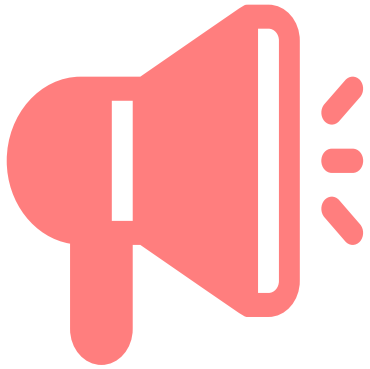
Introduction To Logistic Regression

DATA FOLKZ

CATAPULT DATA LEADERS



DATA FOLKZ®
#CATAPULT DATA LEADERS

Supervised Machine Learning

Types Of Problem Statement

Classification

Used when dependent variable is categorical.

Regression

Used when dependent variable is continuous

Linear Regression

DATA FOLKZ®
#CATAPULT DATA LEADERS

ID	Size	Floor	Broadband Rate	Energy Rating	Rental Price
1	500	4	8	C	320
2	550	7	50	A	380
3	620	9	7	A	400
4	630	5	24	B	390
5	665	8	100	C	385
6	700	4	8	B	410
7	770	10	7	B	480
8	880	12	50	A	600
9	920	14	8	C	570
10	1,000	9	24	B	620

Binary Classification

DATA FOLKZ®
#CATAPULT DATA LEADERS

Table: An email spam prediction dataset.

ID	Suspicious Words	Unknown Sender	Contains Images	Class
376	true	false	true	spam
489	true	true	false	spam
541	true	true	false	spam
693	false	true	true	ham
782	false	false	false	ham
976	false	false	false	ham

Binary Classification

DATA FOLKZ®
#CATAPULT DATA LEADERS

	Bag-of-Words							
ID	MONEY	FREE	FOR	GAMBLING	FUN	MACHINE	LEARNING	SPAM
1	3	0	0	0	0	0	0	true
2	1	2	1	1	1	0	0	true
3	0	0	1	1	1	0	0	true
4	0	0	1	0	3	1	1	false
5	0	1	0	0	0	1	1	false

What is Logistic Regression?

DATA FOLKZ®
#CATAPULT DATA LEADERS

In Linear Regression, We quantify the relationship between dependent variable and independent variable. Here independent variable is a continuous variable

But what will happen if our dependent variable is in the form of **binary categories**. In this situations, we use **Logistic Regression .**

The binary outcome have two potential values, potentially a yes/no question.

Logistic Regression calculates the probability and tells that a predicted value belongs to a specific class.

Lets Explore an Example:

DATA FOLKZ®
#CATAPULT DATA LEADERS

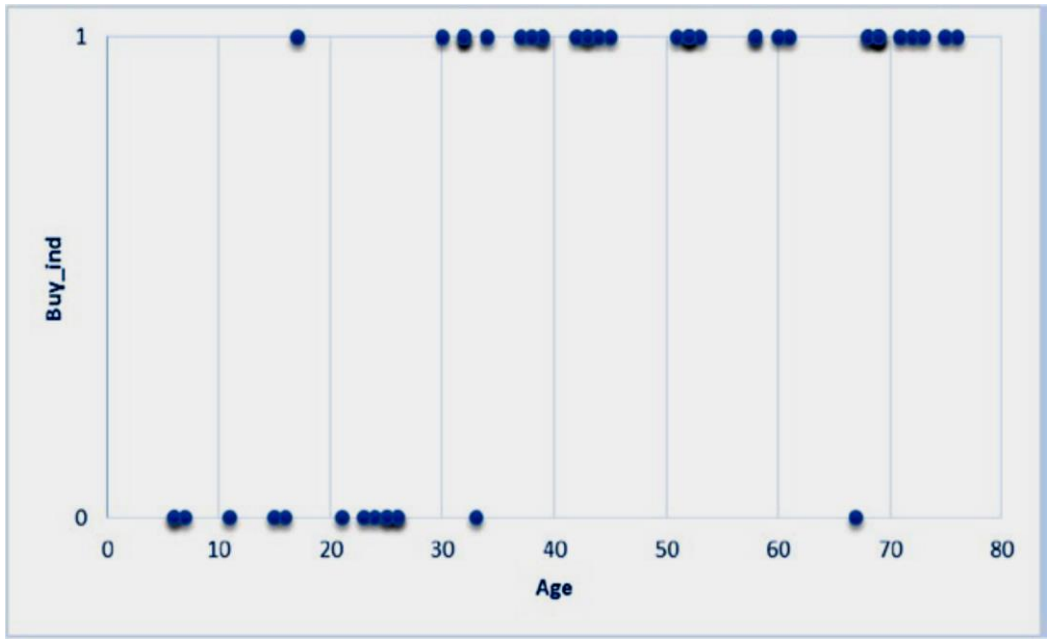
Table shows a table that contains the ice-cream sales data for an ice-cream shop. For now, you are recording just two variables: the age of the customer and whether she buys ice cream. The data set has an indicator variable called "Buy_ind", which takes a **value of 0 if a customer buys ice cream** and **a value of 1 if not**. You are asked to predict ice-cream sales using age as an independent variable.

Age	Buy_id
6	0
25	0
32	1
44	1
34	1
43	1
72	1
67	0
58	1
15	0
42	1

Lets Explore an Example:

DATA FOLKZ®
#CATAPULT DATA LEADERS

Let’s create a scatter plot showing dependent variable on y-axis i.e. “Buy_ind” and independent variable on x-axis i.e. “Age”.



