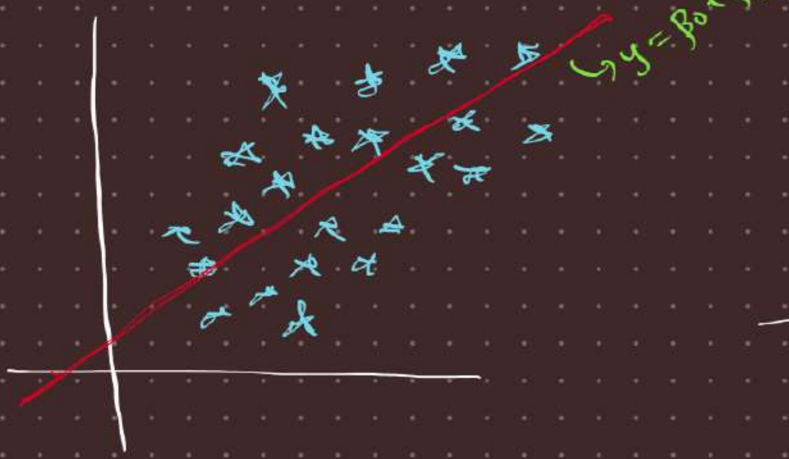
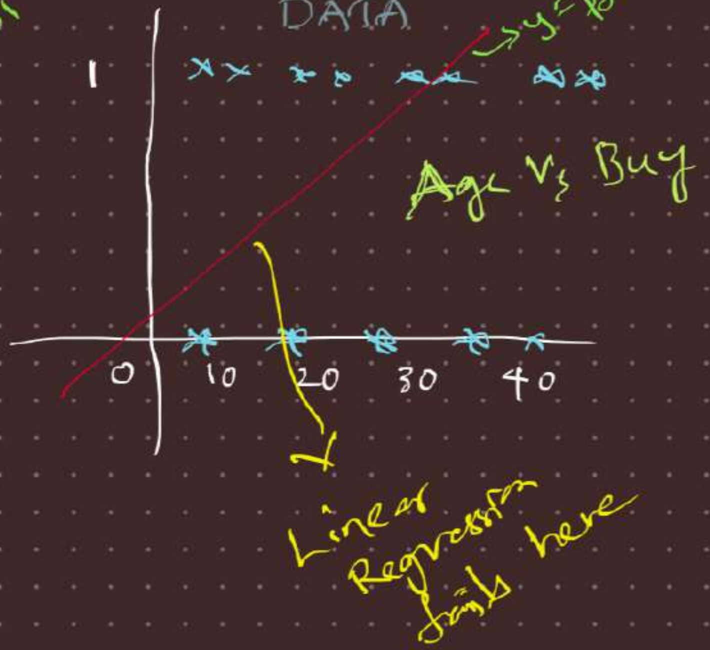
LOGISTIC REGRESSION

→ concept matters ; tool/package doesn't matter

LINEAR DATA



NON-LINEAR DATA



Sales - Buying Vs Not Buying

Marketing - Response Vs No Response

Credit card & Loans - Default vs Non Default

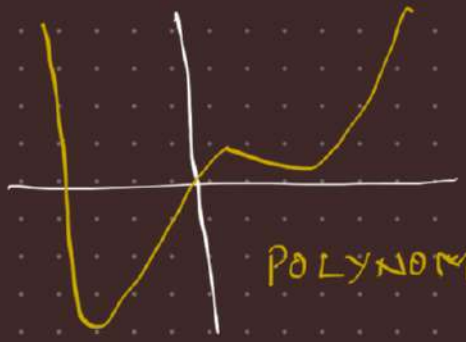
Website - Click vs No click

→ CLASSIFICATION PROBLEM

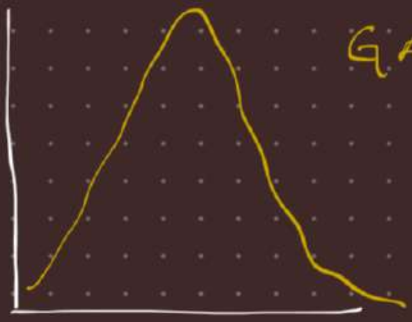
LIN. REG - a good fit if you are predicting a continuous value (target variable)

