

Machine Learning Algorithms

Linear and Logistic Regression

Raghu Prasad K S

Introduction

- Raghu Prasad – BE, MS
- Total of 29 years of experience
- 7 years as a lecturer in an Engineering College
- 22 Years into IT
- Worked with companies like CISCO,CSC,ICICI,First Apex – NTT Data
- Currently into Corporate training and consultancy
- Worked with corporates and public sector
- **Technologies** – Java,Python,Web technologies,Java Script technologies (MEAN stack),IOT,Test Automation,Machine Learning,Artificial Intelligence,ERP,.NET

Topics

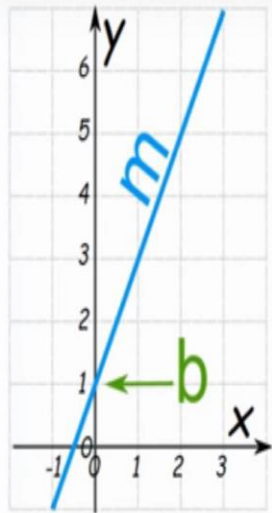
- Supervised machine learning
 - Linear Regression
 - Logistic Regression

Supervised Machine Learning

- **Supervised Learning**
- **How it works:** This algorithm consist of a target / outcome variable (or dependent variable) which is to be predicted from a given set of predictors (independent variables). Using these set of variables, we generate a function that map inputs to desired outputs. The training process continues until the model achieves a desired level of accuracy on the training data.
- **Examples of Supervised Learning: Regression, Decision Tree, Random Forest, KNN, Logistic Regression etc.**

