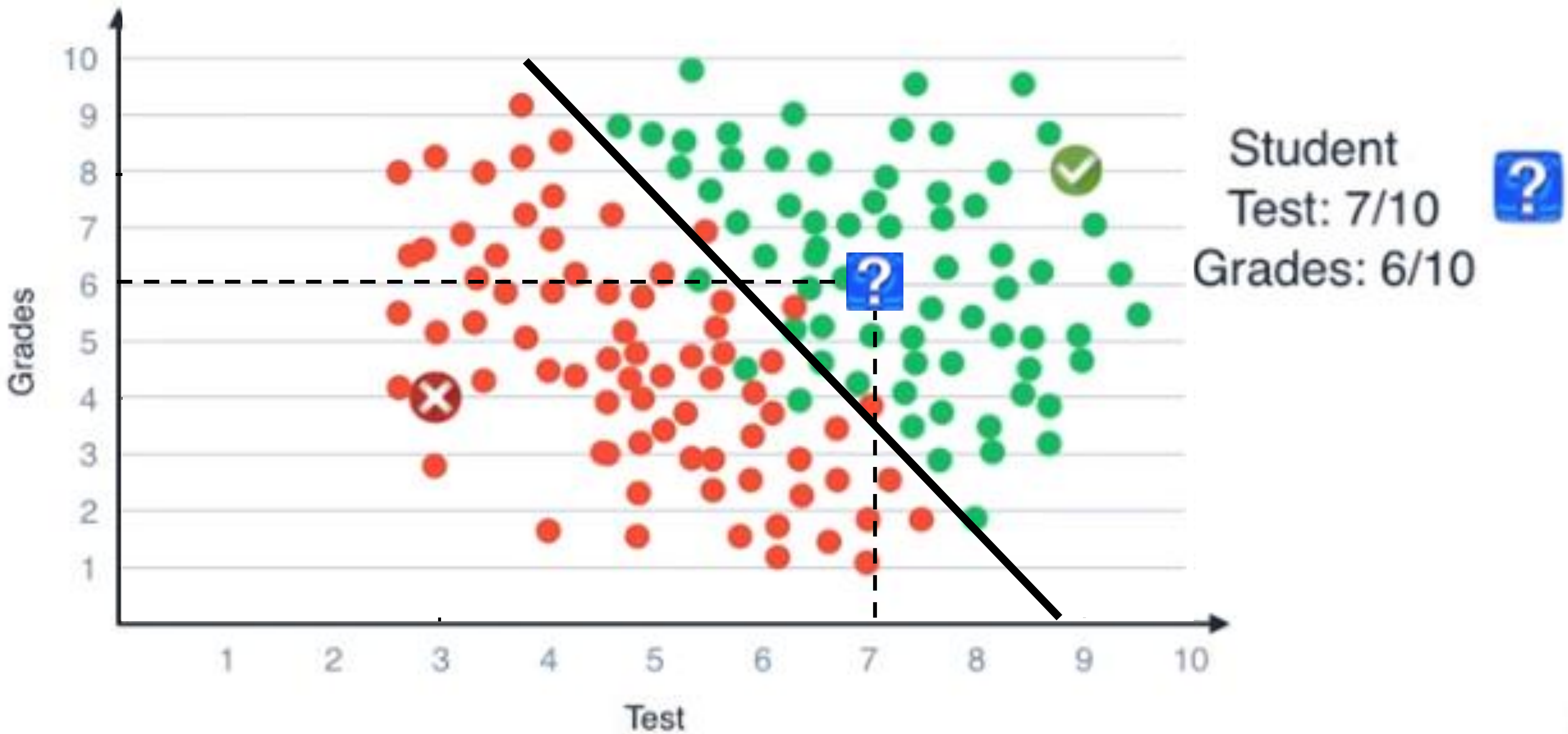
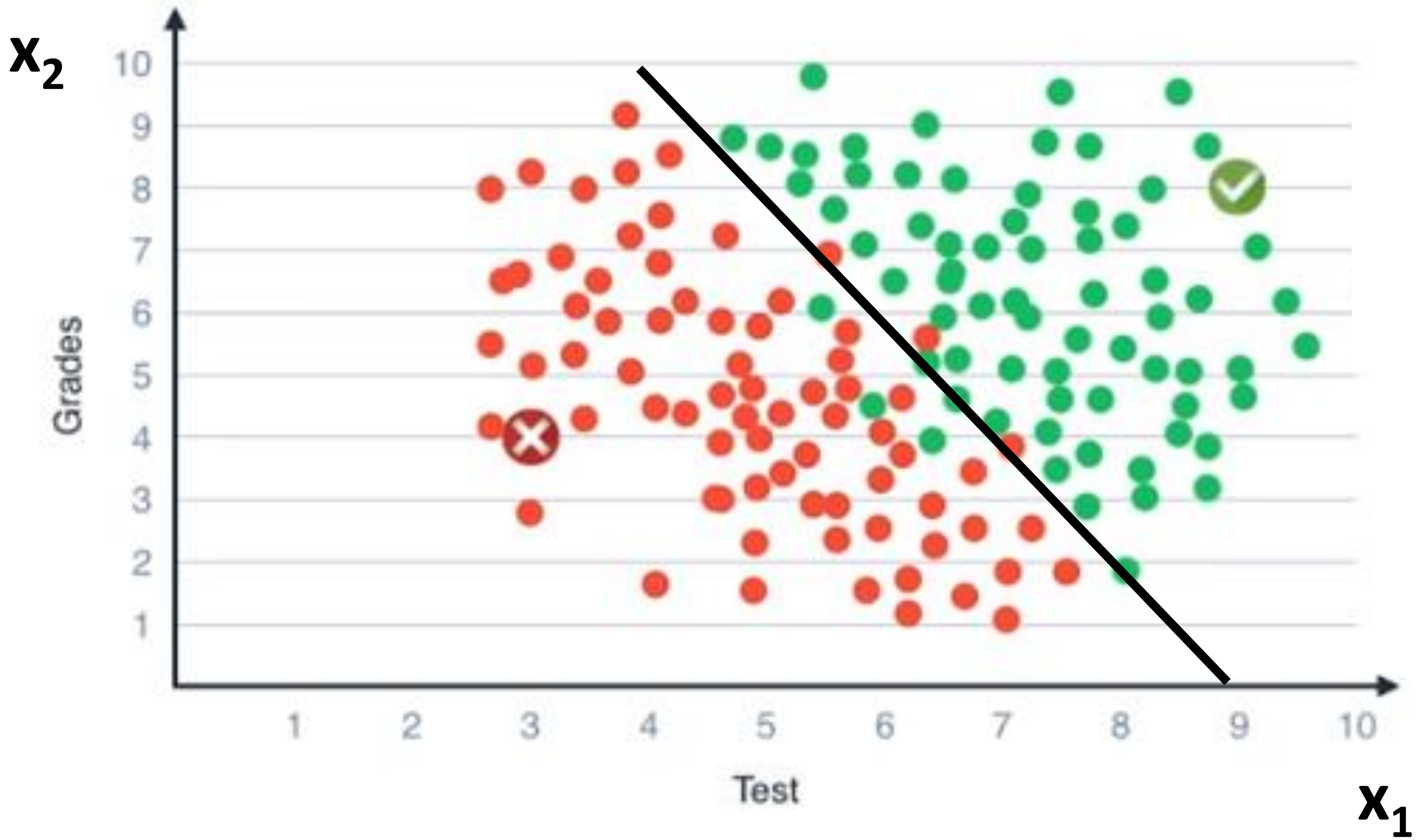
Classification

Test

Grades



Source: Udacity Machine Learning Nano-Degree



BOUNDARY: A LINE $2x_1 + x_2 - 18 = 0$

Score = 2*Test + Grades - 18

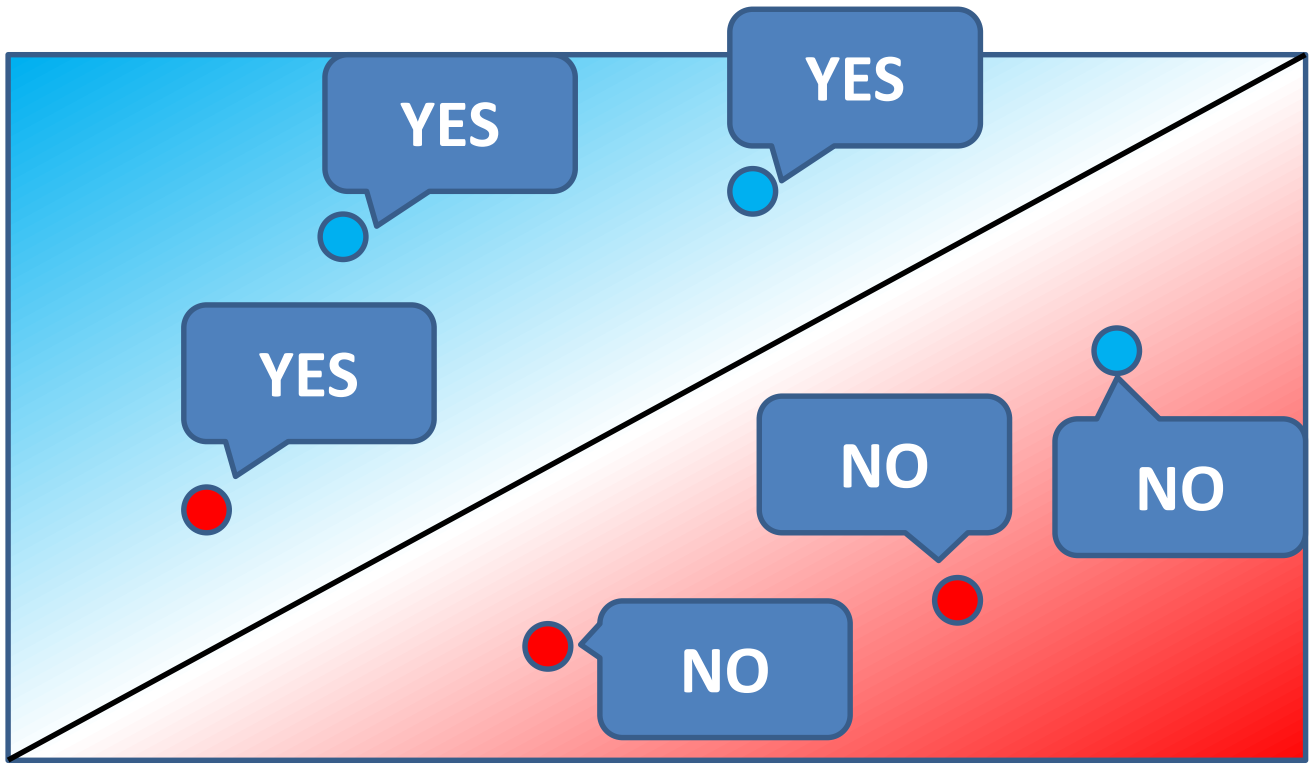
PREDICTION: $\left\{ \begin{array}{l} \text{Score} \geq 0 \ \textbf{ACCEPT} \\ \text{Score} < 0 \ \textbf{REJECT} \end{array} \right\}$

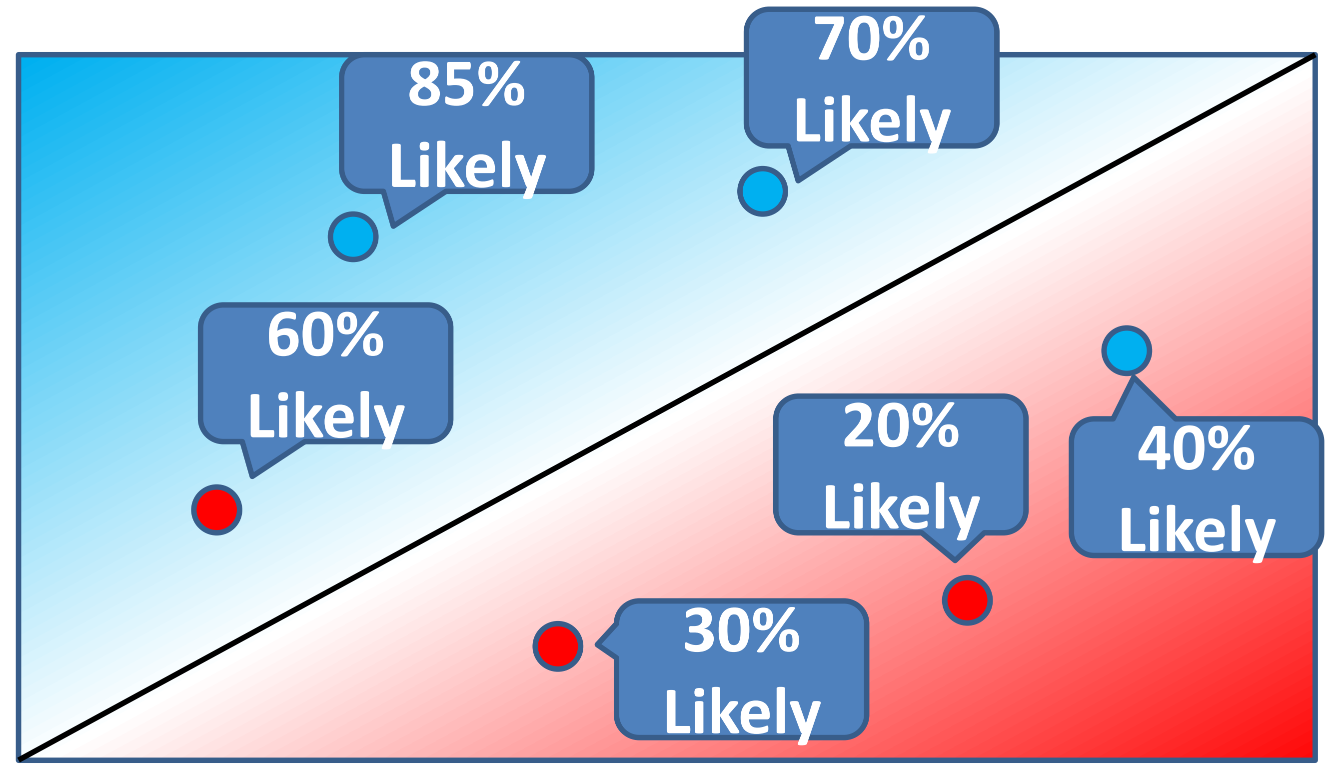
Now that you know the equation for the line ($2x_1 + x_2 - 18=0$), and similarly the "score" ($2x_1 + x_2 - 18$), what is the score of the student who got 7 in the test and 6 for grades?

