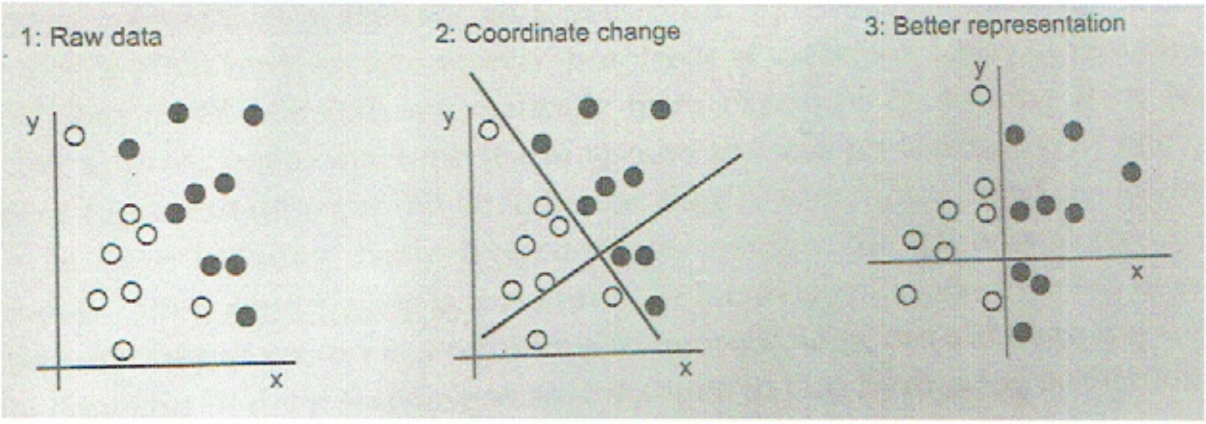
Artificial Intelligence/Machine Learning/Deep Learning:
'Bridging the Skills Gap'

Lesson 3: Classification and Classification Performance Metrics

In this lesson we will briefly repeat what we have seen in terms of classification in lesson 1 and we will go deeper into the performance KPIs for a classifier.

Assume that we want to develop an algorithm that takes as input the (x,y) coordinates of a point and output whether the point is to be black or white. The performance of the algorithm can be defined as the % of the points that are correctly classified.



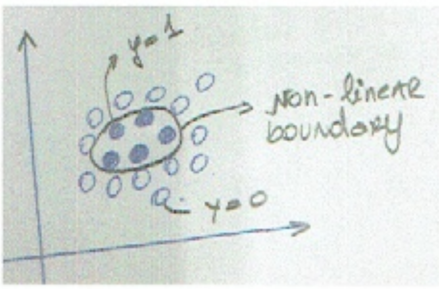
We need a transformation that cleanly separates the black from the white dots. In this case the transformation is simply a coordinate change.

Solution:

- black points: x>0
- white points x<0.

In this case we made the coordinate change by hand but in ML the computer will search for a better representation of the data.

In above example the boundary is linear, but boundaries can also be **non-linear**. In **binary classification** we have only 2 classes: example no tumor (negative class)/tumor (positive class)



We can also have **Multi-Class Classification,** for example recognizing handwritten digits $0 \rightarrow 9$. This is often called **Softmax** but more about this later.

For classification, MSE is unlikely to give a good solution as we are not trying to predict a continuous variable. We need a classifier that converts a linear score to a probability with interval [0,1]

We will look at a bunch of classifiers in this course including:

Classification Algorithm	Coding Exercise (Yes/No)	Comment
K-Nearest Neighbors (KNN)	√	
Perceptron – Multi-Layer Perceptron	X	Perceptron is the simplest form of a Neural Network.
Naïve Bayes	√	
Logistics Regression	√	
Support Vector Machines and Kernel Functions	√	
Decision Trees	√	
Neural Networks	√	Will be discussed during multiple NN sessions

In previous session we discussed the example of classifying credit card transactions into fraud or not-fraud classes. Fraud cases are typically rare and are in the range of 0.3%. In this case we deal with a dataset that is very imbalanced. The machine could perform at 99.7% accuracy if it would decide on not-fraud for all cases. However this is not what we want. We want the machine to find the 0.3% fraud cases. Therefore we will use other metrics as classifier performance KPIs.

