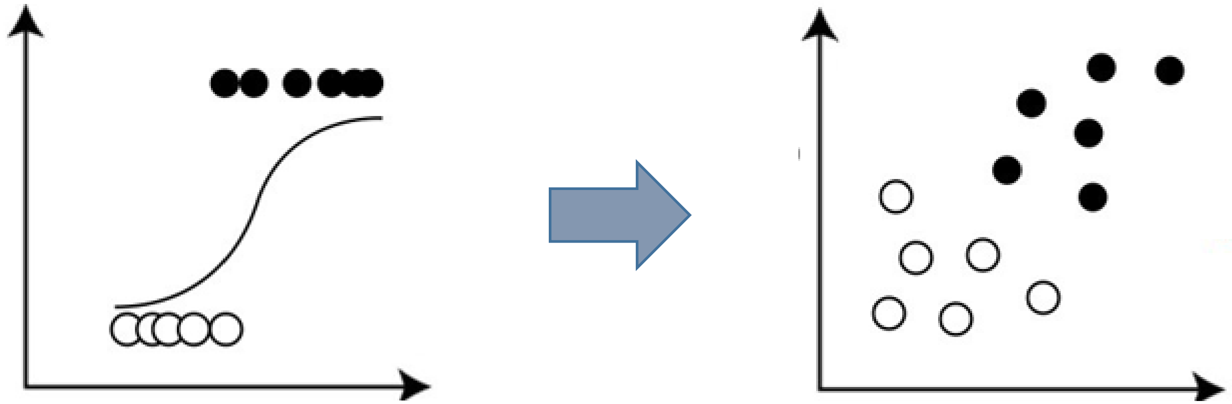
Module 7 – Logistic Regression



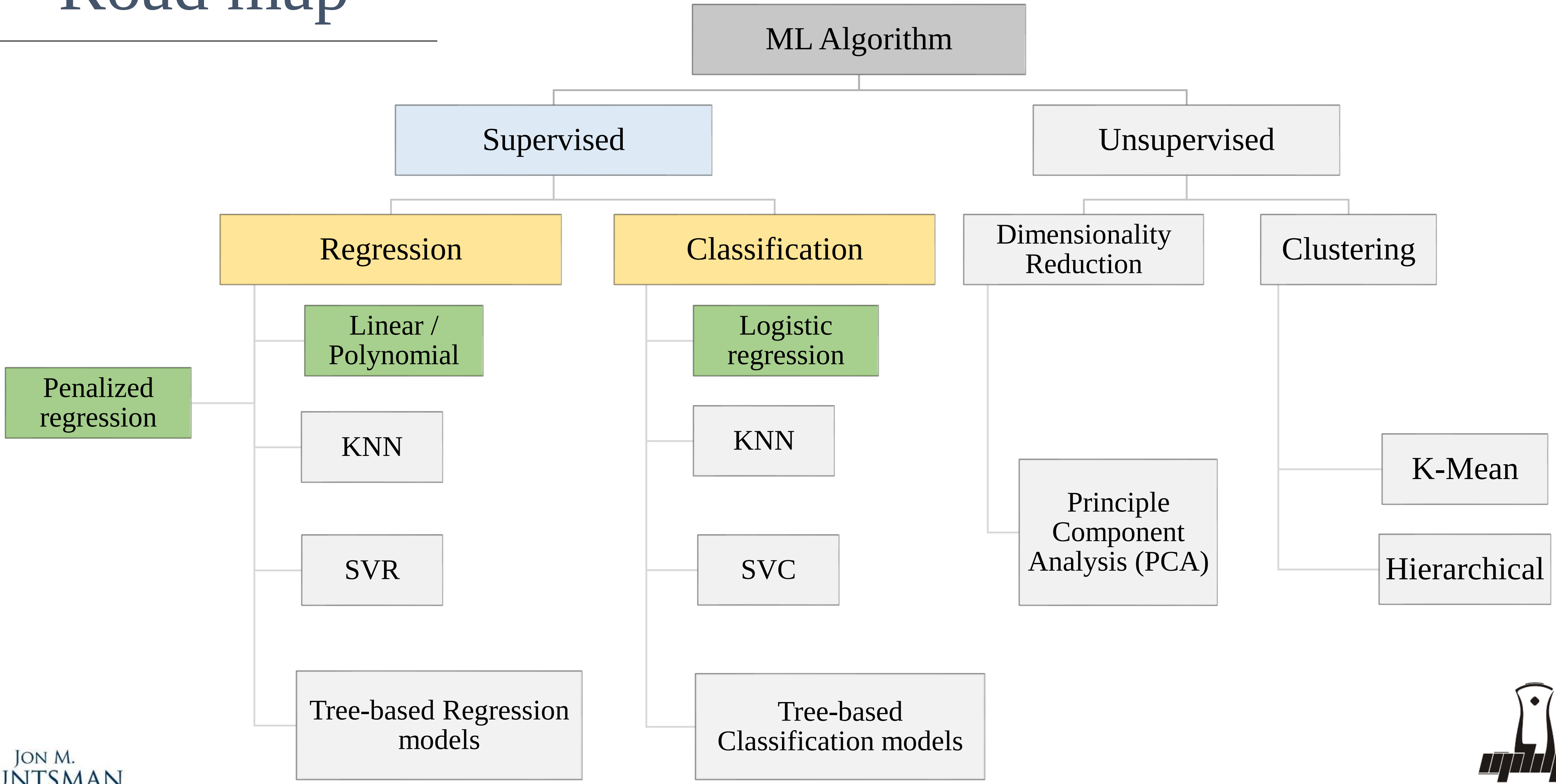
Class Modules

- Module 1- Introduction to Machine 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)

