)$:

ID	x	y	$\hat{y} = f_1(x)$	e	$\|e\|$	e^2
1	2	1	5	-4	4	16
2	3	2	7	- 5	5	25
3	1	2	3	-1	1	1
4	1	1	3	-2	2	4
5	0	1	1	0	0	0
6	5	3	11	-8	8	64
7	4	3	9	-6	6	36
8	6	7	13	-6	6	36
9	5	6	11	-5	5	25
10	3	5	7	-2	2	4

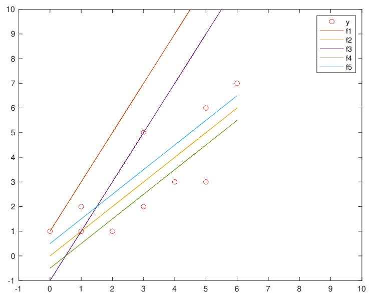
The resulting MAE and MSE values are:

$$\begin{aligned} E_{MAE} &= \frac{1}{N}\sum |e_i| = \frac{1}{10}(4+5+1+2+0+8+6+6+5+2) = 3.9 \\ E_{MSE} &= \frac{1}{N}\sum_{i=1}^{N} e_i^2 = \frac{1}{10}(16+25+1+4+0+64+36+36+25+4) = 21.1 \end{aligned}$$

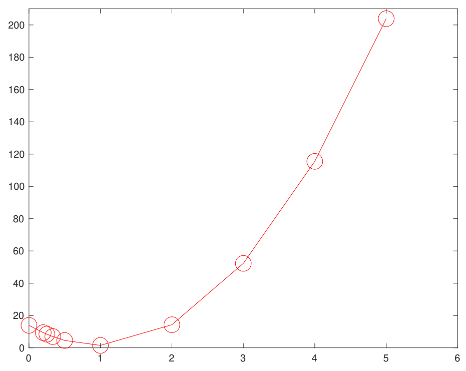
Carrying out the same steps for the remaining 4 models, we obtain the following MAE and MSE values:

Model	MAE	MSE
f_1	3.9	21.1
f_2	1.1	1.5
f_3	2.5	9.5
f_4	1.2	1.85
f_5	1.1	1.65

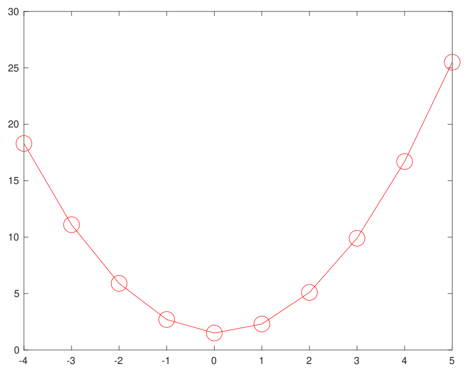
If we use the MAE as our quality metric, we could choose either f_2 or f_5, as they have the lowest MAE value. If we use the MSE, we would choose f_2. For illustrative purposes, the following figure plots each model and the dataset used for testing them:



EXERCISE ♯2 (SOL): For $w_0 = 0$ and w_1 =0, 1/5, 1/4, 1/3, 1/2, 1, 2, 3, 4, 5, plotting the MSE value against w_1 results in:

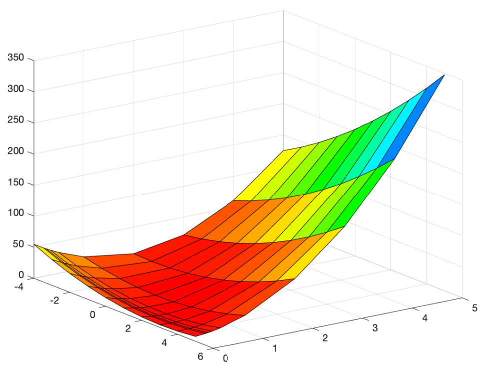


Hence $w_1 = 1$ minimises the MSE (MSE = 1.5). For $w_0 = 1$ and $w_0 =$ -4, -3, -2, -1, 0, 1, 2, 3, 4, 5, we obtain:

