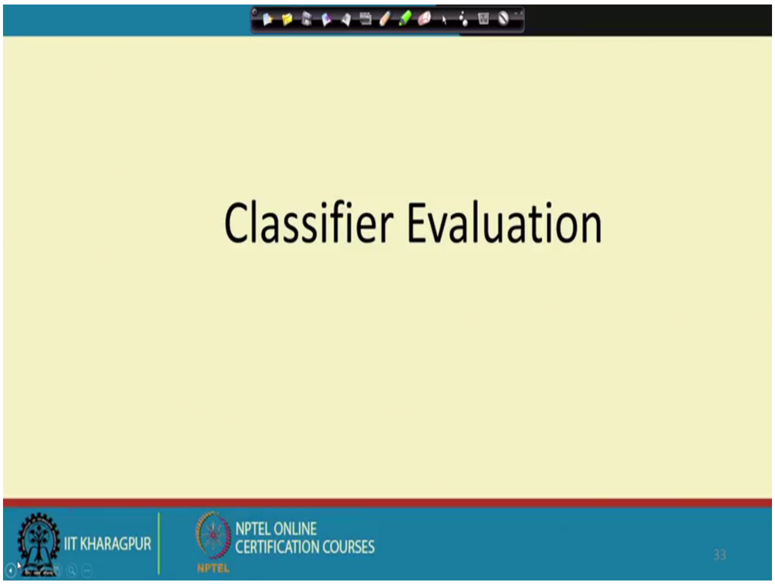
Data Mining
Prof. Pabitra Mitra
Department of Computer Science & Engineering
Indian Institute of Technology, Kharagpur

Lecture – 21
K - Nearest Neighbor – V

In this lecture we discuss; how do we evaluate the classification algorithms, we have designed so far.

(Refer Slide Time: 00:23)

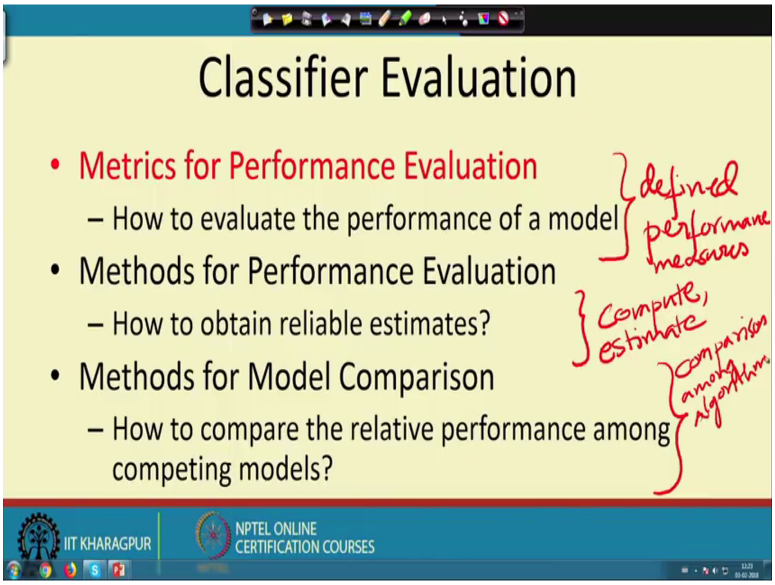


Classifier Evaluation

So, we have looked at 3 classification algorithms. So, far one was the decision tree algorithm, one was the Bayesian classifier and then we studied the nearest neighbor classifier for some value of k. So, now, the question is that how do we know that which classifier to choose, which one is better which one should choose.

It turns out that depending on the problem you are solving some classify there is no classifier which is good for all the problems, some problems some classifier is good, some other problems some other classifier is good. So, it is very important to have quantitative or objective measures of telling how good a classifier is. So, that is what we will discuss in this talk.

(Refer Slide Time: 01:31)



Classifier Evaluation

- Metrics for Performance Evaluation
  - How to evaluate the performance of a model
- Methods for Performance Evaluation
  - How to obtain reliable estimates?
- Methods for Model Comparison
  - How to compare the relative performance among competing models?

defined performance measures

compute, estimate

comparison among algorithm

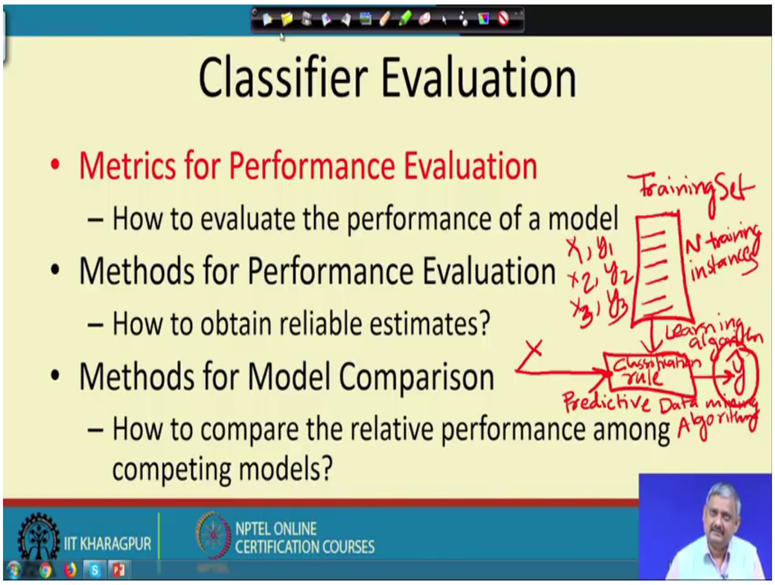
So, we will discuss 3 aspects, one is we define some equation, some metric which given a classifier n a, n a problem it will tell how good the classifier is at that problem it will give some value. So, different type of ways we can evaluate. So, for example, suppose you are a student has given the exam and he has got the so many subjects are there and he has got this amount of marks in each these percentage of marks in each of the subjects how do you tell, do you consider the average of all marks do you consider the highest of all marks and so on. So, different ways you can evaluate.