Source: Udacity Machine Learning Nano-Degree

Source: Udacity Machine Learning Nano-Degree

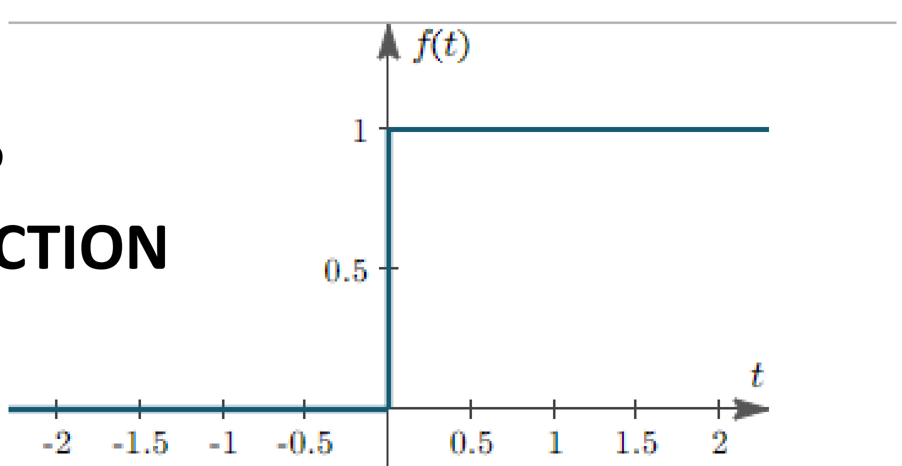
Discrete vs Continuous



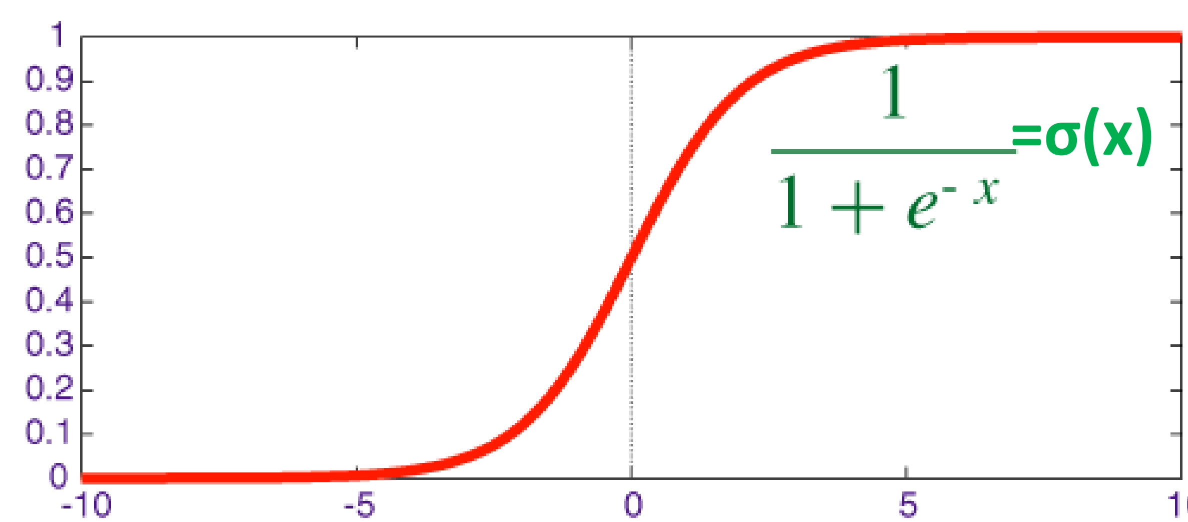


PREDICTION: $\left\{ \begin{array}{l} \text{Score} \geq 0 \ \textbf{YES} \\ \text{Score} < 0 \ \textbf{NO} \end{array} \right\}$

STEP FUNCTION

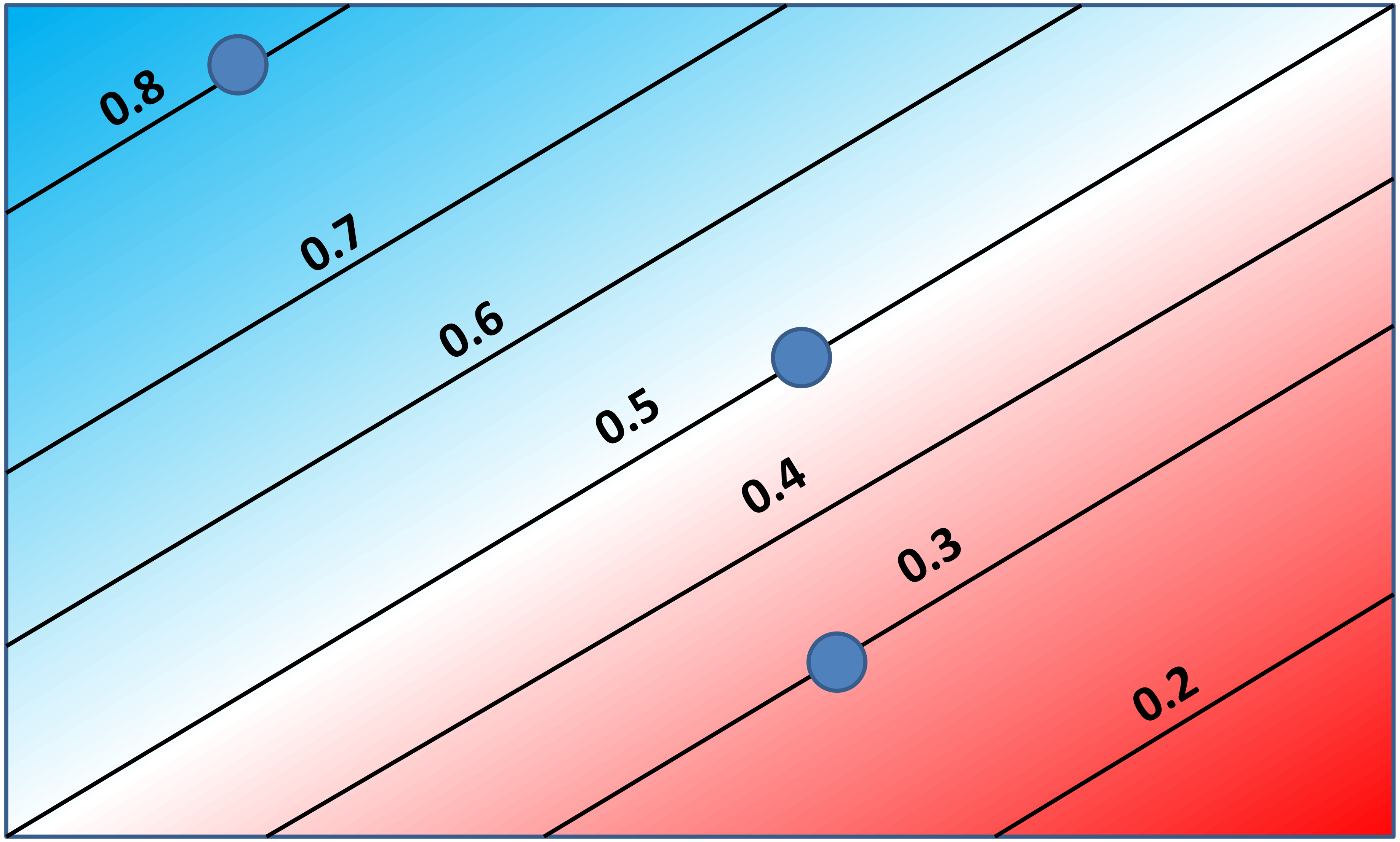


SIGMOID FUNCTION



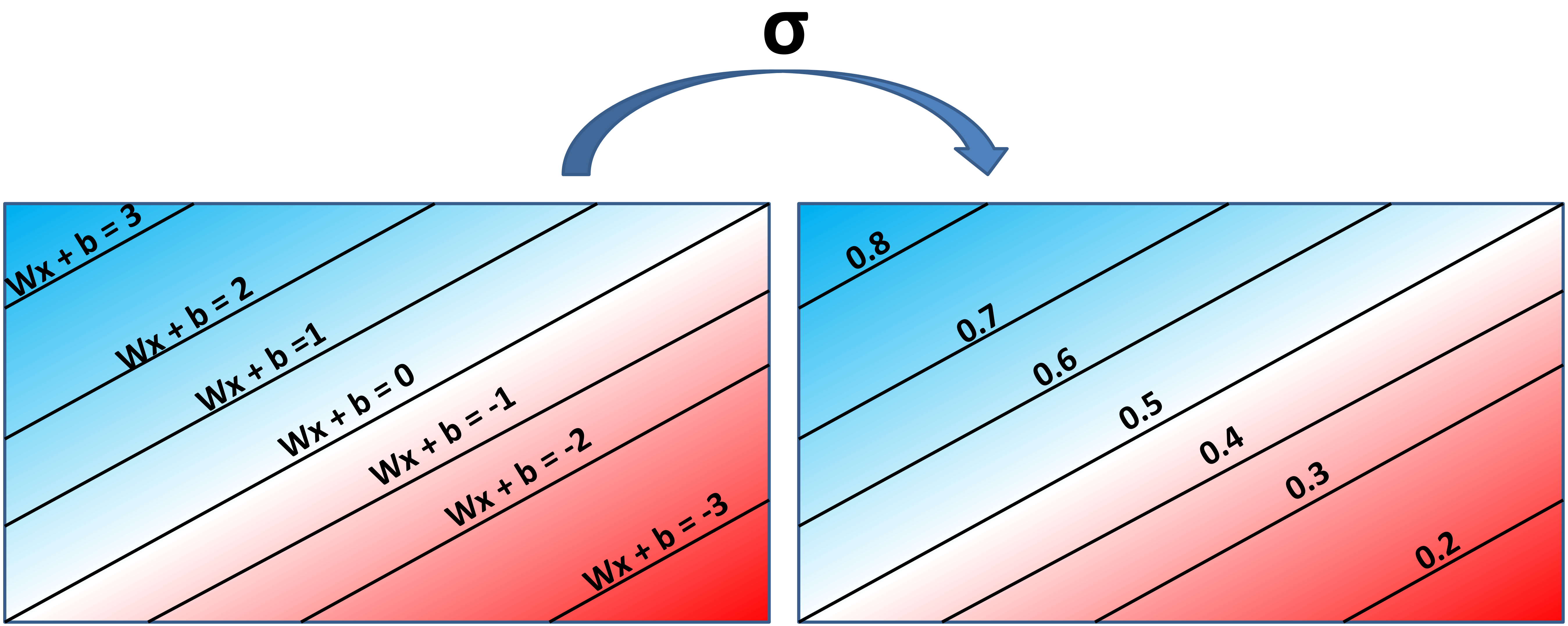
Source: Udacity Machine Learning Nano-Degree

Predictions



Source: Udacity Machine Learning Nano-Degree

Predictions



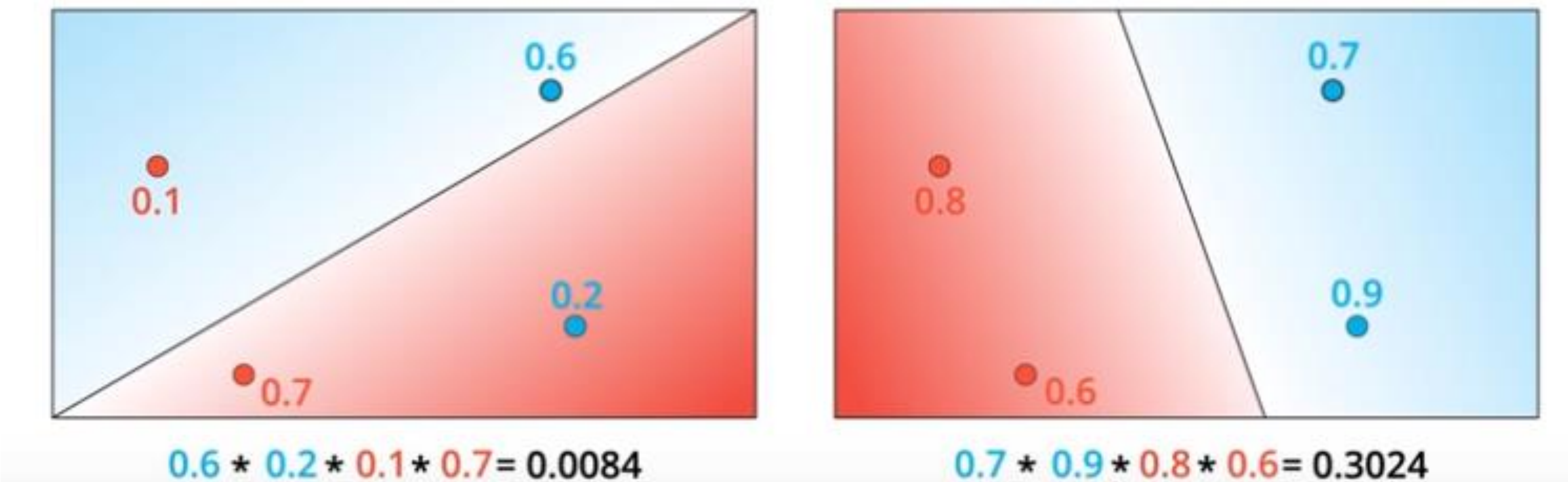
Wx + b

y-hat = σ(Wx + b)

DICE ANALYTICS

Source: Udacity Machine Learning Nano-Degree

Maximum Likelihood



Source: Udacity Machine Learning Nano-Degree

Products of Probabilities

$0.6 * 0.2 * 0.1 * 0.7 = 0.0084$

$0.7 * 0.9 * 0.8 * 0.6 = 0.3024$

Quiz:

What function to use?