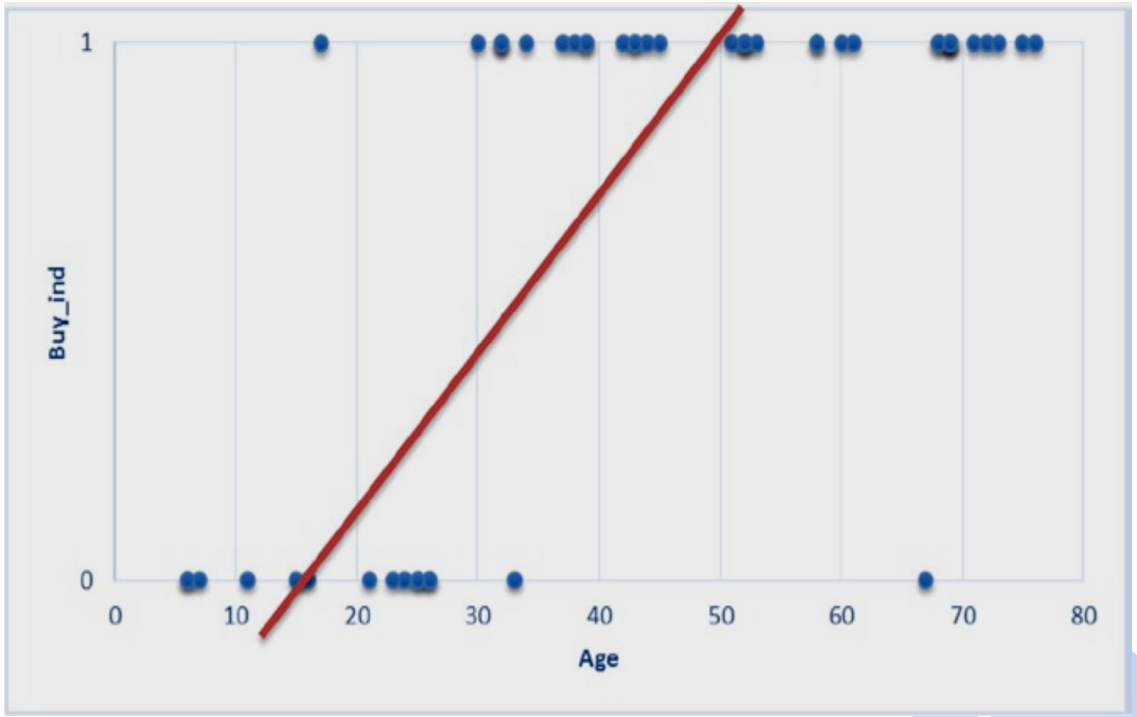
When you take a close look at the data, you can easily observe a pattern in it. You can see that when customer is younger, the buy_ind variable is 0 most of the time, and when customer is older; buy_ind is 1 most of the time.

This shows that older people are not buying ice cream.

Let's Explore an Example:

- The data has some apparent pattern, but you are not able to capture it using the linear regression models. Maybe a linear regression line is not a good fit for this data .

- As the probability can only take values between 0 to 1 , we need a different approach to ensure that our model is appropriate for the data .



Problems with Linear Regression equation

For linear regression, the model is defined by : $y = \beta 0 + \beta 1 x$ -

Lets understand what problems may arise

(i) If we make predictions with a line we can say for values of y<0 predict 0

For values of y>0 predict 1.

But we can clearly see that with a prediction model like this some points are incorrectly classified

(ii) Here we are assuming that all observations have a definite order that is all observations with class 0 should come before all observations with class 1. Even if we shift the threshold we are assuming an order

iii) We are looking for a categorical outcome but we are training the data to fit a regression model which will only produce continuous outcome

iv)For some values of x it may predict a value greater than 1 and for some values a value less than 0 which is not desirable.

Linear Regression is not the best tool for a classification task

Logistic Regression

Logistic Regression is more suited for a Binary Classification task.

Here we predict the likelihood of an event being Yes/No or Positive/Negative

Logistic regression, also takes a linear combination of the features but it connects the outcome to a probability, i.e. y is the probability of a given variable x belonging to a certain class.

So we need a probabilistic function to get the probabilities between 0 to 1

Logistic Regression uses a link function and turns the continuous values of linear combination $y = \beta 0 + \beta 1 x$ into a probability

Link Function

$$g(p(y = 1|X) = \beta_0 + \beta_1 x$$

$$logit(p) = \beta_0 + \beta_1 x$$

The logit of the probability is the logarithm of the odds

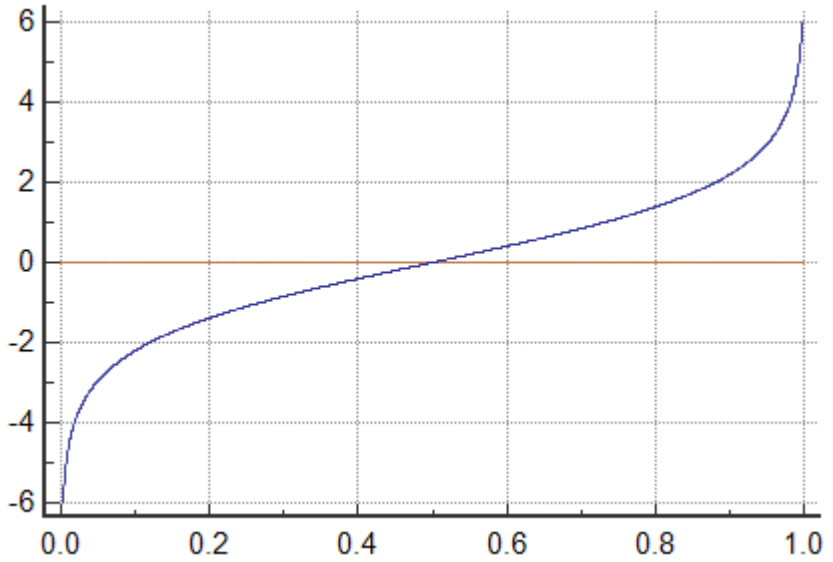
$$logit\left(\frac{p}{1-p}\right) = \beta_0 + \beta_1 x$$

The purpose of the **logit** link is to take a linear combination of the values (which may take any value between ±∞) and convert those values to the scale of a probability, i.e., between 0 and 1

LOGIT FUNCTION – LINK FUNCTION

•The Logit function or the log-odds is the logarithm of the odds
p / (1-p) where p is probability.