Linear Regression – Single Variable

- Prediction of home price



$$\text{price} = m * \text{area} + b$$

$$y = mX + b$$

Slope (or Gradient) Y Intercept

Reference: <https://www.mathsisfun.com/algebra/linear-equations.html>

$$\text{price} = m * \text{area} + b$$

Dependent variable

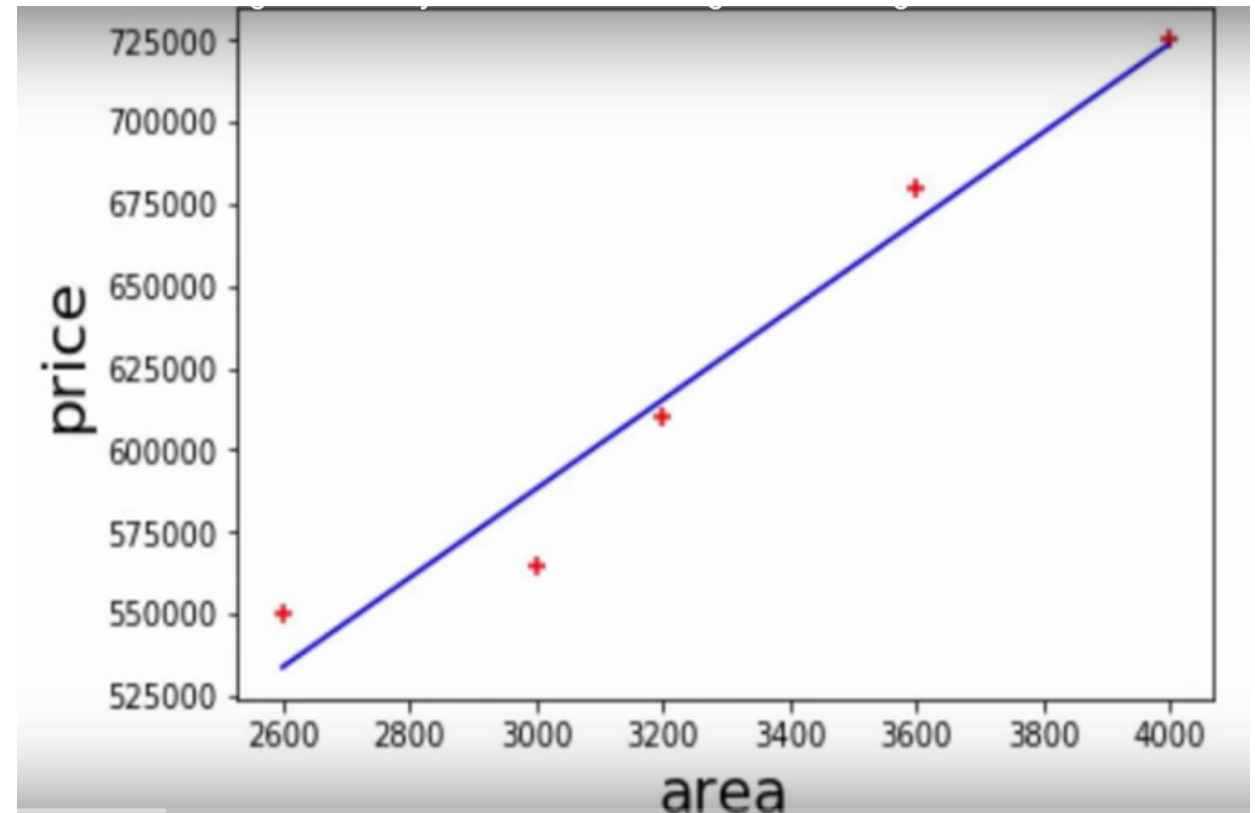
Independent variable

Linear Regression – Single Variable

- Prediction of home price

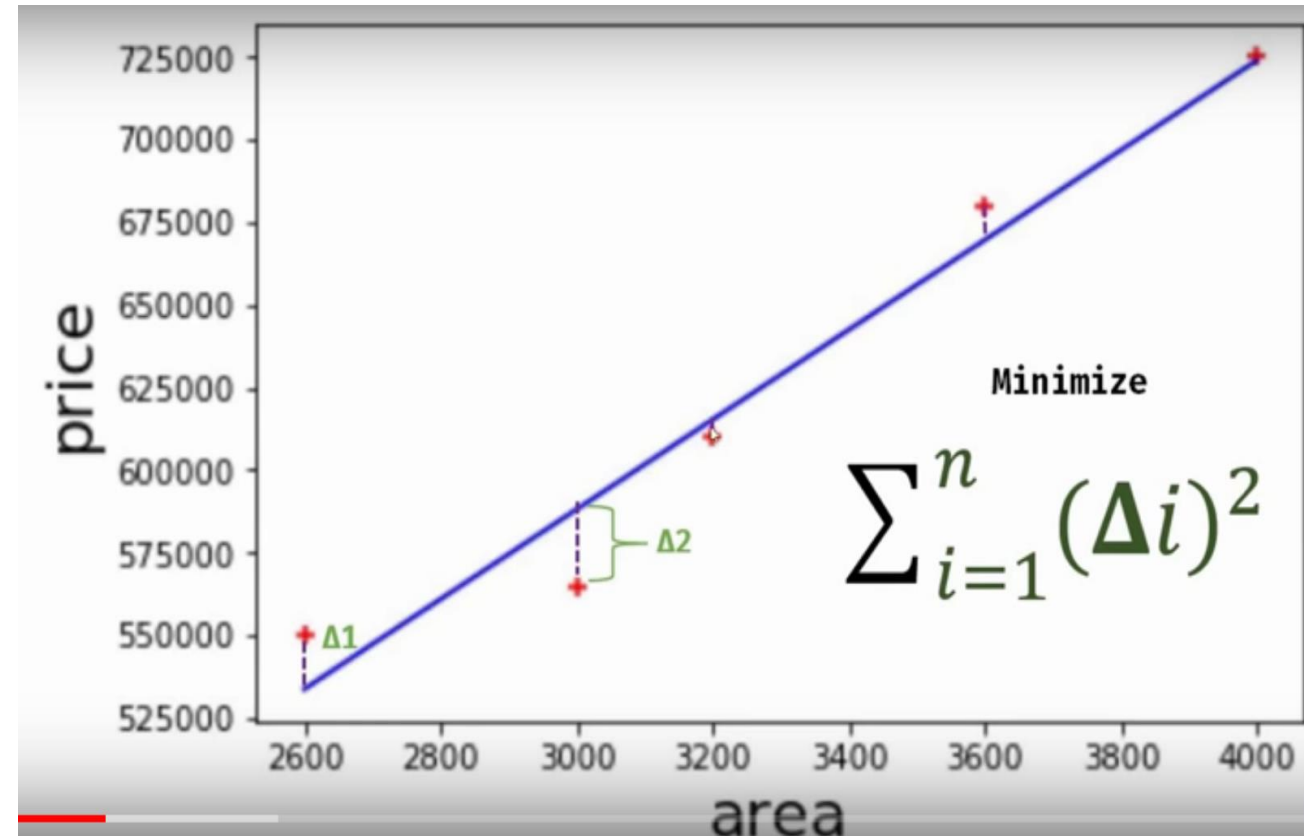
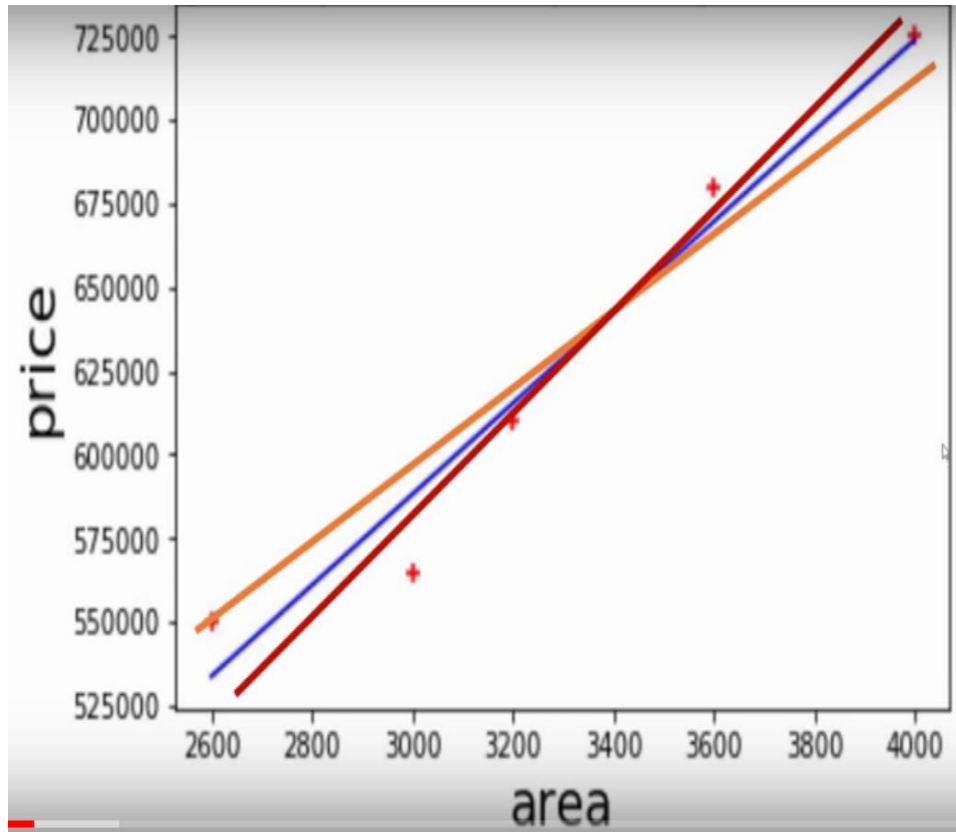
Home prices in Monroe Twp, NJ (USA)

area	price
2600	550000
3000	565000
3200	610000
3600	680000
4000	725000



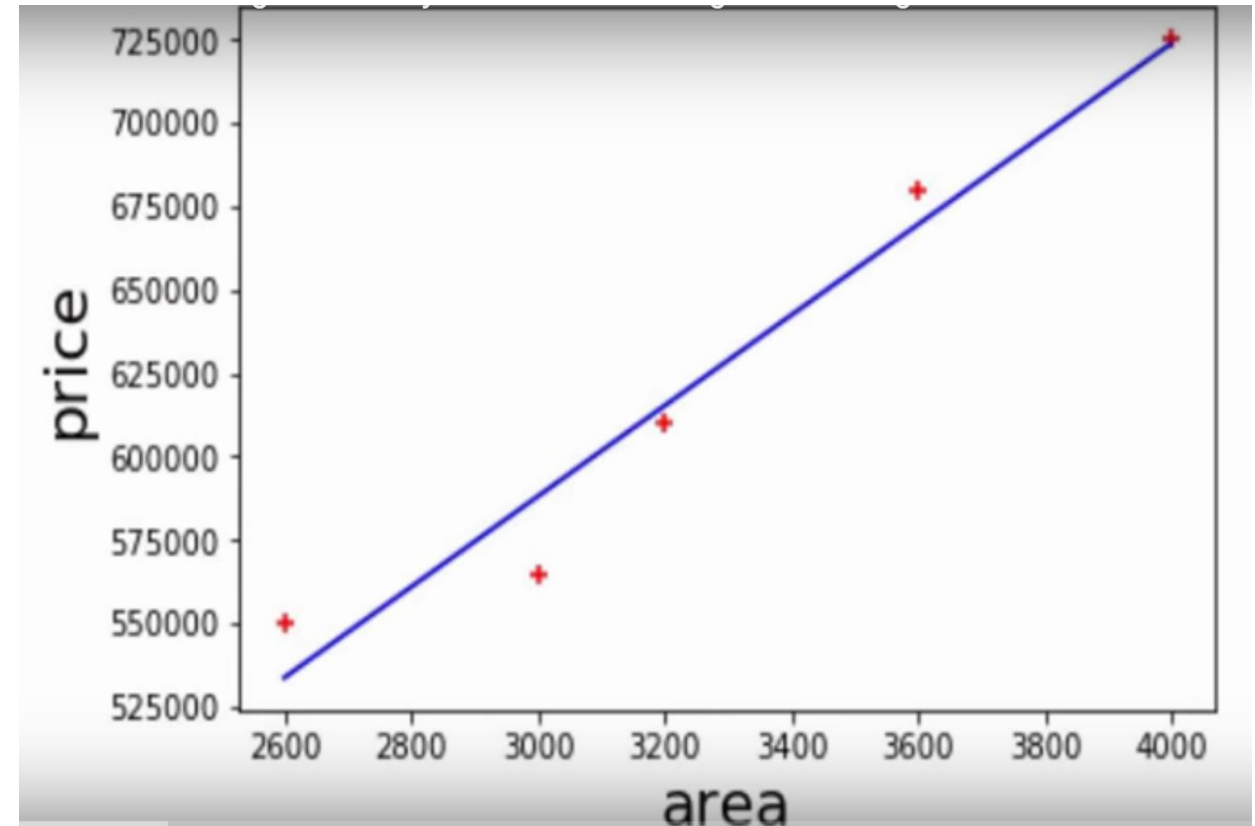
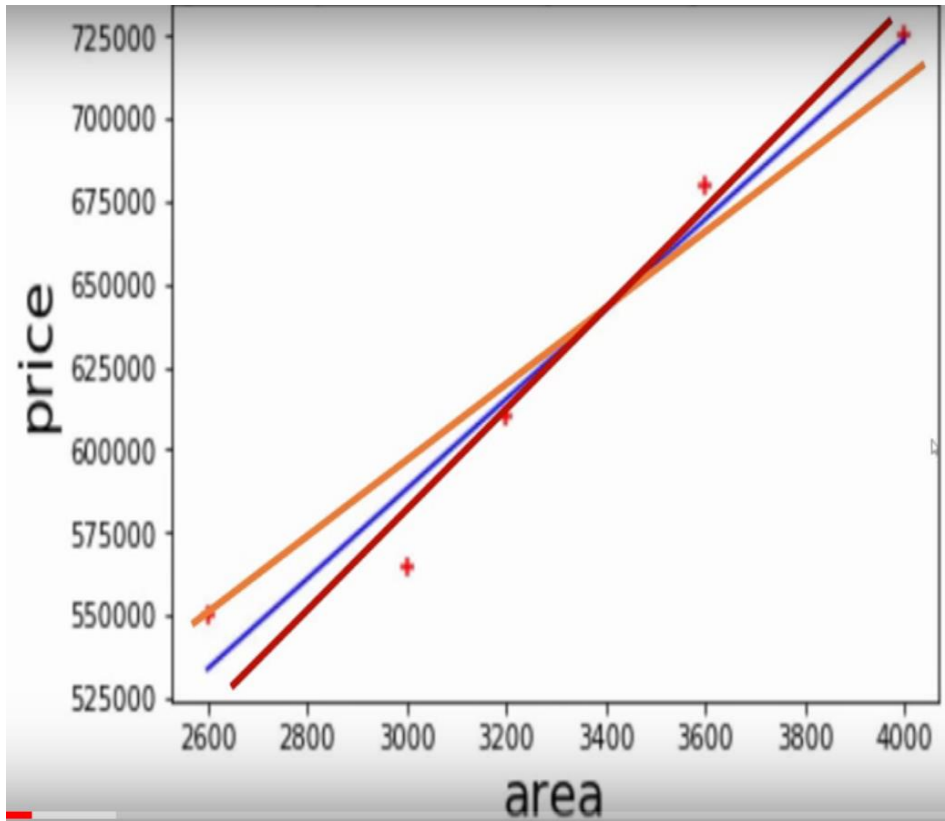
Linear Regression – Single Variable

