ML Session 1: Linear Regression

Till now:

- Python Basics
- Reading Data
- Properties Of Datasets
- Types of Features
- Creating Features
- Creating subsets Of Data
- Working on Data Frames
- Analysing Data (Numerically and Visually)

Linear Regression

- Supervised Model
- Simple problem
- Fits to straight line
- Creates linear Equation

Id	Age	Amount	Salary	Dependents	Sex	Children	Interest Rate
1	23	3432432	434000	2	M	2	9.6
2	54	324000	65000	3	F	0	16.5
3	32	4300000	230900	2	M	2	12

Unsupervised learning

- No target value
- Focus on grouping similar entities by Finding structures in Data
- Finding Anomaly
- Dimensionality reduction

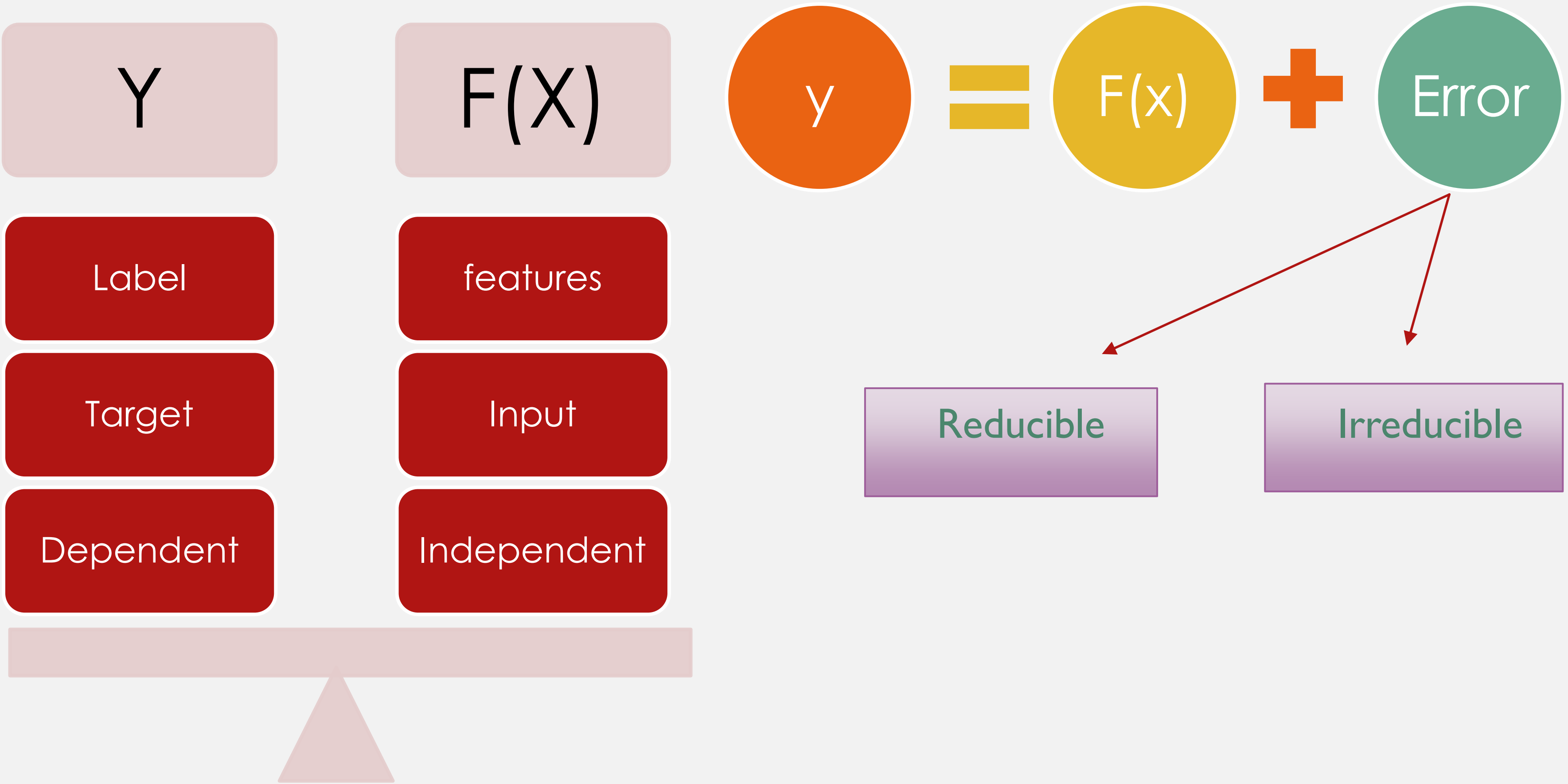
Business Problems to ML Problems

- Planning : Sales, Costs, Quality.
- Predict values : Continuous /Categorical
- Y = f(x) ← Training Model in Supervised Learning
- Yhat = X ← New Y to be predicted.

