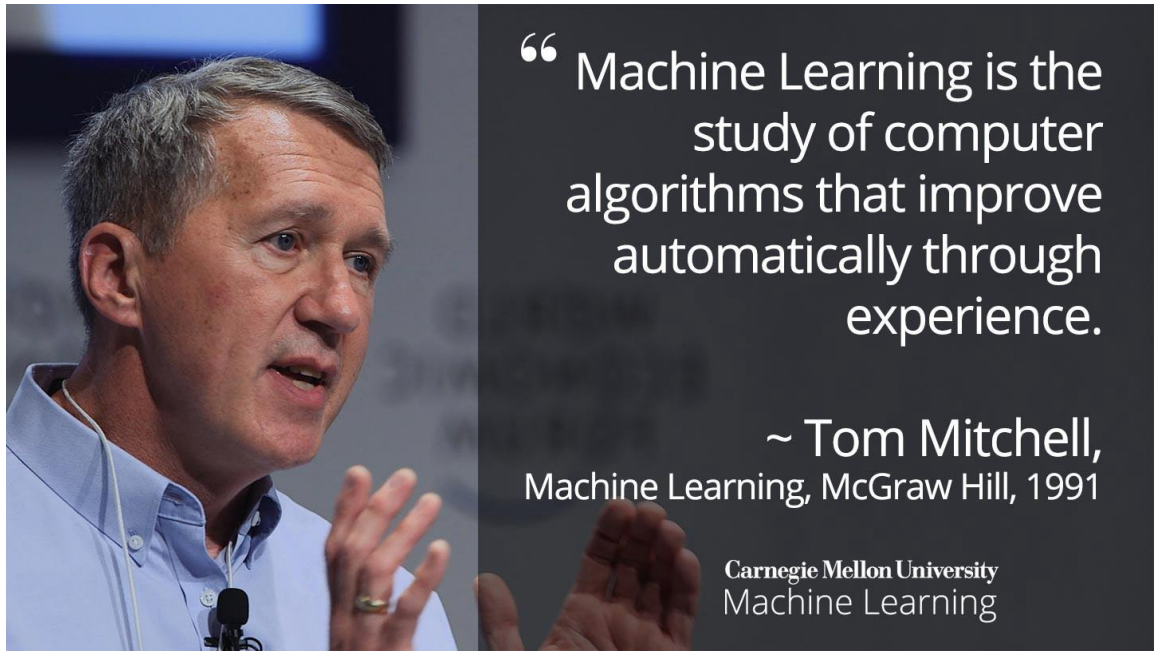
Mašinsko učenje

OSNOVE


“ Machine Learning is the study of computer algorithms that improve automatically through experience.
~ Tom Mitchell,
Machine Learning, McGraw Hill, 1991
Carnegie Mellon University
Machine Learning

Uvod

- Isprogramirati računare tako da uče iz iskustva
  - Google pretraga – rangiranje veb stranica
  - Spam filteri
  - Facebook – prepoznavanje lica na slikama
  - Amazon – preporuka proizvoda


Machine Learning Algorithms
Bayesian
Naive Bayes
Averaged One-Dependence Estimators (AODE)
Bayesian Belief Network (BBN)
Gaussian Naive Bayes
Multinomial Naive Bayes
Bayesian Network (BN)
Decision Tree
Classification and Regression Tree (CART)
Iterative Dichotomiser 3 (ID3)
C4.5
C5.0
Chi-squared Automatic Interaction Detection (CHAID)
Decision Stump
Conditional Decision Trees
M5
Dimensionality Reduction
Principal Component Analysis (PCA)
Partial Least Squares Regression (PLSR
Sammon Mapping
Multidimensional Scaling (MDS)
Projection Pursuit
Principal Component Regression (PCR)
Partial Least Squares Discriminant Analysis
Mixture Discriminant Analysis (MDA)
Quadratic Discriminant Analysis (QDA)
Regularized Discriminant Analysis (RDA)
Flexible Discriminant Analysis (FDA)
Linear Discriminant Analysis (LDA)
Instance Based
k-Nearest Neighbour (kNN)
Learning Vector Quantization (LVQ)
Self-Organizing Map (SOM)
Locally Weighted Learning (LWL)
Clustering
k-Means
k-Medians
Expectation Maximization
Hierarchical Clustering
Deep Learning
Deep Boltzmann Machine (DBM)
Deep Belief Networks (DBN)
Convolutional Neural Network (CNN)
Stacked Auto-Encoders
Ensemble
Random Forest
Gradient Boosting Machines (GBM)
Boosting
Bootstrapped Aggregation (Bagging)
AdaBoost
Stacked Generalization (Blending)
Gradient Boosted Regression Trees (GBRT)
Neural Networks
Radial Basis Function Network (RBFN)
Perceptron
Back-Propagation
Hopfield Network
Regularization
Ridge Regression
Least Absolute Shrinkage and Selection Operator (LASSO)
Elastic Net
Least Angle Regression (LARS)
Rule System
Cubist
One Rule (OneR)
Zero Rule (ZeroR)
Repeated Incremental Pruning to Produce Error Reduction (RIPPER)
Regression
Linear Regression
Ordinary Least Squares Regression (OLSR)
Stepwise Regression
Multivariate Adaptive Regression Splines (MARS)
Locally Estimated Scatterplot Smoothing (LOESS)
Logistic Regression

Mašinsko učenje

- Jedna od oblasti veštačke inteligencije.
- VI programi tipično rešavaju samo jedan problem (Deep Blue može samo da igra šah na šampionskom nivou).
- Mašinsko učenje omogućava pisanje programa koji rešavaju više problema bez menjanja njihovog koda.
- Primer je AlphaGO koji može da igra GO, ali može da nauči da igra Atari igrice.

Mašinsko učenje

Izučavanje algoritama koji

UČE NA OSNOVU PRIMERA I/ILI ISKUSTVA

umesto da koriste hard-kodirana pravila

Mašinsko učenje

Izučavanje algoritama koji

UČE NA OSNOVU PRIMERA I/ILI ISKUSTVA

umesto da koriste hard-kodirana pravila

Mašinsko učenje iz ugla agenta:
Agent se obučava ako posle posmatranja sveta unapređuje svoje performanse na narednim zadacima.

Mašinsko učenje

Definicija učenja:

Računarski program uči iz iskustva ako se njegove performanse pri obavljanju nekog zadatka poboljšavaju sa povećanjem iskustva.

