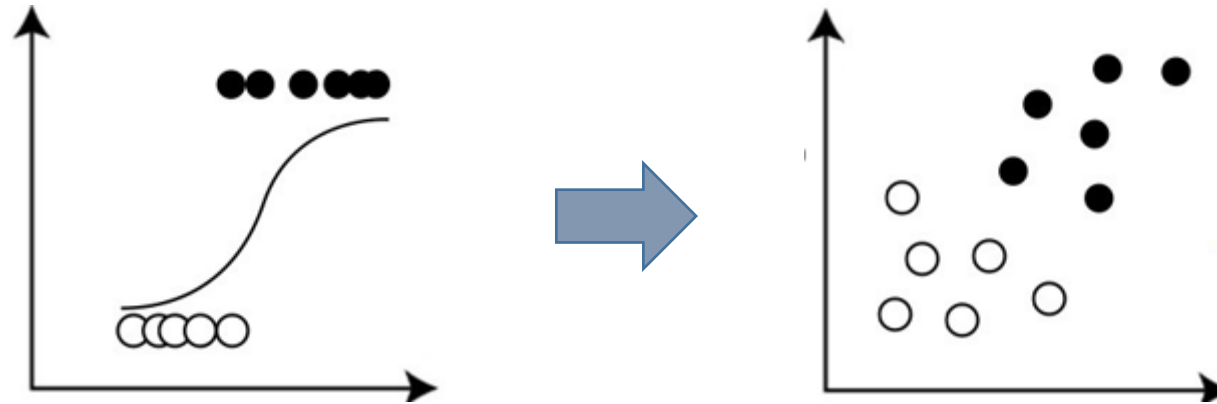




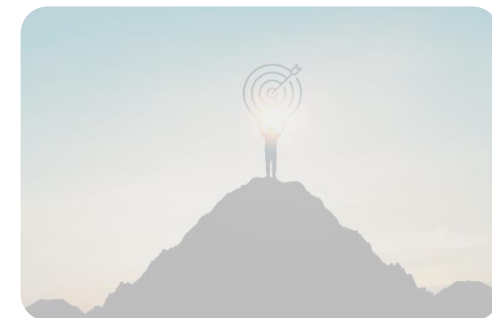
Module 7 – Logistic Regression





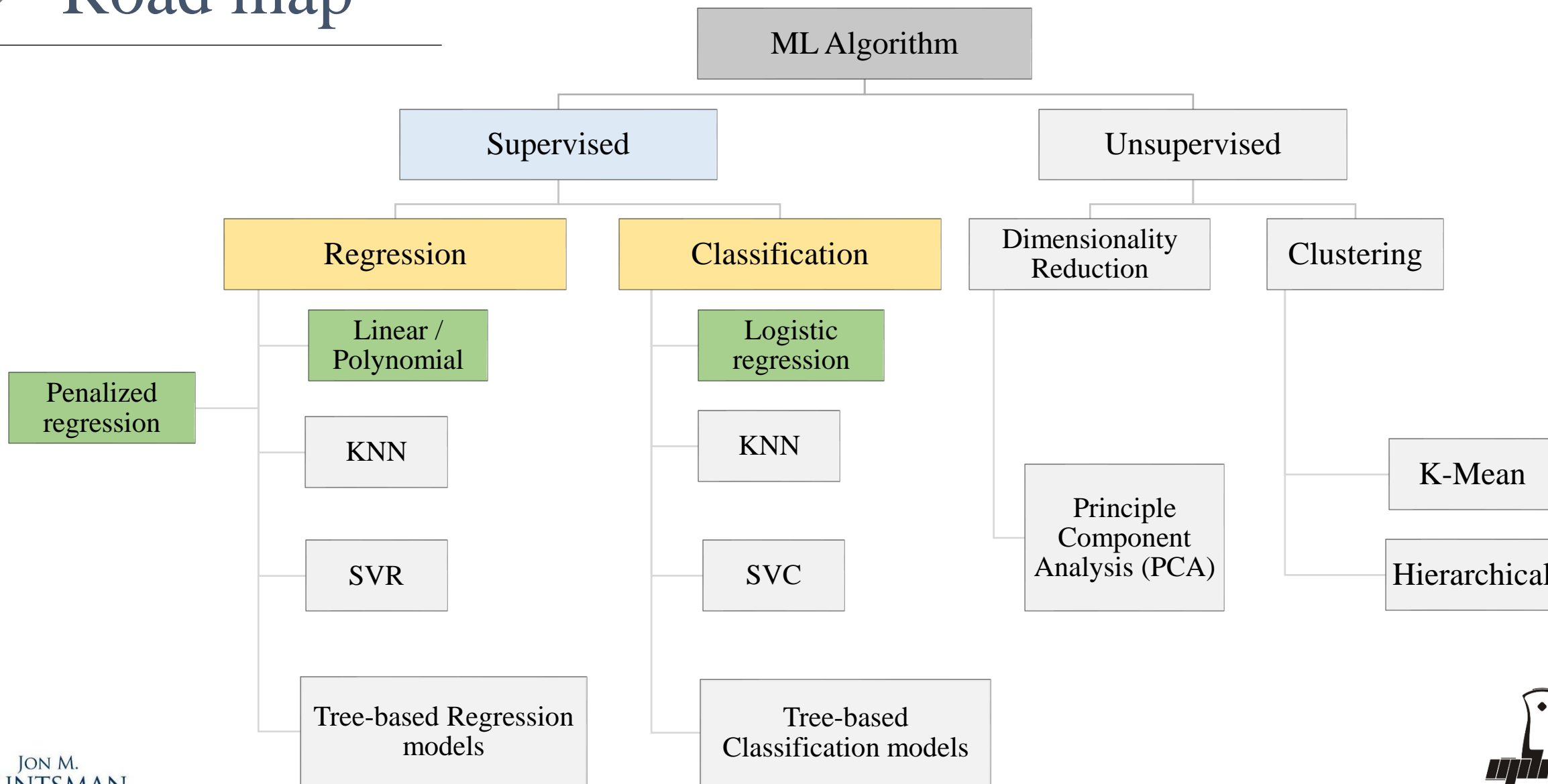
Class Modules

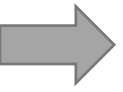
- Module 1- Introduction to Deep Learning
- Module 2- Setting up Machine Learning Environment
- Module 3- Linear Regression (Econometrics approach)
- Module 4- Machine Learning Fundamentals
- Module 5- Linear Regression (Machine Learning approach)
- Module 6- Penalized Regression (Ridge, LASSO, Elastic Net)
- **Module 7- Logistic Regression**
- Module 8- K-Nearest Neighbors (KNN)
- Module 9- Classification and Regression Trees (CART)
- Module 10- Bagging and Boosting
- Module 11- Dimensionality Reduction (PCA)
- Module 12- Clustering (KMeans – Hierarchical)





Road map





Topics

1. Linear probability model (LPM) vs Logistic regression