- Bestfit line



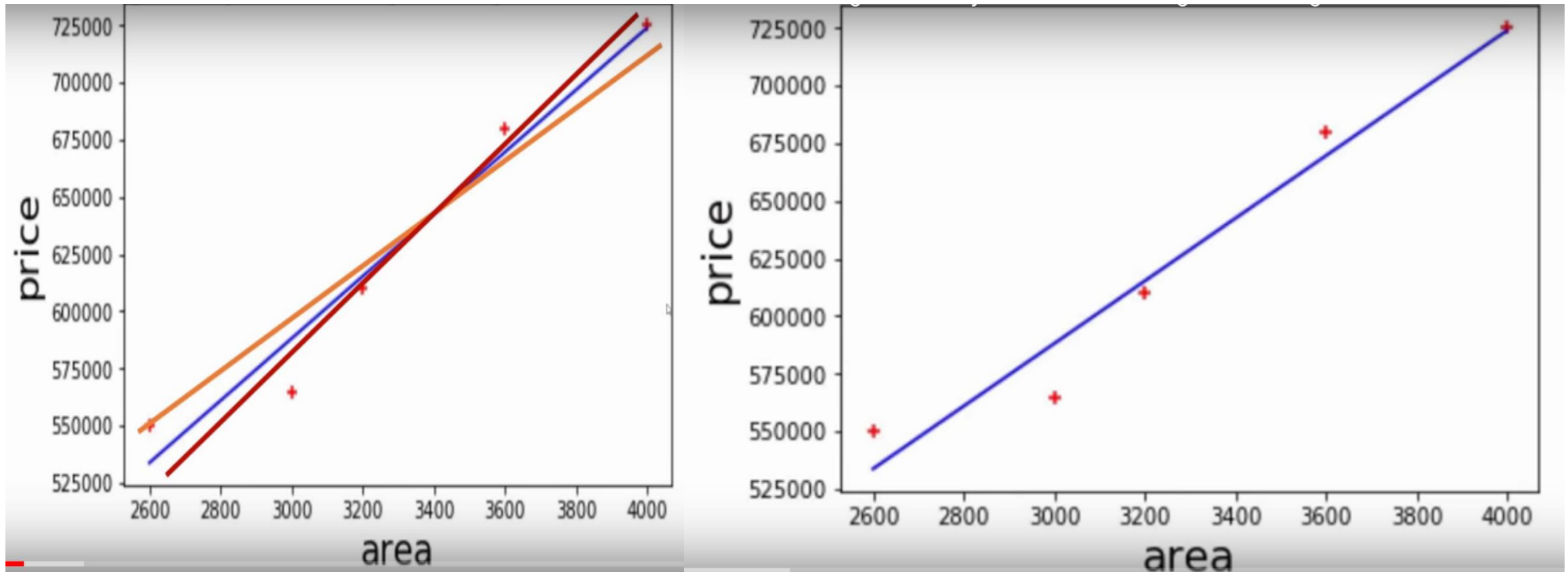
Linear Regression – Single Variable

- Bestfit line



Linear Regression – Single Variable

- Bestfit line



Linear Regression – Multiple Variable

- Home price

Home prices in Monroe Township, NJ (USA)

area	bedrooms	age	price
2600	3	20	550000
3000	4	15	565000
3200		18	610000
3600	3	30	595000
4000	5	8	760000

Dependent variable

Independent variables (**features**)

$$\text{price} = m_1 * \text{area} + m_2 * \text{bedrooms} + m_3 * \text{age} + b$$

Coefficients

Intercept

The diagram shows the equation $\text{price} = m_1 * \text{area} + m_2 * \text{bedrooms} + m_3 * \text{age} + b$. A red arrow points from the label 'Dependent variable' to 'price'. Another red arrow points from the label 'Independent variables (features)' to the group of terms $m_1 * \text{area} + m_2 * \text{bedrooms} + m_3 * \text{age}$. Three purple arrows point from the label 'Coefficients' to m_1 , m_2 , and m_3 . A purple arrow points from the label 'Intercept' to b .

Dummy Variables – One Hot Encoding

- Using Pandas – get_dummies
- Using sklearn – OneHotEncoding

How should we handle text data in numeric model?

town	area	price
monroe township	2600	550000
monroe township	3000	565000
monroe township	3200	610000
monroe township	3600	680000
monroe township	4000	725000
west windsor	2600	585000
west windsor	2800	615000
west windsor	3300	650000
west windsor	3600	710000
robbinsville	2600	575000
robbinsville	2900	600000
robbinsville	3100	620000

town	area	price
1	2600	550000
1	3000	565000
1	3200	610000
1	3600	680000
1	4000	725000
2	2600	585000
2	2800	615000
2	3300	650000
2	3600	710000
3	2600	575000

Dummy Variables – One Hot Encoding

- Using Pandas – get_dummies
- Using sklearn – OneHotEncoding

How should we handle text data in numeric model?

town	area	price
monroe township	2600	550000
monroe township	3000	565000
monroe township	3200	610000
monroe township	3600	680000
monroe township	4000	725000
west windsor	2600	585000
west windsor	2800	615000
west windsor	3300	650000
west windsor	3600	710000
robbinsville	2600	575000
robbinsville	2900	600000
robbinsville	3100	620000

town	area	price
1	2600	550000
1	3000	565000
1	3200	610000
1	3600	680000
1	4000	725000
2	2600	585000
2	2800	615000
2	3300	650000
2	3600	710000
3	2600	575000

Dummy Variables – One Hot Encoding

- Using Pandas – get_dummies
- Using sklearn – OneHotEncoding

Categorical Variables

Nominal

monroe township
robbinsville
west windsor

male
female

green
red
blue

Ordinal

satisfied
neutral
dissatisfied

graduate
masters
phd

high
medium
low

town	area	price	monroe township	west windsor	robbinsville
monroe township	2600	550000	1	0	0
monroe township	3000	565000	1	0	0
monroe township	3200	610000	1	0	0
monroe township	3600	680000	1	0	0
monroe township	4000	725000	1	0	0
west windsor	2600	585000	0	1	0
west windsor	2800	615000	0	1	0
west windsor	3300	650000	0	1	0
west windsor	3600	710000	0	1	0
robbinsville	2600	575000	0	0	1

Logistic Regression – Binary Classification

Linear Regression

1. Home prices
2. Weather
3. Stock price

Predicted value is
continuous

Classification

1. Email is spam or not
2. Will customer buy life insurance?
3. Which party a person is going to vote for?
 1. Democratic
 2. Republican
 3. Independent

Predicted value is
categorical

Classification Types

Will customer buy life insurance?

