Institute of Data

2021

© 2021 Institute of Data

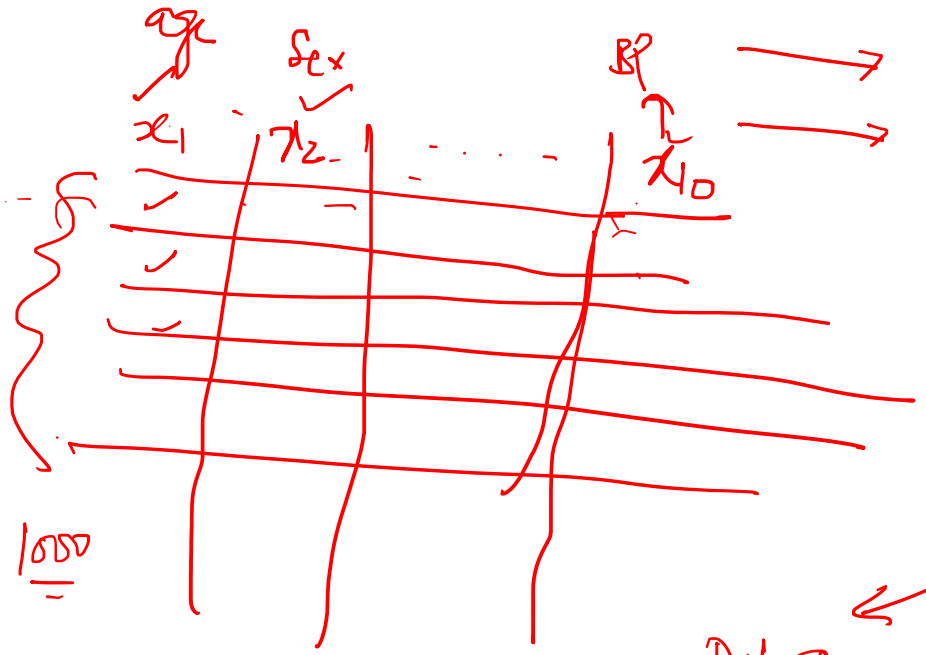
Data Science and AI

Module 5

Supervised ML: Classification

© 2021 Institute of Data

Data set → 10 features



Train Data

Model (Regression)

.fit()

$w_1, w_2, \ldots, w_{10}$

10 Numbers

Parameters (weights)

$$\boxed{\hat{y} = \underline{w}^T \underline{x}}$$

Linear Regression

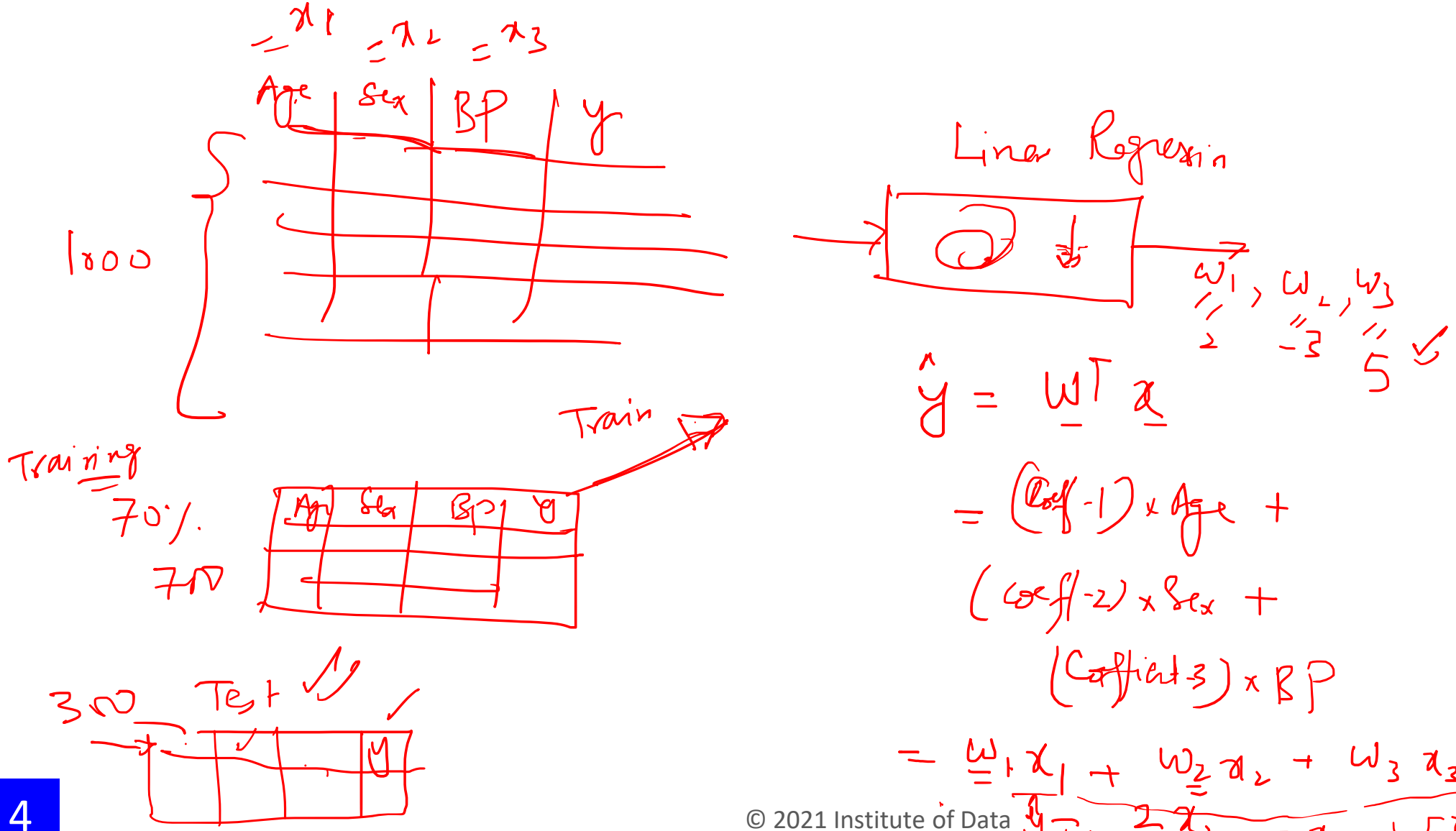
$$\hat{y} = w_1 x_1 + w_2 x_2 + \cdots + w_{10} x_{10}$$

Diabet

x_1 → Age, x_2 → Sex, x_{10} → BP

→ estimate of Prediction

© 2021 Institute of Data



$$\hat{y} = \underline{w}^T \underline{x}$$

$$= (\text{Coeff-1}) \times \text{Age} + (\text{coeff-2}) \times \text{Sex} + (\text{Coefficient-3}) \times \text{BP}$$

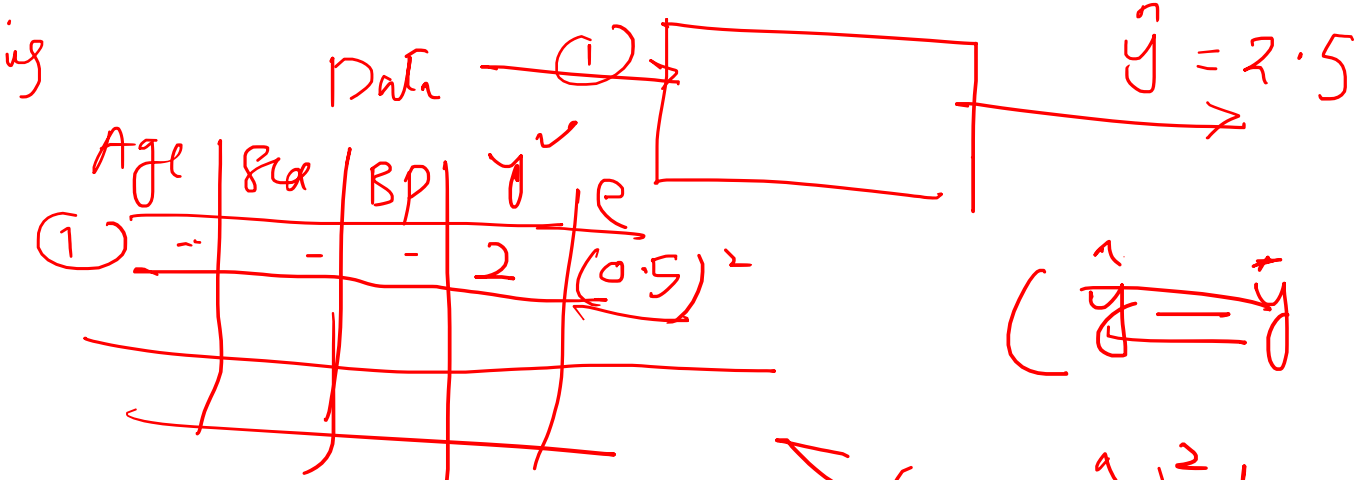
$$= \underline{w}_1 x_1 + \underline{w}_2 x_2 + w_3 x_3$$

$$\boxed{\hat{y} = 2x_1 - 3x_2 + 5x_3}$$

© 2021 Institute of Data

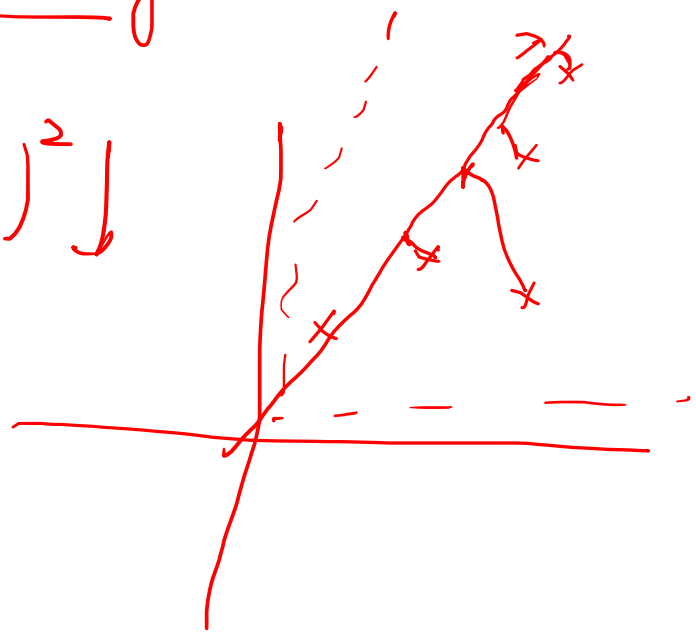
fit ( )

Training Data



$(\hat{y} = y)$

Objective / Cost function $= \sum (y_i - \hat{y}_i)^2$



© 2021 Institute of Data

Institute of Data

Agenda: Module 5

Regression → Real value

v/s

Classification

↳ "Category"

- Introduction to **Classification**
- (1) **Logistic Regression** → "Classification Algo"
- Matrix → **Evaluating** Classification Results
- (2) **Neural Networks** → Deep Neural Network
- (3) **Support Vector Machines** (SVM), ↓ "Perceptron"
- (4) **Bayesian Inference**
- **Applications**

© 2021 Institute of Data

Classification

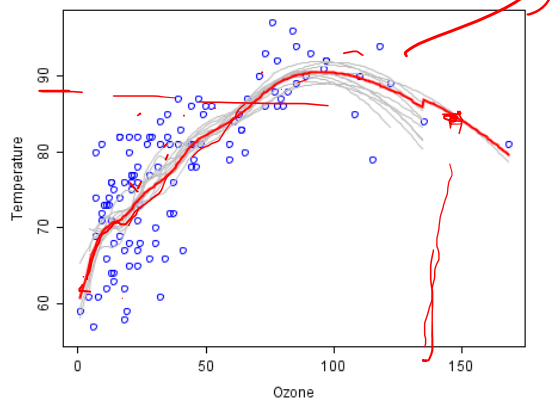
$\hat{y} = \underline{w}^T \underline{x} = w_1 x_1 + \cdots + w_n x_n$

Regression

- train a model by fitting data to a **continuous** response
- *predict* ***continuous*** *numbers*

Real

Data → $f(x)$



Regression

Non-seperable

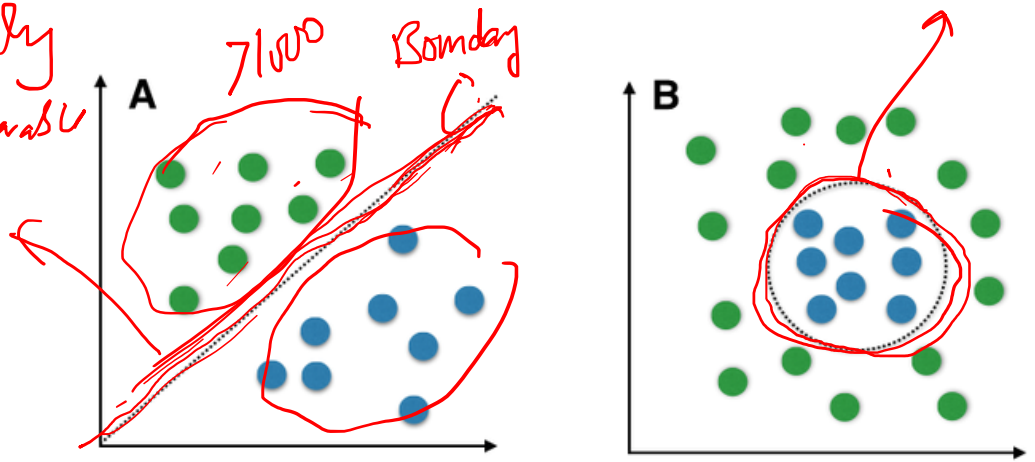
Classification

- train a model by fitting data to a **discrete** response
- predict **class membership**

Linearly seperable

7/000

Bombay



© 2021 Institute of Data

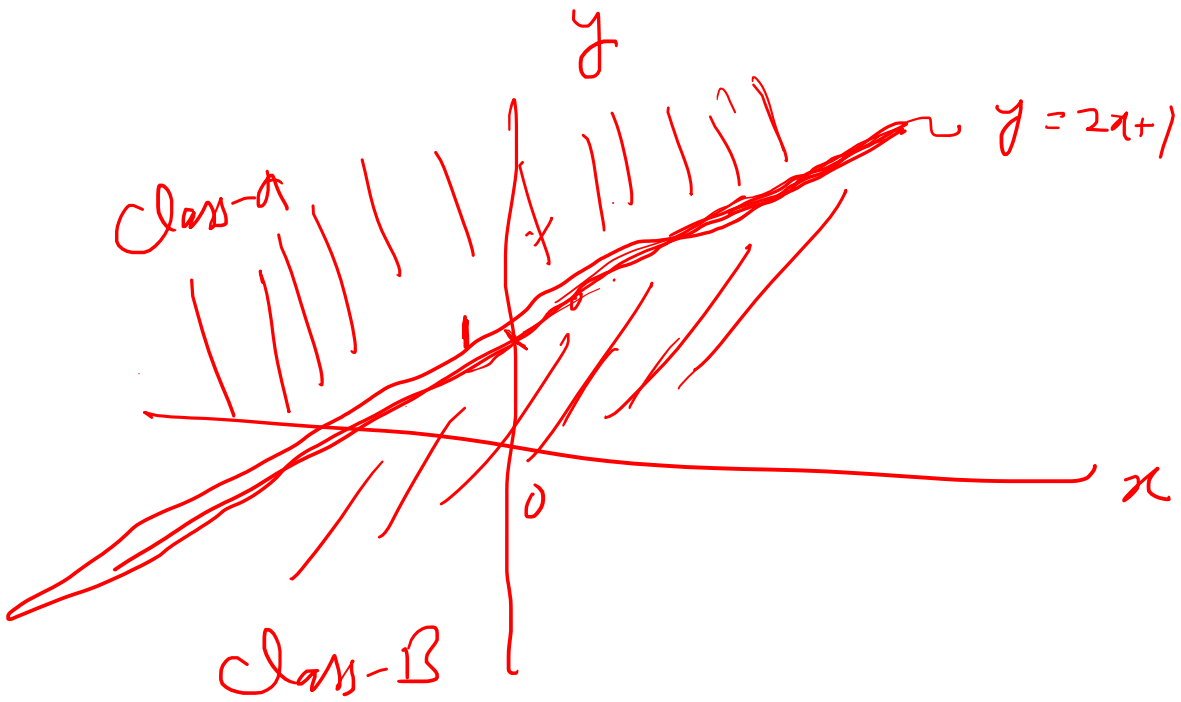
Example

$$\Rightarrow y = 2x + 1$$

$y > 0$

$$\boxed{2x + 1 > 0}$$

$y < 0 \Rightarrow 2x + 1 < 0$



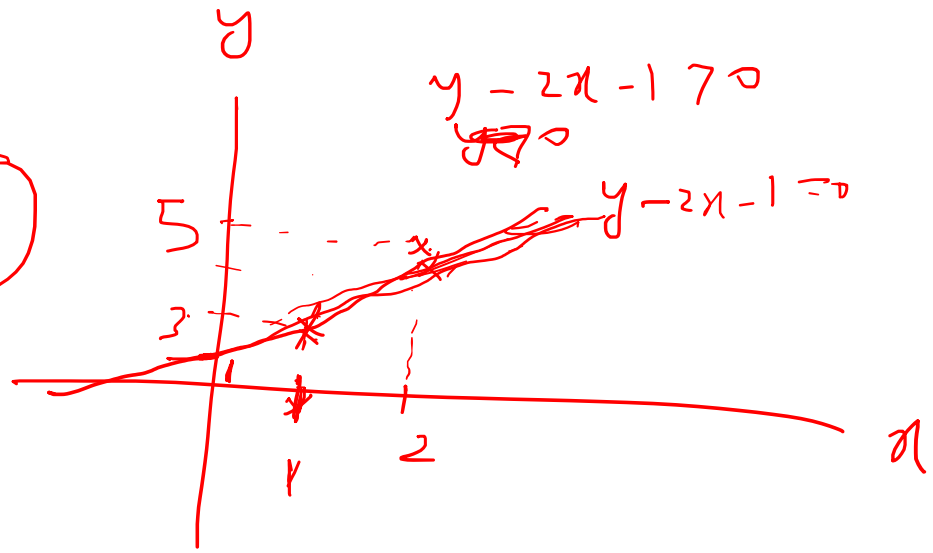
© 2021 Institute of Data

$y = 2x + 1$

$\boxed{y - 2x - 1 = 0}$

$x = 1, y = 3$

$3 - 2 - 1 = 0$



$x = 1, \; y = 4, \quad 4 - 2 - 1 = +ve > 0$

$x = 0.5, \; y = 0.5$

© 2021 Institute of Data

Logistic Regression

- Introduction to **classification**
- Logistic regression **algorithm**
- **Evaluating** classification results
- Measuring the **quality** of classification models
- **Dummy variables**

© 2021 Institute of Data

Classification

examples

- fraud detection
  - True / False
- customer segmentation:
  - frequent, high-value / regular, medium-value / occasional / transient
- credit risk
  - low / medium / high
- disease status
  - NYHA Class I / II / III / IV

© 2021 Institute of Data

Predicting Class Membership by Supervised Machine Learning

training data

- **features**
  - for now, assume these are continuous
- **response**
  - for now, assume this is binary (True/False)
  - Could be multiple classes in some case

goal

- We want to model the conditional probability $Pr(Y = 1|X = x)$ as a function of x
- i.e. predict ***p*(*y* = 1 | *X*)**
  - the probability of ***y*** being True given the predictor(s) ***X***

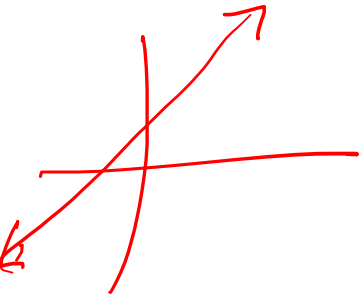
© 2021 Institute of Data

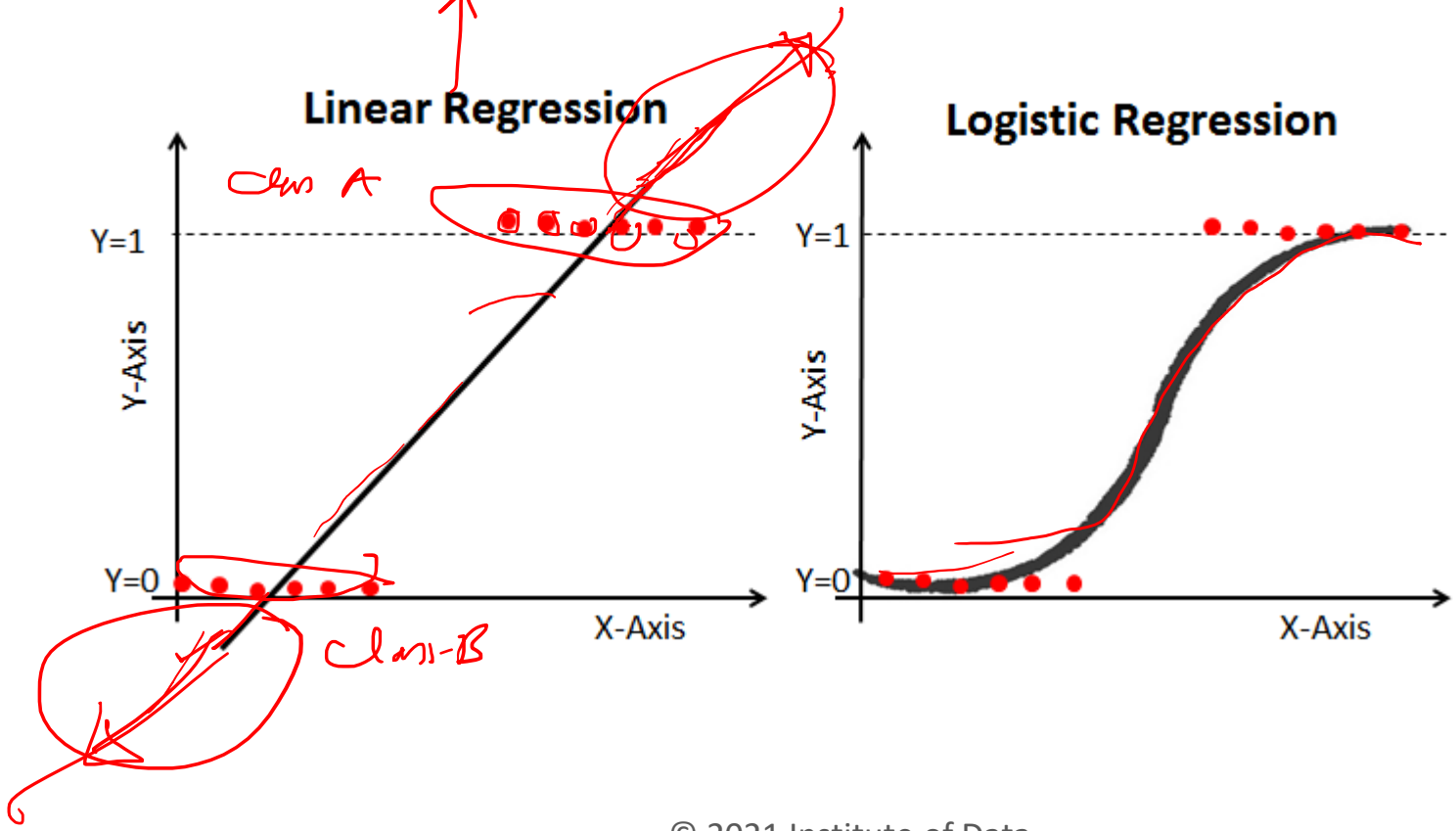
Linear vs Logistic Regression

- What is the difference?

$y = 3x + 1$

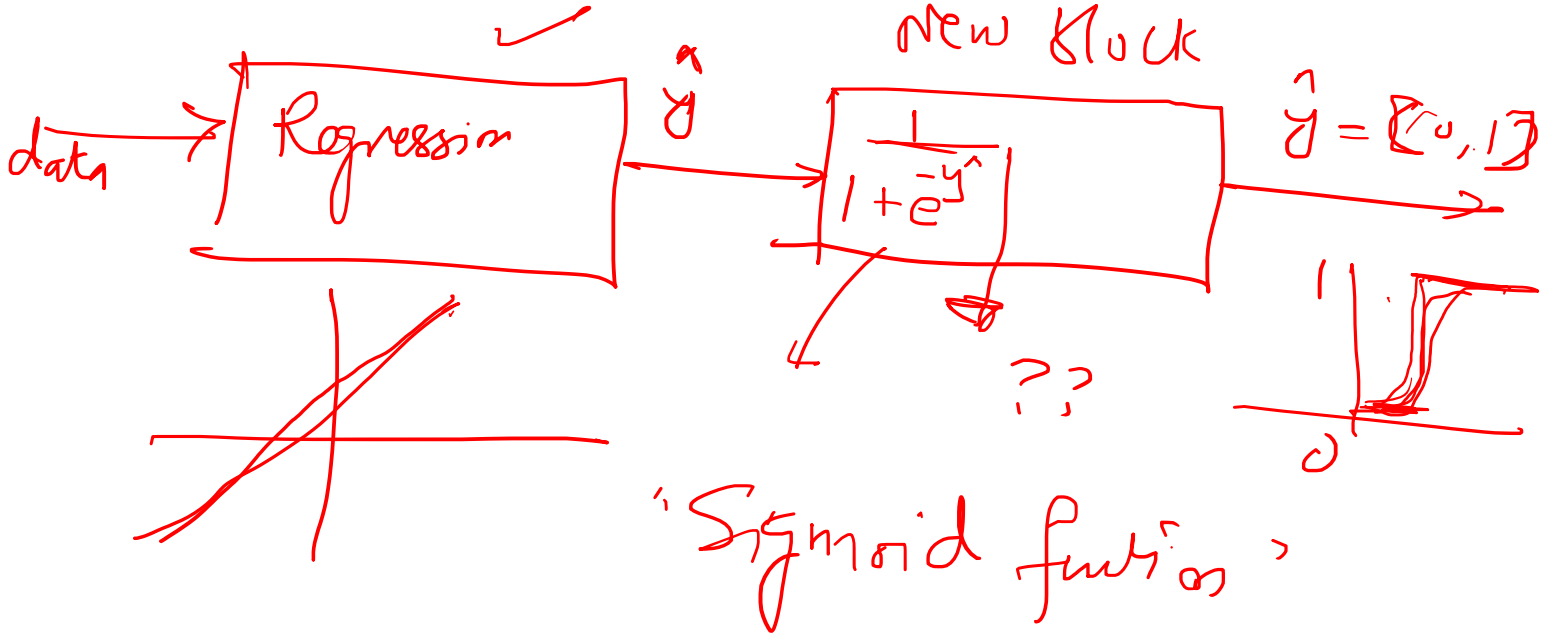
Cannot use it for classifier





© 2021 Institute of Data

Logistic Regression



"Sigmoid function"

© 2021 Institute of Data

Binary Class Prediction

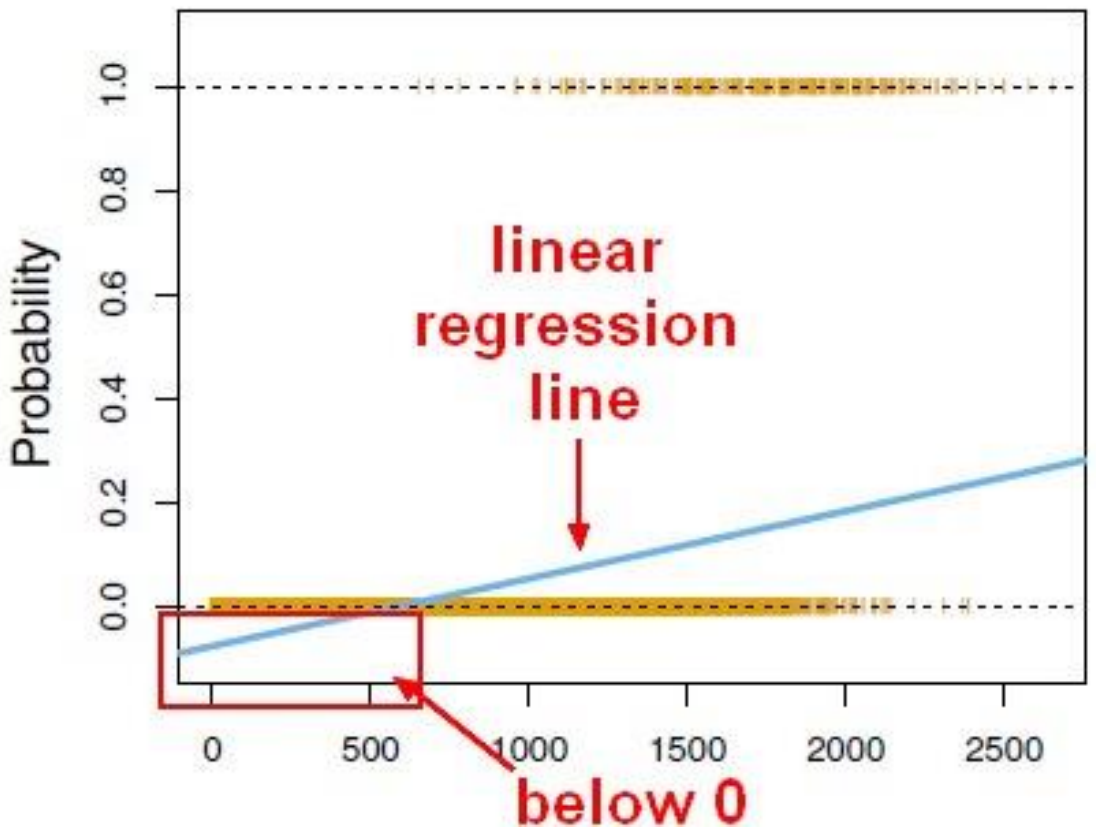
- **can we use linear regression?**

  i.e.t let $p(x)$ be a linear function of x

  $p = \beta_0 + \beta_1 \boldsymbol{X}$

$\boldsymbol{y} \in \{\text{False, True}\}$ ➡ $\boldsymbol{y} = \boldsymbol{\beta X} + \boldsymbol{\varepsilon}$

  - p must be between 0 and 1
  - But linear functions are unbounded.
  - predictions can be outside [0, 1]
  - So, Linear models can't do this.



© 2021 Institute of Data

Binary Class Prediction – cont'd

- The next obvious idea is to let log *p(x)* be a linear function of x

$$log(p) = \beta_0 + \beta_1 \boldsymbol{X}$$

> logarithms are unbounded in only one direction, and linear functions are not.

