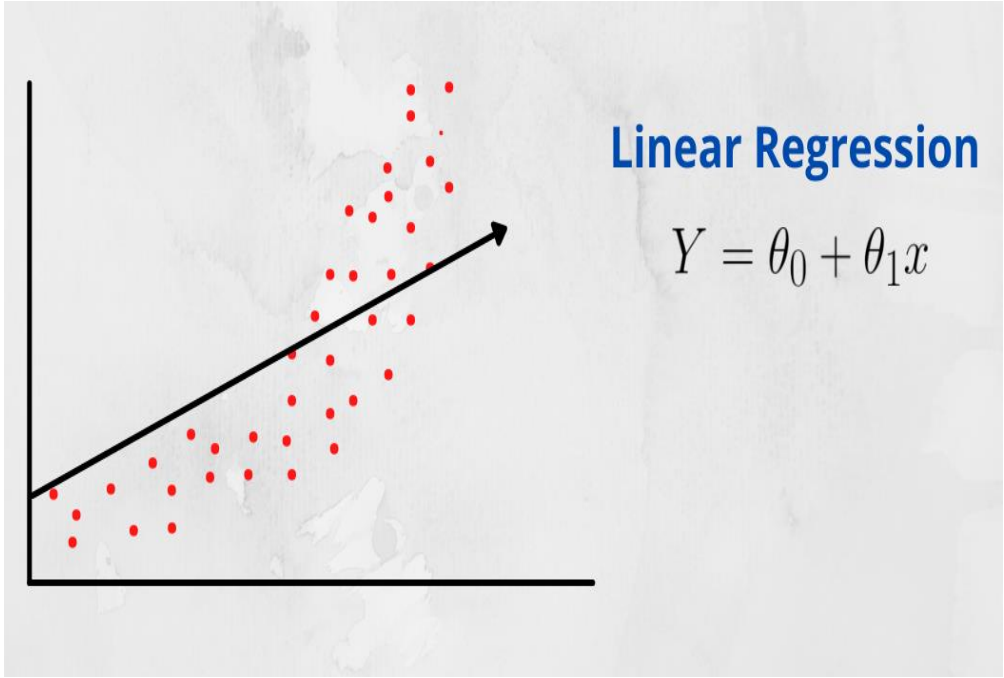
VIT-AP UNIVERSITY

Polynomial Regression

Dr. Kuppusamy .P
Associate Professor / SCOPE

Polynomial Regression

- Simple linear regression algorithm performs better during the relationship between the data is linear.
- If data is non-linear then Linear regression fails to find a best-fit line.
- Let consider the data distribution in the given diagram. The data contain non-linear relationship, and Linear regression h does not perform well i.e., line does not come close to reality.

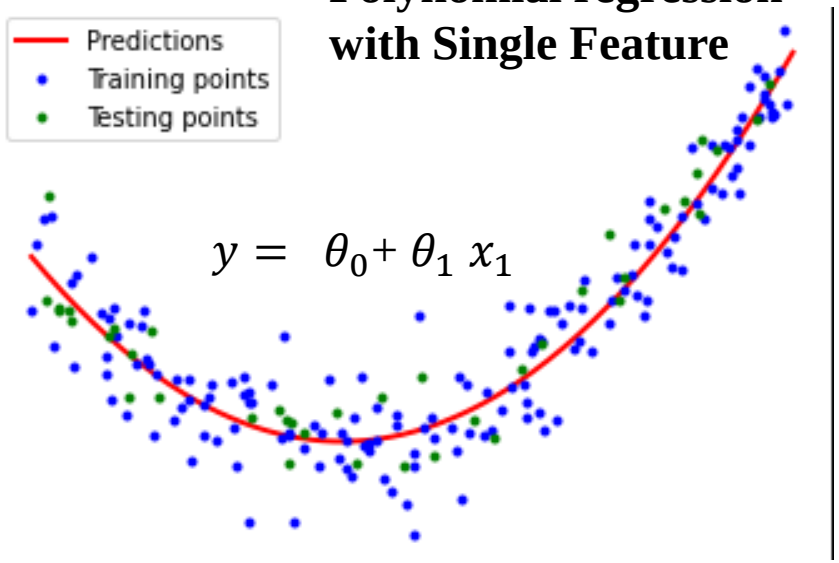


- **Polynomial regression** address this problem by identifying the curvilinear relationship between independent and dependent variables.