Jon M. Huntsman School of Business
Utah State University

Pedram Jahangiry

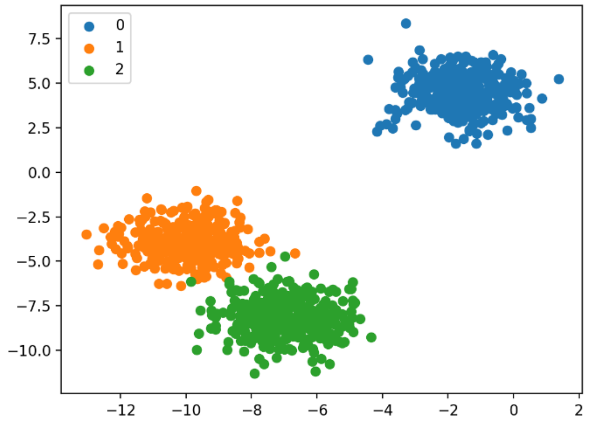
Road map


ML Algorithm
Supervised
Unsupervised
Regression
Classification
Dimensionality Reduction
Clustering
Linear / Polynomial
Penalized regression
KNN
SVR
Tree-based Regression models
Logistic regression
KNN
SVC
Tree-based Classification models
Principle Component Analysis (PCA)
K-Mean
Hierarchical

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
7. Classification performance metrics
   - (accuracy, precision, recall, f1 score, MCC, ROC and AUC)

Classification



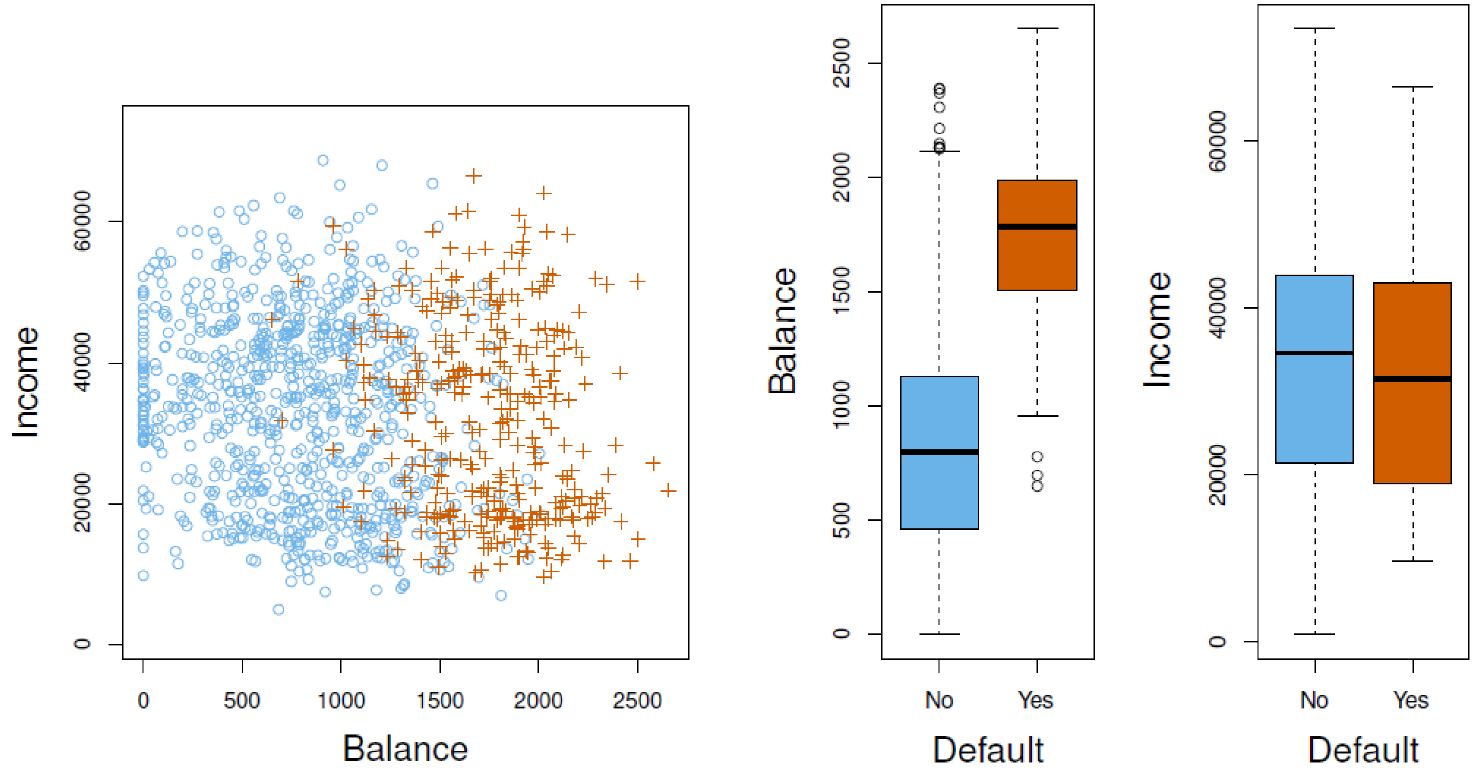
- Qualitative variables can be either nominal or ordinal.
- Qualitative variables are often referred to as **categorical.**
- **Classification** is the process of predicting categorical variables.
- Classification problems are quite common, perhaps even more than regression problems.

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.