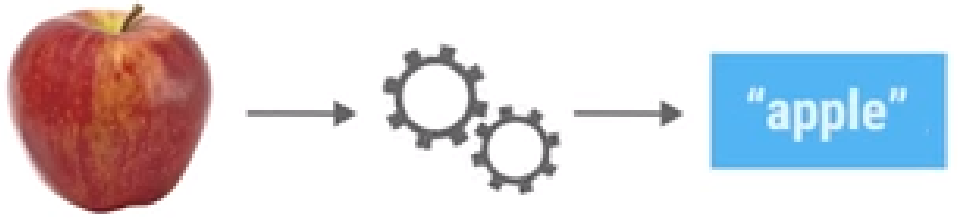
Program koji uči da igra Mice može poboljšavato svoje performanse (sposobnost da pobedi) u igranju igre Mice, kroz iskustvo stečeno igranjem ove igre protiv samog sebe.

Mašinsko učenje

- Računari uče iz podataka
  - Bez eksplicitnog programiranja
  - Uočavaju implicitne paterne u podacima
- Računari su sposobni da odgovore na pitanja poput:
  - “Koja je tržišna vrednost ove kuće?”,
  - “Hoće li se ovoj osobi dopasti taj film?”,
  - “Da li je ovo kancer?”,
  - “Ko je na ovoj slici?”

Primer mašinskog učenja

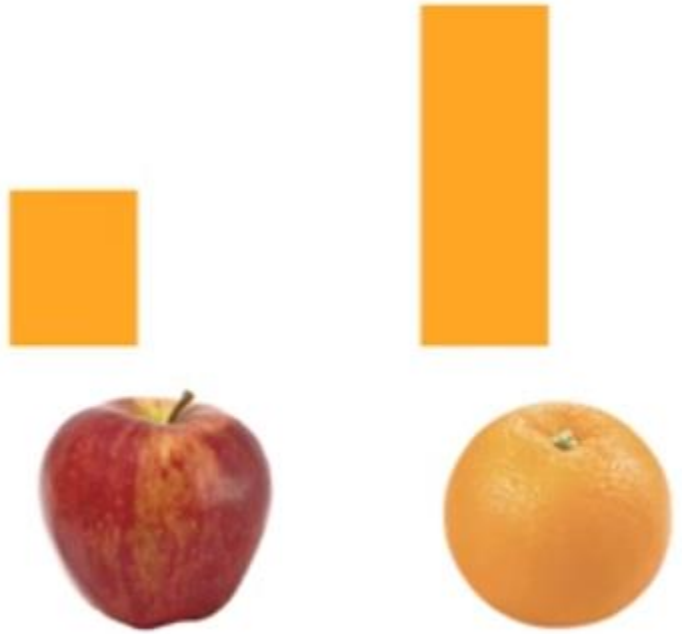
- Napraviti program koji dobija sliku kao ulaz, vrši nekakvu analizu i zaključuje o kojoj vrsti voća je reč.



Primer mašinskog učenja

- Napraviti program koji dobija sliku kao ulaz, vrši nekakvu analizu i zaključuje o kojoj vrsti voća je reč.

- Analizirati piksele slike i napisati gomilu pravila na osnovu kojih će se vršiti zaključivanje?



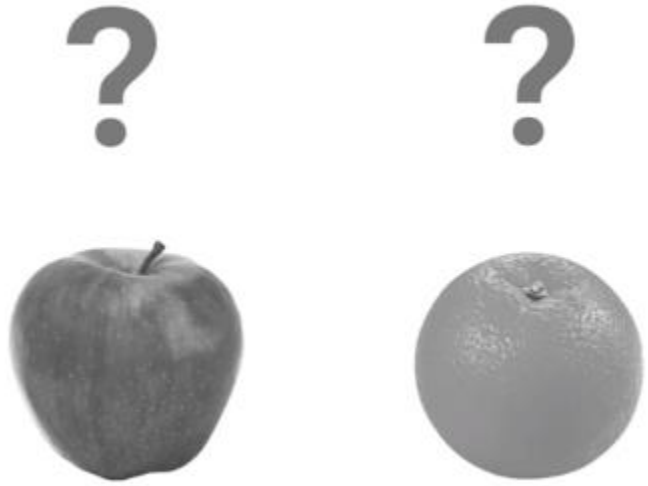
Primer mašinskog učenja

- Napraviti program koji dobija sliku kao ulaz, vrši nekakvu analizu i zaključuje o kojoj vrsti voća je reč.

- Analizirati piksele slike i napisati gomilu pravila na osnovu kojih će se vršiti zaključivanje?

Primer mašinskog učenja

- Napraviti program koji dobija sliku kao ulaz, vrši nekakvu analizu i zaključuje o kojoj vrsti voća je reč.
- Analizirati piksele slike i napisati gomilu pravila na osnovu kojih će se vršiti zaključivanje?
- Može da funkcioniše, ali samo za jednostavne primere.
- Za svaki skup pravila, koliko god veliki, može se pronaći primer za koji pravila neće raditi.



Kako rešiti prethodni problem?

- Potreban nam je algoritam koji će sam pronaći (uočiti) „pravila“, tako da mi ne moramo da ih pišemo.

- Primenom tih „pravila“ računar će za svaki primerak voćke moći da odluči da li se radi o jabuci ili pomorandži.

Kako rešiti prethodni problem?

- Softverski sistem koji će moći da odgovara na pitanje da li je voće jabuka ili pomorandža naziva se **model**, a proces njegovog kreiranja **treniranje**.

- Cilj treniranja je dobijanje tačnog modela koji nam u većini slučajeva daje tačan odgovor.

- Da bi se model trenirao potrebni su podaci.

Trening podaci

Trening podaci (trening skup) su poznati podaci iz kojih algoritam može da uči.

- Na primer, masa i tekstura raznih primeraka voća.

Trening podaci se sastoje iz trening primera: jedan par ulaz-izlaz.

Weight	Texture	Label
150g	Bumpy	Orange
170g	Bumpy	Orange
140g	Smooth	Apple
130g	Smooth	Apple
…	…	…

Trening podaci

Ulaz se sastoji iz više svojstava ili atributa (Features).

- Masa i tekstura voća su svojstva.

- Dobro odabrani atributi, omogućavaju laku klasifikaciju.

- Klasifikator će biti onoliko dobar, koliko su dobro odabrani atributi na osnovu kojih se klasifikacija vrši.

Features

Weight	Texture	Label
150g	Bumpy	Orange
170g	Bumpy	Orange
140g	Smooth	Apple
130g	Smooth	Apple
…	…	…

Trening podaci

- Svaki red u našoj tabeli trening podataka je jedan primer za učenje (Example)

- Jedan primer opisuje jednu voćku.