Second is once you have defined the performance measures how do I compute them rather how do I estimate their value in a faithful manner. So, one is say for example, when you are evaluating a student, one is saying what defines how good a student is, second is do you take actually tests maybe unit tests and say maybe 5 test an year and the at consider the marks of each of these states to properly estimate these measures that we have defined.

And the second is now given these measures and their estimates how do you compare 2 algorithms. So, among algorithms this is actually not an easy problem because maybe with respect to some measure one algorithm is better than the other. So, maybe in maths one student is better than the has got more marks than the other, but maybe in history another measure some other that is that student has got more marks than these. So, if there are multiple measures, it is difficult to say that which algorithm among in this and some aspects some algorithm is good other aspect of some other algorithm is good. So, how do you compare that is what we discuss in this second aspect, third aspect ok.

So, before that let me tell you what is a classification talks basically. So, what you have is let me make the picture clear for what you have.

(Refer Slide Time: 05:00)



Classifier Evaluation

- Metrics for Performance Evaluation
  - How to evaluate the performance of a model
- Methods for Performance Evaluation
  - How to obtain reliable estimates?
- Methods for Model Comparison
  - How to compare the relative performance among competing models?

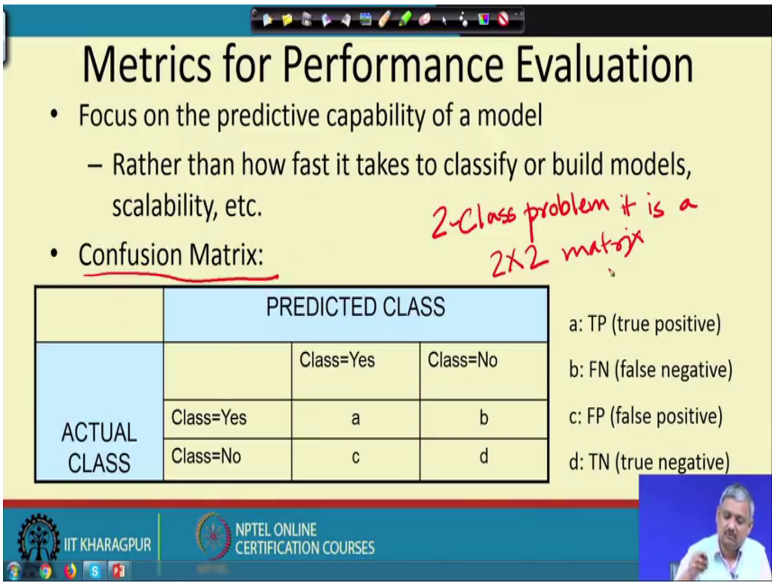
So, you have a training set on n training instances and you have a learning algorithm say a Bayesian classifier or something which gives you a classification rule, gives you a classification rule. What does the classification rule do, if you take in a new instance x whose y is not known note that in the training instance I have a set of x whose class levels I call y is already known supervised learning, we know the class levels of these instances ok.

And then what this classifier does is that it predicts the class level of this new example x. So, it predicts y let me call it as y hat predicts y. So, these algorithms that is why you are also sometimes called predictive data mining algorithms the predict. Now, what this performance evaluation matrix do is that it evaluates this classification rule in terms of how good this prediction matches the actual value say for example, I am predicting a 2 class problem maybe I am predicting tomorrow there will be rain or there would not be rain there will be rain or there would not be rain.

So, there are 2 classes rain or no rain and my rule say predicts rain tomorrow, today it predicts the tomorrow it will rain. I will say my prediction is correct it has done a correct classification, if it matches tomorrows fact that is tomorrow it has rained or not and I will say it is misclassified if it does not match; so, that is the main intuition.

So, we basically find and these what I do I do not do for one instances, suppose goodness will be not just for a single instance even though I will need it only for a single instance is for if you do it 100 times how many times it matches the actual prediction in what percentage of the time it matches the actual prediction.

(Refer Slide Time: 09:06)



Metrics for Performance Evaluation

- Focus on the predictive capability of a model
  - Rather than how fast it takes to classify or build models, scalability, etc.
- Confusion Matrix:

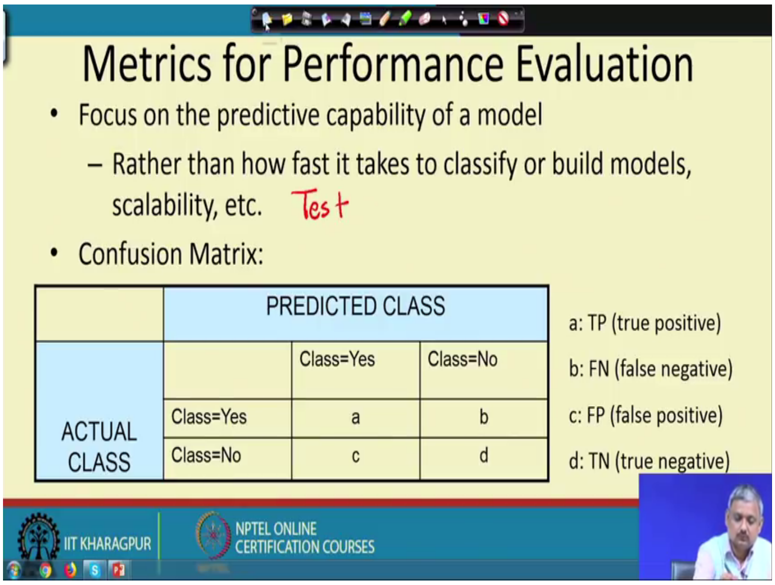
2-Class problem it is a 2X2 matrix

		PREDICTED CLASS	
		Class=Yes	Class=No
ACTUAL CLASS	Class=Yes	a	b
	Class=No	c	d

a: TP (true positive)
b: FN (false negative)
c: FP (false positive)
d: TN (true negative)

So, one way, one step to compute this matrix is to construct what is called a confusion matrix, what is called a confusion matrix. So, what is a confusion matrix? It is a suppose for a 2 class problem, 2 class problem it is a. So, it is a 2 by 2 matrix for a 2 class problem. So, how do I get the matrix, I get the matrix in the following way what I do is that.