Problems

- Banks facing loss due to Loan defaulters:
- Predictors: Customer details, credit history, loan application
- Target: to find if customer will default on loan or not.
- Classification

- Flight prices keep fluctuating based on demands and holidays
- Predictors: Season, nearest holiday, day, month, Origin, Destination
- Target : revised Prices of flight ticket.
- Regression

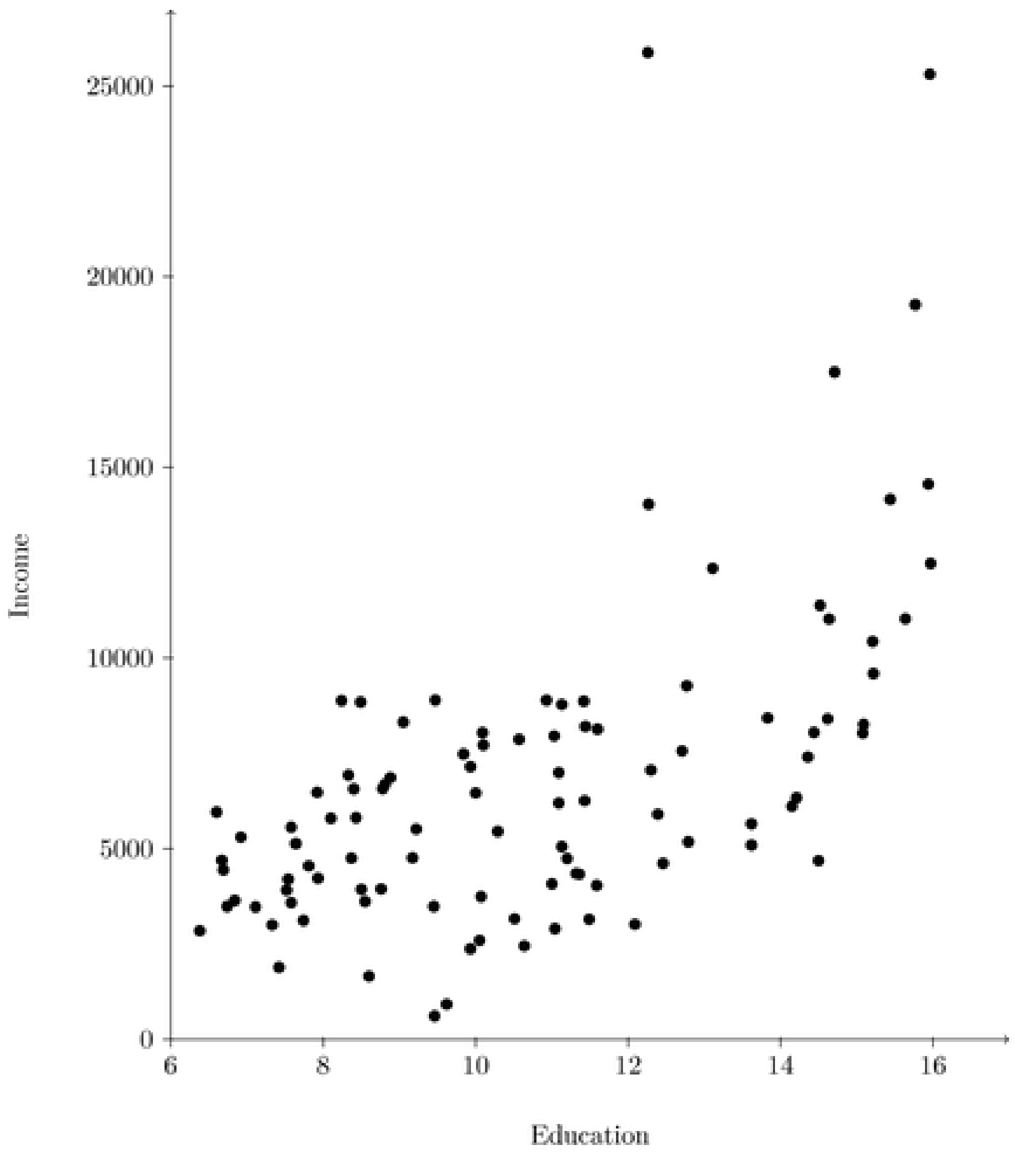


Traditional modelling Methods used to have Biased rules, while the ML models find a general relationship to reduce error while finding the unknown.

MG ANALYTICS

Now:

- Its all about Business.
- Categories Of problems
- What if we make mistakes in Predictions: Cost Functions
- Cost Functions to Parameters: Gradient Descend.



