



# Dealing with Big Data and Machine Learning – Introduction

## Part2

**AI-Amal Program**  
**Amr.Moslim**

Jan.2022



# Today's Agenda



- What is Machine Learning?
- Machine Learning Vs Coding ?
- Machine Learning Algorithm Classes
- Machine Learning Algorithm
- Machine Learning Classifications
- Machine Learning Workflow
- How does Machine Learning work?
- Machine Learning Models Evaluation
- Machine Learning Algorithms
- Machine Learning Applications in Geoscience
- The Road map for start learning Machine Learning

# What's Machine Learning?



Machine learning is the field of **AI** that allows systems to learn from **past data** and make **intelligent decisions** on their own using **algorithms** without **explicitly** programed and **improve** its experience

# Machine Learning vs Coding



Characteristics	Machine Learning Algorithms	Common coding
Objective	To teach the machine to create models to solve the problem without hard coding using data patterns	To use programming language to explicitly code the solution to the problem
Example: $v = d/t$	<pre>Data = (mass, height, width, velocity) Lm = linearregression() Lm.fit() Lm.predict()</pre>	<pre>Data = (d, t) def velocity(d,t):     v = d/t     return (v)</pre>
Tools	Python, R, Scikit learn, Tensorflow, etc...	Python, R, Visual Basic, Java, Go, Excel
Running time	Most of time in data wrangling and model evaluation	Most of the time in coding the problem and solution
Output	ML model and forecast	Data table, graphs, dashboards
Reproducibility	Yes with the same data formats	Yes with the same data formats
Domain knowledge	It is very important and highly recommended	a must

# Machine Learning Algorithm Classes



## Supervised Learning

Labeled data

## Unsupervised Learning

unlabeled data

- ***Data has labels (reference) model should learn.***
- ***Model should be continuously test based on the label prediction or classification.***

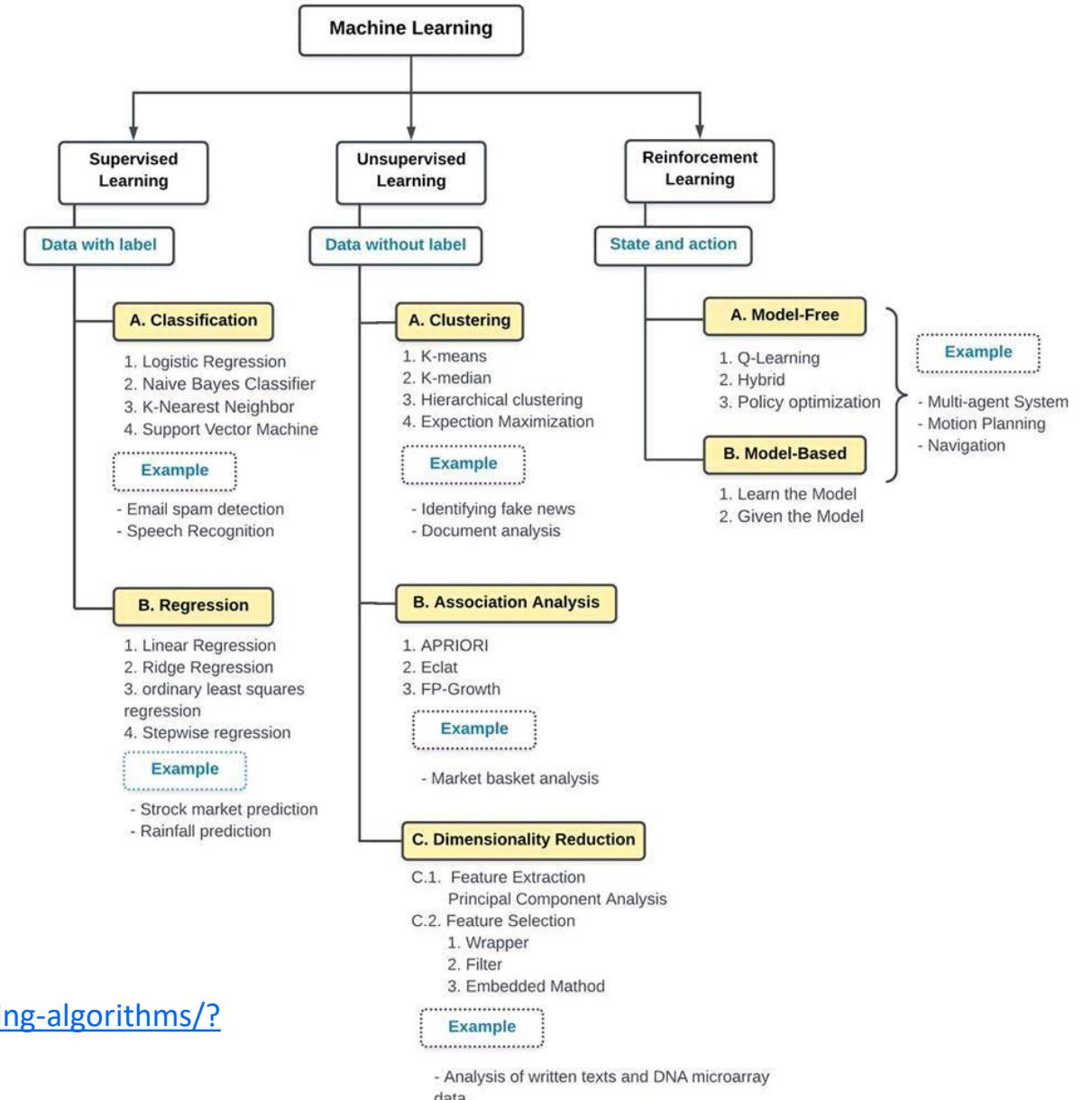
- ***Data has NO labels. Data learn from itself.***
- ***Model should be judged based on certain criteria.***

# Machine Learning Algorithms



Most commonly used Machine learning algorithms:

1. Linear Regression
2. Logistic Regression
3. Decision Tree
4. SVM
5. Naive Bayes
6. kNN
7. K-Means
8. Random Forest
9. Dimensionality Reduction Algorithms PCA
10. Gradient Boosting algorithms
  1. GBM
  2. XGBoost
  3. LightGBM
  4. CatBoost



<https://www.analyticsvidhya.com/blog/2017/09/common-machine-learning-algorithms/?>

# Machine Learning Algorithm Classification



## Supervised Learning

Labeled data prediction

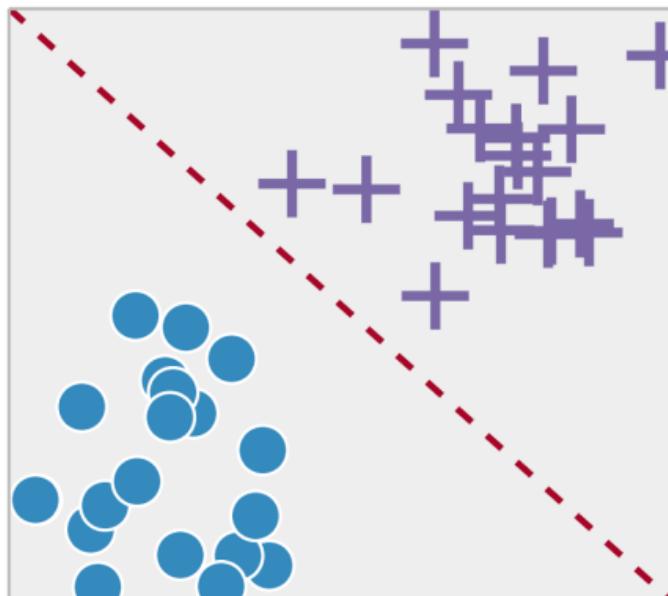
- Regression
- Classification

## Unsupervised Learning

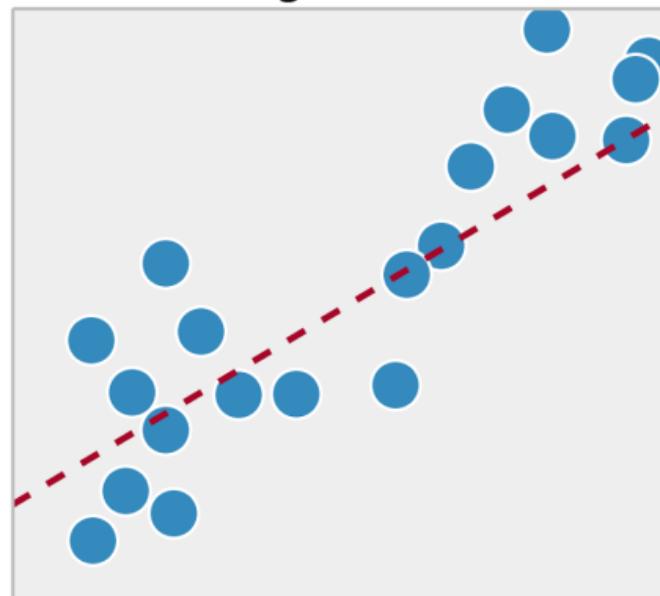
unlabeled data

- Dimensionality reduction
- Clustering

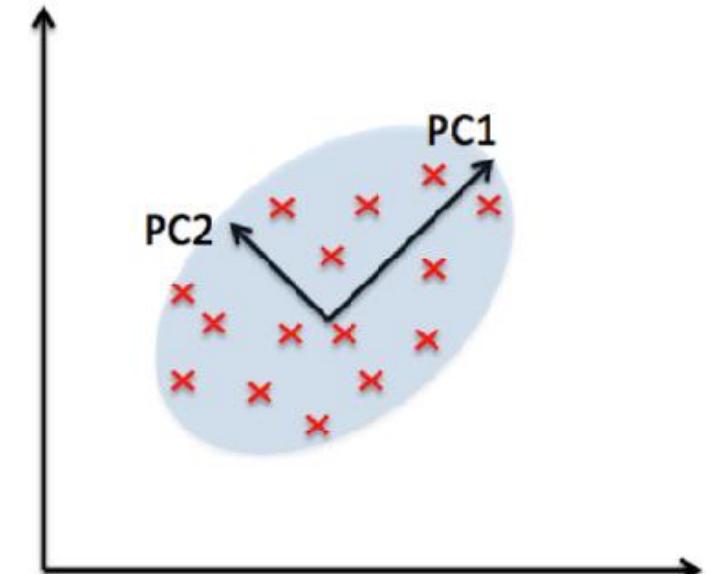
Classification



Regression



Dimensionality reduction



# Machine Learning Algorithm Classification



## Supervised Learning

Labeled data prediction

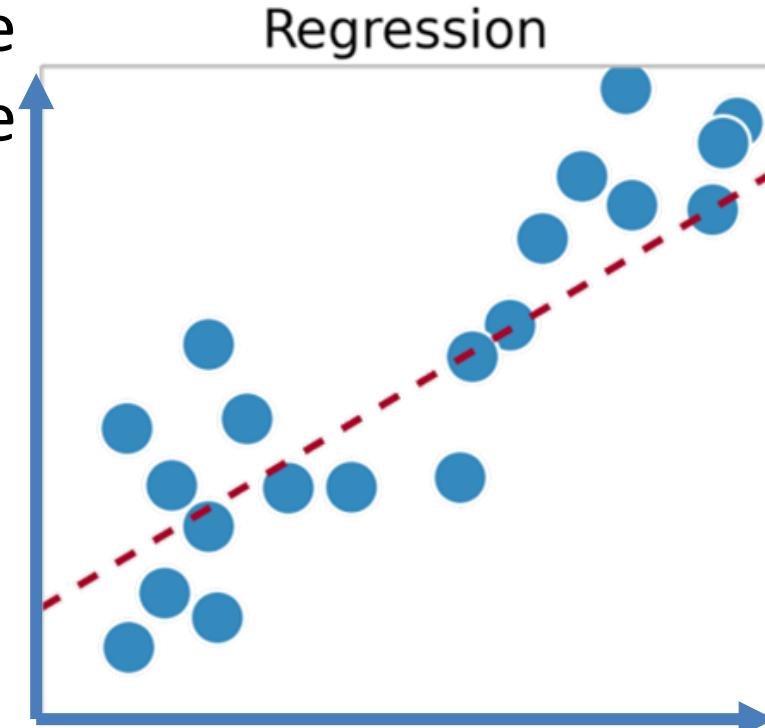
- Regression
- Classification

### Regression:

is a set of statistical processes for estimating the relationships between a dependent variable and one or more independent variables.

- Statistical Modeling Technique
- Types (Linear, Logistic, Polynomial, ...)
- Data is numerical values (Not Categorical)

Example : missing logs predication



# Machine Learning Algorithm Classification



## Supervised Learning

Labeled data prediction

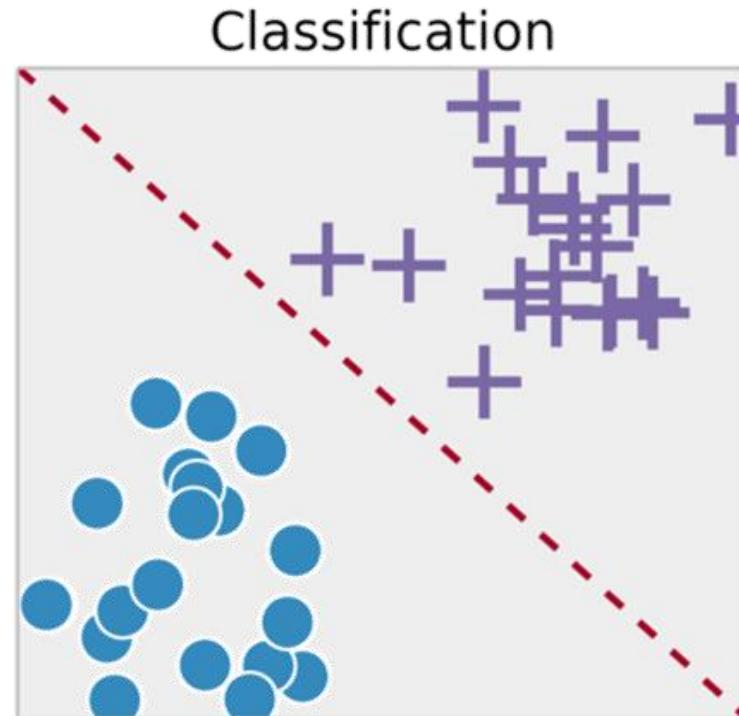
- Regression
- Classification

## Classification: (Categorization)

systematic arrangement in groups or categories  
according to established criteria

- Uses predefined classes
- Belongs to which class

Example : Fraud Detection (Spam / No Spam)  
Facies Classification



# Machine Learning Algorithm Classification



## Unsupervised Learning

unlabeled data

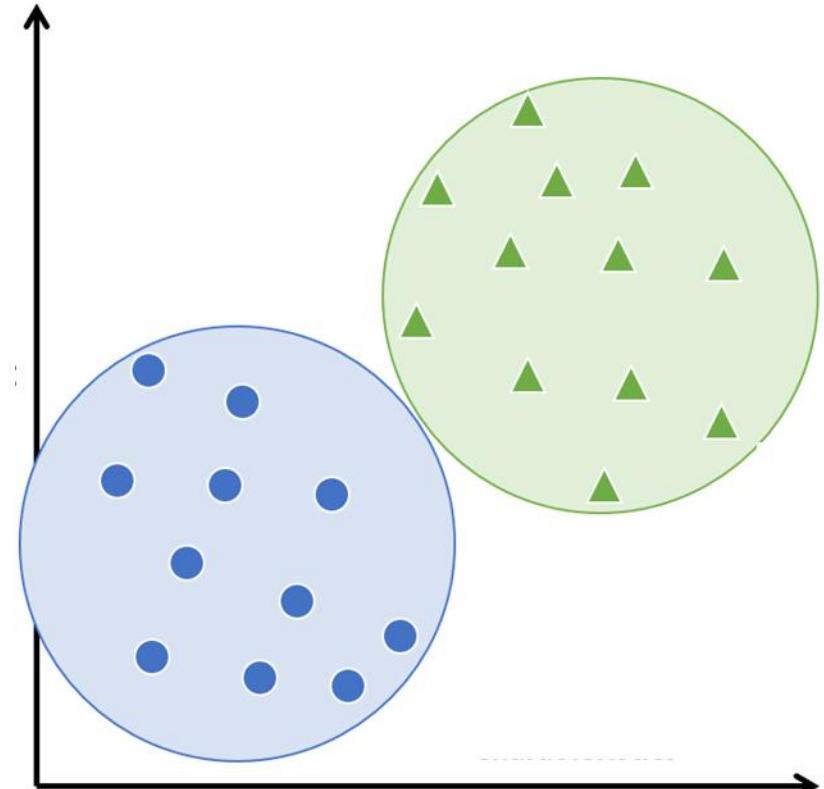
- Dimensionality reduction
- Clustering

### Clustering:

identifies similarities between objects, which it groups according to those characteristics in common and which differentiate them from other groups of objects. These groups are known as "clusters".

