National University of Singapore
School of Computing
IT5005 Artificial Intelligence

Introduction to Learning

1. **Linear Regression Model Fitting**.

   You are given several data points as follows:

x_1	x_2	x_3	y
6	4	11	20
8	5	15	30
12	9	25	50
2	1	3	7

   Apply the Normal Equation formula to obtain a linear regression model that minimizes MSE of the data points.

$$w = (X^T X)^{-1} X^T Y$$

   Solution:

   We can just make use of the Normal Equation to solve for the weights w.

$$X = \begin{bmatrix} 1 & 6 & 4 & 11 \\ 1 & 8 & 5 & 15 \\ 1 & 12 & 9 & 25 \\ 1 & 2 & 1 & 3 \end{bmatrix}, Y = \begin{bmatrix} 20 \\ 30 \\ 50 \\ 7 \end{bmatrix}$$

$$\begin{aligned} w &= (X^T X)^{-1} X^T Y \\ &= \begin{bmatrix} 4 & -5.5 & -7 & 7 \end{bmatrix}^T, \end{aligned}$$

$$\hat{y} = 4 - 5.5x_1 - 7x_2 + 7x_3$$

   Extra questions: Normal Equation needs the calculation of $(X^T X)^{-1}$. But sometimes this matrix is not invertible. When will that happen, and what should we do in that situation?

2. **Examining Cost Functions**.

   For Linear Regression, there are two popular cost functions,

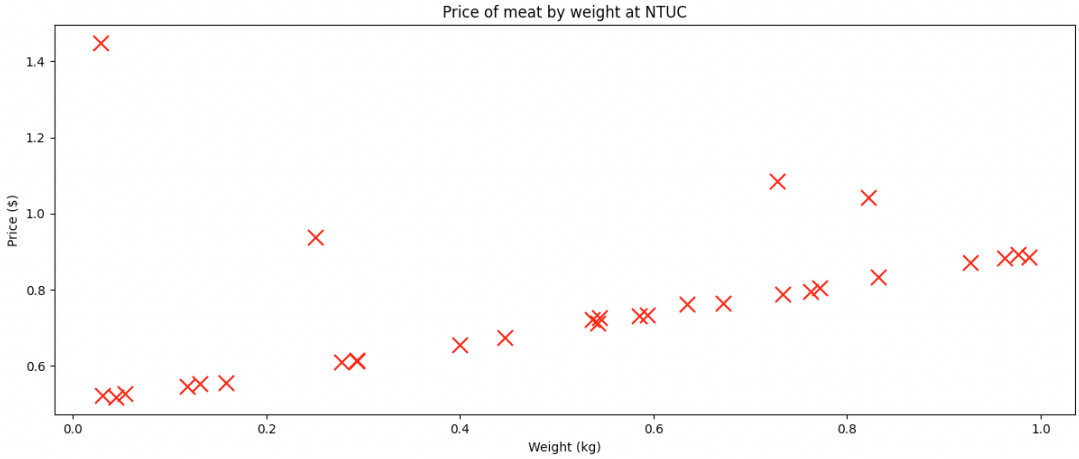
   Mean Squared Error:

$$L(y, \hat{y}) = \frac{1}{2}(y - \hat{y})^2 \tag{1}$$

   and **Mean Absolute Error**:

$$L(y, \hat{y}) = \frac{1}{2}|y - \hat{y}| \tag{2}$$

   (a) Given the scatter plot of a dataset containing the actual weight of meat at NTUC (x) and its price (y), justify your choice of cost function for this problem.



