Machine Learning Lecture Series

Machine Learning Elective IV

CSE21816

UNIT: 02

Lecture: 01

Course Instructor

Dr. Tamal Ghosh

Associate Professor

Computer Science and Engineering

tamal.ghosh1@adamasuniversity.ac.in

Supervised Learning:

Probably Approximately Correct Learning

Probably Approximately Correct Learning

- A good learner will learn with high probability and close approximation to the target concept
- With high probability, the selected hypothesis will have **lower the error** (“Approximately Correct”) with the parameters Ɛ and δ

PAC Learning

- PAC learning, requires
  - small parameters Ɛ and δ,
  - with probability at least (1 – δ), a system learn the concept with error at most Ɛ.
- Ɛ is upper bound on the error in accuracy, i.e. the hypothesis with error less than Ɛ

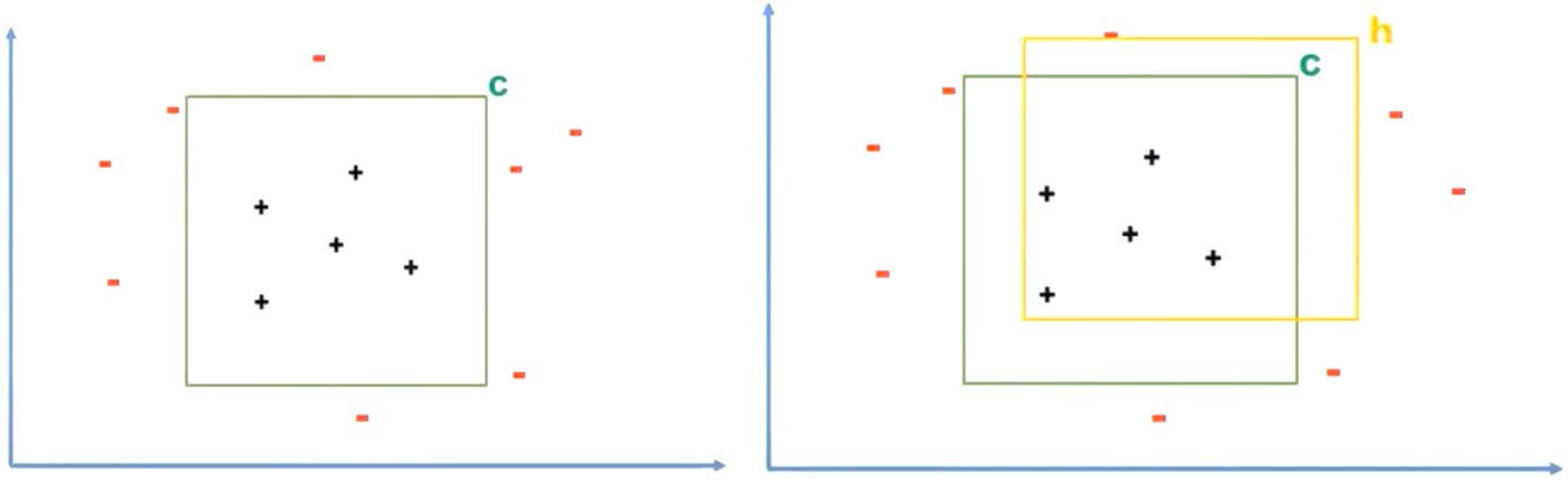
  Accuracy: 1- Ɛ

- δ give the probability of failure in achieving this accuracy δ, $(0<\delta\leq1)$, the hypothesis generated is approximately correct at least $1-\delta$ of the time.

  Confidence: 1- δ

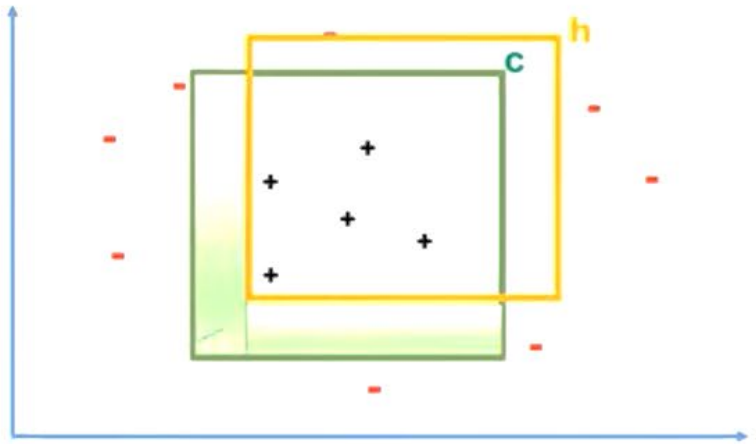
PAC Learning - Example

- N number of Car having Price and Engine power, as training set, (p,e), find the car is family car or not.
- An algorithm gives answer whether the car is family car or not.
- C – Target function
- Instances within rectangle represents family cars and outside are not family cars
- Hypothesis h – closely approximate C, and there may be error region.

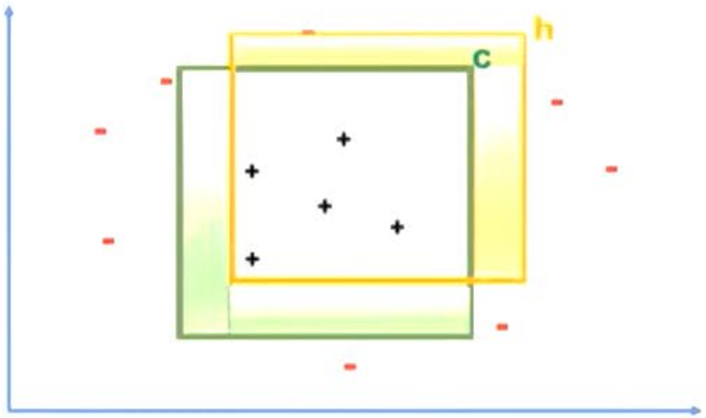


False Positive and False Negative

- Instances lies on shaded region are positive/negative according to our actual function ‘C’, but those are negative/positive based on the hypothesis h. Hence it is called as false negative or false positive



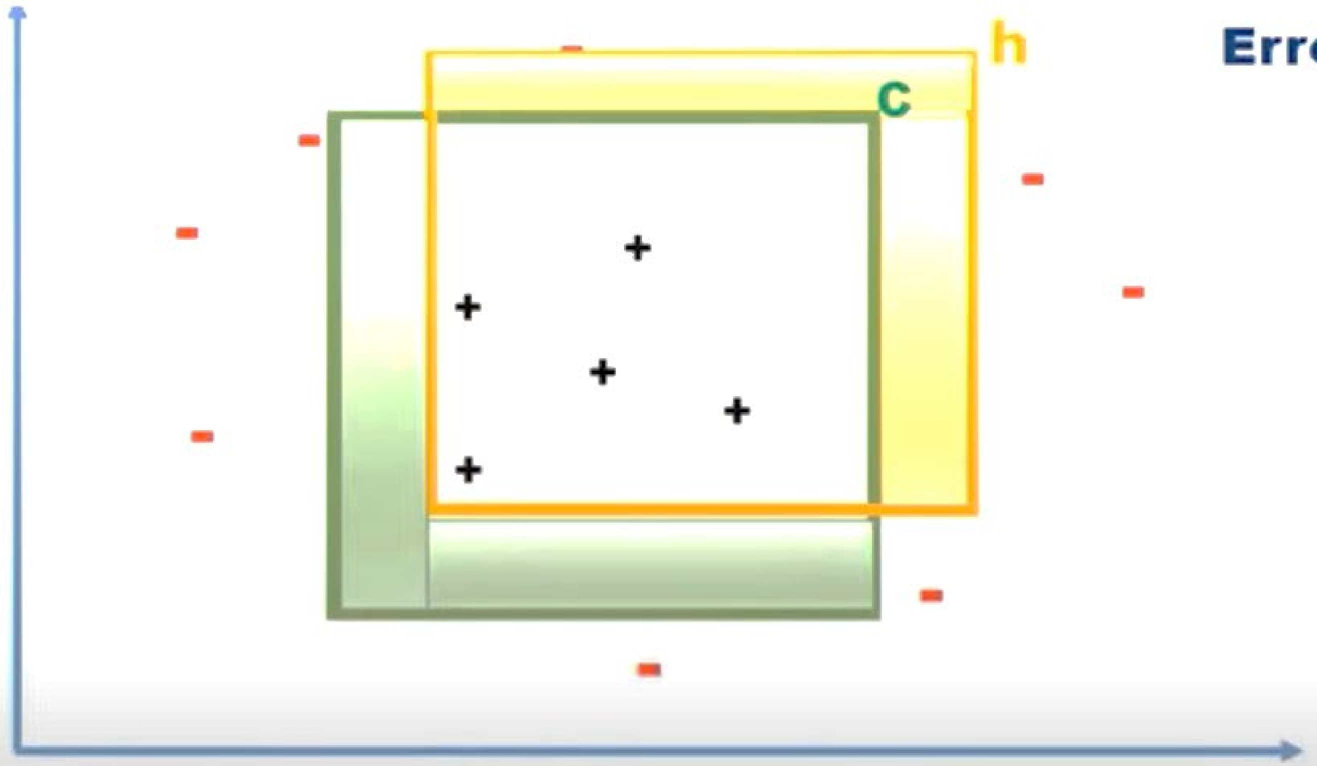
False Negative



False Positive

Error Region

- The probability of error region to be small
- The error region : P(C XOR h) <= Ɛ.



