Machine Learning

Lecture # 2
Data Normalization, KNN

SPLITTING OF TRAINING AND TEST DATA

Dividing Up Data

- We need independent data sets to train, set parameters, and test performance
- Thus we will often divide a data set into three
  - Training set
  - Parameter selection set
  - Test set
- These **must** be independent
- Data set 2 is not always necessary

Dataset

Inputs		Labels
15	95	1
33	90	1
78	70	0
70	45	0
80	18	0
35	65	1
45	70	1
31	61	1
50	63	1
98	80	0
73	81	0
50	18	0

Inputs		Labels
15	95	1
33	90	1
78	70	0
70	45	0
80	18	0
35	65	1
45	70	1
31	61	1
50	63	1
98	80	0
73	81	0
50	18	0

50:50 split

15	95	1
33	90	1
78	70	0
70	45	0
80	18	0
35	65	1

45	70	1
31	61	1
50	63	1
98	80	0
73	81	0
50	18	0

- Can be 70:30 or any other

Estimating the Generalisation Error

- We have a dilemma if we have limited data
  - We want to use as much data as possible for training
  - We need lots of data for estimating the generalisation error
- Obtaining a good estimate of generalisation performance is important for selecting the best parameter values

Cross Validation

- We can solve our dilemma by repeating the training many times on different partitioning
- This is known as K-fold cross validation

D_1	D_2	D_3	D_4	D_5	D_6	D_7	D_8	D_9	D_{10}	D_{11}	D_{12}	D_{13}	D_{14}	D_{15}	D_{16}	D_{17}	D_{18}	D_{19}	D_{20}

$$D = \{D_i\}_{i=1}^{P} \quad D_i = (x_i, y_i)$$

Cross Validation

D_1	D_2	D_3	D_4	D_5	D_6	D_7	D_8	D_9	D_{10}	D_{11}	D_{12}	D_{13}	D_{14}	D_{15}	D_{16}	D_{17}	D_{18}	D_{19}	D_{20}

Test Set

Training Set

5-fold cross-validation

$E_g = \quad 5.1$

Cross Validation

D_1	D_2	D_3	D_4	D_5	D_6	D_7	D_8	D_9	D_{10}	D_{11}	D_{12}	D_{13}	D_{14}	D_{15}	D_{16}	D_{17}	D_{18}	D_{19}	D_{20}

Training Set Test Set Training Set

5-fold cross-validation

$E_9 =$ 3.7

Cross Validation

D_1	D_2	D_3	D_4	D_5	D_6	D_7	D_8	D_9	D_{10}	D_{11}	D_{12}	D_{13}	D_{14}	D_{15}	D_{16}	D_{17}	D_{18}	D_{19}	D_{20}

Training Set Test Set Training Set

5-fold cross-validation

$E_g =$ 4.6

Cross Validation

D_1	D_2	D_3	D_4	D_5	D_6	D_7	D_8	D_9	D_{10}	D_{11}	D_{12}	D_{13}	D_{14}	D_{15}	D_{16}	D_{17}	D_{18}	D_{19}	D_{20}

Training Set Test Set Training Set

5-fold cross-validation

$E_9 =$ 4.6

Cross Validation

D_1	D_2	D_3	D_4	D_5	D_6	D_7	D_8	D_9	D_{10}	D_{11}	D_{12}	D_{13}	D_{14}	D_{15}	D_{16}	D_{17}	D_{18}	D_{19}	D_{20}

Training Set

Test Set

5-fold cross-validation

$E_g =$ 3.3

Cross Validation

D_1	D_2	D_3	D_4	D_5	D_6	D_7	D_8	D_9	D_{10}	D_{11}	D_{12}	D_{13}	D_{14}	D_{15}	D_{16}	D_{17}	D_{18}	D_{19}	D_{20}

$$\langle E_g \rangle = \frac{5.1 + 3.7 + 4.6 + 4.6 + 3.3}{5} = 4.3$$

Cross Validation

D_1	D_2	D_3	D_4	D_5	D_6	D_7	D_8	D_9	D_{10}	D_{11}	D_{12}	D_{13}	D_{14}	D_{15}	D_{16}	D_{17}	D_{18}	D_{19}	D_{20}

Test Set

Training Set

10-fold cross-validation

$$E_g = \quad 5.8$$

Cross Validation

D_1	D_2	D_3	D_4	D_5	D_6	D_7	D_8	D_9	D_{10}	D_{11}	D_{12}	D_{13}	D_{14}	D_{15}	D_{16}	D_{17}	D_{18}	D_{19}	D_{20}

Test Set　　　　Training Set

10-fold cross-validation

$E_g =$ 1.8

Cross Validation

D_1	D_2	D_3	D_4	D_5	D_6	D_7	D_8	D_9	D_{10}	D_{11}	D_{12}	D_{13}	D_{14}	D_{15}	D_{16}	D_{17}	D_{18}	D_{19}	D_{20}

Training Set Test Set Training Set

10-fold cross-validation

$E_9 =$ 4.8

Cross Validation

D_1	D_2	D_3	D_4	D_5	D_6	D_7	D_8	D_9	D_{10}	D_{11}	D_{12}	D_{13}	D_{14}	D_{15}	D_{16}	D_{17}	D_{18}	D_{19}	D_{20}

Training Set Test Set Training Set

10-fold cross-validation

$$E_9 = \quad 3.6$$

Cross Validation

D_1	D_2	D_3	D_4	D_5	D_6	D_7	D_8	D_9	D_{10}	D_{11}	D_{12}	D_{13}	D_{14}	D_{15}	D_{16}	D_{17}	D_{18}	D_{19}	D_{20}

Training Set   Test Set   Training Set

10-fold cross-validation

$E_g =$

7.4

Cross Validation

D_1	D_2	D_3	D_4	D_5	D_6	D_7	D_8	D_9	D_{10}	D_{11}	D_{12}	D_{13}	D_{14}	D_{15}	D_{16}	D_{17}	D_{18}	D_{19}	D_{20}

Training Set Test Set Training Set

10-fold cross-validation

$E_g =$ 0.99

Cross Validation

D_1	D_2	D_3	D_4	D_5	D_6	D_7	D_8	D_9	D_{10}	D_{11}	D_{12}	D_{13}	D_{14}	D_{15}	D_{16}	D_{17}	D_{18}	D_{19}	D_{20}

Training Set Test Set Training Set

10-fold cross-validation

$E_9 =$ 4.5

Cross Validation

D_1	D_2	D_3	D_4	D_5	D_6	D_7	D_8	D_9	D_{10}	D_{11}	D_{12}	D_{13}	D_{14}	D_{15}	D_{16}	D_{17}	D_{18}	D_{19}	D_{20}

Training Set　　　　Test Set　Training Set

10-fold cross-validation

$E_9 =$　　5.4

Cross Validation

D_1	D_2	D_3	D_4	D_5	D_6	D_7	D_8	D_9	D_{10}	D_{11}	D_{12}	D_{13}	D_{14}	D_{15}	D_{16}	D_{17}	D_{18}	D_{19}	D_{20}

Training Set

Test Set

10-fold cross-validation

$E_9 =$

6.2

Cross Validation

D_1	D_2	D_3	D_4	D_5	D_6	D_7	D_8	D_9	D_{10}	D_{11}	D_{12}	D_{13}	D_{14}	D_{15}	D_{16}	D_{17}	D_{18}	D_{19}	D_{20}

Training Set

Test Set

10-fold cross-validation

$E_9 =$

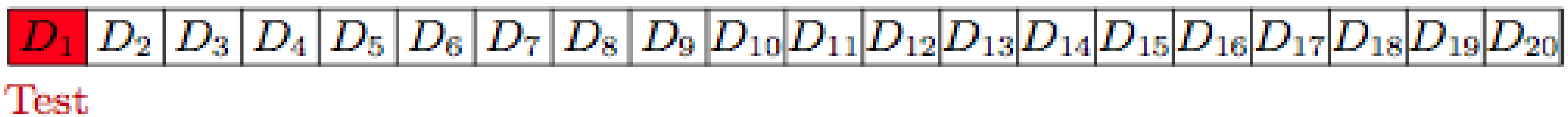
2.7

Cross Validation

D_1	D_2	D_3	D_4	D_5	D_6	D_7	D_8	D_9	D_{10}	D_{11}	D_{12}	D_{13}	D_{14}	D_{15}	D_{16}	D_{17}	D_{18}	D_{19}	D_{20}

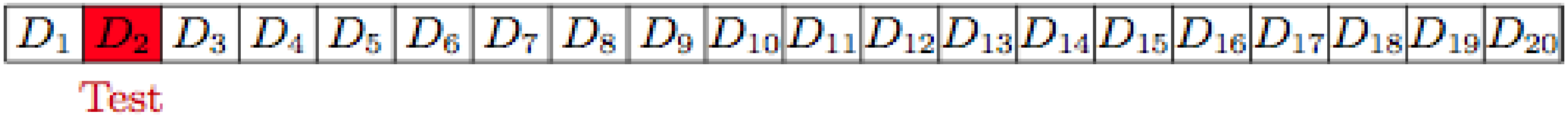
$$\langle E_g \rangle = \frac{5.8 + 1.8 + 4.8 + 3.6 + 7.4 + 0.99 + 4.5 + 5.4 + 6.2 + 2.7}{10} = 4.3$$

Cross Validation



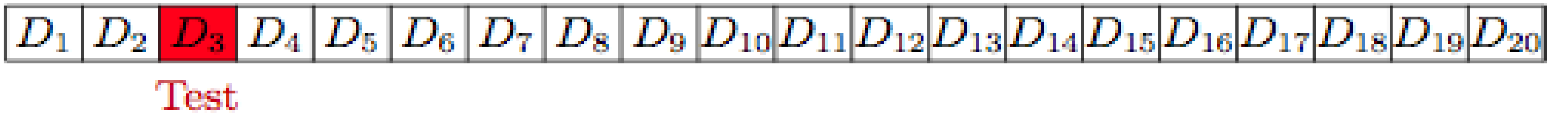
Leave-one-out cross-validation

Cross Validation



Leave-one-out cross-validation

Cross Validation



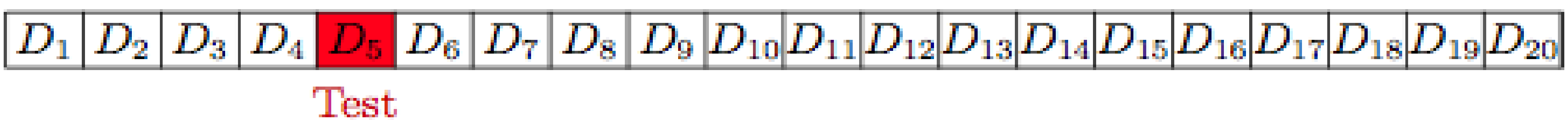
Leave-one-out cross-validation

Cross Validation

D_1	D_2	D_3	D_4	D_5	D_6	D_7	D_8	D_9	D_{10}	D_{11}	D_{12}	D_{13}	D_{14}	D_{15}	D_{16}	D_{17}	D_{18}	D_{19}	D_{20}
			Test																

Leave-one-out cross-validation

Cross Validation



Leave-one-out cross-validation

Cross Validation

D_1	D_2	D_3	D_4	D_5	D_6	D_7	D_8	D_9	D_{10}	D_{11}	D_{12}	D_{13}	D_{14}	D_{15}	D_{16}	D_{17}	D_{18}	D_{19}	D_{20}

$$\langle E_g \rangle = 3.9$$

- Leave-one-out cross-validation is extreme case

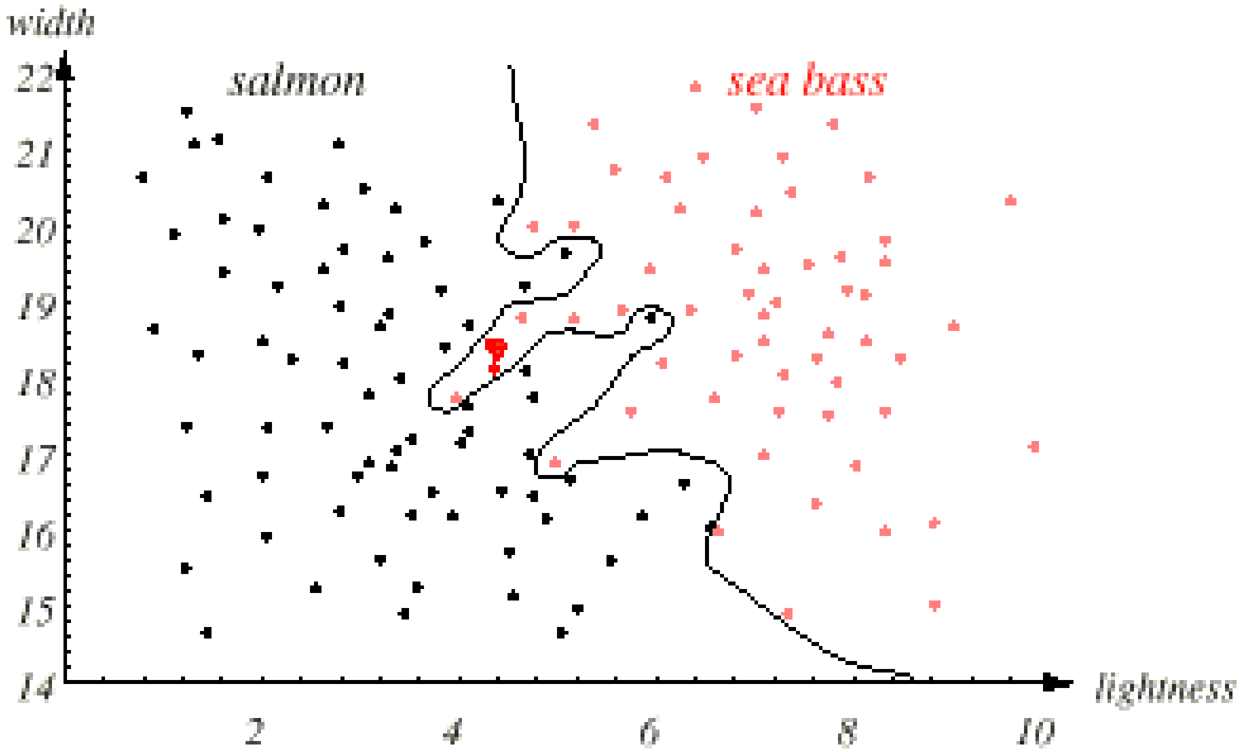
Price of Cross Validation

- Cross-validation is computationally expensive (K-fold cross-validation requires K times as much work)
- There are attempts at estimating generalisation error more cheaply (boot-strapping) methods, but these are not very accurate (https://www.mastersindatascience.org/learning/machine-learning-algorithms/bootstrapping/)
- Cross-validation is only necessary when you have little data

Generalization

- While classes can be specified by training samples with known labels, the goal of a recognition system is to recognize novel inputs
- When a recognition system is over-fitted to training samples, it may give bad performance for typical inputs

OverFitting



PERFORMANCE MEASUREMENTS

R.O.C. Analysis

False positives – i.e. falsely predicting an event
False negatives – i.e. missing an incoming event

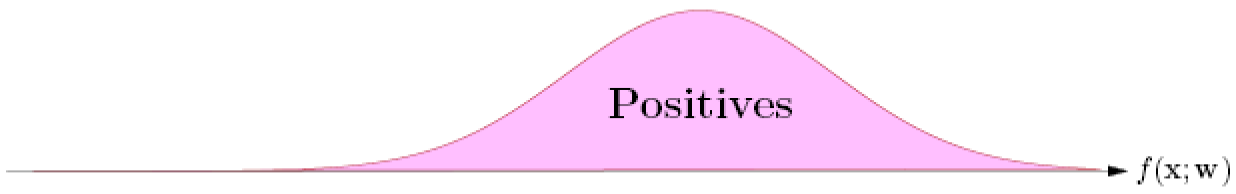
Similarly, we have “true positives” and “true negatives”

