CLASSIFICATION

Courtesy:
Jiawei Han, Micheline Kamber, and Jian Pei
Md Main UddinRony
Prof. Pushpak Bhattacharyya and Aditya M Joshi
Anjali Jivani

CLASSIFICATION EXAMPLE

- An emergency room in a hospital measures 17 variables (e.g., blood pressure, age, etc) of newly admitted patients.
- A decision is needed: whether to put a new patient in an intensive-care unit.
- Due to the high cost of ICU, those patients who may survive less than a month are given higher priority.
- Problem: to predict high-risk patients and discriminate them from low-risk patients.

CLASSIFICATION EXAMPLE

- A credit card company receives thousands of applications for new cards. Each application contains information about an applicant,
  - age
  - Marital status
  - annual salary
  - outstanding debts
  - credit rating
  - etc.
- Problem: to decide whether an application should approved, or to classify applications into two categories, approved and not approved.

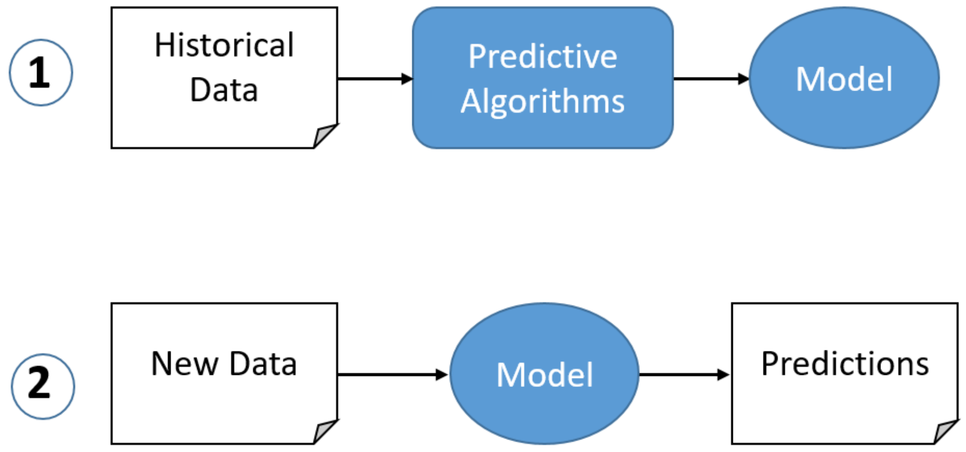
WHAT IS CLASSIFICATION

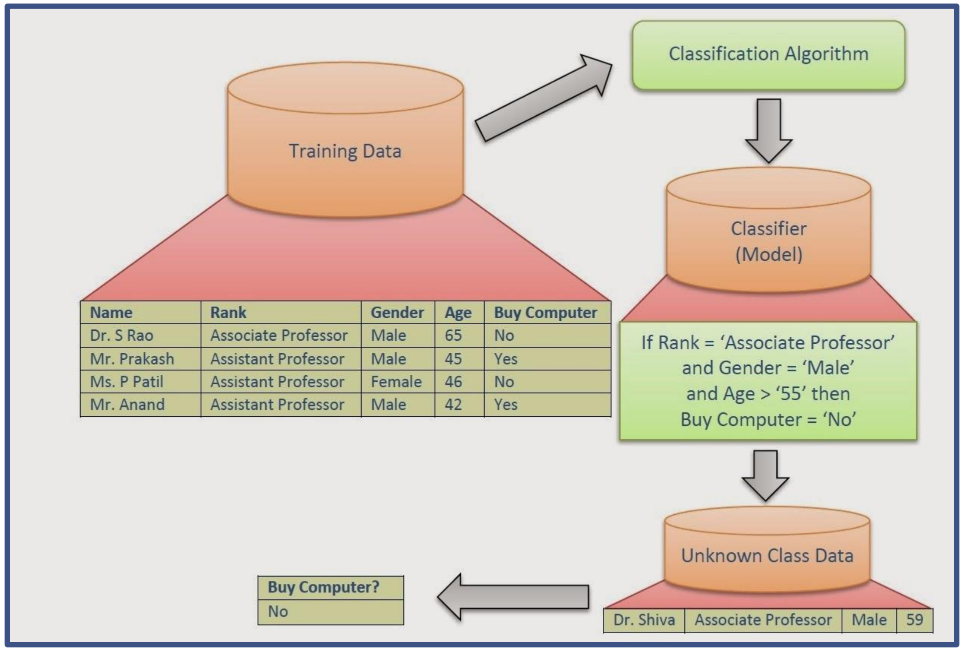
- **Classification**: It is a supervised learning technique in which a model is created based on the historical data with pre-defined labels and used for predicting the classes/labels of unseen data.
- Also known as **Categorization**, is a data mining task that can be used for extracting models describing important classes or to predict future data trends. There are two forms –
  1. Classification
  2. Prediction
- **Classification** models predict categorical class labels. For example, we can build a classification model to categorize bank loan applications as either safe or risky.
- **Prediction** models continuous values; predict the expenditures in dollars of potential customers on computer equipment given their income and occupation.

PHASES OF CLASSIFICATION

There are two phases in the Classification process:

1. **Learning step (training phase)**: construction of classification model using different algorithms to build a classifier by making the model learn using the training set available. The model has to be trained for the prediction of accurate results.
2. **Classification step**: model used to predict class labels and testing the constructed model on test data and hence estimate the accuracy of the classification rules.





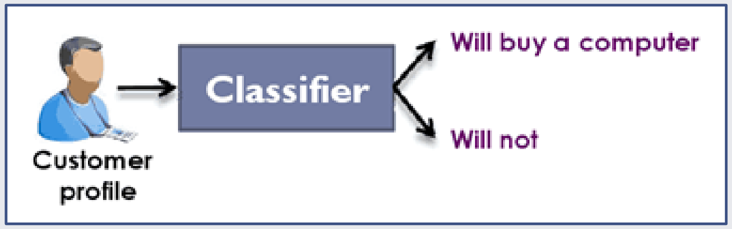
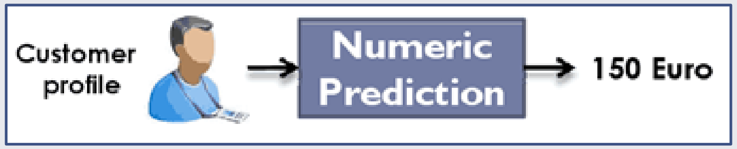
Name	Rank	Gender	Age	Buy Computer
Dr. S Rao	Associate Professor	Male	65	No
Mr. Prakash	Assistant Professor	Male	45	Yes
Ms. P Patil	Assistant Professor	Female	46	No
Mr. Anand	Assistant Professor	Male	42	Yes

If Rank = 'Associate Professor' and Gender = 'Male' and Age > '55' then Buy Computer = 'No'

Dr. Shiva	Associate Professor	Male	59

Buy Computer?
No

CLASSIFICATION *vs.* PREDICTION

CLASSIFICATION	PREDICTION
Predicts categorical class labels (discrete or nominal) e.g. 0 or 1, yes or no.	Models continuous valued functions i.e. predicts unknown or missing values. e.g. price of a house, rate of currency.
Categorical prediction.	Numeric prediction.
Logistic regression	Linear regression
Will a prospective customer buy a computer?  Customer profile Classifier Will buy a computer Will not	How much would a customer spend during a sale? 

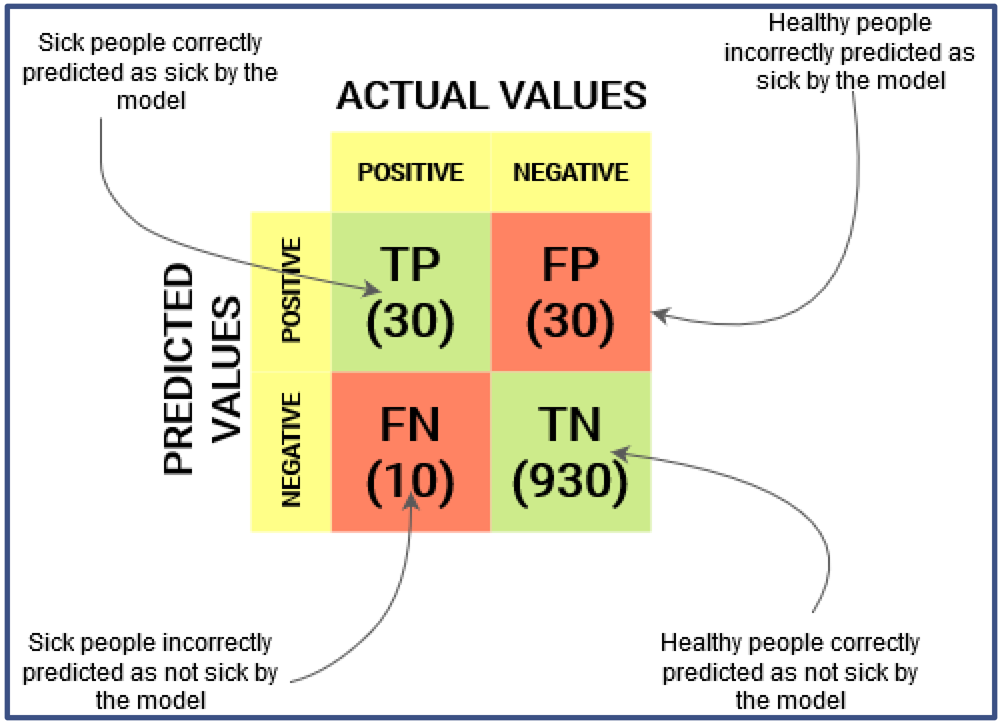
MULTI-LABEL AND MULTI-CLASS

- **Multi-label Classification**
  - One object can belong to more than one class
  - Multi-label classification originated from the investigation of text categorisation problem, where each document may belong to several predefined topics simultaneously.
- **Multi-class Classification**
  - Classification task with more than two classes. Each sample can only be labelled as one class.
  - For example, classification using features extracted from a set of images of fruit, where each image may either be of an orange, an apple, or a pear. Each image is one sample and is labelled as one of the 3 possible classes. Multiclass classification makes the assumption that each sample is assigned to one and only one label - one sample cannot, for example, be both a pear and an apple.
  - Multi-class can be converted to binary classification

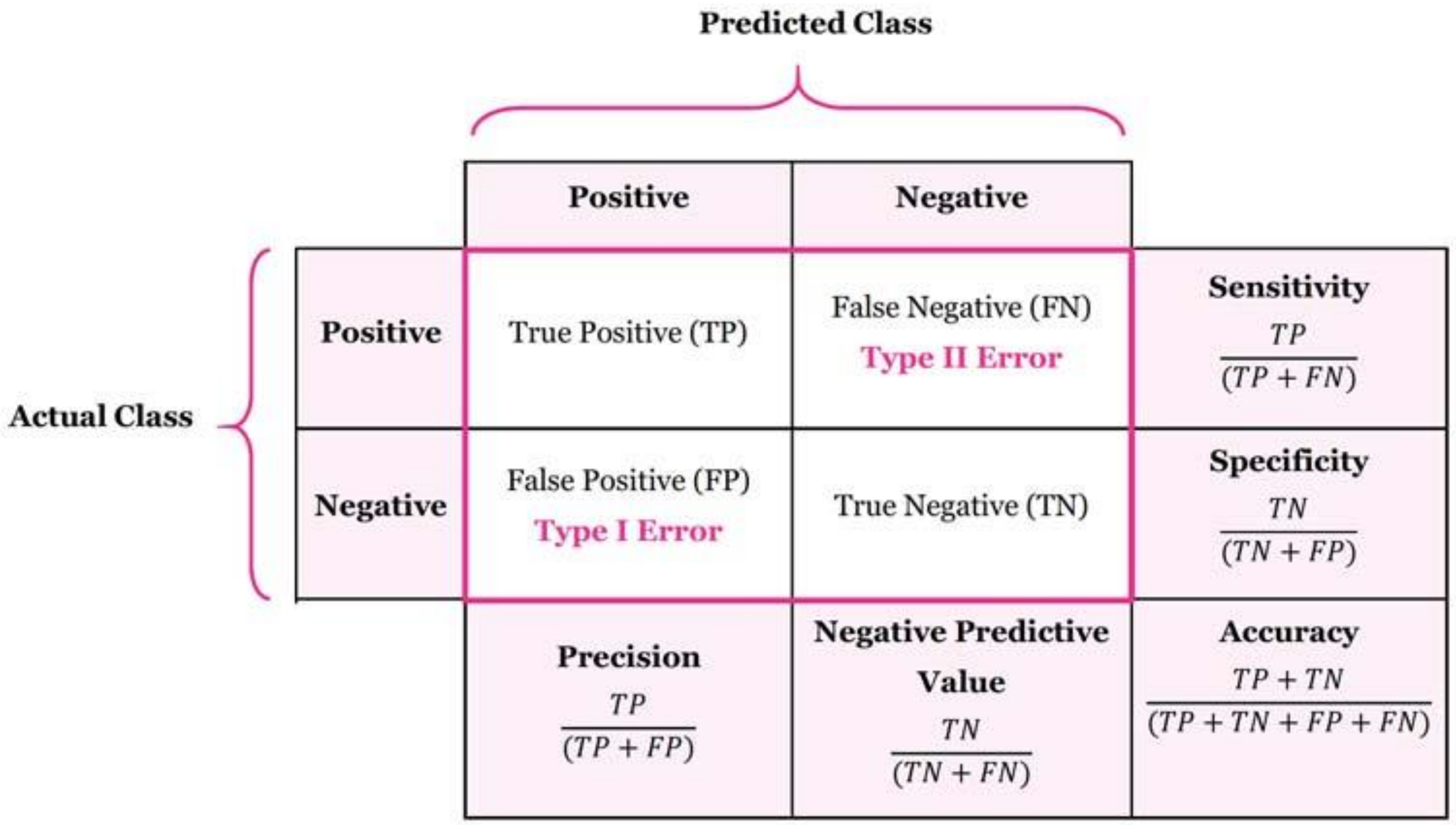
TRAINING AND TESTING DATA

Training Data:
1. Has pre-defined labels.
2. It is divided into two parts – training and testing or validating data.
3. Model created on training data and tested against the validating data whose labels are temporarily removed.
4. Confusion matrix is created to find the predicted labels and their validity.
5. Different measures available to find efficiency of model / classifier.