- NO predefined classes
- Similar data points properties clusters together

Example : Customer Segmentation  
Facies Classification (first time ☺)



# Machine Learning Algorithm Classification



## Unsupervised Learning

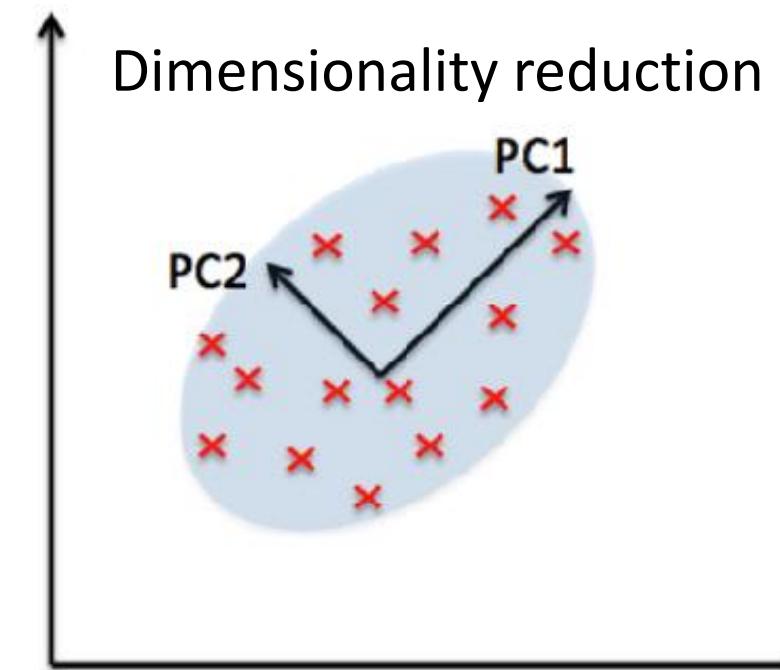
unlabeled data

- Dimensionality reduction
- Clustering

### Dimensionality Reduction:

Analyzing the datasets with an extremely high number of features is often performed to obtain better input features for machine learning algorithms.

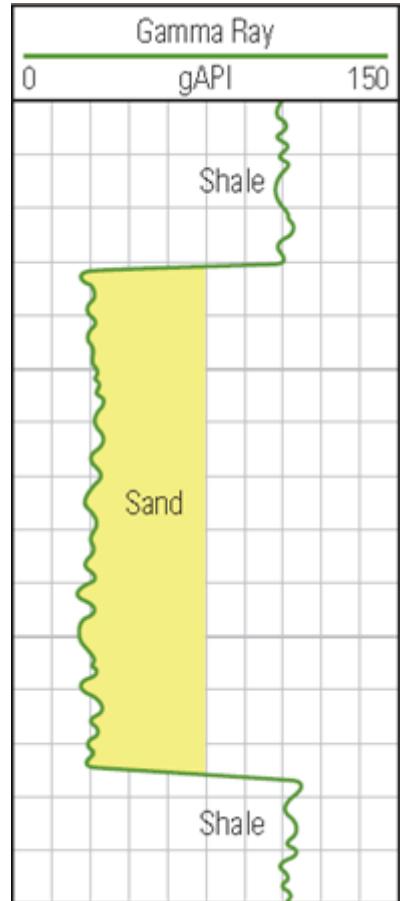
- It improves computational efficiency without sacrificing much on the prediction capability
- removes the collinearity



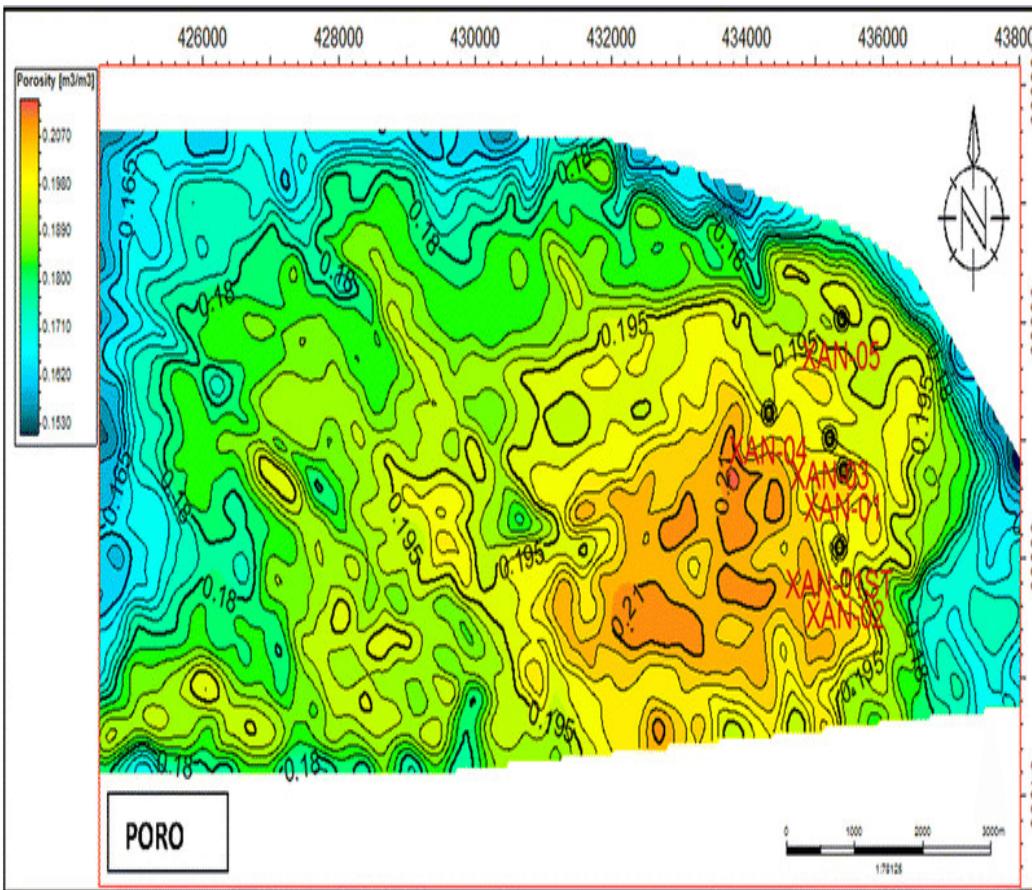
# Machine Learning Algorithm Classification



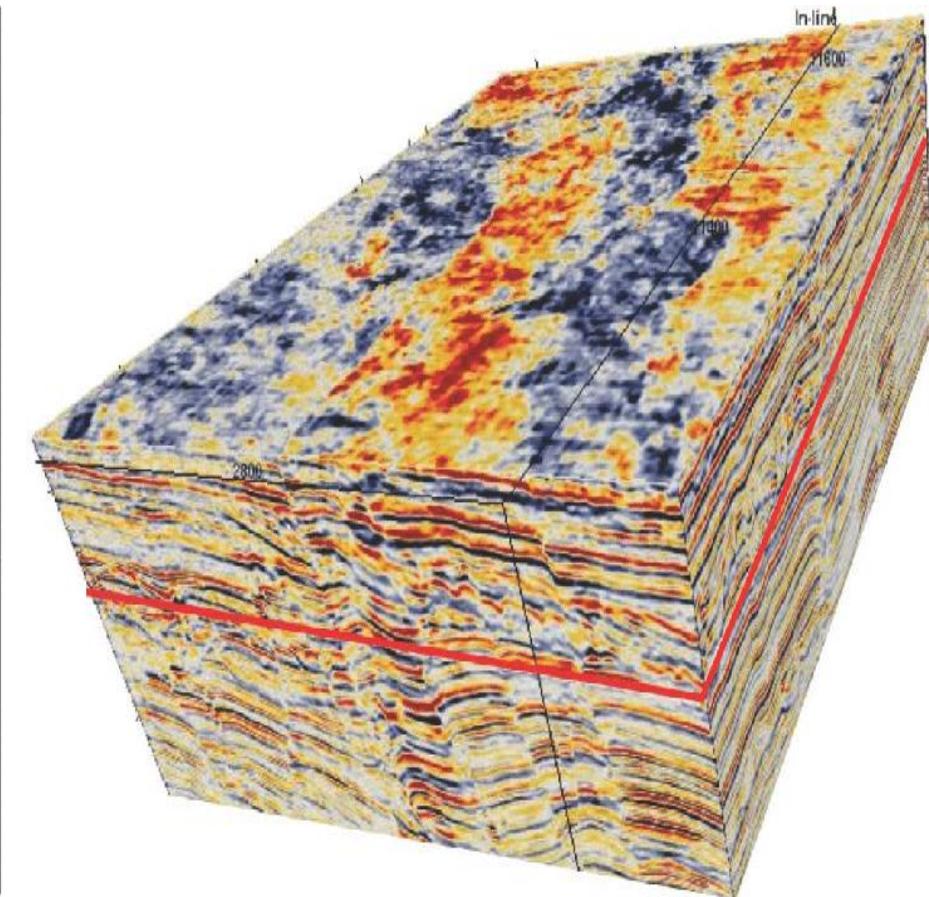
**1D Graph**



**2D Maps**



**3D Cubes**



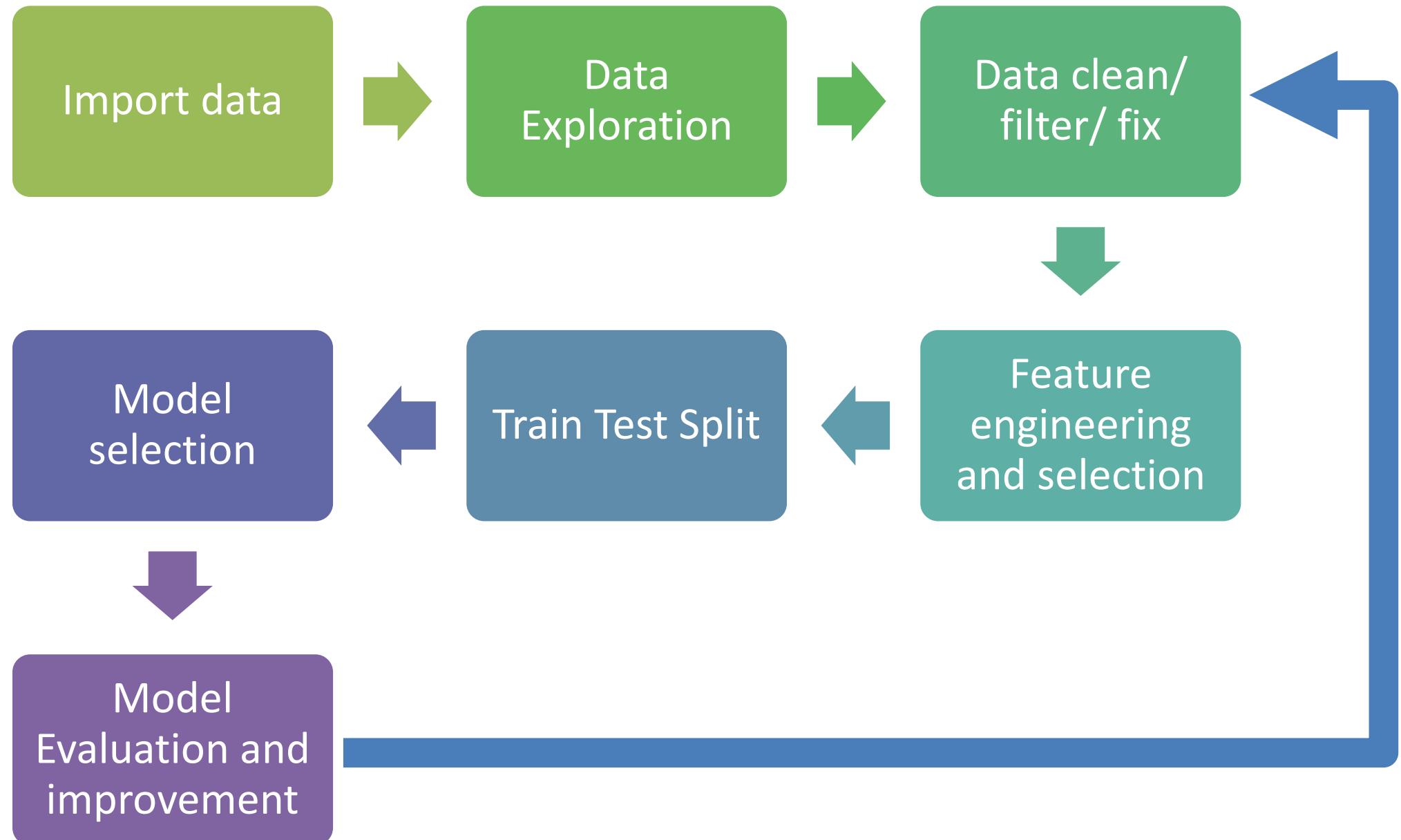
*The curse of dimensionality*

[Richard E. Bellman](#)

---

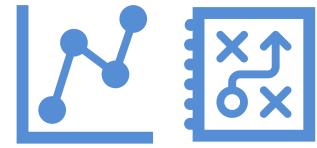
# **MACHINE LEARNING WORKFLOW**

# Machine Learning Work flow





# Data Conditioning





- **Normalization:**

usually means to scale a variable to have values between 0 and 1

$$X_{norm} = \frac{X - X_{min}}{X_{max} - X_{min}}$$

- **Standardization:**

transforms data to have a mean of zero and a standard deviation of 1. not necessary to be between (0,1)

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

$x$  is a data point ( $x_1, x_2 \dots x_n$ ).

$\mu$  is the sample mean.

$\sigma$  is the sample standard deviation.

# Data Conditioning



- **One Hot Encoding :**

GR	DT	Rho	Facies
57	105	2.2	Sand
110	53	2.11	Shale
40	59	2.7	Lime

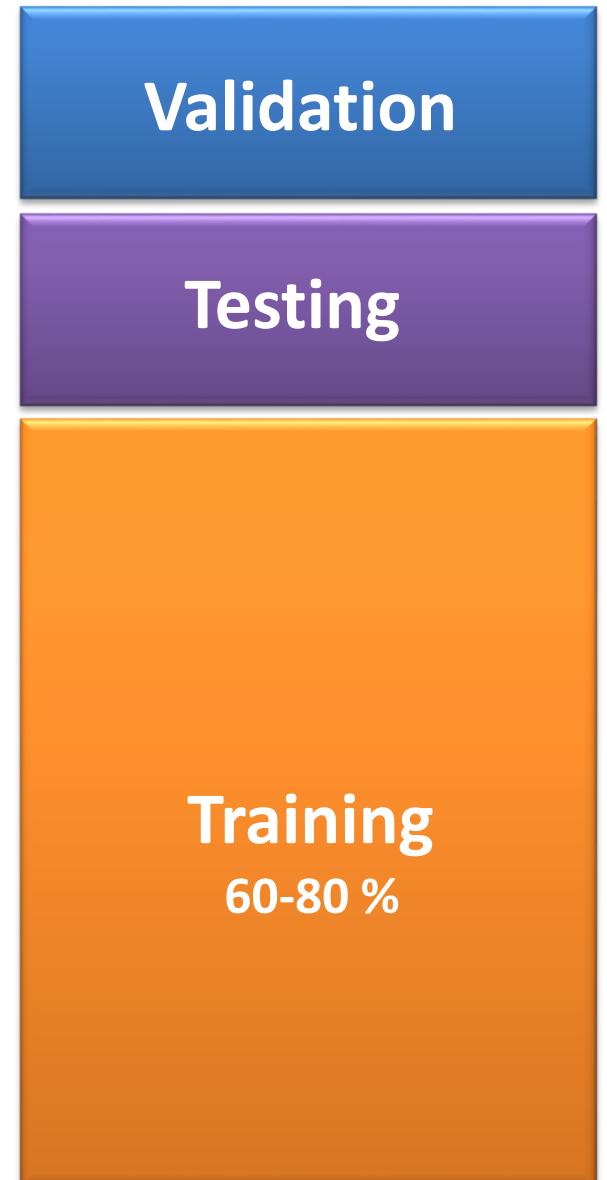
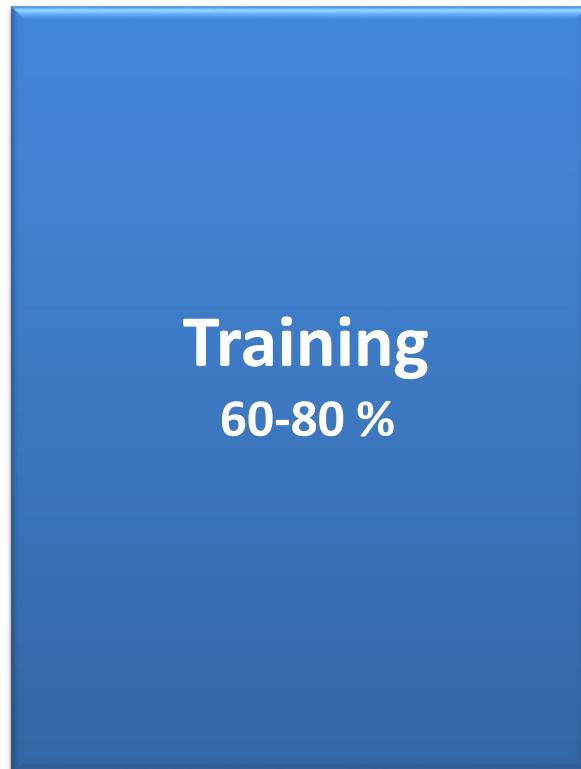


GR	DT	Rho	Sand	Shale	Lime
57	105	2.2	1	0	0
110	53	2.11	0	1	0
40	59	2.7	0	0	1

- **Missing Values Handling :**

- Remove data observation
- Replace Missing data with mean value
- Replace Missing data with value in-between previous and next value

# Train – Test - Split



---

# **ML MODELS EVALUATION**



## Cost Function:

“It is a function that measures the performance of a model for any given data. Cost Function quantifies the error between predicted values and expected values and presents it in the form of a single real number”

Types of Cost functions:

- **MSE**

$$\text{MSE} = \frac{1}{n} \sum_{i=1}^n (Y_i - \hat{Y}_i)^2$$

- **RMSE**

$$\text{RMSE} = \sqrt{\frac{\sum_{i=1}^N (x_i - \hat{x}_i)^2}{N}}$$

- **R2**

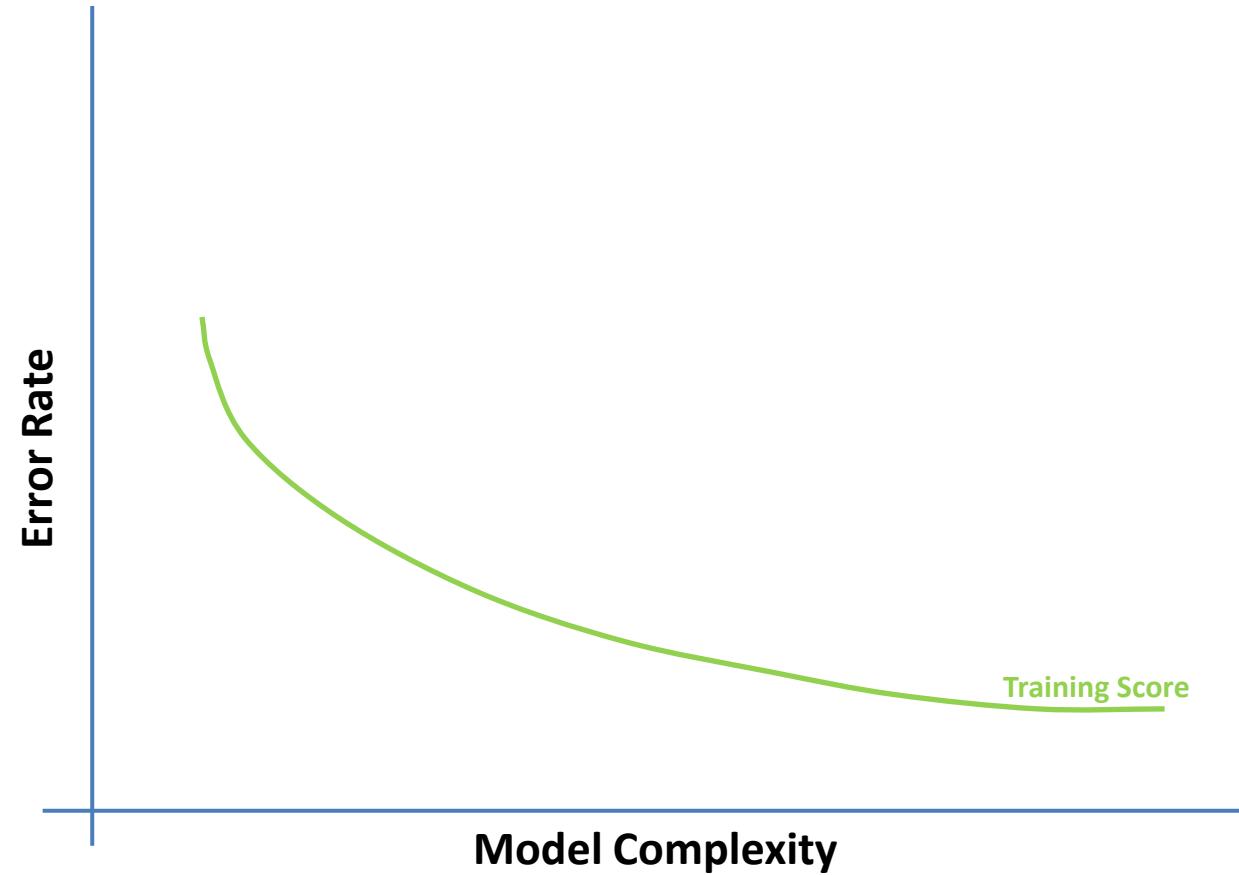
$$R^2 = 1 - \frac{RSS}{TSS}$$

$R^2$  = coefficient of determination

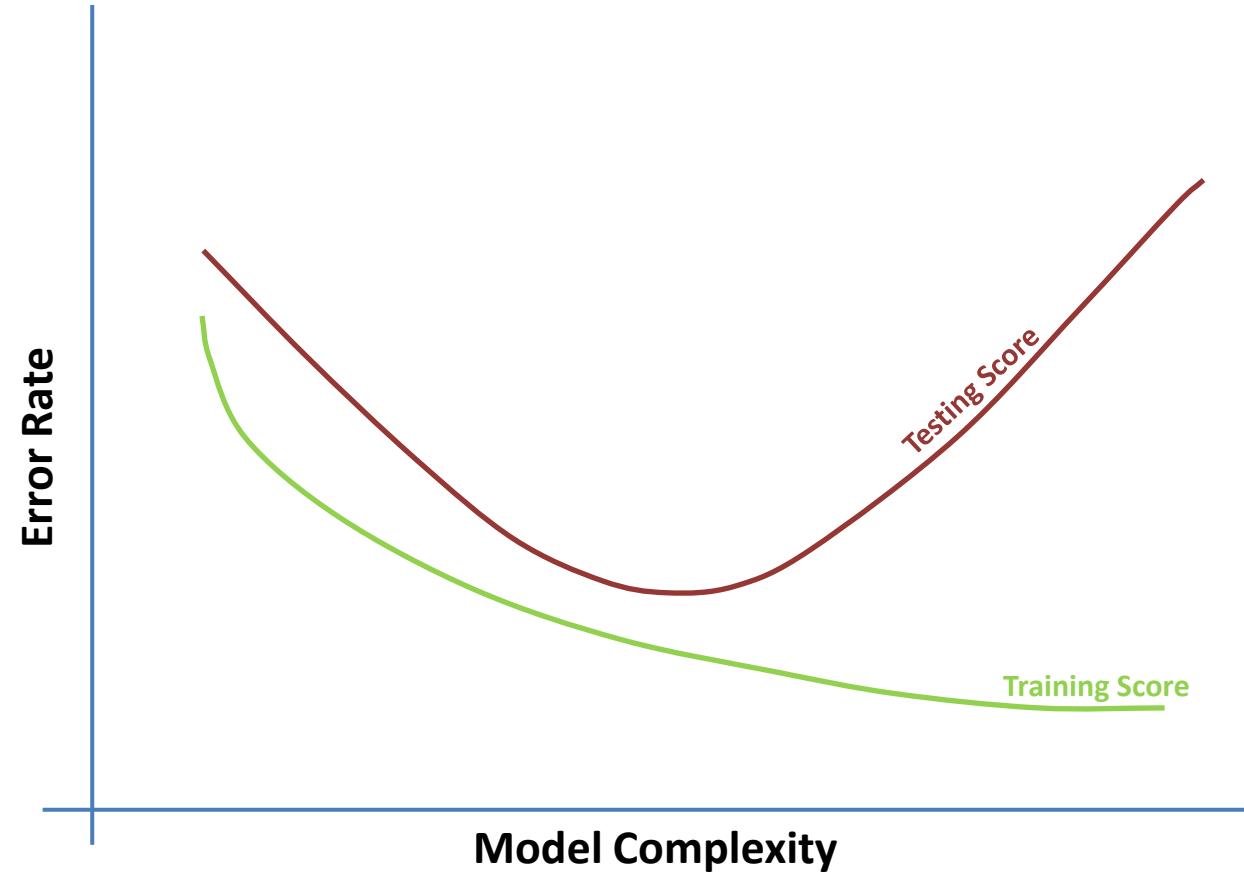
$RSS$  = sum of squares of residuals

$TSS$  = total sum of squares

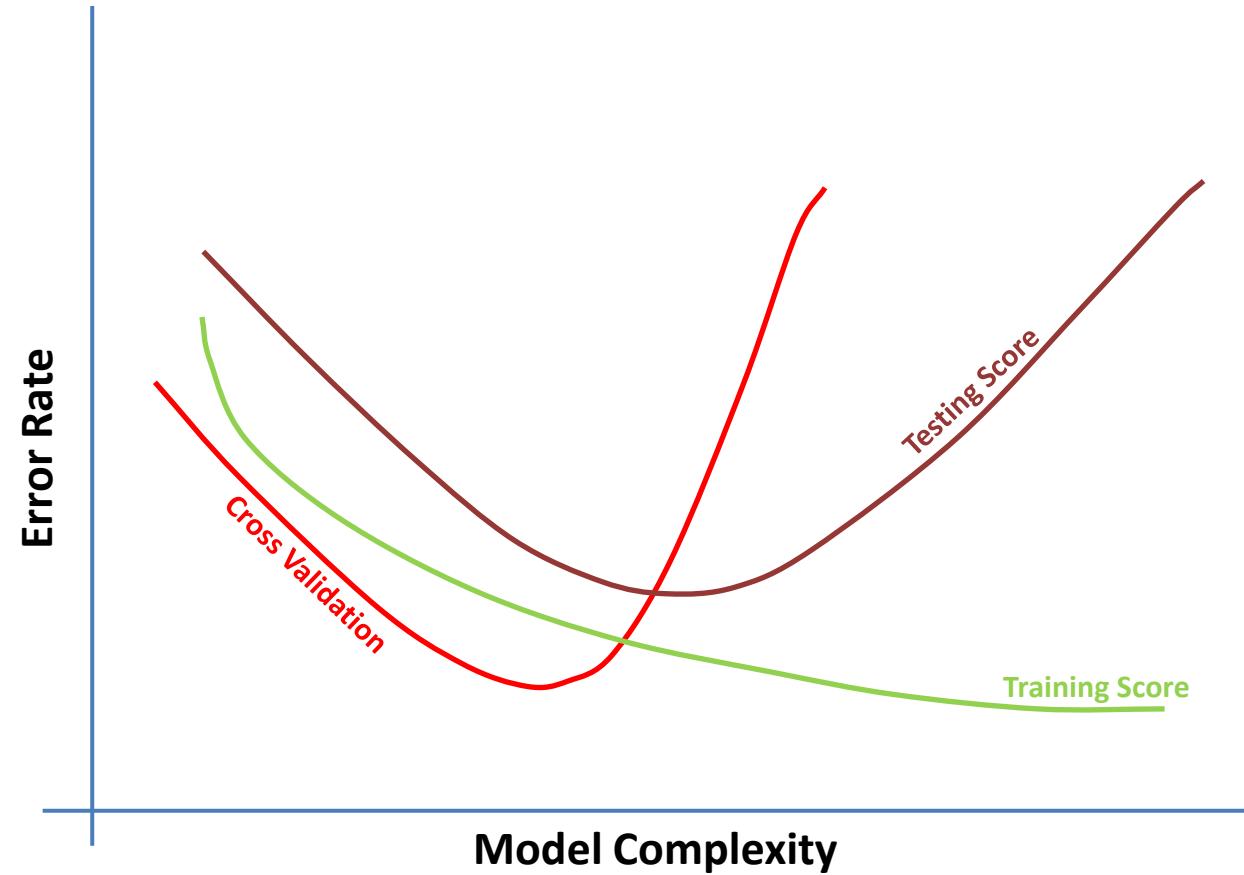
# Model Evaluation - Error vs Model Complexity



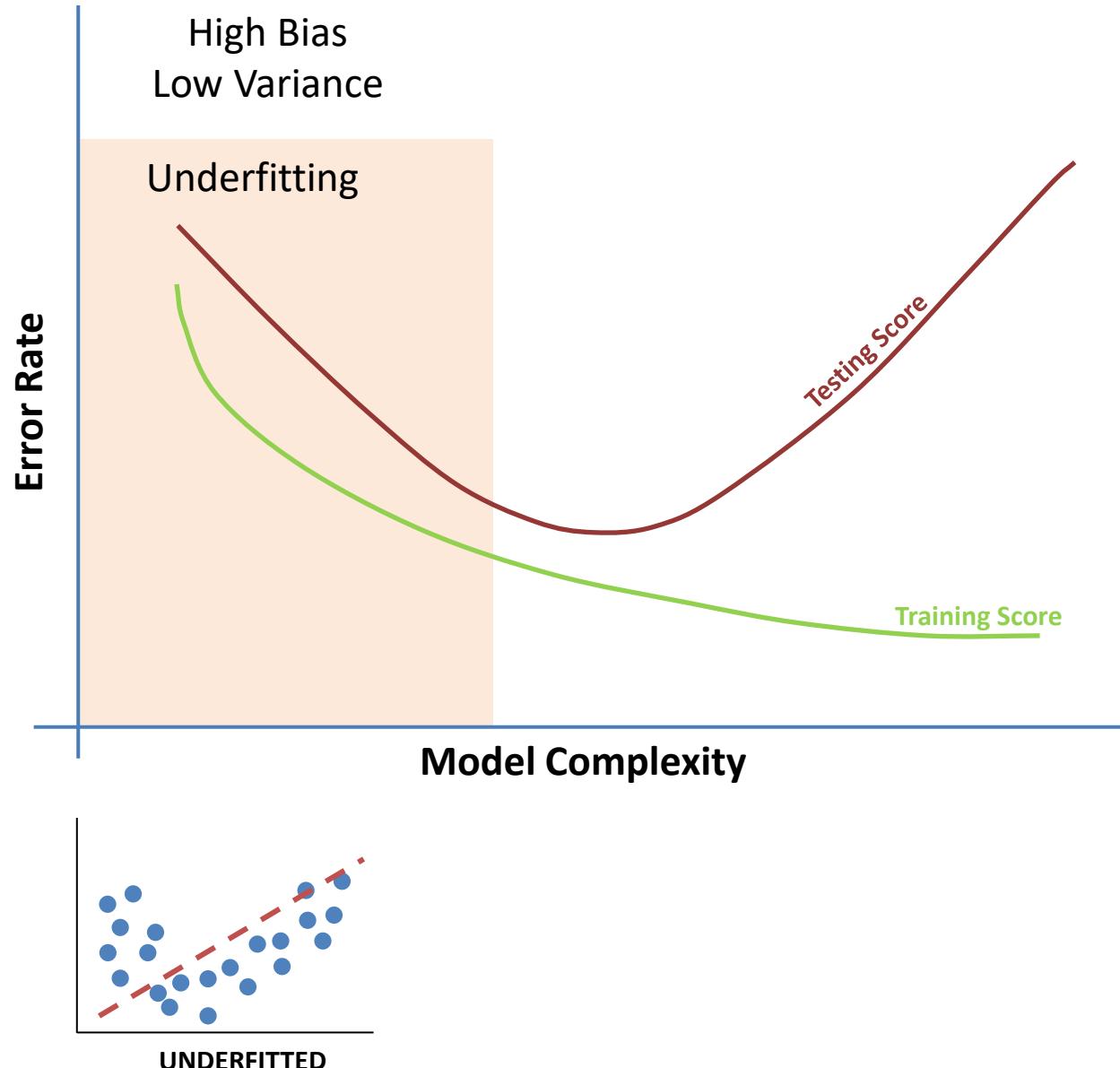
# Model Evaluation - Error vs Model Complexity



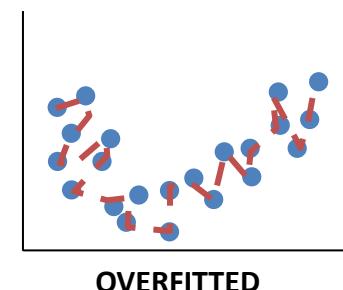
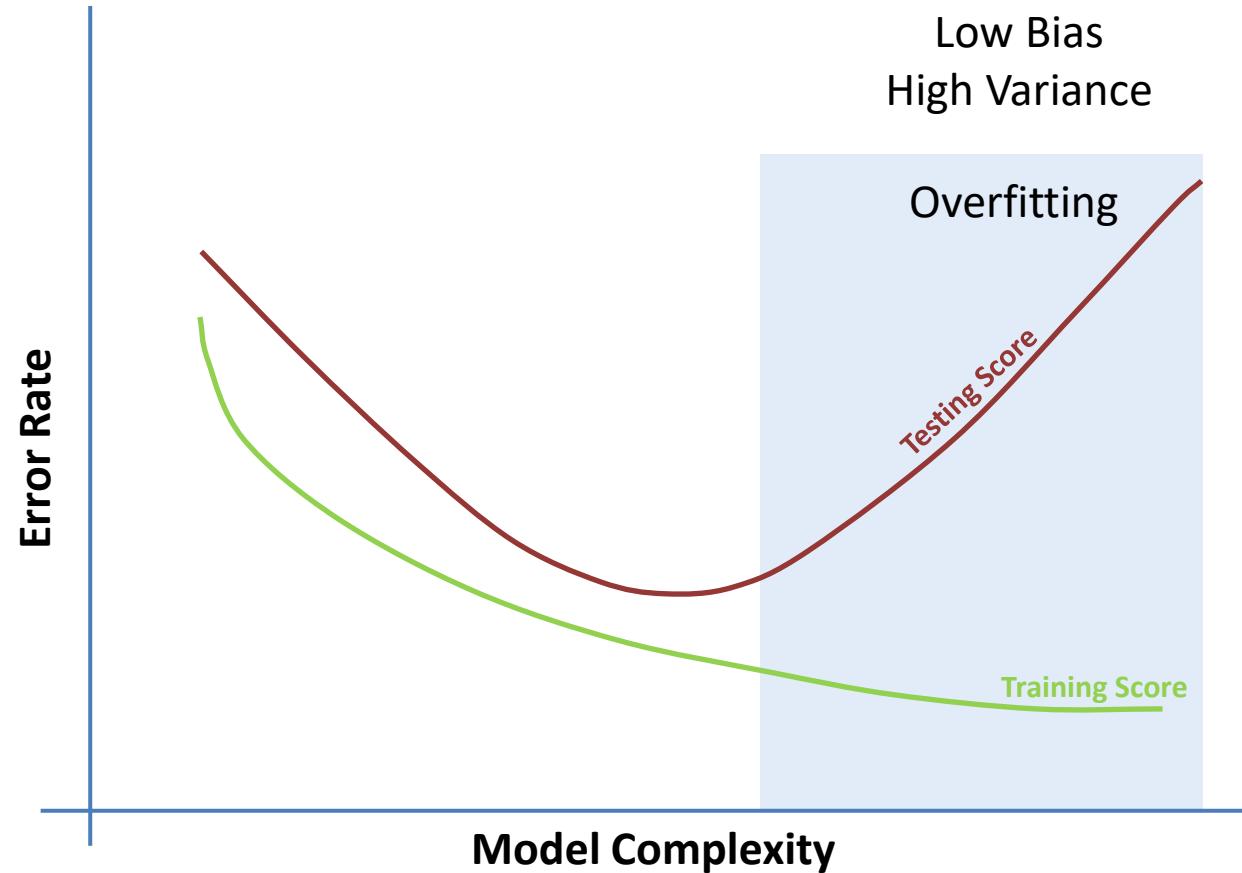
# Model Evaluation - Error vs Model Complexity