$$y = \theta_0 x_0 + \theta_1 x_1 + \theta_2 x_2^2 + \theta_3 x_3^3 + \ldots\ldots + \theta_n x_n^n$$

- In the equation, coefficient $\theta's$ degree is always 1. But θ values to be determined during building the model.
- Degree of order uses as a Hyperparameter
- High degree of polynomial tries to overfit the data and for smaller values of degree, the model tries to underfit.
- So, find the optimum value of a degree.

Polynomial regression with Single Feature



Polynomial Regression

1. Convert the input dataset into polynomial terms using the degree. Here degree acts as y-intercept in linear regression.

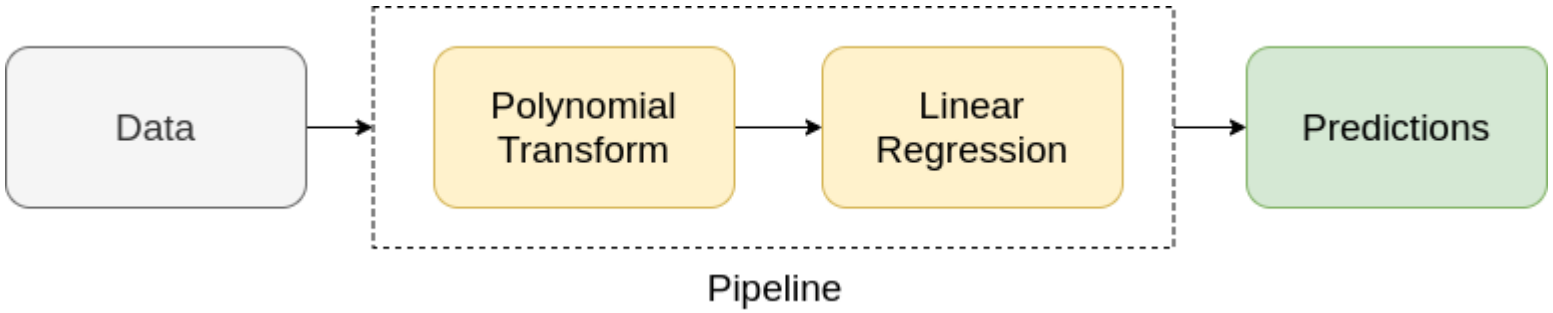
```
from sklearn.preprocessing import PolynomialFeatures
poly = PolynomialFeatures(degree=2, include_bias=True)
x_train_trans = poly.fit_transform(x_train)
x_test_trans = poly.transform(x_test)
```

2. Apply the Linear regression on the transformed data.

```
LinearRegression.fit(x_train_trans, y_train)
```

3. This process performs polynomial regression.

```
from sklearn.pipeline import Pipeline
polynomial_regression = Pipeline([ ("poly_features", polybig_features), ("std_scaler", std_scaler), ("lin_reg", lin_reg) ])
polynomial_regression.fit(X, y)
```



Select the optimal Degree

To avoid overfitting and underfitting problems,

1. Forward selection: increase the degree parameter till you get the optimal result

2. Backward selection: decrease degree parameter till you get optimal

Loss Function Vs Cost Function

- Loss Function is the error for individual data points.
- Loss Function refers to a single training example

Loss Function = *Estimated Value* (y_i') – *Actual Value* (y_i)

- Cost Function is the average of the n-sample error in the data.
- Cost Function refers to the complete training set.

$$J(\theta_0, \theta_1) = \frac{1}{2m}\sum_{i=1}^{m}(\widehat{y_1} - y_i)^2 = \frac{1}{2m}\sum_{i=1}^{m}\left((h_\theta(x_i) - y_i)\right)^2$$

