Logistic Regression

Session-8/9

CLARUSWAY©
WAY TO REINVENT YOURSELF

SUMMARY of PREVIOUS CLASS

Regularization & Regression Project

- Multicollinearity
- Feature Scaling
- Regularization (Ridge, Lasso, Elastic Net)
- Cross-Validation and Grid Search
- One Hot Encoding
- Pipeline

What is Classification?

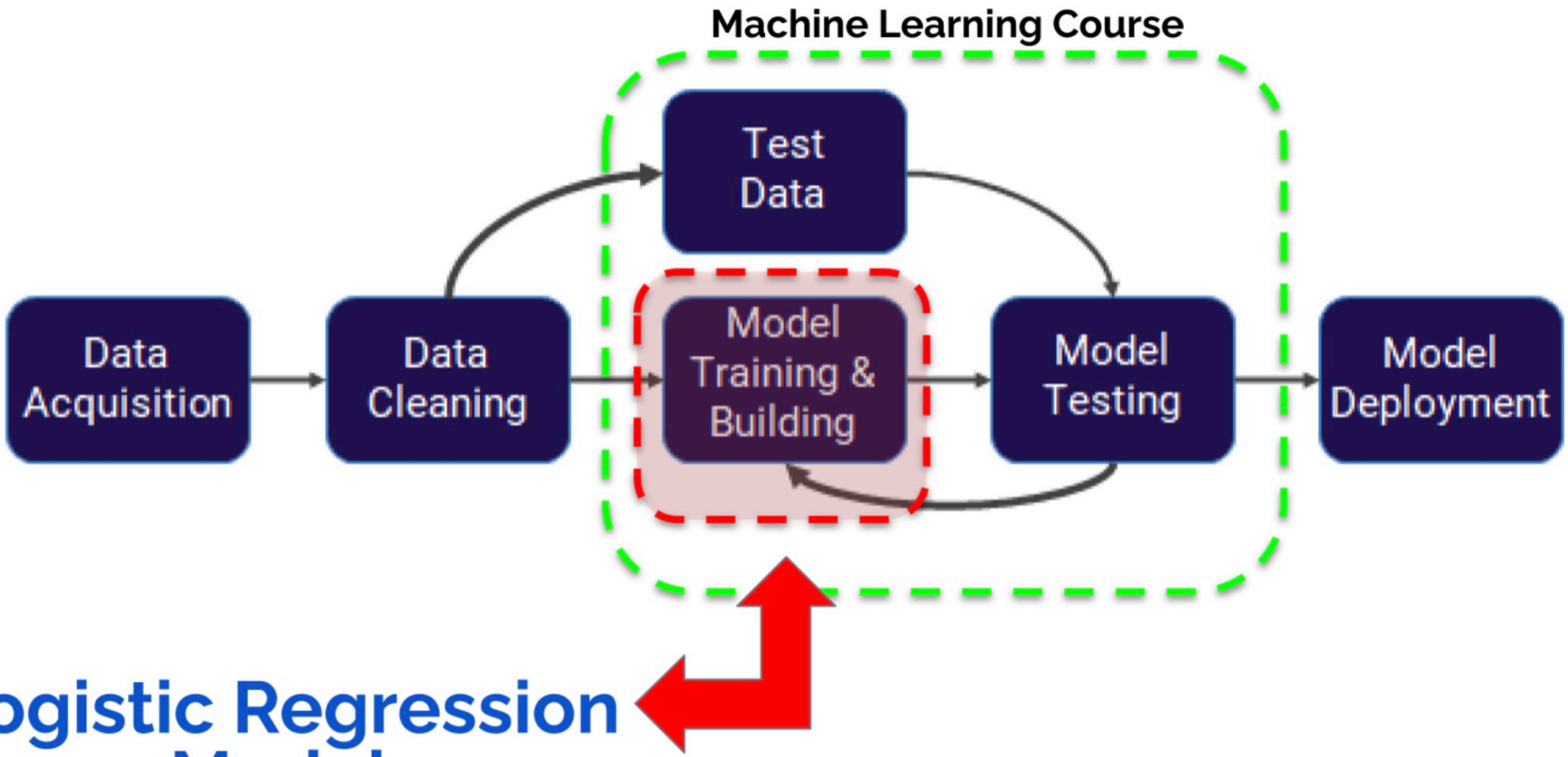
Classification vs Regression

Logistic Regression Theory

Classification Error Metrics

Logistic Regression with Python

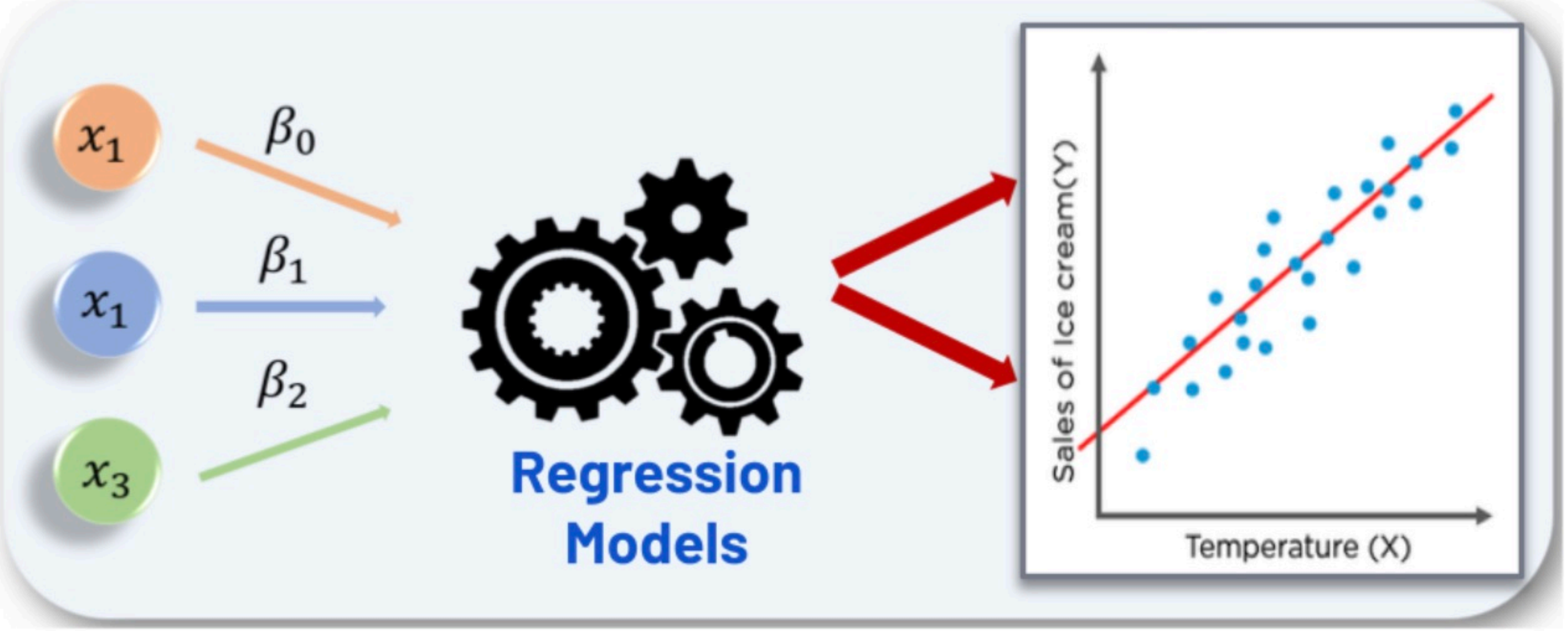
Where are we?

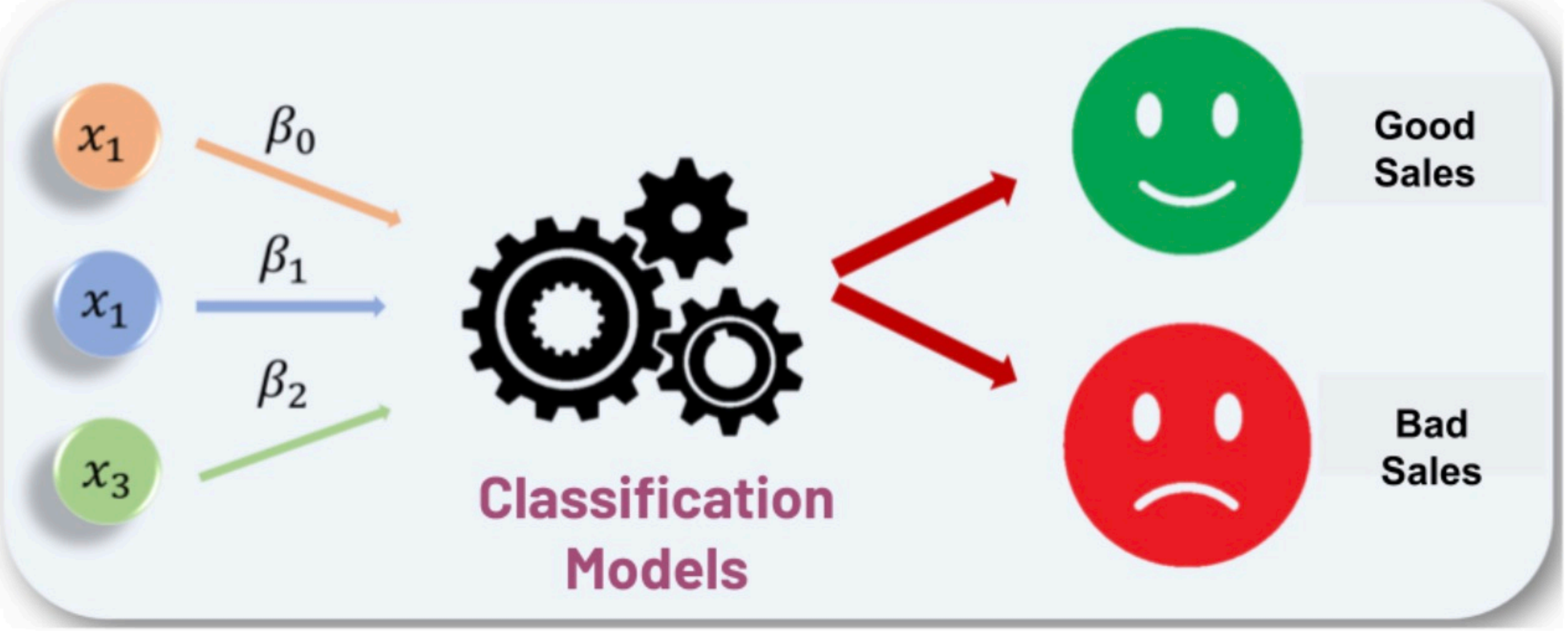


Logistic Regression Model

What is Classification?

Logistic Regression Theory



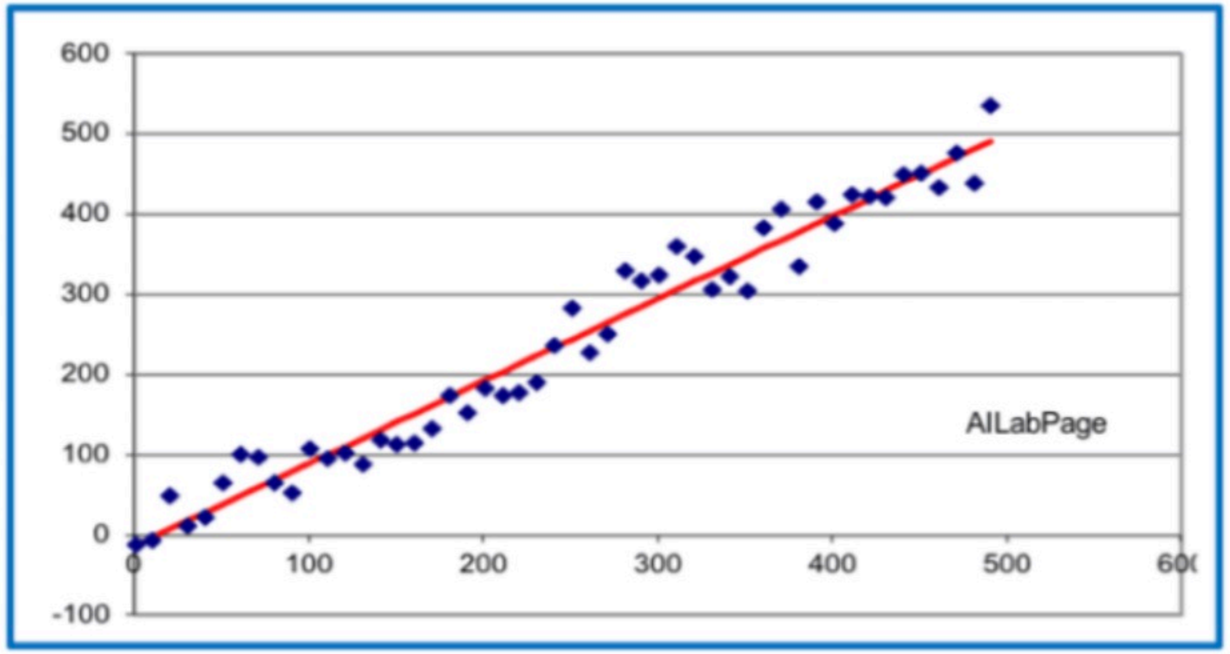


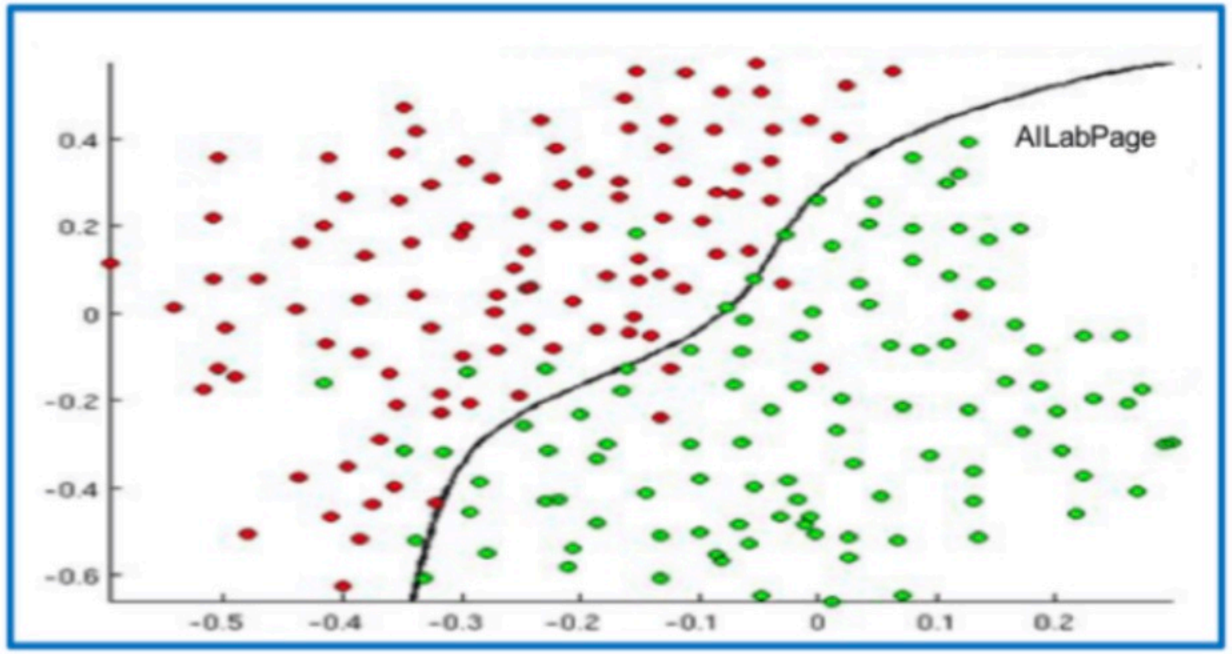
There are many cases where we deal with *input-output relationships*, such as *in* ***regression***, but the output variable does **not always have to be continuous**, sometimes this variable is **<u>categorical</u>**.