> need an approximating function that ensures $\boldsymbol{y} \in [0,1]$

© 2021 Institute of Data

Binary Class Prediction – cont'd

Easiest modification of log *p* which has an unbounded range is the logistic (or logit) transformation

$$\frac{p}{1-p} = \text{odds ratio}$$

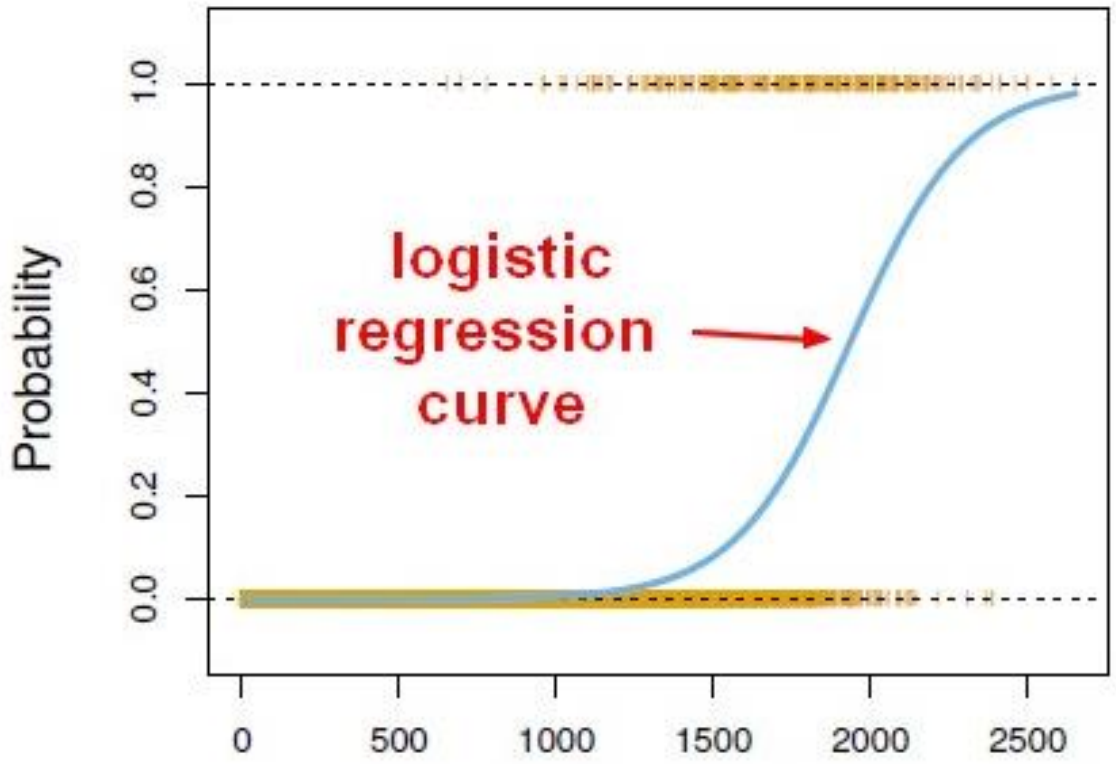
(p = probability of True)

$$\log\left(\frac{p}{1-p}\right) = \text{logit}$$ (*aka* log odds)

solve:

$$\log\left(\frac{p}{1-p}\right) = \beta_0 + \beta_1 \boldsymbol{X}$$

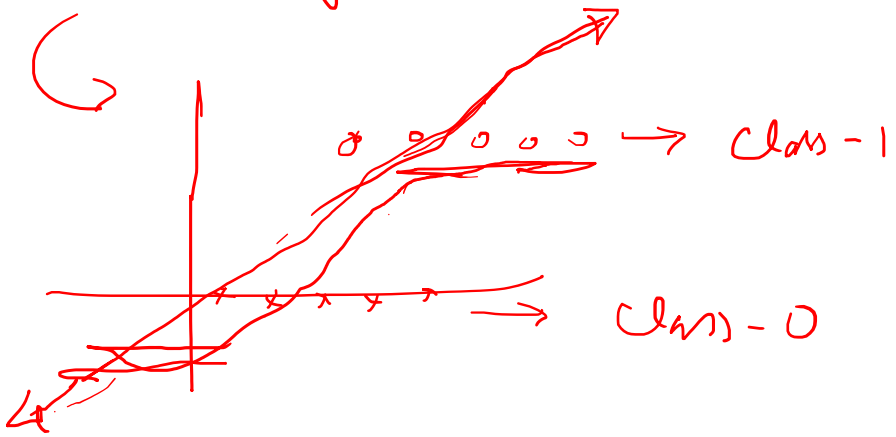
logit function



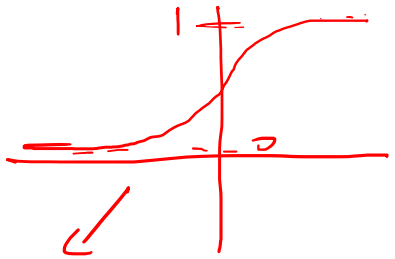
© 2021 Institute of Data

Real value → Regression Vs Classification ← "Category"

Q: How "Logistic Reg" is related to the "linear Regression"



① Logistic Regressor



Sigmoid function

[0, 1]

© 2021 Institute of Data

data x → Linear Reg → $\hat{y} = \beta_0 + \beta_1 x$ → $\frac{1}{1+e^{-\hat{y}}}$ → Prob

$[-\infty, \infty]$ $[0,1]$

Binary classification

- fraudulent
- Non-fraudulent

- Cat
- Dog

- ~~Survey~~ Survive
- Not survived

Email
- Spam
- Not Spam

$P[\text{Cat}/x] > \boxed{0.5} \rightarrow$ "Cat"

$P[\text{Spam}/x] > \boxed{0.5} \rightarrow$ "Spam"

"ROC"

© 2021 Institute of Data

Logistic Regression

- β_0, β_1 are known from regression results:

$$\log\left(\frac{p}{1-p}\right) = \beta_0 + \beta_1 \boldsymbol{X}$$

let $\boldsymbol{X}$ be a new data point for prediction,

calculate $\hat{y}$:

$$\boxed{\hat{y} = \beta_0 + \beta_1 \boldsymbol{X}}$$

then calculate p:

$$p = \frac{e^{\hat{y}}}{e^{\hat{y}} + 1} = \frac{1}{1 + e^{-\hat{y}}}$$

Conditions:

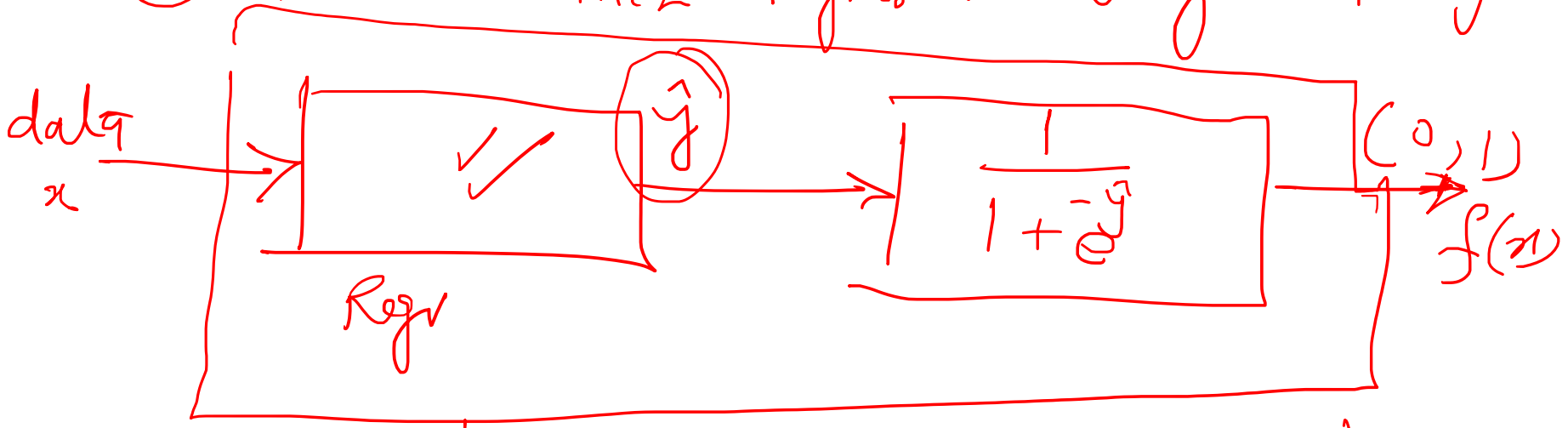
$p > 0$

$p < 1$

© 2021 Institute of Data

Logistic Regression → Classifier (Binary)

② It uses Linear Regression along with Sigmoid



Logistic Regression $f(x) = \frac{1}{1+e^{-\hat{y}}}$

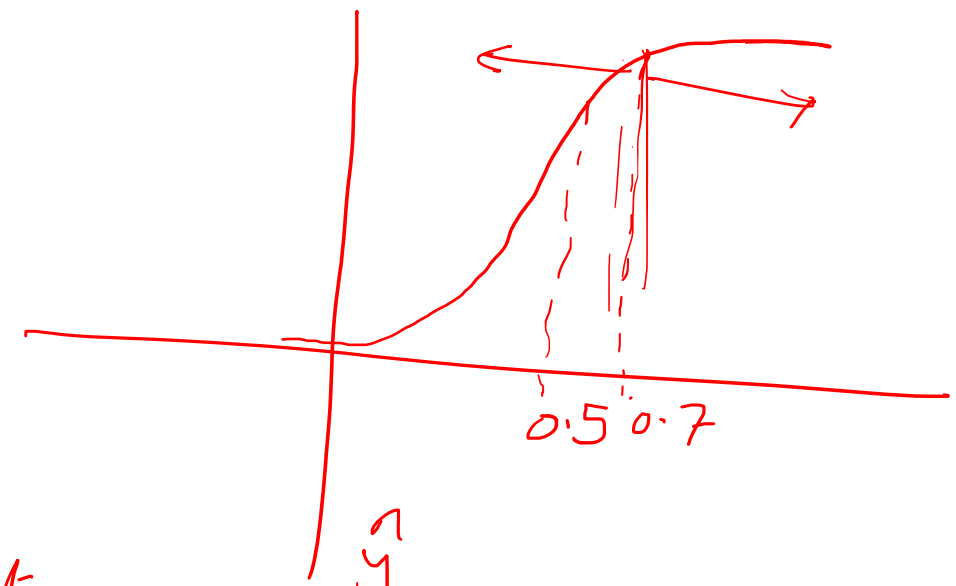
e = Base

$\hat{y} = w_1 x_1 + w_2 x_2 + \cdots + w_n x_n$

© 2021 Institute of Data

$$\frac{1}{1+e^{-x}} = \frac{1}{1+1} = \frac{1}{2}$$

$x = 0,1$



"ROC"

Point

$x = 1$

$$\frac{1}{1+e^{-4}} = \text{Real} \gtrless$$

$\hat{y} = 2x + 2$

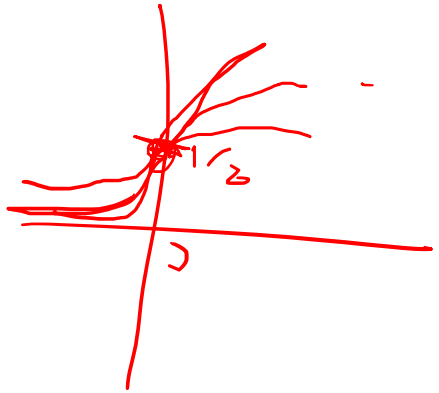
$= 4$

© 2021 Institute of Data

$$\frac{1}{1+e^{-2x}}$$

$$\frac{1}{1+e^{-x}}$$

$$\frac{1}{1+e^{-\alpha x}}$$



$x=0$ $\qquad \frac{1}{1+1} = 1/2$

© 2021 Institute of Data

$$\hat{y} = 2x + 1$$

$x = 1$,

$$\hat{y} = 2 + 1 = 3$$

$$\frac{1}{1+e^{-3}} \Rightarrow (0, 1)$$

$$\hat{y} = w_1 x_1 + w_2 x_2 + \cdots + w_n x_n$$

$$\hat{y}_i = \sum_i w_i x_i$$

$$f(x_i) = \frac{1}{1+e^{-\hat{y}}} = \frac{1}{1+e^{-\sum w_i x_i}}$$

$$\frac{1}{1+e^{-x}} \longrightarrow (0, 1)$$

(Probability) $[0, 1]$

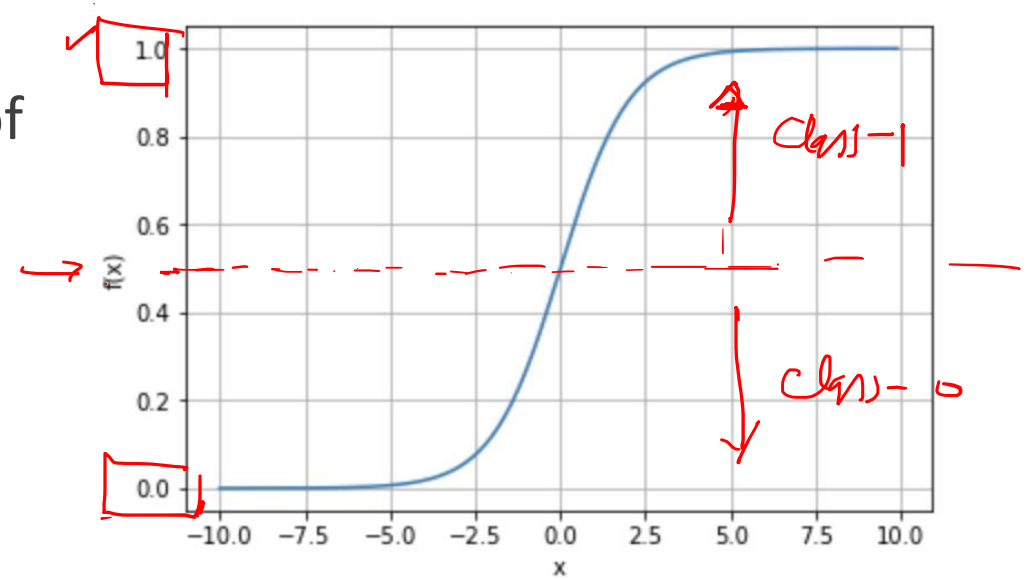
© 2021 Institute of Data

Logistic Regression

Appropriate when predicting a **binary categorical** outcome variable from a set of predictor variables (features) that may be **continuous and/or categorical**

Features should be independent with no missing data.

Logistic Regression typically requires **a relatively large sample size.** A general guideline is that you need at minimum of 10 cases with the least frequent outcome for each independent variable in your model.



Sigmoid or logistic curve

$$y = \sigma(x) = \frac{1}{1+e^{-x}}$$

estimation of Linear Reg

$$P(y=1) = \sigma(w \cdot x + b) = \frac{1}{1+\exp(-(w \cdot x + b))}$$

© 2021 Institute of Data

Q: (i) Given → "The classifier is already Trained"

(Logistic Regression)

* "weights are available".

(ii) Whatever Processing we do with the data, during the Training → The same has to be done during the Test !!

© 2021 Institute of Data

Example

Logistic

y < +ve Review
  -ve Review

"NLP"

- Suppose we are doing binary sentiment classification on movie review text, and we would like to know whether to assign the sentiment class + or - to a review document *doc*

y = +1

Given text

+ve
-ve Review

- Let's assume that we've 6 features
- we already learned a real-valued weight for each of these features,
- The 6 weights(already learnt) corresponding to the 6 features are:

"Infrace"

weights

[2.5, -5.0, -1.2, 0.5, 2.0, 0.7], while b = 0.1

Constant

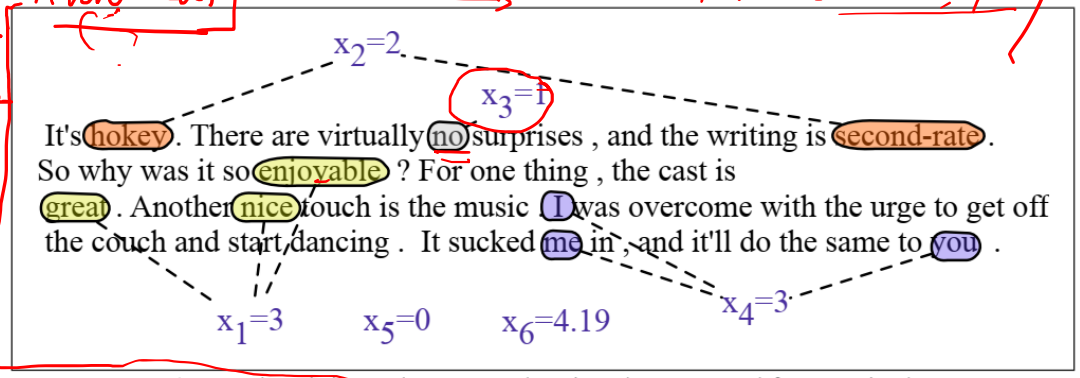
Data ⟹ Feature → [ ] → Traing classifier

Logistic Regression

Find P(Y=1|x) ?

$\hat{y} = w_0 + w_1 x_1 + \cdots + w_6 x_6$

$\hat{y} = 0.1 + 2.5 \times x_1 + 5 \times x_2 + \cdots + 0.7 \times x_6$



A sample mini test document showing the extracted features in the vector x.

Var	Definition	Value
x_1	count(positive lexicon) $\in$ doc)	3
x_2	count(negative lexicon) $\in$ doc)	2
x_3	$\begin{cases} 1 & \text{if "no"} \in \text{doc} \\ 0 & \text{otherwise} \end{cases}$	1
x_4	count(1st and 2nd pronouns $\in$ doc)	3
x_5	$\begin{cases} 1 & \text{if "!"} \in \text{doc} \\ 0 & \text{otherwise} \end{cases}$	0
x_6	log(word count of doc)	ln(66) = 4.19

Numeric

$\hat{y} = 0.1 + 2.5 \times 3 + \cdots + 0.7 \times 4.19$

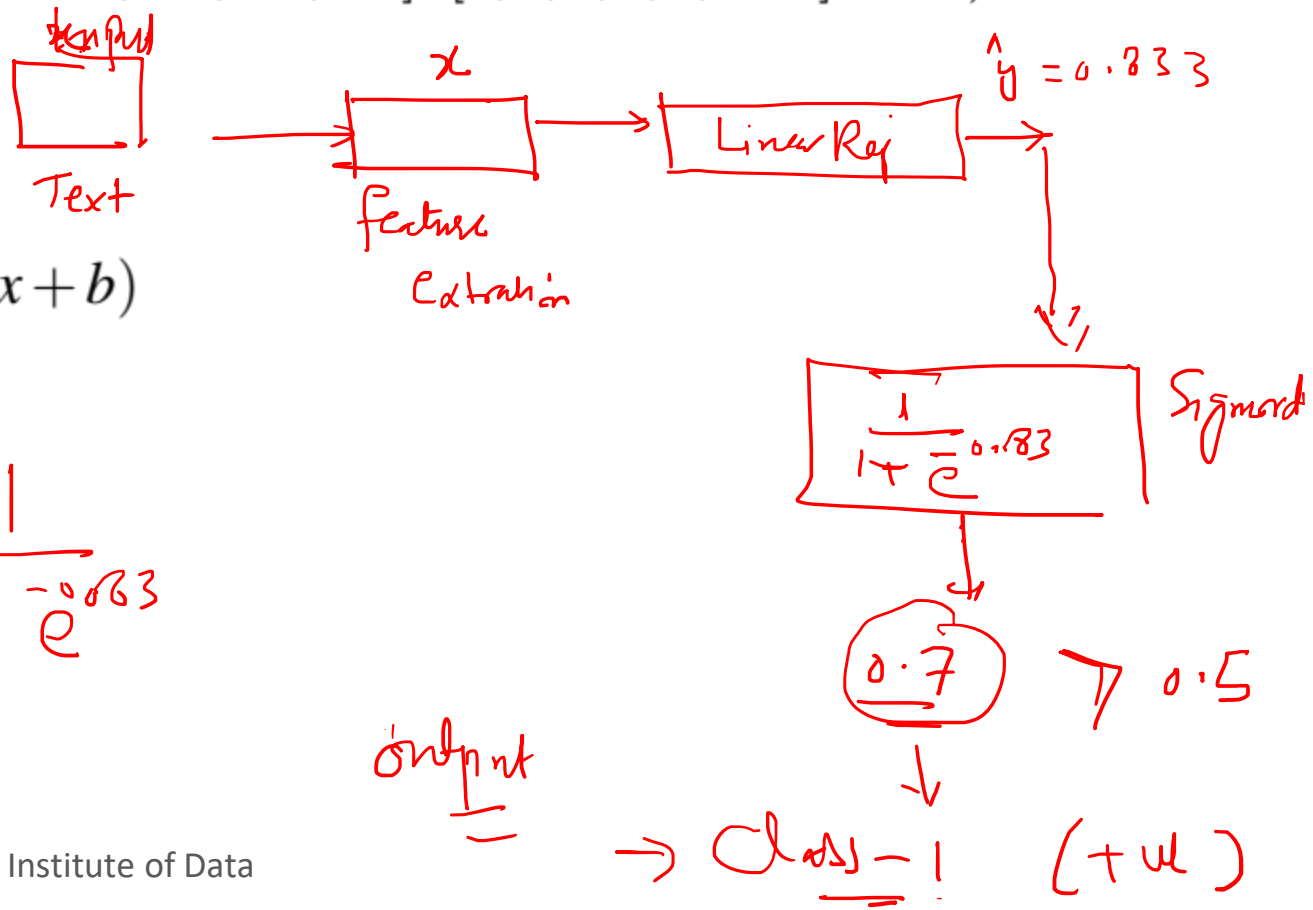
© 2021 Institute of Data

Solution

$$
\begin{aligned}
p(+|x) = P(Y=1|x) &= \sigma(w \cdot x + b) \\
&= \sigma([2.5, -5.0, -1.2, 0.5, 2.0, 0.7] \cdot [3, 2, 1, 3, 0, 4.19] + 0.1) \\
&= \sigma(.833) \\
&= 0.70
\end{aligned}
$$

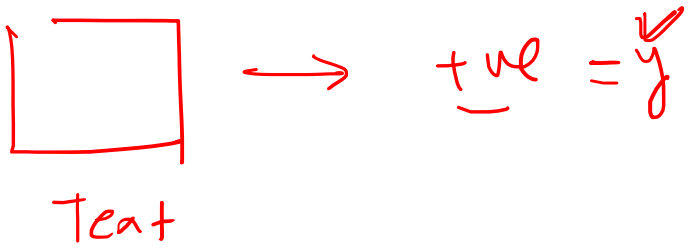
$$
\begin{aligned}
p(-|x) = P(Y=0|x) &= 1 - \sigma(w \cdot x + b) \\
&= 0.30
\end{aligned}
$$

$$\sigma(0.833) = \frac{1}{1+e^{-0.833}}$$



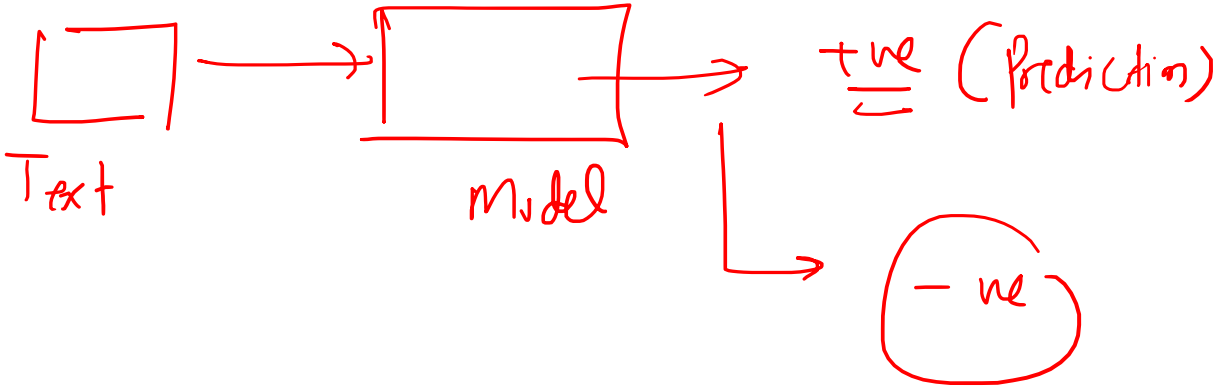
© 2021 Institute of Data

Ex :



TP = 1

Score



© 2021 Institute of Data

Logistic Regression – Scikit-learn

from **sklearn.linear_model** import LogisticRegression

- Can deal with any number of features
- **Features must be numeric**
  - Categorical features should be converted to dummy features
- Can perform **multi-class classification**

- Logistic Reg
- Perceptron
- SVM

Binary

"One Vs Rest"

1 ← Red, Green, Blue → 0

1 ← Red Vs other → 0

1 ← Green Vs other → 0

1 ← Blue Vs other → 0

Demo on Logistic Regression → lab

© 2021 Institute of Data

Evaluating Classification Results

R^2

There are a number of metrics that can be used to evaluate a classification model. Many of these metrics revolve around values drawn from the Confusion Matrix.

Regression

SSR

The **Confusion Matrix** is a table that contains counts of the predictions of the model versus the actuals.

(Model) Prediction

Actual (Label)

	Positive	Negative
Positive	TP	FP
Negative	FN	TN

Note that if we are interested in predicting the opposite class, the entries would be reversed (the positive becomes negative and vice versa).

+ve

"Base"

© 2021 Institute of Data

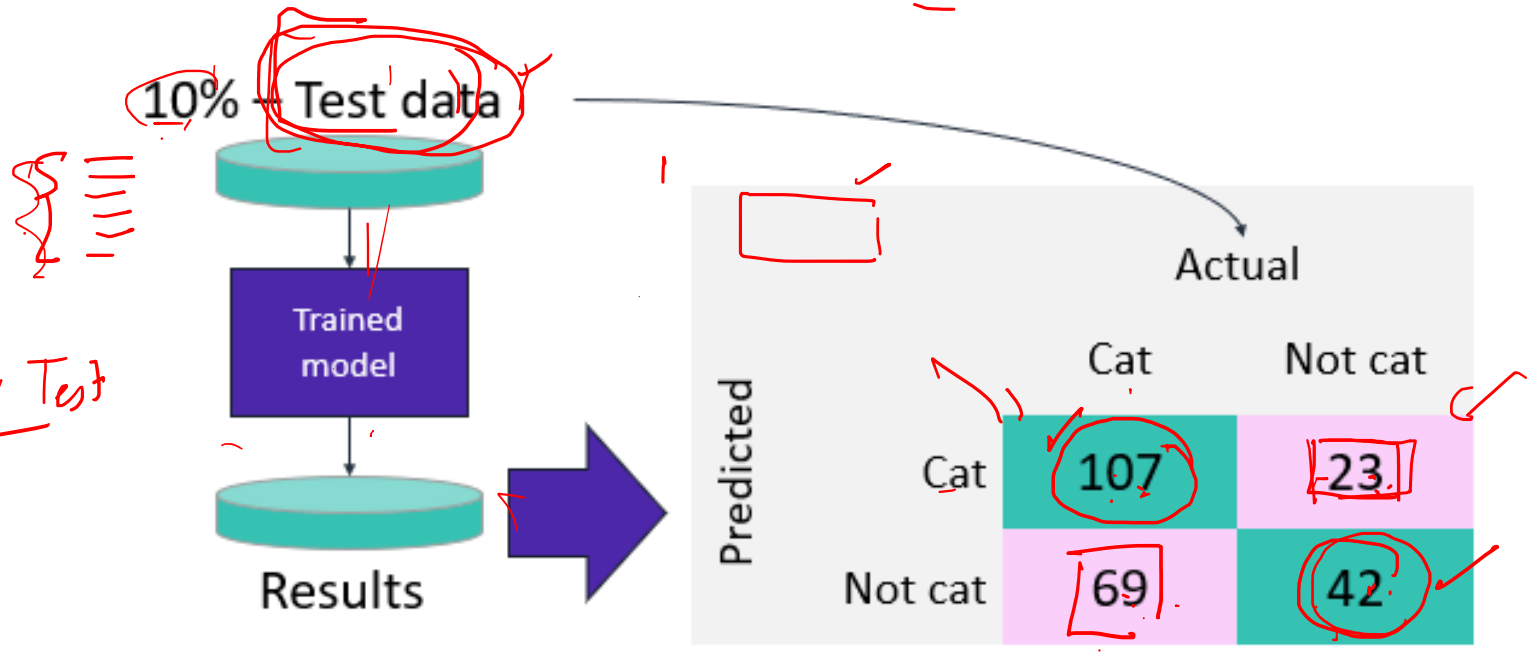
Confusion Matrix

10%

"Positive"

- Image Recognition Model (Cat or Dog i.e. Not Cat)

Q: How Test data much size of

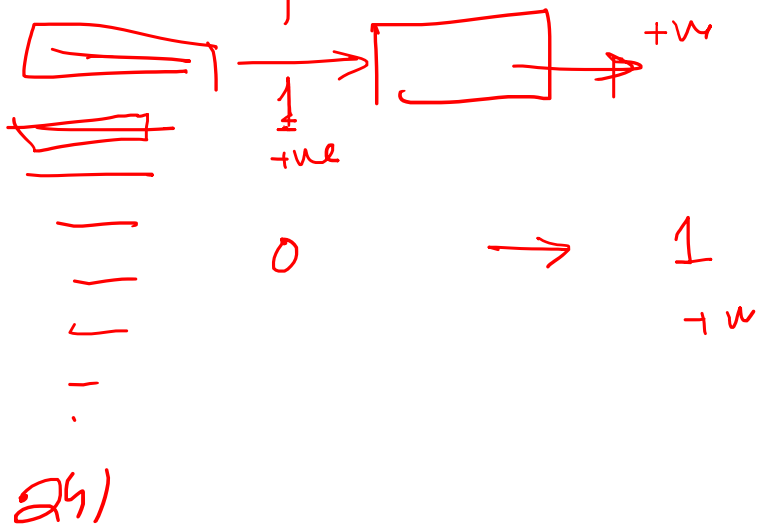


0.9x Test

Actual	Positive	Negative
Prediction		
Positive	True Positive (TP)	False Positive (FP)
Negative	False Negative (FN)	True Negative (TN)

© 2021 Institute of Data

Test Sample



	Actual	Prediction	
+ve	1	1 +ve	TP
−ve	0	1	FP
	1	0	FN
	0	0	TN

Actual

Prediction

TP	FP

© 2021 Institute of Data

Confusion Matrix

- Terminology : Classification

		Actual	
		Cat	Not cat
Predicted	Cat	TP	FP
	Not cat	FN	TN

True positive (TP)
Predicted a cat and it was a cat

False positive (FP)
Predicted ~~not~~ a cat and it was a cat

False negative (FN)
Predicted a cat and it was ~~not~~ a cat

True negative (TN)
Predicted *not* a cat and it was *not* a cat

https://en.wikipedia.org/wiki/Confusion_matrix

© 2021 Institute of Data

Knowledge Check

Goal: Spam Email

- A group of persons perform a Covid test. Explain the confusion matrix for it?

Goal: Fraudulent Transact.

+ve
-ve
features → Covid: +ve, -ve

TP → +ve, +ve

	Actual: +ve	Actual: -ve
Prediction: +ve	TP	FP
Prediction: -ve	FN	TN

Model = Covid = +ve
Actual = Covid = -ve

TN → -ve, -ve

Originally → Covid: +ve
Test → Covid: -ve

© 2021 Institute of Data

Evaluating Classification Results – cont'd

TP	FP
FN	TN

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

8	3
2	2

- **Accuracy**: the fraction of predictions that the model got right. i.e. Number of correct prediction / total prediction
  - Accuracy = Total true prediction / Total sample count
- **Precision**: how many of the prediction were correct. It is a measure of **exactness**. → (Correctness of Model wrt +ve decision)
  - Precision = TP / Total positive prediction (TP + FP)
- **Recall**: how many of the actual positive did the model predict. It is a measure of **completeness**.
  - Recall = TP / Total actual positive (TP + FN)

$$Acc = \frac{10}{10+5} = \left(\frac{10}{15}\right)$$

Actual	Positive	Negative	
Prediction			
Positive	True Positive (TP)	False Positive (FP)	Total predicted positive
Negative	False Negative (FN)	True Negative (TN)	Total predicted negative
	Total actual positive	Total actual negative	Total Sample count

Total true prediction

https://developers.google.com/machine-learning/crash-course/classification/accuracy

© 2021 Institute of Data

Imbalanced

Accuracy $:= \dfrac{TP + TN}{(TP + TN + FN + FP)}$

Total Data

10%

Test data

"99%" → NOT a good Model:

Test data (100 Sample)

	TP	FP
	1	1
	3	90
	FN	TN

+ve = 9

− ve = 91

give → [ ] → Negative

X

or 0 0 0 0 0 0 0 0 0 0

Blue = 9

White = 91

$\frac{91}{100} = 91\%$

© 2021 Institute of Data

- Confusion Matrix
- Accuracy
- Precision : $\frac{TP}{TP+FP}$
- Recall : $\frac{TP}{TP+FN}$

		Actual	
		+ve	-ve
Predicted	+ve	TP	FP
	-ve	FN	TN

Recall = 1, $\boxed{FN = 0}$

© 2021 Institute of Data

Sensitivity (Recall or true positive rate)

TPR

Recall

		Actual	
		Cat	Not cat
Predicted	Cat	TP:107	FP:23
	Not cat	FN:69	TN:42

Sensitivity:

What percentage of cats were correctly identified?

$$\text{sensitivity} = \frac{\text{TP}}{\text{TP} + \text{FN}}$$

$$\text{sensitivity} = \frac{107}{107+69} = 0.6079$$

60% of cats were identified as cats.

$$\text{TPR} = \frac{\text{TP}}{\text{TP}+\text{FN}}$$

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

© 2021 Institute of Data

Specificity

		Actual	
		Cat	Not cat
Predicted	Cat	TP:107	FP:23
	Not cat	FN:69	TN:42

Specificity:

What percentage of ***not*** cats were correctly identified?

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

$$\text{specificity} = \frac{42}{42+23} = 0.6461$$

64% of ***not*** cats were identified as ***not*** cats

© 2021 Institute of Data

Knowledge Check

Question:

- Which model is better?

Model A

Predicted	Actual: Cat	Actual: Not cat
Cat	107	23
Not cat	69	42

FP

Sensitivity	60%
Specificity	64%

Model B

Predicted	Actual: Cat	Actual: Not cat
Cat	148	53
Not cat	28	12

Sensitivity	84%
Specificity	18%

if you want to identify as many cats as possible? Model B is a good answer, if you are not concerned about many false positives—that is, not cats that were incorrectly identified.

if you want to make sure that you identify animals that are not cats? Model A might work, depending upon how many false negatives you can handle.

https://www.analyticsvidhya.com/blog/2020/10/how-to-choose-evaluation-metrics-for-classification-model/

© 2021 Institute of Data

F1 Score

Harmonic mean of the model's precision and recall

$$\text{F1 score} = 2 \cdot \frac{\text{precision} \cdot \text{sensitivity}}{\text{precision} + \text{sensitivity}}$$

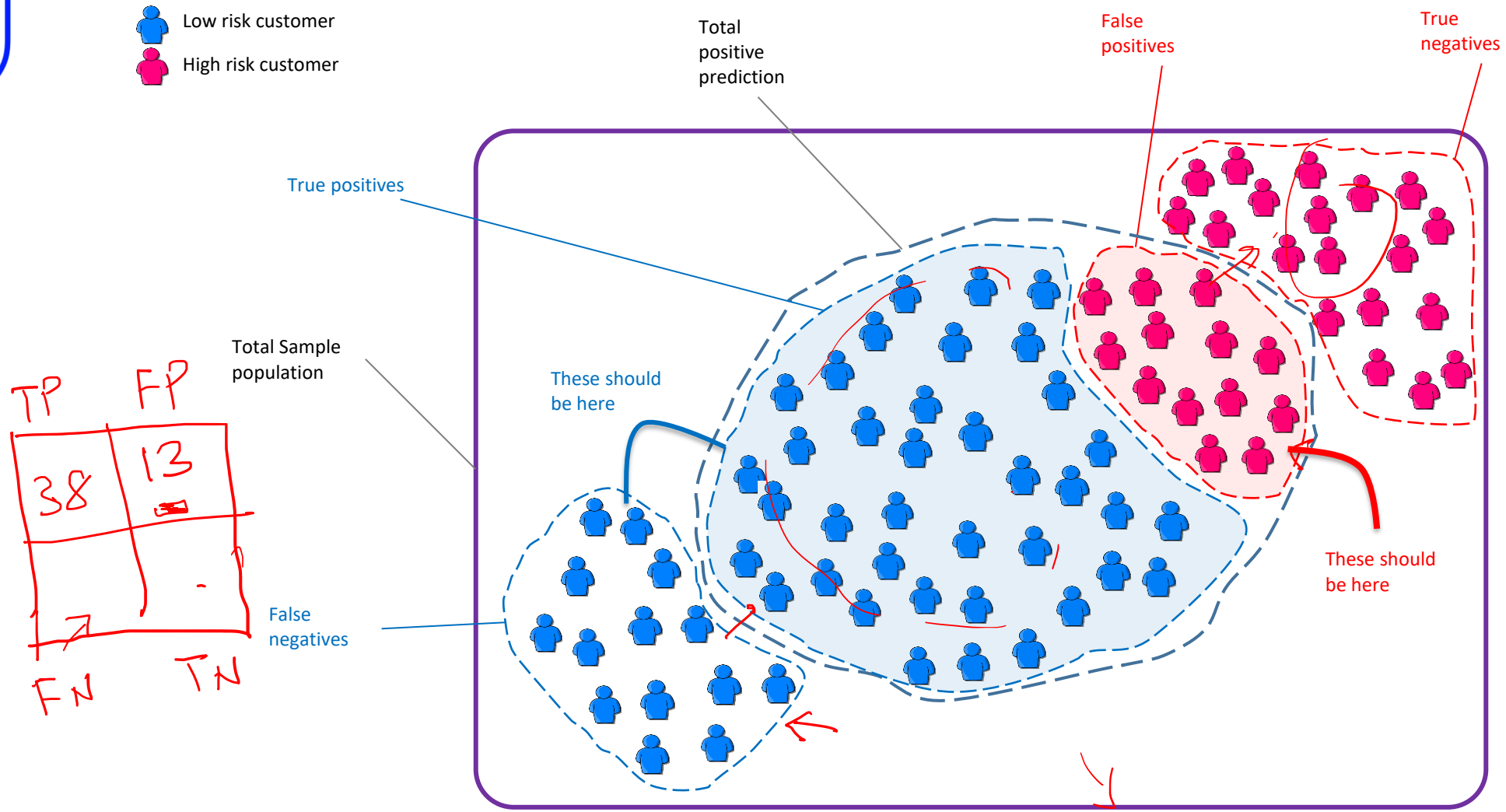
→ Recall

Combines precision and sensitivity

F1 → 1

F1 score is a better measure to use if you are seeking a balance between Precision and Recall.