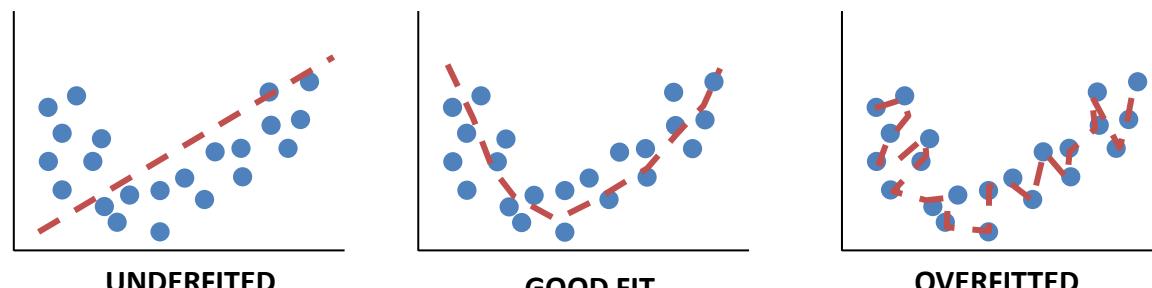
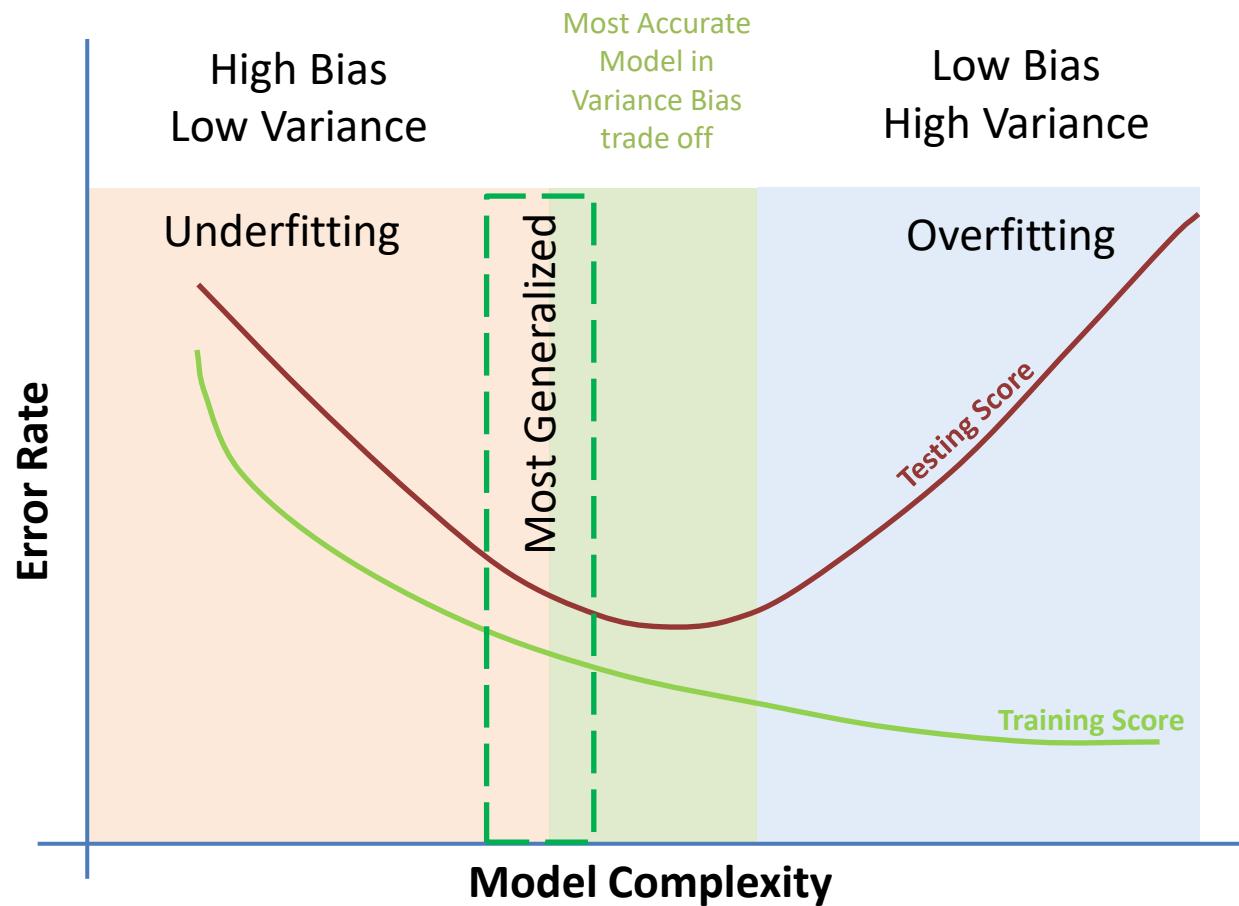
# Model Evaluation – Underfitting



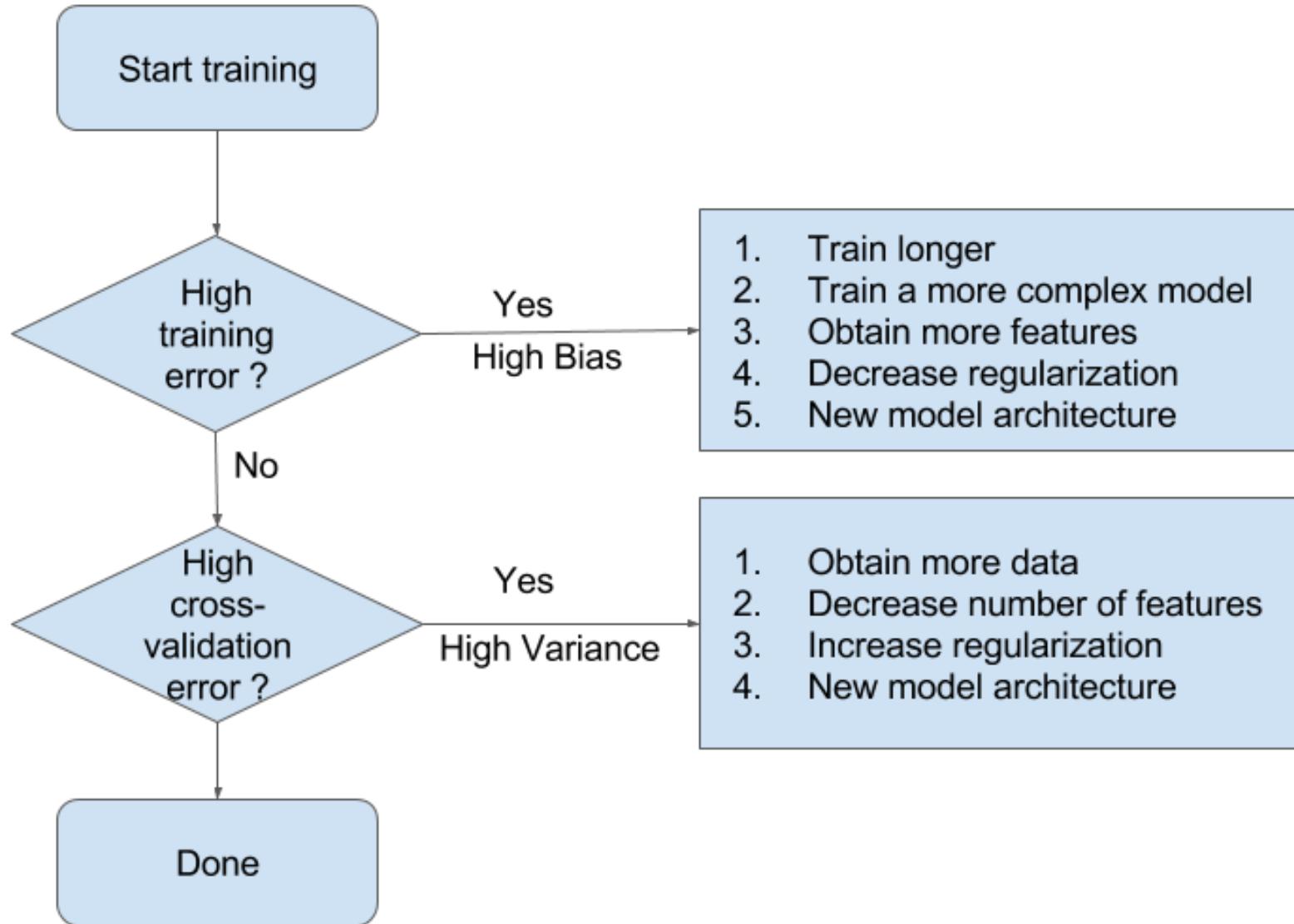
# Model Evaluation - Overfitting



# Model Evaluation - Good Fit



# Model Evaluation - How to Overcome Overfitting / Underfitting



# Model Evaluation – Classification Problems



## Confusion Matrix :

- **Precision:** true positive rate

$$\frac{TP}{TP + FP}$$

- **Recall:** true positive over the 1 class predict

$$\frac{TP}{TP + FN}$$

- **Accuracy:**

$$\frac{TP + TN}{TP + TN + FP + FN}$$

- **F1 Score:**

$$\frac{2 * \text{precision} * \text{Recall}}{\text{Precision} + \text{Recall}}$$

		Predicted Class		
		Positive	Negative	
Actual Class	Positive	True Positive (TP)	False Negative (FN) <b>Type II Error</b>	Sensitivity $\frac{TP}{(TP + FN)}$
	Negative	False Positive (FP) <b>Type I Error</b>	True Negative (TN)	Specificity $\frac{TN}{(TN + FP)}$
	Precision $\frac{TP}{(TP + FP)}$	Negative Predictive Value $\frac{TN}{(TN + FN)}$	Accuracy $\frac{TP + TN}{(TP + TN + FP + FN)}$	

---

# **HOW DOES ML WORK?**

# Linear Regression



- **Objective:**

model the expected value of a continuous variable,  $Y$ , as a linear function of the continuous predictor,  $X$

- **Model structure:**

$$Y = Ax + B$$

- **Model assumptions:**

$Y$  is normally distributed, errors are normally distributed, and independent

- **Parameter estimates and interpretation:**

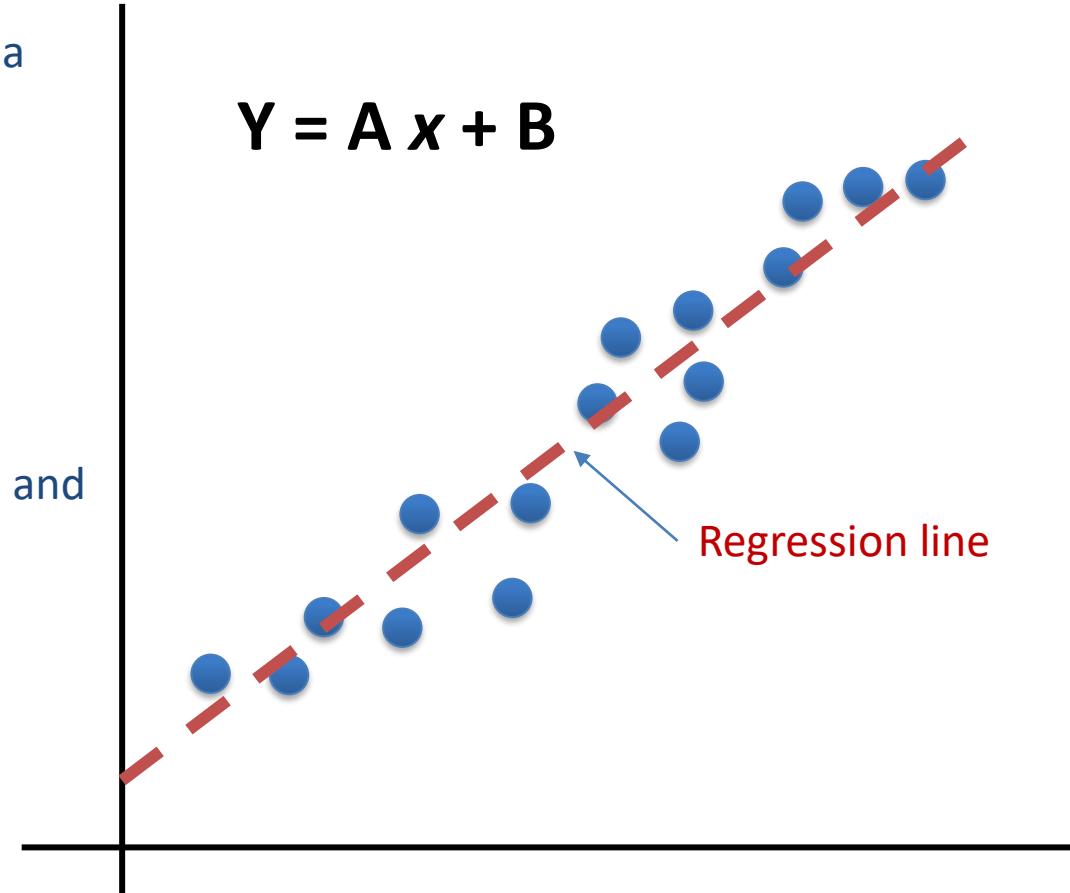
$B$  the intercept, and  $A$  is estimate of the slope

- **Model fit:**

$R^2$ , residual analysis

- **Model selection:**

possible predictors, which variables to include?



# Linear Regression - Gradient Descent



- **Objective:**

To minimize the error function to close to zero (Cost Function) If possible.

- **Function structure:**

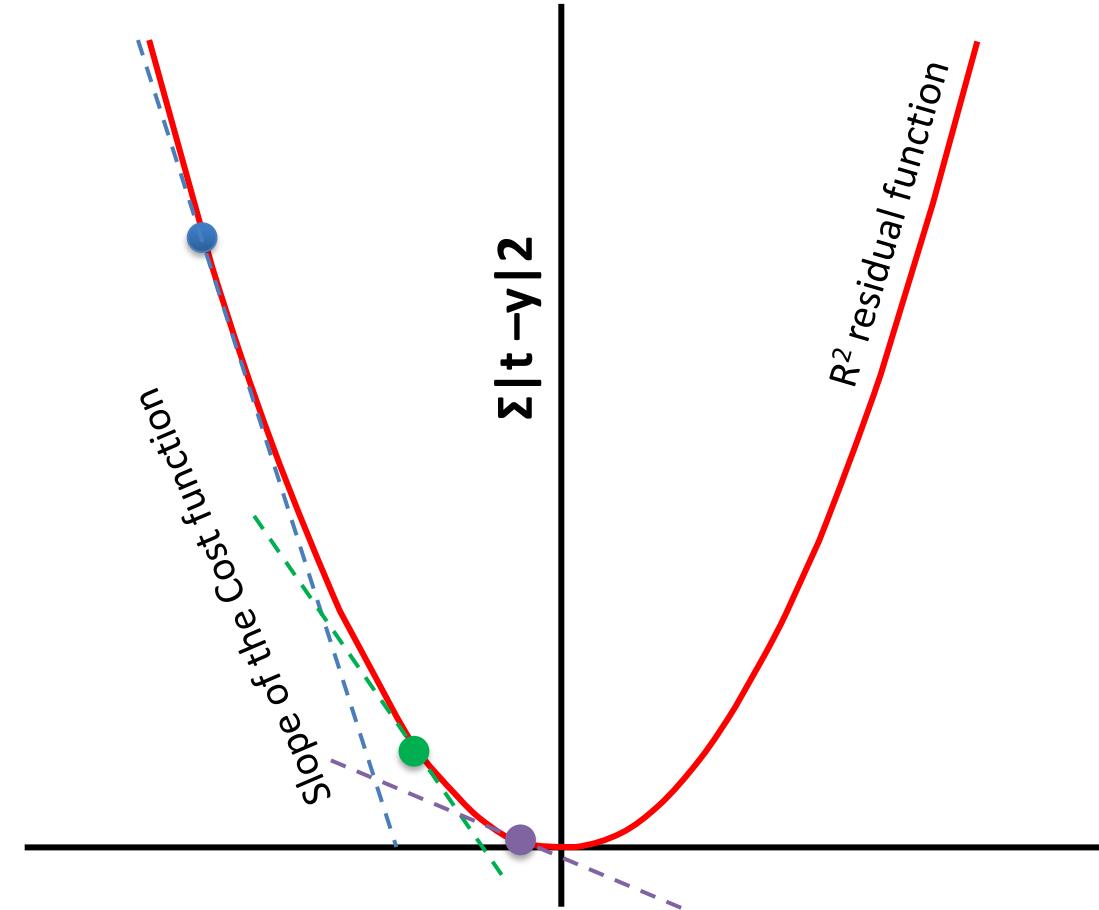
$$\text{Cost function} : \sum |t - y|^2$$

