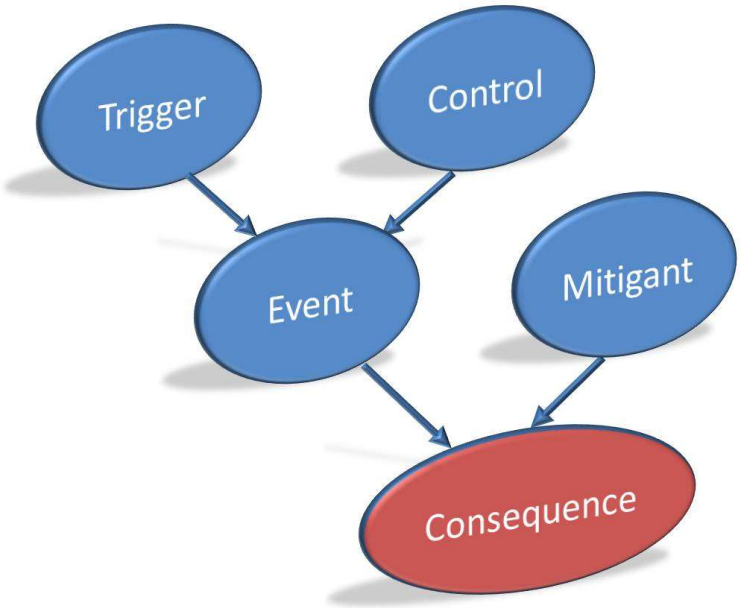


Risk and Decision Making for Data Science and AI

LESSON 8 Learning from Data: Algorithms and their accuracy

Norman Fenton

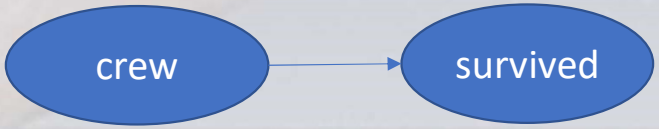
@ProfNFenton

The basic Titanic 'death' statistics

	Total	Died	Survived
Adult Female passengers	408	104	304
Adult Female staff	22	5	17
Child Female passengers	56	28	28
Adult Male passengers	780	647	133
Adult Male staff	886	691	195
Child male passengers	56	28	28
Total Females	486	137	349
Total males	1722	1366	356
Total passengers	1300	807	493
Total crew	908	696	212
Total	**2208**	**1503**	**705**

QUESTION: What is the probability a survivor was a crew member? (note: This is NOT the same as the probability a crew member survived)

Hint: create this model in AgenaRisk (and note that every NPT entry can be taken straight from the tabledata as AgenaRisk will do the necessary 'normalisation' to ensure probabilities sum to 1):



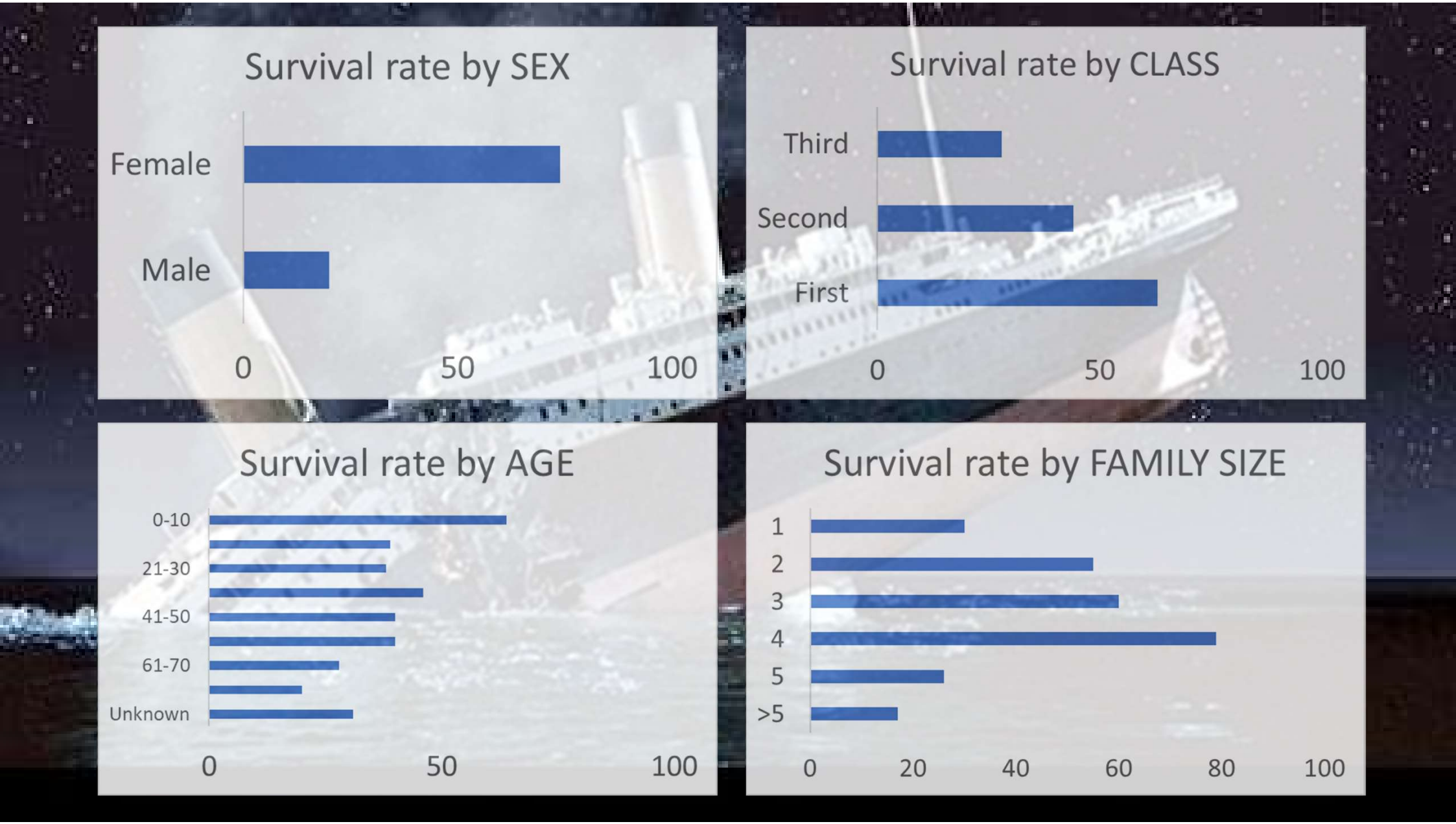
QUESTION: What is the probability a survivor was a man?

Note: people fall for all the classic risk assessment probabilistic fallacies and paradoxes when considering Titanic data

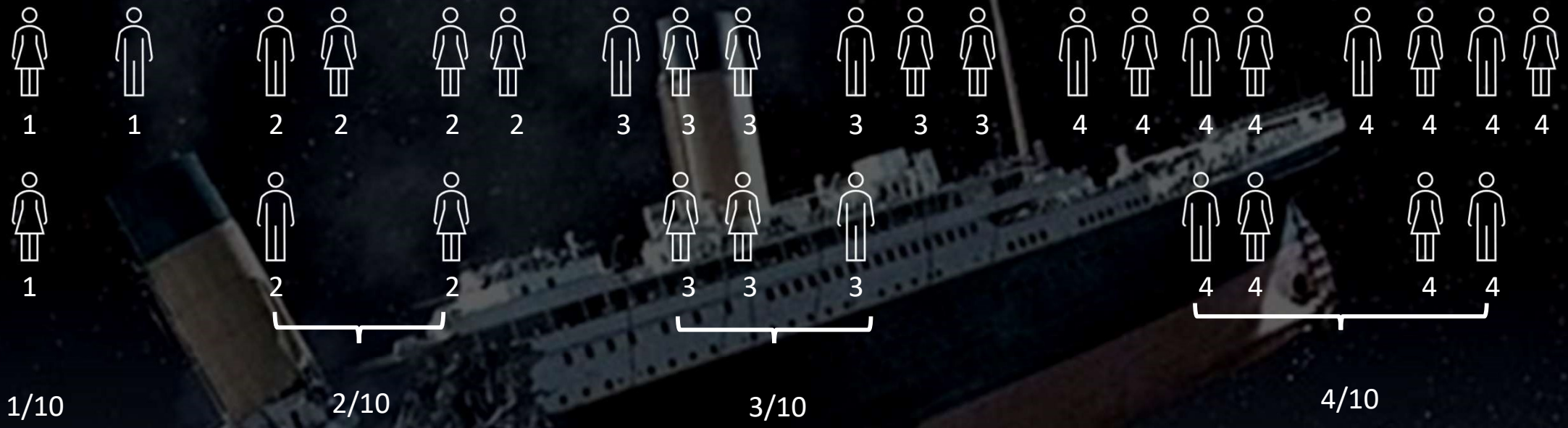
"Contrary to popular myth you were more likely to survive if you were a man than a woman".
False: The probability a man survived is not the same as the probability a survivor was a man
P(S | M) is not equal to P(M | S).
P(S | M) = 21% but P(M | S) = 50.5% In contrast P(S| W) = 72% but P(W | S) = 49.5%
…..there were far more men (1722) than women (486) onboard

"Being a first-class passenger doubles your chance of survival" on the basis that First-class overall survival rate is (62.46%) and the overall average survival rate (31.97%).
But, in fact for the survival rate for 3rd class women (46%) is 13% greater than 1st class men (33%).

https://www.anesi.com/titanic.htm

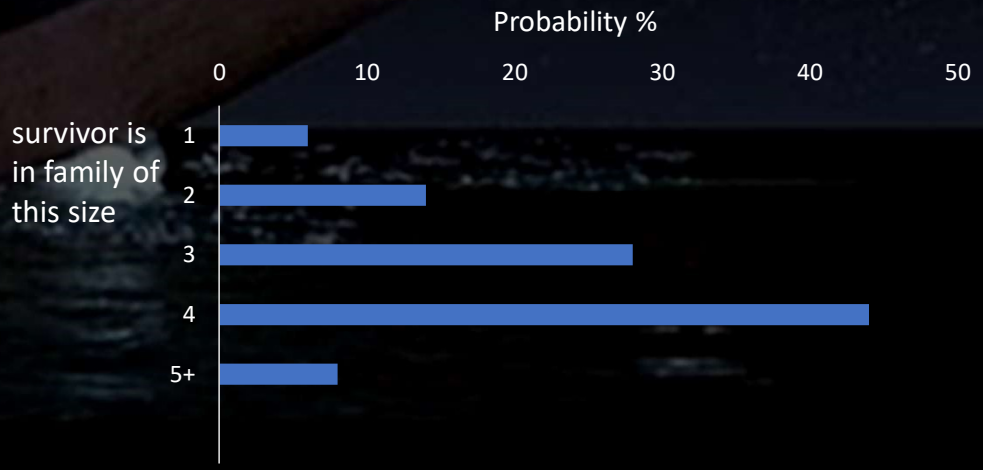

Survival rate by SEX
Female
Male
0
50
100
Survival rate by CLASS
Third
Second
First
0
50
100
Survival rate by AGE
0-10
21-30
41-50
61-70
Unknown
0
50
100
Survival rate by FAMILY SIZE
1
2
3
4
5
>5
0
20
40
60
80
100

And don’t mistake statistical certainties for ‘risk’



So what happened to 5+?

There were very few 5+ families and they were mainly 3rd class



‘Classic’ regression modelling

We want to predict a (numeric) outcome (a variable like “total fatalities”) based on (numeric) attributes believed to impact on it like “temperature”

We ‘fit’ a (linear) regression equation which has the form

Fatalities = $c + t$*temperature

Standard Excel calculates the ‘coefficients’ which are the ‘intercept’ c and the ‘gradient’ t which we also can think of as the ‘weight’ of the attribute “temperature”

In general we may have data on other attributes that are believed to impact on the outcome variables, e.g. “average driving speed”, “number of journeys”

Standard Excel calculates the multiple linear regression equation of the form:

Fatalities = $c + s$*speed + j*journeys + t*temperature

(it refers to c as the ‘intercept’ and s, j, t as the ‘coefficients’ associated with the associated variables).

Other programs like MatLab can ‘fit’ non-linear equations, i.e. polynomials e.g.

Fatalities = $c + s_1$*speed + s_2 speed^2

So the regression equation is just a function *f* that is the best fit:

$$\textbf{Outcome} = f(\textbf{attribute}_1, \textbf{attribute}_2, \ldots, \textbf{attribute}_n)$$

All machine learning algorithms are based on the same principle

We have data for n attributes A1, A2, …, An and the outcome O

record	A1	A2	A3	…	An	O
1	a_{11}	a_{21}	a_{31}	…	a_{n1}	o_1
2	a_{12}	a_{22}	a_{32}	…	a_{n2}	o_2
…	…	…	…	…		
m	a_{1m}	a_{2m}	a_{3m}	…	a_{nm}	o_m

We are seeking an 'algorithm' *f* where

$$\text{Outcome} = f(\text{attribute}_1, \text{attribute}_2, \ldots, \text{attribute}_n)$$

i.e. which takes as its inputs the attribute values and produces a predicted outcome that is the most 'accurate' predictor of the real outcome for those attributes.

