Machine Learning for Soil and Crop Management
Professor. Somsubhra Chakraborty
Agricultural and Food Engineering Department
Indian Institute of Technology, Kharagpur
Lecture 10
Basics of Multivariate Data Analytics (Contd.)

Welcome friends to this tenth lecture of NPTEL online certification course of Machine Learning for Soil and Crop Management. And this is the fifth lecture of week 2. And in this week 2, we are discussing some basics of Multivariate Data Analytics. So, in our previous lectures, we have discussed about the association between multiple variables, how to calculate the correlation, what is covariance.

We have already seen the simple linear regression. We have also seen the diagnostic features of simple linear regression, their assumptions of simple linear regression. We have also seen the multiple linear regression, we have seen different types of data transformation. Box-Cox transformation, centering, scaling, we have seen.

And also in our previous lecture, we have seen the multiple linear regression, pitfalls of multiple linear regression like overfitting, like multicollinearity, what is multicollinearity, we have discussed. How to detect the multicollinearity using VIF for Variance Inflation Factor. So, we have seen that.

And also we have seen, what is overfitting, especially in terms of regression model, we have seen. And also how to detect the multicollinearity, how to resolve the multicollinearity, we have seen. What is overfitting, what is under fitting, how we can detect the overfitting, under fitting, by seeing the training and test error rate along with multiple iterations we have seen. So, we have already discussed that thing.

Then another very important concept we have discussed in our last lecture that is the bias variance tradeoff. In any type of prediction multivariate regression model, the bias and variance are two important aspects and how to make a tradeoff between bias and variance based on the model complexity.

And then model interpretability, we have discussed. But remember, when the model is simple, then it has higher bias but low variance, but when the model is complex, that can show low bias but high variance. So, we have to find a switch spot and we have to find a tradeoff between this bias and variance in case of any prediction model.

So, in this lecture, we are going to start from there and we are going to see how these overfitting can be a problem in case of a classification problem. So, we have seen the prediction. Now, let us see how overfitting can be problematic in a classification problem.

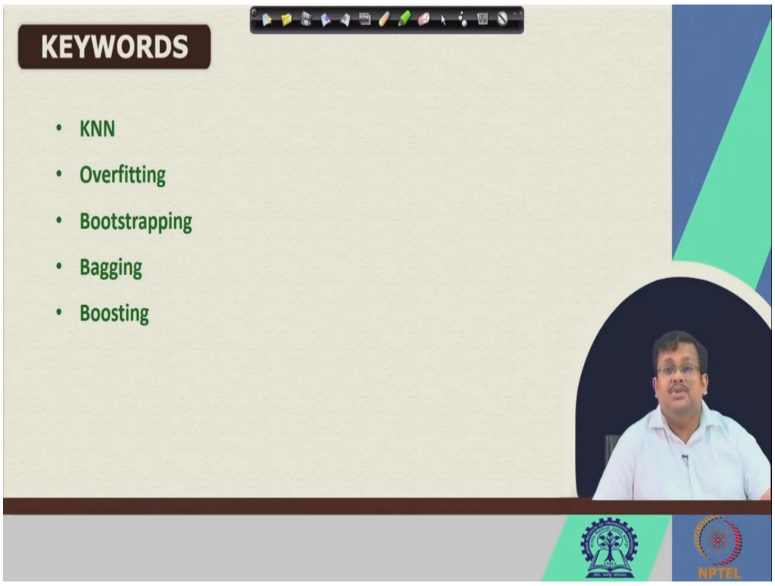
(Refer Slide Time: 03:32)

CONCEPTS COVERED

- Overfitting in KNN
- Avoiding Overfitting
- Bootstrapping
- Bagging
- Boosting

So, we are going to discuss these important concepts like overfitting, specifically giving an example of K Nearest Neighbor regression classification and then, how to avoid the overfitting we are going to discuss. We have seen the overfitting, but how to avoid the overfitting, we are also going to discuss. Then we are going to discuss some of the important concepts like bootstrapping, and then some ensemble method like bagging and boosting, for reducing the overfitting problem.

(Refer Slide Time: 04:16)

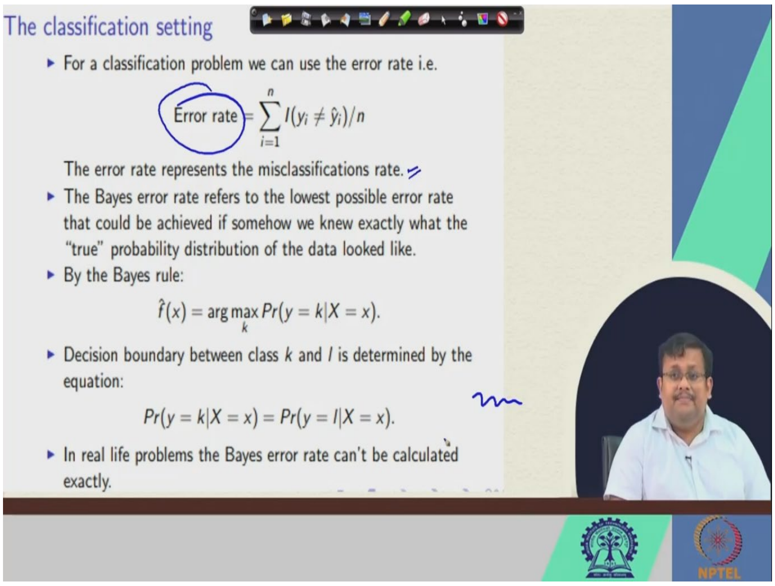


KEYWORDS

- KNN
- Overfitting
- Bootstrapping
- Bagging
- Boosting

So, these are the keywords which we are going to discuss in this lecture. KNN, overfitting, bootstrapping, bagging and boosting, these are the important keywords, which we are going to discuss.

(Refer Slide Time: 04:31)



The classification setting

- For a classification problem we can use the error rate i.e.

$$\text{Error rate} = \sum_{i=1}^{n} I(y_i \neq \hat{y}_i)/n$$

The error rate represents the misclassifications rate.