2. Sigmoid function
3. Logistic regression
4. Regularized logistic regression
5. Maximum Likelihood Estimation
6. MLE and GD

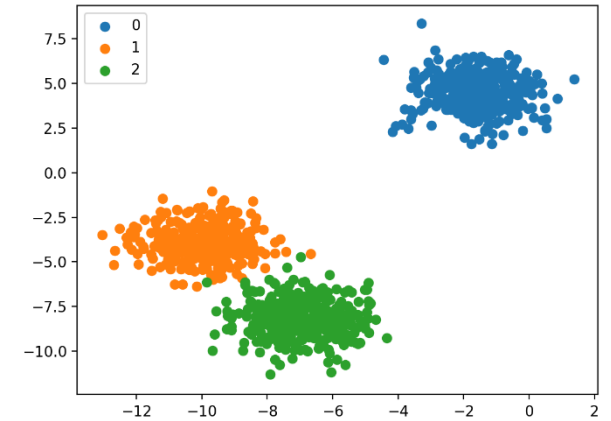


Classification

- Qualitative variables can be either nominal or ordinal.
- Qualitative variables are often referred to as **categorical**.
- **Classification** is the process of predicting categorical variables.
- Classification problems are quite common, perhaps even more than regression problems.

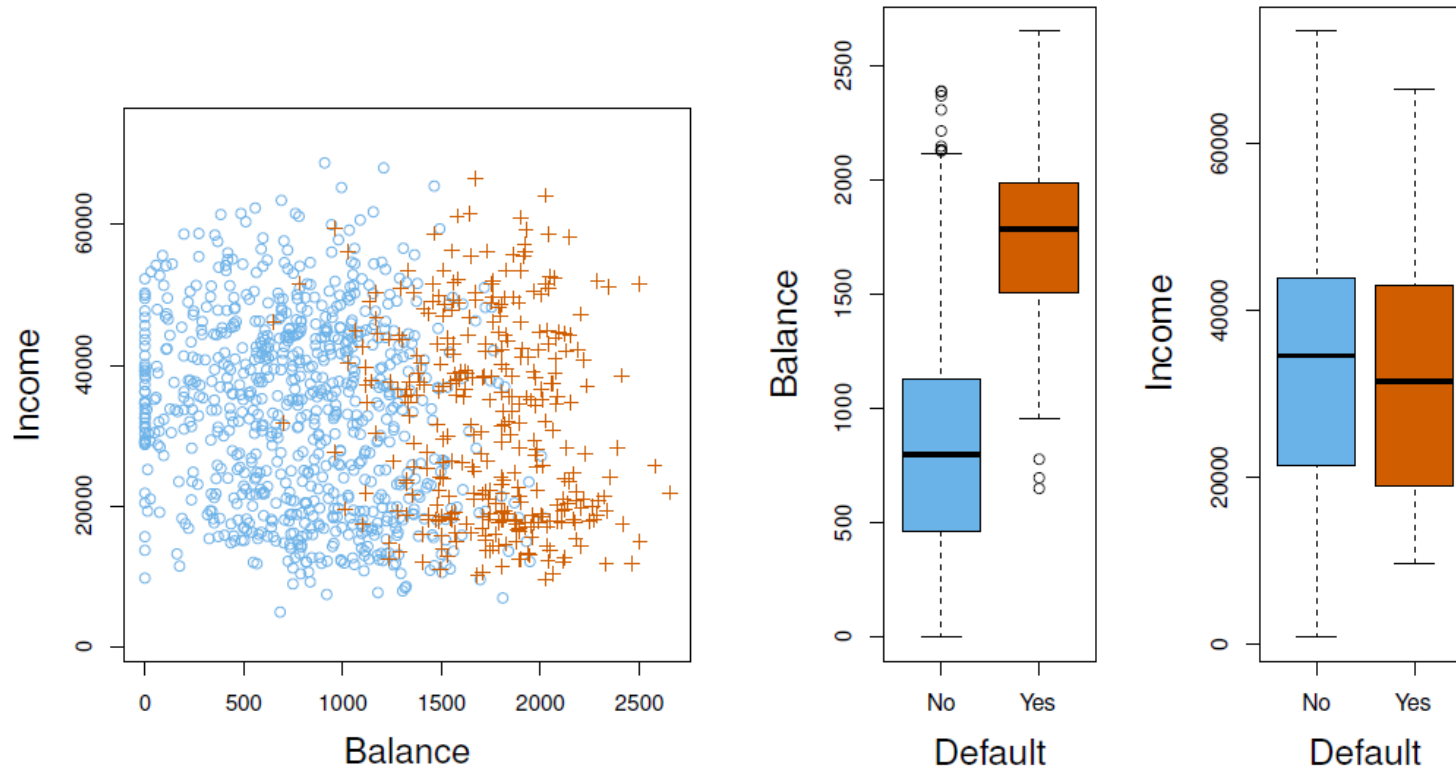
Examples:

- Financial instrument tranches (investment grade or junk)
- Online transactions (fraudulent or not)
- Loan application (approved or denied)
- Credit card default (default or not)
- Car insurance customers (high, medium, low risk)



➔ Credit card default example

- Goal: Build a **classifier** that performs well in **both** train and test set.