Testing Data:
1. Has no labels
2. Classifier predicts the labels


Sick people correctly predicted as sick by the model
ACTUAL VALUES
Healthy people incorrectly predicted as sick by the model
POSITIVE
NEGATIVE
PREDICTED VALUES
POSITIVE
NEGATIVE
TP (30)
FP (30)
FN (10)
TN (930)
Sick people incorrectly predicted as not sick by the model
Healthy people correctly predicted as not sick by the model

CONFUSION MATRIX

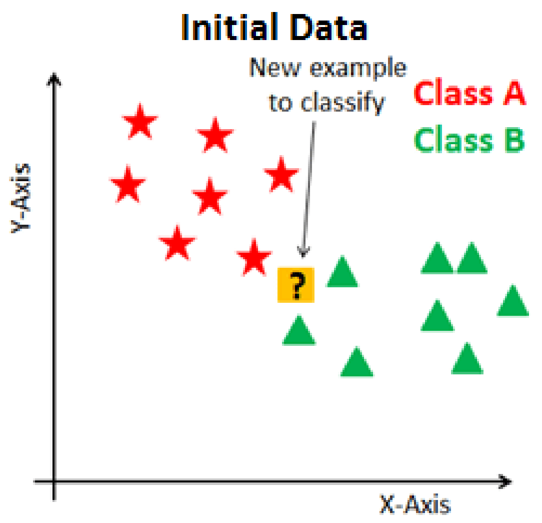


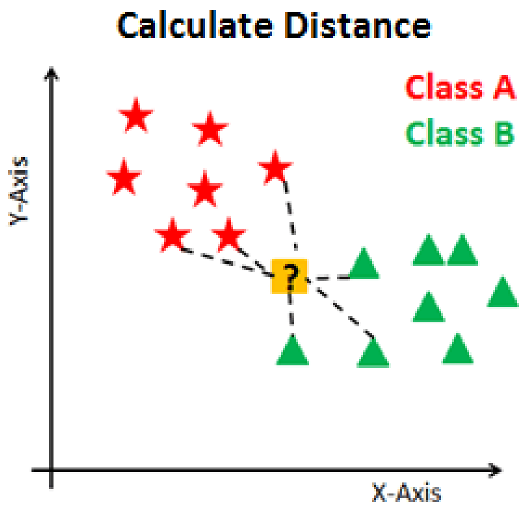
		Predicted Class: Positive	Predicted Class: Negative	
Actual Class	Positive	True Positive (TP)	False Negative (FN) **Type II Error**	**Sensitivity** $\frac{TP}{(TP+FN)}$
	Negative	False Positive (FP) **Type I Error**	True Negative (TN)	**Specificity** $\frac{TN}{(TN+FP)}$
		Precision $\frac{TP}{(TP+FP)}$	**Negative Predictive Value** $\frac{TN}{(TN+FN)}$	**Accuracy** $\frac{TP+TN}{(TP+TN+FP+FN)}$

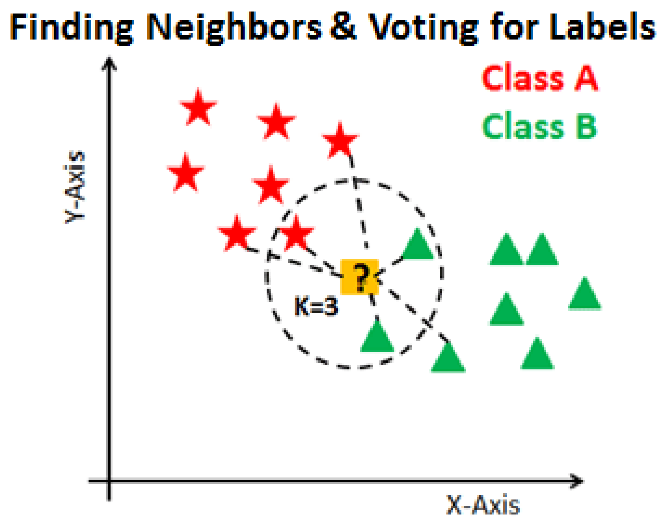
THE K NEAREST NEIGHBOUR

- K nearest neighbours is a simple algorithm that stores all available cases and classifies new cases based on a similarity measure (e.g., distance functions)
- KNN has been used in statistical estimation and pattern recognition already in the beginning of 1970's
- A case is classified by a majority vote of its neighbours, with the case being assigned to the class most common amongst its K nearest neighbours measured by a distance function
- It is a Lazy Learner Algorithm

KNN


Initial Data
New example to classify
Class A
Class B
?
Y-Axis
X-Axis

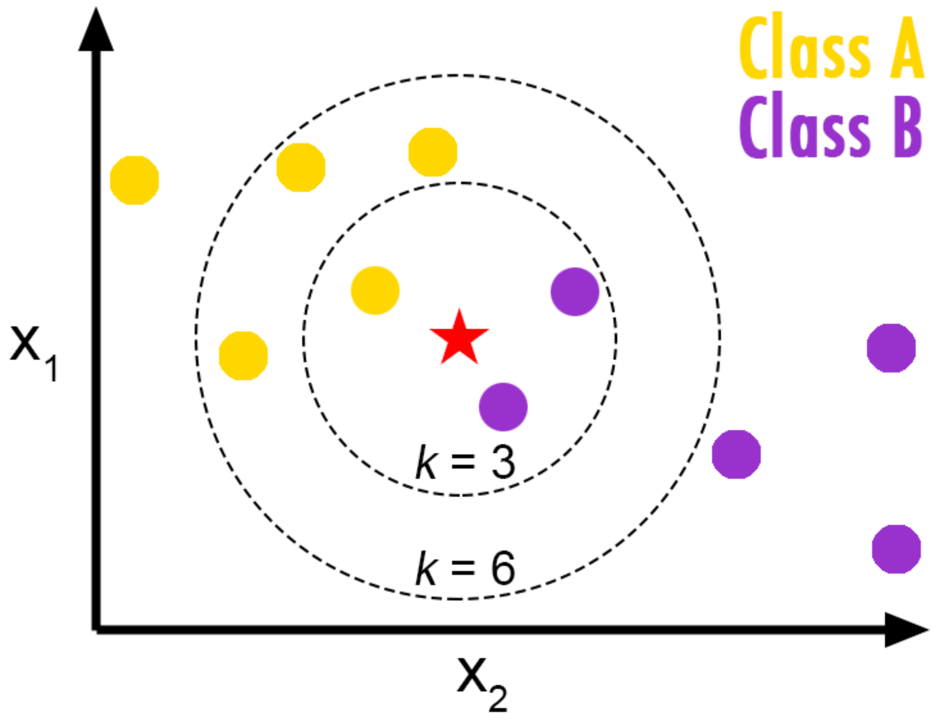

Calculate Distance
Class A
Class B
?
Y-Axis
X-Axis


Finding Neighbors & Voting for Labels
Class A
Class B
?
K=3
Y-Axis
X-Axis

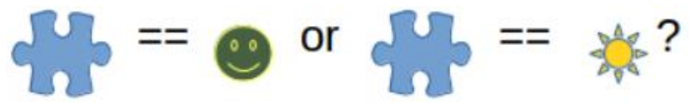
KNN – THE LAZY LEARNER

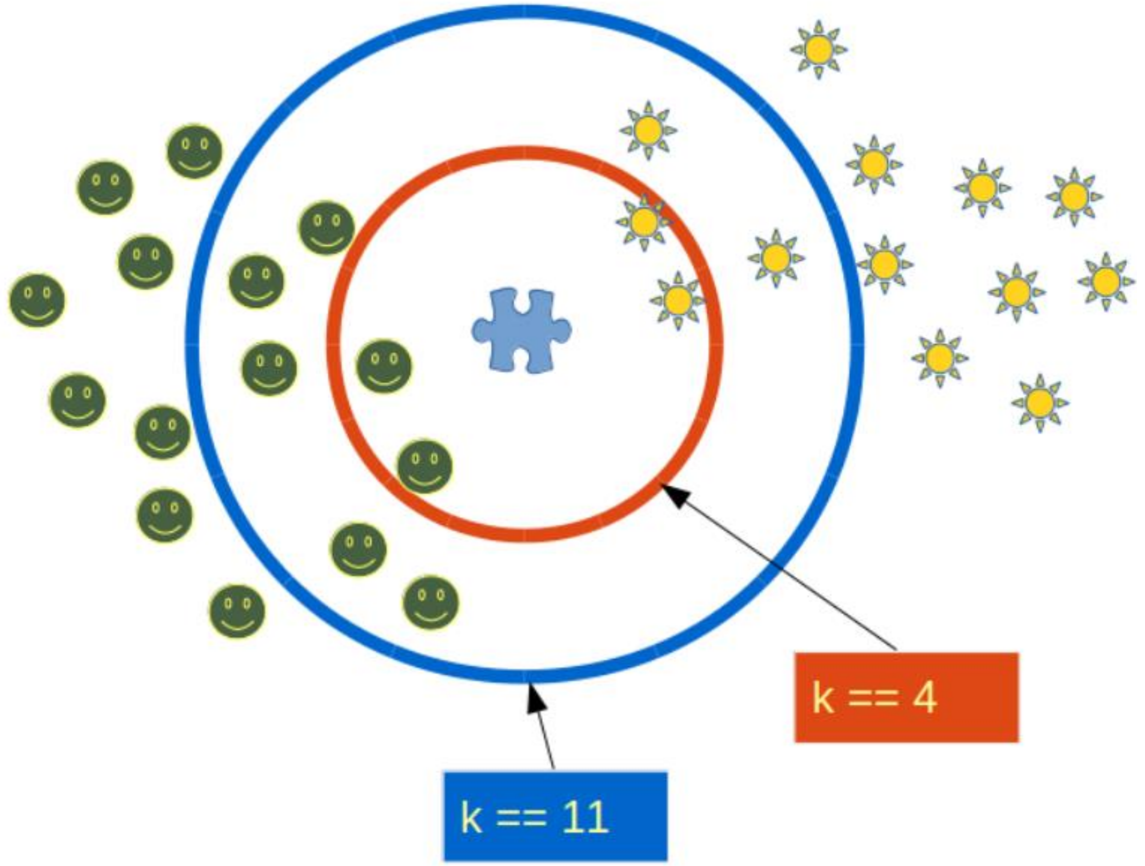
- ‘**Lazy**’: Do not create a model of the training instances in advance.
- When an instance arrives for testing, runs the algorithm to get the class prediction

“One is known by the company one keeps”



KNN





THE K NEAREST NEIGHBOUR

- **What is k**?
  - Value of k determines number of closest neighbours to consider.
  - Majority of vote is commonly used, so the label associated with the majority of the neighbours is used as the label of the new sample.
  - Breaking rule; ex: The label of the closer neighbour is used or the label is chosen randomly among the neighbours.
- **Distance measure:**
  - Need measure to determine 'Closeness' (Distance between sample).
  - Distance measures that can be used:
    - Euclidean
    - Manhattan
    - Cosine
    - …

THE K NEAREST NEIGHBOUR

- **Advantages:**
  - No separate training phase.
  - No separate part where a model is constructed and its parameter is adjusted.
  - Can generate complex decision boundaries.
  - Effective if training data is large.
- **Disadvantages:**
  - Can be slow:
    - Distance between new sample and all samples must be computed to classify new sample.
  - Need to determine value of parameter k
    - If k is too small, then the result can be sensitive to noise points.
    - If k is too large, then the neighbourhood may include too many points from other classes.
  - The choice of the distance measure: Some distance measures can also be affected by the high dimensionality of the data. Attributes may have to be scaled to prevent distance measures from being dominated by one of the attributes.
- These techniques, which are particularly applicable for low dimensional data, can help reduce the computational cost without affecting classification accuracy.
- KNN is particularly well suited for multi-modal classes as well as applications in which an object can have many class labels.

HOW TO DETERMINE K?

- There is no structured method to find the best value for "K". We need to find out with various values by trial and error and assuming that training data is unknown.
- Choosing smaller values for K can be noisy and will have a higher influence on the result.
- Larger values of K will have smoother decision boundaries which mean lower variance but increased bias. Also, computationally expensive.
- In general, practice, choosing the value of **k** is,

  k = sqrt(N), where **N** stands for the **number of samples in your training dataset**.
- Try and keep the value of k odd in order to avoid confusion between two classes of data

KNN EXAMPLE

Customer	Age	Income (K)	No. of cards	Response
John	35	35	3	Yes
Rachel	22	50	2	No
Ruth	63	200	1	No
Tom	59	170	1	No
Neil	25	40	4	Yes
David	**37**	**50**	**2**	?

KNN EXAMPLE

Use L2-norm (Euclidean distance)

Customer	Age	Income (K)	No. of cards	Response	Distance from David
John	35	35	3	Yes	sqrt $[(35-37)^2+(35-50)^2+(3-2)^2]$=**15.16**
Rachel	22	50	2	No	sqrt $[(22-37)^2+(50-50)^2+(2-2)^2]$=**15**
Ruth	63	200	1	No	sqrt $[(63-37)^2+(200-50)^2+(1-2)^2]$=**152.23**
Tom	59	170	1	No	sqrt $[(59-37)^2+(170-50)^2+(1-2)^2]$=**122**
Neil	25	40	4	Yes	sqrt $[(25-37)^2+(40-50)^2+(4-2)^2]$=**15.74**

David	**37**	**50**	**2**	**Yes**

NAÏVE BAYES CLASSIFICATION

- Works based on Bayes' theorem
- Why its is called Naïve?
  - Because it assumes that the presence of a particular feature in a class is unrelated to the presence of any other feature

Bayes Theorem:

- The theorem can be stated mathematically as follow:

$$P(A \mid B) = \frac{P(B \mid A)\,P(A)}{P(B)},$$

- P(A) and P(B) are the probabilities of observing A and B without regard to each other. Also known as **Prior Probability.**
- P(A | B), a conditional (**Posterior**) probability, is the probability of observing event A given that B is true.
- P(B | A) is the conditional (**Posterior**)probability of observing event B given that A is true.

BAYES THEOREM

P(A) = (no. of favourable events) / n,

n – total no. of events

P(A/B) is the conditional probability that given B has occurred what is the probability that A will occur.