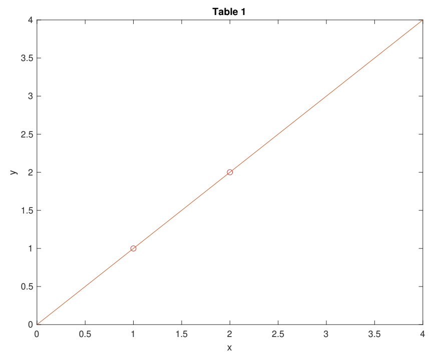


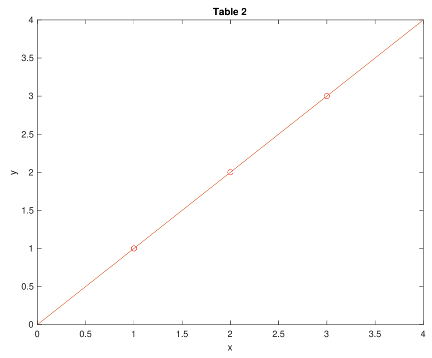
The lowest MSE value is achieved for $w_0 = 0$ (MSE = 1.5). Based on the MSE values along w_0 and w_1, we would sketch a convex MSE surface, whose lowest point is close to the point $w_0 = 1$, $w_1 = 1$. Using the normal equations, the least square solution is $w_0 = 0.43$, $w_1 = 0.89$ (MSE = 1.45).

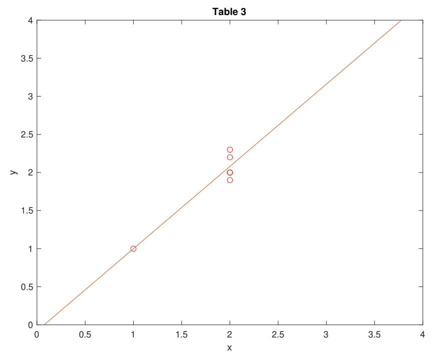
The MSE surface computed on the grid defined by the points w_1 =0, 1/5, 1/4, 1/3, 1/2, 1, 2, 3, 4, 5 and w_0 = -4, -3, -2, -1, 0, 1, 2, 3, 4, 5 is:

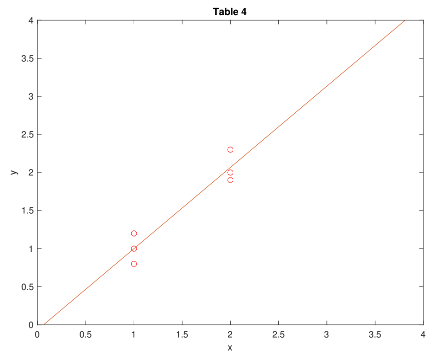


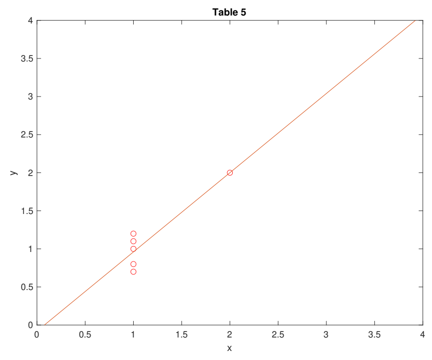
EXERCISE ♯3 (SOL): The following figures show each dataset together with the least squares linear solution:











We will show how to obtain the least squares solution for the dataset shown in Table 5. The design matrix $\boldsymbol{X}$ and label vector $\boldsymbol{y}$ are:

$$\boldsymbol{X} = \begin{bmatrix} 1 & 1 \\ 1 & 1 \\ 1 & 1 \\ 1 & 1 \\ 1 & 1 \\ 1 & 2 \end{bmatrix} \qquad \boldsymbol{y} = \begin{bmatrix} 1 \\ 0.8 \\ 1.2 \\ 1.1 \\ 0.7 \\ 2 \end{bmatrix} \qquad \boldsymbol{X}^T\boldsymbol{X} = \begin{bmatrix} 6 & 7 \\ 7 & 9 \end{bmatrix} \qquad (\boldsymbol{X}^T\boldsymbol{X})^{-1} = \begin{bmatrix} 1.8 & -1.4 \\ -1.4 & 1.2 \end{bmatrix}$$

$$(\boldsymbol{X}^T\boldsymbol{X})^{-1}\boldsymbol{X}^T = \begin{bmatrix} 0.4 & 0.4 & 0.4 & 0.4 & 0.4 & -1 \\ -0.2 & -0.2 & -0.2 & -0.2 & -0.2 & 1 \end{bmatrix} \qquad \boldsymbol{w} = (\boldsymbol{X}^T\boldsymbol{X})^{-1}\boldsymbol{X}^T\boldsymbol{y} = \begin{bmatrix} -0.08 \\ 1.04 \end{bmatrix}$$

The coefficients vector $\boldsymbol{w}$ is the result of multiplying the matrix $(\boldsymbol{X}^T\boldsymbol{X})^{-1}\boldsymbol{X}^T$ and the label vector $\boldsymbol{y}$. The entries of $(\boldsymbol{X}^T\boldsymbol{X})^{-1}\boldsymbol{X}^T$ can be used to quantify the strength of the dependence of each coefficient on one individual sample.