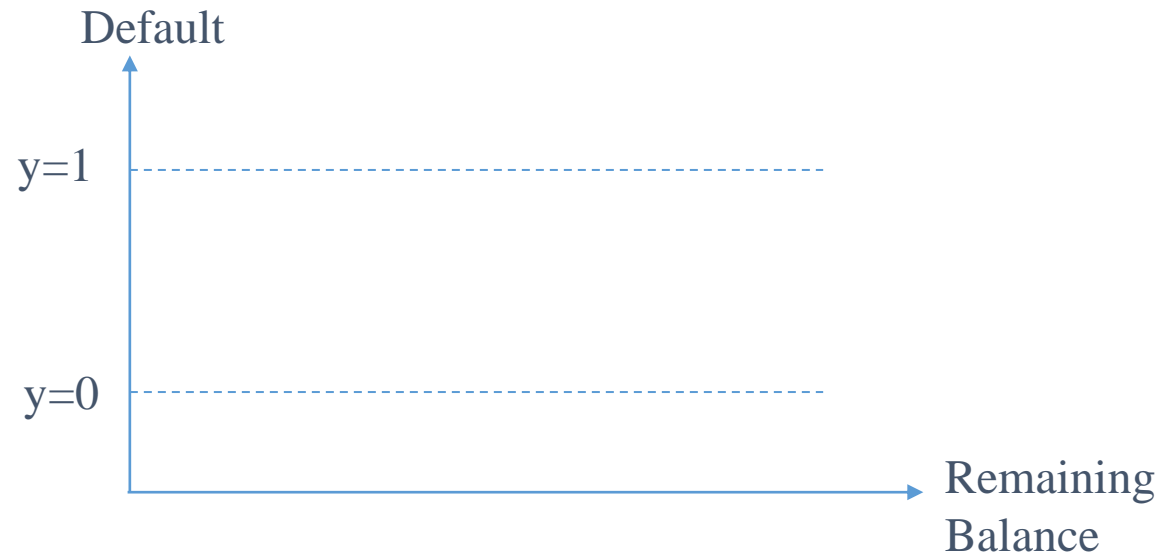


Linear Probability Model (LPM) vs Logistic Regression

Starting with **simple** LPM : $y = \beta_0 + \beta_1 bal + \epsilon$ where, $Y = 1$ for **default** and 0 otherwise.

$$E(Y|bal) = \sum P(y_i|bal) \cdot y_i = \Pr(Y = 1|bal) = P(x) = \beta_0 + \beta_1 bal$$

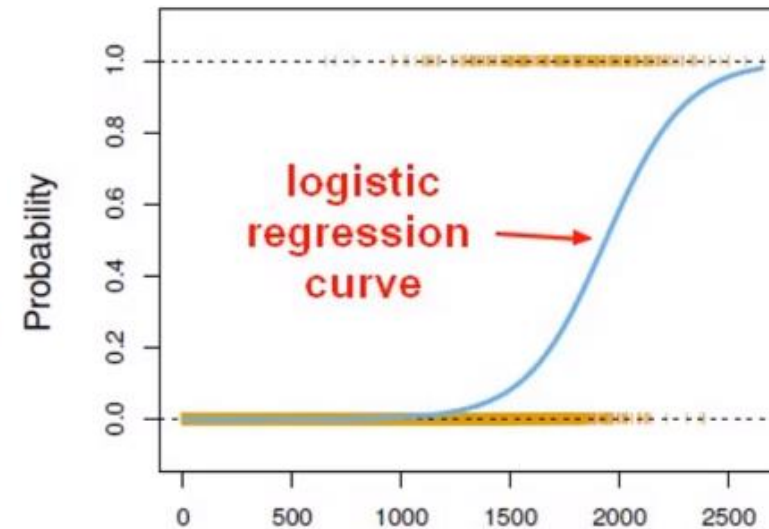
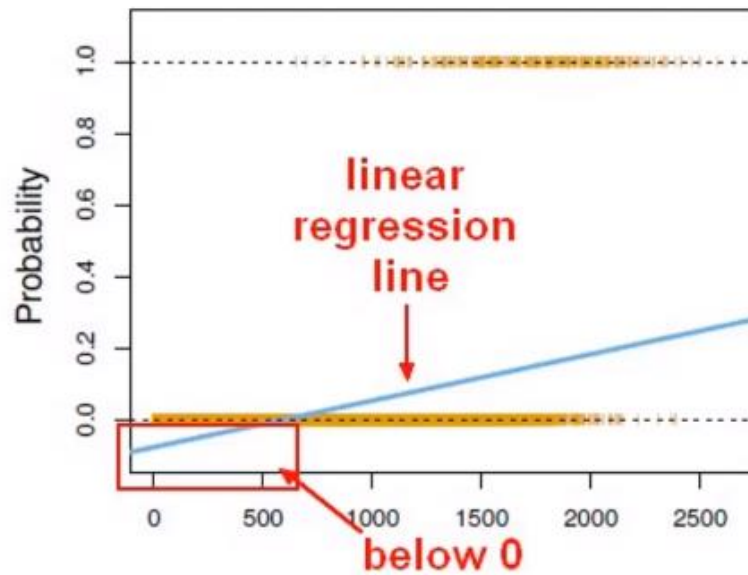
- It seems that simple regression is perfect for this task,
- But what are the caveats?