Examples

Weight	Texture	Label
150g	Bumpy	Orange
170g	Bumpy	Orange
140g	Smooth	Apple
130g	Smooth	Apple
…	…	…

Trening podaci

- Poslednja kolona se naziva Label
- Ona identifikuje vrstu voća koje ima features nabrojane u odgovarajućem redu.

Weight	Texture	Label
150g	Bumpy	Orange
170g	Bumpy	Orange
140g	Smooth	Apple
130g	Smooth	Apple
…	…	…

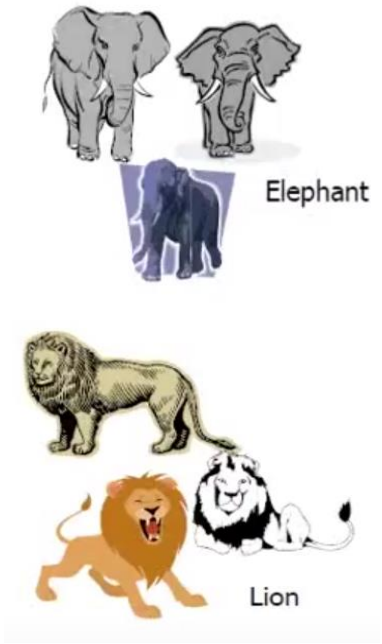
Trening podaci

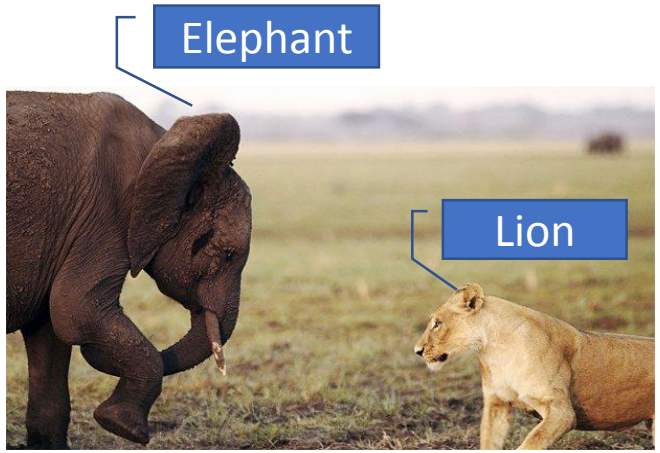
Što više trening podataka imamo, to ćemo biti u mogućnosti da napravimo bolji klasifikator.

Weight	Texture	Label
150g	Bumpy	Orange
170g	Bumpy	Orange
140g	Smooth	Apple
130g	Smooth	Apple
…	…	…

Supervised learning – Nadgledano obučavanje

- Dataset with labels
- Learning from existing data
- Predicting on new, unseen data

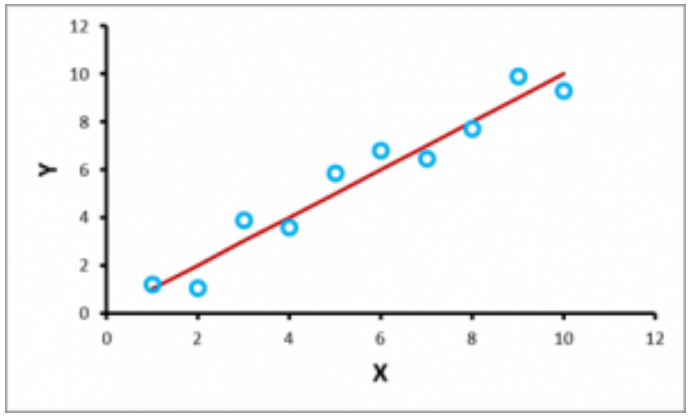


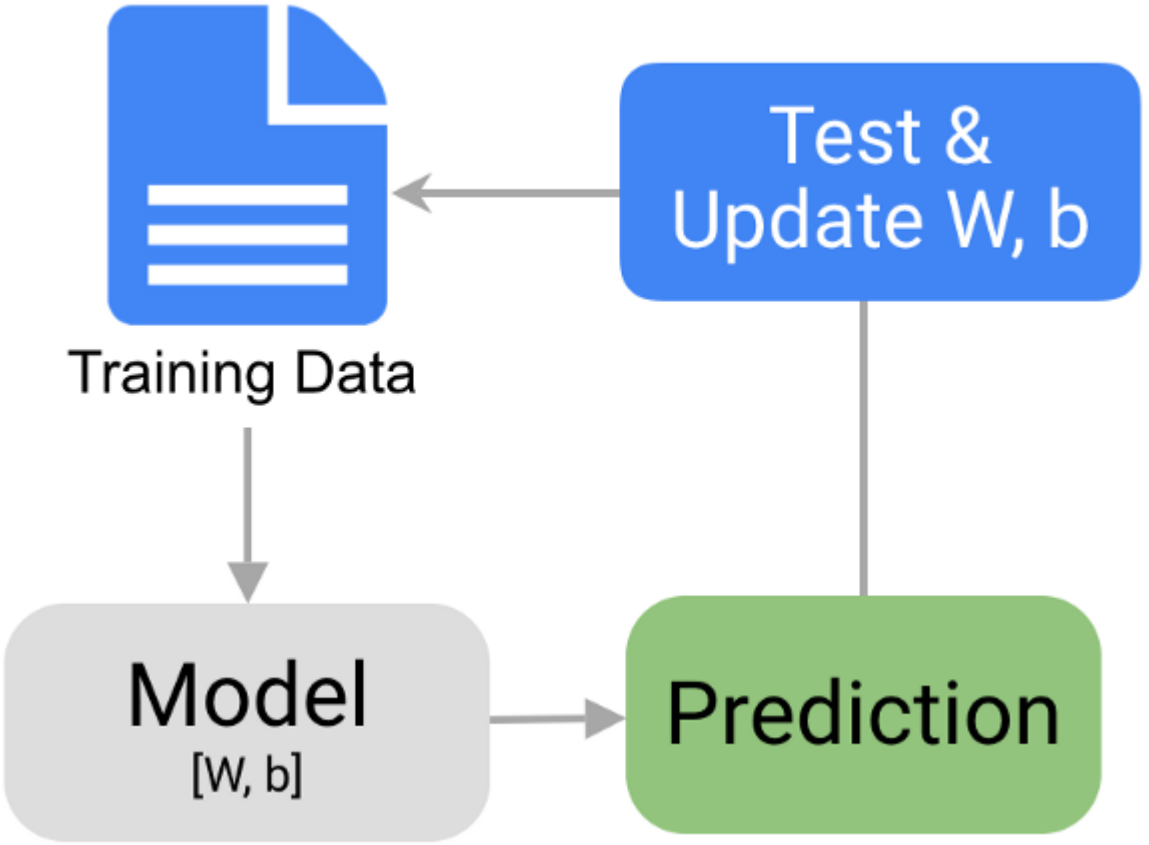


Nadgledano obučavanje - primer

- Tražimo funkciju $f: X \rightarrow Y$ koja preslikava **ulaz** X u **izlaz** Y.