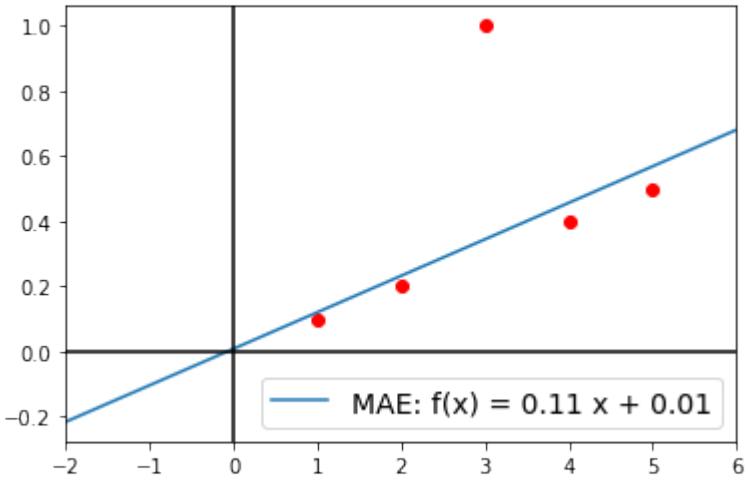
   Solution:

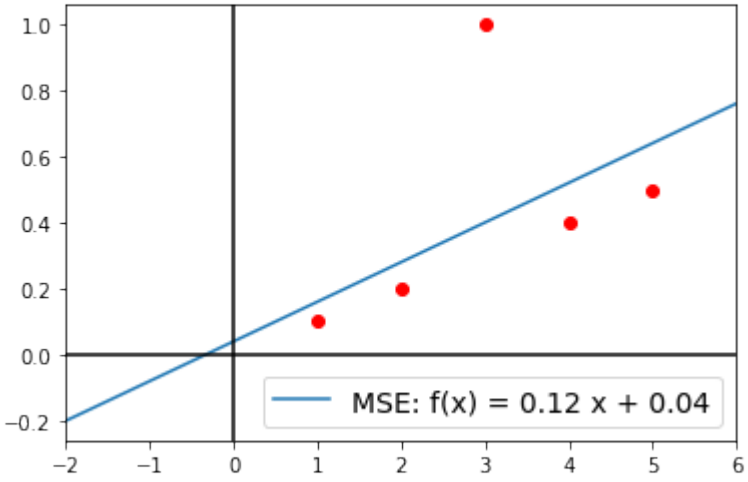
   For this dataset, it contains relatively few outliers and these outliers could be a result of human error. As such, MAE is preferred because it penalises the model less compared to MSE which penalises the outliers more by squaring the error terms, resulting in larger residuals (i.e. vertical differences).

   The choice between MSE and MAE will depend on the goals of the model and the nature of the data. If we consider outliers as important and should be penalized heavily, MSE may be the preferred metric. If outliers are considered less important and should have a smaller impact, MAE may be the preferred metric.

Two examples are shown below when the cost functions are MAE and MSE respectively.

Note: As shown in these examples, outliers can have a greater impact on MSE than MAE, even if the y-values are between 0 and 1.





(b) Can you provide examples of cost functions that are better suited to handle outliers more effectively?

Solution:

Huber loss *is a combination of MSE and MAE and is defined as:*

$$L(y, \hat{y}) = \begin{cases} \frac{1}{2}(y - \hat{y})^2 & \text{for } |y - \hat{y}| \leq \delta \\ \delta \cdot |y - \hat{y}| - \frac{1}{2}\delta^2 & \text{otherwise} \end{cases} \tag{3}$$

where δ is a threshold that determines the transition between the MSE and MAE behaviors.

Log-cosh loss *is defined as:*

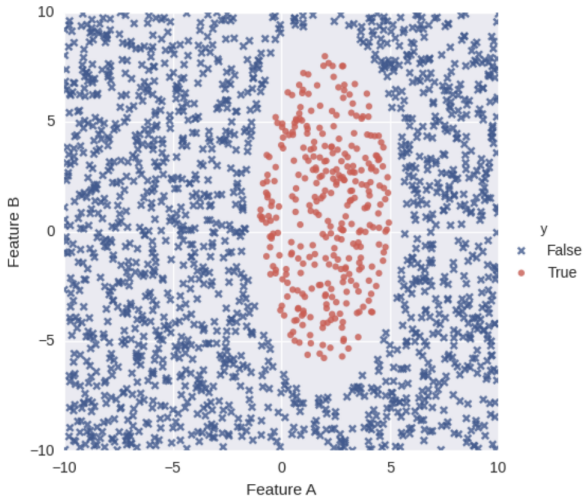
$$L(y, \hat{y}) = \log(\cosh(y_i - \hat{y}_i)) \tag{4}$$