Special machine learning challenges

Missing values

Non-numeric attributes

record	A1	A2	A3	…	An	O
1	a_{11}	a_{21}		Class X	a_{n1}	Yes
2	a_{12}		a_{32}	Class Z	a_{n2}	No
…	…	…	…	…		
m	a_{1m}	a_{2m}	a_{3m}	Class Y	a_{nm}	Yes

Hence, we have to use alternative methods to standard regression
But the overall principle is the same - we are still seeking an 'algorithm' *f* where

Outcome = *f*(attribute$_1$, attribute$_2$,…,attribute$_n$)

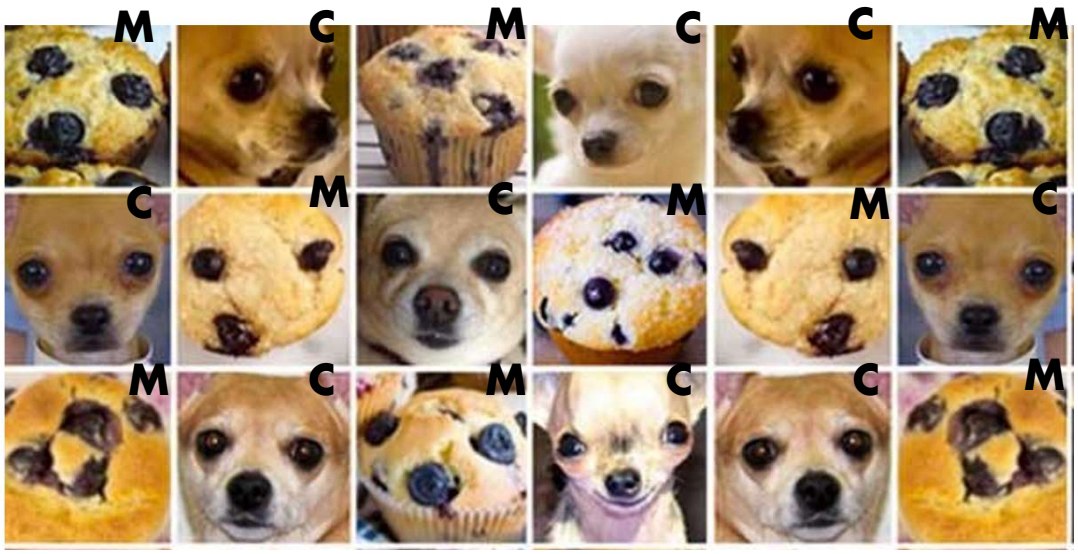
And for which *f* best predicts Outcome

Most datasets are not 'clean' – they require massive amount of effort to get them into a form suitable for applying machine learning algorithms

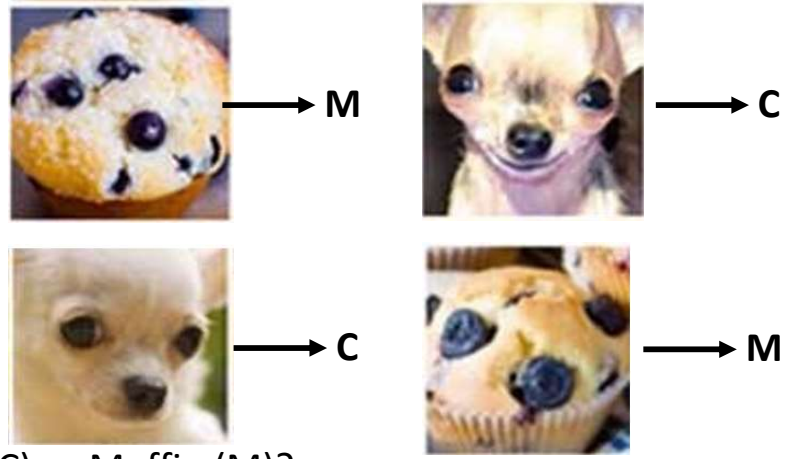
Example: The Titanic dataset

ID	Class	Name	Sex	Age	SibSp	Fare	Embarked	…	Survived
1	3	Braund, Mr. Owen Harris	male	22	1	£7.25	S	…	0
2	1	Cumings, Mrs. John Bradle	female	38	1	£71.28	C	…	1
3	3	Heikkinen, Miss. Laina	female	26	0	£7.93	S	…	1
4	1	Futrelle, Mrs. Jacques Hea	female	35	1	£53.10	S	…	1
5	3	Allen, Mr. William Henry	male	35	0	£8.05	S	…	0
6	3	Moran, Mr. James	male		0	£8.46	Q	…	0
7	1	McCarthy, Mr. Timothy J	male	54	0	£51.86	S	…	0
8	3	Palsson, Master. Gosta Lec	male	2	3	£21.08	S	…	0
9	3	Johnson, Mrs. Oscar W (El	female	27	0	£11.13	S	…	1
10	2	Nasser, Mrs. Nicholas (Ade	female	14	1	£30.07	C	…	1
…	…	…	…	…	…	…	…	…	…

Supervised versus Unsupervised Learning

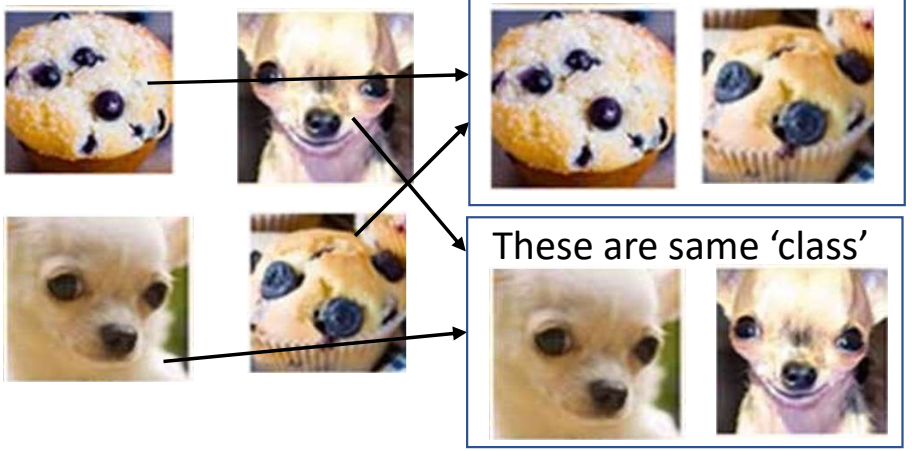


Supervised learning



Chihuahua (C) or Muffin (M)?

Unsupervised Learning



Titanic dataset: supervised learning

Training data: a subset of the passengers

ID	Class	Name	Sex	Age	SibSp	Fare	Embarked	…	Survived
1	3	Braund, Mr. Owen Harris	male	22	1	£7.25	S	…	0
2	1	Cumings, Mrs. John Bradle	female	38	1	£71.28	C	…	1
3	3	Heikkinen, Miss. Laina	female	26	0	£7.93	S	…	1
4	1	Futrelle, Mrs. Jacques Hea	female	35	1	£53.10	S	…	1
5	3	Allen, Mr. William Henry	male	35	0	£8.05	S	…	0
6	3	Moran, Mr. James	male		0	£8.46	Q	…	0
7	1	McCarthy, Mr. Timothy J	male	54	0	£51.86	S	…	0
8	3	Palsson, Master. Gosta Le	male	2	3	£21.08	S	…	0
9	3	Johnson, Mrs. Oscar W (El	female	27	0	£11.13	S	…	1
10	2	Nasser, Mrs. Nicholas (Ade	female	14	1	£30.07	C	…	1
…	…	…	…	…	…	…	…	…	…

From this data we ‘learn’ the outcome (survived) as a function of the other attributes

Test data: the remaining passengers (or a subset of those)

ID	Class	Name	Sex	Age	SibSp	Fare	Embarked	…
892	3	Kelly, Mr. James	male	34.5	0	£7.83	Q	…
893	3	Wilkes, Mrs. James (Ellen I	female	47	1	£7.00	S	…
894	2	Myles, Mr. Thomas Francis	male	62	0	£9.69	Q	…
895	3	Wirz, Mr. Albert	male	27	0	£8.66	S	…
896	3	Hirvonen, Mrs. Alexander	female	22	1	£12.29	S	…
897	3	Svensson, Mr. Johan Cervi	male	14	0	£9.23	S	…
898	3	Connolly, Miss. Kate	female	30	0	£7.63	Q	…
899	2	Caldwell, Mr. Albert Franc	male	26	1	£29.00	S	…
900	3	Abrahim, Mrs. Joseph (Sop	female	18	0	£7.23	C	…
…	…	…	…	…	…	…	…	…

We apply the function (algorithm) to predict the outcome for records where the outcome is (assumed) unknown

Simple Binary Classification Algorithms

Will Titanic passenger survive: Yes or No

Example Algorithm:
If Sex=Female then predict Yes
else predict No

True Positive (TP)

True Negative (TN)

False Positive (TP)

False Negative (FN)

Does patient have condition X: Yes or No

Example Algorithm:
If Diagnostic test is positive then Yes
else No

Will student pass the course: Yes or No

Example Algorithm:
If Student revises > 6 hours then Yes
else No

Will customer default on loan: Yes or No

Will customer renew subscription: Yes or No

etc

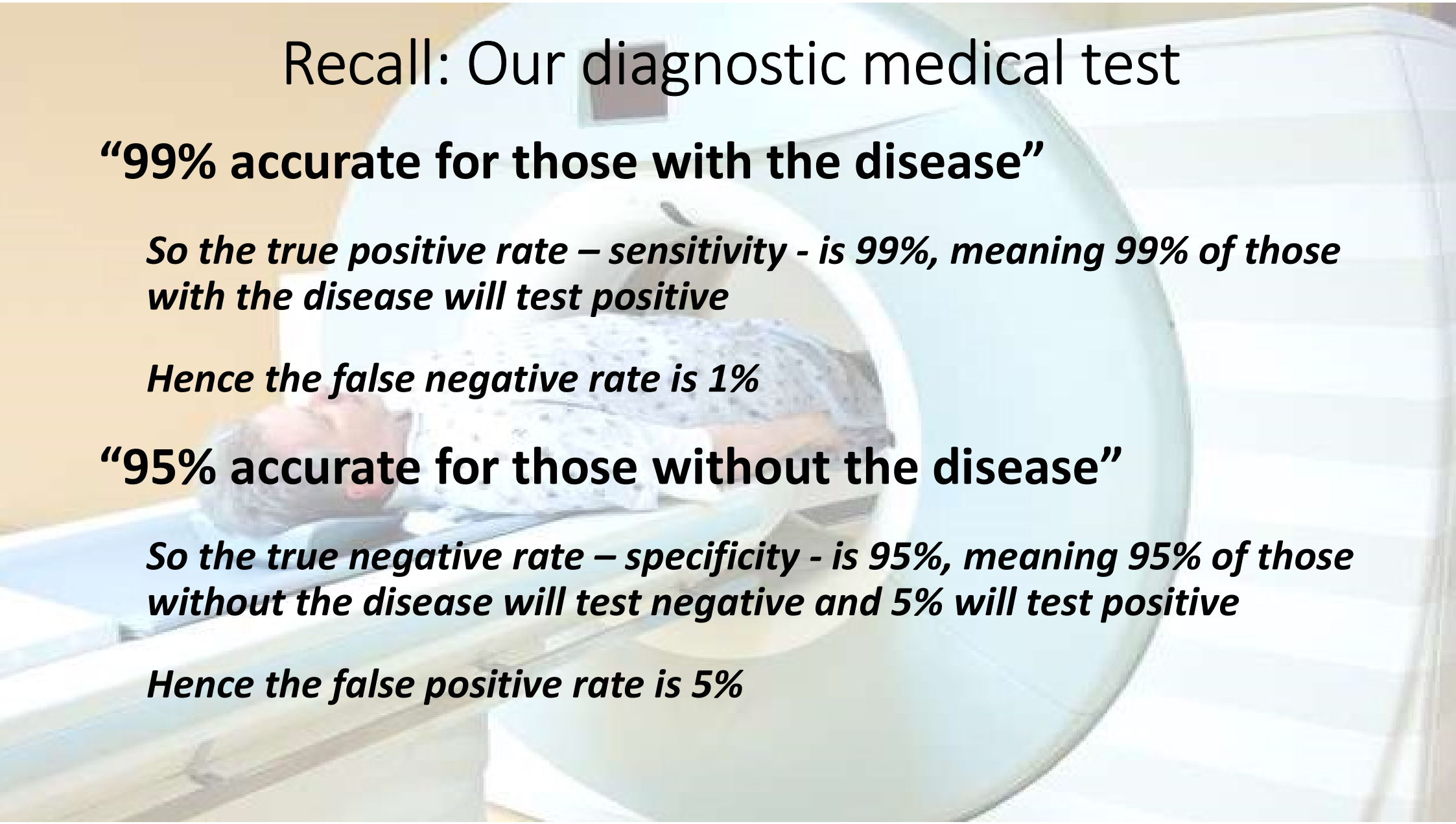
Simple Binary Classification Algorithms: Accuracy*

	Number Predicted to survive	Number Predicted not to survive	Accuracy
Survived	99 (all women) (99 out of all 150 women survived, no men were predicted to survive)	51 (all men) (51 out of the 259 men survived, no women were predicted not to survive)	**True positive (TP) rate** 99/150 who survived correctly predicted to survive. 66% (or 0.66) **False negative (FN) rate (always 1-TP)** 51/150 who survived wrongly predicted not to survive. 34% (or 0.34)
Did not survive	51 (all women) (51 out of the 150 women did not survive, no men were predicted to survive)	208 (all men) (208 out of all 259 men died, no women were predicted not to survive)	**True negative (TN) rate** 208/259 who did not survive correctly predicted not to survive. 80% (or 0.8) **False positive (FP) rate (always 1-TN)** 51/259 who did survive wrongly predicted to survive 20% (or 0.2)

Sensitivity:
True Positive Rate

Specificity:
True Negative Rate

*refers to the 409 passengers in the test dataset



Recall: Our diagnostic medical test

“99% accurate for those with the disease”

So the true positive rate – sensitivity - is 99%, meaning 99% of those with the disease will test positive

Hence the false negative rate is 1%

“95% accurate for those without the disease”

So the true negative rate – specificity - is 95%, meaning 95% of those without the disease will test negative and 5% will test positive

Hence the false positive rate is 5%

Simple measure of overall accuracy of a test or binary classification algorithm

	Number predicted YES	Number predicted NO	Total
Number YES's	*A*	*B*	*A+B*
Number NO's	*C*	*D*	*C+D*

Sensitivity = *A*/(*A*+*B*)

Specificity = *D*/(*C*+*D*)

Accuracy = (*A*+*D*)/(*A*+*B*+*C*+*D*)

Disease test example (YES = has disease)

	Number predicted YES	Number predicted NO	Total
Number YES's	99	1	100
Number NO's	4,995	94,905	99,900

Sensitivity = 99/100 = 99% **Specificity = 94950/99900 = 95%**

Accuracy = (99+94950)/100,000 = 95.05%

Titanic simple example (YES = survive)

	Number predicted YES	Number predicted NO	Total
Number YES's	99	51	150
Number NO's	51	208	259

Sensitivity = 99/150 = 66% **Specificity = 208/259 = 88%**

Accuracy = (99+208)/(150+259) = 76%

But this measure of accuracy takes no account of the confidence with which a prediction is made

The sensitivity/specificity balance

Suppose disease prevalence is 1% (i.e. prior probability of disease is 1/100)

If specificity and sensitivity are both equal to 99% (i.e. probability 99/100)
then the posterior probability of a person testing positive actually having the disease is 0.5 (=50%) (this is also called the **positive predictive value)**.

So out of 100 people testing positive, 50 will actually have the disease.