- $y = w \cdot x + b$





Primer - Trening podaci

```
1 import sklearn
2 features = [[140, "smooth"], [130, "smooth"], [150, "bumpy"], [170, "bumpy"]]
3 labels = ["apple", "apple","orange", "orange"]
```

`features` sadrži prve dve kolone trening seta - input za klasifikator.

`labels` sadrži poslenju kolonu trening seta – output klasifikatora.

http://scikit-learn.org/stable/

Primer - Trening podaci

```
1 import sklearn
2 features = [[140, "smooth"], [130, "smooth"], [150, "bumpy"], [170, "bumpy"]
3 labels = ["apple", "apple","orange", "orange"]
```

`features` sadrži prve dve kolone trening seta - input za klasifikator.

`labels` sadrži poslenju kolonu trening seta – output klasifikatora.

Scikit-learn koristi realne brojeve za features i labels!

```
1 import sklearn
2 features = [[140, 1], [130, 1], [150, 0], [170, 0]]
3 labels = [0, 0, 1, 1]
```

0 – bumpy, 1 - smooth

0 – apple, 1 - orange

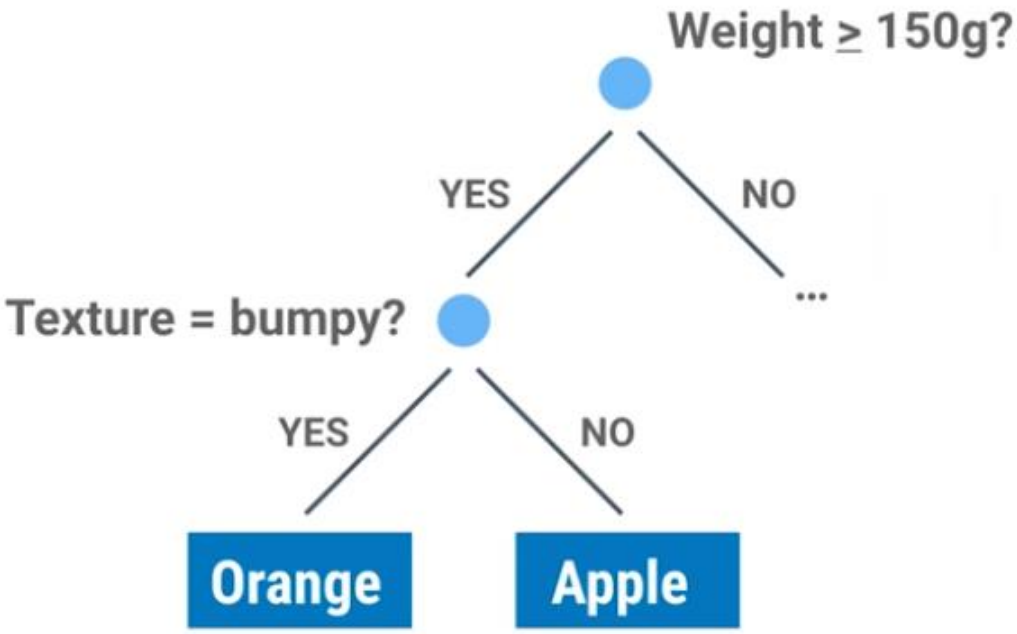
Supervised Learning – korak po korak

1. Sakupiti trening podatke
2. Trenirati klasifikator

Treniranje klasifikatora

- Klasifikatori mogu biti različiti:
  - Decision tree
  - Artificial neural network
  - Support vector machines
  - K nearest neighbours
  - …

Treniranje klasifikatora – Decision tree

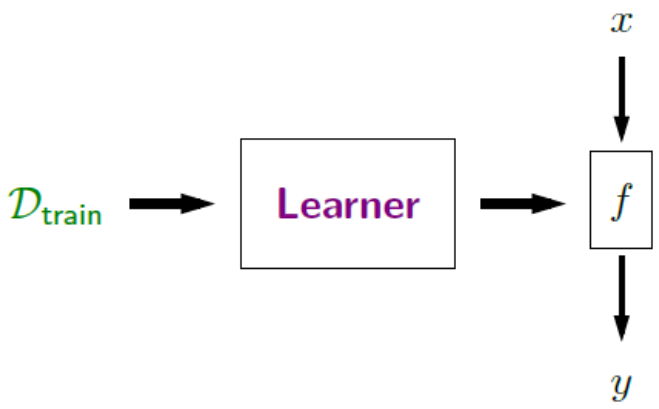


Treniranje klasifikatora – Decision tree

```
from sklearn import tree

features=[[140,1], [130,1], [150, 0], [170,0]]
labels=[0,0,1,1]

clf = tree.DecisionTreeClassifier()
```

- U ovom trenutku klasifikator je poput prazne kutije i nema nikakvo znanje o jabukama i pomorandžama.
- Da bismo ga trenirali potreban nam je algoritam za učenje

Algoritam za učenje

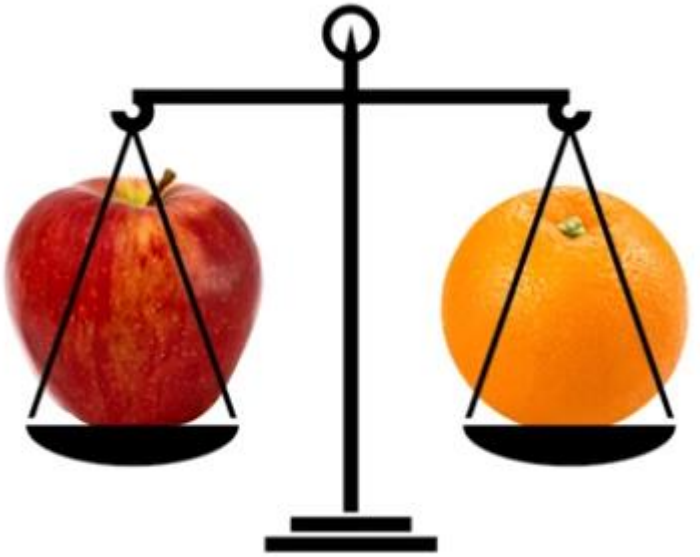


- Ako klasifikator zamislimo kao kutiju pravila koja može da klasifikuje objekte, onda algoritam za učenje kreira ta pravila.
- Učenje se sastoji u produkovanju (obučavanju) prediktora f na osnovu trening podataka.
- Obučeni prediktor treba da bude u stanju da za do tada neviđeni ulaz (ulaz koji nije bio deo trening skupa), predvidi izlaz.