1. Yes
2. No

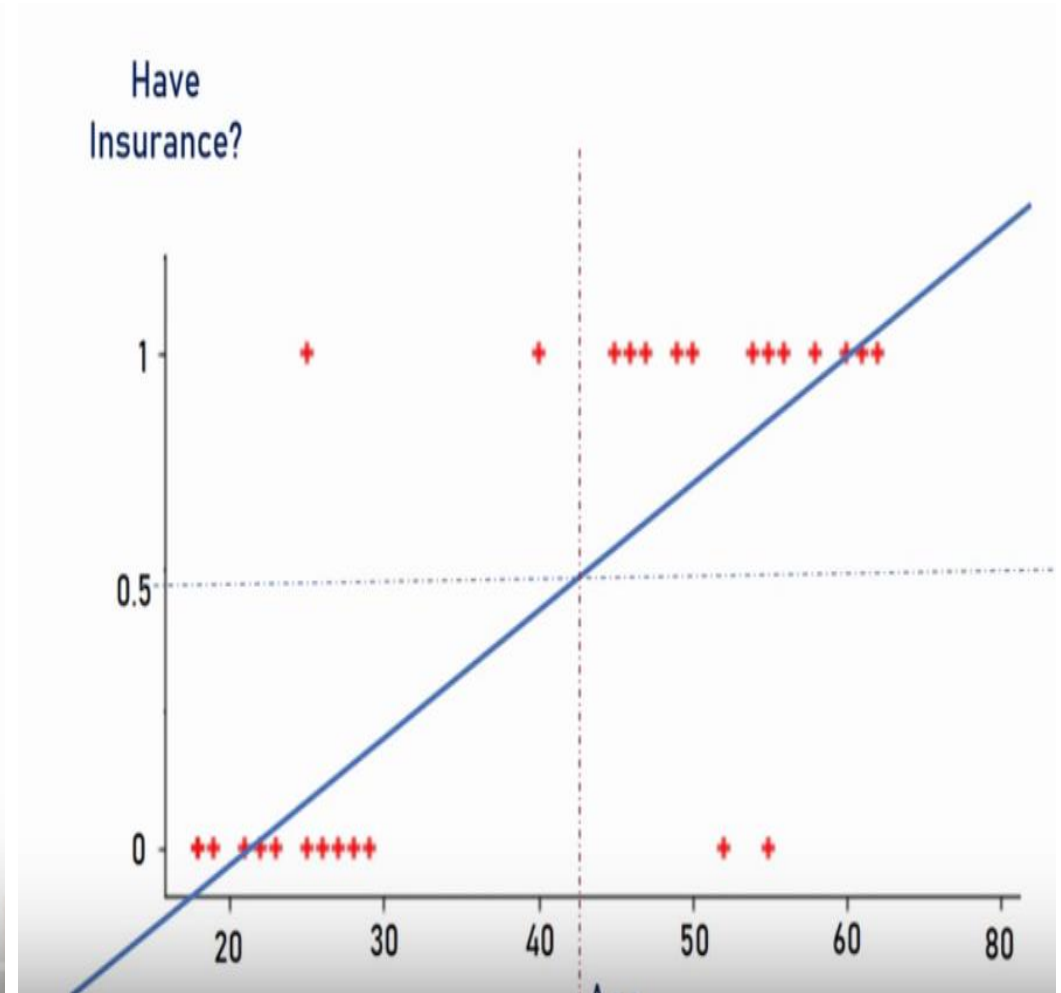
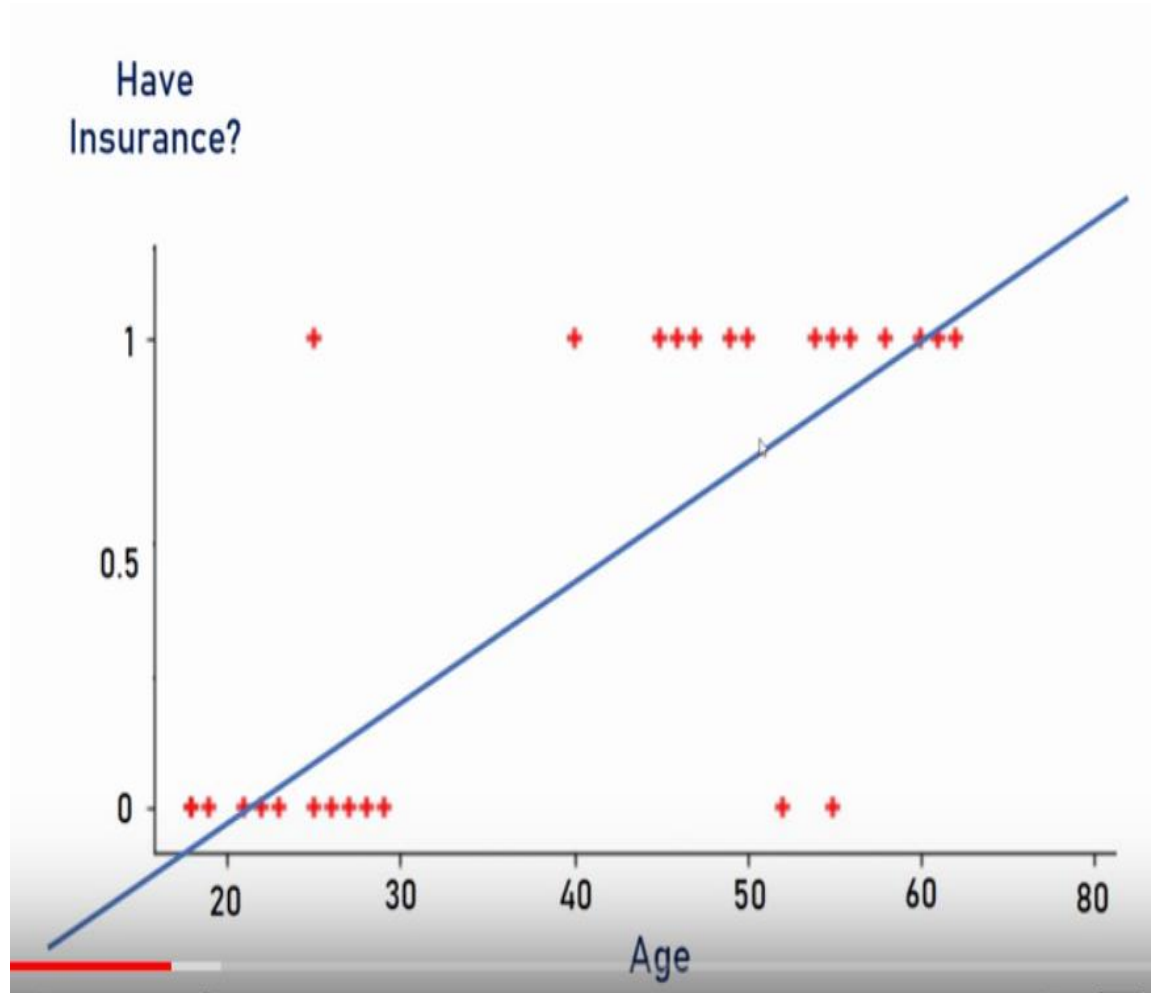
Binary Classification

Which party a person is going to vote for?