P(A/B) = P(A∩B) / P(B) --------(1)

P(B/A) = P(A∩B) / P(A) --------(2)

From (1),

P(A∩B) = P(A/B) P(B) --------(3)

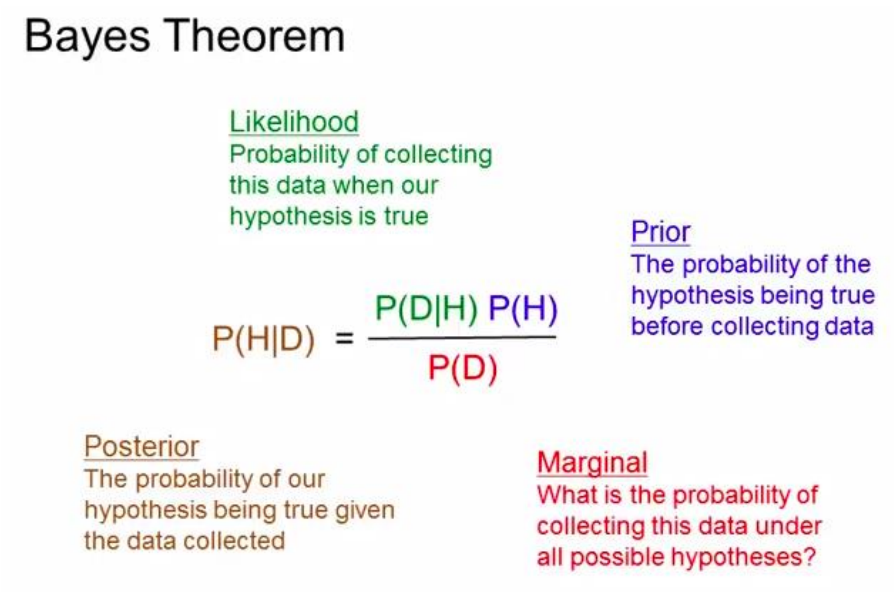
From (2)

P(A∩B) = P(B/A) P(A) --------(4)

From (3) and (4),

P(A/B) = P(B/A) P(A) / P(B) --------(Bayes Theorem)

A and B are independent events



Bayes Theorem

Likelihood
Probability of collecting this data when our hypothesis is true

Prior
The probability of the hypothesis being true before collecting data

$$P(H|D) = \frac{P(D|H)\,P(H)}{P(D)}$$

Posterior
The probability of our hypothesis being true given the data collected

Marginal
What is the probability of collecting this data under all possible hypotheses?

BAYES THEOREM

LIKELIHOOD
The probability of "B" being True, given "A" is True

PRIOR
The probability "A" being True. This is the knowledge.

$$P(A|B) = \frac{P(B|A).P(A)}{P(B)}$$

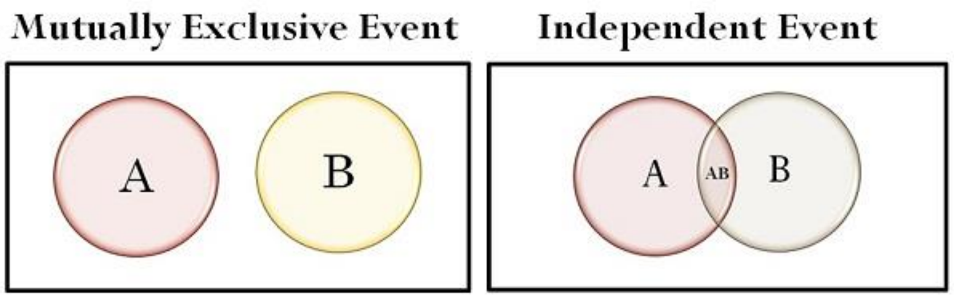
POSTERIOR
The probability of "A" being True, given "B" is True

MARGINALIZATION
The probability "B" being True.

INDEPENDENT EVENTS

BASIS FOR COMPARISON	MUTUALLY EXCLUSIVE EVENTS	INDEPENDENT EVENTS
Meaning	Two events are said to be mutually exclusive, when their occurrence is not simultaneous.	Two events are said to be independent, when the occurrence of one event cannot control the occurrence of other.
Influence	Occurrence of one event will result in the non-occurrence of the other.	Occurrence of one event will have no influence on the occurrence of the other.
Mathematical formula	P(A and B) = 0	P(A and B) = P(A) P(B)
Sets in Venn diagram	Does not overlap	Overlaps

INDEPENDENT EVENTS



The events A and B are independent if,
$P(A \cap B) = P(A) \times P(B)$

$P(A/B) = P(A \cap B) / P(B)$ --------(1)
$P(A/B) = P(A)$, from (1)
$P(B/A) = P(B)$

If A1, A2, A3, … An are independent events then,

$$P(A1 \cap A2 \cap A3 \cap \ldots\ldots An) = P(A1)\, P(A2)\, P(A3) \cdots P(An)$$
$$= \prod_{i=1}^{n} (Ai)$$

THE NAÏVE BAYES UNDERSTANDING

What we want to know; the Posterior probability of Class j given a predictor x

The Likelihood; the probability of the predictor given a Class j. Its computed from the training-set.

The Prior probability of Class j; what we know about the class distribution before we consider x.

$$P(Class_j \mid x) = \frac{P(x \mid Class_j) \times P(Class_j)}{P(x)}$$

The Evidence. In practice, there's interest only in the numerator (denominator is effectively constant)

Applying the independence assumption

$$P(x \mid Class_j) = P(x_1 \mid Class_j) \times P(x_2 \mid Class_j) \times \ldots \times P(x_k \mid Class_j)$$

Substituting the independence assumption, we derive the Posterior probability of Class j given a new instance x' as...

$$P(Class_j \mid x') = P(x'_1 \mid Class_j) \times P(x'_2 \mid Class_j) \times \ldots \times P(x'_k \mid Class_j) \times P(Class_j)$$

THE NAÏVE BAYES UNDERSTANDING

$$P(A \mid B) = \frac{P(B \mid A)\,P(A)}{P(B)},$$

$$\underbrace{P\left(\frac{Ci}{X}\right)}_{\text{Test Data}} = \underbrace{\frac{P\left(\frac{X}{Ci}\right)P(Ci)}{P(X)}}_{\text{Training Data}}$$

- The LHS is the probability that given a test data X, what is the probability that it belongs to class Ci (i = 1..total no. of classes).
- Find probability of test data X belonging to each class. Whichever probability is highest, X belongs to that class. i.e. if it is binary classification, find $P(\frac{C1}{X})$ and $P(\frac{C2}{X})$ and whichever is higher X is assigned to that class.
- In the RHS, the denominator is P(X), which would be same for all the classes. So it can be removed.

THE NAÏVE BAYES EXAMPLE

Given all the previous patient's symptoms and diagnosis, the training set is →

chills	runny nose	headache	fever	flu?
Y	N	Mild	Y	N
Y	Y	No	N	Y
Y	N	Strong	Y	Y
N	Y	Mild	Y	Y
N	N	No	N	N
N	Y	Strong	Y	Y
N	Y	Strong	N	N
Y	Y	Mild	Y	Y

Does the patient with the following symptoms have the flu? →

chills	runny nose	headache	fever	flu?
Y	N	Mild	Y	?

TRAINING SET				
chills	runny nose	headache	fever	flu?
Y	N	Mild	Y	N
Y	Y	No	N	Y
Y	N	Strong	Y	Y
N	Y	Mild	Y	Y
N	N	No	N	N
N	Y	Strong	Y	Y
N	Y	Strong	N	N
Y	Y	Mild	Y	Y

TEST SET (VALIDATION)				
chills	runny nose	headache	fever	flu?
Y	N	Mild	Y	?

Probabilities for class flu = Y	
P(Flu=Y) = 5/8	0.625
P(chills=Y|flu=Y) = 3/5	0.6
P(runny nose=N|flu=Y) = 1/5	0.2
P(headache=Mild|flu=Y) = 2/5	0.4
P(fever=Y|flu=Y) = 4/5	0.8

Probabilities for class flu = N	
P(Flu=N) = 3/8	0.375
P(chills=Y|flu=N) = 1/3	0.333
P(runny nose=N|flu=N) = 2/3	0.666
P(headache=Mild|flu=N) = 1/3	0.333
P(fever=Y|flu=N) = 1/3	0.333

THE NAÏVE BAYES EXAMPLE

Class flu = Y		Class flu = N	
P(Flu=Y)	0.625	P(Flu=N)	0.375
P(chills=Y|flu=Y)	0.6	P(chills=Y|flu=N)	0.333
P(chills=N|flu=Y)	0.4	P(chills=N|flu=N)	0.666
P(runny nose=Y|flu=Y)	0.8	P(runny nose=Y|flu=N)	0.333
P(runny nose=N|flu=Y)	0.2	P(runny nose=N|flu=N)	0.666
P(headache=Mild|flu=Y)	0.4	P(headache=Mild|flu=N)	0.333
P(headache=No|flu=Y)	0.2	P(headache=No|flu=N)	0.333
P(headache=Strong|flu=Y)	0.4	P(headache=Strong|flu=N)	0.333
P(fever=Y|flu=Y)	0.8	P(fever=Y|flu=N)	0.333
P(fever=N|flu=Y)	0.2	P(fever=N|flu=N)	0.666

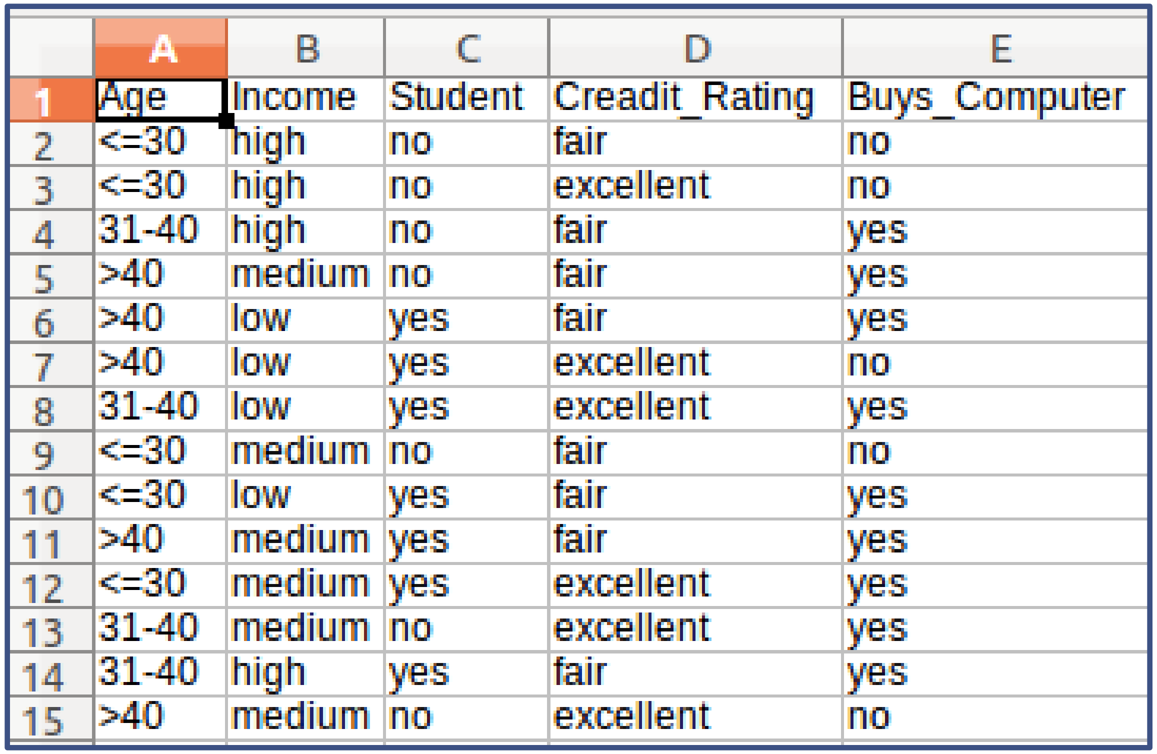
THE NAÏVE BAYES EXAMPLE

$$P(\frac{Ci}{X}) = \frac{P\left(\frac{X}{Ci}\right)P(Ci)}{P(X)}$$

Class flu = Y	Class flu = N
$P\left(\frac{X}{flu = Y}\right)P(flu = Y)$	$P\left(\frac{X}{flu = N}\right)P(flu = N)$
P(chills = Y|flu=Y) * P(runny nose = N|flu=Y) * P(headache = Mild|flu=Y) * P(fever = N|flu=Y) * P(flu=Y)	P(chills = Y|flu=N) * P(runny nose = N|flu=N) * P(headache = Mild|flu=N) * P(fever = N|flu=N) * P(flu=N)
= 0.6 * 0.2 * 0.4 * 0.2 * 0.625 = 0.006	= 0.333 * 0.666 * 0.333 * 0.666 * 0.375 = 0.0184

Class flu = N has higher probability, so the test data belongs to class N.

THE NAÏVE BAYES EXAMPLE



	A	B	C	D	E
1	Age	Income	Student	Creadit_Rating	Buys_Computer
2	<=30	high	no	fair	no
3	<=30	high	no	excellent	no
4	31-40	high	no	fair	yes
5	>40	medium	no	fair	yes
6	>40	low	yes	fair	yes
7	>40	low	yes	excellent	no
8	31-40	low	yes	excellent	yes
9	<=30	medium	no	fair	no
10	<=30	low	yes	fair	yes
11	>40	medium	yes	fair	yes
12	<=30	medium	yes	excellent	yes
13	31-40	medium	no	excellent	yes
14	31-40	high	yes	fair	yes
15	>40	medium	no	excellent	no

Classes:

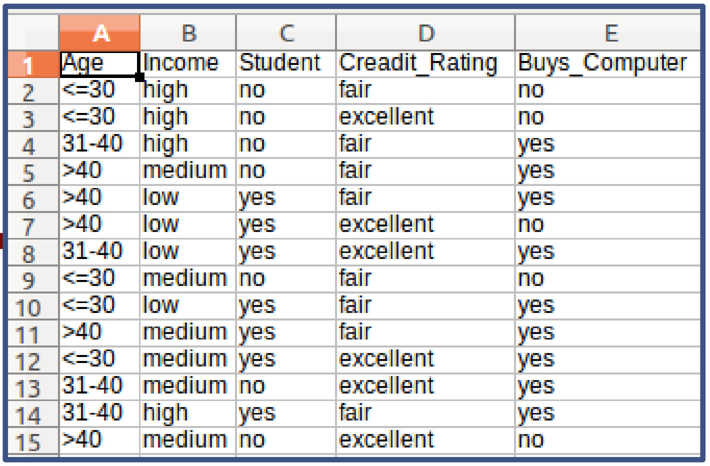
C1:buys_computer = 'yes'

C2:buys_computer = 'no'

Data to be classified:
X = (age <=30, Income = medium, Student = yes, Credit_rating = Fair)

Find whether X buys computer or not.

