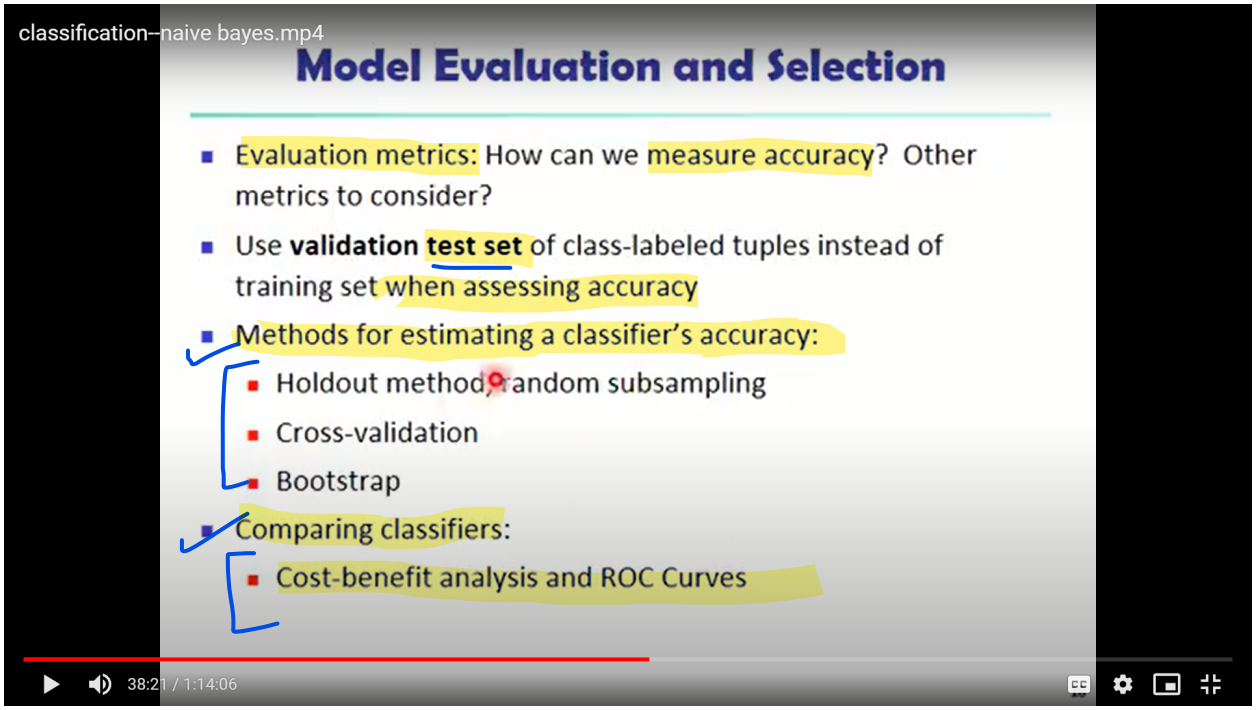
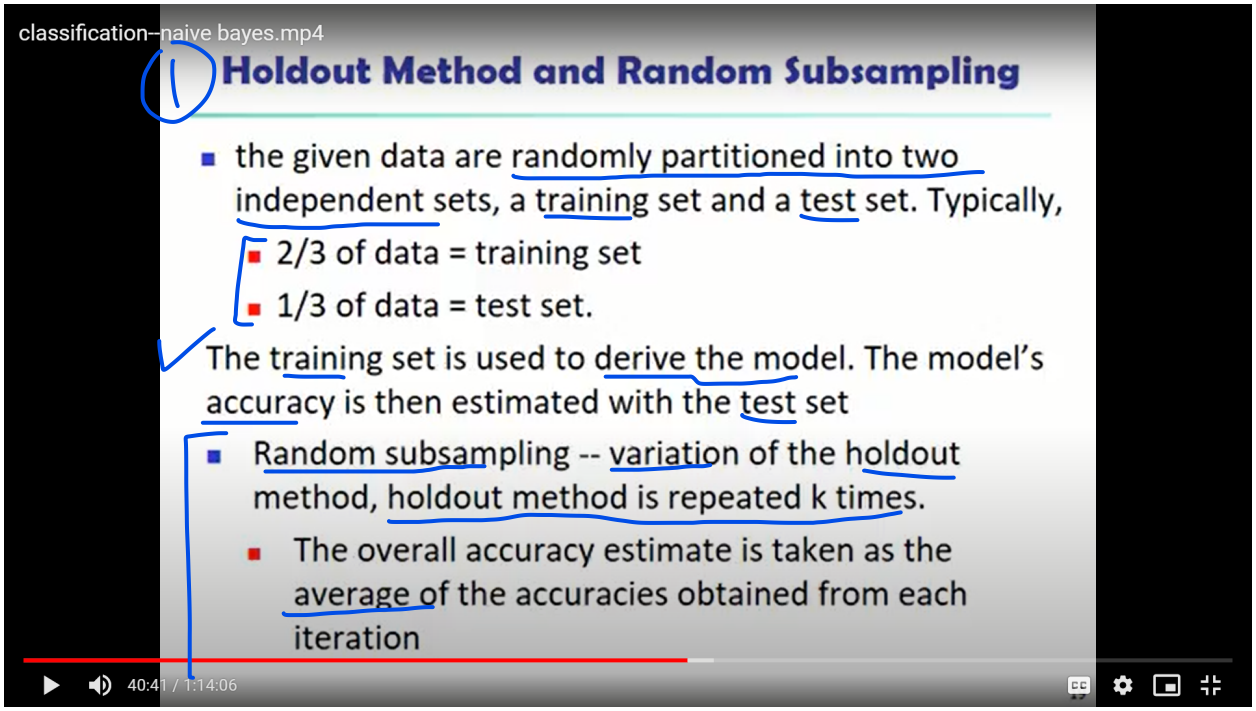
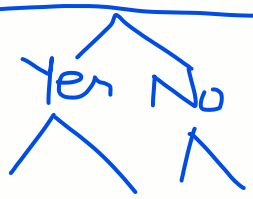