In general if disease prior=*d* and specificity and sensitivity both equal to 1-*d* then the posterior probability of a person testing positive actually having the disease is 0.5

$$P(H \mid E) = \frac{P(E \mid H) \times P(H)}{P(E \mid H) \times P(H) + P(E \mid \text{not } H) \times P(\text{not } H)}$$

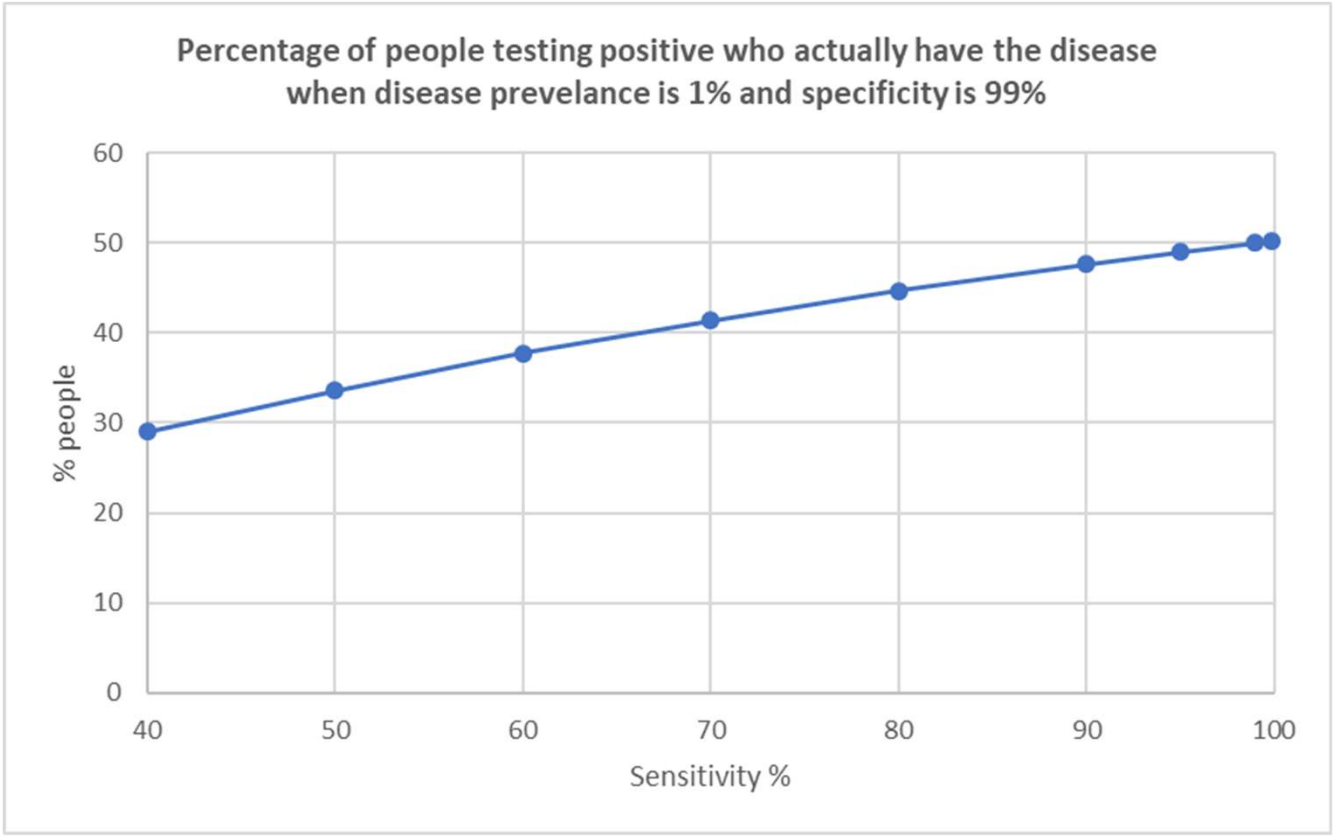
$$P(H \mid E) = \frac{(1-d) \times d}{((1-d) \times d) + (d \times (1-d))} = \frac{1}{2}$$

Sensitivity does not make a big difference

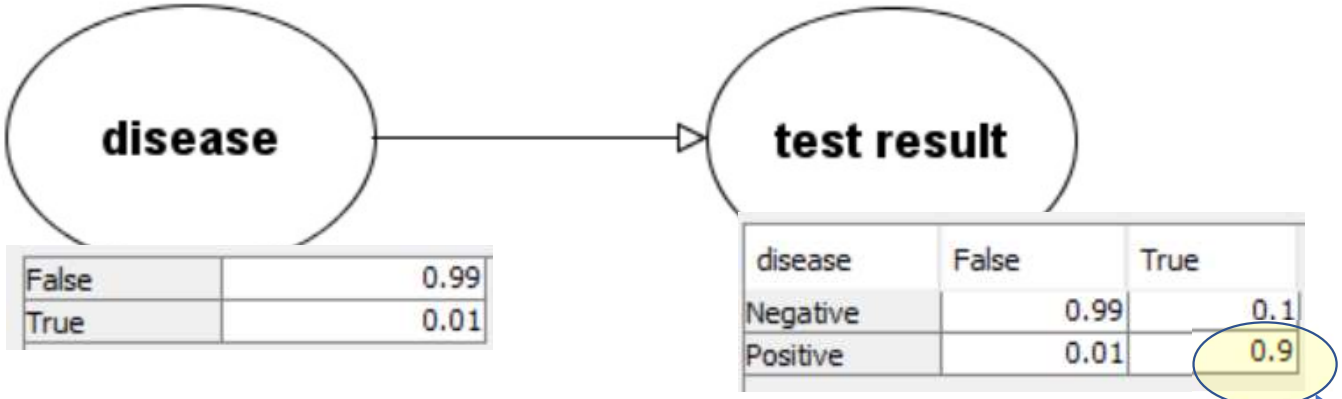
Using the 1% disease prevalence and 99% specificity, what is the probability a patient testing positive has the disease?

29% when SENSITIVITY = 40%
34% when SENSITIVITY = 50%
38% when SENSITIVITY = 60%
41% when SENSITIVITY = 70%
45% when SENSITIVITY = 80%
48% when SENSITIVITY = 90%
49% when SENSITIVITY = 95%
50% when SENSITIVITY = 99%
50.2% when SENSITIVITY = 99.9%

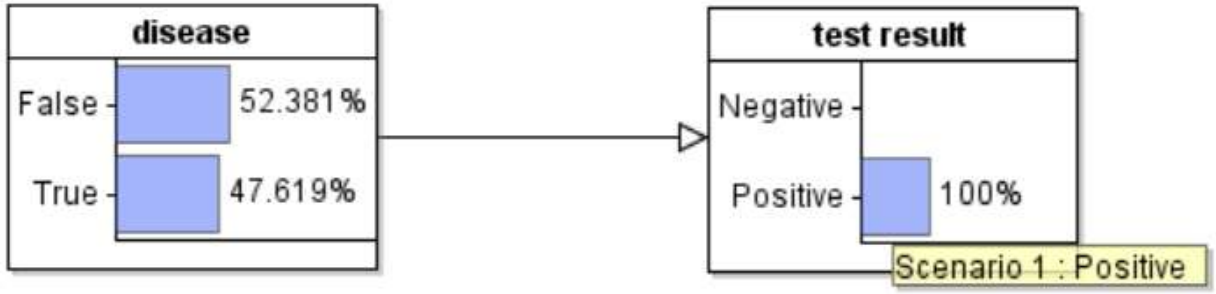
Law of diminishing returns



Experiment in AgenaRisk

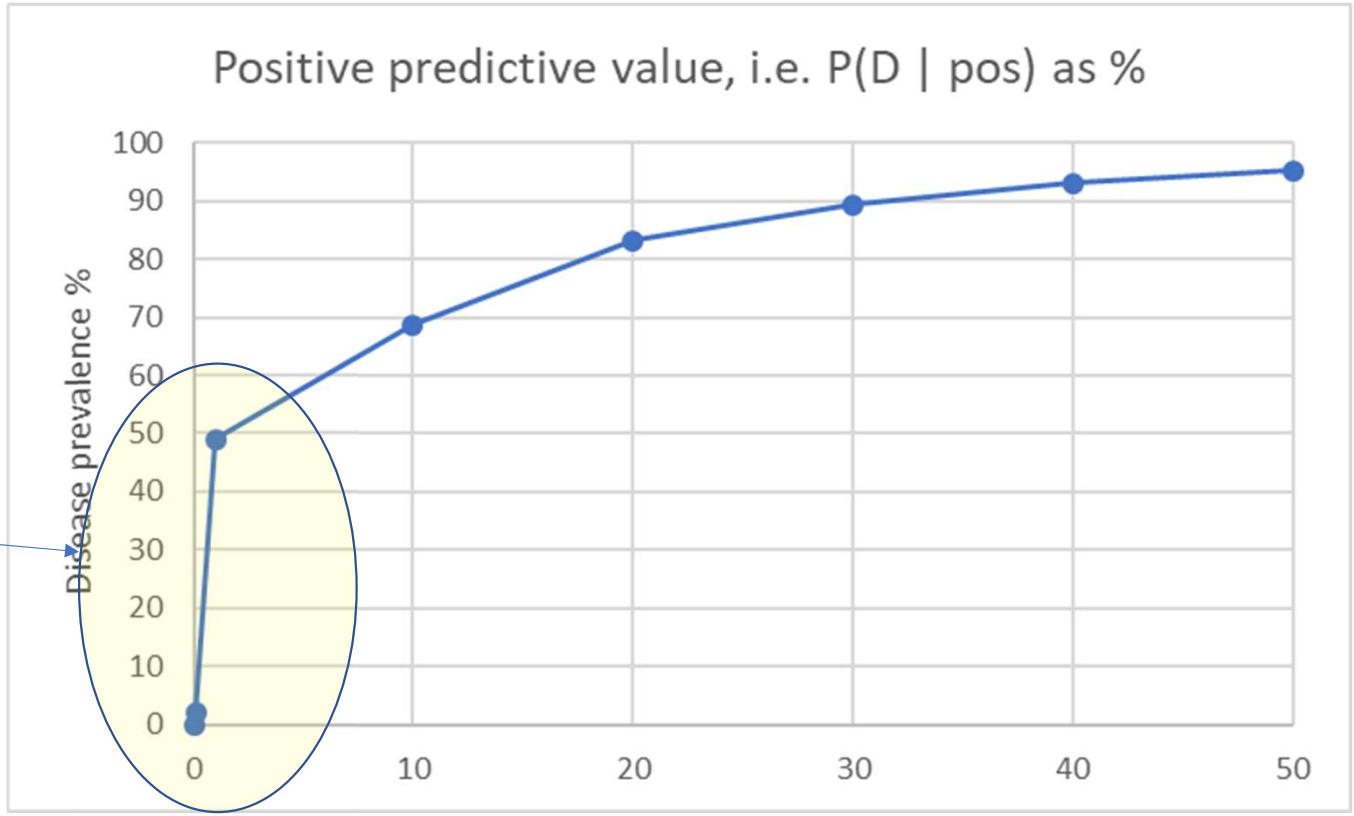


This is the sensitivity value to play around with



Positive predicted value for different disease prevalence rates

Despite high test accuracy the positive predictive value is very low when prevalence rates are below 50%



Assumes:
99% sensitivity and
95% specificity

Quiz Question

An algorithm is developed to recognize whether an image is that of a dog
The test data consists of 100 images
of which 20 are dogs

The algorithm sensitivity (true positive rate) is 95%

Its specificity (true negative rate) is 70%

Question: What is the overall accuracy of this algorithm?

	Number predicted YES	Number predicted NO	Total
Number YES's	*A*	*B*	*A+B*
Number NO's	*C*	*D*	*C+D*

Sensitivity = *A*/(*A*+*B*)

Specificity = *D*/(*C*+*D*)

Accuracy = (*A*+*D*)/(*A*+*B*+*C*+*D*)

Accounting for confidence of predictions

In the Titanic training data 74% of females survived and 19% of males survived.

We can use this information to express the predicted outcome as a probability in our simple algorithm

If x is MALE then p(x survived)=0.19
If x is FEMALE then p(x survived) = 0.74

The probabilistic algorithm is turned back into a binary classification by defining a "cut off point" probability c which must be exceeded for a person to be predicted to survive.

Suppose c = 0.5. Then every female is predicted to survive and every male is predicted not to survive (as in the original algorithm) and we already saw that (because in the test data 66% females survived and 20% males survived) the sensitivity is again 66% and it specificity is 80%.

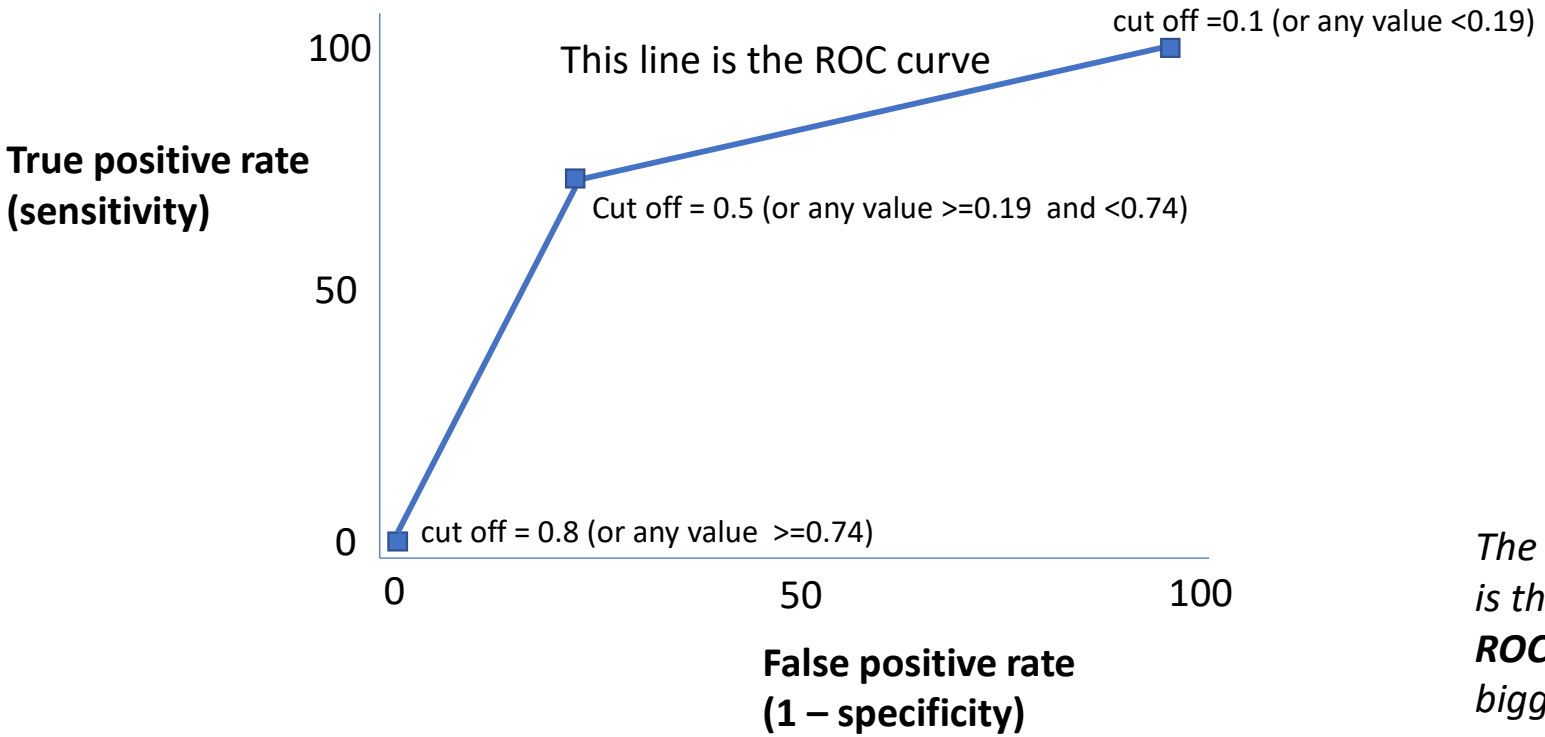
Any value of c which is at least 0.19 and less than 0.74 will have the same sensitivity (66%) and specificity (80%).

