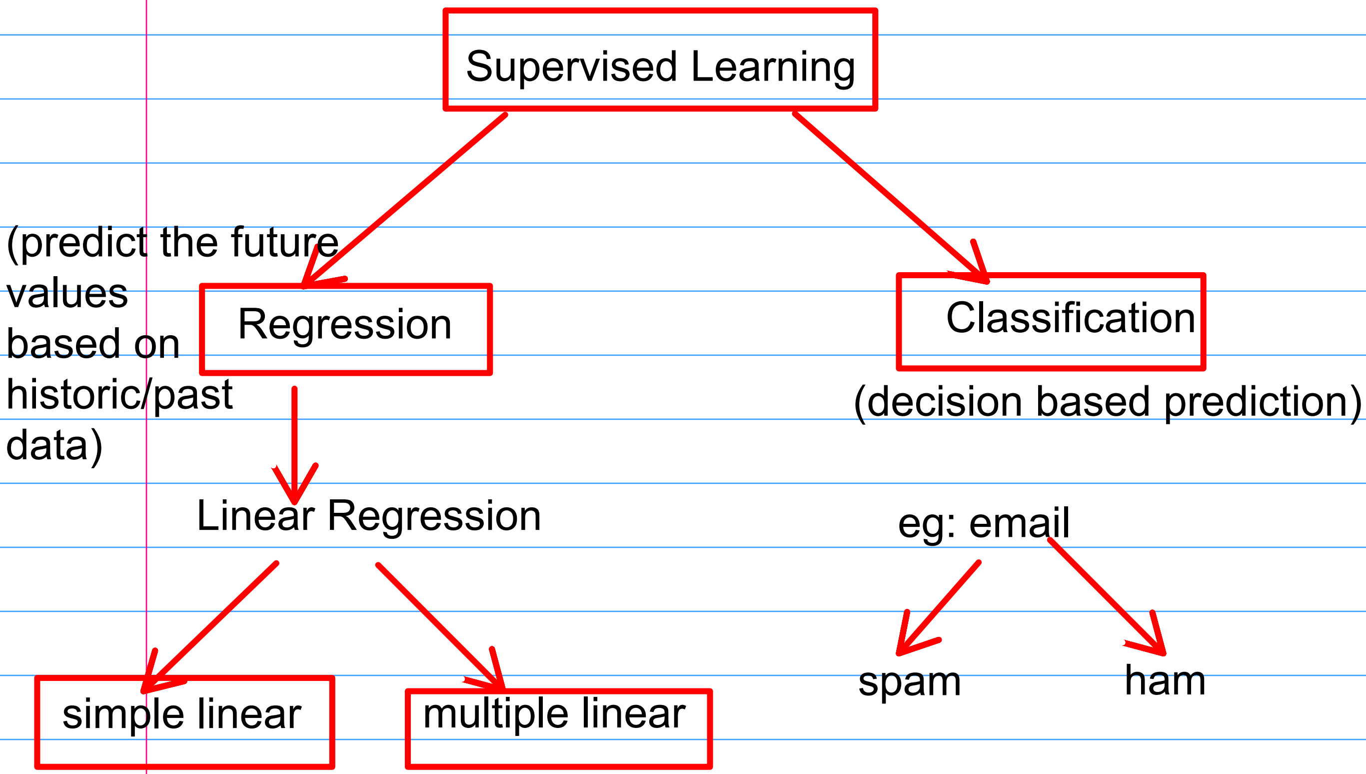
MACHINE LEARNING
Logistic Regression

- Trainer : Sujata Mohite
- Email: sujata.mohite@sunbeaminfo.com


Supervised Learning
(predict the future values based on historic/past data)
Regression
Classification
(decision based prediction)
Linear Regression
simple linear
multiple linear
eg: email
spam
ham

Logistic Regression

- Logistic regression is another supervised learning algorithm which is used to solve the classification problems.
- In **classification problems**, we have dependent variables in a binary or **discrete format** such as 0 or 1.
- It was then used in many social science applications
- Logistic Regression is used when the **dependent variable(target) is categorical** such as 0 or 1, Yes or No, True or False, Spam or not spam, etc.