CLASSIFICATION

Classification vs Regression

Regression vs Classification





Regression

1. The system attempts to predict a value for an input based on past data.
2. Real number / Continuous numbers – Regression problem
3. Example – 1. Temperature for tomorrow

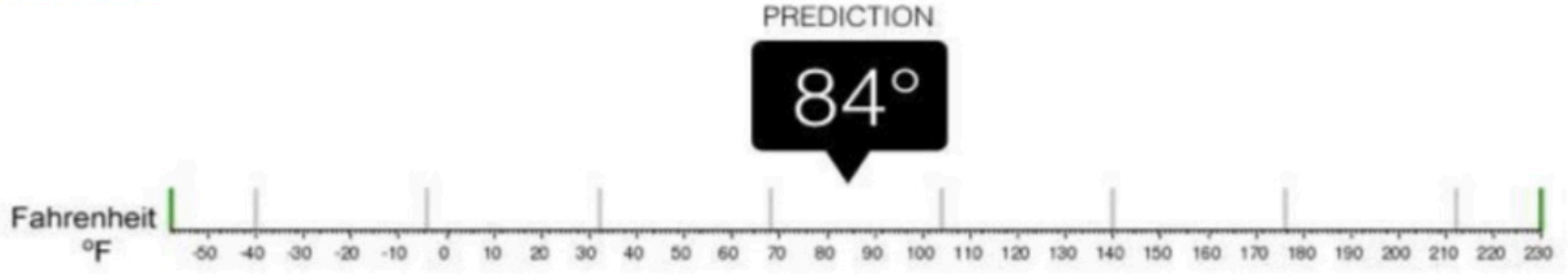
Classification

1. In classification, predictions are made by classifying them into different categories.
2. Discreate / categorical variable – Classification problem
3. Example – 1. Type of cancer 2. Cancer Y/N

Regression vs Classification

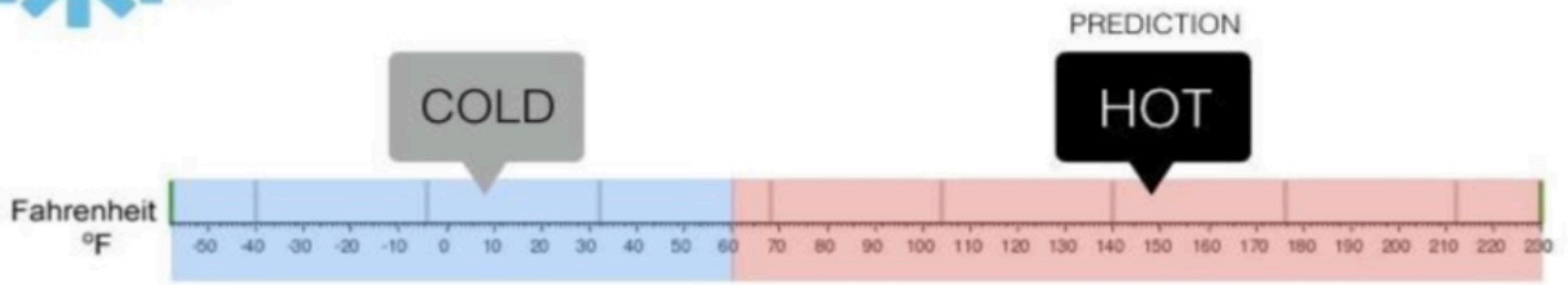
Regression

What is the temperature going to be tomorrow?

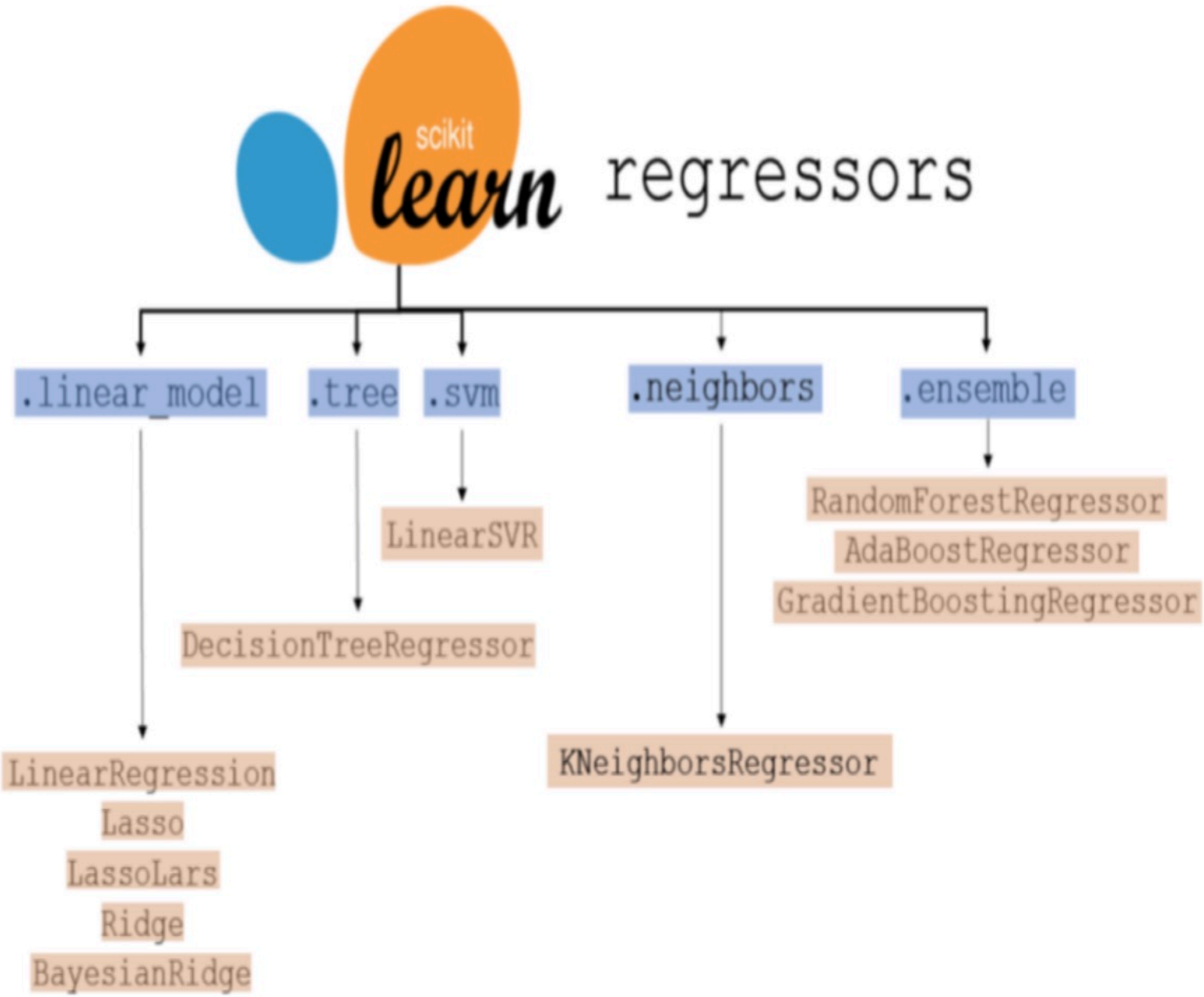


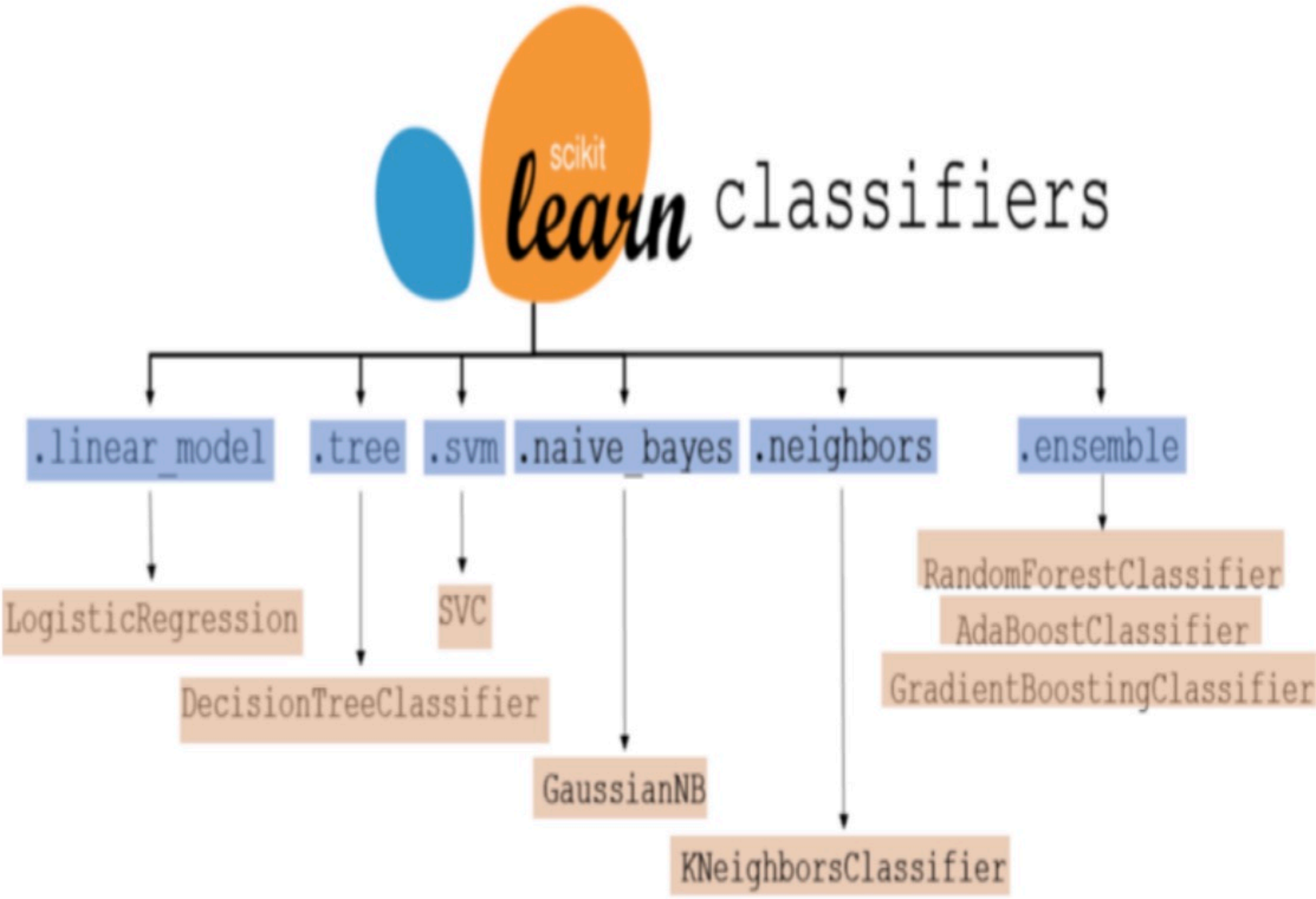
Classification

Will it be Cold or Hot tomorrow?



Regression vs Classification





Regression vs Classification

scikit learn metrics

commonly used metrics marked in green

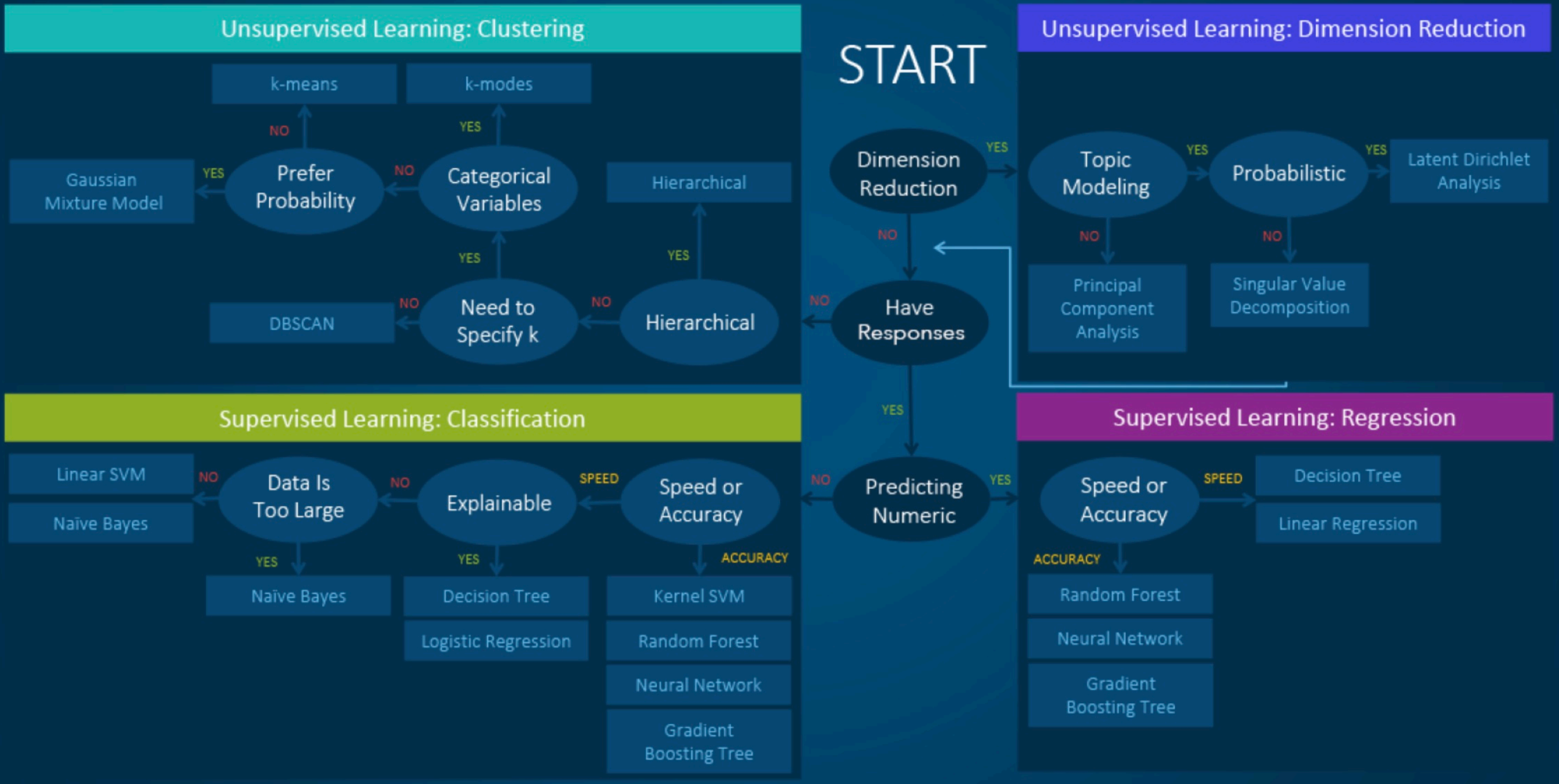
classification

- accuracy_score
- balanced_accuracy_score
- average_precision_score
- brier_score_loss
- f1_score
- log_loss
- precision_score
- recall_score
- jaccard_score
- roc_auc_score

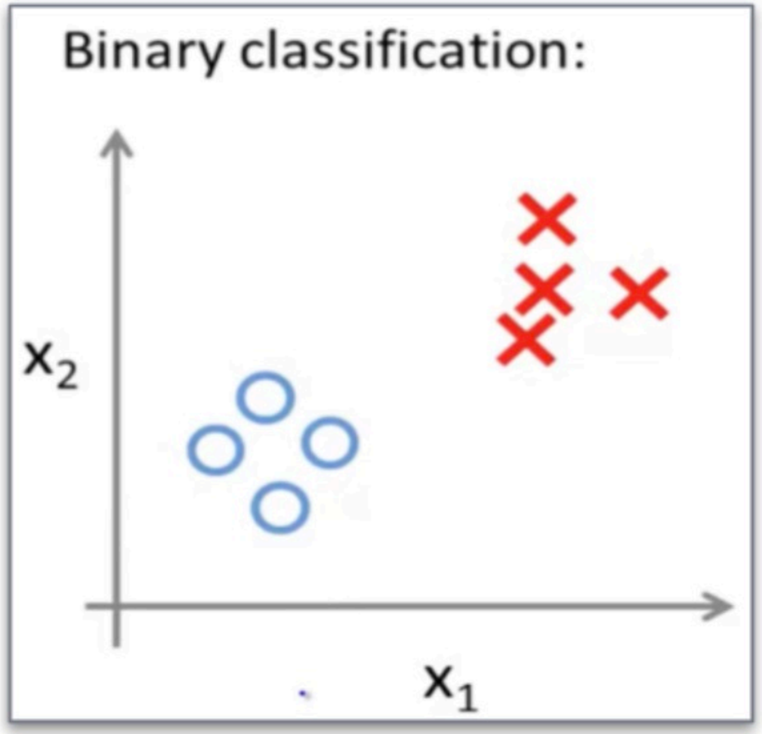
regression

- explained_variance_score
- max_error
- mean_absolute_error
- mean_squared_error
- mean_squared_log_error
- median_absolute_error
- r2_score
- mean_poisson_deviance
- mean_gamma_deviance

Machine Learning Algorithms Cheat Sheet



Logistic Regression Theory



Logistic Regression is a method for **Classification**.