- sin ○
- cos ○
- log ○
- exp ○

Source: Udacity Machine Learning Nano-Degree

Cross Entropy

$0.6*0.2*0.1*0.7 = 0.0084$

$\ln(0.6) + \ln(0.2) + \ln(0.1) + \ln(0.7)$

-0.51 -1.61 -2.3 -0.36

$-\ln(0.6) - \ln(0.2) - \ln(0.1) - \ln(0.7)$

0.51 1.61 2.3 0.36

4.8

$0.7*0.9*0.8*0.6 = 0.3024$

$\ln(0.7) + \ln(0.9) + \ln(0.8) + \ln(0.6)$

-0.36 -0.1 -.22 -0.51

$-\ln(0.7) - \ln(0.9) -\ln(0.8) -\ln(0.6)$

0.36 0.1 .22 0.51

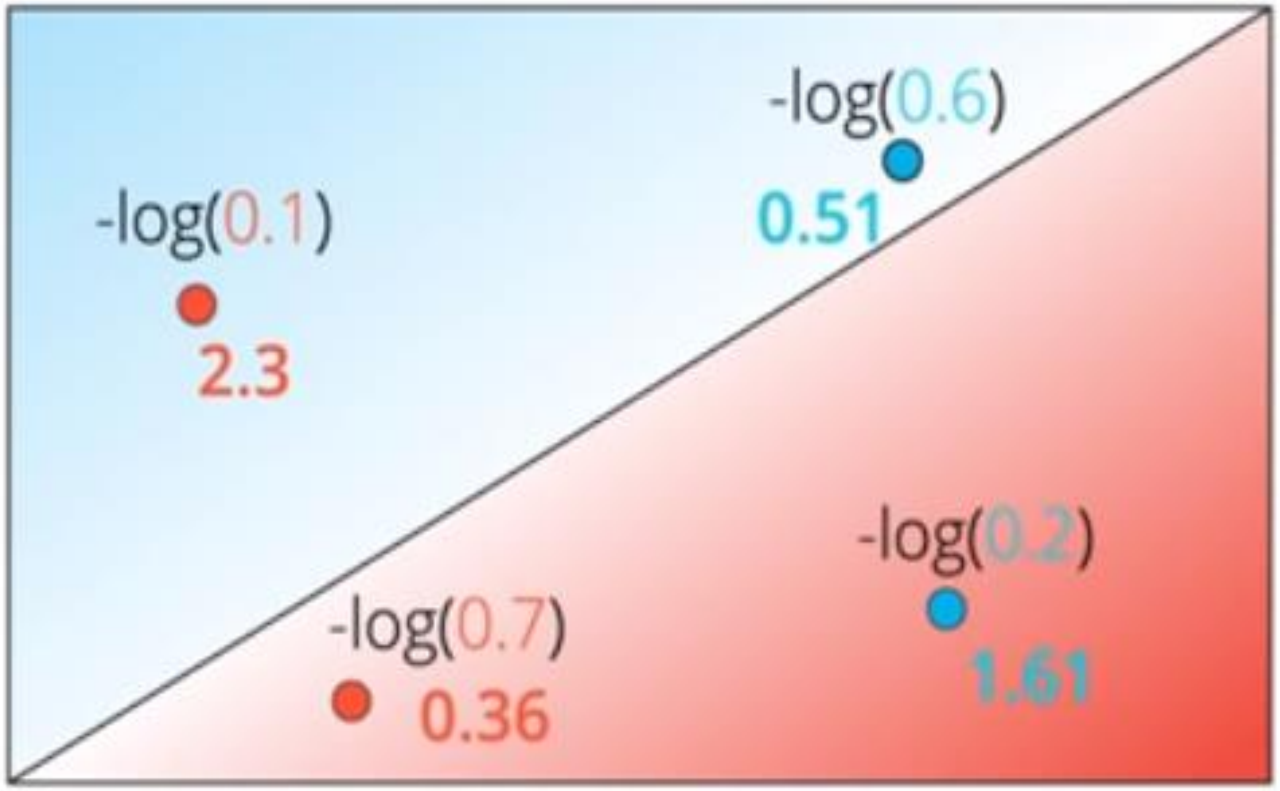
1.2

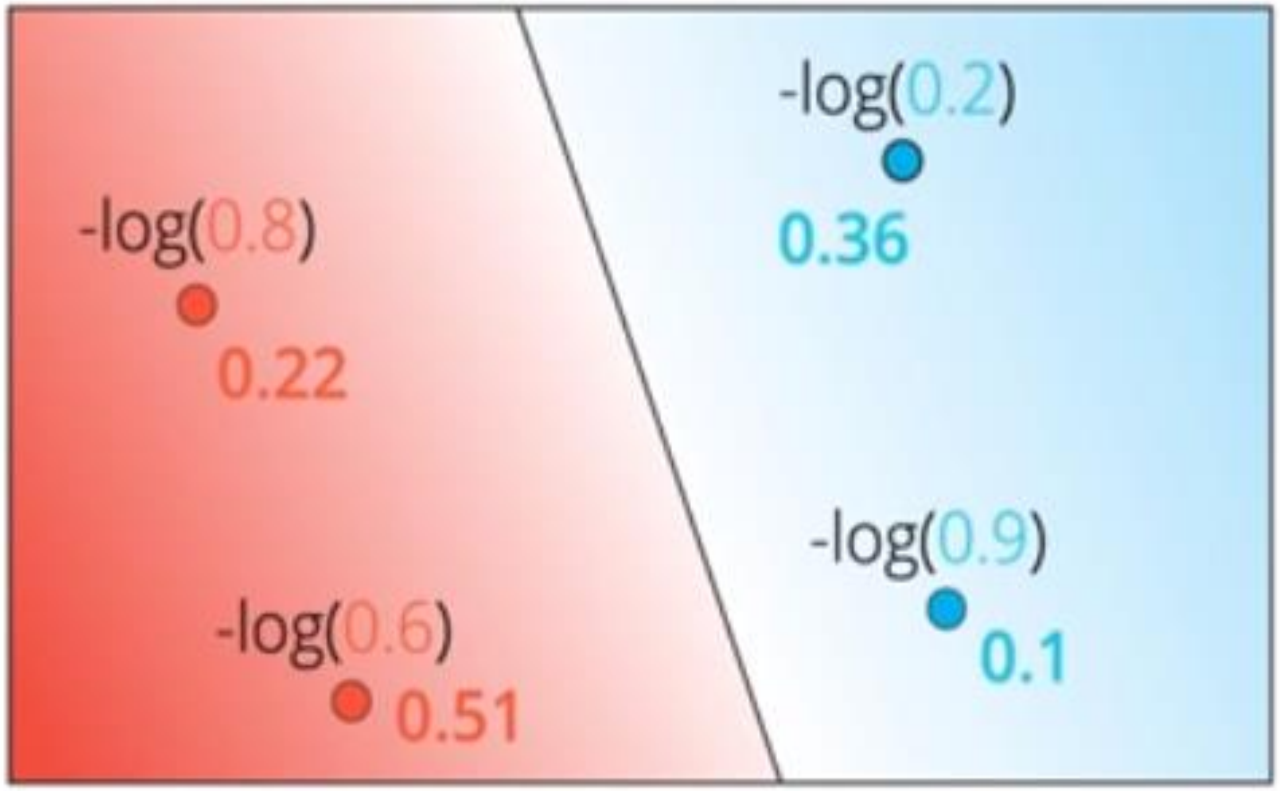
CROSS ENTROPY

DICE ANALYTICS

Source: Udacity Machine Learning Nano-Degree

Cross Entropy





0.6 * 0.2 * 0.1 * 0.7 = 0.0084

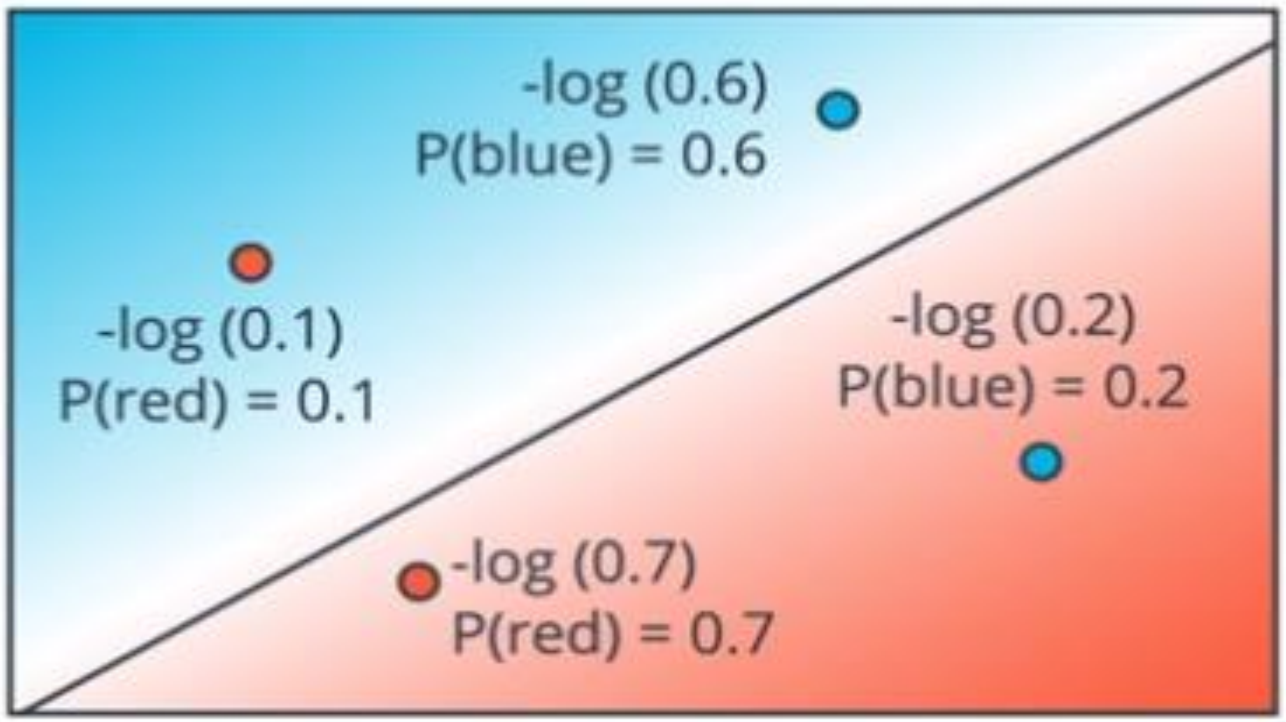
-log(0.6) - log(0.2) - log(0.1) - log(0.7) = 4.8