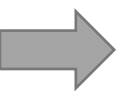




Linear Probability Model (LPM) vs Logistic Regression

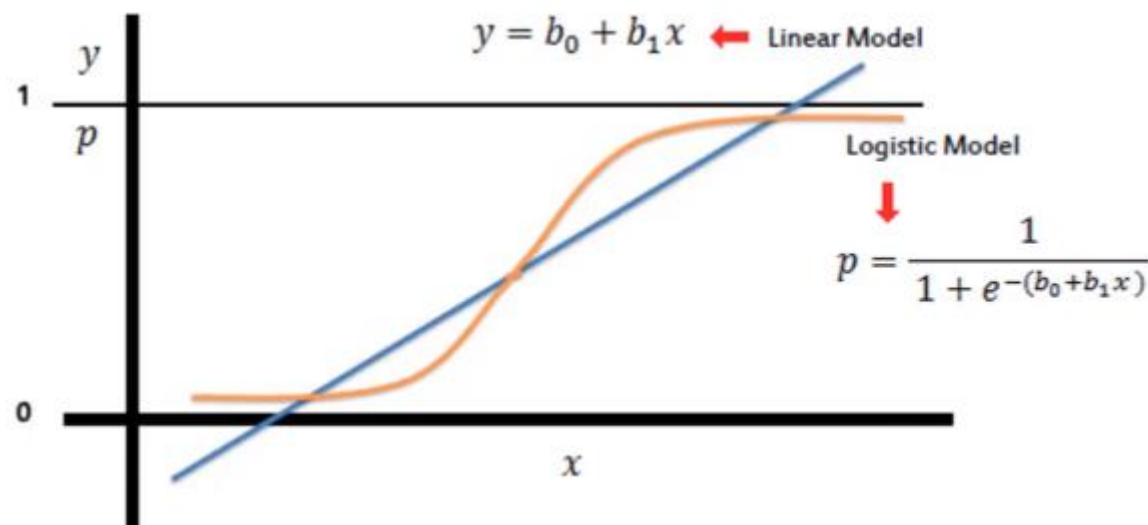
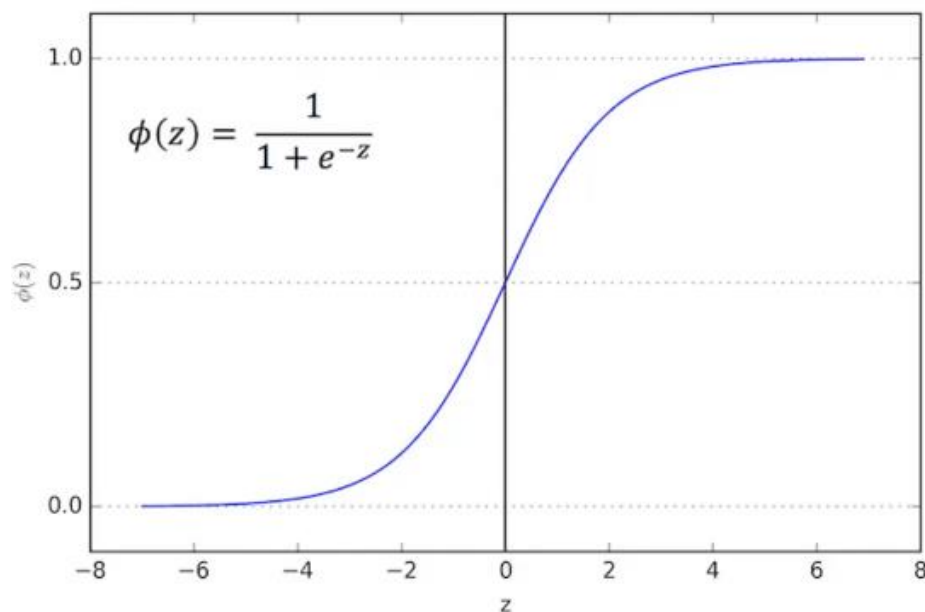
- What else? What if the data set is imbalanced?





Sigmoid Function

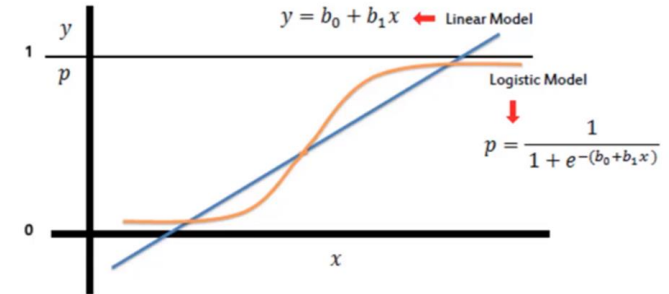
- We need a **monotone** mapping function that has a **range** of $[0,1]$



➔ Logistic Regression (Model)

- The model:

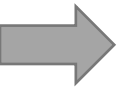
$$f_{w,b}(X) = \frac{1}{1+e^{-(WX+b)}}$$



- In case of two classes, $f_{w,b}(X) = \Pr(Y = 1|x) = p(x)$.
- A bit of rearrangement gives