(Refer Slide Time: 10:37)



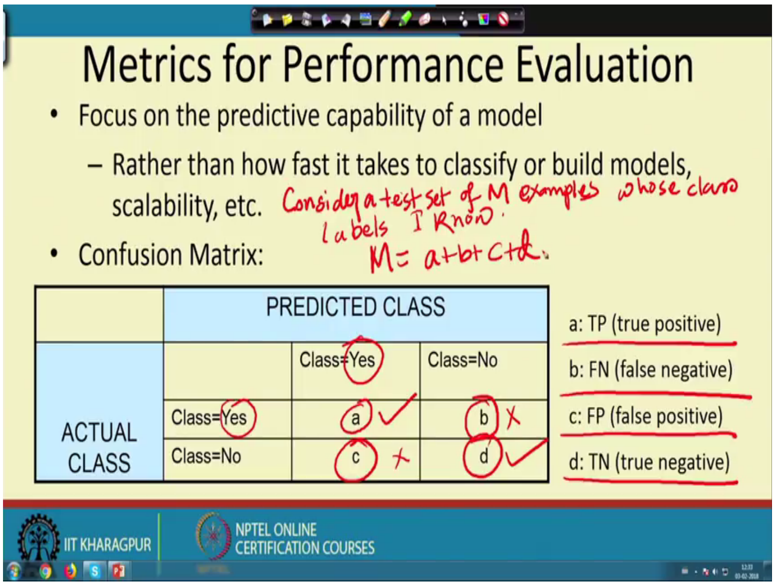
Metrics for Performance Evaluation

- Focus on the predictive capability of a model
  - Rather than how fast it takes to classify or build models, scalability, etc. Test
- Confusion Matrix:

	PREDICTED CLASS		
		Class=Yes	Class=No
ACTUAL CLASS	Class=Yes	a	b
	Class=No	c	d

a: TP (true positive)
b: FN (false negative)
c: FP (false positive)
d: TN (true negative)

(Refer Slide Time: 10:46)



Metrics for Performance Evaluation

- Focus on the predictive capability of a model
  - Rather than how fast it takes to classify or build models, scalability, etc. Considr a test set of M examples whose class labels I know.
- Confusion Matrix: $M = a+b+c+d$.

	PREDICTED CLASS		
		Class=Yes	Class=No
ACTUAL CLASS	Class=Yes	a ✓	b ✗
	Class=No	c ✗	d ✓

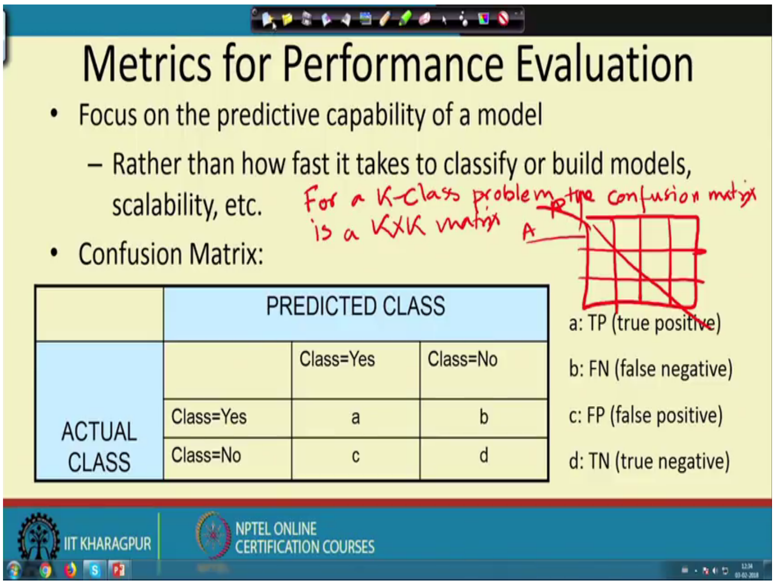
a: TP (true positive)
b: FN (false negative)
c: FP (false positive)
d: TN (true negative)

Of consider a test set of M examples whose class levels I know and let us say a number out of this M examples actually belong to the first class let me call it as a yes class and also predicted to be yes class.

So, a out of M, a out of m and this number I would call it as a true positive, actually positive also predicted as positive these actually are it is a it is the most desirable case then we consider b which is actually positive classified as negative. Actually positive classified as negative, I call this a false negative similarly c is false positive and d which is true negative. So, it is this; a and d which are correct classifications, these and these are two types of errors, you can also easily think that m equals some of this equals the sum of this.

So, this is the fundamental. So, how many these we can easily extend, like I can.