Algoritam za učenje

- Algoritam za učenje pronalazi obrasce u podacima učenje.

- Na primer, može da uoči da je pomorandža obično teža od jabuke.



Algoritam za učenje

- U scikit-u, algoritam za učenje je deo klasifikator objekta i zove se fit.
- Fit pronalazi paterne u podacima.

```
from sklearn import tree

features=[[140,1], [130,1], [150, 0], [170,0]]
labels=[0,0,1,1]

clf = tree.DecisionTreeClassifier()
clf = clf.fit(features, labels)
```

Supervised Learning – korak po korak

1. Sakupiti trening podatke
2. Trenirati klasifikator
3. Iskoristiti klasifikator

Primer

- Upotrebićemo obučeni klasifikator da bismo klasifikovali novu voćku koja ima 160g i neravnu koru

```
from sklearn import tree

features=[[140,1], [130,1], [150, 0], [170,0]]
labels=[0,0,1,1]

clf = tree.DecisionTreeClassifier()
clf = clf.fit(features, labels)

print clf.predict([[160,0]])
```

Da li biste ovu voćku svrstali u jabuke ili pomorandže?

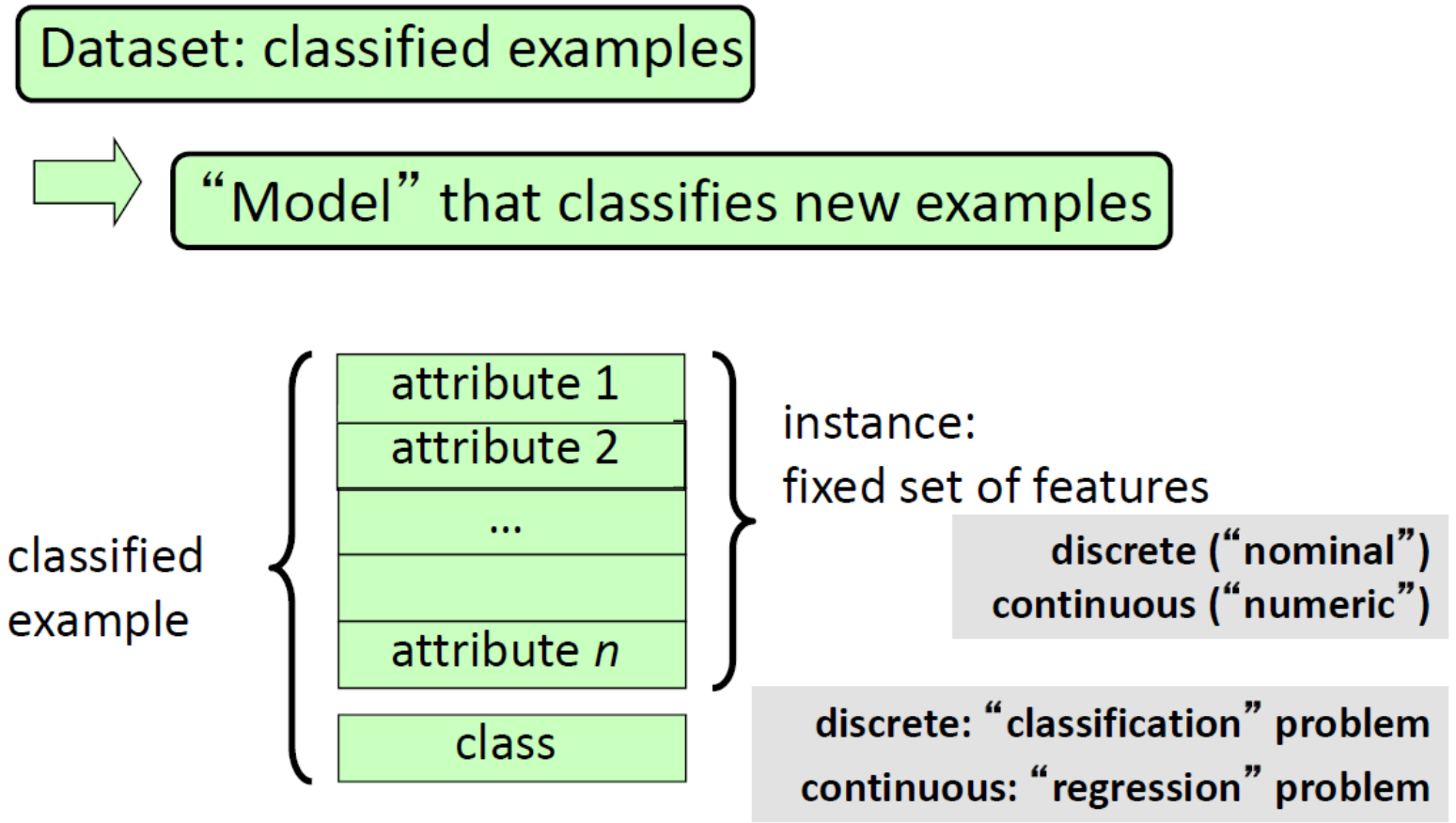
Primer

- Upotrebićemo obučeni klasifikator da bismo klasifikovali novu voćku koja ima 160g i neravnu koru

```
from sklearn import tree

features=[[140,1], [130,1], [150, 0], [170,0]]
labels=[0,0,1,1]

clf = tree.DecisionTreeClassifier()
clf = clf.fit(features, labels)

print clf.predict([[160,0]])
```

Klasifikator predviđa da je u pitanju pomorandža

Summary – supervised learning



Zaključak

- Novi klasifikator za novi problem možete napraviti korišćenjem potpuno istog koda.
- Samo je potrebno da izmenite trening skup.
- Zato je mašinsko učenje mnogo bolje od klasičnog pristupa gde bismo morali da pišemo nova pravila za svaki novi problem.