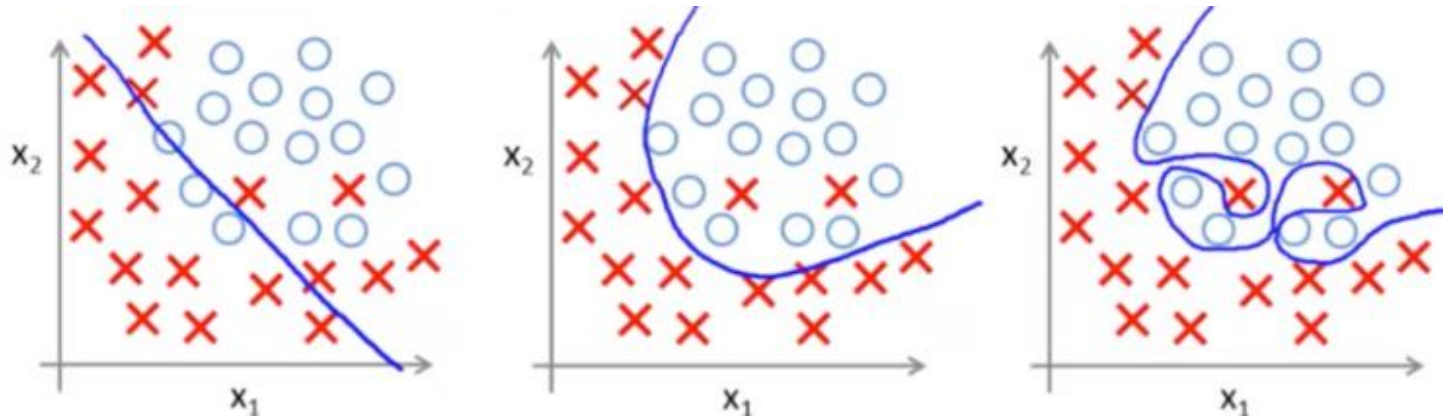
$$\text{Log} \left(\frac{p(x)}{1-p(x)} \right) = WX + b$$

- This monotone transformation is called the **log odds** or **logit** transformation of $p(x)$.
- Logistic regression ensures that our estimates always lie between 0 and 1



Logistic regression fit (Decision boundary)

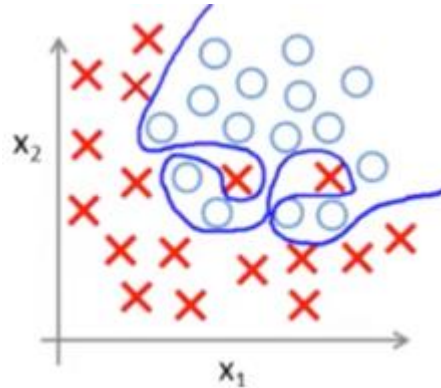
- Depending on how we define $WX + b$, we can get any of the following fits from logistic regression classifier.

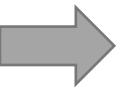


➔ Regularized Logistic regression

- Penalizing the logistic regression by adding L1, L2 or the combination penalty term.

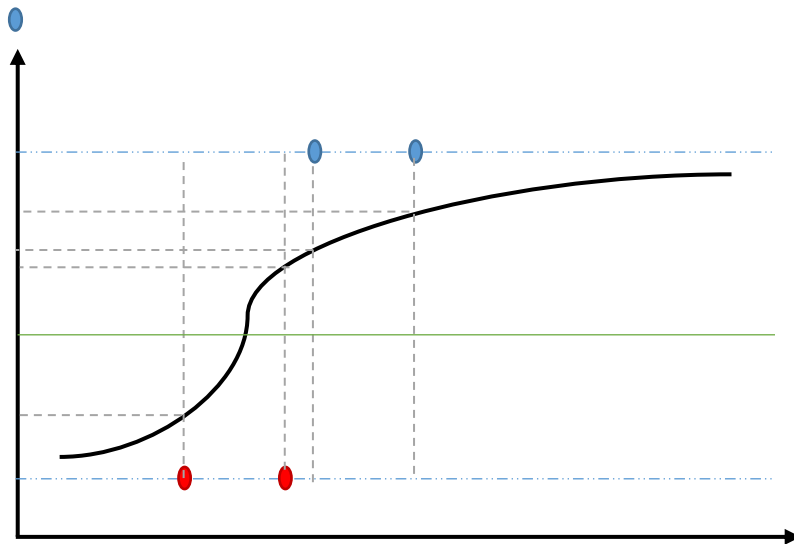
$$J_{w,b} = \text{logistic cost function} + \lambda_1 \sum_{j=1}^D |w_j| + \lambda_2 \sum_{j=1}^D w_j^2$$



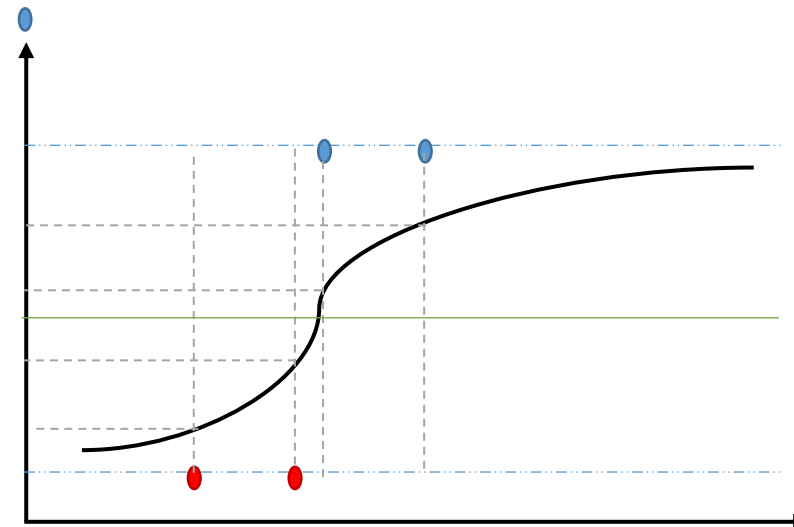


Logistic Regression Estimation (Maximum Likelihood)