Polynomial Regression with Multi-variables

- The multi-variable polynomial equation

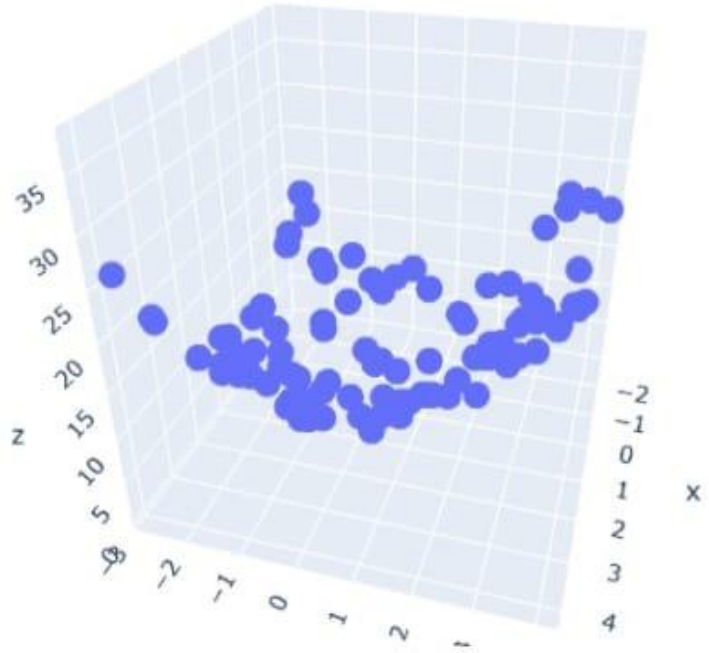
$$y = \theta_0 x_0 + \theta_1 x_1 + \theta_2 x_2^2 + \theta_3 x_3^3 + \ldots\ldots + \theta_n x_n^n$$

- Apply polynomial regression on many features (columns) in input data.
- Visualize the result in 3-dimensional plot.
- Loss Function refers to a single training example

  Loss Function = Estimated Value (y_i') – Actual Value (y_i)

- Cost Function refers to the complete training set.
- Cost function using MSE

$$J(\theta_0, \theta_1) = \frac{1}{2m}\sum_{i=1}^{m}(\widehat{y_1} - y_i)^2 = \frac{1}{2m}\sum_{i=1}^{m}\left((h_\theta(x_i) - y_i)\right)^2$$



- Gradient Descent
- $\theta_j = \boldsymbol{\theta_j - \alpha \frac{\partial}{\partial \theta} J(\theta_0, \theta_1)};\ \ j = 0\ and\ j = 1$

Gradient Descent

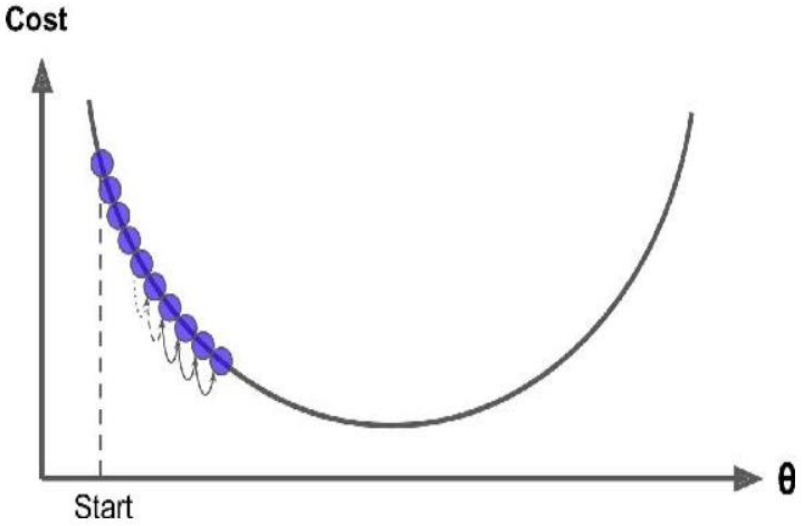
- The size of the steps determined by the learning rate parameter.