Primer realnog data set-a

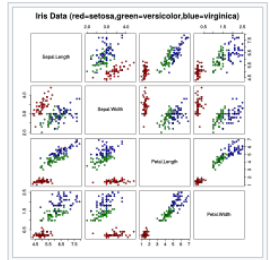
- Iris - https://en.wikipedia.org/wiki/Iris_flower_data_set

Iris flower data set

From Wikipedia, the free encyclopedia

The ***Iris* flower data set** or **Fisher's *Iris* data set** is a multivariate data set introduced by the British statistician and biologist Ronald Fisher in his 1936 paper *The use of multiple measurements in taxonomic problems* as an example of linear discriminant analysis.[1] It is sometimes called **Anderson's *Iris* data set** because Edgar Anderson collected the data to quantify the morphologic variation of *Iris* flowers of three related species.[2] Two of the three species were collected in the Gaspé Peninsula "all from the same pasture, and picked on the same day and measured at the same time by the same person with the same apparatus".[3]

The data set consists of 50 samples from each of three species of *Iris* (*Iris setosa*, *Iris virginica* and *Iris versicolor*). Four features were measured from each sample: the length and the width of the sepals and petals, in centimetres. Based on the combination of these four features, Fisher developed a linear discriminant model to distinguish the species from each other.



Scatterplot of the data set

Contents [hide]

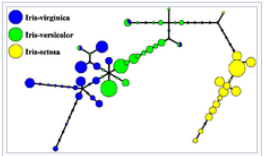
Use of the data set [edit]

Based on Fisher's linear discriminant model, this data set became a typical test case for many statistical classification techniques in machine learning such as support vector machines[5].

The use of this data set in cluster analysis however is not common, since the data set only contains two clusters with rather obvious separation. One of the clusters contains *Iris setosa*, while the other cluster contains both *Iris virginica* and *Iris versicolor* and is not separable without the species information Fisher used. This makes the data set a good example to explain the difference between supervised and unsupervised techniques in data mining: Fisher's linear discriminant model can only be obtained when the object species are known: class labels and clusters are not necessarily the same.[6]

Nevertheless, all three species of *Iris* are separable in the projection on the nonlinear branching principal component.[7] The data set is approximated by the closest tree with some penalty for the excessive number of nodes, bending and stretching. Then the so-called "metro map" is constructed.[4] The data points are projected into the closest node. For each node the pie diagram of the projected points is prepared. The area of the pie is proportional to the number of the projected points. It is clear from the diagram (left) that the absolute majority of the samples of the different *Iris* species belong to the different nodes. Only a small fraction of *Iris-virginica* is mixed with *Iris-versicolor* (the mixed blue-green nodes in the diagram). Therefore, the three species of Iris (*Iris setosa*, *Iris virginica* and *Iris versicolor*) are separable by the unsupervising procedures of nonlinear principal component analysis. To discriminate them, it is sufficient just to select the corresponding nodes on the principal tree.


Ronald Fisher



An example of the so-called "metro map" for the *Iris* data set.[4] Only a small fraction of *Iris-virginica* is mixed with *Iris-versicolor*. All other samples of the different *Iris* species belong to the different nodes.

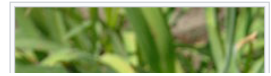
Data set [edit]

The dataset contains a set of 150 records under 5 attributes - Petal Length , Petal Width , Sepal Length , Sepal width and Class.

Fisher's *Iris* Data [hide]

Dataset Order	Sepal length	Sepal width	Petal length	Petal width	Species
1	5.1	3.5	1.4	0.2	*I. setosa*
2	4.9	3.0	1.4	0.2	*I. setosa*
3	4.7	3.2	1.3	0.2	*I. setosa*
4	4.6	3.1	1.5	0.2	*I. setosa*
5	5.0	3.6	1.4	0.3	*I. setosa*
6	5.4	3.9	1.7	0.4	*I. setosa*
7	4.6	3.4	1.4	0.3	*I. setosa*

Unsatisfactory k-means clustering result (the data set does not cluster into the known classes) and actual species visualized using ELKI



Primer realnog data set-a

- Određivanje vrste cveta Iris na osnovu dužine i širine latica

Primer realnog data set-a

- Primer uključuje tri različite vrste irisa

- *Iris setosa*
- *Iris versicolor*
- *Iris virginica*

- U skupu je dato po 50 primera od svake vrste irisa
- Irisu su opisani korišćenjem 4 svojstva (dužime i širine čašičnih listića (sepal) i latica (pedal))

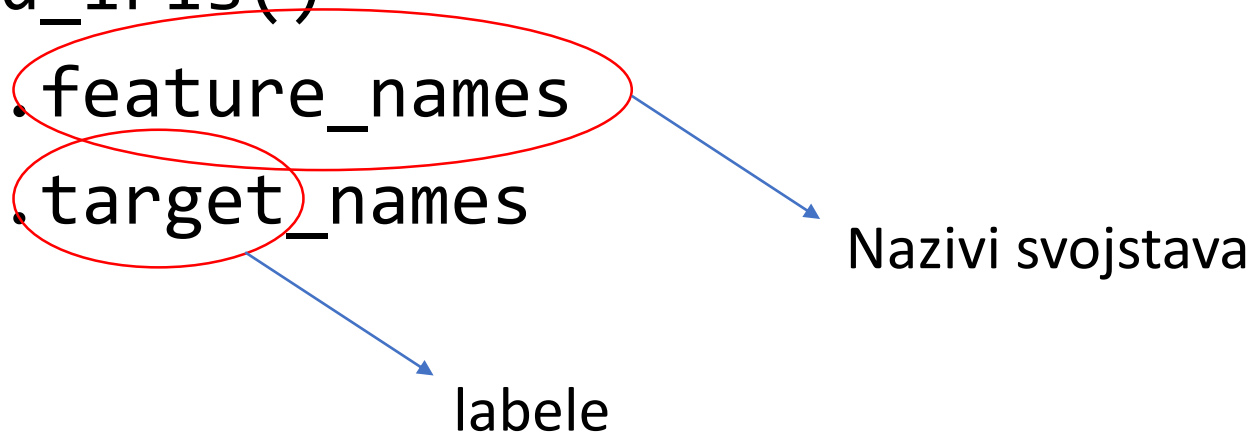
Primer realnog data set-a

Koraci:

- Učitati dataset.
- Obučiti klasifikator.
- Upotrebiti ga za određivanje vrste primera irisa koji nije sadržan u datasetu.
- Vizualizovati decision tree.