X = (age <= 30 , income = medium, student = yes, credit_rating = fair)



	A	B	C	D	E
1	Age	Income	Student	Creadit_Rating	Buys_Computer
2	<=30	high	no	fair	no
3	<=30	high	no	excellent	no
4	31-40	high	no	fair	yes
5	>40	medium	no	fair	yes
6	>40	low	yes	fair	yes
7	>40	low	yes	excellent	no
8	31-40	low	yes	excellent	yes
9	<=30	medium	no	fair	no
10	<=30	low	yes	fair	yes
11	>40	medium	yes	fair	yes
12	<=30	medium	yes	excellent	yes
13	31-40	medium	no	excellent	yes
14	31-40	high	yes	fair	yes
15	>40	medium	no	excellent	no

$P(C_i)$: P(buys_computer = “yes”) = 9/14 = 0.643

P(buys_computer = “no”) = 5/14= 0.357

Compute $P(X|C_i)$ for each class:

P(age = “<=30” | buys_computer = “yes”) = 2/9 = 0.222

P(age = “<= 30” | buys_computer = “no”) = 3/5 = 0.6

P(income = “medium” | buys_computer = “yes”) = 4/9 = 0.444

P(income = “medium” | buys_computer = “no”) = 2/5 = 0.4

P(student = “yes” | buys_computer = “yes) = 6/9 = 0.667

P(student = “yes” | buys_computer = “no”) = 1/5 = 0.2

P(credit_rating = “fair” | buys_computer = “yes”) = 6/9 = 0.667

P(credit_rating = “fair” | buys_computer = “no”) = 2/5 = 0.4

$P(X|C_i)$: P(X|buys_computer = “yes”) = 0.222 x 0.444 x 0.667 x 0.667 = 0.044

P(X|buys_computer = “no”) = 0.6 x 0.4 x 0.2 x 0.4 = 0.019

$P(X|C_i)*P(C_i)$: P(X|buys_computer = “yes”) * P(buys_computer = “yes”) = 0.028

P(X|buys_computer = “no”) * P(buys_computer = “no”) = 0.007

Therefore, X belongs to class (“buys_computer = yes”)

THE ZERO PROBABILITY PROBLEM

- **Naïve Bayesian prediction requires each conditional prob. be non-zero. Otherwise, the predicted prob. will be zero**

$$P(X \mid C_i) = \prod_{k=1}^{n} P(x_k \mid C_i)$$

- **Ex. Suppose a dataset with 1000 tuples, income=low (0), income= medium (990), and income = high (10)**
- **Use Laplacian correction (or Laplacian estimator or smoothing)**

  Adding 1 to each case

  - P(income = low) = 1/1003
  - P(income = medium) = 991/1003
  - P(income = high) = 11/1003
- **The "corrected" prob. estimates are close to their "uncorrected" counterparts**

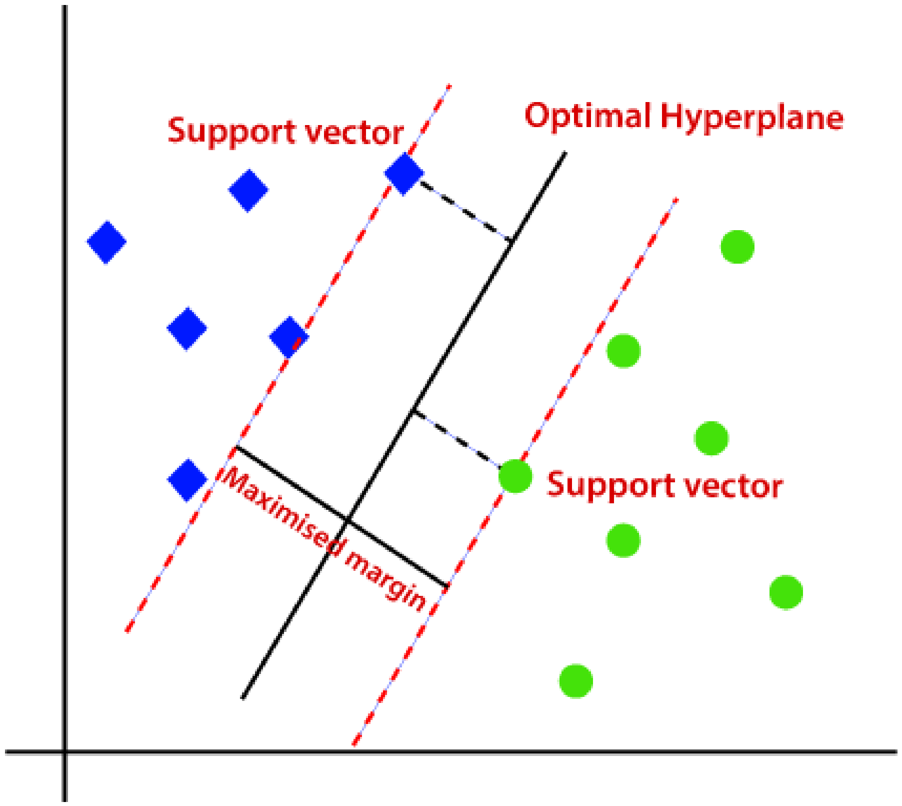
THE NAÏVE BAYES ANALYSIS

- **Advantages**
  - Easy to implement
  - Good results obtained in most of the cases
- **Disadvantages**
  - Assumption: class conditional independence, therefore loss of accuracy
  - Practically, dependencies exist among variables
    - E.g., hospitals: patients: Profile: age, family history, etc. Symptoms: fever, cough etc., Disease: lung cancer, diabetes, etc.
    - Dependencies among these cannot be modeled by Naïve Bayes Classifier
- **How to deal with these dependencies?**

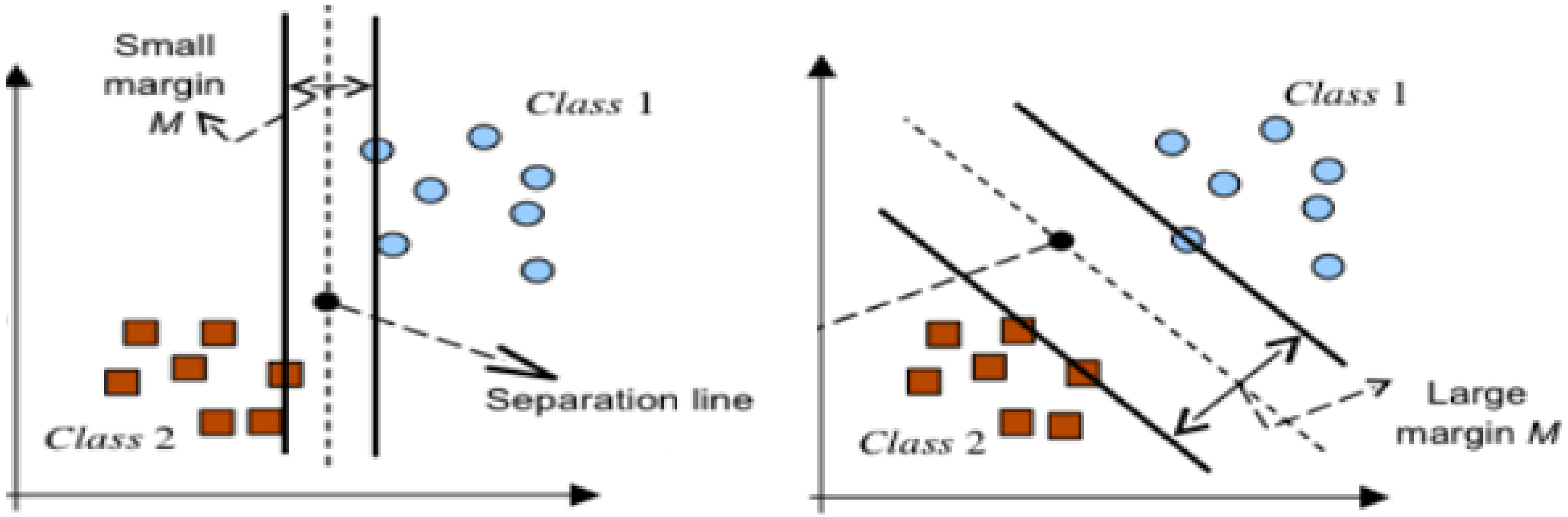
  'Bayesian Belief Networks'

THE SUPPORT VECTOR MACHINE (SVM)

SVM or Support Vector Machine is a linear model for classification and regression problems. It can solve linear and non-linear problems and work well for many practical problems. The idea of **SVM** is simple: The algorithm creates a line or a hyperplane which separates the data into classes.



THE SUPPORT VECTOR MACHINE (SVM)



MODEL EVALUATION

- Use **validation test set** of class-labeled tuples instead of training set when assessing accuracy.
- Methods for estimating a classifier's accuracy:
  1. Holdout method, random subsampling
  2. Cross-validation
  3. Bootstrap
- Comparing classifiers:
  1. Evaluation metrics
  2. Confidence intervals
  3. Cost-benefit analysis and ROC Curves

MODEL EVALUATION

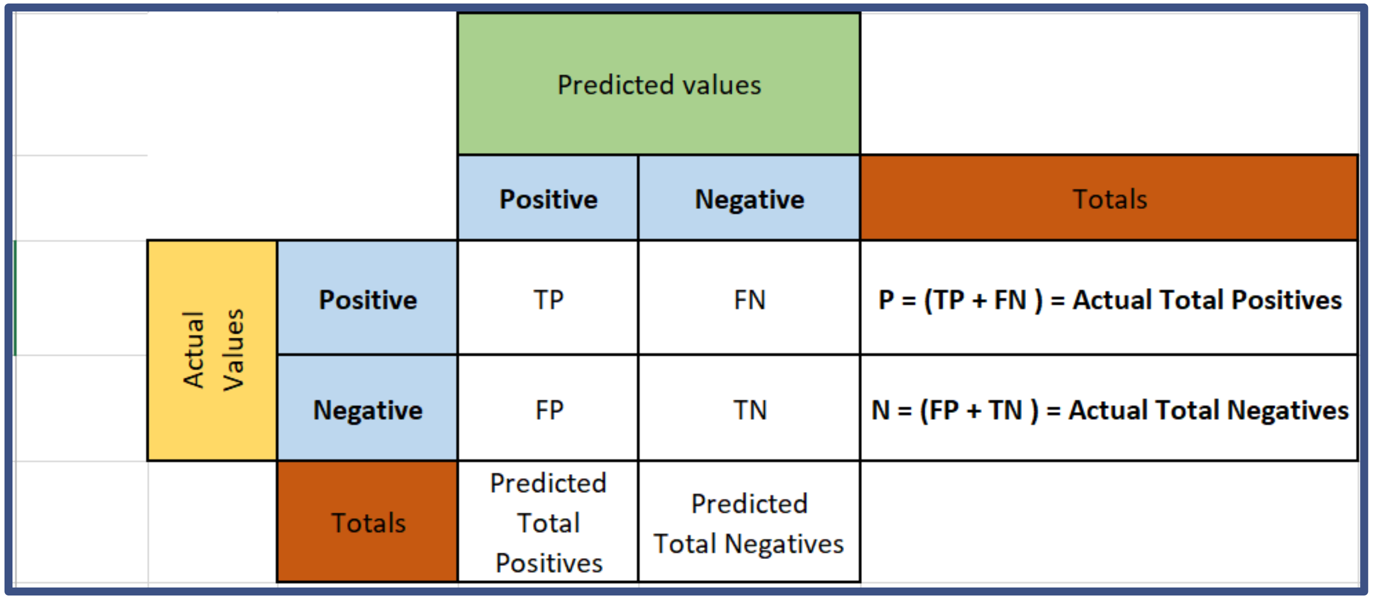
- **Holdout method:**
  - Given training data is randomly partitioned into two independent sets:
    1. Training set (e.g., 2/3) for model construction
    2. Validation Test set (e.g., 1/3) for accuracy estimation
- **Random sampling:**
  - A variation of holdout
  - Repeat holdout k times, accuracy = avg. of the accuracies obtained
- **Cross-validation** (k - fold, where k = 10 is most popular)
  - Randomly partition the data into k mutually exclusive subsets, each approximately equal size
  - At i-th iteration, use D_i as test set and others as training set
  - **Leave-one-out:** k folds where k = # of tuples, for small sized data (one against rest)
  - **Stratified cross-validation:** folds are stratified so that class distribution in each fold is approx. the same as that in the initial data.

MODEL EVALUATION

- **Bootstrap**
- Works well with small data sets
- Samples the given training tuples uniformly *with replacement*
- i.e., each time a tuple is selected, it is equally likely to be selected again and re-added to the training set
- Several bootstrap methods, and a common one is **.632 boostrap**
- A data set with *d* tuples is sampled *d* times, with replacement, resulting in a training set of *d* samples. The data tuples that did not make it into the training set end up forming the test set. About 63.2% of the original data end up in the bootstrap, and the remaining 36.8% form the test set (since $(1 – 1/d)^d \approx e^{-1} = 0.368$)
- Repeat the sampling procedure *k* times, overall accuracy of the model:

$$Acc(M) = \frac{1}{k}\sum_{i=1}^{k}(0.632 \times Acc(M_i)_{test\_set} + 0.368 \times Acc(M_i)_{train\_set})$$

CLASSIFIER EVALUATION METRICS



		Predicted values		
		Positive	**Negative**	Totals
Actual Values	**Positive**	TP	FN	**P = (TP + FN) = Actual Total Positives**
	Negative	FP	TN	**N = (FP + TN) = Actual Total Negatives**
	Totals	Predicted Total Positives	Predicted Total Negatives	

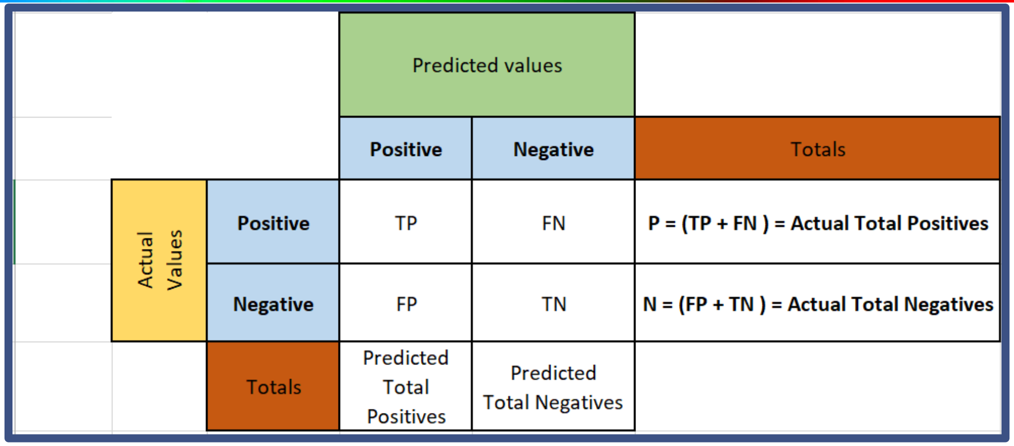
- **Accuracy,** or recognition rate: percentage of test set tuples that are correctly classified.
  Accuracy = (TP + TN)/All (All – TP+FN+FP+TN)
- **Error rate:** 1 – accuracy, or
  Error rate = (FP + FN)/All

CLASSIFIER EVALUATION METRICS

- **Class Imbalance Problem**:
  - One class may be *rare*, e.g. fraud, or HIV-positive
  - Significant *majority of the negative class* and minority of the positive class
  - e.g. in 100 samples, 3 are HIV positive and 97 are negative. If positive are not detected then,

    Accuracy = (TP + TN)/All
    = (0 + 97)/100
    = 97%
  - Though Accuracy is 97% it is not a good classifier as it not considering the False Negative.