For small values of x, $\log(\cosh(x)) \approx \frac{1}{2}x^2$, which is similar to MSE. For larger values of x, $\log(\cosh(x)) \approx |x| - log(2)$, which is similar to MAE. Log cosh approximates MSE and MAE and is similar to the Huber loss function.

Huber loss and Log cosh loss are more robust to outliers compared to MSE and MAE because they don't give as much weight to extreme values. This makes them useful in cases where the presence of outliers might negatively impact model performance if using MSE or MAE.

3. **Linear vs Non-linear Separability**

Bondrewd Workshop is a company that produces cute fluffy bunnies through experimentation and genetic mutations. Quality control for the bunnies is done manually. A group of scientists decide whether a bunny is ready to be released into the wild based on two features: **Feature A** is a bunny's cuteness score and **Feature B** is a bunny's fluffiness score. The figure below show examples of bunnies that have been released and withheld in the past. Each dot corresponds to a bunny, and responds to the following question as true or false: this bunny is ready to be released.

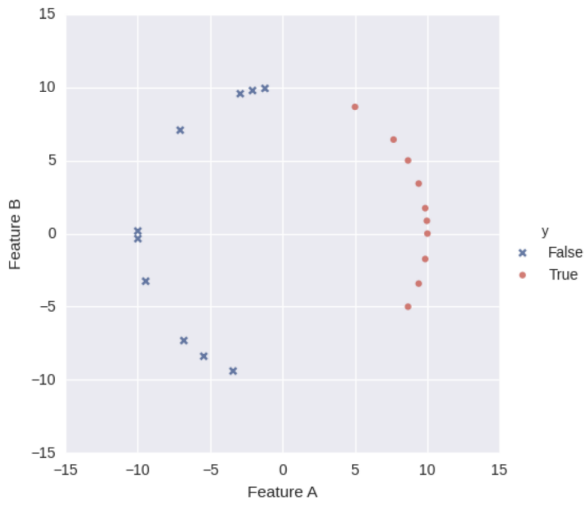


Knowing that you are an ML expert, Bondrewd the CEO has approached you and asked you to automate the decision making process.

(a) Define a reasonable set of features that will perfectly classify whether or not a bunny can be released into the wild.

Solution: *Notice that an ellipse with major and minor axis parallel to y-axis and x-axis is sufficient to classify the data. Hence* (A^2, B^2, A, B) *minimally suffices. For more general ellipses (or conics) you can use the more general set of features:* (A^2, AB, B^2, A, B).

(b) Bondrewd decides to change the production direction in the company. Bondrewed Workshop will be creating fewer, but cuter (and fluffier) bunnies. After more experiments, they have collected the examples again in the figure below.



Define a reasonable set of features that will perfectly classify whether or not a bunny can be released into the wild.

Solution: *We can use* (A) *and that is minimum set of features possible as a single vertical line is enough to classify. We can also use the set from 1(a) since a higher polynomial set of features can degenerate into a line.*

4. **Logistic Regression for Multi-Class Classification**.

   Suppose you have a classification task of deciding whether an animal is a cat, a horse, or an elephant. However, you can't see the animal but you have the information about

   - The weight of the animal (in kilogram)
   - The length of the animal (in meter)

   You, being an ML Expert, suggested to use 3 Logistic Regression models to solve this problem. After training on the training dataset, you get the following parameters:

$$w_{cat} = [4.2, -0.01, -0.12]$$
$$w_{horse} = [-20, -0.08, 35]$$
$$w_{elephant} = [-1250, 0.82, 0.9]$$

   (a) You're given a list of animals with their features. Compute the probability of an animal belonging to a certain class and classify them accordingly.

