# Module : Supervised Learning (Regression) Module 3

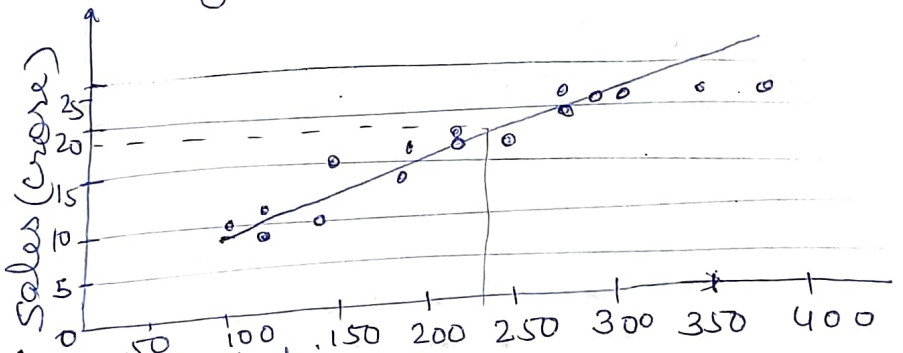
→ Focus is on prediction of future results using linear regression concepts.

→ It is a form of predictive modelling technique which tells us the relationship between the dependent (target variable) and independent variables (predictors)

- Simple Linear Regression
- Multiple linear Regression

## 1. Simple Linear Regression

The most elementary type of regression model is the simple linear regression which explains the relationship between a dependent variable and one independent variable using a straight line. The straight line is plotted on the scatter plot




* Use this straight line model to predict for a new data point.

→ Independent variable (Predictor variable)

Dependent variable (output variable)

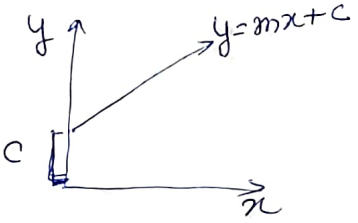
The standard equation of the regression line is given by

$$y = \beta_0 + \beta_1 x$$

$\beta_0$ ← Intercept

$\beta_1$ ← slope

* When $x = 0$ $y = \beta_0$ means there will be still some sales.