•It is a type of function that creates a map of probability values from (0,1) to (-∞,+ ∞).



Logistic Regression

Mathematically speaking, what you achieved is the inverse of the logit function, a function that's called Logistic Function

$$p = \frac{1}{1 + e^{-(\beta_0 + \beta_1 x_1)}} \equiv p(z) = \frac{1}{1 + e^{-z}}$$

Logistic Regression

So, we use Sigmoid Function which is given by :

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

This function will give results between 0 to 1 for any value of x.

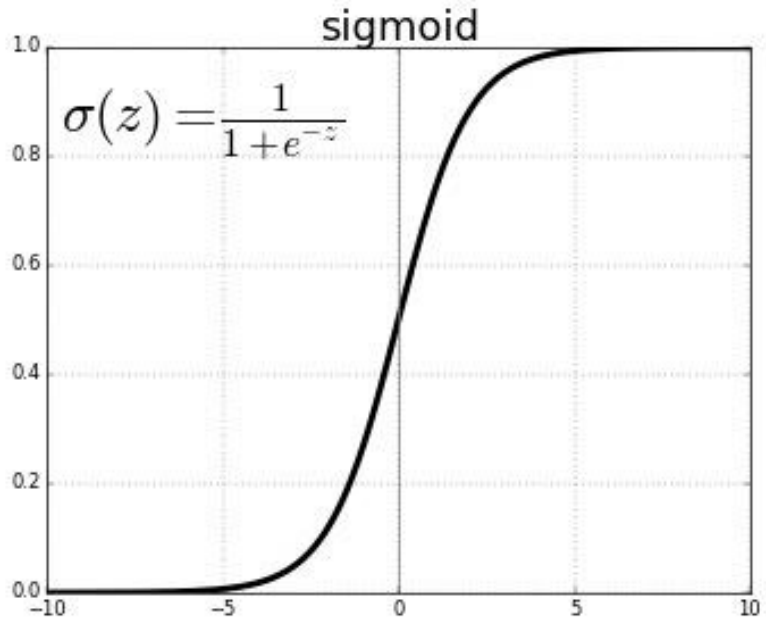
At the end of this process the probability p, the probability of success, is a function of a linear combination.

SIGMOID FUNCTION

•A sigmoid function is a mathematical function having a characteristic "S"-shaped curve or sigmoid curve.

•Used to map the predicted values to probabilities, we use the sigmoid function.

•The function maps any real value into another value between 0 and 1.



Equivalent forms of logistic regression model :

Probability form :

$$p(X) = \frac{e^{\beta_0+\beta_1 X}}{1+e^{\beta_0+\beta_1 X}}.$$

Logit form :

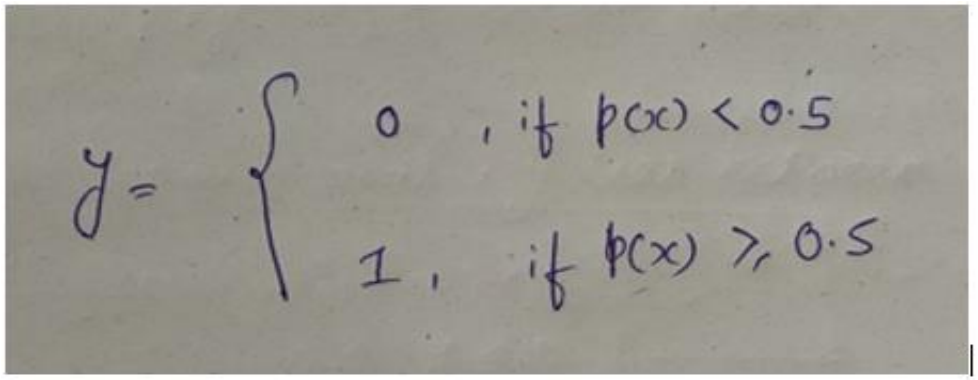
$$\log\left(\frac{p(X)}{1-p(X)}\right) = \beta_0 + \beta_1 X$$

Predictions And Threshold

Our **prediction function** returns a probability score between 0 and 1.

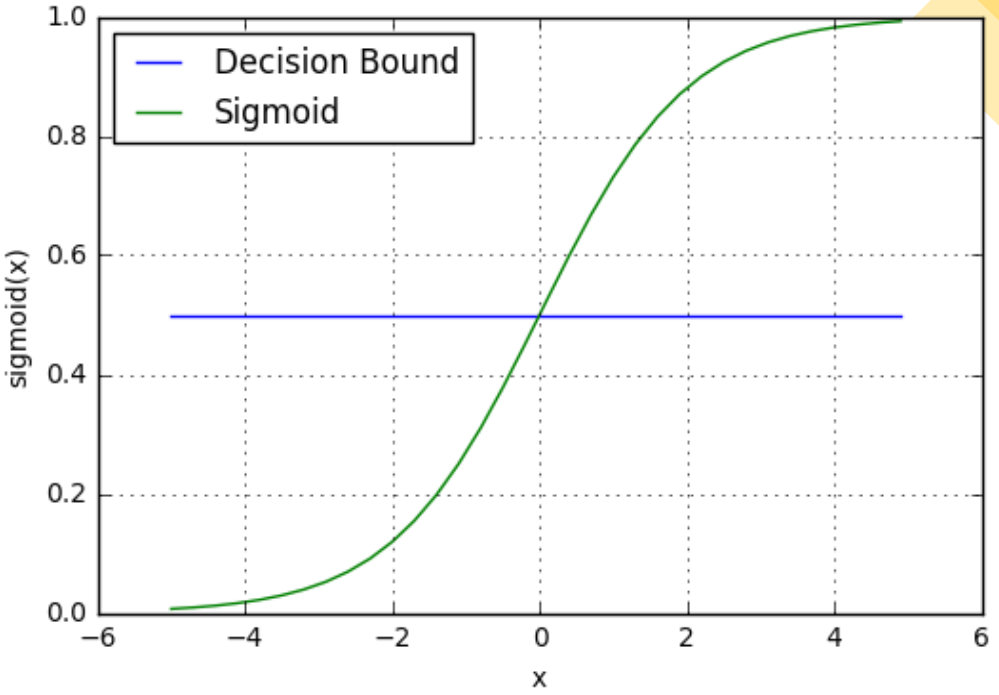
In order to map this to a discrete class (0/1, yes/no), we select a threshold value or tipping point above which we will classify values into class 1 and below which we classify values into class 2.