Error Region : C XOR h

Approximately Correct

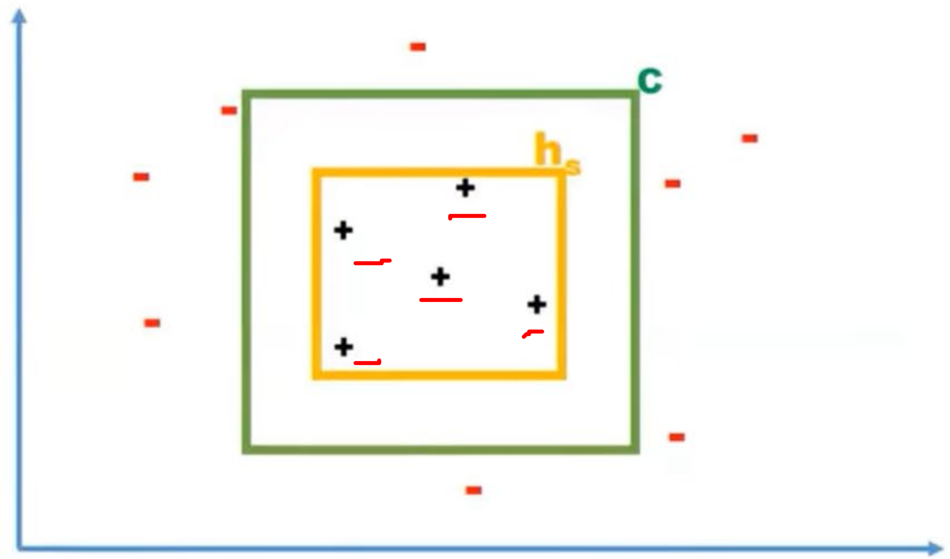
- the hypothesis h, that approximately correct, and error is less than or equal to Ɛ.
- Where 0<= Ɛ<=1/2
- i.e. P(C XOR h) <= Ɛ

Probably Approximately Correct

- Low generalization error with high probability
- [P(Error(h) <= Ɛ)] <= 1- δ
- P(P(C XOR h) <= Ɛ) <= 1- δ

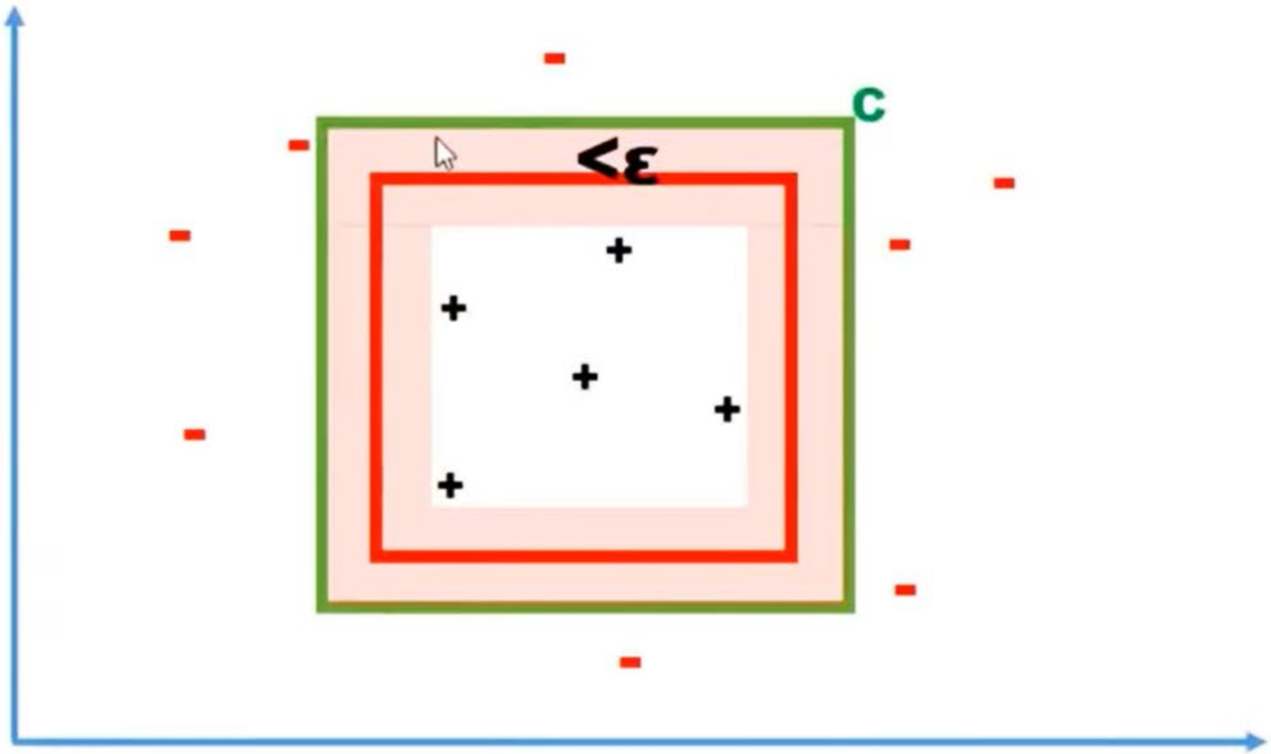
PAC Learnability for Axis-aligned Rectangle

- Specialization:
- h_s is the tightest possible rectangle around a set of positive training examples.
- h_s is subset of C , Hence Error region = C - h