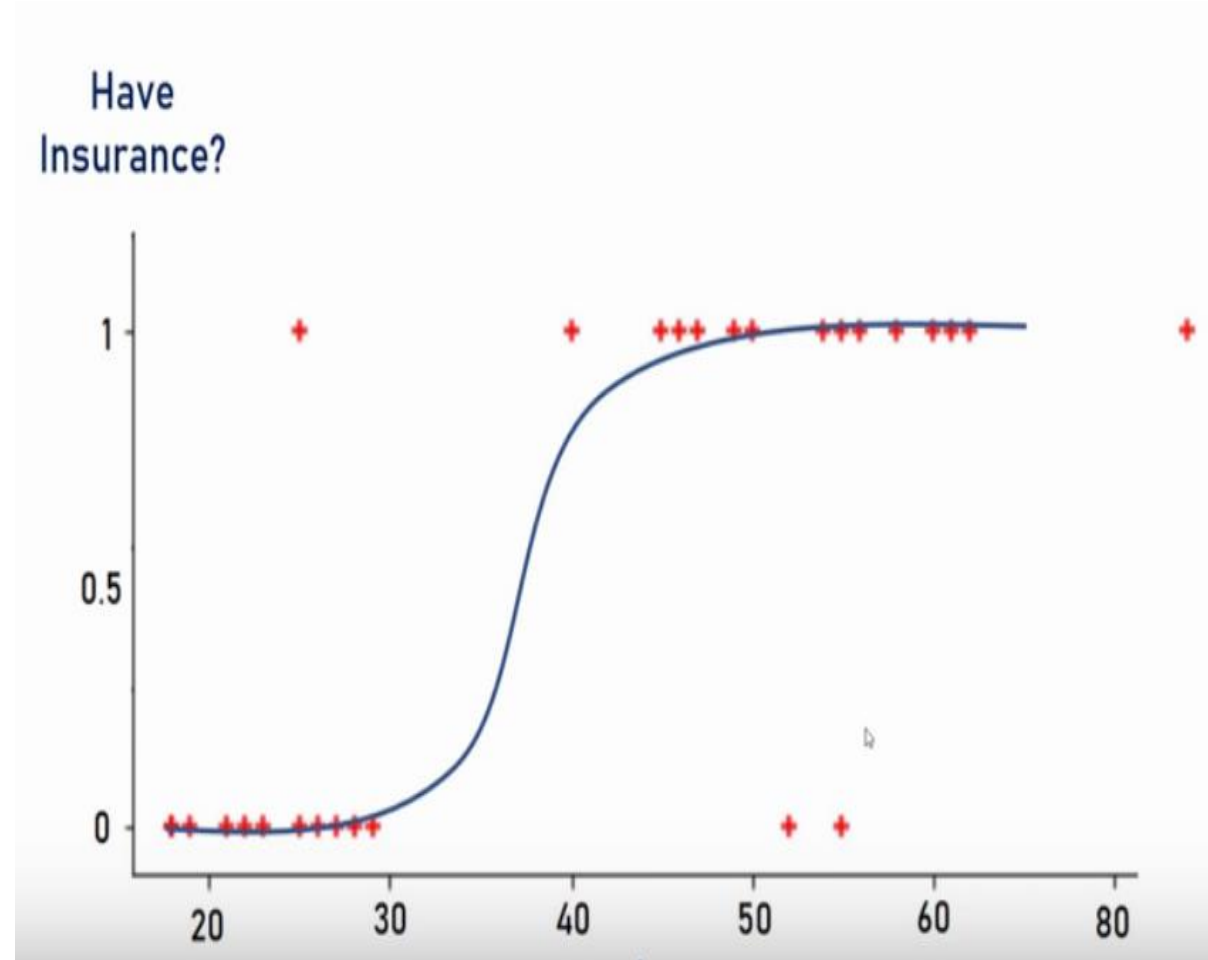
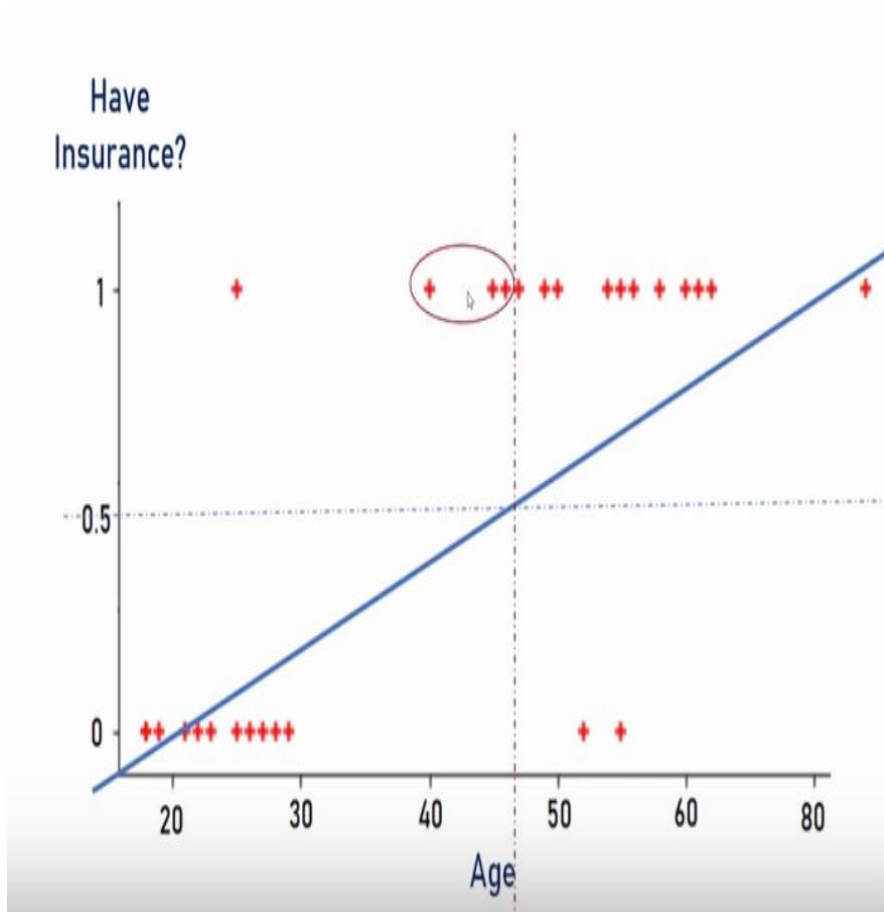
1. Democratic
2. Republican
3. Independent

Multiclass Classification

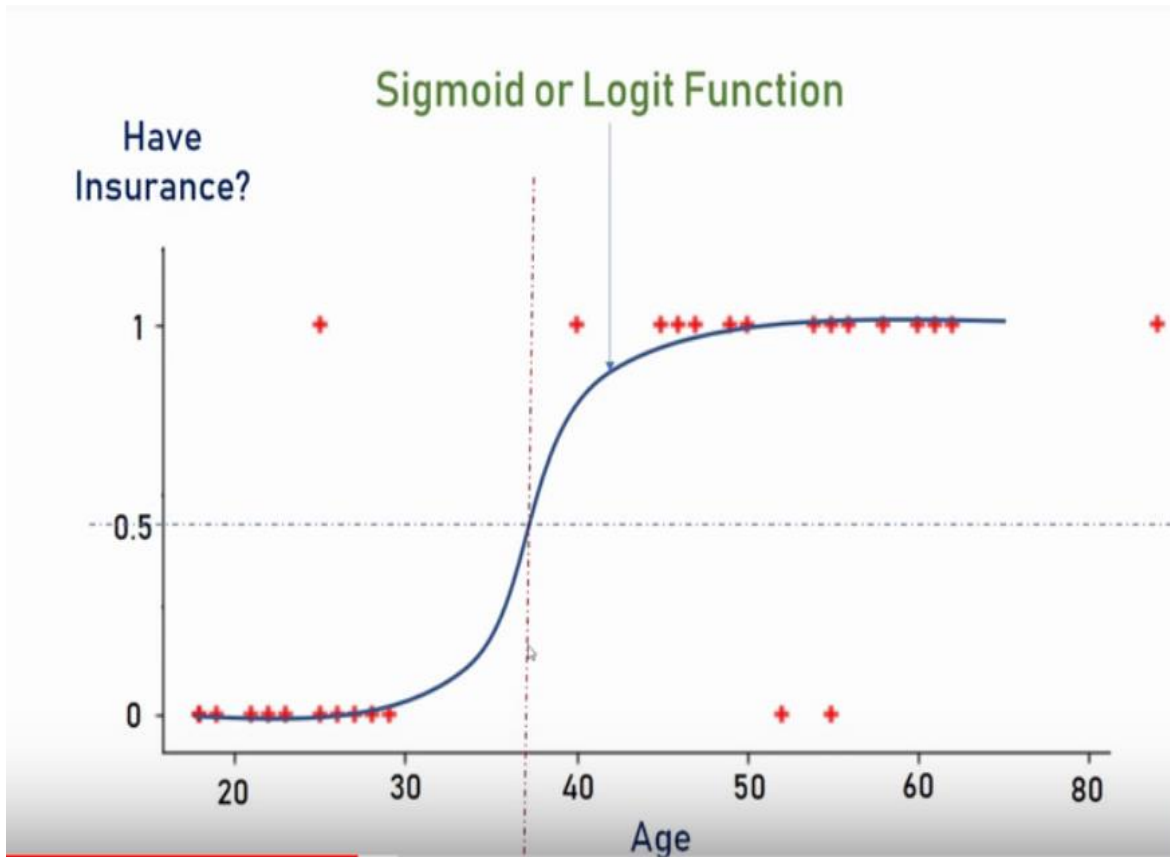
Logistic Regression – Binary Classification