0/1

- The dependent variable is a binary variable that contains data coded as 1 (yes, success, etc.) or 0 (no, failure, etc.)
- Unlike linear regression, logistic regression can directly predict probabilities (values that are restricted to the (0,1) interval)
- Furthermore, those probabilities are well-calibrated when compared to the probabilities predicted by some other classifiers
- Eg:-

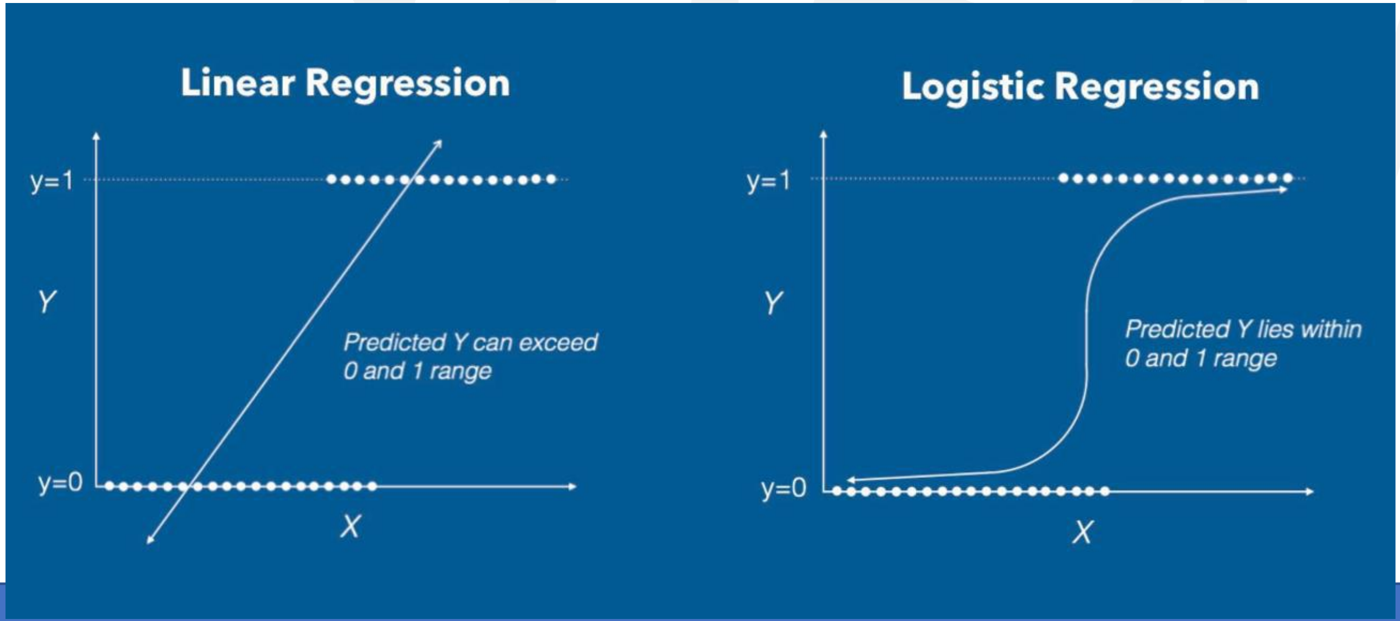
  To predict whether an email is spam (1) or (0)

  Whether the tumors malignant (1) or not (0)

Linear vs Logistic

Linear	Logistic
Target variable is an interval variable	Target variable is a discrete (binary or ordinal) variable
Predicted values are the mean of the target variable at the given values of the input variable	Predicted values are the probability of a particular level(s) of the target variable at the given values of the input variables