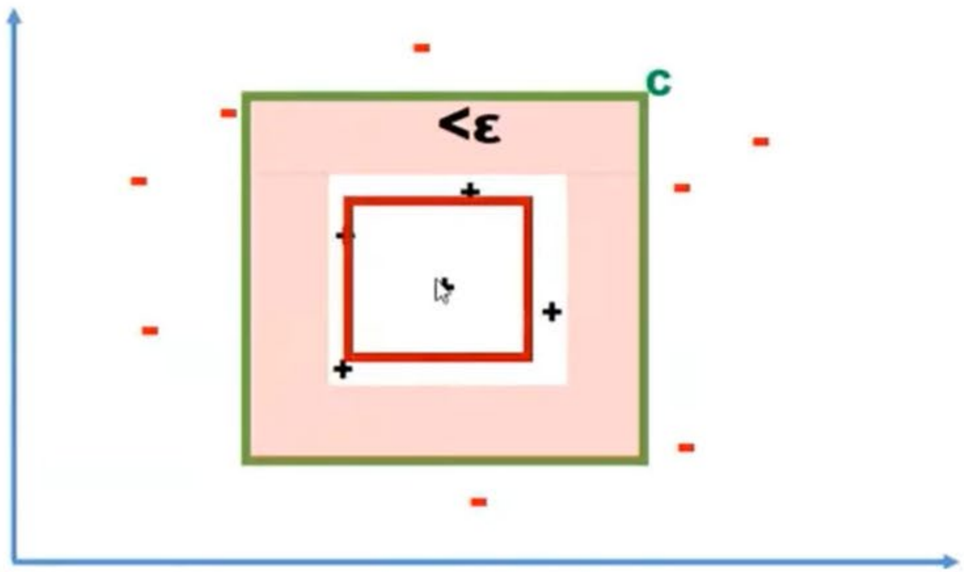


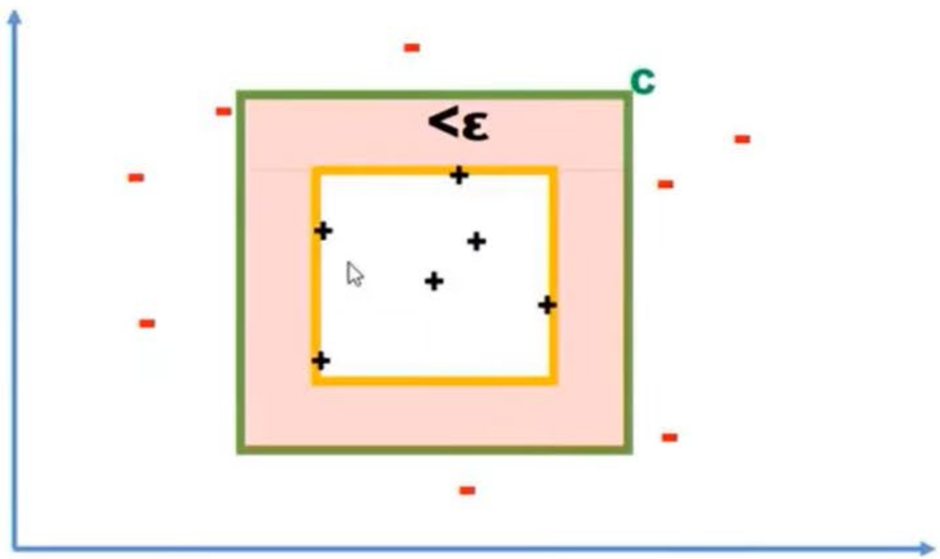
Approximately Correct

- If an hypothesis lies between h and c (shaded region) then it is approximately correct.



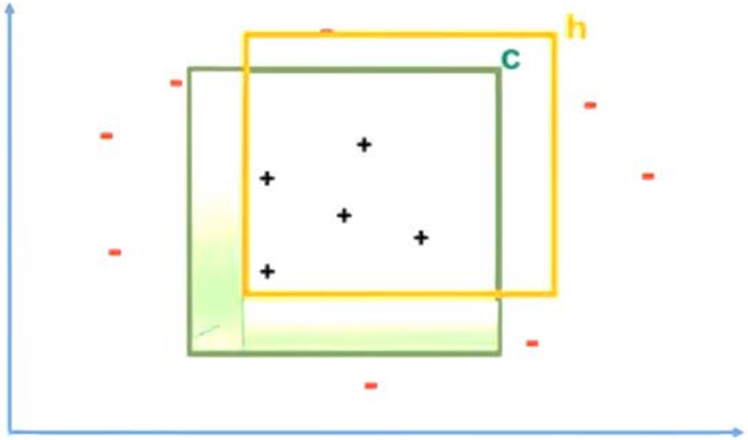
- If the generated hypothesis does not touch any of these region
- Error region is greater than Ɛ and not approximately correct, because the error region got increased.
- Atleast one +ve example at each side of the rectangle



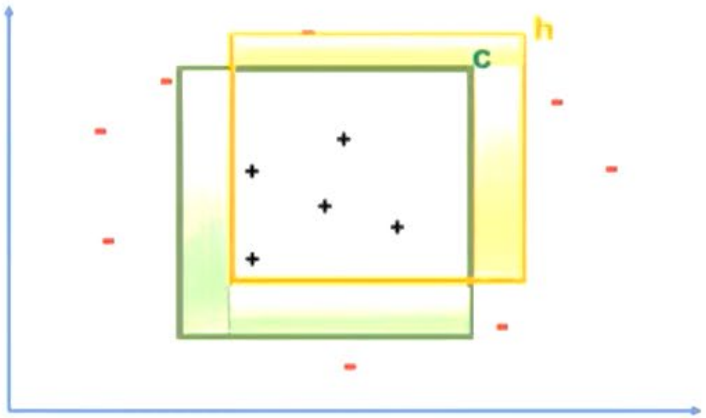


False Positive and False Negative

- Instances lies on shaded region are positive/negative according to our actual function ‘C’, but those are negative/positive based on the hypothesis h. Hence it is called as false negative or false positive



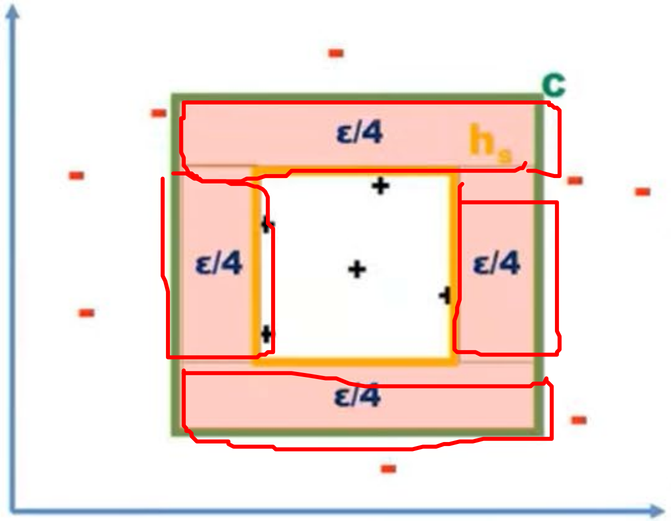
False Positive



False Negative

Error Region

- Error Region = sum of four rectangular strips < Ɛ
- Each strip is at most Ɛ/4
- Probability of positive example falling in any one of the strip (error region = Ɛ/4)
- Probability that a randomly drawn positive example misses a strip = 1- Ɛ/4
- P(m instance miss a strip) = $(1- \varepsilon/4)^m$
- P(m instances miss any strip) < $4(1- \varepsilon/4)^m$
- Finally we get m > 4/ Ɛ log 4/ δ



Example 1

Sl.No.	Error(h1)
1	0.001
2	0.025
3	0.07
4	0.003
5	0.035
6	0.045
7	0.027
8	0.065
9	0.012
10	0.036

- Hypothesis h1 generated the errors with respect to price and engine power of given 10 samples,
- *Given*, Ɛ = 0.05 δ = 0.20
- P(h1) >= 1-δ
- P(h1)= 8/10 = 0.80 (3rd and 8th values are greater than Ɛ)
- Therefore, 0.80 >= (1 - 0.20) i.e. 0.80 = 0.80
- **Hence h1 is probably approximately correct**

Example 2

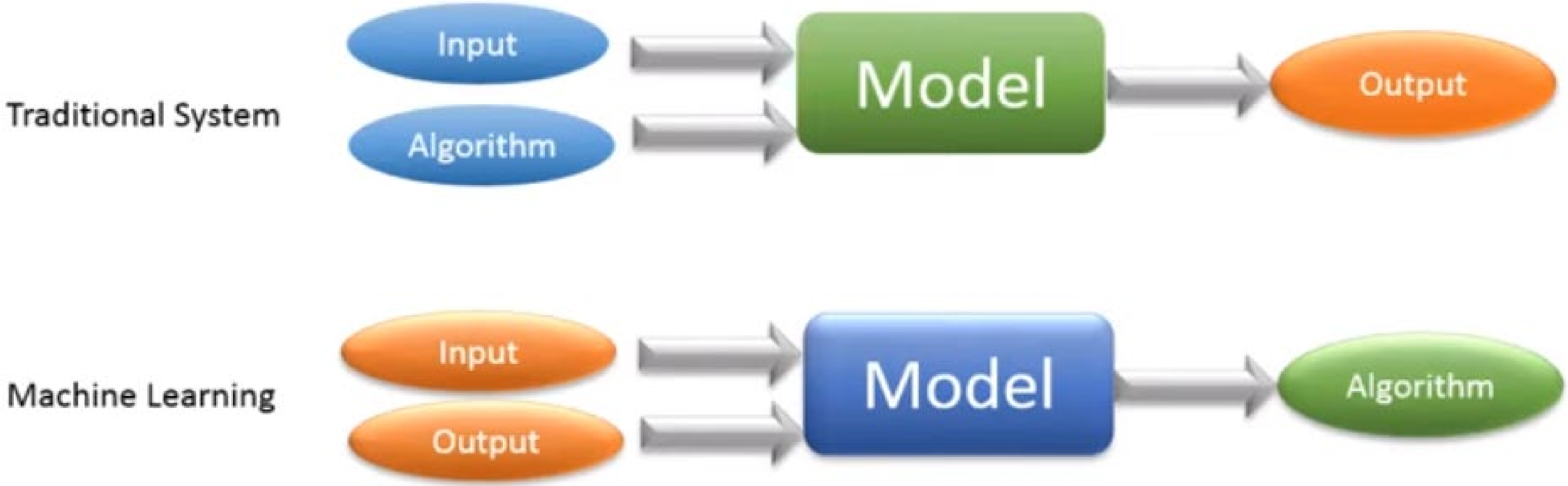
Sl.No.	Error(h2)
1	0.012
2	0.015
3	0.071
4	0.063
5	0.022
6	0.045
7	0.011
8	0.029
9	0.066
10	0.031