If you are predicting class 0 or class 1 ? → Non. linear function

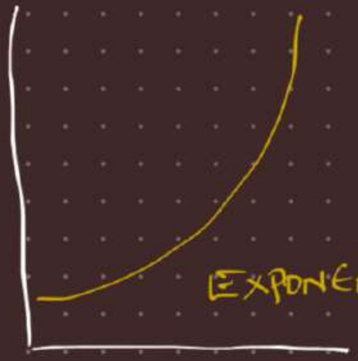
Some Non lincar functions



POLYNOMIAL

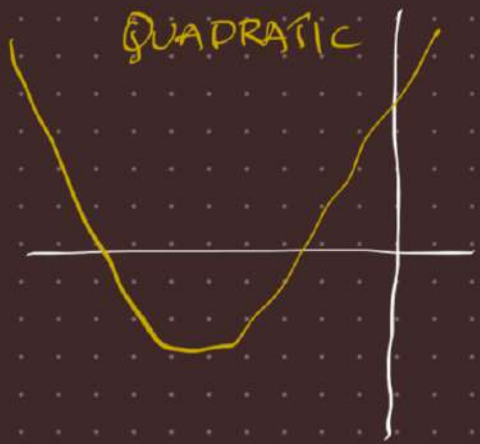


GAUSSIAN

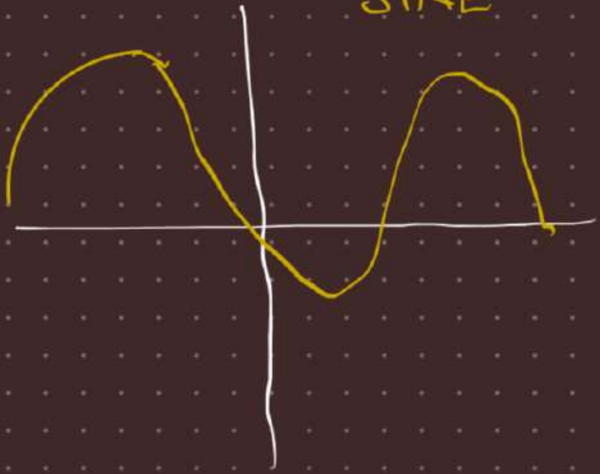


EXPONENTIAL

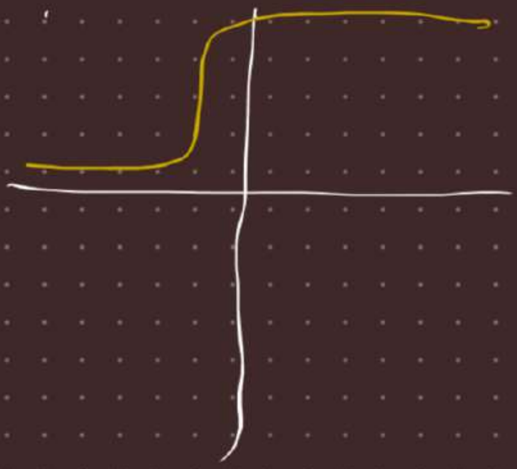
QUADRATIC



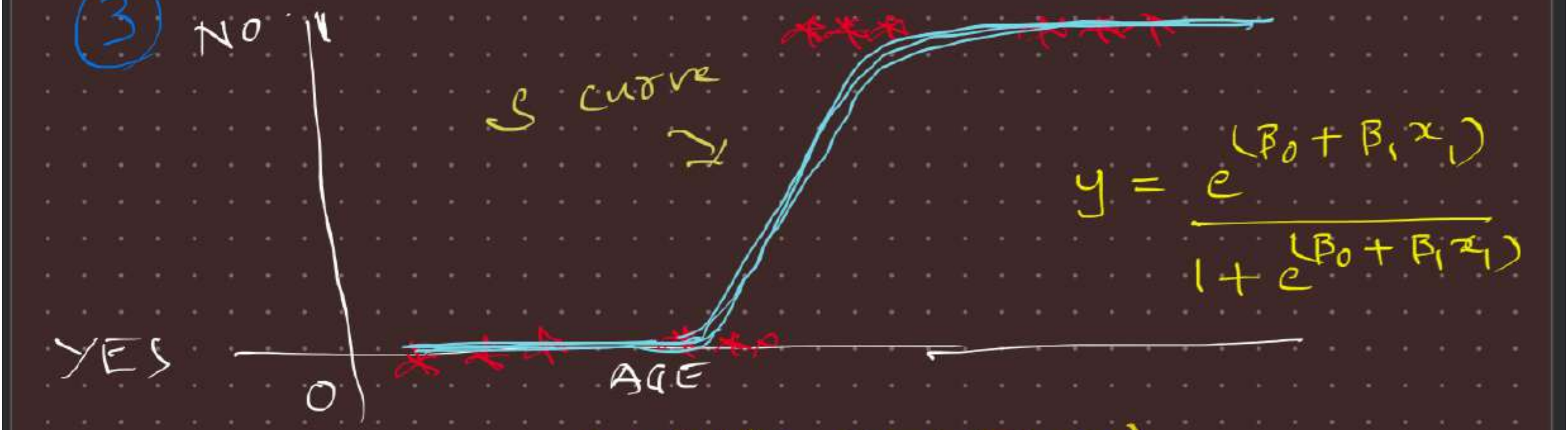
SINE



LOGISTIC



③



$$y = \frac{e^{(\beta_0 + \beta_1 x_1)}}{1 + e^{(\beta_0 + \beta_1 x_1)}}$$

LOGISTIC REGRESSION

Note :

Looking at problem statement
at target variable

should be able to decide

Regression problem

Classification problem

Exercise : Problem statements

① Predicting Loss % → Regression

② Predicting Buying Vs Not buying → classification

③ Predicting no. of customers → regression

④ Predicting response vs no response → classification

⑤ Predicting Revenue → regression

⑥ Predicting the product prize → regression

⑦ Predicting Attrition vs. Retention → classification

⑧ Predicting click Vs No click → classification

⑨ Predicting Fraud Vs Non Fraud → classification

⑩ Predicting the amount of fraud → Regression

4 MULTIPLE LOGISTIC REGRESSION

real world → target variable depends on multiple features

$$y = \frac{e^{\beta_0 + B_1x_1 + \beta_2x_2 + \beta_3x_3 \ldots - \beta_k xc_k}}{1 + e\left(\beta_0 + \beta_1x_1 + \beta_2x_2 + -\ldots\beta_kx_k\right)}$$