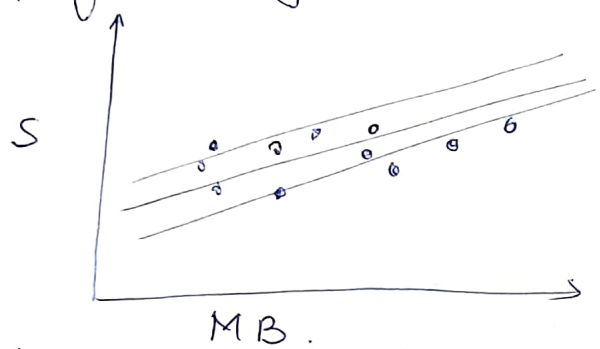



m = slope ↑ shows how strong the relationship between x and y

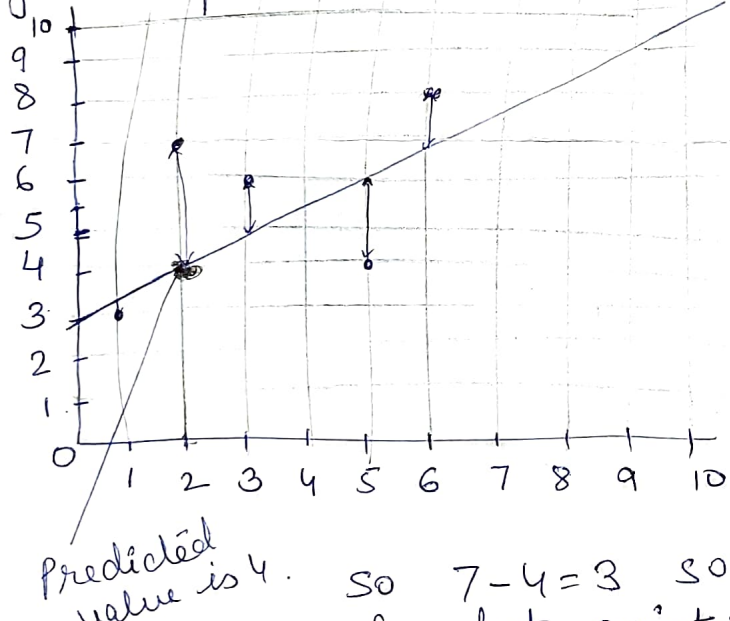
when $x = 0$ $y = c$ ↑ intercept

# Best-Fit Line

We fit straight line to a set of data points. We would like to know what is the best fit line that we can choose for future predictions given the fact that there are multiple straight lines that can fit the given data points.




To find the best-fit line there is a notion called as Residuals. The residual will be associated with every data point




When $x = 2$, actual value of $y = 7$ whereas acc. to straight line model that we have chosen the predicted y value is 4.

So $7 - 4 = 3$ so 3 is called Residual for data point 2.

$$Y = \beta_0 + \beta_1 X$$

$$e_i = y_i - y_{pred}$$

$$e_1^2 + e_2^2 + \dots + e_n^2 = \boxed{RSS}$$ (Residual Sum of Squares)

So RSS needs to be minimized. We pick that straight line whose values of $\beta_0$ and $\beta_1$ is such that the value of RSS is minimized.

RSS can be written in terms of $\beta_0$ and $\beta_1$

i.e. $RSS = (Y_1 - \beta_0 - \beta_1 X_1)^2 + (Y_2 - \beta_0 - \beta_1 X_2)^2 + \dots (Y_n - \beta_0 - \beta_1 X_n)^2$

$$RSS = \sum_{i=1}^{n} (Y_i - \beta_0 - \beta_1 X_i)^2$$

In linear regression the cost function is RSS. The goal is to minimize the cost function. Choose $\beta_0$ and $\beta_1$ such that this value is minimized

## Ways to Minimize Cost Function (RSS)

1. Differentiation
2. Gradient Descent

example

| Marketing Budget (X) (in lacs) | Actual Sales (Y) (in crores) | Predicted sales ($Y_{pred}$) | RSS |
|---|---|---|---|
| 127.4 | 10.5 | | |
| 364.4 | 21.4 | | |
| 150 | 10 | | |
| 128.7 | 9.6 | | |
| 285.9 | 17.4 | | |
| 200 | 12.5 | | |
| 303.3 | 20 | | |
| 315.7 | 21 | | |
| 169.8 | 14.7 | | |
| 104.9 | 10.1 | | |
| 297.7 | 21.5 | | |
| 256.4 | 16.6 | | |
| 249.1 | 17.1 | | |
| 323.1 | 20.7 | | |
| 223 | 15.5 | | |
| 235 | 13.5 | | |
| 200 | 12.5 | | |

$\beta_0 = 3.3525$

$\beta_1 = 0.0528$

## TSS (Total Sum of Squares)

→ is a variation of the values of a dependent variable from the sample mean of the dependent variable.

→ total variation in a sample

$$TSS = (Y_1 - \bar{Y})^2 + \dots (Y_n - \bar{Y})^2 \qquad \bar{Y} = \text{average}$$

$$= \sum_{i=1}^{n} (Y_i - \bar{Y})^2$$

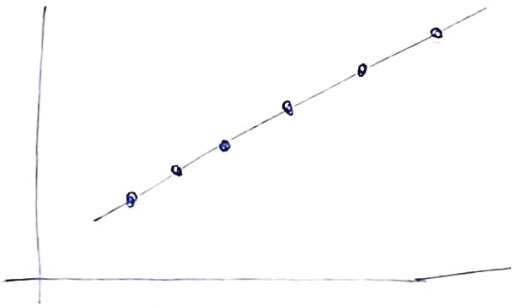
$$R^2 = 1 - \frac{RSS}{TSS}$$

The strength of linear regression model can be assessed using 2 metrics:

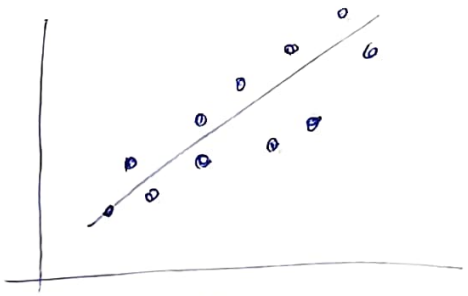
1) $R^2$ or Coefficient of Determination
2) Residual Standard Error (RSE)

## $R^2$ (tells how good a model is)

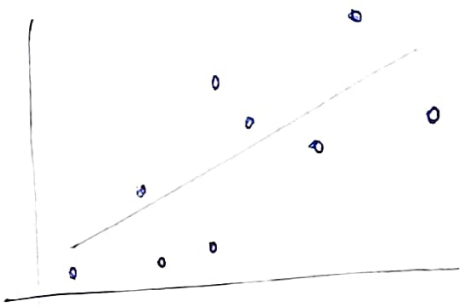
→ it measures how well actual outcomes are replicated by the model, based on the proportion of total variation of outcomes explained by the model. Higher the $R^2$, the better the model fits your data.



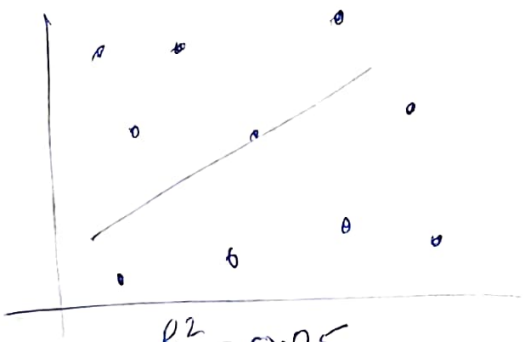
$R^2 = 1$



$R^2 = 0.70$



$R^2 = 0.36$

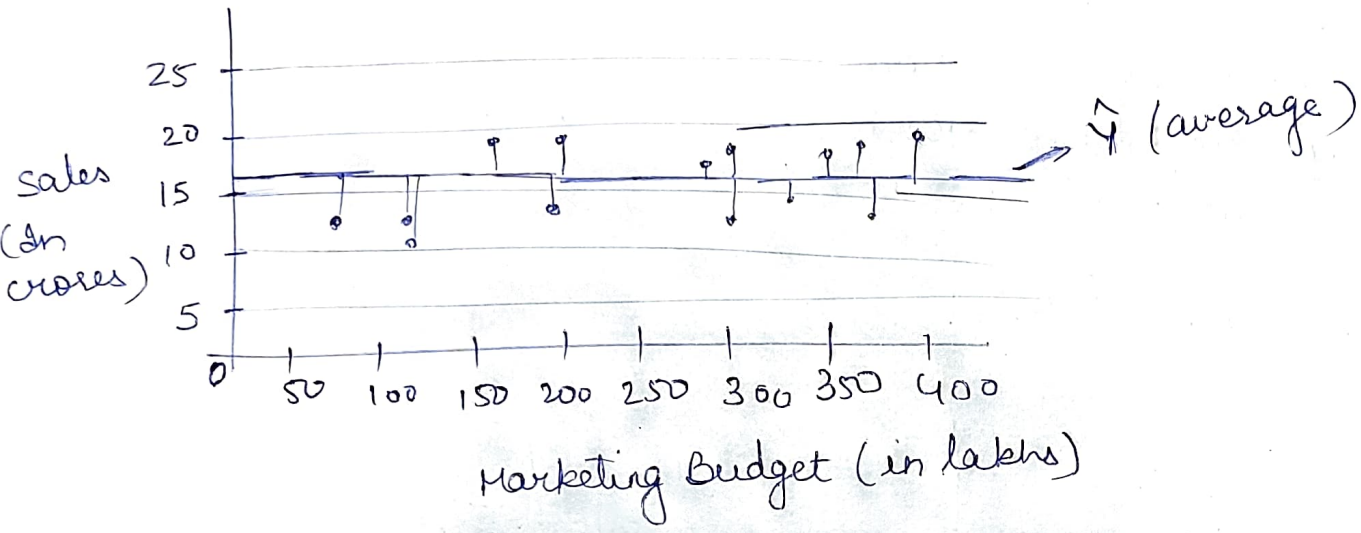


$R^2 = 0.05$

* $R^2$ will lie between 0 and 1

## Importance of RSS/TSS:

If we know nothing about linear regression and still have to draw a line to represent those points, atleast we can have a line pass through the mean of all points. This is the worst possible approximation we can do. TSS gives us the deviation of all points from the mean line.




Apart from $R^2$ there is one more quantity named RSE (Residual Square Error) which is linked to RSS.

$$RSE = \sqrt{\frac{RSS}{df}} \qquad df = n-2$$

$n$ = no. of data points
df = degree of freedom

## Assumptions of Simple Linear Regression

1. There is a linear relationship between X and Y.

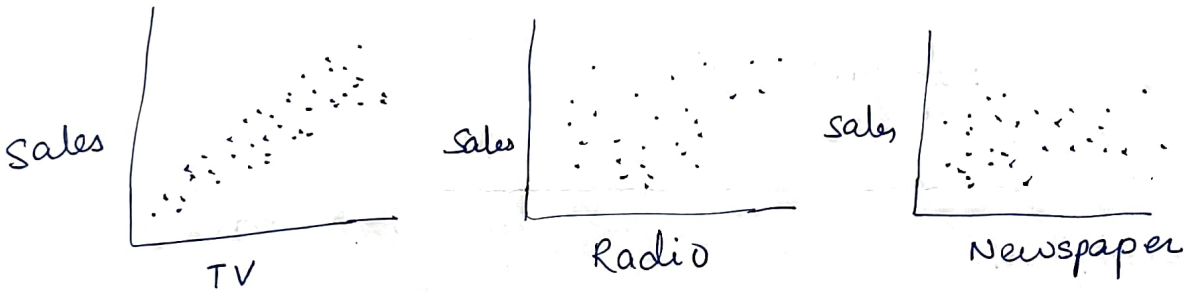
# Multiple Linear Regression

→ It represents relationship between two or more independent input variables and a response variable

→ Multiple linear regression is needed when one variable might not be sufficient to create good model and make accurate predictions

→ The objective of multiple regression is to find a linear regression equation that can best determine the value of dependent variable Y for different values of independent variables in X

⇒ eg Advertising dataset




| Predictors | $R^2$ |
|---|---|
| TV | 0.816 |
| Radio | 0.112 |
| Newspaper | 0.058 |

TV + Newspaper = 0.836

TV + Radio = 0.910

* The more information we fit to the model, the better chances are of explaining the model.

* we can check $R^2$ after adding the variables to see how much the model has improved.

Formulation of MLR

$$Y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 \text{ --- } \beta_n x_n$$

New Considerations

1. Adding more isn't always helpful
   - Model may 'overfit' by becoming too complex
     - Model fits the train set too well, doesn't generalize.
     - high train accuracy, low test accuracy
   - Multicollinearity
     - Associations between predictor variables
2. Feature selection becomes an important aspect.

Overfitting

eg we want to predict if a student will land a job interview based on his/her resume.
Assume we train a model from 10,000 resumes & their outcomes. We try the model on original dataset & it predicts 99% accuracy. but when we run the model on new (unseen) dataset of resumes it gives only 50% accuracy.