(Refer Slide Time: 13:56)



Metrics for Performance Evaluation

- Focus on the predictive capability of a model
  - Rather than how fast it takes to classify or build models, scalability, etc.
- Confusion Matrix:

	PREDICTED CLASS		
		Class=Yes	Class=No
ACTUAL CLASS	Class=Yes	a	b
	Class=No	c	d

a: TP (true positive)

b: FN (false negative)

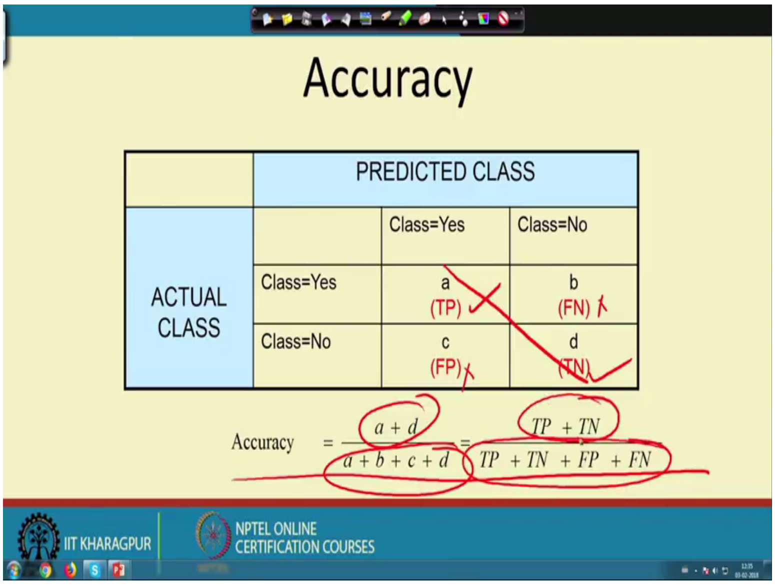
c: FP (false positive)

d: TN (true negative)

For a K-class problem the confusion matrix is a KxK matrix

Like this is actual, this is actual and this is predicted along the rows actual and along the columns predicted. So, we have class 1, class 2 and so on and so on. So, I can extend this concept and the diagonal is the correct classify the correctness. So, this is the fundamental thing.

(Refer Slide Time: 15:04)



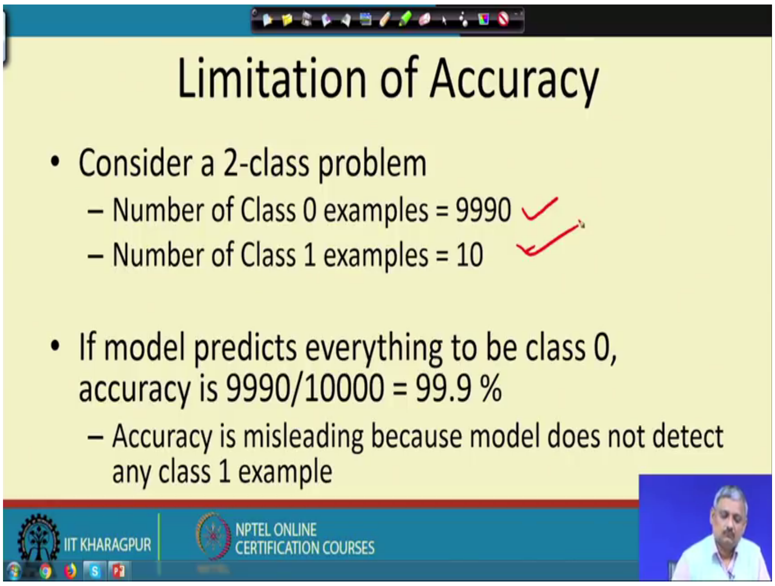
Accuracy

		PREDICTED CLASS	
		Class=Yes	Class=No
ACTUAL CLASS	Class=Yes	a (TP)	b (FN)
	Class=No	c (FP)	d (TN)

$$\text{Accuracy} = \frac{a+d}{a+b+c+d} = \frac{TP + TN}{TP + TN + FP + FN}$$

Now, using this confusion matrix I can define several matrix. So, using this true positive, true negative, false positive. false negative I can define several measures, the most common is accuracy. So, it is the sum of the diagonal elements divided by total number of elements this by this fine, you accept that this is a good measure important measure what else. So, one I am just defining it is very important measure.

(Refer Slide Time: 16:18)

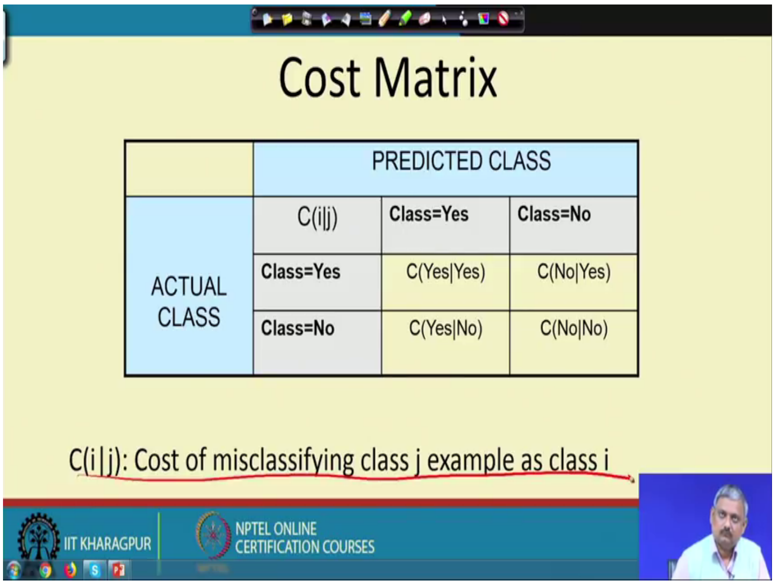


Limitation of Accuracy

- Consider a 2-class problem
  - Number of Class 0 examples = 9990 ✓
  - Number of Class 1 examples = 10 ✓
- If model predicts everything to be class 0, accuracy is 9990/10000 = 99.9 %
  - Accuracy is misleading because model does not detect any class 1 example

But only this is sometimes misleading, take this example if the classes are imbalanced sorry this may happen this is clear.