We can improve on the sensitivity by choosing c to be any value less than or equal to 0.19. In that case every person is predicted to survive. ***This is perfect sensitivity (100%) … but terrible specificity (0%).***

We can improve on the specificity by choosing c to be any value greater than or equal to 0.74. then every person is predicted not to survive. ***This is perfect specificity (100%)….terrible sensitivity (0%).***

Clearly, the accuracy of the algorithm depends on its accuracy for different cut-off points

ROC (Receiver operating characteristic) curve



*The measure of accuracy is the **area under the ROC curve (AUC)**. The bigger the area the more accurate the algorithm*

A useless algorithm

Randomly predicts whether or not a passenger survives based on a coin toss:

For every passenger x
If toss =“Heads”: predict survive
If toss = “Tails” predict not survive

In the test data 150 people survived and 259 did not survive, so assume 50% in each group toss Heads we get:

	Number predicted YES	Number predicted NO
Number YES’s	***75***	***129***
Number NO’s	***75***	***130***

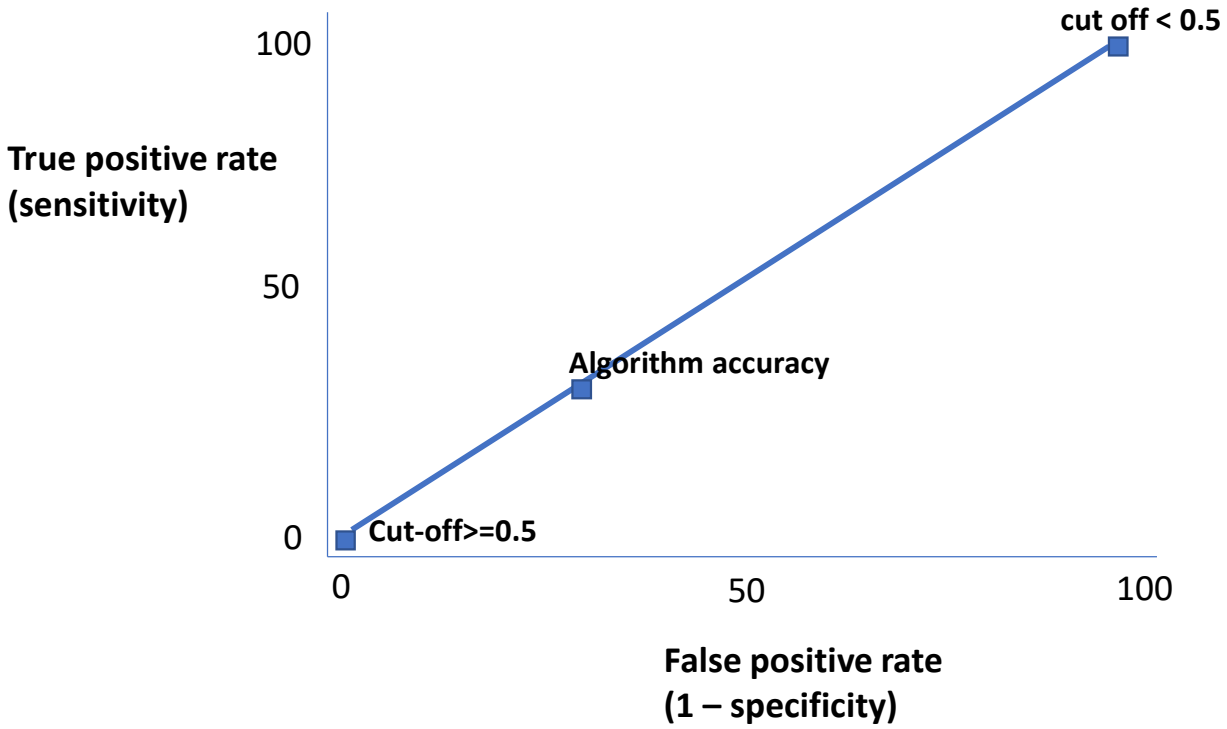
Sensitivity = *37%*

Specificity = *63%*

The probabilistic version of this algorithm is:

For every passenger x
P(x survives)=0.5

ROC curve for useless algorithm



Note that a ‘straight line’ like this represents the least informative type of algorithm

ROC curve: Diagnostic test

Consider our test for the disease D which has sensitivity 99% and specificity 95%.
Imagine a simple algorithm defined as:

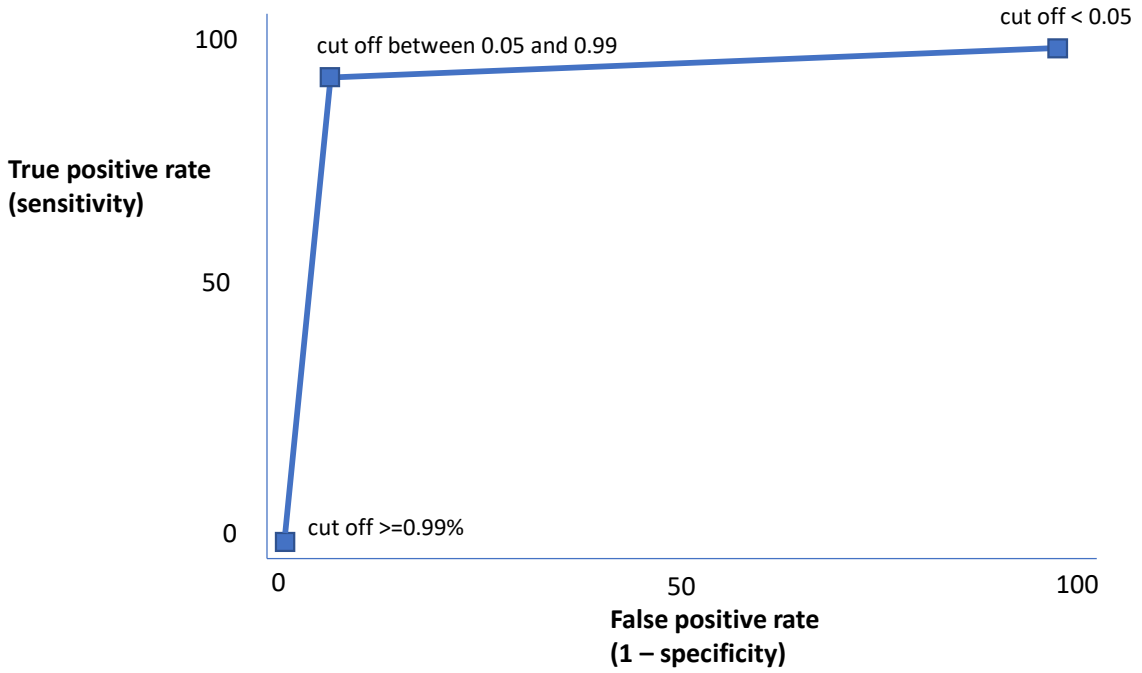
If test is positive then P(D)=0.99
else P(D)=0.05

Patient	Test Result	Prediction of D
1	Positive	99%
2	Negative	5%
3	Negative	5%
4	Positive	99%
…	….	….

If the cut-off is at least 0.99 then nobody is classified as having D (0% true positives but 0% sensitivity).

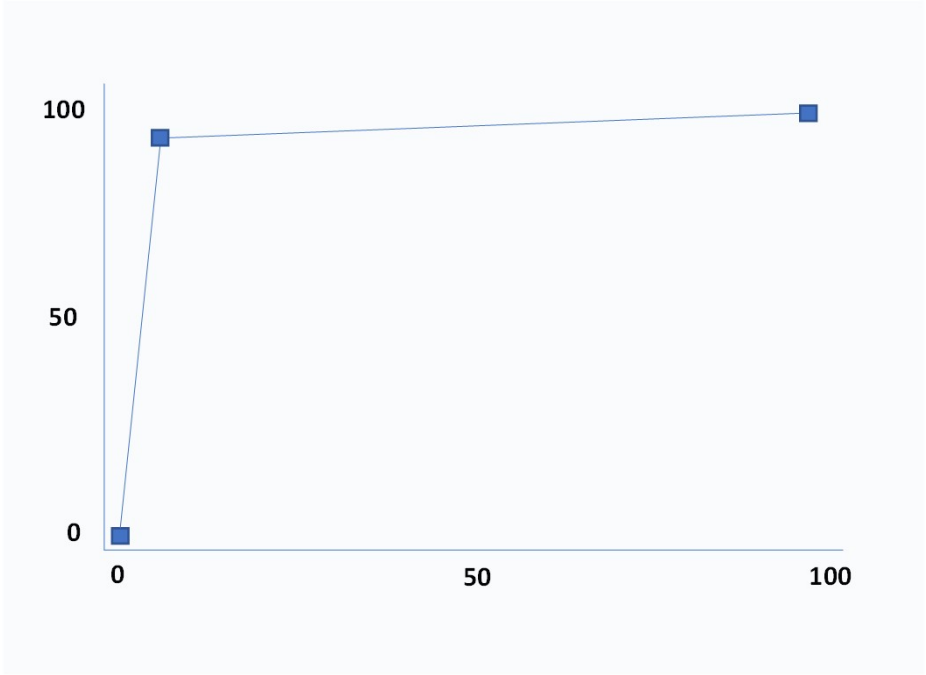
If the cut-off is anything between 0.05 and 0.99 then 99% of people with D will be correctly classified as having D. But there will also be 5% false positives

If the cut-off is anything less than 0.05 then everybody will be classified as having D so 100% sensitivity but also 100% false positives



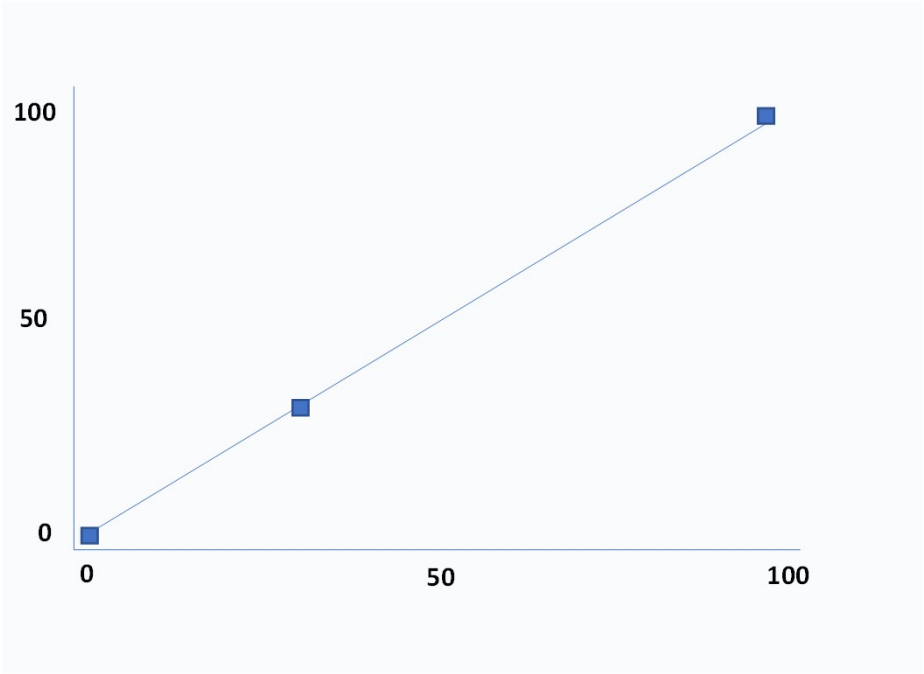
Compare the two ROC curves

Accurate model



Area under the curve is close to 1

Useless coin toss model

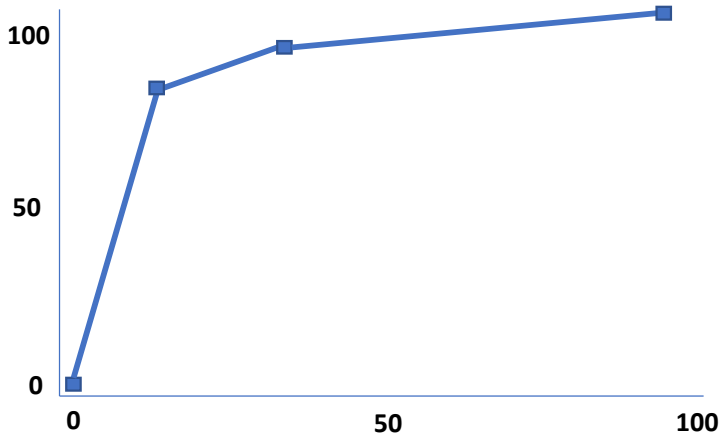


Area under the curve is close to 0.5

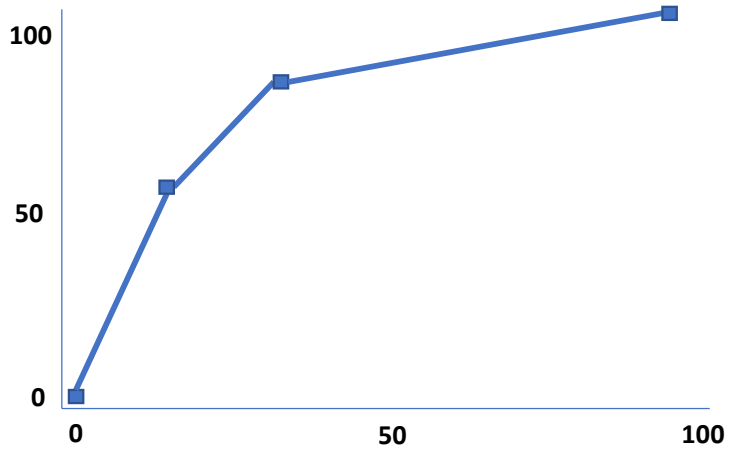
Hence area under the curve is a standard summary measure of model accuracy

Quiz Question: Which ROC curve represents the most accurate algorithm?

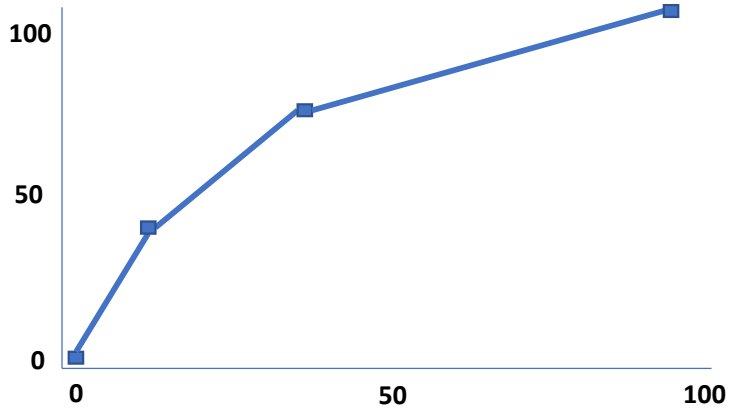
A



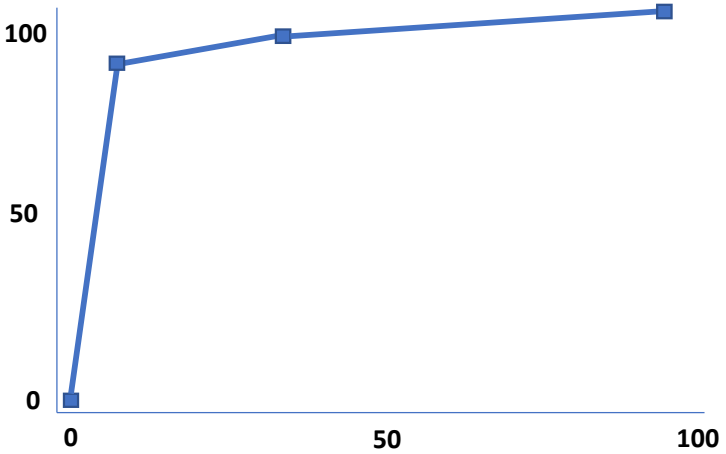
B



C



D



Quiz question

In the Training data 39% of the passengers survived.
So a naïve algorithm could simply randomly assign 39% of the test data passengers to survive.

What is the sensitivity and specificity of this algorithm (recall that in the Test data 150 out of 409 survived)?

What is the probabilistic version of this algorithm?