- The Bayes error rate refers to the lowest possible error rate that could be achieved if somehow we knew exactly what the "true" probability distribution of the data looked like.
- By the Bayes rule:

$$\hat{f}(x) = \arg\max_{k} Pr(y = k|X = x).$$

- Decision boundary between class k and l is determined by the equation:

$$Pr(y = k|X = x) = Pr(y = l|X = x).$$

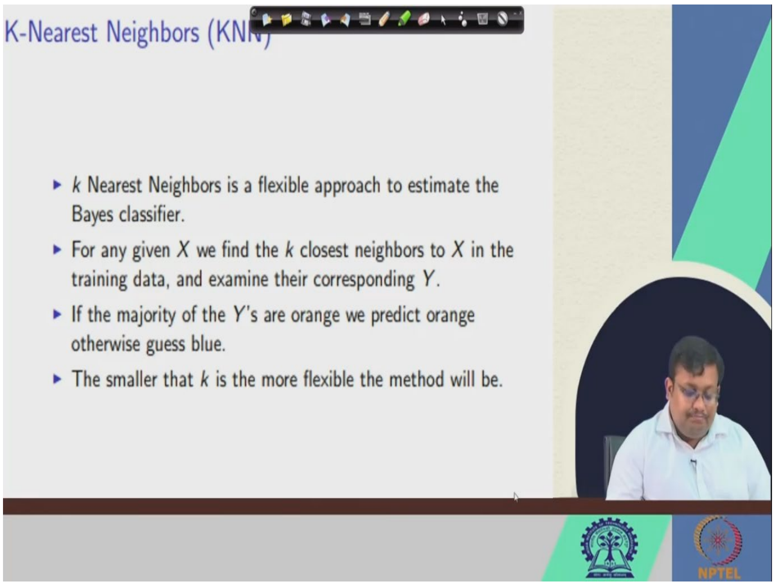
- In real life problems the Bayes error rate can't be calculated exactly.

So, we have seen the overfitting problem in case of prediction setting, when we are having a multivariate prediction model. But, what happens in case of a classification setting. So, for a classification problem, we can use the error rate that is, this error rate is basically summation of these yi not equal to yi hat divided by n.

So, the error rate represent the misclassification rate. So, here these error rate, basically, shows the misclassification rate. Misclassification means when we wrongly classify any sample and wrongly assign a label to that sample that is called misclassification. And this misclassification is determined by using the misclassification rate. So, the misclassification rate can be calculated by this error rate.

So, the Bayes error rate refers to the lowest possible error rate that could be achieved, if somehow we knew exactly what the true probability distribution of the data look like. So, by the Bayes rule, we can see this is the rule and the decision boundary between class K and an I is determined by the this equation. So, in real life problem the bias error rate can be calculated exactly. So, let us see an example in detail.

(Refer Slide Time: 06:18)



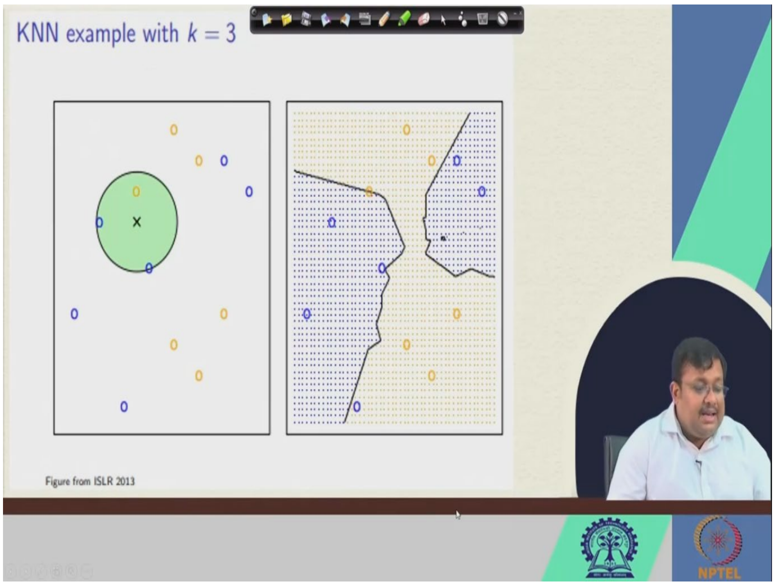
K-Nearest Neighbors (KNN)

- k Nearest Neighbors is a flexible approach to estimate the Bayes classifier.
- For any given X we find the k closest neighbors to X in the training data, and examine their corresponding Y.
- If the majority of the Y's are orange we predict orange otherwise guess blue.
- The smaller that k is the more flexible the method will be.

So, here you can see that it is K-Nearest Neighbor, which is a very important classification method, a clustering method and this K-Nearest Neighbor is a flexible approach to estimate the bias classifier. So, for any given x, we find k closest neighbors to x in the training data and examine their corresponding y.

So, if the majority of the ys are orange, we predict orange otherwise guess blue. So, based on the majority rule, we assign the level to the data. So, the smaller, that k is more flexible the model will be. So, if we reduce the k that will make the more flexible model. So, let us see one example.

(Refer Slide Time: 07:28)

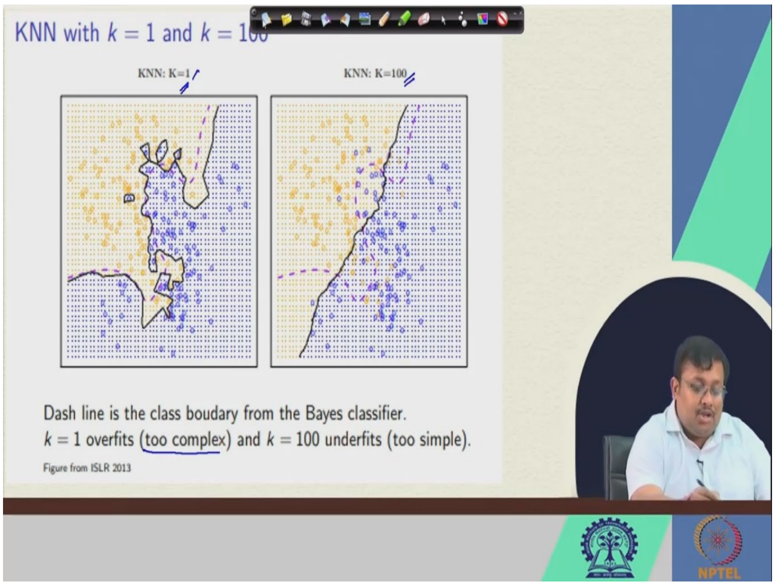


KNN example with $k = 3$

Figure from ISLR 2013

So, here you can see, it is a KNN example with k equal 3. So, based on this, we can find three different clusters, since k equal to 3.