0.7 * 0.9 * 0.8 * 0.6 = 0.3024

-log(0.7) - log(0.9) - log(0.8) - log(0.6) = 1.2

Source: Udacity Machine Learning Nano-Degree

Error Function



$$-\log(0.6) - \log(0.2) - \log(0.1) - \log(0.7) = 4.8$$

0.51 1.61 2.3 0.36

If $y = 1$

$P(\text{blue}) = \hat{y}$

$\text{Error} = -\ln(\hat{y})$

If $y = 0$

$P(\text{red}) = 1 - P(\text{blue}) = 1 - \hat{y}$

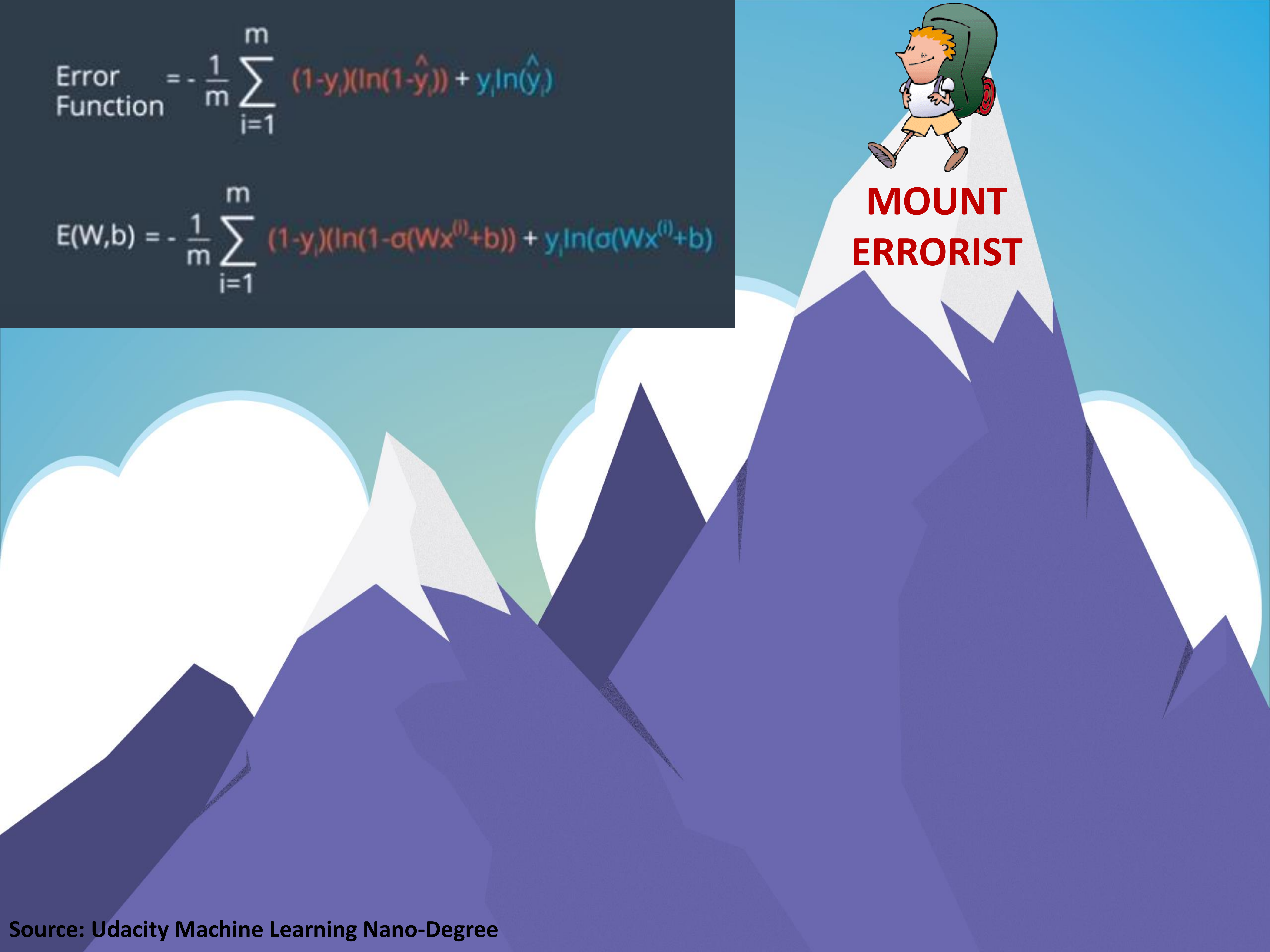
$\text{Error} = -\ln(1 - \hat{y})$

$$\text{Error} = -(1-y)(\ln(1-\hat{y})) - y\ln(\hat{y})$$

Source: Udacity Machine Learning Nano-Degree

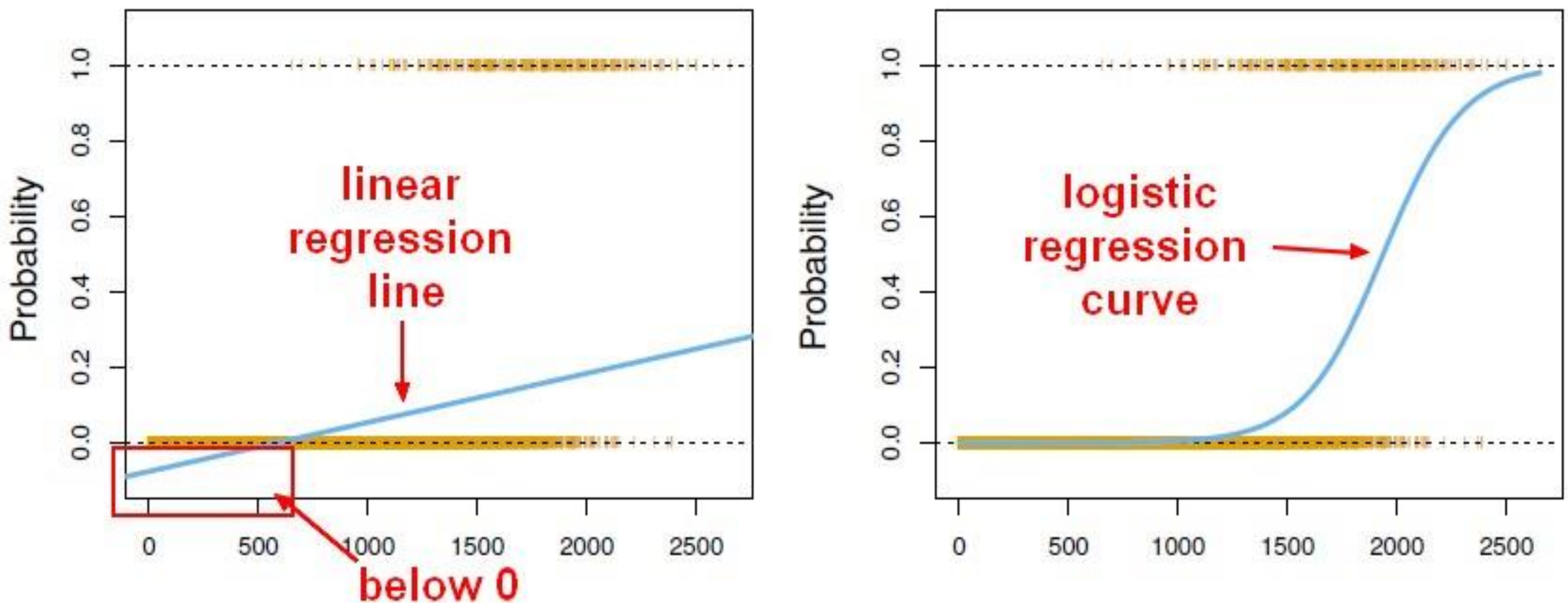
$$\text{Error Function} = -\frac{1}{m}\sum_{i=1}^{m}(1-y_i)(\ln(1-\hat{y}_i)) + y_i\ln(\hat{y}_i)$$

$$E(W,b) = -\frac{1}{m}\sum_{i=1}^{m}(1-y_i)(\ln(1-\sigma(Wx^{(i)}+b)) + y_i\ln(\sigma(Wx^{(i)}+b)$$



Source: Udacity Machine Learning Nano-Degree

Logistic Regression



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

DICE ANALYTICS

Model Evaluation

Confusion Matrix

10, 000 PATIENTS

DIAGNOSIS

PATIENTS

	Diagnosed Sick	Diagnosed Healthy
Sick	1000 True Positives	200 False Negatives
Healthy	800 False Positives	8000 True Negatives

Source: Udacity Machine Learning Nano-Degree

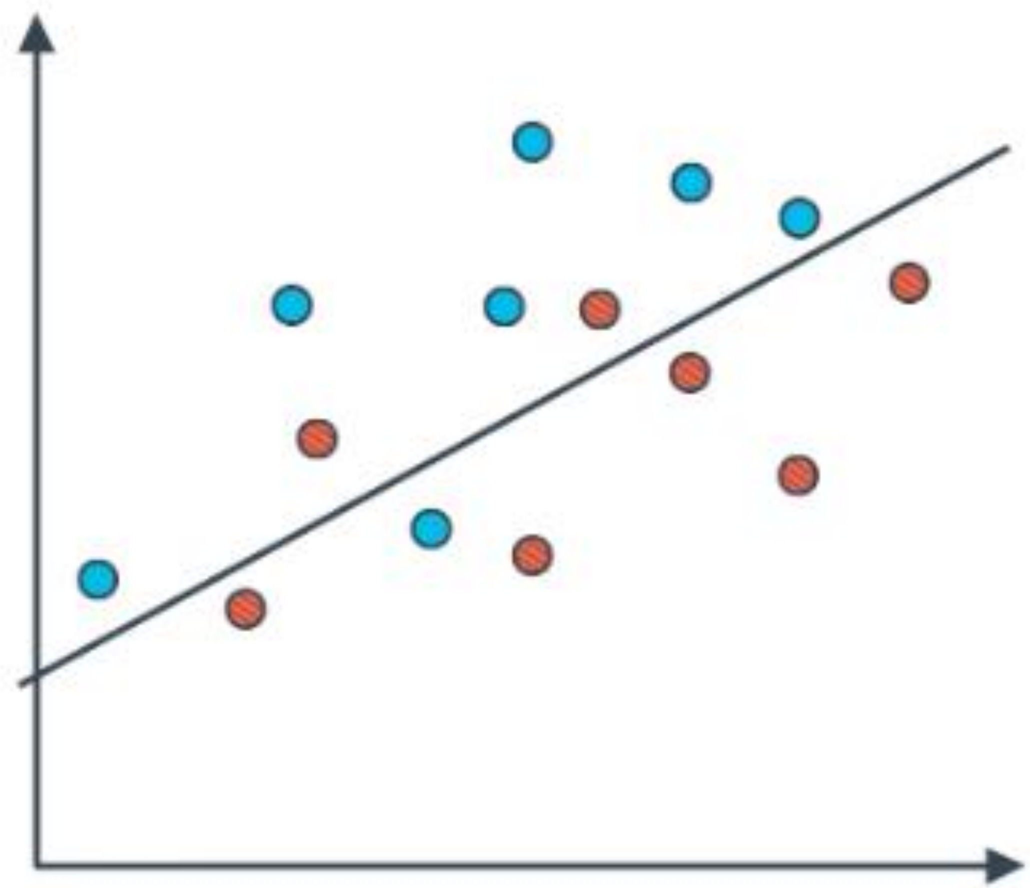
Confusion Matrix

1000 EMAILS

EMAIL \ SPAM	Spam Folder	Inbox
Spam	100 True Positives	170 False Negatives
Not Spam	30 False Positives	700 True Negatives

Source: Udacity Machine Learning Nano-Degree

Confusion Matrix



	Guessed Positive	Guessed Negative
Positive	6 True Positives	1 False Negatives
Negative	2 False Positives	5 True Negatives

TYPE-1 Error

TYPE-2 Error

In this image, the blue points are labelled positive, and the red points are labelled negative. Furthermore, the points on top of the line are predicted (guessed) to be positive, and the points below the line are predicted to be negative.

Source: Udacity Machine Learning Nano-Degree

Accuracy

Out of total patients, how many identified correctly

	Diagnosed Sick	Diagnosed Healthy
Sick	1000 True Positives	200 False Negatives
Healthy	800 False Positives	8000 True Negatives

$$\frac{1000+8000}{1000+8000+200+800} = 90\%$$

Out of total emails, how many identified correctly

	Spam Folder	Inbox
Spam	100 True Positives	170 False Negatives
Not Spam	30 False Positives	700 True Negatives

$$\frac{100+700}{100+700+30+170} = 80\%$$

Source: Udacity Machine Learning Nano-Degree

CREDIT CARD FRAUD

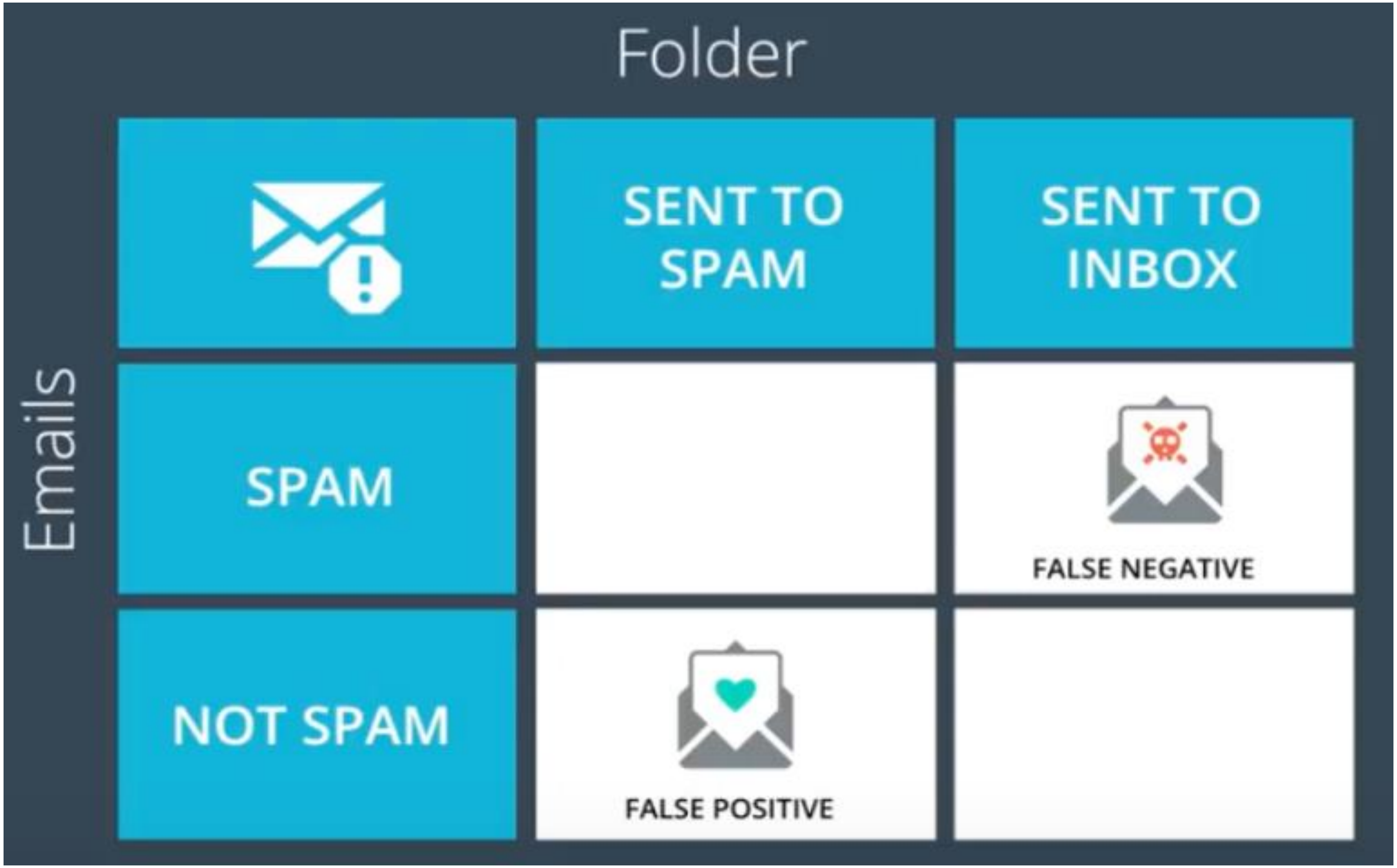
MODEL: ALL TRANSACTIONS ARE FRAUDULENT.

GREAT! NOW I'M CATCHING ALL OF THE FRAUDULENT TRANSACTIONS!

PROBLEM: I'M ACCIDENTALLY CATCHING ALL OF THE GOOD ONES!