Classification Performance Metrics:

- Confusion Matrix
- AUC-ROC = Area Under the Curve - Receiver Operating Characteristic
- F1-Score

CONFUSION MATRIX

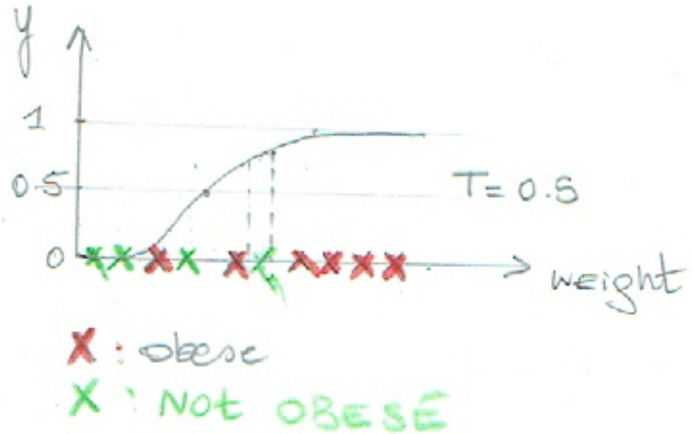
→ BINARY clasification
→ 2×2 MATRIX

<u>example</u>. classifier that predicts if A person IS OBESE given his/her weight

↓

1 feature x

y label
$y \geq 0.5$ obese
$y < 0.5$ NOT OBESE



Predict

TRAINING samples = m

ACTUAL

10	NOT OBESE	OBESE	
NOT OBESE	TN 3	FP 1	4
OBESE	FN 1	TP 5	6
	4	6	

Type 1 ERROR

Type 2 ERROR

TP: TRUE Positive
FP: False Positive
TN: TRUE NEGAtive
FN: FALSE NEGAtive

<u>Accuracy</u> $= \dfrac{TP+TN}{m}$

<u>Recall | sensitivity | TRUE Pos rate</u>

$= \dfrac{5 \to TP}{6 \to TP+FN}$

<u>Precission</u>

$= \dfrac{TP}{TP+FP} = \dfrac{5}{6}$

ROC . RECEIVER OPERATOR characteristic

so far → Treshhold $T = 0.5$

what happens if we change T ?

FPR = predicted obese but person is actually NOT obese

↳ $\dfrac{1}{4} = 25\%$

T = 0 → ALL OBESE

	N	O
N	0	4
O	0	6

FPR = 1
TPR = 1

xx

	N	O
N	2	2
O	0	6

FPR = 0.5
TPR = 1

xxx

	N	O
N	2	2
O	1	5

FPR = 0.5
TPR = 5/6

xxxx

	N	O
N	3	1
O	1	5

FPR = 1/4
TPR = 5/6

xxxxx

	N	O
N	3	1
O	2	4

FPR = 0.25
TPR = 4/6

xxxxxx

	N	O
N	4	0
O	2	4

FPR = 0
TPR = 4/6

xxxxxxx

	N	O
N	4	0
O	3	3

FPR = 0
TPR = 0.5

xxxxxxxx

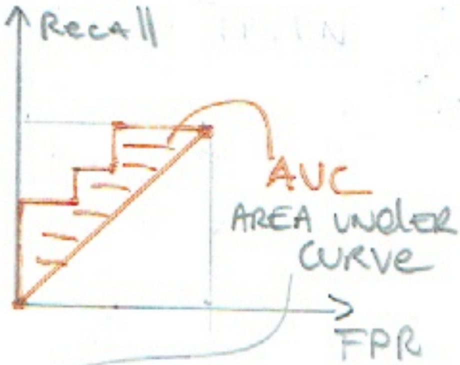
	N	O
N	4	0
O	4	2

FPR =
TPR =

	N	O
N	4	0
O	6	0

→ All Not OBESE

FPR = 0
TPR = 0



$= \frac{FP}{FP + TN}$

ROC
↓
How many FNs ARE you willing to Accept?

to compare classifiers!

Trade off Recall / Precision

ex. binary classifier : cancer / NO CANCER

FP : Predict 'cancer' whereas person is healty.

FN : Predict 'NO CANCER' where person has cancer)

Recall ≈ 1 ← you want to avoid FNs !! ←
→ Avoid Type II !!

$$\boxed{F_1\text{-}SCORE = \frac{2\ Recall \cdot Precission}{Recall + Precission}}$$