Weight (kg)	Length (m)
4.2	0.4
720	2.4
2350	5.5

Table 1: List of animals with unknown class

Solution:
For the first animal,

$$p_{cat} \approx 1 \quad p_{horse} = 0.00177 \quad p_{elephant} \approx 0$$

Hence, we classify the first animal as a cat.
For the second animal,

$$p_{cat} = 0.036 \quad p_{horse} = 0.999 \quad p_{elephant} \approx 0$$

Hence, we classify the second animal as a horse.
For the third animal,

$$p_{cat} \approx 0 \quad p_{horse} \approx 0 \quad p_{elephant} \approx 1$$

Hence, we classify the third animal as an elephant.

(b) What if we want to extend the classification task to classify other animals? Can we train a new model while keeping the weights of the previous models?

Solution: *It depends. For an animal that are very distinct with the three animals, we can create a new logistic regression model without changing the previous weights. However, for classifying a new animal that is similar with one of the classes (e.g, classifying a dog), we need to retrain the old models.*

5. **Precision, recall, F1 score and ROC curve**

Esophageal cancer is a serious and very aggressive disease. In this question, we want to look at the size of a patient's tumor and decide whether the cancer has spread to his or her lymph nodes. Using what we learnt, we use maximum dimension (mm) of esophagus tumor as input, and label 1 if the cancer had spread to their lymph nodes, and 0 otherwise. We derived this machine learning model M which outputs a continuous score for every input sample. Figure 1 shows the results for 20 samples from the model. The actual labels can be either 1 (red, positive label) or 0 (blue, negative label). The model output makes the final classification decision. If a threshold, p is given, model M outputs label 1 if M(x) is greater than or equal to the threshold, otherwise the model outputs 0.

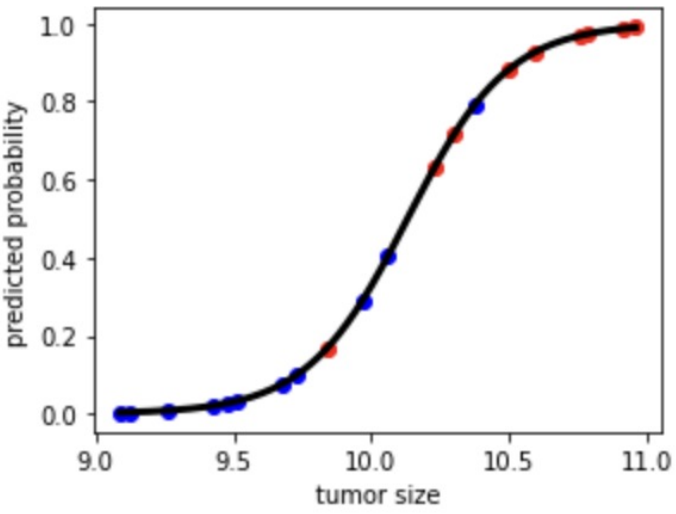


Figure 1: Model probability output and tumor size

(a) For the threshold p = 0.5, come up with the confusion matrix.

(b) For the threshold p = 0.5, find the precision, recall and F1 score.

Hint: Precision = $\frac{TP}{TP+FP}$ *Recall* = $\frac{TP}{TP+FN}$ *F1 score* = $\frac{2TP}{2TP+FP+FN}$