Učitavanje data set-a

```
from sklearn.datasets import load_iris

iris = load_iris()
print iris.feature_names
print iris.target_names
```



Pregled data set-a

- Konkretne vrednosti svojstava za primere podataka u dataset-u sadržane su u promenljivoj data.
- Na primer, ako isprintamo prvi član, videćemo mere za ovaj cvet

```
print iris.data[0]
```

- Target promenljiva sadrži nazive cveća

```
print iris.target[0]
```

Podela Data set-a na trening i testni skup

```
import numpy as np
        #training data - uklanjamo samo po jedan primer cveta od svake vrste
test_idx = [0,50,100]
train_target = np.delete(iris.target, test_idx)
train_data = np.delete(iris.data, test_idx, axis = 0)

        #testing data
test_target = iris.target[test_idx]
test_data = iris.data[test_idx]
```

Obučavanje i testiranje klasifikatora

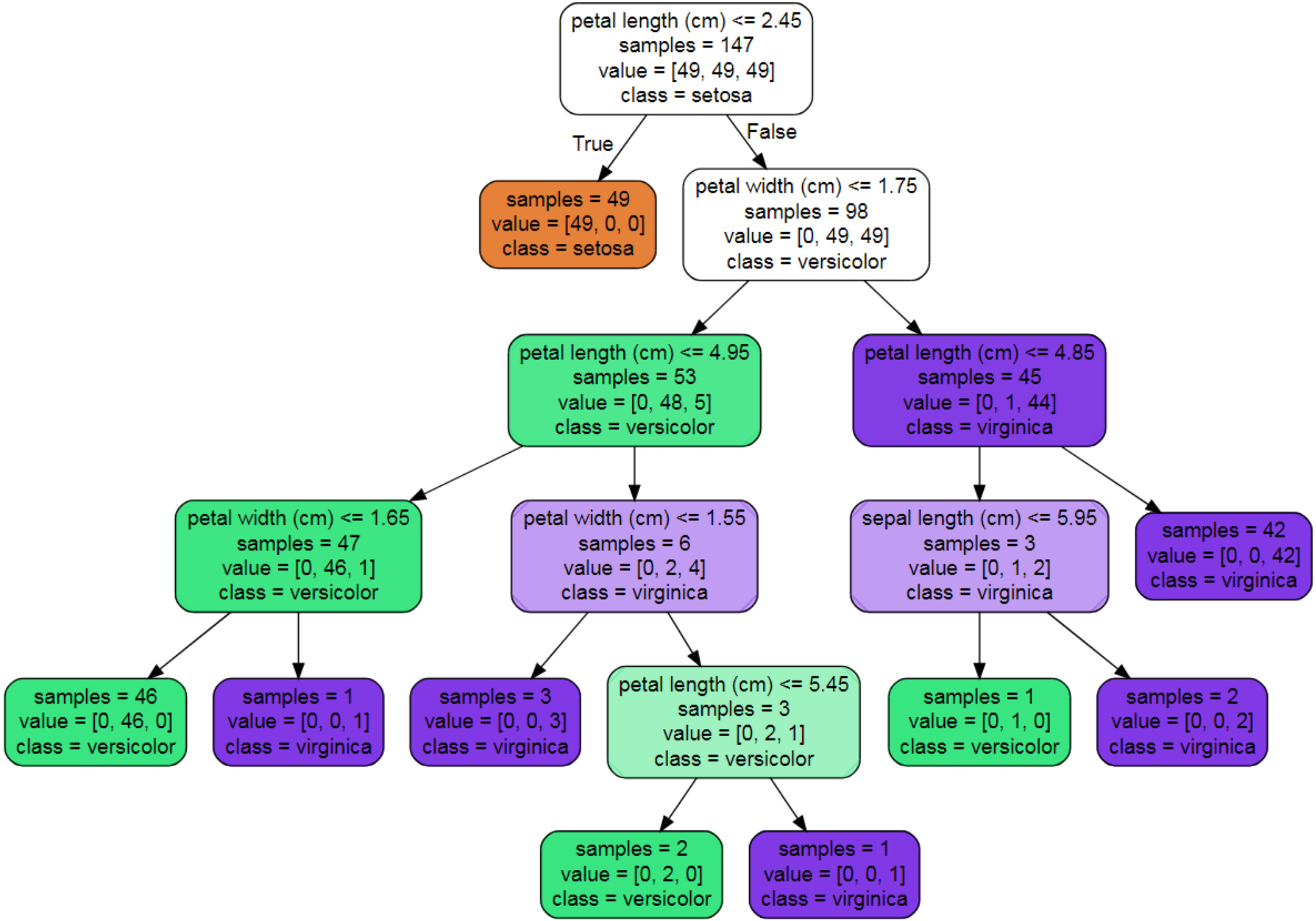
```
from sklearn import tree

clf = tree.DecisionTreeClassifier()
clf.fit(train_data, train_target)

print clf.predict(test_data)
```

Klasifikatoru na ulazu dajemo svojstva (features) naših testnih podataka, a na izlazu dobijamo labele

Vizuelizacija

```
with open("clf.txt", "w") as f:

f = tree.export_graphviz(clf, out_file=f,
                         feature_names = iris.feature_names,
                         class_names=iris.target_names,
                         filled=True,rounded=True,
                         impurity=False)
```

http://webgraphviz.com/


petal length (cm) <= 2.45
samples = 147
value = [49, 49, 49]
class = setosa
True
False
samples = 49
value = [49, 0, 0]
class = setosa
petal width (cm) <= 1.75
samples = 98
value = [0, 49, 49]
class = versicolor
petal length (cm) <= 4.95
samples = 53
value = [0, 48, 5]
class = versicolor
petal length (cm) <= 4.85
samples = 45
value = [0, 1, 44]
class = virginica
petal width (cm) <= 1.65
samples = 47
value = [0, 46, 1]
class = versicolor
petal width (cm) <= 1.55
samples = 6
value = [0, 2, 4]
class = virginica
sepal length (cm) <= 5.95
samples = 3
value = [0, 1, 2]
class = virginica
samples = 42
value = [0, 0, 42]
class = virginica
samples = 46
value = [0, 46, 0]
class = versicolor
samples = 1
value = [0, 0, 1]
class = virginica
samples = 3
value = [0, 0, 3]
class = virginica
petal length (cm) <= 5.45
samples = 3
value = [0, 2, 1]
class = versicolor
samples = 1
value = [0, 1, 0]
class = versicolor
samples = 2
value = [0, 0, 2]
class = virginica
samples = 2
value = [0, 2, 0]
class = versicolor
samples = 1
value = [0, 0, 1]
class = virginica

Supervised learning

- Supervised learning - Agent posmatra neke primere parova ulazno-izlaz i nauči LEARNING funkciju koja preslikava ulaze u izlaze.

