python

Class: Machine Learning

Topic

Classification Problems and the Concept of Generalization

Linear Classification

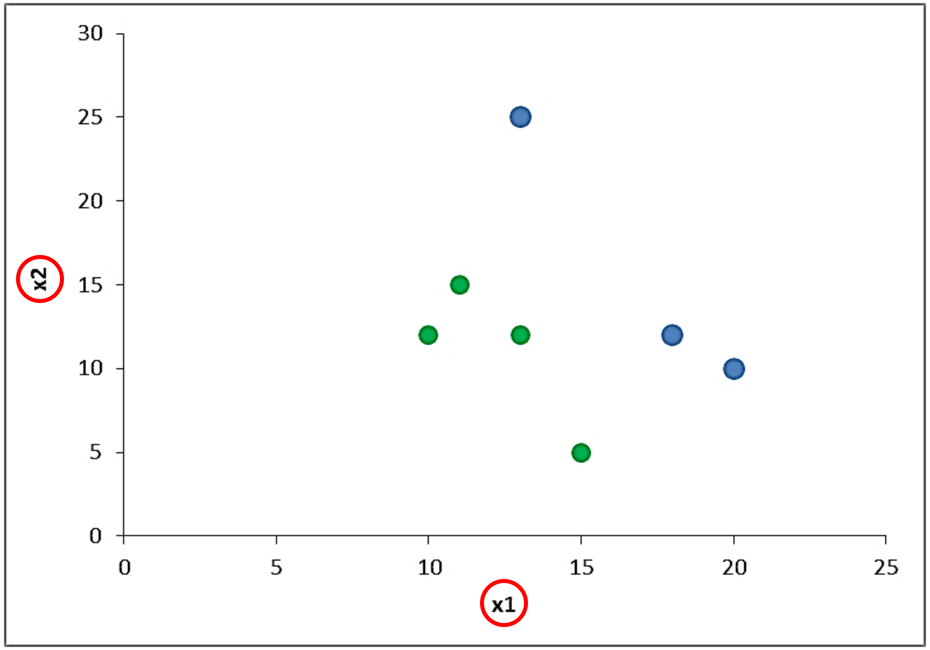
Classification is a problem emerging from **supervised learning** where the response variable is typically a **category**

Category with 2 variables – a **binary classification problem**

More than 2 classes – a **multiclass classification problem**

Linear Classification

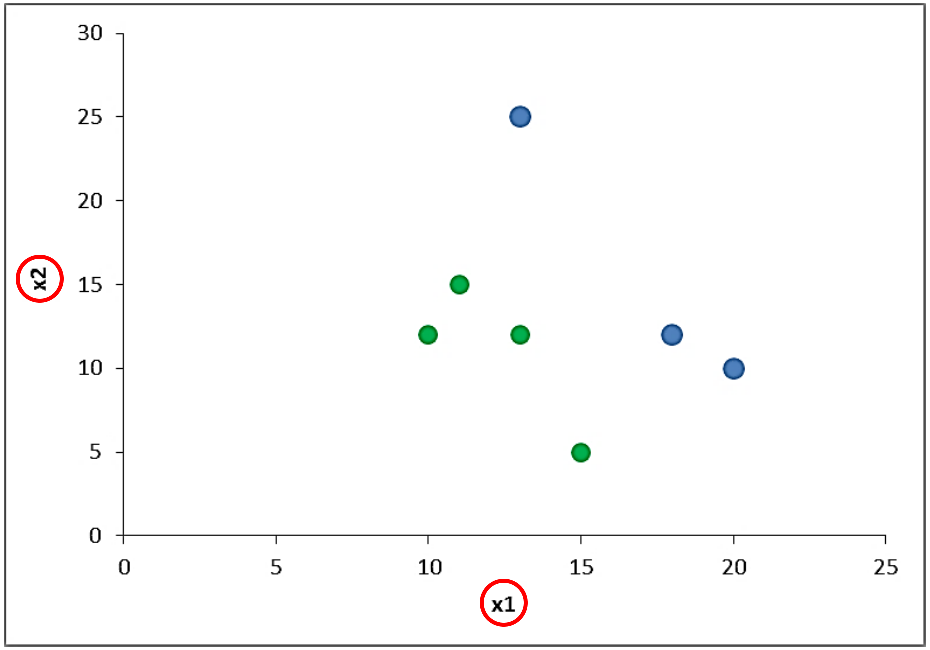
How Does it Work?



This data set has 2 features **x1** and **x2** and a response column (binary variable)

Linear Classification

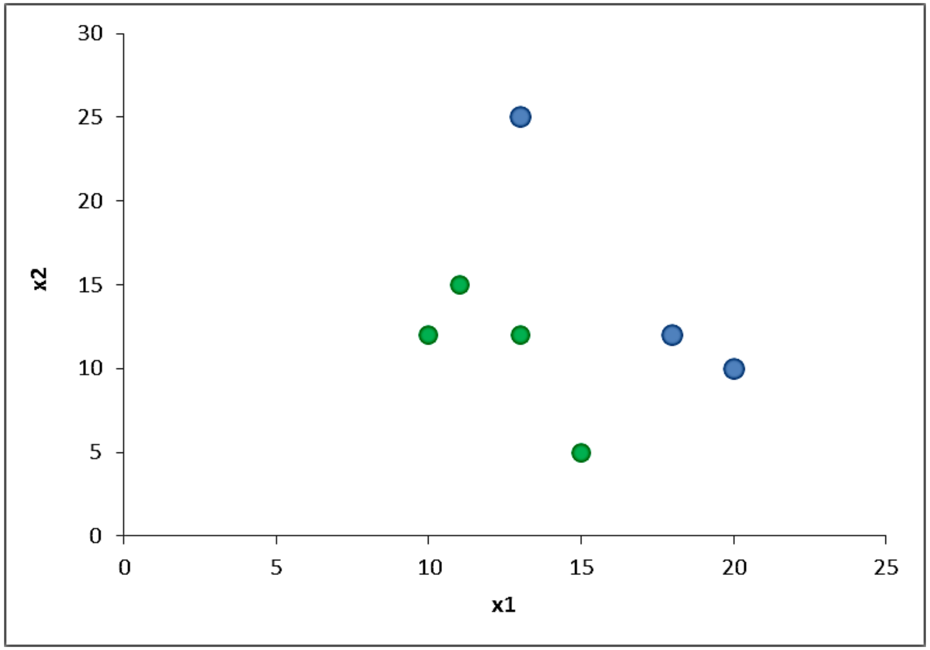
How Does it Work?