EVALUATION METRICS



		Predicted values		
		Positive	Negative	Totals
Actual Values	Positive	TP	FN	P = (TP + FN) = Actual Total Positives
	Negative	FP	TN	N = (FP + TN) = Actual Total Negatives
	Totals	Predicted Total Positives	Predicted Total Negatives	

- **Sensitivity**: Also called **RECALL:**

  True Positive recognition rate. Also called completeness – what % of positive tuples did the classifier label as positive? Perfect score is 1.0

  Recall = TP/P = TP/(TP+FN) (does not consider FP)

- **Specificity**: True Negative recognition rate

  Specificity = TN/N = TN/(TN+FP)

- **Precision:** Also called exactness – what % of tuples that the classifier labeled as positive are actually positive.

  Precision = TP/(TP+FP) (does not consider FN)

EVALUATION METRICS

- **F measure (F$_1$** or **F-score)**: harmonic mean of precision and recall,

$$F_1 = \frac{2}{\frac{1}{\text{recall}} \times \frac{1}{\text{precision}}} = 2 \times \frac{\text{precision} \times \text{recall}}{\text{precision} + \text{recall}}$$
$$= \frac{\text{tp}}{\text{tp} + \frac{1}{2}(\text{fp} + \text{fn})}$$

- **F$_\beta$** measure weighted measure of precision and recall assigns β times as much weight to recall as to precision,

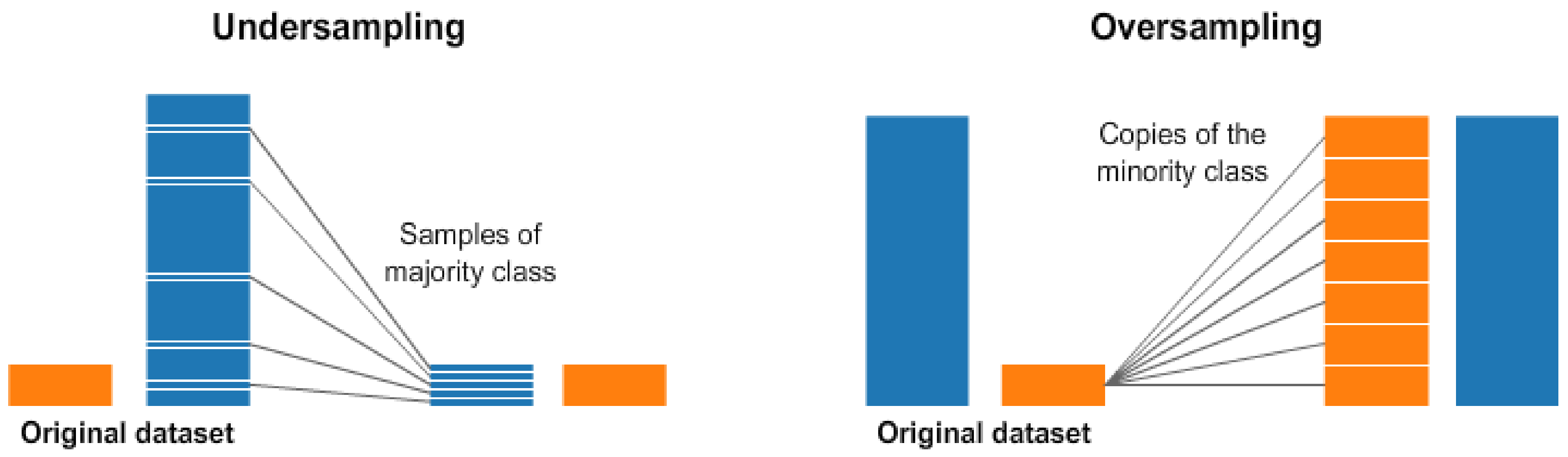
$$F_\beta = (1 + \beta^2) \times \frac{\text{precision} \times \text{recall}}{(\beta^2 \times \text{precision}) + \text{recall}}$$
$$= \frac{(1 + \beta^2)\text{tp}}{(1 + \beta^2)\text{tp} + \beta^2\text{fn} + \text{fp})}$$

HANDLING DATA IMBALANCE

What have datasets in domains like, fraud detection in banking, real-time bidding in marketing or intrusion detection in networks, in common?

- They have less than 1% of rare, but "interesting" events (e.g. fraudsters using credit cards, user clicking advertisement or corrupted server scanning its network).
- Most machine learning algorithms do not work very well with imbalanced datasets.

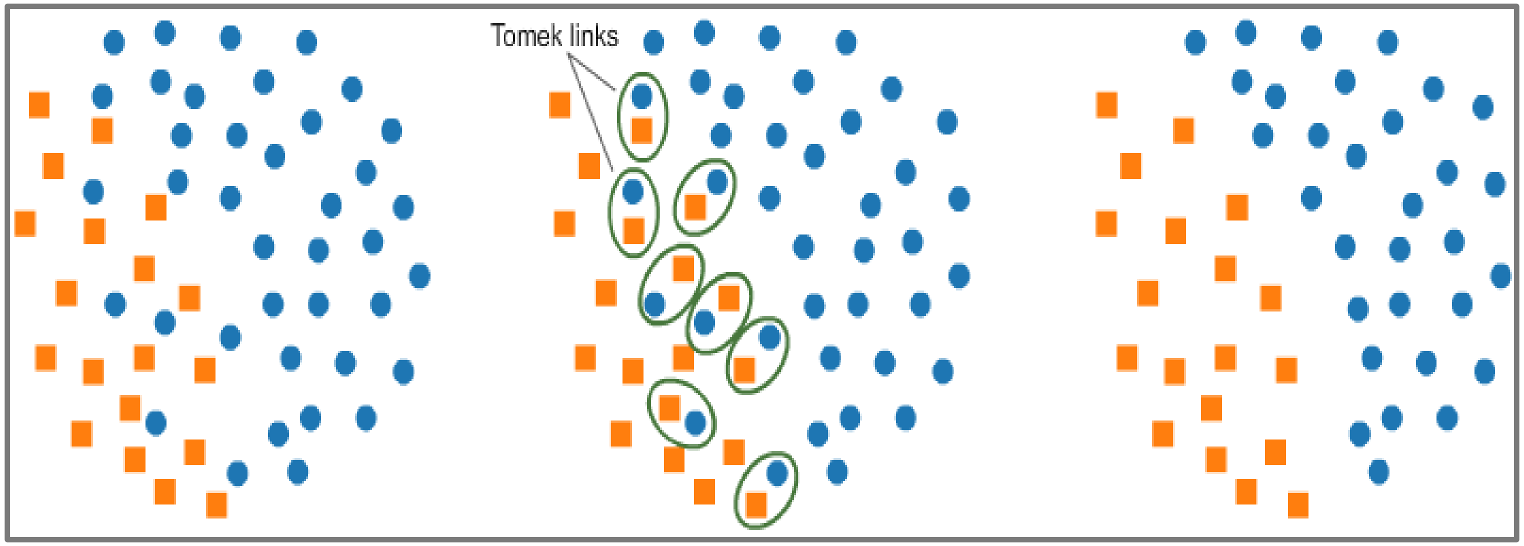
❖ **Random Undersampling and Oversampling**



HANDLING DATA IMBALANCE

Undersampling using Tomek Links: (available in Python)

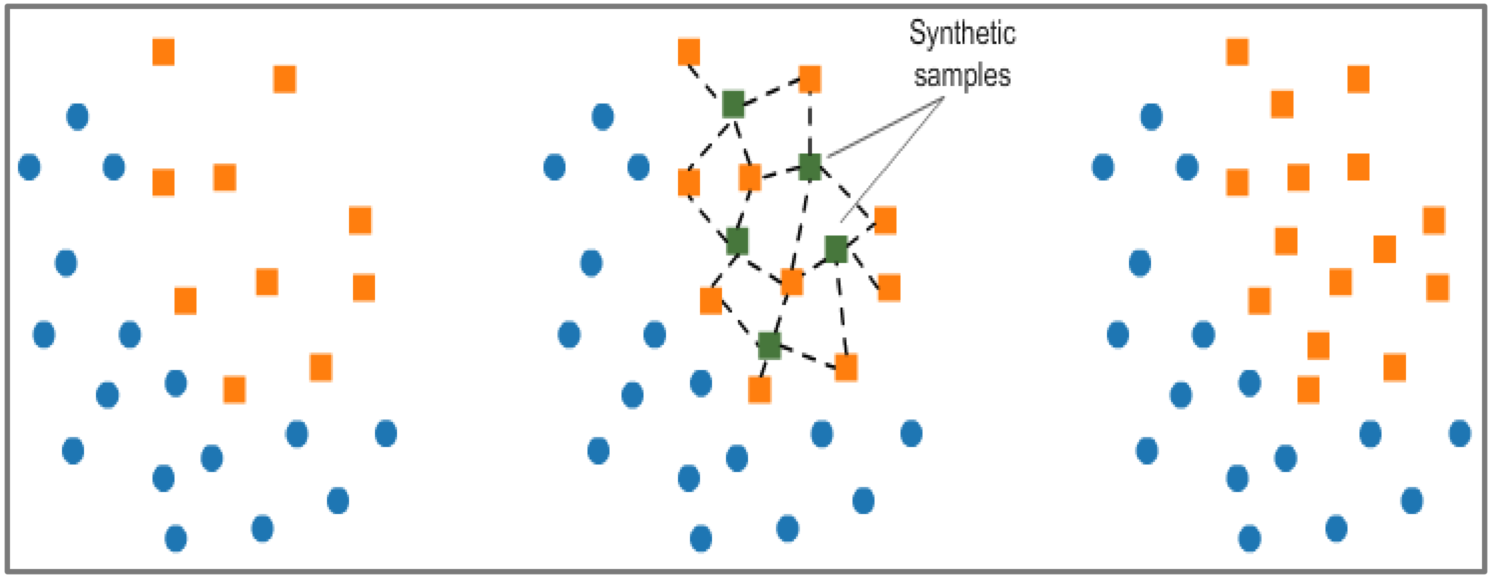
- Tomek links are pairs of examples of opposite classes in close vicinity.
- Removing the majority element from the Tomek link provides a better decision boundary for a classifier.



HANDLING DATA IMBALANCE

Oversampling using SMOTE:

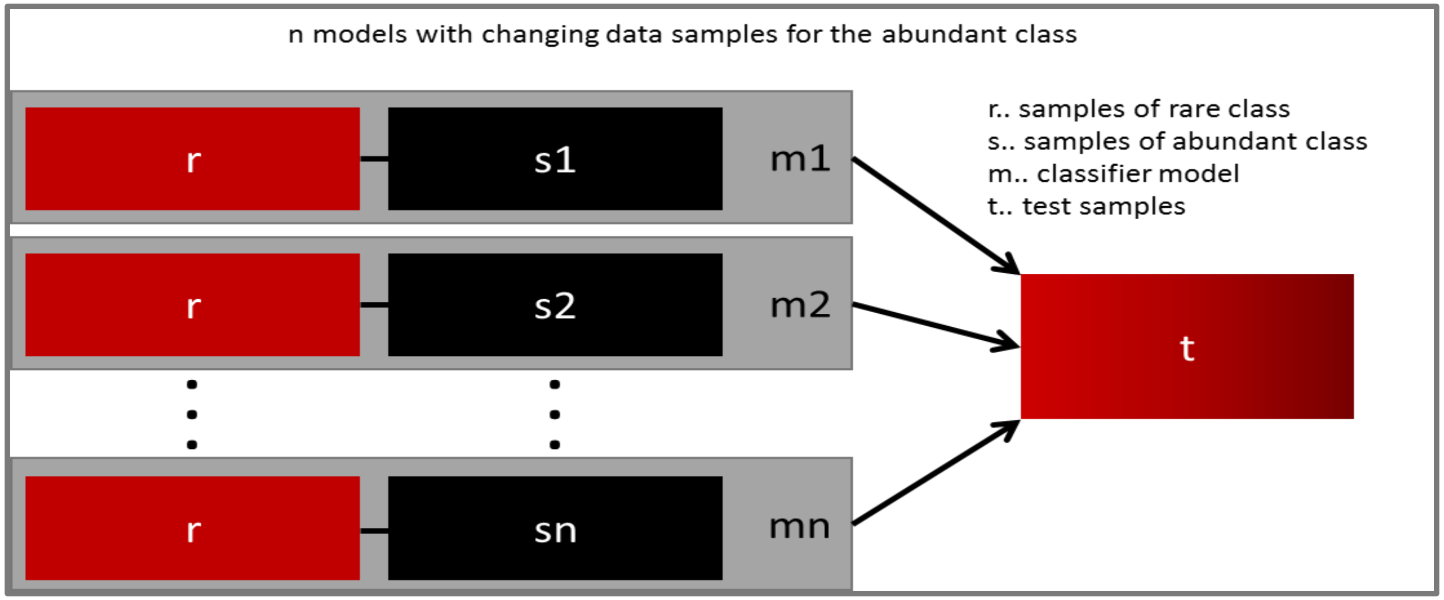
- **SMOTE (Synthetic Minority Oversampling Technique)**
- Synthesize elements for the minority class, in the vicinity of already existing elements. (create artificial elements similar to the minority class)