- Hypothesis h2 generated the errors with respect to price and engine power of given 10 samples,
- *Given*, Ɛ = 0.05 δ = 0.20
- P(h2) >= 1-δ
- P(h2)= 7/10 = 0.70 (3^{rd},4^{th},9^{th} values > Ɛ)
- Here, 0.70 >= (1-0.20) i.e. 0.70 < 0.80
- Hence h2 is not probably approximately correct

Supervised Learning:

Learning a Class From Examples

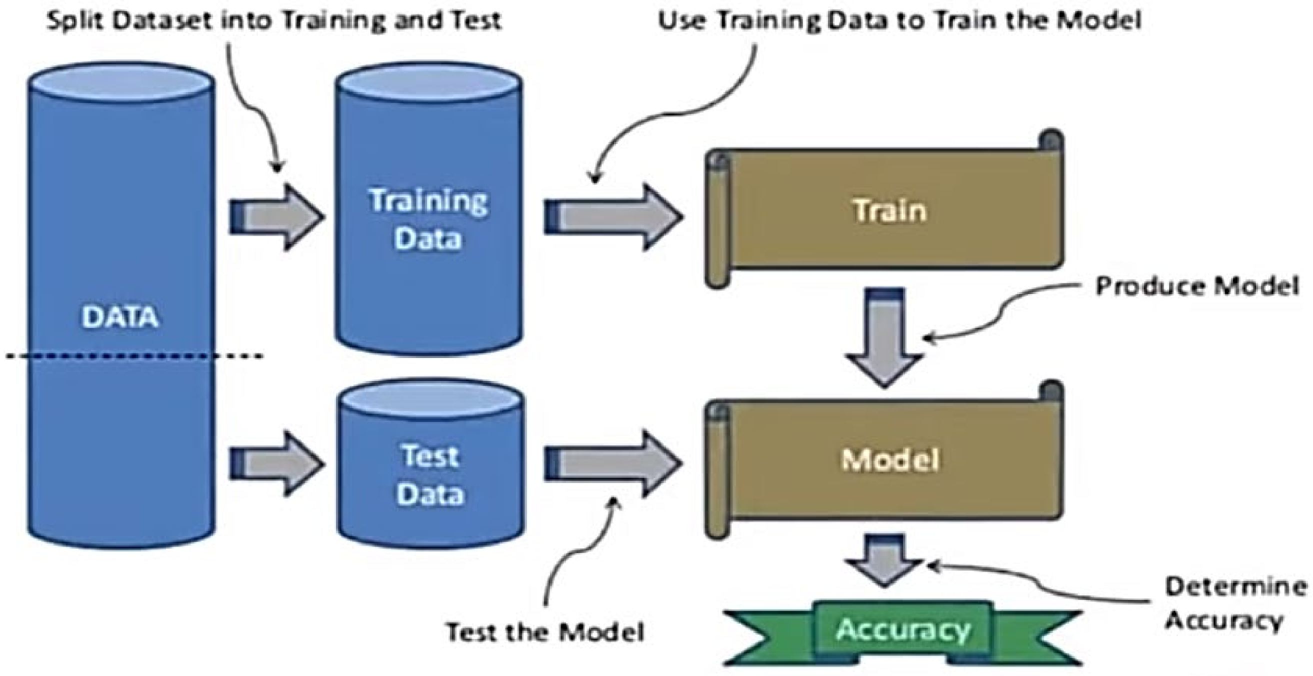
Traditional Vs. Machine Learning Models



Supervised Learning

- In Supervised learning, A model is getting trained on a labelled dataset.
- It is a process of providing input data as well as correct output data, The supervised learning algorithm is to find a mapping function to map the input with the output.

Supervised Learning



Learning a Class from Example

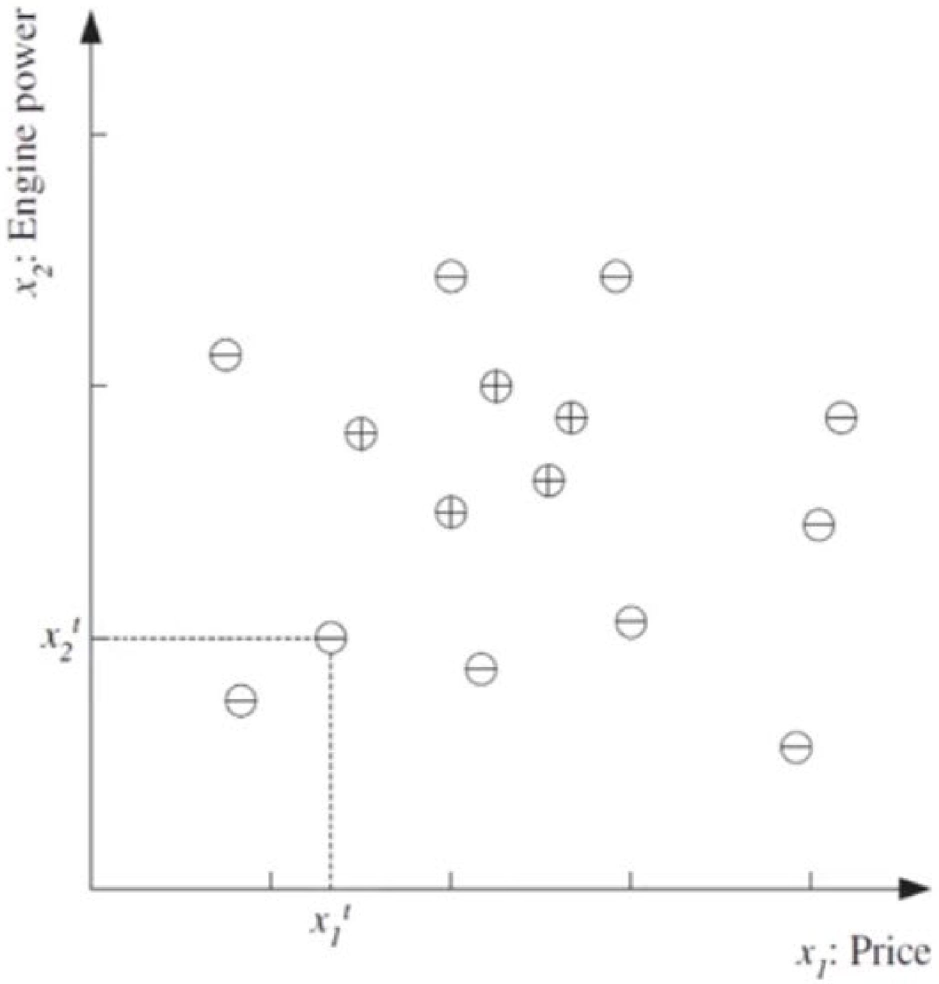
Class-C : “family car.”

Contd…

- Set of cars "Class-C : Family of Cars"
- A group of people look at the cars and label them; family car or not with two attributes the price and engine power.
- The cars that they believe are family cars are positive examples, and the other cars are negative examples.
- People ignore other attributes such as seating capacity and color and consider those of irrelevant.

Training set-Family Car

- The data point corresponds to one sample car
- Coordinates: price and engine power
- ‘+’: positive example of class (a family car),
- ‘–’: negative example (not a family car)



Contd…

Variables ‘x’ and ‘r’

- Price as the first input attribute x1 (e.g., in Rupees)
- Engine power as the second attribute x2 (e.g., engine volume in cubic centi-meters).
- Label denotes its type

$$\boldsymbol{x} = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$

$$r = \begin{cases} 1 & \text{if } \boldsymbol{x} \text{ is a positive example} \\ 0 & \text{if } \boldsymbol{x} \text{ is a negative example} \end{cases}$$

