GSÜSEM

GALATASARAY ÜNİVERSİTESİ
Sürekli Eğitim Uygulama ve Araştırma Merkezi

Sınıflandırma (Classification)

Özlem DURMAZ İNCEL

odurmaz@gmail.com

odincel@gsu.edu.tr

Contents

- Part 1:
  - Introduction to Classification
  - K-Nearest Neighbor Algorithm (KNN) and Example in Python
  - Classifier Evaluation Metrics and Methods in Classification (Introduction)
- Part 2
  - Classifier Evaluation Metrics and Methods in Classification (Continued)
  - Decision Trees and Example in Python
  - Ensemble Methods (Bagging and Boosting)
  - Introduction to SVM and Example in Python (If time permits)

Note and Acknowledgements

- Slides marked with Cognitive Class are the slides of IBM's Cognitive Class, "Machine Learning with Python" Class: https://cognitiveclass.ai/courses/machine-learning-with-python/

Other Resources

- Han, Jiawei, Jian Pei, and Micheline Kamber. Data mining: concepts and techniques. Academic Press, Morgan Kaufmann Publishers, 2001, ISBN 1-55860-489-8

- Machine Learning: Classification by University of Washington, https://www.coursera.org/learn/ml-classification

1. Introduction

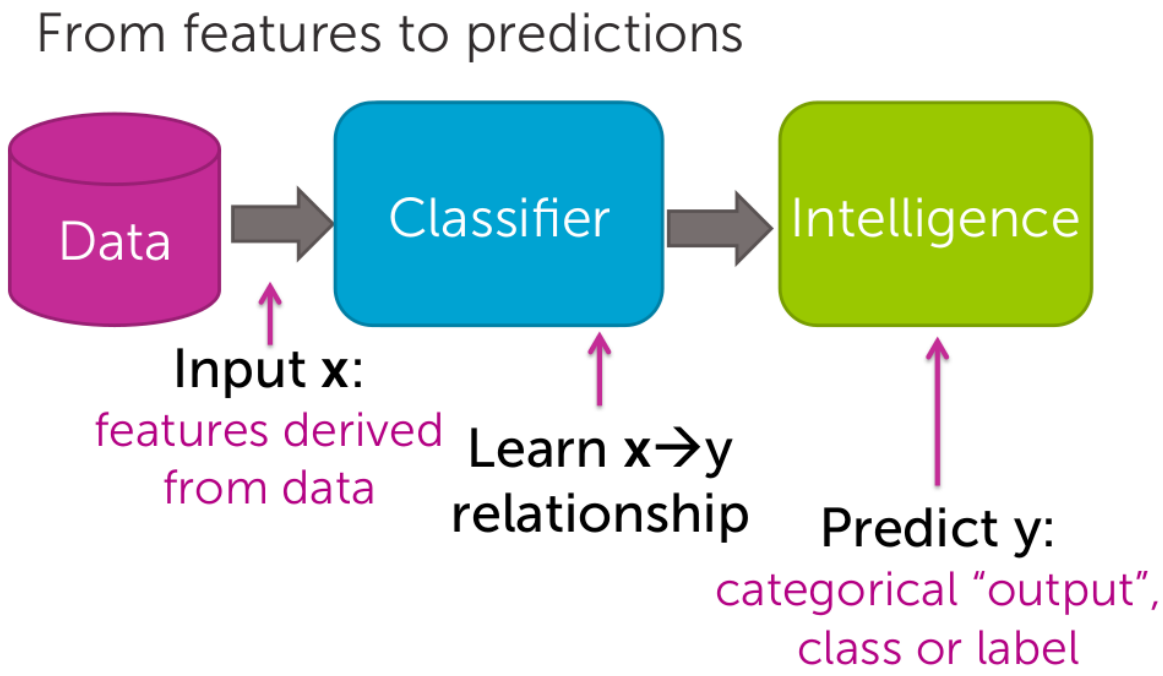
GSÜSEM

GALATASARAY ÜNİVERSİTESİ

Sürekli Eğitim Uygulama ve Araştırma Merkezi

What is Classification? (Sınıflandırma)

- A Supervised learning approach (denetimli öğrenme)
- Categorizing/classifying unknown items into a discrete set of categories or “classes”
- Classification attempts to learn the relationship between a set of feature variables and a target variable of interest.
- The target attribute is a categorical variable with discrete values



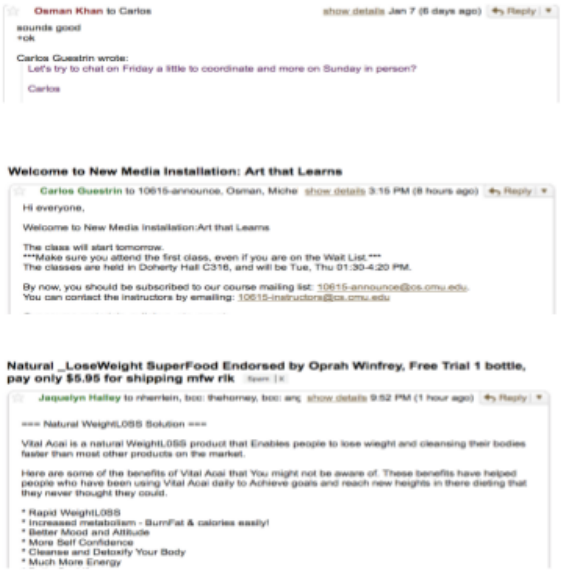
Özniteliklerden Tahminlere

AMAÇ: Bilinmeyen öğeleri ayrık kategoriler veya “sınıflar” halinde ayrıştırma Sınıflandırma, öznitelikler ile bir hedef değişken (sınıf) arasındaki ilişkiyi öğrenmeye çalışır.

Image: Machine Learning: Classification by University of Washington

Classification Examples

Binary Classifier (İkili Sınıflandırıcı)



Input: x

Text of email, sender, IP,...

Not spam

Spam

Output: y

Input: x
Webpage

Education

Finance

Technology

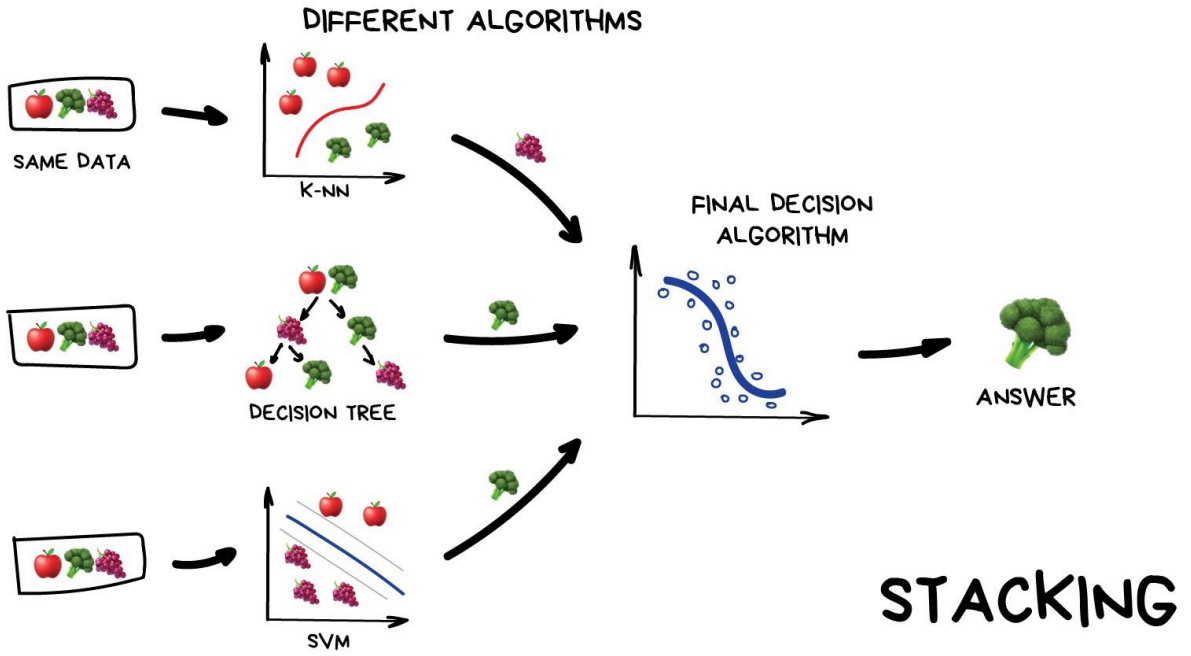
Output: y

Multi-Class Classifier
Çok-Sınıflı Sınıflandırıcı

Image: Machine Learning: Classification by University of Washington

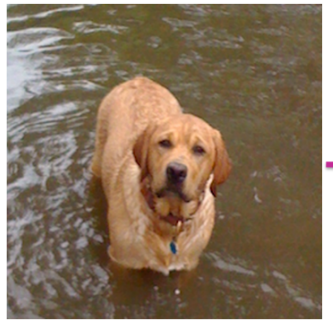
Classification Algorithms

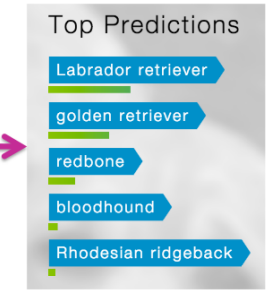
- Decision Trees
- Naive Bayes
- Linear Discriminant Analysis
- K-Nearest Neighbor
- Logistic Regression (Emre Alptekin)
- Neural Networks (Emre Alptekin)
- Support Vector Machines
- Hidden Markov Models
- Etc.



Classification Use Case Examples

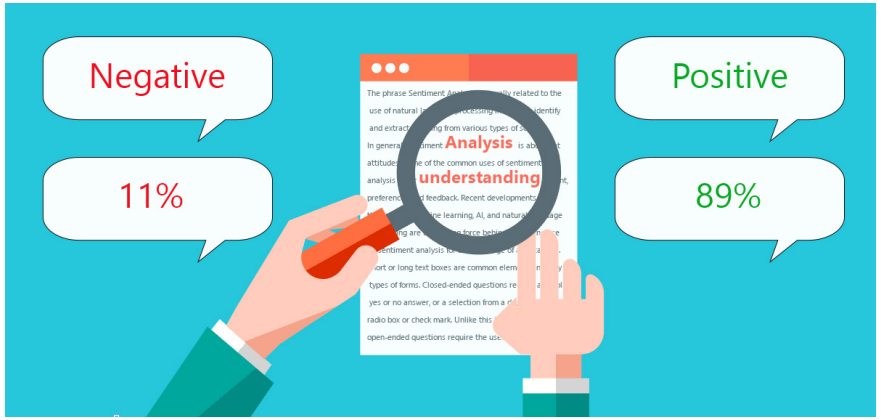
- Speech Recognition
- Spam Detection
- Churn Detection
- Fraud Detection
- Biometric Identification
- Face Recognition
- Document/Image Classification
- Sentiment Prediction
- Medical Diagnosis
- Activity Recognition





Input: x
Image pixels

Output: y
Predicted object



Loan default prediction example

age	ed	employ	income	debtinc	creddebt	othdebt	default
41	3	17	176	9.3	11.35	5.009	1
27	1	10	31	17.3	1.36	4.001	0
40	1	15	55	5.5	0.856	2.169	0
41	1	15	120	2.9	2.69	0.821	0
24	2	2	28	17.3	1.78	3.057	1
41	2	5	25	10.2	0.39	2.157	0
39	1	20	67	30.6	3.8	16.668	0

age	ed	employ	income	debtinc	creddebt	othdebt	default
37	2	16	130	9.3	10.23	3.21	?

Categorical Variable

Let's look at the loan default prediction problem

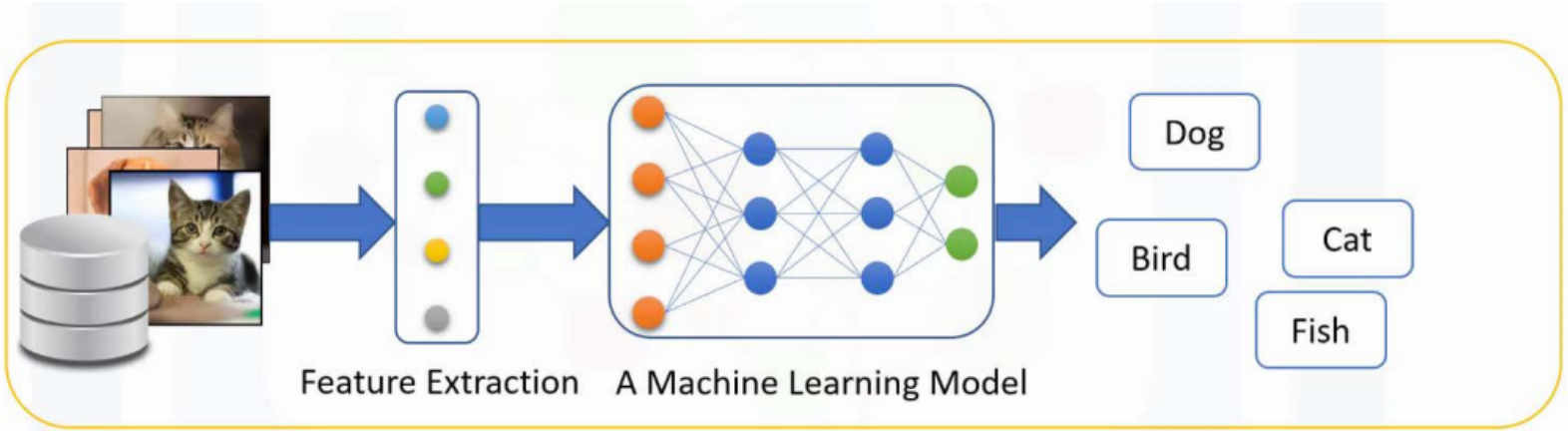
A similar and extended dataset can be found at: https://www.kaggle.com/c/loan-default-prediction/data

This example is a binary classifier with two values.

We can also build classifier models for both binary classification and multi-class classification.

Classification—A Two-Step Process

- **1: Model construction:** describing a set of predetermined classes (MODEL OLUŞTURMA)
  - Each tuple/sample is assumed to belong to a predefined class, as determined by the class label attribute (HER BİR ÖRNEK, ETİKETİ İLE TANIMLI BİR SINIFA AİT)
  - The set of tuples used for model construction is **training** set (EĞİTİM KÜMESİNDEN MODEL)
  - The model is represented as classification rules, decision trees, or mathematical formulae (MODEL: Kurallar, ağaçlar, matematiksel formüller)

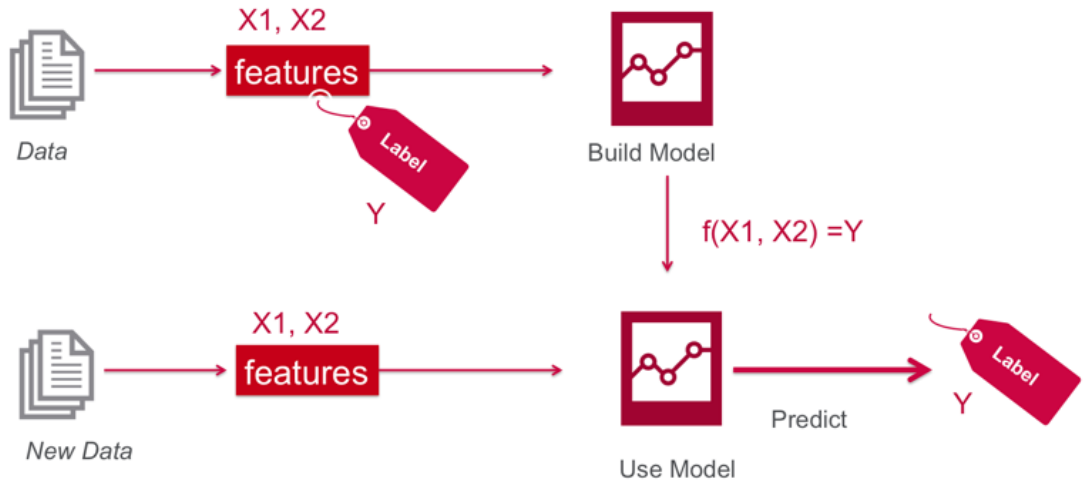


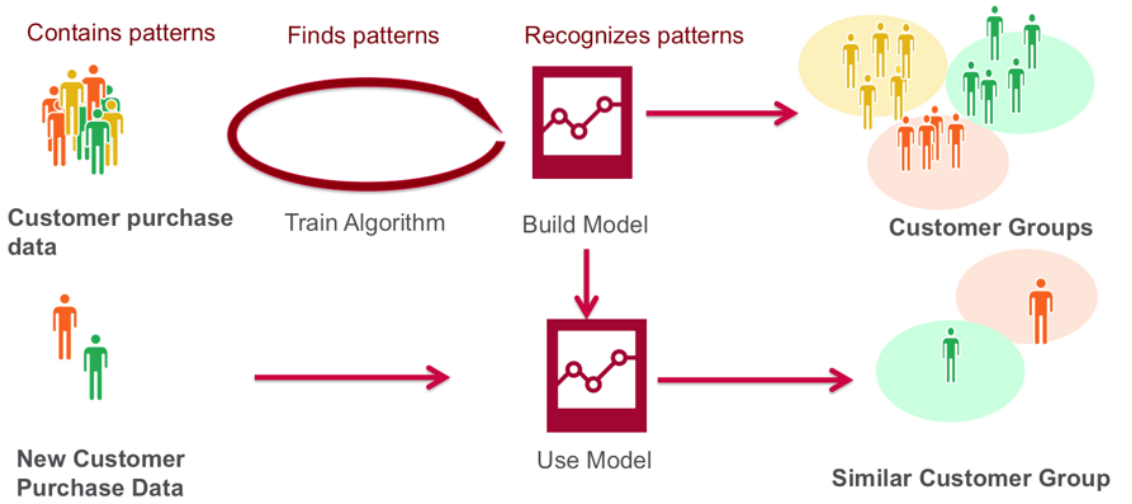
Classification—A Two-Step Process

- **2: Model usage**: for classifying future or unknown objects (Bilinmeyen örnekleri Tahmin Etme)
  - Estimate accuracy of the model (MODELİN BAŞARIMINI TAHMİN ETME)
    - The known label of test sample is compared with the classified result from the model
    - Accuracy rate is the percentage of **test** set samples that are correctly classified by the model (DOĞRULUK ORANI: modele göre doğru şekilde sınıflandırılmış test seti örneklerinin yüzdesi)
    - Test set is independent of training set (otherwise **overfitting**) (TEST KÜMESİ)
  - If the accuracy is acceptable, use the model to classify new data
- Note: If the test set is used to select models, it is called **validation** (test) set (DOĞRULAMA KÜMESİ: Test seti modelleri seçmek için kullanılıyorsa, doğrulama kümesi)

Supervised vs. Unsupervised Learning

- Supervised learning (e.g: classification)
  - Supervision: The training data (observations, measurements, etc.) are accompanied by **labels** indicating the class of the observations
  - New data is classified based on the training set
- Unsupervised learning (e.g: clustering)
  - The class labels of training data is unknown
  - Given a set of measurements, observations etc. with the aim of establishing the existence of classes or clusters in the data