HANDLING DATA IMBALANCE

Ensemble different resampled datasets:

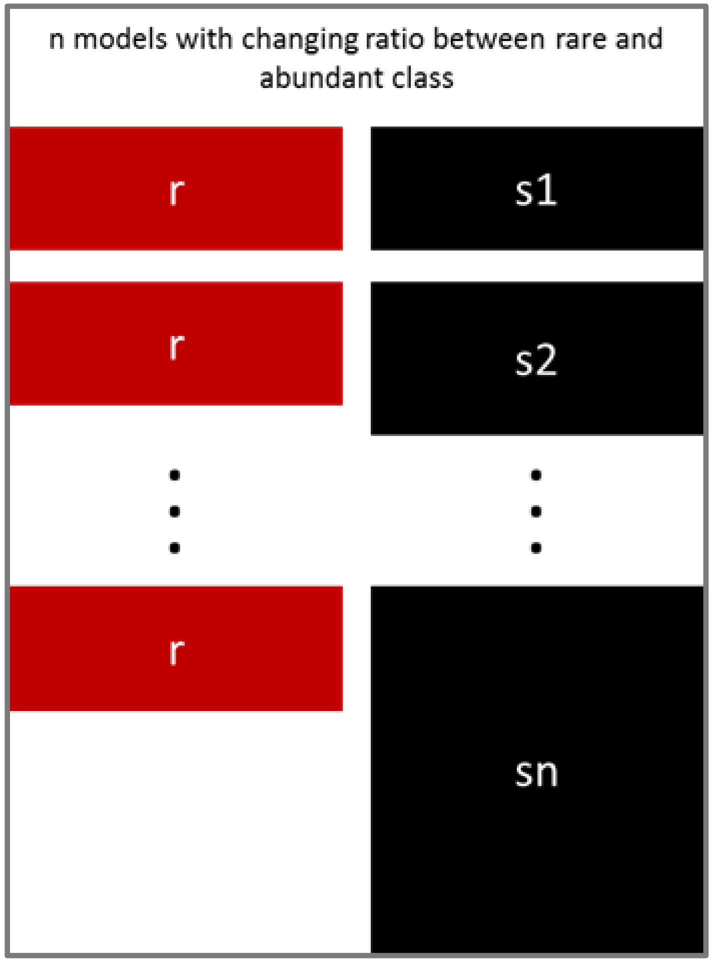
- One easy best practice is building n models that use all the samples of the rare/minor class and n-differing samples of the abundant/major class.
- Given that you want to ensemble 10 models, you would keep e.g. the 1000 cases of the rare class and randomly sample 10000 cases of the abundant class. Then you just split the 10000 cases in 10 chunks and train 10 different models.


n models with changing data samples for the abundant class
r
s1
m1
r
s2
m2
r
sn
mn
t
r.. samples of rare class
s.. samples of abundant class
m.. classifier model
t.. test samples

HANDLING DATA IMBALANCE

Resample with different ratios:

- The previous approach can be fine-tuned by playing with the ratio between the rare and the abundant class.
- The best ratio heavily depends on the data and the models that are used. But instead of training all models with the same ratio in the ensemble, it is worth trying to ensemble different ratios.
- So if 10 models are trained, it might make sense to have a model that has a ratio of 1:1 (rare:abundant) and another one with 1:3, or even 2:1. Depending on the model used this can influence the weight that one class gets.



n models with changing ratio between rare and abundant class
r
s1
r
s2
r
sn

HANDLING DATA IMBALANCE

Assign Class weights in the model:

- Most of the machine learning models provide a parameter called class_weights
- Specify a higher weight for the minority class.
- We penalize our model more when it misclassifies a positive minority example and penalize less when it misclassifies a negative example

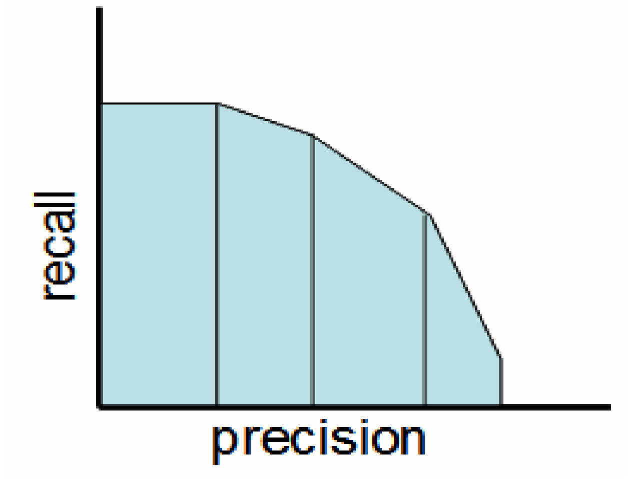
HANDLING DATA IMBALANCE

$$\text{F1 Score} = \frac{2 \times (\text{Precision} \times \text{Recall})}{\text{Precision} + \text{Recall}}$$

Change your Evaluation Metric:

- The F1 Score should be the ***evaluation metric***.
- The F1 score is a number between 0 and 1 and is the harmonic mean of precision and recall.
- ***The F1 score sort of maintains a balance between the precision and recall for your classifier***. If your precision is low, the F1 is low, and if the recall is low again, your F1 score is low.
- E.g. If you are a police inspector and you want to catch criminals, you want to be sure that the person you catch is a criminal (Precision) and you also want to capture as many criminals (Recall) as possible. The F1 score manages this tradeoff.

Precision-Recall Tradeoff



HANDLING DATA IMBALANCE

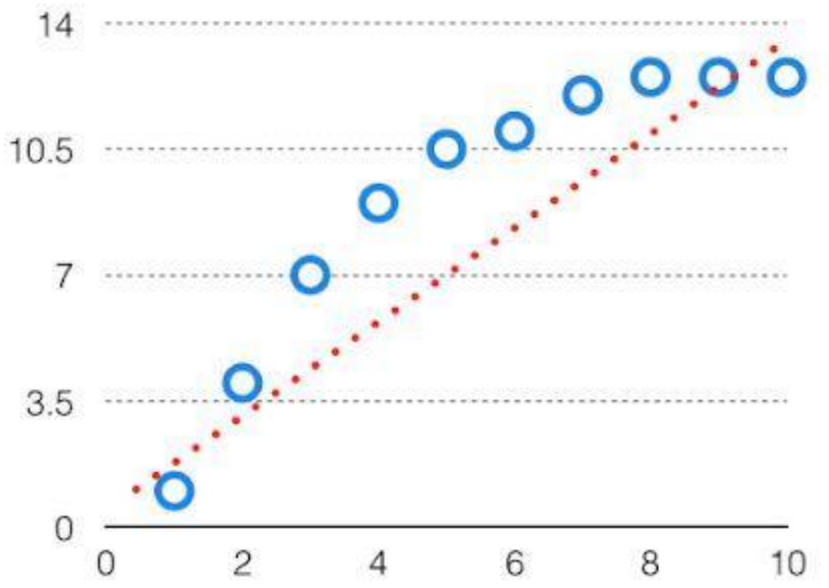
Miscellaneous:

- **Collect more Data**
  Getting more data with more positive examples is going to help your models get a more varied perspective of both the majority and minority classes.
- **Treat the problem as anomaly detection**
  Anomaly detection is the identification of rare items, events or observations which raise suspicions by differing significantly from the majority of the data. You can use Isolation forests or autoencoders for anomaly detection.
- **Model-based**
  Some models are particularly suited for imbalanced datasets. For example, in boosting models, we give more weights to the cases that get misclassified in each tree iteration.

Bias

Bias

- It is the difference between the average prediction of our model and the correct value which we are trying to predict.
- Model with high bias pays very little attention to the training data and oversimplifies the model.
- Being high in bias gives a large error in training as well as testing data.
- By high bias, the data predicted is in a straight line format, thus not fitting accurately in the data in the data set.
- Such fitting is known as **Underfitting of Data**.
- This happens when the hypothesis is too simple or linear in nature.



High Bias

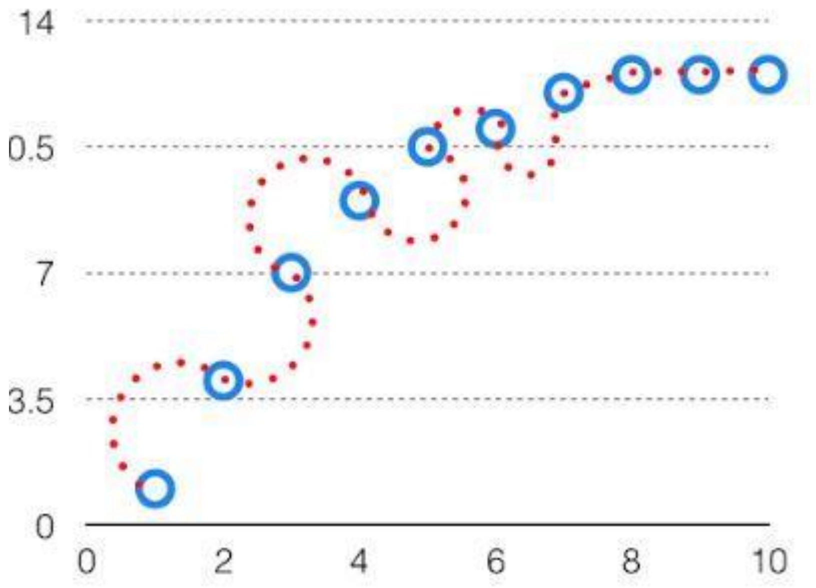
$$h_\theta(x) = g(\theta_0 + \theta_1 x_1 + \theta_2 x_2)$$

Most of the data points do not fall on the curve.

Variance

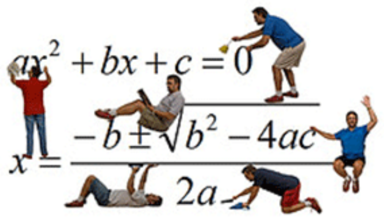
Variance

- It is the variability of model prediction for a given data point or a value which tells us spread of our data.
- Model with high variance pays a lot of attention to training data and does not generalize on the data which it hasn't seen before. (non-linear - polynomial)
- Such models perform very well on training data but has high error rates on test data.
- When a model is high on variance, it is then said to as **Overfitting of Data**.
- Overfitting is fitting the training set accurately via complex curve and high order hypothesis but is not the solution as the error with unseen data is high.



$$h_\theta(x) = \theta_0 + \theta_1 x + \theta_2 x^2 + \theta_3 x^3 + \theta_4 x^4$$

Most of the data points fall on the curve. (also the noise points)

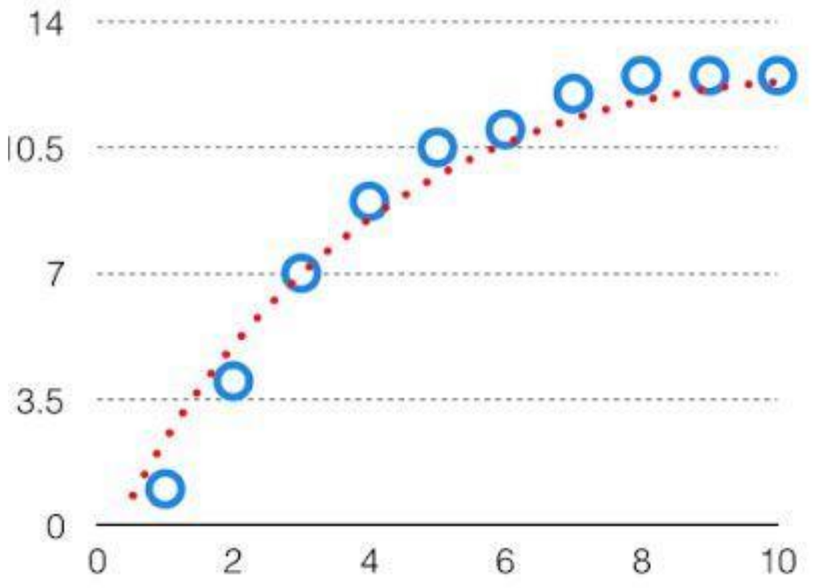


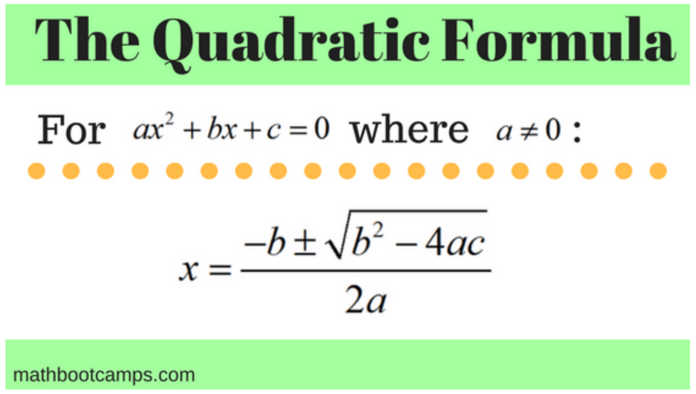
© Ian Vogt 2008

Bias-Variance

Bias-Variance Trade-off

- If the algorithm is too simple (hypothesis with linear eq.) then it may be on high bias and low variance condition and thus is error-prone.
- If algorithms fit is too complex (hypothesis with high degree eq.) then it may be on high variance and low bias. In the latter condition, the new entries will not perform well.
- There is something between both of these conditions, known as Trade-off or Bias Variance Trade-off.
- An algorithm can’t be more complex and less complex at the same time. For the graph, the perfect tradeoff will be like shown on the RHS.