Plotting a scatterplot with these 2 variables

Linear Classification

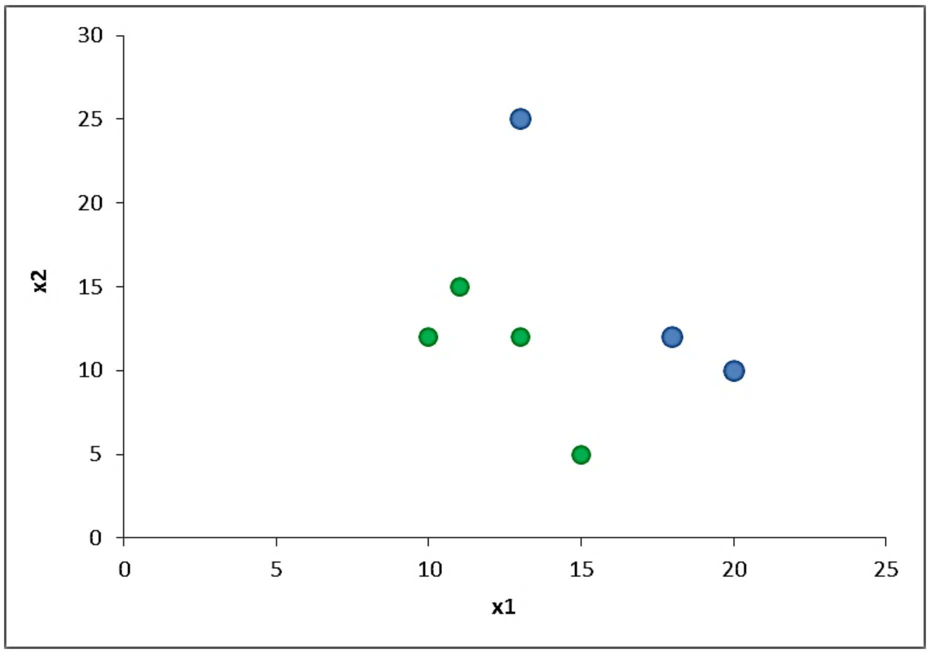
How Does it Work?



Each point is colored green or blue depending on the value of the response

Linear Classification

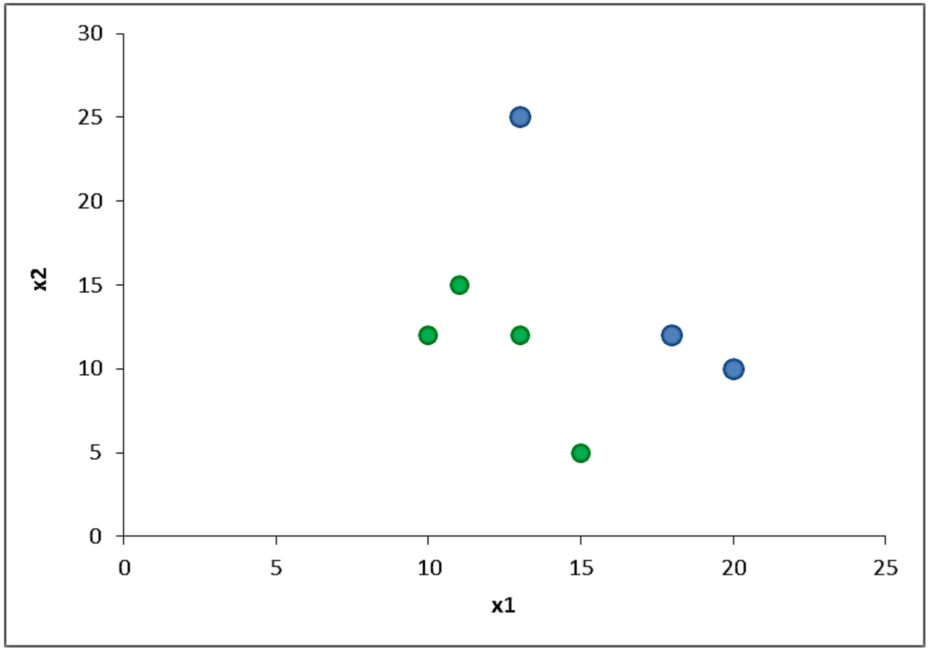
How Does it Work?



Since the response has only 2 categories, one of 2 colors is possible for each point on the scatterplot

Linear Classification

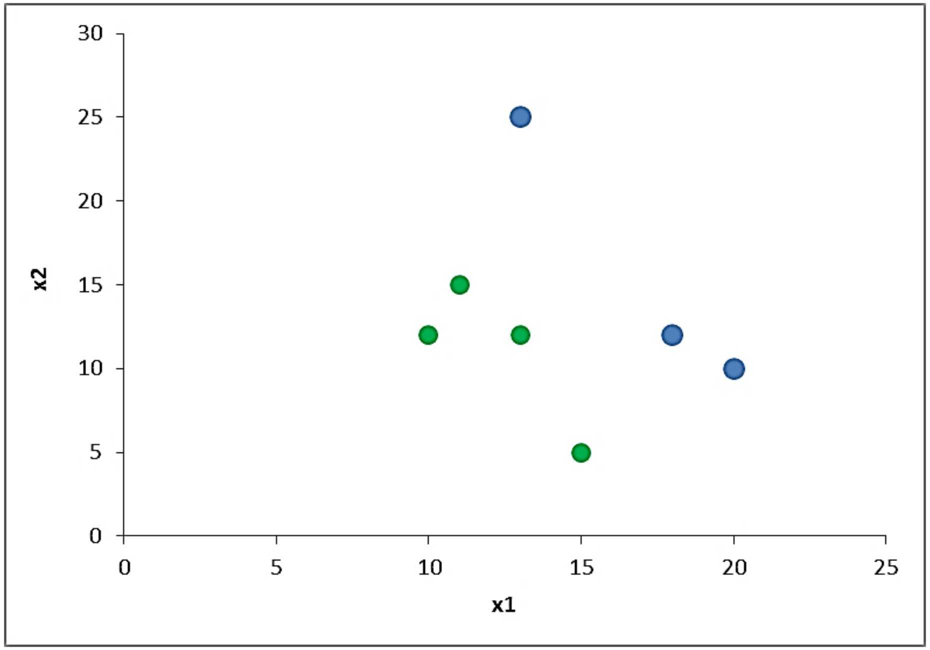
How Does it Work?



Linear classification is simple and intuitive – finding a line that separates the green points from the blue ones

Linear Classification

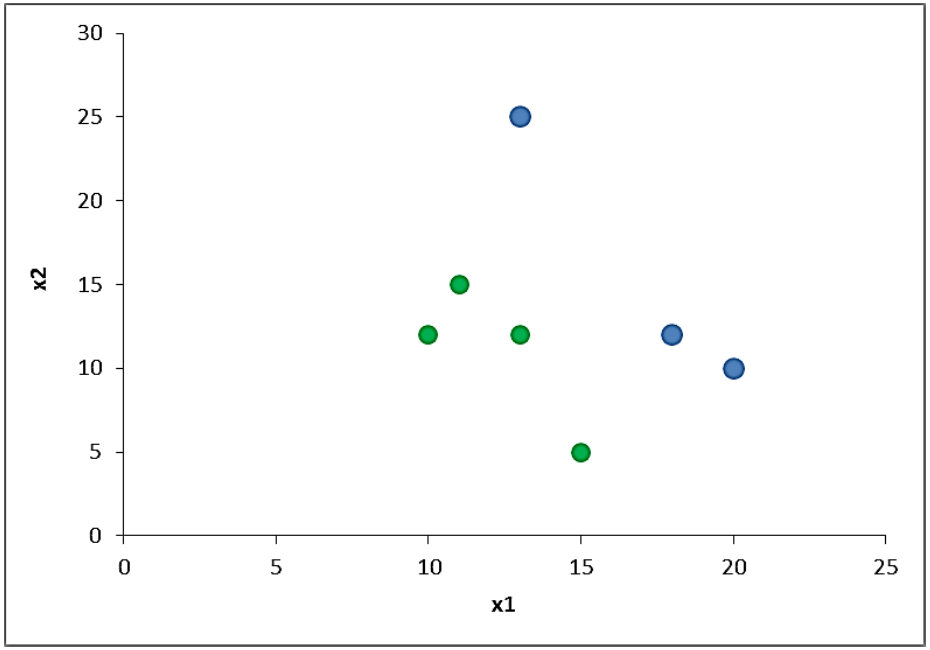
How Does it Work?