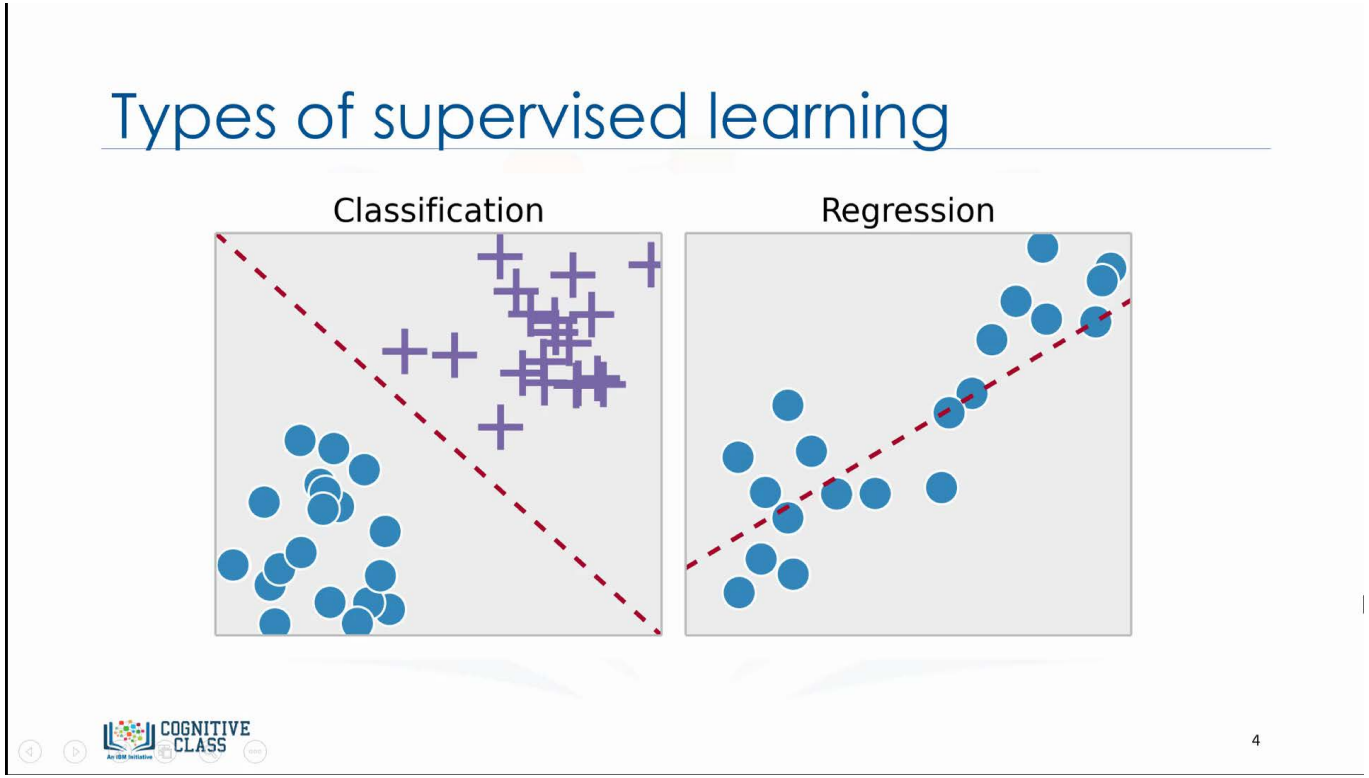


https://dzone.com/articles/demystifying-ai-machine-learning-and-deep-learning

En temel fark: Etiketler Yok

Prediction Problems: Classification vs. Numeric Prediction

- Classification
  - predicts categorical class labels (discrete or nominal)
  - classifies data (constructs a model) based on the training set and the values (class labels) in a classifying attribute and uses it in classifying new data
- Numeric Prediction
  - models continuous-valued functions, i.e., predicts unknown or missing values



2. K-Nearest Neighbor Classification Algorithm

GSÜSEM

GALATASARAY ÜNİVERSİTESİ

Sürekli Eğitim Uygulama ve Araştırma Merkezi

Intro to KNN, Example

- Example: a telecommunications provider has segmented the customer base by service usage patterns, categorizing the customers into four groups (Müşterileri kullanım özelliklerine göre 4 gruba ayırma)
- If demographic data can be used to predict group membership the company can customize offers for individual perspective customers (Demografik veri kullanma): This is a classification problem.
- The example focuses on using demographic data, such as; region, age, and marital status to predict usage patterns. The **target field** called custcat has four possible values:
  - Basic Service, E-Service, Plus Service, and Total Service.

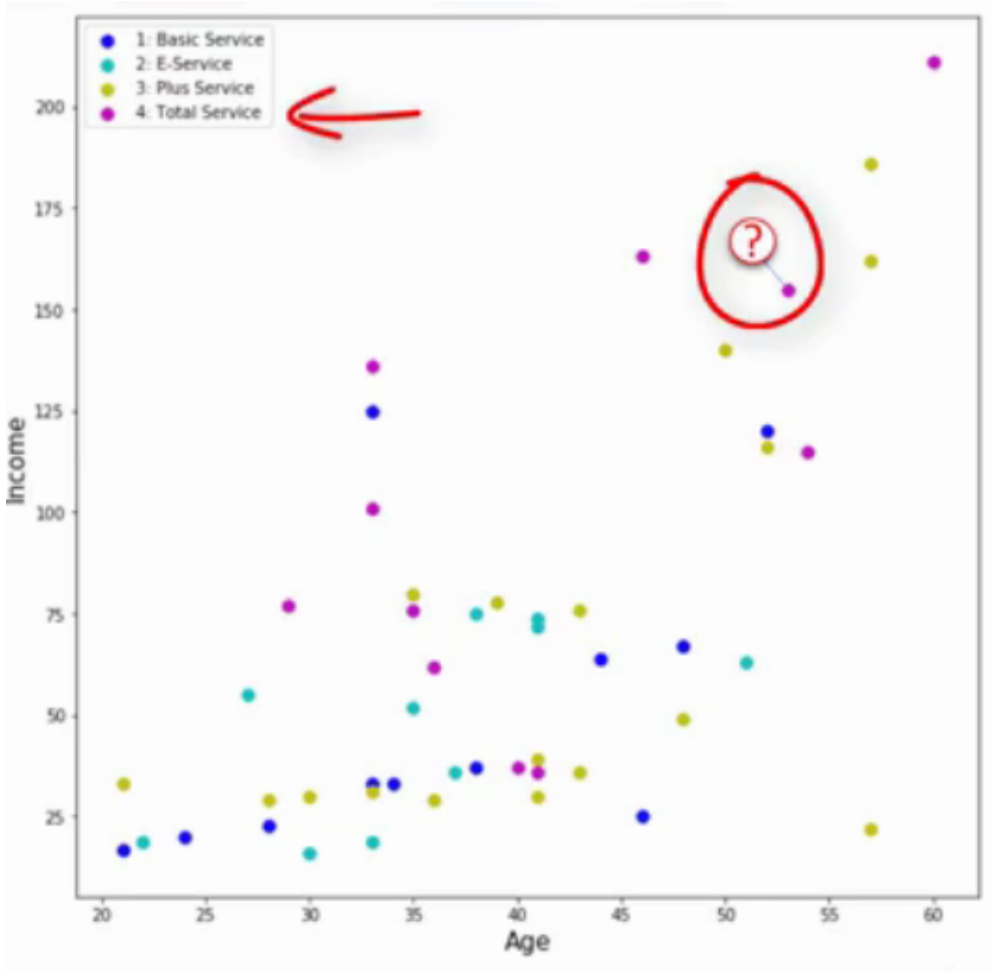
Example

region	age	marital	income	ed	employ	gender	Custcat
2	44	1	64	4	5	0	1
3	33	1	136	5	5	0	4
3	52	1	116	1	29	1	3
2	33	0	33	2	0	1	1
2	30	0	30	1	2	0	3
2	39	1	78	2	16	1	3
3	22	0	19	2	4	1	2
3	35	1	76	2	10	0	4
2	**50**	**1**	**166**	**4**	**31**	**0**	**?**

Value	Label
1	Basic Service
2	E-Service
3	Plus Service
4	Total Service

Determining the Class Using 1st -KNN

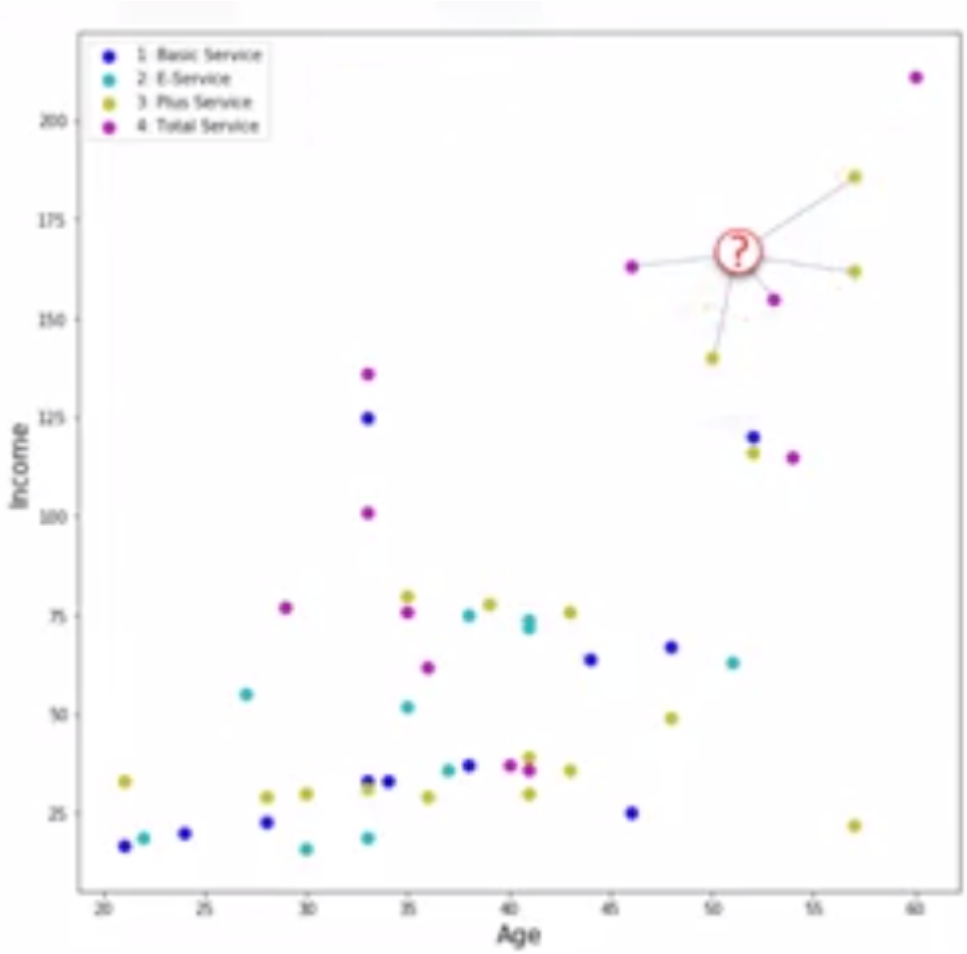
region	age	marital	income	ed	employ	gender	Custcat
2	44	1	64	4	5	0	1
3	33	1	136	5	5	0	4
3	52	1	116	1	29	1	3
2	33	0	33	2	0	1	1
2	30	0	30	1	2	0	3
2	39	1	78	2	16	1	3
3	22	0	19	2	4	1	2
3	35	1	76	2	10	0	4
2	**50**	**1**	**166**	**4**	**31**	**0**	**?**



1NN → 4:Total Service

Determining the Class Using 1st -KNN

region	age	marital	income	ed	employ	gender	Custcat
2	44	1	64	4	5	0	1
3	33	1	136	5	5	0	4
3	52	1	116	1	29	1	3
2	33	0	33	2	0	1	1
2	30	0	30	1	2	0	3
2	39	1	78	2	16	1	3
3	22	0	19	2	4	1	2
3	35	1	76	2	10	0	4
2	**50**	**1**	**166**	**4**	**31**	**0**	**?**

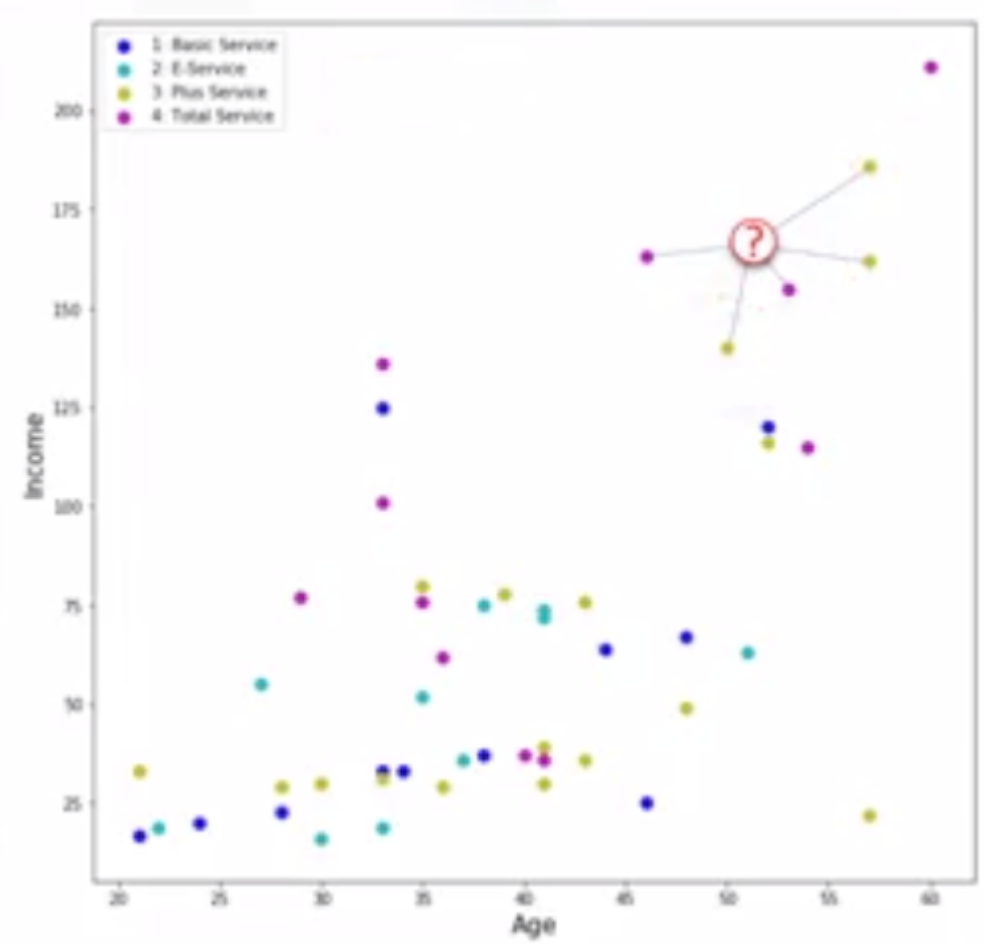


What if we chose the 5 nearest neighbors and did a majority voting?
En yakın 5 komşuyu seçip çoğunluk oylaması yapsaydık?

5NN → 3:Plus Service

What is K-Nearest Neighbors?

- A method that **classifies** cases based on their similarity to other cases (Benzerliğe göre sınıflandırma)
- Data points that are near each other are said to be **neighbors**. (Yakın olan veriler komşudur)
- Based on the idea that similar cases with the same class labels are near each other. (Aynı sınıf etiketi olan noktalar birbirine yakındır)
- Thus, the distance between two cases is a measure of their dissimilarity (MESAFE)



KNN Algorithm

How to select K?
How to calculate the similarity/distance?

1) Pick a value for K – K değerini seç
2) Calculate the distance of the unknown case from all cases Bilinmeyen örneğin diğer örneklere olan mesafesini hesapla
3) Select the K-observations in the training data that are “nearest” to the unknown data point – Eğitim kümesinden, bilinmeyen örneğe en yakın k tane en yakın komşuyu seç
4) Predict the response of the unknown data point using the most popular response value from the K-Nearest Neighbors – K-komşu arasındaki en popüler cevabı seçerek, bilinmeyen noktanın etiketini tahmin et

Closeness – Distance Calculation

- "Closeness" is defined in terms of a distance metric, such as Euclidean distance (Mesafe metriği kullanarak yakınlık hesabı)
- The Euclidean distance between two points or tuples, say, ÖKLİT mesafesi:
  - $X_1 = (x_{11}, x_{12}, \ldots, x_{1n})$ and
  - $X_2 = (x_{21}, x_{22}, \ldots, x_{2n})$, is

$$dist(\boldsymbol{X_1}, \boldsymbol{X_2}) = \sqrt{\sum_{i=1}^{n} (x_{1i} - x_{2i})^2}$$

Calculating the distance in a 1-dimensional Space

- How can we calculate the similarity between two data points?
- We can easily use a specific type of Minkowski distance to calculate the distance of these two customers, which is the Euclidean distance.

Cust 1
Age: 54

Cust 2
Age: 50

$Dis(x_1, x_2) = 4$

$$dist(\boldsymbol{X_1}, \boldsymbol{X_2}) = \sqrt{\sum_{i=1}^{n} (x_{1i} - x_{2i})^2}$$

- What about if we have more than one feature?

Calculating the distance in a multi-dimensional Space

- If we have income, age and education for each customer, we can also use the same distance matrix for multidimensional vectors.
- We have to normalize our feature set to get the accurate dissimilarity measure.

Cust 1
Age: 54 Income:190 Ed: 3

Cust 2
Age: 50 Income:200 Ed:8

Dis(x_1,x_2) = 11.87

$$dist(\boldsymbol{X_1}, \boldsymbol{X_2}) = \sqrt{\sum_{i=1}^{n} (x_{1i} - x_{2i})^2}$$

Normalization?

- Typically, we normalize the values of each attribute before using distance Equation
- This helps prevent attributes with initially large ranges (such as income) from outweighing attributes with initially smaller ranges (such as marital status) - Daha büyük aralıkları olan (gelir gibi) özelliklerin daha küçük aralıkları olan (medeni durum gibi) nitelikleri bastırmasını önler.
- Min-max normalization, for example, can be used to transform a value v of a numeric attribute A to v' 0 in the range [0, 1] by computing

$$v' = \frac{v - min_A}{max_A - min_A}$$

Non-numeric Attributes?

- How can distance be computed for attributes that are not numeric, such as color?
- For categorical attributes, a simple method is to compare the corresponding value of the attribute in tuple X_1 with that in tuple X_2.
  - If the two are identical (e.g., tuples X_1 and X_2 both have the color blue), then the difference between the two is taken as 0.
  - If the two are different (e.g., tuple X_1 is blue but tuple X_2 is red), then the difference is considered to be 1.
  - Other methods may incorporate more sophisticated schemes for differential grading (e.g., where a larger difference score is assigned, say, for blue and white than for blue and black).

Other Similarity Measures?

- Manhattan Distance (City-Block Distance)

$$d(i, j) = |x_{i1} - x_{j1}| + |x_{i2} - x_{j2}| + \cdots + |x_{in} - x_{jn}|$$

- Minkowski Distance

$$d(i, j) = (|x_{i1} - x_{j1}|^p + |x_{i2} - x_{j2}|^p + \cdots + |x_{in} - x_{jn}|^p)^{1/p}$$

where p is a positive integer. Such a distance is also called L_p norm

What is the Best Value for K?