(Refer Slide Time: 07:45)



KNN with $k = 1$ and $k = 100$

Dash line is the class boudary from the Bayes classifier.
$k = 1$ overfits (too complex) and $k = 100$ underfits (too simple).

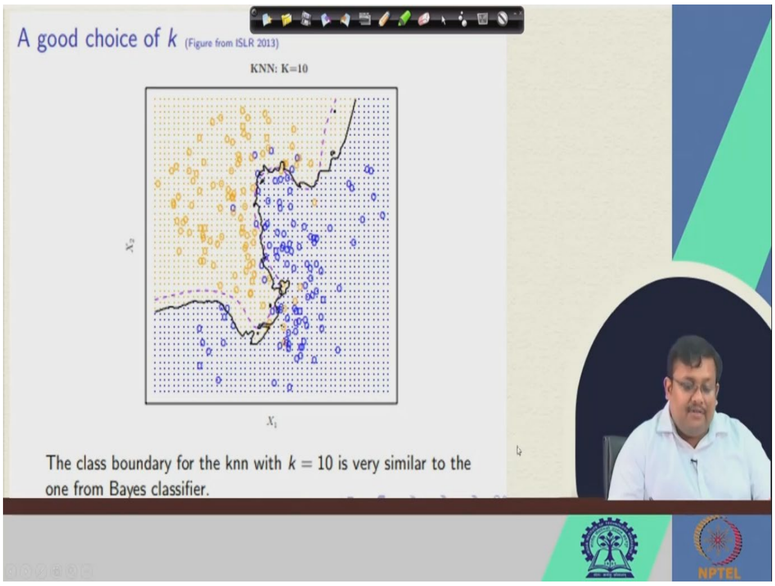
Figure from ISLR 2013

Now, let us see the difference between k equal to 1 and k equal to 100. So, this is k equal to 1, and here whereas, we are fitting a KNN classifier with k equal to 100. So, here this dashed line shows the boundary from the biasing classifier. So, where we are using k equal to 1, that overfits, because, we are perfectly classifying the two groups of samples. So, it is too complex, but, when there is a too much complexity that will also reduce the bias, we are already know that from our previous slide.

If you go back to our previous slide, we have seen that the smaller the k will be the more flexible the method, that means, it will have more flexibility. So, you can see when the k is 1, we can perfectly classify the samples into two groups by using this class boundary. However, when we use the k equal to 100 that underfits, because, that becomes too simple then. So, this is how we should avoid the overfitting and underfitting and find the best possible value of k in case of K-Nearest Neighborhood classification, okay.

So, again when there is a k equal to 1, it perfectly overfits, because it is making a very accurate boundary. However, when it is 100, it becomes too simple and it also underfits. And overfitting occurs when the model becomes too complex, when it learns too much, based on the data and also in the noise, but underfits is when the model is not sufficiently learned that is called underfitting.

(Refer Slide Time: 10:20)

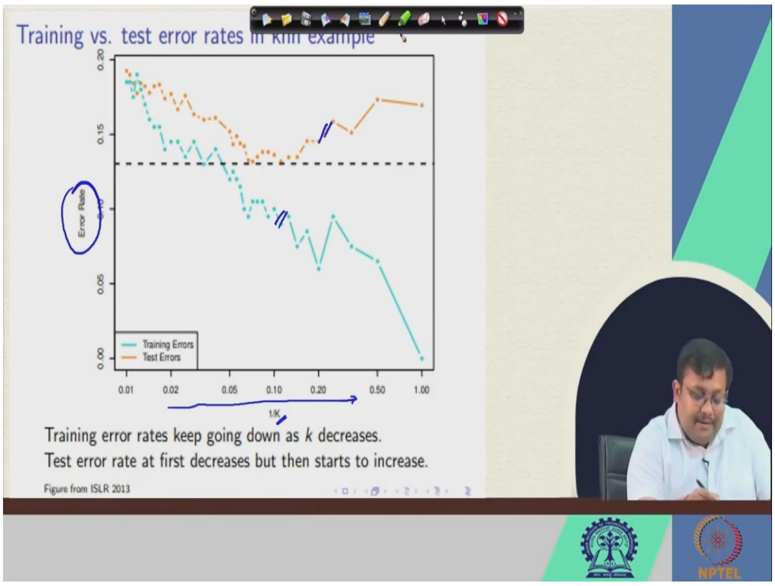


A good choice of k (Figure from ISLR 2013)

The class boundary for the knn with $k = 10$ is very similar to the one from Bayes classifier.

Now, let us see. So, the class boundary, it is here, we can see, this is the class boundary for the KNN with k equal to 10 is very similar to the one from Bayes classifier. So, it is a good choice of k. We can see it is a compromise between k equal to 1 and k equal to 100. Here, we are making a tradeoff between the complexity and the overfitting and underfitting and we are selecting the optimum value of k that is 10.

(Refer Slide Time: 10:50)



Training vs. test error rates in knn example

Training error rates keep going down as k decreases.
Test error rate at first decreases but then starts to increase.