Some examples of classification problems:

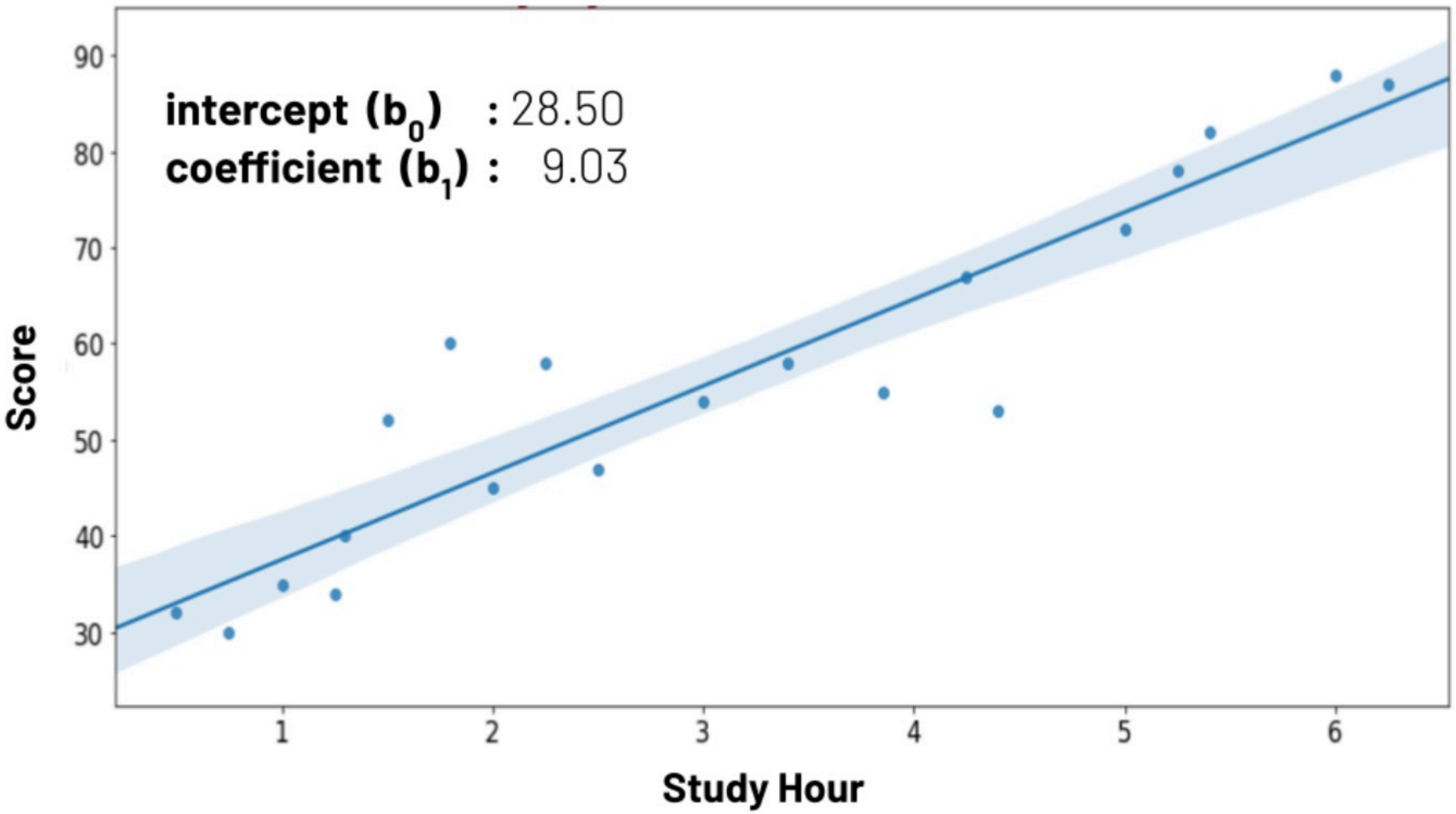
* Clinical trials (**Disease Diagnosis**),

* Scoring and **fraud detection** (where response is binary),

* Predicting the customer **churn**.

Logistic Regression Theory

Student	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
Study Hour	0.5	0.75	1.00	1.25	1.30	1.50	1.80	2.00	2.25	2.50	3.00	3.40	3.85	4.25	4.40	5.00	5.25	5.40	6.00	6.25
Score	32	30	35	34	40	52	60	45	58	47	54	58	55	67	53	72	78	82	88	87



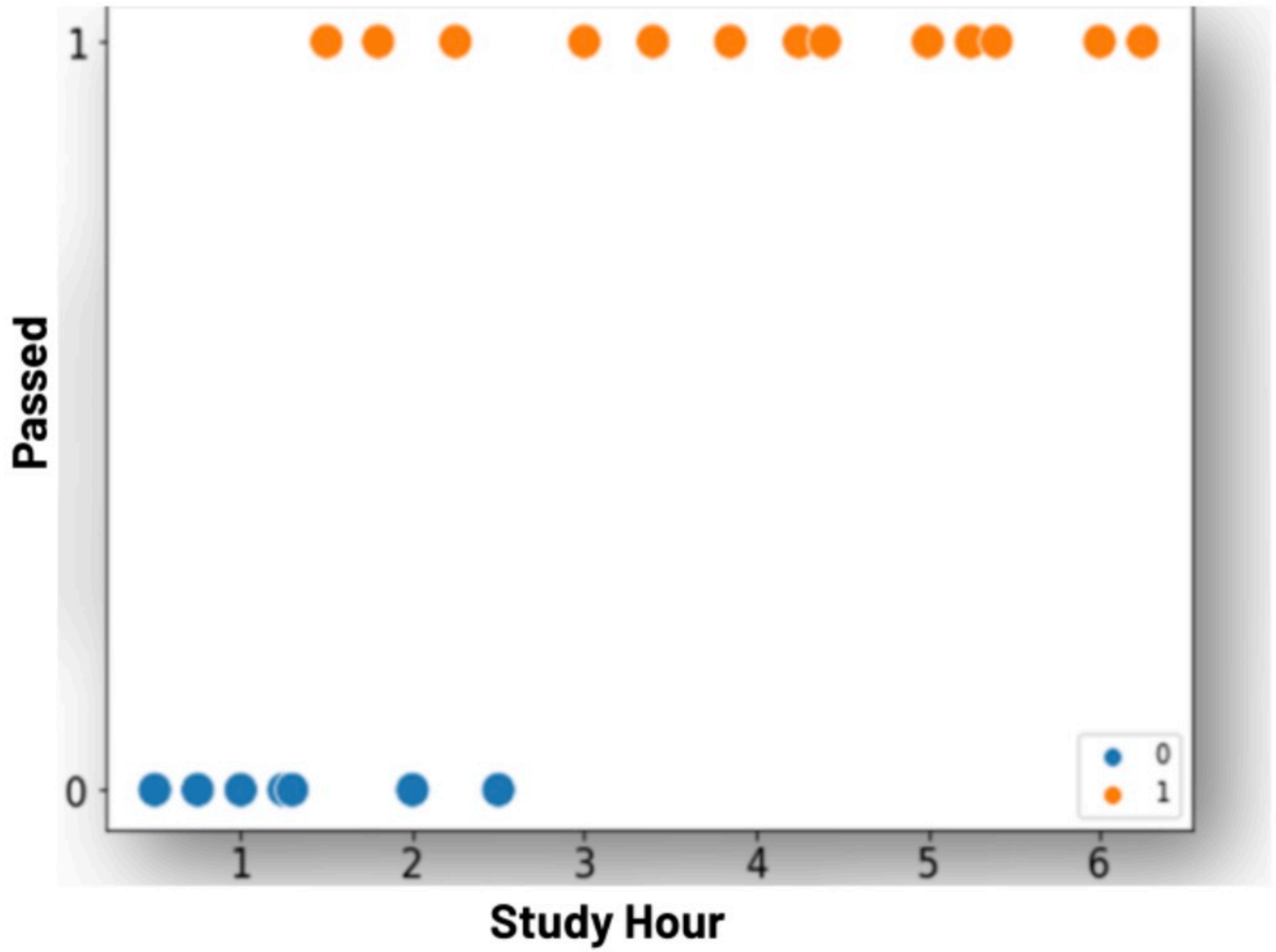
Regression Equation

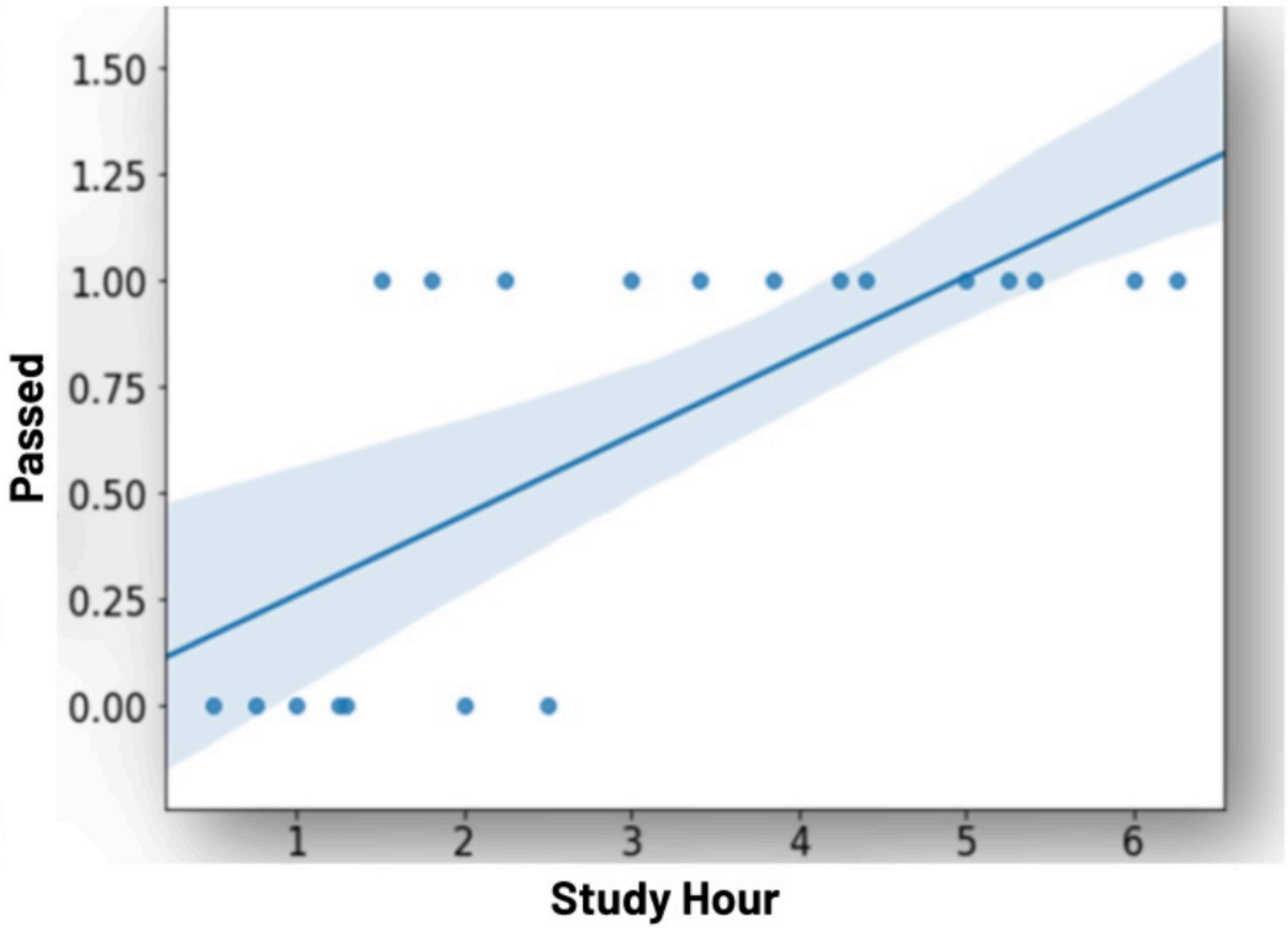
$$y = b_0 + b_1 * X$$

y = 28.50 + 9.03 * Study hour

Logistic Regression Theory

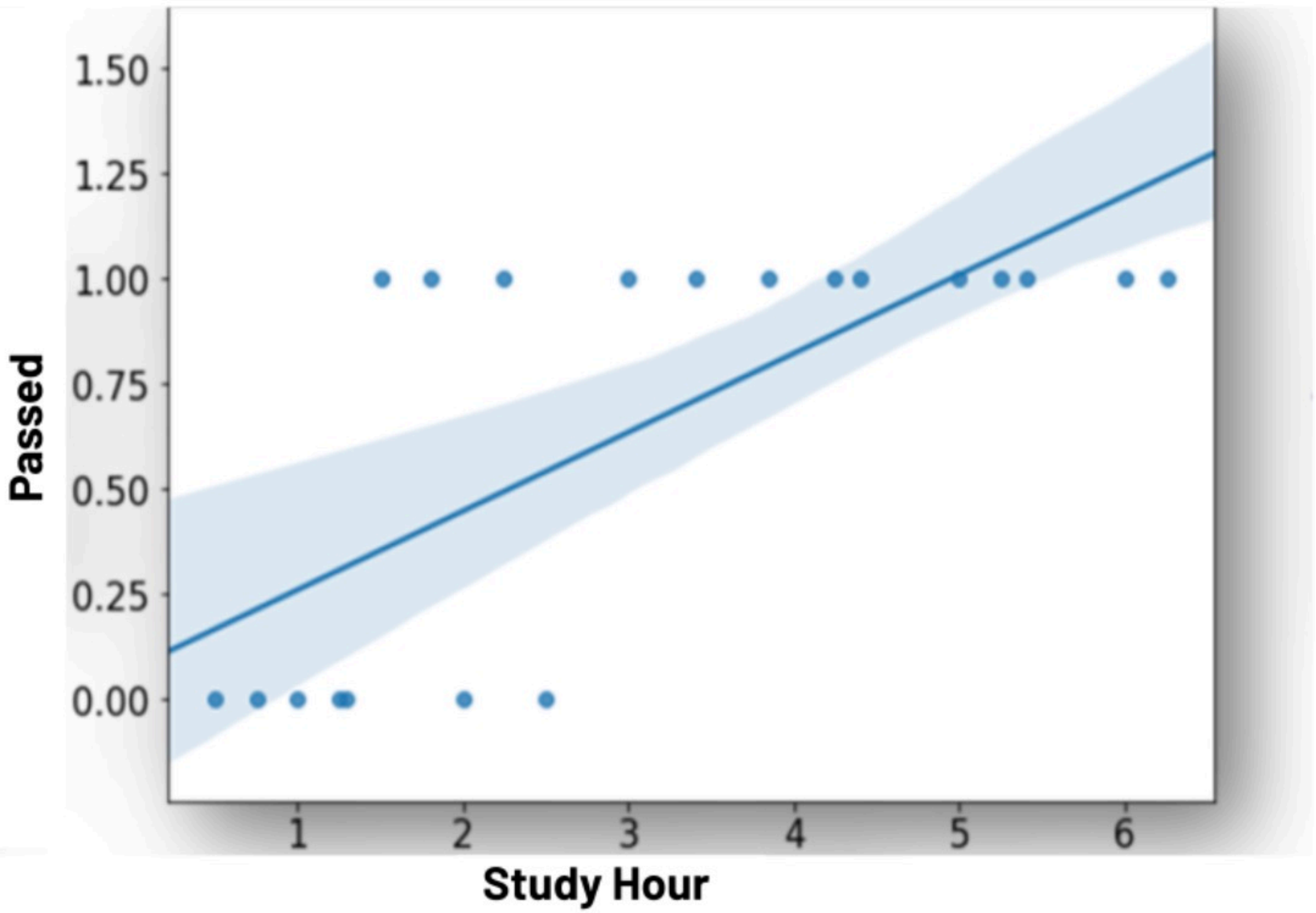
Student	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
Study Hour	0.5	0.75	1.00	1.25	1.30	1.50	1.80	2.00	2.25	2.50	3.00	3.40	3.85	4.25	4.40	5.00	5.25	5.40	6.00	6.25
Passed	0	0	0	0	0	1	1	0	1	0	1	1	1	1	1	1	1	1	1	1