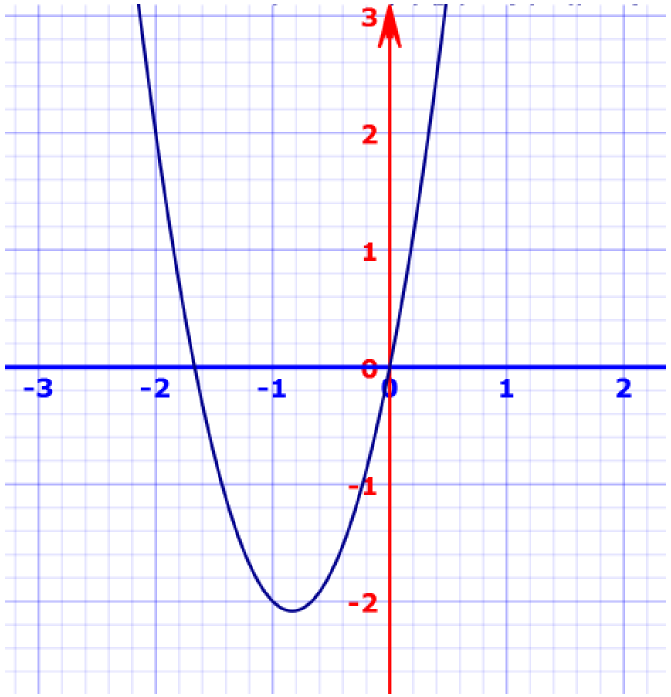
MG ANALYTICS

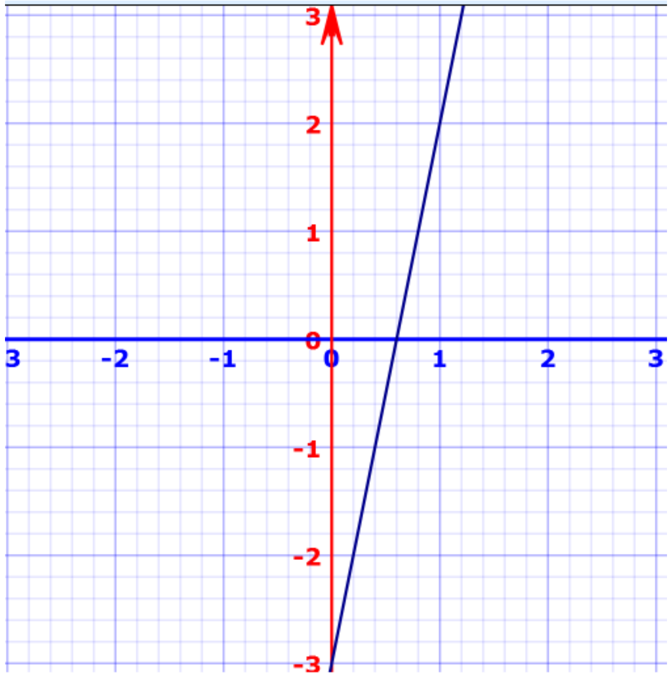
MG ANALYTICS

MG ANALYTICS

Equations : Linear, Quadratic, cubic…

- Y = mx +c
- Y = b1x1 + b2x2 +…..bnxn +b





What is a good Prediction Model

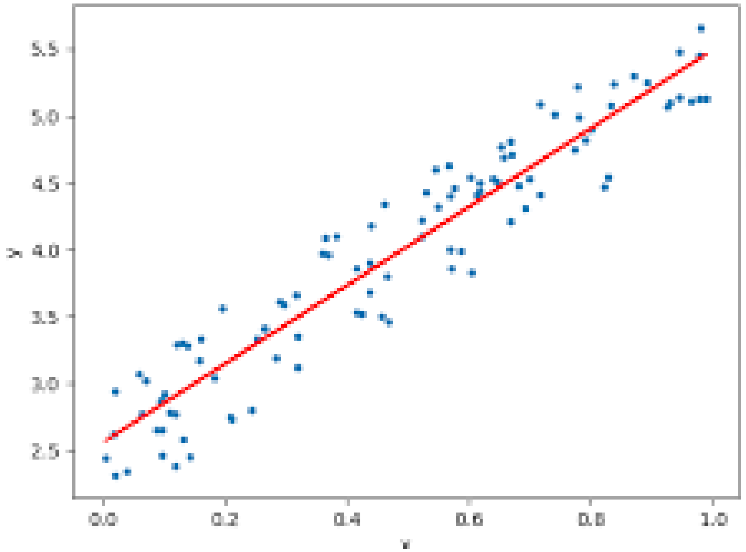
- It gives accurate results.
- The accuracy should not be limited to just one data point.
- It should perform equally well on test data as on training data.
- A simpler model is a better model complex models tend to overfit.

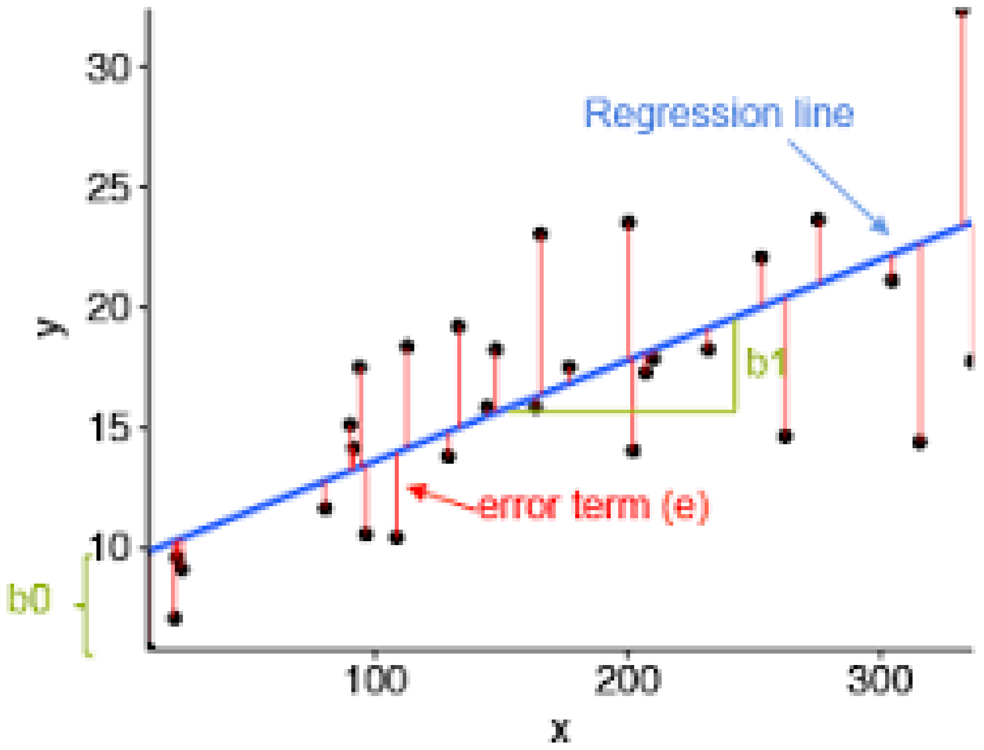
REGRESSION

- **Regression** takes a group of random variables, thought to be predicting Y, and tries to find a mathematical relationship between them.
- This relationship is typically in the form of a straight line (linear **regression**) that best approximates all the individual data points

LINEAR Regression

- **Used** for continuous target values like age, sales, interest Rate, house price
- **Expectation**: learn relationship between independent input data and the dependent target values
- **Calculate** slope and coefficients to create a linear equation which could align to given data properly.
- **Minimize** Cost (Cumulative Error for all the observations).