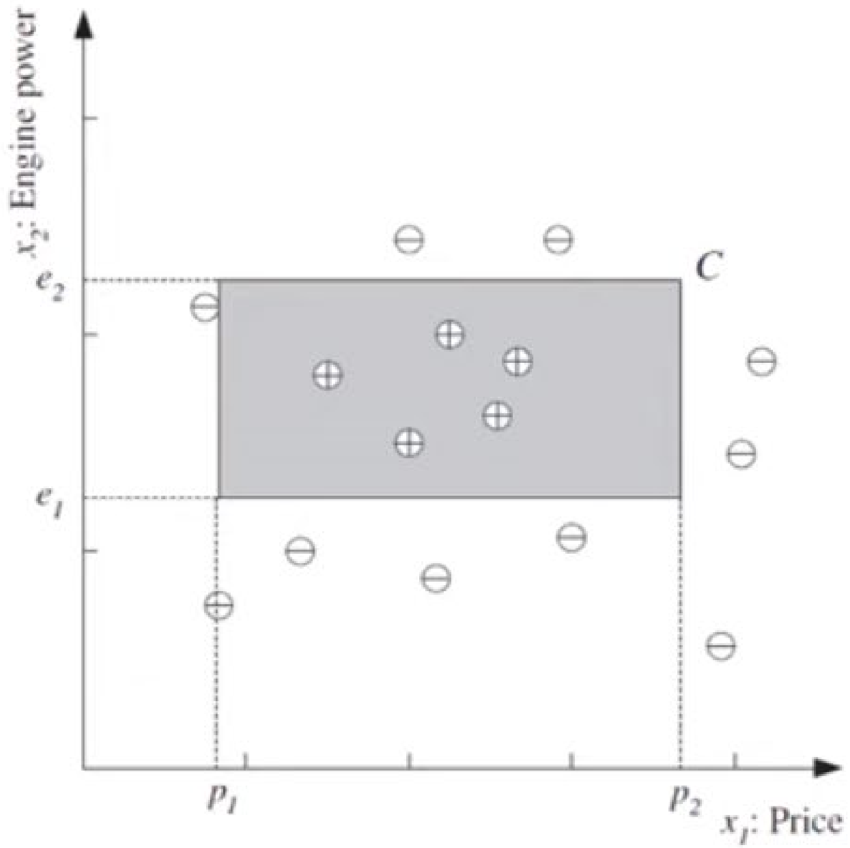
- Each car is represented by such an ordered pair (x, r) and the training set contains N such examples

$$\mathcal{X} = \{\mathbf{x}^t, r^t\}_{t=1}^N$$

- where t indexes the training set.

Example of a Hypothesis class.

- if a car to be a family car, its price and engine power should be in a certain range
- (p1 ≤ price ≤ p2) AND (e1 ≤ engine power ≤ e2) for suitable values of p1, p2, e1, and e2.
- **The class of family car is a rectangle in the price-engine power space.**



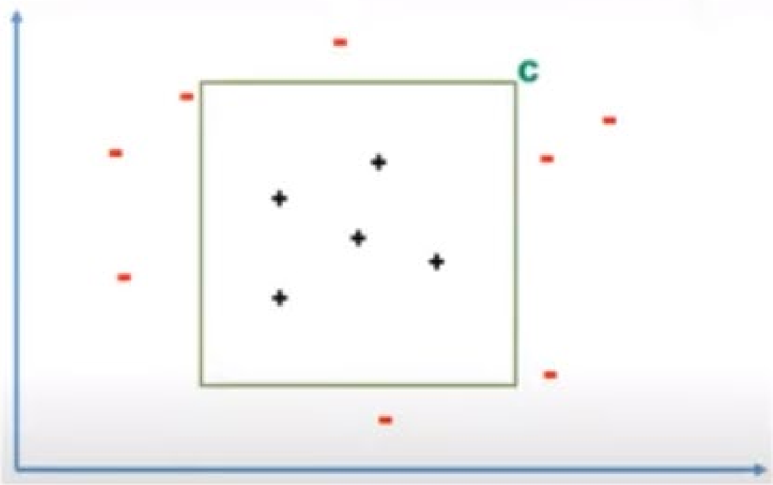
Contd…

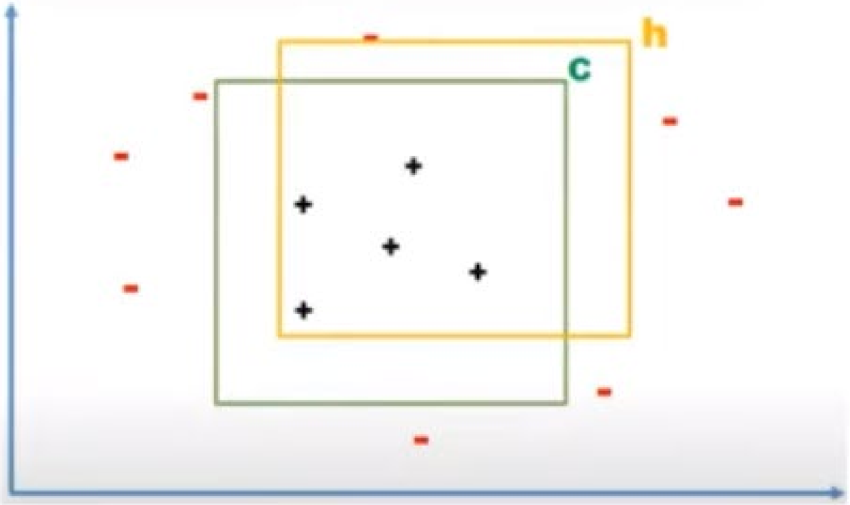
- hypothesis, $h \in H$, specified by a particular quadruple of (p^h1, p^h2, e^h1, e^h2), to approximate C
- the hypothesis h makes a prediction for an instance x such that

$$h(\boldsymbol{x}) = \begin{cases} 1 & \text{if } h \text{ classifies } \boldsymbol{x} \text{ as a positive example} \\ 0 & \text{if } h \text{ classifies } \boldsymbol{x} \text{ as a negative example} \end{cases}$$

Contd…

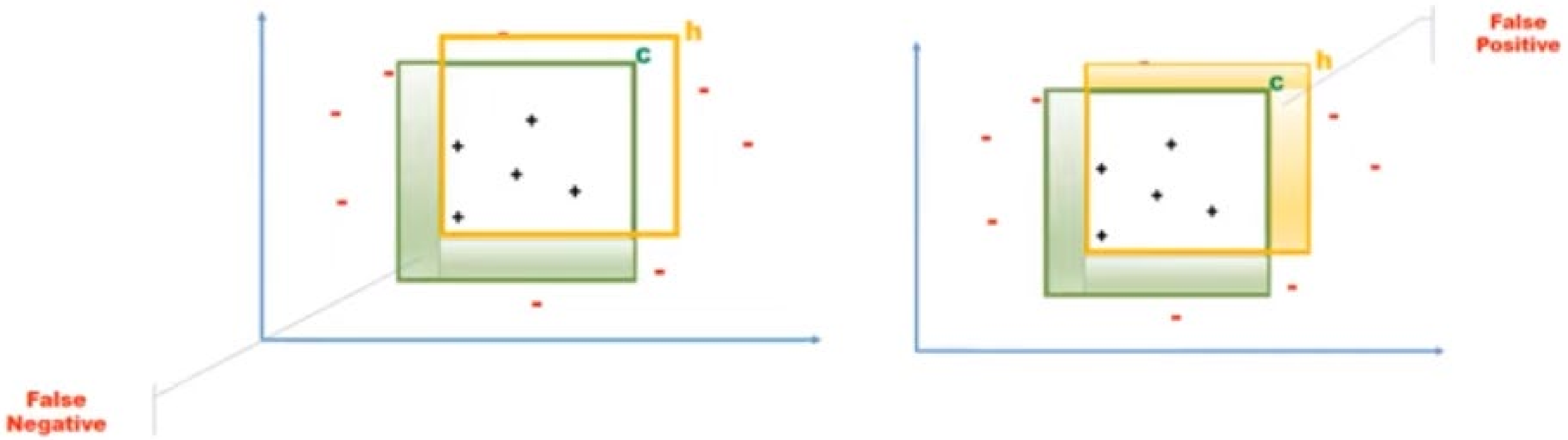
- In real life we do not know C(x), so we cannot evaluate how well h(x) matches C(x).
- C – Target function
- Instances within rectangle represents family cars and outside are not family cars
- Hypothesis h – closely approximate c, and there may be error region.





False Positive and False Negative

- C is the actual class and h is our induced hypothesis.
- The point where C is 1 but h is 0 is a false negative, and
- the point where C is 0 but h is 1 is a false positive.
- true positives and true negatives—are correctly classified.