(Refer Slide Time: 16:54)



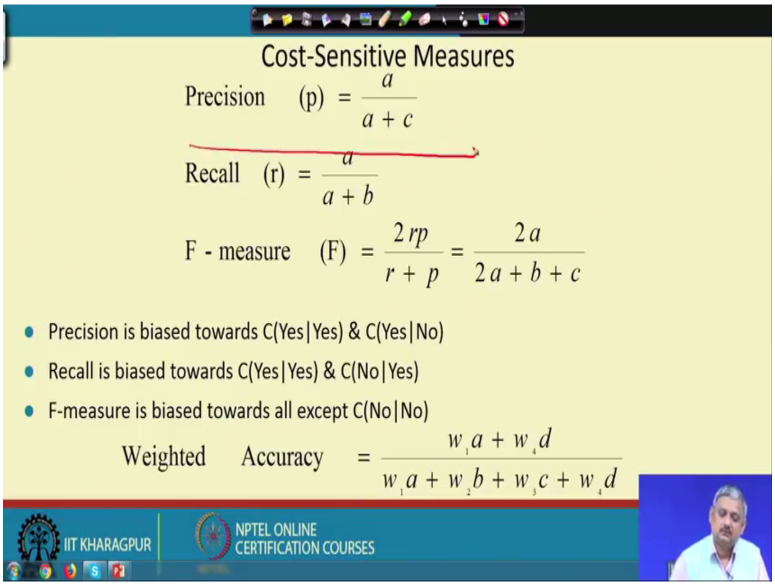
Cost Matrix

	PREDICTED CLASS		
ACTUAL CLASS	C(i|j)	**Class=Yes**	**Class=No**
	Class=Yes	C(Yes|Yes)	C(No|Yes)
	Class=No	C(Yes|No)	C(No|No)

C(i|j): Cost of misclassifying class j example as class i

So, I introduced another matrix called a cost matrix cost of each type of error. So, c is the cost, this is what I want.

(Refer Slide Time: 17:32)



Cost-Sensitive Measures

$$\text{Precision} \quad (p) = \frac{a}{a+c}$$

$$\text{Recall} \quad (r) = \frac{a}{a+b}$$

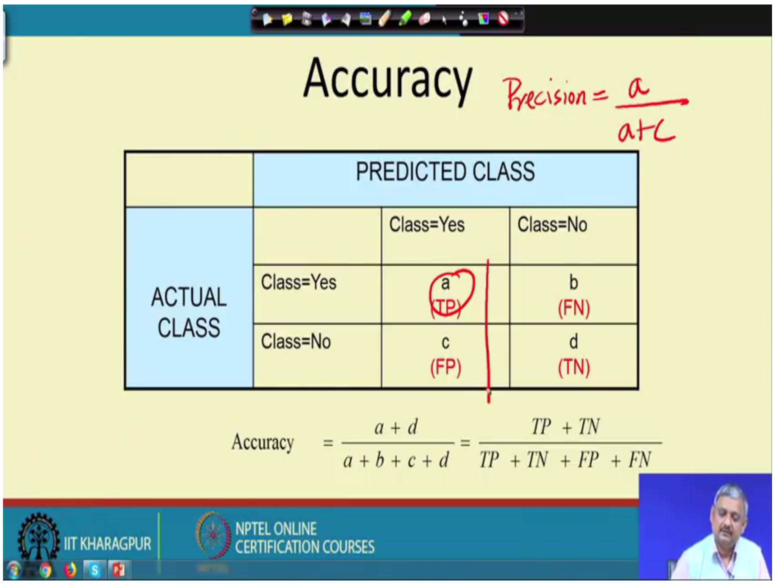
$$\text{F - measure} \quad (F) = \frac{2rp}{r+p} = \frac{2a}{2a+b+c}$$

- Precision is biased towards C(Yes|Yes) & C(Yes|No)
- Recall is biased towards C(Yes|Yes) & C(No|Yes)
- F-measure is biased towards all except C(No|No)

$$\text{Weighted} \quad \text{Accuracy} = \frac{w_1 a + w_4 d}{w_1 a + w_2 b + w_3 c + w_4 d}$$

So, now, I can have several cost sensitive measures using the cost matrix as well as the confusion matrix many you may assume uniform cost also precision, if you go back it is.

(Refer Slide Time: 18:11)



Accuracy

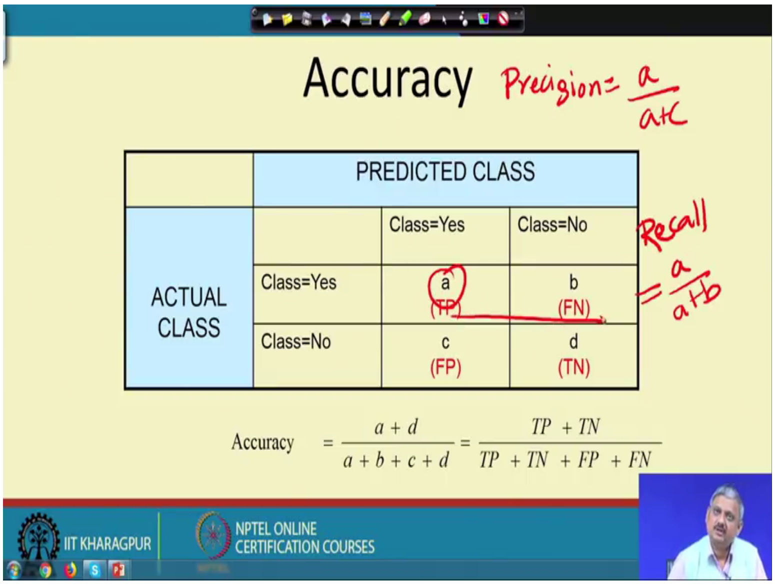
Precision = $\frac{a}{a+c}$

	PREDICTED CLASS	
ACTUAL CLASS	Class=Yes	Class=No
Class=Yes	a (TP)	b (FN)
Class=No	c (FP)	d (TN)

$$\text{Accuracy} = \frac{a+d}{a+b+c+d} = \frac{TP+TN}{TP+TN+FP+FN}$$

This by this ok.