MG ANALYTICS

Linear Regression

- Y = f(x)
- Y = b0 + b1x1 + b2x2 +…..bnxn
- Y => **Target** : price of flight ticket   i.e. **Continuous Value**
- X1, x2, x3 ,x4…. = season, origin, destination, nearest holiday.
- Y and x are constant.
- We try to fit various beta values to bring the predictions close to actual Y values.
- What we predict is **Yhat**
- Y – Yhat = **Error**

Process:

- Convert all variables to **numeric**
- Categorical variables should be converted to **dummy** values
- Remove outlier data
- Impute missing values
- Transformations made to train data should also reflect in test data.
- Metrics : RMSE , MAE
- Goodness of predictions would depend on the scale of train target values.

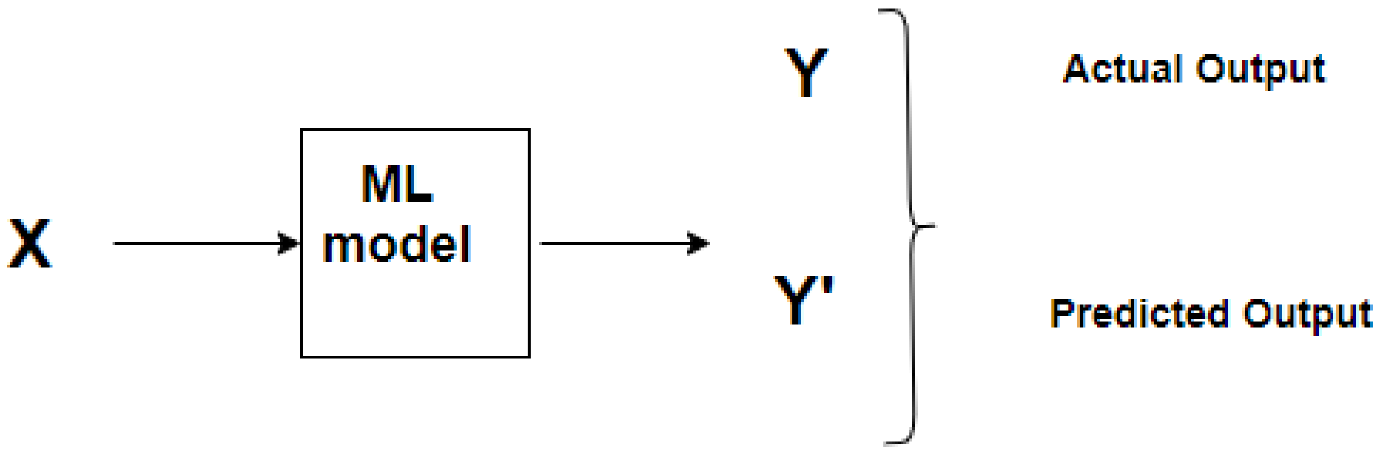
Errors:

Actual price	Predicted price	Error	Absolute Error
1000	1120	120	120
980	1000	20	20
950	1100	50	50
1020	960	-80	80
1050	940	-100	100
		10	370

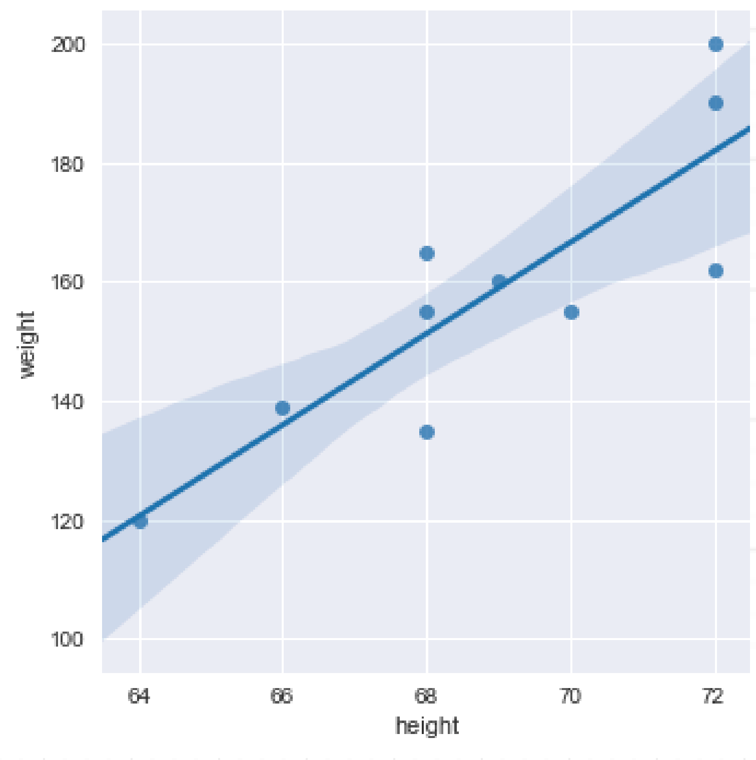
- Our Main focus in Predictive analysis is to Reduce the overall Error.