Once the line is found, all points on one side of that line will be expected to be **green** and those on the other side **blue**

Linear Classification

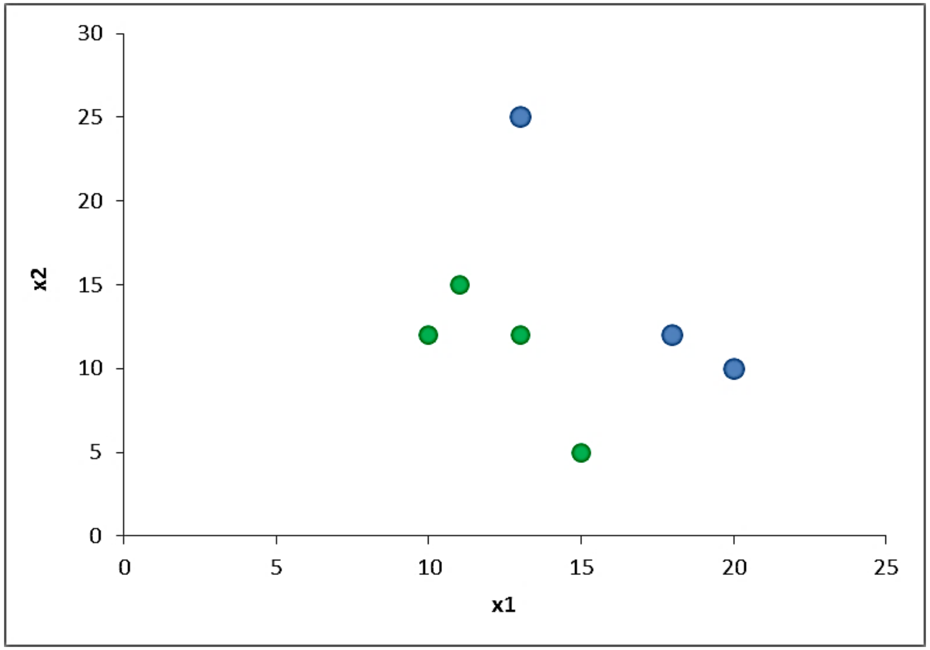
How Does it Work?



Example: All points to the left of the line will be **green** and those to the right will be **blue**

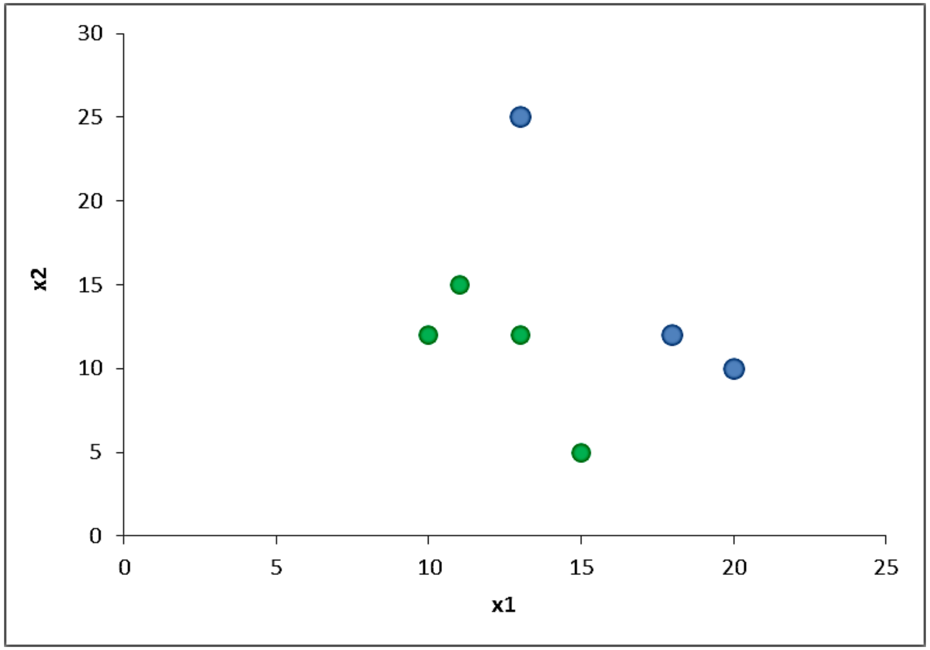
Error for a Classifier

Similar to linear regression, it can be shown that there are infinite such lines possible