$p \geq 0.5, class=1$
$p < 0.5, class=0$



Reference : ML Glossary

Predictions And Threshold



For example, if our threshold was .5 and our prediction function returned .7, we would classify this observation as positive (class 1).

If our prediction was .2 we would classify the observation as negative (class 0) .

Reference : ML Glossary

Cost Function in Linear regression

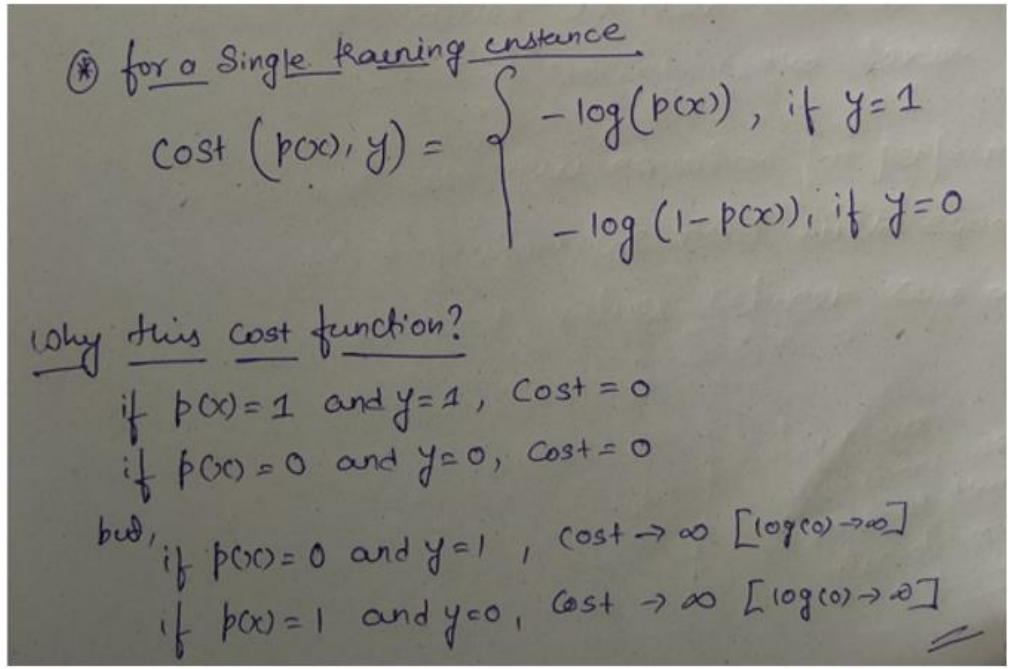
$$Error \;\; E = \sum_{i=1}^{n} (t_i - y_i)^2$$

$$= \sum_{i=1}^{n} (t_i - (w[0] + w[1] * size_i))^2$$

This function represents optimization objective i.e. we create a cost function and minimize it so that we can develop an accurate model with minimum error.

Cost Function

called Cross-Entropy, also known as Log Loss



Where , p(x) = Predicted value
y = actual value

The Cost Function gives us the errors of our predictions and subsequently, is needed for our learning algorithm. Our aim is to minimize the errors of our predictions, i.e., to minimize the cost function.

Cross-entropy loss can be divided into two separate cost functions: one for y=1 and one for y=0 (as shown in the image)

When we are expecting y=1, cost function is –log(p(x))
When we are expecting y=0 , cost function is –log(1-p(x))

In case of y=0 , p(x)=0 and y=1 , p(x)=1 , our cost = 0 ,
Otherwise infinity .

The cost function for the whole training set is given as :

$$J(\theta) = -\frac{1}{m}\Sigma_{i=1}^{m}\left[y^{(i)}log\left(\hat{p}^{(i)}\right) + \left(1-y^{(i)}\right)log\left(1-\hat{p}^{(i)}\right)\right]$$