Logistic Regression – Binary Classification



Logistic Regression – Binary Classification



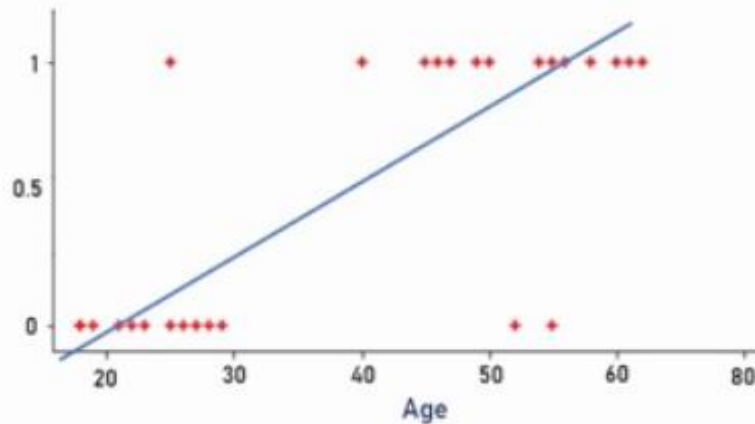
$$\text{sigmoid}(z) = \frac{1}{1 + e^{-z}}$$

e = Euler's number ~ 2.71828

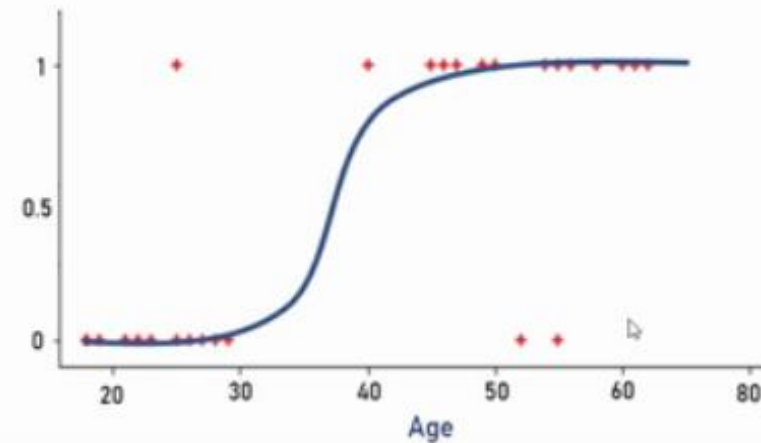
Sigmoid function converts input into range 0 to 1

Logistic Regression – Binary Classification

$$y = m * x + b$$

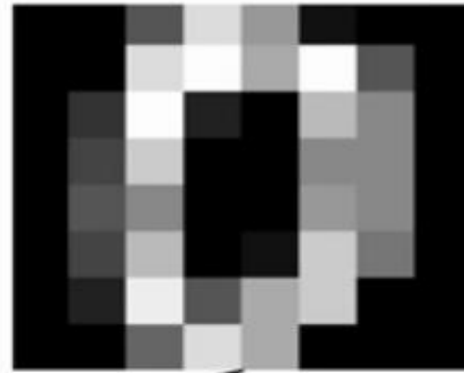


$$y = \frac{1}{1 + e^{-(m*x+b)}}$$

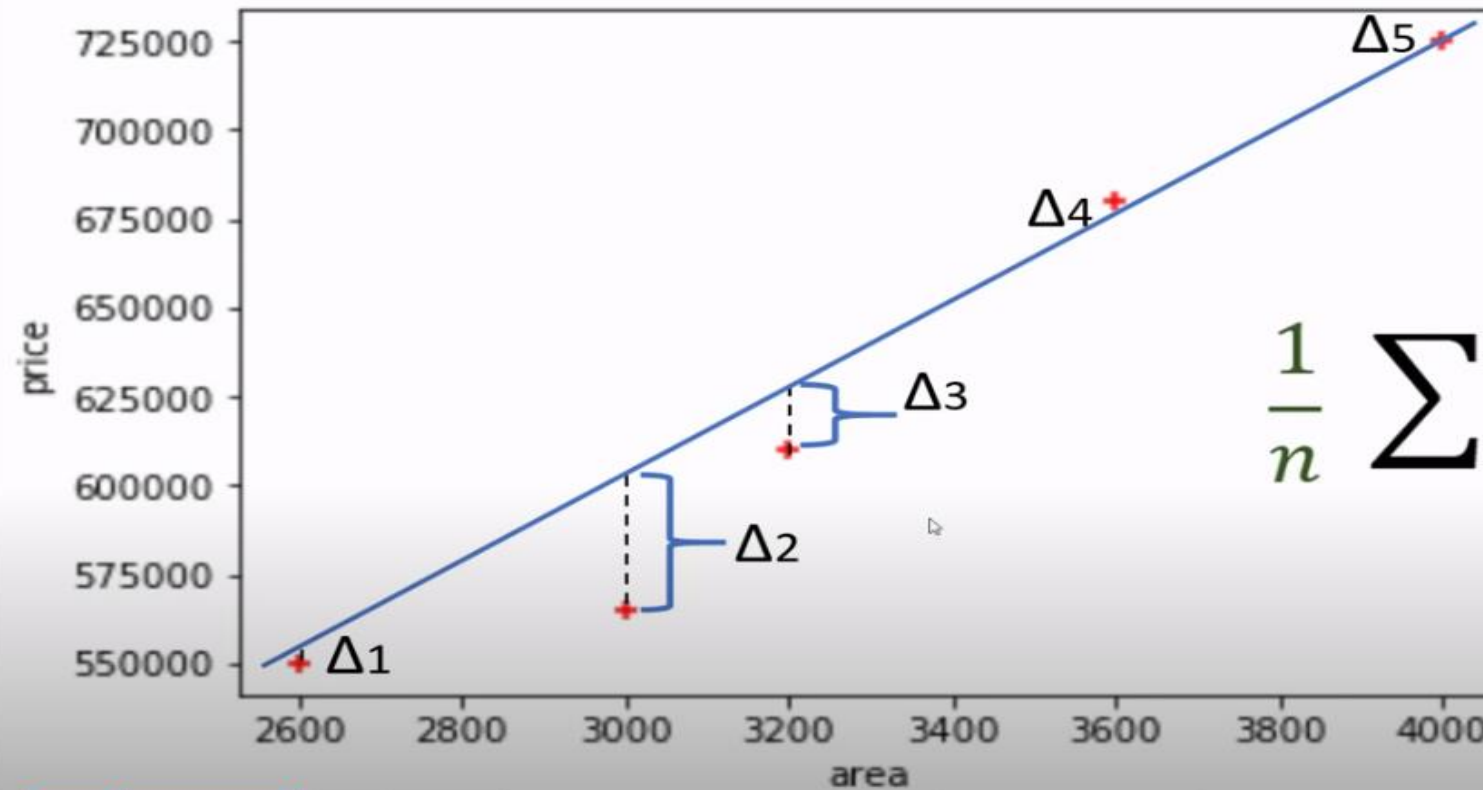


Logistic Regression – Multiclass Classification

Identify hand written digits recognition



Gradient Descent



$$\frac{1}{n} \sum_{i=1}^n (\Delta i)^2$$

Gradient Descent

Mean Squared Error

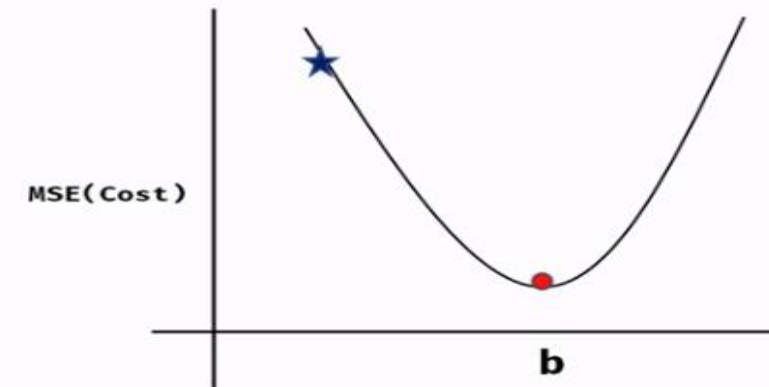
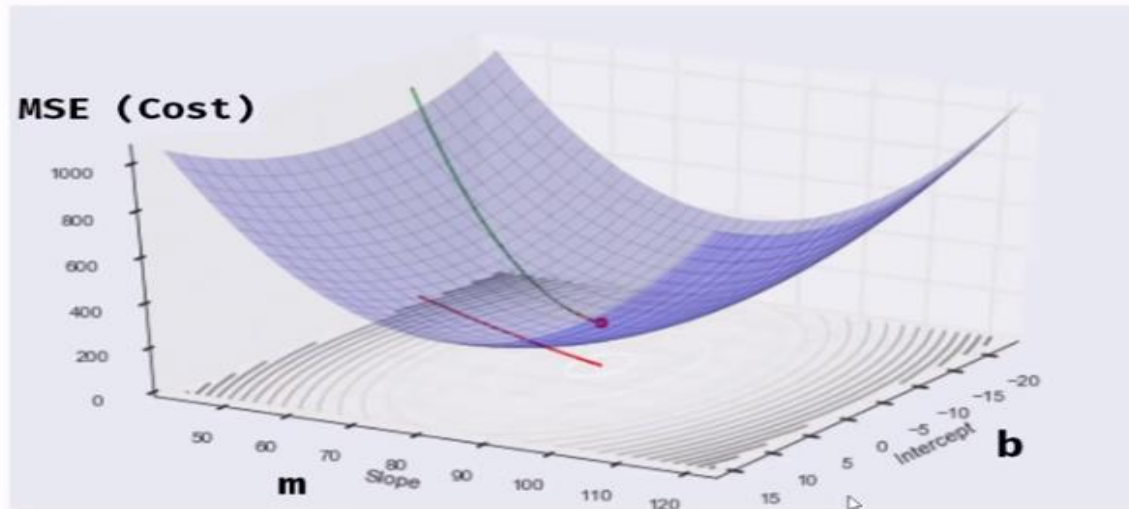
$$mse = \frac{1}{n} \sum_{i=1}^n (y_i - y_{predicted})^2$$