Source: Udacity Machine Learning Nano-Degree

WHICH ONE IS WORST?



Source: Udacity Machine Learning Nano-Degree

DICE ANALYTICS

HOW WE EVALUATE?

HIGH RECALL

HIGH PRECISION

Source: Udacity Machine Learning Nano-Degree

Precision

Out of <u>all patients diagnosed as sick</u>, how many <u>diagnosed sick correctly</u>

	Diagnosed Sick	Diagnosed Healthy
Sick	1000 True Positives	200 False Negatives
Healthy	800 False Positives	8000 True Negatives

$$\frac{1000}{1000+800} = 55.6\%$$

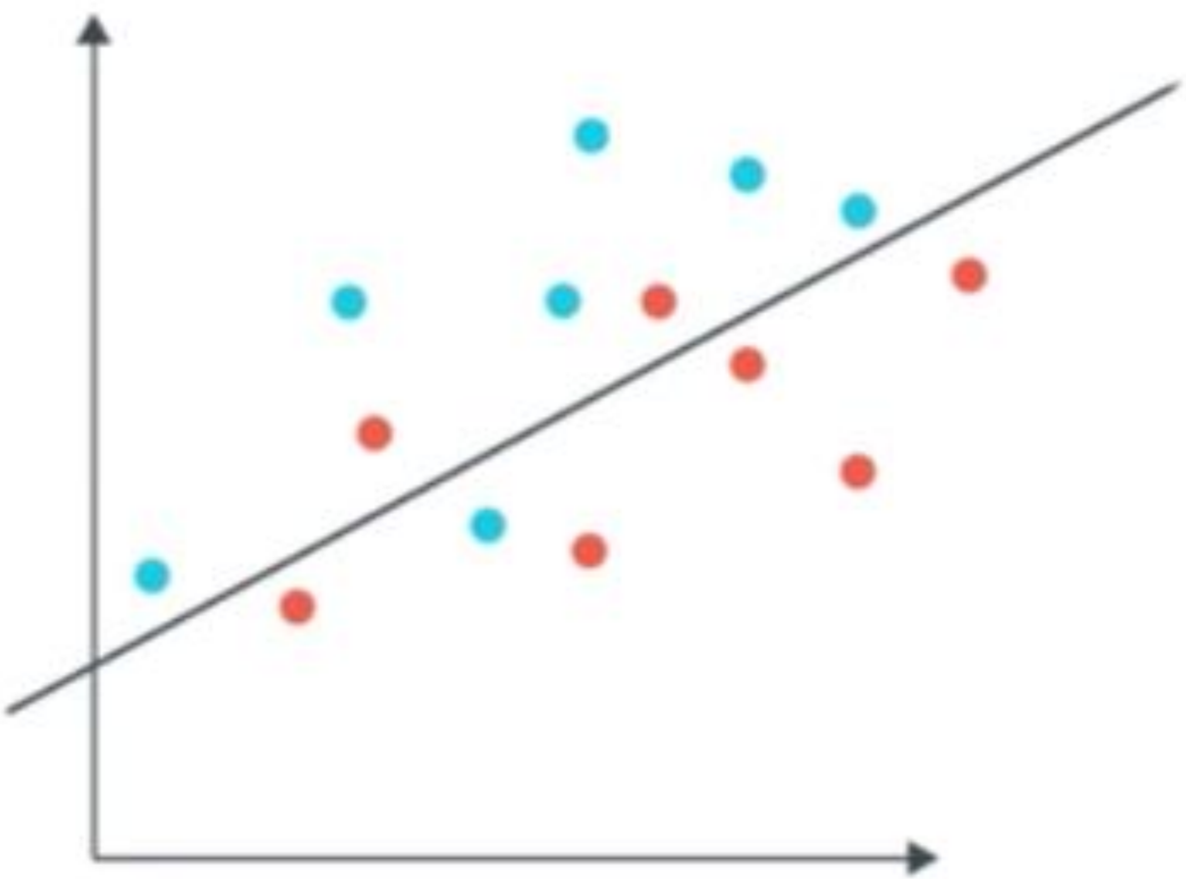
Out of <u>all emails sent to Spam folder</u>, how many <u>emails sent correctly</u>

	Spam Folder	Inbox
Spam	100 True Positives	170 False Negatives
Not Spam	30 False Positives	700 True Negatives

$$\frac{100}{100+30} = 76.9\%$$

Source: Udacity Machine Learning Nano-Degree

QUIZ



OUT OF THE POINTS WE HAVE PREDICTED TO BE POSITIVE, HOW MANY ARE CORRECT?

Source: Udacity Machine Learning Nano-Degree

Recall

Out of all sick patients, how many were correctly diagnosed as sick

	Diagnosed Sick	Diagnosed Healthy
Sick	1000 True Positives	200 False Negatives
Healthy	800 False Positives	8000 True Negatives

$$\frac{1000}{1000+200} = 83.3\%$$

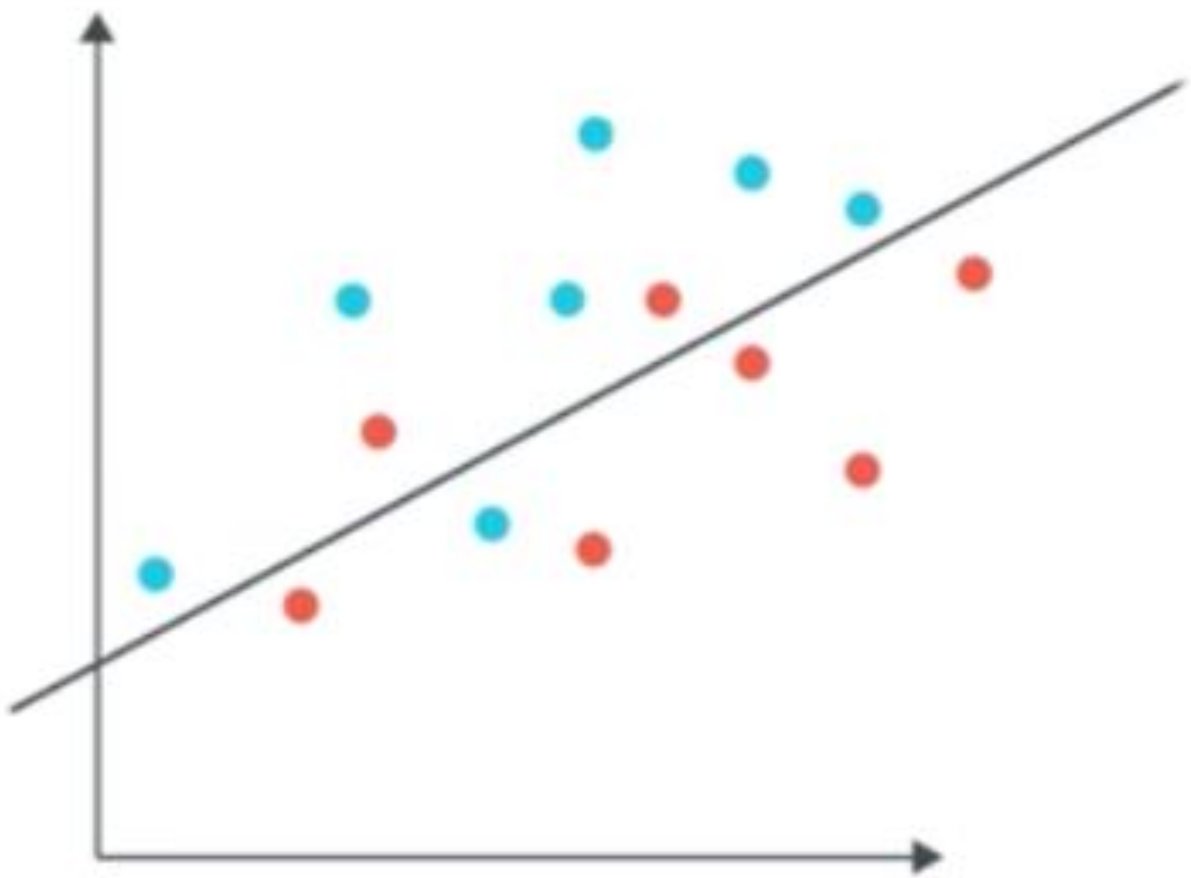
Out of all spam emails, how many were correctly sent to spam folder

	Spam Folder	Inbox
Spam	100 True Positives	170 False Negatives
Not Spam	30 False Positives	700 True Negatives

$$\frac{100}{100+170} = 37\%$$

Source: Udacity Machine Learning Nano-Degree

QUIZ



OUT OF THE POINTS LABELLED "POSTIIVE," HOW MANY DID WE CORRECTLY PREDICT?

Source: Udacity Machine Learning Nano-Degree

Precision vs Recall

MEDICAL MODEL

PRECISION: 55.7%

RECALL: 83.3%

ONE SCORE?

SPAM DETECTOR

PRECISION: 76.9%

RECALL: 37%

Source: Udacity Machine Learning Nano-Degree

F1 Score

MEDICAL MODEL

PRECISION: 55.7%

RECALL: 83.3%

AVERAGE: 69.5%

ONE SCORE?

SPAM DETECTOR

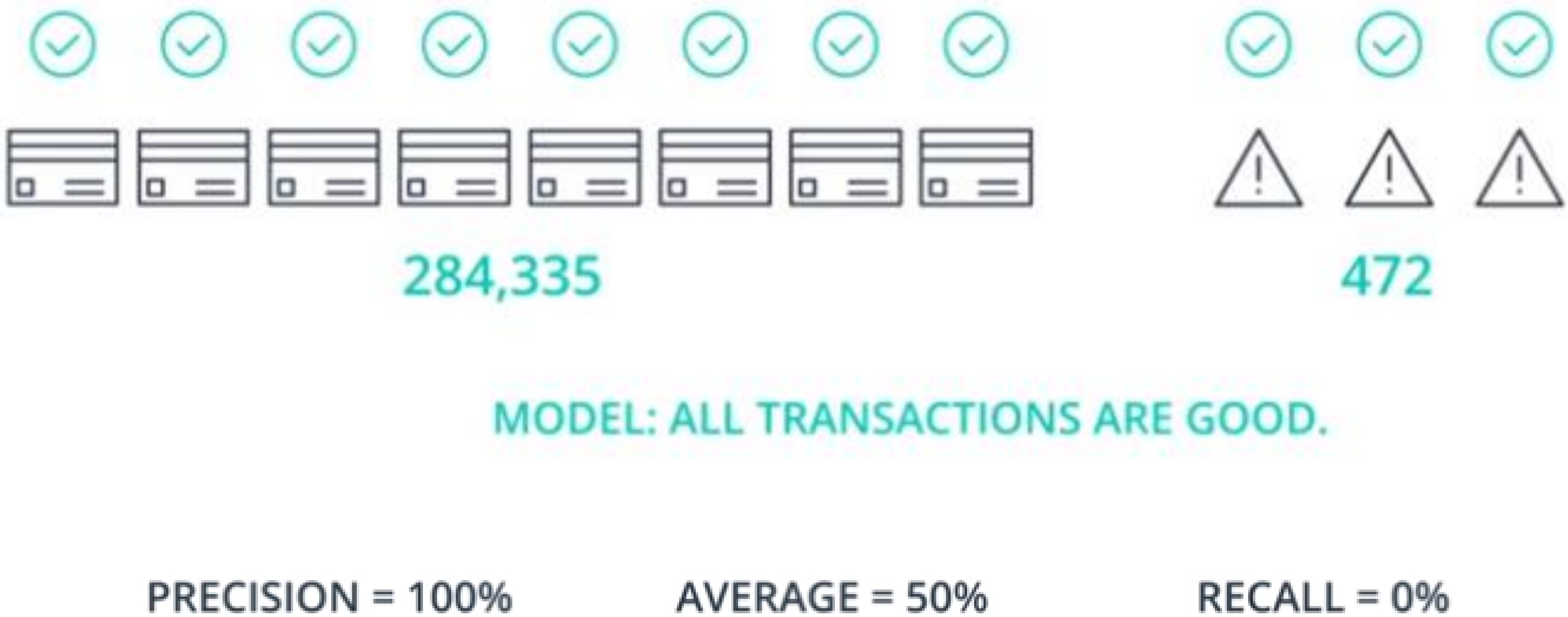
PRECISION: 76.9%

RECALL: 37%

AVERAGE: 56.95%

Source: Udacity Machine Learning Nano-Degree

F1 Score



DICE ANALYTICS

Source: Udacity Machine Learning Nano-Degree

F1 Score

MODEL: ALL TRANSACTIONS ARE FRAUDULENT.