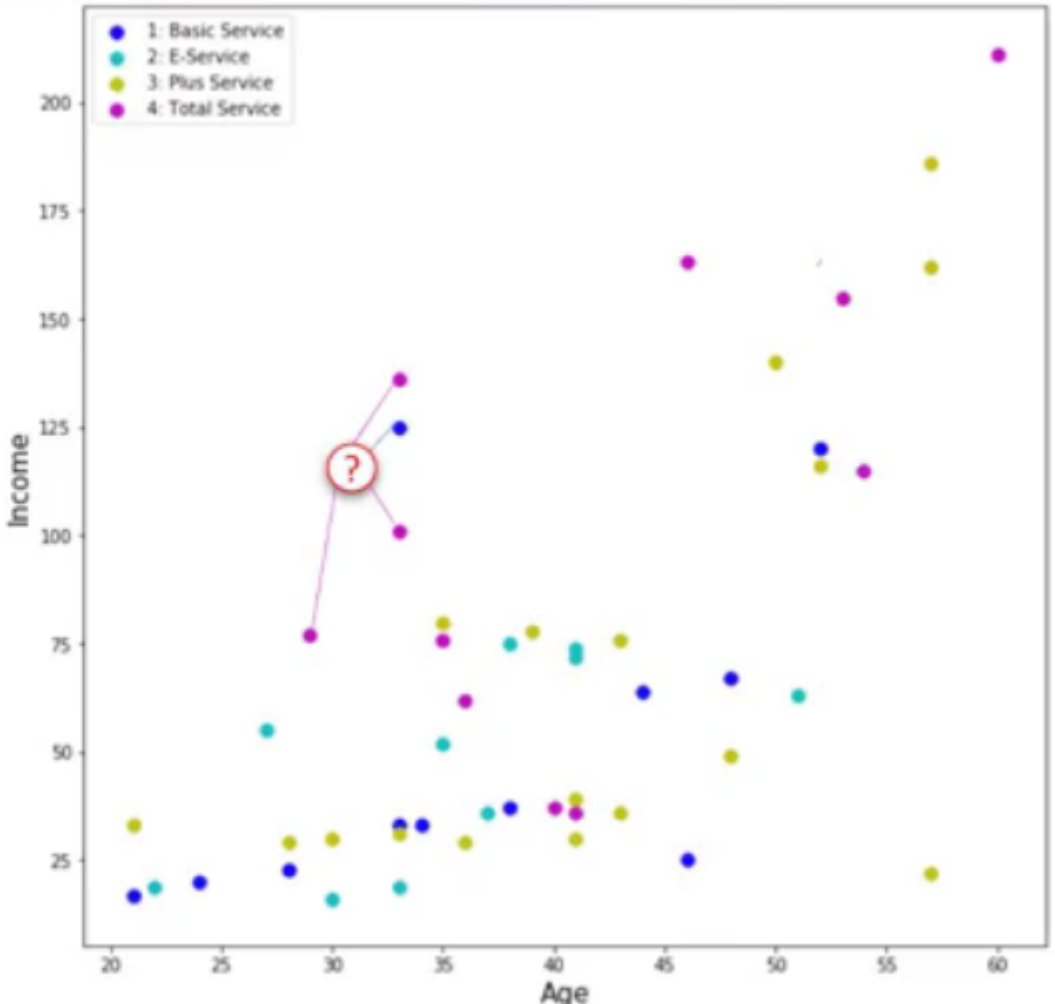
- The general solution is to reserve a part of your data for testing the accuracy of the model.
- Once you've done so, choose K equals one and then use the training part for modeling and calculate the accuracy of prediction using all samples in your test set.
- Repeat this process increasing the K and see which K is best for your model.

Verinin bir kısmı test için ayrılır, K=1'den başlayarak test seti üzerinde doğruluk hesaplanır, K arttırılarak devam edilir ve Knın en iyi sonuç verdiği değer seçilir

What is the Best Value for K?

Common to use also:
k = sqrt(N), where N stands for the number of samples in your training dataset.

- K =1 class 1
- K =20 ?



Summary and Critics for KNN

- Nearest-neighbor classifiers use distance-based comparisons that intrinsically assign equal weight to each attribute. Therefore can suffer from poor accuracy when given noisy or irrelevant attributes.
- The method, however, has been modified to incorporate attribute weighting and the pruning of noisy data tuples.
- The choice of a distance metric can be critical.
- Nearest-neighbor classifiers can be extremely slow when classifying test tuples. If D is a training database of $|D|$ tuples and $k = 1$, then $O(|D|)$ comparisons are required in order to classify a given test tuple.

KNN Summary – A Lazy Learner (or Learning from Your Neighbors)

- Unlike eager learning methods, lazy learners do less work when a training tuple is presented and more work when making a classification or prediction.
- Because lazy learners store the training tuples or “instances,” they are also referred to as instance-based learners, even though all learning is essentially based on instances
- When making a classification or prediction, lazy learners can be computationally expensive. They require efficient storage techniques and are well-suited to implementation on parallel hardware.
- They offer little explanation or insight into the structure of the data.

KNN Example – Customer Categorization

- Imagine a telecommunications provider has segmented its customer base by service usage patterns, categorizing the customers into four groups. If demographic data can be used to predict group membership, the company can customize offers for individual prospective customers. The target field, called “custcat”, has four possible values that correspond to the four customer groups, as follows:",
- 1- Basic Service, 2- E-Service, 3- Plus Service, 4- Total Service
- Our objective is to build a classifier, to predict the class of unknown cases using KNN
- We will use the data file “teleCust1000t.csv” which has 1000 instances

3. Evaluation Metrics in Classification

GSÜSEM

GALATASARAY ÜNİVERSİTESİ

Sürekli Eğitim Uygulama ve Araştırma Merkezi

Classifier Evaluation – Performansı Değerlendirme

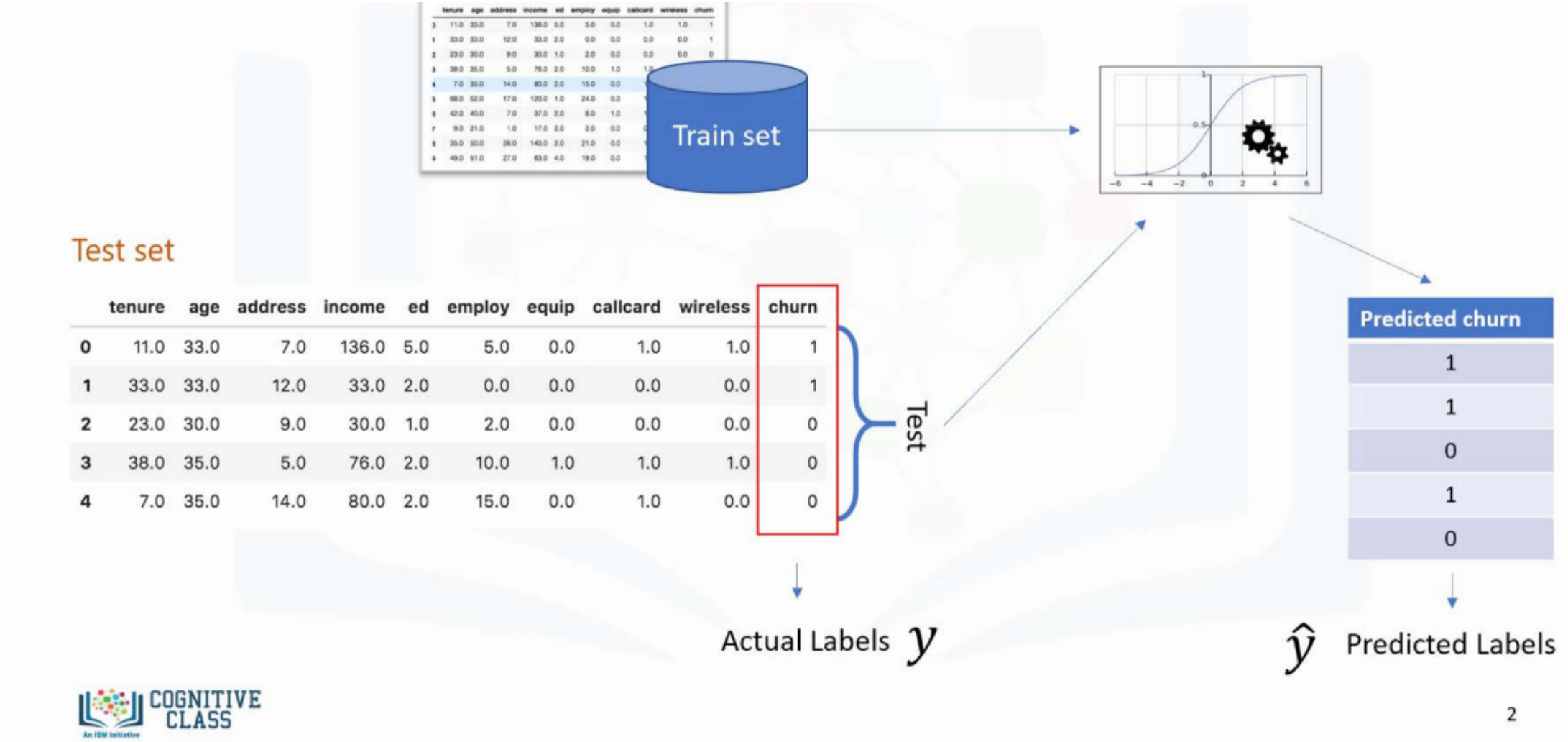
- <u>Evaluation metrics explain the performance of a model.</u>
- We have **trained** the model, and now we want to calculate its success using the **test set.** We pass the test set to our model, and we find the predicted labels.
- Now the question is, “How accurate is this model?” Basically, we compare the actual values in the test set with the values predicted by the model, to calculate the accuracy of the model.

WHY SPLIT? Neden Ayırıyoruz

Overfitting Problem: The model is overly trained to the dataset, which may capture noise and produce a non-generalized model(i.e., during learning it may incorporate some particular anomalies of the training data that are not present in the general data set overall).

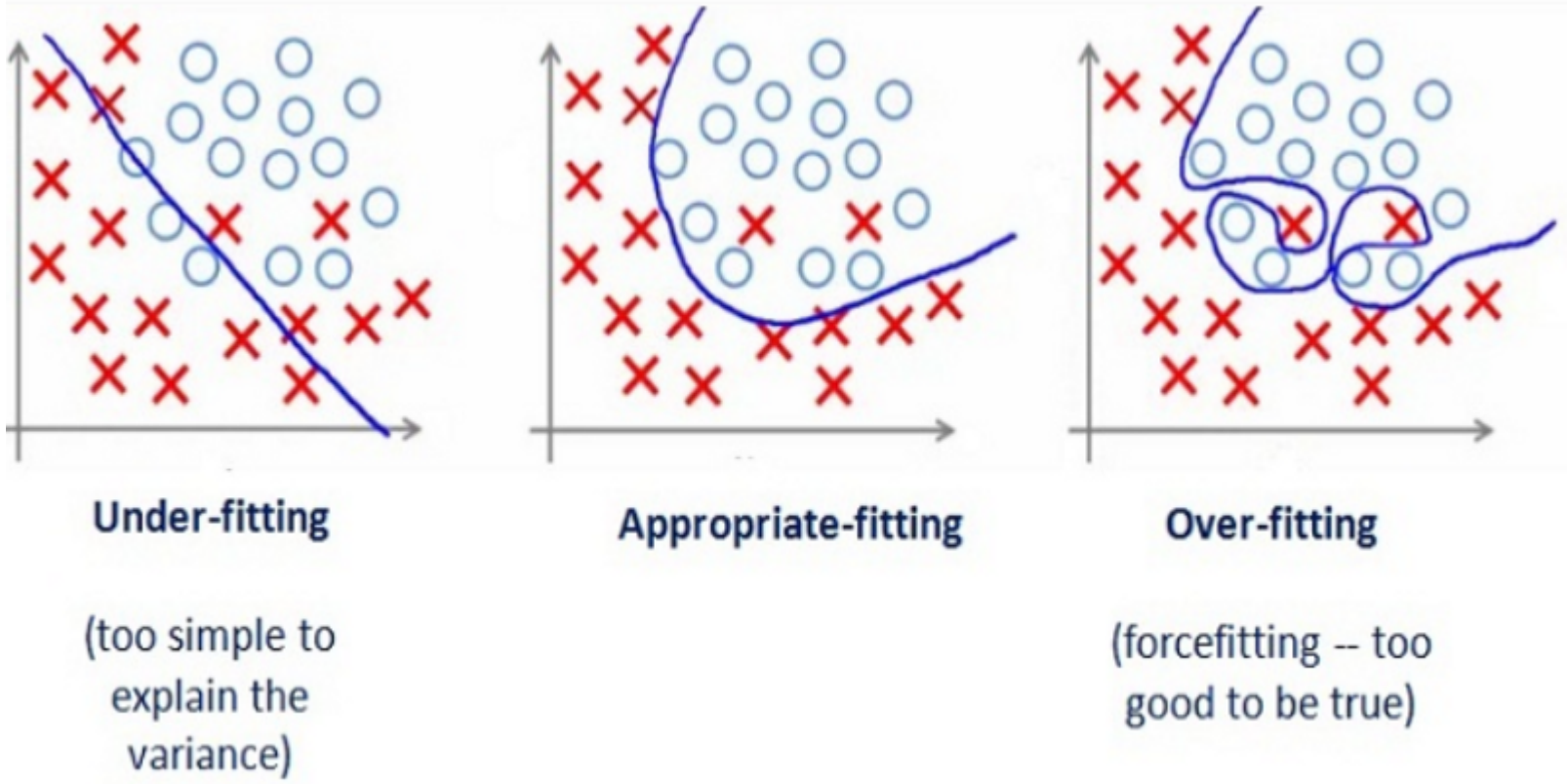
Out of sample accuracy: It is important that our models have a high, out of sample (data that was unseen) accuracy

Classifier Evaluation



	tenure	age	address	income	ed	employ	equip	callcard	wireless	churn
0	11.0	33.0	7.0	136.0	5.0	5.0	0.0	1.0	1.0	1
1	33.0	33.0	12.0	33.0	2.0	0.0	0.0	0.0	0.0	1
2	23.0	30.0	9.0	30.0	1.0	2.0	0.0	0.0	0.0	0
3	38.0	35.0	5.0	76.0	2.0	10.0	1.0	1.0	1.0	0
4	7.0	35.0	14.0	80.0	2.0	15.0	0.0	1.0	0.0	0

Overfitting – Underfitting?



Underfitting: Yetersiz Oturma
Overfitting: Aşırı Oturma, aşırı öğrenme

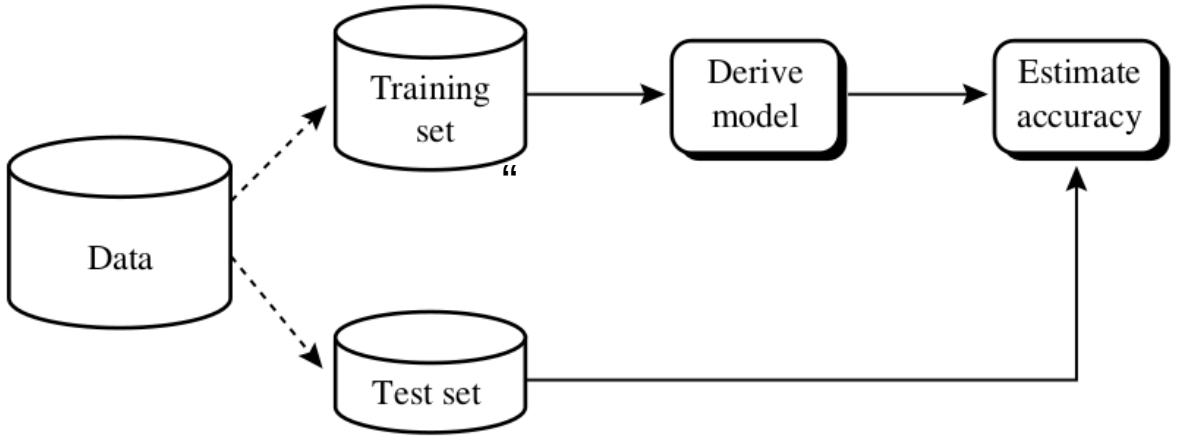
Overfitting refers to a model that models the training data too well.

Overfitting happens when a model learns the detail and noise in the training data to the extent that it negatively impacts the performance of the model on new data.

Underfitting refers to a model that can neither model the training data nor generalize to new data.

Holdout Method

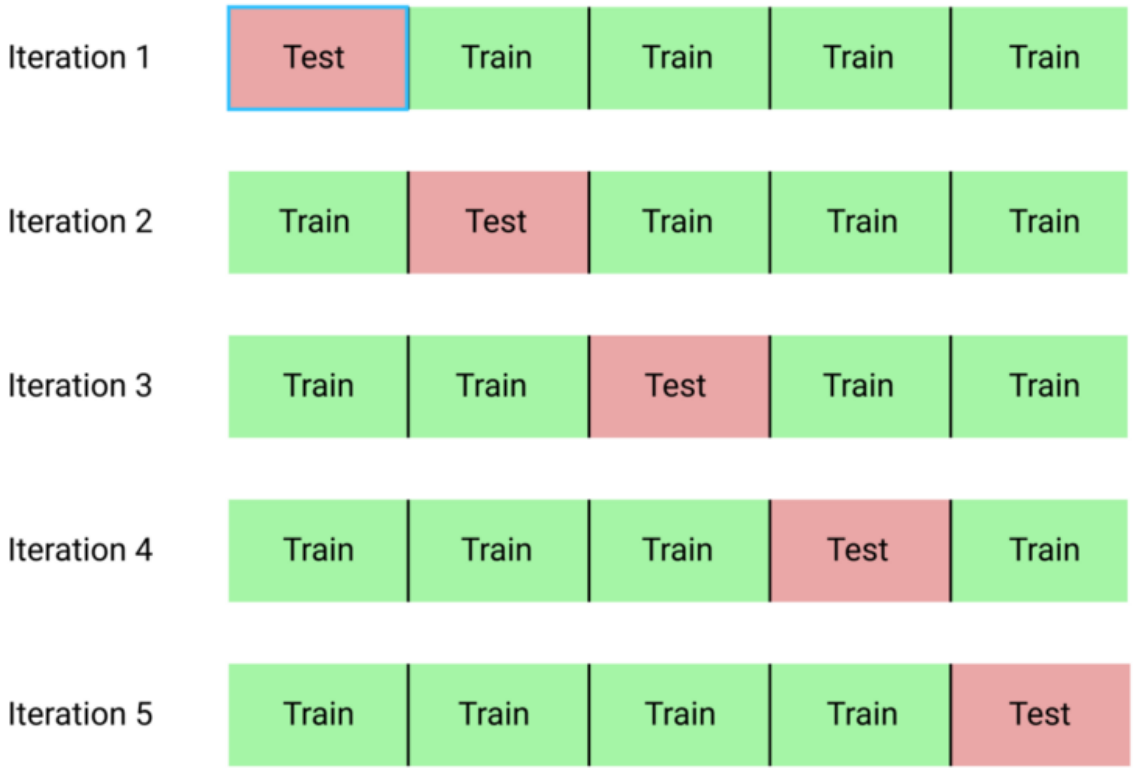
- The given data are randomly partitioned into two independent sets, a training set and a test set.
- Typically, two-thirds of the data are allocated to the training set, and the remaining one-third is allocated to the test set.
- The training set is used to derive the model, whose accuracy is estimated with the test set.