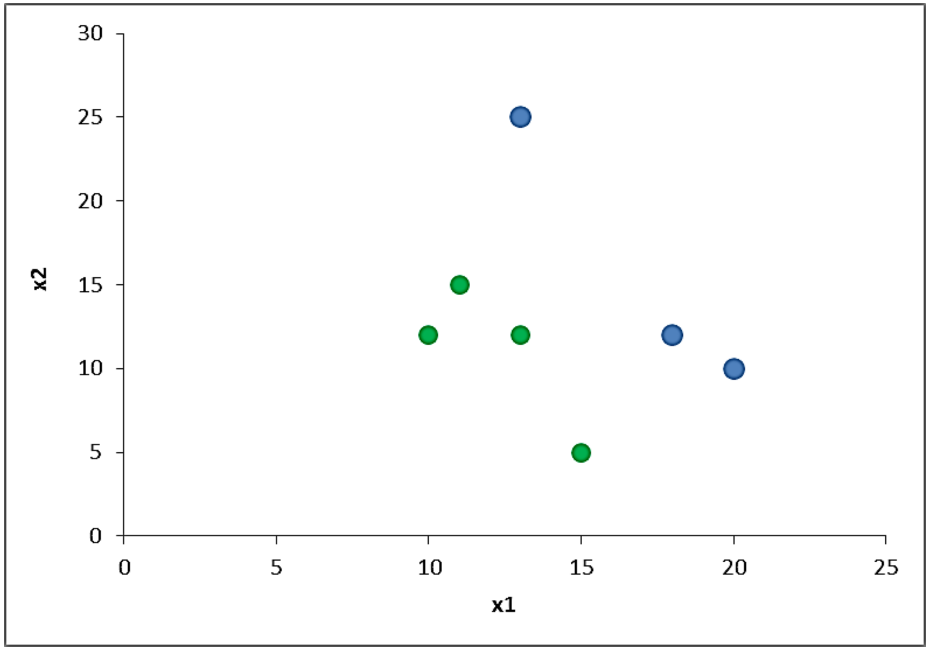
Error for a Classifier

As earlier, all lines will not be equally good

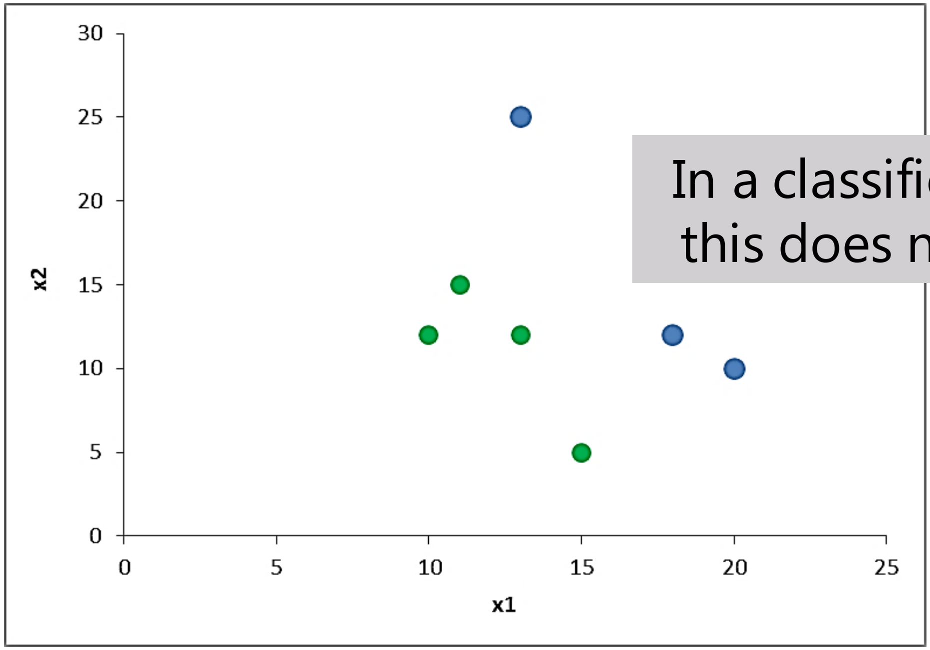


Error for a Classifier

In linear regression, measuring squared error between 2 lines helps in comparing and determining which line better summarizes the data

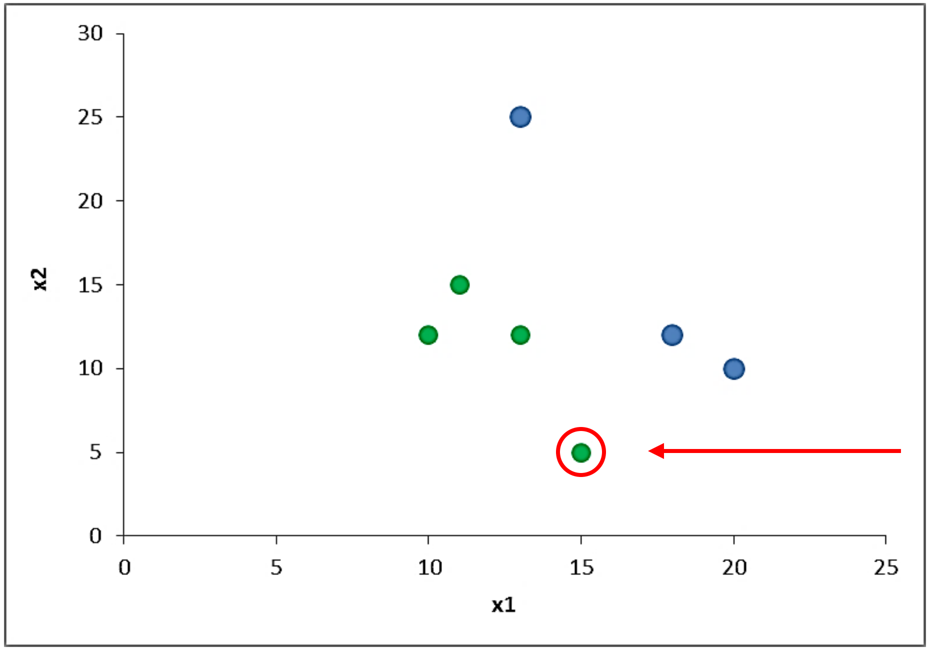


Error for a Classifier



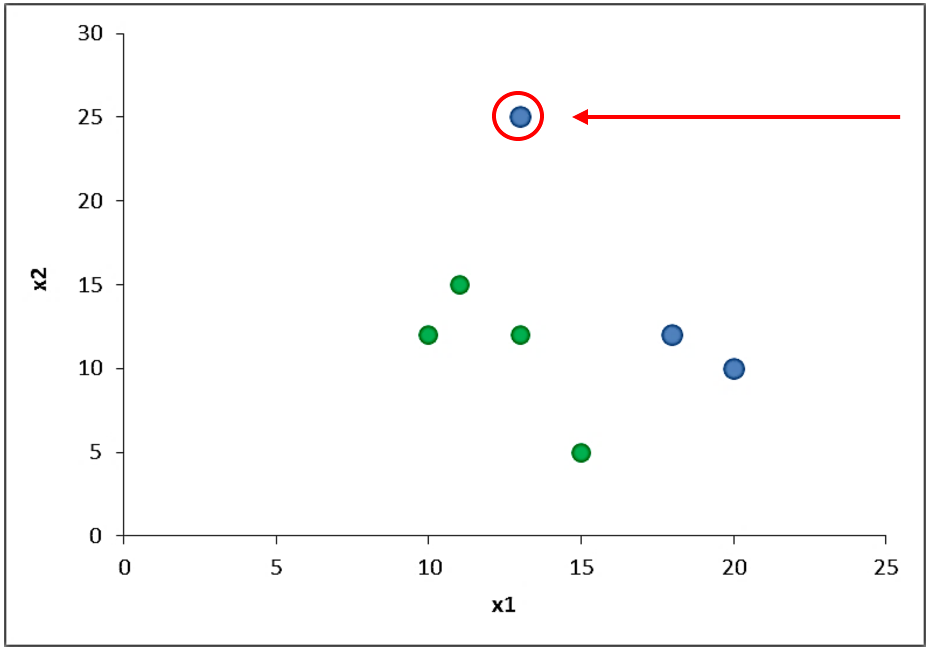
In a classification problem, this does not make sense!