Draw the ROC curve for this algorithm

ROC curve example: Predicting rain tomorrow

Our model is based on the type of weather today and the following data

Weather on any day	Probability %	Probability rain next day
No rain	70	25
Light rain	15	30
Heavy rain	10	50
Storm	5	60

Algorithm to predict rain next day

- If no rain today predict probability 25%
- If light rain today predict probability 30%
- If heavy rain today predict probability 50%
- If storm today predict probability 60%

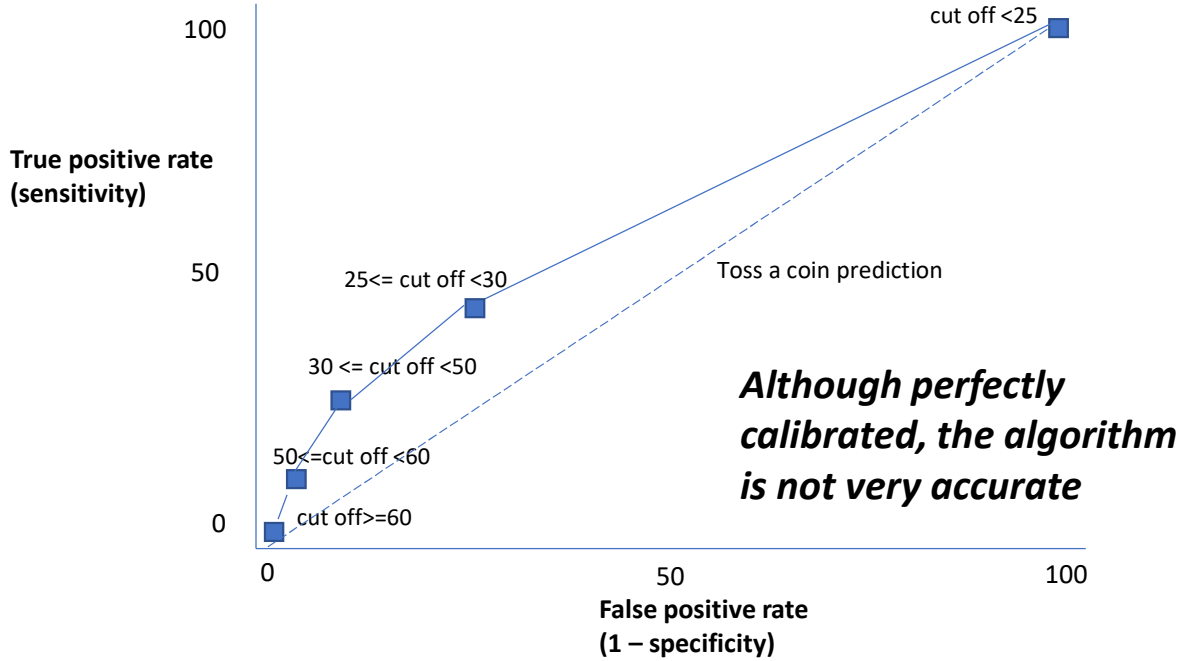
If cut-off >= 60% then we never predict rain so 0% true positives, 0% false positive.

If 50%< =cut-off < 60% then 10% true positives, 3% false positive.

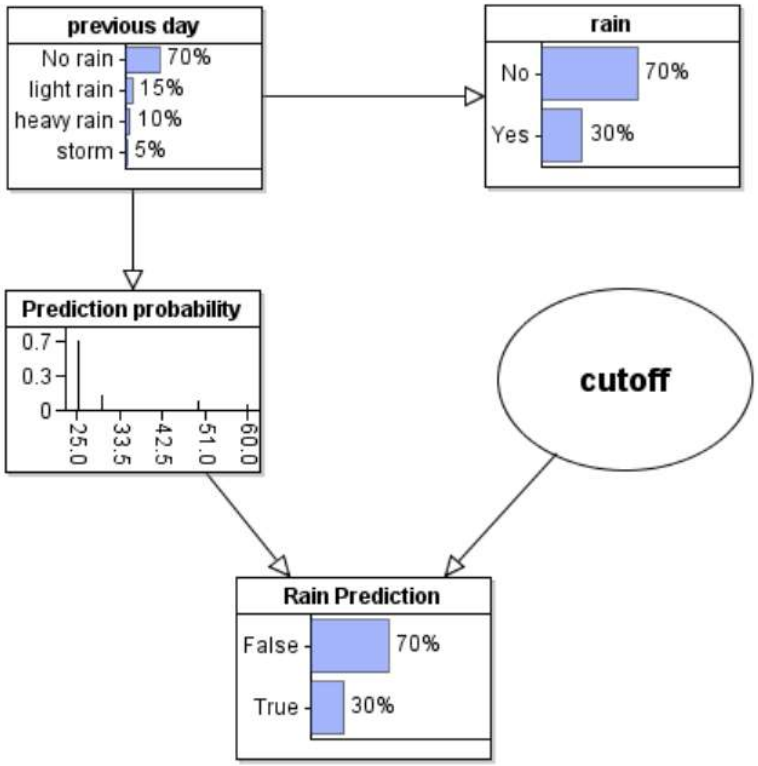
If 30%<= cut-off < 50% then 27% true positives, 10% false positive.

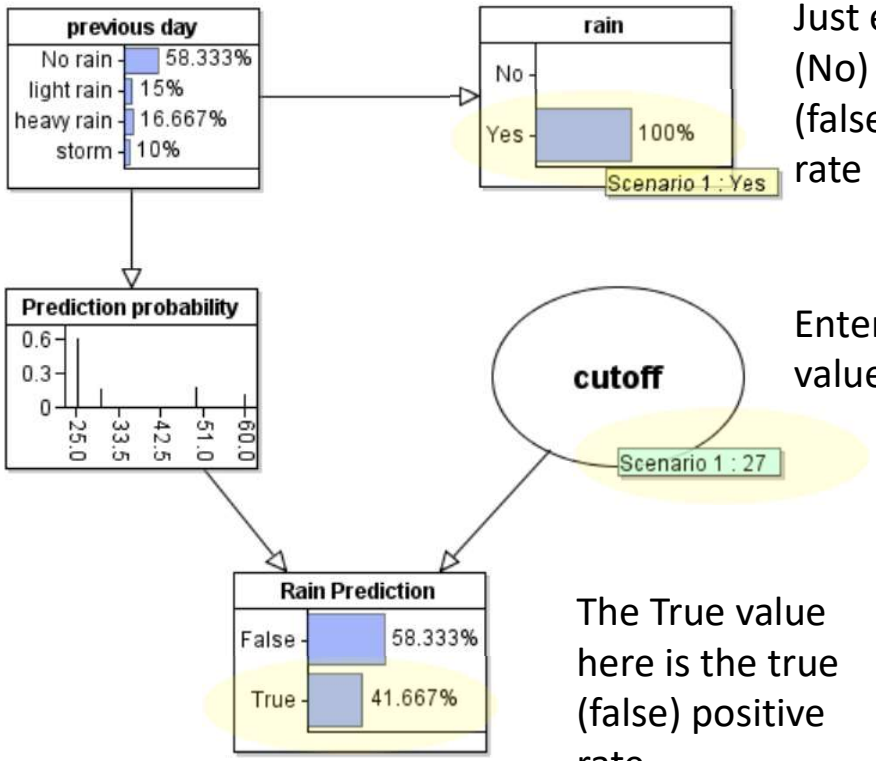
If 25%<=cut-off < 30% then 42% true positives, 25% false positive.

If cut-off <25% then we always predict rain so 100% true positives, 100% false positive.



How did I calculate the false positives/negatives?





Just enter Yes (No) to get true (false) positive rate

Enter any cutoff value

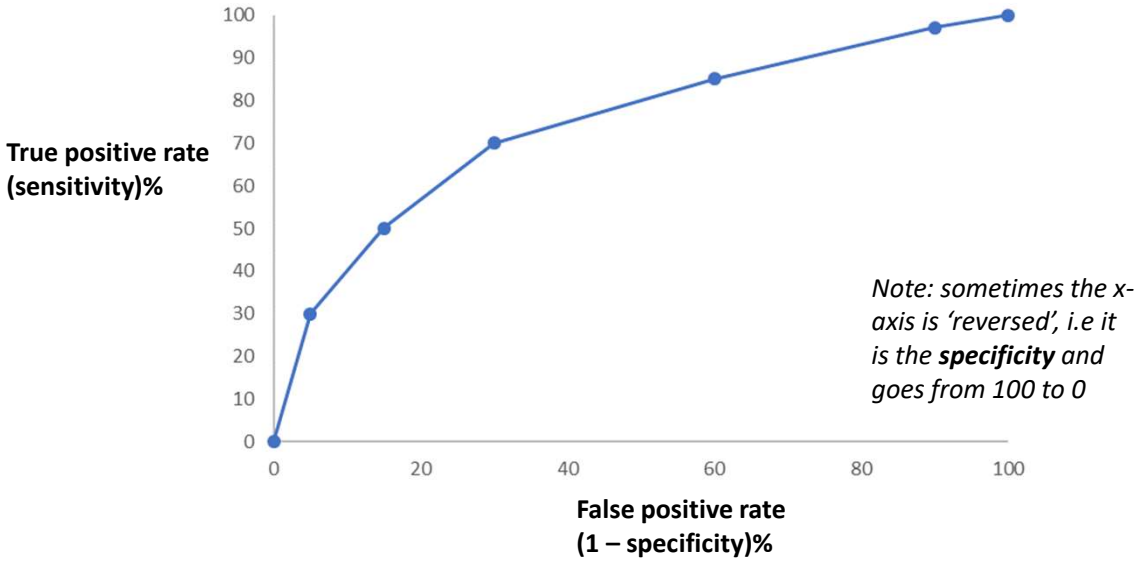
The True value here is the true (false) positive rate

ROC curve summary

Used to measure overall accuracy of any (binary) classification system/model (i.e. one that tries to predict whether an input belongs to class *D* or class *not D)* where the system produces a probability of the input being a *D* or not.

Whether or not we actually classify the input as a *D* or not depends on which probability 'cut-off' point we use. For example if the cut-off point is 90% then only inputs for which the model determines the probability of *D* ≥ 90% will be classified as being *D*.

The ROC is simply a plot of the true positive rate (sensitivity) of the classification on the y-axis against the false positive rate (1 – specificity)% on the x-axis ***for a set of cut-off points between 0 and 100%***

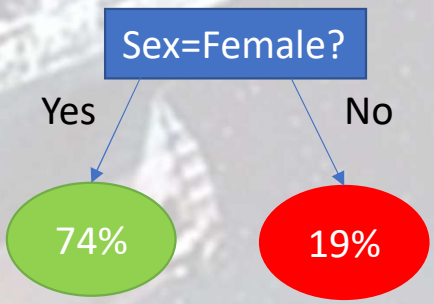


Learning method: Classification Trees

Series of Yes/No questions (e.g. for TITANIC data: “Male?”, “1st Class” etc.)

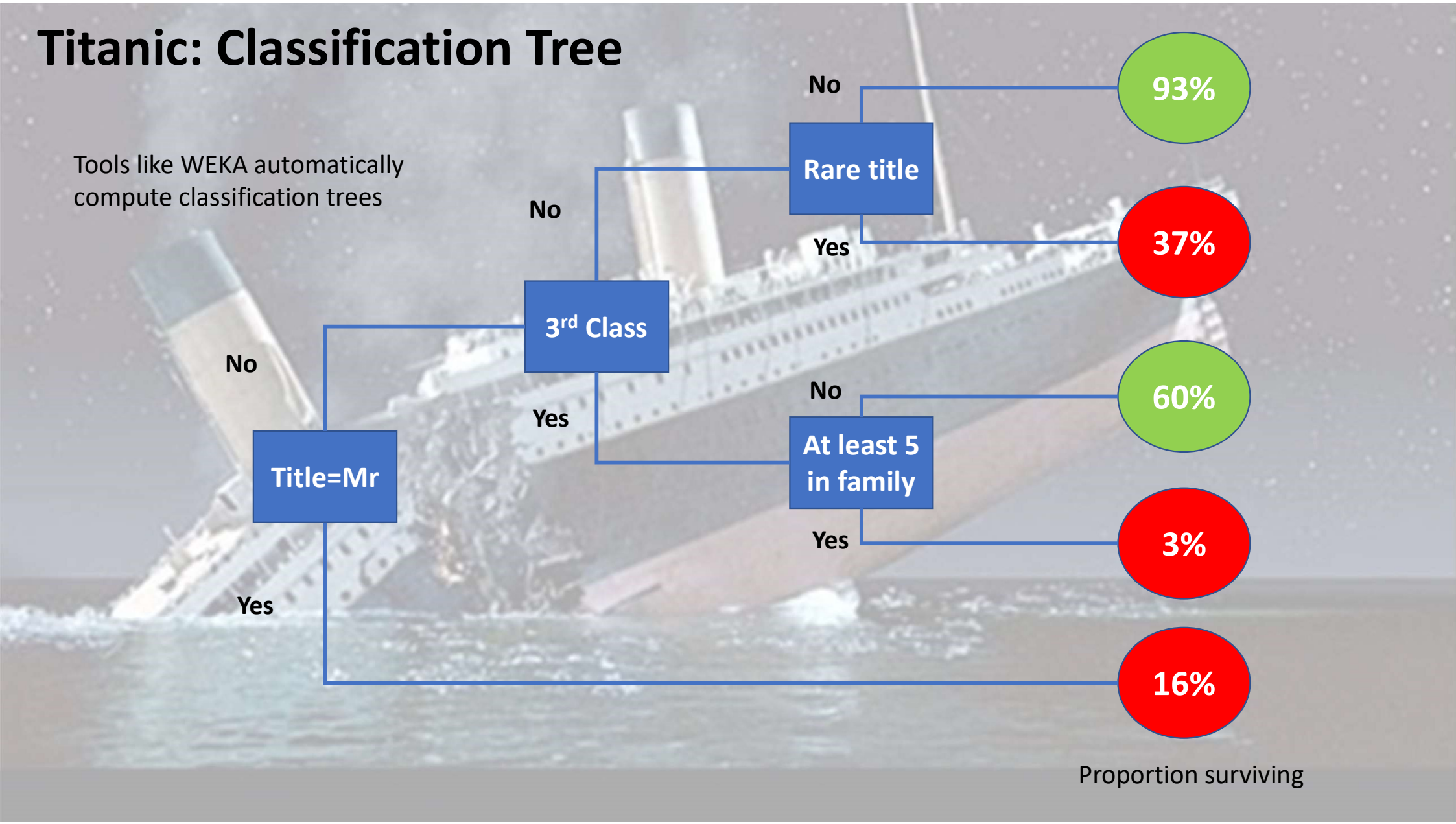
Answer to each decides next question to be asked until no further questions are needed.

Each of the training data records is assigned to the ‘branch’ which correctly classifies it, and for each branch we assign the proportion of those records for which the outcome was YES (for TITANIC this means the proportion classified by that branch who survived).