For Small Learning rate:

- If choose learning rate is too small, the algorithm takes many iterations to achieve the minimum.

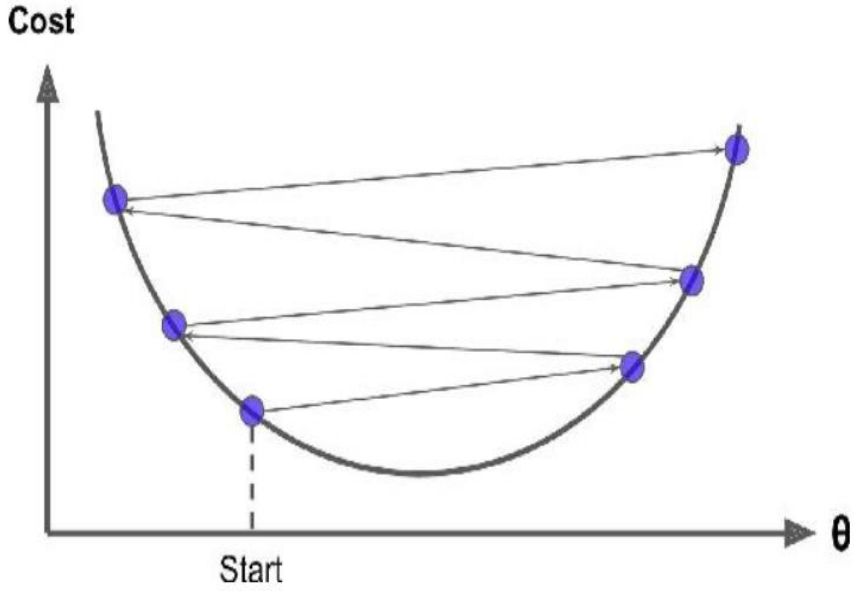
$$\theta_0 := \theta_0 - \alpha \frac{1}{m} \sum_{i=1}^{m} (h_\theta(x_i) - y_i)$$

$$\theta_1 := \theta_1 - \alpha \frac{1}{m} \sum_{i=1}^{m} ((h_\theta(x_i) - y_i) x_i)$$



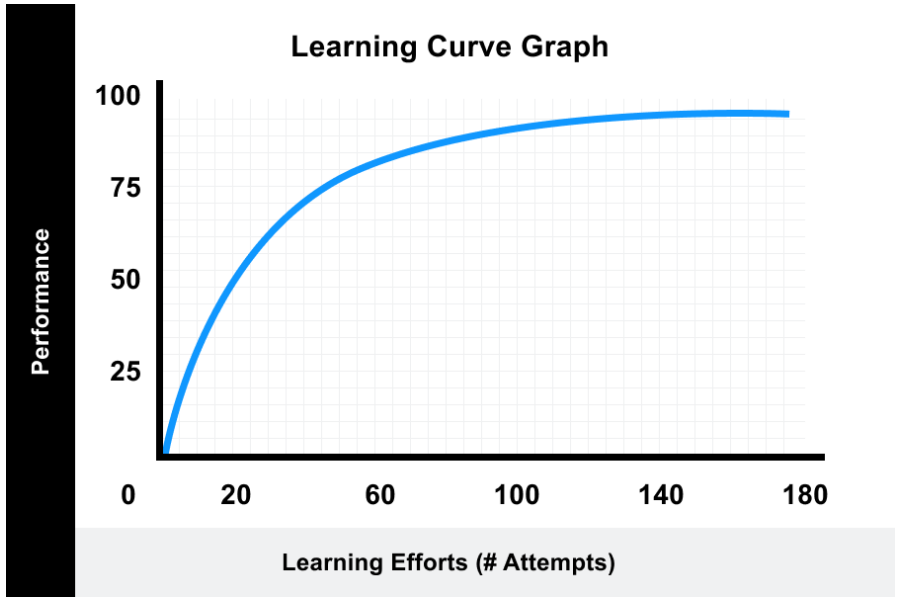
For High Learning rate:

- If choose learning rate is too high, the algorithm may jump across the valley in finding solution even higher than before.