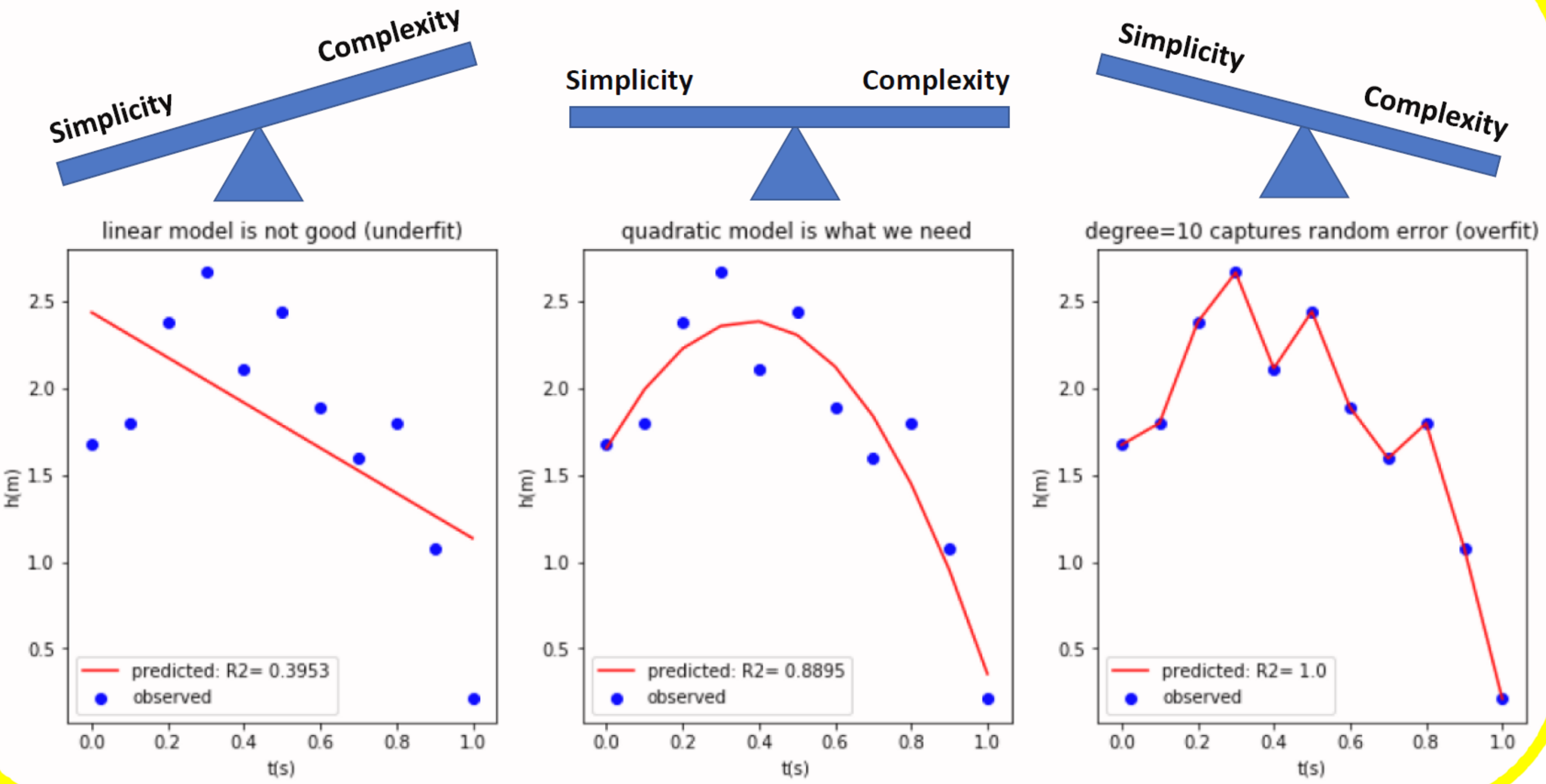

The Quadratic Formula

For $ax^2 + bx + c = 0$ where $a \neq 0$:

$$x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$

mathbootcamps.com

Bias-Variance



Irreducible error

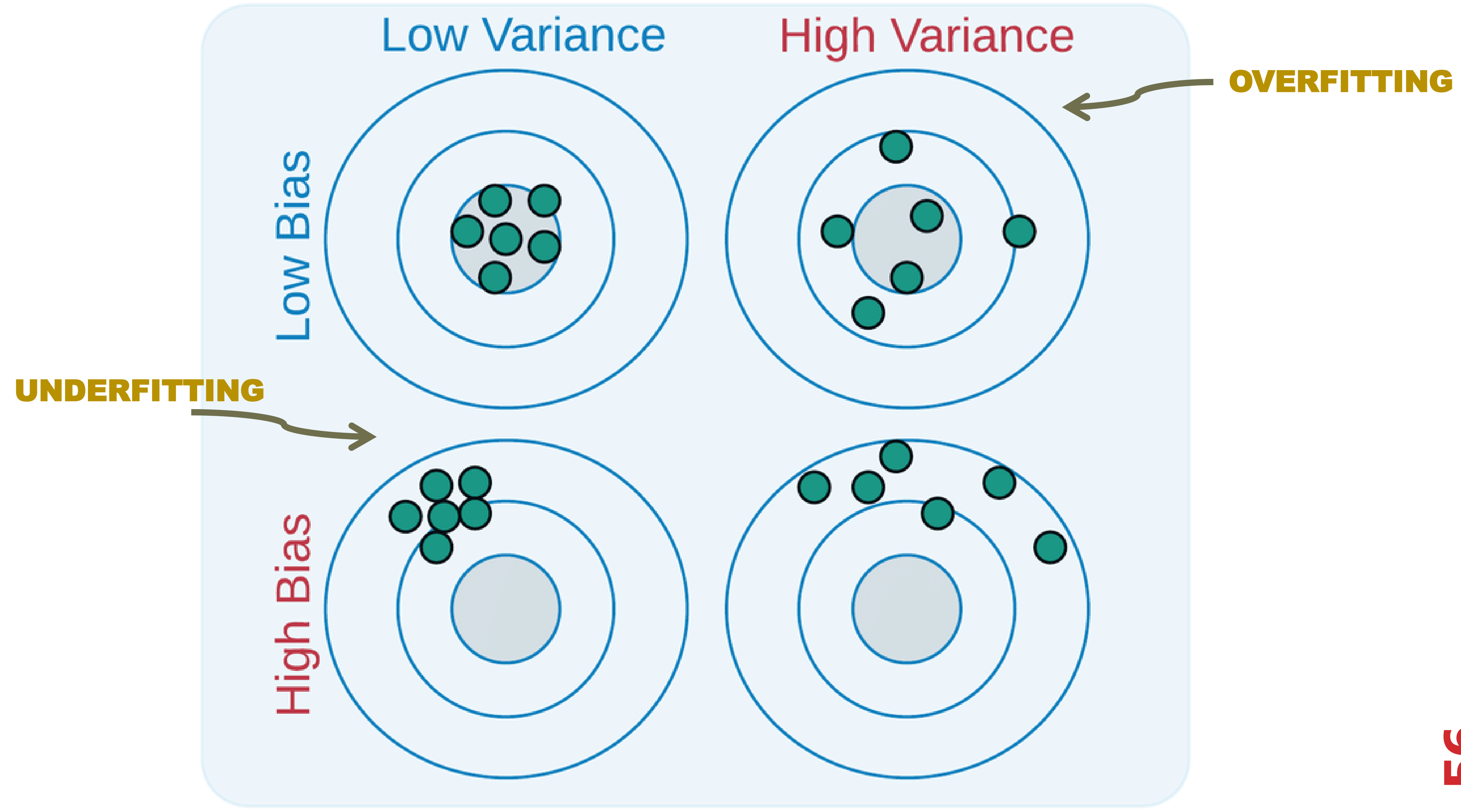
Irreducible error (Noise)

- It is the error that can’t be reduced by creating good models.
- It is a measure of the amount of noise in our data.
- Here it is important to understand that no matter how good we make our model, our data will have certain amount of noise or irreducible error that can not be removed.
- Bias and variance are reducible errors.
- The total error is calculated using the following formula.

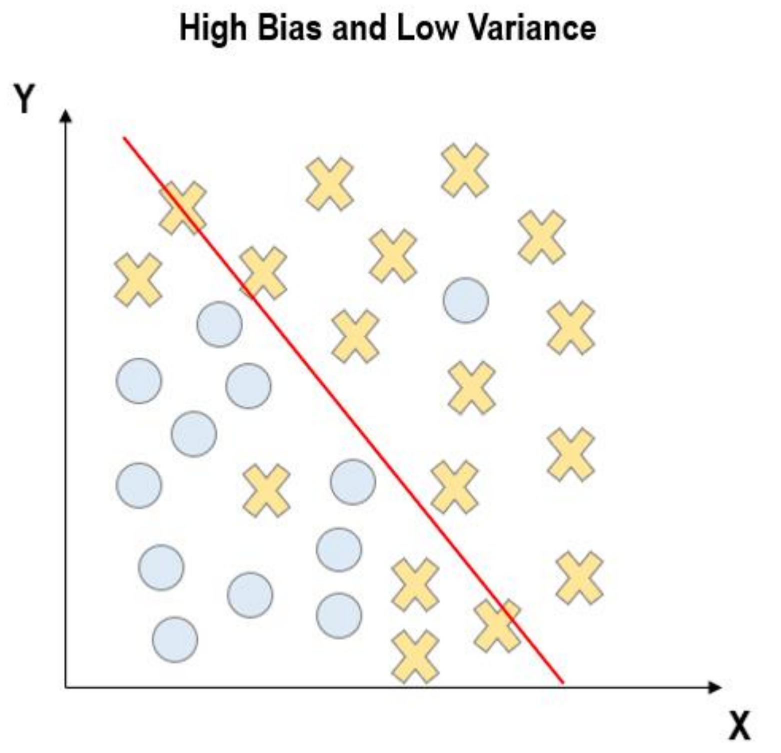
Total Error = Bias^2 + Variance + Irreducible Error

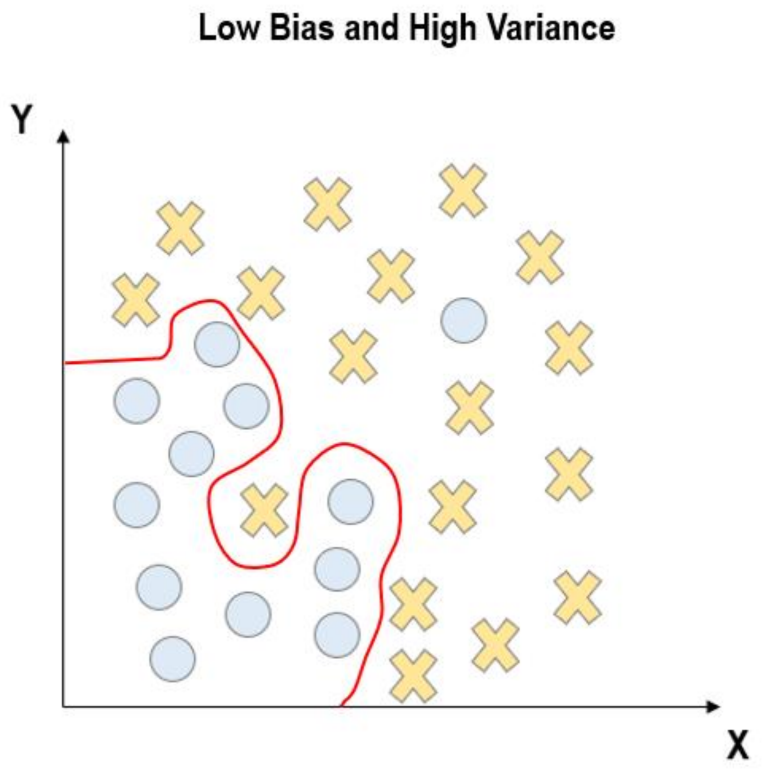
Bias-Variance

In the diagram, centre of the target is a model that perfectly predicts correct values.



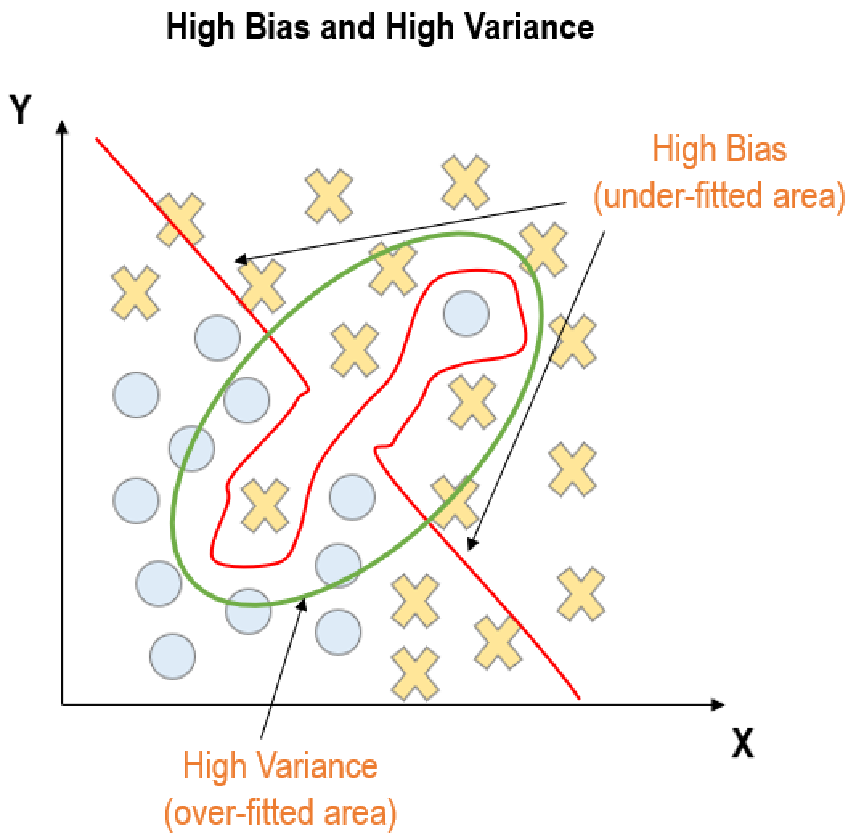
Bias-Variance

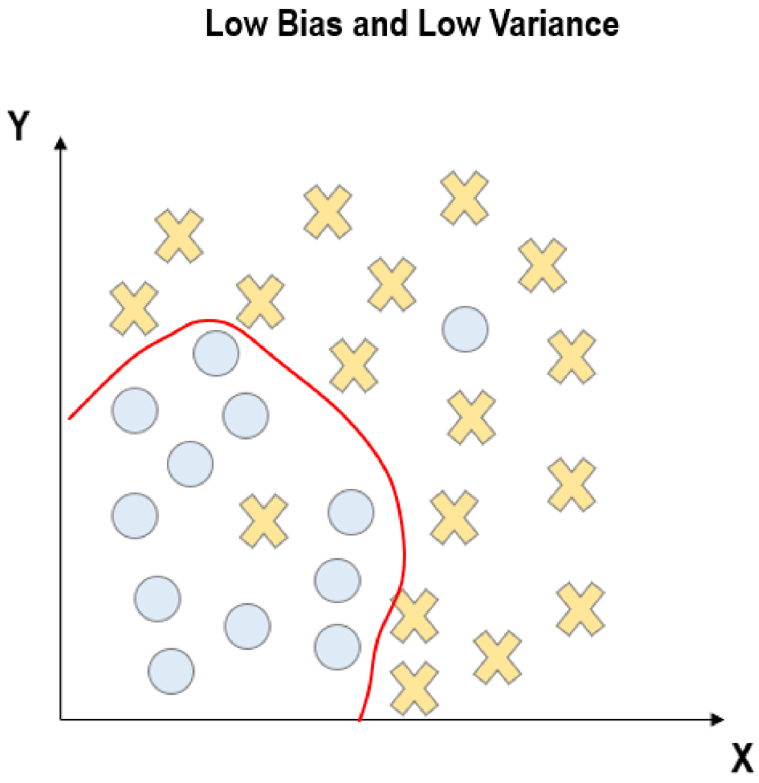




1. High bias – Data is not modelled correctly. Some points lie on the line too. (underfittiing)
2. High variance – Data is modelled as per the training set values. (overfitting)