Figure from ISLR 2013

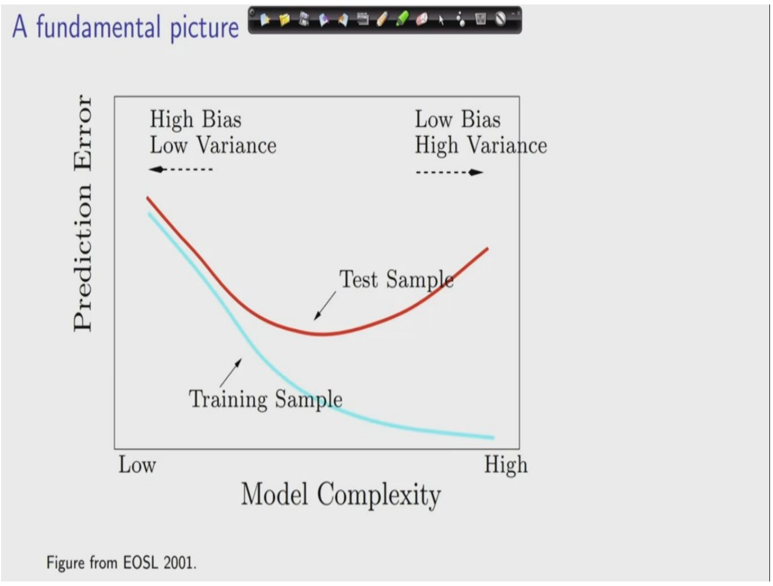
So, if we see, if we plot the error rate and 1 by k for both training error and test error, we again divide the data into training data set and test data set, so again we see the training error and test error with increasing values of 1 by k, so, we can see here, the training error rate keep going down as k decreases. So, as k decreases, the value of 1 by k increases. So, the training error is going down. That means, when the k decreases means, it is becoming too complex. So, in that way, that is creating overfitting also.

So, here you can see the test error at first decreases and then it starts to increase, just like we have seen, in case of Bayes variance tradeoff and in case of our previous lecture, when the model become too complex and over fitted that taste error first decrease up to a certain point and then they start increasing.

So, that particular thing is happening here, because we are decreasing the k. When we are decreasing the k that means, we are increasing the model complexity or classification complexity, just like we have seen in case of k equal to 1. So, although, the training error is decreasing, because we are learning too much or we are becoming too flexible with the training data set.

So, the training error is decreasing continuously with a decrease in the value of k. But that has very little implication for predicting or for classifying unknown data set that is test data set. So, here we can see clearly the test data set is showing overfitting, so, our model is overfitted. Our classification error model is overfitted.

(Refer Slide Time: 13:24)

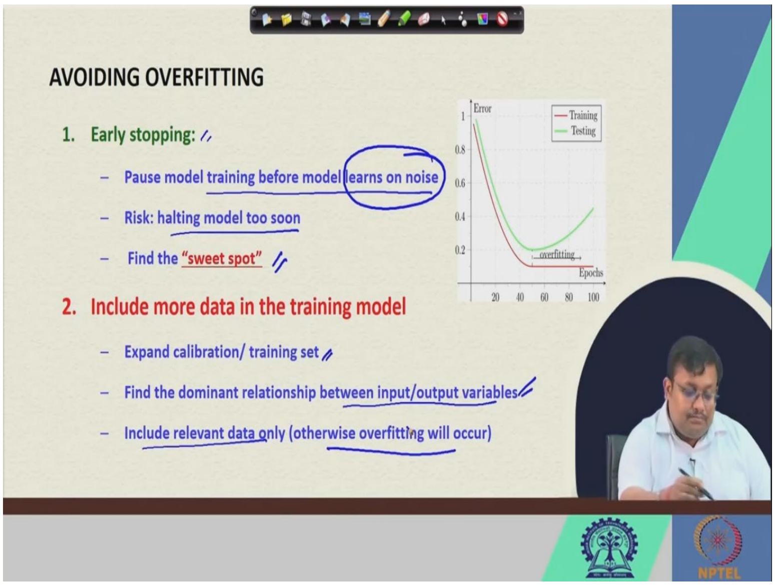


A fundamental picture

Figure from EOSL 2001.

Now, so, again this concept comes in this case, we can just see here, when the model complexity increases from low to high, the training sample error goes down and the test sample error, first initially decrease, but then again it increase. So, this same concept of this fundamental picture of these overfitting occurs in case of classification using the K-nearest neighbor model.

(Refer Slide Time: 13:54)



AVOIDING OVERFITTING

1. Early stopping:
   - Pause model training before model learns on noise
   - Risk: halting model too soon
   - Find the "sweet spot"
2. Include more data in the training model
   - Expand calibration/ training set
   - Find the dominant relationship between input/output variables
   - Include relevant data only (otherwise overfitting will occur)

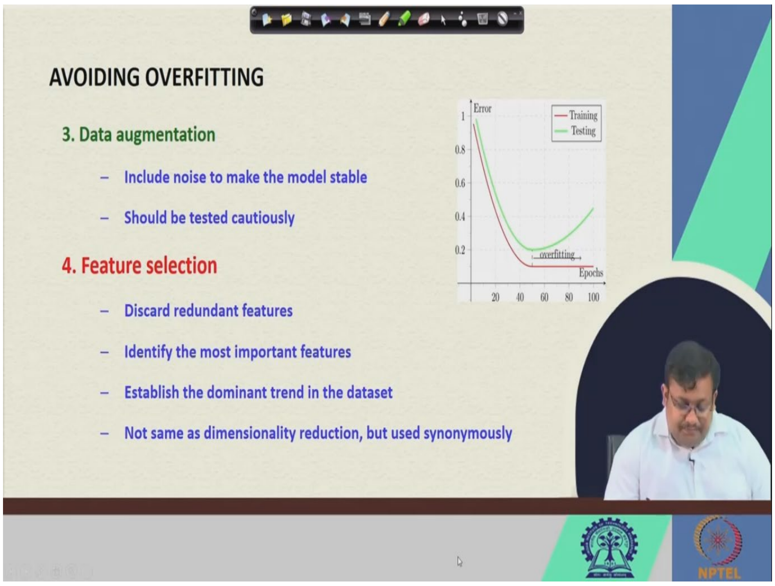
Now, so, we have seen the implication of overfitting, in case of training data set, in case of prediction model and classification model. Now, let us see how we can avoid the overfitting problem. This is very important aspect, because you always want your model not to give over, you always want your model to be robust, you do not want your model to be overfitted, which do not have any which does not have any practical utility.

So, how we can avoid the overfitting? You can avoid overfitting by early stopping. So, you can see here, early stopping means you can pause model training before the model learns on noise. So, when we train our model to for too long, then only that our model learns on the noise. But, if we stop, if we pause the model training before it learns on noise that could remove that overfitting problem that is called early stopping.