Learning Curve

- A learning curve is a correlation between a learner's performance on a task and the number of attempts or time required to complete the task.
- Learning curve can be represented as a direct proportion on a graph.
- The learning curve shows a learner’s (model) efficiency in a task that improves over time.

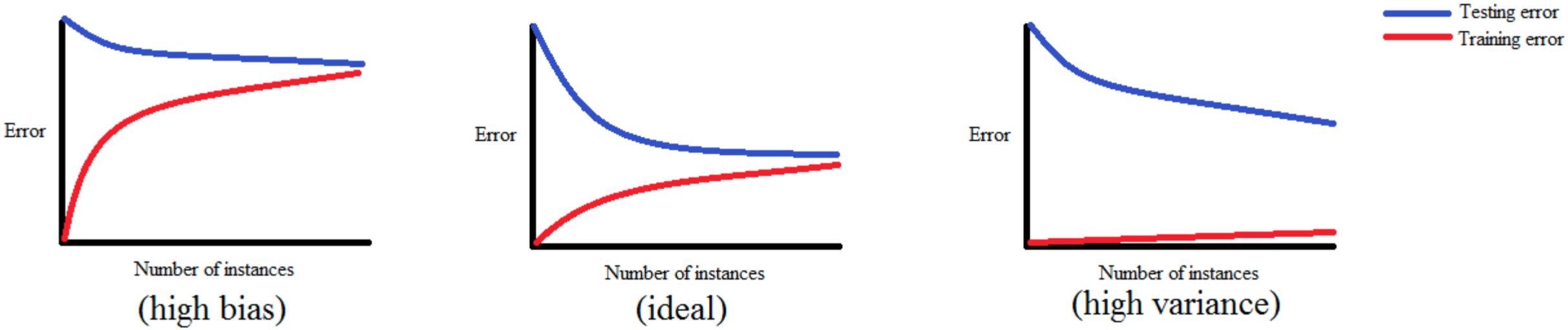


Learning Curve in Machine Learning

- Learning curve shows changes in the training and validation errors with respect to the number of training examples used in training the model.
- If a model is balanced, both errors converge to small values when the training sample size increases.

Bia and Variance

- **Bias** is defined as the mistake (error) caused by the model's simple assumptions in fitting the data.
- A high bias represents the model is unable to capture training data patterns that leads to under-fitting.
- **Variance** is defined as the error caused by the complicated model tries to match the training data.
- When a model has a high variance, it passes over the majority of data points that leads the data to overfit.
- As the model complexity grows, the bias reduces while the variance increases, and vice versa.
- A good model should have minimal variance and bias. However, both is nearly impossible.
- So, a **trade-off** must be made to build a strong model that performs well on both train and unseen data.