© 2021 Institute of Data



Actual	Positive	Negative	
Prediction			
Positive	True Positive (TP)	False Positive (FP)	Total predicted positive
Negative	False Negative (FN)	True Negative (TN)	Total predicted negative
	Total actual positive	Total actual negative	Total Sample count

Total true prediction

Note: icons are positioned where they are for illustration purpose only.

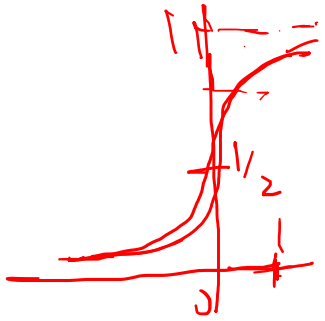
Accuracy = Total true prediction / Total sample count
Precision = TP / Total positive prediction (TP + FP)
Recall = TP / Total actual positive (TP + FN)

© 2021 Institute of Data

"Threshold"

- ROC →

Models return probabilities:



Predicted
0.96 → Cat
0.77 → Cat
0.58 → Cat
0.01 → Dog
0.47 → Dog

0.5

if [predicted] >= threshold — It's a cat!

if [predicted] < threshold — NOT cat!

$> 0.4 \rightarrow$ Cat

$Y = 0, 1$

Changing the threshold can change the prediction.

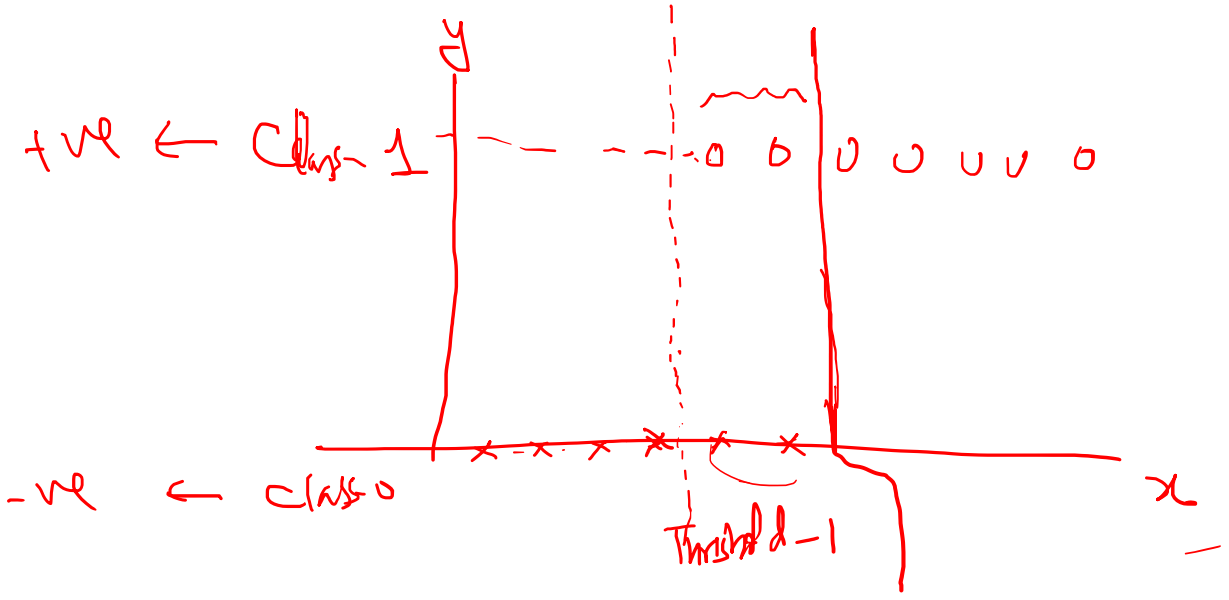
$x \rightarrow \boxed{} \rightarrow P[Y=1/x]$

$P[Y=0/x] = 1 - P[Y=1/x]$

© 2021 Institute of Data

ROC

Ex:



	TP	FP
	7	2
	0	4
	FN	TN

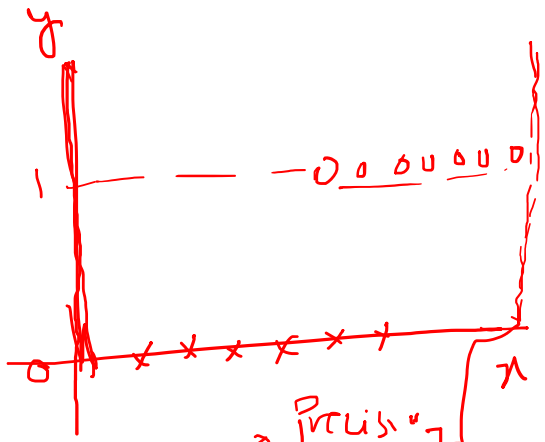
TPR $= \frac{7}{7} = 1$

FPR $= \frac{2}{2+4} = 1/3$

$(1, 1/3)$

(TPR, FPR)

5	0
2	6



Precision $= \frac{7}{7+6}$

TP	
7	6
0	0

0	0
7	6

© 2021 Institute of Data

Threshold

Ex:

Pred	Act	
1	1	TP
0	0	TN
1	0	
0	1	

- 10 data points with three different thresholds

ID	Actual	Prediction Probability	>0.6	>0.7	> 0.8	Metric
1	0	0.98	1	1	1	
2	1	0.67	1	0	0	
3	1	0.58	0	0	0	
4	0	0.78	1	1	0	
5	1	0.85	1	1	1	
6	0	0.86	1	1	1	
7	0	0.79	1	1	0	
8	0	0.89	1	1	1	
9	1	0.82	1	1	1	
10	0	0.86	1	1	1	

$$TPR = \frac{True\ Positive}{True\ Positive + False\ Negative}$$

$$FPR = \frac{False\ Positive}{False\ Positive + True\ Negative}$$

Threshold > 0.6

TP	FP
3	6
1	0
FN	TN

$TPR = \text{Recall} = \frac{3}{3+1} = 3/4$

© 2021 Institute of Data

Threshold

- 10 data points with three different thresholds

ID	Actual	Prediction Probability	>0.6	>0.7	> 0.8	Metric
1	0	0.98	1	1	1	
2	1	0.67	1	0	0	
3	1	0.58	0	0	0	
4	0	0.78	1	1	0	
5	1	0.85	1	1	1	
6	0	0.86	1	1	1	
7	0	0.79	1	1	0	
8	0	0.89	1	1	1	
9	1	0.82	1	1	1	
10	0	0.86	1	1	1	
			0.75	0.5	0.5	TPR
			1	1	0.66	FPR
			0	0	0.33	TNR
			0.25	0.5	0.5	FNR

$$TPR = \frac{True\ Positive}{True\ Positive + False\ Negative}$$

$$FPR = \frac{False\ Positive}{False\ Positive + True\ Negative}$$

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

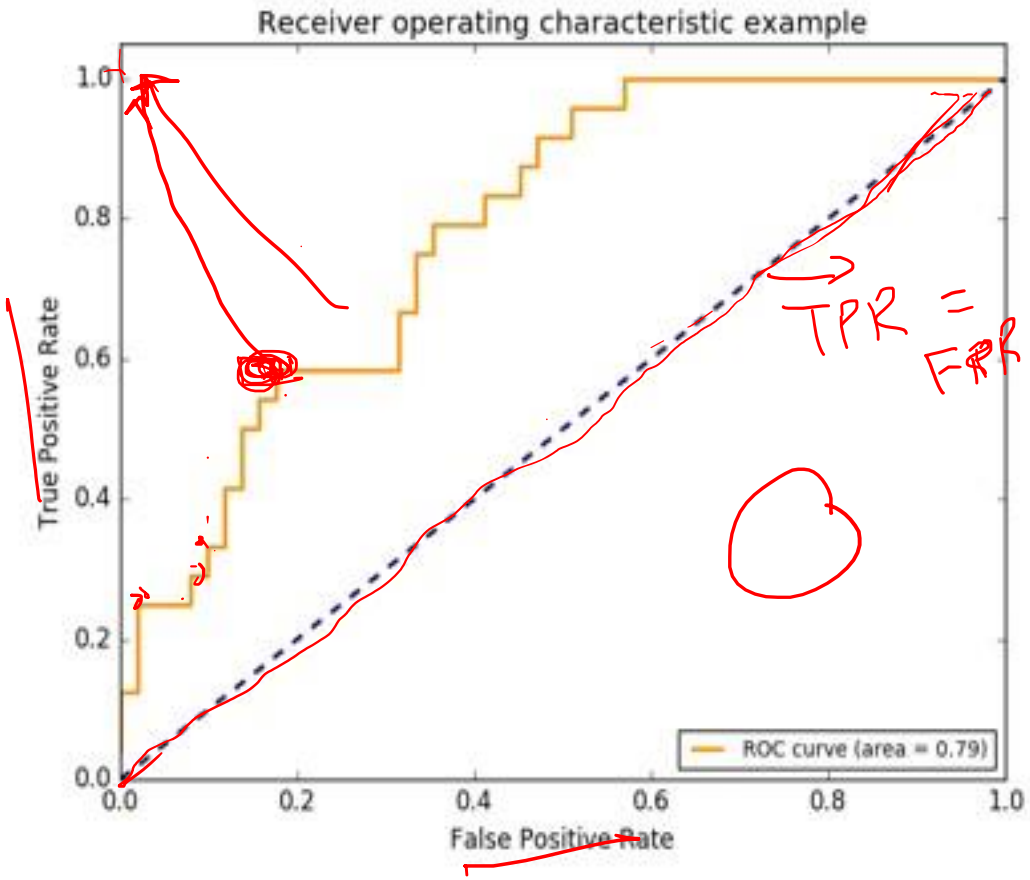
© 2021 Institute of Data

Evaluating Classification Results – cont'd

Sklear

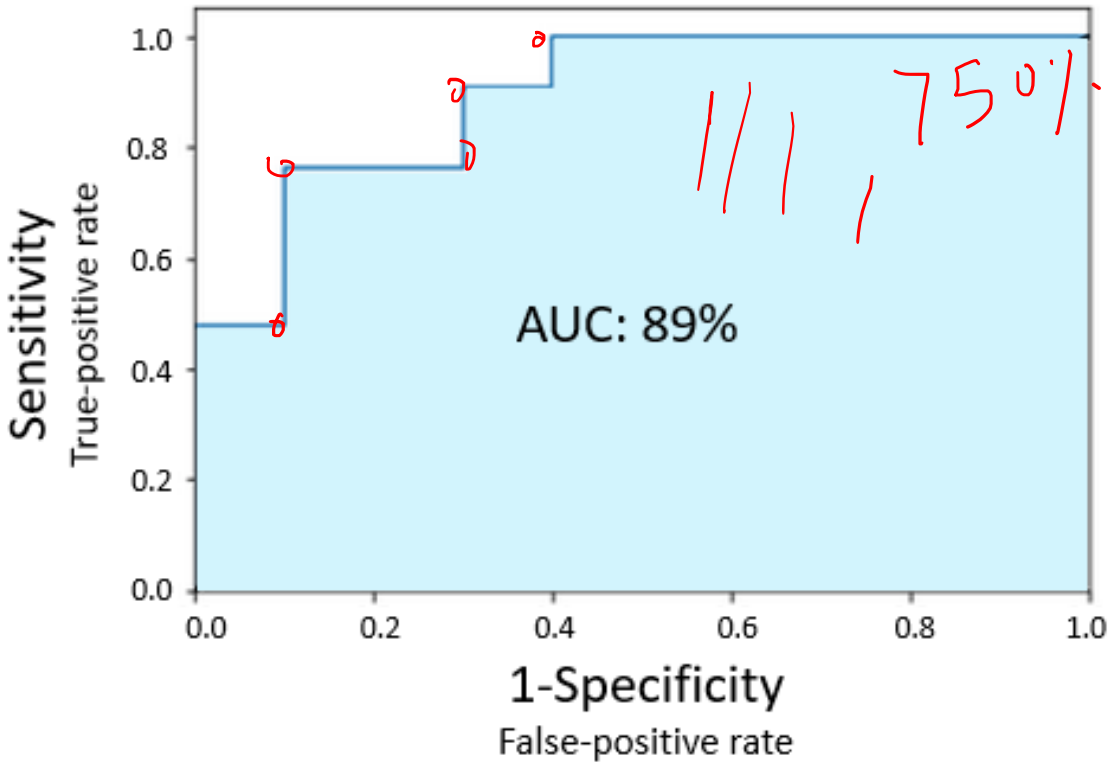
Receiver operating characteristics (ROC) curve

- Compares True Positive Rate and False Positive Rate
- **True Positive Rate (TPR)** = Recall
- **False Positive Rate (FPR)** is FP/ Total negative count (FP+ TN)
- ROC plots **TPR** vs **FPR** by varying threshold over the entire range of threshold settings. It depicts relative **trade-offs between true positive (benefits) and false positive (costs)**
- **Area Under Curve (AUC)** is equal to the probability that the model will rank a randomly chosen positive instance higher than a randomly chosen negative one.



© 2021 Institute of Data

AUC-ROC



Area under the curve

AUC-ROC is used to easily compare one model to another.

© 2021 Institute of Data

Evaluating Classification Results – cont'd

- There are many other metrics and many other names for the same metrics
- It is better to stick with a small number of metrics that make sense in your domain before using other metrics
- **Accuracy, precision, recall** and **AUC** are the most common metrics

threshold

Confusion Matrix

© 2021 Institute of Data

Dummy Variables

How can we use categorical variables in an algorithm that requires numerical predictors?

- ***ordinal categoricals***
  - can be converted to a sequence of integers, if it makes sense to do so

cold	cool	moderate	warm	hot
1	2	3	4	5

✓

  - the above implies hot < warm < moderate < cool < cold
    ... which makes sense

© 2021 Institute of Data

Dummy Variables

How can we use categorical variables in an algorithm that requires numerical predictors?

- ***cardinal categoricals***

apples	bananas	peaches	oranges	pears
1	2	3	4	5

> *this implies pears > oranges > peaches > bananas > apples ... does not make sense!*

> must convert to dummy variables instead

© 2021 Institute of Data

Cardinal Dummy Variables

- full definition (number of variables = number of categories):
  - fruit_apples: 1 = apples, 0 = no apples
  - fruit_bananas: 1 = bananas, 0 = no bananas
  - fruit_peaches: 1 = peaches, 0 = no peaches
  - fruit_oranges; 1 = oranges, 0 = no oranges
  - fruit_pears; 1 = pears, 0 = no pears
- compact definition (number of variables is one less than number of categories):
  - fruit_bananas: 1 = bananas, 0 = apples
  - fruit_peaches: 1 = peaches, 0 = no peaches
  - fruit_oranges; 1 = oranges, 0 = no oranges
  - fruit_pears; 1 = pears, 0 = no pears

Python:

pandas.get_dummies

© 2021 Institute of Data

Lab 5.1: Logistic Regression

- Purpose:
  - To predict survival amongst Titanic passengers using the LogisticRegression() method of Scikit-Learn

- Materials:
  - ‘Lab 5.1.ipynb’

© 2021 Institute of Data

Discussion

- Strengths & weaknesses of logistic regression

© 2021 Institute of Data

Institute of Data

Perceptron (Neural Networks)

Supervised Learning

Regression

Classification

① Logistic Regression

② "Perceptron"

CNN

③ SVM

- **Biological** and **artificial** neurons
- **Activation** functions
- **Linear regression** with a perceptron
- **Linear classification** with a perceptron
- **Back propagation**
- **Gradient descent**
- Practical implementations

→ Deep NN

© 2021 Institute of Data

How does a nerve cell make a decision?

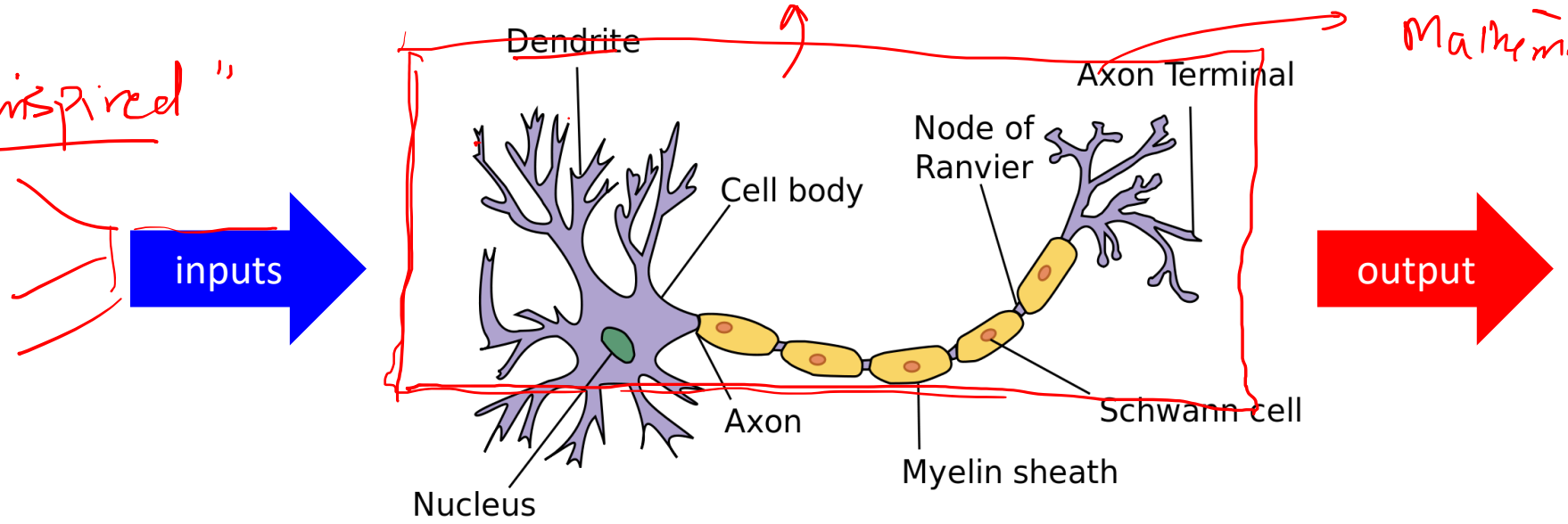
- Neuron receives inputs at antennae-like structures (‘**dendrites**’)
- Each incoming connection is **dynamically strengthened or weakened** by:
  - frequency of use (‘weighting’)
  - neurotransmitters

"Non-Linear"

- **Weighted inputs** are summed in the cell body (transformed into a new signal)
- New signal is **propagated** along the cell’s axon to be detected by other neurons

"Nature-inspired"

Mathematical Relation



© 2021 Institute of Data

The Perceptron

- A **basic imitation** of the natural neuron
- Inputs are given **weights**
- **Weighted** signals are summed
- summed signal is transformed by an **activation function** to produce an output

$y = z = w_0 + w_1 x_1 + \cdots + w_n x_n$

$= \sum_{i=0} w_i x_i$

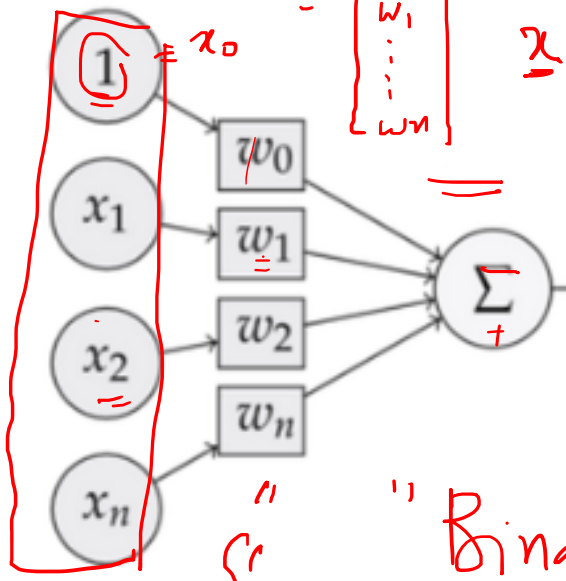
$z = w_0^T x \rightarrow$ Single value

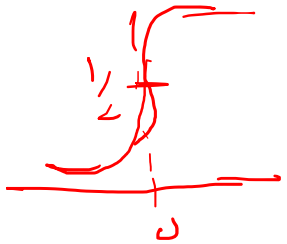
$= w^T x$

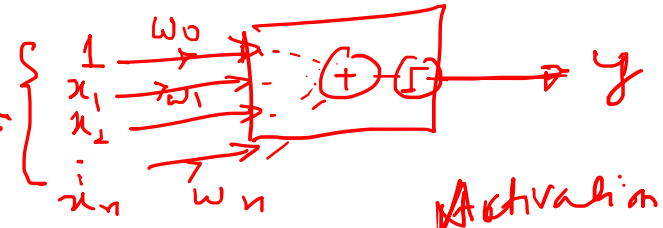
$y = \Phi(z)$

$y = \Phi(w^T x)$

$x = \begin{bmatrix} 1 \\ x_1 \\ \vdots \\ x_n \end{bmatrix}$ $w = \begin{bmatrix} w_0 \\ w_1 \\ \vdots \\ w_n \end{bmatrix}$



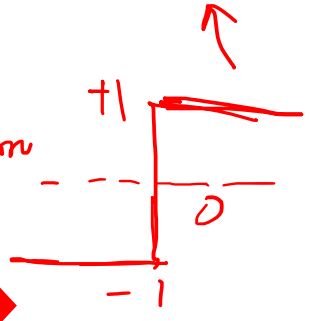




Activation function

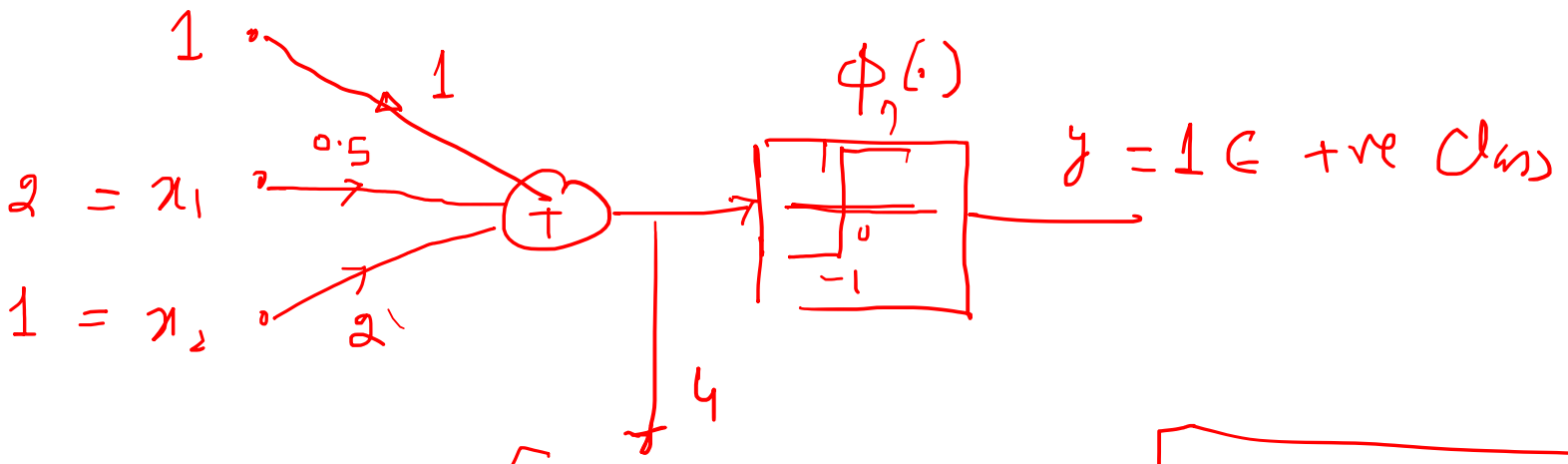
"Non-Linear Unit"

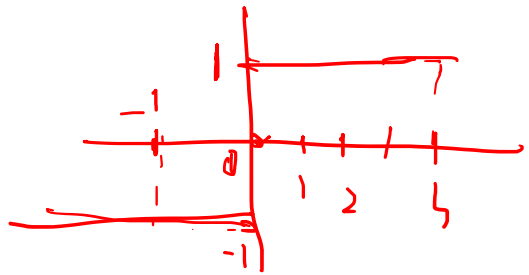
"Signum"



"Binary Classifier"

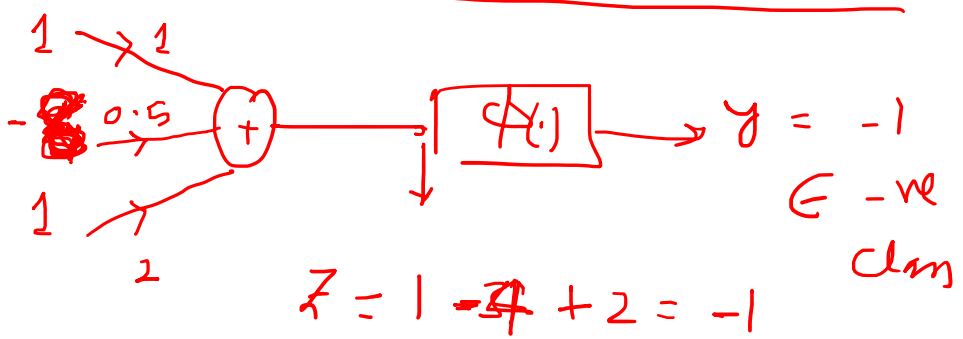
© 2021 Institute of Data





$$z = 1\times1 + 2\times0.5 + 1\times2$$
$$= 1+1+2 = 4$$

$$z = \underline{w}^T \underline{x}$$
$$= \begin{bmatrix} 1 & 0.5 & 2 \end{bmatrix} \begin{bmatrix} 1 \\ 2 \\ 1 \end{bmatrix} = 4$$



© 2021 Institute of Data

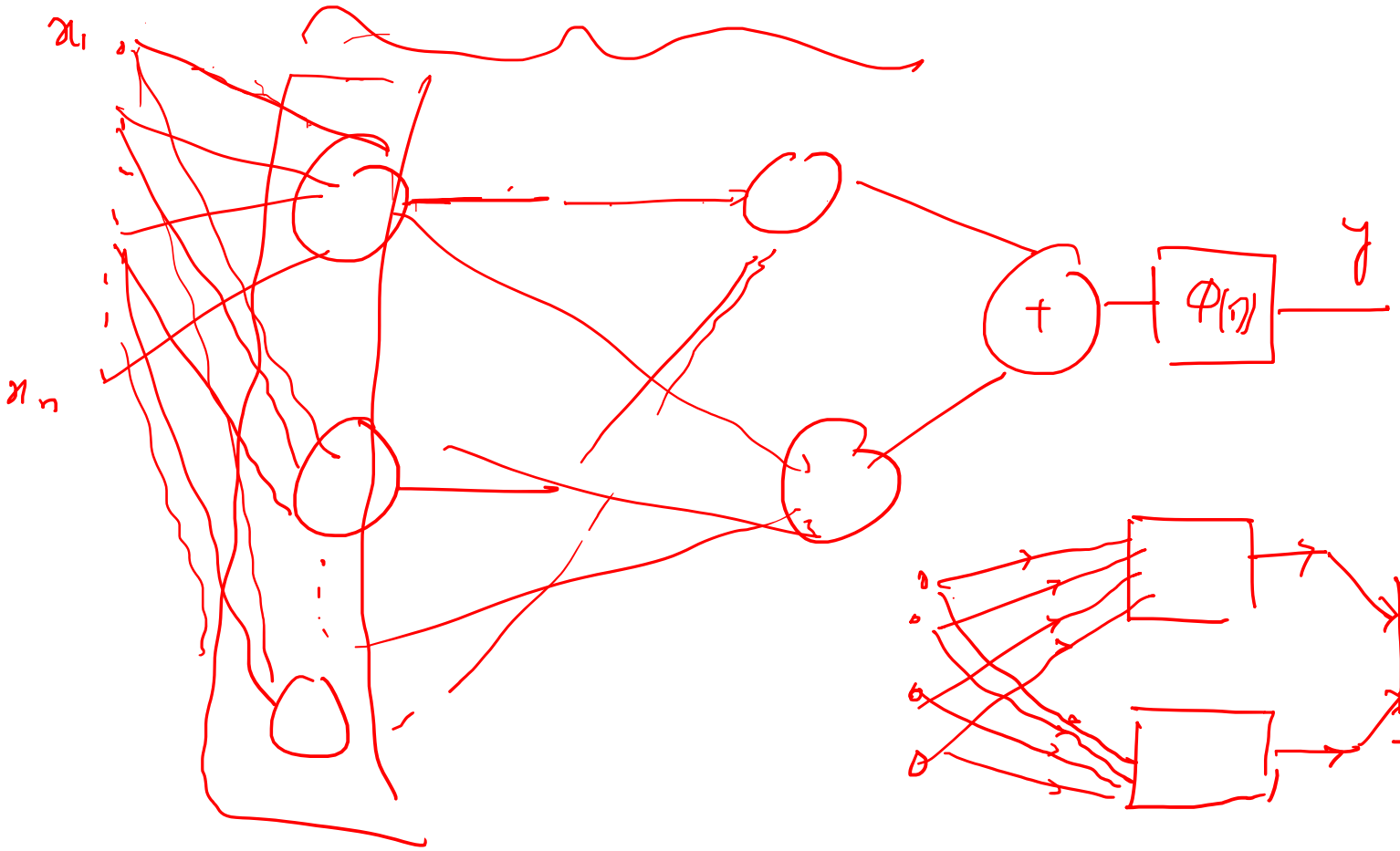
DNN

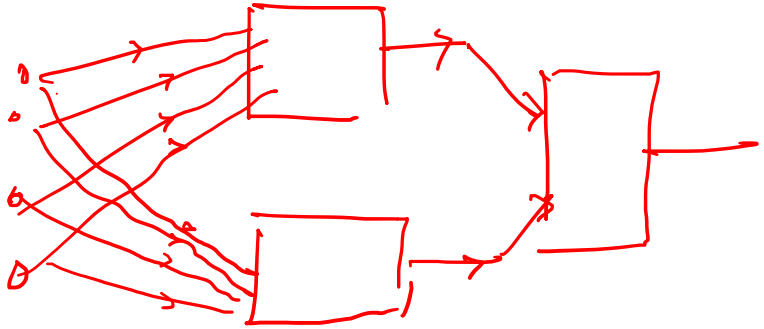
GPU

filter

filter

filter

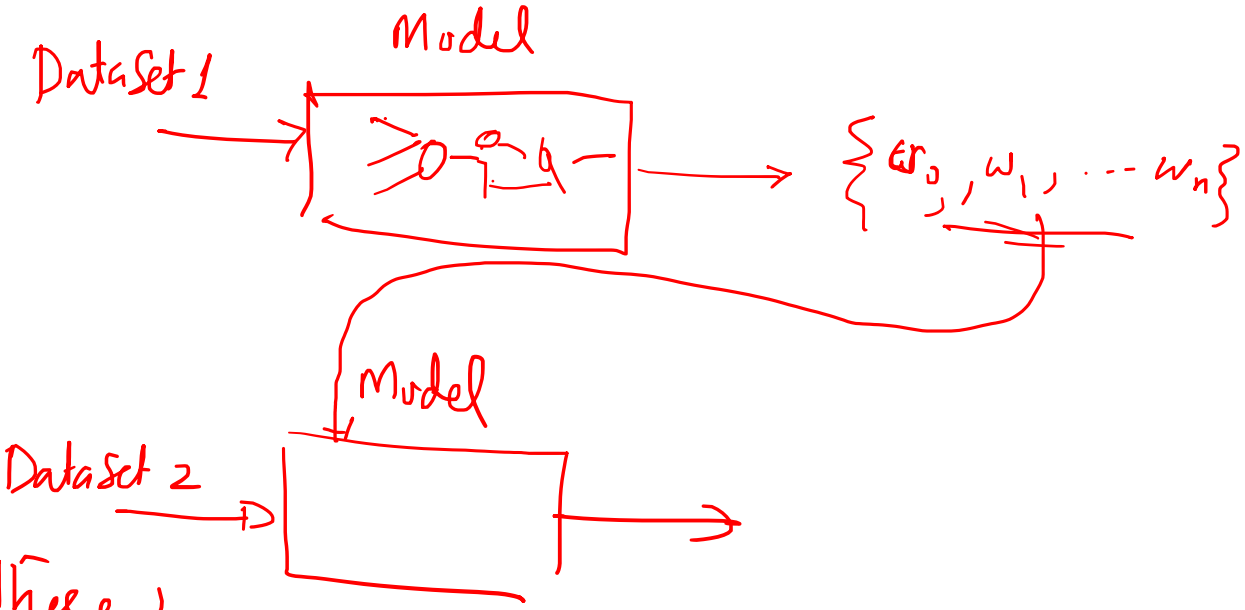




© 2021 Institute of Data

→ [$\underline{w}$ = random] → Algorithm

→ Transferred learning :