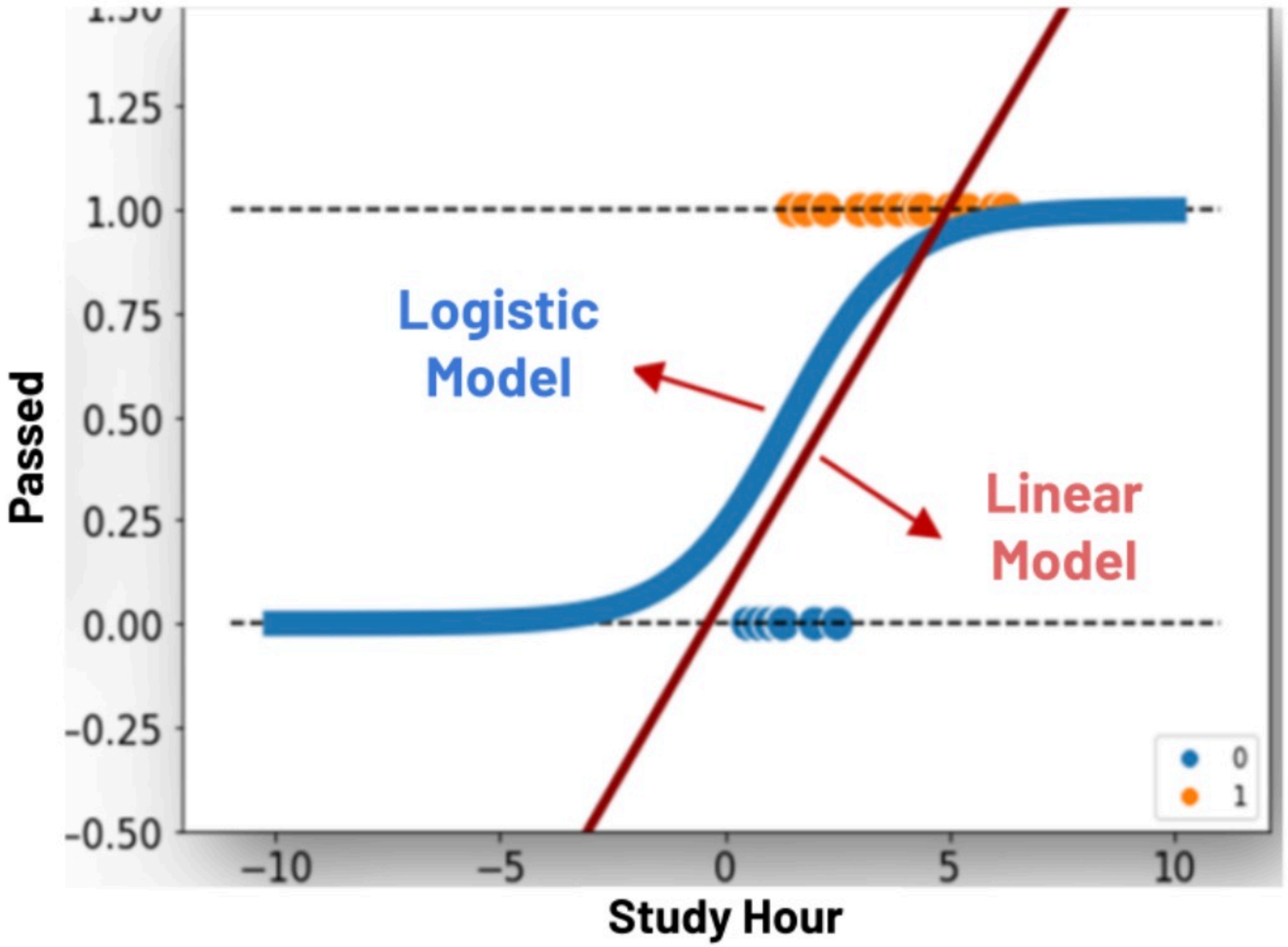




Logistic Regression Theory

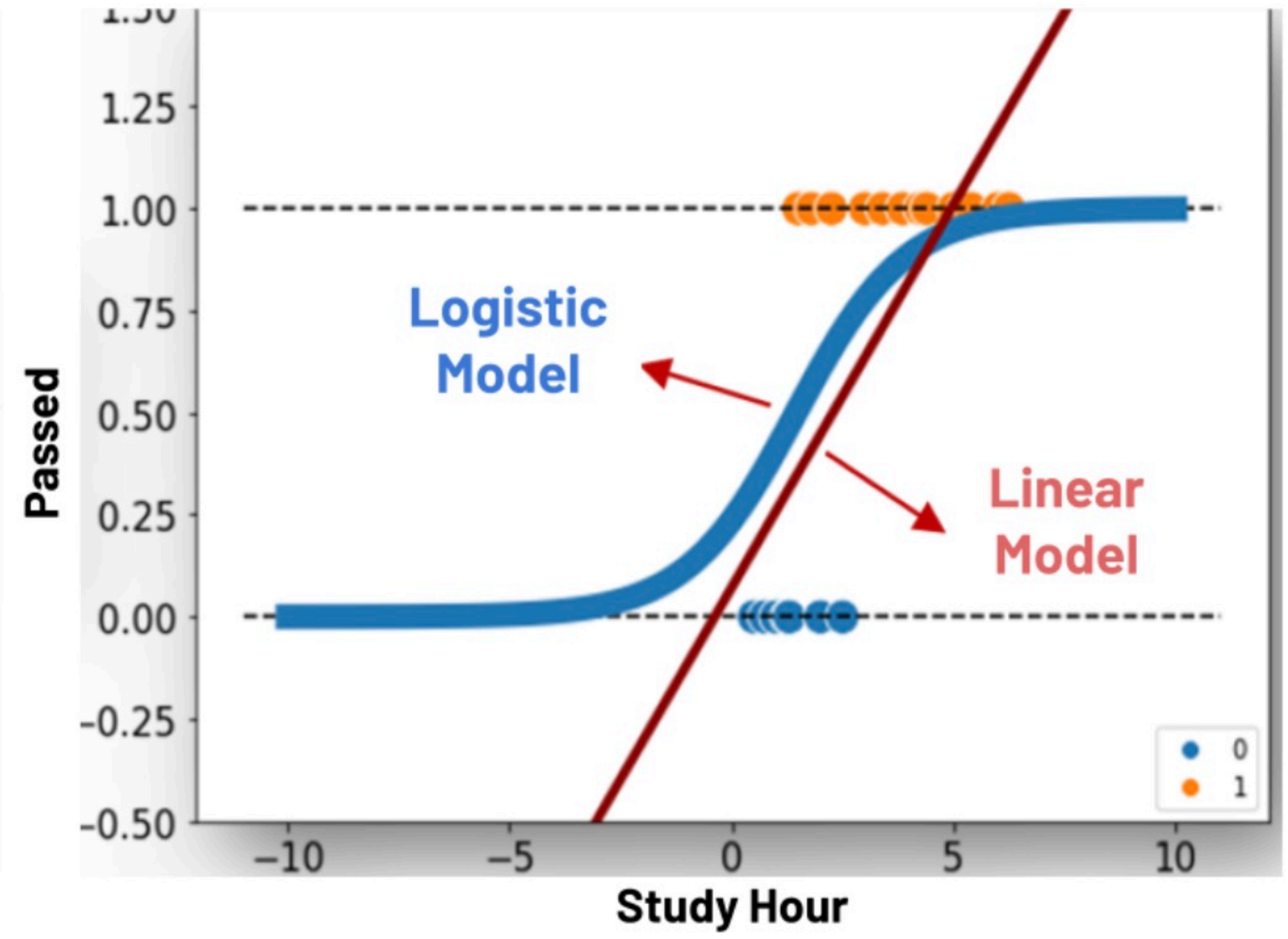
Student	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
Study Hour	0.5	0.75	1.00	1.25	1.30	1.50	1.80	2.00	2.25	2.50	3.00	3.40	3.85	4.25	4.40	5.00	5.25	5.40	6.00	6.25
Passed	0	0	0	0	0	1	1	0	1	0	1	1	1	1	1	1	1	1	1	1





Logistic Regression Theory

Student	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
Study Hour	0.5	0.75	1.00	1.25	1.30	1.50	1.80	2.00	2.25	2.50	3.00	3.40	3.85	4.25	4.40	5.00	5.25	5.40	6.00	6.25
Passed	0	0	0	0	0	1	1	0	1	0	1	1	1	1	1	1	1	1	1	1

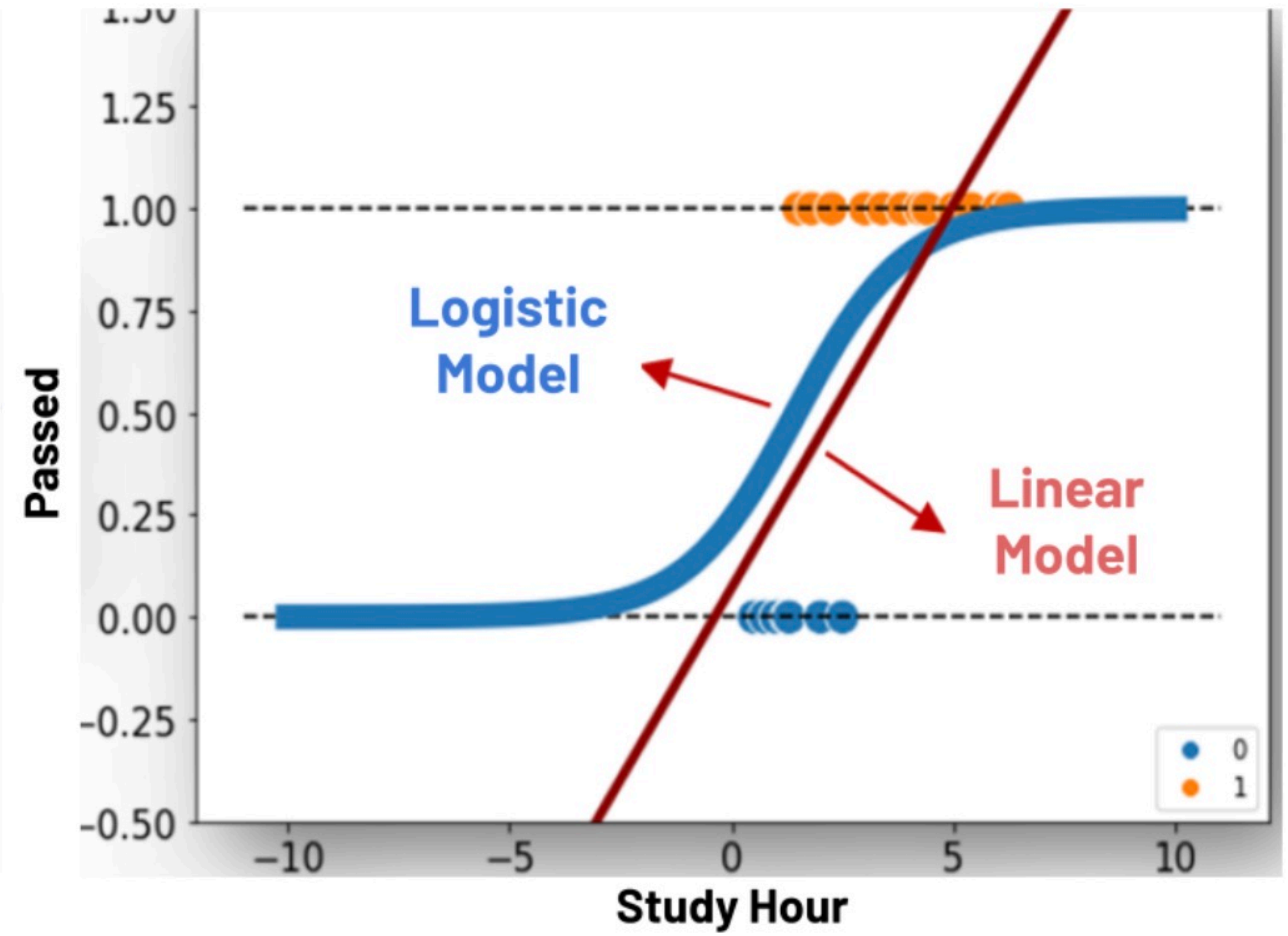


We get an output from the logistic regression model as **passed or not passed** (1 or 0).

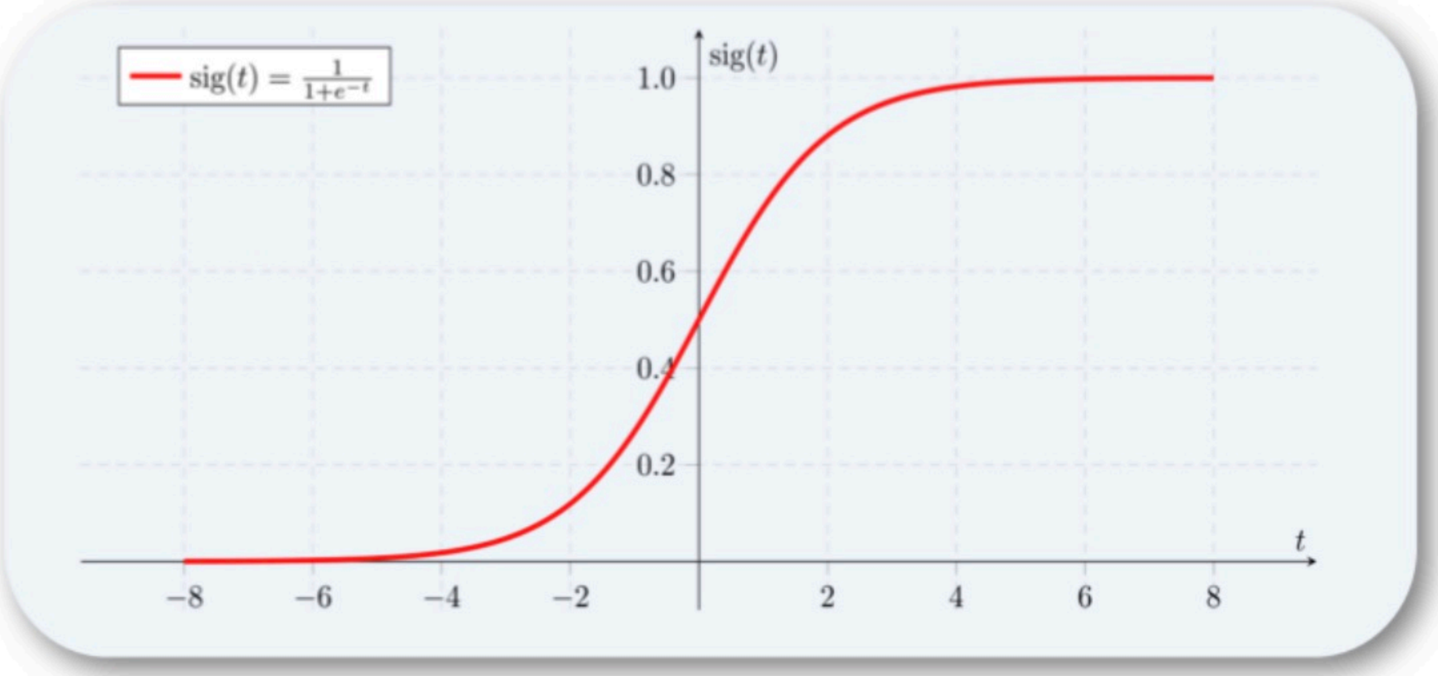
However, if we go into this diagram a little more, we will see that we pass one more function to squeeze the values between 0 and 1. This function is the **<u>sigmoid function</u>**.

Logistic Regression Theory

Student	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
Study Hour	0.5	0.75	1.00	1.25	1.30	1.50	1.80	2.00	2.25	2.50	3.00	3.40	3.85	4.25	4.40	5.00	5.25	5.40	6.00	6.25
Passed	0	0	0	0	0	1	1	0	1	0	1	1	1	1	1	1	1	1	1	1

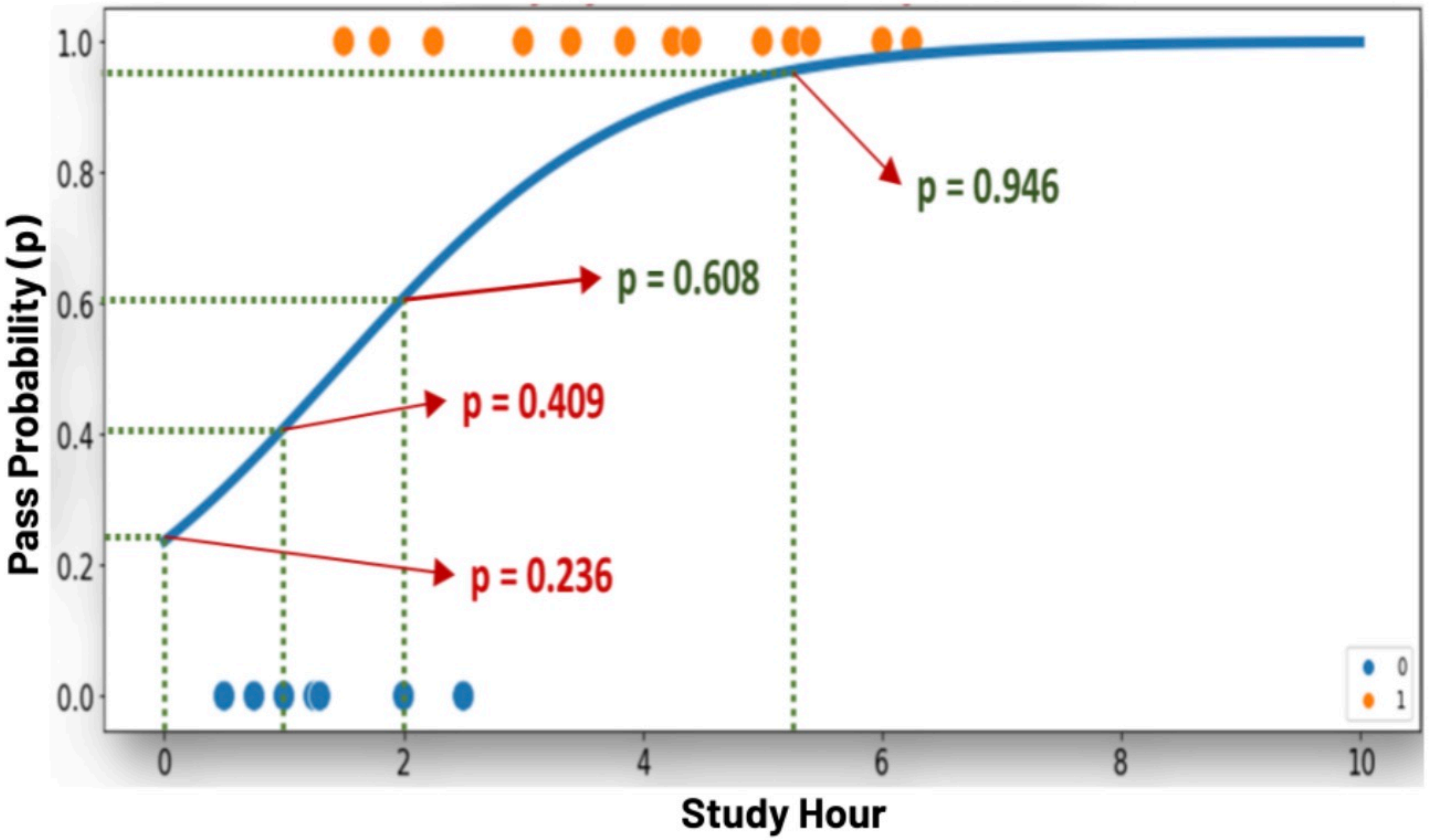


Sigmoid Function



Logistic Regression Theory

Student	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
Study Hour	0.5	0.75	1.00	1.25	1.30	1.50	1.80	2.00	2.25	2.50	3.00	3.40	3.85	4.25	4.40	5.00	5.25	5.40	6.00	6.25
Passed	0	0	0	0	0	1	1	0	1	0	1	1	1	1	1	1	1	1	1	1



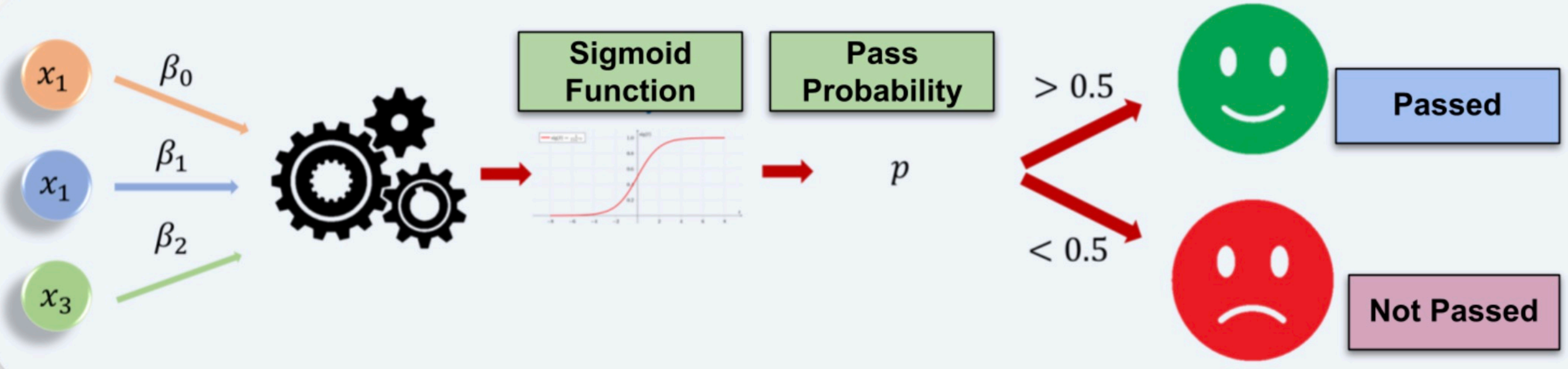
While the **probability of passing a course** for a student who

- **never studied is 23%**,
- **1h** studied is **41%,**
- **2h** studied is **61%,**
- **5h 30m** studies is **95%**

While estimating with **logistic regression**, we can work with the probability we specified, or we can also ask our model to return a value as **1 or 0** directly.