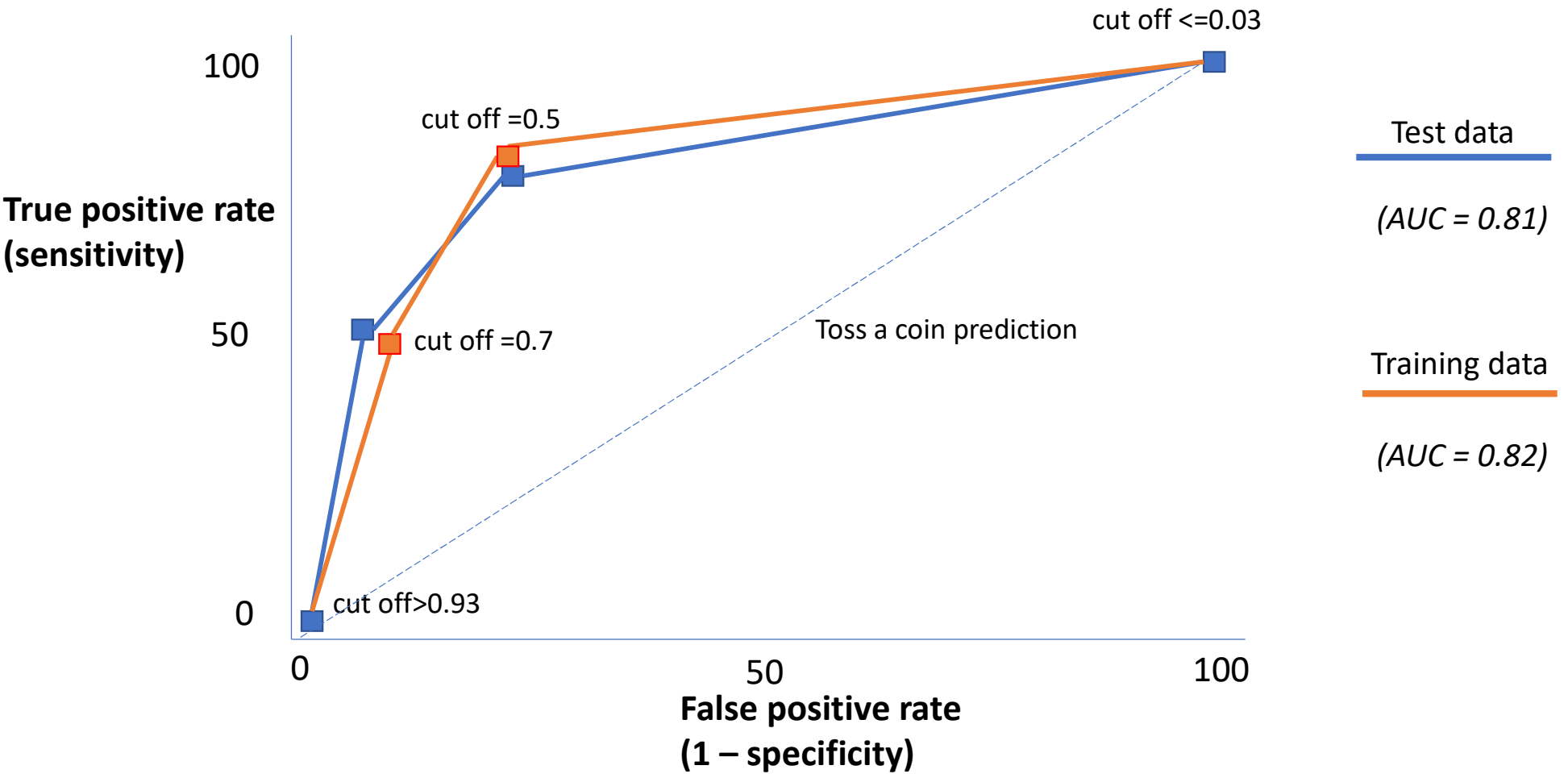


Titanic: Classification Tree

Tools like WEKA automatically compute classification trees

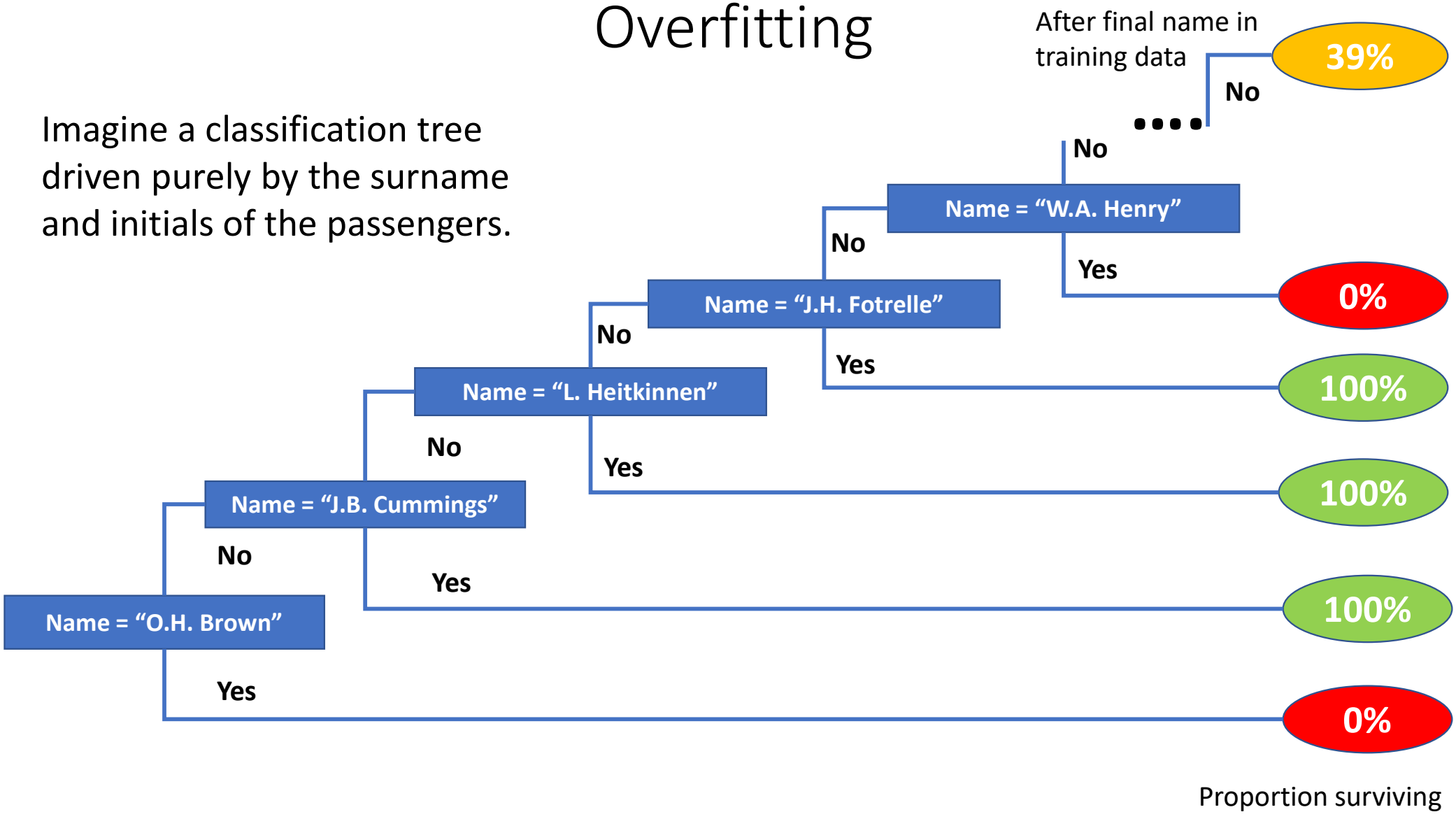


ROC curve for Titanic Classification Tree Algorithm



Overfitting

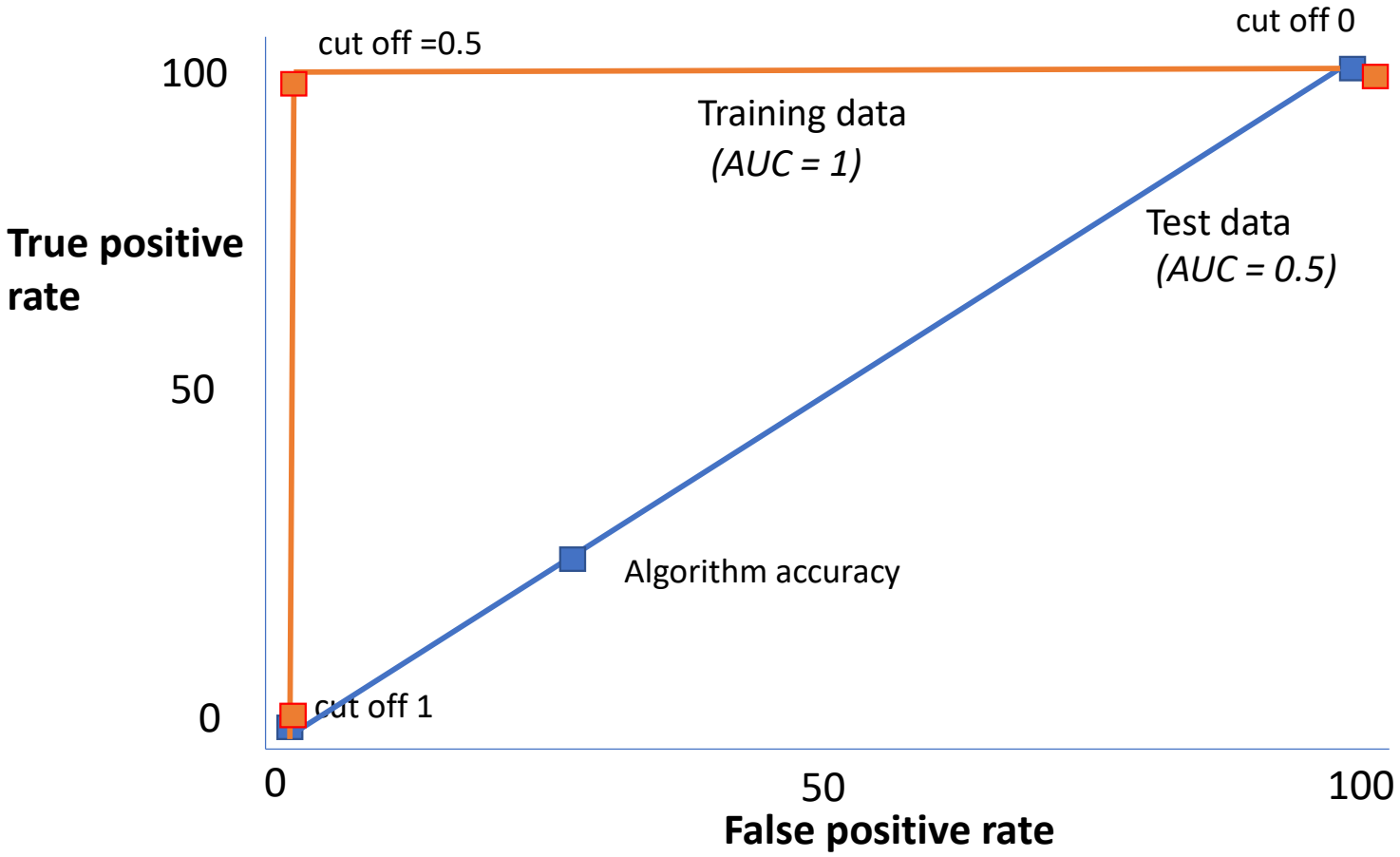
Imagine a classification tree driven purely by the surname and initials of the passengers.



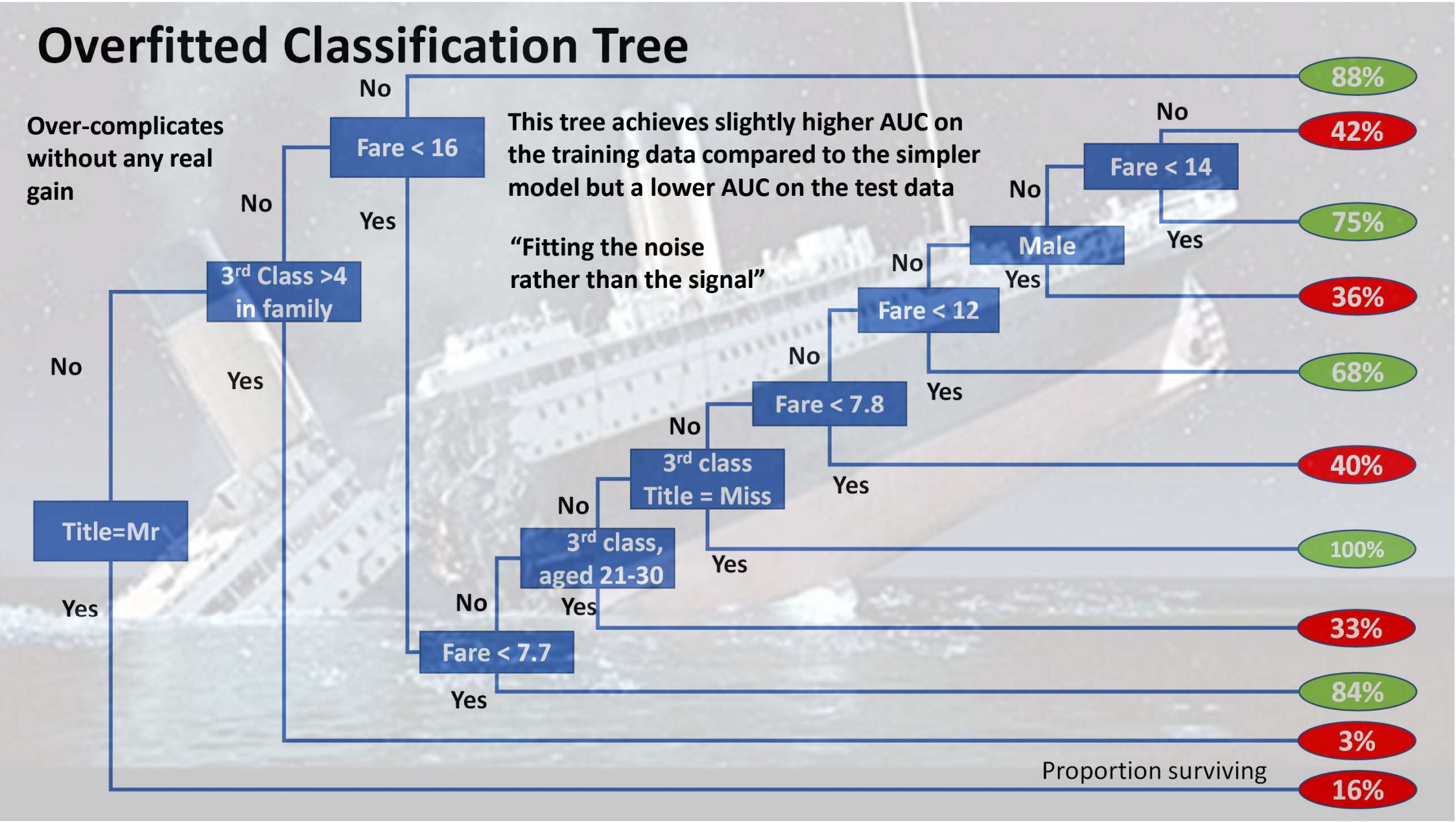
Overfitting

Outcome for every passenger in the training set is predicted perfectly for any cut-off between 0 and 1

But for every passenger in the test set the predicted probability of survival is 39%