But there is a risk too. Because, halting model will be, we can in that way; we can halt the model too soon. So, we have to find this sweet spot as I have already mentioned couple of times. Also, another way is to include the more data in the training model could be another solution. So, you can expand the calibration or training set, you can expand, you can include more data into your training model, to avoid overfitting and you can find that dominant relationship between input and output variables. This is another way.

And the third way is, include relevant data only otherwise overfitting will occur. So, that is why you have to be very very careful about multicollinearity and you should remove the unnecessary variables, which are highly correlated with other independent variables. So, you should remove them by using the VI F score and you can keep only the uncorrelated factors or variables in your model for training your model, and otherwise these overfitting may happen.

(Refer Slide Time: 16:53)



AVOIDING OVERFITTING

3. Data augmentation
   - Include noise to make the model stable
   - Should be tested cautiously

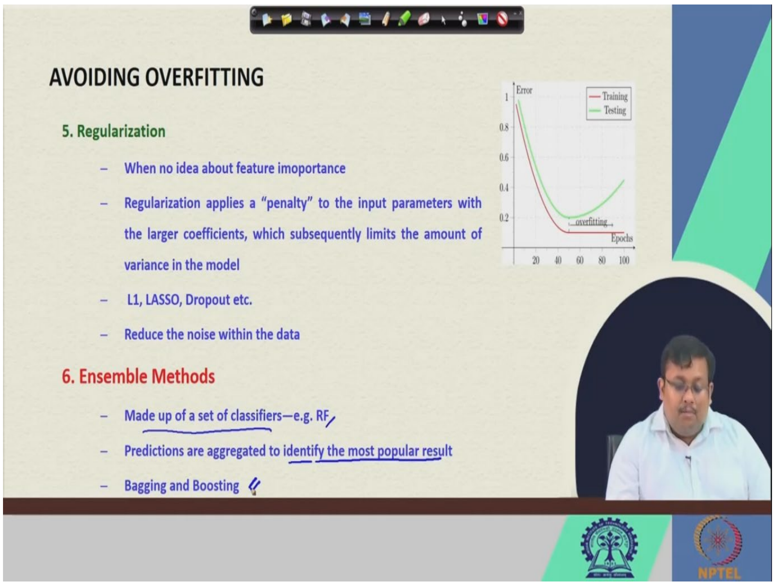
4. Feature selection
   - Discard redundant features
   - Identify the most important features
   - Establish the dominant trend in the dataset
   - Not same as dimensionality reduction, but used synonymously

Third way of removing the overfitting is the data augmentation. So, in the data augmentation, what happens, if you include noise to make the model stable? So, it is a unique thing, I mean, here we are including noise to make the model stable, but we should do it, we should test these data augmentation very, very cautiously.

The fourth one is feature selection. Now, this feature selection, we have already an overview in our first week. Now, feature selection means you have to discard the redundant features. That means you are discarding those features, who are which are showing the multicollinearity. This is one of the way. Or you can identify the most important features and keep them only. This is another way of feature selection.

Then you can establish that dominant trend in the data set that is another way. And this feature selection is not actually equal to dimensionality reduction, you will come across this term very frequently, reducing the dimension or dimensional reduction, but they are not, although they are used synonymously, they are not same. We will see them in details in our upcoming lectures. So, feature selection is another very effective way for dealing with the overfitting problem.

(Refer Slide Time: 18:34)



AVOIDING OVERFITTING

5. Regularization

- When no idea about feature importance
- Regularization applies a "penalty" to the input parameters with the larger coefficients, which subsequently limits the amount of variance in the model
- L1, LASSO, Dropout etc.
- Reduce the noise within the data

6. Ensemble Methods

- Made up of a set of classifiers—e.g. RF
- Predictions are aggregated to identify the most popular result
- Bagging and Boosting

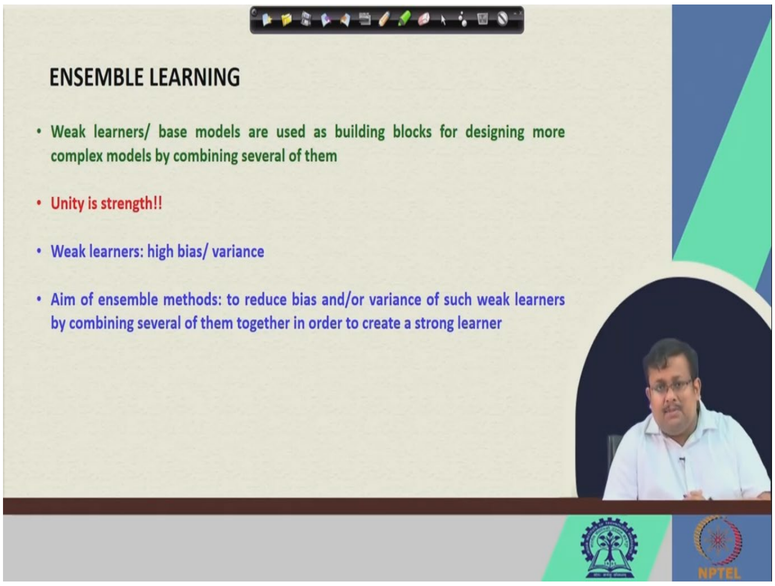
Now, the another method for reducing the overfitting is a regularization method. Now, what is regularization method? Regularization method is, when we do not have any idea about feature importance. When we do not have any idea about the feature importance, so in that case, regularization applies a penalty to the input parameters with a larger coefficient, which subsequently limits the amount of variance in the model. So, these are called regularization methods. We are adding a penalty to the input parameters with the larger coefficients.

So, we first identify which are the larger coefficients and then we apply a penalty to those parameters, input parameters, which subsequently limits the amount of variance in the model. So, that is how this regularization method works. Some examples of regularization methods are L1, Lasso Dropout etcetera and it can reduce the noise within the data. So, it these regularization methods are helpful for reducing the noise within the data.

Another method is called, the another major way of avoiding overfitting is, the use of ensemble methods. Now, the ensemble methods are made up of a set of classifiers, for example, in case of random forest. Random forest is a tree-based classifier. So, we will see that these types of ensemble methods are made up of a set of classifiers.

And predictions are here, in ensemble methods predictions are aggregated to identify the most popular results and some examples are bagging and boosting. We are going to discuss these two terms in more details in our coming slides.