0/1



Logistic Regression

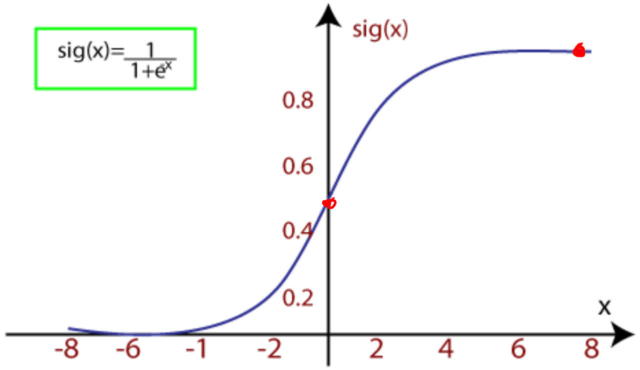
activation function

- Logistic regression uses **sigmoid function** or logistic function which is a complex cost function. This sigmoid function is used to model the data in logistic regression. The function can be represented as:

$$f(x)=\frac{1}{1+e^{-x}}$$

- f(x)=Output between the 0 and 1 value.
- x=input to the function
- e=base of natural logarithm.

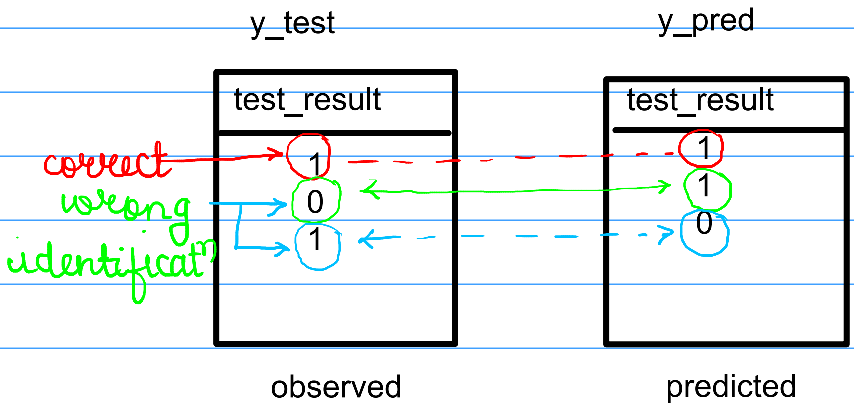
- When we provide the input values (data) to the function, it gives the S-curve as follows:



Classification Model Evaluation Metrics

- For evaluation of classification model, following metrics are used
- Confusion Matrix
- F1 Score
- Auc-Roc
  curve

age	physical_score
20	40
30	20
45	35



right identification = [0,0] + [1,1]
= 0 + 1
= 1

wrong identification = [1,0] +[0,1]
= 1 + 1
= 2

$$\text{accuracy} = \frac{\text{right identification}}{\text{total}}$$

$$= \frac{1}{3}$$

$$= 0.33$$

$$= 33\%$$

my model accuracy is 33%

Confusion Matrix

- A confusion matrix is an N X N matrix, where N is the number of classes being predicted
- The confusion matrix provides more insight into not only the performance of a predictive model, but also which classes are being predicted correctly, which incorrectly, and what type of errors are being made.

Observed	Predicted	
1	1	TP
1	0	FN
0	0	TN
0	1	FP

		Predicted condition	
	Total population = P + N	Yes 1 Predicted condition positive (PP)	No 0 Predicted condition negative (PN)
Actual condition	Actual condition positive (P) Yes 1	[1,1] **True positive (TP),** hit	**False negative (FN),** $\boxed{\text{Type II error,}}$ [1,0] miss, underestimation
	Actual condition negative (N) No 0	[0,1] **False positive (FP),** $\boxed{\text{Type I error,}}$ false alarm, overestimation	[0,0] **True negative (TN),** correct rejection

TP vs FP vs TN vs FN

Cat

Cat

Cat

Cat

No Cat

No Cat

No Cat

Cat

Cat

TP vs FP vs TN vs FN

FP

Cat

TP

Cat

FP

Cat

TP

Cat

TN

No Cat

FN

No Cat

TN

No Cat

TP

Cat

TP

Cat

TP = 4

FP = 2

Total P = 6

TN = 2

FN = 1

Total N = 3

Total = 9

Accuracy

Cat

Cat

Cat

Cat

No Cat

No Cat

No Cat

Cat

Cat

How many we got right ?