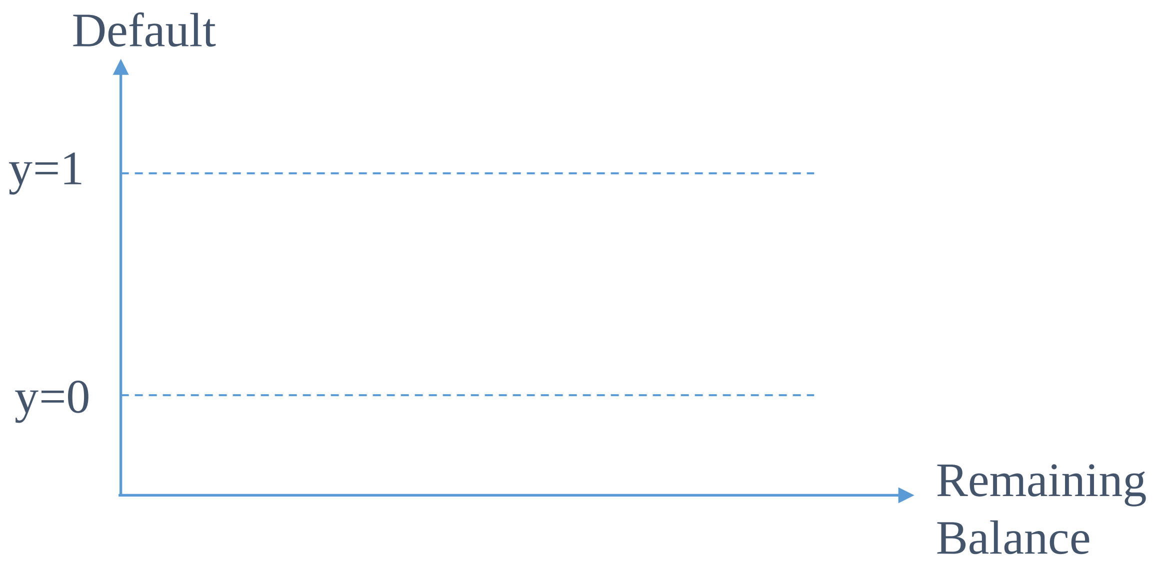


Linear Probability Model (LPM) vs Logistic Regression

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

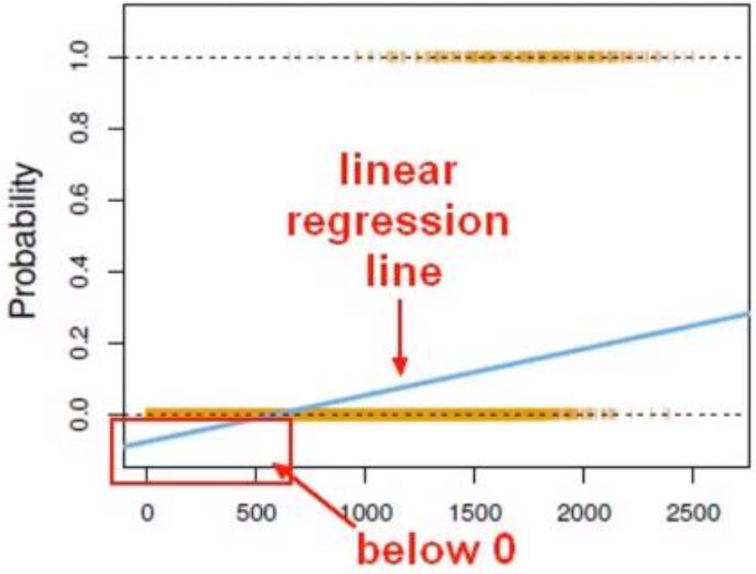
$$E(Y|bal) = \sum P(y_i|bal).y_i = \Pr(Y = 1|bal) = P(x) = \beta_0 + \beta_1 bal$$

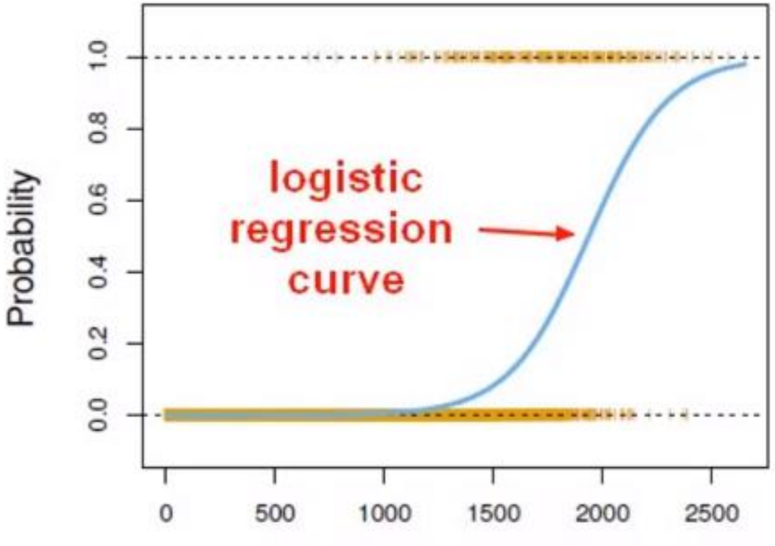
- It seems that simple regression is perfect for this task,
- But what are the caveats?