Mean Squared Error

$$mse = \frac{1}{n} \sum_{i=1}^n (y_i - (mx_i + b))^2$$

Cost Function

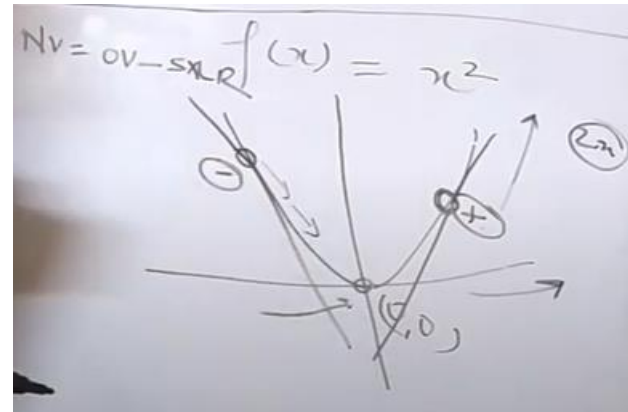
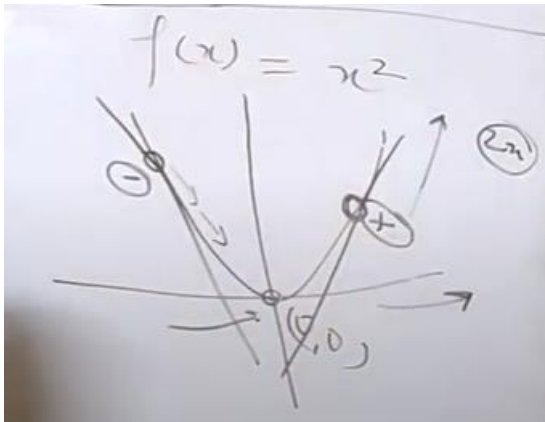
Gradient Descent



Gradient descent

- An algorithm to minimize a function by optimizing parameters
- Example – Maths test max marks 50
- Guess marks – 40. Its too much
- Guess marks – 30 .Its too less
- Guess marks – 35 .. Correct
- $\text{New Value} = \text{Old Value} - \text{step size}$
- $\text{Step size} = \text{Learning rate} * \text{slope}$

Gradient descent



$$J(m, c) = \frac{1}{n} \sum_{i=1}^n (Y_i - (mx_i + c))^2$$

Hand-drawn graph showing the cost function $J(m, c)$ as a function of m and c . The graph shows a 3D surface representing the cost function, with axes labeled m and c . The surface is a paraboloid, and the minimum point is marked with a circled '1' and labeled $J(m, c)$.

Gradient Descent

X	Y
1	2
3	4

← Training data

initial assumptions, $c=0, m=1$

$$J(m, c) = [2 - (c + m \cdot 1)]^2 + [4 - (c + 3m)]^2$$

$$\frac{\partial J}{\partial c} = -2[2 - (c + m)] + [-2(4 - (c + 3m))]$$

$$= -2[2 - (1)] + [-2(4 - 3)]$$

$$= -2[1] + [-2] = -4$$

$$c_{\text{new}} = c_{\text{old}} - LR \times (-4)$$

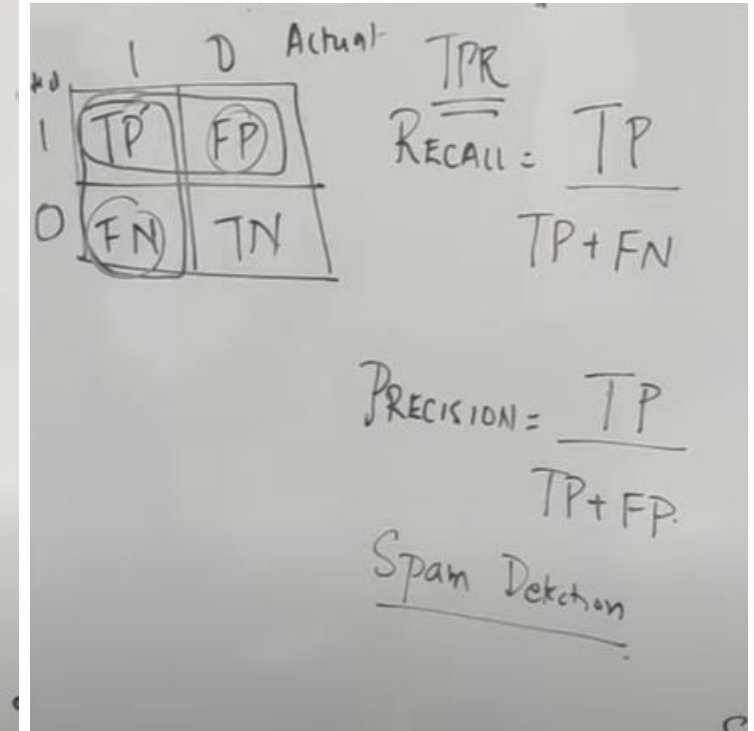
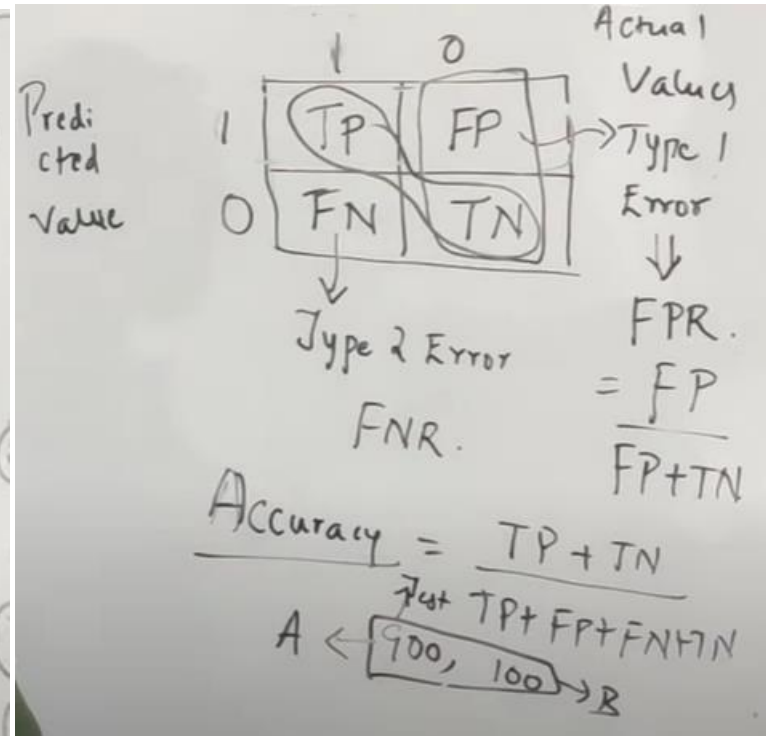
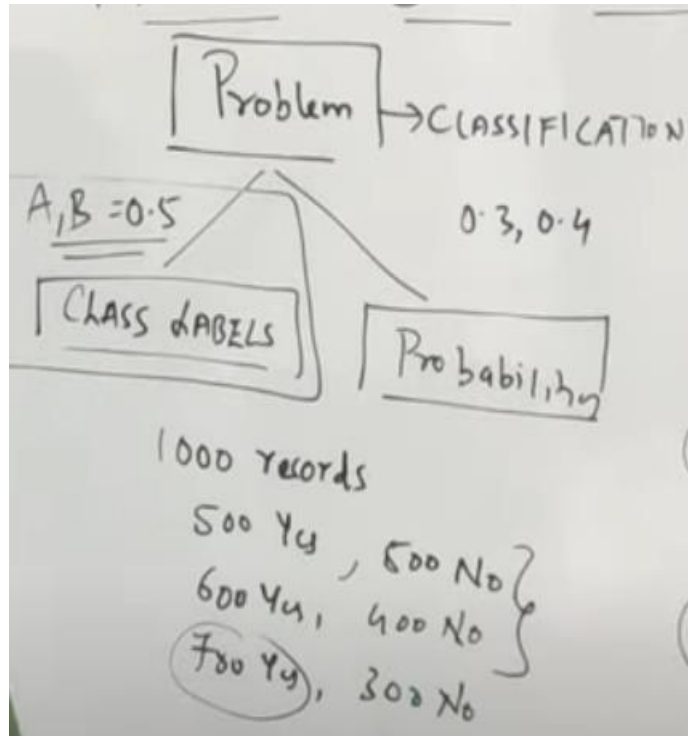
$$= 0 - (0.001 \times (-4))$$