Logistic Regression Theory

Student	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
Study Hour	0.5	0.75	1.00	1.25	1.30	1.50	1.80	2.00	2.25	2.50	3.00	3.40	3.85	4.25	4.40	5.00	5.25	5.40	6.00	6.25
Passed	0	0	0	0	0	1	1	0	1	0	1	1	1	1	1	1	1	1	1	1



We will take the corresponding **probability value (p)** according to the values of X_1, ... X_n as the result from the sigmoid function.(This function takes any value and ***produces a value (p)*** between 0 and 1.)

Logistic Regression Theory

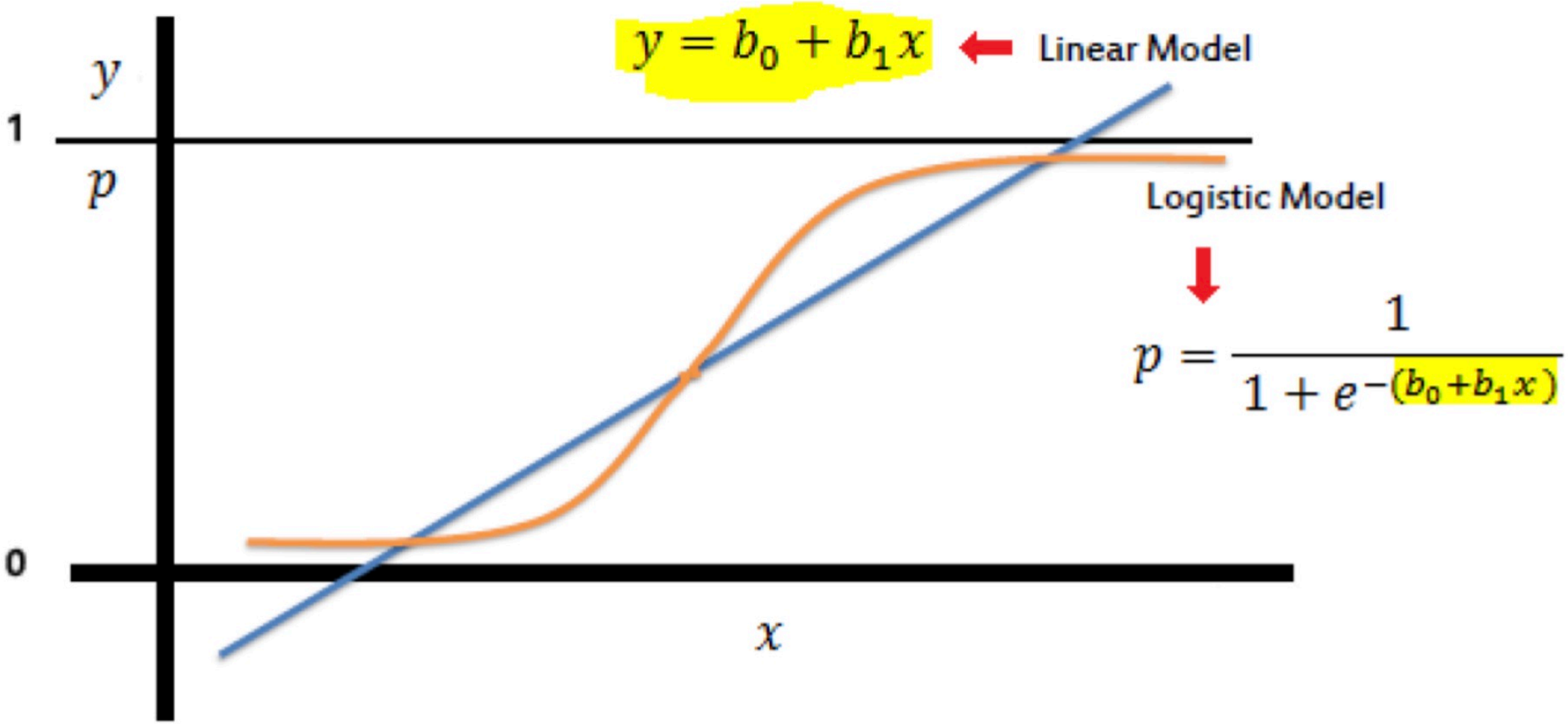
$$P(z) = \frac{1}{1 + e^{-z}}$$

```
def sigmoid(z):
   return 1 / (1 + np.exp(-z))
z = np.dot(X, weight)
h = sigmoid(z)
```

$$\text{income} = \beta_0 + \beta_1 \text{age} \longrightarrow P(\text{income} > 4000) = \frac{1}{1 + e^{-(\hat{\beta}_0 + \hat{\beta}_1 \text{age})}}$$

$$z = \beta_0 + \beta_1\, age$$

Logistic Regression Theory

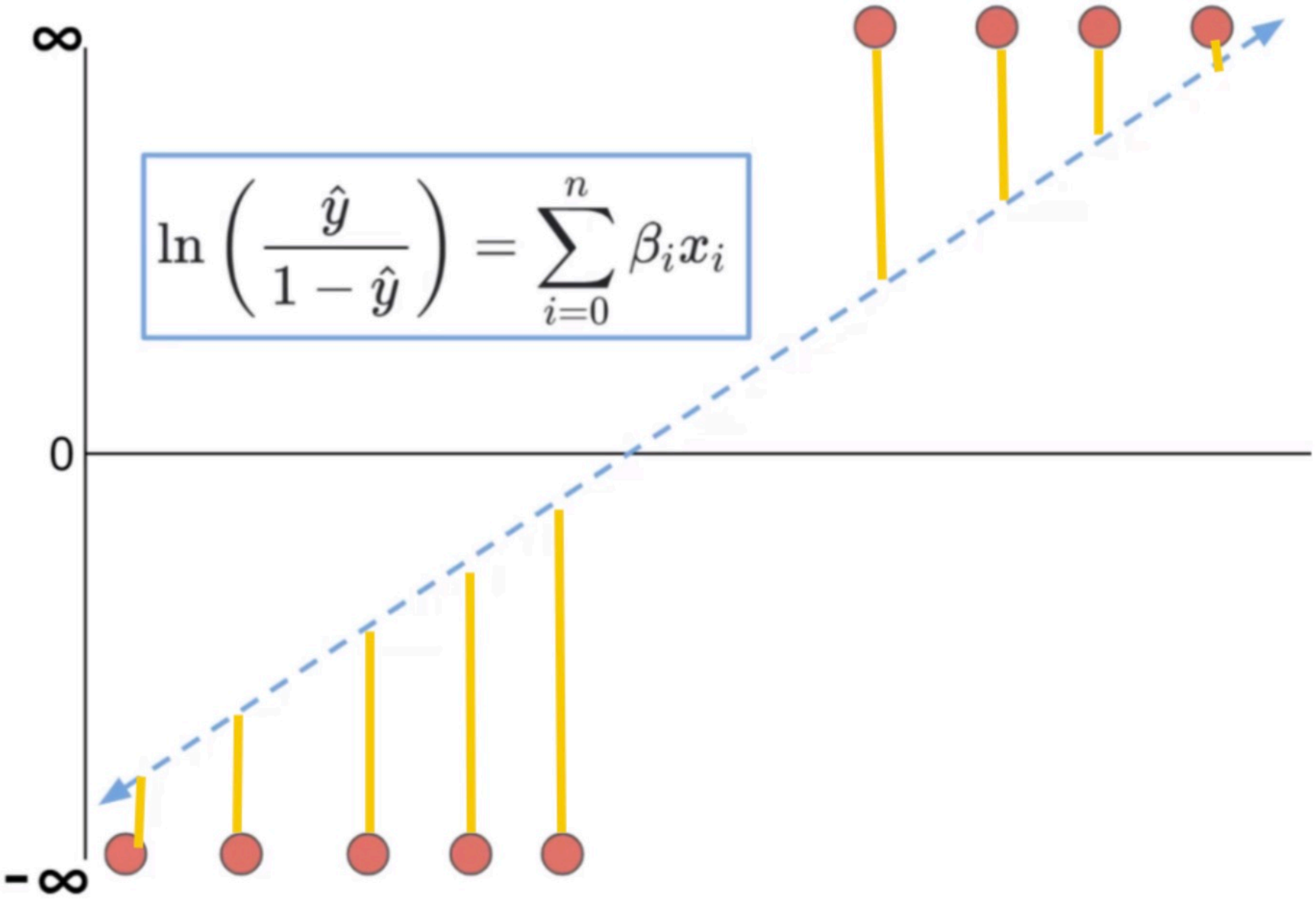


Logistic Regression Theory

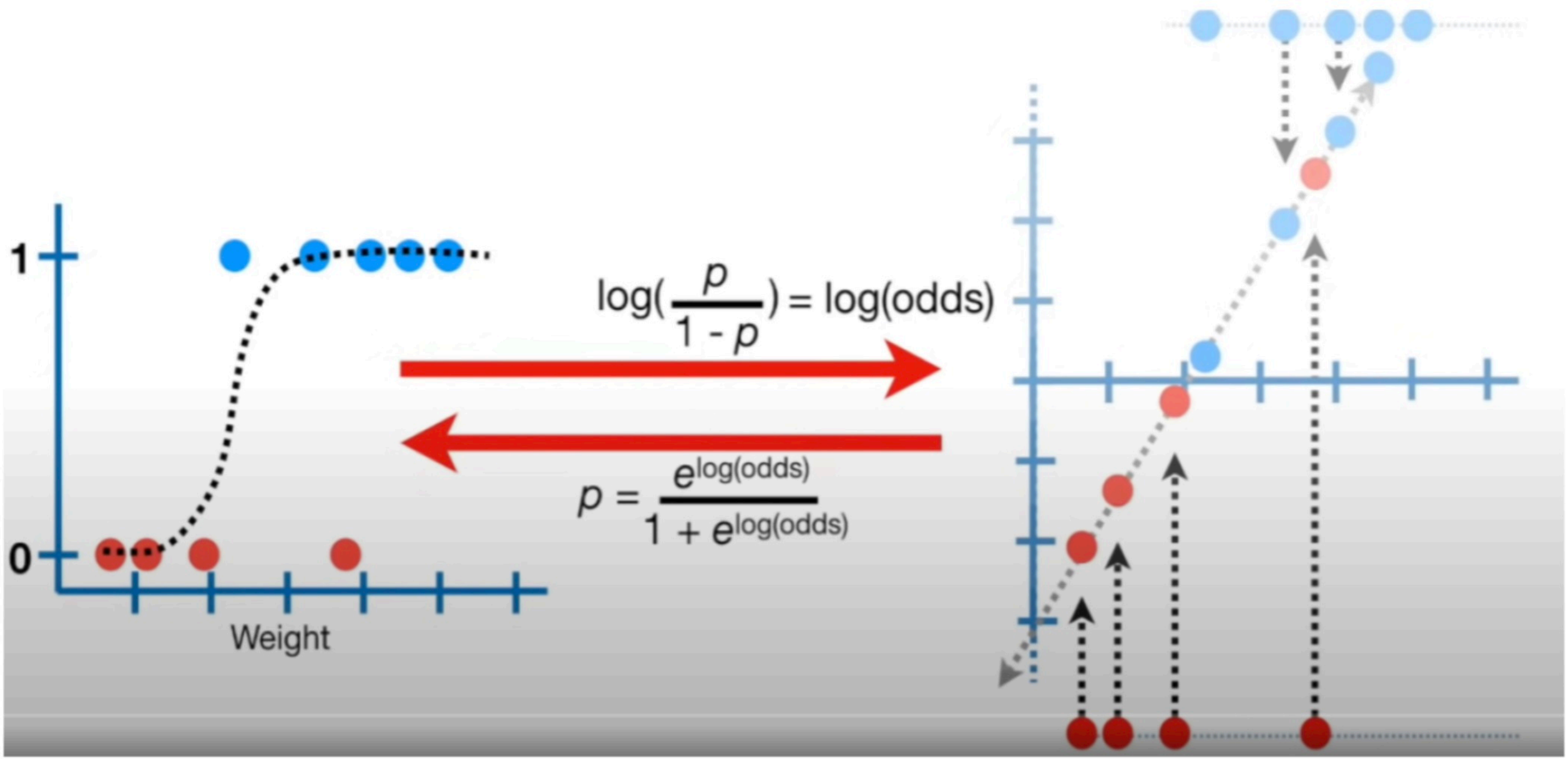
$$\lim_{p \to 1} ln(\frac{p}{1-p}) = \infty$$

$$ln(\frac{0.5}{1-0.5}) = 0$$

$$\lim_{p \to 0} ln(\frac{p}{1-p}) = -\infty$$



Logistic Regression Theory



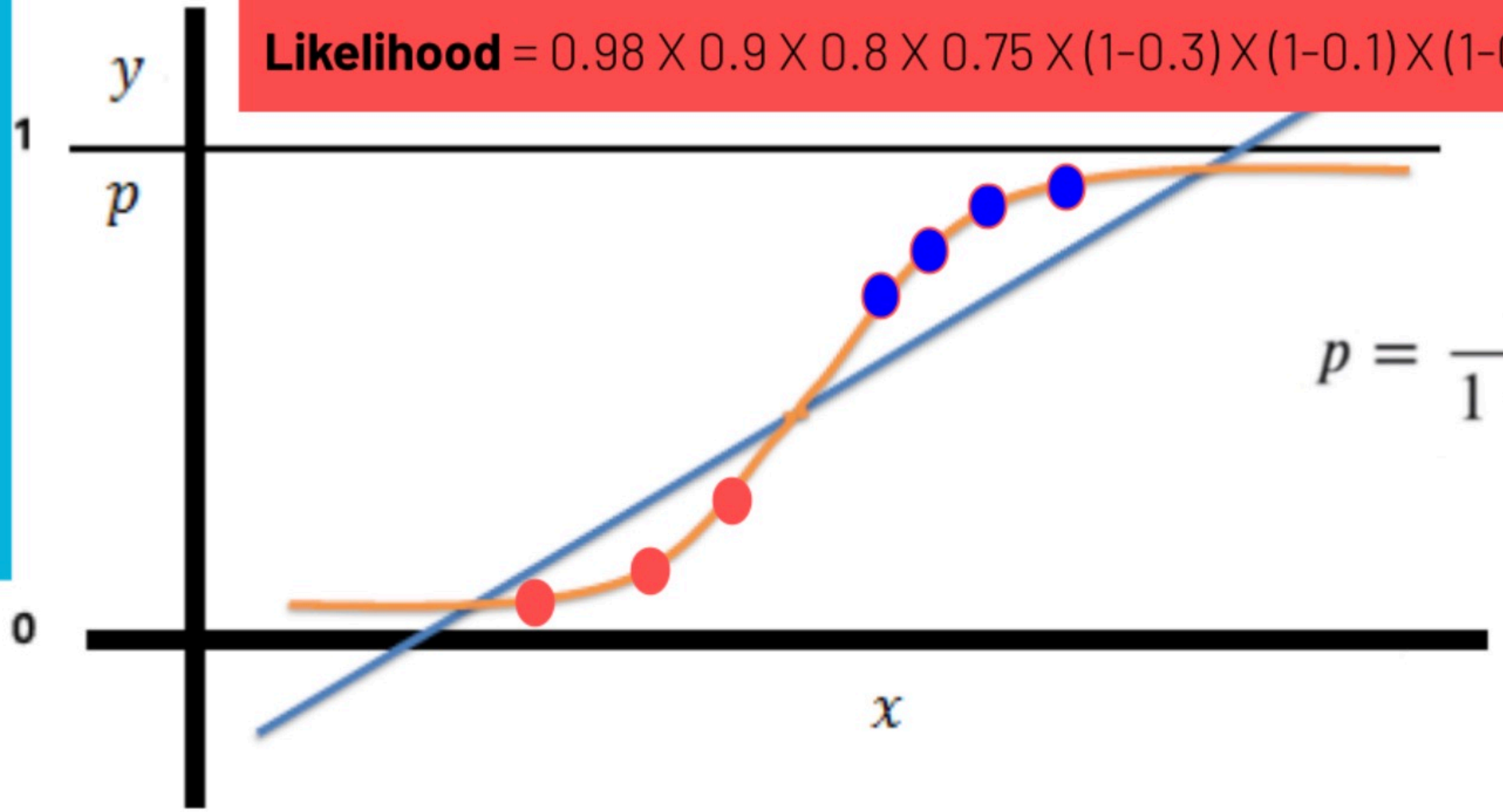
Picture credit :StatQuest

Logistic Regression Theory

Likelihood = Product of probabilities of belonging to a class

The MLE (*Maximum Likelihood Estimation*) seeks out a solution in which the predicted log-odds better represent the observed result.