- **Model assumptions:**

Slope of the *cost function*  $\approx$  Zero, then it is the best prediction

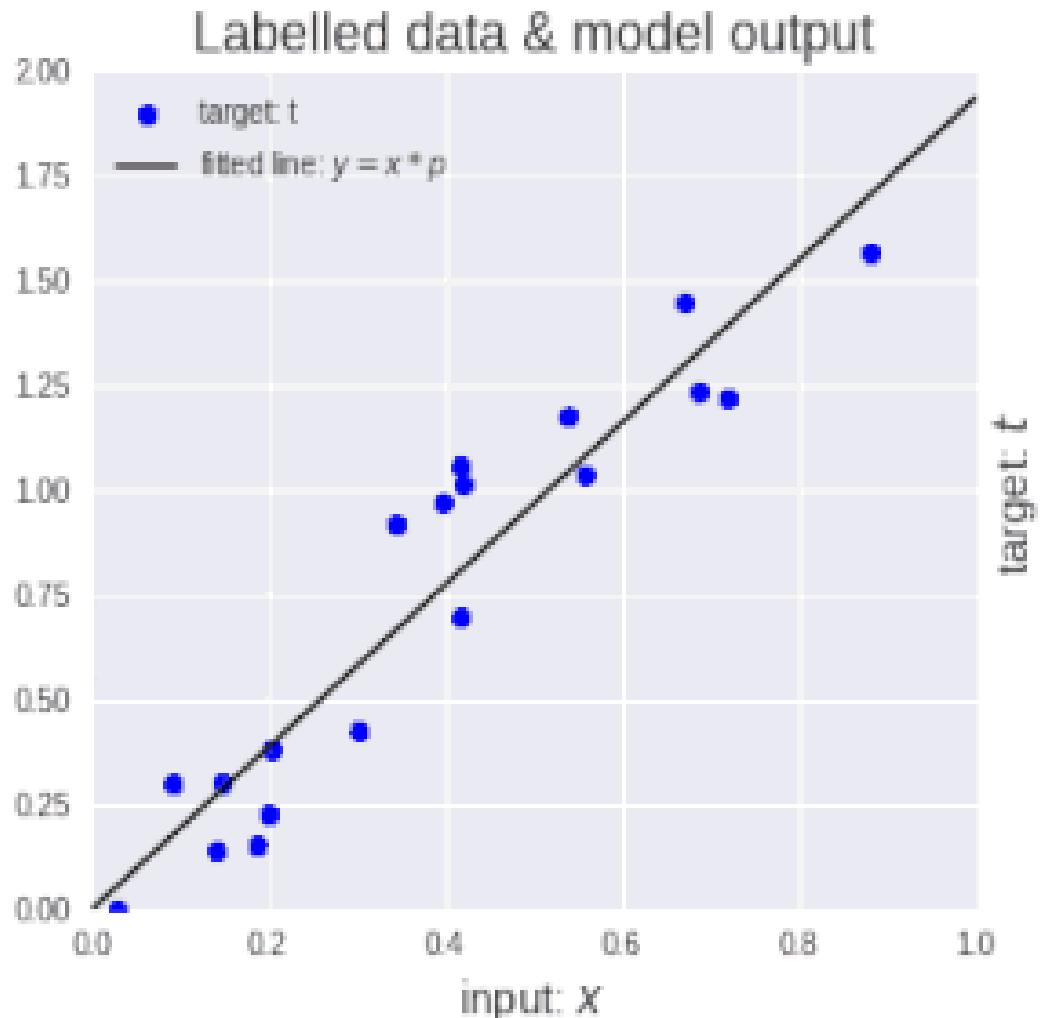
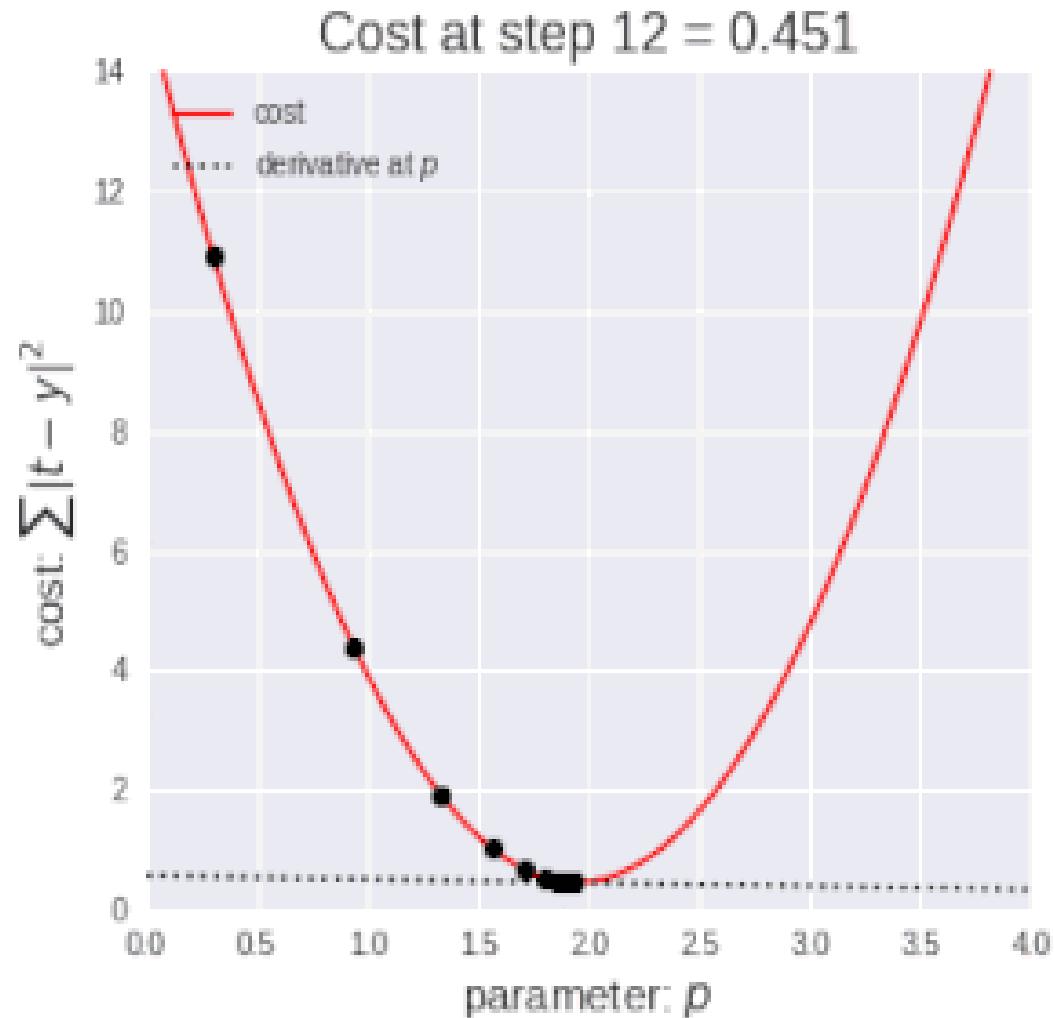
- **Parameter estimates and interpretation:**

- Slope first derivative over certain iterations,
- Learning rate



Y : Cost Function (Loss function, Error)  
A : Slope  
x : N# of iterations

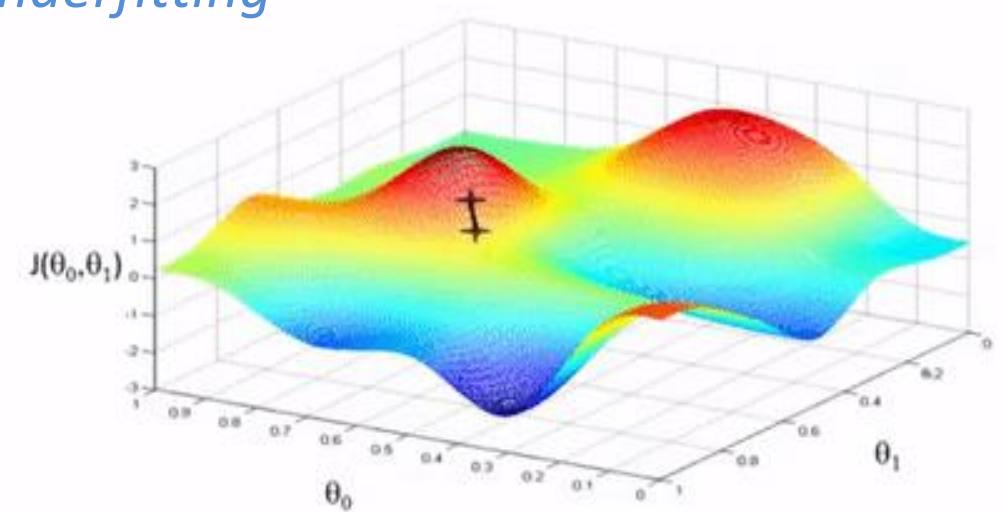
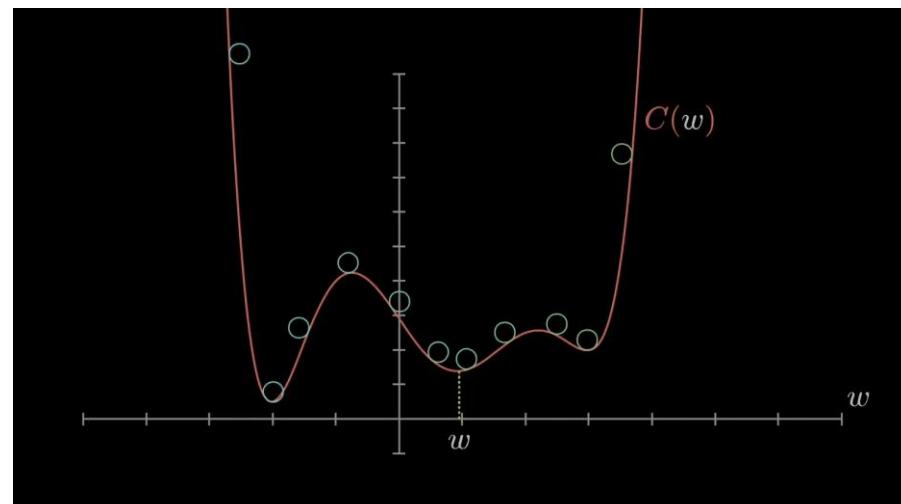
# Linear Regression - Gradient Descent



# Gradient Descent



- Gradient descent is based on *calculus*.
- Gradient descent is different from one algorithm to another based on the complexity of the algorithm and no# of variables (dimensions)
- It always has *local minima*.
- Learning rate is the essential step to reach a healthy GD
- Learning rate can be cause of *overfitting or underfitting*



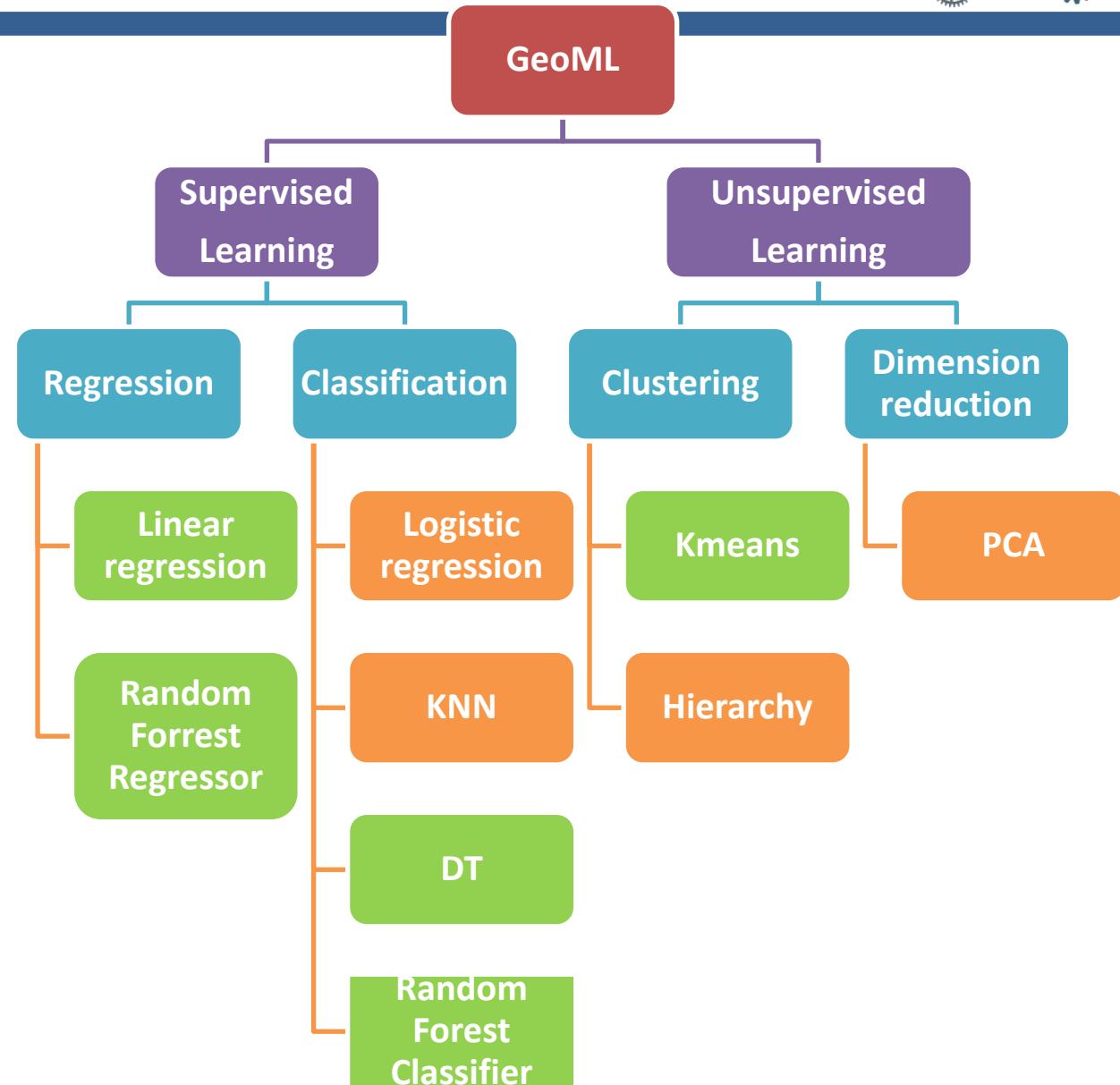
# **Machine Learning Algorithms**

# Machine Learning Algorithms



Most commonly used Machine learning algorithms:

1. Linear Regression
2. Logistic Regression
3. Decision Tree
4. Support Vector Machine (SVM)
5. Naive Bayes
6. K-Nearest Neighbors (KNN)
7. K-Means
8. Random Forest (RF)
9. Dimensionality Reduction Algorithms (PCA)
10. Gradient Boosting algorithms
11. Artificial Neural Network



# K-Means Clustering



- ***Objective:***

To be able to cluster the data based on the input variable or variables and find cluster centroids

- ***Model structure:***

$$\text{Euclidean distance} = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$

- ***Model assumptions:***

- Cluster centroids are in the middle of each cluster
- Each cluster has a centroid and data is scattered around it

- ***Parameter estimates and interpretation:***

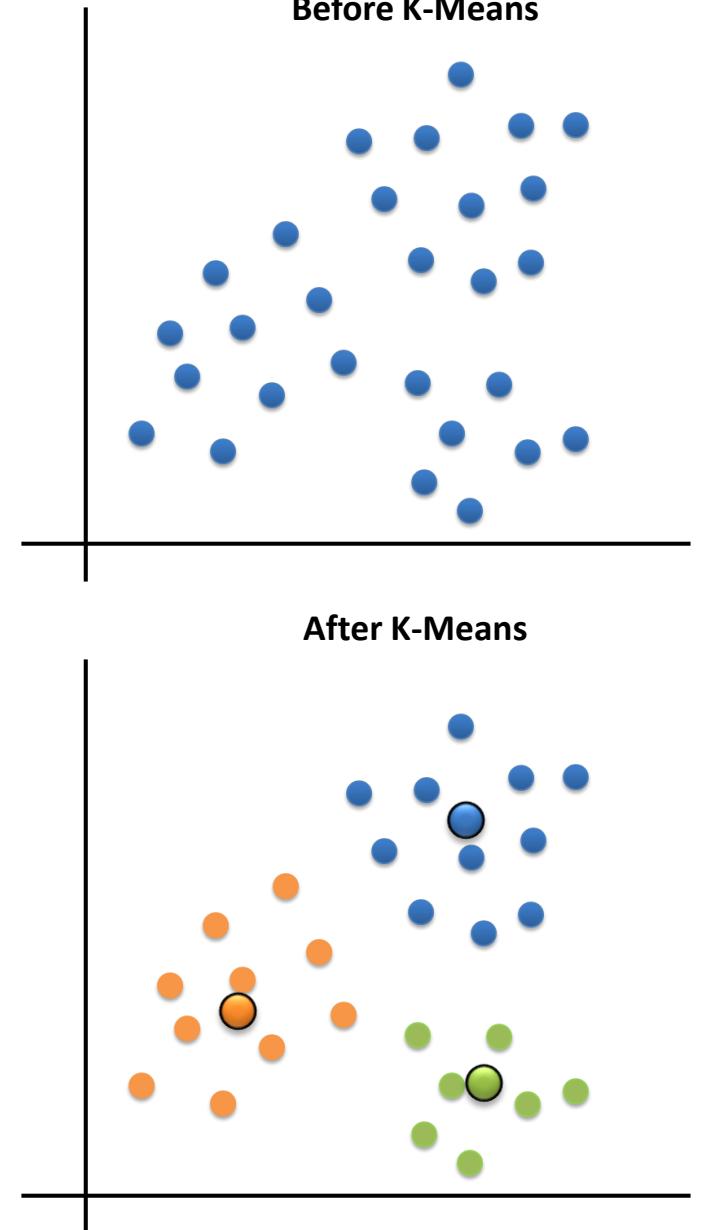
Find Euclidian distance that correspond to each centroid