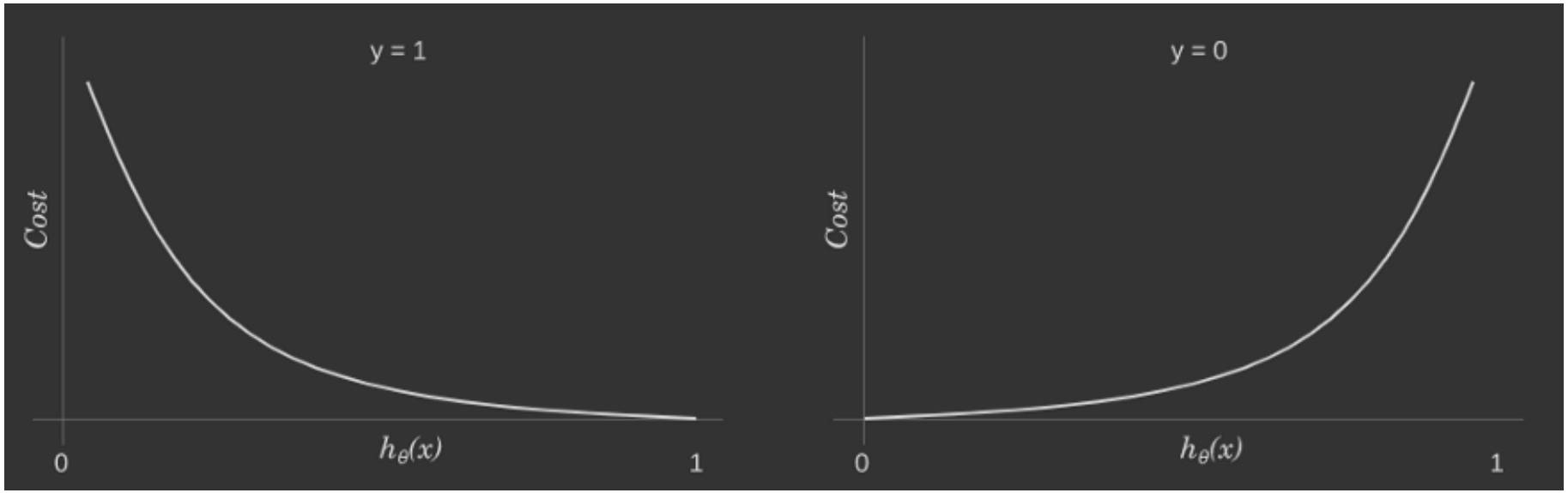
Cost Function

DATA FOLKZ®
#CATAPULT DATA LEADERS

called Cross-Entropy, also known as Log Loss

-log(h(x))

-log(1- h(x))



In the above graph ,
The figure on left hand side is for y=1 , where we can see, when the predicted probability (x-axis) is close to 1, the cost is less and when the predicted probability is close to 0, loss approaches infinity.

The figure on right hand side is for y=0 , where we can see, when the predicted probability (x-axis) is close to 0, the cost is less and when the predicted probability is close to 1, loss approaches infinity.

Gradient Descent

DATA FOLKZ®
#CATAPULT DATA LEADERS

The main goal of Gradient descent is to **minimize the cost value.** i.e. min J(***θ***).
Now to minimize our cost function we need to run the gradient descent function on each parameter i.e.

$$\theta_j := \theta_j - \alpha \frac{\partial}{\partial \theta_j} J(\theta)$$

Gradient Descent

DATA FOLKZ®
#CATAPULT DATA LEADERS

Table shows a sample dataset with a categorical target feature. This dataset contains measurements of the revolutions per minute (RPM) that power station generators are running at, the amount of vibration in the generators (VIBRATION), and an indicator to show whether the generators proved to be working or faulty the day after these measurements were taken

A dataset listing features for a number of generators.

ID	RPM	VIBRATION	STATUS
1	568	585	good
2	586	565	good
3	609	536	good

Gradient Descent

DATA FOLKZ®
#CATAPULT DATA LEADERS

$$\mathbb{M}_{\mathbf{w}}(\langle \text{RPM}, \text{VIBRATION} \rangle) = \frac{1}{1 + e^{-(-0.4077 + 4.1697 \times \text{RPM} + 6.0460 \times \text{VIBRATION})}}$$

$$P(t = faulty|\mathbf{d}) = \mathbb{M}_{\mathbf{w}}(\mathbf{d})$$

$$P(t = good|\mathbf{d}) = 1 - \mathbb{M}_{\mathbf{w}}(\mathbf{d})$$

Implementation

DATA FOLKZ®
#CATAPULT DATA LEADERS

Implementing Logistic Regression:

```
From sklearn.linear_model import LogisticRegression
Model= LogisticRegression( )
Model.fit(X_train,  y_train)
Model.predict(X_test)
```

Classification Metrics

Confusion Matrix:

It is a tabular representation of Actual Vs Predicted values.

It allows the visualization of the performance of an algorith

A confusion matrix is a summary of prediction results on a classification problem. A binary classifier can give four types of errors

Predicted Values \ Actual Values	Positive (1)	Negative (0)
Positive (1)	TP	FP
Negative (0)	FN	TN

Terminologies

True Positive (TP): Observation is positive, and is predicted to be positive. You predicted that an email is spam and it actually is.

False Negative (FN): Observation is positive, but is predicted negative You predicted that an email is not spam(ham) and actually it is spam. It is also known as Type 2 Error.

Terminologies

True Negative (TN): Observation is negative, and is predicted to be negative. You predicted that an email is not spam and it is actually not spam

False Positive (FP): Observation is negative, but is predicted positive.
You predicted that an email is spam but actually is not.
It is also known as Type 1 Error

Confusion Matrix Based Performance Measures

$$TPR = \frac{TP}{(TP + FN)}$$

$$TNR = \frac{TN}{(TN + FP)}$$

$$FPR = \frac{FP}{(TN + FP)}$$

$$FNR = \frac{FN}{(TP + FN)}$$

There are strong relationships between these measures, for example: FNR = 1 − TPR, and FPR = 1 − TNR. All these measures can have values in the range [0, 1]. Higher values of TPR and TNR indicate better model performance, while the opposite is the case for FNR and FPR.

DATA FOLKZ®
#CATAPULT DATA LEADERS

Precision

The confusion matrix allows many other measure of performance to evaluate the model

.

“Out of all the total predicted positive classes , how many Results are actually positive”

$$\textit{Precision} = \frac{TP}{TP + FP}$$

It is also known as Positive predictive value (PPV). It is calculated by dividing the total number of correctly classified positive examples by the total number of predicted positive examples.