How to prevent overfitting

1. Cross validation (K-Fold)

Divides input dataset into K groups of samples of equal sizes. The samples are called folds. For each learning set, the prediction function uses K-1 folds and rest folds are used for the test set.


2. Removing irrelavant features

3. Regularization – If overfitting occurs when a model is too complex, reducing the number of features makes sense. Regularization methods like Lasso can be beneficial for removal of features.

## Multicollinearity

Thought experiment: Predictors $x_1$, $x_2$ are exactly the same. Is there a difference in effects of the following

a) $2x_1 + 8x_2$
b) $10x_1$
c) $10x_2$
d) $9x_1 + 1x_2$

} outcomes are same

If there is a change in outcome, it is difficult to identify whether this change is because of $x_1$ or $x_2$

* Don't know where the variation is coming from.
* Strong relationship b/w different variables (they will not be same in real scenario)

### Multicollinearity affects

- Interpretation - does change in Y when all all other are held constant doesn't apply.
- Inference - coefficients swing widly.

### Multicollinearity does not affect

- predictions, precision of the predictions
  * for inference we need to know from where the variation is coming from but for purely prediction it does not matter if we are correctly predicting the outcome.

### Dealing with multicollinearity

1) Drop variables that is highly correlated with others. (As other two variables bring the same info then no need the

2. Drop the variable which is not useful for business (when you need to decide which variable to drop if two variables are related to each other)

3. Create new variable if two variables are correlated to each other.

4. Transform variable
   - Principal Component Analysis (PCA)
   - Partial least square (PLS)

## Feature Selection (Subset Selection).

Suppose we have p variables, there can be $2^p$ models.

How to select features:

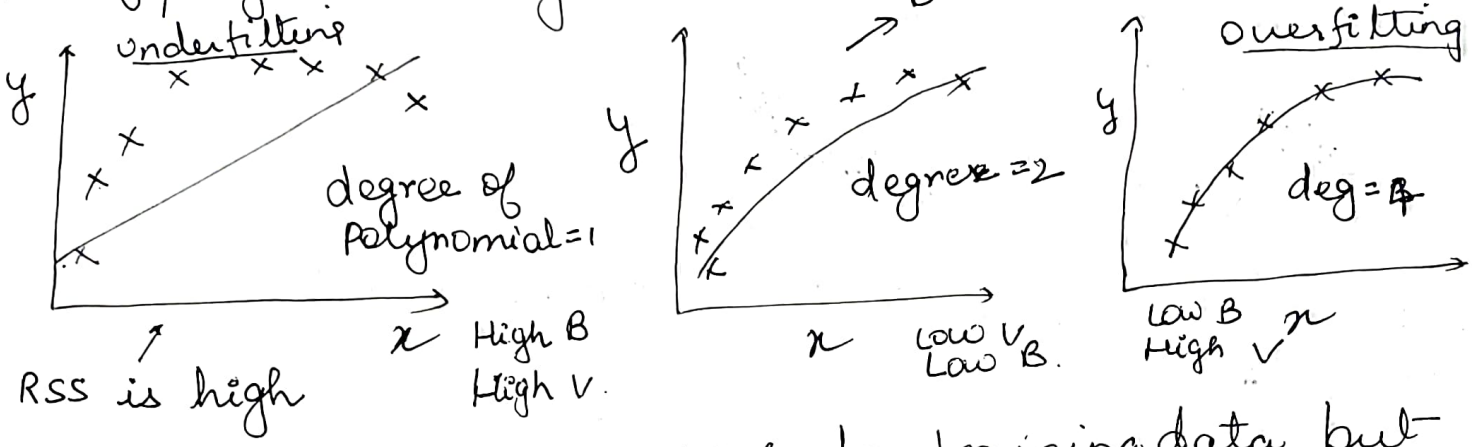
1. Manual feature elimination
   → Build model:
   → Drop features that are least helpful in prediction (high p value) → tells whether there is a relationship b/w Predictor variable & O/P var.
   → Drop features that are redundant (using correlations, VIF)
   → Rebuild model and repeat
   * with less variables (say 10/20) it is feasible.