① feature selection (H will be there)

② Images → "CNN" : → extract features automatically

© 2021 Institute of Data

Perceptron Activation Function

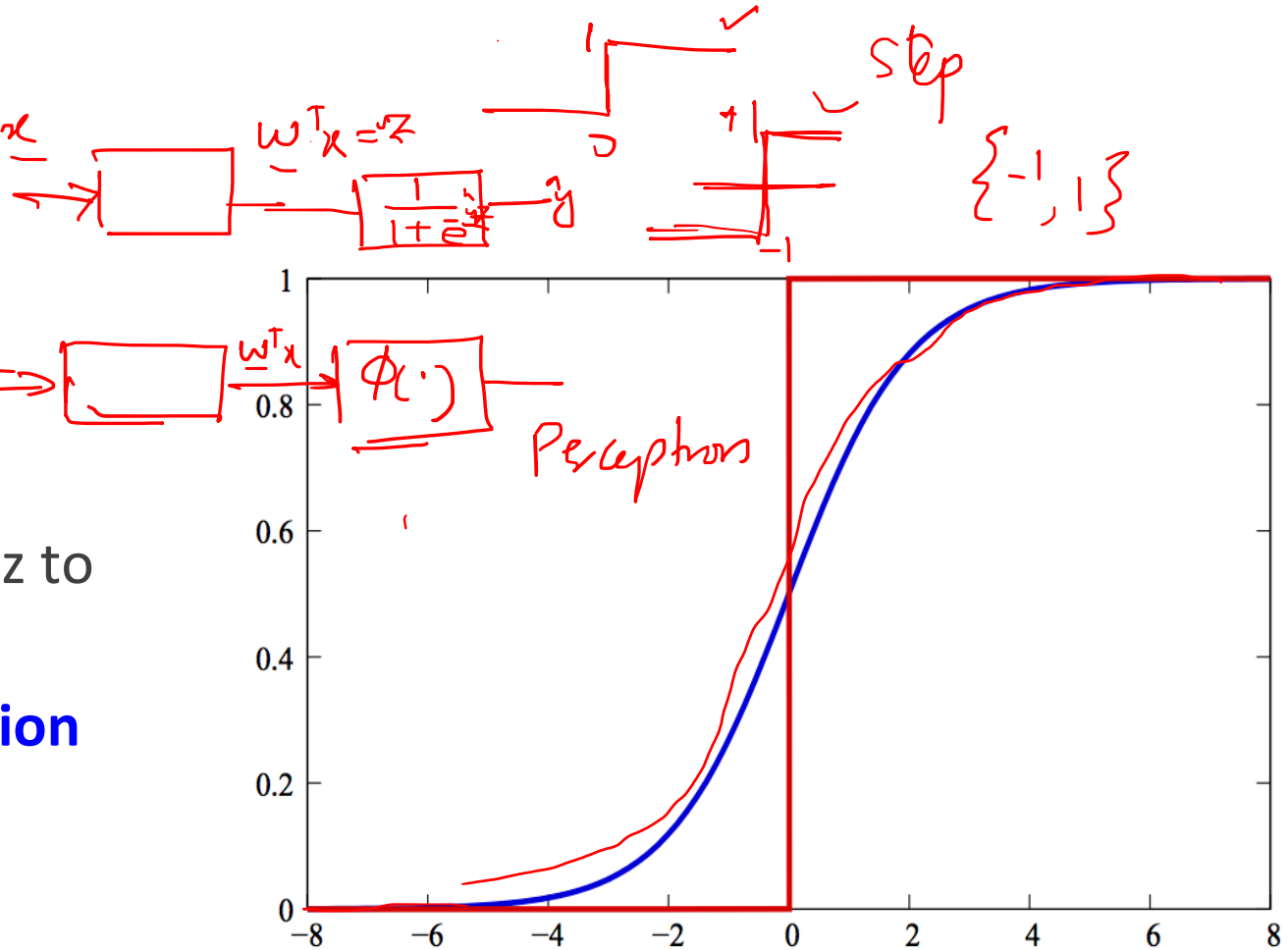
- z is the **weighted sum of inputs** (similar to Logistic Regression):

$$z = \sum_{j=0}^{n} w_j \, x_j$$

- A transfer function f(z) converts z to the output of the node
- f(z) is called the **activation function**

example:

$$f_{log}(z) = \frac{1}{1 + e^{-z}}$$

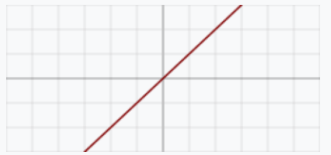


© 2021 Institute of Data

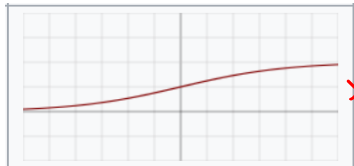
Perceptron Activation Functions

Activation function

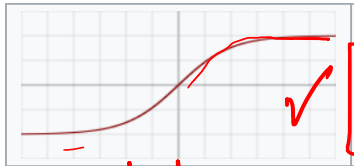
- Keras



$$f(x) = x$$



$$f(x) = \frac{1}{1+e^{-x}}$$



$$f(x) = \tanh(x) = \frac{2}{1+e^{-2x}} - 1$$

DNN

$$f(x) = \begin{cases} 0 & \text{for } x < 0 \\ x & \text{for } x \geq 0 \end{cases}$$

- Linear
- Logistic
- Hyperbolic tangent
- **Rectified linear unit (ReLU)**

unit

+ve → +1

-ve → -1

© 2021 Institute of Data

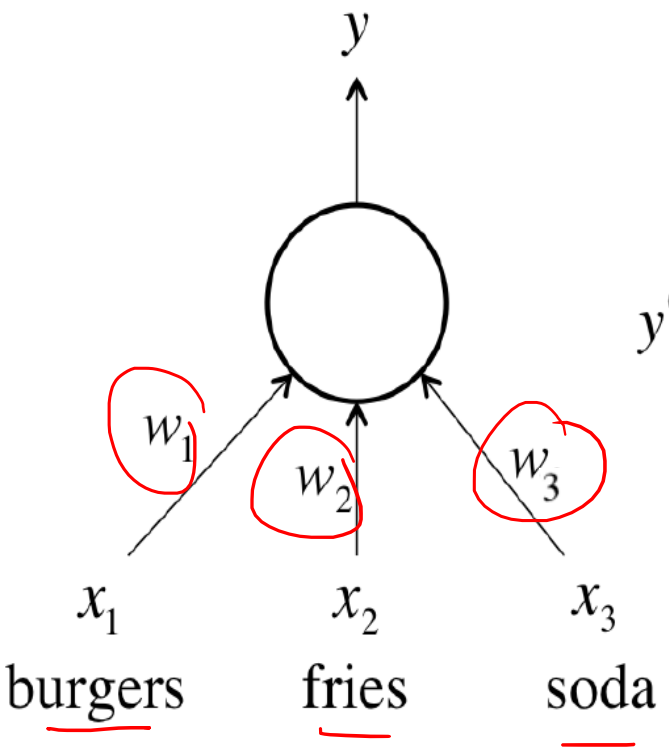
Linear Regression Perceptron

- Example: given a set of fast food orders and total prices, can we predict the unit prices?
- let

  x_1 = number of burgers
  x_2 = number of fries
  x_3 = number of sodas
  y = total price
  for any order
- object:

  compute the weights
  w_1, w_2, w_3
  that minimise the residuals of y

100 USD



$$y^{(i)} = w_1 x_1^{(i)} + w_2 x_2^{(i)} + w_3 x_3^{(i)}$$

© 2021 Institute of Data

Linear Classification Perceptron

- Given
  - a set of *n*-D **features** $\boldsymbol{X}$
  - a set of **labels** (response var) $\boldsymbol{y}$

- Objective:

  compute the weights $w_1, w_2, \ldots w_n$

  that describe the ***hyperplane*** that separates points $\boldsymbol{X}$ by class $\boldsymbol{y}$

  example: labels $\boldsymbol{y} \in \{-1, 1\}$

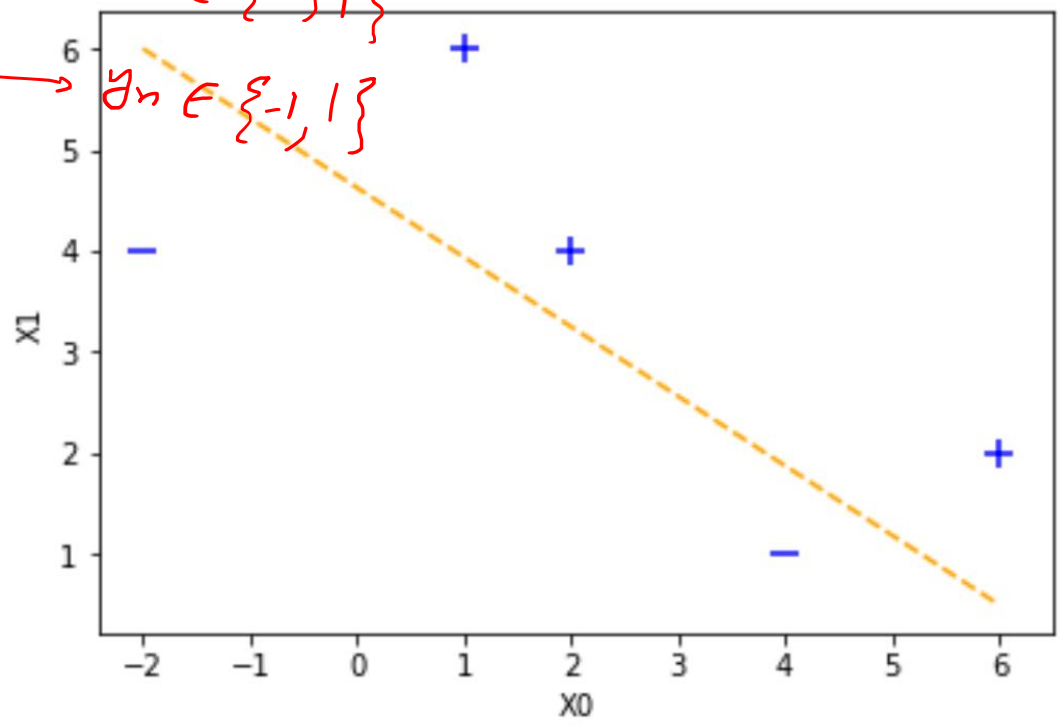
SEE LAB 5.2

$x_1 \rightarrow y_1 \in \{-1, 1\}$

$x_2 \rightarrow y_2 \in \{-1, 1\}$

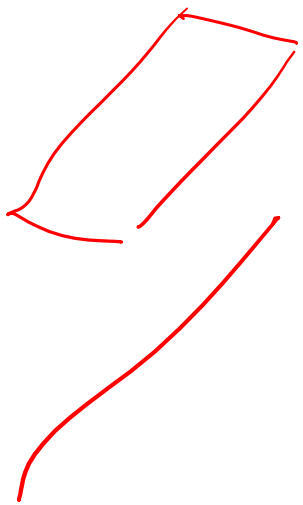
$\vdots$

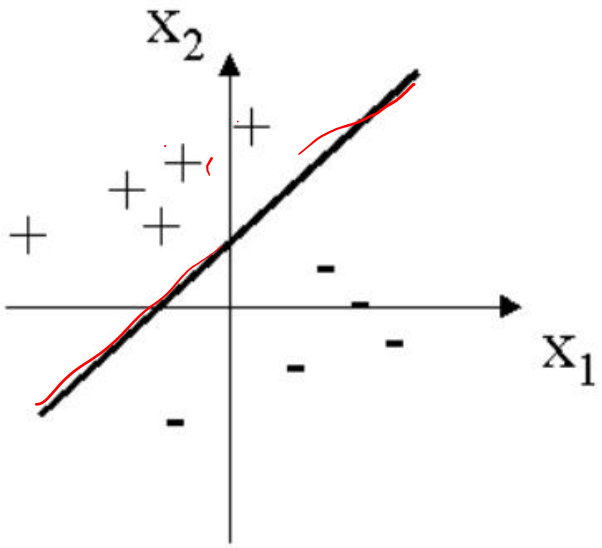
$x_n \rightarrow y_n \in \{-1, 1\}$



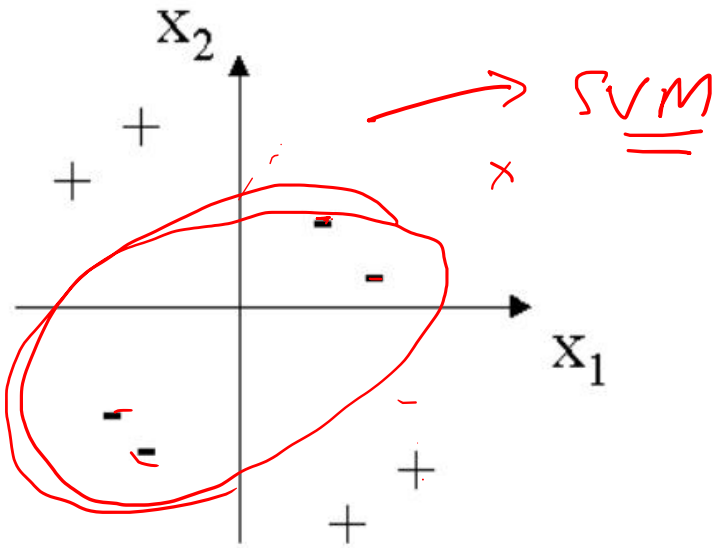
© 2021 Institute of Data

Linearly separable





"Linearly Separable"



Not Linearly Separable

$$w_1 x_1 + w_2 x_2 + \theta = 0$$

© 2021 Institute of Data

Details on plotting straight lines

- x1+x2-1=0
- Separable line (X1+x1-1 > 0)

- x1+x2-2 =0 (intercept change---moving line along vertical axis)

- x1+2*x2-1=0 (slope---tilting line)

- x1+2*x2-2=0

- Norm vector: (indicates the direction in which that function is increasing rapidly)

© 2021 Institute of Data

Back Propagation

"DNN"

How to **compute the weights**?

- Iteratively **adjust the weights** over the entire dataset until the algorithm converges to a solution:

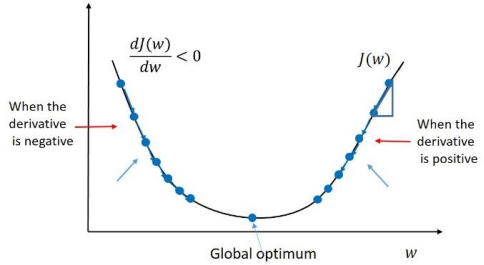
  error (at output) < *tolerance*

© 2021 Institute of Data

How to implement back propagation?

Gradient descent

- Error at a node is a function of the weights on the inputs flowing into the node
- Use gradient of error function to step in direction of decreasing error until minimum is found
- Update weight with a Learning Rate times the error and direction of the error (which is the derivative of the cost function)



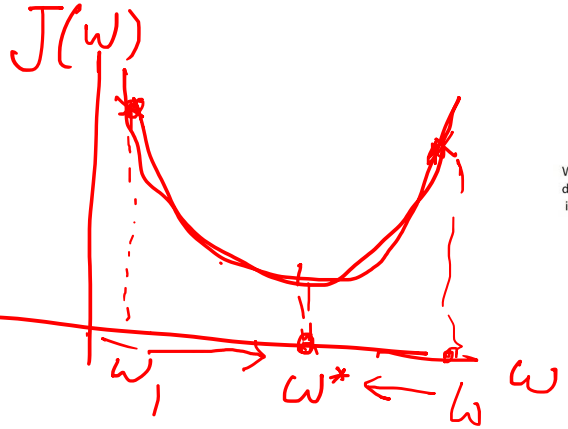
© 2021 Institute of Data

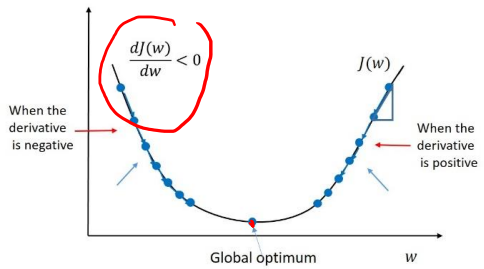
Objective

Gradient Descent

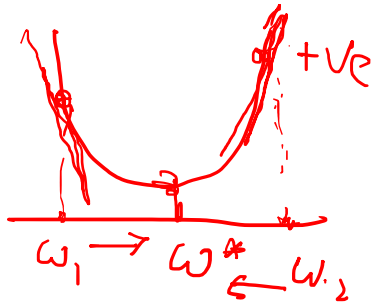
w = randomly

$$\boxed{w_{new} = w_{old} - \eta \frac{\partial}{\partial w} J(w)}$$





$$w_{new} = w_1 - \eta(-ve) = w_1 + \eta(+ve)$$



$$w_{new} = w_2 - \eta(+ve)$$

$$= w_2 - (-ve)$$

© 2021 Institute of Data

Stochastic Gradient Descent

w = w + Learning Rate * (expected - predicted) * x

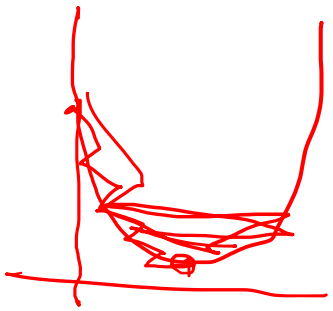
new old

Constant

error

W += '

SGD =

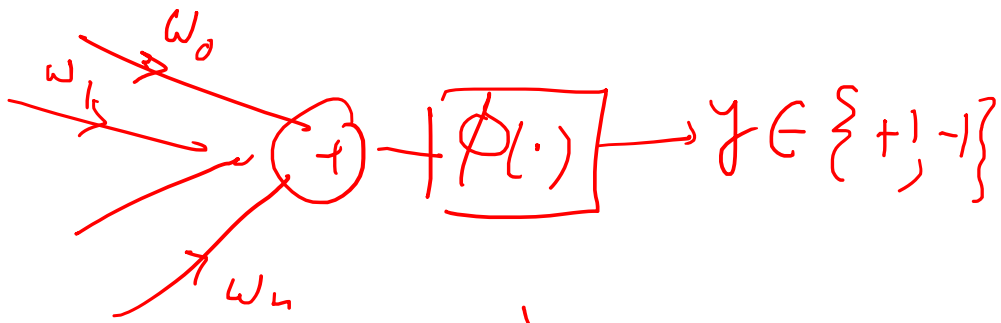


© 2021 Institute of Data

*

Perceptron → Basic NN → Human Brain

$$y = \phi(\underline{w}^T \underline{x})$$



Calculate the weight:

→ Stochastic Descent Algo

* Random weights →
* ~~Using~~ Misclassification → weights are updated

Based on ⇒

© 2021 Institute of Data

The Perceptron in Practice

https://demonstrations.wolfram.com/PerceptronAlgorithmInMachineLearning/

https://www.cs.cornell.edu/courses/cs4780/2018fa/lectures/lecturenote03.html

https://en.wikipedia.org/wiki/Perceptron

© 2021 Institute of Data

Perceptron Learning

GD : $\frac{\partial}{\partial x}$

'SGD' → doesn't

$w_i += w_i + eta\ x_i\ y_i$

Desired output $d(n) = \begin{cases} +1 & if\ x(n) \in set\ A \\ -1 & if\ x(n) \in set\ B \end{cases}$

Class A +ve | class B -ve

1. Select random sample from training set as input
2. If classification is correct, do nothing
3. If classification is incorrect, modify the weight vector *w* using

$w^T x$

new

old

$$w_i = w_i + \eta d(n) x_i(n)$$

weight ~~eq~~ update Equation

Repeat this procedure until the entire training set is classified correctly

https://www.youtube.com/watch?v=4Gac5I64LM4

© 2021 Institute of Data

Example of Perceptron learning

© 2021 Institute of Data

x_1	x_2	y
1	1	1

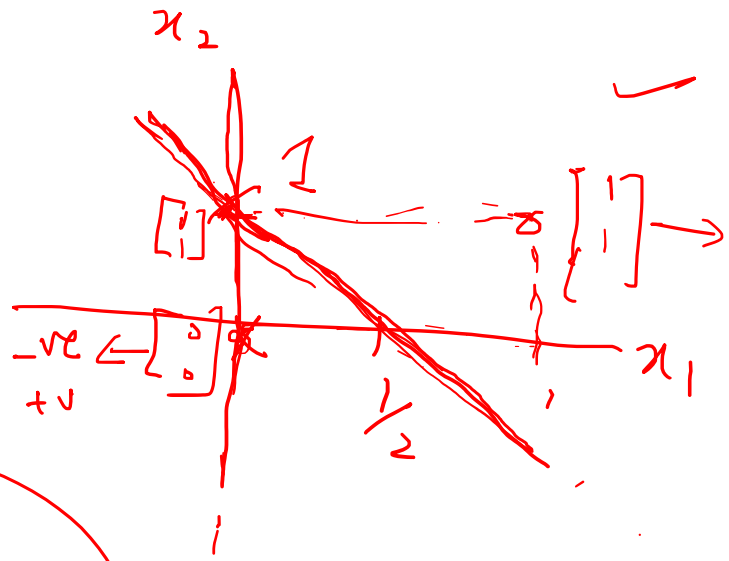
$$2 \cdot x_1 + 1 \cdot x_2 - 1 = 0$$

$$\boxed{\underline{w}^T \underline{x} = 0}$$

$$\underline{w} = \begin{bmatrix} -1 \\ 2 \\ 1 \end{bmatrix} \qquad \underline{x} = \begin{bmatrix} 1 \\ x_1 \\ x_2 \end{bmatrix}$$

$$-1 \times 1 + 2x_1 + x_2 = 0$$

$$\boxed{-1 + 2x_1 + x_2 = 0}$$



$\begin{bmatrix} 1 \\ 1 \end{bmatrix} \rightarrow$ -ve

$$\underline{x} = \begin{bmatrix} 1 \\ 0 \\ 1 \end{bmatrix}$$

$\boxed{\underline{w}^T \underline{x} = 0} \rightarrow$ will on the line

$$\underline{x} = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}$$

$\underline{w}^T \underline{x} = 2 > 0 \rightarrow$ above the line

$$\underline{x} = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}$$

$\underline{w}^T \underline{x} = -K < 0$

© 2021 Institute of Data

$$\boxed{2x + y = 0}$$

$$\begin{bmatrix} \cancel{1} \\ \cancel{-1/2} \end{bmatrix} \qquad \begin{bmatrix} 1/2 \\ -1 \end{bmatrix} \rightarrow \text{on the}$$

LHS :

$$2 \times \frac{1}{2} - 1 = 0 = \underline{RHS} \qquad w^T x = 0$$

$$w^T x > 0$$

$$2x + y > 0 \quad \text{above the line}$$

$$2x + y < 0 \quad \text{below the line}$$

$$w^T x < 0$$

© 2021 Institute of Data

* Any Linear Eq → Line, Plane, Hyperplane

$$w_0 x_1 + w_1 x_1 + \cdots + w_n x_n = 0$$

$$\rightarrow \boxed{\underline{w}^T \underline{x} = 0}$$

$w^T \underline{x} > 0 \rightarrow$ above

$w^T \underline{x} < 0 \rightarrow$ below

$w^T \underline{x} = 0 \rightarrow \underline{on}$

© 2021 Institute of Data

Initial guess

$[-2, 1] \rightarrow +1$

$[1, 1] \rightarrow -1$

- We have six data point (3 for each class and are linearly separable)

$3 \rightarrow +ve$

$3 \rightarrow -ve$

Initial Values:

$\eta = 0.2$

random weight

$$w = \begin{pmatrix} 0 \\ 1 \\ 0.5 \end{pmatrix}$$

$y = 1 \cdot w_0 + w_1 x_1 + w_2 x_2$

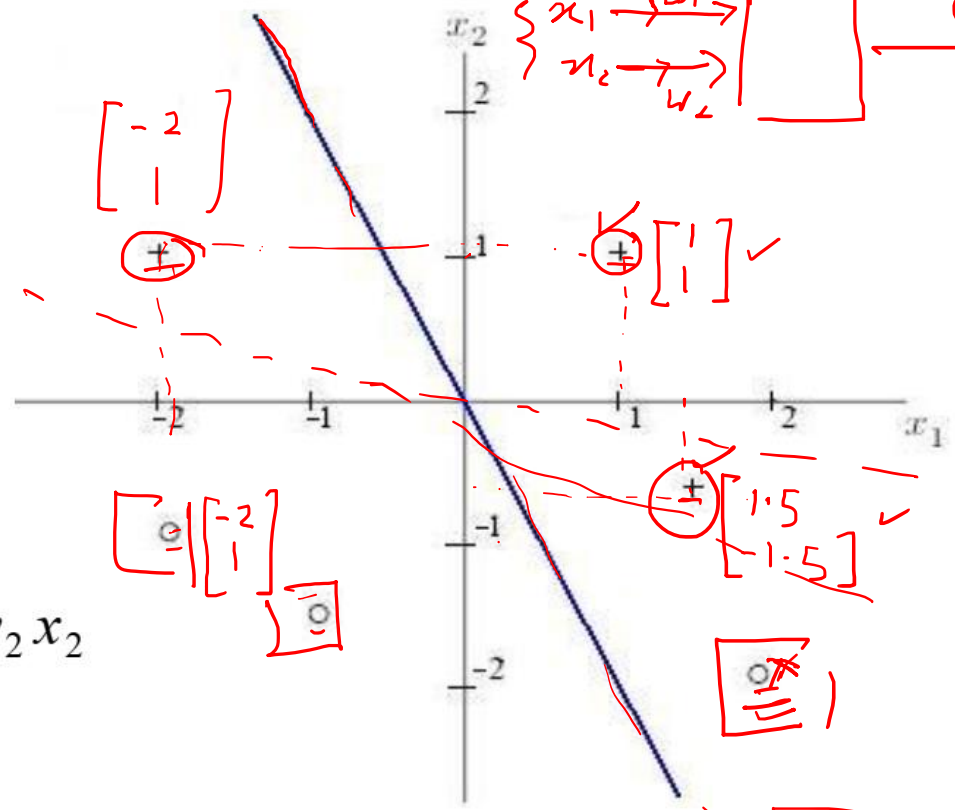
$= w^T x$

$$0 = w_0 + w_1 x_1 + w_2 x_2$$

$$= 0 + x_1 + 0.5 x_2$$

$$\Rightarrow x_2 = -2x_1$$

$= \begin{bmatrix} w_0 \\ w_1 \\ w_2 \end{bmatrix} \quad x = \begin{bmatrix} 1 \\ x_1 \\ x_2 \end{bmatrix}$



$1 \rightarrow w_0$, $x_1 \rightarrow w_1$, $x_2 \rightarrow w_2$ $\rightarrow y$

$\begin{bmatrix} -2 \\ 1 \end{bmatrix}$ $\begin{bmatrix} 1 \\ 1 \end{bmatrix}$ $\begin{bmatrix} 1.5 \\ -1.5 \end{bmatrix}$

$\begin{bmatrix} x_1 \\ x_2 \\ -1 \end{bmatrix}$ $\quad w^T x - b = 0$

© 2021 Institute of Data

Checking individual classification

- First one is correctly classified

$\boxed{x_2 = -2x_1}$

$\eta = 0.2$

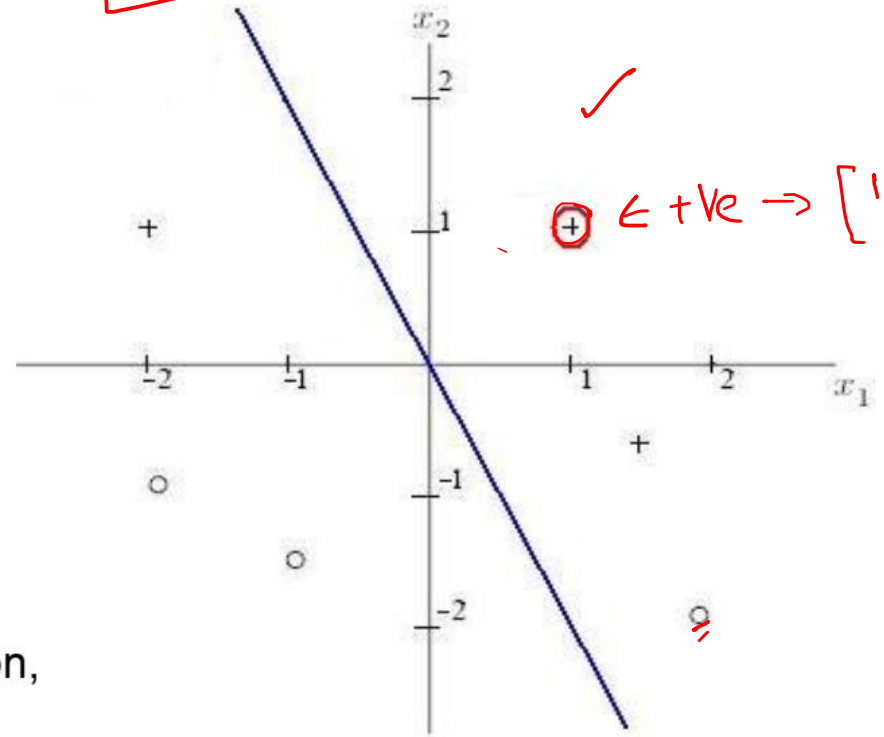
$$w = \begin{pmatrix} 0 \\ 1 \\ 0.5 \end{pmatrix}$$

$\underline{x} = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}$

$x_1 = 1,\ x_2 = 1$

$\boxed{w^T x > 0}$

Correct classification, no action



$\leftarrow$ +ve $\rightarrow \begin{bmatrix} 1 \\ 1 \end{bmatrix}$

© 2021 Institute of Data

Checking individual classification

- Wrongly classified, so need to update the weights (new line)

x_1	x_2	y
2	-2	-ve = -1

$d(n) = -1$

$\eta = 0.2$

$$w = \begin{pmatrix} 0 \\ 1 \\ 0.5 \end{pmatrix}$$

$x = \begin{bmatrix} 1 \\ 2 \\ -2 \end{bmatrix}$

$\boxed{w^T x > 0}$

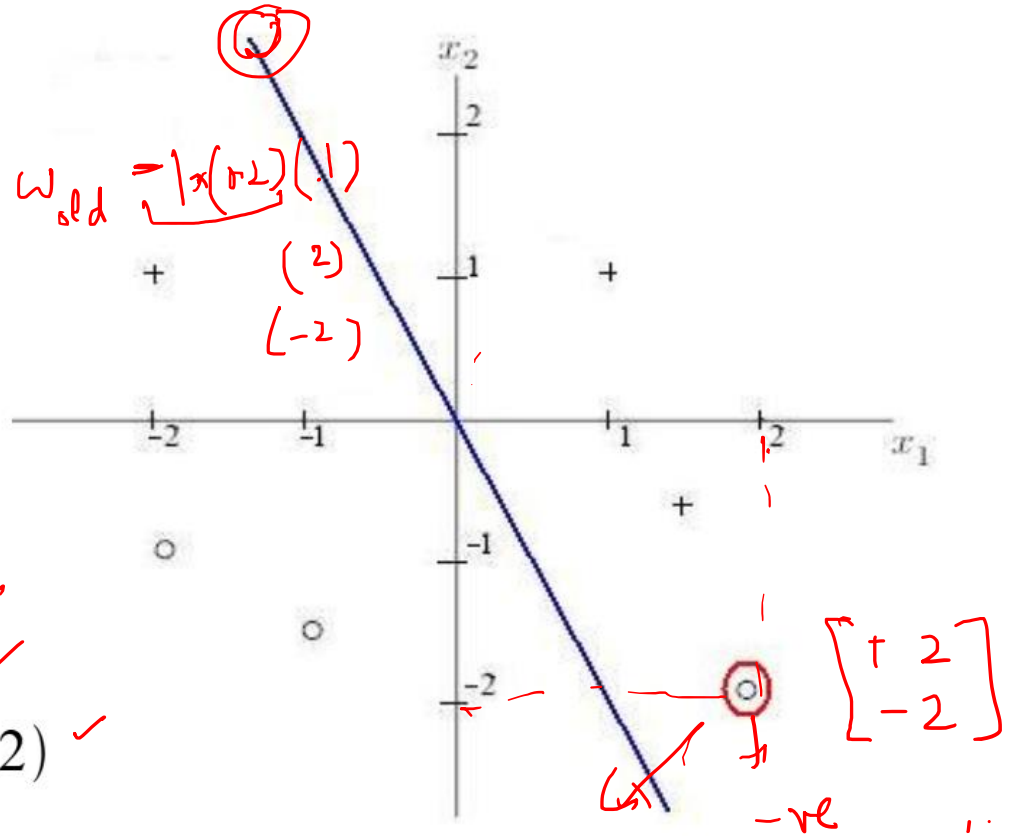
$x_1 = 2,\ x_2 = -2$

$$w_0 = w_0 - 0.2 * 1$$
$$w_1 = w_1 - 0.2 * 2$$
$$w_2 = w_2 - 0.2 * (-2)$$

$w_{new} = w_{old} - \eta x(n)(1)$

(2)

(-2)