Bias-Variance





1. High bias, high variance – Combination of overfitting and underfitting
2. Low bias, low variance – just right model.

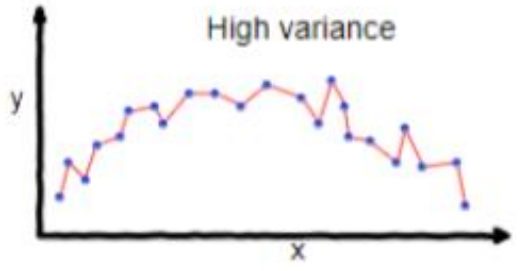
Bias-Variance

- In supervised learning, **underfitting** happens when a model unable to capture the underlying pattern of the data. These models usually have high bias and low variance. It happens when we have very less amount of data to build an accurate model or when we try to build a linear model with a nonlinear data. Also, these kind of models are very simple to capture the complex patterns in data like Linear and logistic regression.

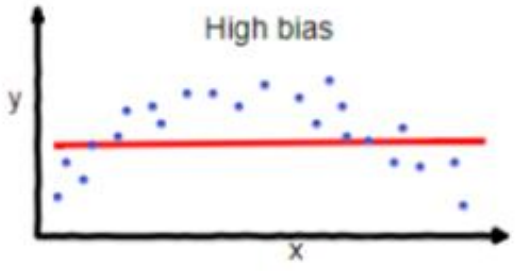
- In supervised learning, **overfitting** happens when our model captures the noise along with the underlying pattern in data. It happens when we train our model a lot over noisy dataset. These models have low bias and high variance. These models are very complex like Decision trees which are prone to overfitting.

- Check out the following video:
  https://youtu.be/EuBBz3bI-aA

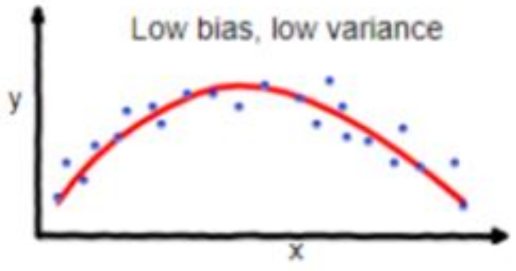
Bias-Variance



overfitting



underfitting



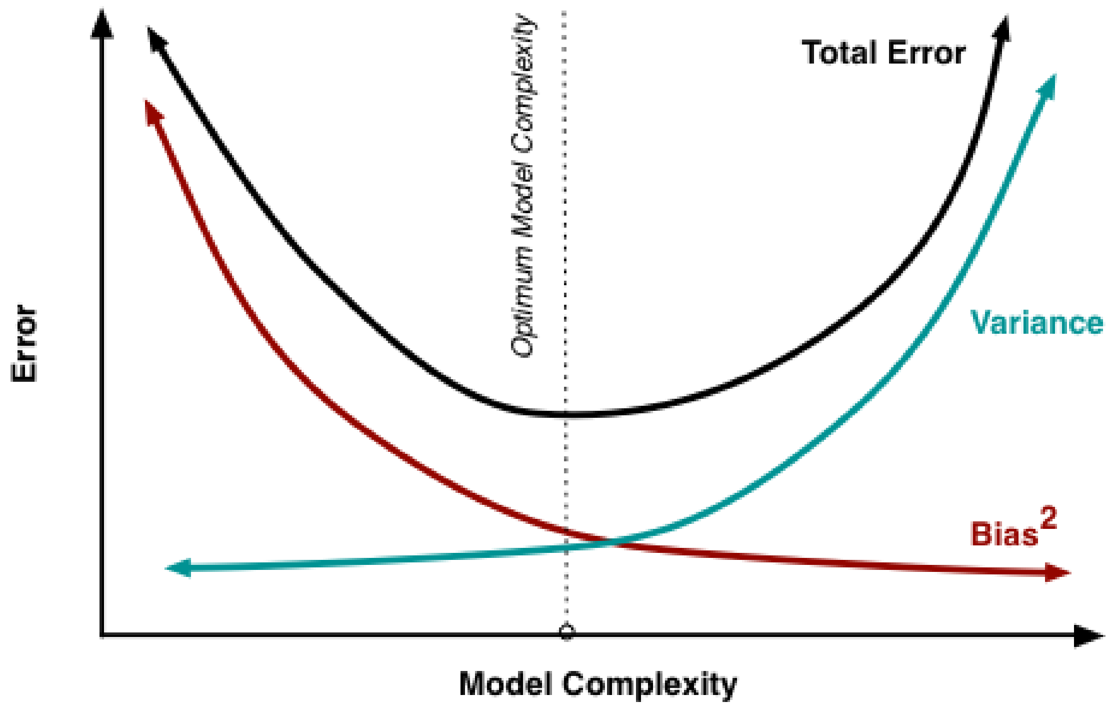
Good balance

- If our model is too simple and has very **few parameters** then it may have **high bias** and low variance.
- On the other hand if our model has large number of parameters then it's going to have high variance and low bias. So we need to find the right/good balance without overfitting and underfitting the data.

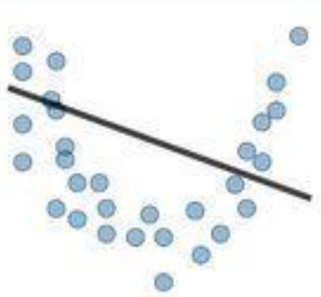
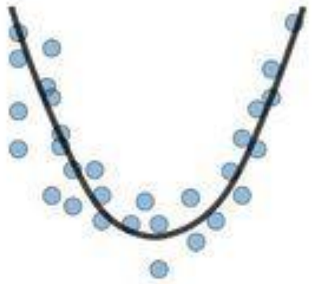
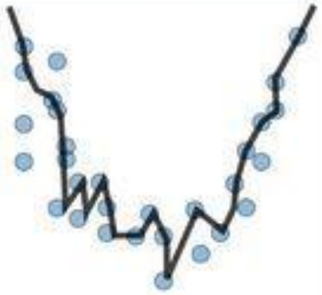
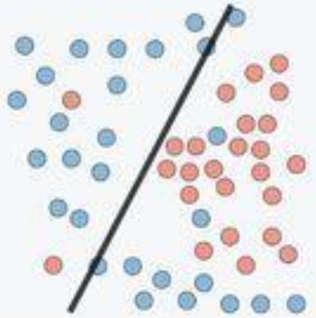
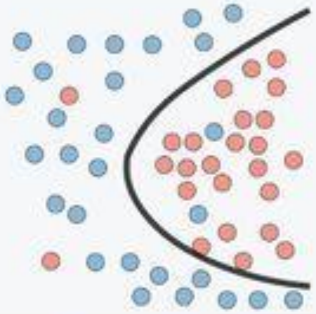
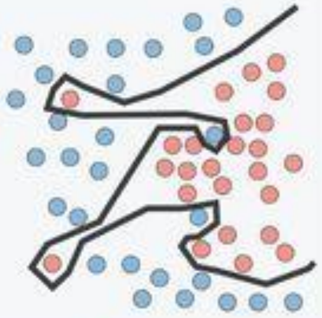
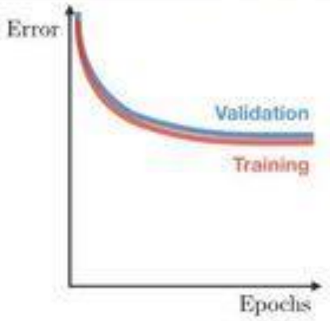
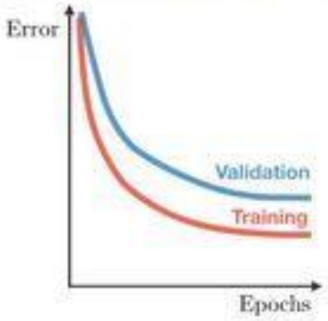
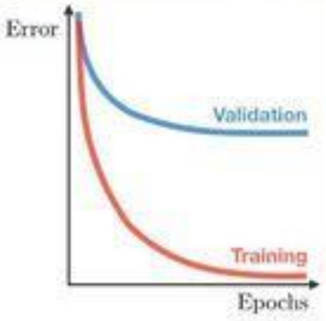
Bias-Variance

To build a good model, we need to find a good balance between bias and variance such that it minimizes the total error.

Total Error = Bias^2 + Variance + Irreducible Error



An optimal balance of bias and variance would never overfit or underfit the model.

	Underfitting	Just right	Overfitting
Symptoms	• High training error • Training error close to test error • High bias	• Training error slightly lower than test error	• Very low training error • Training error much lower than test error • High variance
Regression illustration			
Classification illustration			
Deep learning illustration			
Possible remedies	• Complexify model • Add more features • Train longer		• Perform regularization • Get more data

Bias-Variance

Detection of High Bias

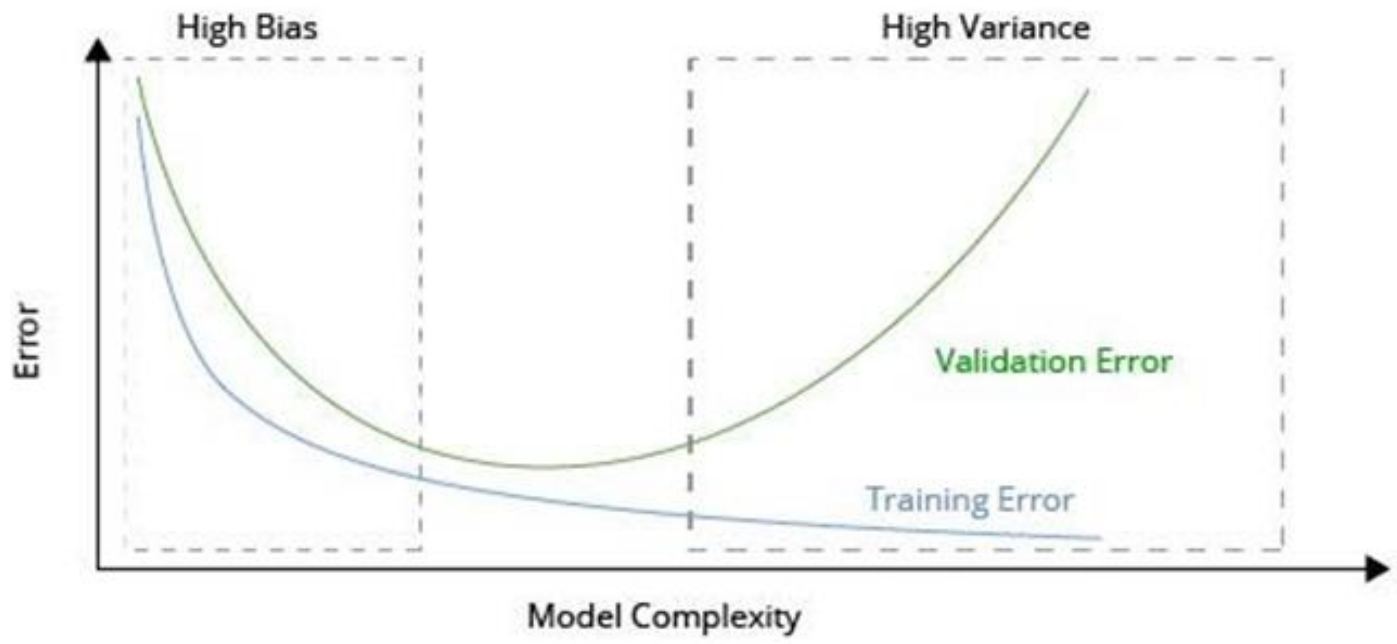
- ➢ The model suffers from a very low Training Accuracy.
- ➢ The Validation accuracy is low and similar in magnitude to the training accuracy.
- ➢ A model with high bias suffers from underfitting.

Detection of High Variance

- ➢ The model suffers from a very low Training Error (high accuracy).
- ➢ The Validation error is very high as compared to the Training Error.
- ➢ A model with high variance suffers from overfitting.

Bias-Variance

Example:- To Detect a model suffering from High Bias and Variance is shown below figure:
Reduce variance – increasing training data set size, reduce features
Reduce bias – increase number of features



Summary

- Classification is a form of data analysis that extracts models describing important data classes.
- Effective and scalable methods have been developed for decision tree induction, Naive Bayesian classification, rule-based classification, and many other classification methods.
- Evaluation metrics include: accuracy, sensitivity, specificity, precision, recall, F measure, and F_{β} measure.
- Stratified k-fold cross-validation is recommended for accuracy estimation. Bagging and boosting can be used to increase overall accuracy by learning and combining a series of individual models.
- Significance tests and ROC curves are useful for model selection.
- There have been numerous comparisons of the different classification methods; the matter remains a research topic
- No single method has been found to be superior over all others for all data sets
- Issues such as accuracy, training time, robustness, scalability, and interpretability must be considered and can involve trade-offs, further complicating the quest for an overall superior method

END