Note :

customer churned out } customer leaving

Model building must be always followed by Model validation.

Earlier, for Linear Regression, model validation → R^2

For Logistic Regression (classification problem)

Model validation : ⟶ next page

5

LOGISTIC REGRESSION : MODEL VALIDATION

→ model.predict()

x_1,	x_2,	x_3	------	x_K	y_{actual}	y_{pred}
—	—	—		—	1	1
—	—	—		—	0	1
—	—	—		—	1	1
—	—	—		—	0	0
—	—	—		—	1	0
—	—	—		—	0	0

		Ypred	
		0	1
Yactual	0	a	b
	1	c	d

→ Confusion matrix

$$\text{Accuracy} = \frac{a+d}{(a+b+c+d)}$$

What is the guerentee that this "Accuracy" remains the same for new data.

To solve this problem, we need to do "Cross Validation"

6 LOGISTIC REGRESSION : CROSS VALIDATION

Lets say, dataset = 10,000 records

80,000 (training data)

20,000 (testing data)

Train data Accuracy = 92%
Test data Accuracy = 60%
} this model is suffering from **OVERFITTING**

Train Accuracy = 60% } this model is suffering from **UNDERFITTING**

Overfitting : Train accuracy is very high but test accuracy is significantly low

Thats the reason, validation alone is not sufficient, you need to do cross validation.

How do you do that?

Divide the data into train data & test data

In sklearn, → **model_selection.train_test_split**