Theoretical Pipeline for Linear Regression.

1) Define Hypothesis

$$h\theta(x) = \theta_0 x_0 + \theta_1 x_1 + \theta_2 x_2 \cdots \theta_d x_d$$

$$\therefore h\theta(x) = \sum_{i=1}^{d} (\theta_i x_i) + \theta_0 x_0$$

$$\therefore h\theta(x) = \sum_{i=1}^{d} (\theta_i x_i) = \theta^T x$$

2) Define a cost function

$$J(\theta) = \frac{1}{2} \sum_{i=1}^{n} (h\theta(x^{(i)}) - y^{(i)})^2$$

$h\theta(x^{(i)})$: hypothesis prediction; $y^{(i)}$: expected prediction

$J(\theta)$ should be lowest as possible.

3) Algorithms to minimize the cost function

1) Gradient Descent
   a) Stochastic gradient Descent.
   b) Minibatch gradient Descent.
2) Normal Equation
3) Probabilistic Interpretation

1) Gradient Descent.

$$\therefore \theta^{(t+1)} = \theta^{(t)} - \alpha \nabla_\theta J(\theta^{(t+1)})$$

we will get new θ after substracting the product of learning rate and the

gradient of cost function. or the derivative of cost function from the previous θ.

$$\theta^{(t+1)} = \theta^{(t)} - \alpha \left[\sum_{i=1} (\theta^T x^{(i)} + y^{(i)}) - x^{(i)} \right]$$

This is the formula for the batch gradient descent, where the whole data (all samples) will be run for each iteration, which will consume lot of computational power.

Therefore, the stochastic gradient descent is designed which takes only one sample as a input for the new θ.

∴ Stochastic gradient Descent -

$$\tilde{J}(\theta) = \frac{1}{2} (\theta^T x^{(k)} - y^{(k)})^2$$

2) Normal Equation.

- This algorithm gives only one θ.
- Does not require learning rate.

$$\therefore \theta = (x^T x)^{-1} x^T y$$

- when x is invertible.
- Used when dataset is not large.

3) Probabilistic Interpretation:

- Likelihood of θ

$$L(\theta) = \prod_{i=1}^{n} P(y^{(i)} | x^{(i)}; \theta)$$

$$\therefore \boxed{L(\theta) = \prod_{i=1}^{n} \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left[\frac{-y^{(i)} - \theta^T x^{(i)}}{2\sigma^2}\right]}$$

- log likelyhood

$$l(\theta) = \log[L(\theta)]$$

$$\therefore l(\theta) = \sum_{i=1}^{n} \left[\log \frac{1}{\sqrt{2\pi\sigma^2}} + \log \exp\left[\frac{-y^{(i)} - \theta^T x^{(i)}}{2\sigma^2}\right]\right]$$

Choose θ to maximize $l(\theta)$

$$\boxed{\arg\max_{\theta} l(\theta) = \arg\min_{\theta} J(\theta)}$$

4) Regularization

- Regularization is performed during the cost function minimization
- It is a technique used to prevent the overfitting by adding penalty to the cost function.

$$\boxed{J(\theta) = \frac{1}{2}\left[\sum_{i=1}^{m} (h\theta(x^{(i)}) - y^{(i)}) + \lambda \sum_{i=1}^{m} \theta_j^2\right]}$$

Regularization parameter.

Regularization term.

5) Feature Scaling :

- It is used to make the model work good.
- By making sure that features are on similar scale.

e.g. $-1 < x_i > +1$.

Ways to do feature scaling -

1) Normalization (min-max scaling)
2) Standardization
3) Robust Scaling
4) Log Transformation.

1) Normalization

$$x' = \frac{x - x_{min}}{x_{max} - x_{min}}$$

2) Standardization

$$x' = \frac{x - \mu}{\sigma}$$

μ - mean , σ = standard deviation.

3) Robust Scaling.

$$x' = \frac{x - \text{median}(x)}{IQR(x)}$$

4) Log Transformation

$$x' = \log(x+1)$$

DATE :

6) Performance checking.

Evaluation of the model is done for checking whether the model is working good or not.

1) MSE (Mean Square Error)
2) MAE) (Mean Absolute Error)
3) R^2 (R^2 score)
4) RMSE (Root mean square error).

1) $$\text{MSE} = \frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y})^2$$

2) $$\text{MAE} = \frac{1}{N}\sum_{i=1}^{N}|y_i - \hat{y}|$$

3) $$R^2 = 1 - \frac{\Sigma(y_i - \hat{y})^2}{\Sigma(y_i - \bar{y})^2} = \frac{(\text{Sum of Residual})^2}{(\text{Total})^2} = \frac{RSS}{TSS}$$

4) $$\text{RMSE} = \sqrt{\text{MSE}}$$