2. Automated approach
   → Top n features select
   → Forward / Backward / Stepwise selection
   → Regularization

3. Balanced Approach
   tuning) + manual (fin
   automated (coarse
   selec

# Overfitting, Underfitting, Bias, Variance

Ex of polynomial regression.

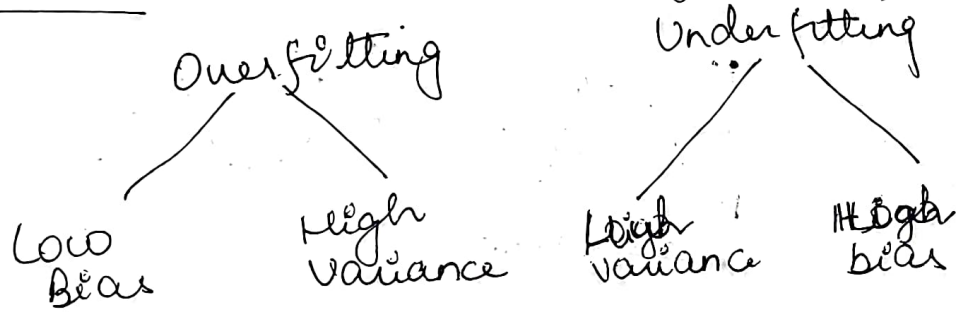



Overfitting - accuracy is high for training data but low for testing data

Underfitting - accuracy is low for training data as well as testing data

Bias - error in training data

Variance - error in testing data

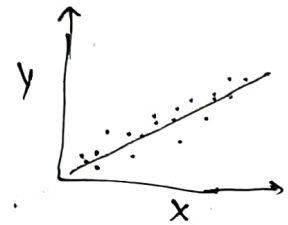



2019:
1, 4, 6, 9, 12, 15, 18, 19, 20, 22, 23, 24, 25, 27, 29, 32, 34, 37, 42, 44, 46, 51, 58, 62, 64.

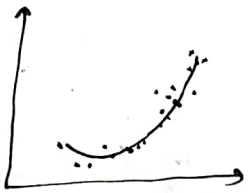
The standard linear model (OLS) perfoms poorly in a situation where we have large multivariate data set containing a number of variables. A better alternative is the penalized regression allowing to create a linear regression model that is penalized for having too many variables in the model by adding a constraint in the equation. The consequence of imposing this penalty is to shrink the coefficients values towards zero. This allows less contributive variables to have coefficient close to 0 or equal to 0.