(Refer Slide Time: 21:01)



ENSEMBLE LEARNING

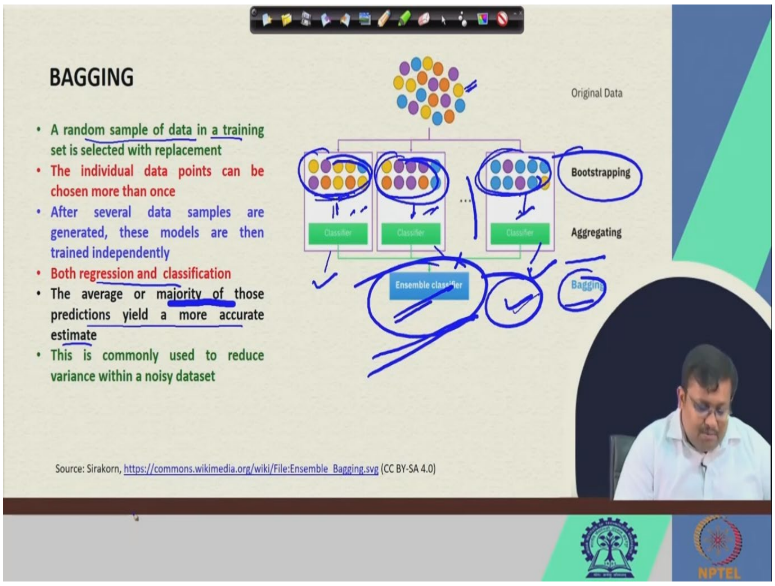
- Weak learners/ base models are used as building blocks for designing more complex models by combining several of them
- Unity is strength!!
- Weak learners: high bias/ variance
- Aim of ensemble methods: to reduce bias and/or variance of such weak learners by combining several of them together in order to create a strong learner

So, what is ensemble learning? Ensemble learning is a method where weak learners or base models are used to build the strong learner. Or in other words, weak learners or base model are used as building blocks for designing more complex model by combining several of them. So, this is how we develop a strong learning algorithm or strong learner, based on some weak learner or very basic models.

Because these ensemble learning is based on the principle of unity is strength. Unity is strength is the there, the principle, major principle of this ensemble learning methods. The weak learners are always have generally high bias or variance. So, when we have too simple model or too complex model, both of them could be weak learners, because, they could have either high bias or variance.

So, the aim of ensemble method is to reduce the bias and all variants of such weak learners by combining several of them together in order to create a strong learner. So, when there is a weak model or when there is a weak learner based on the simplistic model or too complex model, which can show either high bias or high variance, these type of problems can be solved when we use ensemble learning. And ultimately, we combine these weak learners together to get a strong learner.

(Refer Slide Time: 23:04)



BAGGING

- A random sample of data in a training set is selected with replacement
- The individual data points can be chosen more than once
- After several data samples are generated, these models are then trained independently
- Both regression and classification
- The average or majority of those predictions yield a more accurate estimate
- This is commonly used to reduce variance within a noisy dataset

Source: Sirakorn, https://commons.wikimedia.org/wiki/File:Ensemble_Bagging.svg (CC BY-SA 4.0)

Now, let us see what is begging. Those two bagging and boosting are two methods, ensemble methods, learning methods we are going to see in details. Now, remember bagging is a short form of bootstrap aggregation. Bootstrap aggregation. So, what is bootstrapping? And remember, the major difference between bagging and boosting is, bagging is a parallel approach, whereas boosting is an inter-connected approach or sequential approach.

Let us see what is bagging. So, what is called a bootstrap sample? Bootstrap sample means a random sample suppose. We are taking random sample of data in a training set, say suppose, this is a training set and we are selecting the training data, suppose this is an original data. So, from the original data, suppose, we are selecting a training set with replacement, that means, one sample can occur multiple times. So, this is known as the bootstrap sample.

This is also known as sample with replacement. So this is called a bootstrap sample. This is another bootstrap sample. This is another bootstrap sample and the process is known as bootstrapping. So, here again I am repeating a random sample of data in a training set is selected with replacement and the individual data points can be chosen more than once. As we are seeing here in these bootstrap sample, the individual data points are selected more than once.

Now, after several samples are generated, these models are then trained independently. So, once we select these training data set, we are training this training data set, we are training this training data set, we are training this training data set, let us fit the individual classifier, let us fit an individual classifier and aggregate. So, we can an aggregate their results. So, we can apply this bootstrapping method or bagging method to both regression and classification.

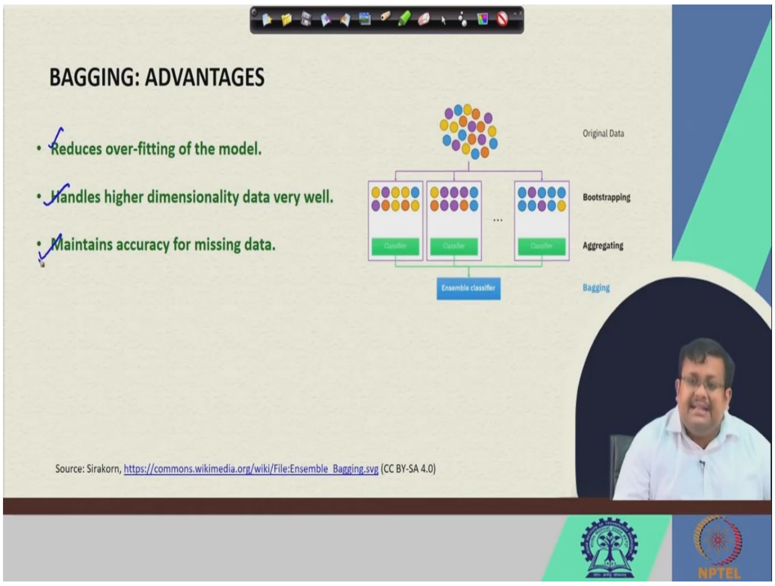
Here I am giving a classification example, but this can be applied to both regression and classification. Remember that the average or majority of those predictions yield a more accurate estimate. So, if suppose this classifier gives a classification, this classifier, it gives the classification, this classifier gives the classification.