Recall

**“Out of all Actual Positives Values,
how much We have predicted correctly”**

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

It is also known as True Positive Rate.
It is calculated as the number of correct positive predictions divided by total number of positives.
Both precision and recall can assume values in the range [0, 1], and higher values in both cases indicate better model performance.

DATA FOLKZ®
#CATAPULT DATA LEADERS

F1 Score

Since we have two measures (Precision and Recall),
it helps to have a measurement that represents both of them. We calculate an F-measure, which uses Harmonic Mean in place of Arithmetic Mean as it punishes the extreme values more.
The F-Measure will always be nearer to the smaller value of Precision or Recall.
Higher value indicates better performance

$$F\text{-}measure = \frac{2*Recall*Precision}{Recall + Precision}$$

DATA FOLKZ®
#CATAPULT DATA LEADERS

F1 Score

Precision, recall, and the F1 measure work best in prediction problems with binary target features and place an emphasis on capturing the performance of a prediction model on the positive, or most important, level.

These measures place less emphasis on the performance of the model on the negative target level.

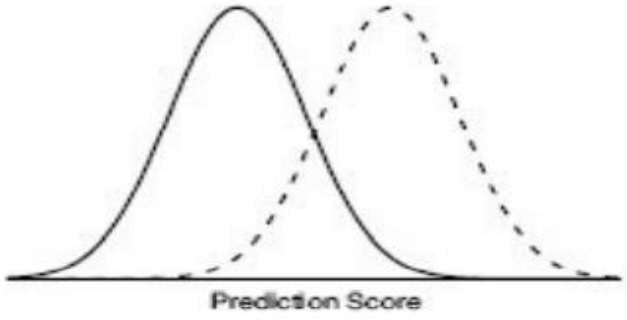
This is appropriate in many applications. For example, in medical applications, a prediction that a patient has a disease is much more important than a prediction that a patient does not.

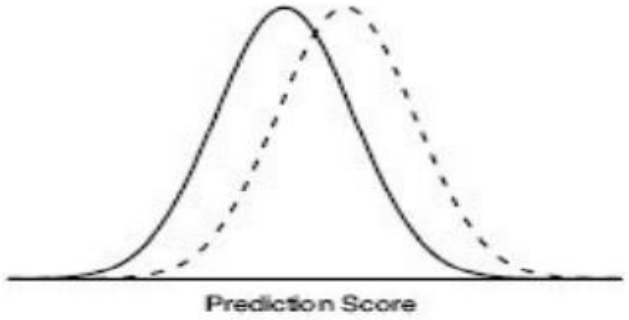
In many cases, however, it does not make sense to consider one target level as being more important.

Prediction score distributions for two different prediction models for two classes . The distributions in (a) are much better separated than those in (b).

ID	Target	Prediction	Score	Outcome
7	ham	ham	0.001	TN
11	ham	ham	0.003	TN
15	ham	ham	0.059	TN
13	ham	ham	0.064	TN
19	ham	ham	0.094	TN
12	spam	ham	0.160	FN
2	spam	ham	0.184	FN
3	ham	ham	0.226	TN
16	ham	ham	0.246	TN
1	spam	ham	0.293	FN

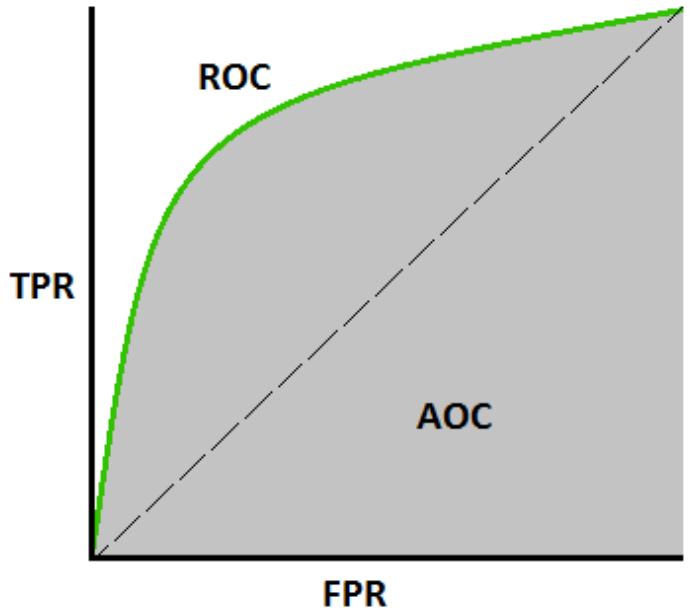
ID	Target	Prediction	Score	Outcome
5	ham	ham	0.302	TN
14	ham	ham	0.348	TN
17	ham	spam	0.657	FP
8	spam	spam	0.676	TP
6	spam	spam	0.719	TP
10	spam	spam	0.781	TP
18	spam	spam	0.833	TP
20	ham	spam	0.877	FP
9	spam	spam	0.960	TP
4	spam	spam	0.963	TP





ROC – AUC Curve

ROC – AUC curves is a performance measurement for classification problem at various thresholds. **ROC** is a probability curve and **AUC** represents degree or measure of separability. It stands for "Area under the ROC Curve.

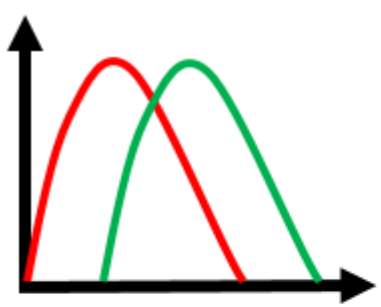


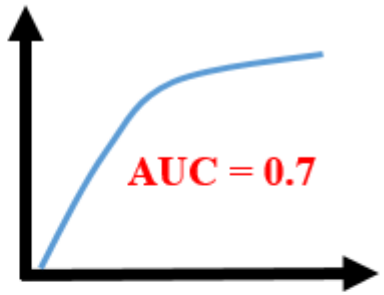
IAs the strength of a predictive model increases, the ROC curve moves farther away from the random line toward the top left hand corner of ROC space—toward a TPR of 1.0 and an FPR of 0.0.