# Assumptions of Linear Regression

1. There is a linear relationship b/w x and y

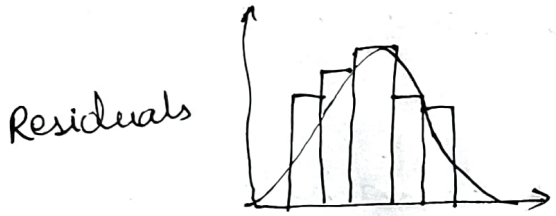



Linear

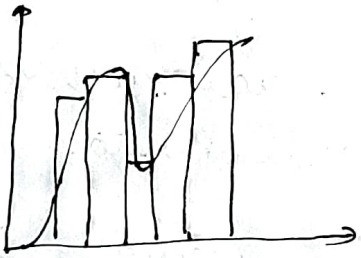


non linear

2. Error terms are distributed normally with mean equal to 0

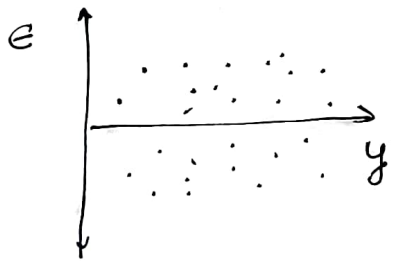



Normally distributed

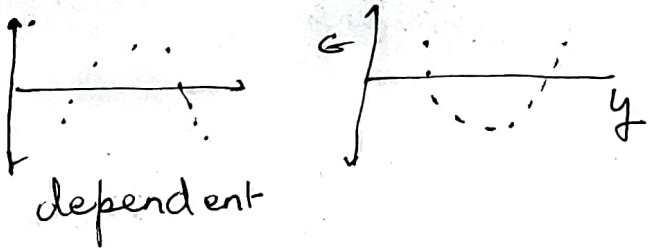


Not normally distributed

3. Error terms are independent of each other

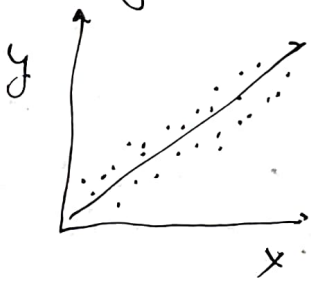



∈ independent




dependent

4. Variance should not increase or decrease with change in error values



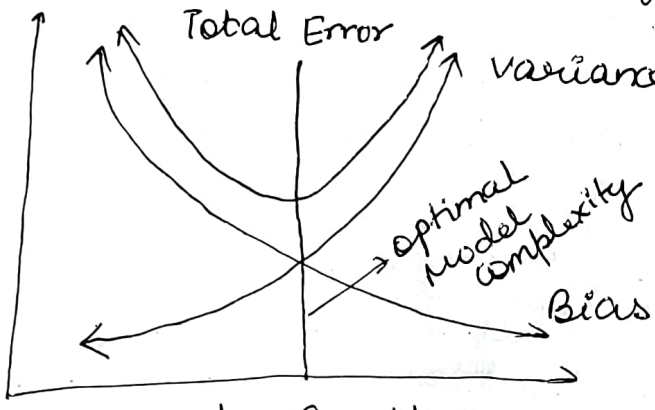

Constant variance

Changing variance

# Regularization



When a model performs really well on the data that it used to train it but does not perform well with unseen data, there is a problem of 'Overfitting'




High Bias & Low Variance - model doesn't perform well on training data (~~underfitting~~) but perform well on test data

When model is simple bias is high and variance is low. As the model becomes complex there is low bias & high variance which is the prob of ~~underfit~~ overfitting

Simple model → high bias, Low variance

complex model → Low bias, High variance

* In both cases the error would be high.
* we need a balance between bias and variance such that both are as low as possible.
* One way to manage Overfitting is Regularization

Regularization helps in managing model complexity by shrinking the model coefficient estimates towards 0. This discourages the model from becoming too complex thus avoiding risk of overfitting

$$\hat{y}_i = b_0 + b_1 x_1 + b_2 x_2$$

$$RSS = \sum_{i=1}^{n} (y - \hat{y}_i)$$

↑ cost func$^{n}$

Model complexity depends on

- Magnitude of coefficients
- Number of coefficients

→ ordinary least square

In OLS model we want to estimate the coefficients of the model for which RSS/cost is minimum. Optimizing this cost results in least possible bias though it may be overfitted

Model complexity can be taken care of by taking care of model coefficients. More extreme is the value of coefficients more complex is the model & higher are the chances of model overfitting. In order to handle this we use Regularization

Add penalty to the cost term (cost func") = RSS + penalty

(helps to shrink model coefficients to 0)

We penalize the model by adding a penalty which moves the coefficients towards 0 or they may be entirely removed from the model. This avoids model becoming complex & hence avoids risk of overfitting.