Error for a Classifier



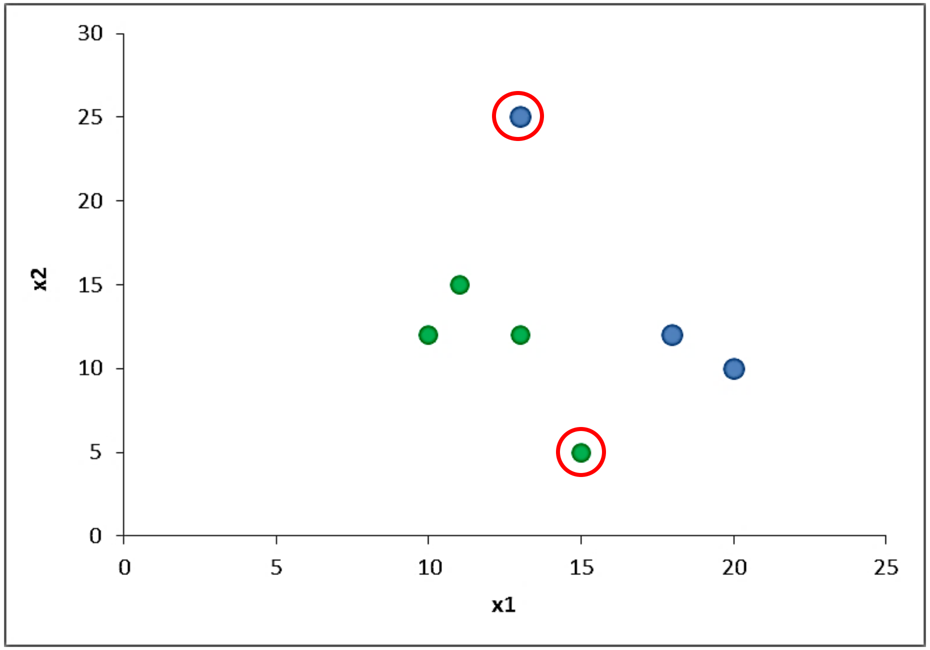
Observed class is **green**

Error for a Classifier



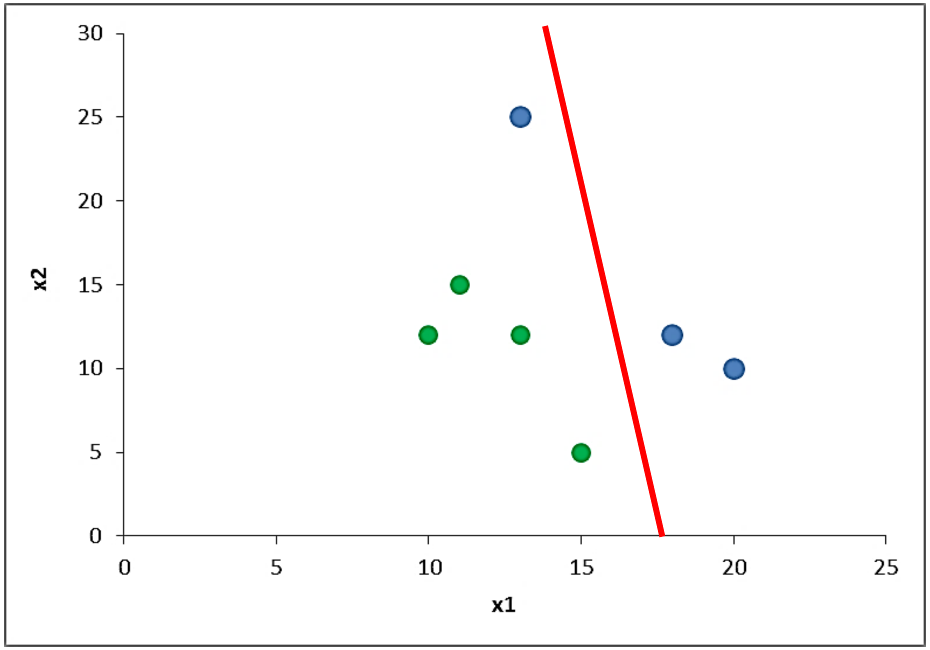
Predicted class is **blue**

Error for a Classifier



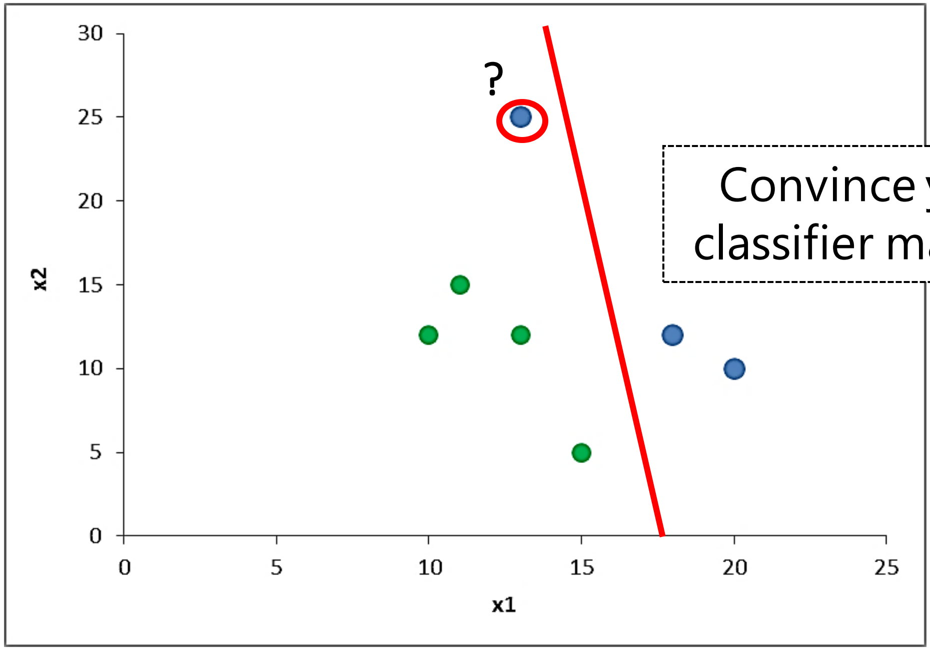
The arithmetic operation, green minus blue, is not valid

Error for a Classifier



This straight line tries to classify the given dataset

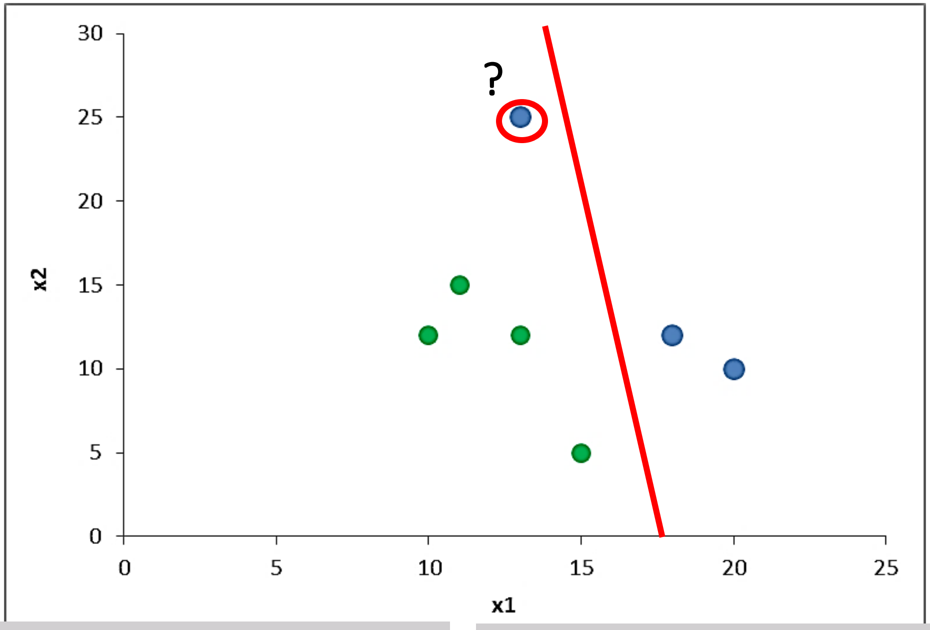
Error for a Classifier



Convince yourself that this classifier makes one mistake!