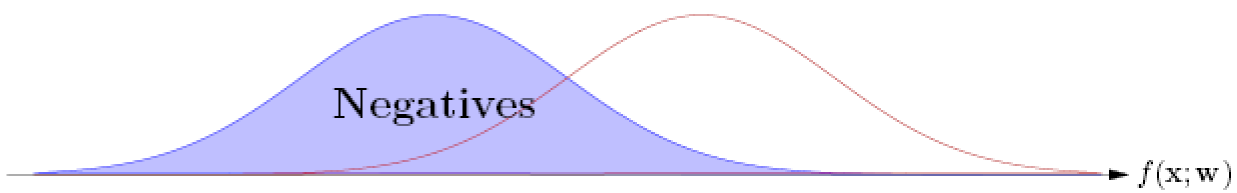
		Prediction	
		0	*1*
Truth	*0*	TN	**FP**
	1	**FN**	TP

Accuracy Measures

- Accuracy
  - = (TP+TN)/(P+N)
- Sensitivity or true positive rate (TPR)
  - = TP/(TP+FN) = TP/P
- Specificity or TNR
  - = TN/(FP+TN) = TN/N
- Positive Predictive value (Precision) (PPV)
  - = Tp/(Tp+Fp)
- Recall
  - = Tp/(Tp+Fn)

ROC Curve

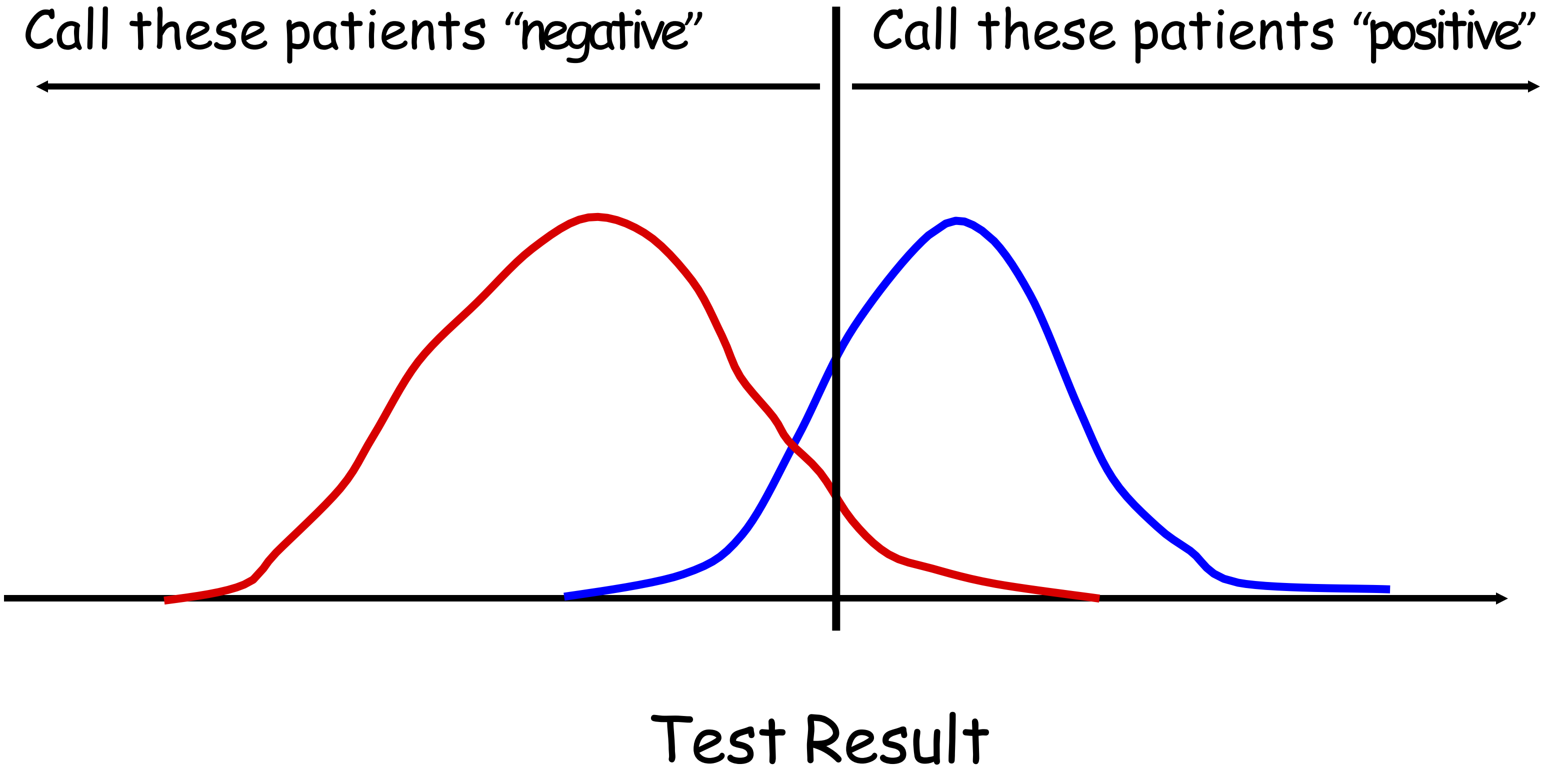



Negatives
$f(\mathbf{x}; \mathbf{w})$

Choosing the threshold

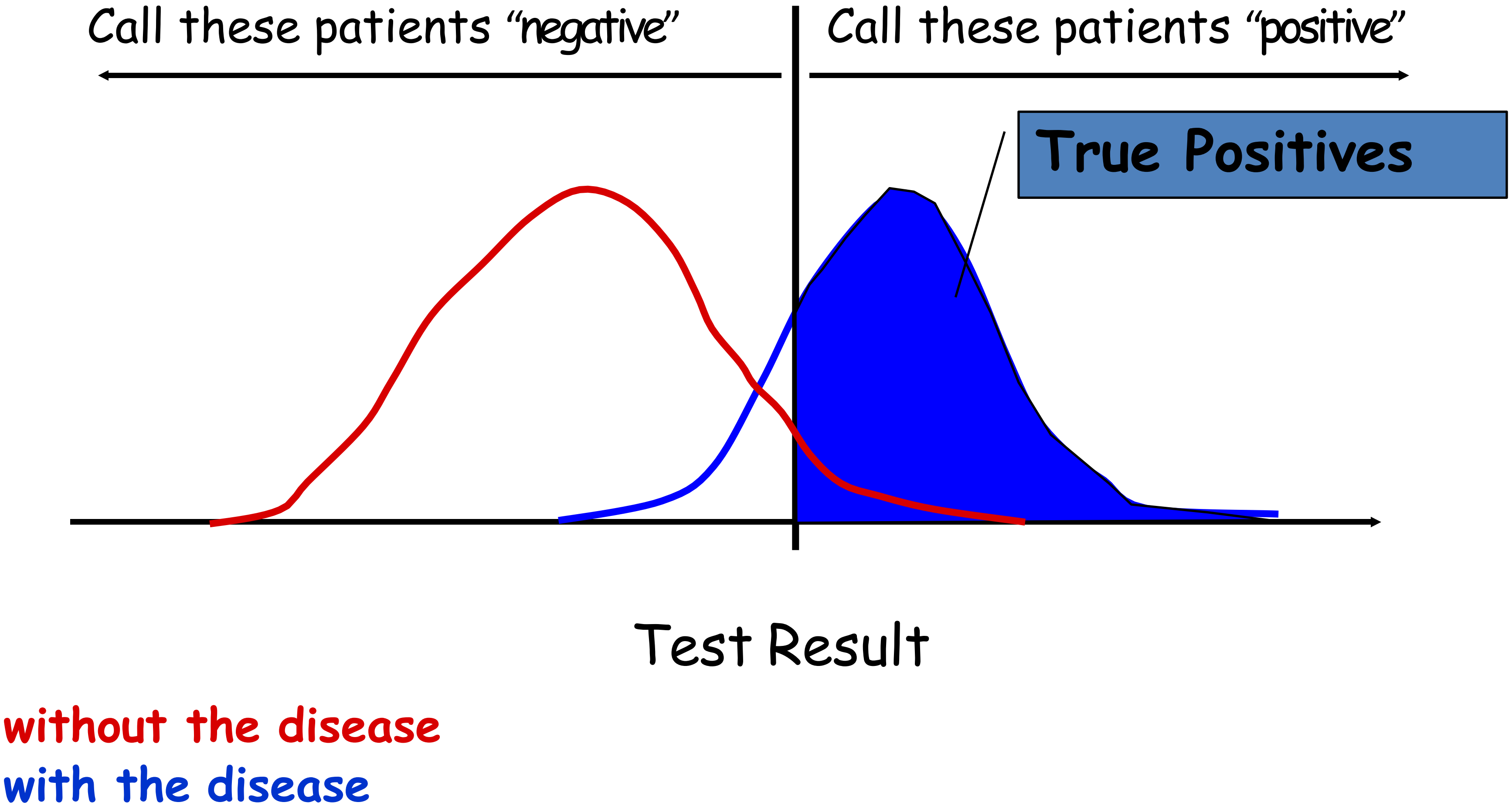
- Where should we set the threshold
- We could choose the **equal error rate** point where the errors in positive set equals the errors in the negative set
- Want to see all the options
- The receiver operating characteristic (ROC) curve is a standard way to test this