Linear Probability Model (LPM) vs Logistic Regression

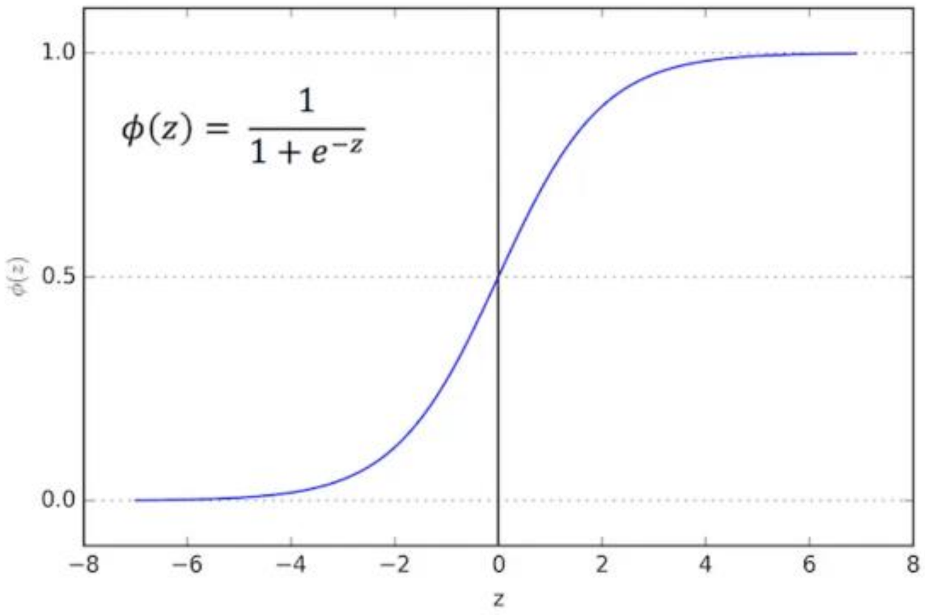
- What else? What if the data set is imbalanced?

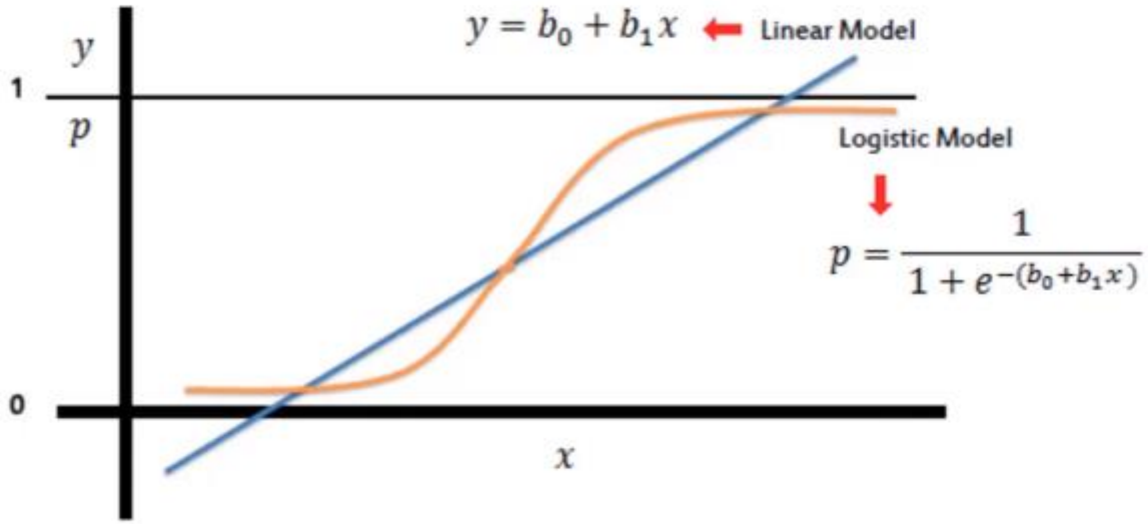




Sigmoid Function

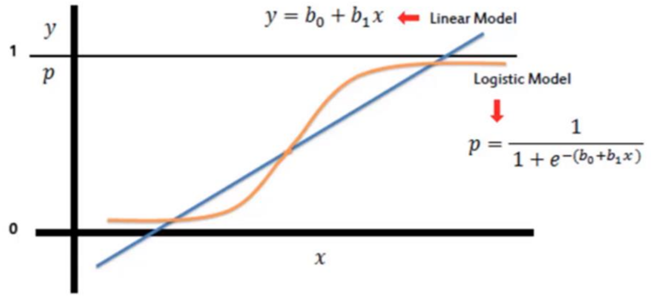
- We need a monotone mapping function that has a range of [0,1]





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

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

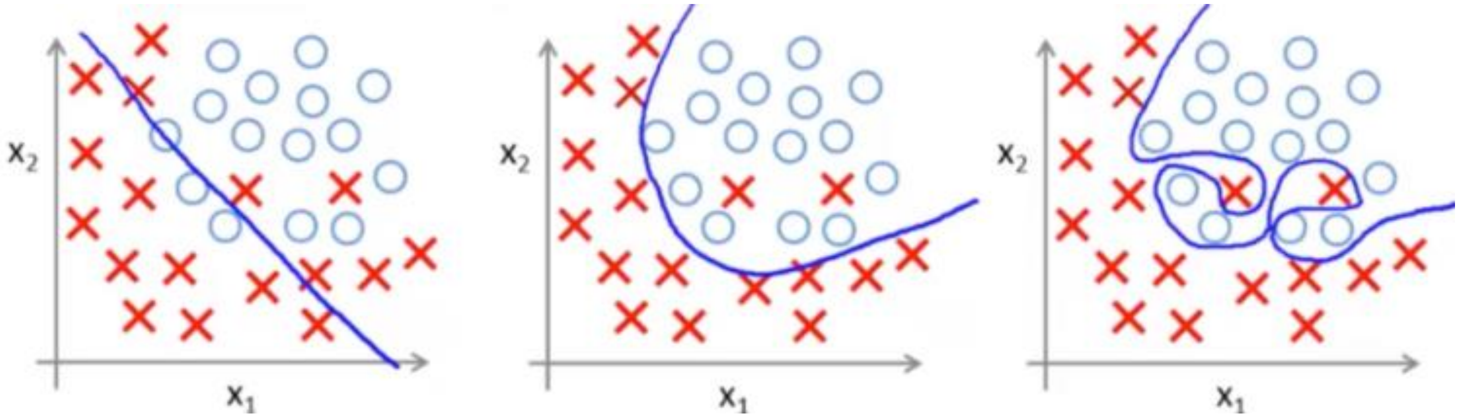
- This monotone transformation is called the log odds or logit transformation of $p(x)$.
- Logistic regression ensures that our estimates always lie between 0 and 1