Accuracy : How many we got right?

FP	TP	FP	TP	TN	FN	TN	TP	TP
Cat	Cat	Cat	Cat	No Cat	No Cat	No Cat	Cat	Cat

Correct = TP + TN / Total

= 6 / 9

= 2/3

= 0.66

Accuracy

- Percentage of correct predictions out of all the observations.
- Prediction correct only if actual value matches
- Accuracy $= \dfrac{\text{Correct predition}}{\text{Total cases}} \text{ X } 100$
- Accuracy $= \dfrac{\text{TP+TN}}{\text{TP+TN+FP+FN}} \text{ x } 100$

Precision

- Precision talks about how precise/accurate your model is out of those predicted positive, how many of them are actual positive
- Precision is a good measure to determine, when the costs of False Positive is high
- For instance, in email spam detection, a false positive means that an email that is non-spam (actual negative) has been identified as spam (predicted spam).
- The email user might lose important emails if the precision is not high for the spam detection model.

$$\text{Precision} = \frac{True\ Positive}{True\ Positive + False\ Positive} = \frac{\text{TP}}{\text{TP+FP}} \text{ x } 100$$

$$= \frac{True\ Positive}{Total\ Predicted\ Positive}$$

Precision

Cat

Cat

Cat

Cat

No Cat

No Cat

No Cat

Cat

Cat

Out of all Cat predictions how many we got right ?

Precision : Out of all Cat predictions how many we got right ?

FP

Cat

TP

Cat

FP

Cat

TP

Cat

1

TN

No Cat

FN

No Cat

2

TN

No Cat

TP

Cat

TP

Cat

True positive = 4
Total positive = 6
Precision of + ve = 4/6 = 2/3= 0.66

True Negative = 2
Total Negative = 3
Precision of -ve = 2/3 = 0.66

Recall

- Recall actually calculates how many of the Actual Positives our model capture through labelling it as Positive (True Positive)
- ▪ Applying the same understanding, we know that Recall shall be the model metric we use to select our best model when there is a high cost associated with False Negative
- ▪ For instance, in fraud detection or sick patient detection, if a fraudulent transaction (Actual Positive is predicted as non-fraudulent (Predicted Negative), the consequence can be very bad for the bank

$$\text{Recall} = \frac{\textit{True Positive}}{\textit{True Positive} + \textit{False Negative}} = \frac{TP}{TP+FN}$$

$$= \frac{\textit{True Positive}}{\textit{Total Actual Positive}}$$

Recall









Cat | Cat | Cat | Cat | No Cat | No Cat | No Cat | Cat | Cat

Out of all Cat truth how many we got right ?

Recall : Out of all Cat truth how many we got right ?

								
	1		2	1		2	3	4
FP	TP	FP	TP	TN	FN	TN	TP	TP
Cat	Cat	Cat	Cat	No Cat	No Cat	No Cat	Cat	Cat

True positive = 4

Total Actual positive = 5

Recall of + ve = 4/5 = 0.80

$$\text{Recall} = \frac{\textit{True Positive}}{\textit{True Positive} + \textit{False Negative}} = \frac{TP}{TP+FN}$$

True Negative = 2

Total Actual Negative = $\boxed{4}$

Recall of -ve = 2/4 = 0.50

$$\left\{ \text{Recall} = \frac{\text{True Negative}}{\text{True Negative+False Positive}} \right\}$$