Likelihood = 0.98 X 0.9 X 0.8 X 0.75 X (1-0.3) X (1-0.1) X (1-0.05) = **0.31**



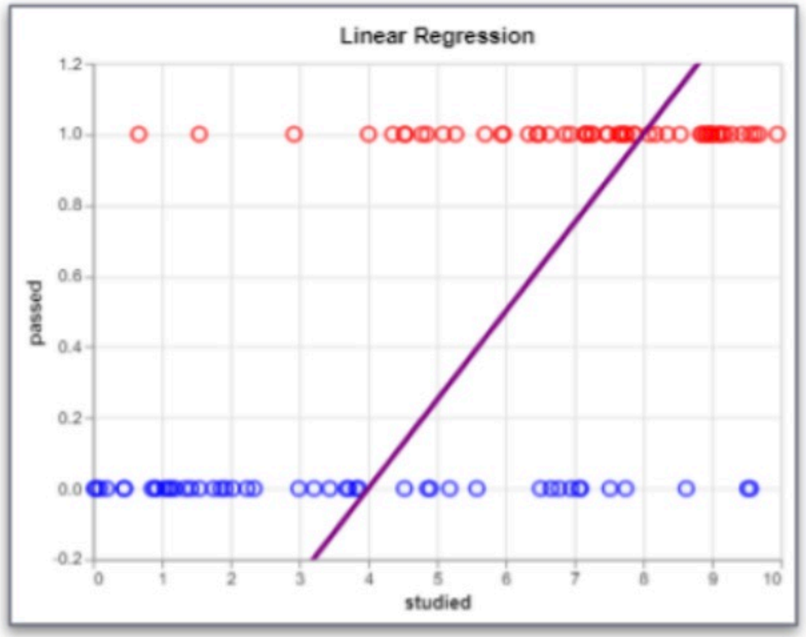
$$p = \frac{e^{\ln(odds)}}{1 + e^{\ln(odds)}}$$

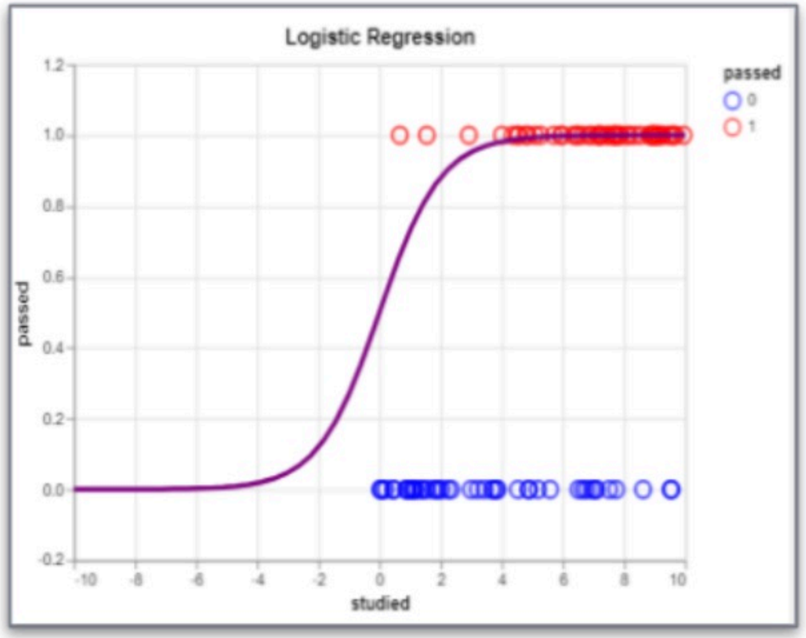
Cost Function (log loss)

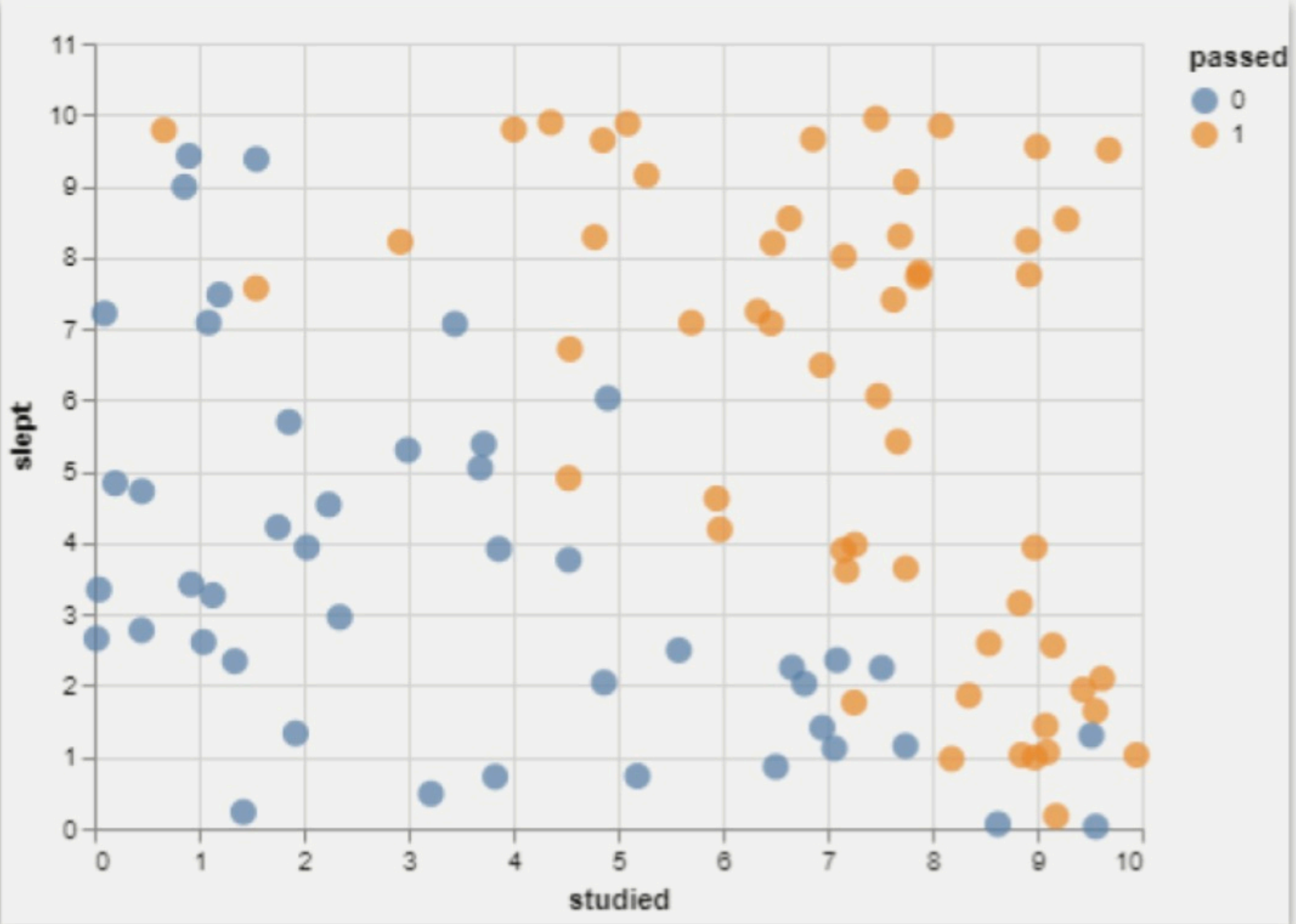
$$J(\mathbf{x}) = -\frac{1}{m}\sum_{j=1}^{m}\left(y^j \log\left(\frac{1}{1+e^{-\sum_{i=0}^{n}\beta_i x_i^j}}\right) + (1-y^j)\log\left(1-\frac{1}{1+e^{-\sum_{i=0}^{n}\beta_i x_i^j}}\right)\right)$$

Logistic Regression Theory

	studied	slept	passed
0	4.855064	9.639962	1
1	8.625440	0.058927	0
2	3.828192	0.723199	0
3	7.150955	3.899420	1
4	6.477900	8.198181	1
...	...	...	...
95	0.022280	2.658428	0
96	7.630637	7.405351	1
97	3.684997	5.049965	0
98	7.484260	6.059396	1
99	2.030708	3.937267	0







CLASSIFICATION

Logistic Regression Theory

Pros & Cons

Pros:

– Works well with linearly separable data

– Simple to interpret and easy to implement

– Both training and prediction are fast

– No assumptions about distributions of classes in feature space.

Cons:

– Prone to overfitting if data is small and have many features

– Sensitive to number of categorical features

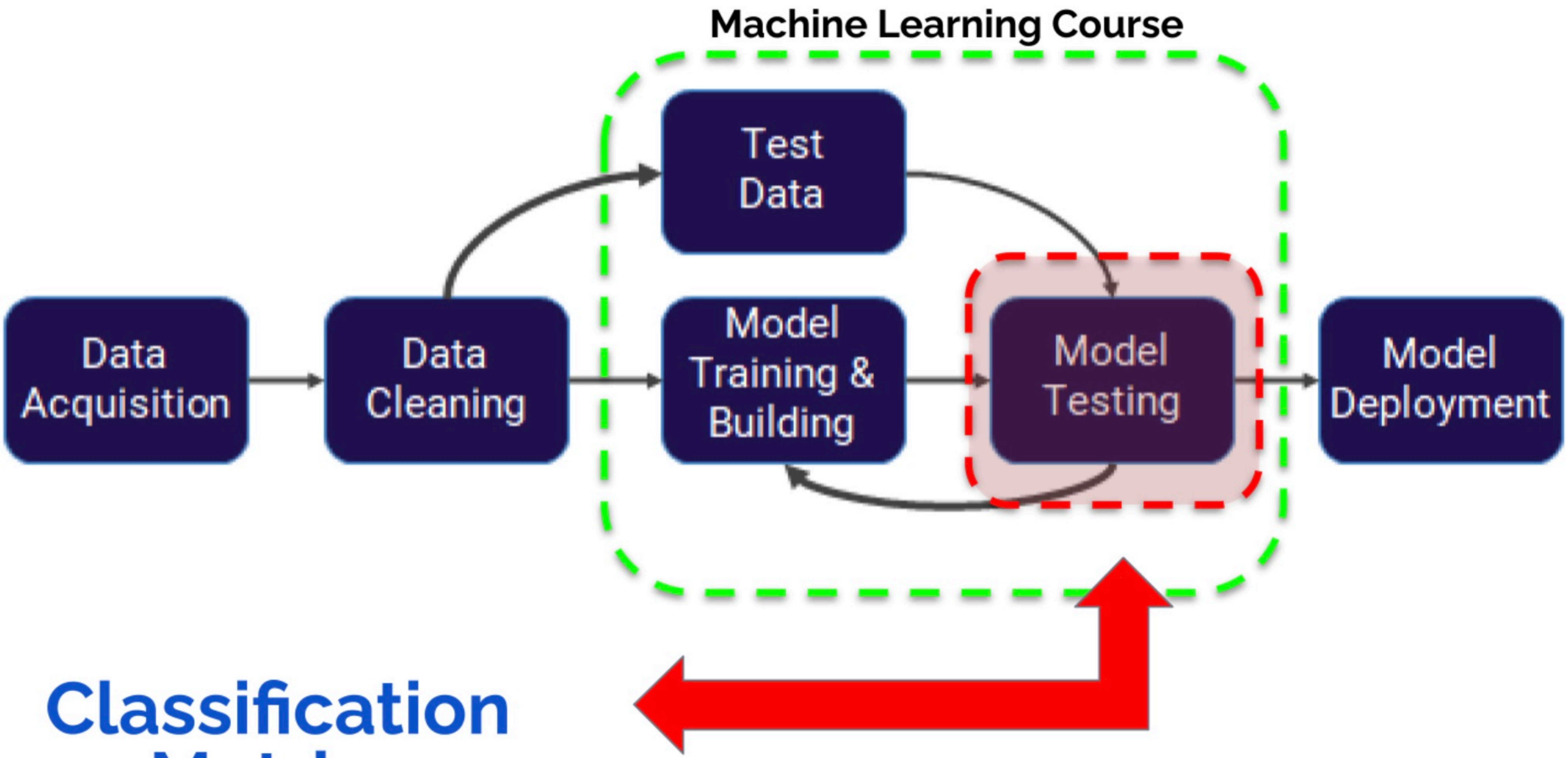
– Linear decision boundaries

– Needs correlation between features and target

Evaluating Performance

* *Classification Error Metrics*

Where are we?



Classification Metrics

Classification Error Metrics -Confusion Matrix-

		Predicted class	
		+	-
Actual class	+	**TP** True Positives	**FN** False Negatives Type II error
	-	**FP** False Positives Type I error	**TN** True Negatives

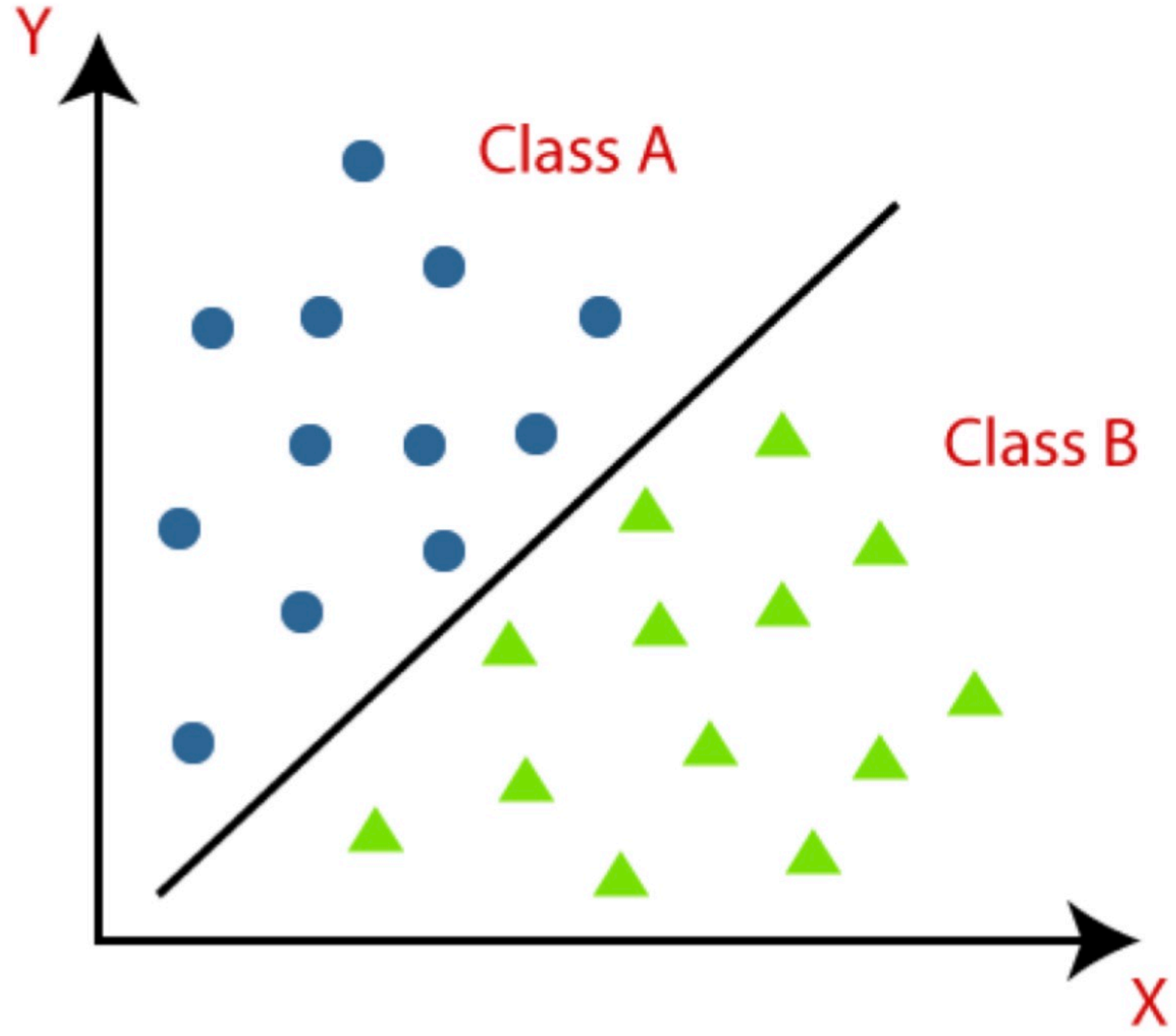
In Python

```
15]: confusion_matrix(y, y_pred)

15]: array([[448,  52],
            [121, 147]], dtype=int64)
```

Actual Cancer:	1	1	0	0	1	0	0	0	0	1 ....
Predicted Cancer:	0	1	1	0	1	0	0	0	0	0 ....
	✗	✓	✗	✓	✓	✓	✓	✓	✓	✗
	FN	**TP**	**FP**	**TN**	**TP**	**TN**	**TN**	**TN**	**TN**	**FN** ...