Jon M. Huntsman School of Business
Utah State University

Pedram Jahangiry

Logistic regression fit (Decision boundary)

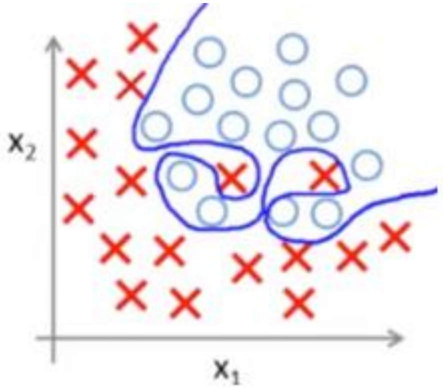
- Depending on how we define $WX + b$, we can get any of the following fits from logistic regression classifier.



Regularized Logistic regression

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

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

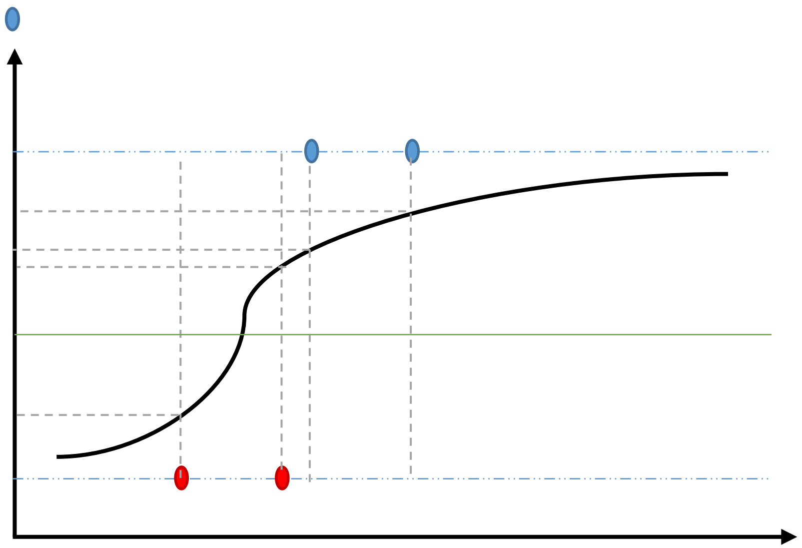


JON M. HUNTSMAN SCHOOL OF BUSINESS
UtahStateUniversity

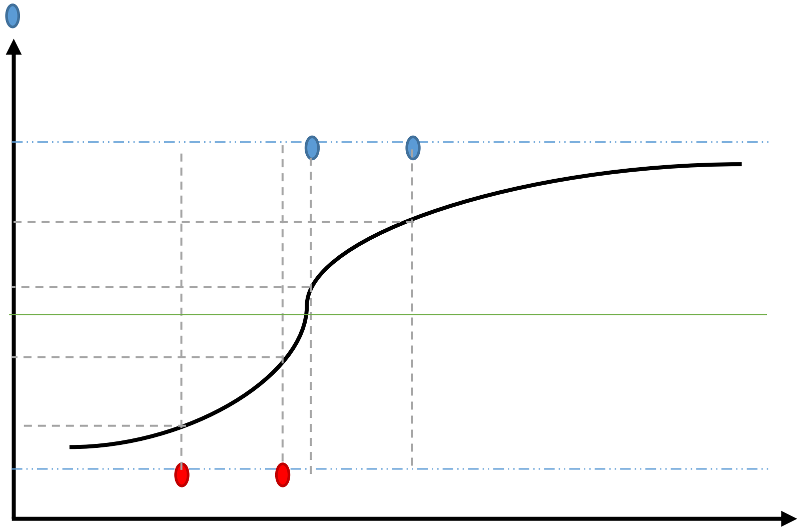
Pedram Jahangiry

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.



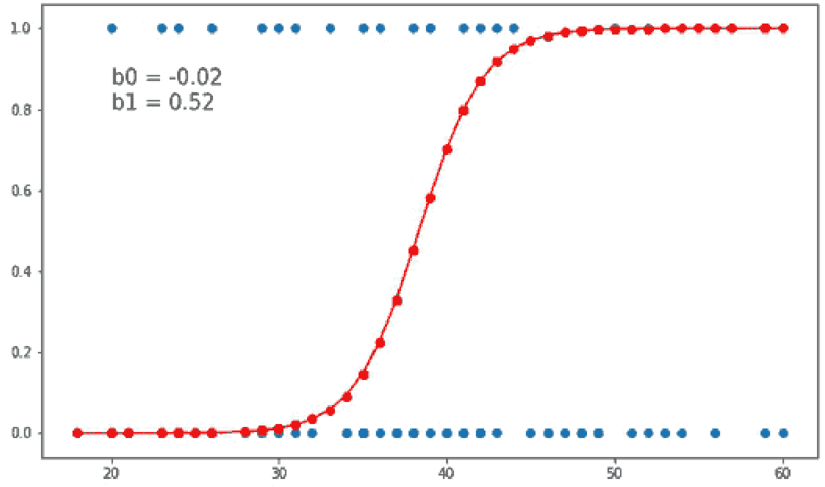
$L = 0.9 * 0.8 \ * (1 - 0.75) * (1 - 0.2) = 0.144$



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

Logistic Regression (Maximum Likelihood)

- MLE in action!