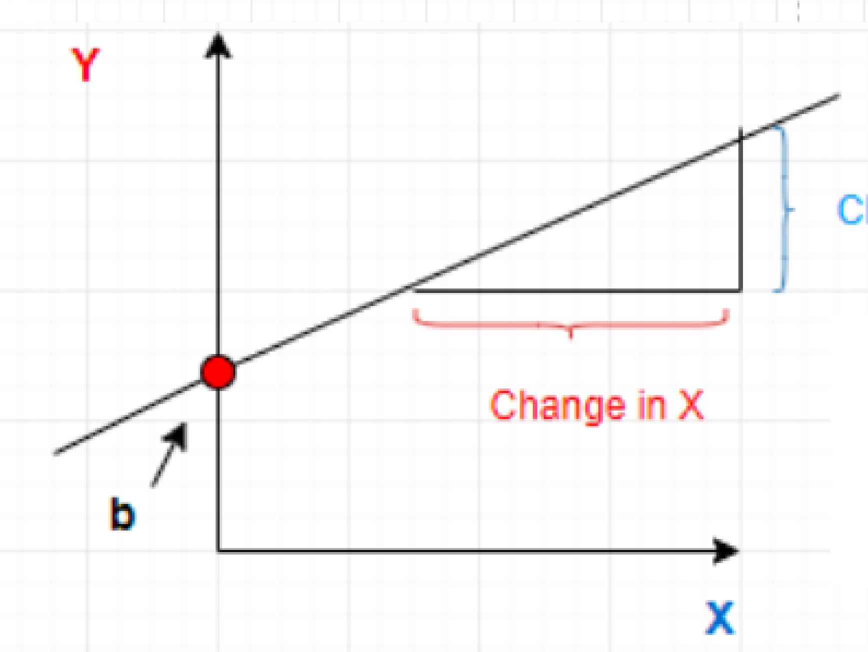
$$\hat{y_i} = f(X_i)$$
$$y_i = \hat{y_i} + \epsilon_i$$

- We need to see how weights in Equation impact errors.



Error = Y'(Predicted) - Y(Actual)