Contd…

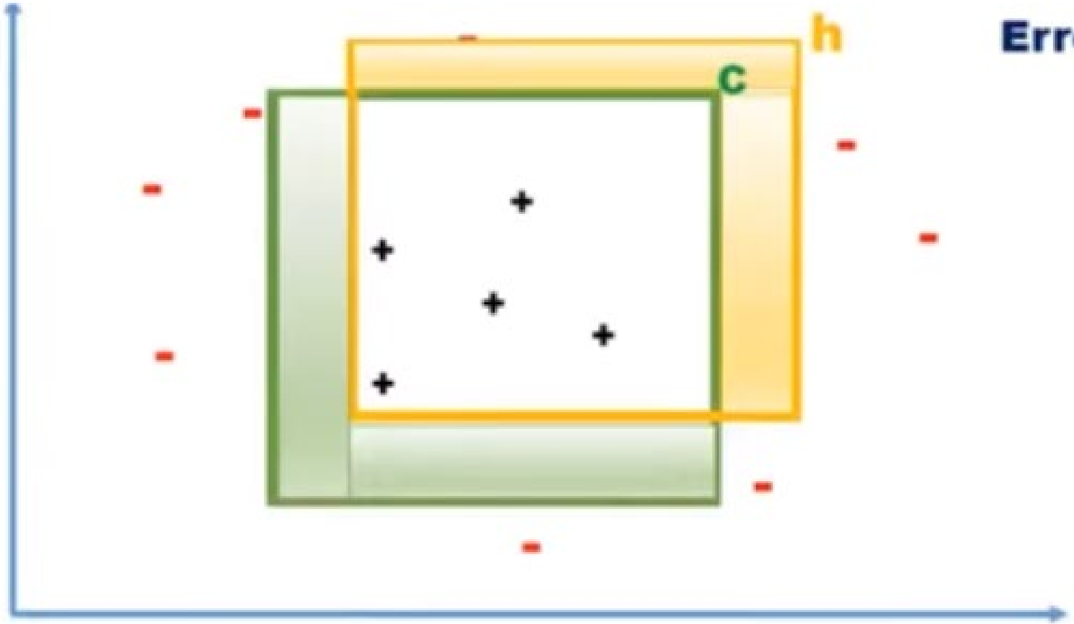
Error

- The empirical error is the proportion of training instances where predictions of h do not match the required values given in X.
- The error of hypothesis h given the training set X is

$$E(h|\mathcal{X}) = \sum_{t=1}^{N} 1(h(\mathbf{x}^t) \neq r^t)$$

- our example, the hypothesis class H is the **set of all possible rectangles.**
- Each quadruple (p^h1, p^h2, e^h1, e^h2), defines one hypothesis, h, from H,

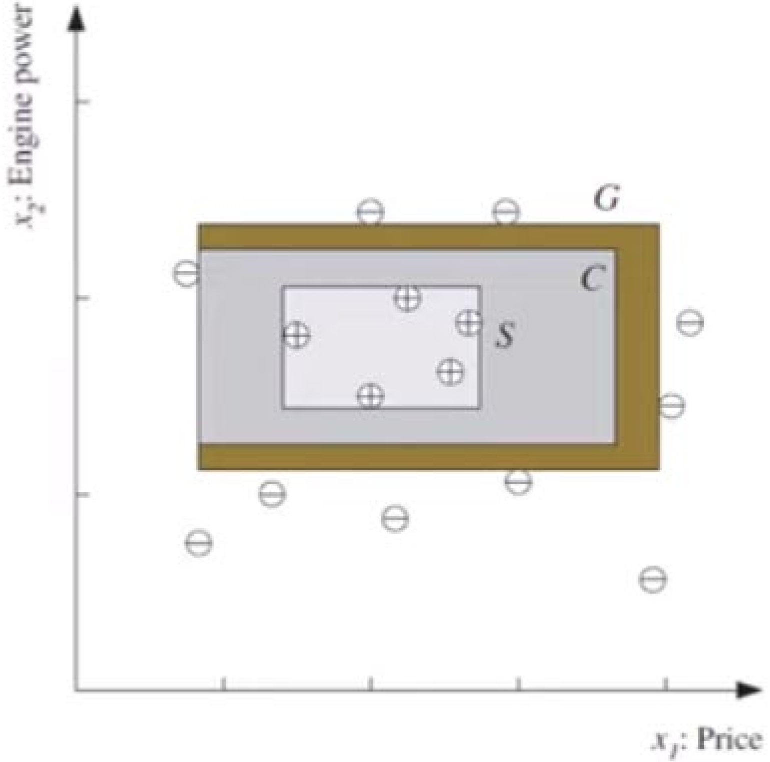
- The probability of error region to be small
- The error region : P(C XOR h) <= Ɛ.



Error Region : C XOR h

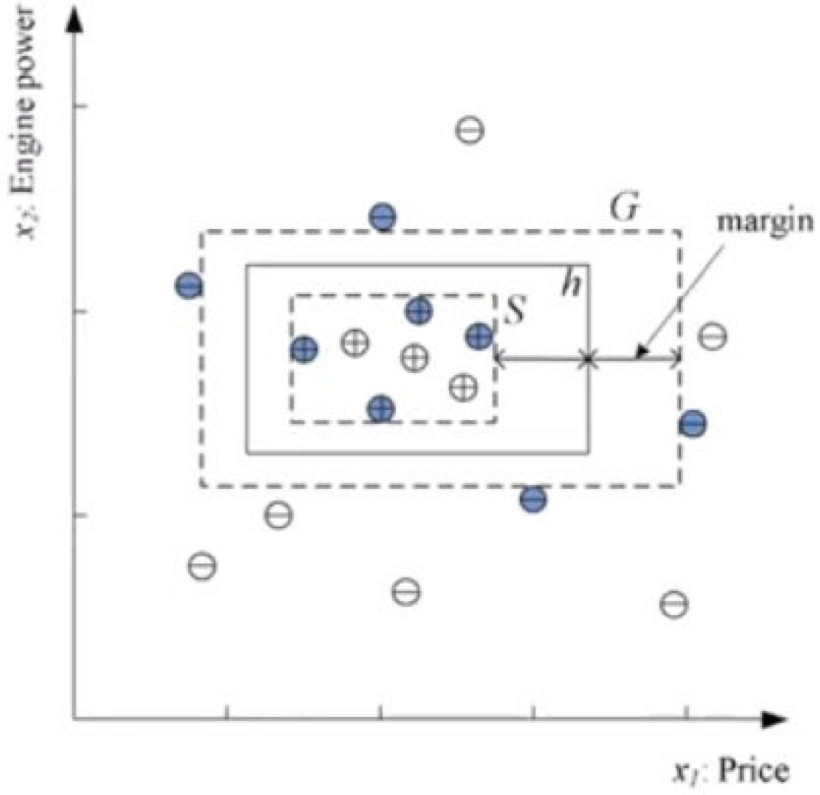
Most General and Most Specific Hypothesis

- Generalization—that is, how well our hypothesis will correctly classify future examples that are not part of the training set.
- Most general hypothesis, G, is the largest rectangle, that includes all the positive examples and none of the negative examples.
- Most specific hypothesis, S, that is the hypothesis tightest rectangle that includes all the positive examples and none of the negative examples.
- Note that the actual class C may be larger than S but is never smaller.



Margin

- The margin, which is the distance between the boundary and the instances closest to it.



Classification Model

- Let us consider two examples, say
  - ‘predicting whether a tumour is malignant or non-malignant’ and
  - ‘price prediction in the domain of real estate’.
- both are the problems related to prediction.

1. The tumour prediction, we are trying to predict which category or class, i.e. ‘malignant’ or ‘non-malignant’, an unknown input data related to tumour belongs to.
2. The price prediction, trying to predict an absolute value and not a class.

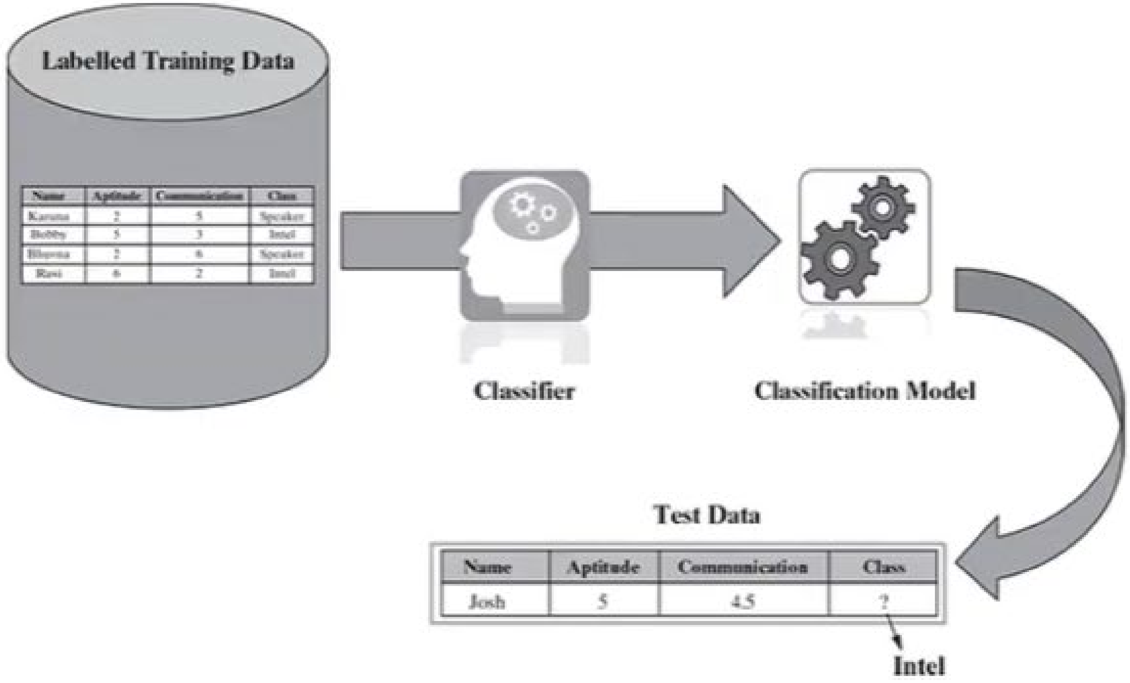
- The problem to predict a categorical or nominal variable, then it is known as a classification problem.