- ***Model fit:***

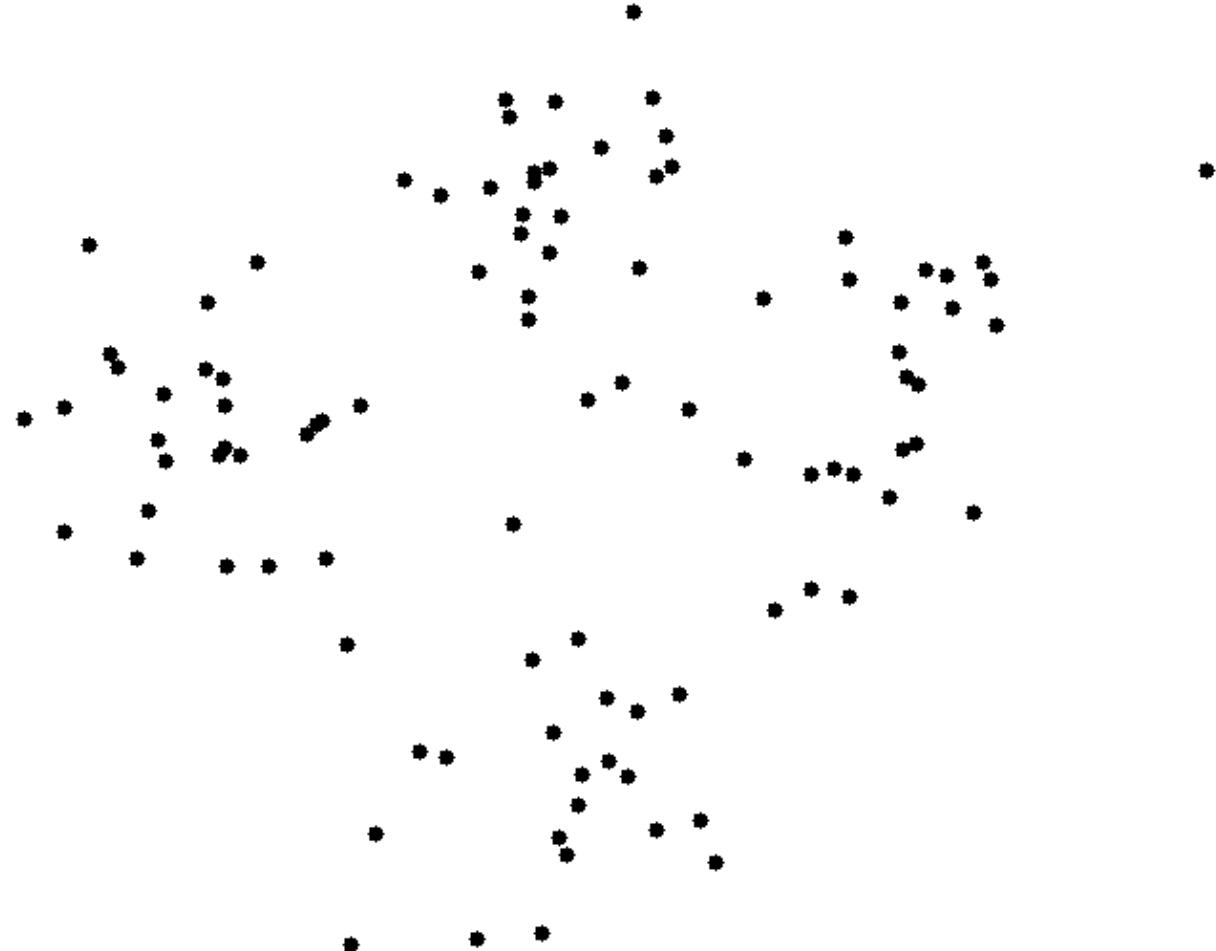
$R^2$ , residual analysis

- ***Model selection:***

How Many Centroids ?



# K-Means Classification



# Decision Tree



- ***Objective:***

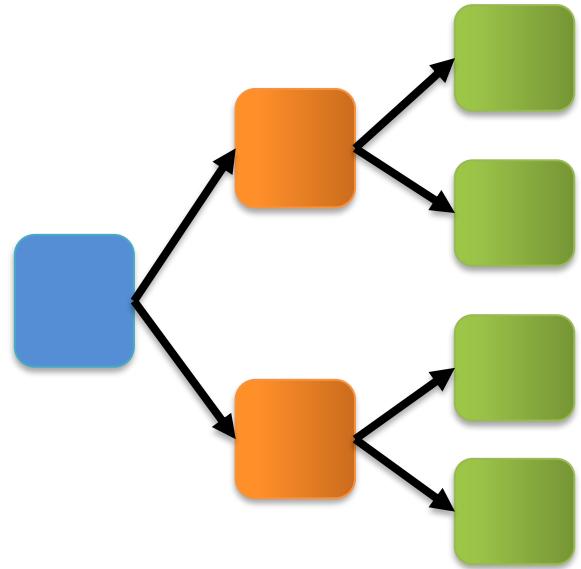
To be able to build decision boundaries based on the maximum variance between variables

- ***Model structure:***

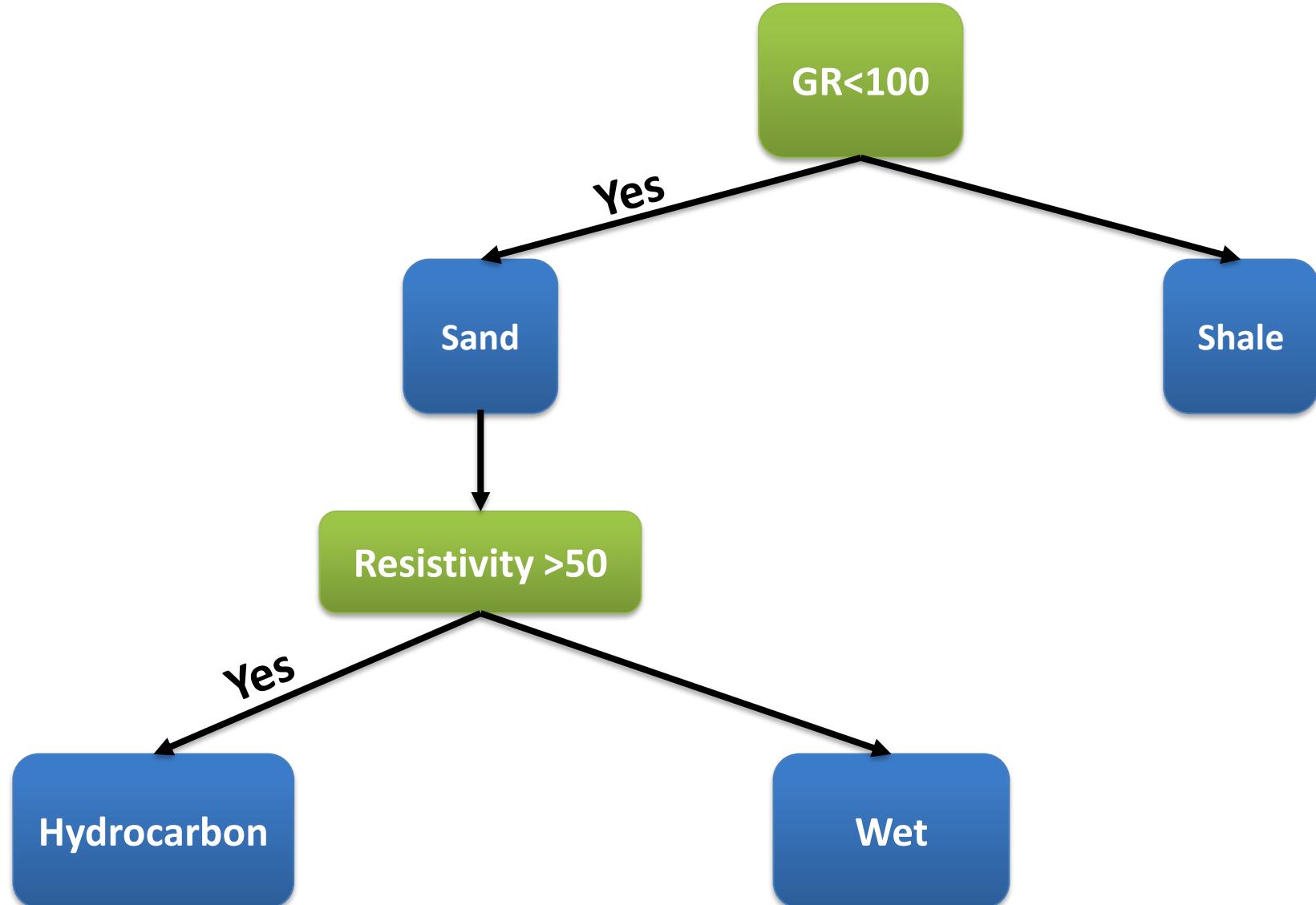
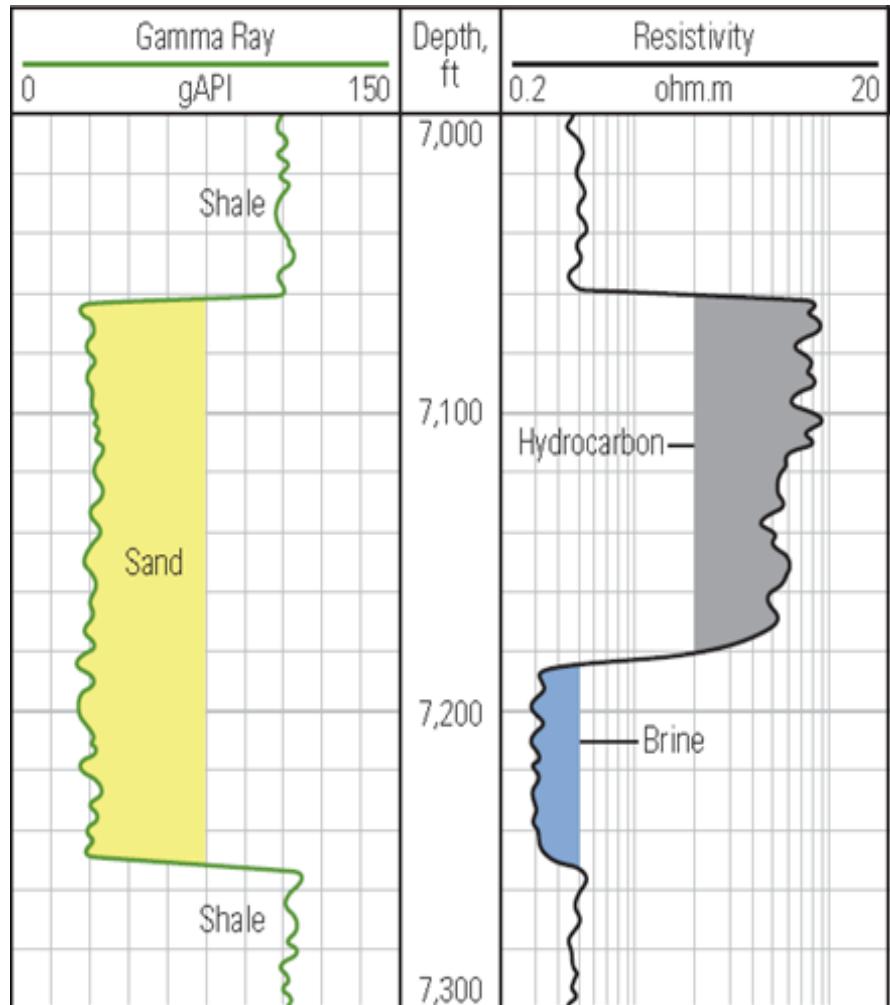
- Pick the best attribute/feature. The best attribute is one which best splits or separates the data.
- Ask the relevant question.
- Follow the answer path.
- Go to step 1 with different attribute until you arrive to the answer.

- ***Model fit:***

$R^2$ , Confusion Matrix



# Decision Tree



# Random Forest



- **Objective:**

To be able to build decision boundaries based on the maximum variance between variables

- **Model structure:**

Step 1 : Select random samples from a given dataset.

Step 2 : Construct a decision tree for every sample.

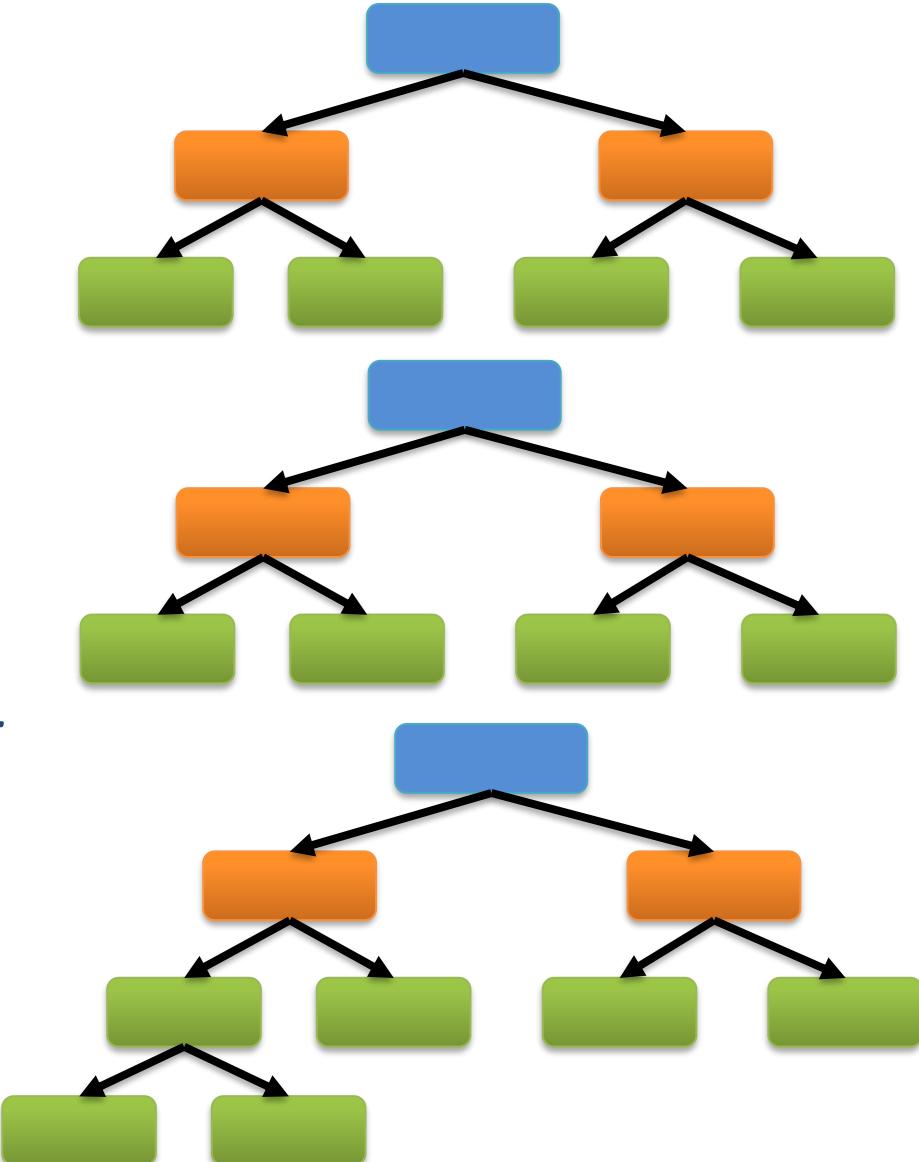
Step 3 : Get the prediction result from every decision tree.

Step 4 : Voting will be performed for every predicted result.

Step 5 : Select the most voted prediction result as the final prediction result.

- **Model fit:**

$R^2$ , Confusion Matrix



# Neural Network



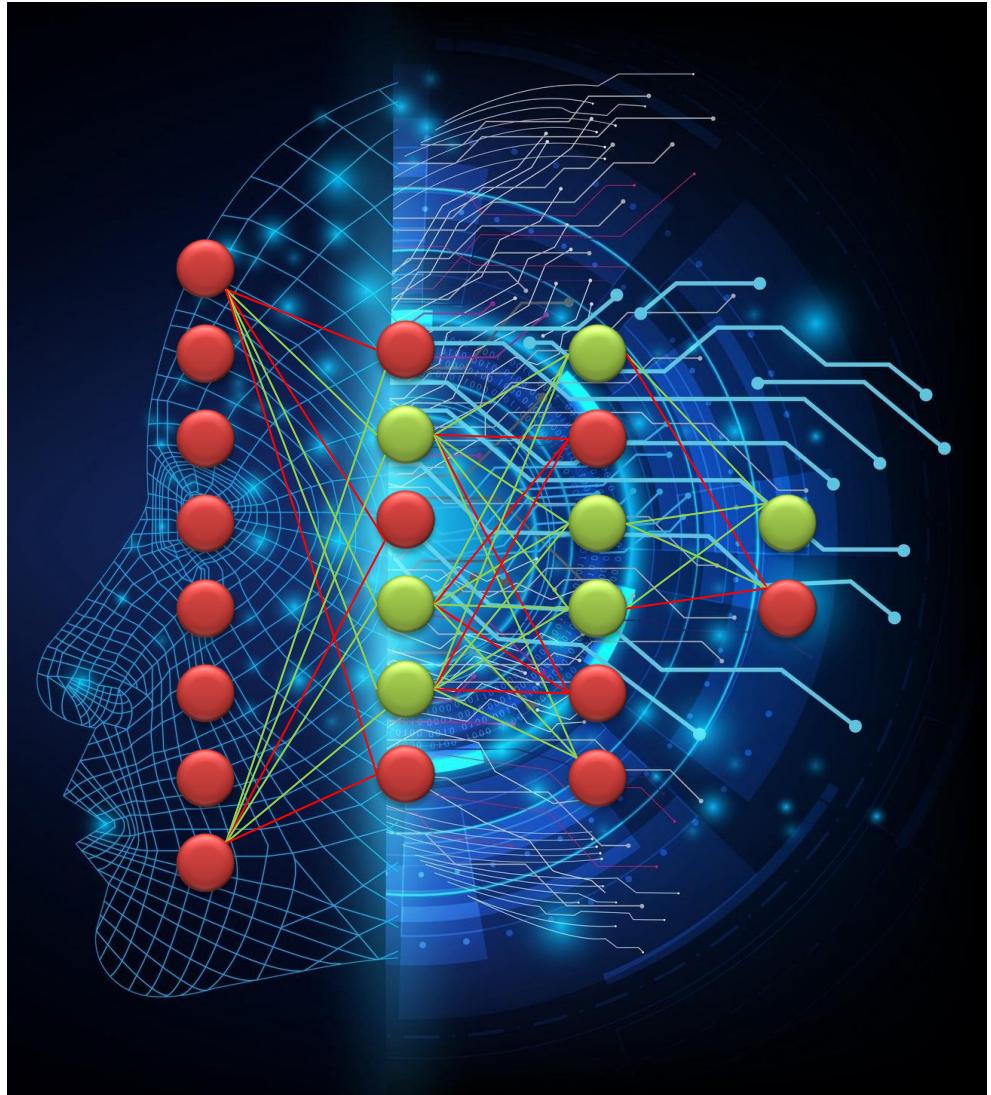
- **Introduction**
- **ANN Structure**
- **How ANN Works**
- **Activation Functions**



# Neural Network - Introduction



- In 1943, Warren McCulloch and Walter Pitts first made a trial to create a computational model from human neural networks.
- Neural Network structure is composed of :
  - Input Layer
  - Hidden Layers
  - Output Layers
- Composed of two algorithmic stages :
  - Forward Propagation
  - Backward Propagation
- Neural Network can be used for prediction or classification based on the nature of the output layer.