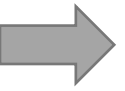
- In logistic regression, instead of minimizing the average loss, we **maximize** the **likelihood** of the training data according to our model. This is called **maximum likelihood estimation**.
- What is the likelihood function?
- The likelihood function describes the **joint probability of the observed data** as a function of the **parameters** of the model.



$$L = 0.9 * 0.8 * (1 - 0.75) * (1 - 0.2) = 0.144$$

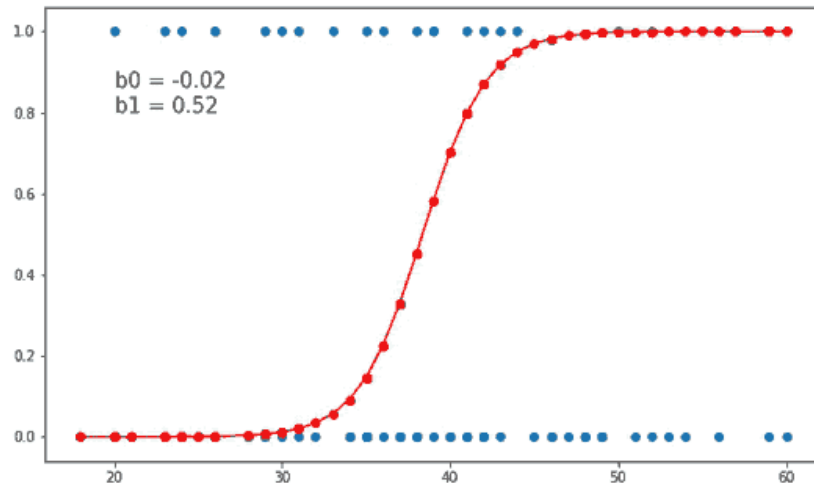


$$L = 0.85 * 0.6 * (1 - 0.4) * (1 - 0.2) = 0.244$$



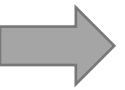
Logistic Regression (Maximum Likelihood)

- MLE in action!



$$f_{w^*,b^*}(X) = \frac{1}{1+e^{-(w^*X+b^*)}}$$

$$L_{w,b} = \prod_i f_{w,b}(x_i)^{y_i} (1 - f_{w,b}(x_i))^{1-y_i}$$



Logistic Regression (Objective function)

- Maximizing the likelihood function:

$$\text{Max} \{L_{w,b} = \prod_i f_{w,b}(x_i)^{y_i} (1 - f_{w,b}(x_i))^{1-y_i} \}$$

- Solution:** In practice, it is more convenient to maximize the **log-likelihood** function. This log-likelihood maximization, gives us w^* and b^* . There is **no closed form solution** to this optimization problem. We need to use **gradient descent**.
- We are now ready to make **predictions**.

$$f_{w^*,b^*}(X) = \frac{1}{1+e^{-(w^*X+b^*)}}$$

- Depending on how we define the probability threshold, we can classify the observations. In practice, the choice of the threshold could be different depending on the problem.

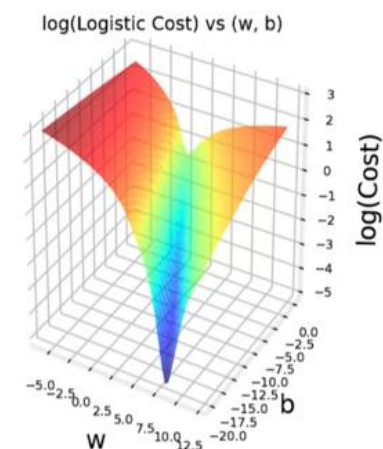
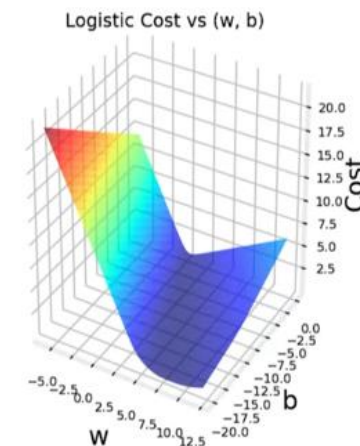
MLE and Gradient Descent

Simplified **loss** function

$$L(f_{\vec{w},b}(\vec{x}^{(i)}), y^{(i)}) = \begin{cases} -\log(f_{\vec{w},b}(\vec{x}^{(i)})) & \text{if } y^{(i)} = 1 \\ -\log(1 - f_{\vec{w},b}(\vec{x}^{(i)})) & \text{if } y^{(i)} = 0 \end{cases}$$

$$L(f_{\vec{w},b}(\vec{x}^{(i)}), y^{(i)}) = -y^{(i)}\log(f_{\vec{w},b}(\vec{x}^{(i)})) - (1 - y^{(i)})\log(1 - f_{\vec{w},b}(\vec{x}^{(i)}))$$

$$\begin{aligned} \text{cost } J(\vec{w}, b) &= \frac{1}{m} \sum_{i=1}^m [L(f_{\vec{w},b}(\vec{x}^{(i)}), y^{(i)})] \\ &= \frac{1}{m} \sum_{i=1}^m [y^{(i)}\log(f_{\vec{w},b}(\vec{x}^{(i)})) + (1 - y^{(i)})\log(1 - f_{\vec{w},b}(\vec{x}^{(i)}))] \end{aligned}$$





Logistic regression output for credit card default example

$$P(\text{default}|\text{bal}, \text{inc}) = \frac{1}{1 + e^{-(b + w_1(\text{bal}) + w_2(\text{inc}))}}$$



		Predictions (Decision boundary)	
		0 No Default	1 Default
Actual	0 No Default	TN=1933	FP=3
	1 Default	FN=44	TP=20