Classification Model

- Classification algorithm is used to identify the category of new data on the basis of training data.
- In classification, a program learn from given dataset (training data) then classify new data (test data) into number of classes or groups.
  - Yes/No, Cat/Dog, Red/Green/Blue, Spam/not spam etc.
- The classes can be called as targets or labels or categories.

Classification Model

- A classification model is obtained from the labelled training data by a classifier algorithm.
- On the basis of the model, a class label (e.g. ‘Intel’ as in the case of the test data) is assigned to the test data.



Classification Model

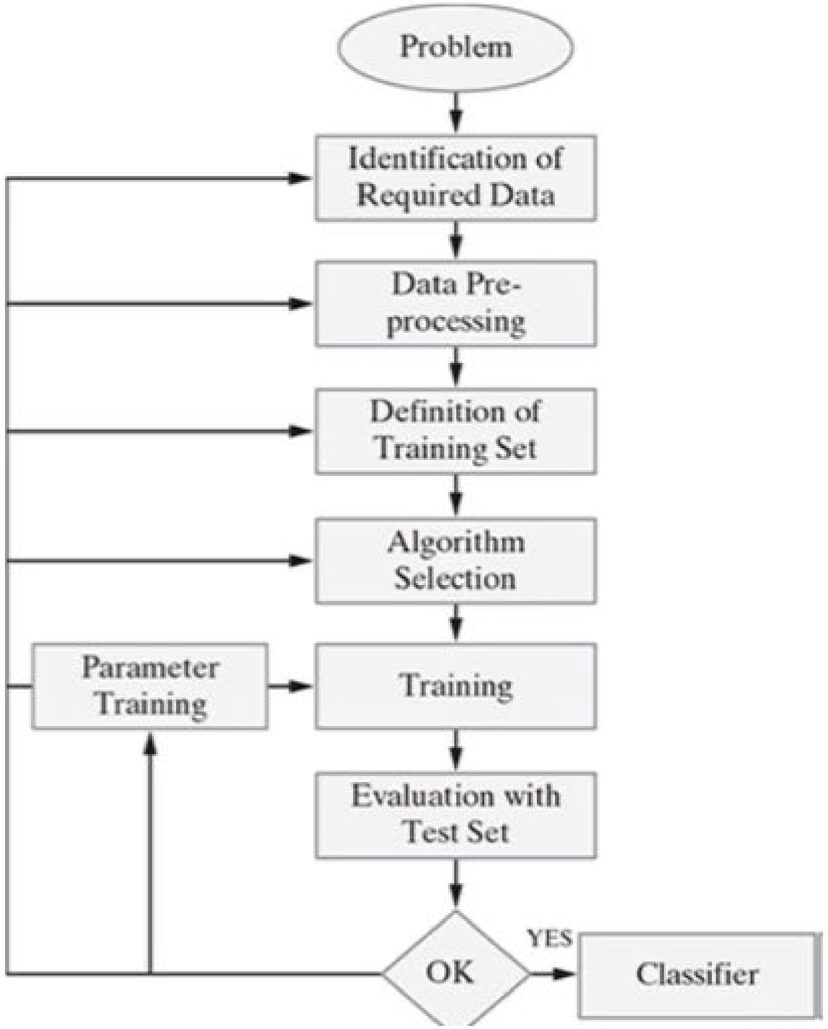
- A critical classification problem in the context of the banking domain is identifying potentially fraudulent transactions.
- Because there are millions of transactions which have to be scrutinized to identify whether a particular transaction might be a fraud transaction,
- it is not possible for any human being to carry out this task.
- Machine learning solves this problem efficiently, on the basis of the past transaction data, labelled as fraudulent, all new incoming transactions are marked or labelled as usual or suspicious.
- The suspicious transactions are subsequently segregated for a closer review.

Classification Model

- Some typical classification problems include the following:
- Image classification
- Disease prediction
- Win–loss prediction of games
- Prediction of natural calamity such as earthquake, flood, etc.
- Handwriting recognition

Classification Learning Steps

1. Problem Identification:
2. Identification of Required Data:
3. Data Pre-processing:
4. Definition of Training Data Set:
5. Algorithm Selection:
6. Training:
7. Evaluation with the Test Data Set:



Classification Learning Steps

Problem Identification

- Identifying the problem is the first step in the supervised learning model.
- The problem needs to be a well-formed problem,
- i.e. a problem with well-defined goals and benefit, which has a long-term impact.

Identification of Required Data:

- On the basis of the problem identified above, the required data set that exactly represents the identified problem needs to be evaluated.
- For example: If the problem is to predict whether a tumour is malignant or not,
- then the corresponding patient data sets related to malignant tumour and normal tumours are to be identified.

Data Pre-processing:

- The data is gathered from different sources, it is usually collected in a raw format and is not ready for immediate analysis.
- Data pre-processing refers to the transformations applied to the identified data before feeding the same into the algorithm.
- This is related to the cleaning/transforming the data set.
- This step ensures that all the unnecessary/irrelevant data elements are removed.
- And the data is ready to be fed into the machine learning algorithm.

Definition of Training Data Set:

- Before starting the analysis, the user should decide what kind of data set is to be used as a training set.
- a set of ‘input meta-objects’ and corresponding ‘output meta-objects’ are gathered.
- Thus, a set of data input (X) and corresponding outputs (Y) is gathered either from human experts or experiments.
- The training set needs to be actively representative of the real-world use of the given scenario.

Classification Learning Steps

Algorithm Selection:

- This is the most critical step of supervised learning model.
- This involves determining the structure of the learning function and the corresponding learning algorithm.
- On the basis of various parameters, the best algorithm for a given problem is chosen.

Training:

- The identified learning algorithm will run on the gathered training set, with the required control parameters as input to the algorithm
- These parameters (inputs given to algorithm) may also be adjusted by optimizing performance on a subset (called as validation set)

Evaluation with the Test Data Set:

- Training data is run on the algorithm, and its performance is measured here.
- If a suitable result is not obtained, further training of parameters may be required.

K-Nearest Neighbor (KNN) Algorithm

- The kNN algorithm is a simple but extremely powerful supervised learning algorithm.
- K-NN algorithm
  - stores all the available data and
  - classifies a new data point based on the similarity.
- The kNN is used in classifications or predictions, the grouping of an individual data point.
- This means when new data appears then it can be easily classified into a well suite Category/Class.

