Security Analytics

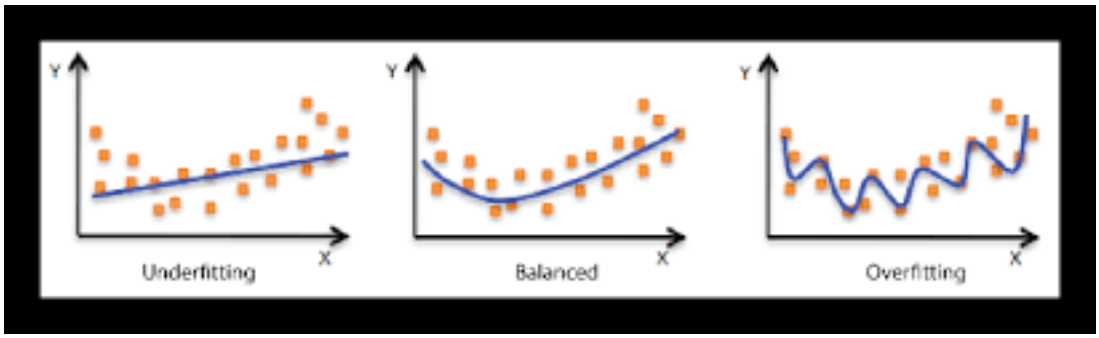
Assignment 1

Student Name: Chandrika Mukherjee
Student ID: 32808289

Problem 1

Background : part 1

- **Generalization:** How well is a trained model to classify or forecast unseen data. a generalized model should work for all subsets of unseen data. Diversity of Input is important factor in order to keep the model generalized, therefore, error rate doesn't vary when testing against unseen data.
- **Overfitting:** When a very complex model is learnt, the training output exactly fits the training actual output, but while testing the model against unseen data, the model performs poorly. Overfitting model learns the variability in the training data very well which includes noise too, therefore, it has high variance.
- **Underfitting :** When a simple model is learnt that performs poorly on training data and testing data both. Underfitting model has high bias(predictions are very far from target values).



- **Regularization:** When a model learns the outliers in the training data, the model becomes very complex which overfits the data. The coefficients take larger value when the model learns the outliers. Therefore, to penalize the coefficients, regularization is applied on the model. Regularization parameter controls the strength of Regularization.
- **No free lunch theorem:** According to this theorem, there is no single best optimization algorithm. All optimization algorithm performs equally well when their performances are averaged over all possible object functions. Suppose, a model A performs better than model B against a dataset, there will be another dataset for which B will perform better than A.
- **Occam's razor:** According to this principle, We should prefer simple models with fewer coefficients over complex models. Complex models overfits the training data.

- **Independent and identically distributed data points:** When all the data points come from same probability distribution and their occurrences are independent of each other (there are no overall trend in the data points). Example- unbiased coin toss, unbiased dice roll etc.
- **Cross-validation:** To understand how well a model will generalize an independent/unseen data, cross validation is performed.
  - Holdout Method: Splitting a dataset into training and testing set. The model is trained against the training data and later the model is tested against the testing/validation data.
  - K-Fold Cross Validation: We iterate K times on the whole dataset. a data point within the dataset is given the opportunity to be used in one test case and rest of the (K-1) times, it will be used as training data.
- **Degrees of freedom:** The number of independent parameters for a system is described as the degree of freedom. When degree of freedom is more, the model is expected to overfit the training data.

part 2

Given, the observations of two different coin tosses as follows -
$Coin_1 = H, H, H, H, T, T, H, H, H, H, T, T, H, H, H, H, T$
$Coin_2 = H, H, T, T, T, T, H, H, T, T, T, T, H, H, T, T, T$

Coin tosses are Independent and Identically Distributed(i.i.d.) random variables. This implies that the coin toss events does not follow any trend and their occurrences are independent of each other. Also, for coin toss, only possible outcome is H and T. So, even if the observation suggests that for both the cases, if another toss is performed, will get T in both. But as these coin toss events are i.i.d., the probability of getting H and T are same (0.5).

part 3

- $$X = \begin{pmatrix} 1 & x_1 & x_1^2 & \cdots & x_1^M \\ 1 & x_2 & x_2^2 & \cdots & x_2^M \\ \vdots & \vdots & \ddots & \vdots & \\ 1 & x_N & x_N^2 & \cdots & x_N^M \end{pmatrix}_{N,M+1}$$

  Converting given X matrix to Z matrix using $z_{i,j} = x_i^j$ rule

- $$Z = \begin{pmatrix} 1 & z_{1,1} & z_{1,2} & \cdots & z_{1,M} \\ 1 & z_{2,1} & z_{2,2} & \cdots & z_{2,M} \\ \vdots & \vdots & \ddots & \vdots & \\ 1 & z_{N,1} & z_{N,2} & \cdots & z_{N,M} \end{pmatrix}_{N,M+1}$$

  Z matrix corresponding to given X matrix

$$t = \begin{bmatrix} t_1 \\ t_2 \\ t_3 \\ \vdots \\ t_N \end{bmatrix}_{N,1}$$ t is the output matrix

$$w = \begin{bmatrix} w_0 \\ w_1 \\ w_2 \\ w_3 \\ \vdots \\ w_M \end{bmatrix}_{M+1,1}$$ w is the weight matrix

- total number of samples is N
- number of features is M

$$\begin{aligned} E(w) &= \frac{1}{2}\sum_{i=1}^{N}(w^T Z_i - t_i)^2 \\ &= \frac{1}{2}||Zw - t||_2^2 \\ &= \frac{1}{2}(Zw - t)^T(Zw - t) \\ &= \frac{1}{2}(w^T Z^T - t^T)(Zw - t) \\ &= \frac{1}{2}(w^T Z^T Zw - w^T Z^T t - t^T Zw + t^T t) \\ &= \frac{1}{2}w^T Z^T Zw - w^T Z^T t + \frac{1}{2}t^T t \end{aligned} \tag{1}$$

Taking derivative with respect to w and Using $\frac{\partial W^T AW}{\partial W} = 2\text{AW}$ where A is Symmetric Matrix

$$\begin{aligned} \frac{\partial E}{\partial w} &= \frac{1}{2} * 2Z^T Zw - Z^T t + 0 \\ &= Z^T Zw - Z^T t \end{aligned} \tag{2}$$

Setting derivative to zero gives,

$$w^* = (Z^T Z)^{-1} Z^T t \tag{3}$$

Resulting Polynomial in terms of w^* -

$$\begin{aligned} y(x, w^*) &= \sum_{j=0}^{M} (w^*)^T x_j \\ &= \sum_{j=0}^{M} ((Z^T Z)^{-1} Z^T t)^T x_j \\ &= \sum_{j=0}^{M} t^T Z((Z^T Z)^{-1})^T x_j \\ &= \sum_{j=0}^{M} t^T Z((Z^T Z)^T)^{-1} x_j \qquad\qquad using\ \ (A^T)^{-1} = (A^{-1})^T \\ &= \sum_{j=0}^{M} t^T Z(Z^T Z)^{-1} x_j \end{aligned} \tag{4}$$

Z and Z^T are equivalent to X and X^T respectively, therefore, we can write as follows,

$$y(x, w^*) = \sum_{j=0}^{M} t^T X(X^T X)^{-1} x_j$$