$\begin{bmatrix} 1 & 2 \\ & -2 \end{bmatrix}$

-ve

Miss classification

© 2021 Institute of Data

updated line equation

$\eta = 0.2$

new

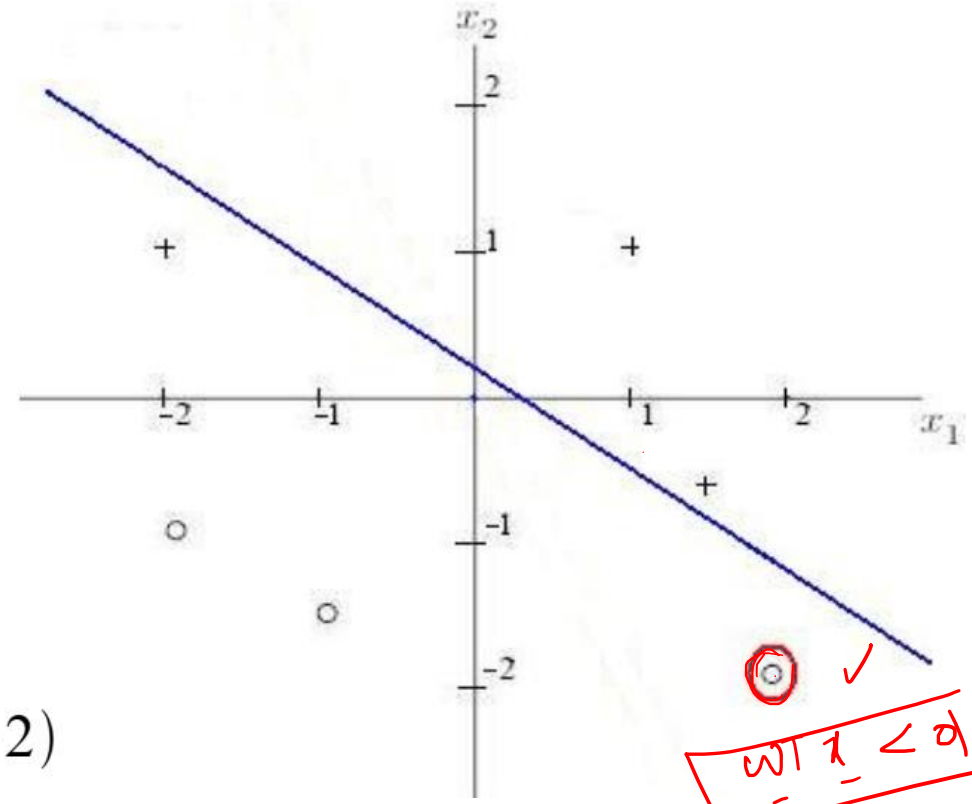
$$w = \begin{pmatrix} -0.2 \\ 0.6 \\ 0.9 \end{pmatrix}$$

$x_1 = 2,\ x_2 = -2$

$$w_0 = w_0 - 0.2 * 1$$

$$w_1 = w_1 - 0.2 * 2$$

$$w_2 = w_2 - 0.2 * (-2)$$



$w^T x < 0$

© 2021 Institute of Data

No change in line (Correct classification)

x_1	x_2	y
-2	1	+1

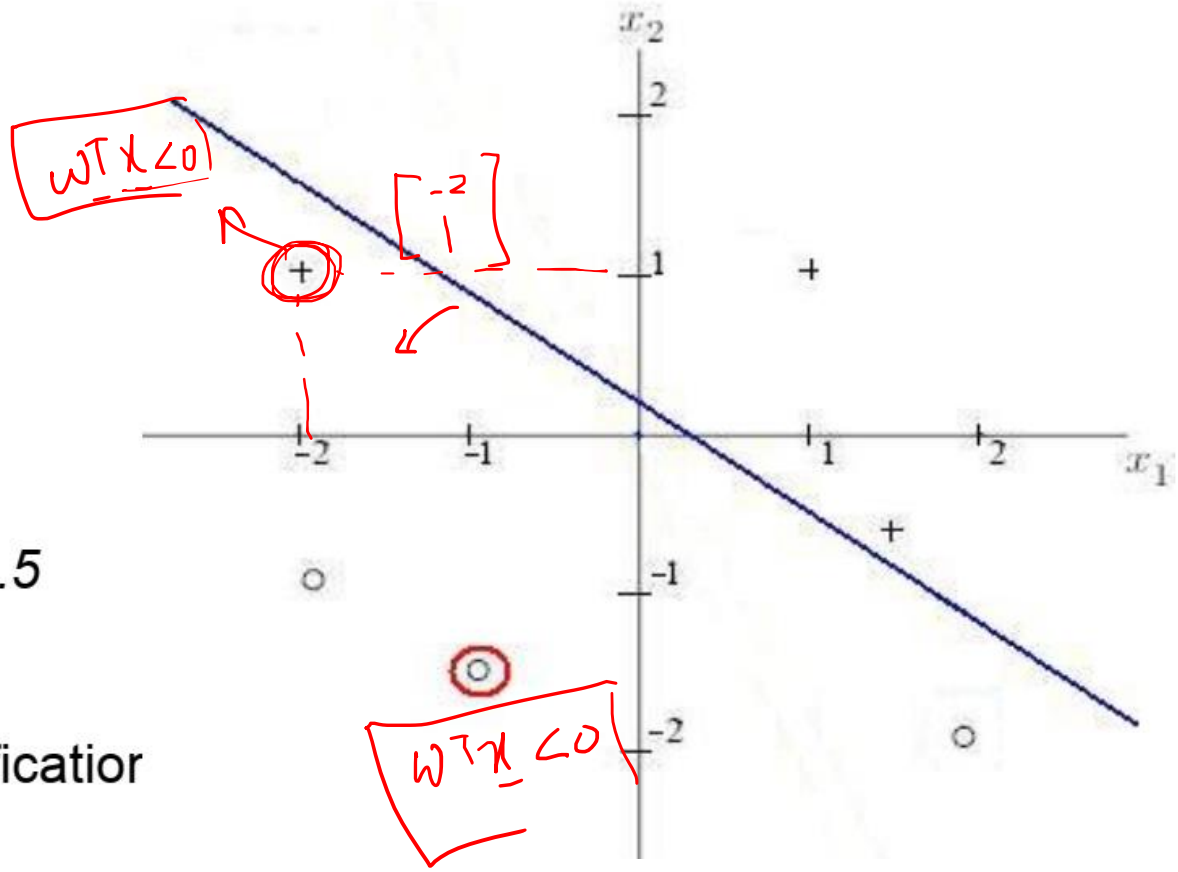
$\eta = 0.2$

$$w = \begin{pmatrix} -0.2 \\ 0.6 \\ 0.9 \end{pmatrix}$$

$x_1 = -1,\ x_2 = -1.5$

$w^T x < 0$

Correct classificatior
no action



© 2021 Institute of Data

No change in line (Correct classification)

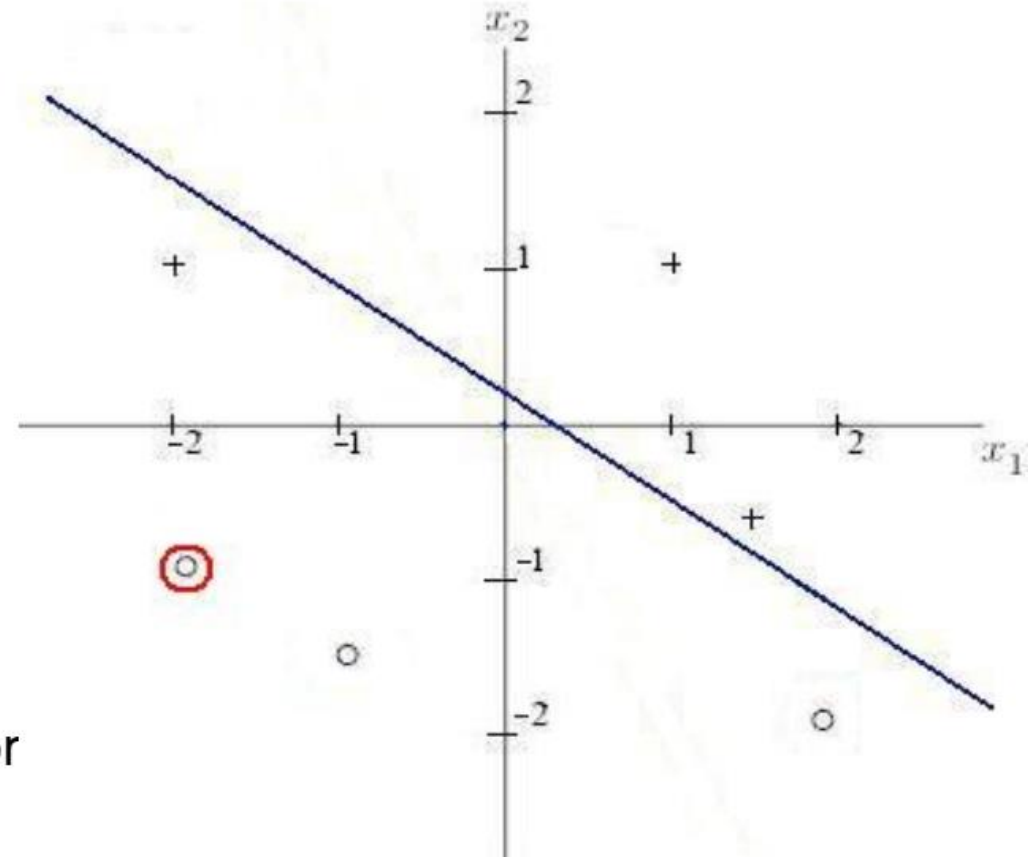
$\eta = 0.2$

$$w = \begin{pmatrix} -0.2 \\ 0.6 \\ 0.9 \end{pmatrix}$$

$x_1 = -2,\ x_2 = -1$

$w^T x < 0$

Correct classificatior
no action



© 2021 Institute of Data

Misclassification

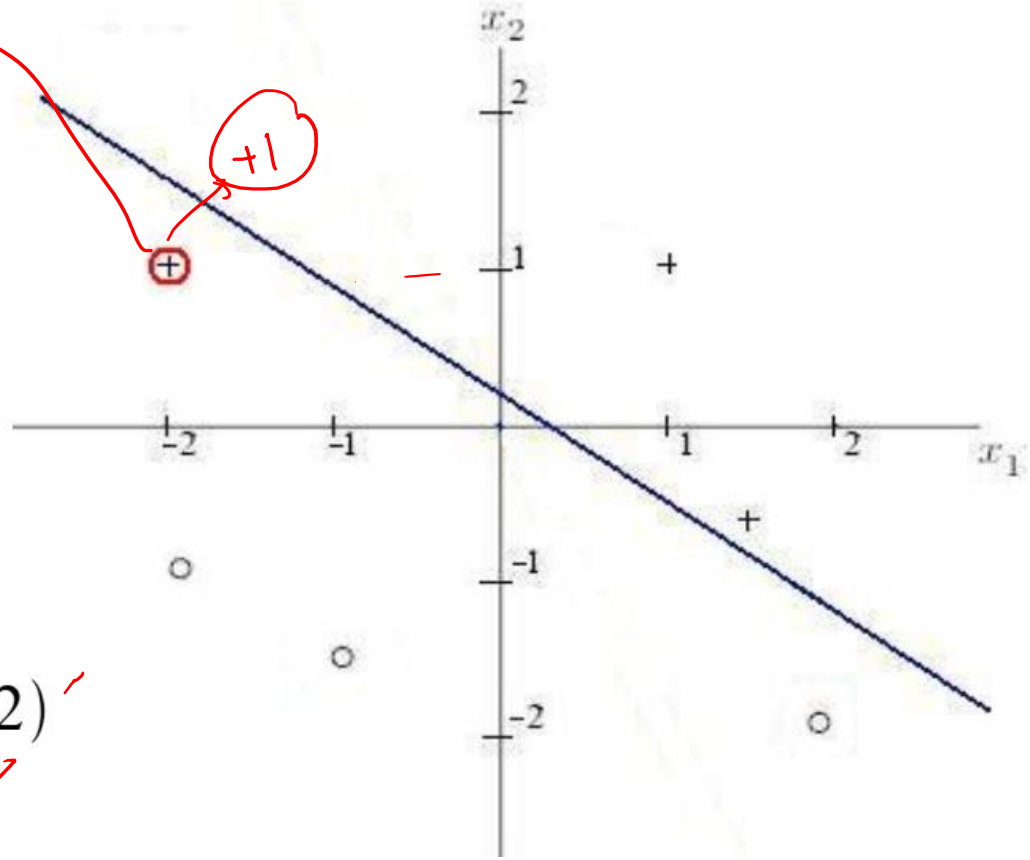
$x = \begin{bmatrix} 1 \\ -2 \\ 1 \end{bmatrix}$

$\eta = 0.2$

$$w = \begin{pmatrix} -0.2 \\ 0.6 \\ 0.9 \end{pmatrix}$$

$x_1 = -2,\ x_2 = 1$

$$w_0 = w_0 + 0.2 * 1$$
$$w_1 = w_1 + 0.2 * (-2)$$
$$w_2 = w_2 + 0.2 * 1$$



© 2021 Institute of Data

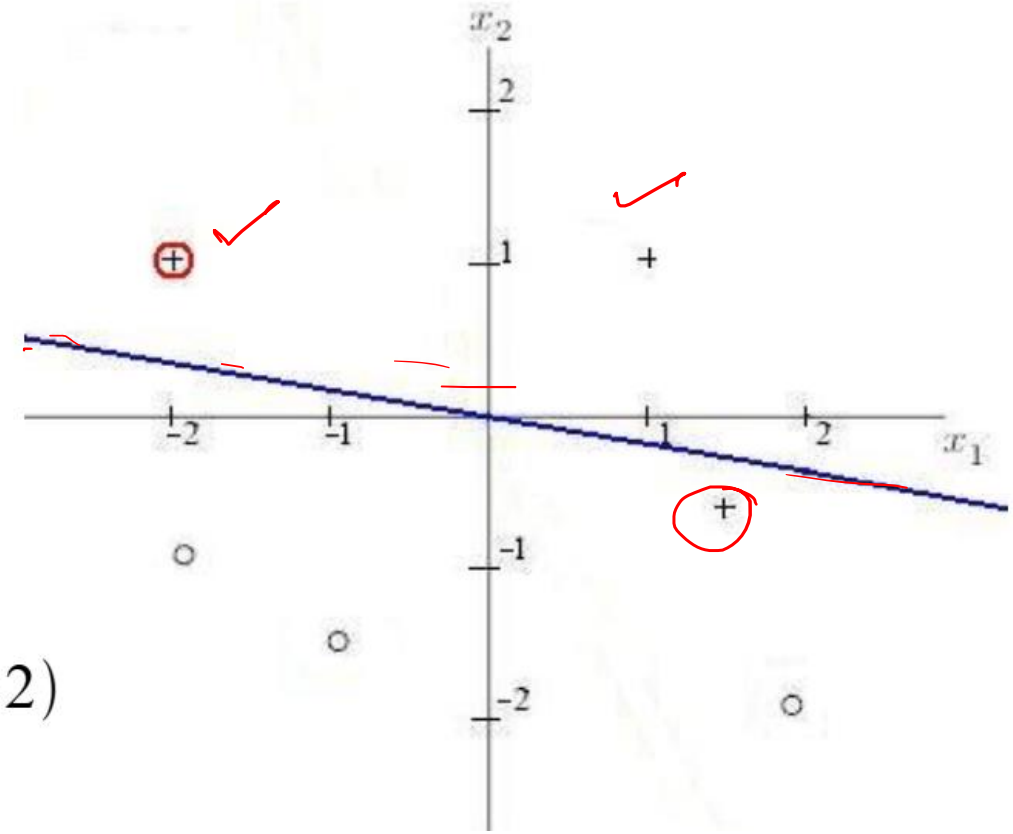
New update line

$\eta = 0.2$

$$w = \begin{pmatrix} 0 \\ 0.2 \\ 1.1 \end{pmatrix}$$

$x_1 = -2,\ x_2 = 1$

$$w_0 = w_0 + 0.2 * 1$$
$$w_1 = w_1 + 0.2 * (-2)$$
$$w_2 = w_2 + 0.2 * 1$$



© 2021 Institute of Data

Second round

- Misclassification

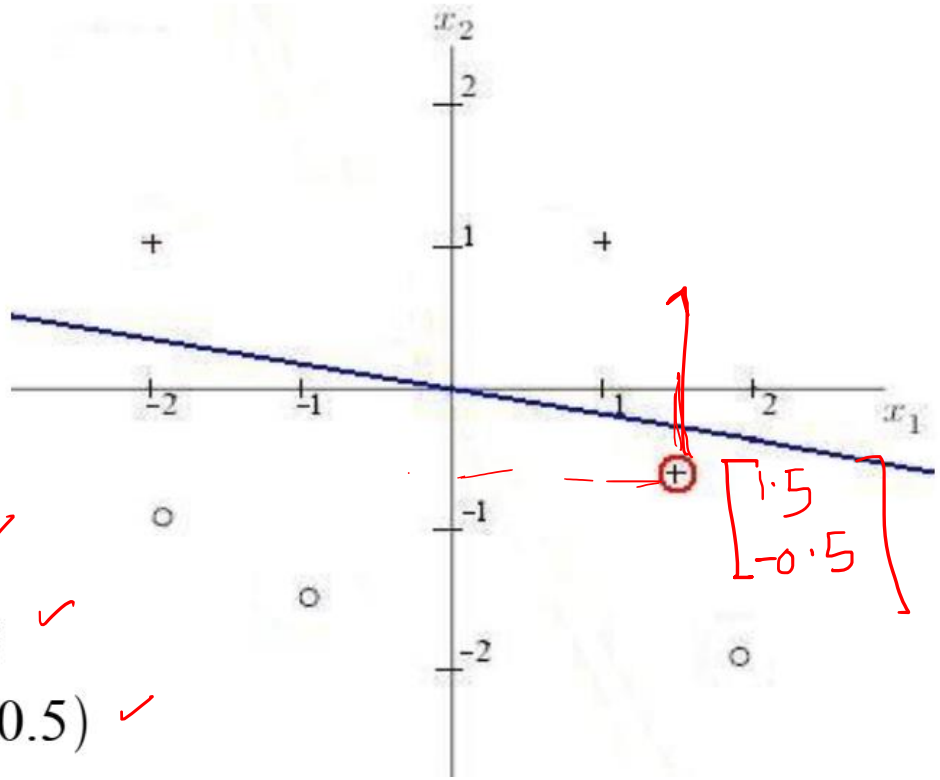
$\eta = 0.2$

$$w = \begin{pmatrix} 0 \\ 0.2 \\ 1.1 \end{pmatrix}$$

$$\underline{x} = \begin{bmatrix} 1 \\ 1.5 \\ -0.5 \end{bmatrix}$$

$x_1 = 1.5,\ x_2 = -0.5$

$$w_0 = w_0 + 0.2 * 1$$
$$w_1 = w_1 + 0.2 * 1.5$$
$$w_2 = w_2 + 0.2 * (-0.5)$$



© 2021 Institute of Data

Updated weights

- Proper classification

$w^T x \rightarrow np.dot(_, 1)$

$$\begin{bmatrix} 0.2 \\ 0.5 \\ 1 \end{bmatrix}$$

$\eta = 0.2$

$$\boxed{w = \begin{pmatrix} 0.2 \\ 0.5 \\ 1 \end{pmatrix}}$$

$$\underline{x} = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$$

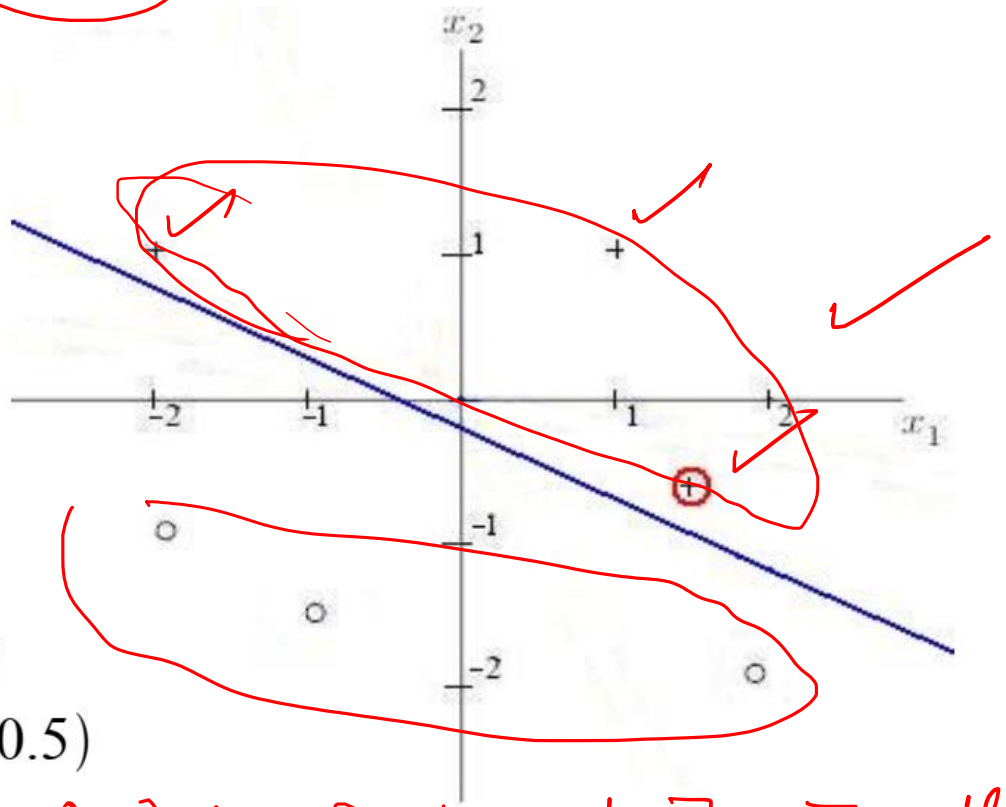
$x_1 = 1.5,\ x_2 = -0.5$

$$w_0 = w_0 + 0.2 * 1$$

$$w_1 = w_1 + 0.2 * 1.5$$

$$w_2 = w_2 + 0.2 * (-0.5)$$

$$\underline{x} = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}$$



$\underline{w}^T \underline{x} = \underline{x}^T \underline{w} = 0.2 + 0.5 + 1 = 1.7 > +ve$

© 2021 Institute of Data

The Perceptron in Practice

Logistic Regression

Naive Baye's

- in the *burgers/fries/sodas* example there should be an exact solution
  - (unless the unit process vary over the course of the dataset, in which case we would end up predicting the average price of each item)

Perceptron

SVM

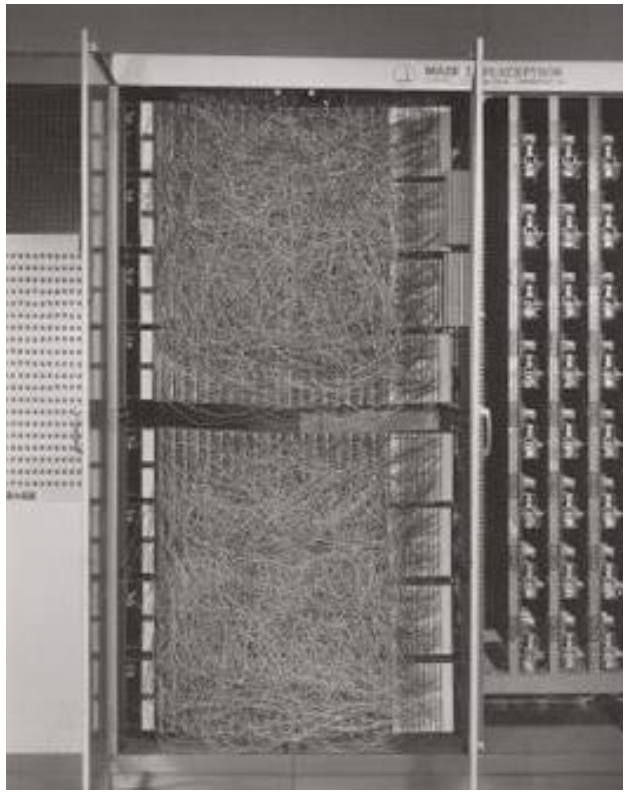
- in general ...
  - the data will not permit an exact solution
  - presence of local minima may cause convergence to fail
- in practice ...
  - use many perceptrons (***nodes***), organised into ***layers*** to learn more complex problems
    - > ~~neural network and Deep Learning~~ "Back Propagation"

© 2021 Institute of Data

Lab 5.2: Classification with a Perceptron

- Purpose:
  - To evaluate a simple perceptron for predicting classes from numeric features.

- Materials:
  - ‘Lab 5.2.ipynb’



Mark 1 Perceptron

By Source (WP:NFCC#4), Fair use, https://en.wikipedia.org/w/index.php?curid=47541432

© 2021 Institute of Data

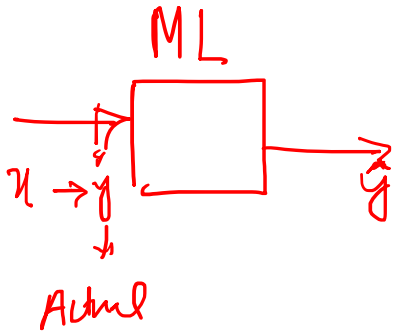
Discussion

- Strengths & weaknesses of perceptron-based prediction

© 2021 Institute of Data

Support Vector Machines (SVM)

- Concepts
- Linear SVMs
- Nonlinear SVMs
- Limitations
- Applications



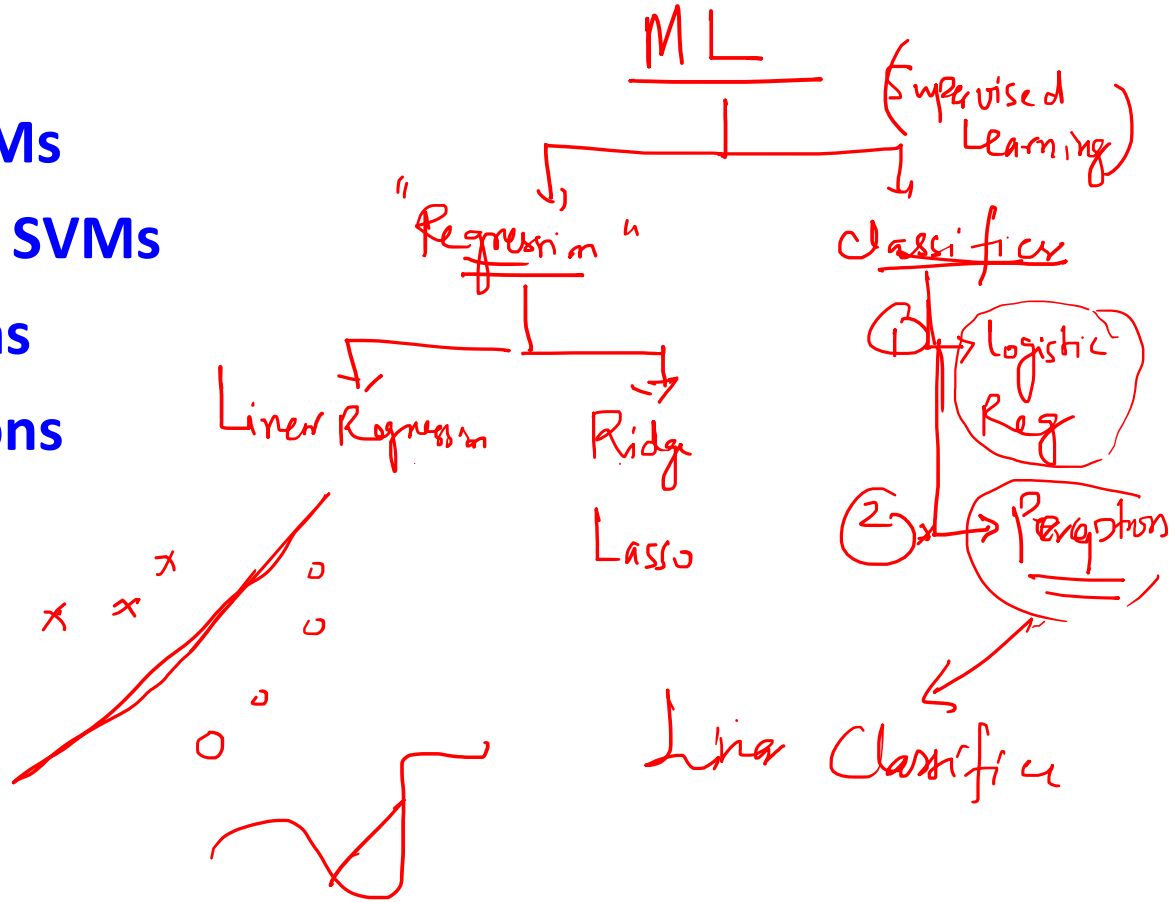
$e = (y - \hat{y})^2$

→ Line :

→ Plane :

→ Hyperplane :

$$\boxed{w^T x + b = 0}$$



© 2021 Institute of Data

Institute of Data

$$\boxed{w_{new} = w_{old} + (w^T x_i) y_i}$$

$$f[g(f(x))]$$

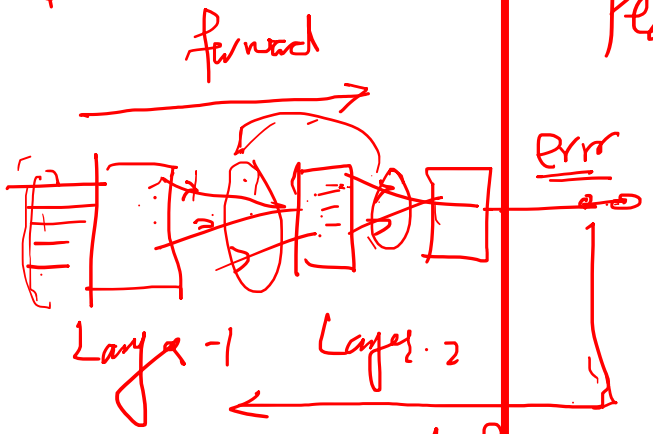
"Classifier" → Minimizing cost/objective function

* optimum Solution (Parameters) → "optimization"

(i) "Stochastic Gradient Descent"

(ii) SGD .....