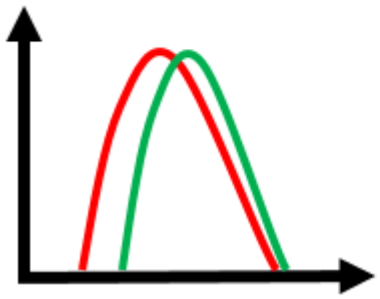
So, the ROC curve gives us an immediate visual indication of the strength of a model—the closer the curve is to the top left, the more predictive the model

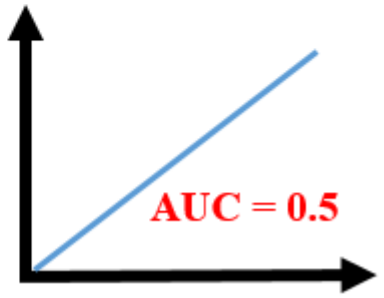
When AUC is 0.7, it means there is 70% chance that model will be able to distinguish between positive class and negative class.

Area Under Curve (AUC)





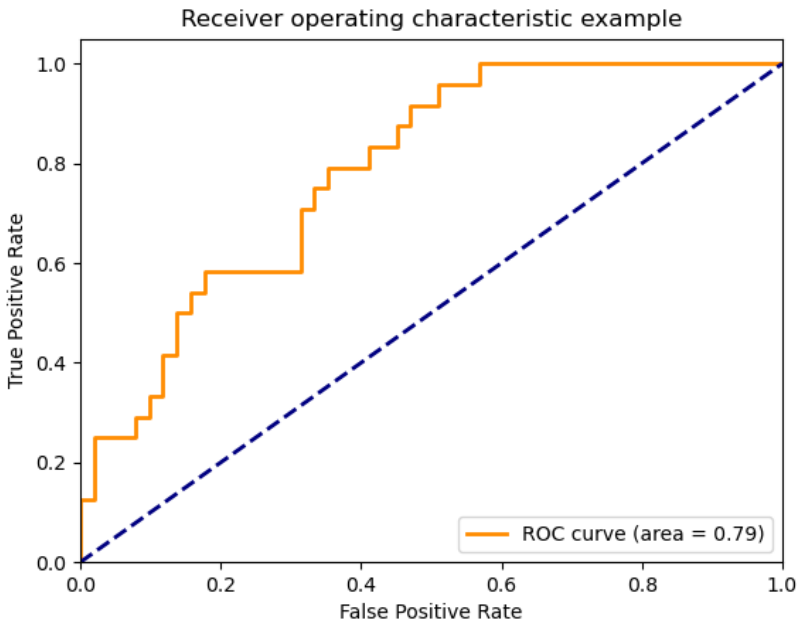




AUC represents degree or measure of separability. It tells how much model is capable of distinguishing between classes.

Higher the AUC, better the model is at predicting 0s as 0s and 1s as 1s.
By analogy, Higher the AUC, better the model is at distinguishing between patients with disease and no disease .

Receiver Operating Characteristics (ROC)