Error for a Classifier

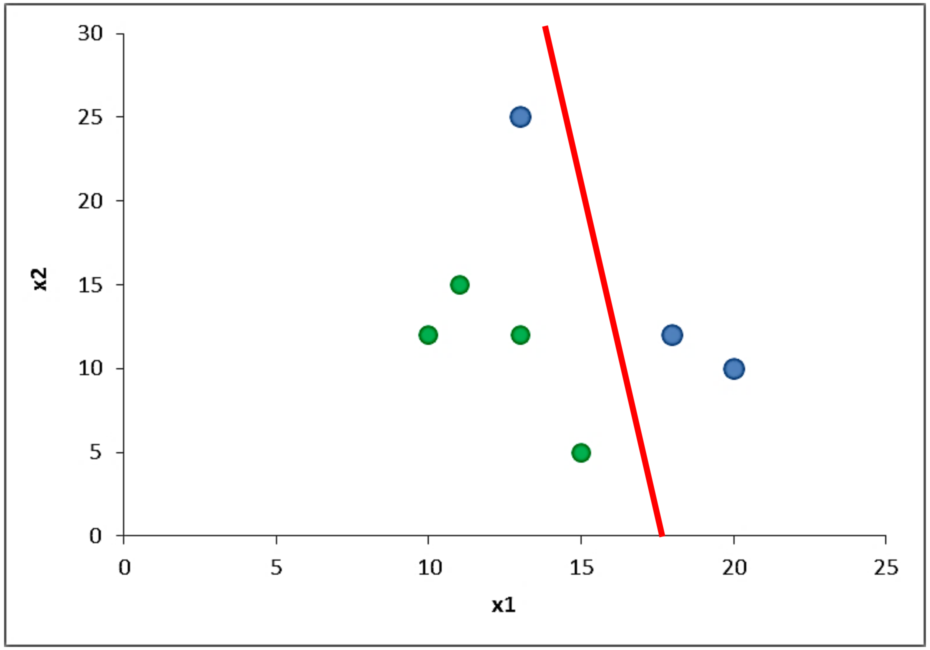
Misclassification rate is most popularly used as an error metric for determining the performance of a classifier



Comparing the predicted values with the expected values over the entire dataset

Computing the average number of cases where the predicted and the observed categories are not equal

Error for a Classifier



There are other measures to quantify the performance of a classifier

Linearity and Non-Linearity

Classification problem – Straight lines that could separate the blue points from the green ones

Linear Models

Regression problem – equation for the function of a straight line

Linearity and Non-Linearity

Things are not always linear!

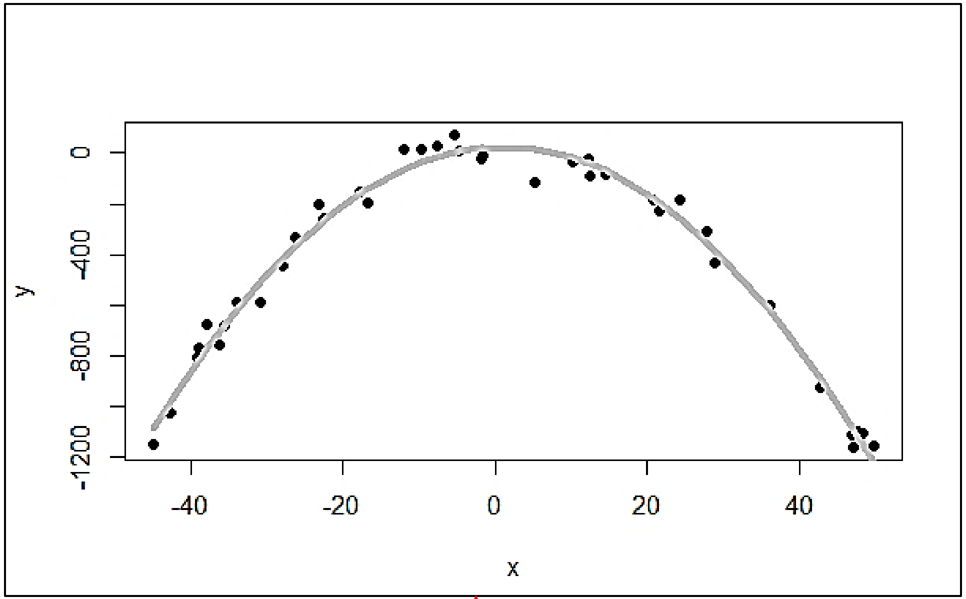
Real life datasets and problems can be highly **non-linear**

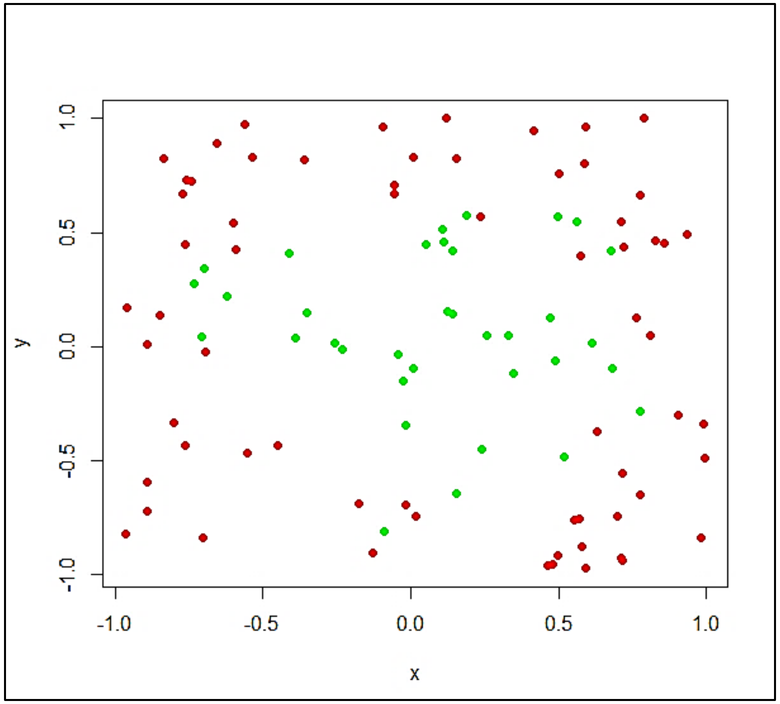
Non-linearity comes in different ways

Example: A feature when plotted versus the response, reveals a non-linear curve

This can be addressed with the existing methods of linear models but requires some feature engineering