Perceptron : ['weight update Equation]



→ "Chain Rule"

Neural Network s → Multiple Layers

→ Back Propagation To update the weights

© 2021 Institute of Data

Institute of Data

"Gradient Descent" (GD)

Objective function $= \|y - \hat{y}\|^2 = \sum (y_i - \hat{y}_i)^2 = J$

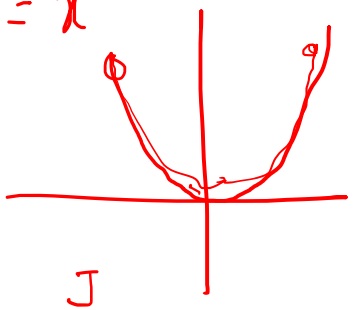
$x \longrightarrow$ Model $\longrightarrow \hat{y} \neq y$

$\hat{y}$: estimate, y: Actual

Cost function $\leftarrow \sum (y - \hat{y})^2$

$= \sum e^2$

$y = x^2$



J

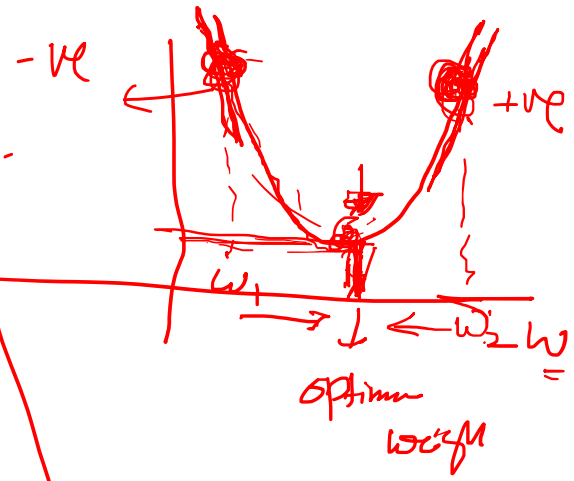
$\rightarrow \frac{\partial J}{\partial x} \rightarrow -ve$

$\frac{\partial J}{\partial x} \rightarrow +ve$

$$\boxed{w_{new} = w_{old} - \eta \frac{\partial J}{\partial w}}$$

η: rate of rate

Cost $= J$



w_1: $\frac{\partial J}{\partial w_1} \rightarrow -ve$

$w_{new} = w_{old} - \eta(-ve)$

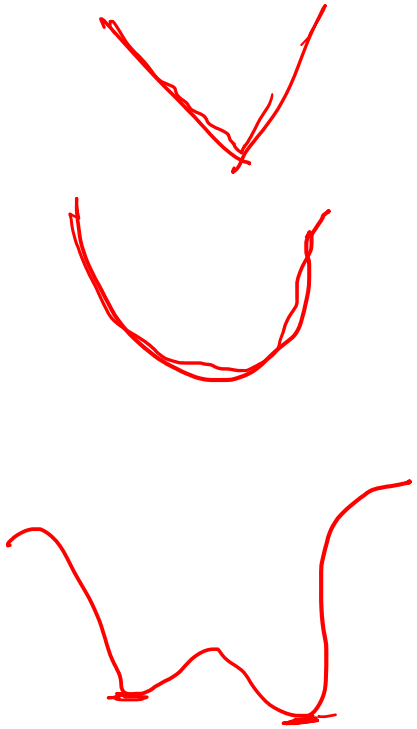
$= w_{old} + \eta \rightarrow$ increase

w_2: $\frac{\partial J}{\partial w_2} \rightarrow +ve$

$w_{new} = w_{old} - \eta(+ve)$

$= w_{old} - \eta(+ve)$

© 2021 Institute of Data



Cost function → (Square) loss : Convex Cost function

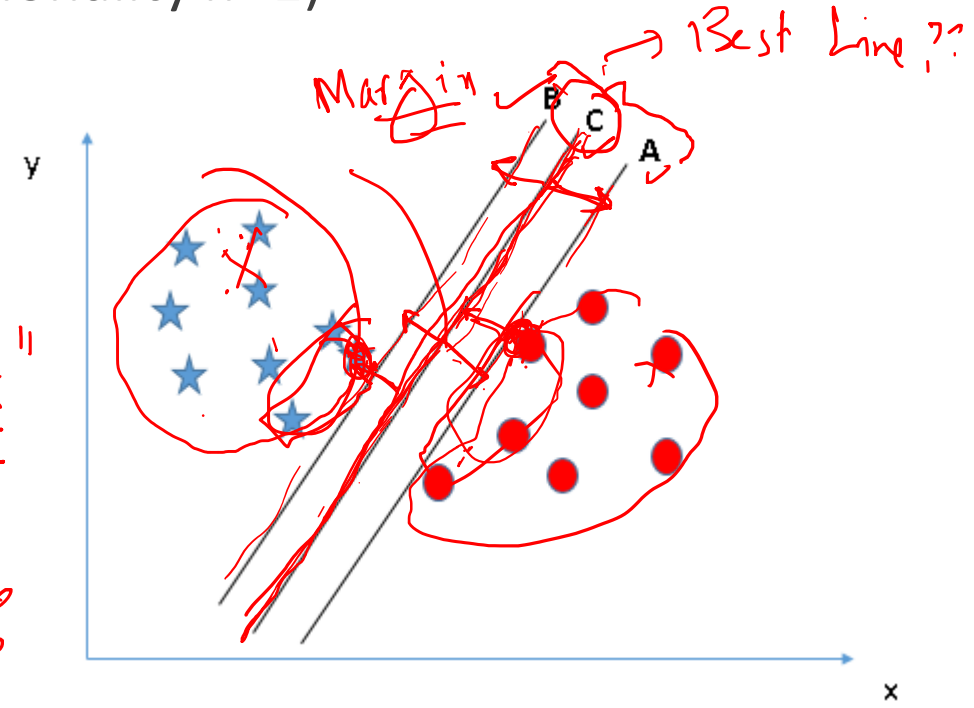
© 2021 Institute of Data

Support Vector Machines

- A **linear** algebraic method for separating ***n*-dimensional** data into classes
  - Data points are separated by a ***hyperplane*** (i.e. a boundary that has dimensionality *n*−1)

example:

2D

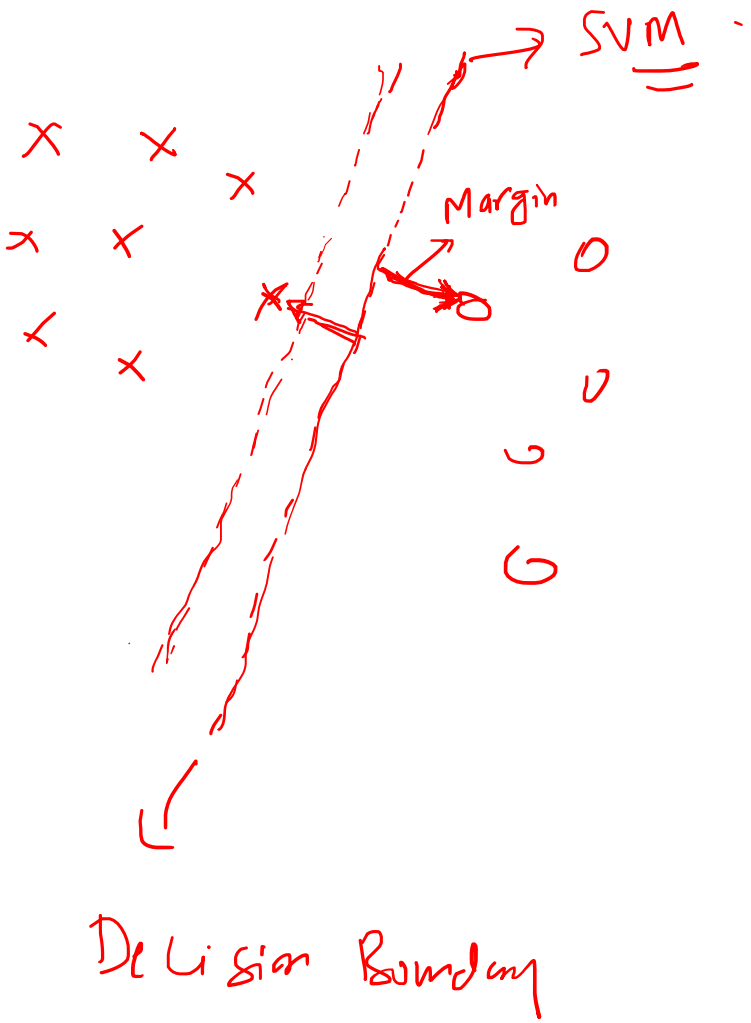


- lines A, B, C are **hyperplanes** in a 2D space
- each line correctly separates the two classes
- line C is preferred, because it **maximises** the average squared distance (***margin***) between the boundary and the points

① RBF [Radial Basis function]

② "Kernel Trick"

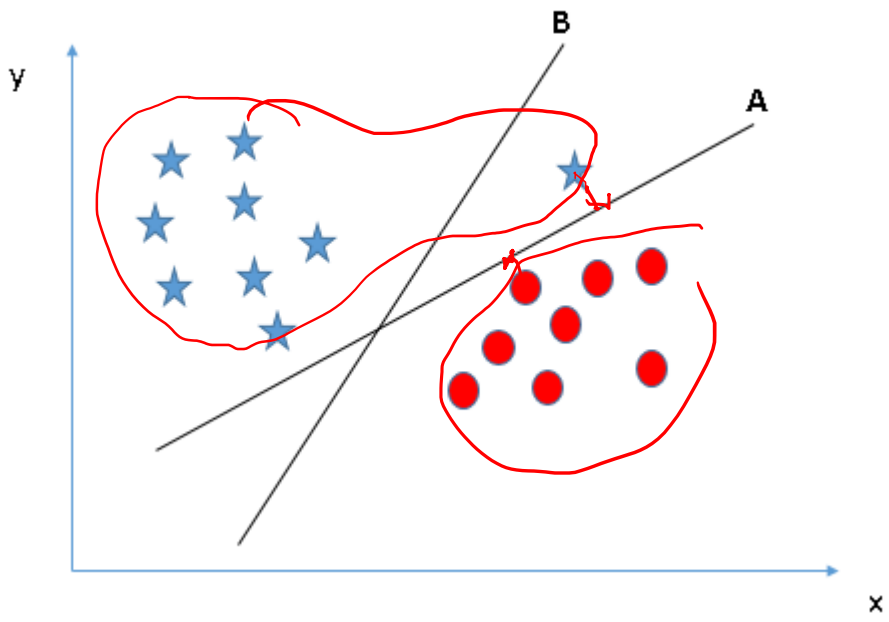
© 2021 Institute of Data



* Margin
* Decision Boundary

© 2021 Institute of Data

Support Vector Machines – cont'd



- In general, infinite hyperplanes are possible
- line B gives a larger margin
- line A correctly separates the classes
- line B will be chosen
  - **maximum separation** of classes is the first priority

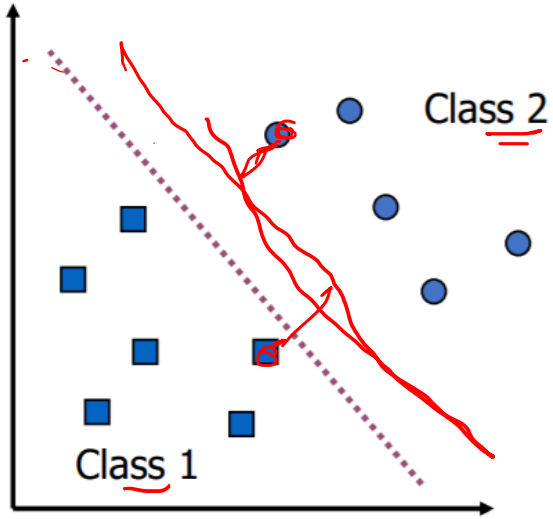
Support Vector Machine (SVM) finds an optimal solution

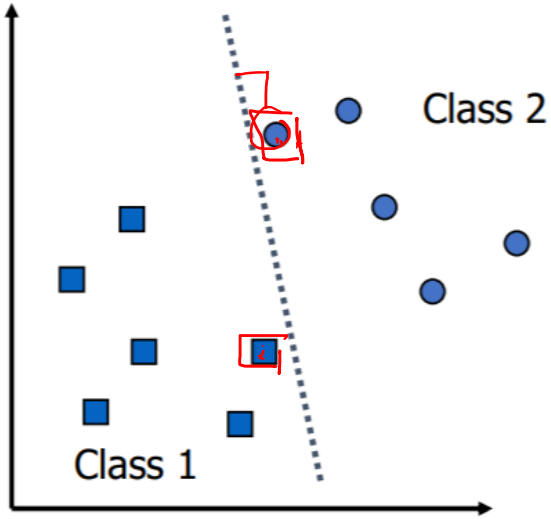
© 2021 Institute of Data

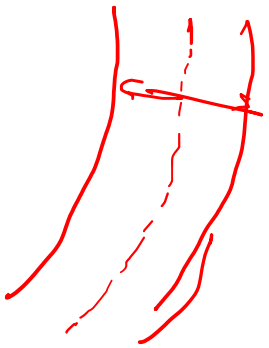
Support Vector Machines – cont'd

Street width

- Example of Bad Decision Boundaries





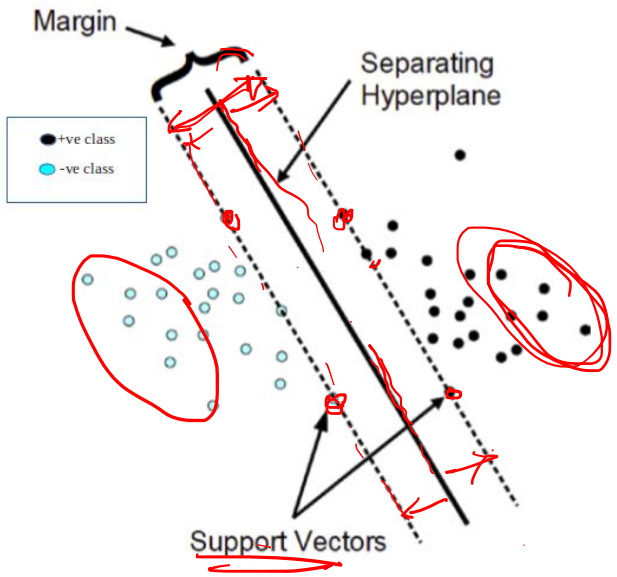


© 2021 Institute of Data

Support Vector Machines – cont'd

Hard Margin

- Support vectors are the data points that lie closest to the decision surface (or hyperplane)
- They are the data points most difficult to classify
- They have direct bearing on the optimum location of the decision surface



© 2021 Institute of Data

Support Vector Machines – cont'd

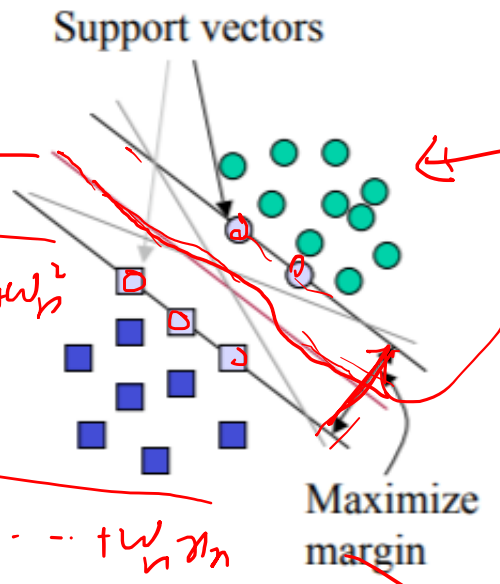
- SVMs maximize the margin (Winston terminology: the 'street') around the separating hyperplane.
- The decision function is fully specified by a (usually very small) subset of training samples, the support vectors.

$\min \|\underline{w}\|$

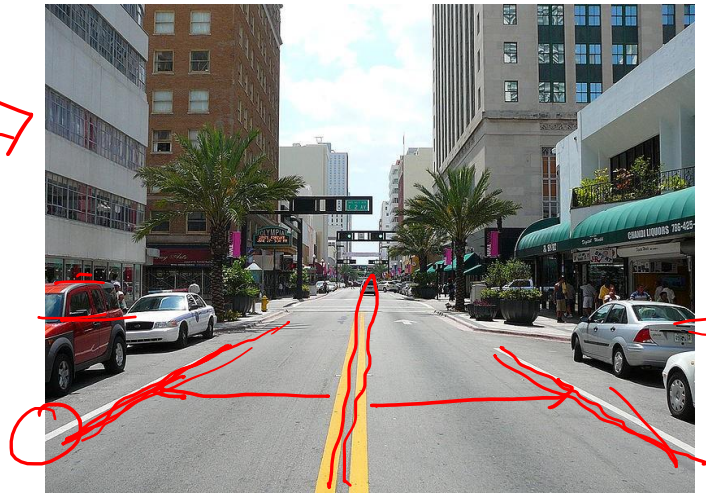
s.t. $\underline{w}^T \underline{x} - \gamma = 0$

$\|\underline{w}\| = \sqrt{w_0^2 + w_1^2 + \cdots + w_n^2}$

$\underline{w}_0 + \underline{w}_1 x_1 + \underline{w}_2 x_2 + \cdots + \underline{w}_n x_n - \gamma = 0$



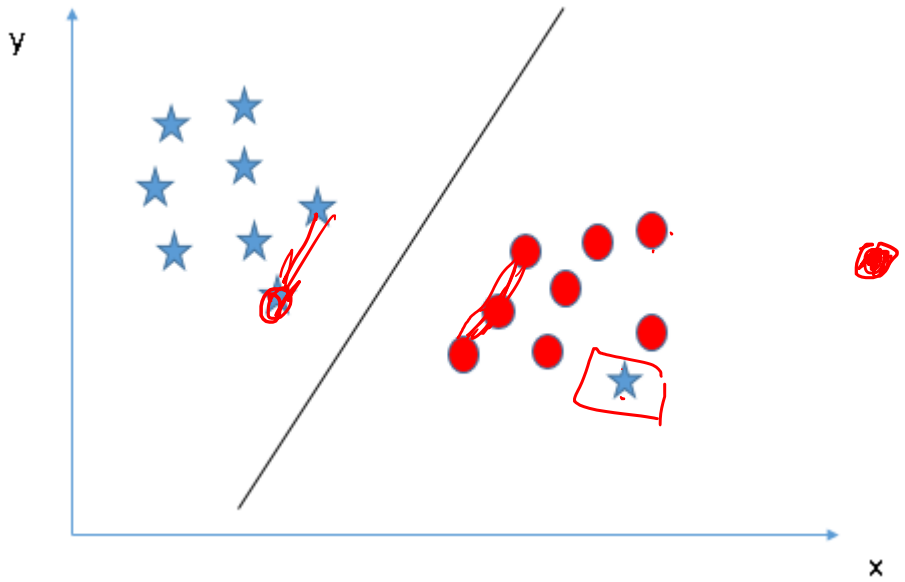
$\frac{2}{\|\underline{w}\|}$



Support Vectors

© 2021 Institute of Data

Support Vector Machines – cont'd



- the line separates the classes except for one outlier
- the presence of the outlier does not shift the line to the right
  - **SVMs are robust against outliers**

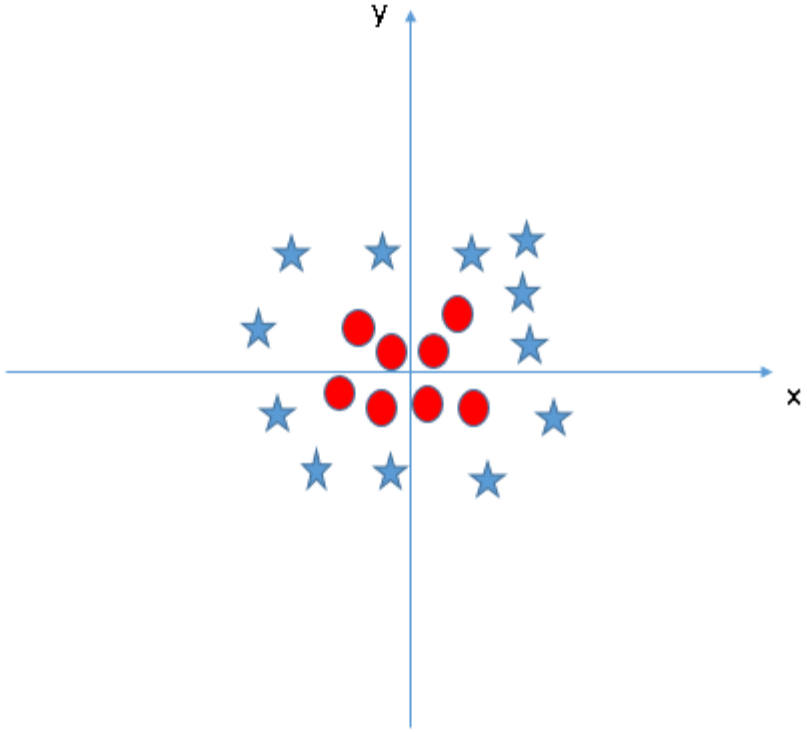
© 2021 Institute of Data

Support Vector Machines – cont'd

Kernel: "Non-linear Transformation"

"Non-linear SVM"

Not Linearly Seperable

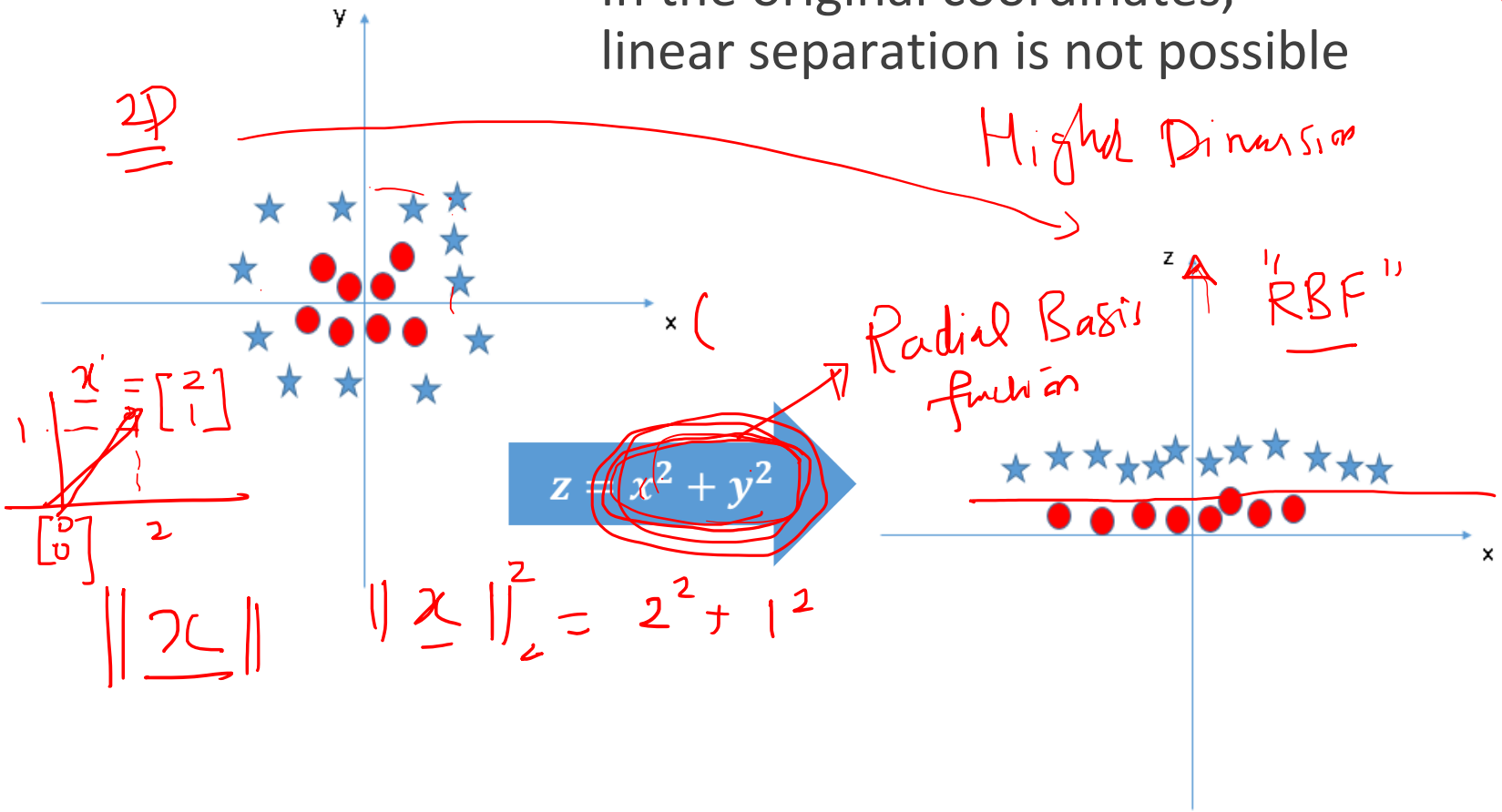


- there is no line that can separate these classes
  - will an SVM fail, here?
  - *Hint: we could* ***transform*** *the coordinates or create a new feature*

© 2021 Institute of Data

Support Vector Machines – cont'd

- In the original coordinates, linear separation is not possible



$e^{-\gamma\|x-c\|} \rightarrow \boxed{RBF}$

$\phi(\cdot) \rightarrow$ Transformation

- in these **coordinate**, there *is* a line that can separate the classes

  > the **kernel trick**:

  - an inseparable problem can be transformed into a separable problem in a higher dimension
  - the transform function is called the ***kernel***

© 2021 Institute of Data

Linear SVMs (Detail)

Assume linear separability for now (we will relax this later)

- In 2 dimensions, can be separated by a line – in higher dimensions, need hyperplanes

(i) Hard margin → "Separable"

- classes are separable by the margin around the solution hyperplane

(ii) Soft margin → "not separable"

SVM

- classes are not separable
- the best-fit hyperplane is computed by minimising the *hinge loss* function
  - incorporates a parameter that defines the trade-off between maximising the margin and minimising the number of misclassified points

© 2021 Institute of Data

Linear SVM with Hard Margin

$\underline{w}^T \underline{x} - b = 0$ $\quad (x_1, y_1) = \underline{x}_p$

$|\underline{w}^T \underline{x}_p - b| / \|\underline{w}\|$

- equation of hyperplane is

$$\vec{w} \cdot \vec{x} - b = 0$$

- compute $\vec{w}$, b so as to maximise width of margin

$$2/\|\vec{w}\|$$

Optimization of maximizing the margin ('street width') to reduce the number of weights that are nonzero to just a few that correspond to the support vectors

$2x + 3y - 1 = 0$

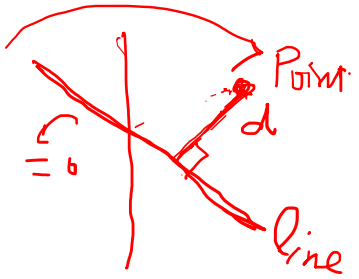
$\begin{bmatrix} 2 \\ 1 \end{bmatrix}$

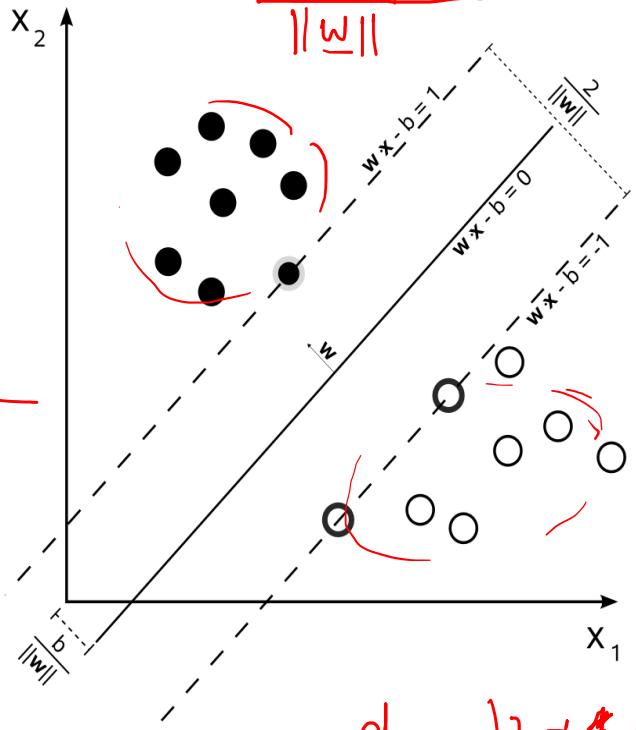
$\dfrac{|2 \times 2 + 3 \times 1 - 1|}{\sqrt{2^2 + 3^2}}$

Point $x_p(x_1, y_1)$

$\underline{w}^T \underline{x} + b = 0$

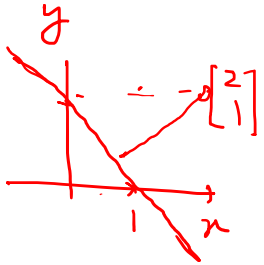
$d = \dfrac{|\underline{w}^T \underline{x}_p - b|}{\|\underline{w}\|}$





$1x + 1y - 1 = 0$

$x_p = \begin{bmatrix} 2 \\ 1 \end{bmatrix}$



$d = \dfrac{|2 + 1 - 1|}{\sqrt{1^2 + 1^2}} = \dfrac{2}{\sqrt{2}} = \sqrt{2}$

https://en.wikipedia.org/wiki/Distance_from_a_point_to_a_line

© 2021 Institute of Data

Linear SVM with Hard Margin

SVM

- equation of hyperplane is

$$\boxed{\vec{w} \cdot \vec{x} - b = 0}$$

$w_1 x_1 + w_2 x_2 + w_0 = 0$

$\|w\| \to \min$

- compute $\vec{w}$, b so as to maximise width of margin

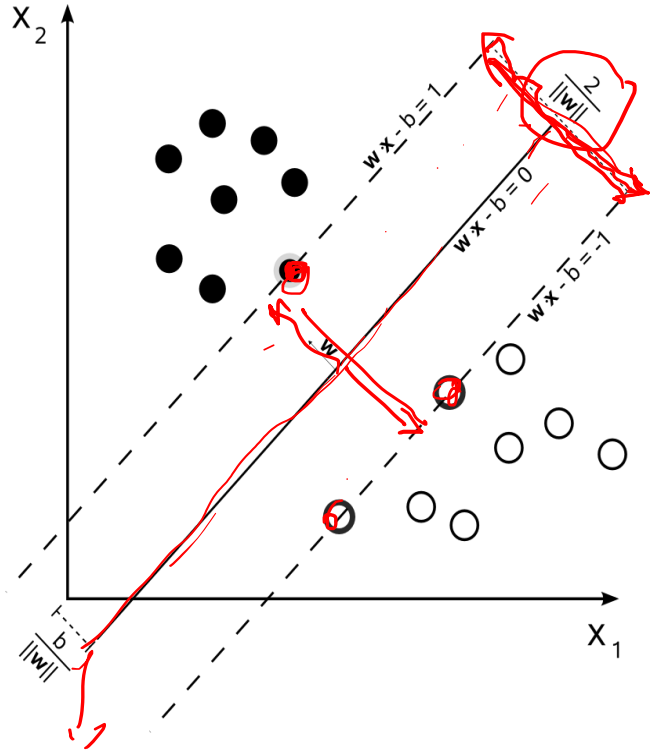
$$\boxed{2/\|\vec{w}\|} \to \max$$

$\|w\| = \sqrt{w_0^2 + w_1^2 + w_2^2}$

$\frac{1}{x}$ ↓ min

Optimization of maximizing the margin ('street width') to reduce the number of weights that are nonzero to just a few that correspond to the support vectors

Visual $\|w\|$??



$w^T x - b = 0$ s.t. $\min \|w\|$ → Constrained optimization Pb

- find weights

https://en.wikipedia.org/wiki/Distance_from_a_point_to_a_line

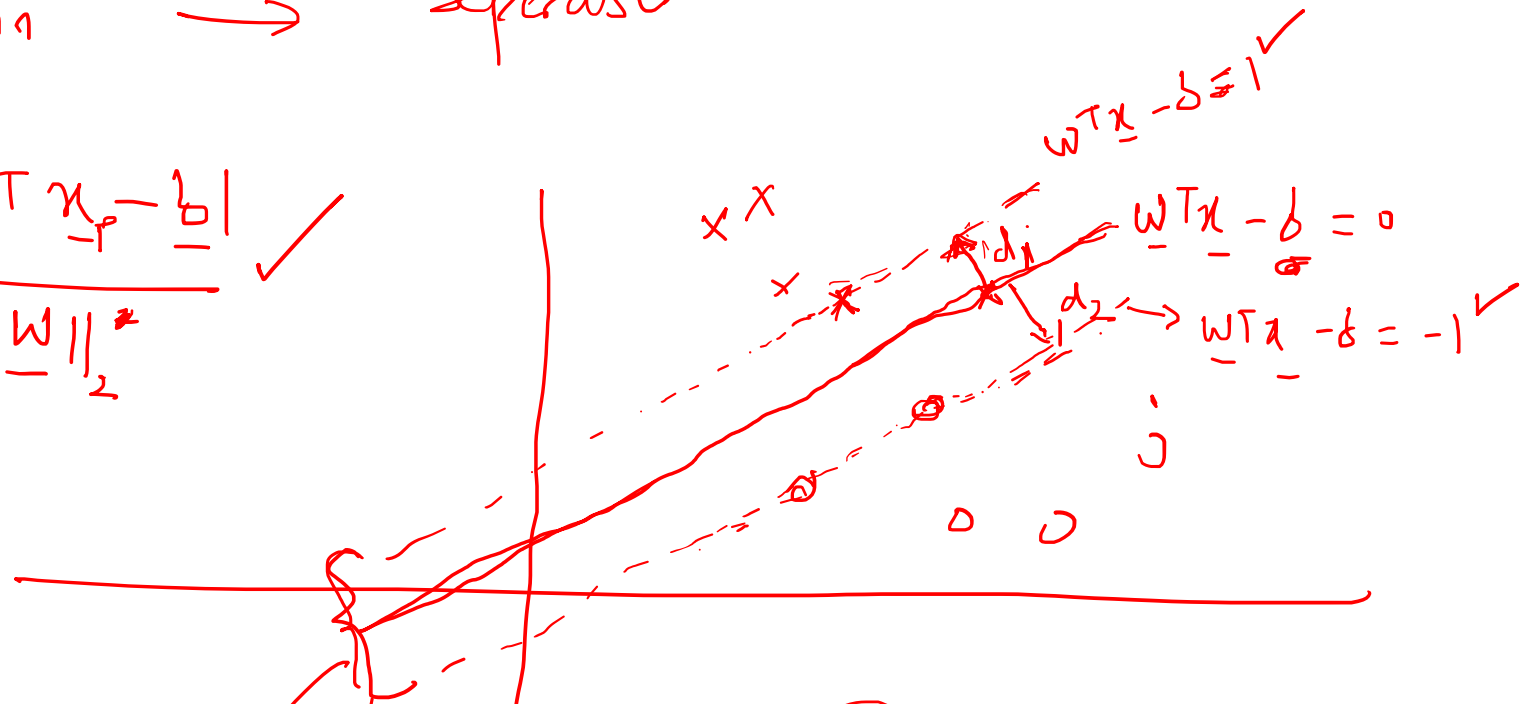
© 2021 Institute of Data

① Hard Margin → Seperable

$$d = \frac{|\underline{w}^T \underline{x}_p - \underline{b}|}{\|\underline{w}\|_2^2}$$

$$d_1 = \frac{1}{\|\underline{w}\|_2^2}$$

$$d_2 = \frac{|-1|}{\|\underline{w}\|} = \frac{1}{\|\underline{w}\|}$$



$$\frac{1}{\|\underline{w}\|} + \frac{1}{\|\underline{w}\|} = \frac{2}{\|\underline{w}\|}$$

© 2021 Institute of Data

$$\left\{ \begin{array}{l} \min \|\underline{W}\| \\ \underline{W}^T \underline{x} - \underline{b} = 0 \end{array} \right\}$$

$\longrightarrow$ Class -

$\underline{W}^T \underline{x} - b = 0 \longleftarrow$

$$\begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} = x_p$$

Testing Part

$$\left\{ \begin{array}{l} \underline{W}^T x_p - b \Leftarrow 0 \\ \underline{W}^T x_p - b > 0 \end{array} \right.$$

© 2021 Institute of Data

Linear SVM with Hard Margin

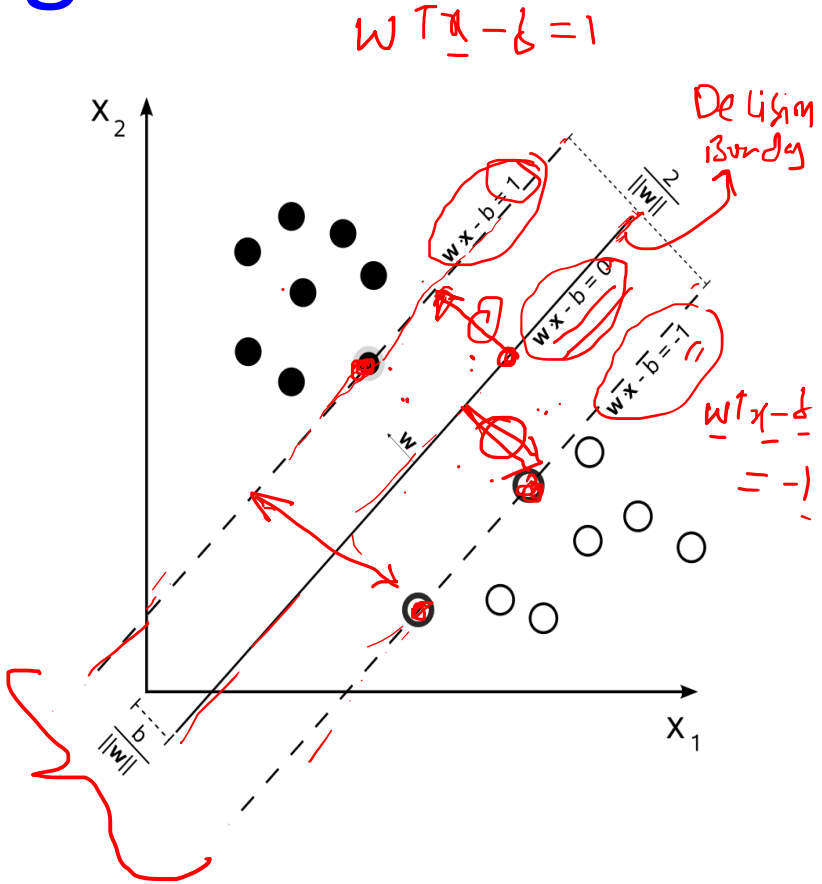
- equation of hyperplane is

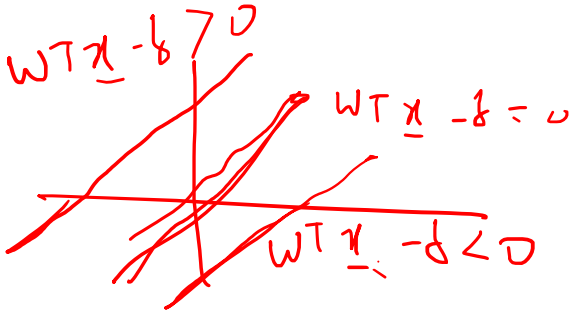
$$\vec{w} \cdot \vec{x} - b = 0$$

- compute $\vec{w}$, b so as to maximise width of margin

$$2/\|\vec{w}\|$$

Optimization of maximizing the margin ('street width') to reduce the number of weights that are nonzero to just a few that correspond to the support vectors





https://en.wikipedia.org/wiki/Distance_from_a_point_to_a_line

© 2021 Institute of Data

Linear SVM with Soft Margin

Classes → not Linearly seperable

- equation of hyperplane is

$$\vec{w} \cdot \vec{x} - b = 0$$

- compute $\vec{w}$, b so as to minimise

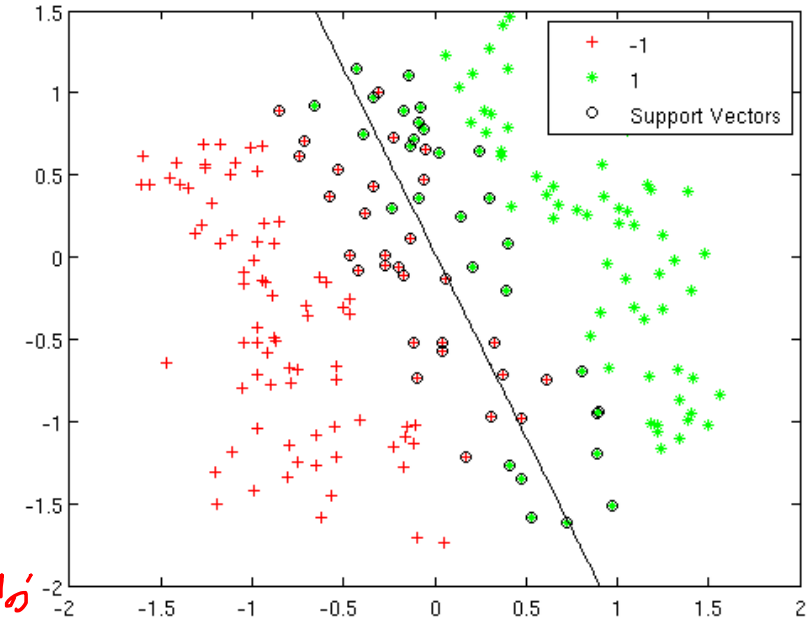
Cost function $J(\theta)$

$$= \left[\frac{1}{n} \sum_{i=1}^{n} \max\left(0, 1 - \boxed{y_i(\vec{w} \cdot \vec{x}_i - b)}\right) \right] + \lambda \|\vec{w}\|^2$$

"Hinge Loss"

Regularization

$y_i(w^T x_i - b)$ +ve



Nb. This plot also shows that only the nearest points are used to compute the boundary. These points are called the ***support vectors***.