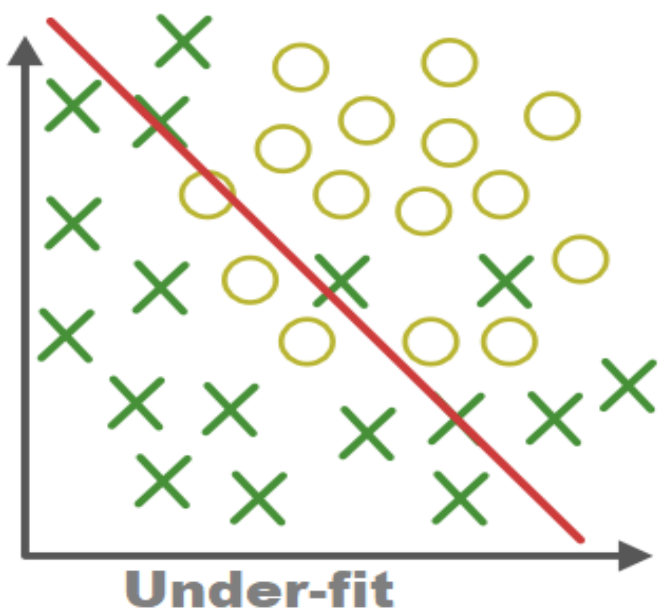
Data Distribution Vs Learning Curve

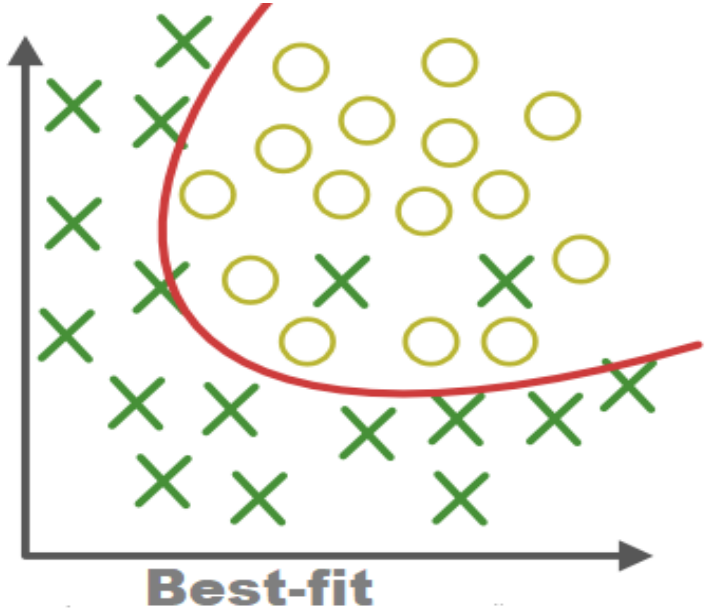
Underfitting

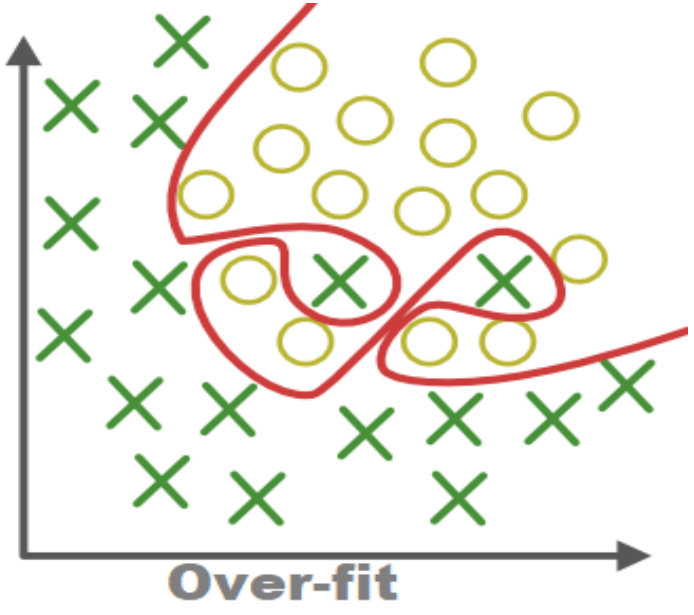
- If a model contains **high bias**, model is **underfitting** with the data.
- In this case, both errors fail to decrease even increases the training set.

Overfitting

- Sometime model performs well with the training data but does not perform well with the test data i.e., model is **overfitting** with the training data due to data with **high variance** and **Low bias**.
- In this case, increasing the training sample size decreases the training error but it fails to decrease the validation (test) error.







Regularization

- **Regularization** approach prevents the model from overfitting by adding extra information (training samples) to the training dataset.
- It maintain all variables or features in the model by **reducing the magnitude** of the variables for better performance and generalization of the model.
- Primary process is to regularize or reduce the coefficient (weight) of features toward zero i.e., reduce the magnitude of the features without changing the number of features.

Working Principle

- Regularization works by adding a penalty or complexity term to the complex model.
- Let's consider the simple linear regression equation:

$$y = \theta_0 x_0 + \theta_1 x_1 + \theta_2 x_2 + \ldots\ldots + \theta_n x_n + b$$

- x_0 is bias, θ_0, θ_1,… θ_n are Weights or Parameters and b is an intercept.