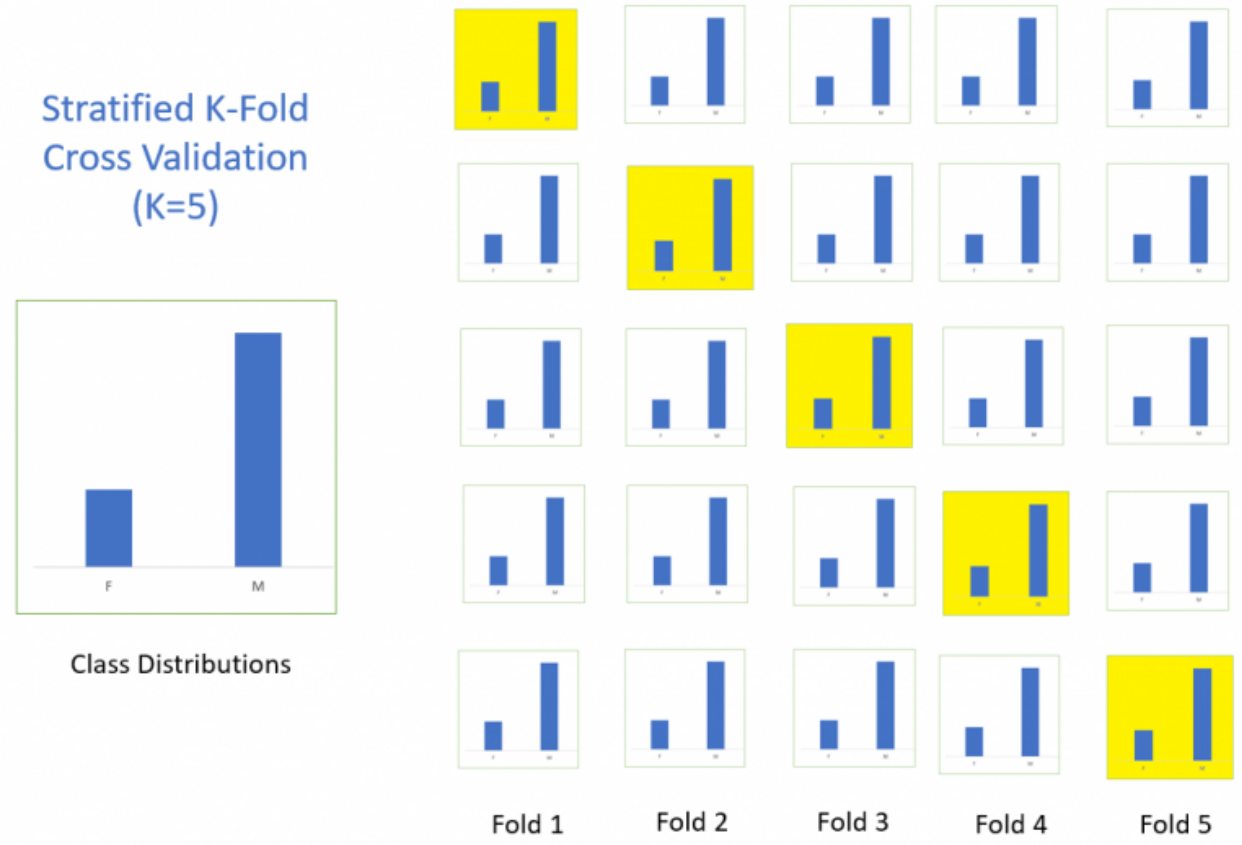
Cross -Validation Method

- In k-fold cross-validation, the initial data are randomly partitioned into k mutually exclusive subsets or “folds,” $D_1, D_2, \ldots, D_k$, each of approximately equal size.
- Training and testing is performed k times. In iteration i, partition D_i is reserved as the test set, and the remaining partitions are collectively used to train the model.
- Unlike the holdout method, here, each sample is used the same number of times for training and once for testing. For classification, the accuracy estimate is the overall number of correct classifications from the k iterations, divided by the total number of tuples in the initial data.

Cross -Validation Method



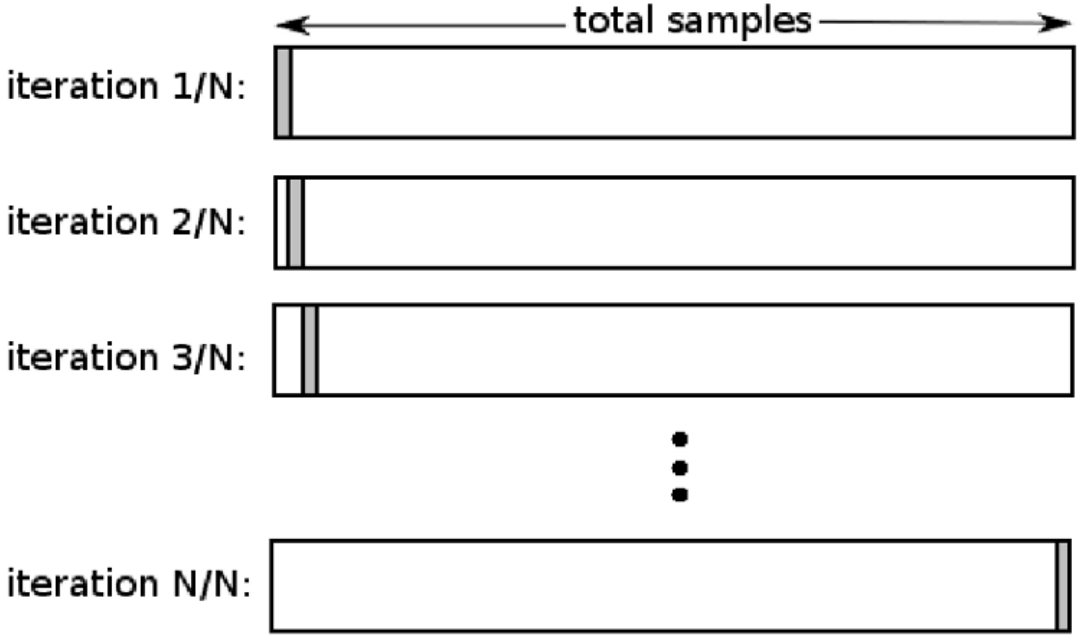
Cross Validation in general



Stratified Cross Validation

Leave-One-Out Cross Validation

- 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.
- In general, stratified 10-fold cross-validation is recommended for estimating accuracy (even if computation power allows using more folds) due to its relatively low bias and variance.



Confusion Matrix

- Given m classes, a confusion matrix is a table of at least size m by m. An entry, $CM_{i,j}$ in the first m rows and m columns indicates the number of tuples of class i that were labeled by the classifier as class j. Each column of the matrix represents the instances in a predicted class, while each row represents the instances in an actual class.

Predicted Class

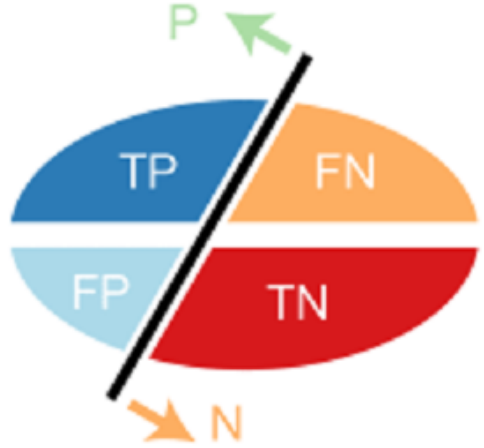
Actual Class	**C1**	**C2**
C1	True Positives	False Negatives
C2	False Positives	True Negatives

True Positive (TP): Correct hit

True Negative (TN): Correct rejection

False Positive (FP): False Alarm

False Negative (FN): Miss



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**

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?
- Perfect score is 1.0
- Inverse relationship between precision & recall

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

- ***F* measure (*F_1* or *F*-score)**: harmonic mean of precision and recall,

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

Classifier Evaluation Metrics: Example

Actual Class\Predicted class	cancer = yes	cancer = no	Total	Recognition(%)
cancer = yes	**90**	**210**	300	30.00 (*sensitivity*
cancer = no	**140**	**9560**	9700	98.56 (*specificity*)
Total	230	9770	10000	96.40 (*accuracy*)

- *Precision* = 90/230 = 39.13% *Recall* = 90/300 = 30.00%

F-Measure

F1-score or F-score: harmonic mean of precision and recall,

$$F\_score = \frac{recall \times precision}{(recall + precision)/2}$$

Actual Class\Predicted class	cancer = yes	cancer = no	Total	Recognition(%)
cancer = yes	**90**	**210**	300	30.00 (*sensitivity*
cancer = no	**140**	**9560**	9700	98.56 (*specificity)*
Total	230	9770	10000	96.40 (*accuracy*)

- *Precision* = 90/230 = 39.13% *Recall* = 90/300 = 30.00% F-Measure = 33.96%

Multi-class Calculation

- Let's say we have three classes: Red, Yellow, Other. How do we compute these for each class?

Actual Class\Predicted class	red	yellow	Other	Total
red	**100**	30	50	180
yellow	20	**90**	40	150
other	30	20	**120**	170
total	150	140	210	400

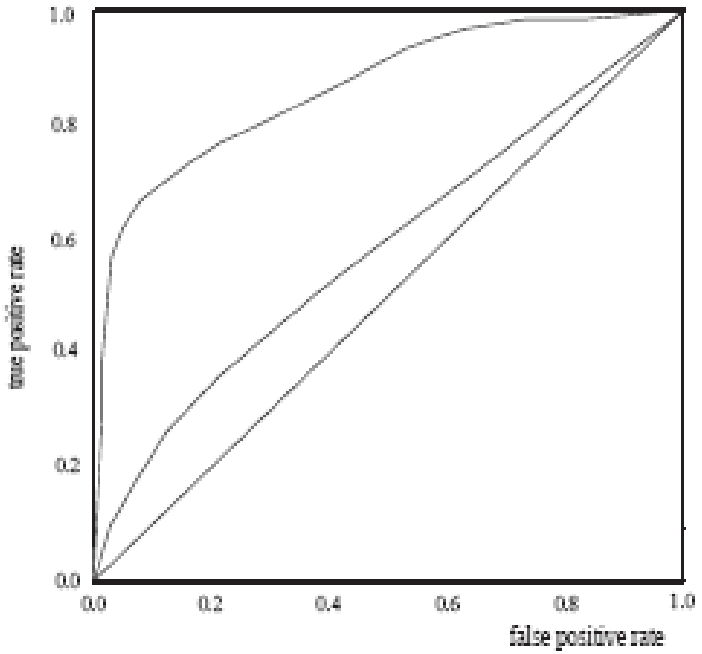
- *Precision (TP/TP+FP):*
  - *Red* = 100/150
  - Yellow = 90/140
  - Other = 120/210
- *Recall (TP/TP+FN):*
  - *Red* = 100/180
  - Yellow = 90/150
  - Other = 120/170

Other Measures?

- **Speed**
  - time to construct the model (training time)
  - time to use the model (classification/prediction time)
- **Robustness**: handling noise and missing values
- **Scalability**: efficiency in disk-resident databases
- **Interpretability:** understanding and insight provided by the model
- Other measures, e.g., goodness of rules, such as decision tree size or compactness of classification rules

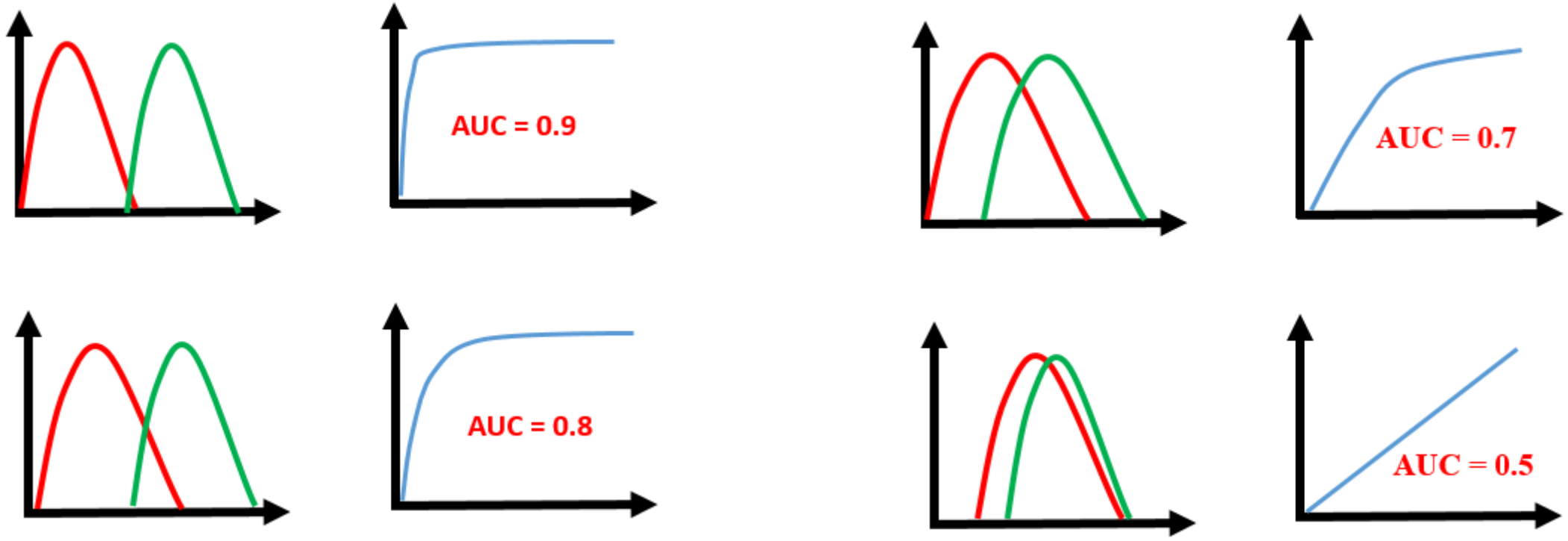
Model Selection – ROC and AUC

- **ROC** (Receiver Operating Characteristics) curves: for visual comparison of classification models
- Shows the trade-off between the true positive rate and the false positive rate
- The area under the ROC curve (AUC) is a measure of the accuracy of the model
- Rank the test tuples in decreasing order: the one that is most likely to belong to the positive class appears at the top of the list
- The closer to the diagonal line (i.e., the closer the area is to 0.5), the less accurate is the model



- Vertical axis represents the true positive rate
- Horizontal axis rep. the false positive rate
- The plot also shows a diagonal line
- A model with perfect accuracy will have an area of 1.0

Model Selection – ROC and AUC



Overfitting – Underfitting?

When two models have similar classification error on the validation set, pick the simpler one

Complexity	Train error	Validation error
Simple	0.23	0.24
Moderate	0.12	0.15
Complex	0.07	0.15
Super complex	0	0.18

Same validation error

pick

bad!

Overfit

Eğer iki modelin doğrulama seti üzerindeki sınıflandırma hatası aynıysa, basit olan seçilir

Image: Machine Learning: Classification by University of Washington

4. Decision Tree Classification Algorithm

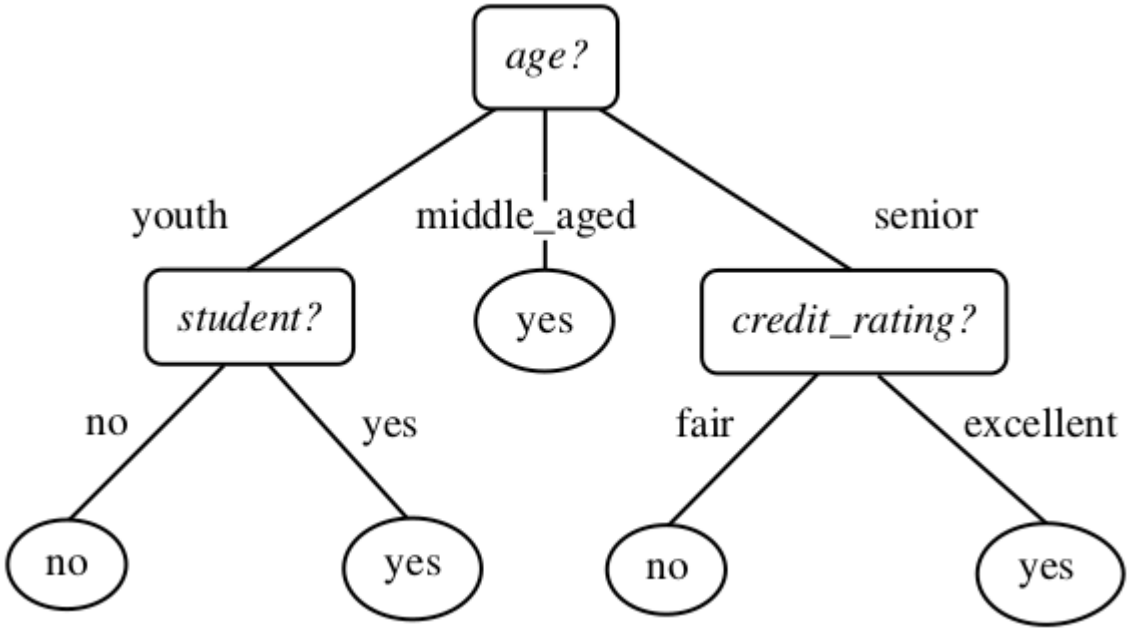
GSÜSEM

GALATASARAY ÜNİVERSİTESİ

Sürekli Eğitim Uygulama ve Araştırma Merkezi

What is a Decision Tree?

- A decision tree is a flowchart-like tree structure, where each internal node (nonleaf node) denotes a test on an attribute, each branch represents an outcome of the test, and each leaf node (or terminal node) holds a class label (Akış Diyagramı – Yapraklar: Sınıf, Ara Düğümler: Test/Karar, Dal: Test Sonucu)
- The basic intuition behind a decision tree is to map out all decision paths in the form of a tree



A decision tree for the concept buys computer, indicating whether a customer at AllElectronics is likely to purchase a computer.

Each internal (nonleaf) node represents a test on an attribute.

Each leaf node represents a class (either buys computer = yes or buys computer = no).