(Refer Slide Time: 18:53)



Accuracy

Precision = $\frac{a}{a+c}$

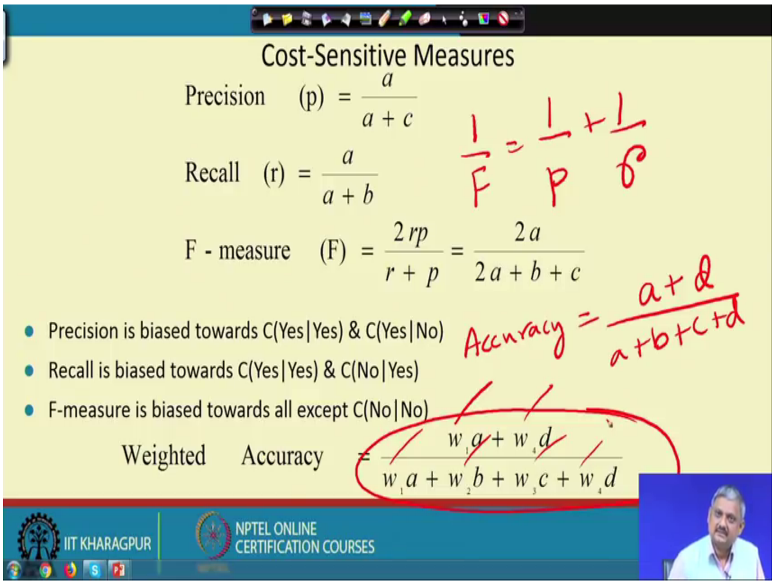
	PREDICTED CLASS	
ACTUAL CLASS	Class=Yes	Class=No
Class=Yes	a (TP)	b (FN)
Class=No	c (FP)	d (TN)

Recall = $\frac{a}{a+b}$

$$\text{Accuracy} = \frac{a+d}{a+b+c+d} = \frac{TP+TN}{TP+TN+FP+FN}$$

Similarly, you have recall which is a by a plus b. So, these are 2 measures commonly used, we have a combination of precision recall called F measure, it is the harmonic mean, it is the harmonic mean.

(Refer Slide Time: 19:41)



Cost-Sensitive Measures

$$\text{Precision} \quad (p) = \frac{a}{a+c}$$

$$\text{Recall} \quad (r) = \frac{a}{a+b}$$

$$\text{F - measure} \quad (F) = \frac{2rp}{r+p} = \frac{2a}{2a+b+c}$$

- Precision is biased towards C(Yes|Yes) & C(Yes|No)
- Recall is biased towards C(Yes|Yes) & C(No|Yes)
- F-measure is biased towards all except C(No|No)

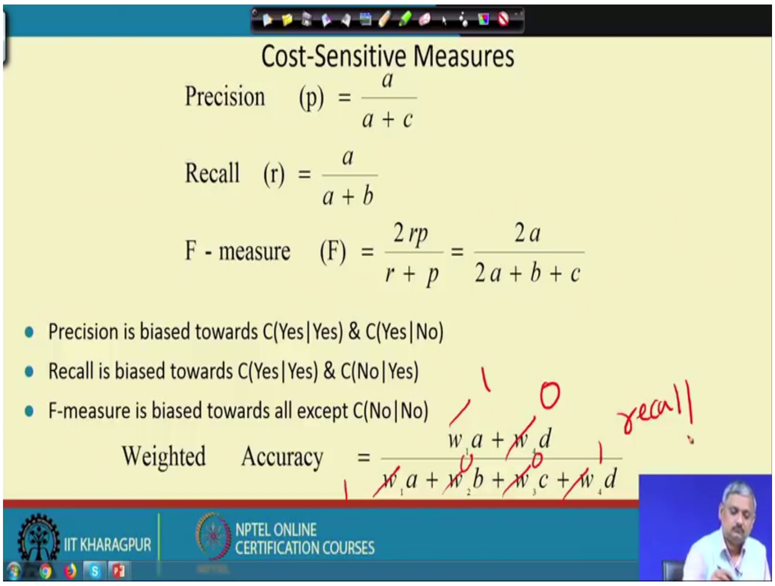
$$\text{Weighted Accuracy} = \frac{w_1 a + w_4 d}{w_1 a + w_2 b + w_3 c + w_4 d}$$

$$\frac{1}{F} = \frac{1}{p} + \frac{1}{r}$$

$$\text{Accuracy} = \frac{a+d}{a+b+c+d}$$

in general you can have a weighted accuracy, remember the cost matrix where have a weight as costs and. So, just accuracy is w 1, w 2; w all these are 0 one all these are equal and 1 accuracy.