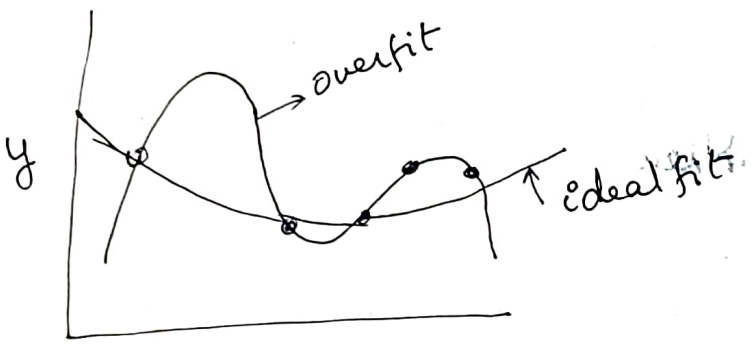
$$\text{cost func}^n = \text{RSS} + \text{penalty}$$

adding penalty will minimize the cost func

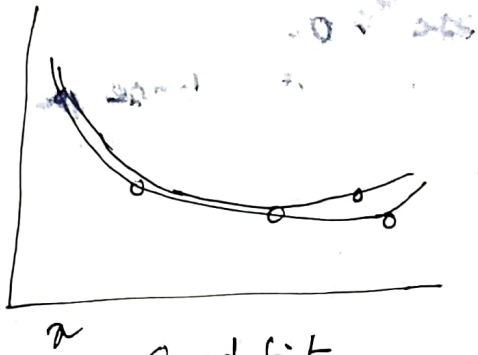
The coefficients that we get are given to training data may not be the best (more bias). Although with this minor compromise in bias, the variance may see a marked reduction

* With regularization we compromise by allowing a little bias for a significant gain in variance.

## How does fit looks after Regularization




Bad fit




Good fit
(After regularization)

When we perform regularization, it has a smoothening effect on the model fit. Curve smoothens out & the fit is close to what we want it to be.

### Points to Remember

1. We want our model to work well on unseen data without losing out on identifying the underlying patterns in the data.
2. We know that the more extreme the values of model coefficients, higher are the chances of model overfitting
3. Shrink the coefficients towards 0
4. Two techniques – Ridge and Lasso.

### Ridge Regression

For regularization we add a penalty term to the cost function such that this penalty trade bigger for a significant reduction in variance

Cost func$^n$ for OLS: $RSS = \sum_{i=1}^{n} (y_i - \hat{y}_i)^2$

Cost func$^n$ for Ridge: $cost = \sum_{i=1}^{n} (y_i - \hat{y}_i)^2 + \lambda \underbrace{\sum_{j=1}^{p} \beta_j^2}_{\text{penalty}}$

(square sum of all model coefficients)

If $\lambda = 0$ then penalty term = 0 & there will be no shrinkage of model coefficients and they would be same as those of OLS. However when $\lambda$ moves towards higher values, the shrinkage penalty increases, pushing the coeff towards 0 which may lead to ~~overfitting~~ underfitting choosing appropriate value of $\lambda$ is crucial.

$\lambda$ helps to determine how much we wish to regularize the model

$\lambda$ higher → lower value of model coeff & more regularization

$\lambda$ → tuning parameter

Disadvantage → ~~reduce all~~ include all predictors in the final model. May not affect the accuracy of the predictions but can make model interpretation challenging when no. of predictors is very large.

* No feature selection

## Lasso Regression

Primary difference between Ridge and Lasso is the penalty term

$$\text{Cost} = \sum_{i=1}^{n} (y_i - \hat{y}_i)^2 + \lambda \sum_{j=1}^{p} |\beta_j|$$

1. Like Ridge regression, the Lasso shrinks the coefficients estimates towards zero.
2. Penalty in Lasso forces some of the coefficient estimates to be exactly equal to zero - the Lasso performs variable selection.
3. selecting good value of $\lambda$ is crucial (Hyperparameter tuning)
4. As $\lambda$ increases, variance decreases & bias increases
5. variable selection results in models that are easier to interpret.