PRECISION = 472/284,807 = 0.16%

RECALL = 472/472 = 100%

AVERAGE = 50.08%

Source: Udacity Machine Learning Nano-Degree

DICE ANALYTICS

F1 Score

Y

PRECISION = 1

RECALL = 0

AVERAGE = 0.5

HARMONIC MEAN = 0

PRECISION = 0.2

RECALL = 0.8

AVERAGE = 0.5

HARMONIC MEAN = 0.32

ARITHMETIC MEAN=
(X+Y)/2

HARMONIC MEAN=
2*(XY)/(X+Y)

~~ARITHMETIC MEAN (PRECISION,RECALL)~~

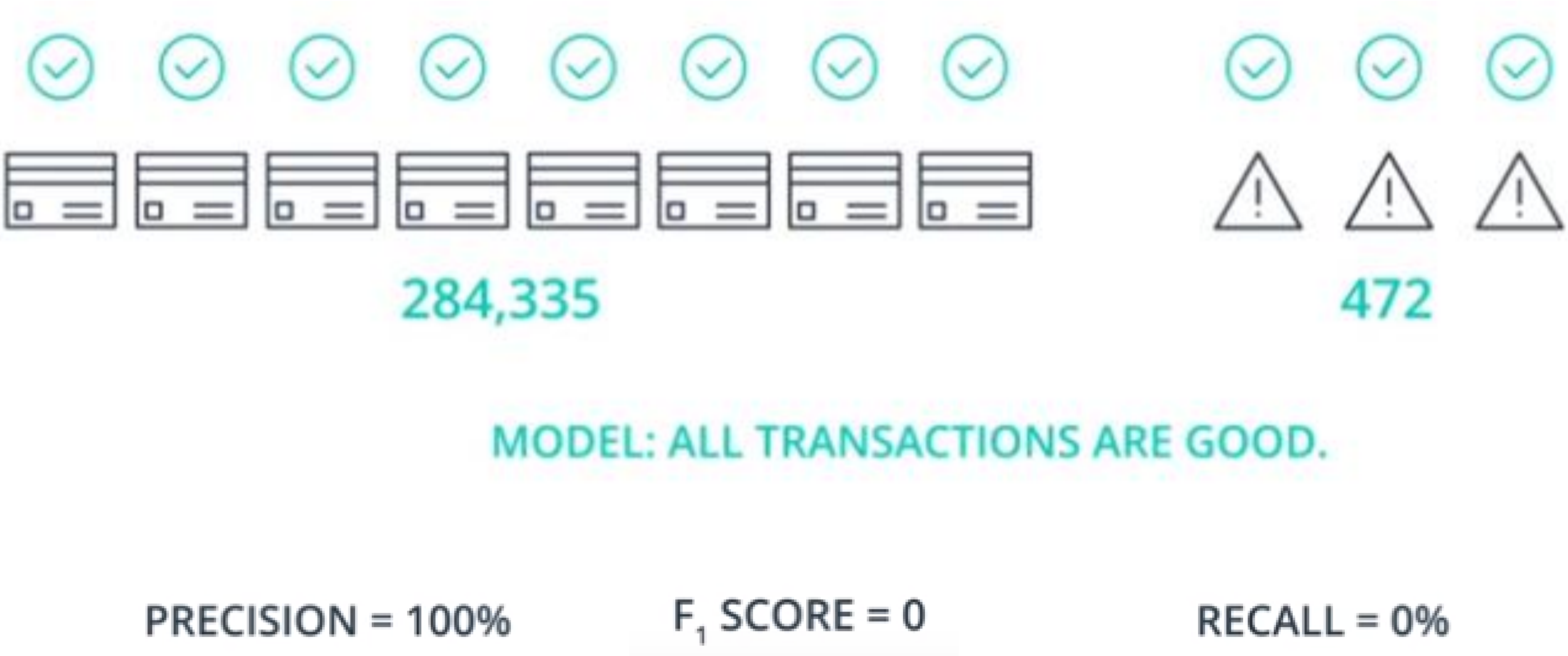
F1 SCORE= HARMONIC MEAN(PRECISION,RECALL)

X

DICE ANALYTICS

Source: Udacity Machine Learning Nano-Degree

F1 Score



Source: Udacity Machine Learning Nano-Degree

Confusion Matrix

n=165	Predicted: NO	Predicted: YES
Actual: NO	50	10
Actual: YES	5	100

	Predicted: NO	Predicted: YES
Actual: NO	TN	FP
Actual: YES	FN	TP

- **true positives (TP):** These are cases in which we predicted yes (they have the disease), and they do have the disease.
- **true negatives (TN):** We predicted no, and they don't have the disease.
- **false positives (FP):** We predicted yes, but they don't actually have the disease. (Also known as a "Type I error.")
- **false negatives (FN):** We predicted no, but they actually do have the disease. (Also known as a "Type II error.")

Confusion Matrix

n=165	Predicted: NO	Predicted: YES	
Actual: NO	TN = 50	FP = 10	60
Actual: YES	FN = 5	TP = 100	105
	55	110	

Accuracy: Overall, how often is the classifier correct?

(TP+TN)/total = (100+50)/165 = 0.91

Precision: When it predicts yes, how often is it correct?

TP/predicted yes = 100/110 = 0.91

True Positive Rate: When it's actually yes, how often does it predict yes?

TP/actual yes = 100/105 = 0.95

also known as "Sensitivity" or "Recall"

DICE ANALYTICS

Confusion Matrix

n=165	**Predicted: NO**	**Predicted: YES**	
Actual: NO	TN = 50	FP = 10	60
Actual: YES	FN = 5	TP = 100	105
	55	110	

False Positive Rate: When it's actually no, how often does it predict yes?

FP/actual no = 10/60 = 0.17

Specificity: When it's actually no, how often does it predict no?

TN/actual no = 50/60 = 0.83

equivalent to 1 minus False Positive Rate

F1 Score: This is a weighted average of the true positive rate (recall) and precision.

2 * (precision * recall)/(precision + recall)

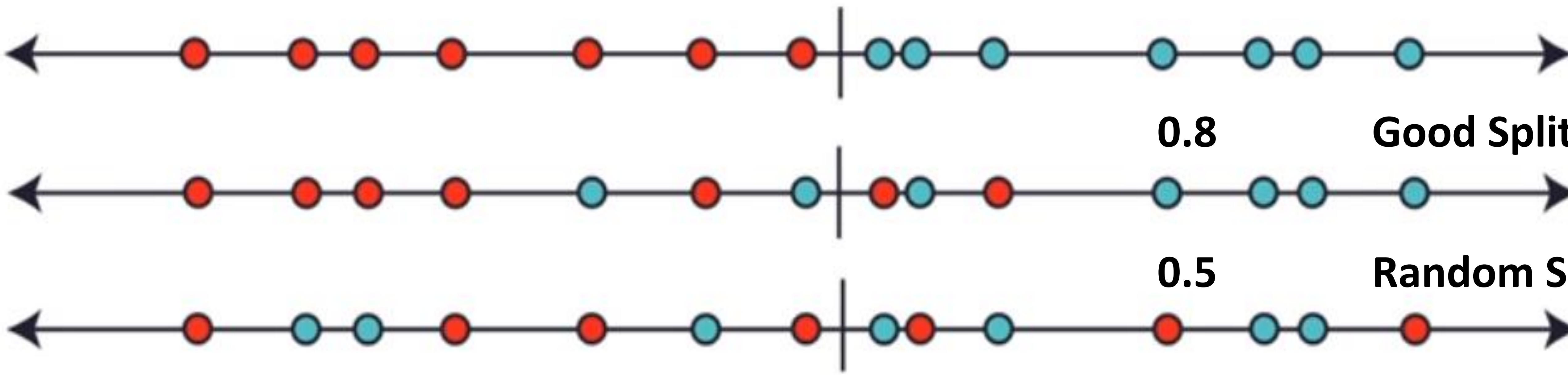
DICE ANALYTICS

Receiver Operating Characteristic

1.0 **Perfect Split**

0.8 **Good Split**

0.5 **Random Split**



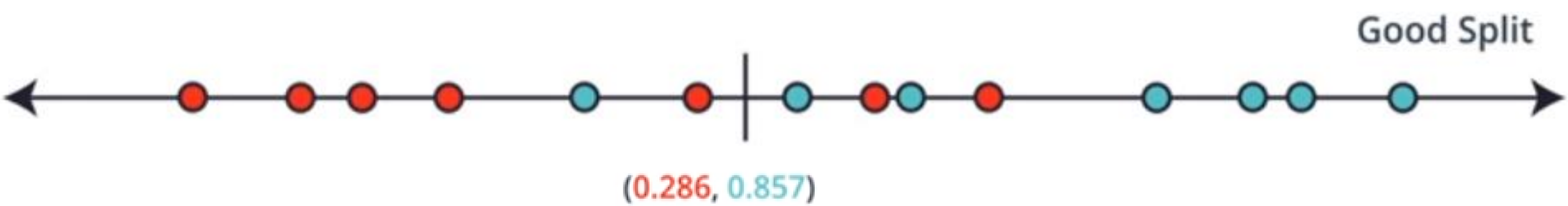
Source: Udacity Machine Learning Nano-Degree

DICE ANALYTICS

ROC Curve

$$\text{True Positive Rate} = \frac{\text{TRUE POSITIVES}}{\text{ALL POSITIVES}} = \frac{6}{7}$$

$$\text{False Positive Rate} = \frac{\text{FALSE POSITIVES}}{\text{ALL NEGATIVES}} = \frac{2}{7}$$

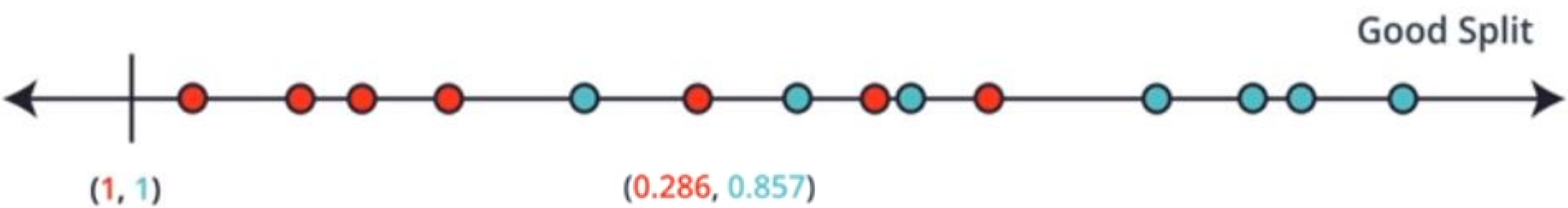


Source: Udacity Machine Learning Nano-Degree

ROC Curve

$$\text{True Positive Rate} = \frac{\text{TRUE POSITIVES}}{\text{ALL POSITIVES}} = \frac{7}{7} =$$

$$\text{False Positive Rate} = \frac{\text{FALSE POSITIVES}}{\text{ALL NEGATIVES}} = \frac{7}{7} =$$



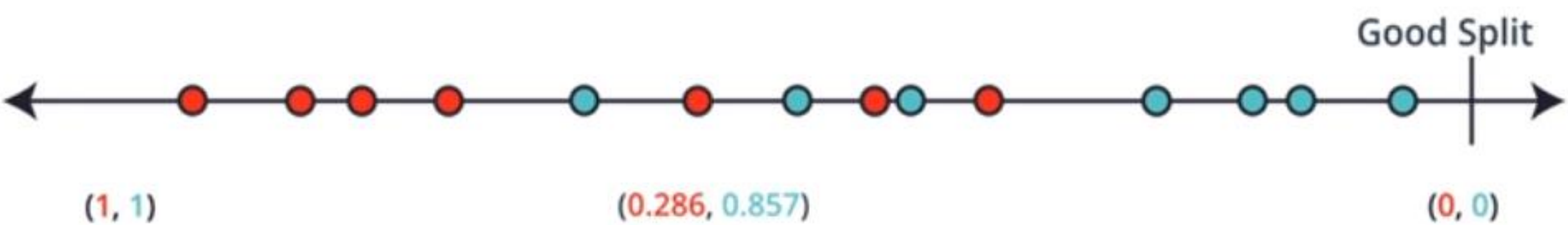
Source: Udacity Machine Learning Nano-Degree

DICE ANALYTICS

ROC Curve

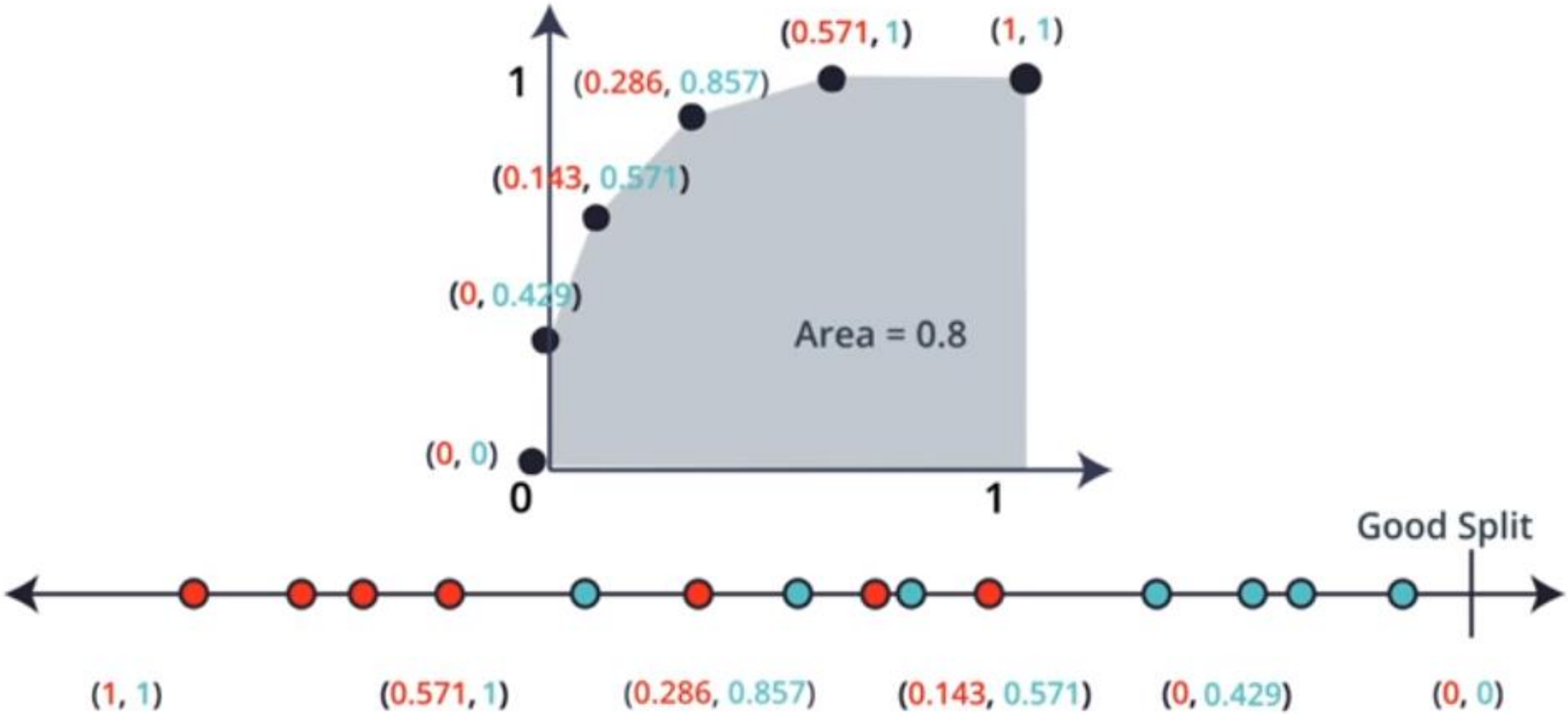
True Positive Rate = $\dfrac{\text{TRUE POSITIVES}}{\text{ALL POSITIVES}} = \dfrac{0}{7} =$