(Refer Slide Time: 20:42)



Cost-Sensitive Measures

$$\text{Precision} \quad (p) = \frac{a}{a+c}$$

$$\text{Recall} \quad (r) = \frac{a}{a+b}$$

$$\text{F - measure} \quad (F) = \frac{2rp}{r+p} = \frac{2a}{2a+b+c}$$

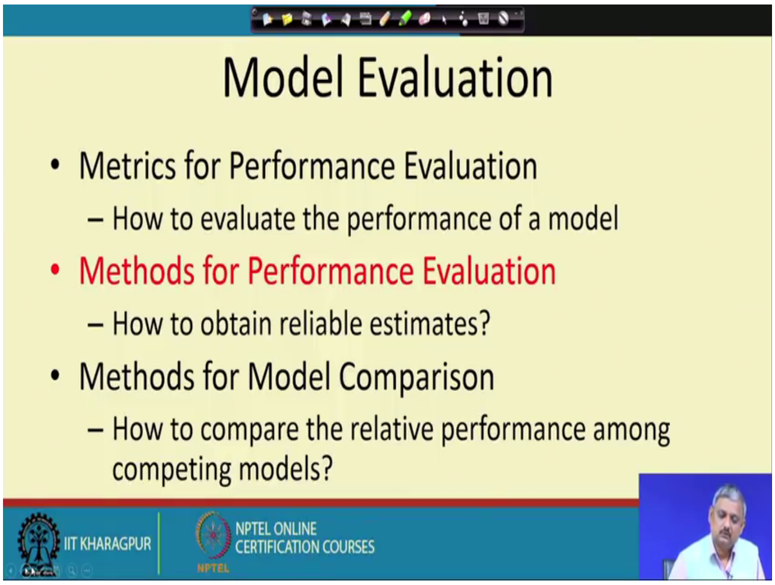
- Precision is biased towards C(Yes|Yes) & C(Yes|No)
- Recall is biased towards C(Yes|Yes) & C(No|Yes)
- F-measure is biased towards all except C(No|No)

$$\text{Weighted Accuracy} = \frac{w_1 a + w_4 d}{w_1 a + w_2 b + w_3 c + w_4 d}$$

1 0 recall

If this is equal to 1, this is equal to 0 then weighted accuracy equal to precision equal to precision. Similarly, if this is equal to one this equal to 0 and this is equal to 1, 0, 0, 1 you have recall.

(Refer Slide Time: 21:31)

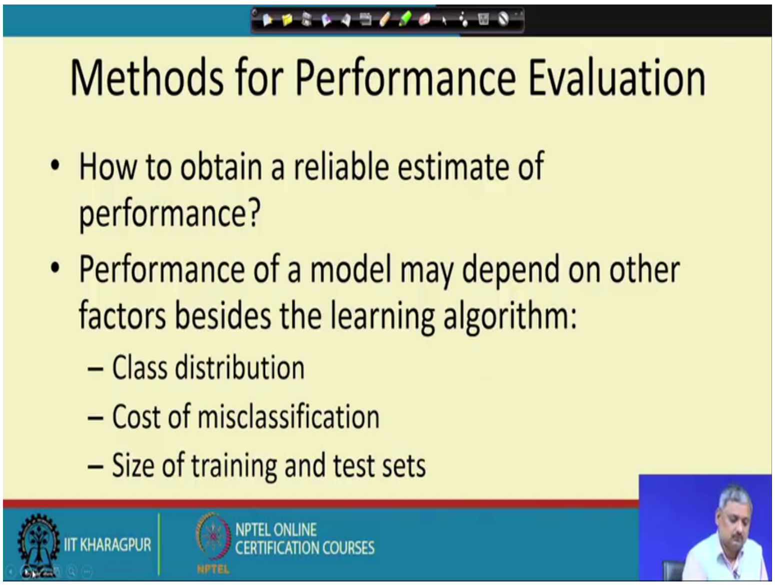


Model Evaluation

- Metrics for Performance Evaluation
  - How to evaluate the performance of a model
- Methods for Performance Evaluation
  - How to obtain reliable estimates?
- Methods for Model Comparison
  - How to compare the relative performance among competing models?

Now, the next question how to get a estimate of this accuracy precision and so on; what size training set you test set you need?

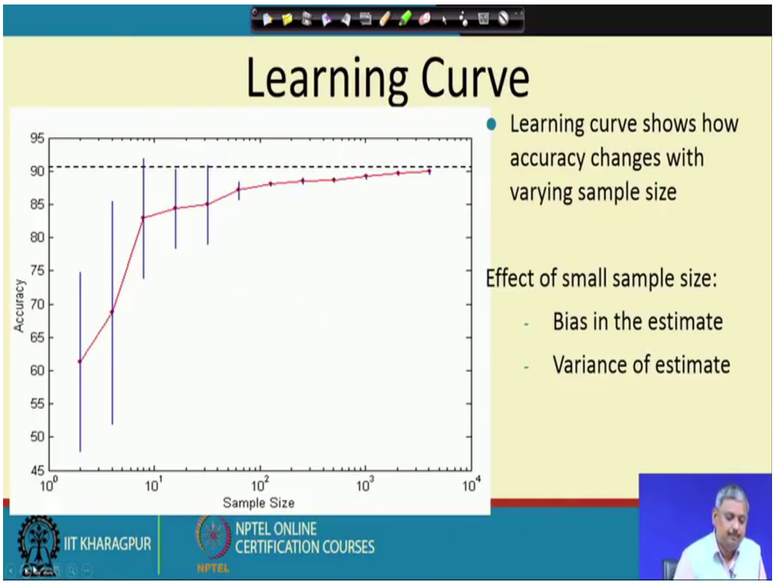
(Refer Slide Time: 21:51)



Methods for Performance Evaluation

- How to obtain a reliable estimate of performance?
- Performance of a model may depend on other factors besides the learning algorithm:
  - Class distribution
  - Cost of misclassification
  - Size of training and test sets

(Refer Slide Time: 21:53)