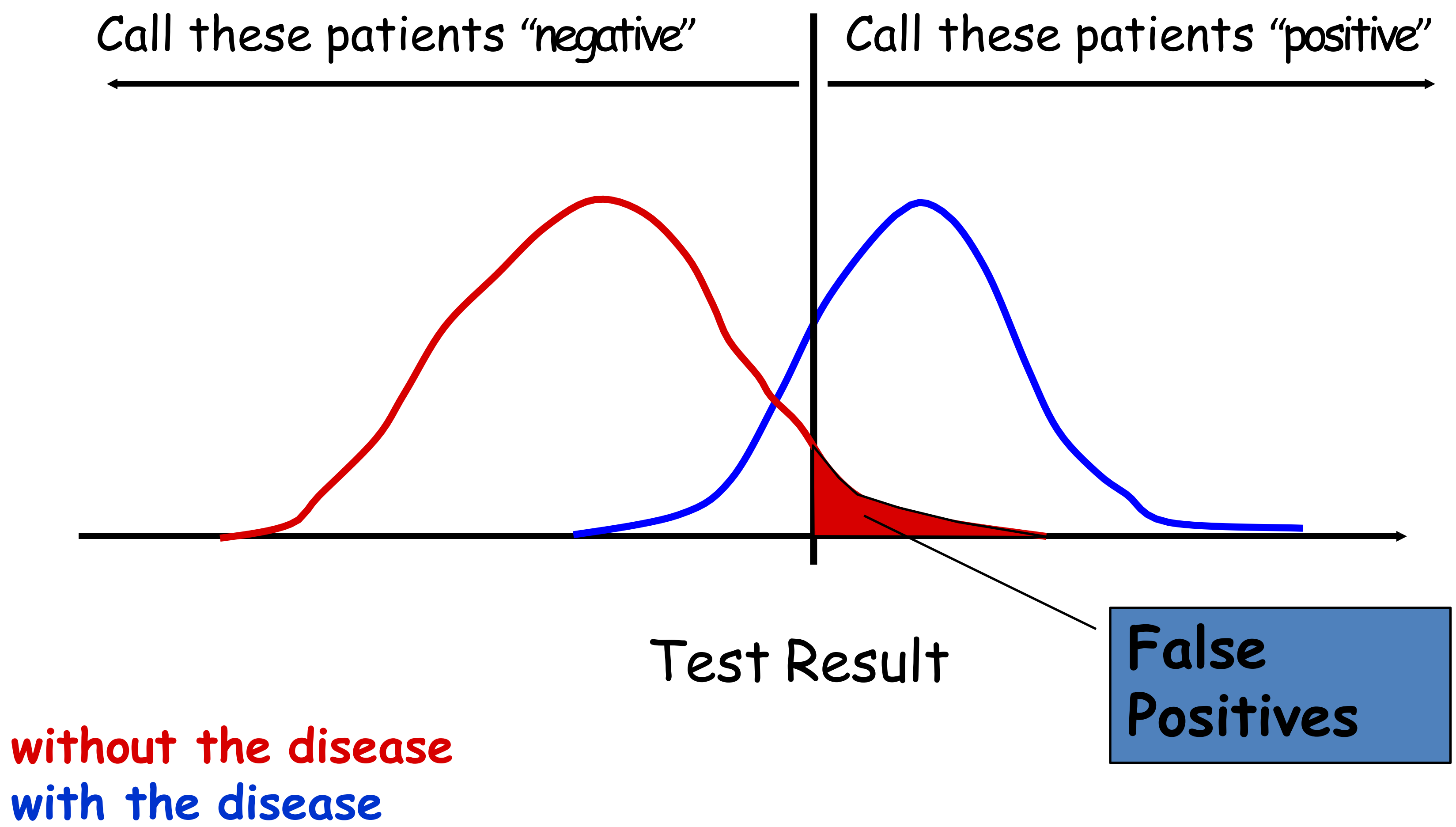
Threshold

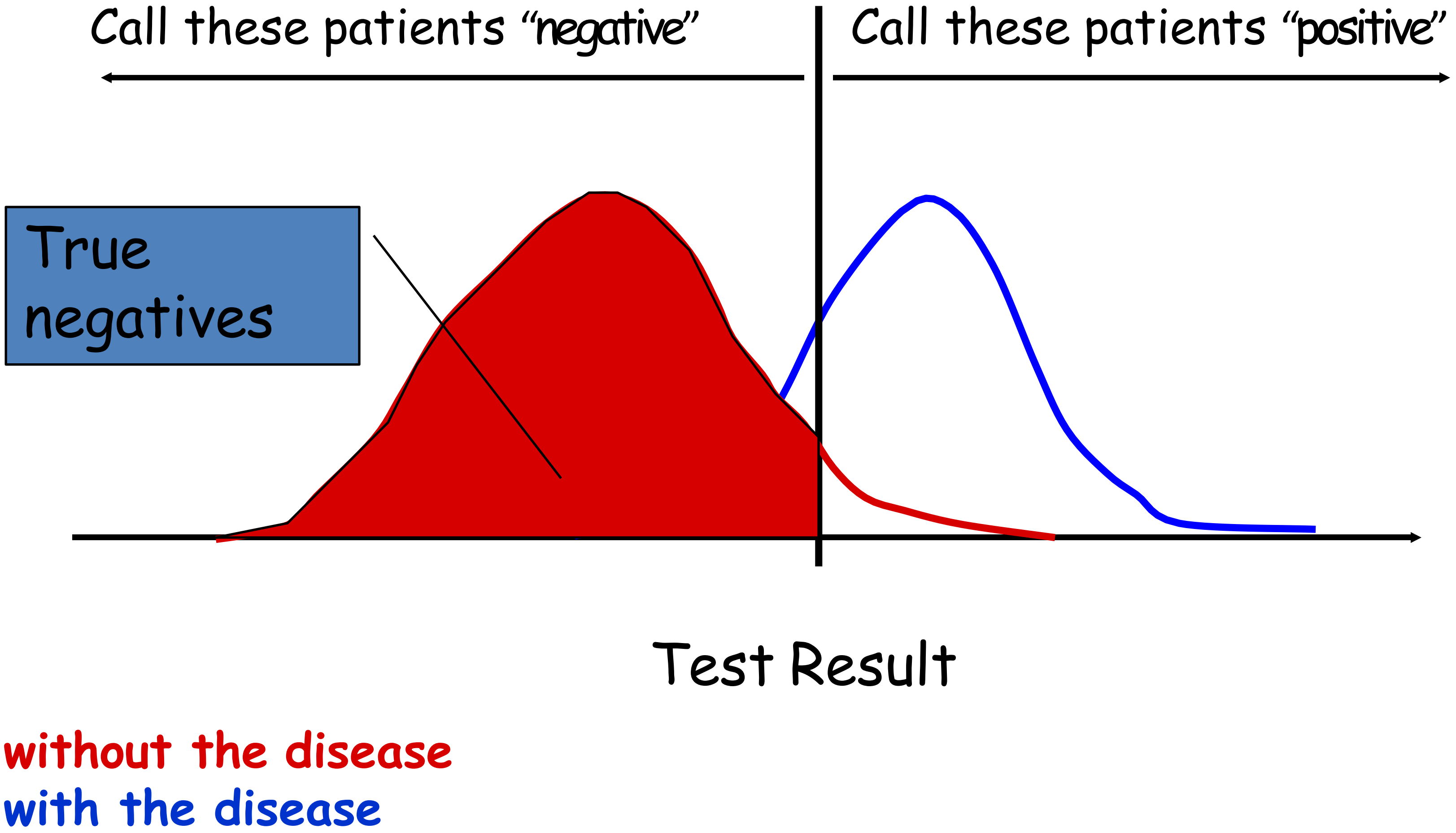


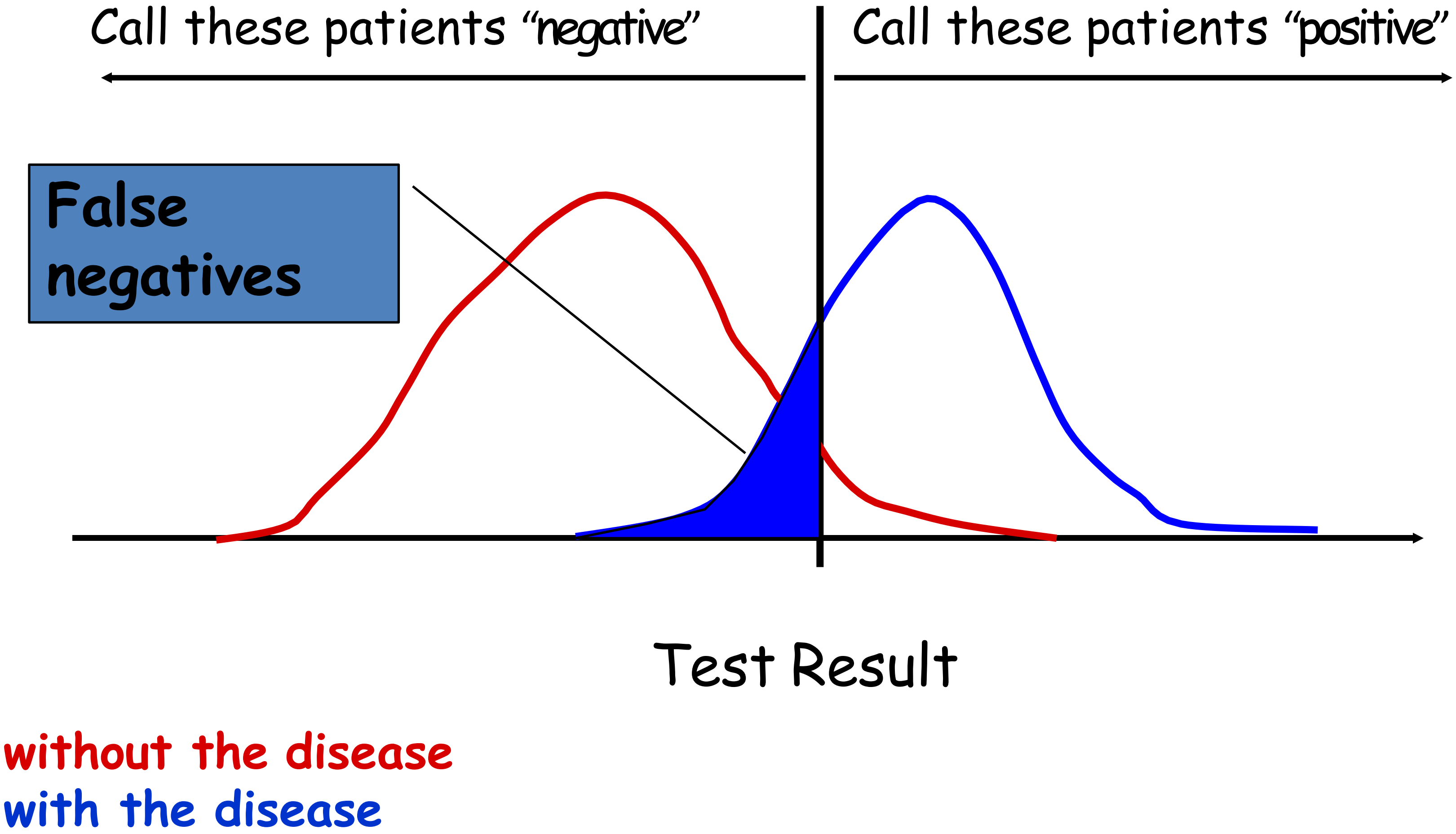
Call these patients "negative"
Call these patients "positive"
Test Result

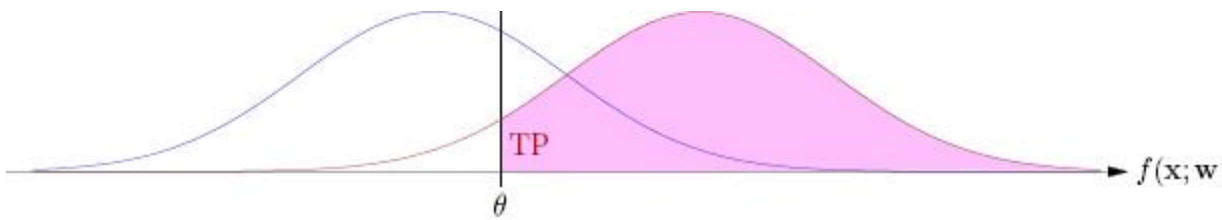
Some definitions ...



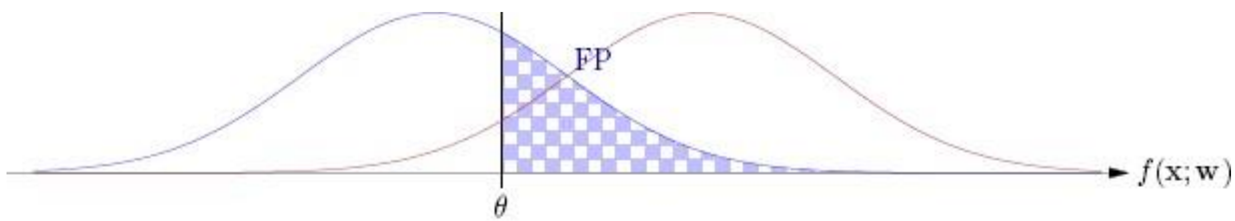

Call these patients "negative"
Call these patients "positive"
Test Result
False Positives
without the disease
with the disease


Call these patients "negative"
Call these patients "positive"
True negatives
Test Result
without the disease
with the disease


Call these patients "negative"
Call these patients "positive"
False negatives
Test Result
without the disease
with the disease