An Example Problem

Drug Prescription Problem

A list of patients and the drugs that they responded to

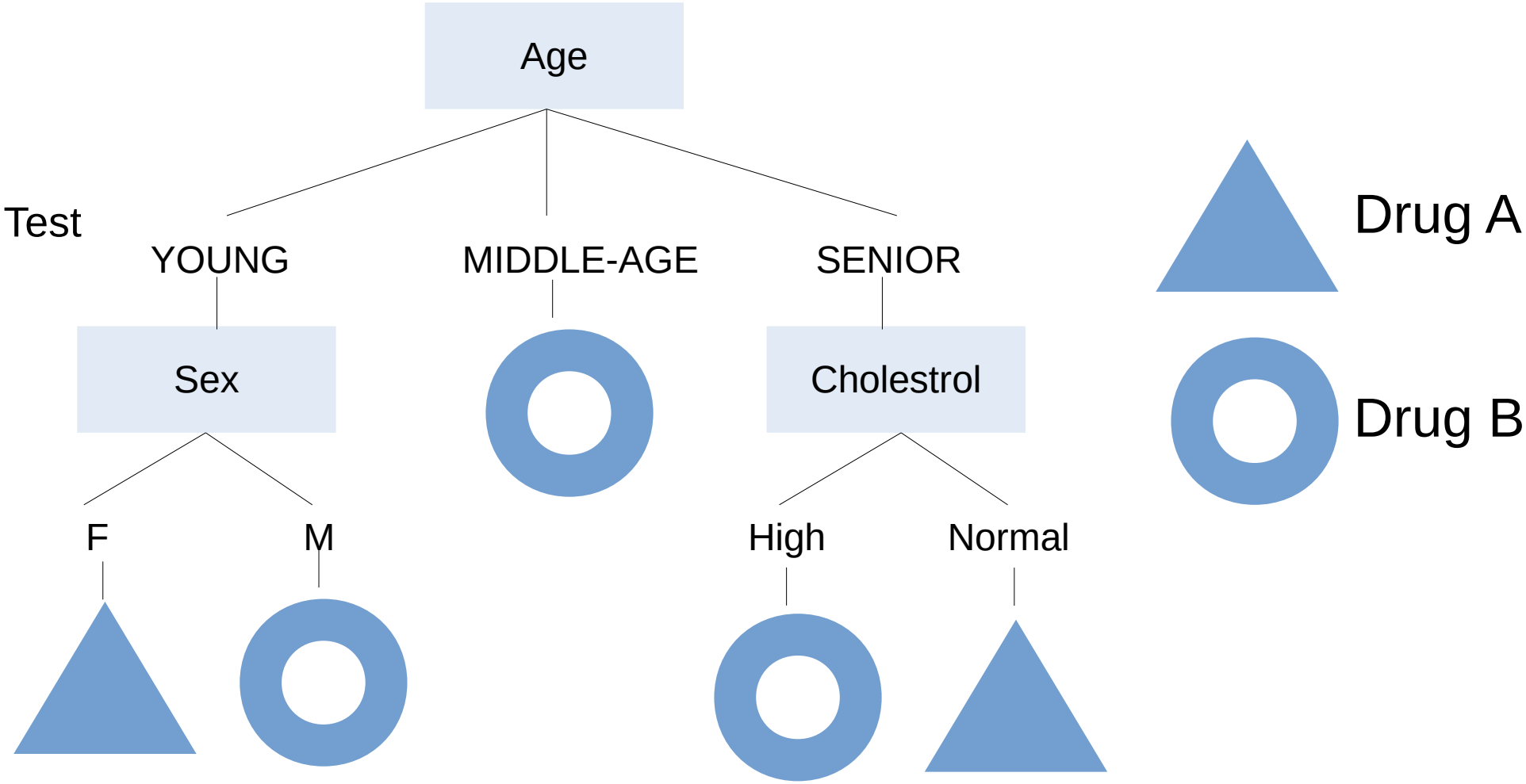
Predict which drug is good for a new patient

Age	Sex	BP	Cholesterol	Na_to_K	Drug
23	F	HIGH	HIGH	25.355	drugA
47	M	LOW	HIGH	13.093	drugB
47	M	LOW	HIGH	10.114	drugB
28	F	NORMAL	HIGH	7.798	drugA
61	F	LOW	HIGH	18.043	drugB
22	F	NORMAL	HIGH	8.607	drugA
49	F	NORMAL	HIGH	16.275	drugB
41	M	LOW	HIGH	11.037	drugA
60	M	NORMAL	HIGH	15.171	???

Building a Decision Tree with the training set

Internal Node: Test
Branch: Result of a Test
Leaf: Class

Ara Düğüm: Test
Dal: Test Sonucu
Yaprak: Sınıf



Decision tree learning algorithm

- Step 1: Start with an empty tree
- Step 2: Select a feature to split data
- For each split of the tree:
  - Step 3: If nothing more to, make predictions
  - Step 4: Otherwise, go to Step 2 & continue (recurse) on this split

Problem 1: Feature split selection

Problem 2: Stopping condition

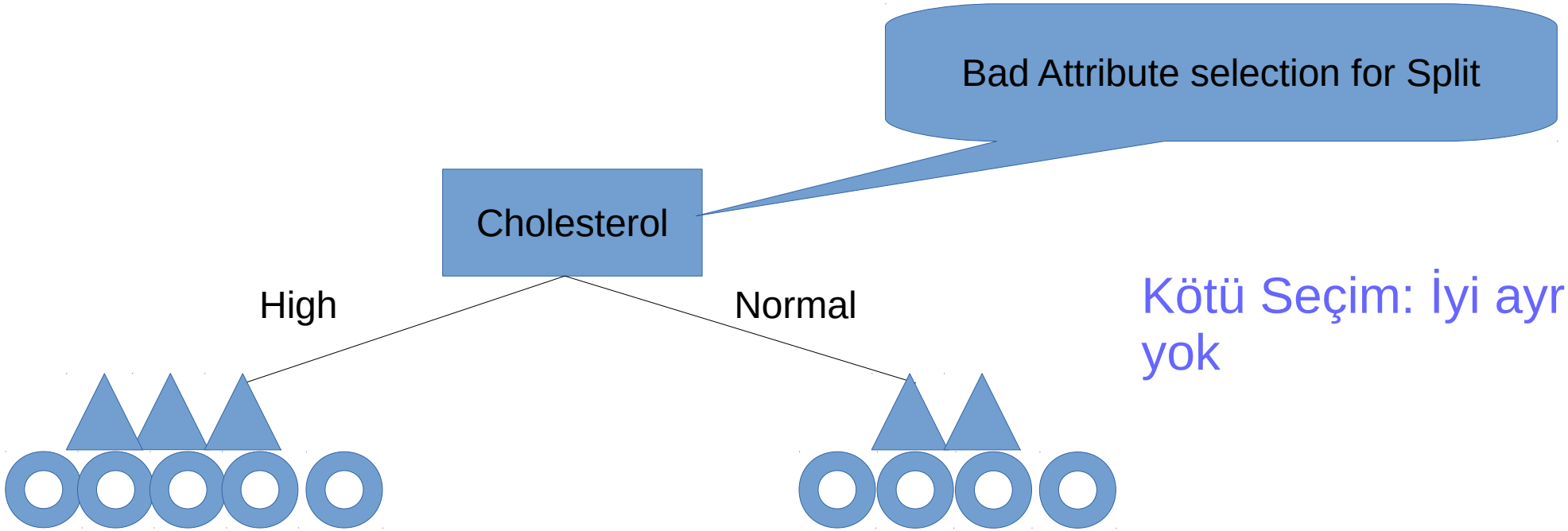
Recursion

Image: Machine Learning: Classification by University of Washington

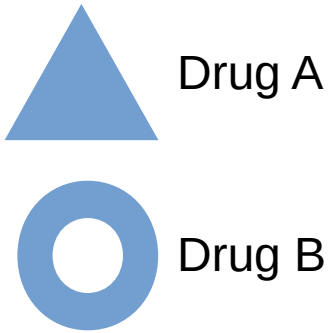
Decision Tree Learning Algorithm

- The basic idea behind any decision tree algorithm is as follows:
  - Choose the best attribute(s) to split the remaining instances and make that attribute a decision node Bölme için en iyi Özniteliği Seç
  - Repeat this process for recursively for each child – Her çocuk düğüm için yineleyerek devam et
  - Stop when: Dur
    - All the instances have the same target attribute value: Verideki tüm örnekler aynı sınıfa aitse
    - There are no more attributes – Bakacak öznitelik kalmadıysa
    - There are no more instances – Örnek veri kalmadıysa
- After building this tree, you can use it to predict the class of unknown cases TAHMİN

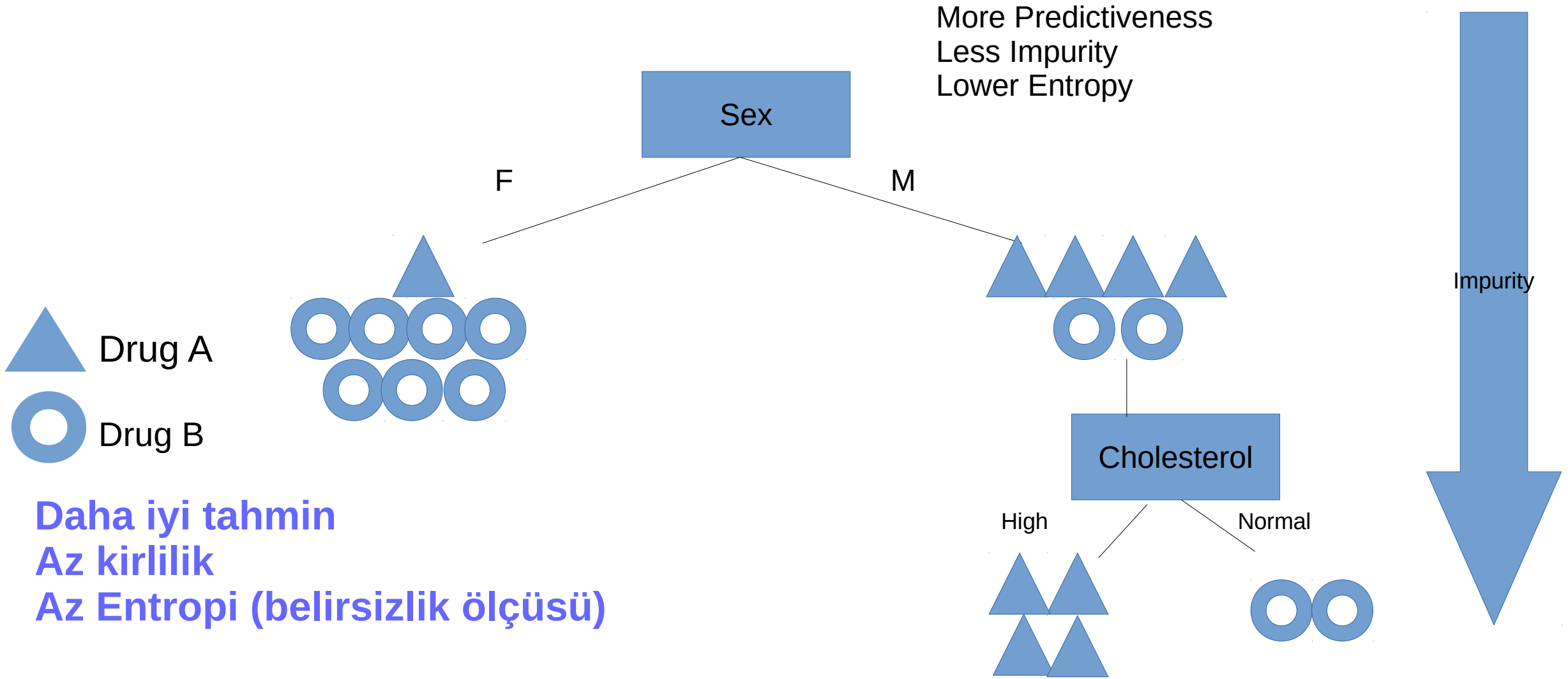
Which attribute is the Best?
Hangi öznitelik en iyisi



Kötü Seçim: İyi ayrışma yok



Which attribute is the Best?



Daha iyi tahmin
Az kirlilik
Az Entropi (belirsizlik ölçüsü)

Entropy

- Measure of randomness or uncertainty (Rassalık ya da Belirsizlik Ölçüsü)
- The lower the entropy, the less uniform the distribution, the purer the node, lower uncertainty – Entropi düştükçe, belirsizlik düşer
- Entropy, Info(D) is just the average amount of information needed to identify the class label of a tuple in D- bir örneğin sınıf etiketini tanımlamak için gereken ortalama bilgi miktarıdır

1 DrugA 7 DrugB	Entropy is low
3 DrugA 5 DrugB	Entropy is High
0 DrugA 8DrugB	Entropy 0
4 Drug A 4 DrugB	Entropy 1

$$Info(D) = -\sum_{i=1}^{m} p_i \log_2(p_i)$$

Note that the minus sign takes care of the fact that p_i is a fraction.

How to Calculate Entropy

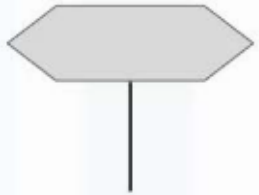
Patient ID	Age	Sex	BP	Cholesterol	Drug
p1	Young	F	High	Normal	Drug A
p2	Young	F	High	High	Drug A
p3	Middle-age	F	Hiigh	Normal	Drug B
p4	Senior	F	Normal	Normal	Drug B
p5	Senior	M	Low	Normal	Drug B
p6	Senior	M	Low	High	Drug A
p7	Middle-age	M	Low	High	Drug B
p8	Young	F	Normal	Normal	Drug A
p9	Young	M	Low	Normal	Drug B
p10	Senior	M	Normal	Normal	Drug B
p11	Young	M	Normal	High	Drug B
p12	Middle-age	F	Normal	High	Drug B
p13	Middle-age	M	High	Normal	Drug B
p14	Senior	F	Normal	High	Drug A

S: [9 B, 5 A]

E = - p(B)log(p(B)) - p(A)log(p(A))

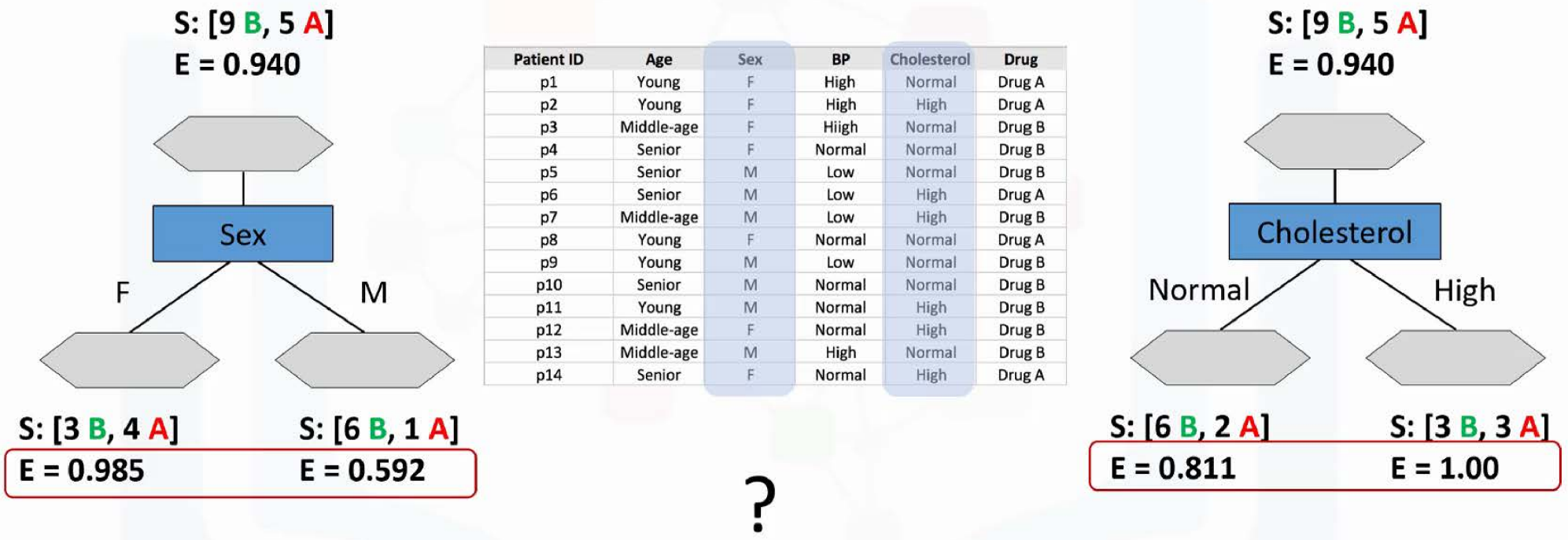
E = - (9/14)log(9/14) - (5/14)log(5/14)

E = 0.940



$$Info(D) = -\sum_{i=1}^{m} p_i \log_2(p_i)$$

Which attribute is the best?



Patient ID	Age	Sex	BP	Cholesterol	Drug
p1	Young	F	High	Normal	Drug A
p2	Young	F	High	High	Drug A
p3	Middle-age	F	Hiigh	Normal	Drug B
p4	Senior	F	Normal	Normal	Drug B
p5	Senior	M	Low	Normal	Drug B
p6	Senior	M	Low	High	Drug A
p7	Middle-age	M	Low	High	Drug B
p8	Young	F	Normal	Normal	Drug A
p9	Young	M	Low	Normal	Drug B
p10	Senior	M	Normal	Normal	Drug B
p11	Young	M	Normal	High	Drug B
p12	Middle-age	F	Normal	High	Drug B
p13	Middle-age	M	High	Normal	Drug B
p14	Senior	F	Normal	High	Drug A

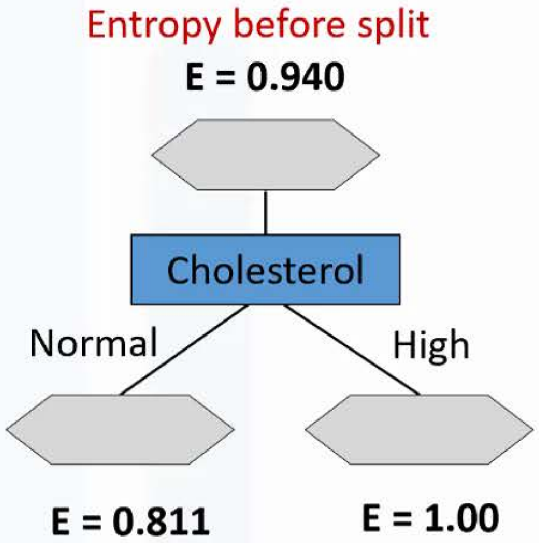
The tree with the higher Information Gain after splitting.

Bölmeden sonra Bilgi Kazancı Daha Yüksek olan Ağaç

What is information gain?

Information gain is the information that can increase the level of certainty after splitting.

Information Gain = (Entropy before split) – (weighted entropy after split)

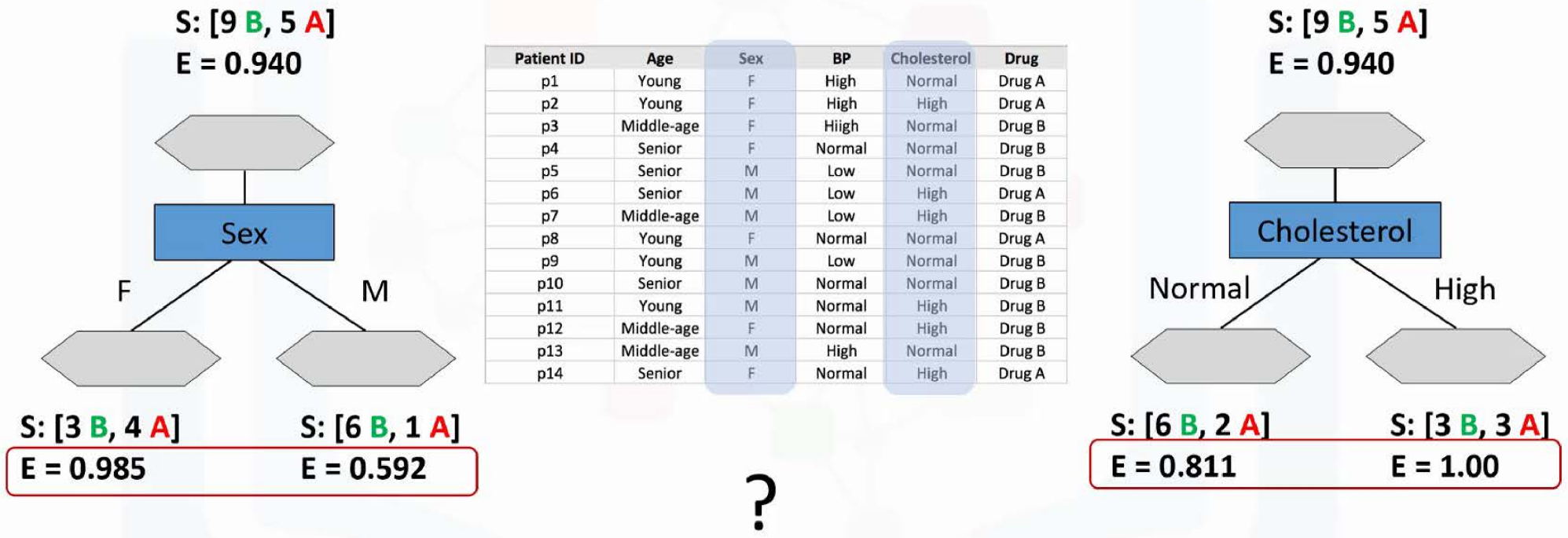


- The attribute with the highest information gain is chosen as the splitting attribute for node N. This attribute minimizes the information needed to classify the tuples in the resulting partitions

Bilgi Kazancı: Kesinlik seviyesini arttıran bilgi miktarı

Bölme için bilgi kazancını arttıran özellik seçilir

Which attribute is the best?