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

$$L_{w,b} = \prod_{i} f_{w,b}(x_i)^{y_i}\left(1 - f_{w,b}(x_i)\right)^{1-y_i}$$

JON M. HUNTSMAN SCHOOL OF BUSINESS
UtahStateUniversity

Pedram Jahangiry

Logistic Regression (Objective function)

- Maximizing the likelihood function:

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

- **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**.

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

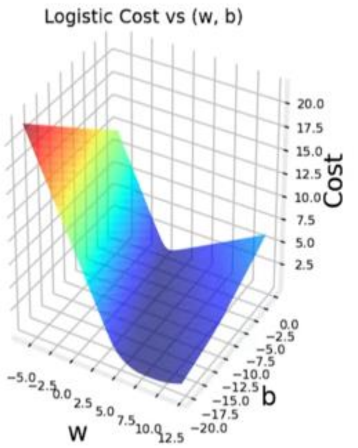
- Depending on how we define the probability threshold, we can classify the observations. In practice, the choice of the threshold could be different depending on the problem.

MLE and Gradient Descent

Simplified loss function

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

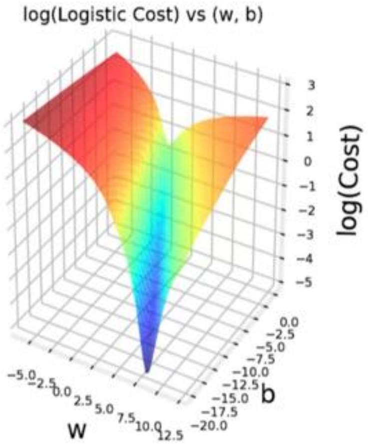
$$\boxed{L\left(f_{\vec{w},b}(\vec{x}^{(i)}), y^{(i)}\right) = -\,y^{(i)}\log\left(f_{\vec{w},b}(\vec{x}^{(i)})\right) - (1 - y^{(i)})\log\left(1 - f_{\vec{w},b}(\vec{x}^{(i)})\right)}$$



cost

$$J(\vec{w}, b) = \frac{1}{m}\sum_{i=1}^{m}\left[\underbrace{L\left(f_{\vec{w},b}(\vec{x}^{(i)}), y^{(i)}\right)}\right]$$

$$= -\frac{1}{m}\sum_{i=1}^{m}\left[y^{(i)}\log\left(f_{\vec{w},b}(\vec{x}^{(i)})\right) + (1 - y^{(i)})\log\left(1 - f_{\vec{w},b}(\vec{x}^{(i)})\right)\right]$$

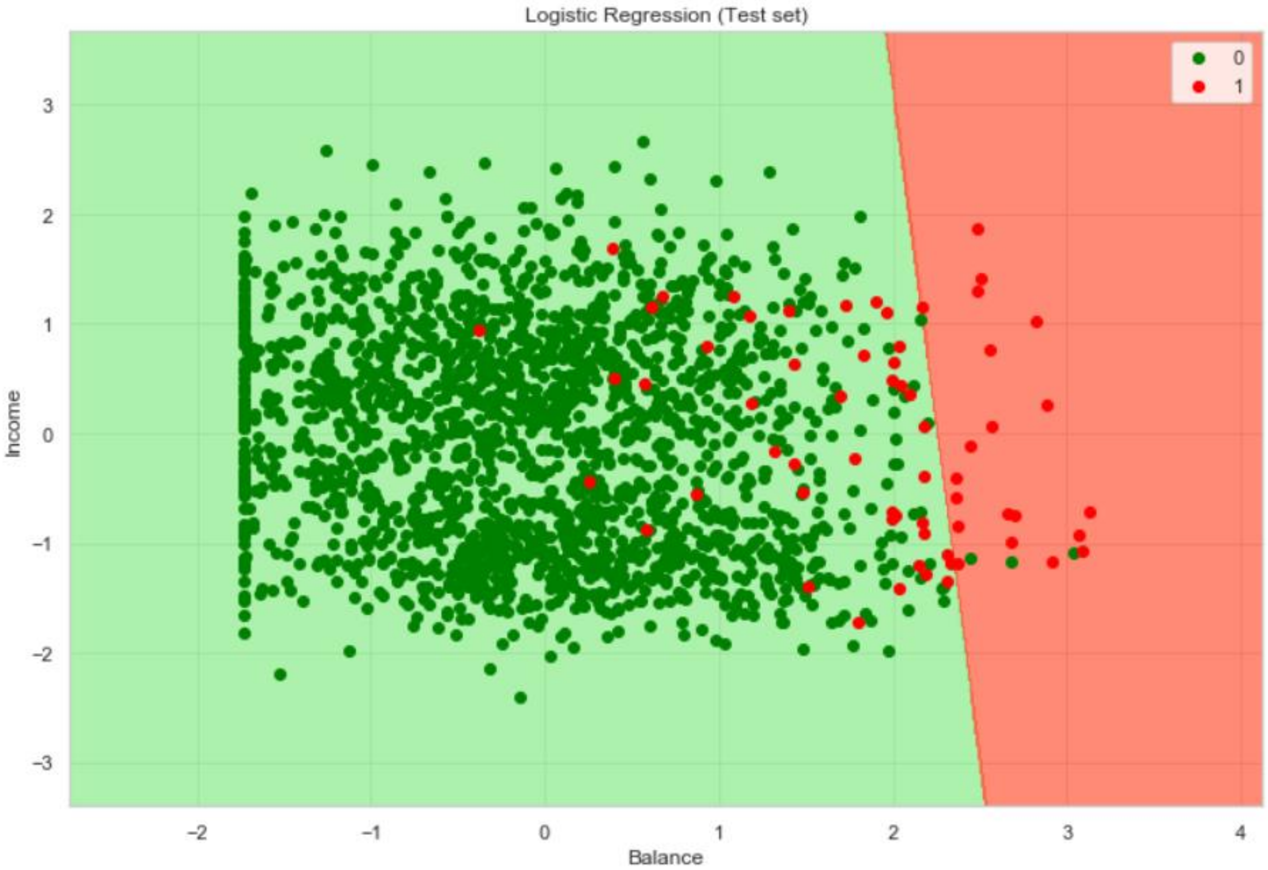


Jon M. Huntsman School of Business
Utah State University

Pedram Jahangiry

Logistic regression output for credit card default example

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



		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 eror

		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

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

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

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.