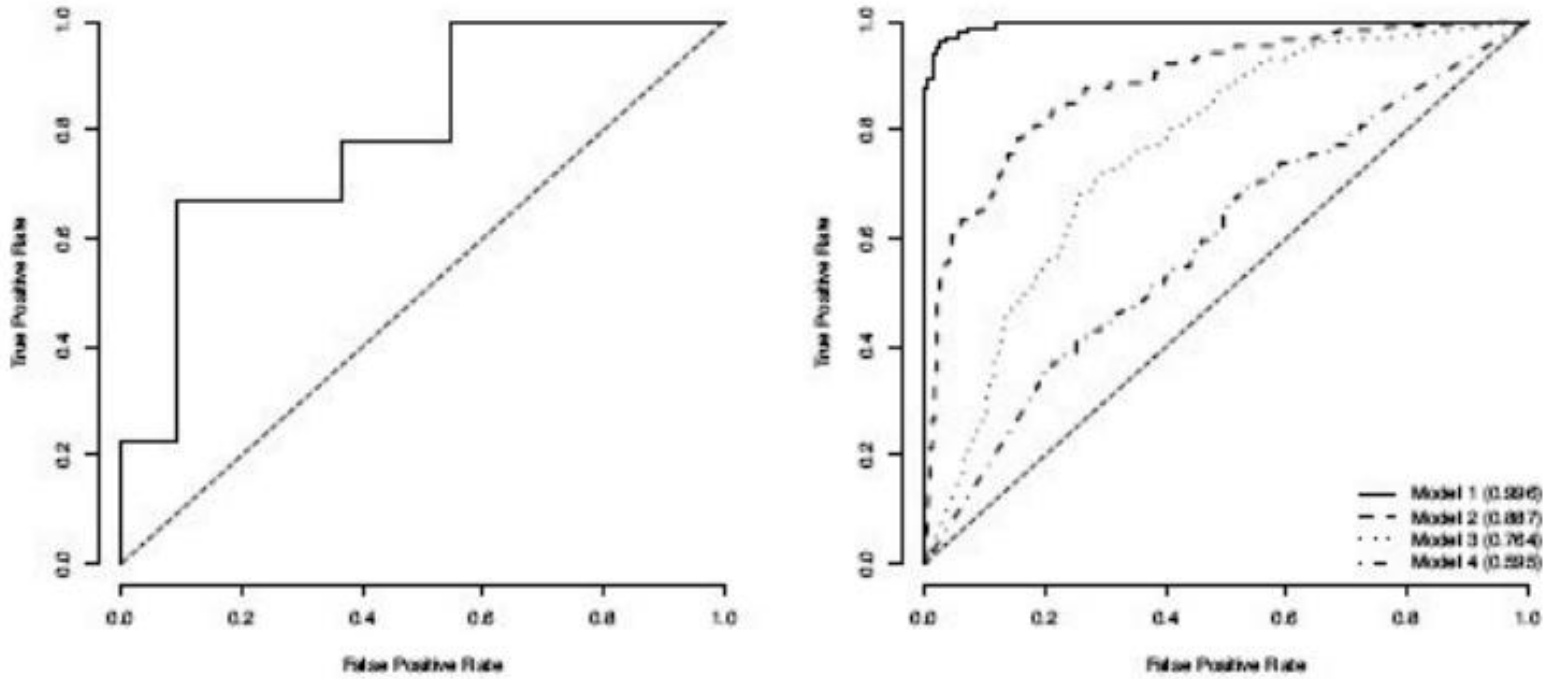


- The ROC curve is created by plotting the true positive rate (TPR) against the false positive rate (FPR) at various threshold settings. The true-positive rate is also known as sensitivity/recall

TPR = TP/TP+FN

- The false-positive rate is also known as probability of false alarm

FPR= FP/FP+TN

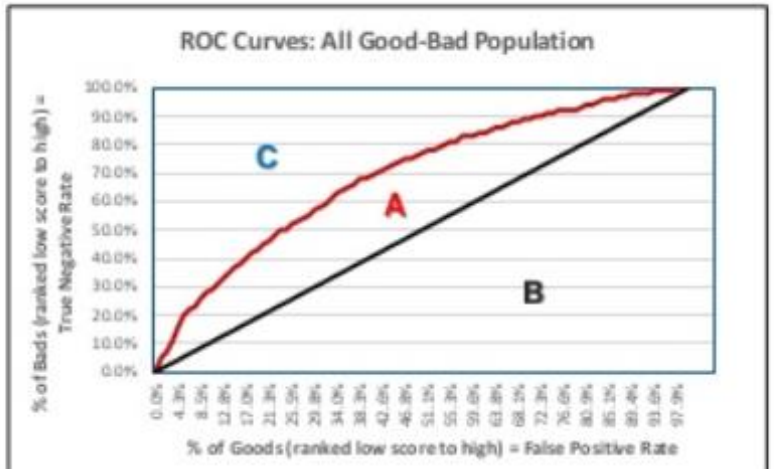


(a) (b)

(a) A complete ROC curve for the email classification example; (b) a selection of ROC curves for different models trained on the same prediction task.

Model Gini

- AUC = Area A + Area B
- Gini Index = Area A / (Area A + Area C)
  - Even more convenient: Gini Index = 2*AUC - 1



Gini coefficient is one of the most popular metrics used by the financial industry for evaluating the performance of credit score models.
It is directly related to the area under the ROC curve i.e AUC curve .

The Gini coefficient is calculated from the area under the curve (AUC) as **2AUC – 1.**
A 74% area under the curve becomes a Gini coefficient of 48% . The Gini coefficient hence effectively ranges between 0% and 100%

Relation between Sensitivity, Specificity, FPR and Threshold

•Sensitivity and Specificity are inversely proportional to each other. When we increase Sensitivity, Specificity decrease and when we increase Specificity, Sensitivity decreases.

•When we decrease the threshold, we get more positive values thus it increases the sensitivity and decreasing the specificity. Similarly, when we increase the threshold, we get more negative values thus we get higher specificity and lower sensitivity.

Metrics at Different Threshold

	TNR	TPR	f1	Precision	Recall
Threshold					
1.000000	0.999905	0.047089	0.089869	0.982143	0.047089
0.845047	0.977476	0.220890	0.310283	0.521212	0.220890
0.778575	0.951815	0.345034	0.387873	0.442857	0.345034
0.715466	0.916556	0.459760	0.415796	0.379505	0.459760
0.651414	0.870557	0.568493	0.415780	0.327739	0.568493
0.590597	0.820186	0.676370	0.410390	0.294556	0.676370
0.504493	0.731895	0.773973	0.369507	0.242685	0.773973
0.420640	0.634765	0.863014	0.334939	0.207792	0.863014
0.258778	0.437940	0.954623	0.272051	0.158629	0.954623

An Example showing change in Metrics With different threshold value.

Implementation

DATA FOLKZ®
#CATAPULT DATA LEADERS

Implementing Logistic Regression:

```
From sklearn.linear_model import LogisticRegression
Model= LogisticRegression( )
Model.fit(X_train,  y_train)
Model.predict(X_test)
```

Implementation

DATA FOLKZ®
#CATAPULT DATA LEADERS

Implementing Logistic Regression:

```
From sklearn.linear_model import LogisticRegression
Model= LogisticRegression( )
Model.fit(X_train,  y_train)
Model.predict(X_test)
```

The ROC curve is drawn by plotting a point for every feasible threshold value and joining them. Figure 8.12(a)[429] shows a complete ROC curve for the email predictions in Table 8.13[427]. A line along the diagonal of ROC space from (0, 0) to (1, 0), shown as a dotted line in Figure 8.12(a)[429], is a reference line representing the expected performance of a model that makes random predictions. We always expect the ROC curve for a trained model to be above this random reference line.[15] In fact as the strength of a predictive model increases, the ROC curve moves farther away from the random line toward the top left hand corner of ROC space—toward a TPR of 1.0 and an FPR of 0.0. So, the ROC curve gives us an immediate visual indication of the strength of a model—the closer the curve is to the top left, the more predictive the model.


DATA FOLKZ®
#CATAPULT DATA LEADERS
Thank you