Classification metrics





Topics

- Classification performance metrics
 - a) Accuracy
 - b) Precision
 - c) Recall
 - d) F1 score
 - e) MCC
 - f) ROC and AUC

		Predictions	
		0 negative	1 positive
Actual	0 negative	TN	FP
	1 positive	FN	TP



Confusion Matrix

		Predictions	
		0 negative	1 positive
Actual	0 negative	TN	FP*
	1 positive	FN**	TP

FP* Type I error

FN** Type II error

		predicted class		
		class 1	class 2	class 3
actual class	class 1	True positives		
	class 2		True positives	
	class 3			True positives



Accuracy, Precision, Recall and F1score

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

		Predictions	
		0 negative	1 positive
Actual	0 negative	TN	FP
	1 positive	FN	TP

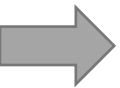
While **recall** expresses the ability to find all **relevant** instances in a dataset, **precision** expresses the proportion of the data points our model says was relevant were actually relevant.

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

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

$$F1\ Score = 2 * \frac{PR}{P + R}$$

F1 uses the **harmonic** mean instead of a simple average because it punishes extreme values.

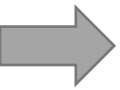


MCC (Matthews Correlation Coefficient)

$$\text{MCC} = \frac{\text{TP} \cdot \text{TN} - \text{FP} \cdot \text{FN}}{\sqrt{(\text{TP} + \text{FP})(\text{TP} + \text{FN})(\text{TN} + \text{FP})(\text{TN} + \text{FN})}}$$

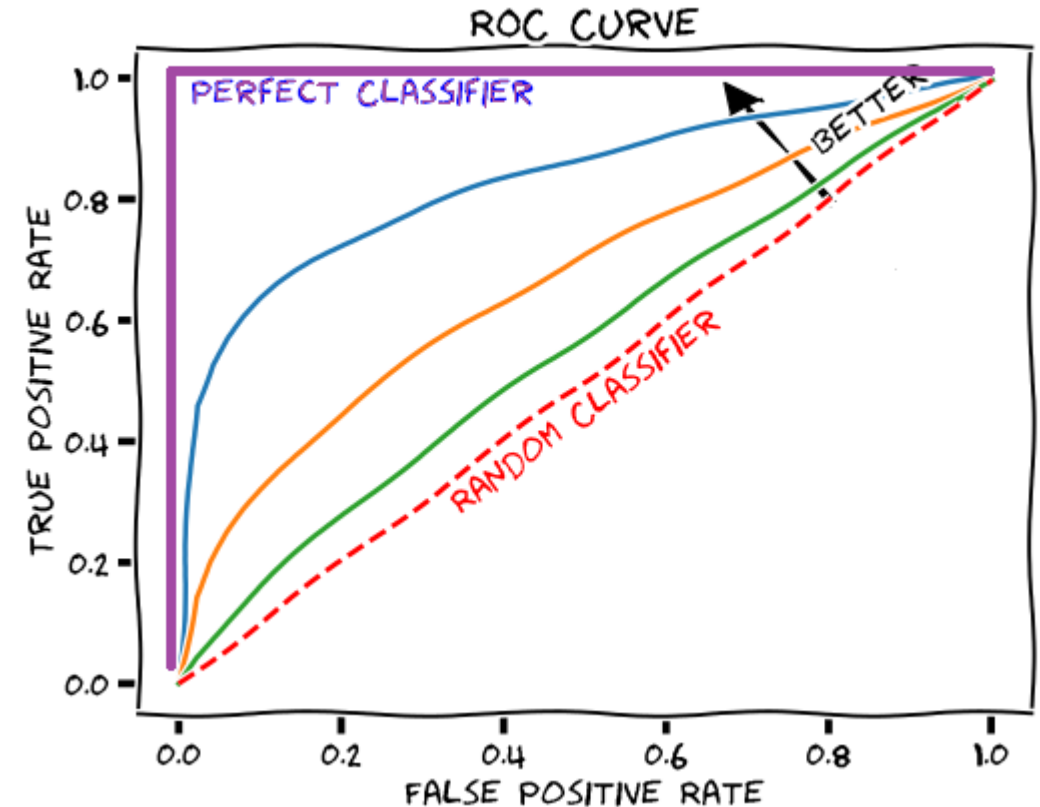
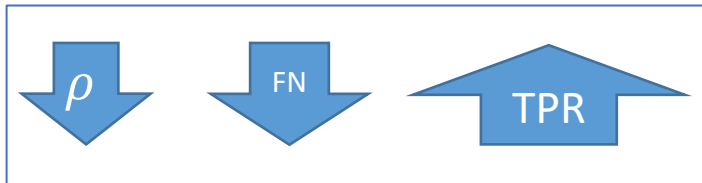
- Accuracy and error rates are misleading for imbalanced data sets.
- Precision, recall or even f1 score will not consider the **true negatives** (TN)
- MCC is one of the most **informative** metrics for any binary classifier.
- MCC returns a value between -1 and +1.
 - ❑ +1 represents a **perfect prediction**,
 - ❑ 0 represents **no better than a random** prediction,
 - ❑ -1 indicates **total misclassification**