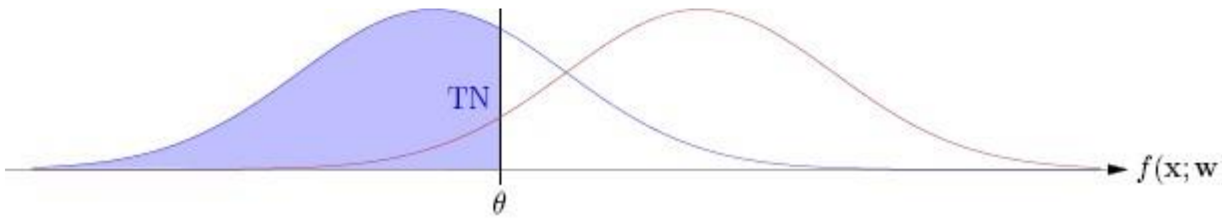


True Positives (TP) = 93.3%



True Positives (TP) = 93.3%

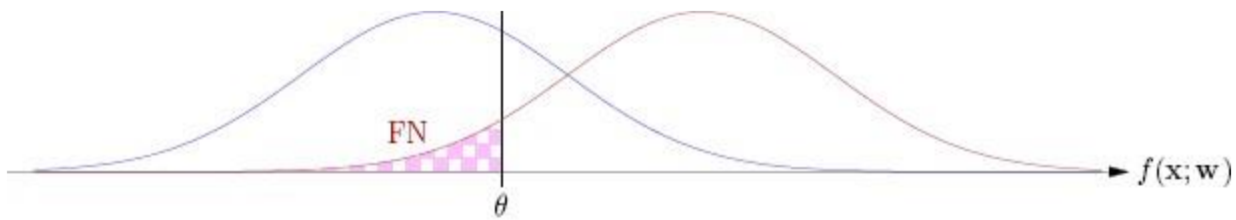
False Positives (FP)= 30.9%



True Positives (TP) = 93.3%

False Positives (FP)= 30.9%

True Negatives (TN) = 69.1%

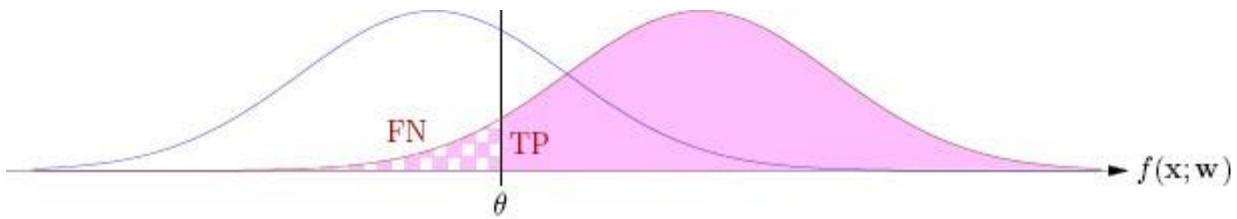


True Positives (TP) = 93.3%

False Positives (FP)= 30.9%

True Negatives (TN) = 69.1%

False Negatives (FN) = 6.68%



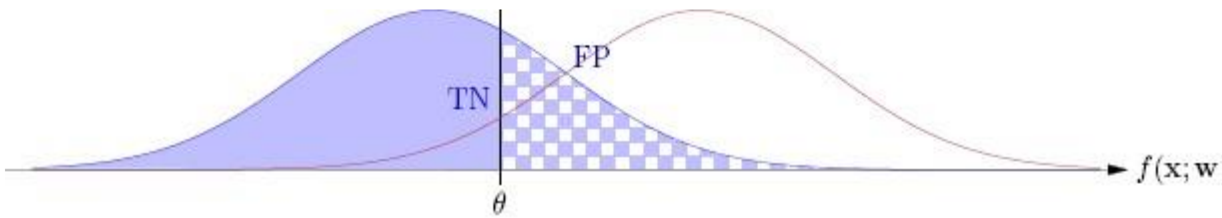
True Positives (TP) = 93.3%

False Positives (FP)= 30.9%

True Negatives (TN) = 69.1%

False Negatives (FN) = 6.68%

TPR (sensitivity) = $\frac{TP}{P} = \frac{TP}{TP+FN} = 0.933$



True Positives (TP) = 93.3%

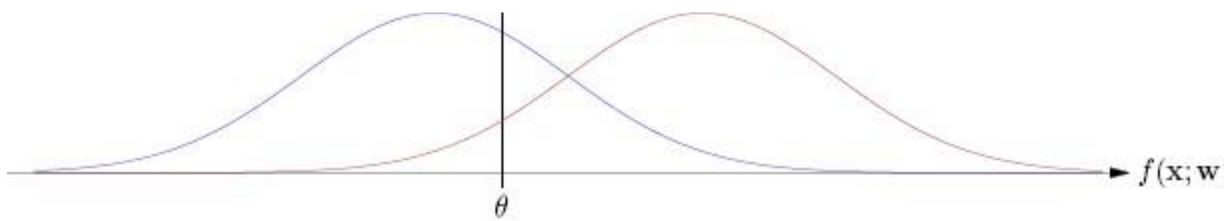
False Positives (FP)= 30.9%

True Negatives (TN) = 69.1%

False Negatives (FN) = 6.68%

TPR (sensitivity) $= \frac{TP}{P} = \frac{TP}{TP+FN} = 0.933$

FPR (1-specificity) $= \frac{FP}{N} = \frac{FP}{FP+TN} = 0.309$



True Positives (TP) = 93.3%

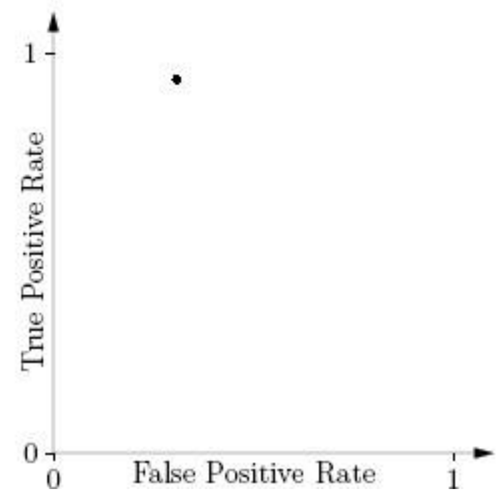
False Positives (FP)= 30.9%

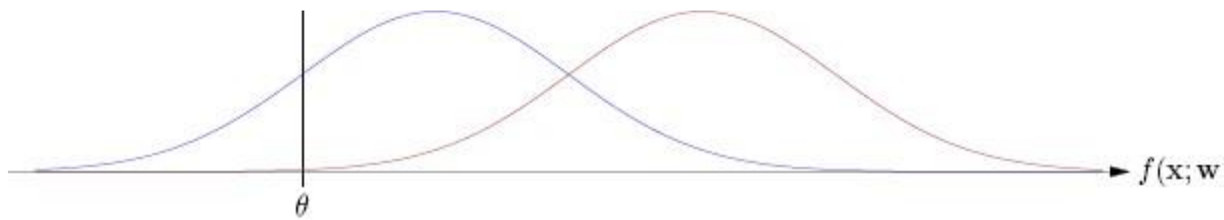
True Negatives (TN) = 69.1%

False Negatives (FN) = 6.68%

TPR (sensitivity) = $\frac{TP}{P} = \frac{TP}{TP+FN} = 0.933$

FPR (1-specificity) = $\frac{FP}{N} = \frac{FP}{FP+TN} = 0.309$





True Positives (TP) = 99.9%

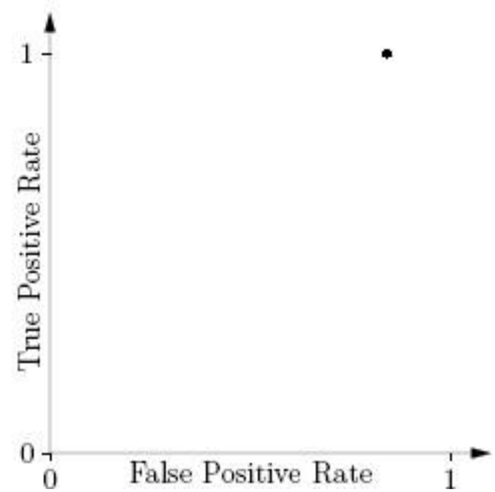
False Positives (FP)= 84.1%

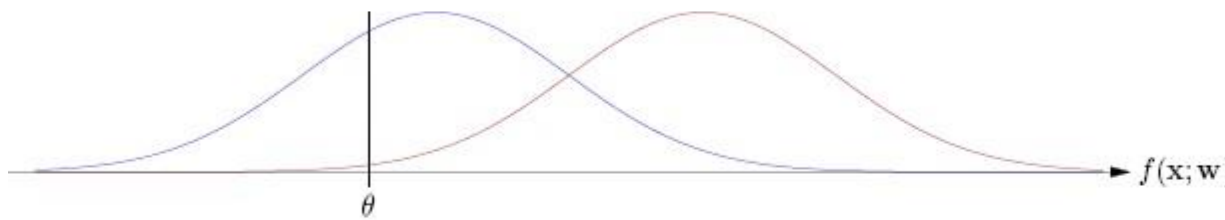
True Negatives (TN) = 15.9%

False Negatives (FN) = 0.135%

TPR (sensitivity) $= \frac{TP}{P} = \frac{TP}{TP+FN} = 0.999$

FPR (1-specificity) $= \frac{FP}{N} = \frac{FP}{FP+TN} = 0.841$





True Positives (TP) = 99.4%

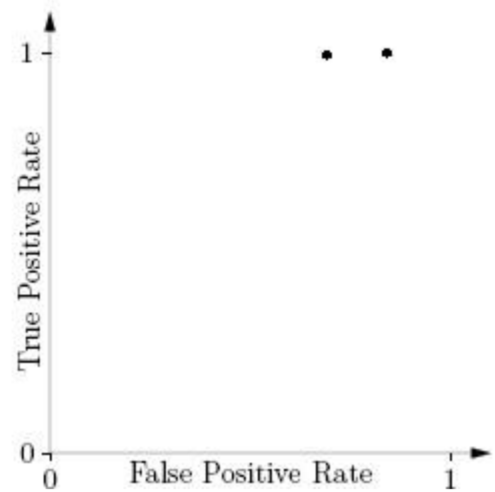
False Positives (FP)= 69.1%

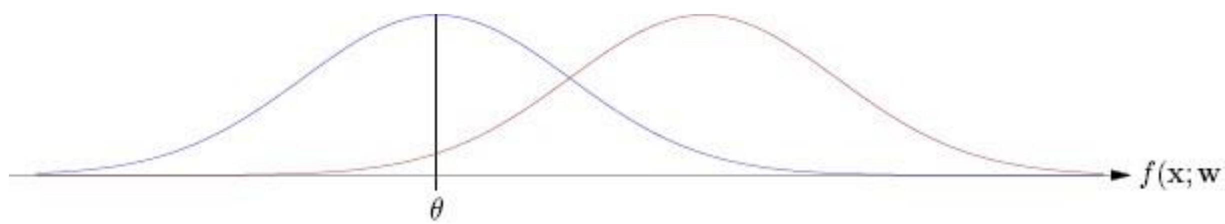
True Negatives (TN) = 30.9%

False Negatives (FN) = 0.621%

TPR (sensitivity) = $\frac{TP}{P} = \frac{TP}{TP+FN} = 0.994$

FPR (1-specificity) = $\frac{FP}{N} = \frac{FP}{FP+TN} = 0.691$





True Positives (TP) = 97.7%

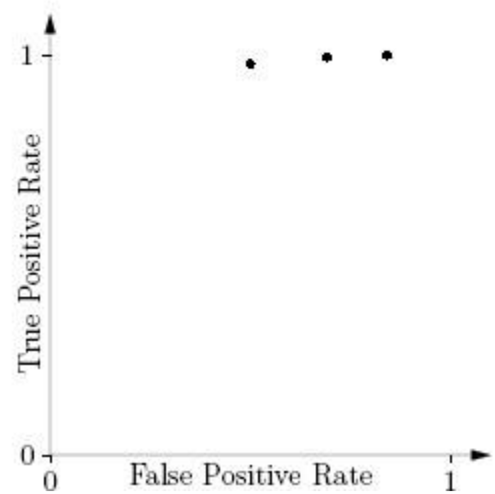
False Positives (FP)= 50%

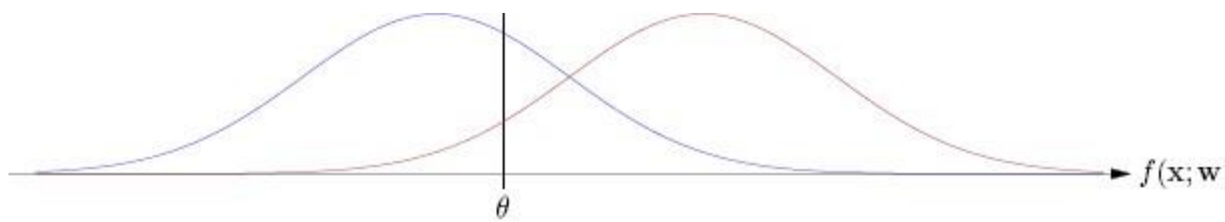
True Negatives (TN) = 50%

False Negatives (FN) = 2.28%

TPR (sensitivity) = $\frac{TP}{P} = \frac{TP}{TP+FN} = 0.977$

FPR (1-specificity) = $\frac{FP}{N} = \frac{FP}{FP+TN} = 0.5$





True Positives (TP) = 93.3%

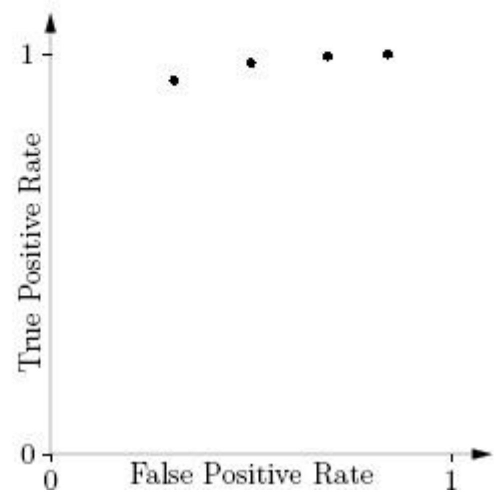
False Positives (FP)= 30.9%

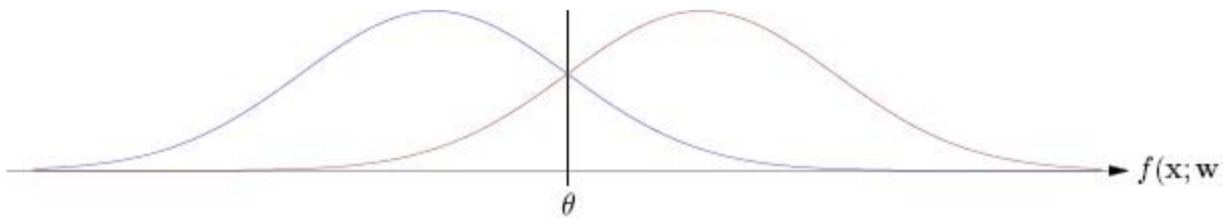
True Negatives (TN) = 69.1%

False Negatives (FN) = 6.68%

TPR (sensitivity) = $\frac{TP}{P} = \frac{TP}{TP+FN} = 0.933$

FPR (1-specificity) = $\frac{FP}{N} = \frac{FP}{FP+TN} = 0.309$





True Positives (TP) = 84.1%

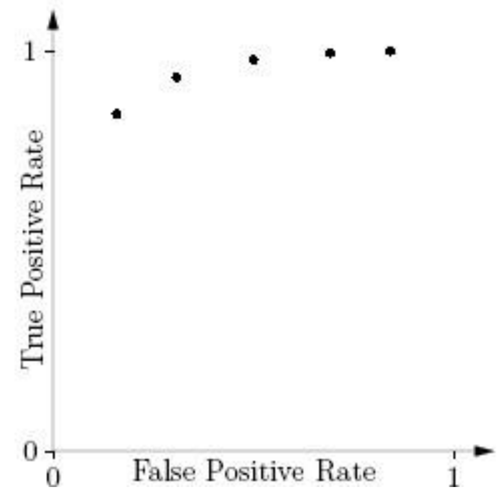
False Positives (FP)= 15.9%

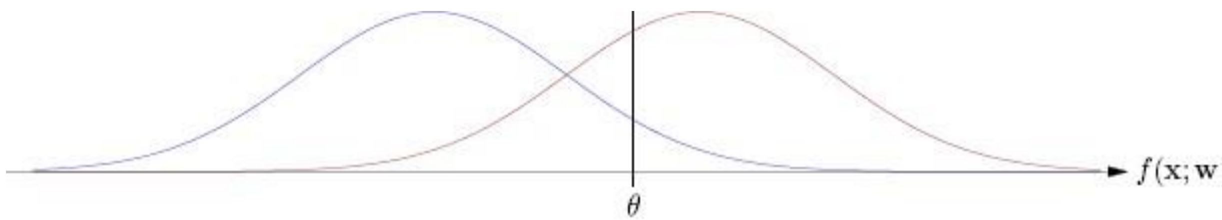
True Negatives (TN) = 84.1%

False Negatives (FN) = 15.9%