© 2021 Institute of Data

Linear SVM with Soft Margin

$-(w^T x - b) < 0$

- equation of hyperplane is

$$\vec{w} \cdot \vec{x} - b = 0$$

- compute $\vec{w}$, b so as to minimise

$$J(x) = \left[\frac{1}{n}\sum_{i=1}^{n} \max\left(0, 1 - y_i(\vec{w} \cdot \vec{x}_i - b)\right)\right] + \lambda\|\vec{w}\|^2$$

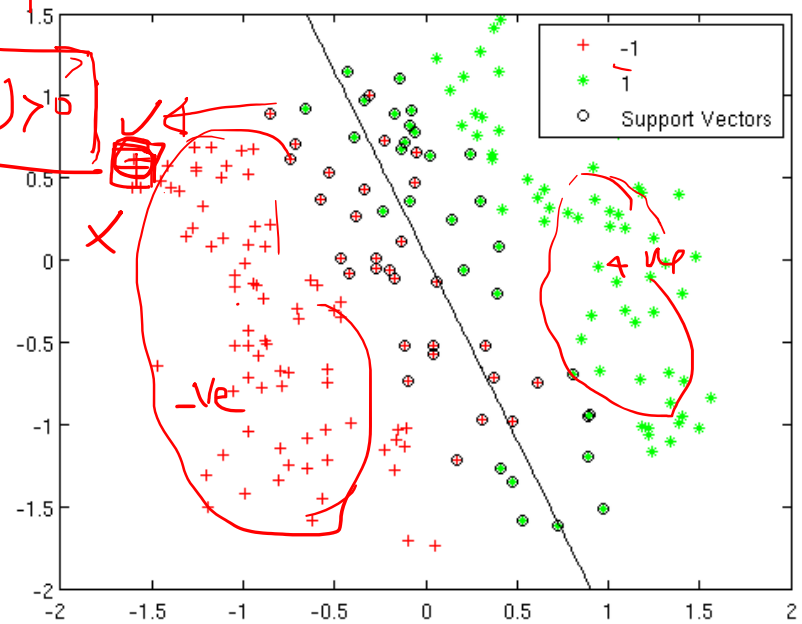
$\max(0, 1 - y_i(w^T x - b))$

$\max(0, 1 - (+ve)) = \max(0, -ve) = 0$

"Correct Classification"

$\boxed{y_i(w^T x - b) > 0}$

Single



Nb. This plot also shows that only the nearest points are used to compute the boundary. These points are called the ***support vectors***.

© 2021 Institute of Data

x_i	y_i Label
x x x x	+1

	y_i Label
o o o	-1

$y_i = +1$

$w^T x - b = 0$

$w^T x - b > 0$

+ve :

y

$w^T x - b > 0$

$\boxed{y_i(w^T x - b) > 0}$

$y_i = \{\pm 1\}$

$w^T x - b < 0$

$y_i \in \{+1, -1\}$

Conclusion

$w^T x - b < 0$

$\boxed{y_i(w^T x - b) > 0}$

$\boxed{y_i(w^T x - b) > 0}$

"Correct classification"

$y_i \to -ve$

© 2021 Institute of Data

© 2021 Institute of Data

$y = \max(0, x) \rightarrow$ ReLU $\rightarrow$

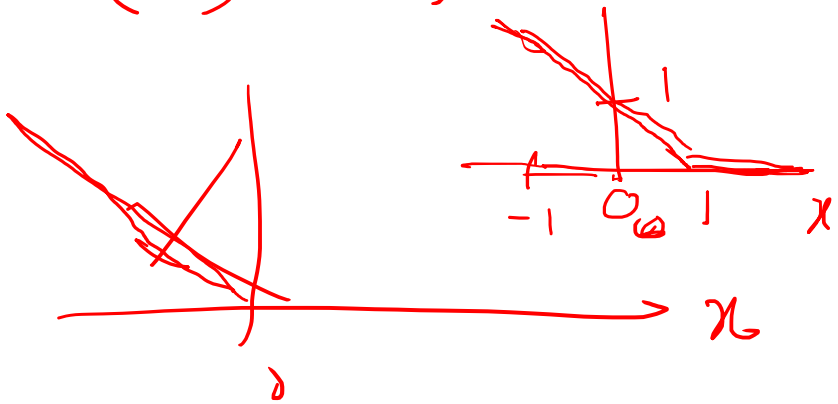
$y = \max(0, -1) = 0$

$y = \max(0, 0) = 0$

$y = \max(0, 1) = 1$

$y = \max(0, 1 - x)$

Hinge loss



© 2021 Institute of Data

Linear SVM with Soft Margin

$C = 1$

Hinge loss

Regularize

$$\min_{\mathbf{w}\in\mathbb{R}^d} \|\mathbf{w}\|^2 + C\sum_i^N \underbrace{\max(0, 1 - y_i f(\mathbf{x}_i))}_{\text{loss function}}$$

$(w^T x - b)$

$\max(0, 1 - \text{Nmbr}) = 0$

Points are in three categories:

$y_i(w^T x - b) > 1$

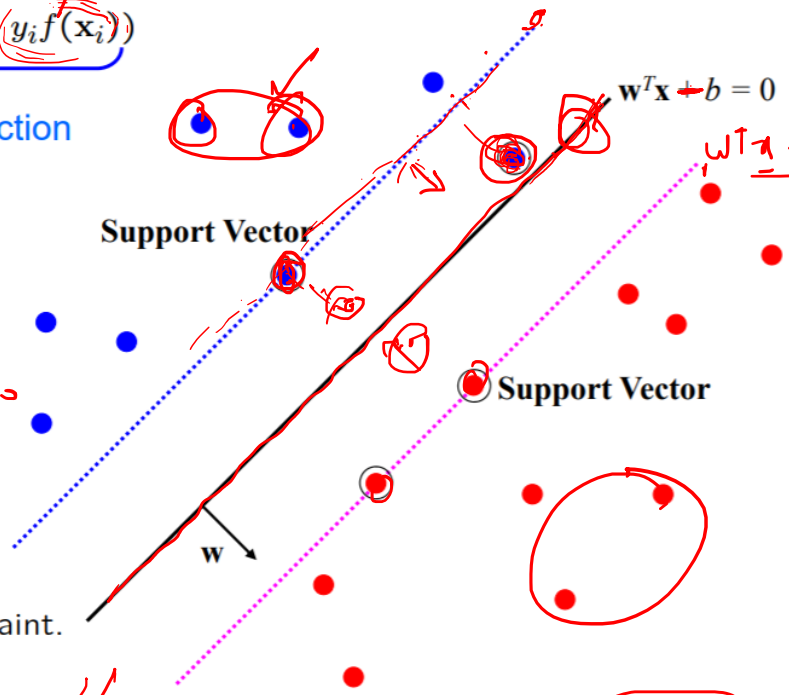
1. $y_i f(x_i) > 1$
   Point is outside margin.
   No contribution to loss

$y_i(w^T x - b) = 1$

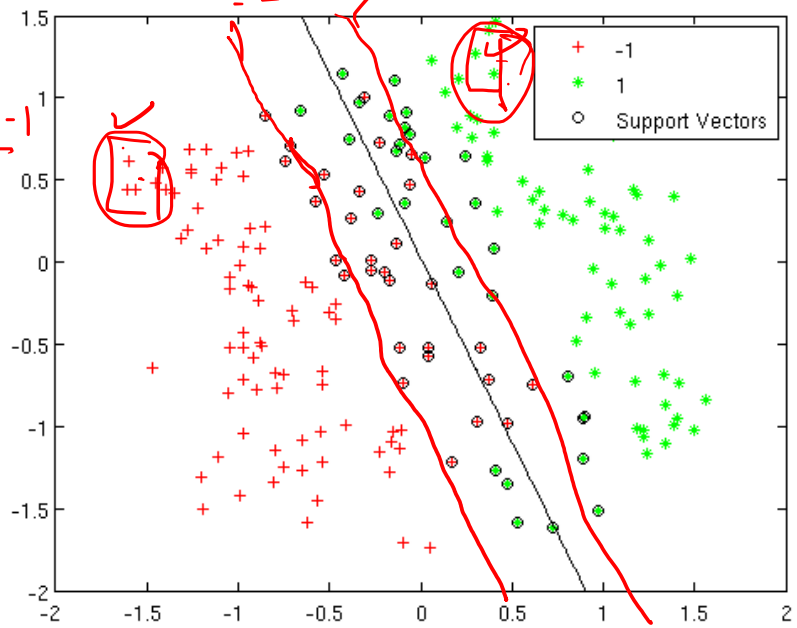
2. $y_i f(x_i) = 1$ $\max(0, 0) = 0$
   Point is on margin.
   No contribution to loss.
   As in hard margin case.

$y_i(w^T x - b) < 1$

3. $y_i f(x_i) < 1$
   Point violates margin constraint.
   Contributes to loss



$w^T x - b = 1$



$\max(0, 1 - 0.2) = \max(0, 0.8) = 0.8$

© 2021 Institute of Data

$$J(x) = \frac{1}{n}\sum \max(0, 1 - y_i(w^T x - b)) + \lambda \|w\|$$

$$\max(0, 1 - y_i(w^T x_i - b))$$

① : $\max(0, 1 - (+2))$

$= \max(0, -1) = 0$

② Support Vector

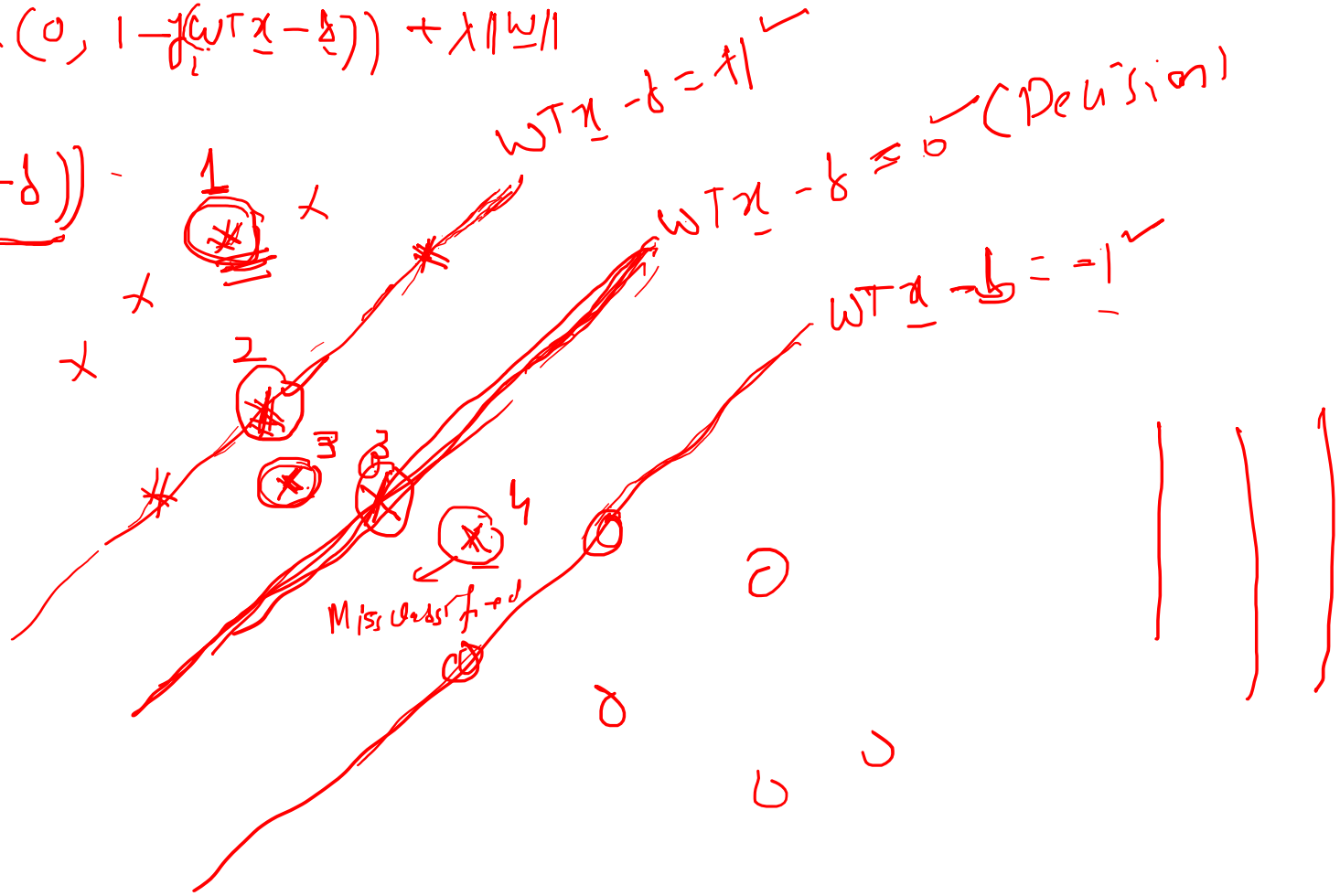
$\max(0, 1 - (1))$

$= \max(0, 0) = 0$

③ $\max(0, 1 - 1.5)$

$= \max(0, -0.5) = 0.5$

④ $\max(0, 1 - (-0.5)) = \max(0, 1.5) = 1.5$



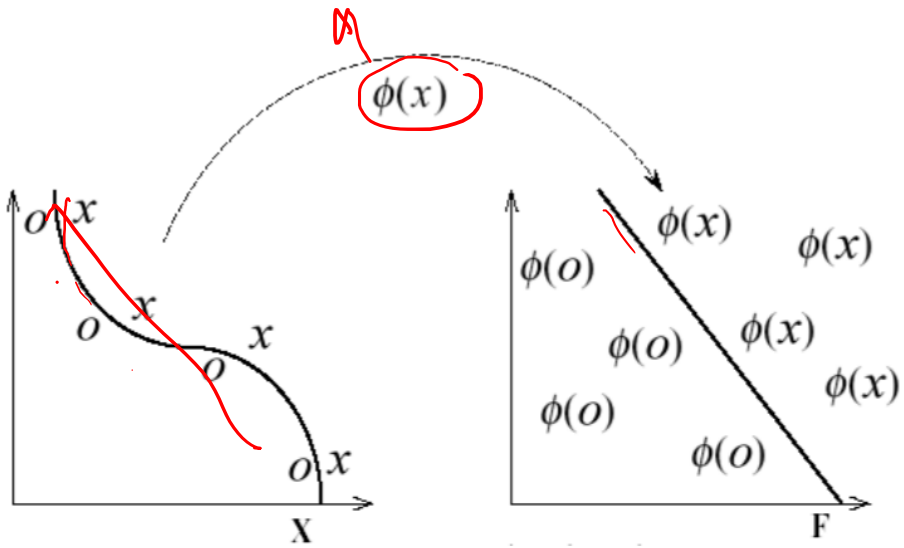
© 2021 Institute of Data

Nonlinear SVMs

The function of kernel is to take data as input and transform it into the required form

"Kernel"

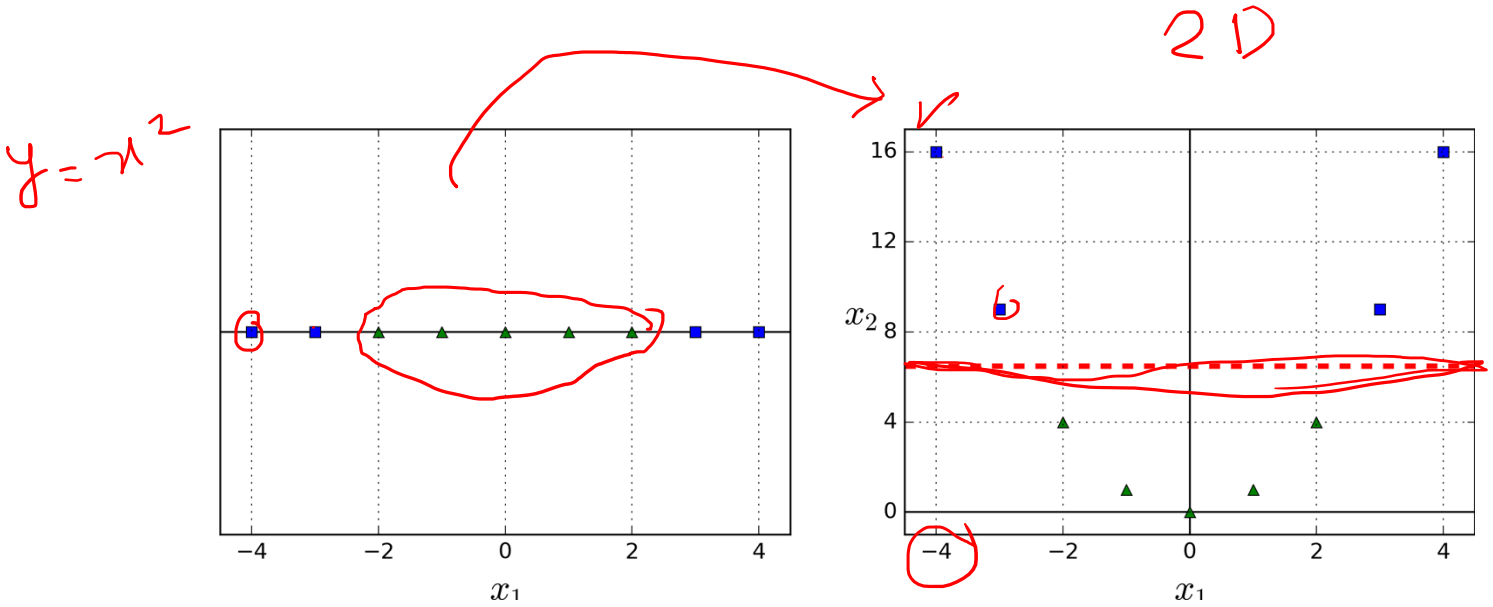
- Polynomial kernel x^2, x^3
  - with or without constant term
- Gaussian radial basis function (RBF) $e^{-\gamma x^2}$
- Hyperbolic tangent $\tanh(\gamma x)$



© 2021 Institute of Data

Nonlinear SVM

- One approach to handling nonlinear datasets is to add more features, such as polynomial features
- In some cases, this can result in a linearly separable dataset



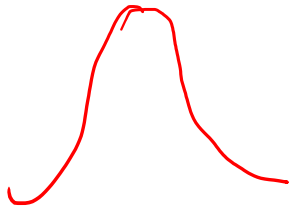
Add a second feature$x_2 = x_1^2$, the resulting 2D dataset is perfectly linearly separable.

© 2021 Institute of Data

$$\phi(x) = e^{-\gamma \|x - c\|}$$

$$x = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \xrightarrow{\phi(\cdot)} y = e^{-\gamma \|x - c\|}$$

Gaussian



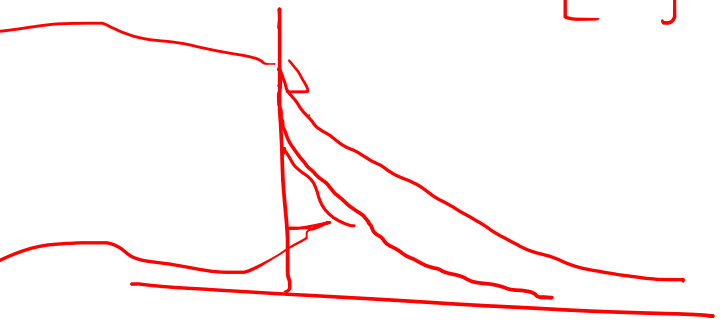
$$e^{-x^2/4\sigma}$$

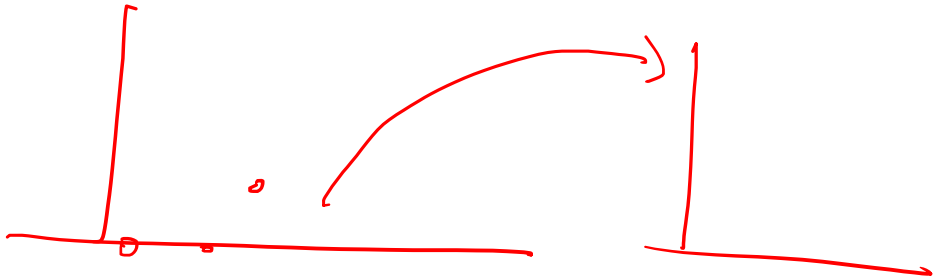
$$\frac{1}{\sqrt{2\pi}\,\sigma}\left[e^{-\frac{(x-\mu)^2}{2\sigma^2}}\right]$$

$$e^{-\frac{(x-2)^2}{2}}$$

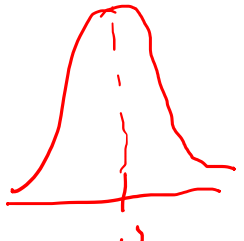
$$e^{-x}$$

$$e^{-3x}$$

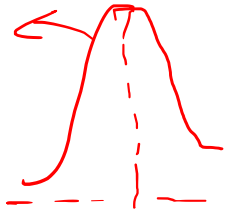




$$e^{-(x)^2/2}$$



$0 = \text{mean}$



$2 = \text{mean}$

© 2021 Institute of Data

① Parameters → weights
that we get after the
Training

② Hyper-Parameters (free-knobs)
↳ that are tunned according
the data
→ These are set before
running the algorithm

© 2021 Institute of Data

Nonlinear SVM

$y = e^{-\gamma\|x - c\|}$

$c = \begin{bmatrix} c_1 \\ c_2 \end{bmatrix} \quad \underline{x} = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$

- Radial Basis function (RBF)

→ Hyperplane → Support vector + Decision Boundary, Support vector -

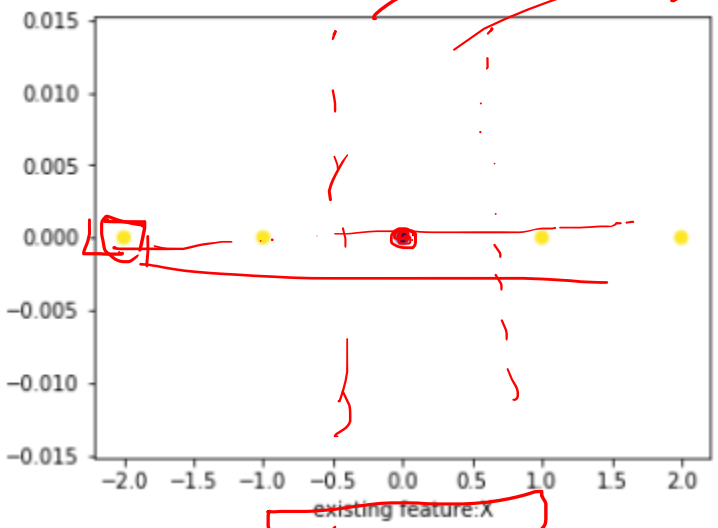
→ Hinge loss function

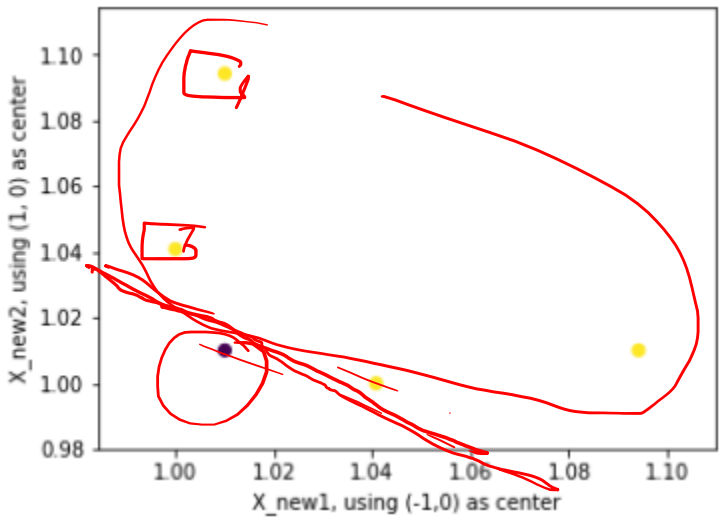
→ Kernel → "RBF"

$$\boxed{\phi(x, center) = exp(-\gamma\|x - center\|^2)}$$

$\|\underline{x} - \underline{c}\| = \sqrt{(x_1 - c_1)^2 + (x_2 - c_2)^2}$

$\phi(\cdot)$ → new





gamma = 0.1

Center = (-1,0)

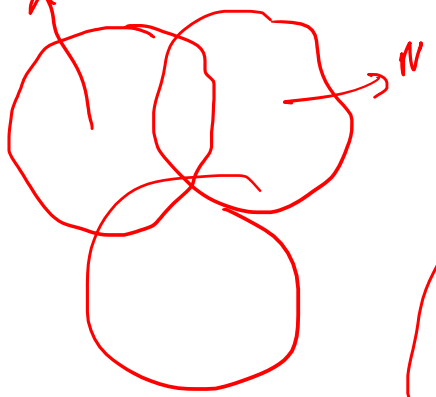
For (2,0),
Xnew = np.exp(np.power(-(0.1*(2+1)),2))
= 1.01

https://medium.com/bite-sized-machine-learning/support-vector-machine-explained-soft-margin-kernel-tricks-3728dfb92cee

© 2021 Institute of Data

ML

Hacking



① Theory Part

→ "loss function"

fit(C, r, k, .)

② Implementation → Sklean.

→ Optimization Algo → Complexity Hidde

→ Result

© 2021 Institute of Data

Limitations of SVMs

- Generalization error
  - margin is too small (data too close together) to accurately separate all classes
- Requires full labelling of input data
- Class membership probabilities are uncalibrated
- Does not natively handle more than 2 classes
  - multiclass problems must be reduced to a series of binary problems before applying SVMs
- Solution parameters are difficult to interpret

© 2021 Institute of Data

Applications of SVMs

- Text categorisation
- Image classification
- Handwriting interpretation
- Protein classification
- Customer segmentation
- Almost all the applications where ANN is used

"XGBoost"

© 2021 Institute of Data

SVMs in Python

- Predictors **X**
- Response **y**
- Test data **Xtest**

- c = **regularization** parameter: controls sensitivity to outliers
- gamma = kernel coefficient ('rbf', 'poly' and 'sigmoid' kernels): controls influence of nearby points

```
from sklearn import svm

# Create SVM classification object:
model = svm.svc(kernel = 'linear', c = 1, gamma = 1)

# Train model:
model.fit(X, y)

# Evaluate quality of fit:
model.score(X, y)

#Predict output for test data:
predicted = model.predict(Xtest)
```

https://scikit-learn.org/stable/modules/generated/sklearn.svm.SVC.html

https://www.youtube.com/watch?v=N1vOgolbjSc

© 2021 Institute of Data

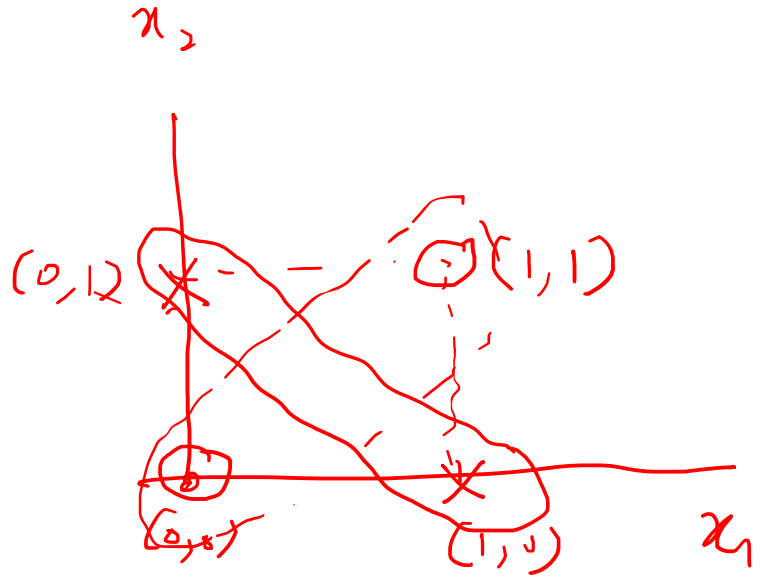
Ex: XOR

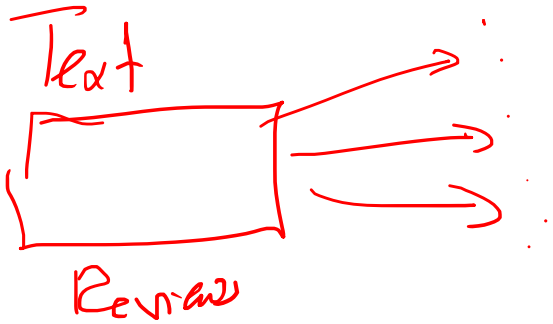
x_2	x_1	XOR
0	0	0 ✓
0	1	1
1	0	1
1	1	0 ✓

① Logistic Reg

② Perceptron

→ ③ SVM





© 2021 Institute of Data

Lab 5.3: Support Vector Machines

- Purpose:
  - To apply the SVM method to linear and nonlinear classification problems.