So, suppose this is classifying is correct, and this classifier giving this ultimate label as cross and this classifier is giving the ultimate level as this tick. So, based on the majority of the vote, we can see that the ultimate, the ensemble classifier will give the final output as tick. So, based on the majority vote, this will be determined, the final outcome will be determined. So, average.

So, in case of regression problem, we generally take the average from individual prediction, individual models, however, in case of classification problem, we take the majority of the prediction and ultimately get the final ensemble classifier. So, we believe these ensemble classifiers is more stronger that these individual weak learners, because way we are combining from these weak learners, we are getting more stronger model because of the assumption that these models, these individual weak learners, will not do, will not wrongly predict in all the instances.

So, if you take a majority of the vote, we can get the more stronger learner that is called ensemble classifier. So, this is called bootstrap aggregation. And you can see all these are going parallel, so we call it bagging method. So, this is commonly used to reduce the variance within their noisy data. So, this is called bagging method.

(Refer Slide Time: 27:43)



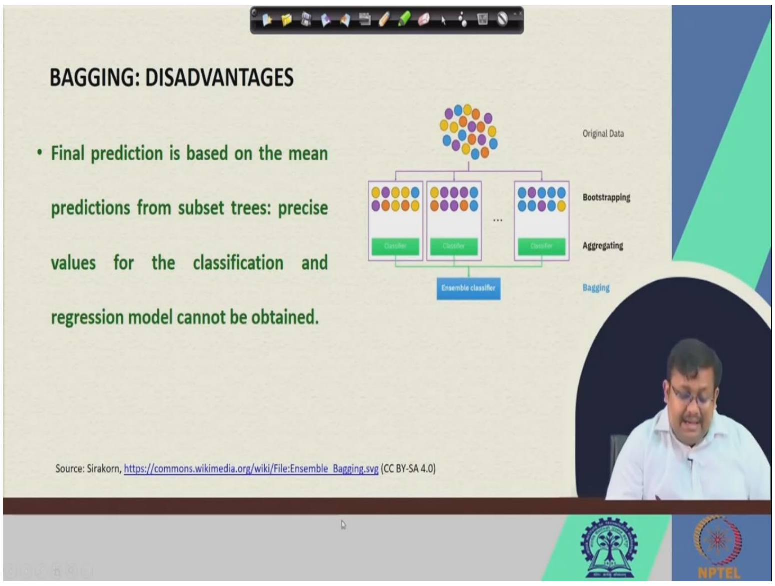
BAGGING: ADVANTAGES

- ✓Reduces over-fitting of the model.
- ✓Handles higher dimensionality data very well.
- ✓Maintains accuracy for missing data.

Source: Sirakorn, https://commons.wikimedia.org/wiki/File:Ensemble_Bagging.svg (CC BY-SA 4.0)

Now, what are the advantages and disadvantages? Bagging method has several advantages. First of all, it can reduce the overfitting of the model, it can handle higher dimensionality data, high dimensional data, where there are huge number of variables involved and it can maintain the accuracy of the missing data also. So, these are the advantage.

(Refer Slide Time: 28:05)



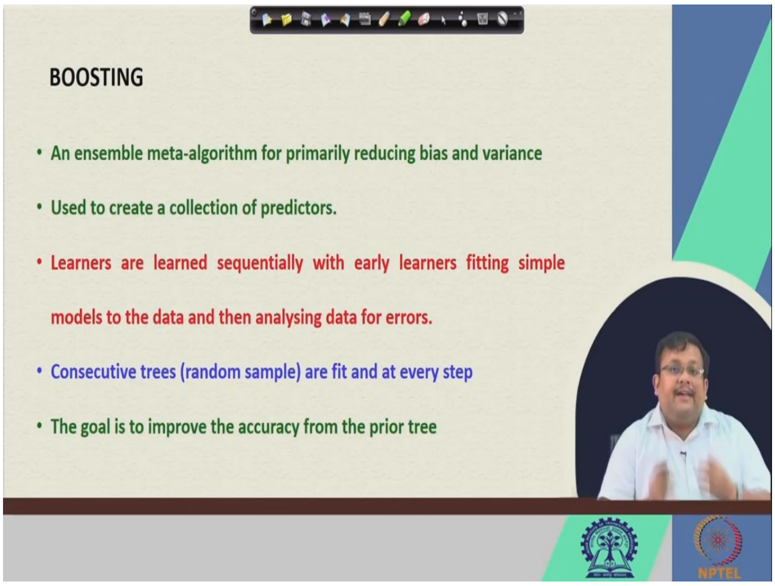
BAGGING: DISADVANTAGES

- Final prediction is based on the mean predictions from subset trees: precise values for the classification and regression model cannot be obtained.

Source: Sirakorn, https://commons.wikimedia.org/wiki/File:Ensemble_Bagging.svg (CC BY-SA 4.0)

However, there are some disadvantages also. Disadvantage is final prediction is based on the mean prediction from the subset trees. So, precise values of the classification regression model cannot be obtained. So, whatever we are getting is ensemble model outcome, based on either the averaging or from the majority votes from individual classified results will not get. So, these are the some of the disadvantage of bagging.

(Refer Slide Time: 28:37)



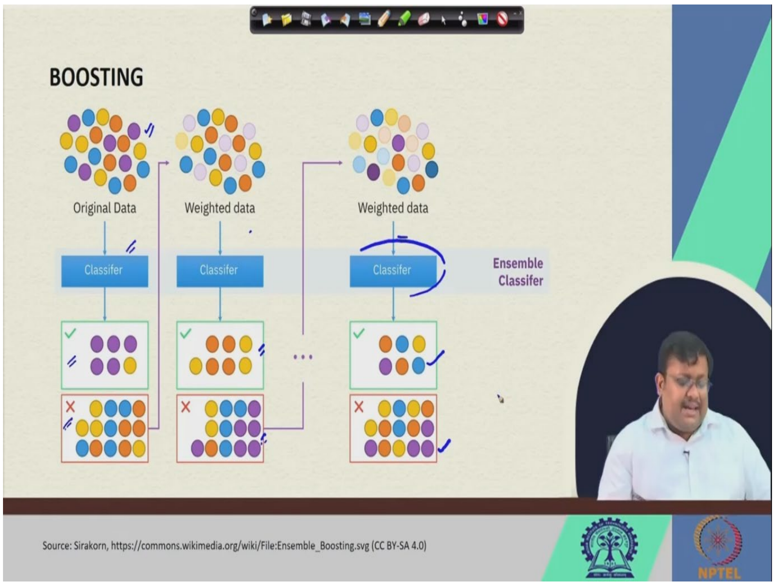
BOOSTING

- An ensemble meta-algorithm for primarily reducing bias and variance
- Used to create a collection of predictors.
- Learners are learned sequentially with early learners fitting simple models to the data and then analysing data for errors.
- Consecutive trees (random sample) are fit and at every step
- The goal is to improve the accuracy from the prior tree

Now, next and the last point is boosting. So, what is boosting? Basically, boosting is an ensemble meta-algorithm, for primarily reducing the bias and variance. And generally it is used to create a collection of predictors. So, here I have already told you, that it is a sequential process.