TPR (sensitivity) $= \frac{TP}{P} = \frac{TP}{TP+FN} = 0.841$

FPR (1-specificity) $= \frac{FP}{N} = \frac{FP}{FP+TN} = 0.159$





True Positives (TP) = 69.1%

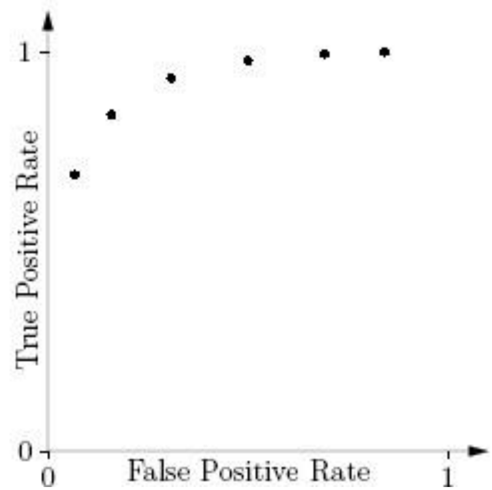
False Positives (FP)= 6.68%

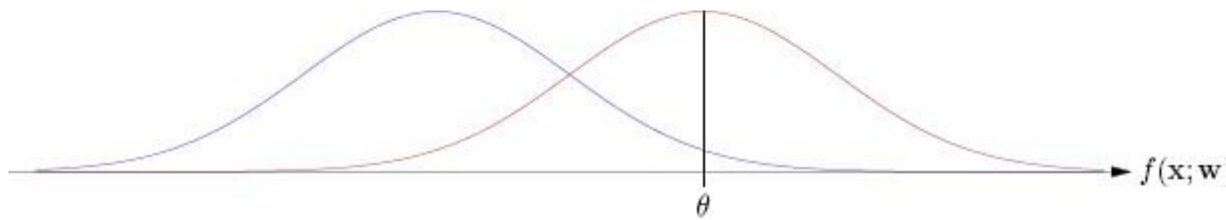
True Negatives (TN) = 93.3%

False Negatives (FN) = 30.9%

TPR (sensitivity) = $\frac{TP}{P} = \frac{TP}{TP+FN} = 0.691$

FPR (1-specificity) = $\frac{FP}{N} = \frac{FP}{FP+TN} = 0.0668$





True Positives (TP) = 50%

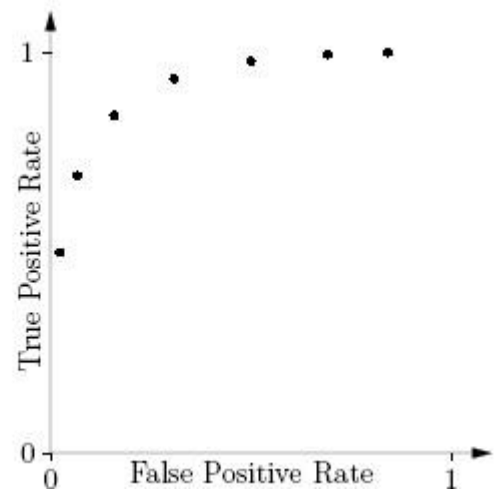
False Positives (FP)= 2.28%

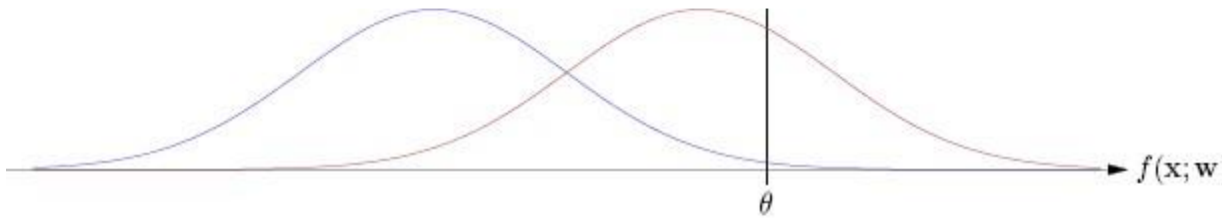
True Negatives (TN) = 97.7%

False Negatives (FN) = 50%

TPR (sensitivity) = $\frac{TP}{P} = \frac{TP}{TP+FN} = 0.5$

FPR (1-specificity) = $\frac{FP}{N} = \frac{FP}{FP+TN} = 0.0228$





True Positives (TP) = 30.9%

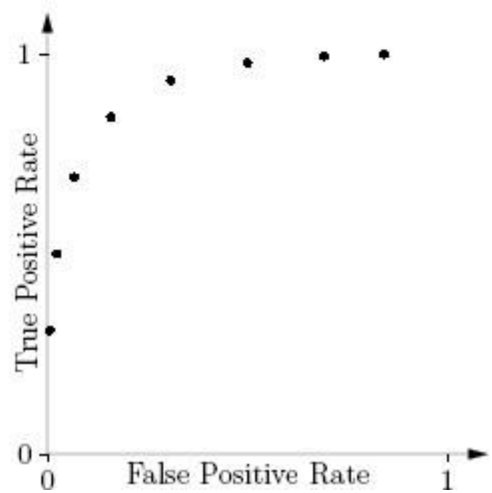
False Positives (FP)= 0.621%

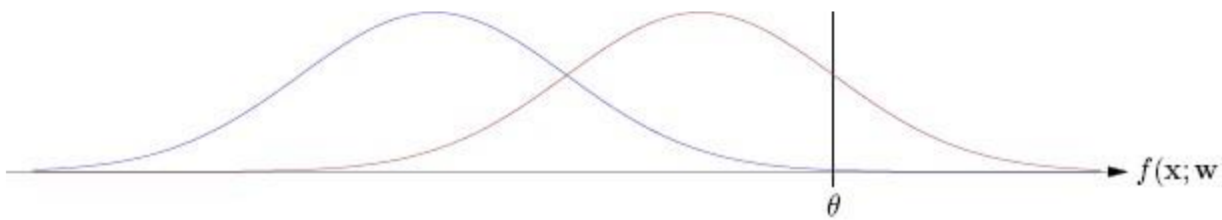
True Negatives (TN) = 99.4%

False Negatives (FN) = 69.1%

TPR (sensitivity) = $\frac{TP}{P} = \frac{TP}{TP+FN} = 0.309$

FPR (1-specificity) = $\frac{FP}{N} = \frac{FP}{FP+TN} = 0.00621$





True Positives (TP) = 15.9%

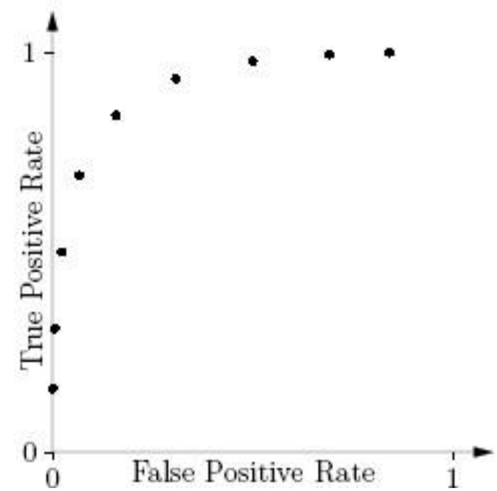
False Positives (FP)= 0.135%

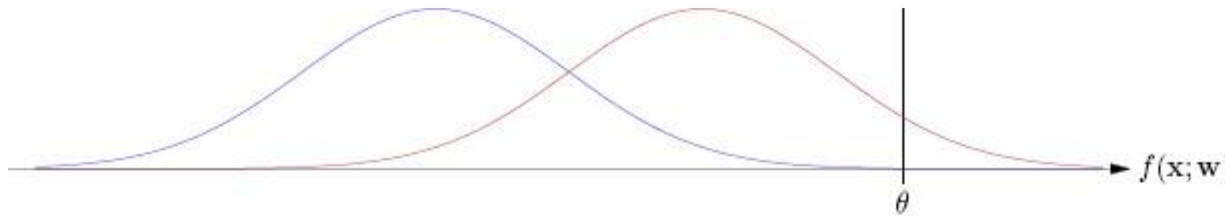
True Negatives (TN) = 99.9%

False Negatives (FN) = 84.1%

TPR (sensitivity) = $\frac{TP}{P} = \frac{TP}{TP+FN} = 0.159$

FPR (1-specificity) = $\frac{FP}{N} = \frac{FP}{FP+TN} = 0.00135$





True Positives (TP) = 6.68%

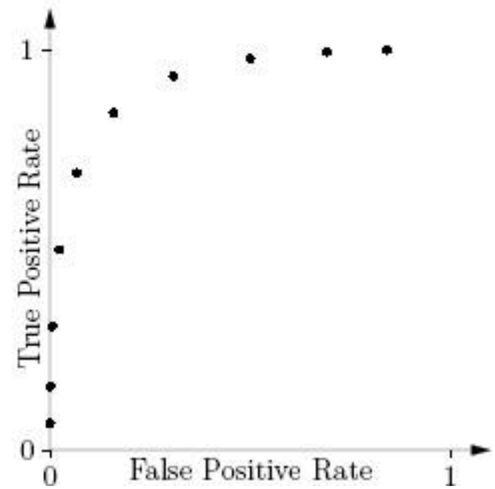
False Positives (FP)= 0.0233%

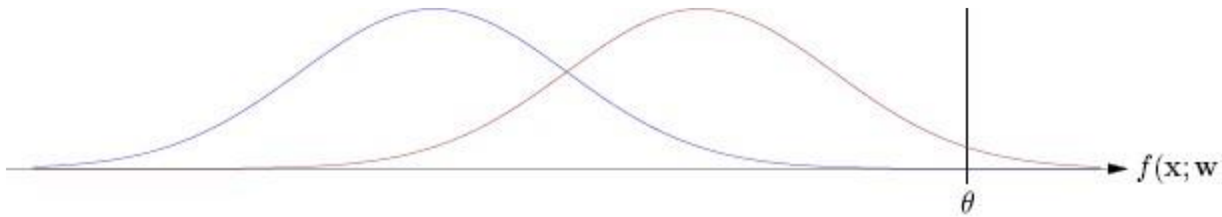
True Negatives (TN) = 100%

False Negatives (FN) = 93.3%

TPR (sensitivity) $= \frac{TP}{P} = \frac{TP}{TP+FN} = 0.0668$

FPR (1-specificity) $= \frac{FP}{N} = \frac{FP}{FP+TN} = 0.000233$





True Positives (TP) = 2.28%

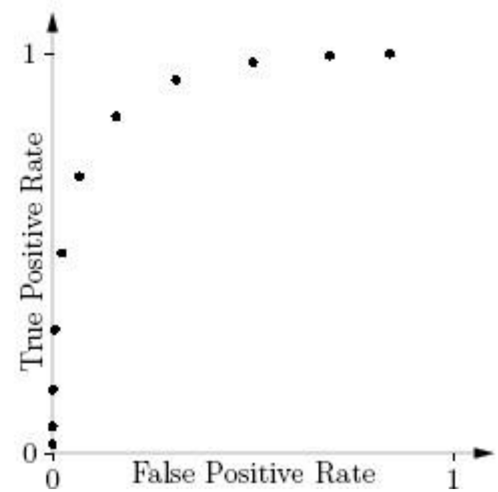
False Positives (FP)= 0.00317%

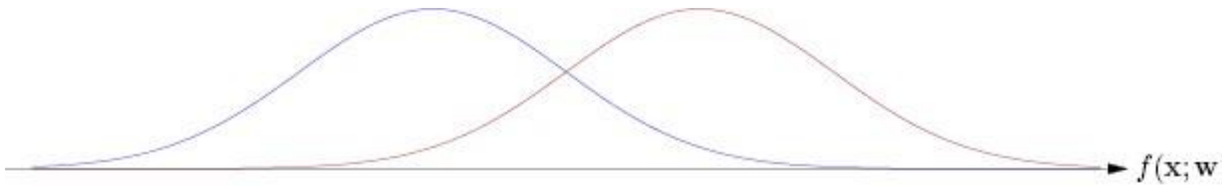
True Negatives (TN) = 100%

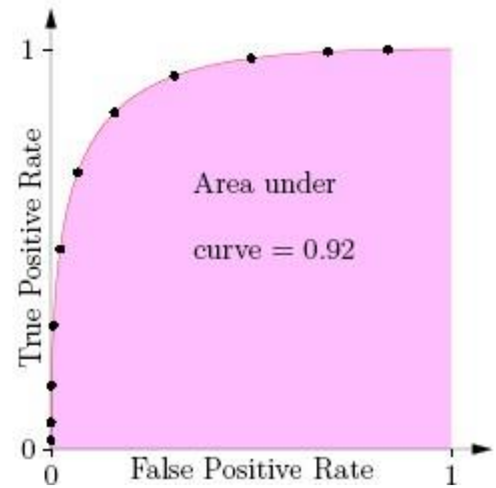
False Negatives (FN) = 97.7%

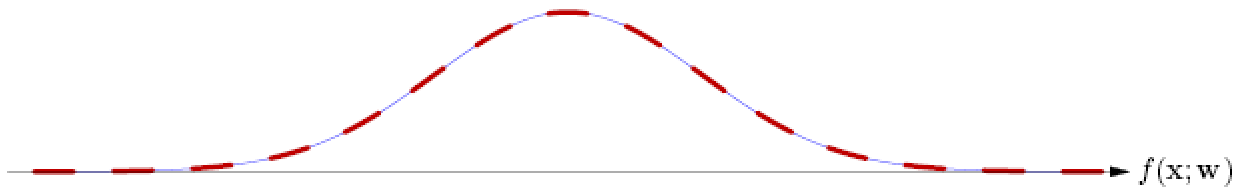
TPR (sensitivity) = $\frac{TP}{P} = \frac{TP}{TP+FN}$ = 0.0228

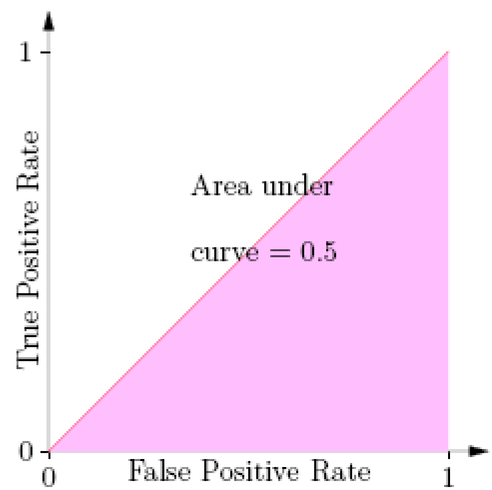
FPR (1-specificity) = $\frac{FP}{N} = \frac{FP}{FP+TN}$ = 3.17e-05

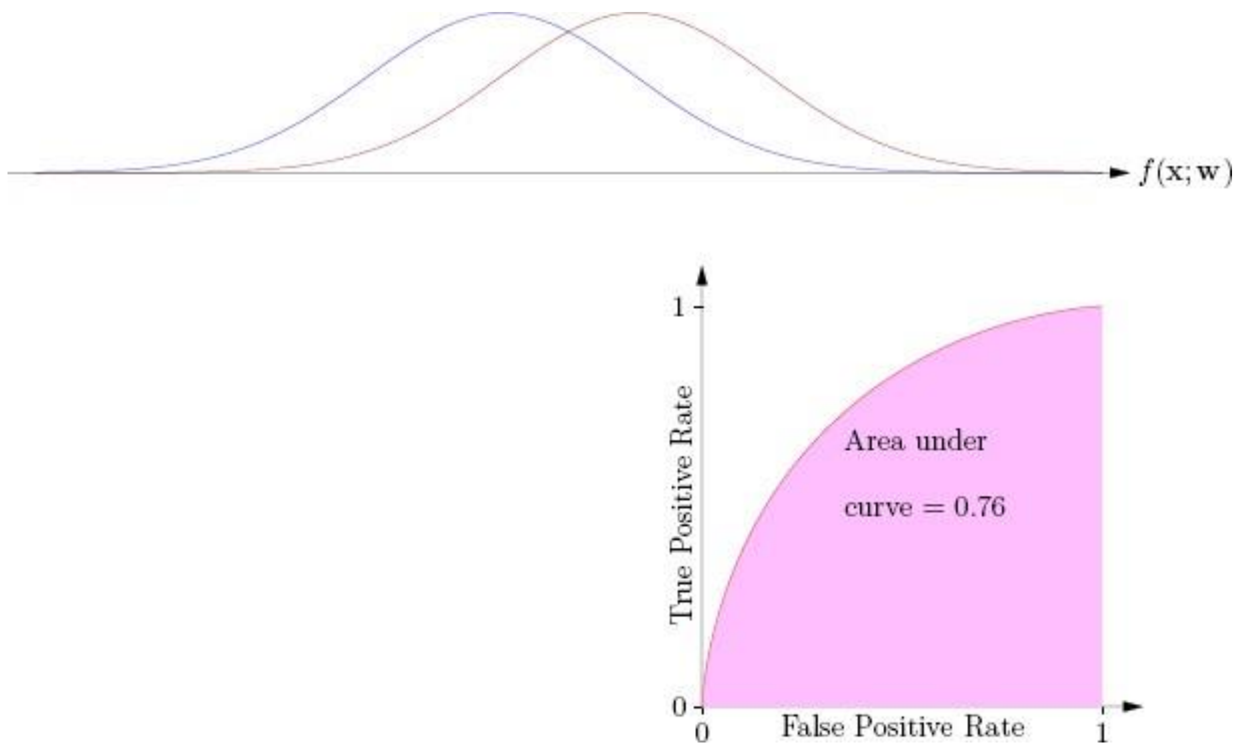



$f(\mathbf{x};\mathbf{w})$

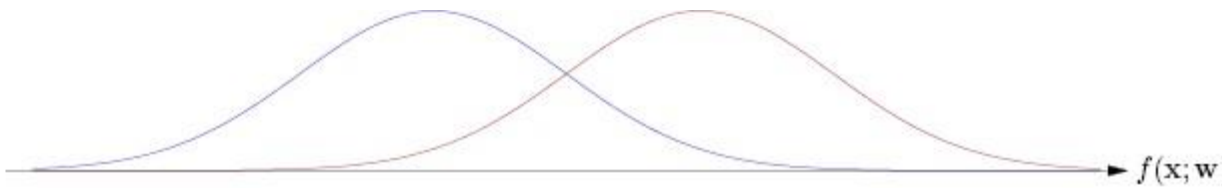

1
0
0
1
True Positive Rate
False Positive Rate
Area under
curve = 0.92

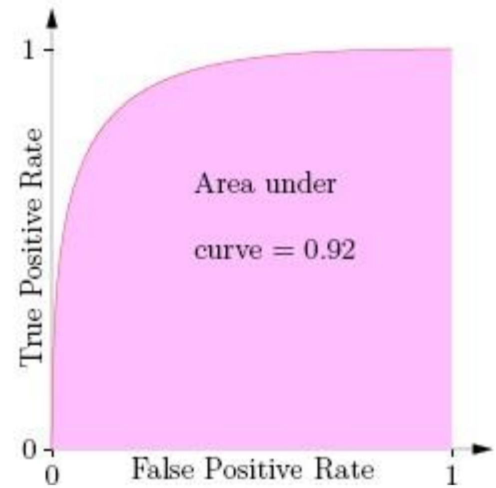

$f(\mathbf{x};\mathbf{w})$

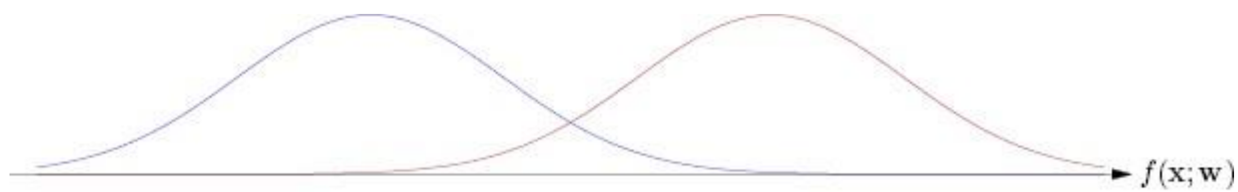

1
0
0
1
True Positive Rate
False Positive Rate
Area under
curve = 0.5

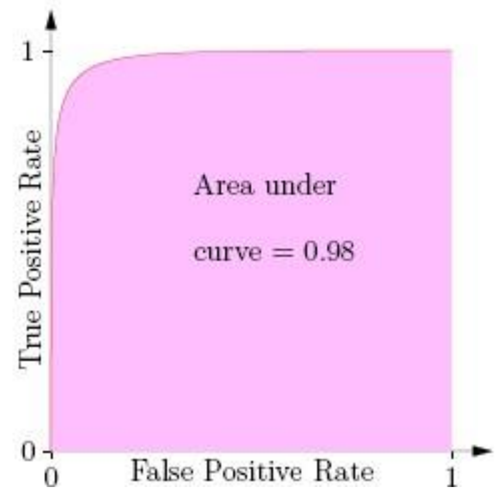


$f(\mathbf{x};\mathbf{w})$
1
0
True Positive Rate
Area under
curve = 0.76
0
False Positive Rate
1