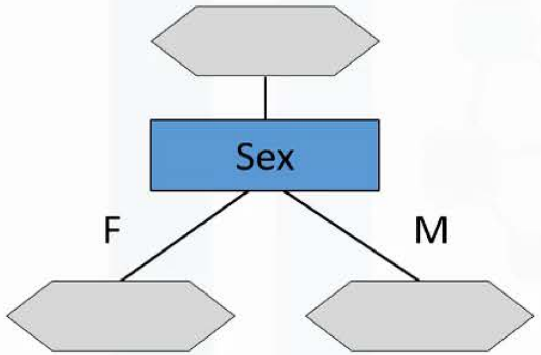
Patient ID	Age	Sex	BP	Cholesterol	Drug
p1	Young	F	High	Normal	Drug A
p2	Young	F	High	High	Drug A
p3	Middle-age	F	Hiigh	Normal	Drug B
p4	Senior	F	Normal	Normal	Drug B
p5	Senior	M	Low	Normal	Drug B
p6	Senior	M	Low	High	Drug A
p7	Middle-age	M	Low	High	Drug B
p8	Young	F	Normal	Normal	Drug A
p9	Young	M	Low	Normal	Drug B
p10	Senior	M	Normal	Normal	Drug B
p11	Young	M	Normal	High	Drug B
p12	Middle-age	F	Normal	High	Drug B
p13	Middle-age	M	High	Normal	Drug B
p14	Senior	F	Normal	High	Drug A

The tree with the higher Information Gain after splitting.

Bölmeden sonra Bilgi Kazancı Daha Yüksek olan Ağaç

Which attribute is the best?

S: [9 B, 5 A]
E = 0.940



S: [3 B, 4 A]
E = 0.985

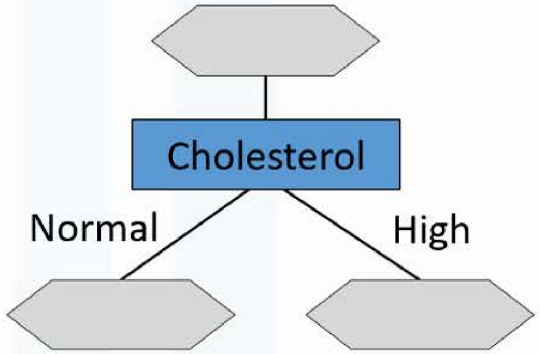
S: [6 B, 1 A]
E = 0.592

Gain (s, Sex)
= 0.940 – [(7/14)0.985 + (7/14)0.592]
=0.151

Patient ID	Age	Sex	BP	Cholesterol	Drug
p1	Young	F	High	Normal	Drug A
p2	Young	F	High	High	Drug A
p3	Middle-age	F	Hiigh	Normal	Drug B
p4	Senior	F	Normal	Normal	Drug B
p5	Senior	M	Low	Normal	Drug B
p6	Senior	M	Low	High	Drug A
p7	Middle-age	M	Low	High	Drug B
p8	Young	F	Normal	Normal	Drug A
p9	Young	M	Low	Normal	Drug B
p10	Senior	M	Normal	Normal	Drug B
p11	Young	M	Normal	High	Drug B
p12	Middle-age	F	Normal	High	Drug B
p13	Middle-age	M	High	Normal	Drug B
p14	Senior	F	Normal	High	Drug A

?

S: [9 B, 5 A]
E = 0.940



S: [6 B, 2 A]
E = 0.811

S: [3 B, 3 A]
E = 1.00

Gain (s, Cholesterol)
= 0.940 – [(8/14).811 + (6/14)1.0]
=0.048

Which attribute is the Best?

- Different criteria:
- ID3 splits attributes based on their ***entropy (Information Gain)***. Entropy is maximized when there is an equal chance of all values for the target attribute (i.e. the result is random). ID3 splits on attributes with the lowest entropy
- For continuous attribute: Partition the continuous value of attribute A into a discrete set of intervals
- Other criteria
  - Gain Ratio
  - Gini Index, etc.

Classification

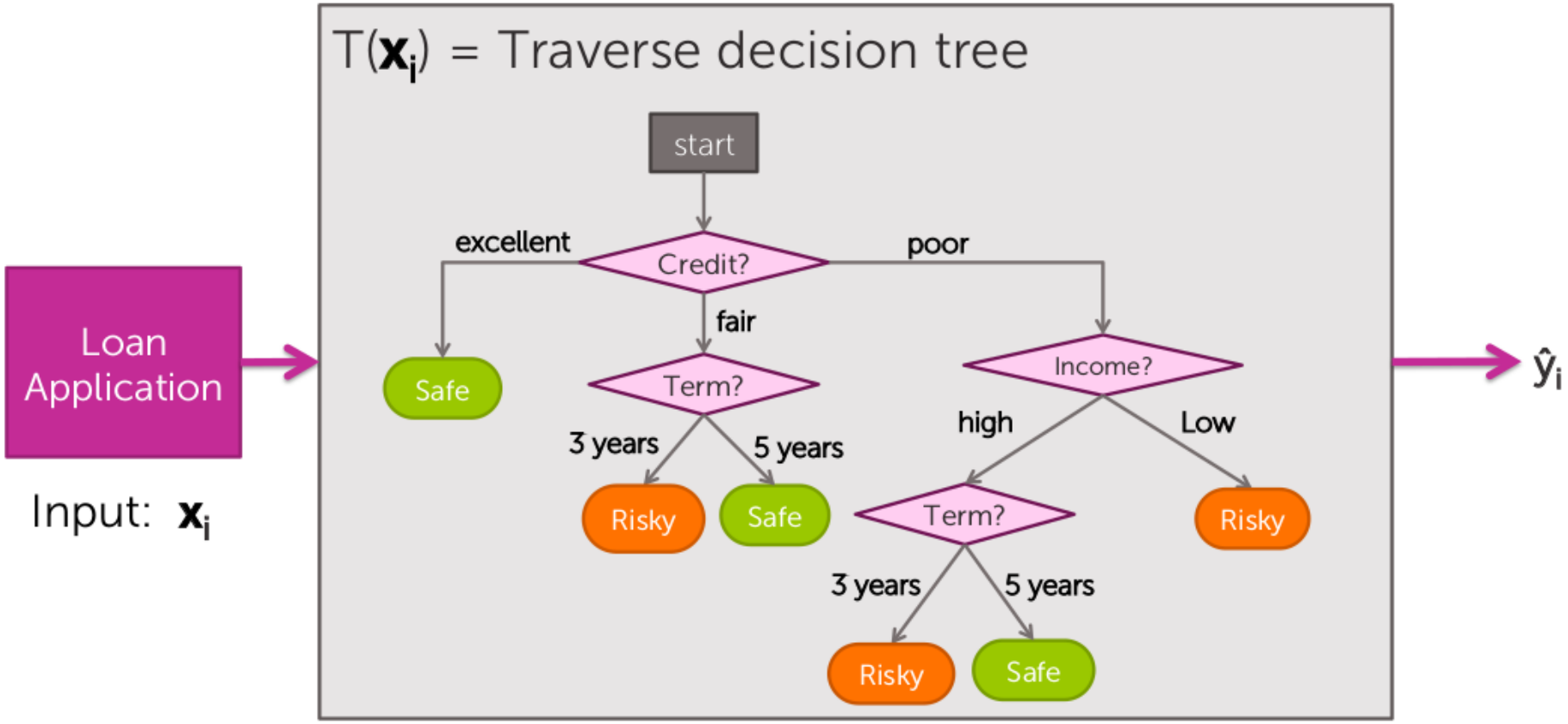


Image: Machine Learning: Classification by University of Washington

Classification

$\mathbf{x}_i$ = (Credit = poor, Income = high, Term = 5 years)

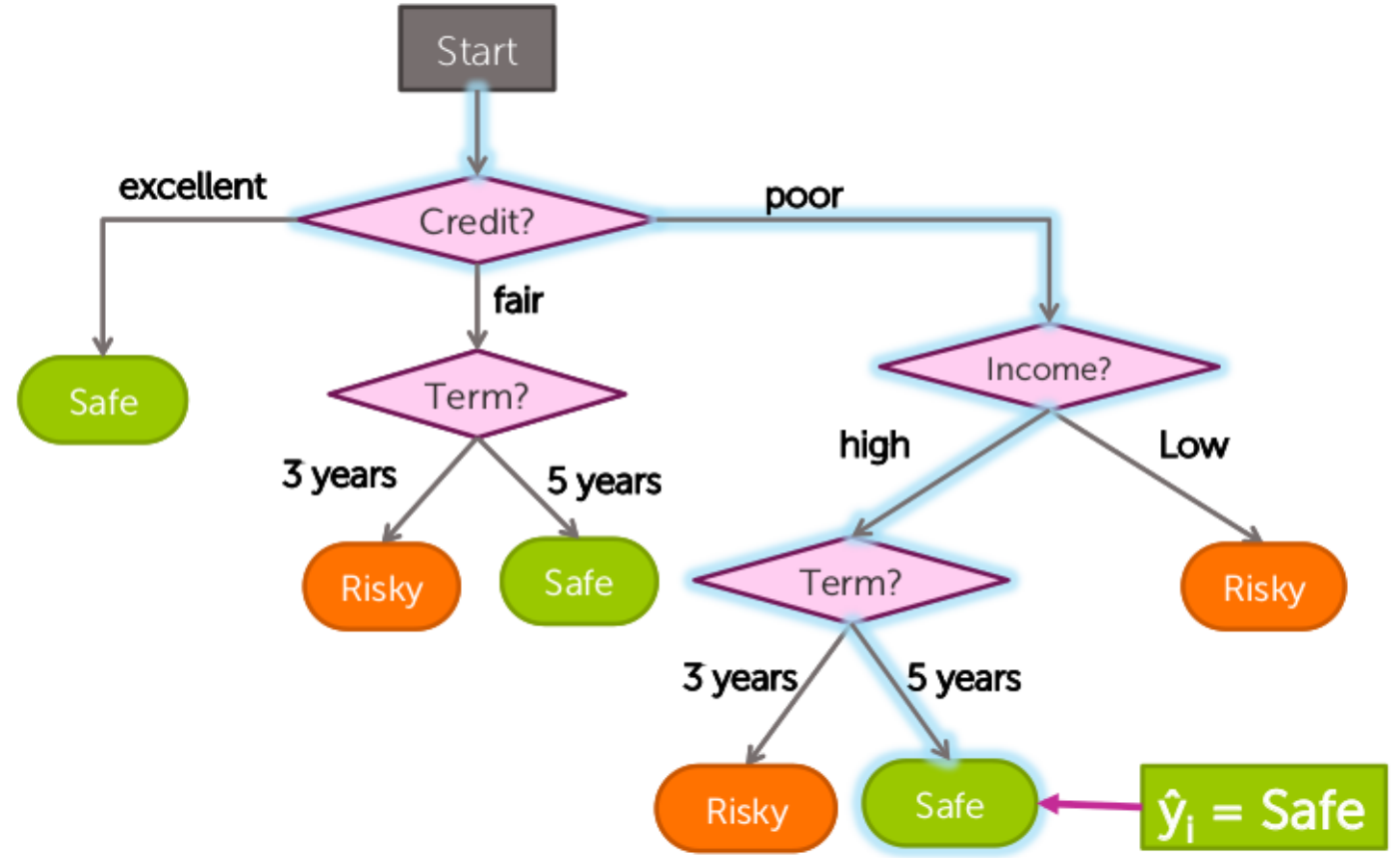


Image: Machine Learning: Classification by University of Washington

Tree Pruning - Preventing Overfitting

- When a decision tree is built, many of the branches will reflect anomalies in the training data due to noise or outliers. Tree pruning methods address this problem of overfitting the data.

- Pruned trees tend to be smaller and less complex and, thus, easier to comprehend. They are usually faster and better at correctly classifying independent test data (i.e., of previously unseen tuples) than unpruned trees.
- 2 Basic Methods
  - Pre-pruning
  - Post-pruning

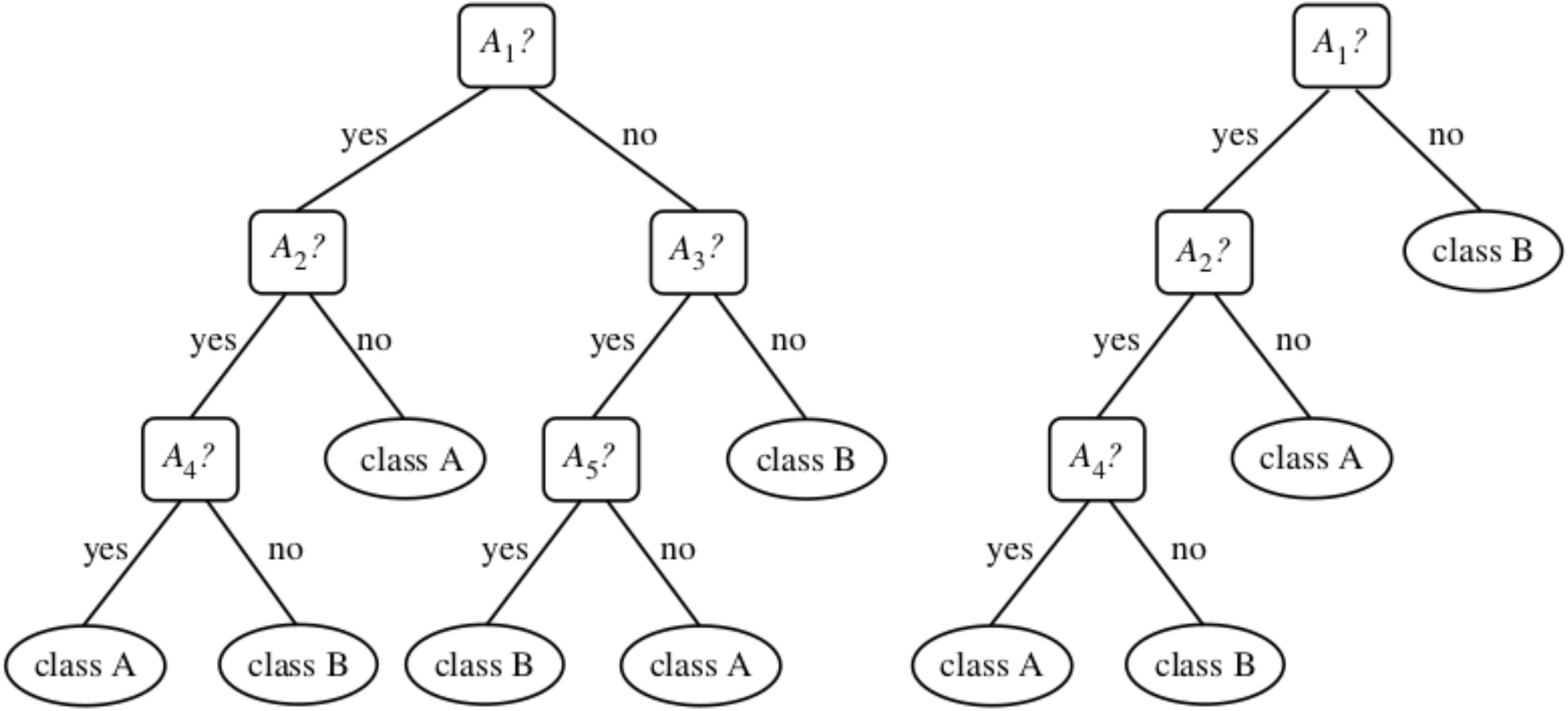
Tree Pruning - Prepruning

- a tree is “pruned” by halting its construction early (e.g., by deciding not to further split or partition the subset of training tuples at a given node).
- Upon halting, the node becomes a leaf.
- When constructing a tree, measures such as information gain, can be used to assess the goodness of a split. If partitioning the tuples at a node would result in a split that falls below a prespecified threshold, then further partitioning of the given subset is halted.

Tree Pruning - Postpruning

- Removes subtrees from a “fully grown” tree.
- A subtree at a given node is pruned by removing its branches and replacing it with a leaf. The leaf is labeled with the most frequent class among the subtree being replaced.
-
- For example, the subtree at node “A 3 ?” in the next slide has the most common class “class B.” In the pruned version of the tree, the subtree in question is pruned by replacing it with the leaf “class B.”

Tree Pruning



Summary and Critics for Decision Trees

- Efficiency becomes an issue of concern when these algorithms are applied to the mining of very large real-world databases.
- The pioneering decision tree algorithms that we have discussed so far have the restriction that the training tuples should reside in memory.
- Most often, the training data will not fit in memory!
- More recent decision tree algorithms that address the scalability issue for trees from very large training sets include SLIQ and SPRINT, both of which can handle categorical and continuous-valued attributes.

Decision Tree Application – Drug Prescription

Drug Prescription Problem

A list of patients and the drugs that they responded to

- Our objective is to build a classifier, to predict which drug is good for a new patient using decision tree
- We will use the data file “drug200.csv” which has 200 instances with the following attributes:
  - Age,Sex,BP,Cholesterol,Na_to_K,Drug

6. Ensemble Methods

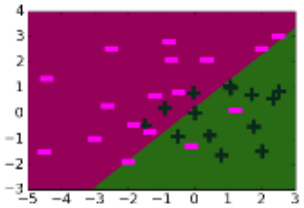
GSÜSEM

GALATASARAY ÜNİVERSİTESİ

Sürekli Eğitim Uygulama ve Araştırma Merkezi

Ensemble Methods – Increasing the Accuracy

Simple (weak) classifiers are good!



Logistic regression w. simple features

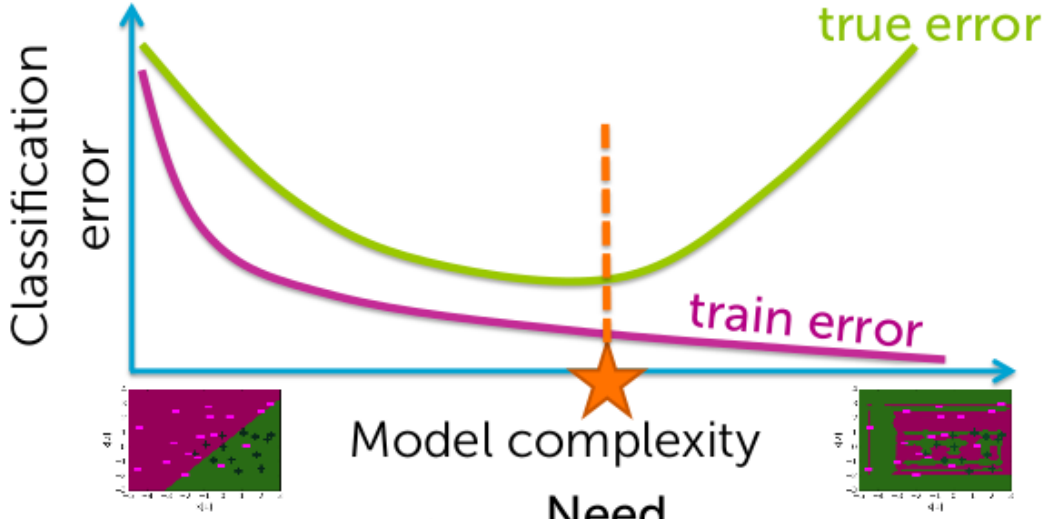
Shallow decision trees

Decision stumps

Low variance. Learning is fast!