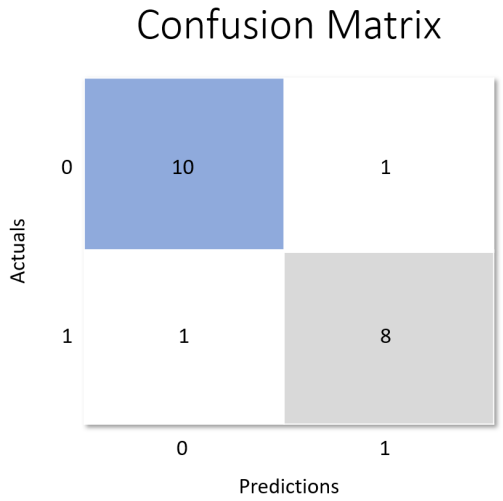


Figure 2: Confusion matrix

Solution: *We find that TP = 8, FP = 1, TN = 10, FN = 1. Precision, Recall, and F1 scores can be calculated as follows.*

$$Precision = \frac{TP}{TP + FP} = \frac{8}{8+1} = \frac{8}{9}$$

$$Recall = \frac{TP}{TP + FN} = \frac{8}{8+1} = \frac{8}{9}$$

$$F1 = \frac{2 * TP}{2 * TP + FP + FN} = \frac{2 * 8}{2 * 8 + 1 + 1} = \frac{8}{9}$$

(c) Based on figure 1, derive the ROC curve.

Hint: TPR = $\frac{TP}{ActualPositive} = \frac{TP}{TP+FN}$ *FPR =* $\frac{FP}{ActualNegative} = \frac{FP}{TN+FP}$

Hint 2: Tabulate a confusion matrix and from there, calculate the true positive rates and false positive rates. Mark out the corresponding point on the graph.

Repeat this for at least four different thresholds.

(d) Based on the ROC curve you derived, decide which threshold you want to choose among p = 0.2, p = 0.5 and p = 0.8.

Solution: *Among these three thresholds, we should choose p = 0.5. p = 0.2 and p = 0.5 gives the same true positive rate, but false positive rate for p = 0.2 is higher. p = 0.5 and p = 0.8 gives the same false positive rate, but true positive rate for p = 0.5 is higher. Hence we choose p = 0.5.*

(e) In this question's case for detecting tumours, should we maximize precision or recall? Explain the reason for your choice.

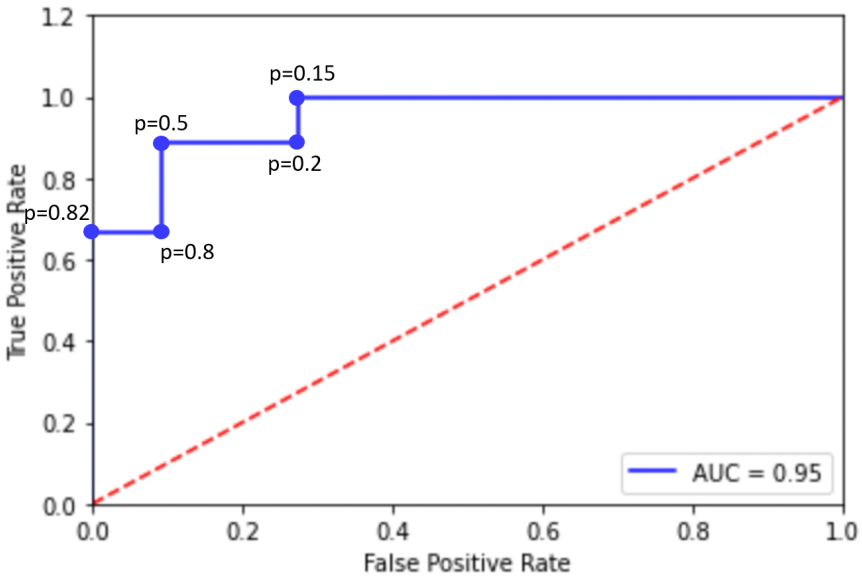


Figure 3: ROC curve

Solution: *If cancer detection is being performed as a regular check up, then precision should be maximized; as we do not want to start cancer treatment on a person unless we are sure that he has cancer. On the other hand, if cancer detection is being performed as part of cancer treatment progress monitoring, then recall should be maximized; as we do not want to stop the ongoing treatment unless we are sure that there is no residual tumour cell left in the patient.*

(f) Suppose now we want to detect plagiarism instead, should we maximize precision or recall? Explain the reason for your choice.