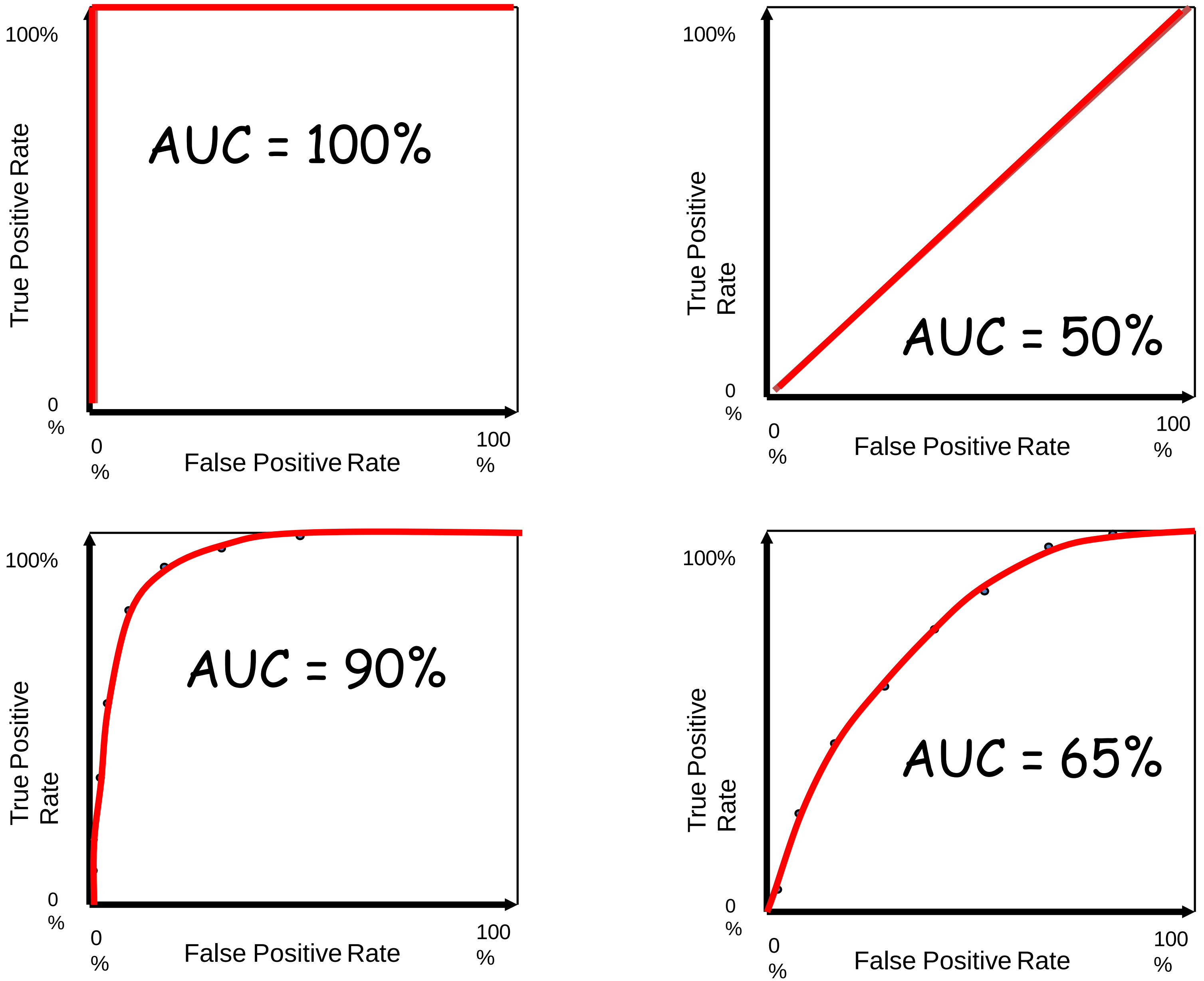

$f(\mathbf{x};\mathbf{w})$


1
True Positive Rate
Area under
curve = 0.92
0
0
False Positive Rate
1


$f(\mathbf{x};\mathbf{w})$


1
0
0
1
True Positive Rate
False Positive Rate
Area under
curve = 0.98

AUC for ROC curves



Data Normalization

- Between 0 to 1

$$((x-\min(x))/(\max(x)-\min(x)))$$

- Between -1 to 1

$$((x-\min(x))/(\max(x)-\min(x)))*2-1$$

Data Normalization

$$x_{ki} \to \frac{x_{ki} - \mu_i}{\sigma_i},$$

$$\mu_i = \frac{1}{P} \sum_{k=1}^{P} x_{ki},$$

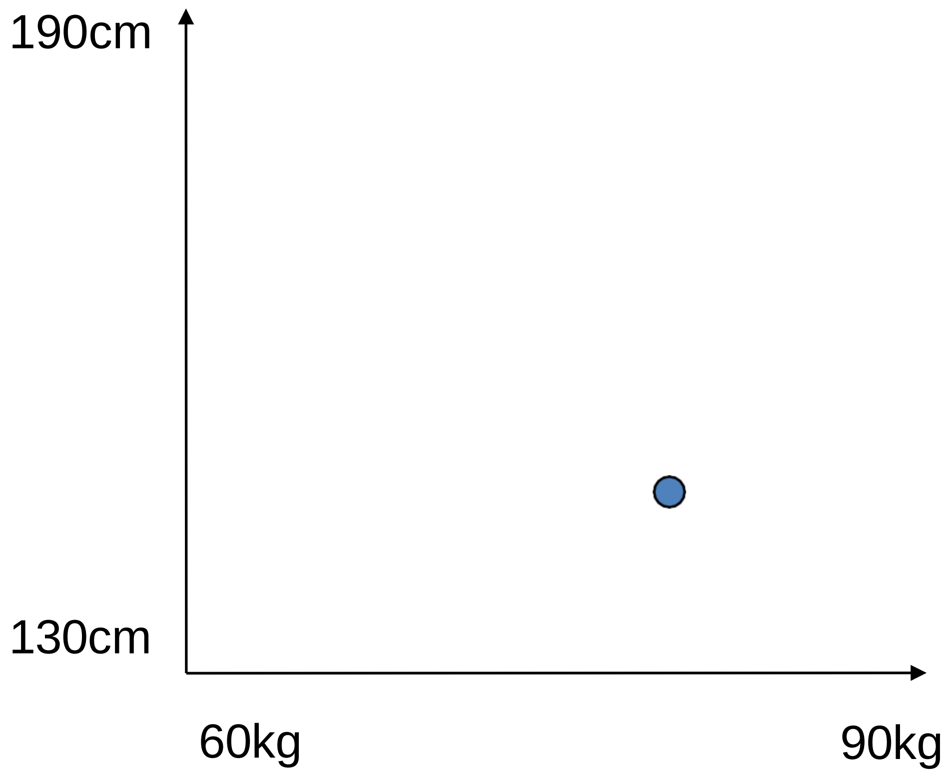
$$\sigma_i = \sqrt{\frac{1}{P-1} \sum_{k=1}^{P} (x_{ki} - \mu_i)^2}$$

Classification Example

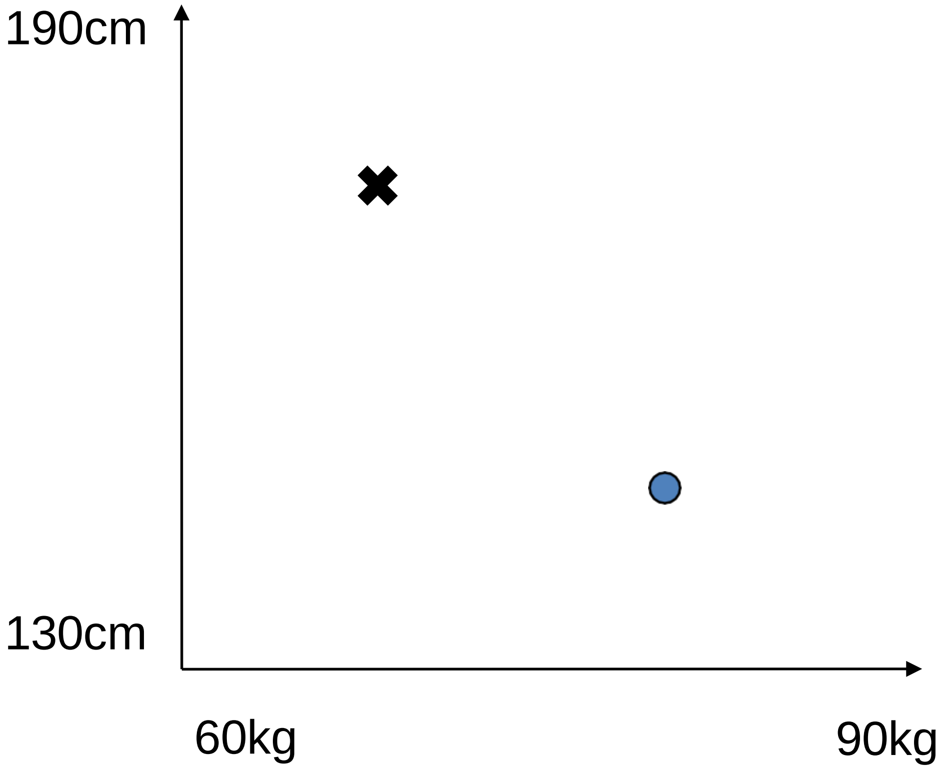
Can we LEARN to recognise a rugby player?

What are the “features” of a rugby player?

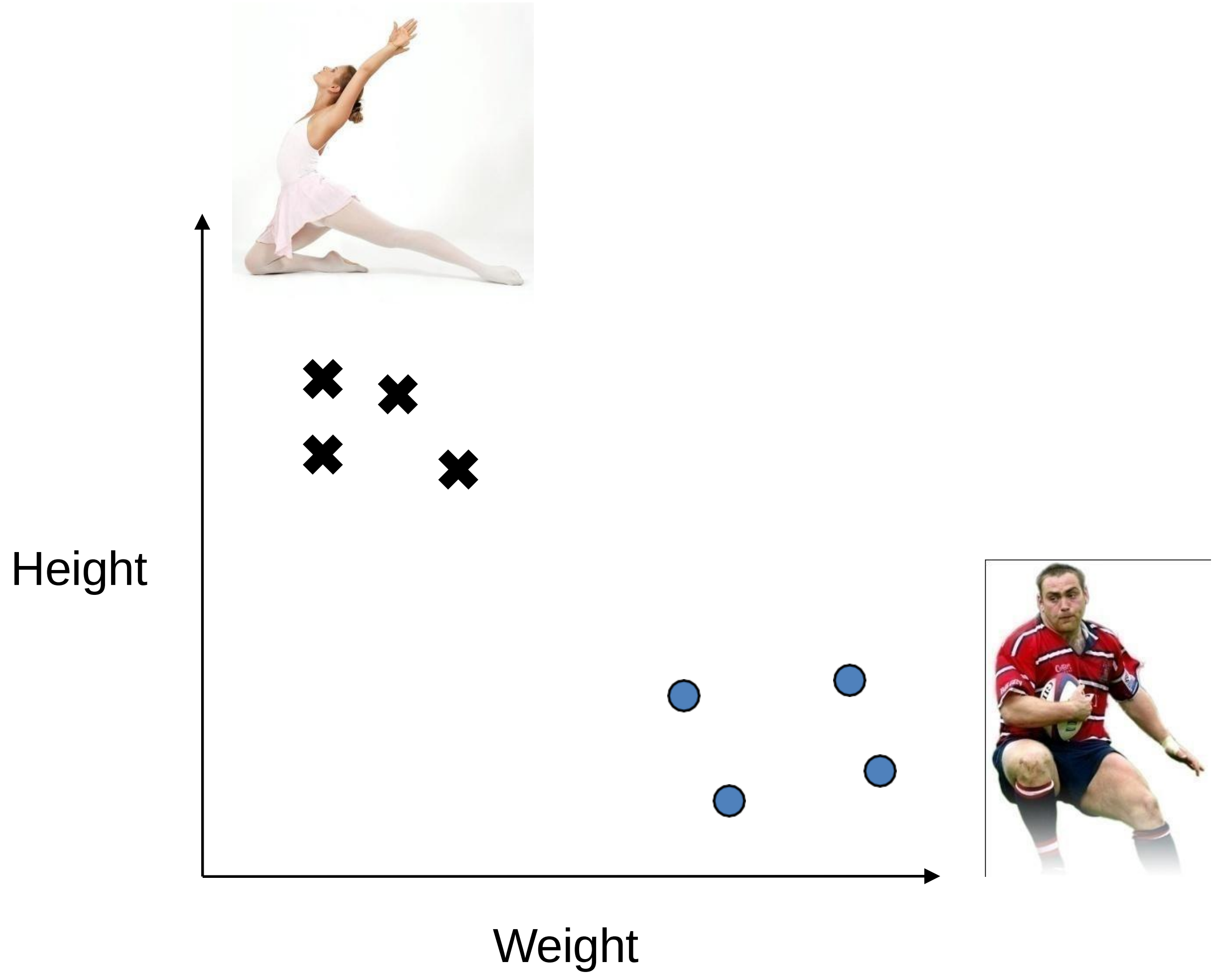
Rugby players = short + heavy?



Ballet dancers = tall + skinny?

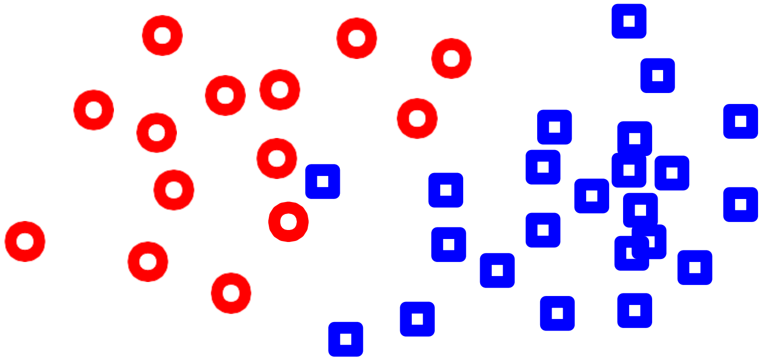


Rugby players "cluster" separately in the space.



K Nearest Neighbors

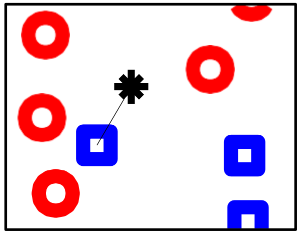
Nearest Neighbour Rule



Consider a two class problem where each sample consists of two measurements (x,y).

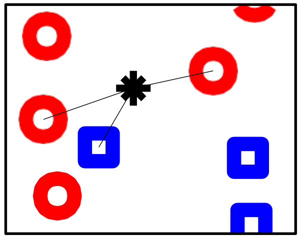
For a given query point q, assign the class of the nearest neighbour.

$k = 1$

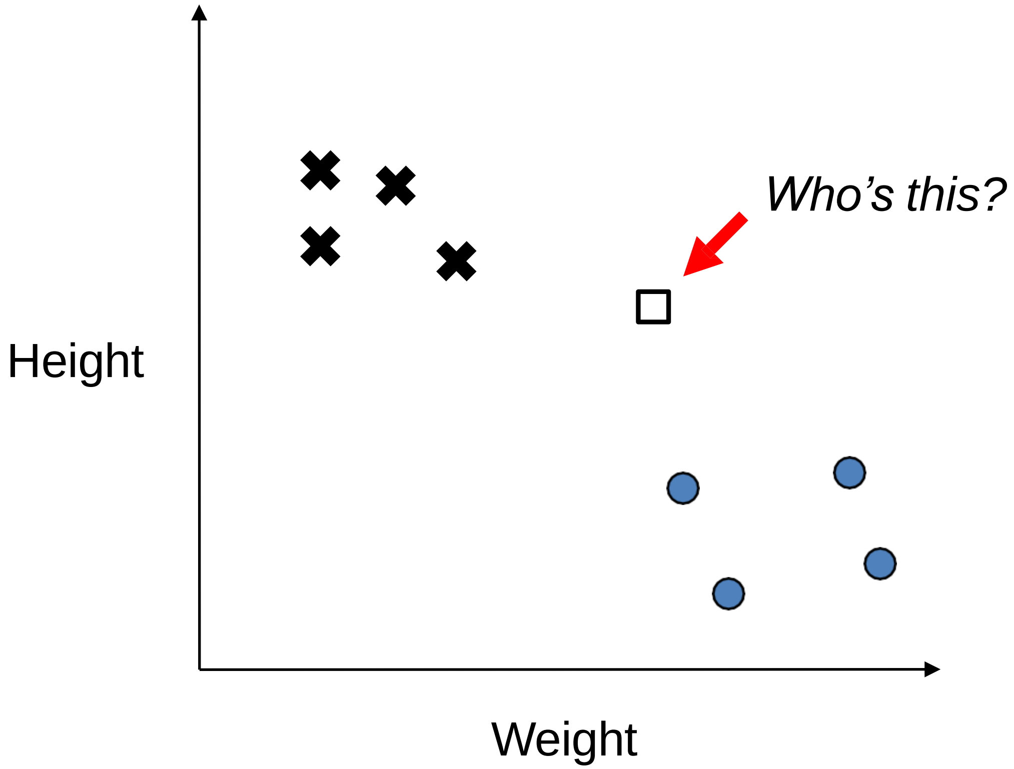


Compute the k nearest neighbours and assign the class by majority vote.