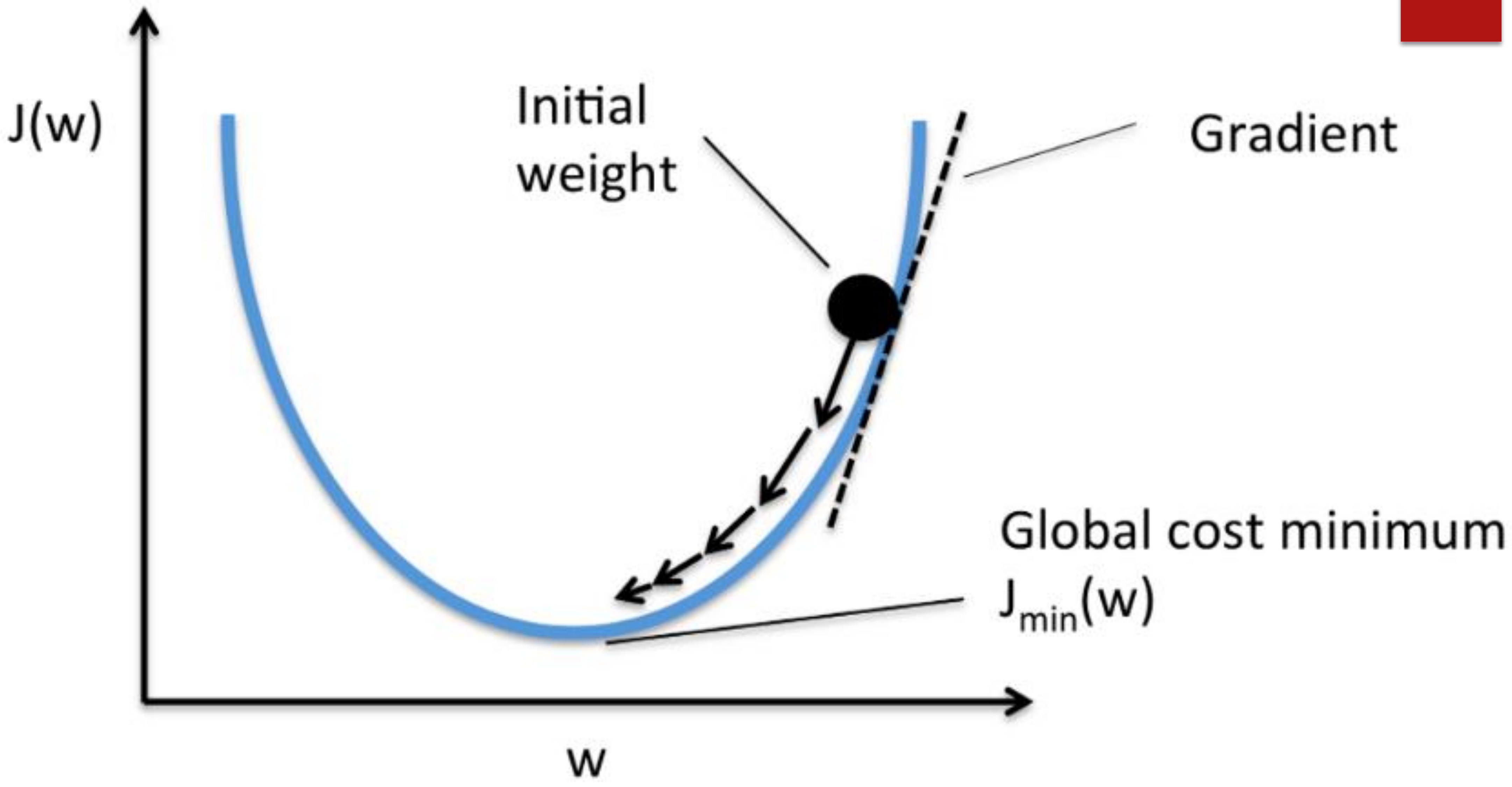




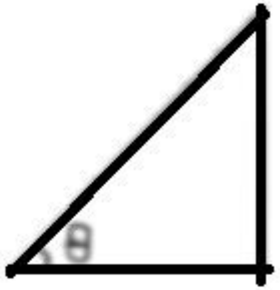
$$\text{Slope} = \frac{\text{Change in Y}}{\text{Change in X}}$$

Cost and Gradient Descent

- **Cost function** is something we want to minimize. For example, our **cost function** might be the sum of squared errors over the training set.
- **Gradient descent** is a method for finding the minimum of a **function** of multiple variables. So we can use **gradient descent** as a tool to minimize our **cost function**.



MG ANALYTICS



$$\tan(\theta) = \frac{\Delta f}{\Delta \beta}$$

$$\frac{\delta f}{\delta \beta} = \frac{\Delta f}{\Delta \beta}$$

$$\Delta f = \frac{\delta f}{\delta \beta} * \Delta \beta$$

MG ANALYTICS

$$f(X_i) = \beta_0 + \beta_1 X_{1i} + \beta_2 X_{2i} \ldots\ldots$$

$$\text{Error} = y_i - \hat{y}_i$$

$$\text{Error} = y_i - (\beta_0 + \beta_1 X_{1i} + \beta_2 X_{2i})$$

$$\text{Cost} = \sum_{i=1}^{n} \text{Error}_i^{\,2}$$

$$\sum_{i=1}^{n} |\text{Error}_i|$$

MG ANALYTICS

- Cost is determined by Beta values as Y and X values are constant, hence we try to find Beta values such that Cost is minimum.

$$X = \begin{bmatrix} 1 & x_{11} & x_{21} & x_{31} & \dots & x_{p1} \\ 1 & x_{12} & x_{22} & x_{32} & \dots & x_{p2} \\ \dots & \dots & \dots & \dots & \dots & \\ 1 & x_{1n} & x_{2n} & x_{3n} & \dots & x_{pn} \end{bmatrix}$$

$$Y = \begin{bmatrix} y_1 \\ y_2 \\ .. \\ y_n \end{bmatrix}$$

$$\beta = \begin{bmatrix} \beta_0 \\ \beta_1 \\ .. \\ \beta_p \end{bmatrix}$$

$$\sum_{i=1}^{n} \epsilon_i{}^2 = \sum_{i=1}^{n} (y_i - \beta_0 - \beta_1 X_{1i} - \beta_2 X_{2i})^2$$

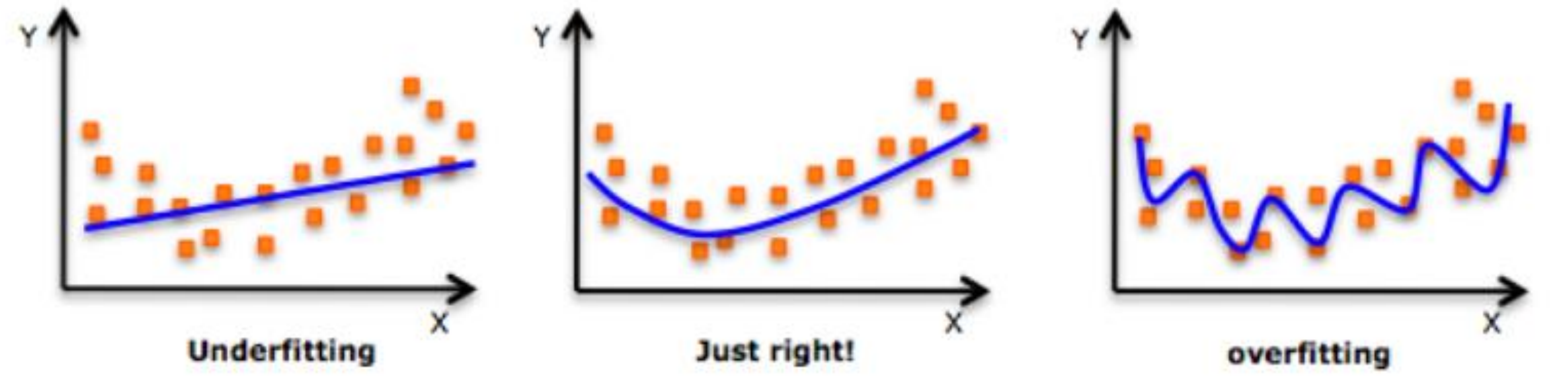
$$Cost = \frac{1}{N}\sum_{i=1}^{N}(Y' - Y)^2$$

Update rules:

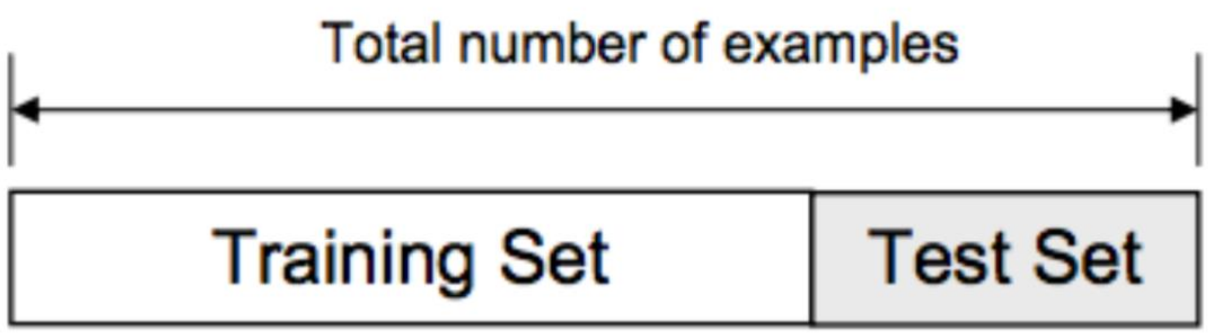
W – ∂SSE/∂w

So, update rules:

1.New w = w – r * **∂SSE/∂w**



An example of overfitting, underfitting and a model that's "just right!"