# Neural Network - Structure

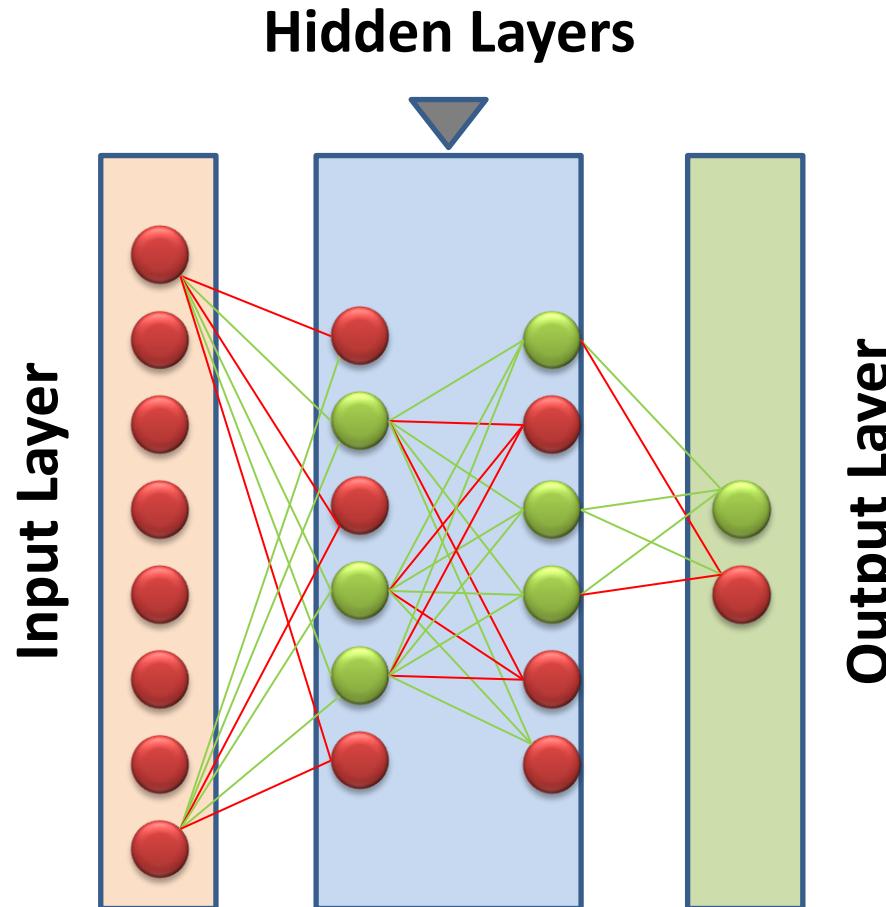


## Input Layer:

- Number of nodes is equal to number of input variables.
- Each node is associated with a randomly selected weight.

## Hidden Layer(s):

- Number of layers and Nodes are dependent on the problem.
- Each connection is passing through Activation function that affects the next layer



## Hidden Layers

## Output Layer:

- **In case of prediction,**
  - the output layer will be consisting of 1 node that result in the predication.
- **In case of classification:**
  - the number of nodes will be equal to the classes we need to predict associated with probabilities for each class.
  - The higher probability will be the predicted class.

# Neural Network – How it works

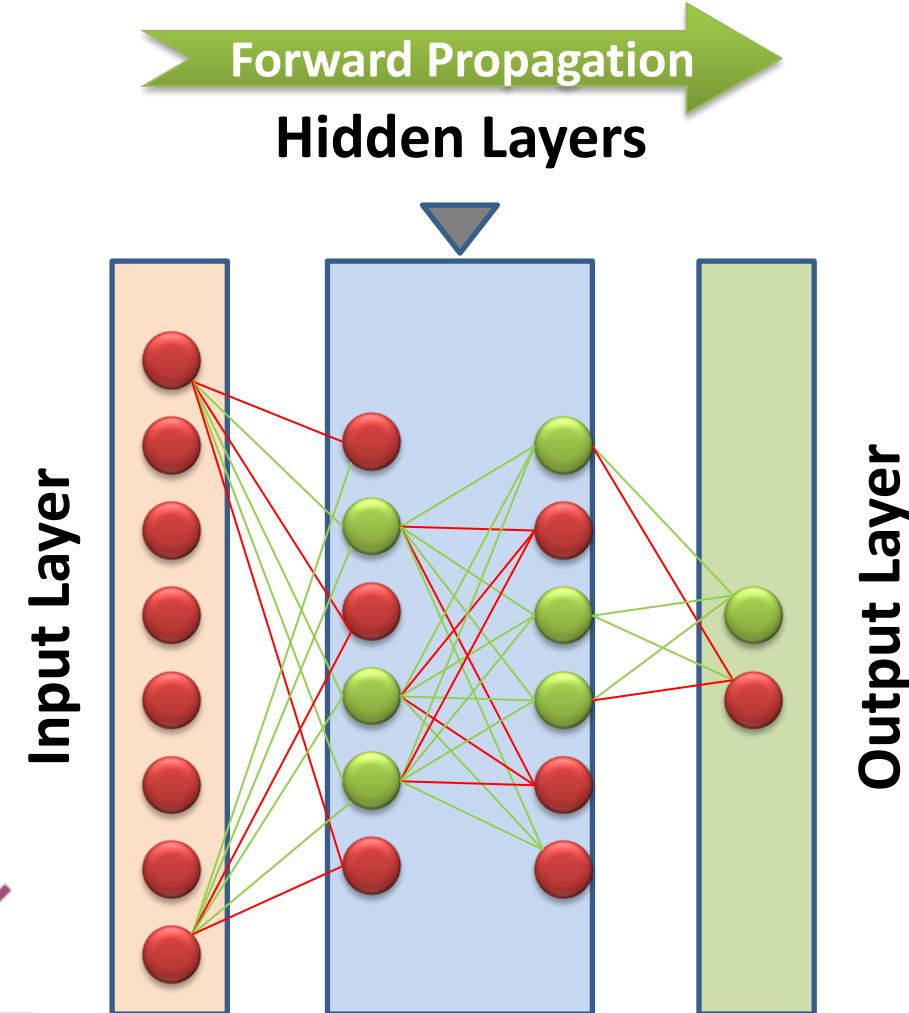
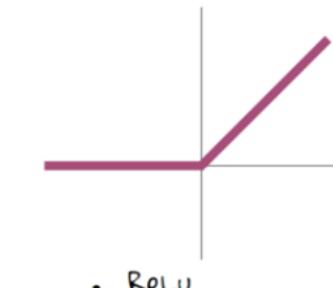
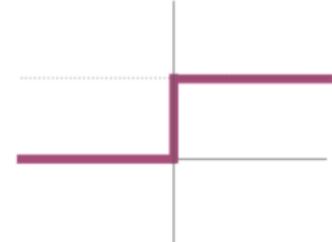
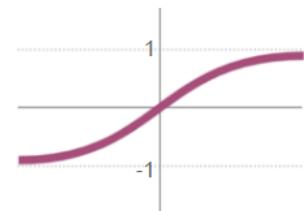
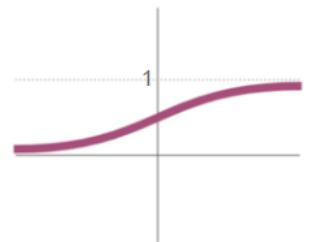


## 1. Forward propagation:

- Random weights assigned to inputs nodes
- ***Activation function*** applied to each node to change the weight.

## Activation Function:

- The model needs non-linear change to the values before submitting the next outcome values in hidden layer.
- Helps the model to capture non-linearities within the data.
- Different types of activation functions:



# Neural Network – How it works



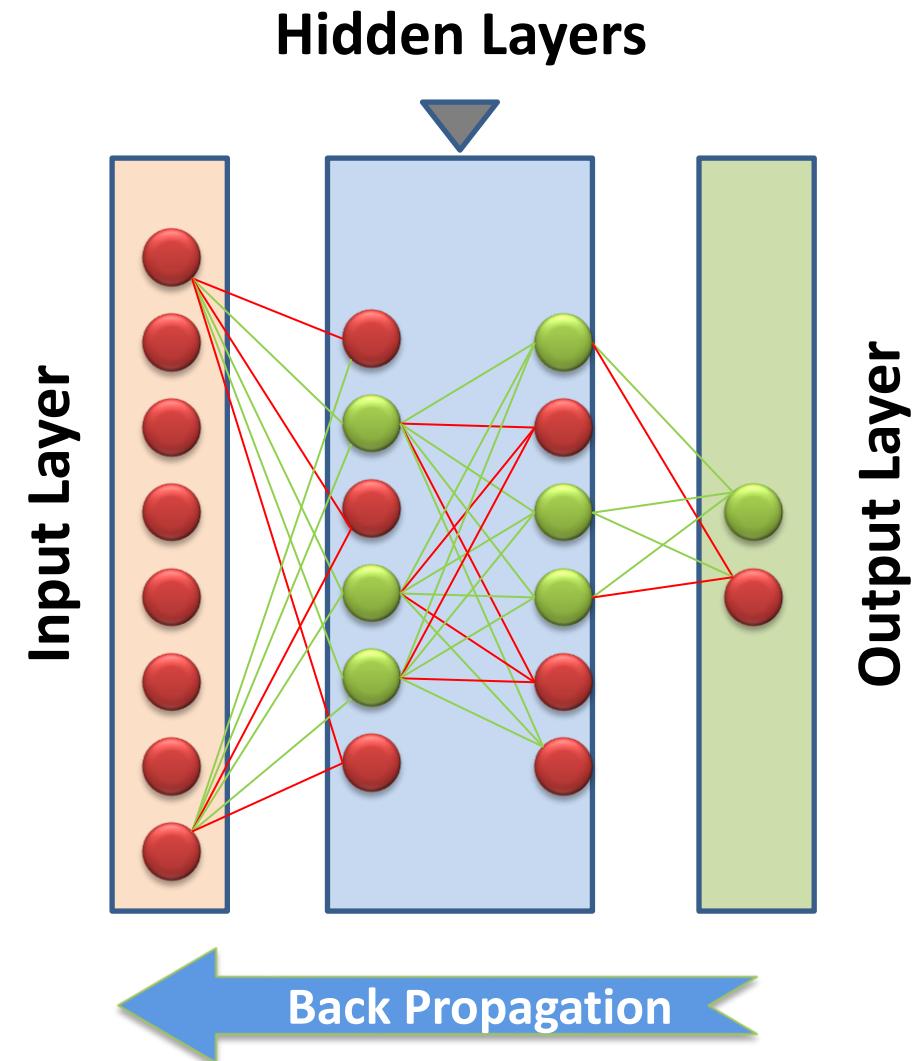
## 2. Back propagation:

- Cost function estimation.
- Update the weights by the errors difference, then
- Iterate.....

That's called **EPOCH**

**EPOCH:**

Is the time consumed to do one feed forward propagation and backpropagation in any Neural Network.



# Machine Learning Algorithms



Algorithm	Accuracy	Used for	Training time	Noise Dealing	Scaling/Normalize	Limitations
Linear Regression	Low - intermediate	Regression	Rapid	NO	Yes	Weak predictive on correlated variables
Logistic Regression	Low - intermediate	Classification	Rapid	NO	Yes	Sensitive to background noise. Limited by high number of features
Naïve Bayes	intermediate	Classification	Rapid	Yes	NO	Assume features are independent
KNN	intermediate	Both	Rapid	NO	Yes	Distance based algorithm
K- Means	intermediate	Classification	Rapid	NO	Yes	Distance Based algorithm
DT	Intermediate - High	Both	Rapid	NO	NO	Risk Overfitting
Random Forest	High	Both	Intermediate	Yes	NO	Risk Overfitting
SVM	High	Classification	Rapid	Yes	Yes	Risk Overfitting Black box Algorithm
ANN	High	Both	Rapid	Yes	Yes	Risk Overfitting Needs computational Power Black box Algorithm