$k = 3$

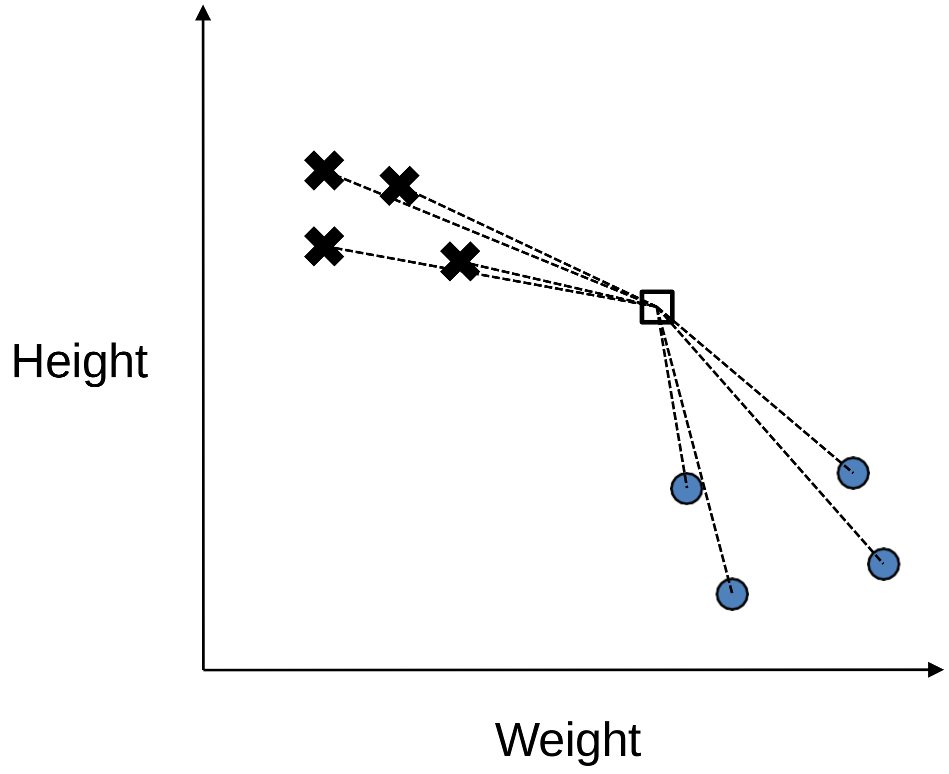


The K-Nearest Neighbour Algorithm



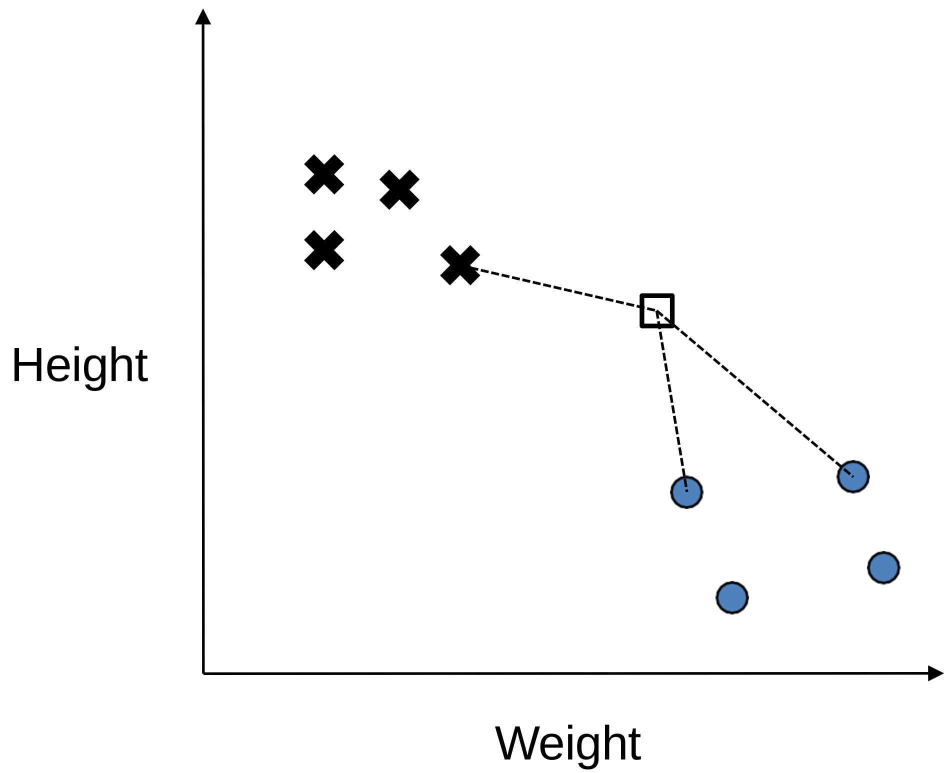
The K-Nearest Neighbour Algorithm

1. *Measure distance to all points*



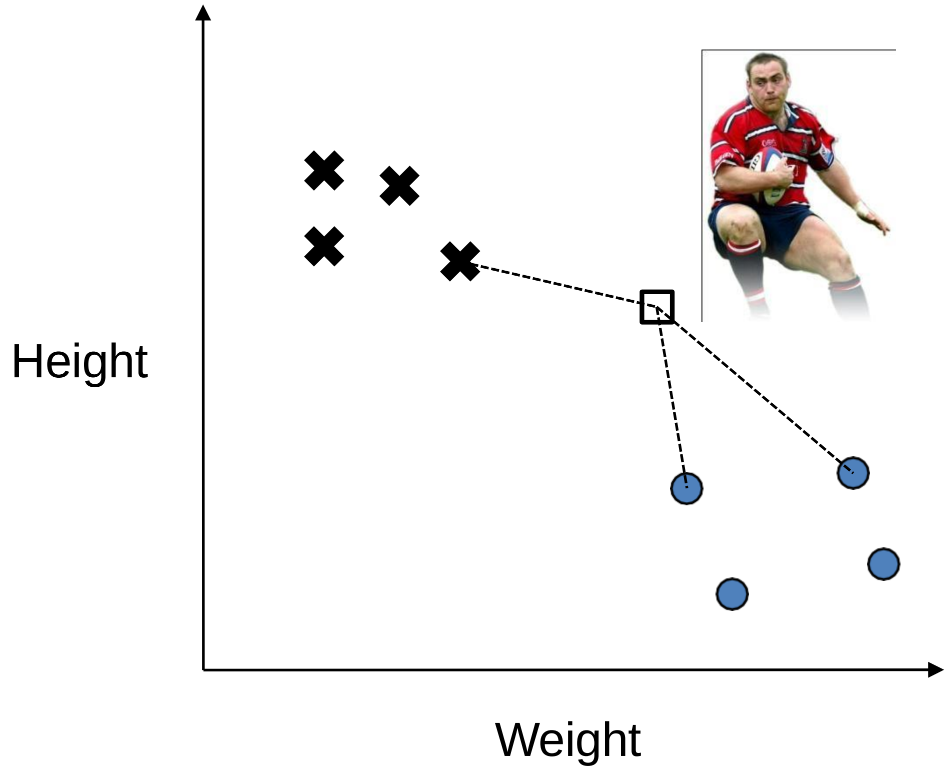
The K-Nearest Neighbour Algorithm

1. *Measure distance to all points*
2. *Find closest "k" points*   ← **(here k=3, but it could be more)**



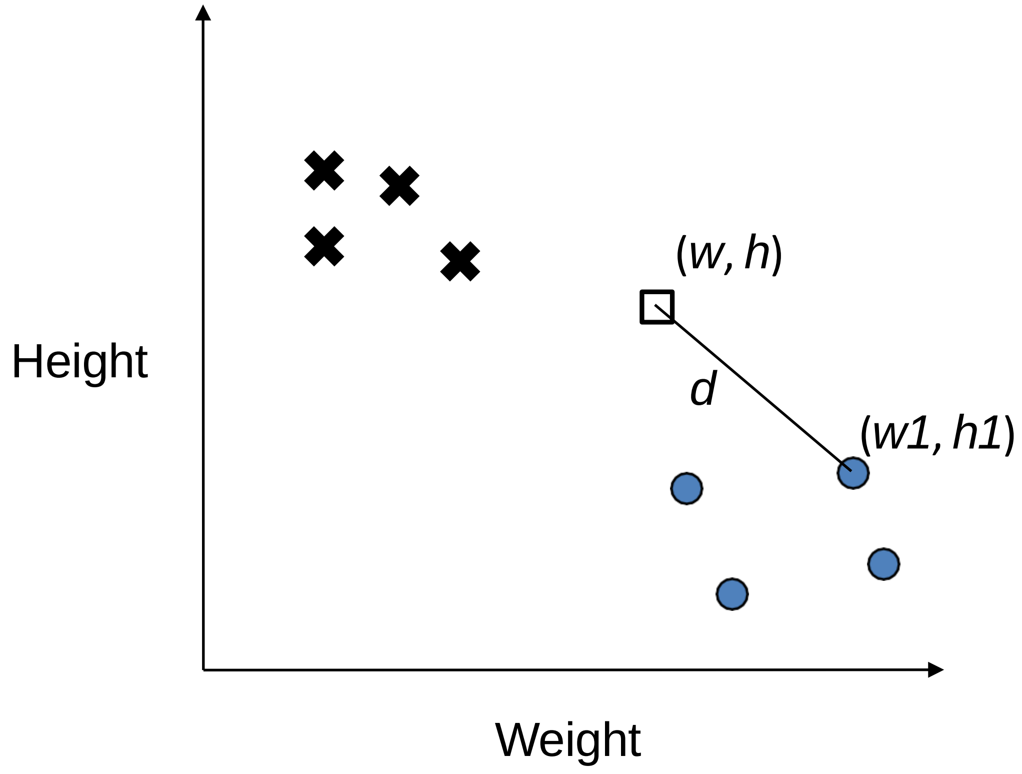
The K-Nearest Neighbour Algorithm

1. *Measure distance to all points*
2. *Find closest "k" points* ← **(here k=3, but it could be more)**
3. *Assign majority class*



“Euclidean distance”

$$d = \sqrt{(w - w_1)^2 + (h - h_1)^2}$$



The K-Nearest Neighbour Algorithm

for each testing point

measure distance to every training point find the k closest points

identify the most common class among those k

predict that class

end

- Advantage: Surprisingly good classifier!
- Disadvantage: Have to store the entire training set in memory

Euclidean distance still works in 3-d, 4-d, 5-d, etc….

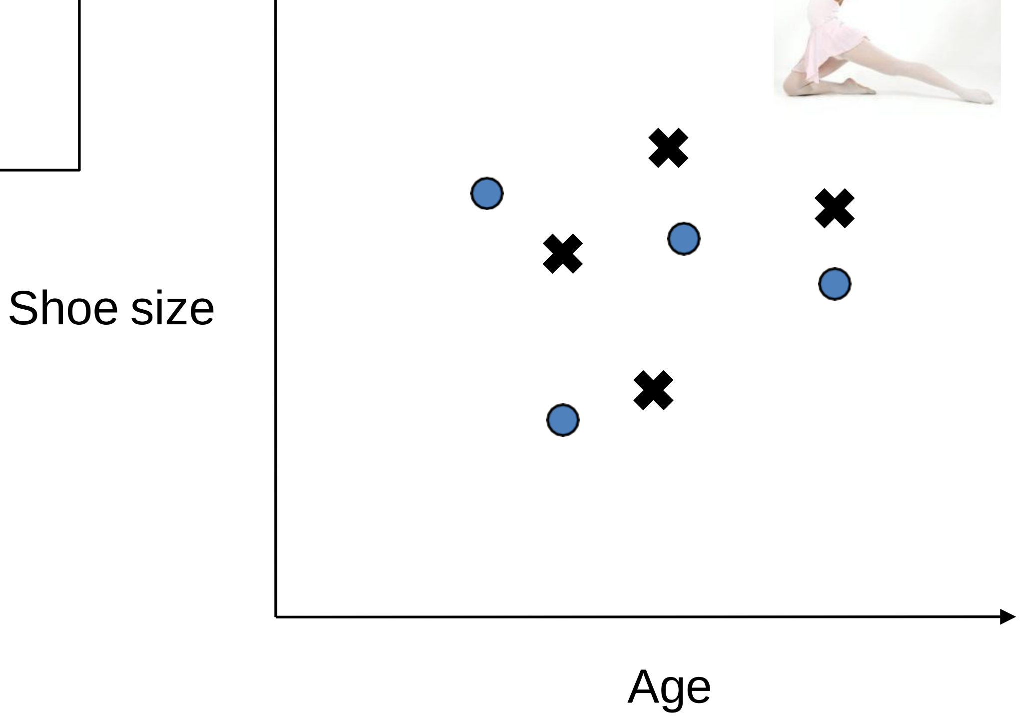
$$d = \sqrt{(x - x_1)^2 + (y - y_1)^2 + (z - z_1)^2}$$

x = *Height*
y = *Weight*
z = *Shoe size*

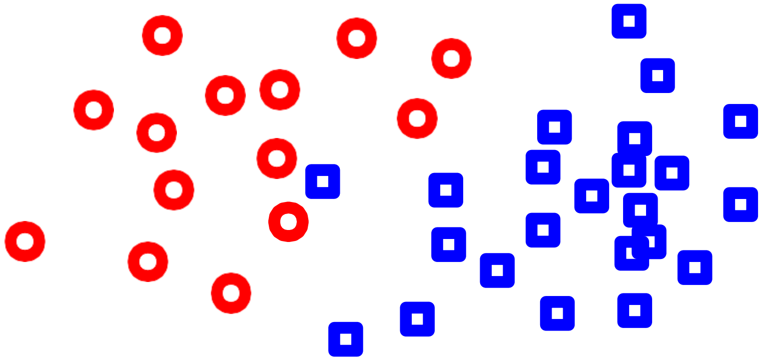
Choosing the wrong features makes it difficult, too many and it's computationally intensive.

Possible features:
- Shoe size ✓
- Height
- Age ✓
- Weight

?



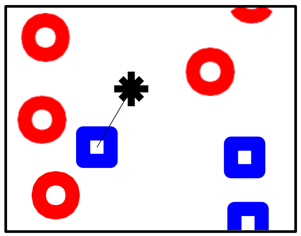
Nearest Neighbour Rule



Consider a two class problem where each sample consists of two measurements (x,y).

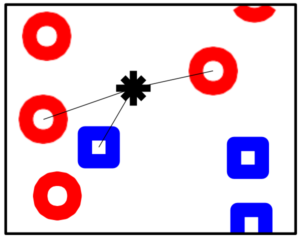
For a given query point q, assign the class of the nearest neighbour.