Regularization

- Cost function for linear model is

$$\text{Cost Function} = \text{Error}\,(y_i, y_i') = J(\theta) = \sum_{i=1}^{m}(y_i - y_i')^2 = \sum_{i=1}^{m}(y_i - \sum_{j=1}^{n}\theta_j * X_{ij})^2$$

- Linear model optimizes the θ_0 and b to minimize the error.

  M= No. of Samples, n - No. of features. θ_0 is not penalized.

- Loss function for linear model is Least Squared Error (Residual Sum of Squares) or MSE.

Types for Regularized Linear Models

- Ridge Regression **(L2 regularization**)
- Lasso (Least Absolute Shrinkage and Selection Operator) Regression **(L1 regularization**)

Ridge Regression (L2 regularization)

- Ridge regression is defined as Linear regression that uses **L2 regularization**
- In Ridge regression, a small amount of bias is added to get better long-term predictions.
- Ridge regression is a regularization technique that reduces the complexity of the model.
- It is also called as **L2 regularization**.
- The cost function added with the penalty term **bias** is called Ridge Regression penalty.
- It is calculated by multiplying with the lambda to the **squared** weight **(coefficient)** of each individual feature.
- The equation for the cost function in ridge regression:

$$\text{Cost Function} = \text{Error}\,(y_i, y_i') = J(\theta) = \sum_{i=1}^{m}(y_i - y_i')^2 = \sum_{i=1}^{m}(y_i - \sum_{j=1}^{n}\theta_j * X_{ij})^2 + \lambda \sum_{j=1}^{n}\theta_j^2$$

- λ is regularized parameter. M= No. of Samples, n – No of features, Lambda range is 0 to 1. θ_0 is not penalized.
- The penalty term $\lambda \sum_{j=1}^{n}\theta_j^2$ regularizes the coefficients of the model.

Ridge Regression (L2 regularization)

- This regularization reduces the amplitudes of the coefficients that decreases the model's complexity.
- If the values of λ tend to zero, the equation becomes the cost function of the linear regression model.
- Hence, the model will resemble the linear regression model for the minimum value of λ.
- A general linear or polynomial regression performance is poor when high collinearity between the independent variables.
- Ridge regression addresses this issue.
- Ridge regression solves the problems even problem contains more parameters than samples.

Lasso (Least Absolute Shrinkage and Selection Operator) Regression (L1 regularization)

- Lasso regression is also regularization technique that reduces the complexity of the model.
- The penalty term contains only the **absolute weights**.
- It can shrink the slope to 0 due to absolute values. But Ridge Regression can only shrink it near to 0.
- It is also called as L1 regularization.
- The equation for the cost function of Lasso regression:

$$\text{Cost Function} = \text{Error}\,(y_i, y_i') = J(\theta) = \sum_{i=1}^{m}(y_i - y_i')^2 = \sum_{i=1}^{m}(y_i - \sum_{j=1}^{n}\theta_j * X_{ij})^2 + \lambda \sum_{j=1}^{n}|\theta_j|$$

- M= No. of Samples, n – No of features, Lambda range is 0 to 1. θ_0 is not penalized.
- The penalty term $\lambda \sum_{j=1}^{n}|\theta_j|$ regularizes the coefficients of the model.
- In Lasso regression, Some of the features are completely neglected for model evaluation.
- This help us to reduce the overfitting in the model as well as the feature selection.

- **Difference between Ridge Regression and Lasso Regression**
- Ridge regression is mostly used to reduce the overfitting in the model, and it includes all the features present in the model. It reduces the complexity of the model by **shrinking the coefficients**.
- Lasso regression helps to reduce the overfitting in the model as well as **feature selection**.

VIT-AP UNIVERSITY

References

1. Tom M. Mitchell, Machine Learning, McGraw Hill , 2017.

2. EthemAlpaydin, Introduction to Machine Learning (Adaptive Computation and Machine Learning), The MIT Press, 2017.

3. Wikipedia