---

# **Machine Learning Applications in Oil and Gas Industry**

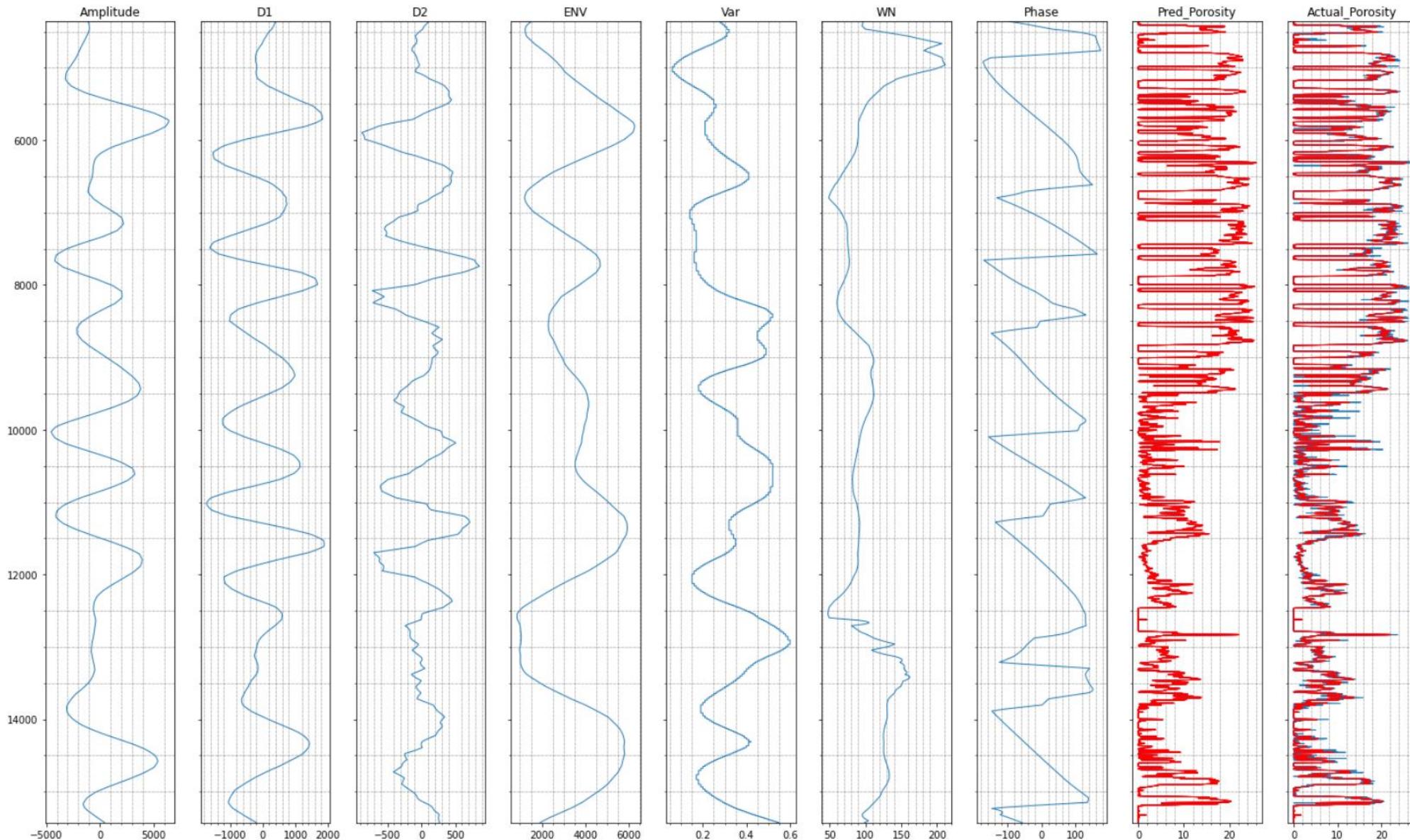


# Machine Learning Application Examples

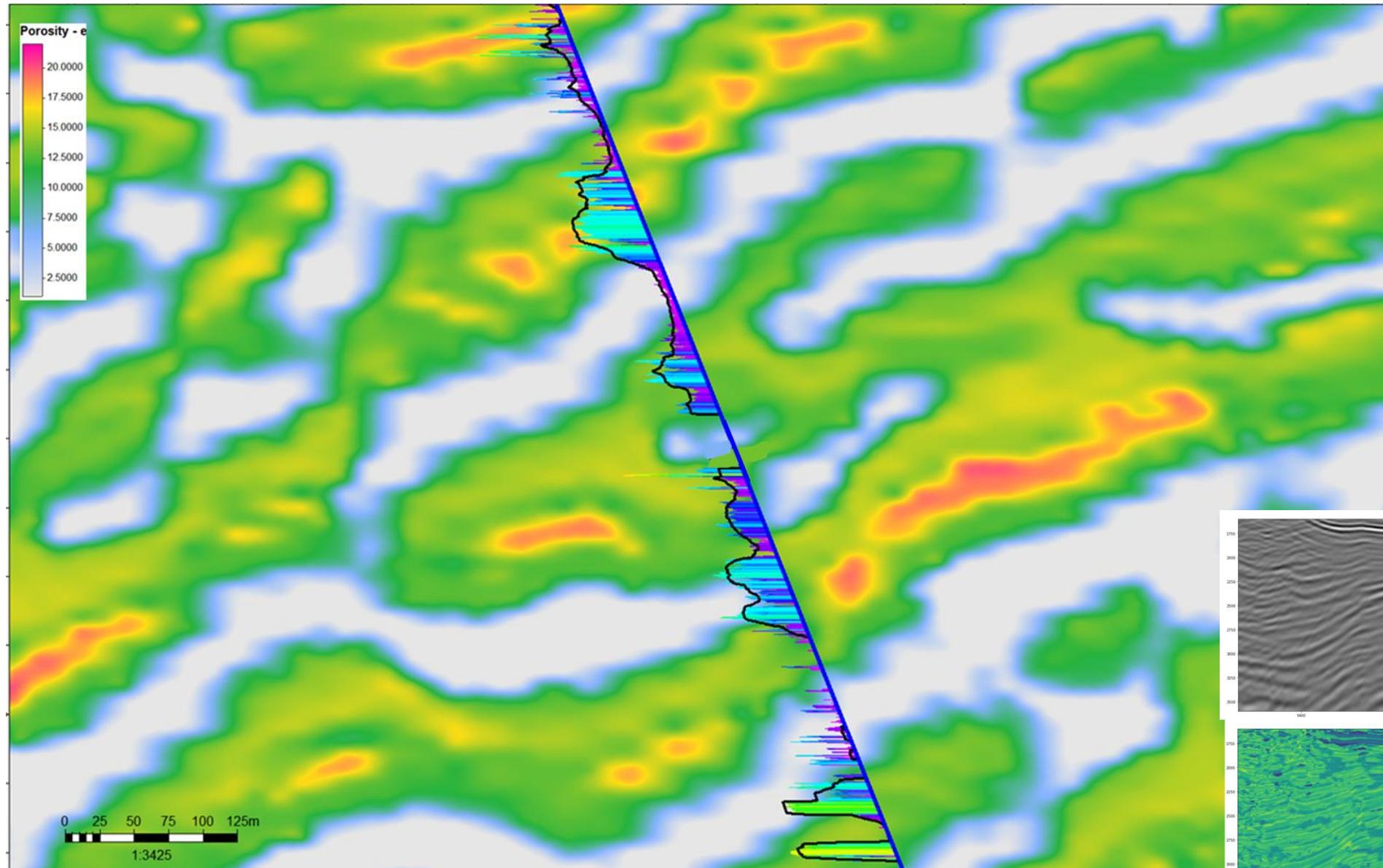
- Facies Classification using well log data
- Porosity Prediction using seismic attributes
- Permeability Prediction using Petrophysical volumes
- Facies Classification using seismic attributes
- Seismic Data inversion using Multi solver algorithms.

# Using Seismic Attributes to predict Porosity logs . . .

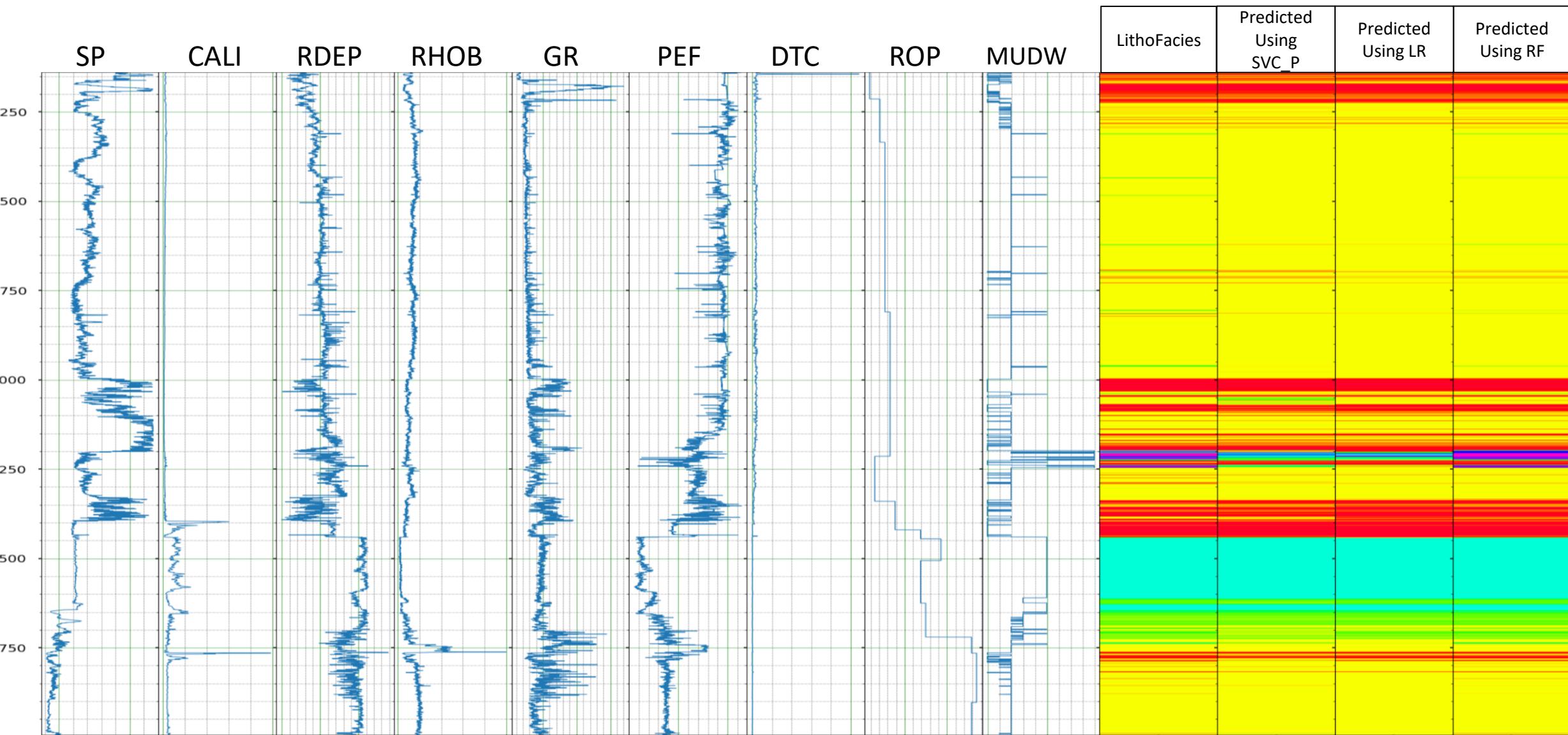
Comparison between the **Predicted** and **Actual**



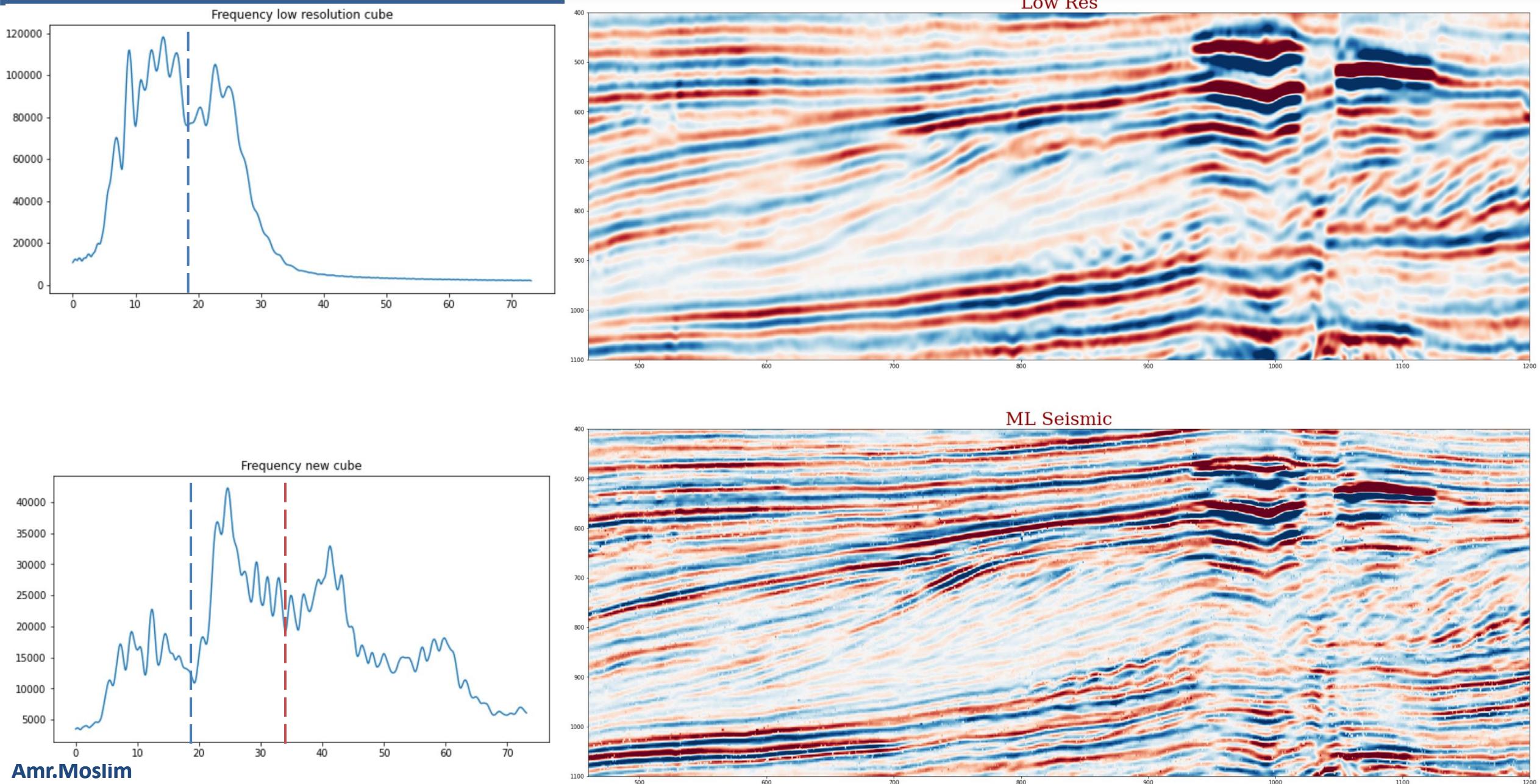
# Porosity log vs 3D Predicted Porosity Cube



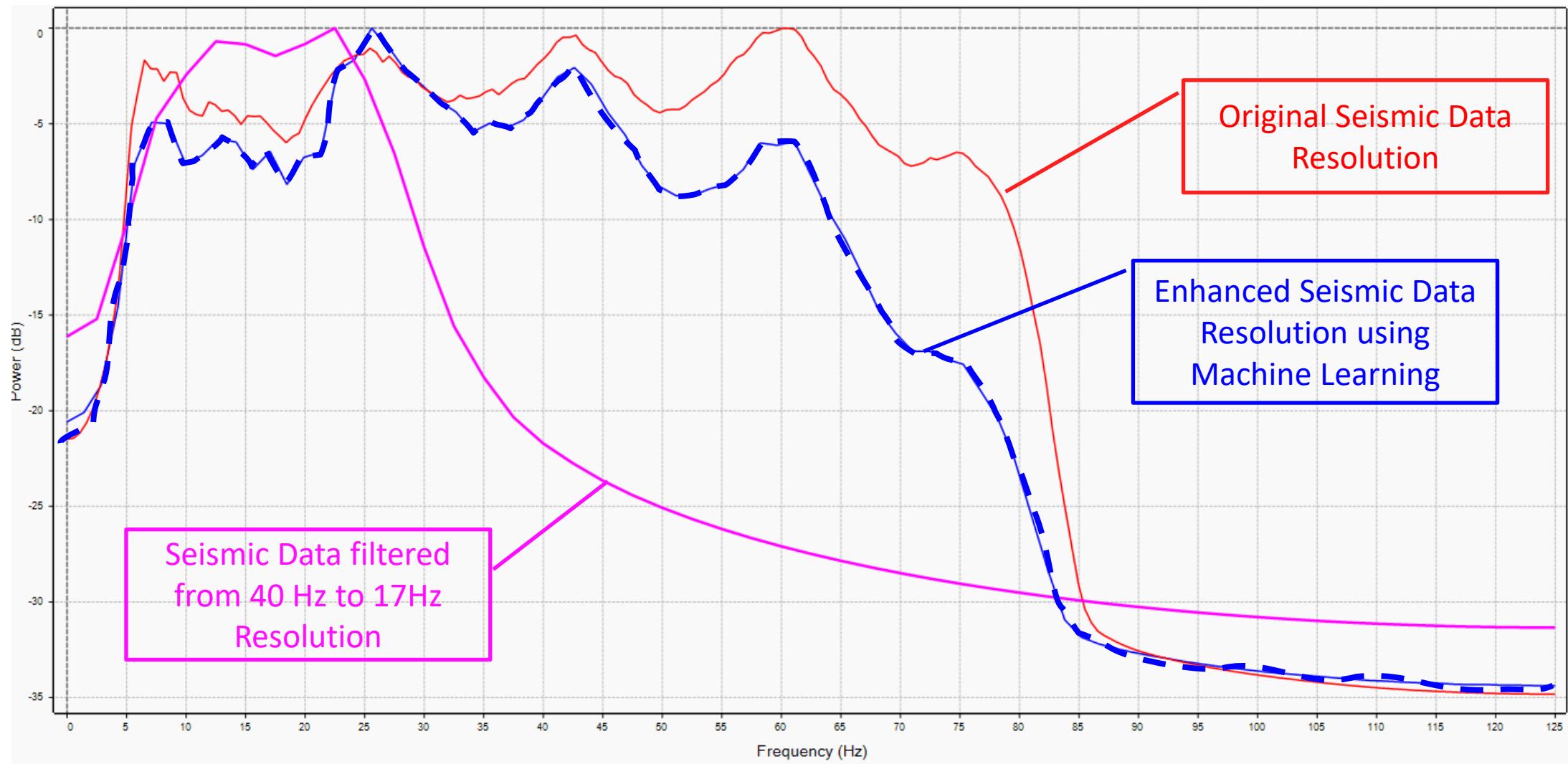
# Facies Classification Using Well Logs . . .



# Seismic Data Enhanced Resolution



# Seismic Data Enhanced Resolution



\*Amplitude spectrum curves calculated using PETREL Software

# The Road Map

**Python:** Numpy,  
Matplotlib, pandas  
**R:** Tidyverse- tidymodels  
- ggplot



Phase 1 : Learn  
a Programming  
Language  
( Python, R,SQL)

Phase 2 : Data  
Wrangling  
Techniques &  
Database

**Python:** Scipy, Obspy,  
statmodels, Plotly,  
**R:** Tidyverse- tidymodels  
- ggplot

Probabilities, center  
measures, variation

Phase 3 :  
Mathematics  
( Statistics, Linear  
Algebra,  
Calculus)

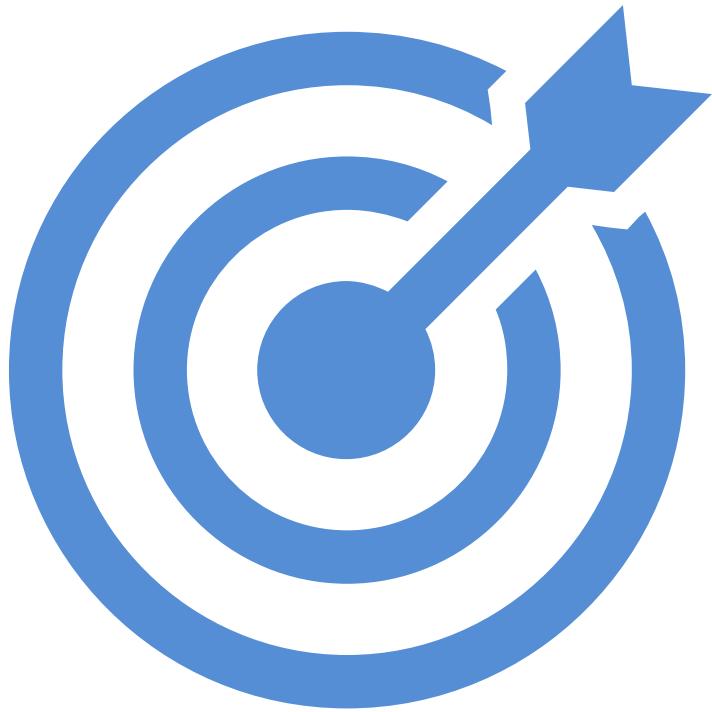
Geoscience Package

- Welly : reading / write well logs las files
- Lasio : reading / write well logs las files
- Segyio : seismic Segy files reading / writing and manipulation.
- Petropy : Petrophysical evaluation

Phase 4 :  
Machine  
Learning  
Algorithms

Machine Learning packages

- Tensorflow : Neural Network and Deep learning
- Keras: ML algorithms
- Scikit Learn: ML algorithms and model evaluations



**Thank You for Your Attention**

Amr Moslim