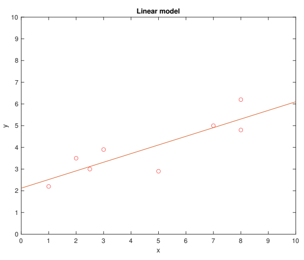
The least square solutions are: $\boldsymbol{w} = [0, 1]^T$ (Table 1), $\boldsymbol{w} = [0, 1]^T$ (Table 2), $\boldsymbol{w} = [-0.08, 1.08]^T$ (Table 3), $\boldsymbol{w} = [-0.06, 1.06]^T$ (Table 4), $\boldsymbol{w} = [-0.08, 1.04]^T$ (Table 5)

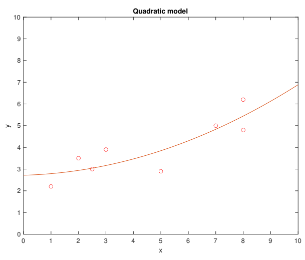
Exercise ♯4 (Sol): The mathematical expressions of a linear, quadratic and cubic model are $y = w_0 + w_1 x$, $y = w_0 + w_1 x + w_2 x^2$ and $y = w_0 + w_1 x + w_2 x^2 + w_3 x^3$, respectively. The corresponding design matrices $\boldsymbol{X}_L$ (linear), $\boldsymbol{X}_Q$ (quadratic) and $\boldsymbol{X}_C$ (cubic) are

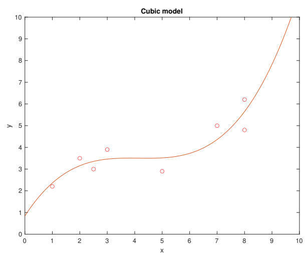
$$\boldsymbol{X}_L = \begin{bmatrix} 1 & 1 \\ 1 & 2 \\ 1 & 3 \\ 1 & 5 \\ 1 & 7 \\ 1 & 8 \\ 1 & 2.5 \\ 1 & 8 \end{bmatrix} \quad \boldsymbol{X}_C = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 2 & 4 \\ 1 & 3 & 9 \\ 1 & 5 & 25 \\ 1 & 7 & 49 \\ 1 & 8 & 64 \\ 1 & 2.5 & 6.25 \\ 1 & 8 & 64 \end{bmatrix} \quad \boldsymbol{X}_C = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 2 & 4 & 8 \\ 1 & 3 & 9 & 27 \\ 1 & 5 & 25 & 125 \\ 1 & 7 & 49 & 343 \\ 1 & 8 & 64 & 512 \\ 1 & 2.5 & 6.25 & 15.625 \\ 1 & 8 & 64 & 512 \end{bmatrix} \quad \boldsymbol{y} = \begin{bmatrix} 2.2 \\ 3.5 \\ 3.9 \\ 2.9 \\ 5 \\ 6.2 \\ 3 \\ 4.8 \end{bmatrix}$$

Using the normal equations, the least squares solutions are $\boldsymbol{w}_L = [2.12, 0.40]^T$, $\boldsymbol{w}_Q = [2.72, 0.03, 0.04]^T$ and $\boldsymbol{w}_C = [0.84, 1.95, -0.47, 0.04]^T$.

Plotting the resulting models we obtain:



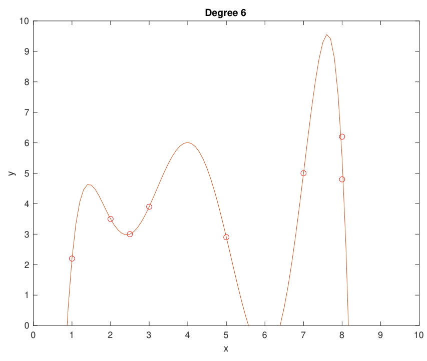


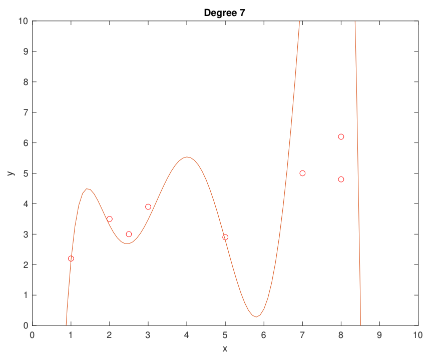


Their MSE values are MSE$_L$= 0.42, MSE$_Q$= 0.40 and MSE$_C$= 0.28 respectively. The cubic model achieves the lowest MSE value. Comparing models based on their training MSE error is a mistake: we need to use a separate dataset to evaluate the future deployment performance. We are not given a separate dataset, hence we cannot say that one of the models is better than the rest.