Overfitted Classification Tree



Learning method: Logistic Regression

A	B	C	D	E	F	
Pclass	**Sex**	**Age**	**SibSp**	**Parch**	**Fare**	**Survived**
3	male	22	1	0	£7.25	0
1	female	38	1	0	£71.28	1
3	female	26	0	0	£7.93	1
1	female	35	1	0	£53.10	1
3	male	35	0	0	£8.05	0
3	male		0	0	£8.46	0
1	male	54	0	0	£51.86	0
3	male	2	3	1	£21.08	0
3	female	27	0	2	£11.13	1
2	female	14	1	0	£30.07	1
3	female	4	1	1	£16.70	1
1	female	58	0	0	£26.55	1
3	male	20	0	0	£8.05	0
3	male	39	1	5	£31.28	0
3	female	14	0	0	£7.85	0
2	female	55	0	0	£16.00	1
3	male	2	4	1	£29.13	0
2	male		0	0	£13.00	1
3	female	31	1	0	£18.00	0
3	female		0	0	£7.23	1
2	male	35	0	0	£26.00	0
2	male	34	0	0	£13.00	1

Convert textual classifications into numeric (e.g male=0; female=1)

As with standard (linear) regression a best fit 'line' is computed with the form

$$R = k + aA + bB + cC + dD + eF$$

where *k* is the intercept, and *a, b, c, d, e, f* are the constant coefficients ('weights') associated with the respective attributes
But this regression equation *R* corresponds to

$$log_2 \frac{p}{1-p}$$

where *p* is the predicted probability the outcome is true (=1)
So we can rearrange the formula to compute *p* as:

$$p = \frac{1}{1 + 2^{-R}}$$

Excel add-in automatically computes the logistic regression coefficients

Logistic regression example

Pclass	Sex	Survived	R	p
3	0	0	-1.77	0.23
1	1	1	2.51	0.85
3	1	1	0.45	0.58
1	1	1	2.51	0.85
3	0	0	-1.77	0.23
3	0	0	-1.77	0.23
1	1	0	2.51	0.85
3	0	0	-1.77	0.23
3	1	1	0.45	0.58
2	1	1	1.48	0.74
3	1	1	0.45	0.58
1	1	1	2.51	0.85
3	0	0	-1.77	0.23
3	0	0	-1.77	0.23
3	1	0	0.45	0.58
2	1	1	1.48	0.74
3	0	0	-1.77	0.23
2	0	1	-0.74	0.37
3	1	0	0.45	0.58
3	1	1	0.45	0.58
2	0	0	-0.74	0.37
2	0	1	-0.74	0.37

Excel add-in automatically computes the logistic regression coefficients:

	coeff b
Intercept	1.318232
Pclass	-1.02621
Sex	2.215847

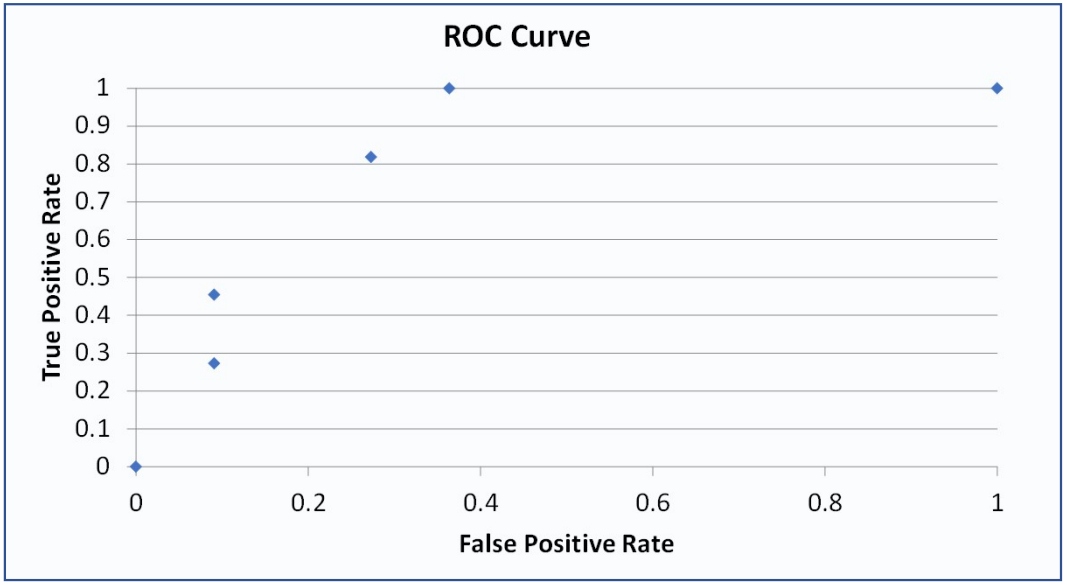
Hence we get the logistic regression equation *R*

$$R = 1.318 - 1.026\,Pclass + 2.216\,Sex$$

And hence we can compute *p*

$$p = \frac{1}{1 + 2^{-R}}$$

Logistic regression example



The tool automatically produces the ROC curve

And all of the data relating to accuracy

Classification Table					ROC Table							
	Suc-Obs	Fail-Obs			*p-Pred*	*Failure*	*Success*	*Fail-Cum*	*Suc-Cum*	*FPR*	*TPR*	*AUC*
Suc-Pred	9	3	12					0	0	1	1	0.636364
Fail-Pred	2	8	10		0.146739	7	0	7	0	0.363636	1	0.090909
	11	11	22		0.324275	1	2	8	2	0.272727	0.818182	0.14876
					0.611932	2	4	10	6	0.090909	0.454545	0
Accuracy	0.818182	0.727273	0.772727		0.814822	0	2	10	8	0.090909	0.272727	0.024793
					0.924691	1	3	11	11	0	0	0
Cutoff	0.5											0.900826

Quiz Question

The data here (which is also in a spreadsheet you can download on QMPlus) shows the relationship between number of hours a student revises for a test and whether or not they pass (1 = Pass, 0 = Fail)

The logistic regression equation for Pass had just two parameters: the 'intercept' and the 'Hours' coefficient. What are they?

Hours	Pass
0.5	0
0.75	0
1	0
1.25	0
1.5	0
1.75	0
1.75	1
2	0
2.25	1
2.5	0
2.75	1
3	0
3.25	1
3.5	0
4	1
4.25	1
4.5	1
4.75	1
5	1
5.5	1

Logistic regression example

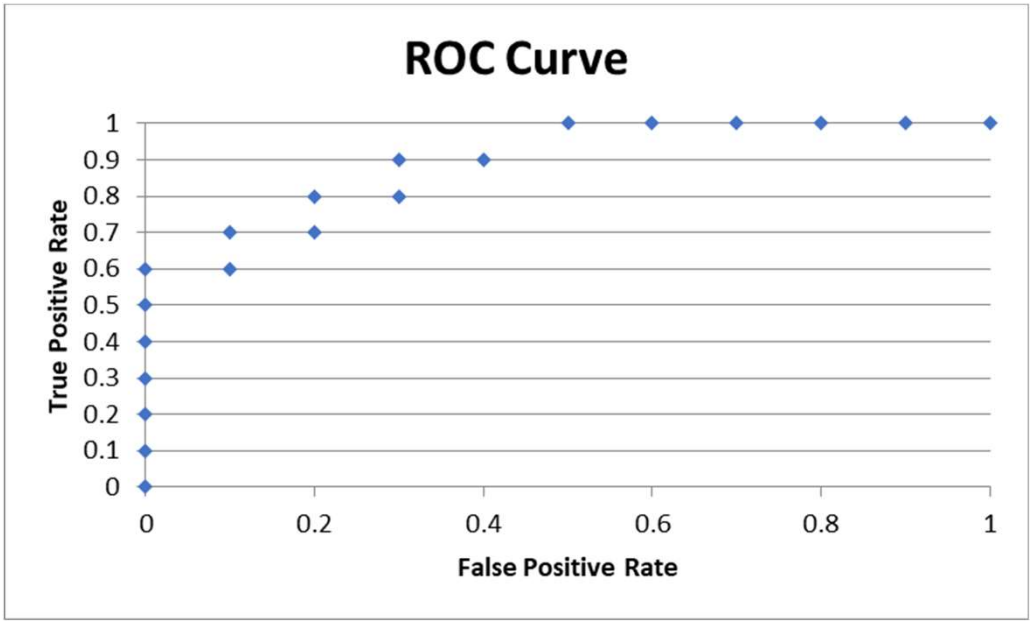
Hours	Pass
0.5	0
0.75	0
1	0
1.25	0
1.5	0
1.75	0
1.75	1
2	0
2.25	1
2.5	0
2.75	1
3	0
3.25	1
3.5	0
4	1
4.25	1
4.5	1
4.75	1
5	1
5.5	1

Prediction if student passes course based on hours of revision

$$log_2 \frac{p}{1-p} = b_0 + b_1 Hours$$

$$p = \frac{1}{1 + 2^{-(b_0 + b_1 Hours)}} = \frac{1}{1 + 2^{-(-4.078 + 1.5 Hours)}}$$

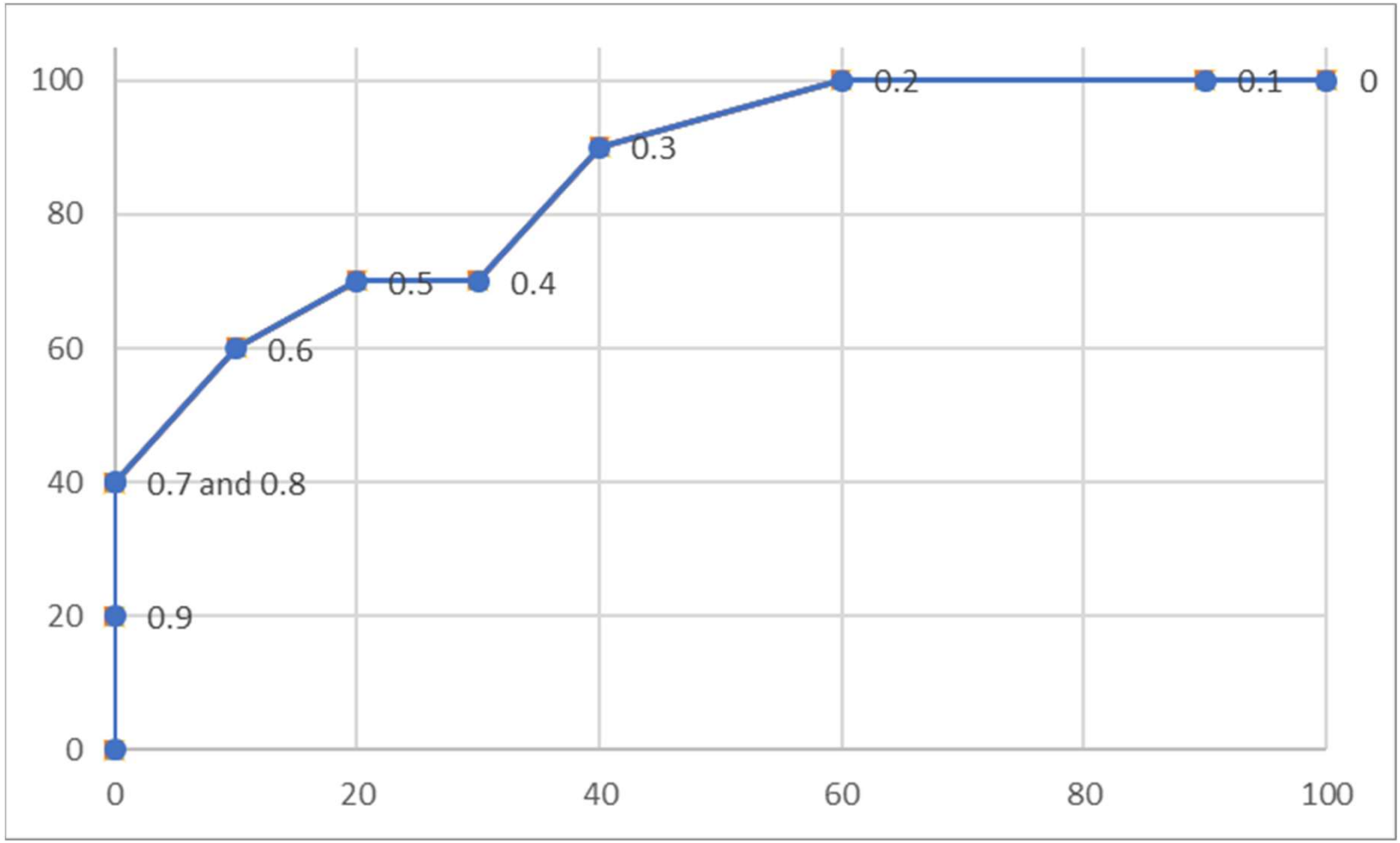
	coeff b
Intercept	-4.07771
Hours	1.504645



Student course pass example: full analysis for manual ROC

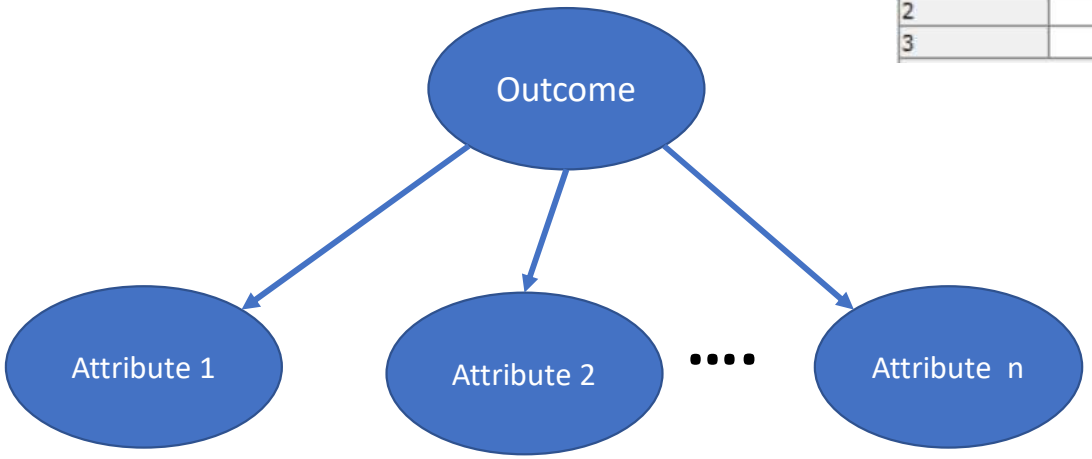
				CUT-OFF										
Hours	***Pass***		***LogReg predict***	***0***	***0.1***	***0.2***	***0.3***	***0.4***	***0.5***	***0.6***	***0.7***	***0.8***	***0.9***	***1***
0.5	0		0.090562	FP	TN	TN	TN	TN	TN	TN	TN	TN	TN	TN
0.75	0		0.11437	FP	FP	TN	TN	TN	TN	TN	TN	TN	TN	TN
1	0		0.143449	FP	FP	TN	TN	TN	TN	TN	TN	TN	TN	TN
1.25	0		0.178433	FP	FP	TN	TN	TN	TN	TN	TN	TN	TN	TN
1.5	0		0.219759	FP	FP	FP	TN	TN	TN	TN	TN	TN	TN	TN
1.75	0		0.267539	FP	FP	FP	TN	TN	TN	TN	TN	TN	TN	TN
1.75	1		0.267539	TP	TP	TP	FN	FN	FN	FN	FN	FN	FN	FN
2	0		0.321429	FP	FP	TN	FP	TN	TN	TN	TN	TN	TN	TN
2.25	1		0.380534	TP	TP	TP	TP	FN	FN	FN	FN	FN	FN	FN
2.5	0		0.443405	FP	FP	FP	FP	FP	TN	TN	TN	TN	TN	TN
2.75	1		0.508144	TP	TP	TP	TP	TP	TP	FN	FN	FN	FN	FN
3	0		0.57261	FP	FP	FP	FP	FP	FP	TN	TN	TN	TN	TN
3.25	1		0.634701	TP	TP	TP	TP	TP	TP	TP	FN	FN	FN	FN
3.5	0		0.692614	FP	FP	FP	FP	FP	FP	FP	TN	TN	TN	TN
4	1		0.791209	TP	TP	TP	TP	TP	TP	TP	FN	FN	FN	FN
4.25	1		0.83092	TP	TP	TP	TP	TP	TP	TP	TP	TP	FN	FN
4.5	1		0.864372	TP	TP	TP	TP	TP	TP	TP	TP	TP	FN	FN
4.75	1		0.892066	TP	TP	TP	TP	TP	TP	TP	TP	TP	FN	FN
5	1		0.914663	TP	TP	TP	TP	TP	TP	TP	TP	TP	TP	FN
5.5	1		0.94744	TP	TP	TP	TP	TP	TP	TP	TP	TP	TP	FN
				10/10 FP	9/10 FP	6/10 FP	4/10 FP	3/10 FP	2/10 FP	1/10 FP	0/10 FP	0/10 FP	0/10 FP	0/10 FP
				10/10 TP	10/10 TP	10/10 TP	9/10 TP	7/10 TP	7/10 TP	6/10 TP	4/10 TP	4/10 TP	2/10 TP	0/10 TP