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

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

MCC (Matthews Correlation Coefficient)

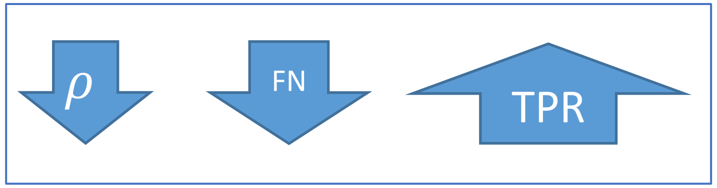
$$\mathrm{MCC} = \frac{\mathrm{TP} \cdot \mathrm{TN} - \mathrm{FP} \cdot \mathrm{FN}}{\sqrt{(\mathrm{TP} + \mathrm{FP})(\mathrm{TP} + \mathrm{FN})(\mathrm{TN} + \mathrm{FP})(\mathrm{TN} + \mathrm{FN})}}$$

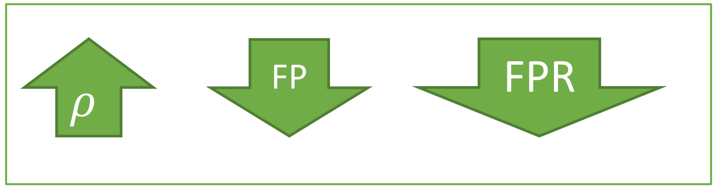
- Accuracy and error rates are misleading for imbalanced data sets.
- Precision, recall or even f1 score will not consider the true negatives (TN)
- MCC is one of the most informative metrics for any binary classifier.
- MCC returns a value between -1 and +1.
  - +1 represents a perfect prediction,
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

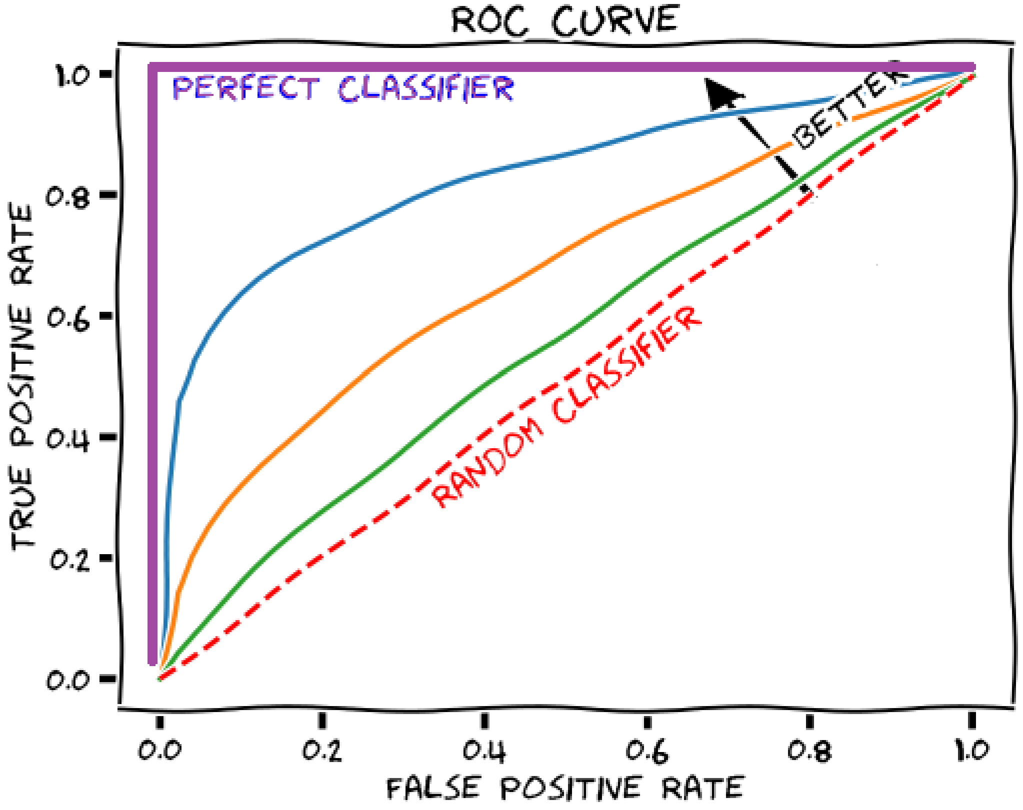
		Predictions	
		0 negative	1 positive
Actual	0 negative	TN	FP
	1 positive	FN	TP

ROC (Receiver Operating Characteristic)