classification--naive bayes.mp4
Model Evaluation and Selection
Evaluation metrics: How can we measure accuracy? Other metrics to consider?
Use validation test set of class-labeled tuples instead of training set when assessing accuracy
Methods for estimating a classifier's accuracy:
Holdout method, random subsampling
Cross-validation
Bootstrap
Comparing classifiers:
Cost-benefit analysis and ROC Curves
38:21 / 1:14:06


classification--naive bayes.mp4
Holdout Method and Random Subsampling
the given data are randomly partitioned into two independent sets, a training set and a test set. Typically,
2/3 of data = training set
1/3 of data = test set.
The training set is used to derive the model. The model's accuracy is then estimated with the test set
Random subsampling -- variation of the holdout method, holdout method is repeated k times.
The overall accuracy estimate is taken as the average of the accuracies obtained from each iteration
40:41 / 1:14:06

Cross-Validation

- In k-fold cross-validation, the initial data are randomly partitioned into k mutually exclusive subsets or "folds," D1, D2,..., Dk , each of approximately equal size.
- Training and testing is performed k times.
- In iteration i, partition Di is reserved as the test set, and the remaining partitions are collectively used to train the model.
- each sample is used the same number of times for training and once for testing.
- Accuracy is the overall number of correct classifications from the k iterations, divided by the total number of tuples in the initial data.

- Leave-one-out is a special case of k-fold cross-validation where k is set to the number of initial tuples. That is, only one sample is "left out" at a time for the test set.
- In stratified cross-validation, the folds are stratified so that the class distribution of the tuples in each fold is approximately the same as that in the initial data



Bootstrap

- the bootstrap method samples the given training tuples uniformly with replacement.

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

same tuple can repeat

Classifier Evaluation Metrics: Confusion Matrix

Confusion Matrix:

Actual class\Predicted class	C_1	$\neg C_1$
C_1	**True Positives (TP)**	**False Negatives (FN)**
$\neg C_1$	**False Positives (FP)**	**True Negatives (TN)**

Example of Confusion Matrix:

Actual class\Predicted class	buy_computer = yes	buy_computer = no	Total
buy_computer = yes	**6954**	**46**	7000
buy_computer = no	**412**	**2588**	3000
Total	7366	2634	10000

- Given *m* classes, an entry, $CM_{i,j}$ in a **confusion matrix** indicates # of tuples in class *i* that were labeled by the classifier as class *j*
- May have extra rows/columns to provide totals

Classifier Evaluation Metrics: Accuracy, Error Rate, Sensitivity and Specificity

A\P	C	¬C	
C	TP	FN	P
¬C	FP	TN	N
	P'	N'	All

- **Classifier Accuracy,** or recognition rate: percentage of test set tuples that are correctly classified

  Accuracy = (TP + TN)/All

- **Error rate:** *1 – accuracy*, or

  Error rate = (FP + FN)/All

- **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
  - **Sensitivity**: True Positive recognition rate
    - **Sensitivity = TP/P**
  - **Specificity**: True Negative recognition rate
    - **Specificity = TN/N**

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

Classifier Evaluation Metrics: Precision and Recall, and F-measures

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

$$precision = \frac{TP}{TP + FP}$$

- **Recall:** completeness – what % of positive tuples did the classifier label as positive?

$$recall = \frac{TP}{TP + FN}$$

- Perfect score is 1.0
- Inverse relationship between precision & recall
- ***F* measure (*F*$_1$ or *F*-score)**: harmonic mean of precision and recall,

$$F = \frac{2 \times precision \times recall}{precision + recall}$$

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

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