F1 Score

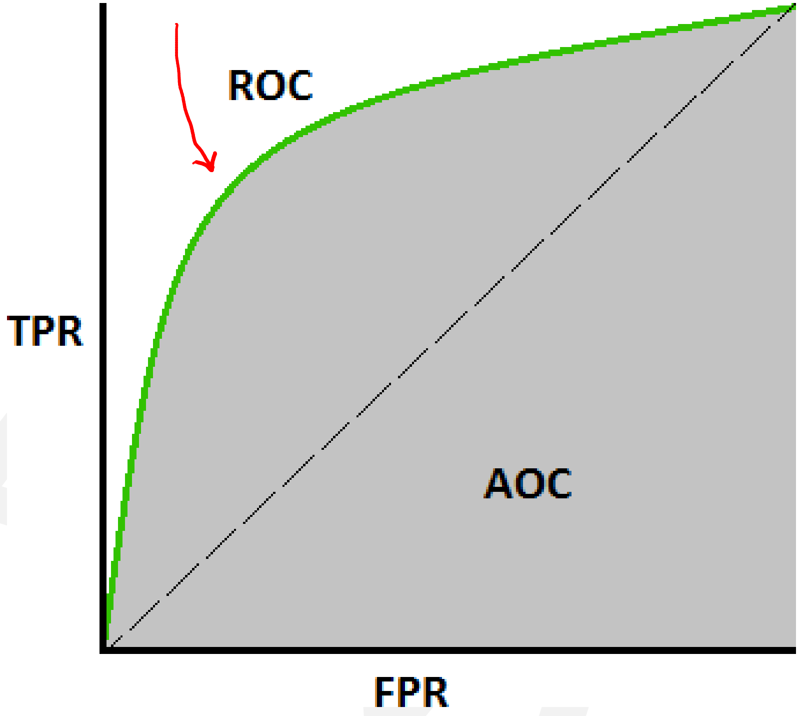
- The F1 score is the harmonic mean of the precision and recall
- The highest possible value of an F-score is 1.0, indicating perfect precision and recall, and the lowest possible value is 0, if either the precision or the recall is zero

$$F1 = 2 \times \frac{Precision * Recall}{Precision + Recall}$$

- Good performance = good F1 score

Receiver Operating Characteristic (ROC)

- ROC curve is a metric that assesses the model ability to distinguish between binary classes
- It is created by plotting the true positive rate (TPR) against the false positive rate (FPR) at various threshold settings
- The TPR is also known as sensitivity, recall or probability of detection in machine learning
- The FPR is also known as the probability of false alarm and can be calculated as 1– specificity
- Points above the diagonal line represent good classification (better than random)
- The model performance improves if it becomes skewed towards the upper left corner



Receiver Operating Characteristic (ROC)

TPR (True Positive Rate) / Recall /Sensitivity

$$\text{TPR /Recall / Sensitivity} = \frac{TP}{TP + FN}$$

Image 3

Specificity

$$\text{Specificity} = \frac{TN}{TN + FP}$$

Image 4

FPR

$$\text{FPR} = 1 - \text{Specificity}$$

$$= \frac{FP}{TN + FP}$$

Performance Measures

Classification

- Accuracy Score : measures how often the classifier correctly predicts
- F1-Score : the harmonic mean of precision and recall.
- Confusion Matrix :It is a matrix of size 2×2 for binary classification with actual values on one axis and predicted on another

$$\textbf{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}$$

$$F1 = 2.\ \frac{Precision \ \times Recall}{Precision + Recall}$$

$$Precision = \frac{TruePositive}{TruePositive + FalsePositive}$$

$$Recall = \frac{TruePositive}{TruePositive + FalseNegative}$$

		ACTUAL	
		Negative	Positive
PREDICTION	Negative	TRUE NEGATIVE	FALSE NEGATIVE
	Positive	FALSE POSITIVE	TRUE POSITIVE

ML Model Training Flow

original dataset ---> (x,y)

↓

supply the train & test data(x,y) ---> x_train,x_test,y_train,y_test

↓

select the model ---> model=algos()
(select the algos)

↓

train the model ---> model.fit()
(make your ML model to learn)

↓

test data

Prediction ---> model.predict(data)

↓

Data visualization ---> matplotlib / seaborn

THANK YOU!