- Materials:
  - ‘Lab 5.3.ipynb’

https://www.kaggle.com/maxl11/svm-demo

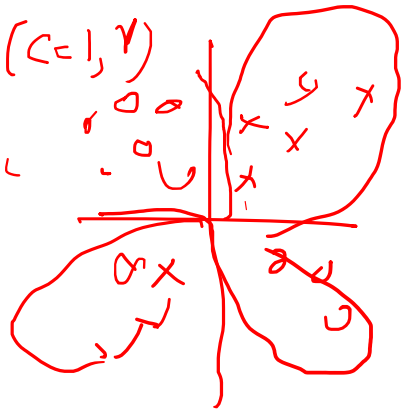
© 2021 Institute of Data

Discussion

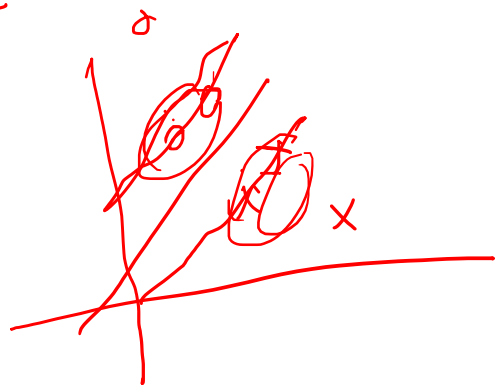
SVM → theory (SV)
→ Implementation (Tool) fit(C=1, γ)
→ optimization → Quadratic Programming ← $O(n^2)$

$O(n)$

- Strengths & weaknesses of SVMs



- Can be used as linear and non-linear classifier
- Effective in high dimensional spaces
- Effective in cases where the number of dimensions is greater than the number of samples.
- Memory efficient (uses support vectors only for decision boundaries)
- Robust to outliers
- Can reduce the redundant information



"Memory efficient" ⇒

+

"not Computationally efficient"

$O(n^2)$

© 2021 Institute of Data

Discussion

- Doesn't perform well when we have large data set because the required training time is higher → $O(n^2)$
- Non-probabilistic
- Choosing a good kernel is not easy
- The SVM hyper parameters are Cost -C and gamma. It is not that easy to fine-tune these hyper-parameters

© 2021 Institute of Data

Institute of Data

- Logistic Reg
- Perceptron
- SVM
- → Bayesian Classifier

Bayesian Inference

Approach

- **Frequentist** vs **Bayesian** probability
- **Bayes' theorem**
- Example: Disease detection
- **Bayesian modelling**
- **Naïve Bayes Classification**

0.3

P[Rain | Heavy Clouds]

"Bayes's Theorem":

→ Probability theorem

© 2021 Institute of Data

Frequentist Probability

SG

48%, 52%

P[F] = 0.48, P[M] = .05

- Probability:
  - a proportion of outcomes

- In the frequentist approach, we have some **sample results** from which we calculate the **frequency** of the *positive* result
  - we estimate the future probability of a positive result based entirely on this sample

'Axioms'

1. $P[S] = 1$
2. $0 \leq P[A] \leq 1$ (Cannot be -ve, $\not> 1$)
3. $P[A \cup B] = P[A] + P[B]$, $P[A \cap B] = \$$

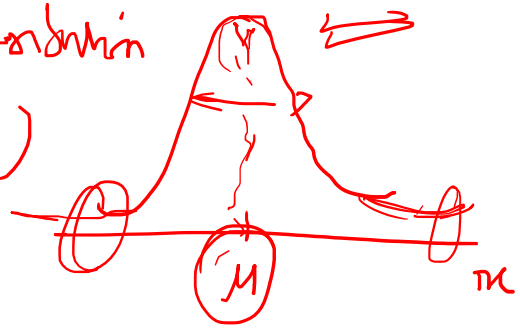
© 2021 Institute of Data

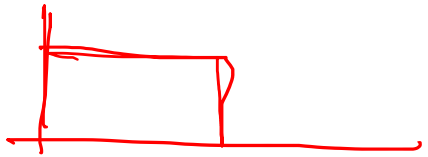
Bayesian Probability

CLT

Normal distribution

$N(\mu, \sigma^2)$





"Prior distribution"

- Probability:
  - "a degree of belief"
  - "probability is orderly opinion, and that inference from data is nothing other than the revision of such opinion in the light of relevant new information."

- In the Bayesian approach, we also have pre-existing information (a sample or distribution) about something that has a causal connection to the thing we are sampling
  - we use this "**prior** distribution" in addition to the current sample to estimate the future probability of a positive result

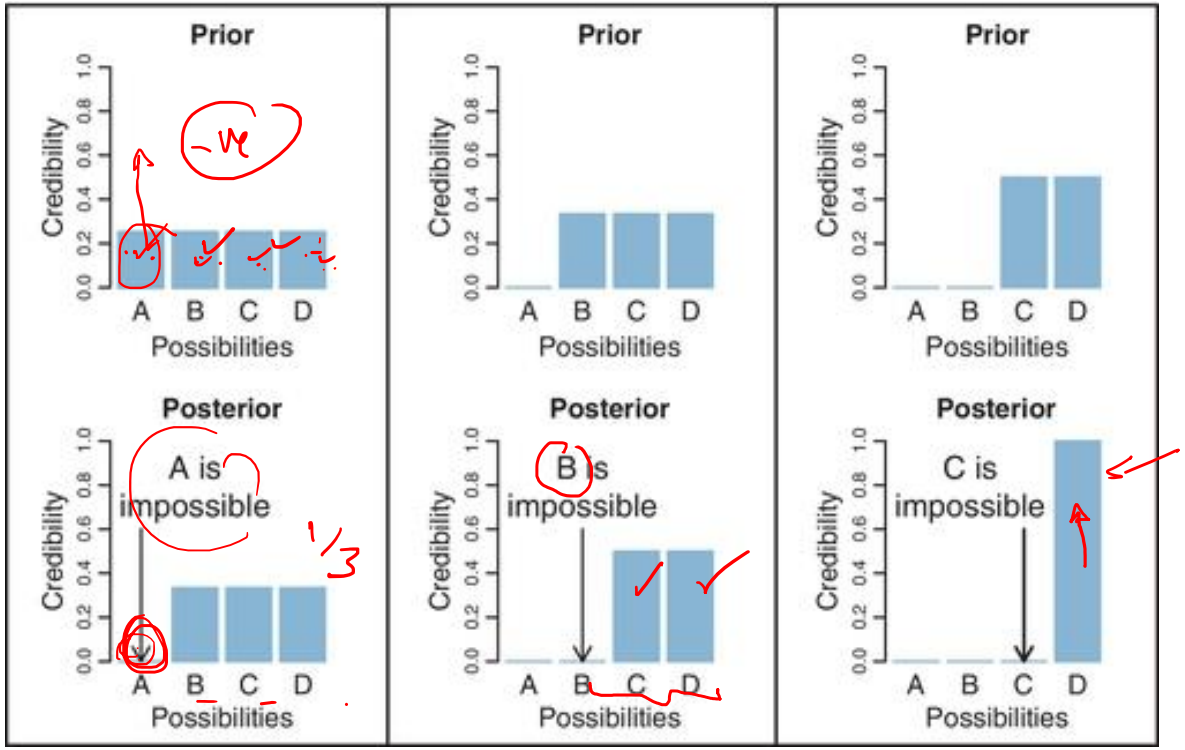
© 2021 Institute of Data

Bayesian Probability

+ve → "Could" "Belief"

$\frac{1}{4} = 0.25$

- There are four possible causes (A,B,C,D) of the outcome to be explained.
- We make new observations that rule out candidate cause A, B, C in steps.
- For excluding each cause, we re-allocate the creditability (Probability), called Posterior.
-



Causes, labeled A, B, C, and D, are mutually exclusive

© 2021 Institute of Data

Bayes' Theorem – cont'd

- suppose event A leads to event B, or that B is seen to be associated with A
- let

"Conditional Prob"

$$\boxed{p(B|A)}$$

"occurred"

$P[+ve \mid A = -ve]$

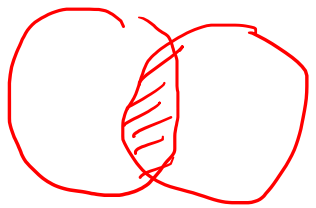
be the probability of B given A

- then

Bayes's

$$\boxed{p(A|B) = \frac{p(A)\,p(B|A)}{p(B)}}$$

$[0,1]$



$$\boxed{P[A|B] = \frac{P[A \cap B]}{P[B]}}$$

is the probability of A given B

> *the likelihood that A must have happened given that B happened*

© 2021 Institute of Data

$P[A]$, $P[A \cap B]$

$$P[A \mid B] = \frac{P[A \cap B]}{P[B]} \rightarrow (1)$$

Email < SPAM, HAM

"Account" → $P[A]$

"Transaction" → $P[B]$

$$P[A] = \frac{P[A \cap B]}{P[B]} \Rightarrow P[A \cap B] = P[A \mid B] \cdot P[B]$$

$$P[A \cap B] = P[A] \cdot P[B] \Rightarrow (1)$$

Independent

$$P[A \mid B] = P[A]$$

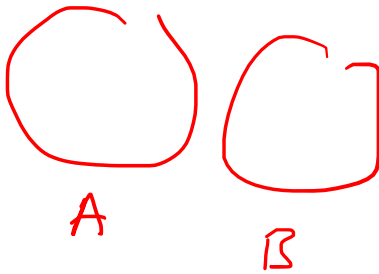
$$P[A \cap B] = 0.2 \times 0.2$$

"Joint"

"Independence"

A is Independent of B

$$P[A \mid B] = P[A]$$



Naive Bayes

↓ "Independence"

© 2021 Institute of Data

① $P[A|B] = \dfrac{P[A \cap B]}{P[B]} \rightarrow ①$

$P[A] = P[A|B]$

with Independence $P[A \cap B] = P[A] \cdot P[B] \rightarrow$

② $P[B|A] = \dfrac{P[A \cap B]}{P[A]}$

Bayes's

Bayes's $\rightarrow$

$$\boxed{P[B|A] = \frac{P[A|B] \cdot P[B]}{P[A]}}$$

A = Rain in SG

B = Cloud in SG

① P[may true | Colleges have infection]

② P[I get infection | ~~Any~~ infected]

M1 $\rightarrow$ F $\rightarrow$ G

M2 $\rightarrow$ F $\rightarrow$ G

fally

© 2021 Institute of Data

Example : Type of Call — Voice (V), Data (D)

Duration of Call — Long (L), Brief (D)

	V	D
L	0.4	0.3
B	0.2	0.1

AND

$P[B \cap V]$

Joint

$P[L \cap V] = P[\text{Long Duration Voice Call}]$

$P[L] = P[LV] + P[LD] = 0.4 + 0.3$
$= 0.7$

$P[B|V] = \dfrac{P[B \cap V]}{P[V]} = \dfrac{0.2}{0.4 + 0.2} = \dfrac{0.2}{0.6} = 1/3$

Condition

© 2021 Institute of Data

Bayes' Theorem – cont'd

$$p(A|B) = \frac{p(A)\,p(B|A)}{p(B)}$$

- $p(A), p(B)$ are ***marginal*** probabilities
- $p(A|B), p(B|A)$ are ***conditional*** probabilities
- $p(A)$ is the ***prior probability***
- $p(B)$ is the ***evidence***
- $\frac{p(B|A)}{p(B)}$ is the ***support*** that B provides for A

What if A is independent of B

- No support of B for A

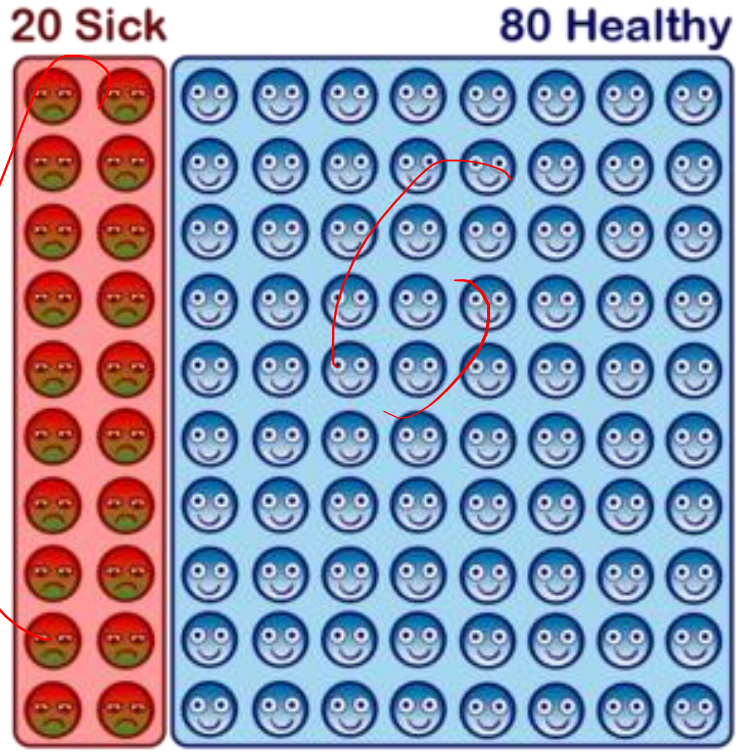
© 2021 Institute of Data

Example: Disease Detection → Bayes' Theorem

100

- prior knowledge:
  - 20% of population has the disease
    P / (P + N) = 0.20 ✓ $P[\text{Sick}^+] = \frac{20}{100}$
  - in a random sample of 100 people, 20 will have the disease (on average)

$\frac{20}{100}$ $\frac{80}{100}$ Actual



https://arbital.com/p/bayes_frequency_diagram/

© 2021 Institute of Data

Example: Disease Detection

- sensitivity of test:
  - 90% of subjects with disease are correctly detected

  TPR = 0.90 = TP / (TP + FN)

  TP = 0.90 * 20 = 18

$\frac{18}{18+2} = TPR$

Actual

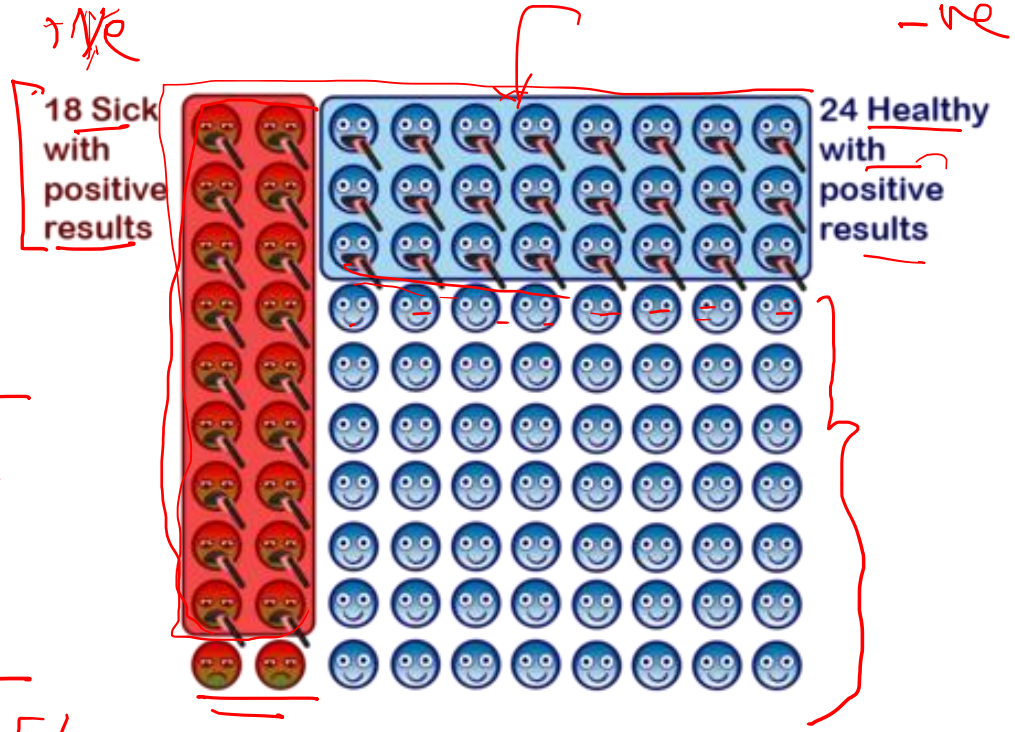
Prediction	+ve	-ve
+ve	18 TP	24 FP
-ve	2 FN	56 TN

- FPR (1 – specificity) of test:
  - 30% of disease-free subjects are erroneously detected

  FPR = 0.30 = FP / (FP + TN)

  FP = 0.30 * 80 = 24

$FPR = \frac{24}{24+56}$



> *if a person tests positive, what is the probability that they are sick?*

$P[\text{sick} \mid +ve] = \frac{18}{18+24} =$

© 2021 Institute of Data

Example: Disease Detection

- probability that subject with a positive test result is actually sick:

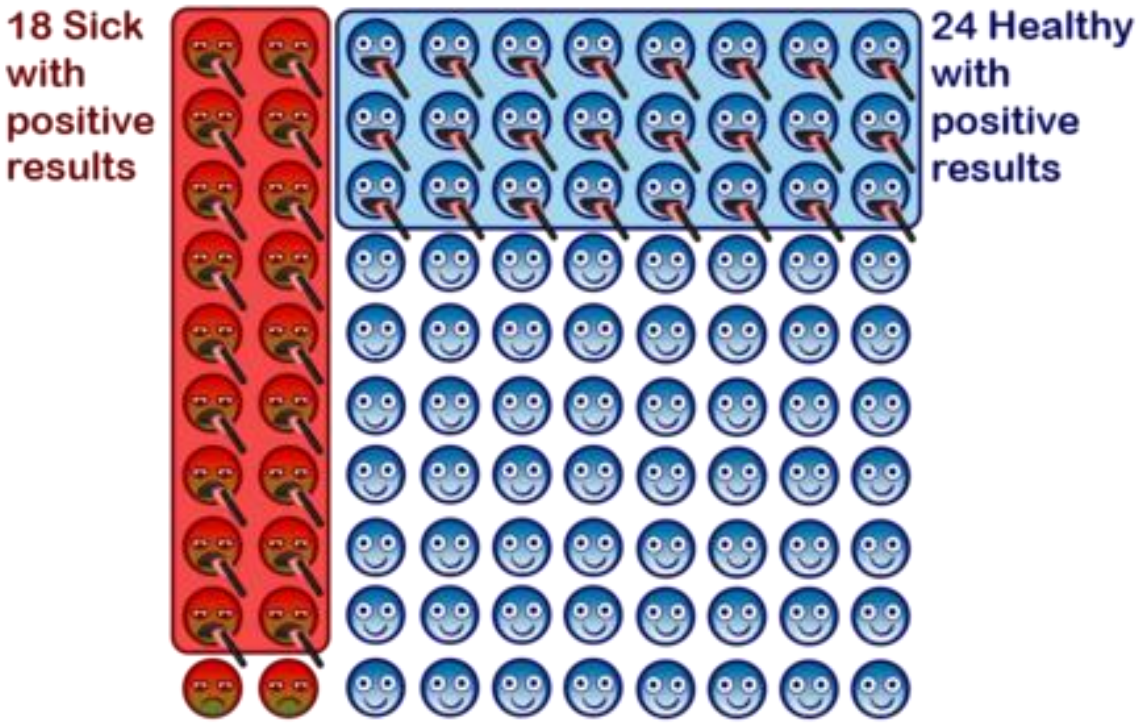
$$p(sick^{+} \mid test^{+}) = TP / (TP + FP)$$

$$= 18 / (18 + 24)$$

$$= 18 / 42$$

$$\approx 0.43$$

P[‘Sick’] gets updated from 0.2 to 0.43 after getting the test result



https://arbital.com/p/bayes_frequency_diagram/

© 2021 Institute of Data

Bayesian Modelling

- **Model-based** approach
  - results not dependent on arbitrary ***p*-value, confidence interval**

- **Naïve Bayes** assumption
  - **Predictors are independent**
    - variables come from distributions that do not interact

- prior distribution is not usually known explicitly
  - type of distribution must be assumed:
    - from domain knowledge
    - by choosing a mathematically sound distribution
  - parameters of distribution are fitted to the prior data

© 2021 Institute of Data

How to Choose the Prior Distribution?

- Informative, **empirical**:
  - Data from **related experiments** informs our prior beliefs
  - Prior beliefs will influence final predictions
- Informative, **non-empirical**:
  - Prefer certain values over others (e.g. need to **regularise coefficients**)
  - Prior beliefs will influence final predictions
- Informative, **domain-knowledge**:
  - No supporting data, but certain facts are known to be more true than others
  - Prior beliefs will influence final predictions
- **Non-informative**:
  - Prior beliefs will have little to no effect on our final assessment
  - Current data alone will determine final predictions

© 2021 Institute of Data

Beta Distribution

- a probability density function suitable for modelling the random behaviour of percentages and proportions

$$\text{PDF} = \frac{x^{\alpha-1}\,(1-x)^{\beta-1}}{B(\alpha,\beta)}$$

$\alpha, \beta > 0$

B is the beta function

https://homepage.divms.uiowa.edu/~mbognar/applets/beta.html

https://en.wikipedia.org/wiki/Beta_distribution

© 2021 Institute of Data

Bayesian Modelling – cont'd

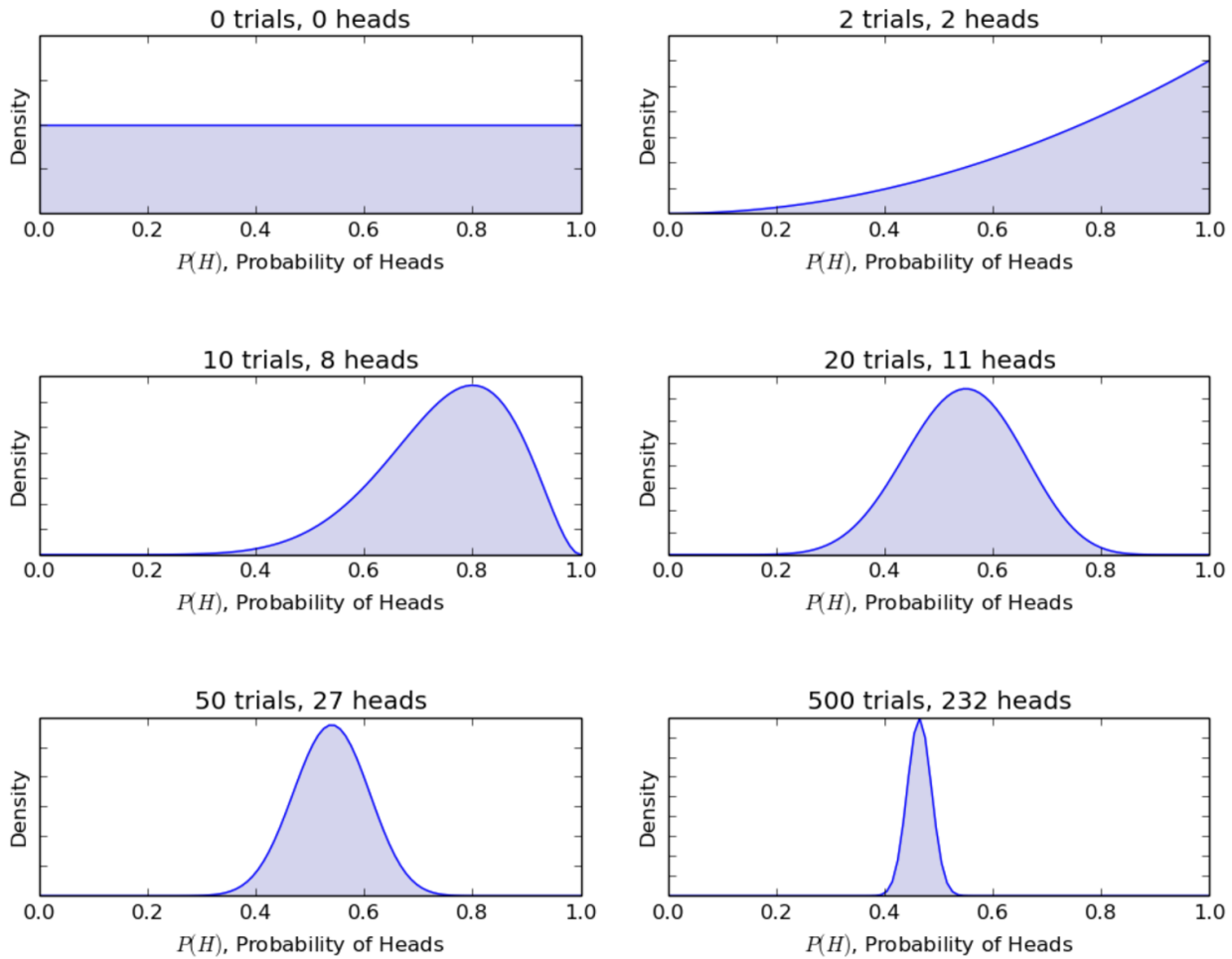
What can we use for the prior probability if we have no information?

> **a beta distribution**

$$\mu = \frac{\alpha}{\alpha + \beta}$$

$$\sigma = \sqrt{\frac{\alpha\beta}{(\alpha+\beta)^2(\alpha+\beta+1)}}$$

Prior = Beta(alpha, beta)
If we get m success, n failure
Posterior= Beta (alpha+m, beta+n)



© 2021 Institute of Data

Bayesian Modelling – Example

Data :

Weather	Play
Sunny	No
Overcast	Yes
Rainy	Yes
Sunny	Yes
Sunny	Yes
Overcast	Yes
Rainy	No
Rainy	No
Sunny	Yes
Rainy	Yes
Sunny	No
Overcast	Yes
Overcast	Yes
Rainy	No

Target

Frequency Table		
Weather	No	Yes
Overcast		4
Rainy	3	2
Sunny	2	3
Grand Total	5	9

Likelihood table				
Weather	No	Yes		
Overcast		4	=4/14	0.29
Rainy	3	2	=5/14	0.36
Sunny	2	3	=5/14	0.36
All	5	9		
	=5/14	=9/14		
	0.36	0.64		

P[overcast] = 0.29

Problem: Players will play if weather is sunny. Is this statement correct?

Yes

NO

We can solve it using the method of posterior probability.

P[Play | Sunny]

© 2021 Institute of Data

Bayesian Modelling – Example

Weather	Play
Sunny	No
Overcast	Yes
Rainy	Yes
Sunny	Yes
Sunny	Yes
Overcast	Yes
Rainy	No
Rainy	No
Sunny	Yes
Rainy	Yes
Sunny	No
Overcast	Yes
Overcast	Yes
Rainy	No

Frequency Table		
Weather	No	Yes
Overcast		4
Rainy	3	2
Sunny	2	3
Grand Total	5	9

Yes
No

Likelihood table				
Weather	No	Yes		
Overcast		4	=4/14	0.29
Rainy	3	2	=5/14	0.36
Sunny	2	3	=5/14	0.36
All	5	9		
	=5/14	=9/14		
	0.36	0.64		

3/9

Problem: Players will play if weather is sunny. Is this statement is correct?

$$P[B|A] = \frac{P[A|B] \cdot P[B]}{P[A]}$$

P(Yes | Sunny) = P(Sunny | Yes) * P(Yes) / P (Sunny)
Here we have P (Sunny |Yes) = 3/9 = 0.33, P(Sunny) = 5/14 = 0.36, P(Yes)= 9/14 = 0.64
Now, P (Yes | Sunny) = 0.33 * 0.64 / 0.36 = 0.60, which has higher probability.

→ Yes !! ✓

© 2021 Institute of Data

Naïve Bayes Classification

- Probabilistic classification methods

- Assumptions
  - predictors are independent (hence ‘naïve’)
  - predictors are normally distributed

- Applications (multiclass classification)
  - text classification (spam, topic)
  - medical diagnosis
  - fraud detection
  - insurance risk category

https://www.youtube.com/watch?v=Q8l0Vip5YUw

© 2021 Institute of Data

Naïve Bayes Classification – cont'd

① Independent

O(n)

Bernoulli Gaussian

- Advantages
  - Algorithms **scale linearly** with number of variables (fast to predict class on large data set)
  - perform well in multi class prediction
  - Uses marginal distributions of variables
  - perform well in case of categorical input variables compared to numerical variable(s)
  - When assumption of independence holds, a Naive Bayes classifier performs better compare to other models like logistic regression and you need less training data.
  - Complement Naïve Bayes is good if data is imbalance

- Disadvantages
  - **Correlated features** bias the model (assumption of **independent predictors**)
  - Absolute probabilities cannot be relied upon
    - only the **probability rankings** should be used
  - Assigns zero probability **("Zero Frequency")** if a new category appears in test data
    - training data must span all possible levels of categorical features

      or
    - apply a smoothing technique (Sklearn uses Laplace estimation)

© 2021 Institute of Data

Naïve Bayes Classification in Python

sklearn.naive_bayes

- GaussianNB
  - assumes normally distributed features

text classification

- BernoulliNB
  - binary/boolean features

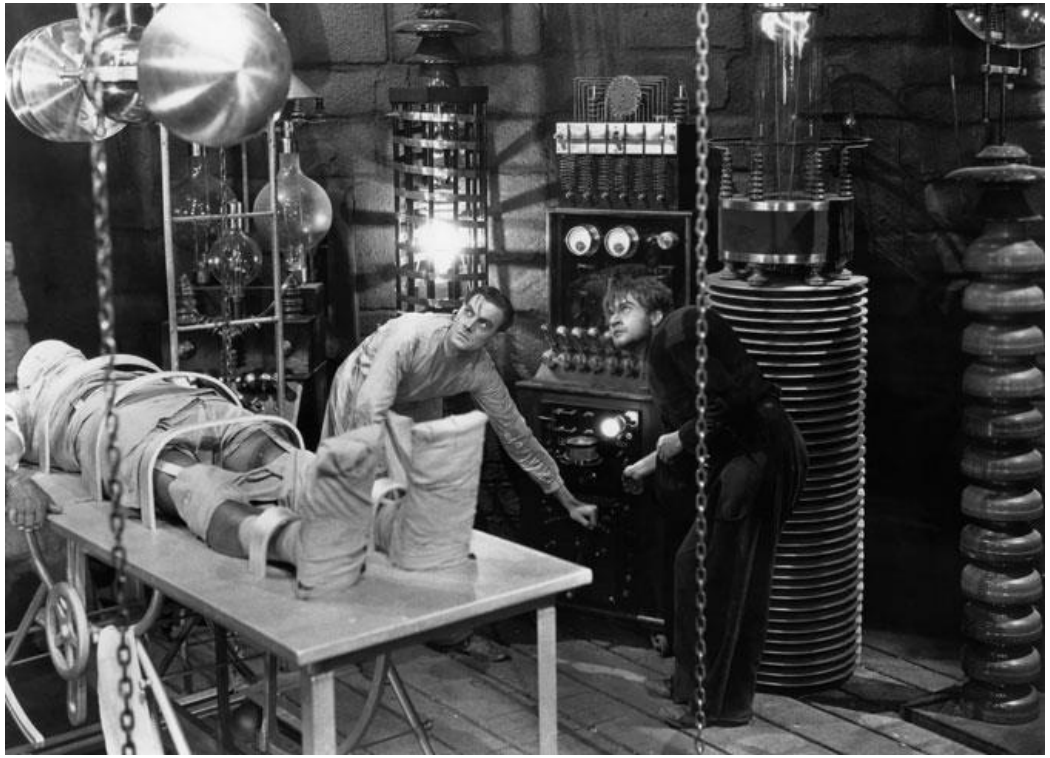
Yes
NO

- MultinomialNB
  - discrete features (multinomial distribution)
    - e.g. word counts for text classification
  - uses smoothing to account for features not present in the training set

© 2021 Institute of Data

Lab 5.4: Bayesian Inference

- Purpose:
  - To explore methods for Bayesian modelling and classification in Python
- Materials:
  - ‘Lab 5.4.ipynb’
- Resources:
  - census data:
    - https://archive.ics.uci.edu/ml/datasets/Adult



© 2021 Institute of Data

Discussion

- more on Bayesian inference:
  - https://www.analyticsvidhya.com/blog/2016/06/bayesian-statistics-beginners-simple-english/
  - https://towardsdatascience.com/bayesian-statistics-for-data-science-45397ec79c94

- to be covered in a future module:
  - decision trees (nonlinear classification methods)

© 2021 Institute of Data

Questions?

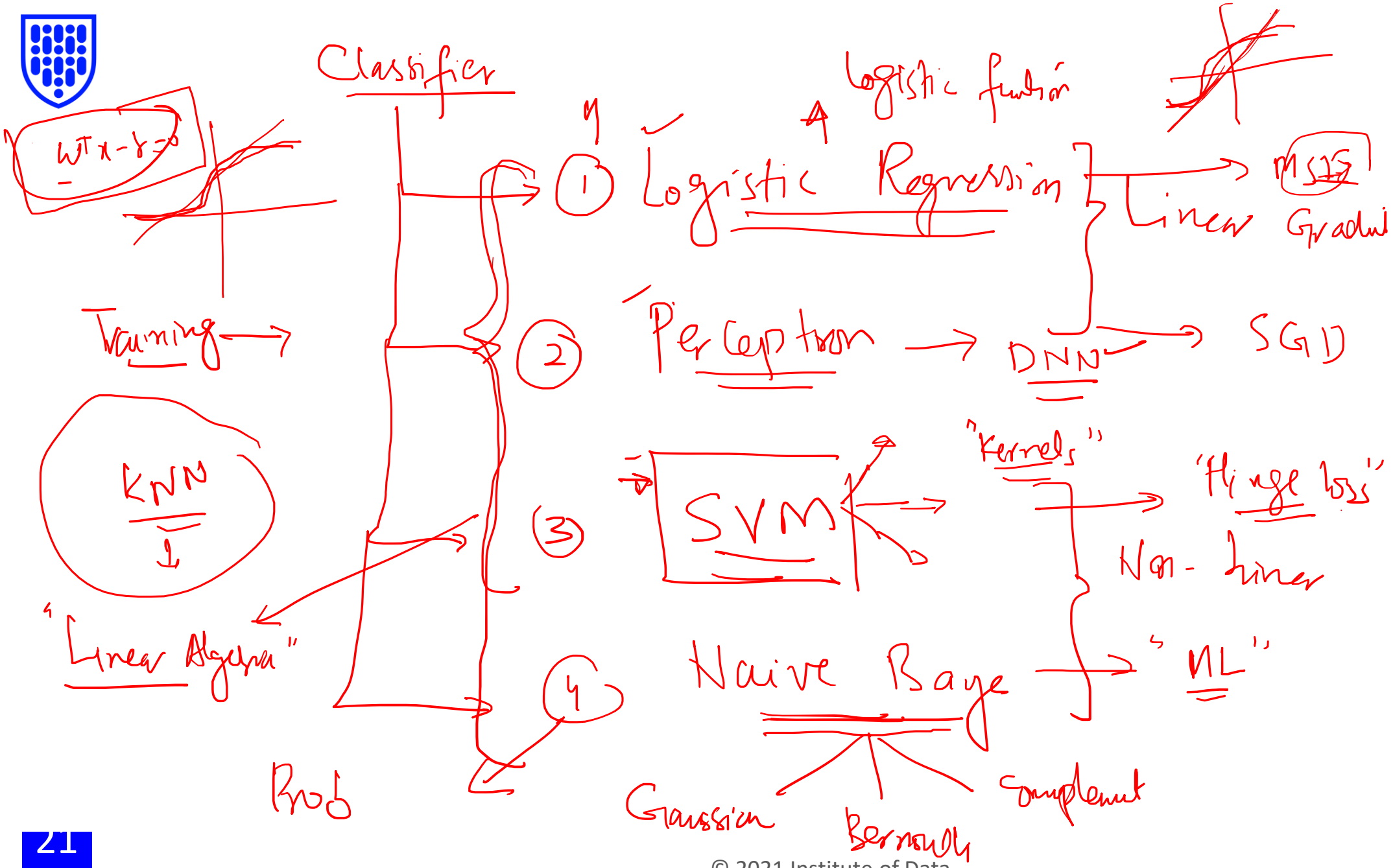
© 2021 Institute of Data

Appendices

© 2021 Institute of Data

End of Presentation!

© 2021 Institute of Data



© 2021 Institute of Data