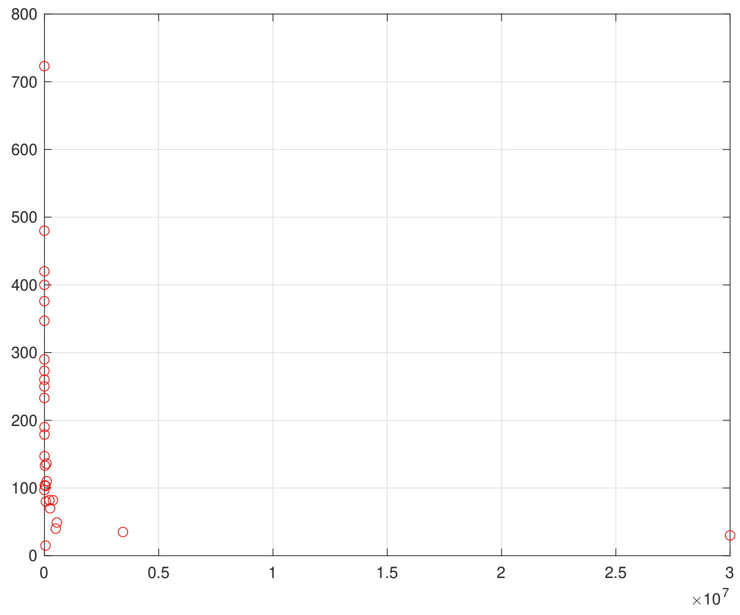
When the number of coefficients of a polynomial model is equal or greater than the number of samples, in principle we could find a solution that achieves perfect predictions on the training dataset. Our models are however likely to be memorising irrelevant details at the expense of deteriorating their generalisation ability.

This dataset consists of 8 samples and in principle there exists a polynomial of degree 7 or higher that predicts perfectly all the labels in the training dataset (MSE = 0). However, notice that samples 6 and 8 share the same predictor but have different labels. No model can return 2 different labels for the same predictor. As a result of this, the matrix $\boldsymbol{X}^T\boldsymbol{X}$ is singular and not invertible. Any numerical result produced in this scenario is likely to be useless. The MSE of the least square polynomial of order 6 is MSE$_6$=0.12, and of order 7 is MSE$_7$=71.70.



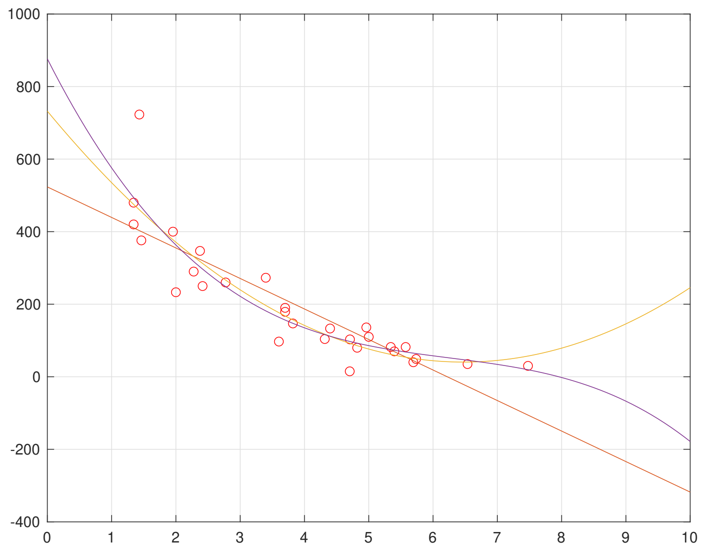


EXERCISE ♯5 (SOL):
Plotting the dataset as it is given to us, it is hard to see any meaningful structure, as most of the samples appear concentrated within a narrow range of body mass values:



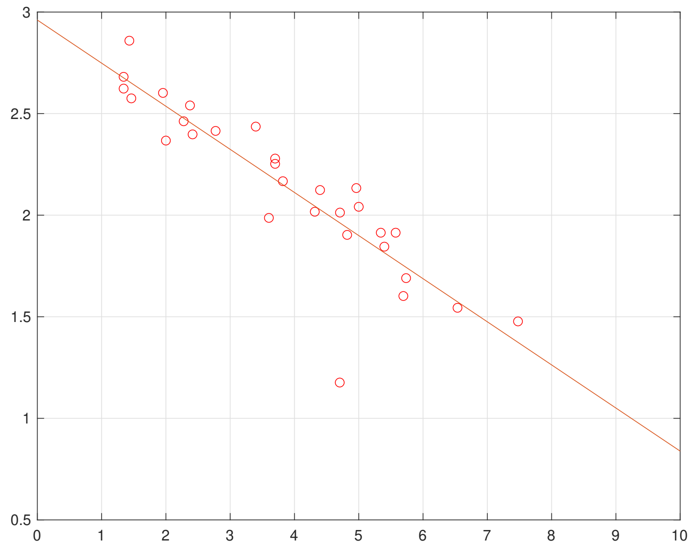
Removing samples with large body mass values would not help much, as the problem is that the body mass values span several orders of magnitude. It makes more sense to transform the dataset using a logarithmic scale. To do this, we simply need to calculate the logarithm base 10 of each body mass. For instance, the body mass of a meerkat is BM = 5000 g, taking logarithms we obtain log_{10}(BM) = 3.7. Instead of using the body mass as our predictor, we can use the logarithm base 10 of the BM.

Fitting a linear, quadratic and cubic model to this transformed dataset, we get the solutions plotted in the following figure:

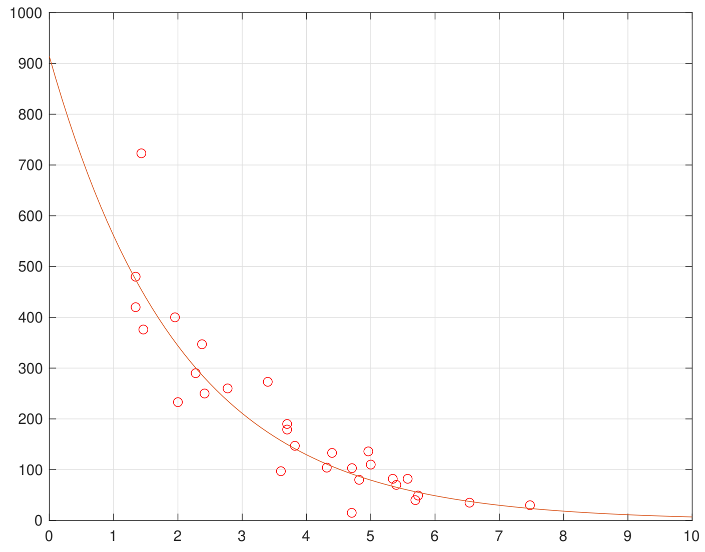


This visualisation shows how well each model fits the (training) dataset. We could also obtain their training MSE values. However, remember that the training MSE value should not be used to compare models: we need to challenge each model with unseen data. Interestingly, even though we do not have a separate dataset to test the three models, we can use domain knowledge to evaluate them. Note that the linear and cubic models predict negative heart rates, which makes no sense. The quadratic model predicts an increase in heart rate as the body mass increases, which is questionable too. Therefore, in addition to data-driven test tasks, we can consider other approaches which use what we know about the problem at hand.

If we apply a logarithmic transformation to both the body mass and heart rate, we would observe that the transform samples seem to lie on a straight line. Let's fit a linear model to this transformed dataset (the coefficients of this model are $w_0 = 2.96$ and $w_1 = -0.2$):



And then transform the heart rate back to its original scale:



The resulting model is a so-called exponential model.

EXERCISE ♯6 (SOL): A regression model that predicts several labels can be seen as several ordinary regression models operating in parallel. Let K denote the number of labels and $\boldsymbol{w}_1, \boldsymbol{w}_2, \ldots \boldsymbol{w}_K$ be the coefficients for each one of the individual regression models. Using matrix notation, we can define a matrix $\boldsymbol{W} = [\boldsymbol{w}_1, \boldsymbol{w}_2, \ldots, \boldsymbol{w}_K]$ containing the coefficients of all the individual models. Then, we can predict $\boldsymbol{y}$ as $\boldsymbol{y} = \boldsymbol{W}^T \boldsymbol{x}$.