Student course pass example: full analysis for manual ROC



Learning method: Naïve Bayes

Assumes the ‘naïve’ causal structure

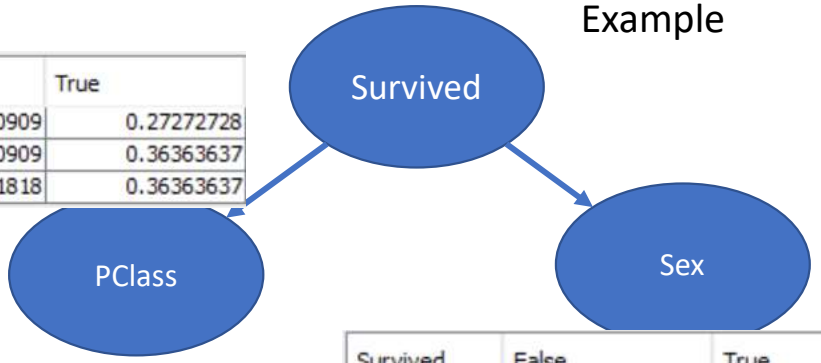


Estimate prob(Attribute *i* | Outcome) from the training data

Then compute
prob (Outcome | Attribute 1, Attribute 2, …, Attribute n)
using Bayes

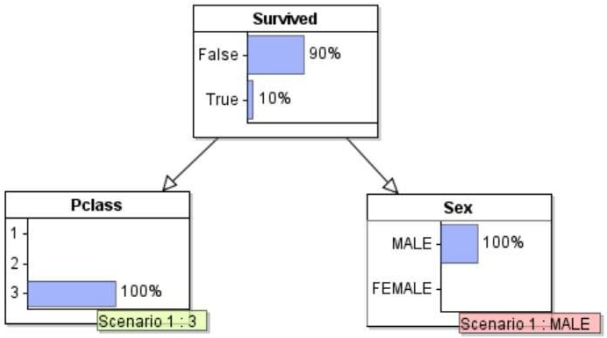
Example

Survived	False	True
1	0.09090909	0.27272728
2	0.09090909	0.36363637
3	0.8181818	0.36363637



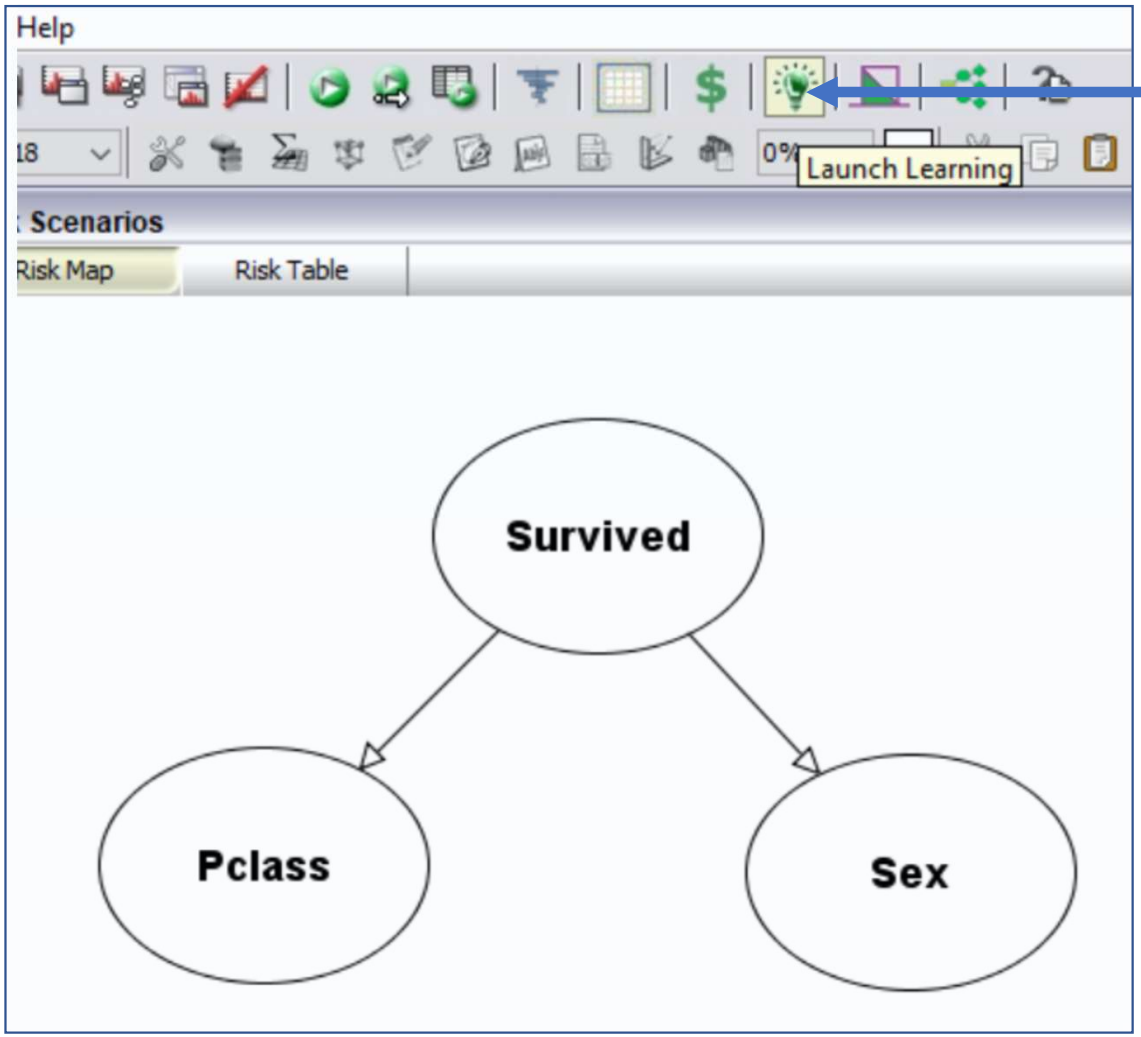
Survived	False	True
MALE	0.72727275	0.18181819
FEMALE	0.27272728	0.8181818

Estimate prob(PClass| Survived)
prob(Sex | Survived)



Learning method: Naïve Bayes

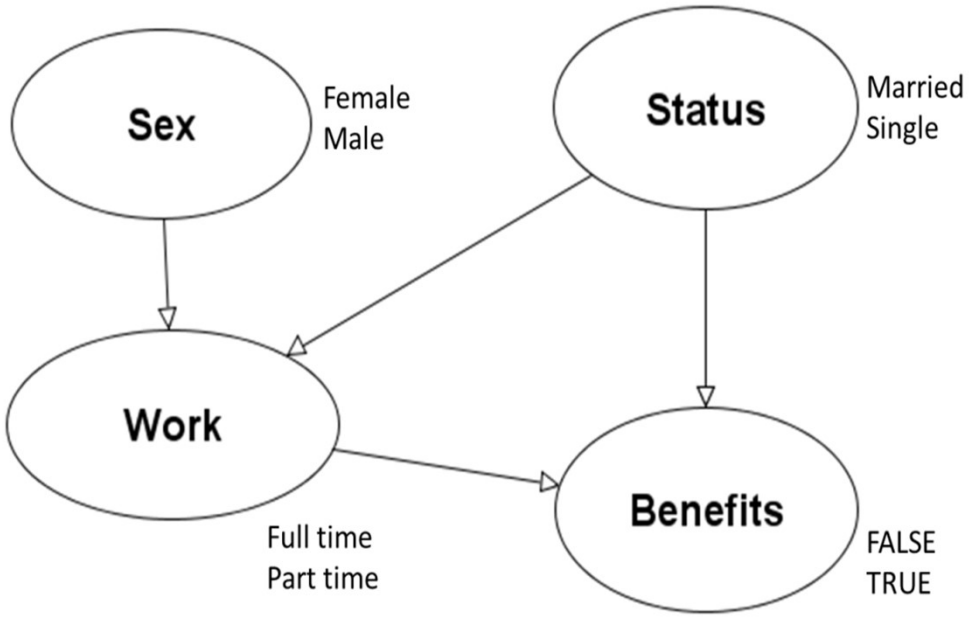
Done automatically in AgenaRisk once the graph structure is defined



Learning Tool

Opens a dialog that enables you to select a csv file with the data (assumes heading and state names are same as in the model)

Learning NPTs from Data alone



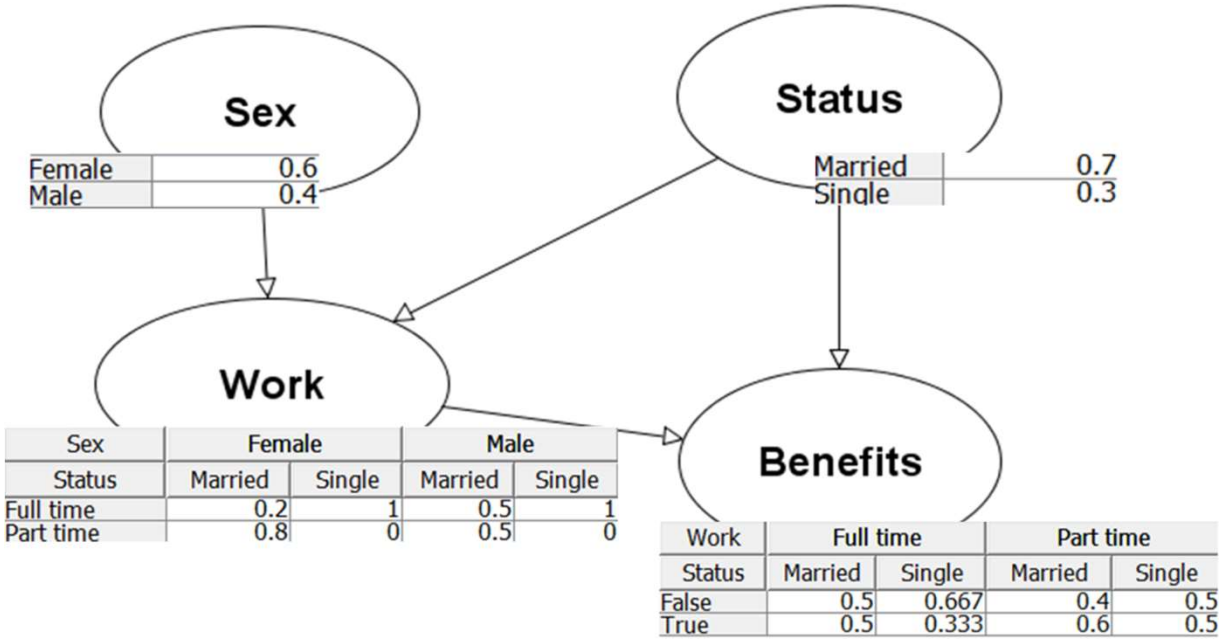
(Library Model: Book2\"17.2 Simple Sex benefits")

Sex	Status	Work	Benefits
Female	Single	Full time	FALSE
Female	Married	Part time	TRUE
Male	Single	Full time	FALSE
Female	Married	Part time	FALSE
Female	Married	Full time	FALSE
Male	Married	Part time	TRUE
Female	Married	Part time	FALSE
Male	Married	Full time	TRUE
Female	Married	Part time	TRUE
Male	Single	Full time	TRUE

(this dataset is in the same library folder)

Note we have no 'missing' data here

Learning NPTs from Data alone



Sex	
Female	0.6
Male	0.4

Status	
Married	0.7
Single	0.3

Sex	Female		Male	
Status	Married	Single	Married	Single
Full time	0.2	1	0.5	1
Part time	0.8	0	0.5	0

Work	Full time		Part time	
Status	Married	Single	Married	Single
False	0.5	0.667	0.4	0.5
True	0.5	0.333	0.6	0.5

The state combination (Part time, Single) is not observed at all in the data set.

We only have one observation for (Female, Single) the fact that this one observation is ‘Full time’ means that we have ‘learnt’ that 100% of single females work full time.

Types of algorithms for supervised learning from data

- Decision trees
- Logistic regression
- Naïve Bayes
- Full Bayesian network
- Random forests

  Large number of trees each producing a classification, with final classification decided by majority vote
- Support vector machines

  Find linear combinations of features that best split different outcomes
- K-nearest neighbours

  Classifies according to majority outcome among close cases in training set
- Neural networks

  Layers of nodes – each depending on previous layer by weights like a series of logistic regressions piled on top of each other. Weights learned by optimization procedure.

But beware

None of the methods are truly based purely on ‘machine learning’

We always must make decisions about:

- Which attributes to include/exclude
- How to ‘discretize’ the data

There is inevitably some ‘cleaning’ of the data and transforming state labels

We have assumed no ‘missing values’ in the data

Even if there are a LOT of training data, it is unlikely there will be much (or even ANY) data for all combinations of attributes state values

With sufficient data all of the methods achieve reasonable accuracy – and generally there is little to choose between them

None of the methods can learn about ‘causality’

Summary

- We have focused on supervised learning from data. This requires a training set where the outcome for each record is known. The challenge is to learn how to accurately predict the outcome from the attribute values and to test the accuracy of the predictions on a test dataset where the outcome is not given in advance.
- In contrast with unsupervised learning there is no given outcome and the challenge is to find relationships between records.
- Where the outcome is non-numeric we cannot use classical regression methods. We have focused on methods where the outcome is binary, i.e. binary classification methods.
- For binary classification methods, accuracy is measured by true and false positives (sensitivity and specificity).
- Generally it is better to predict the probability of an outcome and in such cases the accuracy of an algorithm is measured by the AUC - area under the ROC curve.
- There are many different pure machine learning methods. But they produce results of similar accuracy and all suffer the same weaknesses

Additional Material