KNN Algorithm

- Input: Training data set, test data set (or data points), value of 'k' (i.e. number of nearest neighbours to be considered)
- Steps:
- Do for all test data points
- Calculate the distance (usually Euclidean distance) of the test data point from the different training data points.
- Find the closest 'k' training data points, i.e. training data points whose distances are least from the test data point.
- If k = 1
- Then assign class label of the training data point to the test data point
- Else
- Whichever class label is mostly present in the training data points, assign that class label to the test data point
- End do

KNN : Example

- Let us try to understand the algorithm with a simple data set.
- A Student data set consists of 15 students, studying in a class.
- Each of the students has been assigned a score on a scale of 10 on two performance parameters –
- 'Aptitude' and 'Communication'.
- Also, a **class value** is assigned to each student based on the following criteria:
- 1. good communication skills & good level of aptitude have been classified as 'Leader'
- 2. good communication skills but not so good level of aptitude have been classified as 'Speaker'
- 3. not so good communication skill but a good level of aptitude have been classified as 'Intel'

Name	Aptitude	Communication	Class
Karuna	2	5	Speaker
Bhuvna	2	6	Speaker
Gaurav	7	6	Leader
Parul	7	2.5	Intel
Dinesh	8	6	Leader
Jani	4	7	Speaker
Bobby	5	3	Intel
Parimal	3	5.5	Speaker
Govind	8	3	Intel
Susant	6	5.5	Leader
Gouri	6	4	Intel
Bharat	6	7	Leader
Ravi	6	2	Intel
Pradeep	9	7	Leader
Josh	5	4.5	Intel

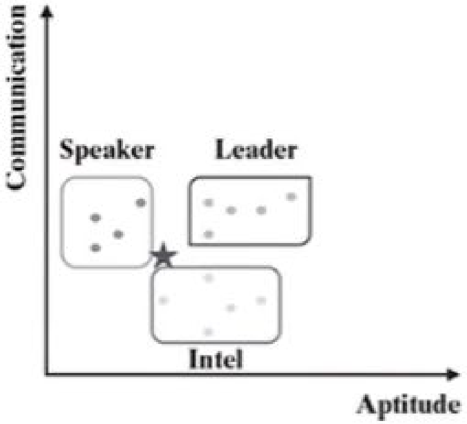
KNN: Example

- To build a classification model, a part of the labelled input data is retained as test data.
- The remaining portion of the input data is used to train the model – hence known as training data.
- The test data is used to evaluate the performance of the model.
- the record of the student named Josh is assumed to be the test data.

	Name	Aptitude	Communication	Class
Training Data	Karuna	2	5	Speaker
	Bhuvna	2	6	Speaker
	Gaurav	7	6	Leader
	Parul	7	2.5	Intel
	Dinesh	8	6	Leader
	Jani	4	7	Speaker
	Bobby	5	3	Intel
	Parimal	3	5.5	Speaker
	Govind	8	3	Intel
	Susant	6	5.5	Leader
	Gouri	6	4	Intel
	Bharat	6	7	Leader
	Ravi	6	2	Intel
	Pradeep	9	7	Leader
Test Data	Josh	5	4.5	Intel

KNN: Example

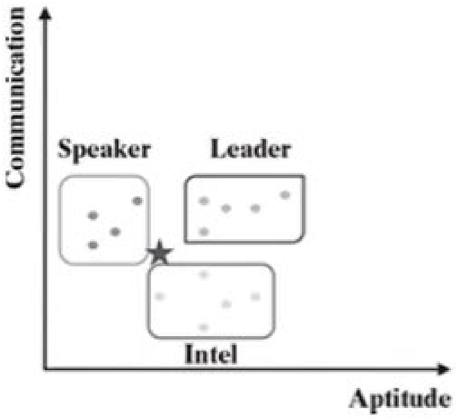
- considering the features 'Aptitude' and 'Communication' can be represented as dots in a two-dimensional feature space.
- the training data points having the same class value are coming close to each other.
- The feature 'Name' is ignored because, as we can understand, it has no role to play in deciding the class value.
- * in the diagram is test data set



Name	Aptitude	Communication	Class
Karuna	2	5	Speaker
Bhuvna	2	6	Speaker
Gaurav	7	6	Leader
Parul	7	2.5	Intel
Dinesh	8	6	Leader
Jani	4	7	Speaker
Bobby	5	3	Intel
Parimal	3	5.5	Speaker
Govind	8	3	Intel
Susant	6	5.5	Leader
Gouri	6	4	Intel
Bharat	6	7	Leader
Ravi	6	2	Intel
Pradeep	9	7	Leader
Josh	5	4.5	???

KNN: Example

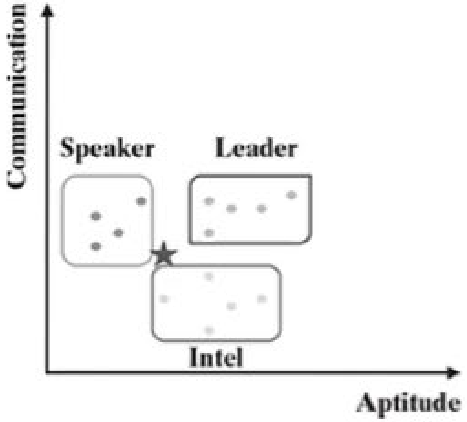
- To find the nearest neighbours of the test data point, Euclidean distance of the different dots need to be calculated from the asterisk.
- If k = 1, only the closest training data element is considered.
- If k=3, only three nearest neighbours or three training data elements closest to the test data element are considered.
- The class label of that data element is directly assigned to the test data element.



Name	Aptitude	Communication	Class	Distance	$k = 1$	$k = 2$	$k = 3$
Karuna	2	5	Speaker	3.041			
Bhuvna	2	6	Speaker	3.354			
Parimal	3	5.5	Speaker	2.236			
Jani	4	7	Speaker	2.693			
Bobby	5	3	Intel	1.500			1.500
Ravi	6	2	Intel	2.693			
Gouri	6	4	Intel	1.118	1.118	1.118	1.118
Parul	7	2.5	Intel	2.828			
Govind	8	3	Intel	3.354			
Susant	6	5.5	Leader	1.414			
Bharat	6	7	Leader	2.693			
Gaurav	7	6	Leader	2.500			
Dinesh	8	6	Leader	3.354			
Pradeep	9	7	Leader	4.717			
Josh	5	4.5	???				

KNN: Example

- Gouri and Bobby have class value 'Intel', while Susant has class value 'Leader'.
- In this case, the class value of Josh is decided by **majority voting.**
- Because the class value of 'Intel' is formed by the majority of the neighbours, the class value of Josh is assigned as 'Intel'.
- This same process can be extended for any value of k.



Name	Aptitude	Communication	Class	Distance	$k = 1$	$k = 2$	$k = 3$
Karuna	2	5	Speaker	3.041			
Bhuvna	2	6	Speaker	3.354			
Parimal	3	5.5	Speaker	2.236			
Jani	4	7	Speaker	2.693			
Bobby	5	3	Intel	1.500			1.500
Ravi	6	2	Intel	2.693			
Gouri	6	4	Intel	1.118	1.118	1.118	1.118
Parul	7	2.5	Intel	2.828			
Govind	8	3	Intel	3.354			
Susant	6	5.5	Leader	1.414			
Bharat	6	7	Leader	2.693			
Gaurav	7	6	Leader	2.500			
Dinesh	8	6	Leader	3.354			
Pradeep	9	7	Leader	4.717			
Josh	5	4.5	???				

KNN: Selecting the K Value

- If the value of k is very large (in the extreme case equal to the total number of records in the training data), the class label of the majority class of the training data set will be assigned to the test data regardless of the class labels of the neighbours nearest to the test data.
- If the value of k is very small (in the extreme case equal to 1), the class value of a noisy data or outlier in the training data set which is the nearest neighbour to the test data will be assigned to the test data.
- The best k value is somewhere between these two extremes.

KNN: Selecting the K Value

- Few strategies are adopted by machine learning practitioners to arrive at a value for k.
- 1. k equal to the square root of the number of training records.
- 2. test several k values on a variety of test data sets and choose the one that delivers the best performance.
- 3. choose a larger value of k, but apply a weighted voting process in which the vote of close neighbours is considered more influential than the vote of distant neighbours.

KNN: A Lazy Learner

- The eager learners follow the general steps of machine learning,
- i.e. perform an abstraction of the information obtained from the input data and then follow it through by a generalization step.
- In the kNN algorithm, these steps are completely skipped.
- It stores the training data and directly applies the philosophy of nearest neighbourhood finding to arrive at the classification.
- there is no learning happening in the real sense.
- Therefore, kNN falls under the category of lazy learner.

KNN: Advantages and Disadvantages

- Strengths
- Extremely simple algorithm – easy to understand
- Very effective in certain situations,
- Very fast or almost no time required for the training phase
- Weaknesses
- Does not learn anything in the real sense.
- Classification is done completely on the basis of the training data.
- If the training data does not represent the problem domain systematically, the algorithm fails to make an effective classification.
- Also, a large amount of computational space is required to load the training data for classification.

KNN: Applications

- The recommender systems, recommend users, with different items which are similar to a particular item that the user seems to like.
- The liking pattern may be revealed from past purchases or browsing history and the similar items are identified using the kNN algorithm.
- Information retrieval (concept search), Searching documents/ contents similar to a given document/content.

Next Class

- Decision Tree
- Random Forest
- Regression
- Logistics Regression

Thank you for your participation.

For any clarification write to:

tamal.ghosh1@adamasuniversity1.ac.in