False Positive Rate = $\dfrac{\text{FALSE POSITIVES}}{\text{ALL NEGATIVES}} = \dfrac{0}{7} =$



Source: Udacity Machine Learning Nano-Degree

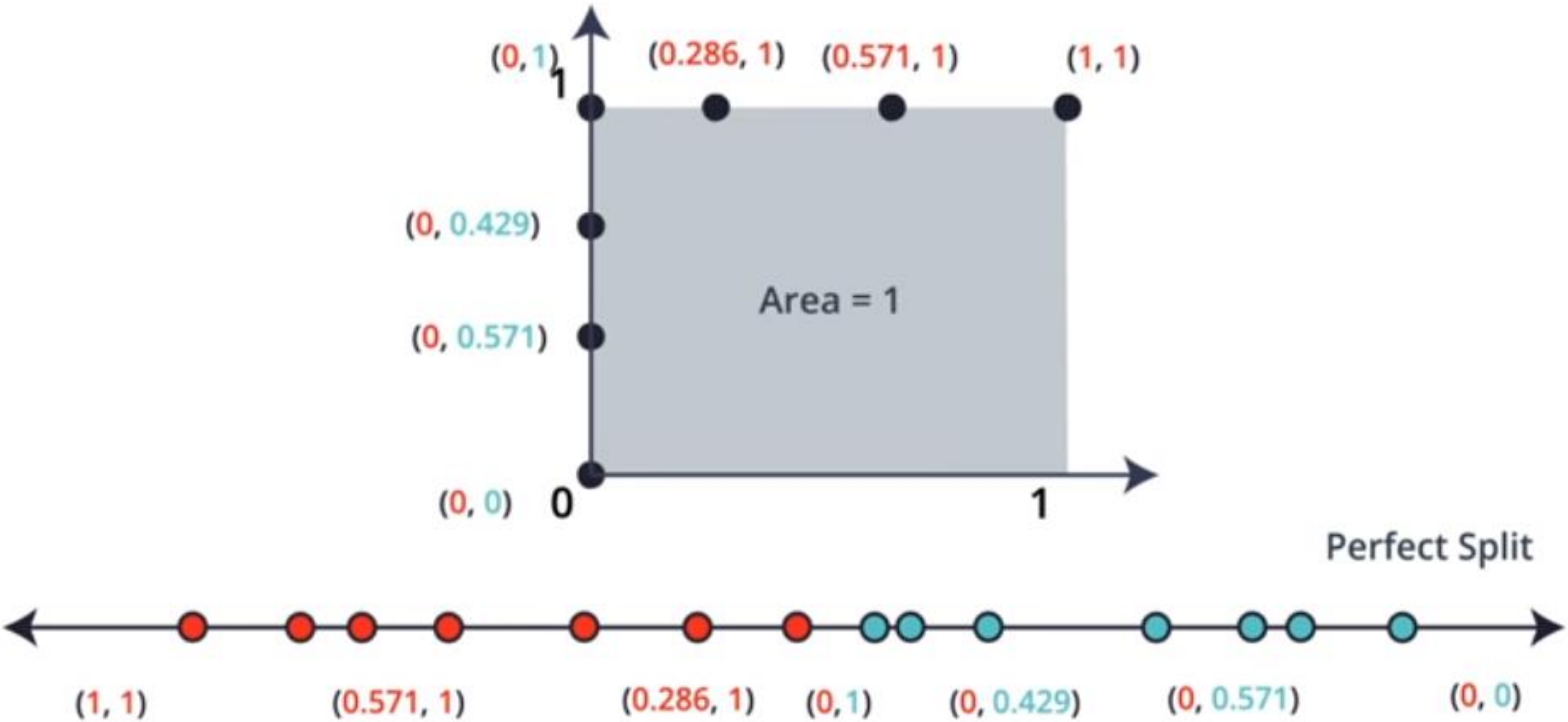
ROC Curve



Source: Udacity Machine Learning Nano-Degree

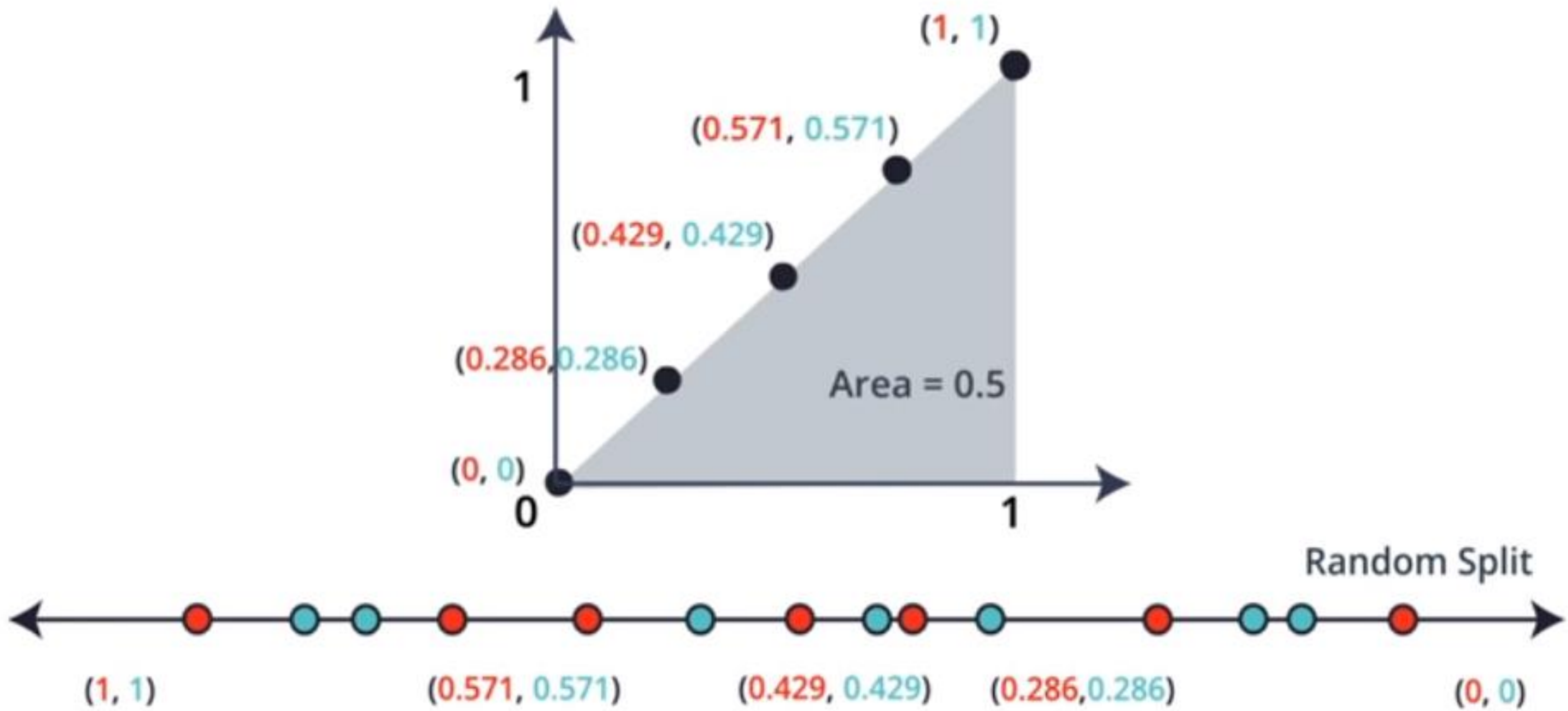
DICE ANALYTICS

ROC Curve



Source: Udacity Machine Learning Nano-Degree

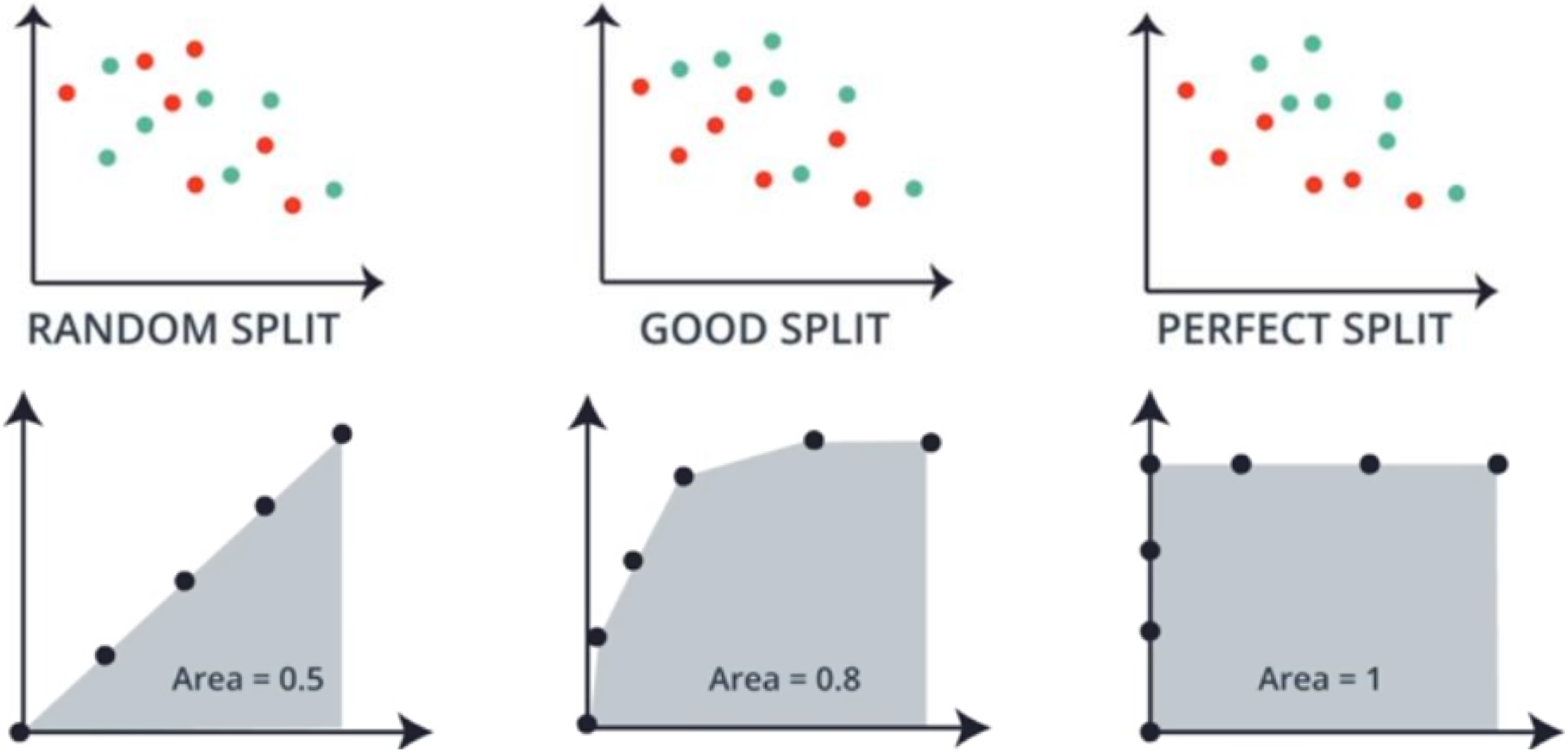
ROC Curve



Source: Udacity Machine Learning Nano-Degree

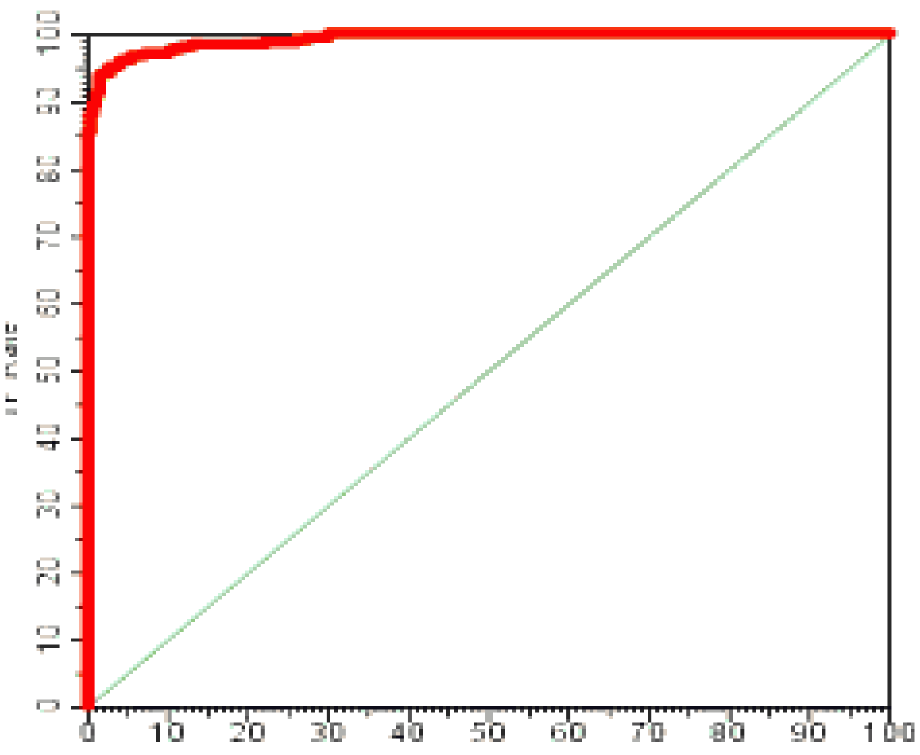
DICE ANALYTICS

AREA UNDER ROC Curve

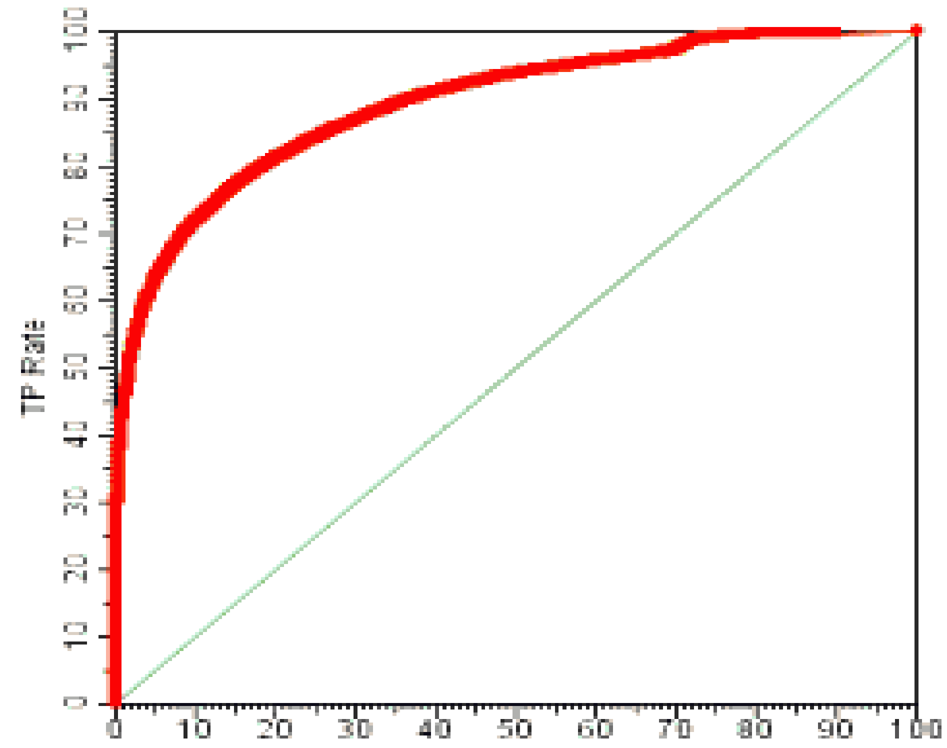