Classification Error Metrics



Confusion Matrix

* Accuracy
* Recall
* Precision
* F1-Score

Classification Error Metrics -Accuracy-

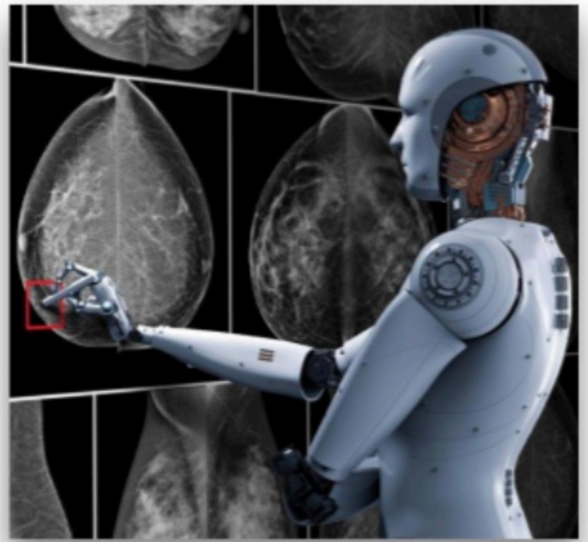
* Most *basic* and *common* metric to evaluate our model.

Actual Labels:	1	1	0	1	0	0	0	1	1	0
Predicted Labels:	0	1	1	1	0	0	0	1	1	1
	✗	✓	✗	✓	✓	✓	✓	✓	✓	✗

$$\frac{\text{All correctly predicted values (7)}}{\text{All predicted values (10)}} \times 100 \Rightarrow \textbf{Accuracy = 70 \%}$$

Why is Accuracy <u>not</u> a good metric?

<u>Cancer Detection</u> Example:



Actual Cancer:	1	1	0	0	1	0	0	0	0	1	....
Predicted Cancer:	0	1	1	0	1	0	0	0	0	0	....
	✗	✓	✗	✓	✓	✓	✓	✓	✓	✗	

$$\frac{\text{All correctly predicted values (60)}}{\text{All predicted values (63)}} \times 100 \rightarrow$$

Accuracy = 95 %

(PERFECT ????)

Accuracy is very high, but <u>missed 2 actual</u> patient.

Other Metrics to Evaluate our Models
-Sensitivity (Recall)-

		Predicted class	
		+	-
Actual class	+	**TP** True Positives	**FN** False Negatives Type II error
	-	**FP** False Positives Type I error	**TN** True Negatives

- Correctly predicted as **positives** compared to **total number of positives**.
- Fraction of positives that were correctly identified.

$$\textbf{Sensitivity} = \frac{\textbf{TP}}{\textbf{TP+FN}} = \frac{2}{2+2}$$

Actual Cancer:	1	1	0	0	1	0	0	0	0	1	....
Predicted Cancer:	0	1	1	0	1	0	0	0	0	0	....
	FN	TP	FP	TN	TP	TN	TN	TN	TN	FN	...

Sensitivity
50 %

Other Metrics to Evaluate our Models
-Precision-

		Predicted class +	Predicted class -
Actual class	+	**TP** True Positives	**FN** False Negatives Type II error
	-	**FP** False Positives Type I error	**TN** True Negatives

- Correctly predicted as **positives** compared to **total predicted of positives.**
- Accuracy of positive predictions.

$$\textbf{Precision} = \frac{TP}{TP+FP} = \frac{2}{2+1}$$

Actual Cancer:	1	1	0	0	1	0	0	0	0	1 ....
Predicted Cancer:	0	1	1	0	1	0	0	0	0	0 ....
	✗	✓	✗	✓	✓	✓	✓	✓	✓	✗
	FN	TP	FP	TN	TP	TN	TN	TN	TN	FN ...

Precision 67 %

Other Metrics to Evaluate our Models -Specificity-

		Predicted class	
		+	-
Actual class	+	**TP** True Positives	**FN** False Negatives Type II error
	-	**FP** False Positives Type I error	**TN** True Negatives

Correctly predicted as **negatives** compared to **total number of negatives.**

$$\textbf{Specificity} = \frac{TN}{TN+FP} = \frac{58}{58+1}$$

Actual Cancer:	1	1	0	0	1	0	0	0	0	1	....
Predicted Cancer:	0	1	1	0	1	0	0	0	0	0	....
	FN	TP	FP	TN	TP	TN	TN	TN	TN	FN	...

Specificity 98 %

Other Metrics to Evaluate our Models
-F1 Score-

		Predicted class +	Predicted class -
Actual class	+	**TP** True Positives	**FN** False Negatives Type II error
	-	**FP** False Positives Type I error	**TN** True Negatives

- **Harmonic Mean** of Precision and Recall.
- Mostly **<u>used for unbalanced</u>** distribution.

$$F_1 = 2 * \frac{precision * recall}{precision + recall}$$

Actual Cancer:	1	1	0	0	1	0	0	0	0	1	....
Predicted Cancer:	0	1	1	0	1	0	0	0	0	0	....
	✗	✓	✗	✓	✓	✓	✓	✓	✓	✗	
	FN	**TP**	**FP**	**TN**	**TP**	**TN**	**TN**	**TN**	**TN**	**FN**	...

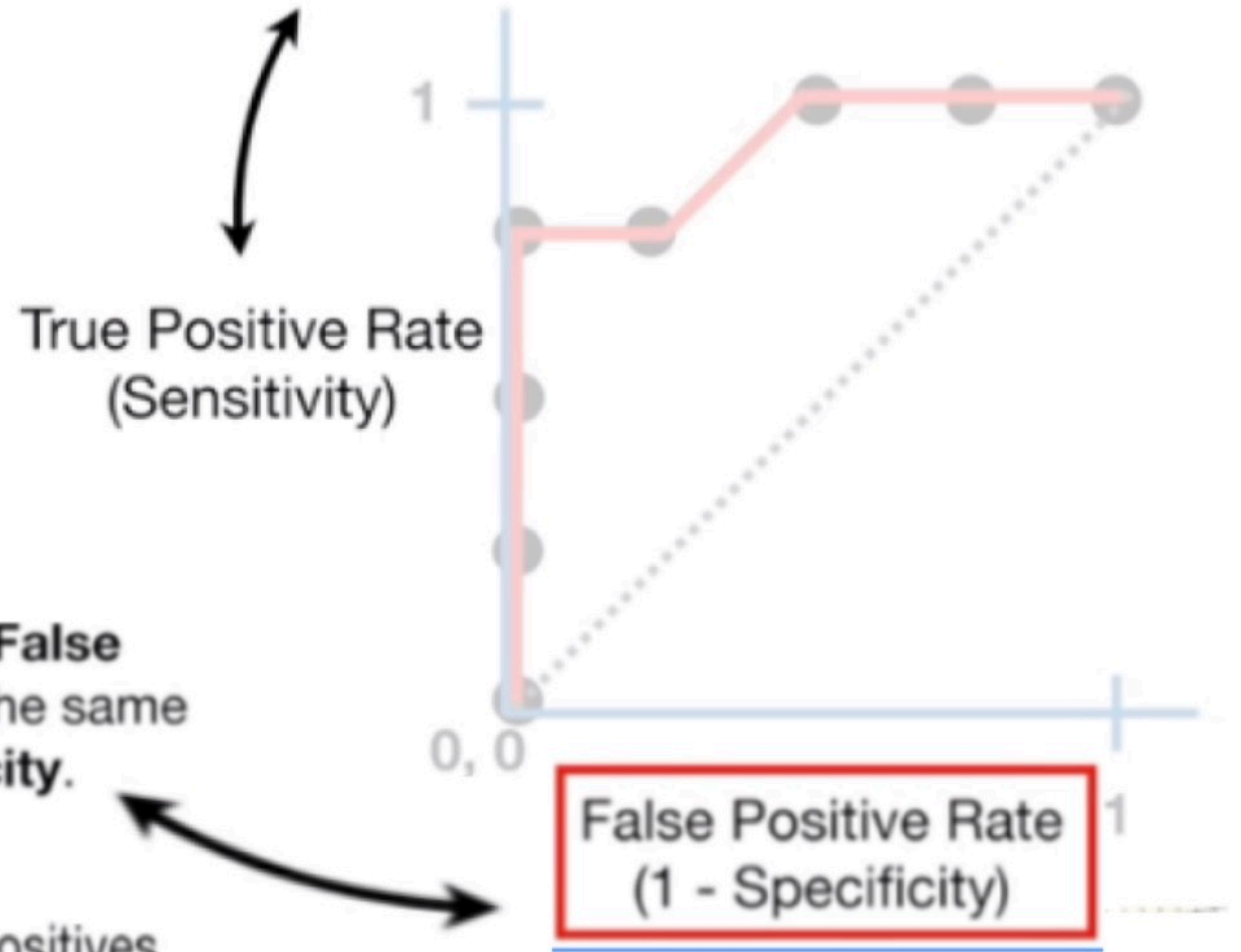
F1 Score 57 %

Other Metrics to Evaluate our Models -ROC/AUC-

$$\text{True Positive Rate} = \text{Sensitivity} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Negatives}}$$

Receiver
Operator
Characteristics

Area
Under
Curve

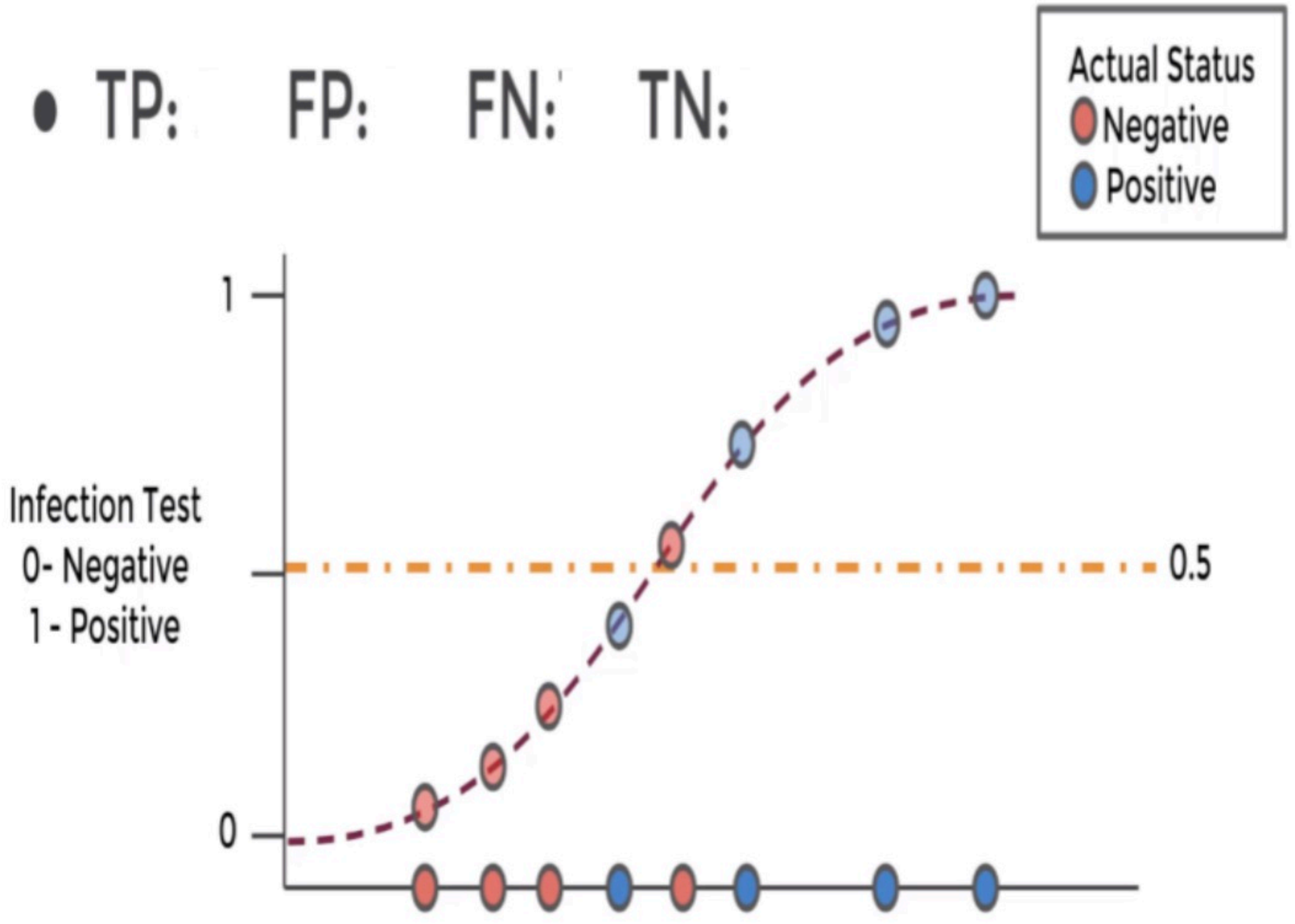


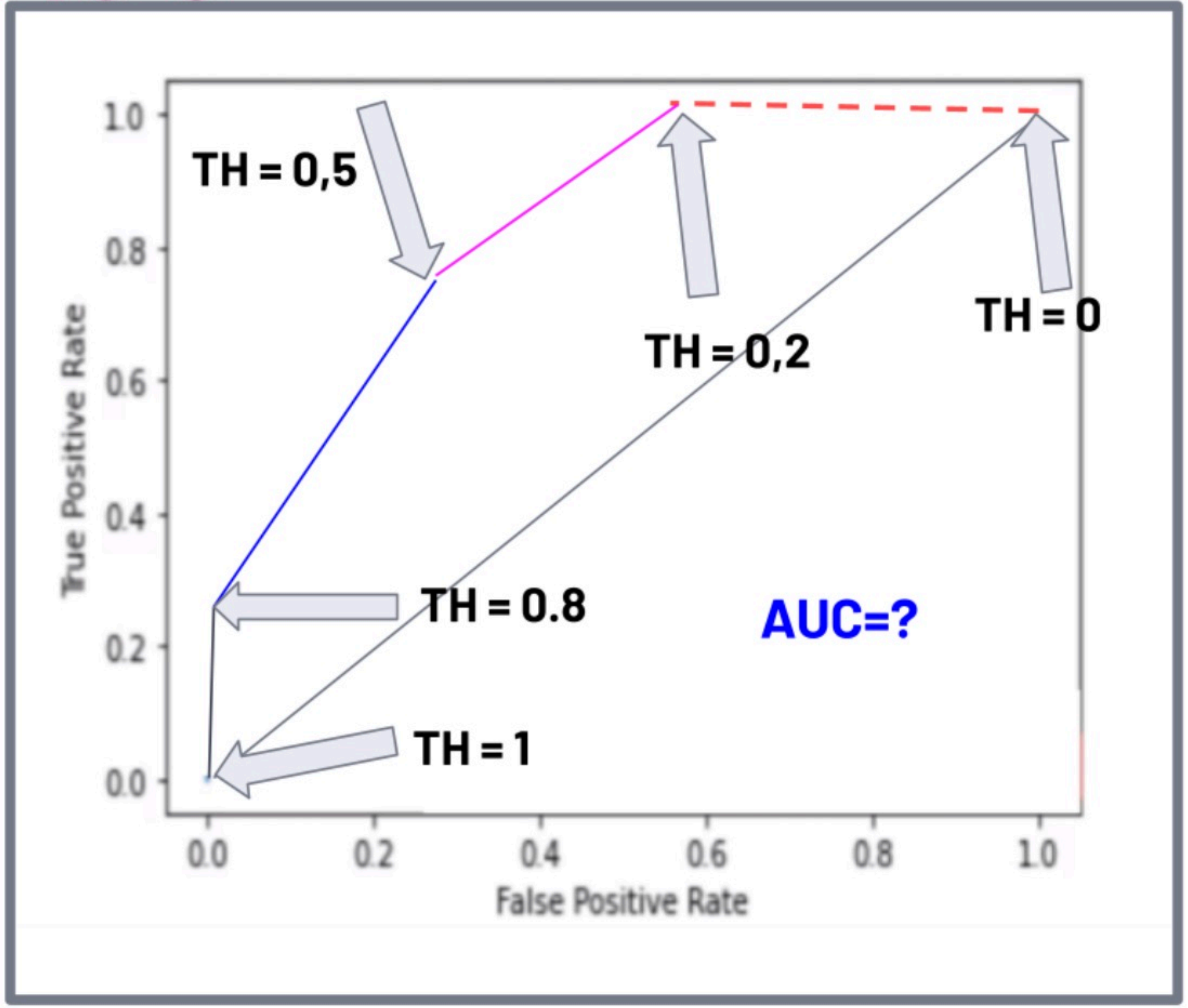
The x-axis shows the **False Positive Rate**, which is the same thing as **1 - Specificity**.