But high bias…

Finding a classifier that's just right



Weak learner → Need stronger learner

Option 1: add more features or depth
Option 2: ?????

Image: Machine Learning: Classification by University of Washington

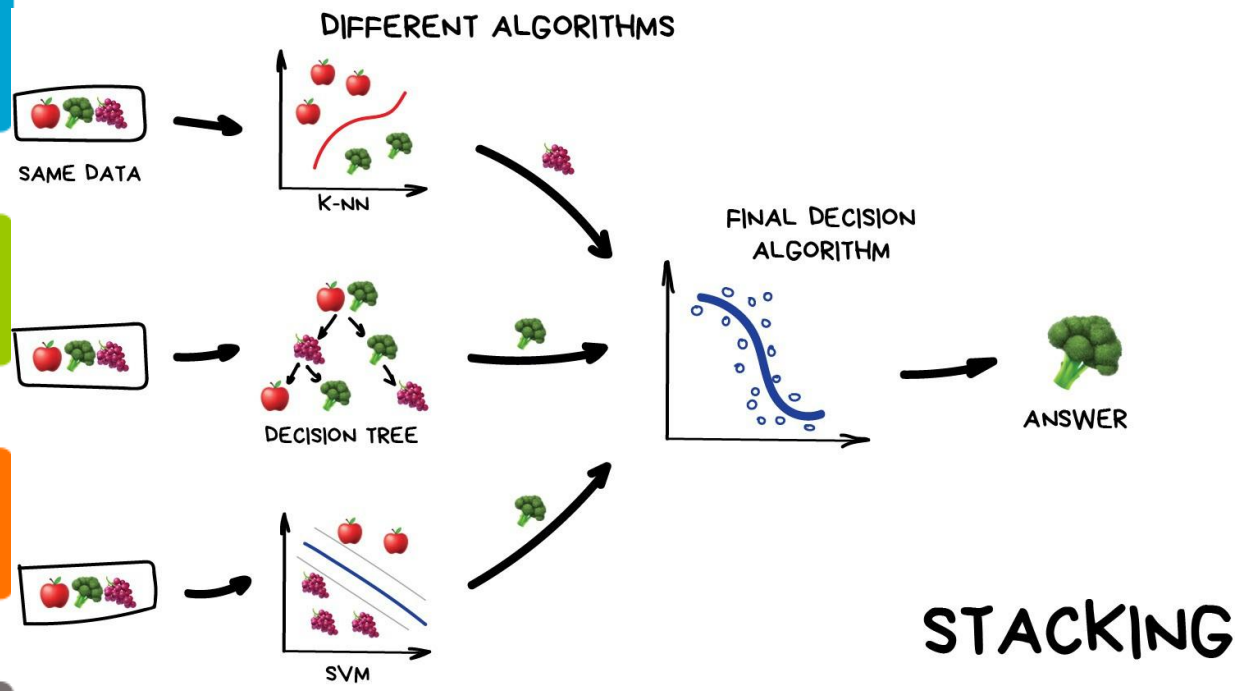
Boosting Question?

"Can a set of weak learners be combined to create a stronger learner?" *Kearns and Valiant (1988)*

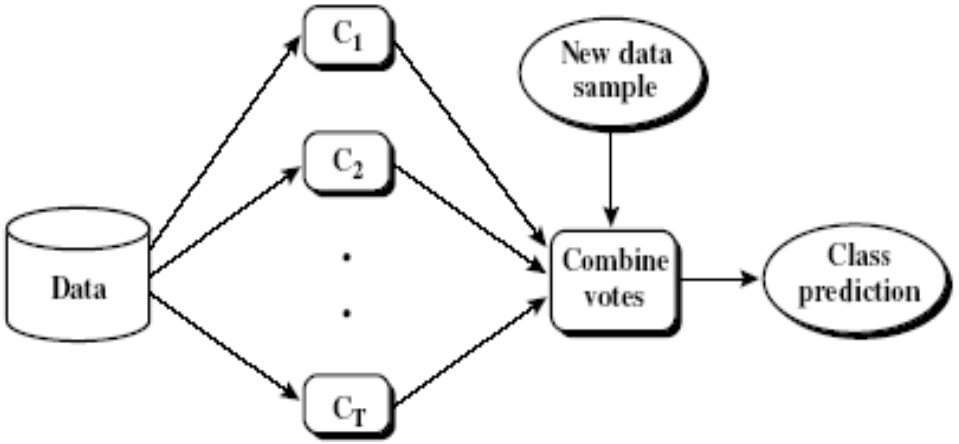
Yes! *Schapire (1990)*

Boosting

Amazing impact: • simple approach • widely used in industry • wins most Kaggle competitions



Boosting Question?



- Ensemble methods
  - Use a combination of models to increase accuracy
  - Combine a series of k learned models, M1, M2, …, Mk, with the aim of creating an improved model M*
- Popular ensemble methods
  - Bagging: averaging the prediction over a collection of classifiers
  - Boosting: weighted vote with a collection of classifiers
  - Ensemble: combining a set of heterogeneous classifiers

Bagging

Stands for “bootstrap aggregation”, each training set is a bootstrap sample

The bagged classifier often has significantly greater accuracy than a single one derived from D, the original training data.

+ more robust to the effects of noisy data

+ reduces the variance of the individual classifiers

Algorithm: Bagging. The bagging algorithm—create an ensemble of models (classifiers or predictors) for a learning scheme where each model gives an equally-weighted prediction.

Input:

- D, a set of d training tuples;
- k, the number of models in the ensemble;
- a learning scheme (e.g., decision tree algorithm, backpropagation, etc.)

Output: A composite model, $M*$.

Method:

(1) **for** $i = 1$ to k **do** // create k models:
(2) create bootstrap sample, D_i, by sampling D with replacement;
(3) use D_i to derive a model, M_i;
(4) **endfor**

To use the composite model on a tuple, X:

(1) **if** classification **then**
(2) let each of the k models classify $\boldsymbol{X}$ and return the majority vote;
(3) **if** prediction **then**
(4) let each of the k models predict a value for $\boldsymbol{X}$ and return the average predicted value;

Boosting: Focus learning on “hard” points

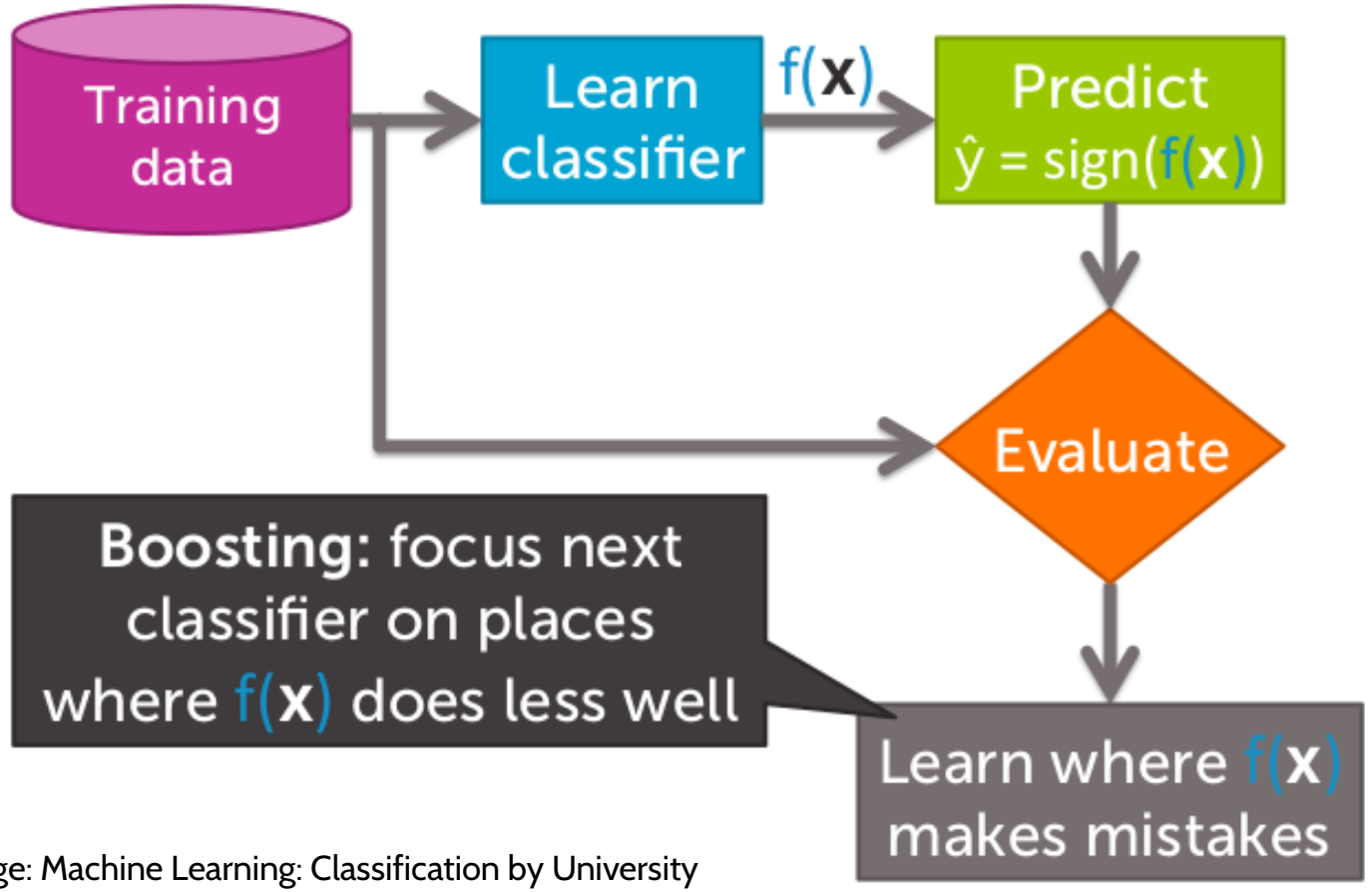


Image: Machine Learning: Classification by University of Washington

Weights are assigned to each training tuple (equal at start). A series of k classifiers is iteratively learned. After a classifier Mi is learned, the weights are updated to allow the subsequent classifier, M i+1 , to “pay more attention” to the training tuples that were misclassified by Mi.

The final boosted classifier, M*, combines the votes of each individual classifier, where the weight of each classifier’s vote is a function of its accuracy.

Example: AdaBoost

- Start same weight for all points: $\alpha_i = 1/N$

- For t = 1,…,T
  - Learn $f_t(\mathbf{x})$ with data weights α_i
  - Compute coefficient $\hat{w}_t$
  - Recompute weights α_i
  - Normalize weights α_i

- Final model predicts by:

$$\hat{y} = sign\left(\sum_{t=1}^{T} \hat{\mathbf{w}}_t f_t(\mathbf{x})\right)$$

$$\hat{\mathbf{w}}_t = \frac{1}{2}\ln\left(\frac{1 - weighted\_error(f_t)}{weighted\_error(f_t)}\right)$$

$$\alpha_i \leftarrow \begin{cases} \alpha_i e^{-\hat{w}_t}, & \text{if } f_t(\mathbf{x}_i) = y_i \\ \alpha_i e^{\hat{w}_t}, & \text{if } f_t(\mathbf{x}_i) \neq y_i \end{cases}$$

$$\alpha_i \leftarrow \frac{\alpha_i}{\sum_{j=1}^{N} \alpha_j}$$

Random Forest (Breiman 2001)

- Random Forest:
  - Each classifier in the ensemble is a decision tree classifier and is generated using a random selection of attributes at each node to determine the split
  - During classification, each tree votes and the most popular class is returned
- Two Methods to construct Random Forest:
  - Forest-RI (random input selection): Randomly select, at each node, F attributes as candidates for the split at the node. The CART methodology is used to grow the trees to maximum size
  - Forest-RC (random linear combinations): Creates new attributes (or features) that are a linear combination of the existing attributes (reduces the correlation between individual classifiers)
- Comparable in accuracy to Adaboost, but more robust to errors and outliers
- Insensitive to the number of attributes selected for consideration at each split, and faster than bagging or boosting

7. Support Vector Machine Classification Algorithm

GSÜSEM

GALATASARAY ÜNİVERSİTESİ
Sürekli Eğitim Uygulama ve Araştırma Merkezi

Support Vector Machines

- A Support Vector Machine is a supervised algorithm that can classify cases by finding a separator.

  1)Mapping data to a **high dimensional** feature space so that points can be categorized, even when the data are not otherwise linearly separable.

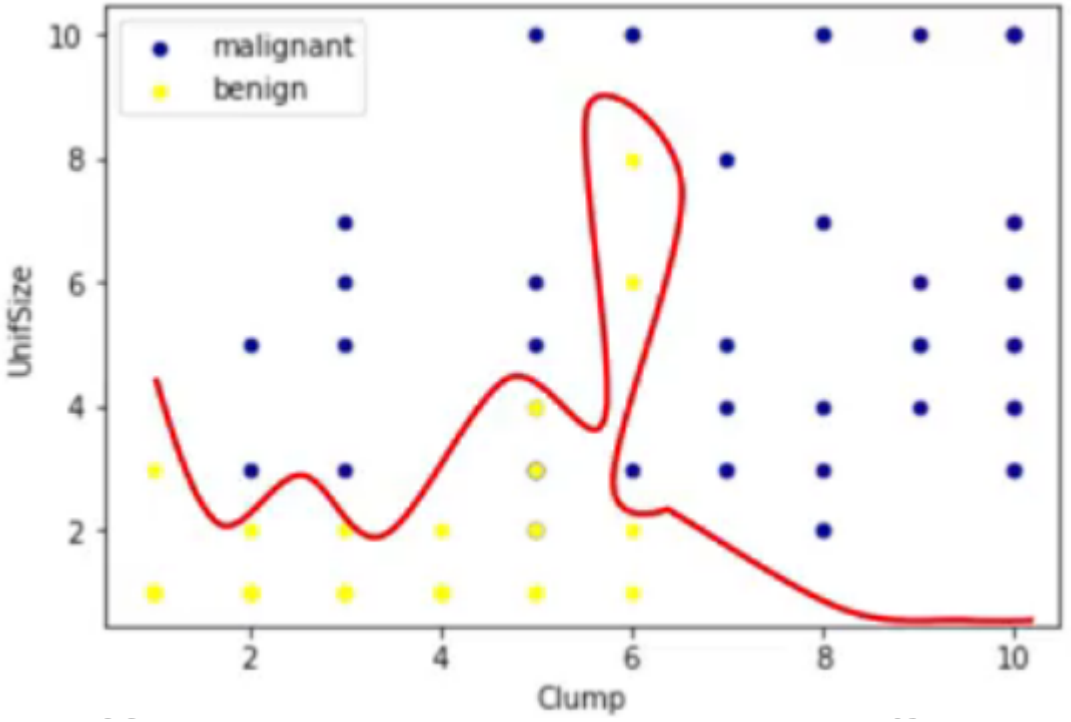
  2) Finding a separator

- In a nutshell:
  - It uses a nonlinear mapping to transform the original training data into a higher dimension. Within this new dimension,
  - it searches for the linear optimal separating hyperplane (that is, a “decision boundary” separating the tuples of one class from another).

Support Vector Machine - Example

A dataset containing characteristics of thousands of human cell samples extracted from patients who were believed to be at risk of developing cancer, being benign and malignant.
We can use the values of these cell characteristics to give an early indication

Clump	UnifSize	UnifShape	MargAdh	SingEpiSize	BareNuc	BlandChrom	NormNucl	Mit	Class
5	1	1	1	2	1	3	1	1	benign
5	4	4	5	7	10	3	2	1	benign
3	1	1	1	2	2	3	1	1	malignant
6	8	8	1	3	4	3	7	1	benign
4	1	1	3	2	1	3	1	1	benign
8	10	10	8	7	10		7	1	malignant
1	1	1	1	2	10	3	1	1	benign
2	1	2	H	2	1	3	1	1	benign
2	1	1	1	2	1	1	1	5	benign
4	2	1	1	2	1	2	1	1	benign

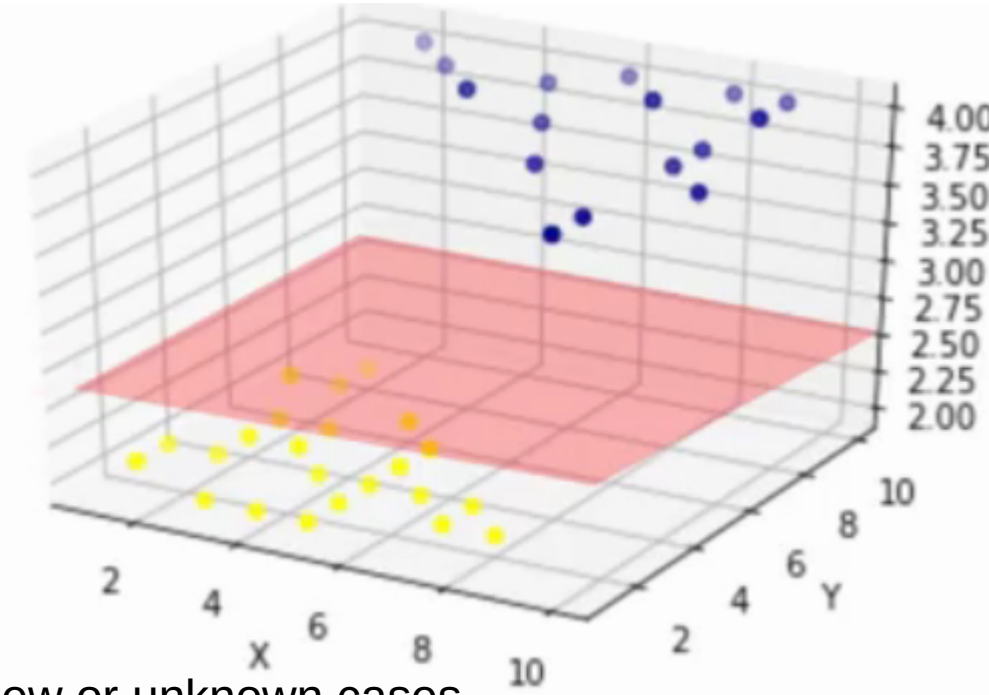


The two categories can be separated with a curve but not a line, a linearly non separable data set

Support Vector Machine - Example

- A Support Vector Machine is a supervised algorithm that can classify cases by finding a separator.
  1)Mapping data to a **high dimensional** feature space so that points can be categorized, even when the data are not otherwise linearly separable.
  2) Finding a separator