Source: Udacity Machine Learning Nano-Degree

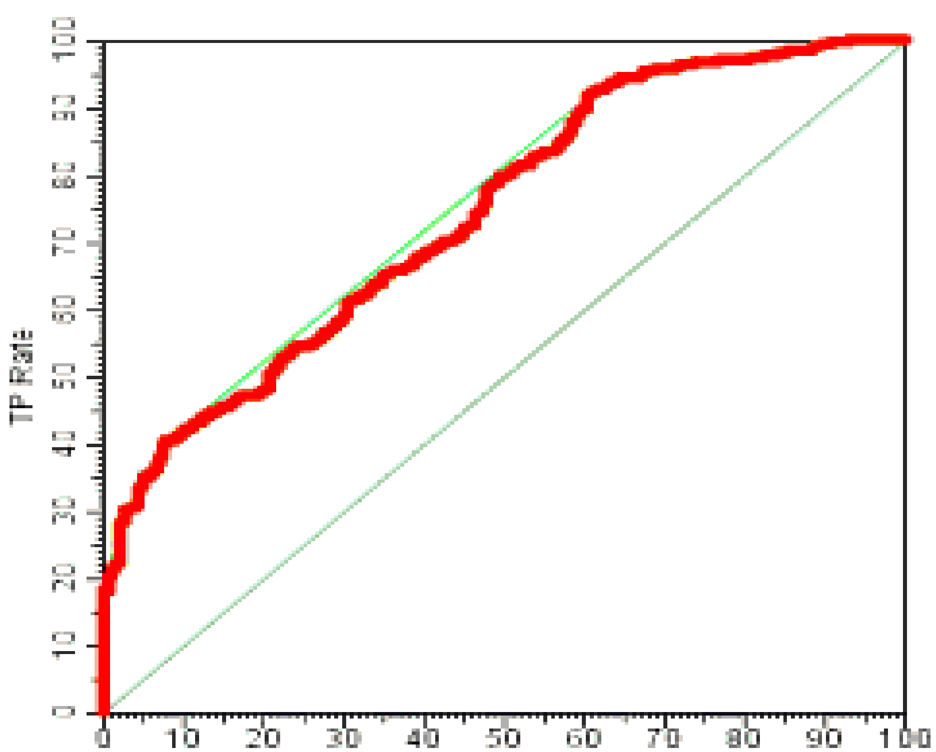
ROC AUC Curve



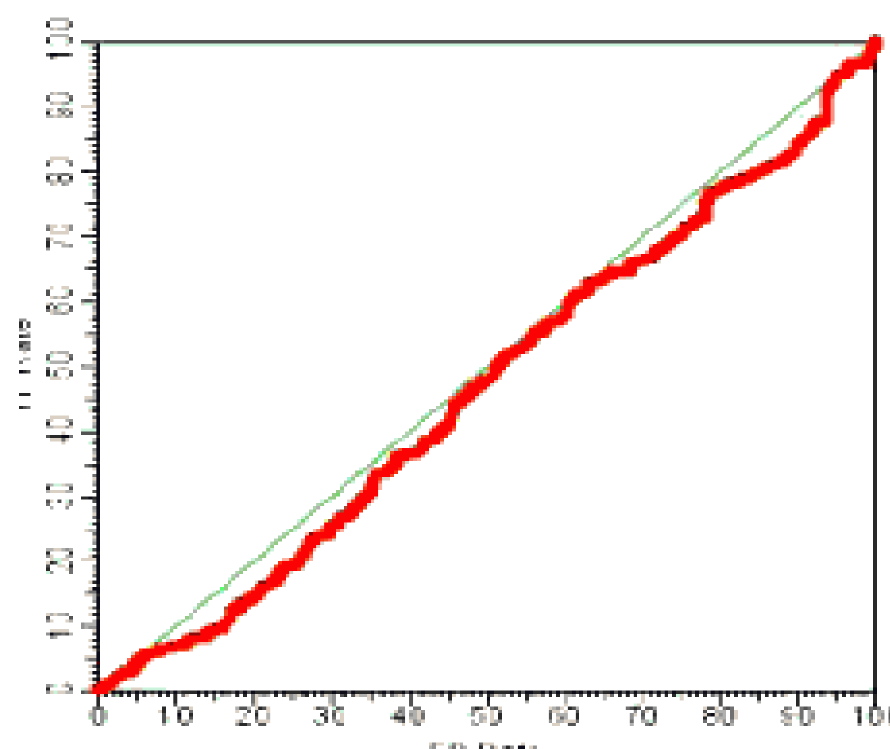
good separation



reasonable



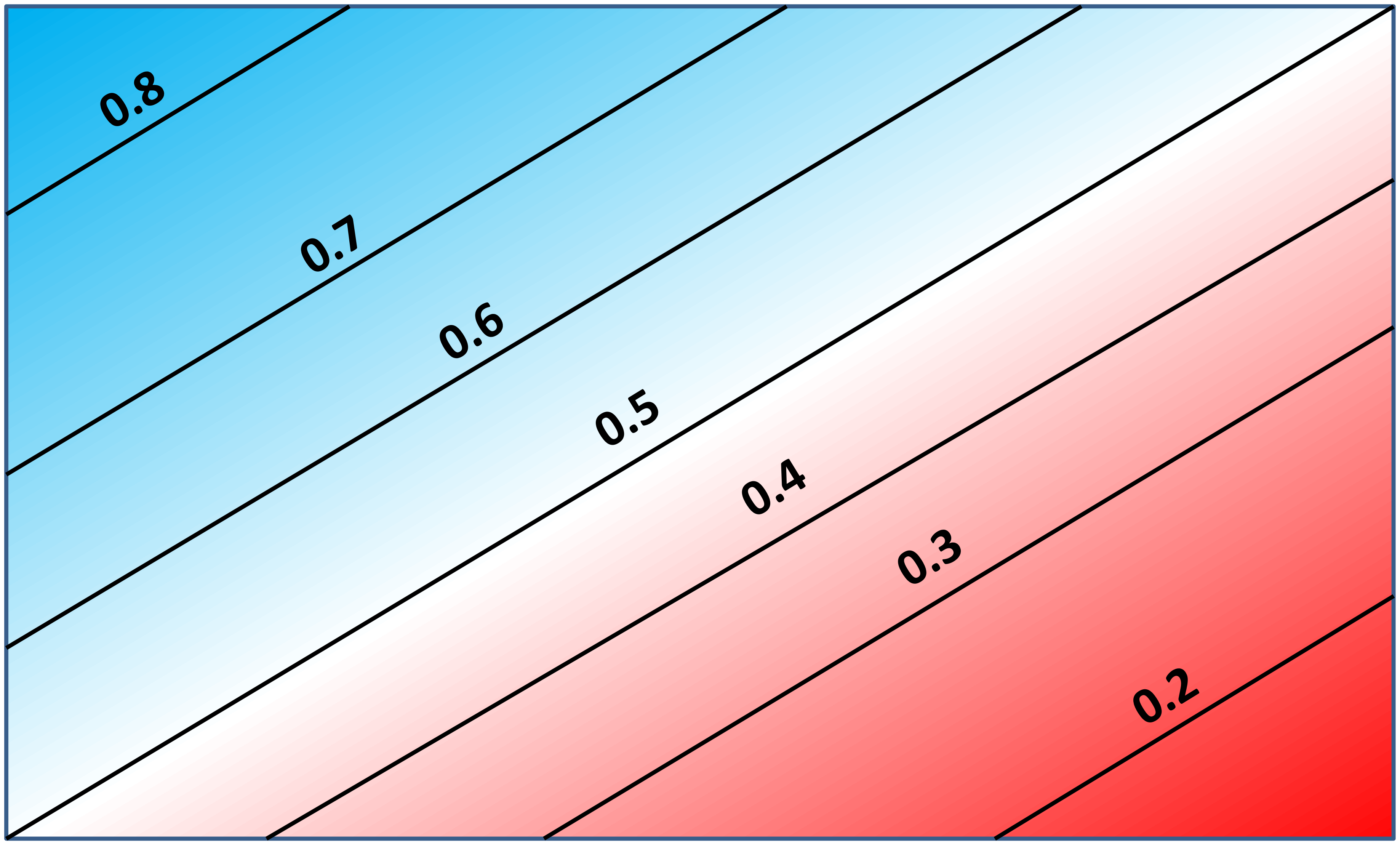
poor separation



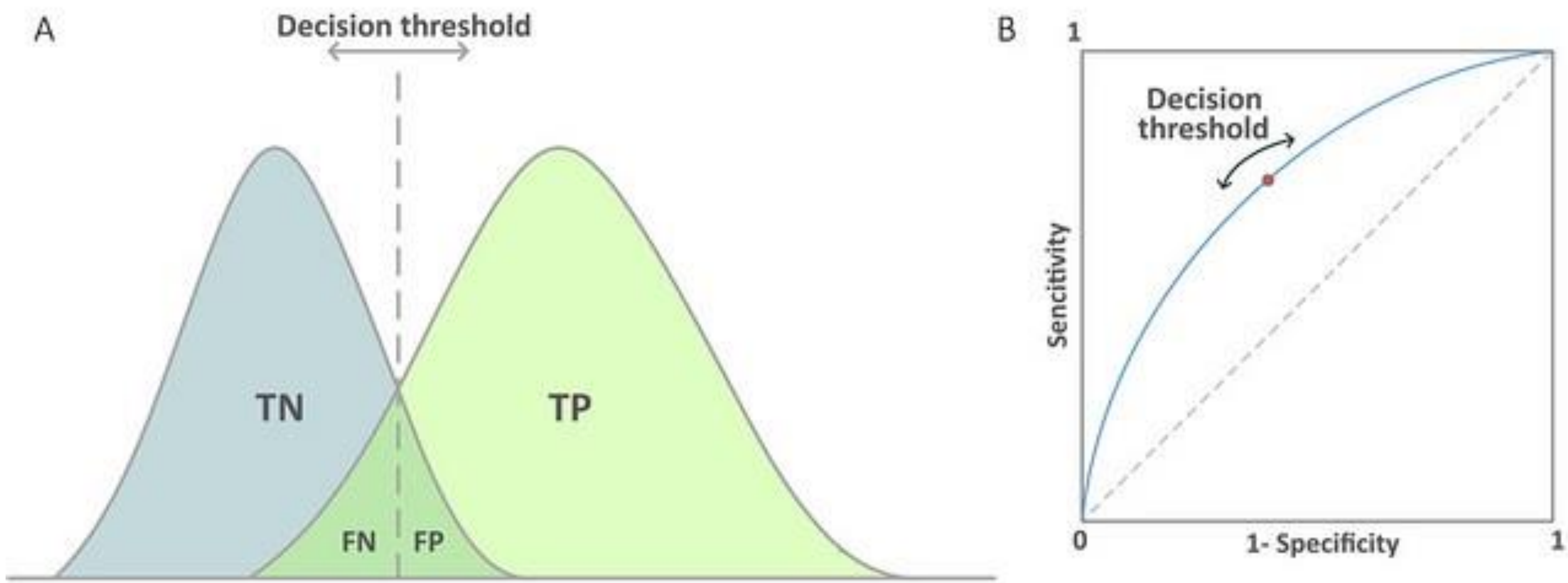
random separation

DICE ANALYTICS

Classifier Decision Threshold



Classifier Decision Threshold



The goal is to outline how to move the decision threshold to in Figure A, reducing false negatives or reducing false positives as per domain knowledge

Let’s Practice