$k = 1$

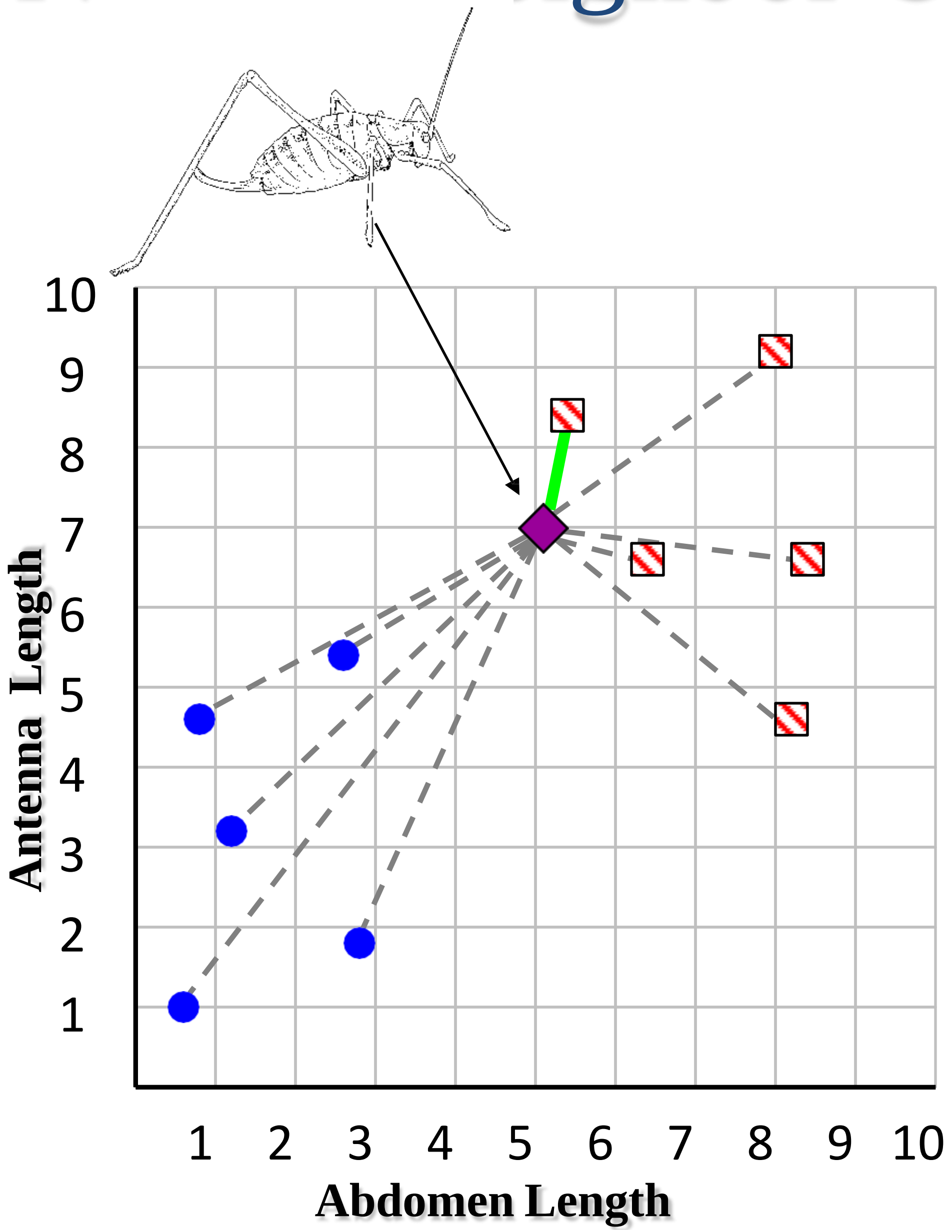


Compute the k nearest neighbours and assign the class by majority vote.

$k = 3$



Nearest Neighbor Classifier



If the **nearest** instance to the previously unseen instance **is a Katydid**

class is **Katydid**

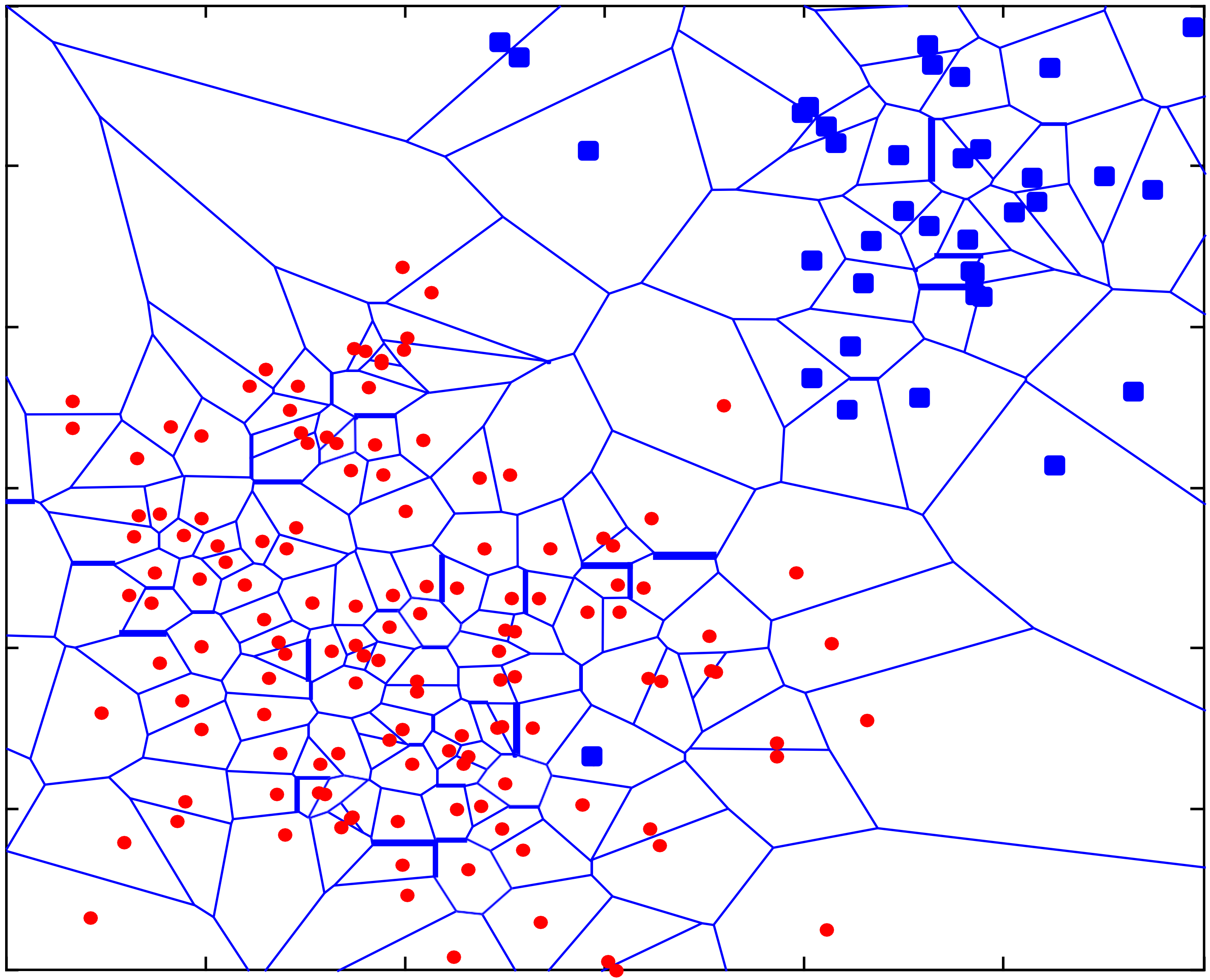
else

class is **Grasshopper**

Katydids

Grasshoppers

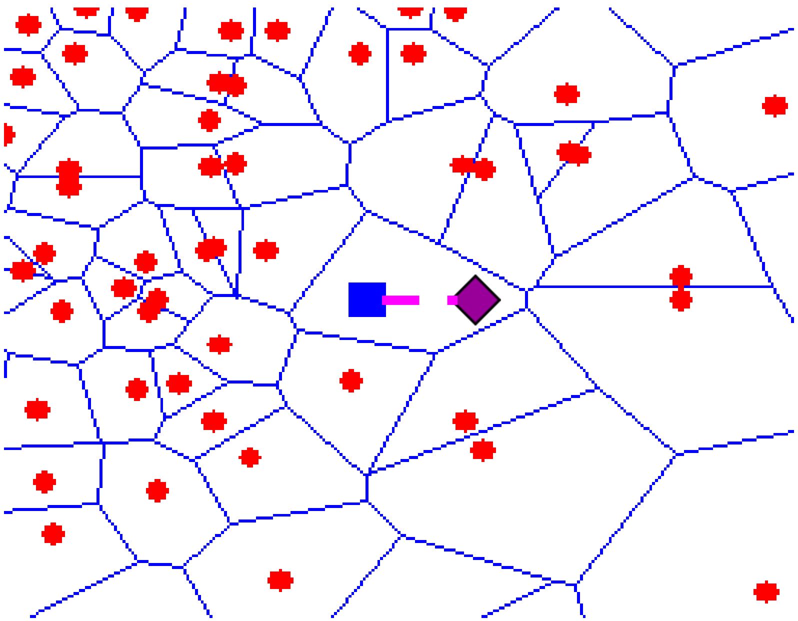
The nearest neighbor algorithm is sensitive to outliers…



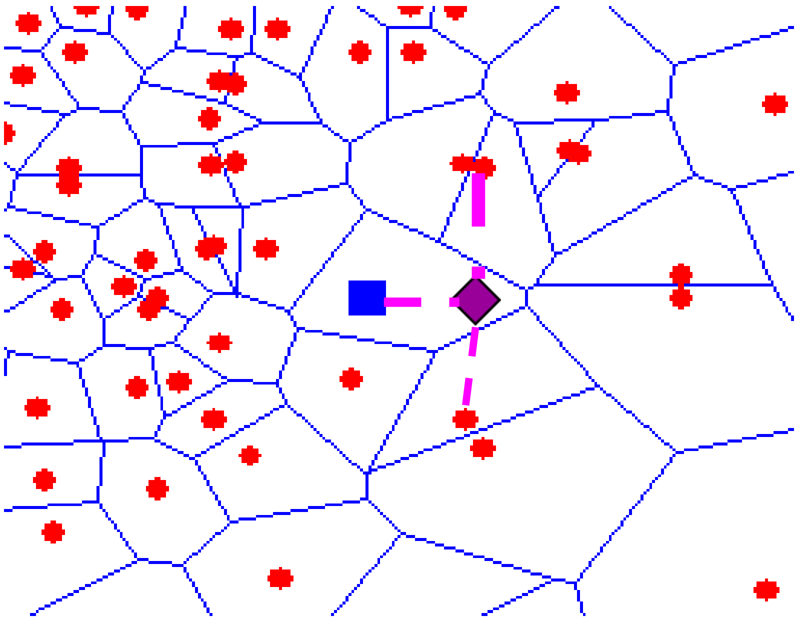
The solution is to…

We can generalize the nearest neighbor algorithm to the K- nearest neighbor (KNN) algorithm.

We measure the distance to the nearest K instances, and let them vote. K is typically chosen to be an odd number.



K = 1



K = 3

K-Nearest Neighbour Model

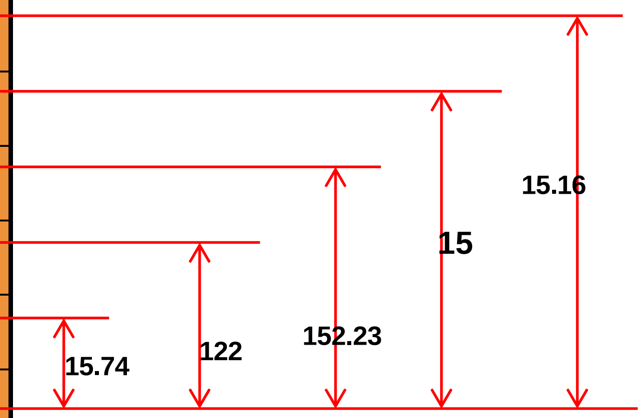
- Example : Classify whether a customer will respond to a survey question using a 3-Nearest Neighbor classifier

Customer	Age	Income	No. credit cards	Response
John	35	35K	3	No
Rachel	22	50K	2	Yes
Hannah	63	200K	1	No
Tom	59	170K	1	No
Nellie	25	40K	4	Yes
David	37	50K	2	?

K-Nearest Neighbour Model

- Example : 3-Nearest Neighbors

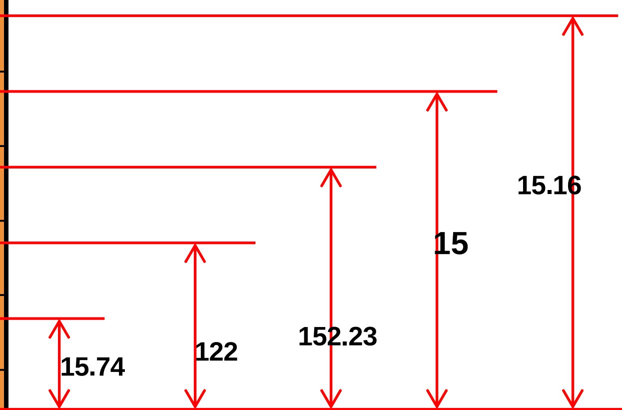
Customer	Age	Income	No. credit cards	Response
John	35	35K	3	No
Rachel	22	50K	2	Yes
Hannah	63	200K	1	No
Tom	59	170K	1	No
Nellie	25	40K	4	Yes
David	37	50K	2	?



K-Nearest Neighbour Model

- Example : 3-Nearest Neighbors

Customer	Age	Income	No. credit cards	Response
John	[illegible]	[illegible]	[illegible]	No
Rachel	[illegible]	[illegible]	[illegible]	Yes
Hannah	63	200K	1	No
Tom	59	170K	1	No
Nellie	[illegible]	[illegible]	[illegible]	Yes
David	37	50K	2	?

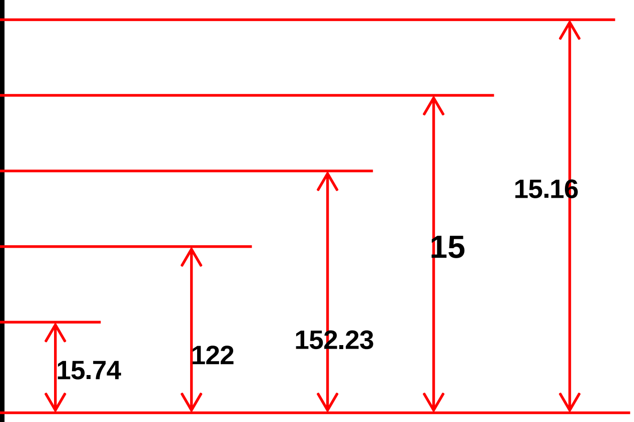


Three nearest ones to David are: No, Yes, Yes

K-Nearest Neighbour Model

- Example : 3-Nearest Neighbors

Customer	Age	Income	No. credit cards	Response
John	[illegible]	[illegible]	[illegible]	No
Rachel	[illegible]	[illegible]	[illegible]	Yes
Hannah	63	200K	1	No
Tom	59	170K	1	No
Nellie	[illegible]	[illegible]	[illegible]	Yes
David	37	50K	2	? Yes



Three nearest ones to David are: No, Yes, Yes

K-Nearest Neighbour Model

- Example: For the example we saw earlier, pick the best K from the set {1, 2, 3} to build a K-NN classifier

Customer	Age	Income	No. credit cards	Response
John	35	35K	3	No
Rachel	22	50K	2	Yes
Hannah	63	200K	1	No
Tom	59	170K	1	No
Nellie	25	40K	4	Yes
David	37	50K	2	?

Acknowledgements

- Introduction to Machine Learning, Alphaydin
- Statistical Pattern Recognition: A Review – A.K Jain et al., PAMI (22) 2000
- Pattern Recognition and Analysis Course – A.K. Jain, MSU
- *Pattern Classification"* by Duda et al., John Wiley & Sons.

Material in these slides has been taken from, the following resources

Thank you