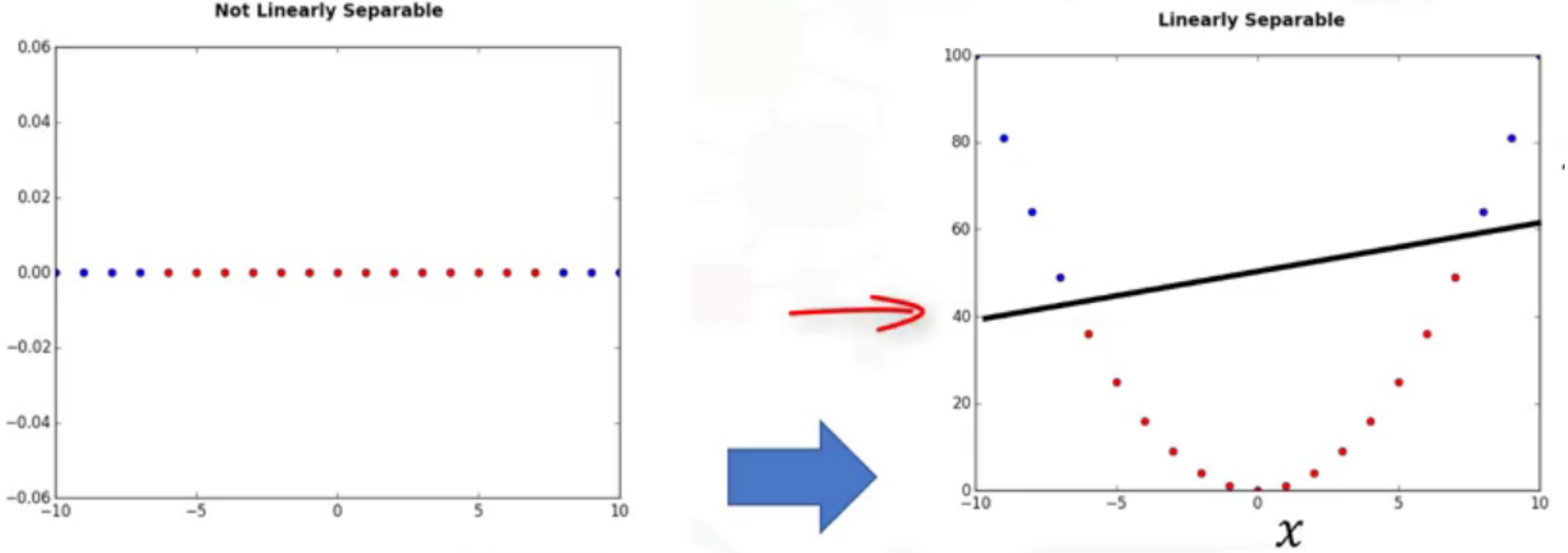
Clump	UnifSize	UnifShape	MargAdh	SingEpiSize	BareNuc	BlandChrom	NormNucl	Mit	Class
5	1	1	1	2	1	3	1	1	benign
5	4	4	5	7	10	3	2	1	benign
3	1	1	1	2	2	3	1	1	malignant
6	8	8	1	3	4	3	7	1	benign
4	1	1	3	2	1	3	1	1	benign
8	10	10	8	7	10		7	1	malignant
1	1	1	1	2	10	3	1	1	benign
2	1	2	H	2	1	3	1	1	benign
2	1	1	1	2	1	1	1	5	benign
4	2	1	1	2	1	2	1	1	benign



the separator is shown as a plane. This plane can be used to classify new or unknown cases. Therefore, the SVM algorithm outputs an optimal hyperplane that categorizes new examples

Data Transformation

Kernelling: Mapping data into a higher-dimensional space.
Kernel Function: The mathematical function used for the transformation, can be:
Linear, polynomial, Radial Basis Function, and sigmoid.



$$\phi(x) = [x, x^2]$$

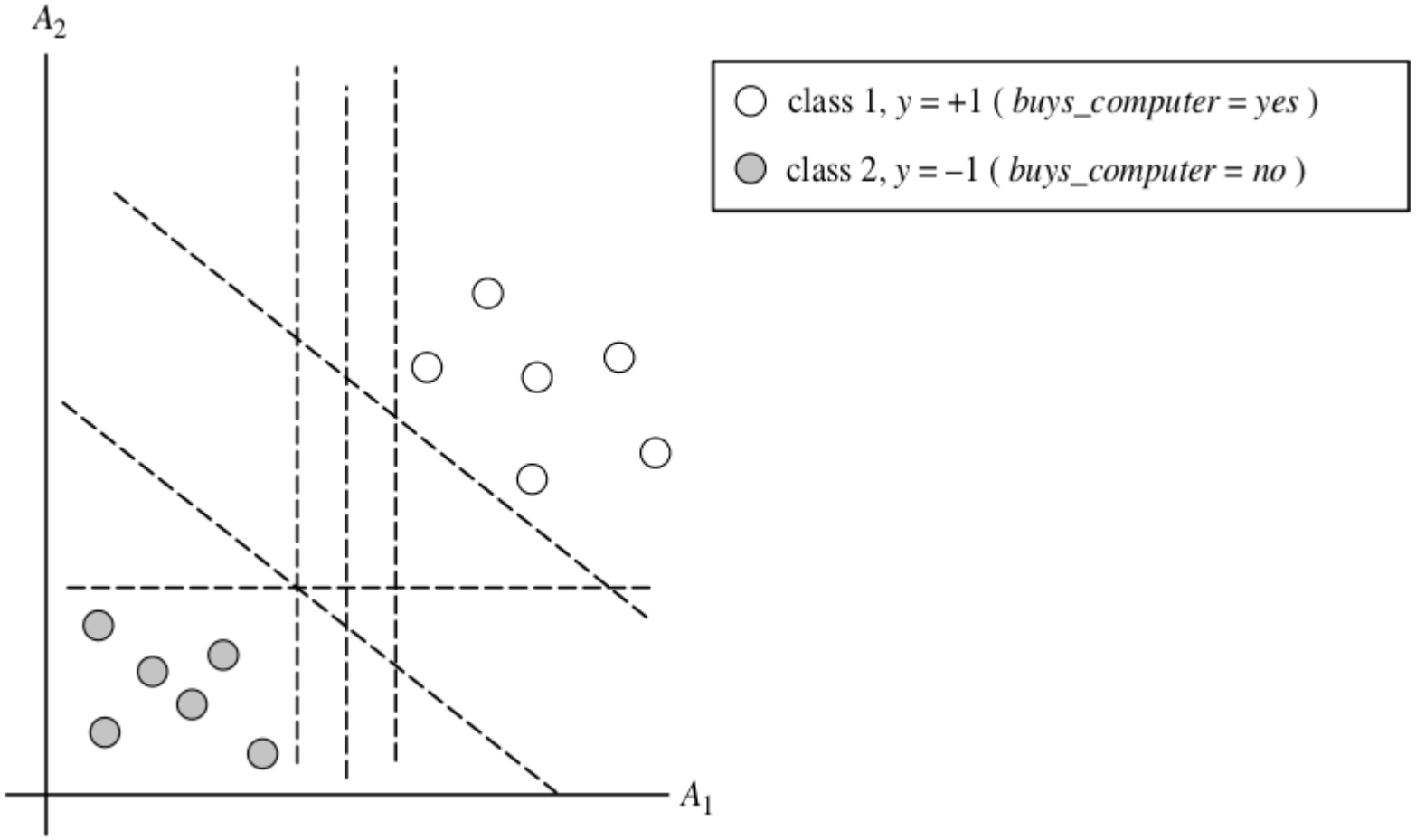
Example Kernel Functions

Polynomial kernel of degree h : $K(X_i, X_j) = (X_i \cdot X_j + 1)^h$

Gaussian radial basis function kernel : $K(X_i, X_j) = e^{-\|X_i - X_j\|^2 / 2\sigma^2}$

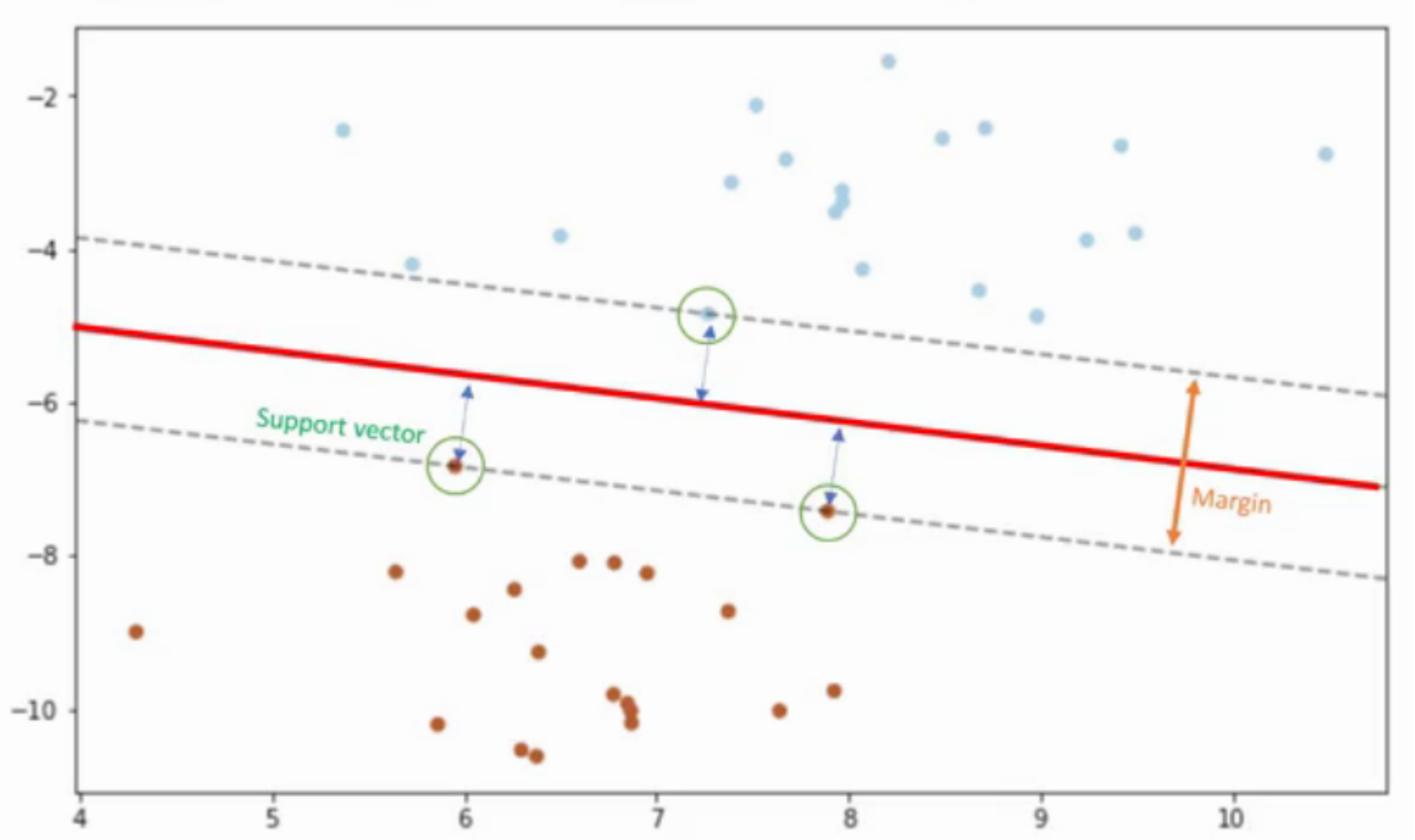
Sigmoid kernel : $K(X_i, X_j) = \tanh(\kappa X_i \cdot X_j - \delta)$

Finding the Hyperplane



There are an infinite number of (possible) separating hyperplanes or "decision boundaries."
Which one is best?

How to Find the Hyperplane?



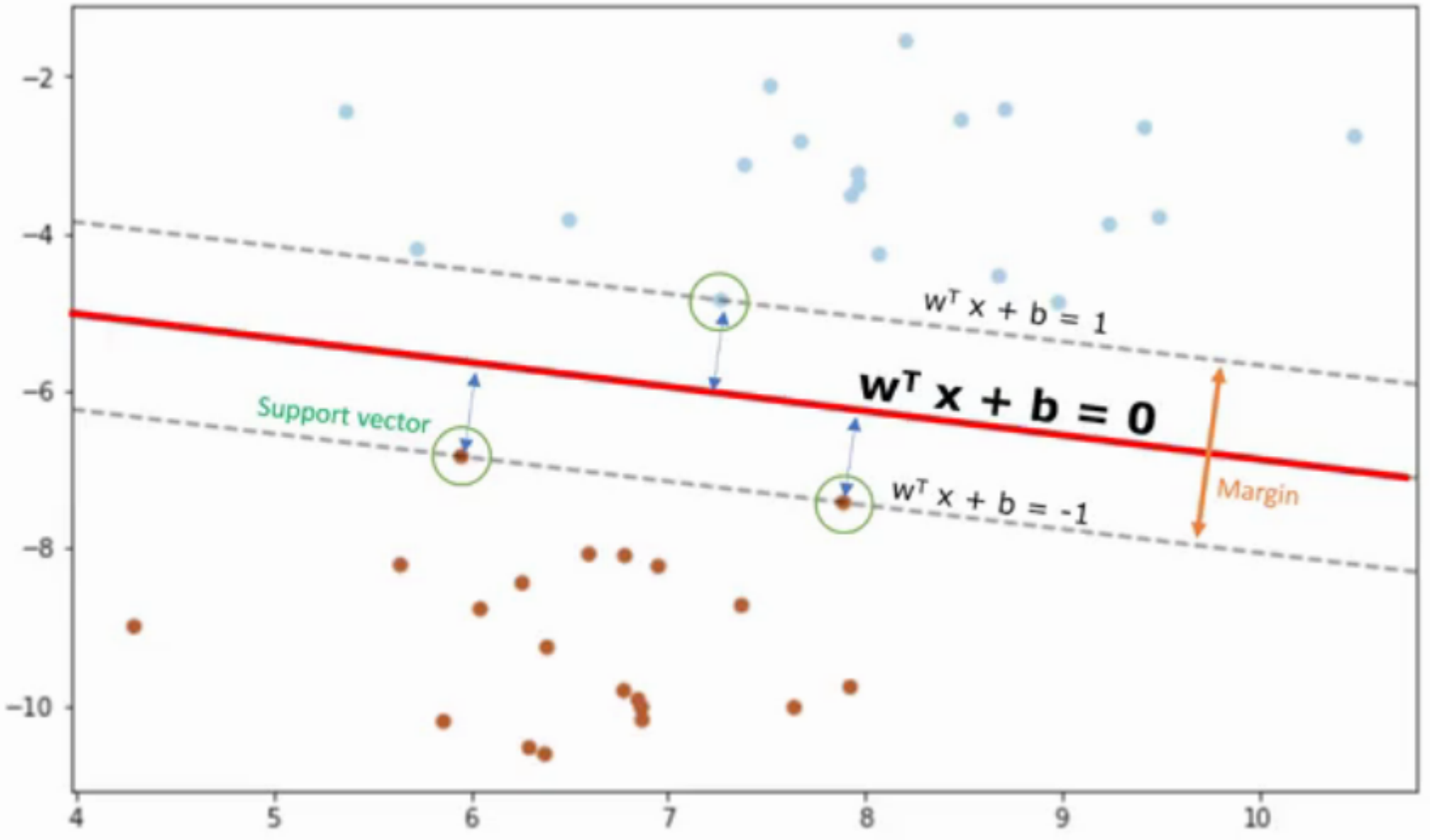
Choose the best hyperplane that represents the largest separation or **margin** between the two classes
Or choose a hyperplane with as big a margin as possible.

Examples closest to the hyperplane are **support vectors.**

How to Find the Hyperplane?

Find $\mathbf{w}$ and b such that
$\Phi(\mathbf{w}) = \frac{1}{2}\,\mathbf{w}^T\mathbf{w}$ is minimized;
and for all $\{(\mathbf{x_i}, y_i)\}$: $y_i\,(\mathbf{w}^T\mathbf{x_i} + b) \geq 1$

- hyperplane is learned from training data using an optimization procedure that maximizes the margin
- hyperplane and boundary decision lines, have their own equations (Gradient Descent can be used)
- The output of the algorithm is the values w and b for the line.
- Classifications are done using this estimated line.
- If the equation returns a value greater than 0,
- then the point belongs to the first class which is above the line, and vice-versa.



Multi-Class Problems?

- SVM classifiers can be combined for the multiclass case.
- A simple and effective approach, given m classes, trains m classifiers, one for each class (where classifier j learns to return a positive value for class j and a negative value for the rest).
- A test tuple is assigned the class corresponding to the largest positive distance.

Multiclass Classification

- Classification involving more than two classes (i.e., > 2 Classes)
- **Method 1. One-vs.-all (OVA):** Learn a classifier one at a time
  - Given m classes, train m classifiers: one for each class
  - Classifier j: treat tuples in class j as positive & all others as negative
  - To classify a tuple X, the set of classifiers vote as an ensemble
- **Method 2. All-vs.-all (AVA):** Learn a classifier for each pair of classes
  - Given m classes, construct m(m-1)/2 binary classifiers
  - A classifier is trained using tuples of the two classes
  - To classify a tuple X, each classifier votes. X is assigned to the class with maximal vote
- **Comparison**
  - All-vs.-all tends to be superior to one-vs.-all
  - Problem: Binary classifier is sensitive to errors, and errors affect vote count

Summary and Critics for SVM

- Advantages
  - Accurate in high-dimensional spaces.
  - Use a subset of training points in the decision function called, support vectors, so it's also memory efficient.

- Disadvantages
  - prone for over-fitting if the number of features is much greater than the number of samples.
  - do not directly provide probability estimates,
  - SVMs are not very efficient computationally if your dataset is very big

SVM Example – Cancer Prediction

- The example is based on a dataset that is publicly available from the UCI Machine Learning Repository (Asuncion and Newman, 2007)[http://mlearn.ics.uci.edu/MLRepository.html]. The dataset consists of several hundred human cell sample records, each of which contains the values of a set of cell characteristics.
- The Class field contains the diagnosis, as confirmed by separate medical procedures, as to whether the samples are benign (value = 2) or malignant (value = 4).
- Our objective is to build a classifier, to predict the class of unknown cases using SVM
- We will use the data file “cell_samples.csv”

Semi-Supervised Classification

- Semi-supervised: Uses labeled and unlabeled data to build a classifier
- Self-training:
  - Build a classifier using the labeled data
  - Use it to label the unlabeled data, and those with the most confident label prediction are added to the set of labeled data
  - Repeat the above process
  - Adv: easy to understand; disadv: may reinforce errors
- Co-training: Use two or more classifiers to teach each other
  - Each learner uses a mutually independent set of features of each tuple to train a good classifier, say f_1
  - Then f_1 and f_2 are used to predict the class label for unlabeled data X
  - Teach each other: The tuple having the most confident prediction from f_1 is added to the set of labeled data for f_2, & vice versa
- Other methods, e.g., joint probability distribution of features and labels

Active Learning

- Class labels are expensive to obtain
- Active learner: query human (oracle) for labels
- Pool-based approach: Uses a pool of unlabeled data
  - L: a small subset of D is labeled, U: a pool of unlabeled data in D
  - Use a query function to carefully select one or more tuples from U and request labels from an oracle (a human annotator)
  - The newly labeled samples are added to L, and learn a model
  - Goal: Achieve high accuracy using as few labeled data as possible
- Evaluated using learning curves: Accuracy as a function of the number of instances queried (# of tuples to be queried should be small)
- Research issue: How to choose the data tuples to be queried?
  - Uncertainty sampling: choose the least certain ones
  - Reduce version space, the subset of hypotheses consistent w. the training data
  - Reduce expected entropy over U: Find the greatest reduction in the total number of incorrect predictions

Summary (I)

- 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.
- 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.

Summary (II)

- 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

Data Types (to clarify)