In case of bootstrapping, in case of boosting it is a parallel process, but this is a sequential process. So, here learners are learned sequentially with early learners, fitting simple model to the data and then analyzing the data for the errors. And then, consecutive trees or random sample are fit at every step. So, the goal of this boosting is to improve the accuracy from the prior tree.

(Refer Slide Time: 29:26)

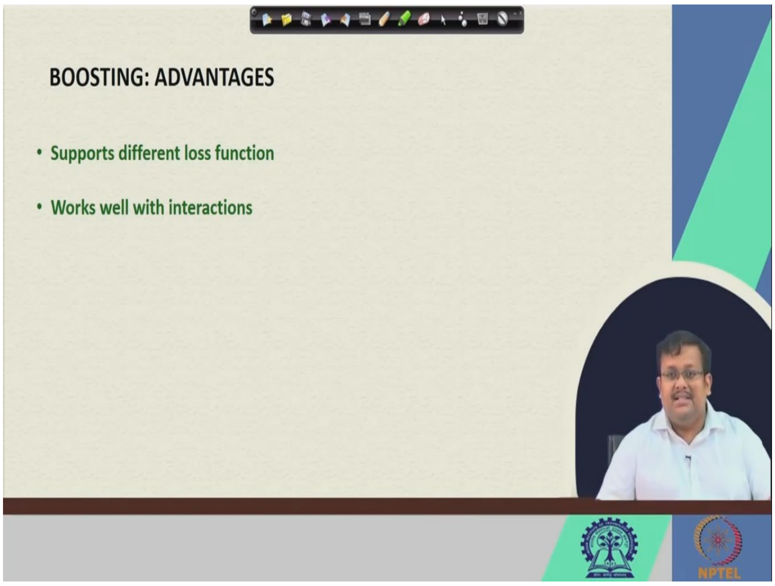

BOOSTING

Source: Sirakorn, https://commons.wikimedia.org/wiki/File:Ensemble_Boosting.svg (CC BY-SA 4.0)

So, this picture gives a very good explanation of the boosting. So, you can see we have an original data set. So, from this original data set, let us fit a classifier and we can see that some of the samples will be correctly classified and rest of the samples will be misclassified. So, we take these misclassified samples, we give them some weightage and fit a next classification algorithm.

Then we can see, some of them will be correctly classified, some of them will be wrongly classified. And then,, this misclassified sample will go after getting, and will again give some weightage, so that to reduce their chance of misclassification. And then again, they will be classified, and then we will see the correctly classified and misclassified. So this is an ensemble classifier and you can see here, the operation is going on sequentially. So, this the difference between bagging and boosting.

(Refer Slide Time: 30:39)

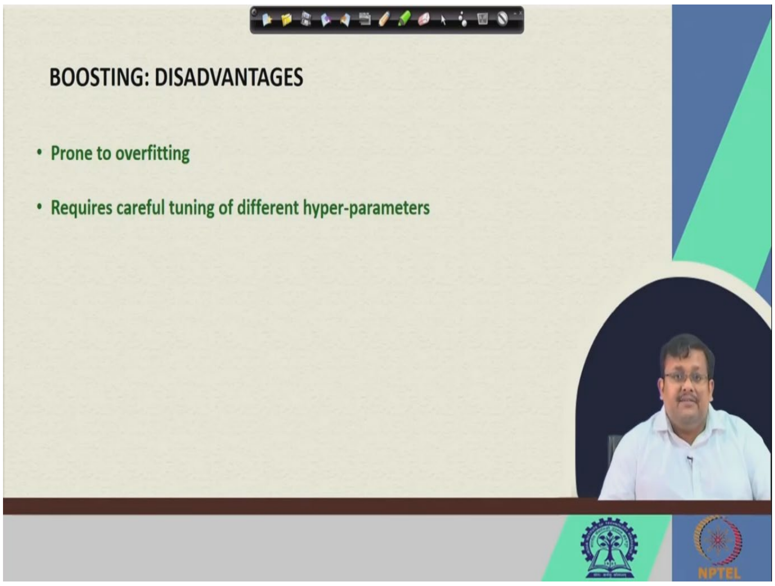


BOOSTING: ADVANTAGES

- Supports different loss function
- Works well with interactions

So, what are the advantages of the boosting? The advantages boosting is, it supports different loss function and it works well with interactions. These are some of the bosting advantages.

(Refer Slide Time: 30:52)



BOOSTING: DISADVANTAGES

- Prone to overfitting
- Requires careful tuning of different hyper-parameters

However, there is a disadvantage also. First of all, the boosting is sometimes prone to overfitting, and it requires careful tuning of different hyper parameters before run the boosting method. There are different types of boosting algorithm like exiboost, databoost algorithm. So, we will see their application in our subsequent lectures.

So, guys, we have seen now, the details of multicolinearity, we have seen the details of overfitting, how to reduce the overfitting, in case of using different types approaches, we have seen boosting, we have seen bagging, we have seen, what is bootstrapping. So, I hope that in this chapter you have gained some useful knowledge for handling with multi variate data and to remove some of the inherent problems, while dealing with multi variate data.

So, let us wrap up our lecture here. And from next week onwards, we will actually see the application of different machine learning, classification, clustering, and prediction model, application on agricultural problems. So we will start next week with principle component analysis. So, please stay tuned and join with me in our next set of lectures. Thank you very much.