		Predictions	
		0 negative	1 positive
Actual	0 negative	TN	FP
	1 positive	FN	TP



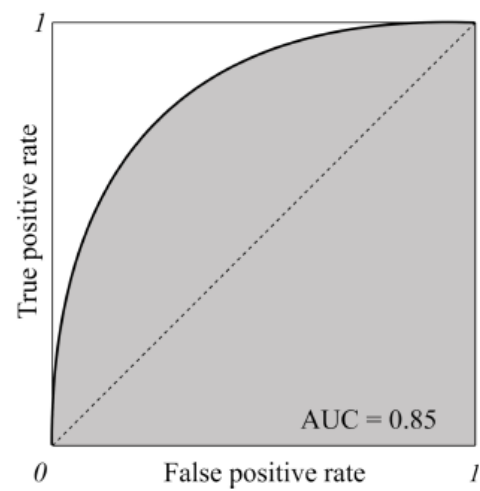
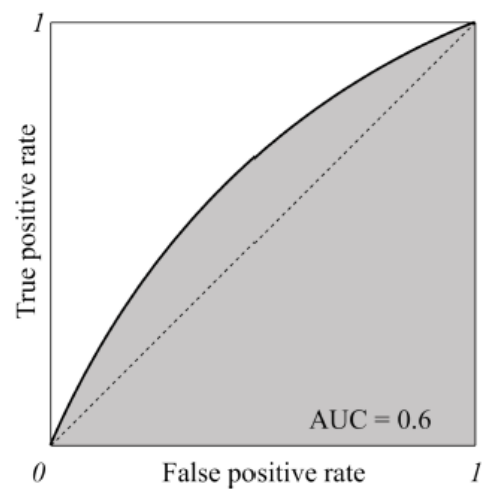
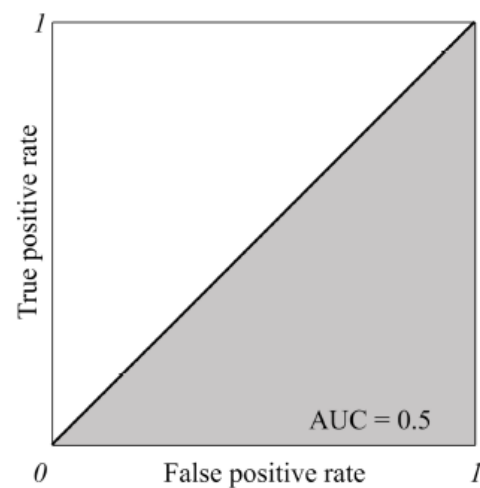
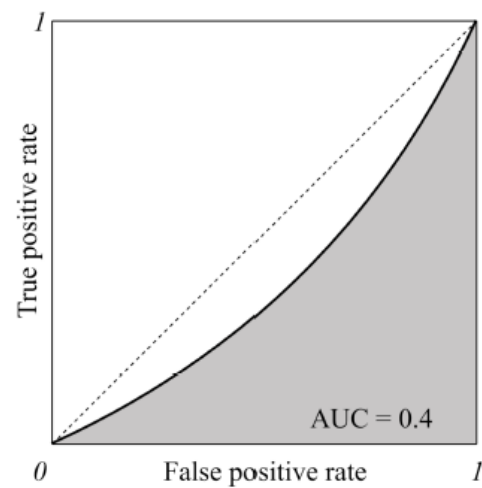
ROC (Receiver Operating Characteristic)

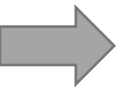
		Predictions		
		0 negative	1 positive	
Actual	0 negative	TN	FP	False Positive Rate = $\frac{FP}{FP + TN}$
	1 positive	FN	TP	True Positive Rate = $\frac{TP}{TP + FN}$





AUC





Some other classification metrics

		True condition			
Total population		Condition positive	Condition negative	Prevalence = $\frac{\Sigma \text{ Condition positive}}{\Sigma \text{ Total population}}$	Accuracy (ACC) = $\frac{\Sigma \text{ True positive} + \Sigma \text{ True negative}}{\Sigma \text{ Total population}}$
Predicted condition	Predicted condition positive	True positive	False positive, Type I error	Positive predictive value (PPV), Precision = $\frac{\Sigma \text{ True positive}}{\Sigma \text{ Predicted condition positive}}$	False discovery rate (FDR) = $\frac{\Sigma \text{ False positive}}{\Sigma \text{ Predicted condition positive}}$
	Predicted condition negative	False negative, Type II error	True negative	False omission rate (FOR) = $\frac{\Sigma \text{ False negative}}{\Sigma \text{ Predicted condition negative}}$	Negative predictive value (NPV) = $\frac{\Sigma \text{ True negative}}{\Sigma \text{ Predicted condition negative}}$
		True positive rate (TPR), Recall, Sensitivity, probability of detection, Power = $\frac{\Sigma \text{ True positive}}{\Sigma \text{ Condition positive}}$	False positive rate (FPR), Fall-out, probability of false alarm = $\frac{\Sigma \text{ False positive}}{\Sigma \text{ Condition negative}}$	Positive likelihood ratio (LR+) = $\frac{\text{TPR}}{\text{FPR}}$	Diagnostic odds ratio (DOR) = $\frac{\text{LR+}}{\text{LR-}}$
		False negative rate (FNR), Miss rate = $\frac{\Sigma \text{ False negative}}{\Sigma \text{ Condition positive}}$	Specificity (SPC), Selectivity, True negative rate (TNR) = $\frac{\Sigma \text{ True negative}}{\Sigma \text{ Condition negative}}$	Negative likelihood ratio (LR-) = $\frac{\text{FNR}}{\text{TNR}}$	
				F ₁ score = $2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}$	



Class Modules

- ✓ Module 1- Introduction to Deep Learning
- ✓ Module 2- Setting up Machine Learning Environment
- ✓ Module 3- Linear Regression (Econometrics approach)
- ✓ Module 4- Machine Learning Fundamentals
- ✓ Module 5- Linear Regression (Machine Learning approach)
- ✓ Module 6- Penalized Regression (Ridge, LASSO, Elastic Net)
- ✓ Module 7- Logistic Regression
- Module 8- K-Nearest Neighbors (KNN)
- Module 9- Classification and Regression Trees (CART)
- Module 10- Bagging and Boosting
- Module 11- Dimensionality Reduction (PCA)
- Module 12- Clustering (KMeans – Hierarchical)