$$\text{False Positive Rate} = (1 - \text{Specificity}) = \frac{\text{False Positives}}{\text{False Positives} + \text{True Negatives}}$$

Picture credit :StatQuest

Other Metrics to Evaluate our Models -ROC/AUC-

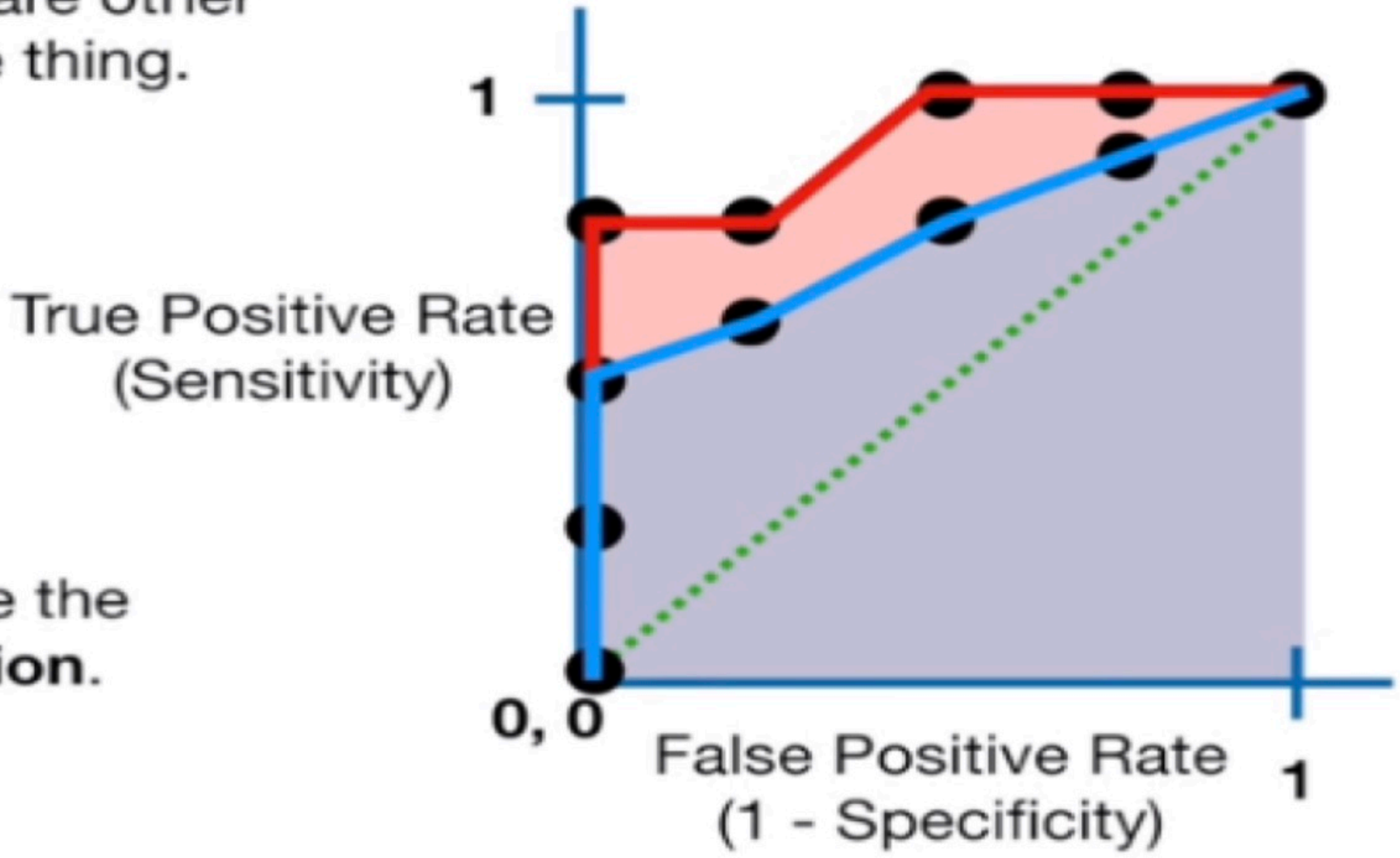




Other Metrics to Evaluate our Models -ROC/AUC-

Although **ROC** graphs are drawn using **True Positive Rates** and **False Positive Rates** to summarize confusion matrices, there are other metrics that attempt to do the same thing.

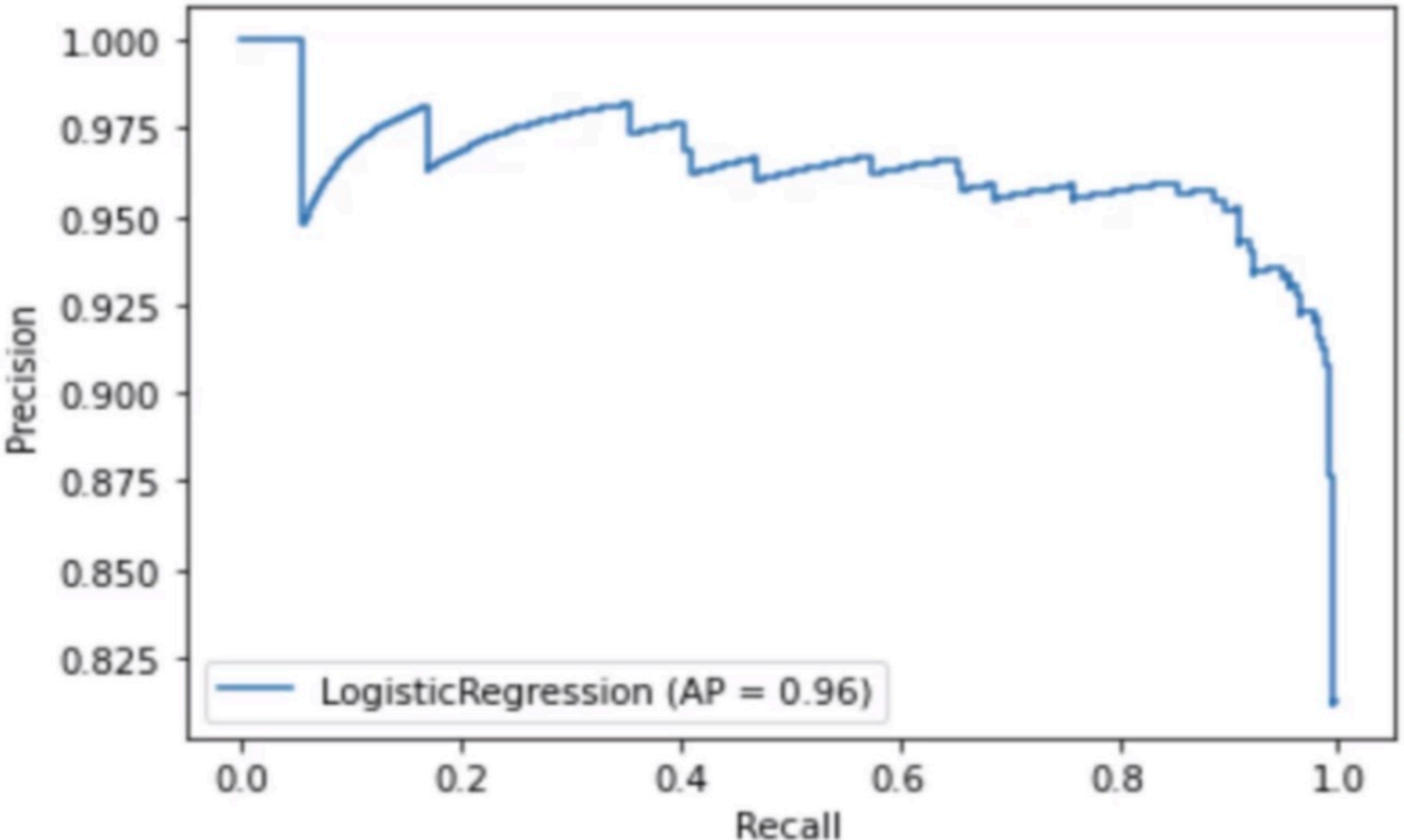
For example, people often replace the **False Positive Rate** with **Precision**.



Picture credit :StatQuest

Other Metrics to Evaluate our Models -ROC/AUC-

$$\text{Precision} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Positives}}$$ = **Precision** is the proportion of positive results that were correctly classified



Classification Error Metrics

	Predicted class POSITIVE (spam)	Predicted class NEGATIVE (normal)	
Actual class POSITIVE (spam)	TRUE POSITIVE (TP) 320	FALSE NEGATIVE (FN) 43	*Recall* $= \frac{TP}{TP+FN} = \frac{320}{320+43} = 0.882$
Actual class NEGATIVE (normal)	FALSE POSITIVE (FP) 20	TRUE NEGATIVE (TN) 538	
	Precision $= \frac{TP}{TP+FP} = \frac{320}{320+20} = 0.941$		

Accuracy =

$$\frac{\sum TP + \sum TN}{\sum \text{total population}}$$

F1 Score =

$$\frac{2}{\frac{1}{\text{Recall}} + \frac{1}{\text{Precision}}}$$

Confusion Matrix

Logistic Regression with Python

Be ready for

Logistic Regression with Python 1-2

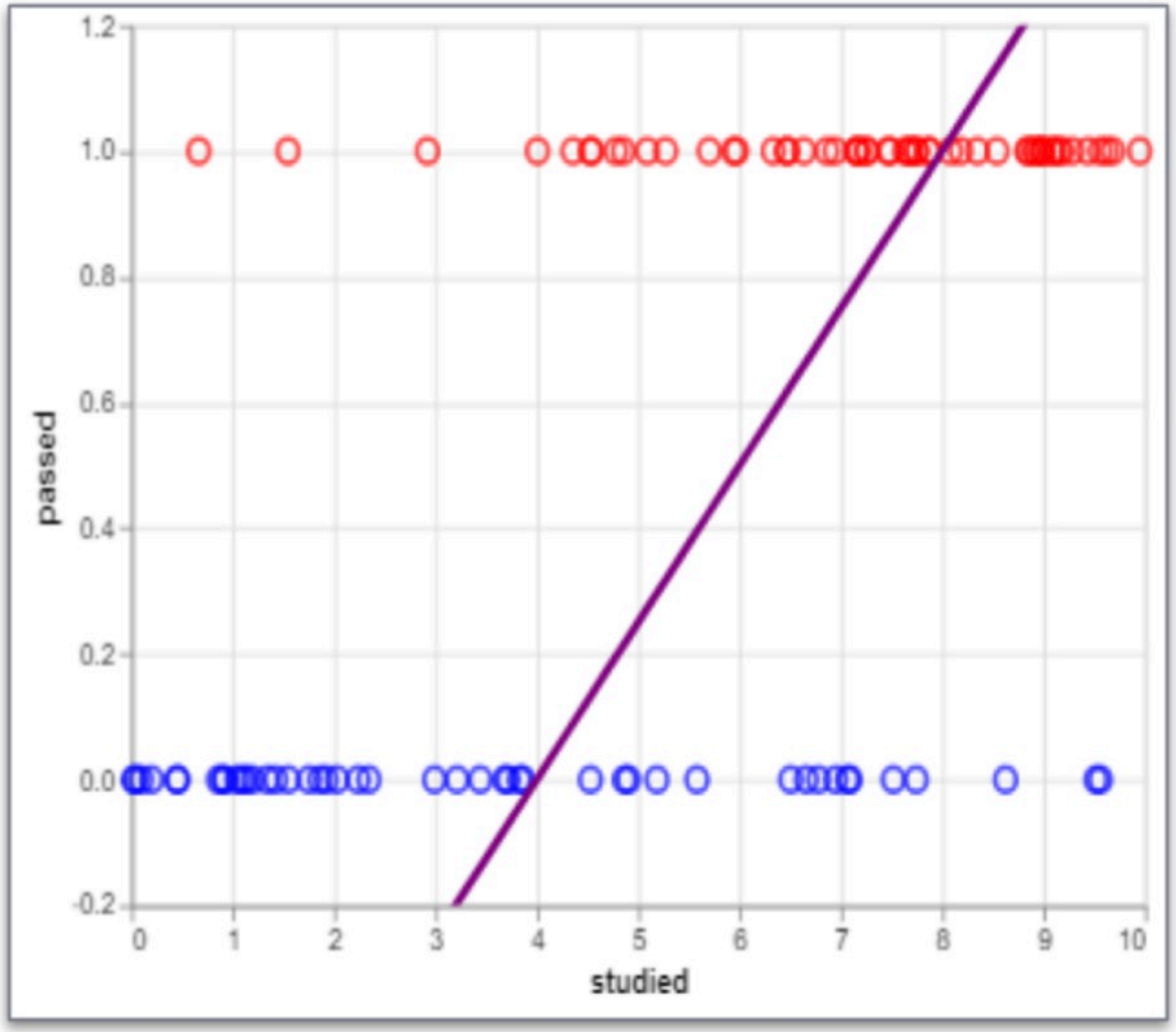
Session

Logistic Regression Theory

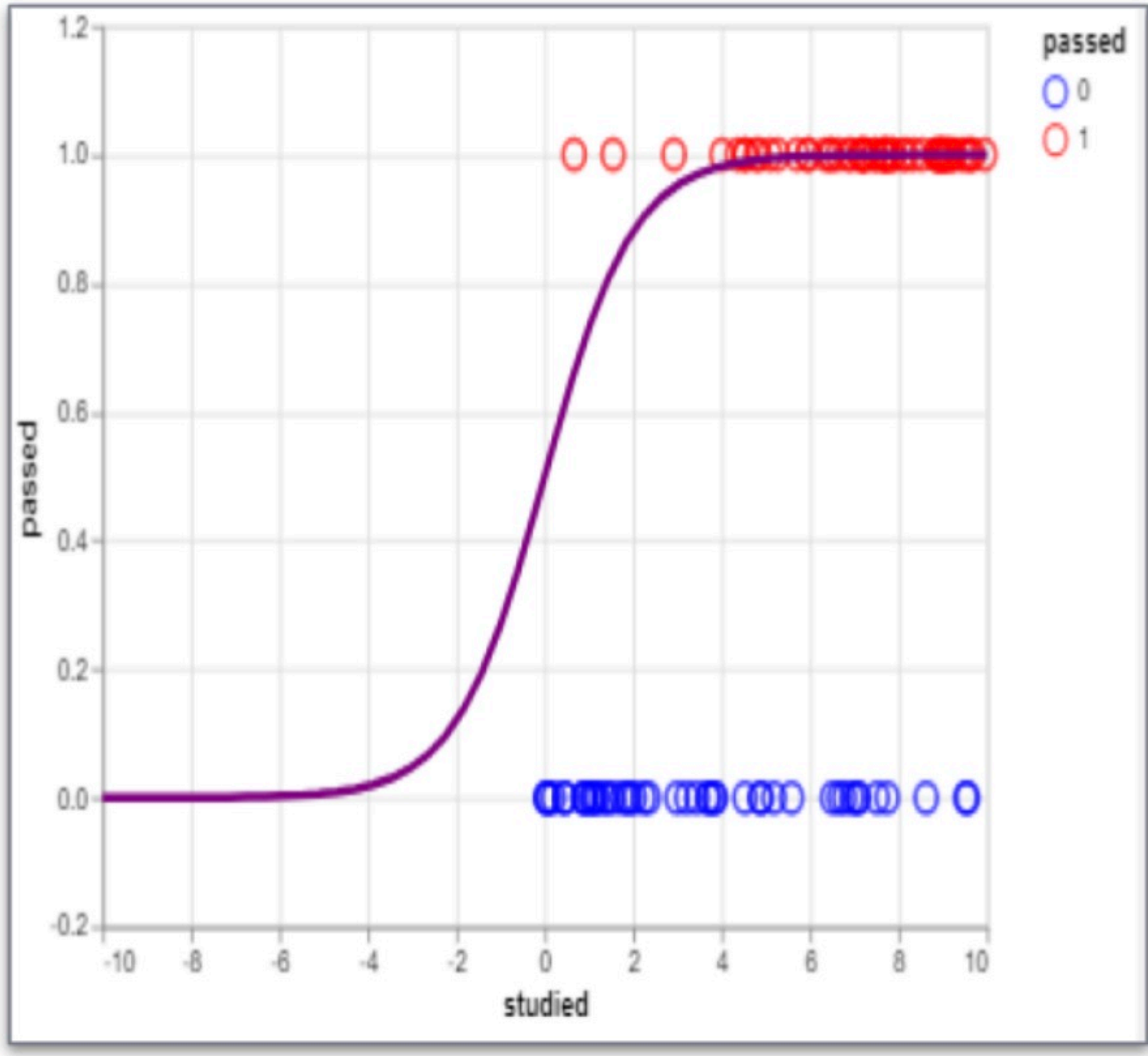
Another Sample:

	studied	slept	passed
0	4.855064	9.639962	1
1	8.625440	0.058927	0
2	3.828192	0.723199	0
3	7.150955	3.899420	1
4	6.477900	8.198181	1
...	...	...	...
95	0.022280	2.658428	0
96	7.630637	7.405351	1
97	3.684997	5.049965	0
98	7.484260	6.059396	1
99	2.030708	3.937267	0

DATA



Linear Regression



Logistic Regression