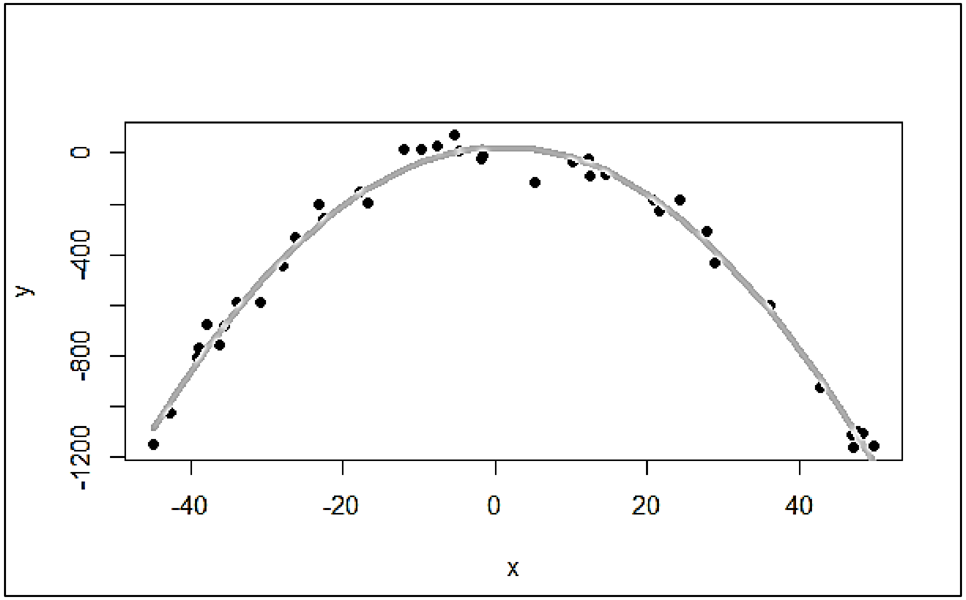
Linearity and Non-Linearity

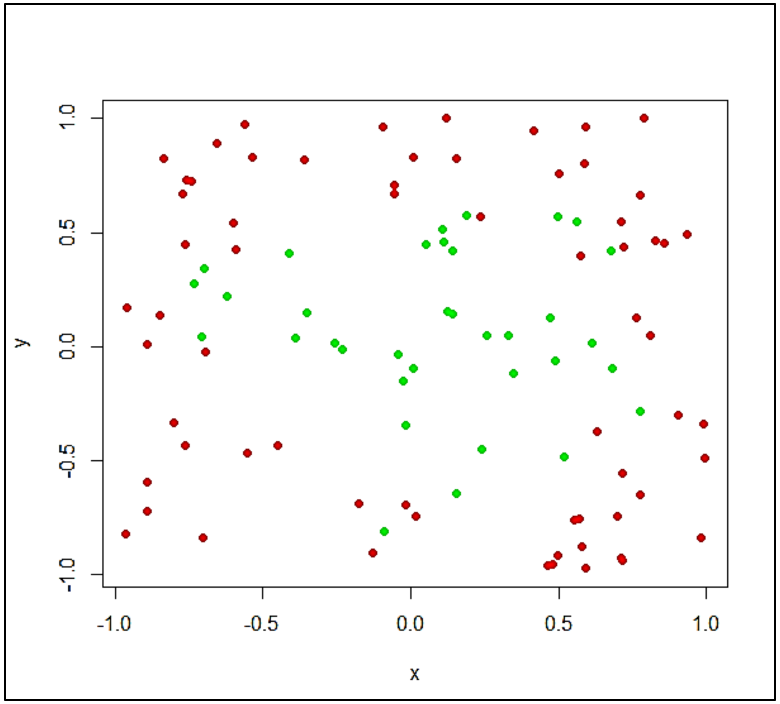




Consider adding square of the feature as a new feature in the dataset

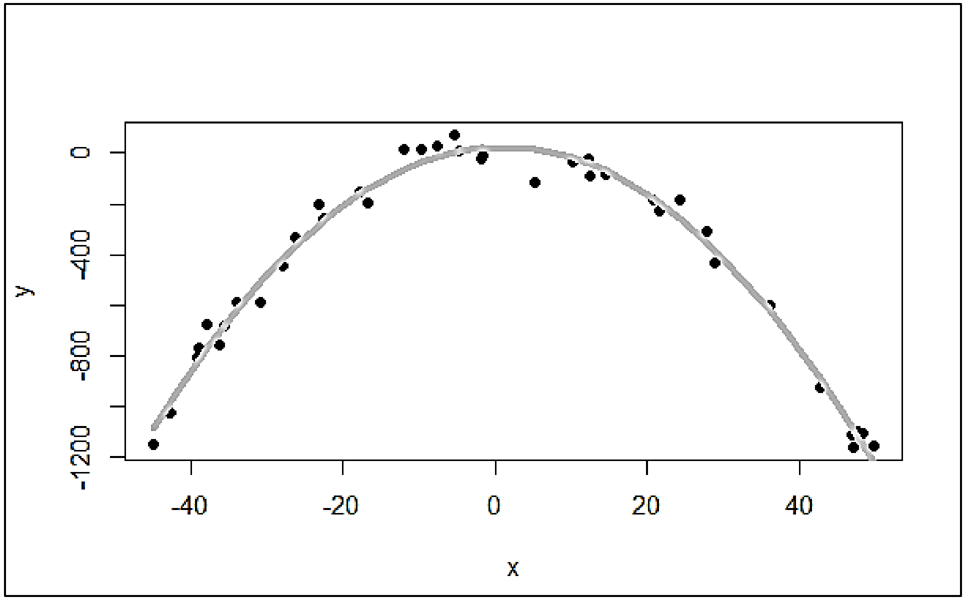
Linearity and Non-Linearity

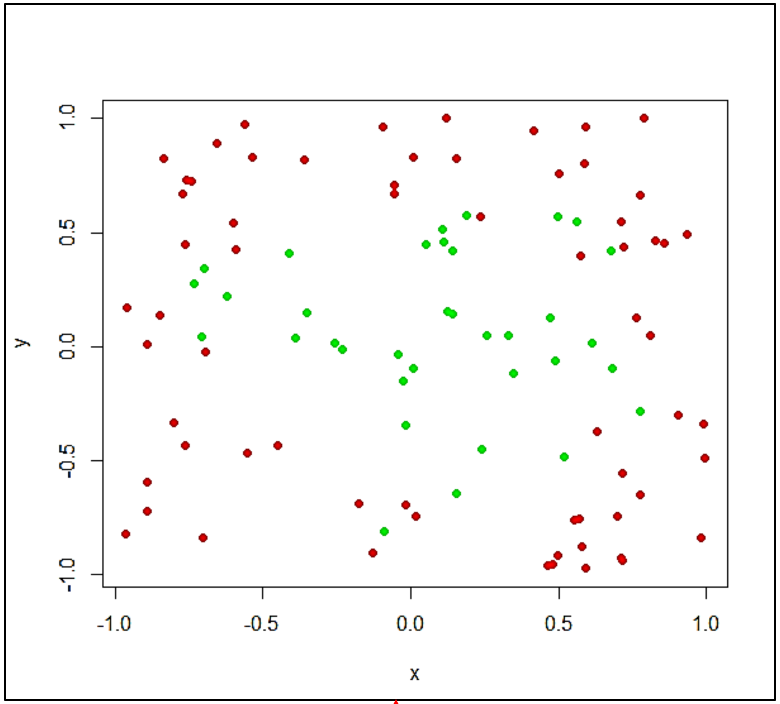




More tricky!