Solution: *In this case, we should maximize precision. This is because we don't want to wrongly accuse those who did not plagiarize. Therefore, we should minimize false positives.*

Bonus: Suppose we were helping banks with credit card fraud detection. In such a case, should we maximize precision or recall? Explain the reason for your choice.

Solution: *In this case, there isn't a clear guideline on which to maximize. If we allow a lot of false positives (if fraud is the positive label here), we will incorrectly think that some transactions are fraudulent and block them, causing inconvenience to the customers. On the other hand, if we allow for a lot of false negatives, then we will miss a lot of fraudulent cases, and the bank will lose a lot of money. Given these, in this case, we should strike a balance between recall and precision.*

6. **Perceptron Learning Algorithm**.

Consider the dataset shown in Table 2. Mr. Aiken would like to build a linear classification model. To this end, he initialized the weight vector as $\mathbf{w} = [0, 1, 0]$. He is keen on solving this problem by hand. Assume that Aiken is using perceptron learning algorithm with a learning rate of 0.1. You need to help him in getting started by answering the following questions:

Data Index	Feature A	Feature B	Label
1	1	0	-1
2	1	1	-1
3	1	2	1
4	2	1	1

(a) Using the initial weight vector, find the predicted label for each data point.

(b) Do the weights need update? If yes, perform one iteration of the weight update.

(c) Could Rosenblat's PLA converge to a solution? Provide the rationale.

Solution:

(a) Let $\hat{y}$ be the predicted label.

$$\hat{y} = sgn(\mathbf{w}^T\mathbf{x})$$

where $\mathbf{w}^T = [0, 1, 0]$

$\mathbf{x}$ *is the feature vector with Feature A and Feature B as elements with bias term augmented*

Data Index	*Feature A*	*Feature B*	*Label*	*Predicted Label*
1	*1*	*0*	*-1*	*1*
2	*1*	*1*	*-1*	*1*
3	*1*	*2*	*1*	*1*
4	*2*	*1*	*1*	*1*

(b) Yes, the weights need the update as two data indices 1 and 2 are misclassified. Select the first misclassified data index, i.e., $\mathbf{x}^{(1)}$, for update. An iteration of the weight update:

$$\begin{aligned} \mathbf{w}_1 &= \mathbf{w}_0 + \eta(y^{(1)} - \hat{y}^{(1)})\mathbf{x}^{(1)} \\ &= \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} + 0.1 * (-1 - 1) * \begin{bmatrix} 1 \\ 1 \\ 0 \end{bmatrix} \\ &= \begin{bmatrix} -0.2 \\ 0.8 \\ 0 \end{bmatrix} \end{aligned} \tag{5}$$

(c) PLA could converge to a solution as the data points belonging to different classes can be linearly separated as shown in the Figure 4.

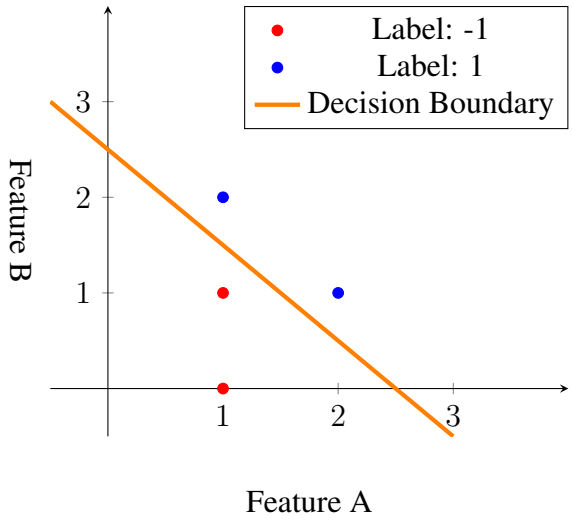


Figure 4: Plot of data points with a decision boundary