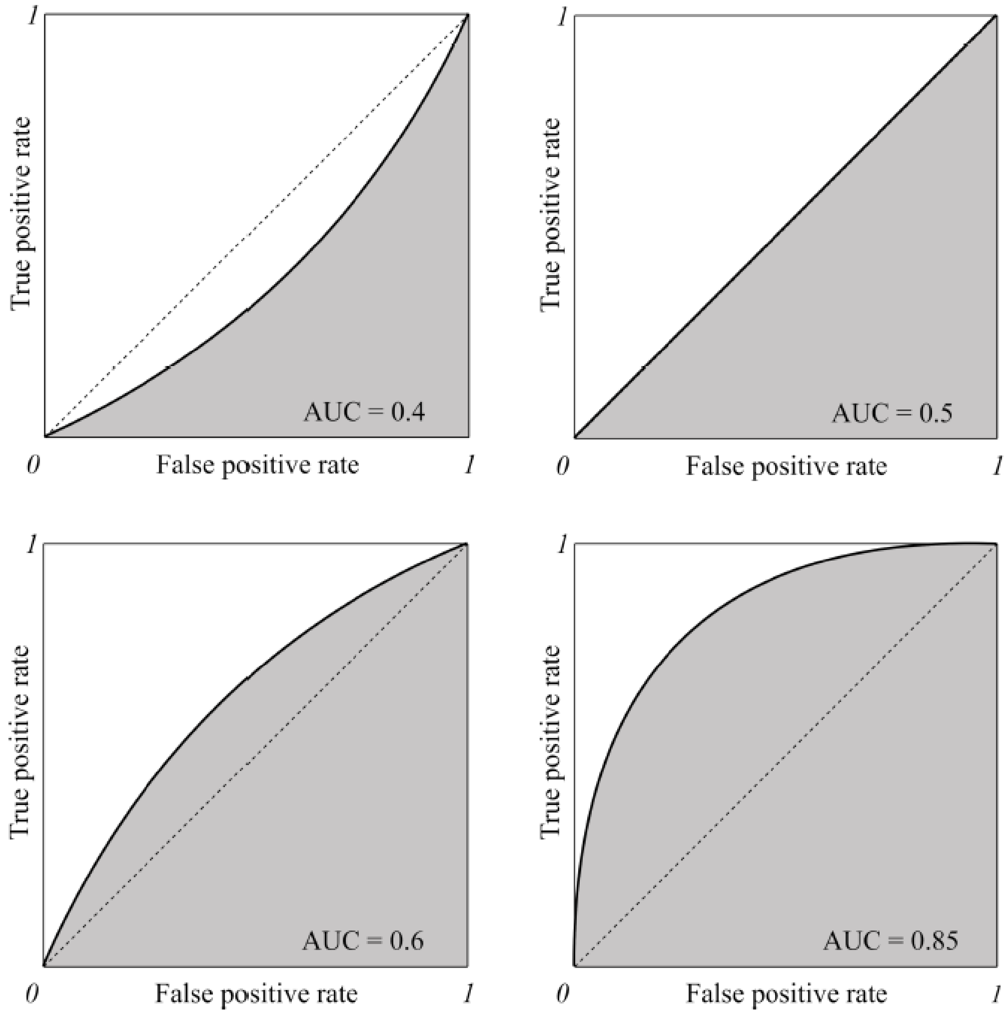
		Predictions: 0 negative	Predictions: 1 positive	
Actual	0 negative	TN	FP	$False\ Positive\ Rate = \frac{FP}{FP + TN}$
Actual	1 positive	FN	TP	$True\ Positive\ Rate = \frac{TP}{TP + FN}$







AUC


True positive rate
False positive rate
AUC = 0.4
AUC = 0.5
AUC = 0.6
AUC = 0.85
0
1

Some other classification metrics

<table>
<tr><td colspan="2"></td><td colspan="2">True condition</td><td colspan="3"></td></tr>
<tr><td></td><td>Total population</td><td>Condition positive</td><td>Condition negative</td><td>Prevalence $= \frac{\Sigma \text{ Condition positive}}{\Sigma \text{ Total population}}$</td><td colspan="2">Accuracy (ACC) = $\frac{\Sigma \text{ True positive} + \Sigma \text{ True negative}}{\Sigma \text{ Total population}}$</td></tr>
<tr><td rowspan="2">Predicted condition</td><td>Predicted condition positive</td><td>True positive</td><td>False positive, Type I error</td><td>Positive predictive value (PPV), Precision = $\frac{\Sigma \text{ True positive}}{\Sigma \text{ Predicted condition positive}}$</td><td colspan="2">False discovery rate (FDR) = $\frac{\Sigma \text{ False positive}}{\Sigma \text{ Predicted condition positive}}$</td></tr>
<tr><td>Predicted condition negative</td><td>False negative, Type II error</td><td>True negative</td><td>False omission rate (FOR) = $\frac{\Sigma \text{ False negative}}{\Sigma \text{ Predicted condition negative}}$</td><td colspan="2">Negative predictive value (NPV) = $\frac{\Sigma \text{ True negative}}{\Sigma \text{ Predicted condition negative}}$</td></tr>
<tr><td colspan="2" rowspan="2"></td><td>True positive rate (TPR), Recall, Sensitivity, probability of detection, Power = $\frac{\Sigma \text{ True positive}}{\Sigma \text{ Condition positive}}$</td><td>False positive rate (FPR), Fall-out, probability of false alarm $= \frac{\Sigma \text{ False positive}}{\Sigma \text{ Condition negative}}$</td><td>Positive likelihood ratio (LR+) $= \frac{\text{TPR}}{\text{FPR}}$</td><td rowspan="2">Diagnostic odds ratio (DOR) $= \frac{\text{LR+}}{\text{LR−}}$</td><td rowspan="2">$F_1$ score = $2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}$</td></tr>
<tr><td>False negative rate (FNR), Miss rate $= \frac{\Sigma \text{ False negative}}{\Sigma \text{ Condition positive}}$</td><td>Specificity (SPC), Selectivity, True negative rate (TNR) $= \frac{\Sigma \text{ True negative}}{\Sigma \text{ Condition negative}}$</td><td>Negative likelihood ratio (LR−) $= \frac{\text{FNR}}{\text{TNR}}$</td></tr>
</table>

Class Modules

- ✓ Module 1- Introduction to Machine 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)

Jon M. Huntsman School of Business
UtahStateUniversity

Pedram Jahangiry