Linearity and Non-Linearity





Linear models still work here, but require special kinds of modification

How Do You Choose a Classifier?

Do you start with a simple linear classifier?

?

Do you produce a highly non-linear classifier using polynomial functions of different features?

How Do You Choose a Classifier?

There is no formula

How Do You Choose a Classifier?

Data Science is as much Art as it is Science

Exploratory analysis

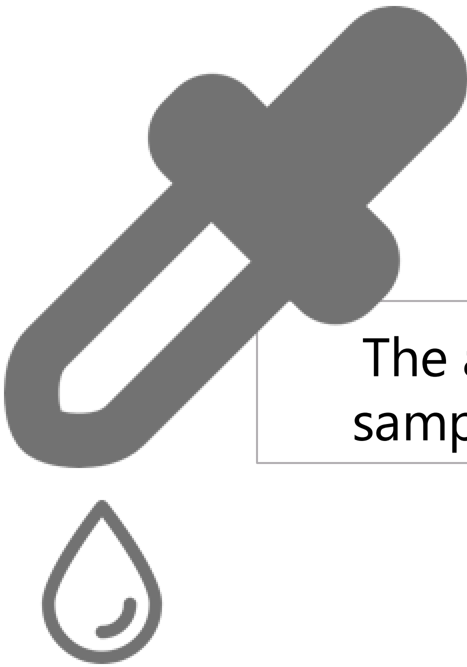
Plot your data in as many ways as possible

Each plot provides a view of the data from different perspectives

Gives an idea of the kind of feature engineering and transformations required

Creativity

Let’s Move On



The algorithms or models are a sample from an infinite universe

Samples and Populations

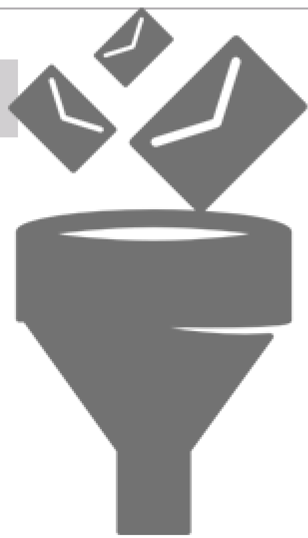
People around the world use e-mails

Each e-mail user receives multiple e-mails per day

This can vary from 1 to 100s, depending on the user

It is not feasible to collect all the data and train the algorithm on the entire data

E-mail spam classification example



Sample of e-mails is used as training dataset

Samples and Populations

Do you think there are only 29000 customers who use a credit card?

No!

Credit default data set example

The data is a **sample** of customers

Samples and Populations

Example: Using a classifier trained on 1000 sample e-mails, would be put to use for a user whose e-mail data was not a part of the sample

Unseen data

The computer sees this **sample** data and tries to best learn the algorithm that can predict a certain response of choice

The real application of these models is done when they are applied to unseen data

The computer or the algorithm has not seen it earlier

Samples and Populations

Unseen data

Machine learning models to be valid and usable in real life, have to perform well

Generalization

It is the ability of a machine learning model to perform well on unseen data

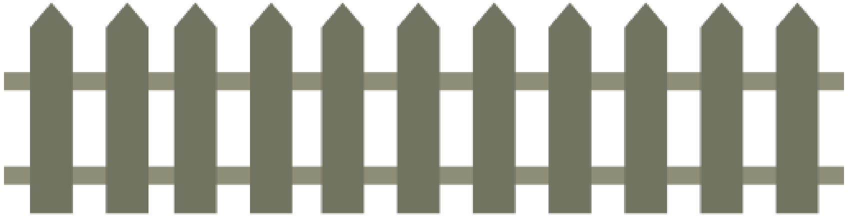
It is a highly important criteria for a model to be deployed in real life

Generalization

Example: Setting thresholds

Only those models will be deployed on production systems if a generalization metric defined on the model is above the threshold value

Depending on how good the model is required to be, this threshold can be low or very high



Generalization

Keep this principle in mind when choosing what kind of functional form you want your regression model or classification model to take

Browse through the materials on Bias Variance Tradeoff available on the internet

Bias Variance Tradeoff

Simpler models tend to generalize well

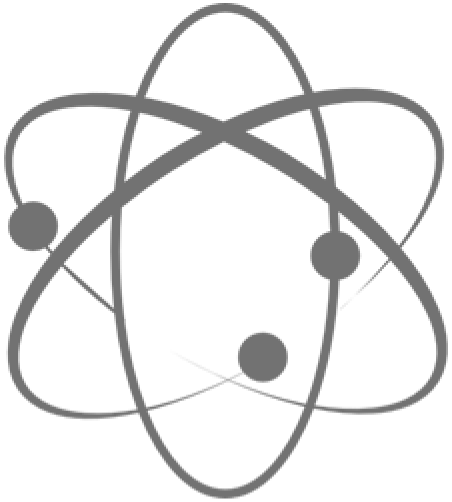
The more complicated the model, a greater chance of it performing poorly on unseen data even though it might perform almost perfectly on the training data

Generalization

Building Models

Each available method comes with its own theoretical implications

As a beginner, do not get bogged down by the weight of mathematical theory



Generalization

Building Models

Break up the original data

Train | Test | Validate

This can be done in **80-10-10** or **70-20-10**

Ensure that the majority is spent on **training** the algorithm

Generalization

Building Models

1 Have a list of models to work with

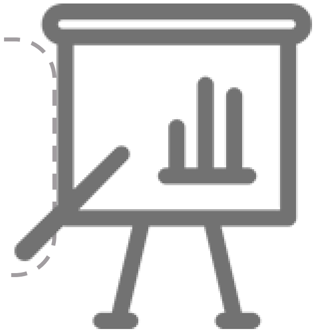
Typically, outputs from an initial round of exploratory analysis

Generalization

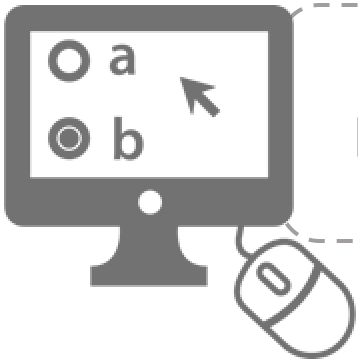
Building Models

2

Estimate model parameters on **train** for each of the candidate models



Select the model that has the lowest error



For each of these models, get the prediction on the **testing** split and compute the errors

3

This is the **final model**

Generalization

Building Models

④

Once the final model is selected, predict on the **validate** split and get the error

This error measure computed on the **validate** split is based on data that the final model has never seen

It was trained on the **train** split and compared with other models on the **test** split

The data in **validate** split is completely new for the model

A performance measure on this **split** can be thought of as a proxy for real world performance for the model

Generalization

Building Models

This is a heuristic rather than a rigorously defined approach

There are other ways to find out the error measures such that they reflect performance on unseen data

Example: K-Fold Cross Validation

Recap

Classification Problems and the Concept of Generalization

Linear Classification

Error for a Classifier

Linearity and Non-Linearity

Choosing a Classifier

Samples and Populations

Generalization

Next

K-Nearest Neighbor and Summing Up the End-to-End Workflow