Learning Curve

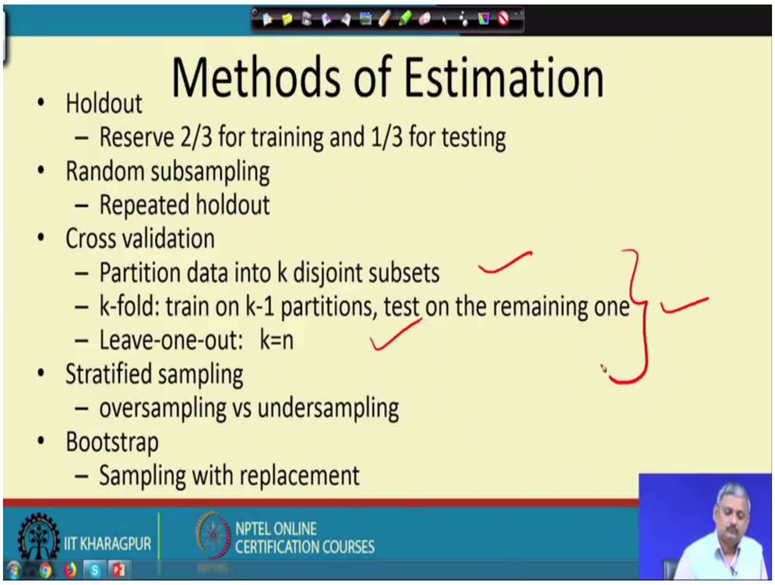
- Learning curve shows how accuracy changes with varying sample size

Effect of small sample size:

- Bias in the estimate
- Variance of estimate

These are the methods.

(Refer Slide Time: 22:04)



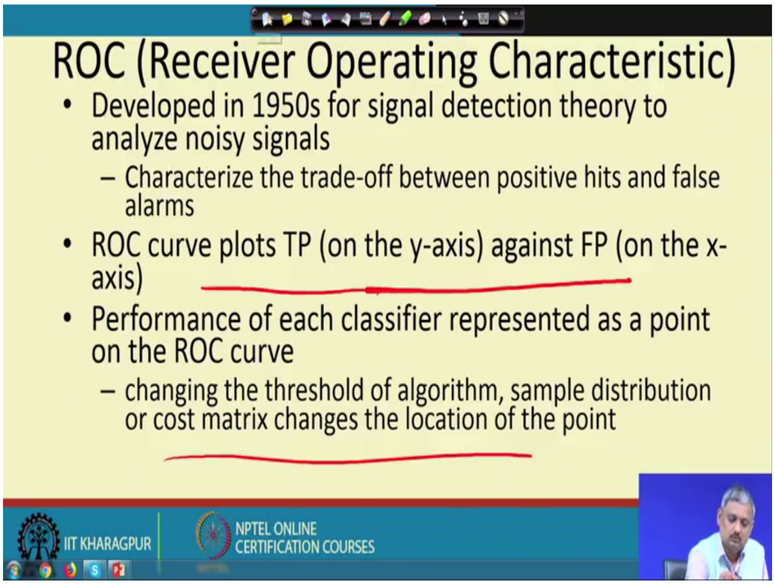
Methods of Estimation

- Holdout
  - Reserve 2/3 for training and 1/3 for testing
- Random subsampling
  - Repeated holdout
- Cross validation
  - Partition data into k disjoint subsets
  - k-fold: train on k-1 partitions, test on the remaining one
  - Leave-one-out: k=n
- Stratified sampling
  - oversampling vs undersampling
- Bootstrap
  - Sampling with replacement

So, this is cross validation, the most common method these are the steps partition train k fold do it average you take. So, this is clear the cross validation method, how to estimate it each of these methods, gives a good method of estimating.

So, what you do you take a training set on that you find this take several training set take the average of these values that is a good estimate. How do you compare models?

(Refer Slide Time: 23:07)

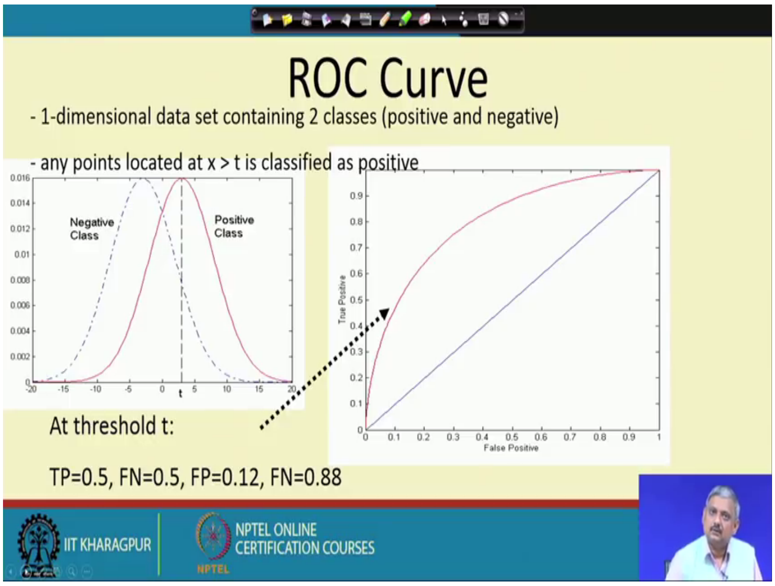


ROC (Receiver Operating Characteristic)

- Developed in 1950s for signal detection theory to analyze noisy signals
  - Characterize the trade-off between positive hits and false alarms
- ROC curve plots TP (on the y-axis) against FP (on the x-axis)
- Performance of each classifier represented as a point on the ROC curve
  - changing the threshold of algorithm, sample distribution or cost matrix changes the location of the point

ROC curve, this is the definition true positive, false positive and this is important all right. So, this looks like this.

(Refer Slide Time: 23:27)



ROC Curve

- 1-dimensional data set containing 2 classes (positive and negative)

- any points located at x > t is classified as positive

At threshold t:

TP=0.5, FN=0.5, FP=0.12, FN=0.88

So, area under ROC curve is a good measure. So, these are the measures so I am not listing all of them you can go through them I can go through them test of significance many thing. So, that is how you measure it.

So, I think you have got an idea of how to evaluate the classifiers, you if we have a training set and it is set and these are commonly used for the classification. So, if you apply this classification algorithm we have to use this. So, thank you for today will go into the next topic in my next lecture.

Thank you.