$$= 0.004$$

Metrics In Classification

- 1. Confusion Matrix
- 2. False Positive Rate – FPR (Type 1 Error)
- 3. False Negative Rate – FNR (Type 2 Error)
- 4. Recall (TPR, Sensitivity)
- 5. Precision (+ve Prod Val)
- 6. Accuracy
- 7. F1 Score (F Beta)
- 8. Cohen Kappa
- 9. RoC Curve, AUC Score
- 10. PR Curve

Metrics In Classification



Metrics In Classification

Actual \ Predicted

1	0
1	0

TP (True Positive) FP (False Positive)
FN (False Negative) TN (True Negative)

↓ ↓ ↓

TPR (True Positive Rate) → Cancer or Not

RECALL = $\frac{TP}{TP + FN}$

PRECISION = $\frac{TP}{TP + FP}$

Span Detection

$$F_{\text{Beta}} = \frac{(1 + \beta^2) \text{Precision} \times \text{Recall}}{\beta^2 \text{Precision} + \text{Recall}}$$

$$F_{\text{Beta}} = \frac{(1 + \beta^2) \text{Precision} \times \text{Recall}}{\beta^2 \text{Precision} + \text{Recall}}$$

$\beta = 1$

$\beta = 0.5$

$\beta = 2$ = Harmonic mean

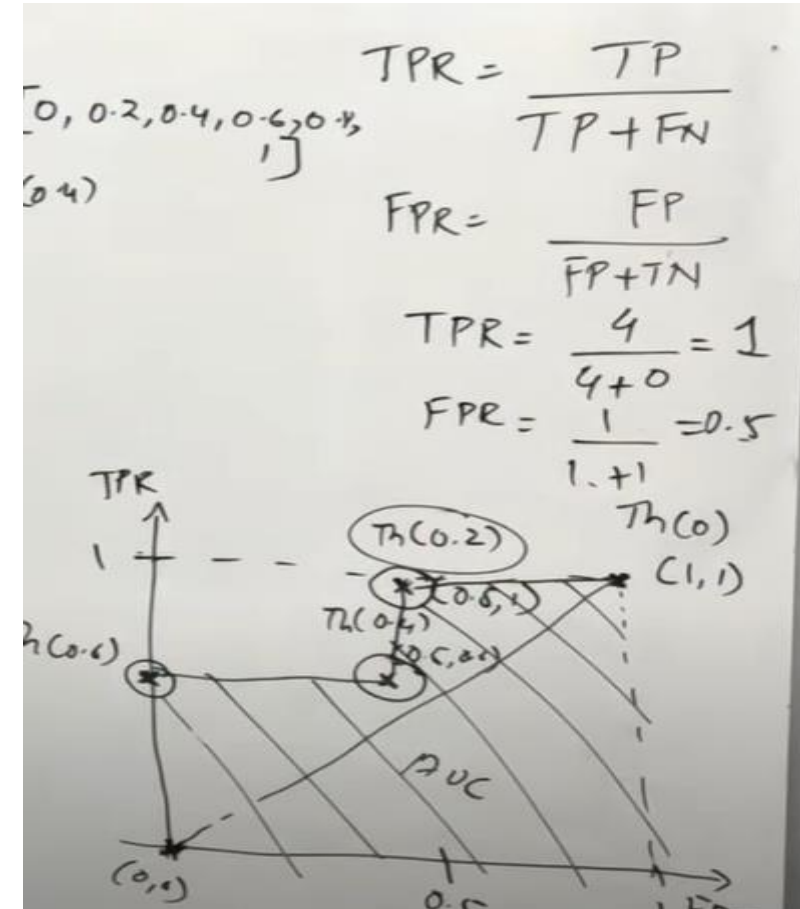
$\left\{ = \frac{2xy}{x+y} \right\}$

Metrics In Classification

ROC AND AUC [0, 0.2, 0.4, 0.6, 0.8, 1]

y	\hat{y}_{prob}	$\hat{y}(0)$	$\hat{y}(0.2)$	$\hat{y}(0.4)$
1	0.8	1	1	
0	0.96	1	1	
1	0.4	1	1	
1	0.3	1	1	
0	0.2	1	0	
1	0.7	1		

TPR ↑



Precision, Recall, F1 Score

Truth										
Prediction	Dog	Dog	Dog	No Dog	Dog	No Dog	Dog	No Dog	Dog	Dog
	✓	✗	✓		✓		✗		✓	✗

True Positive = 4

False Positive = 3












Precision, Recall, F1 Score

Truth										
Prediction	Dog	Dog	Dog	No Dog	Dog	No Dog	Dog	No Dog	Dog	Dog
				✗		✓		✗		

True **Negative** = 1

False **Negative** = 2

Precision, Recall, F1 Score

										
Truth										
Prediction	<div>Dog</div>	<div>Dog</div>	<div>Dog</div>	<div>No Dog</div>	<div>Dog</div>	<div>No Dog</div>	<div>Dog</div>	<div>No Dog</div>	<div>Dog</div>	<div>Dog</div>
	✓	✗	✓	✗	✓	✓	✗	✗	✓	✗

How many we got right? → 5

Accuracy → $5/10 \rightarrow 0.5$

Precision, Recall, F1 Score



Truth										
Prediction	Dog	Dog	Dog	No Dog	Dog	No Dog	Dog	No Dog	Dog	Dog
	✓	✗	✓		✓		✗		✓	✗

True **Positive** = 4

False **Positive** = 3

Precision is out of all **dog predictions** how many you got it right?

$$\text{Precision} = 4 / 7 = \mathbf{0.57}$$

$$\text{Precision} = TP / (TP + FP)$$



Precision, Recall, F1 Score

Truth										
Prediction	<div>Dog</div>	<div>Dog</div>	<div>Dog</div>	<div>No Dog</div>	<div>Dog</div>	<div>No Dog</div>	<div>Dog</div>	<div>No Dog</div>	<div>Dog</div>	<div>Dog</div>
	✓	✗	✓		✓		✗		✓	✗

Recall is out of all **dog truth** how many you got it right?

Total Dog truth samples = 6

True **Positive** = 4

$$\text{Recall} = 4 / 6 = 0.67$$

$$\text{Recall} = TP / (TP + FN)$$

Precision, Recall, F1 Score

For **precision**, think about **predictions** as
your base

For **recall**, think about **truth** as your base

Precision, Recall, F1 Score



Truth										
Prediction	Dog	Dog	Dog	No Dog	Dog	No Dog	Dog	No Dog	Dog	Dog
				✗		✓		✗		

$$\text{Precision} = 1 / 3 = 0.33$$

$$\text{Recall} = 1 / 4 = 0.25$$

F1 Score

- $F1 = 2 * (\text{Precision} * \text{Recall}) / (\text{precision} + \text{recall})$

Hands-on session

- Linear Regression
- Logistic Regression

References

- <https://www.skillbasics.com/>
- 3Blue1BrownSeries
- <https://www.youtube.com/watch?v=IHZwWFHWa-w>

Thank You