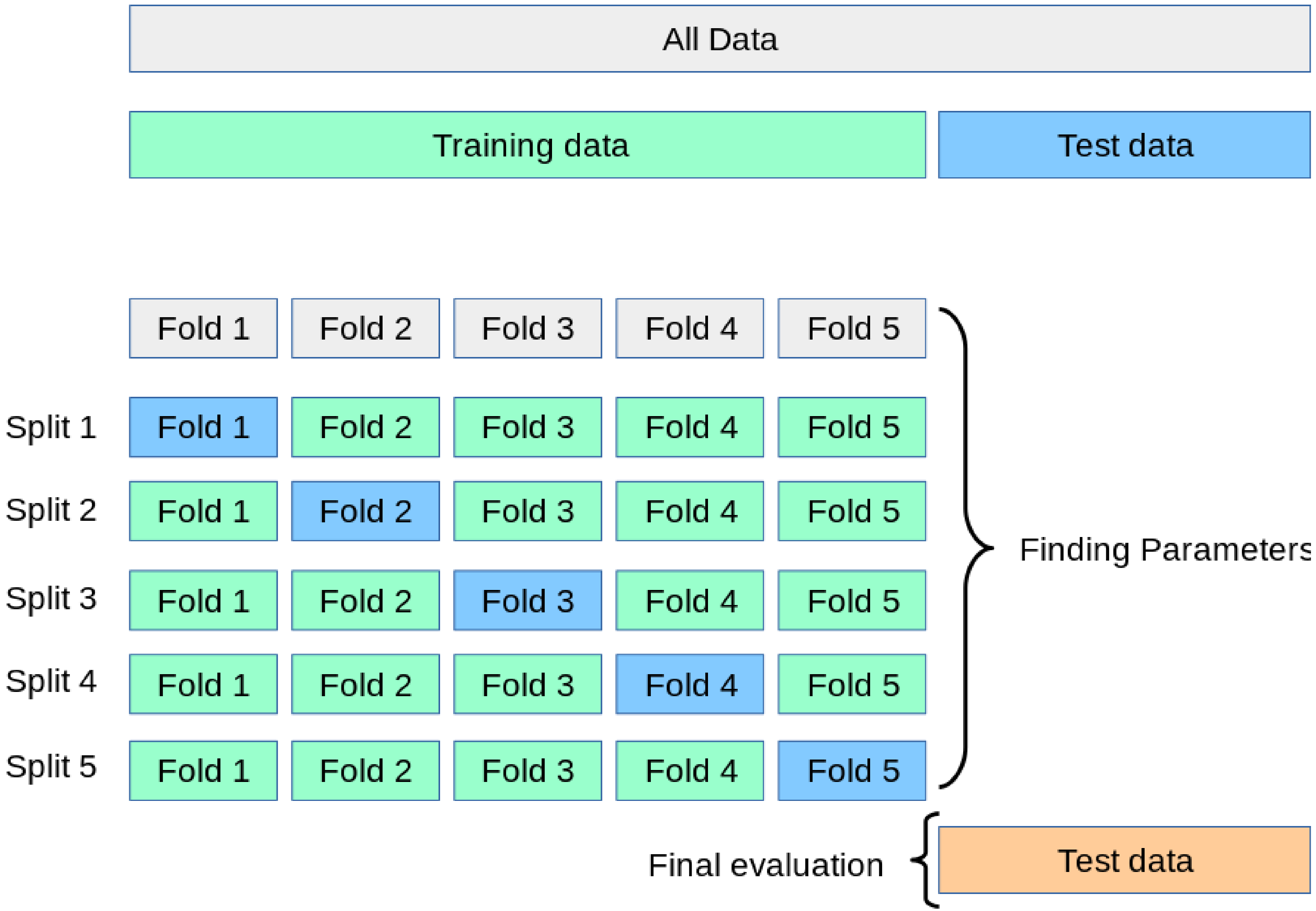


MG ANALYTICS

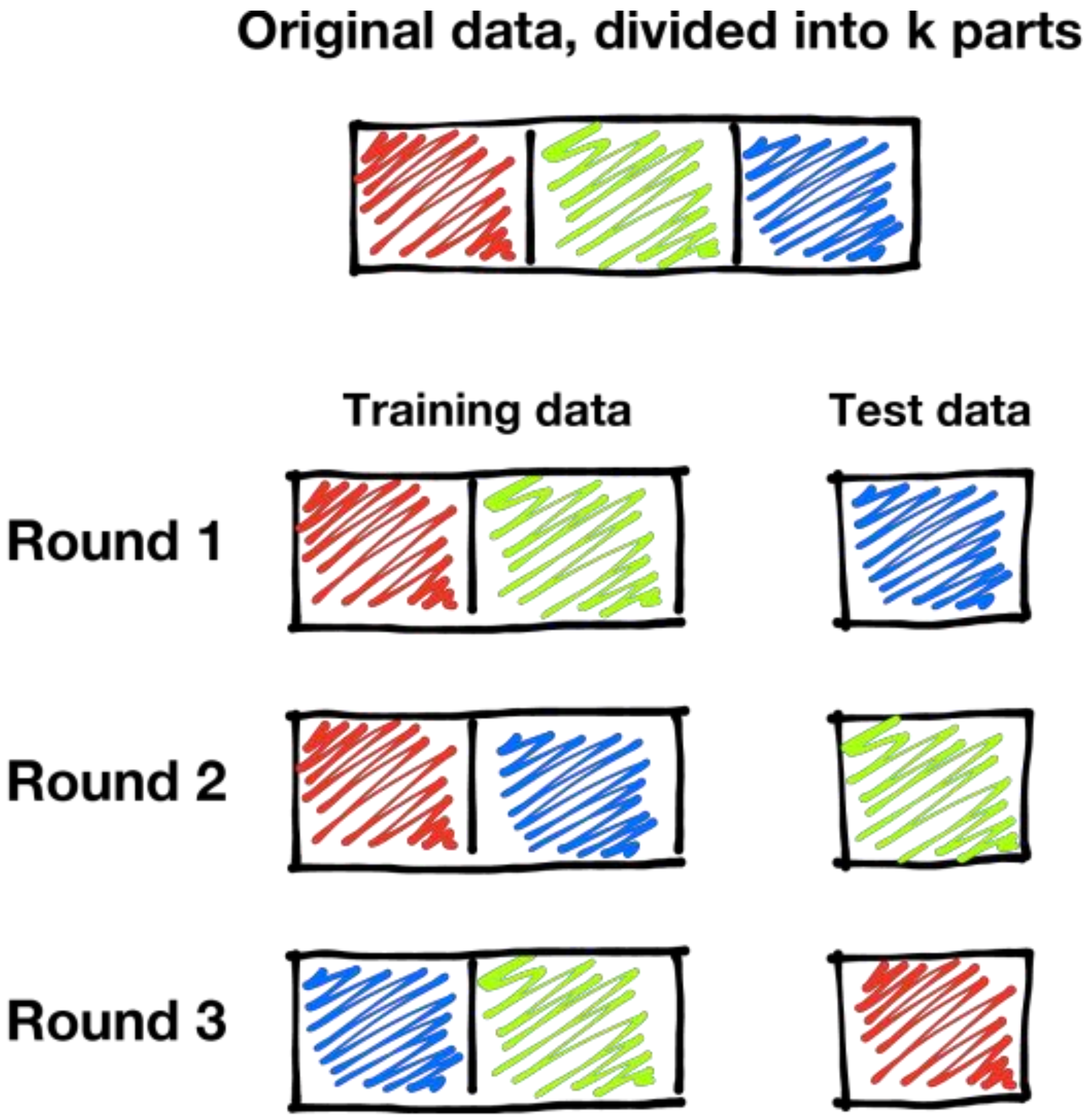
Cross Validation – Holdout Method | Train Test Split

- The holdout method is the simplest kind of cross validation. The data set is separated into two sets, called the training set and the testing set.
- The model is built using the training set only.

- Dis-Advantages:
- The evaluation may depend heavily on which data points end up in the training set and which end up in the test set.
- Thus the evaluation may be significantly different depending on how the division is made.



MG ANALYTICS



MG ANALYTICS

Regularization

- *The model will have a low accuracy if it is overfitting.*
- Model tries to capture the noise in the training dataset.
- ***Noise : the data points that don’t really represent the true properties of the data, but random chance*.**
- Learning such data points, makes the model more flexible, at the risk of overfitting.
- Regularization shrinks the coefficient estimates towards zero.
- ***Discourages learning a more complex or flexible model, so as to avoid the risk of overfitting.***

Ridge Regression

$$\sum_{i=1}^{n}\left(y_i-\beta_0-\sum_{j=1}^{p}\beta_j x_{ij}\right)^2+\lambda\sum_{j=1}^{p}\beta_j^2=\mathrm{RSS}+\lambda\sum_{j=1}^{p}\beta_j^2$$

Lasso Regression

$$\sum_{i=1}^{n}\left(y_i-\beta_0-\sum_{j=1}^{p}\beta_j x_{ij}\right)^2+\lambda\sum_{j=1}^{p}|\beta_j| =\mathrm{RSS}+\lambda\sum_{j=1}^{p}|\beta_j|.$$

MG ANALYTICS

Comparison

- Ridge shrinks the coefficients for least important predictors, very close to zero.
  - It will never make them exactly zero.
  - Final model will include all predictors.
- Lasso, has the effect of forcing some of the coefficient estimates to be exactly equal to zero when the tuning parameter λ is sufficiently large.
  - **the lasso method also performs variable selection.**