Given a **training set** of N example input–output pairs

$$(x_1, y_1), (x_2, y_2), \ldots (x_N, y_N) \ ,$$

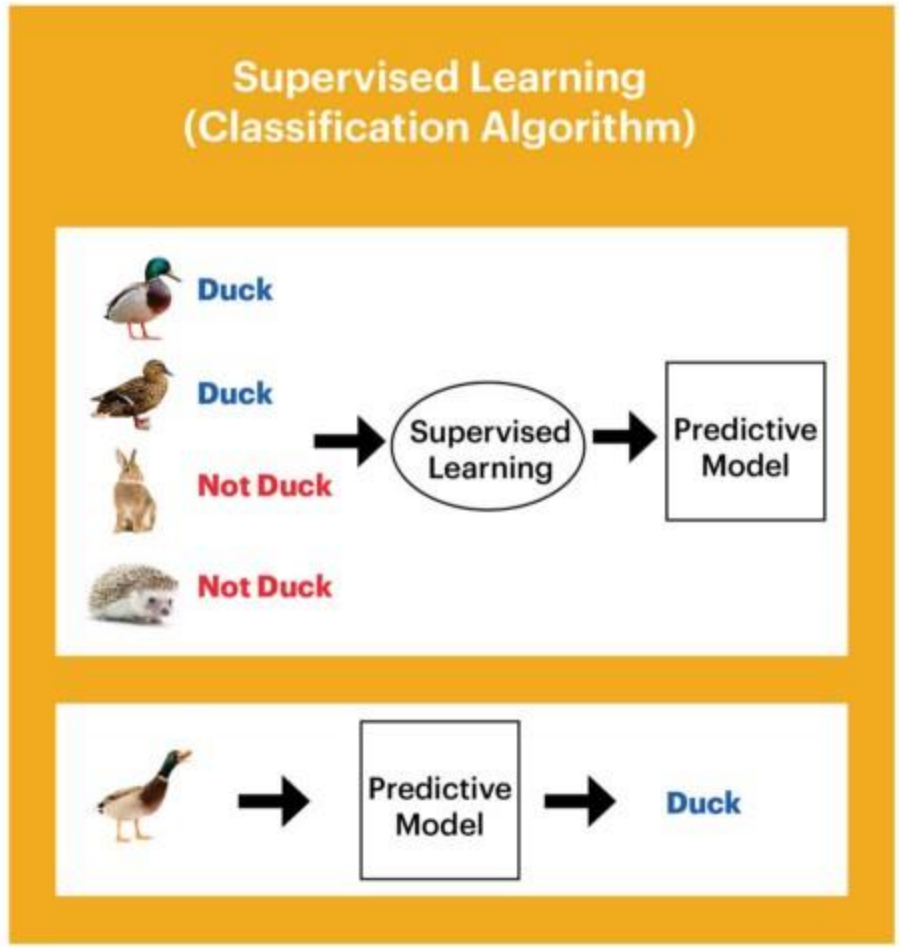
where each y_j was generated by an unknown function $y = f(x)$, discover a function h that approximates the true function f.

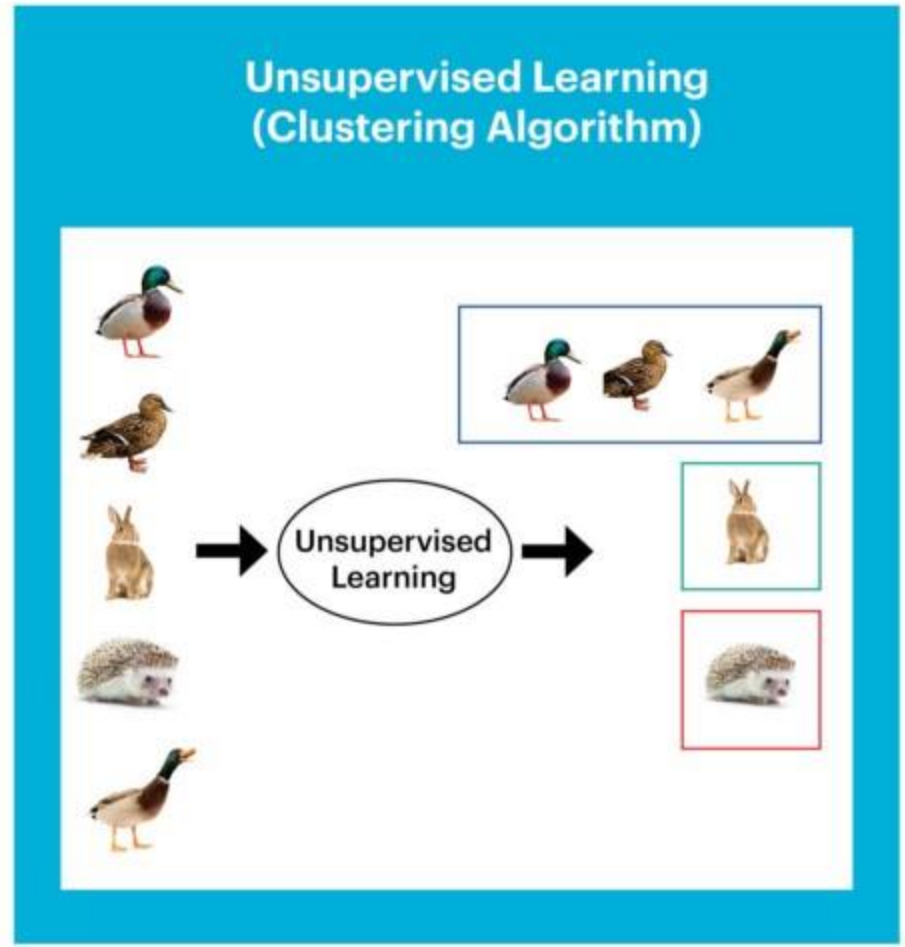
Primeri supervised learning-a

- Klasifikacija –izlaz je jedna vrednost iz konačnog skupa vrednosti
  - Given email, predict if it’s spam or not (discrete)
- Regresija – izlaz je kontinualna vrednost
  - Predict price of the houses based on their size and location
- Ova podela nije strogo formalno, jer tehnički, binarna klasifikacija može biti podvedena pod regresioni problem sa labelama -1 i +1

Primeri algoritama supervised learning-a

- Klasifikacija:
  - Decision Trees – stabla odlučivanja
  - Logistička regresija
  - Artificial Neural Networks – veštačke neuronske mreže
  - Support Vector Machines – metoda potpornih vektora
  - Naive Bayes – naivni Bajesovski klasifikator
  - K nearest neighbour – k najbližih suseda
- Regresija:
  - Linearna regresija

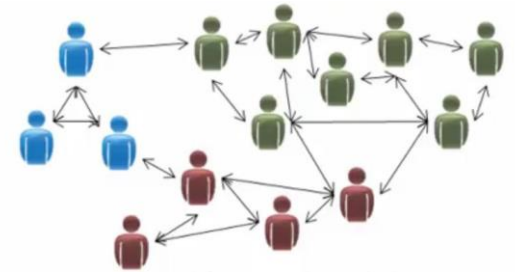




Western Digital.

Unsupervised learning – nenadgledano učenje

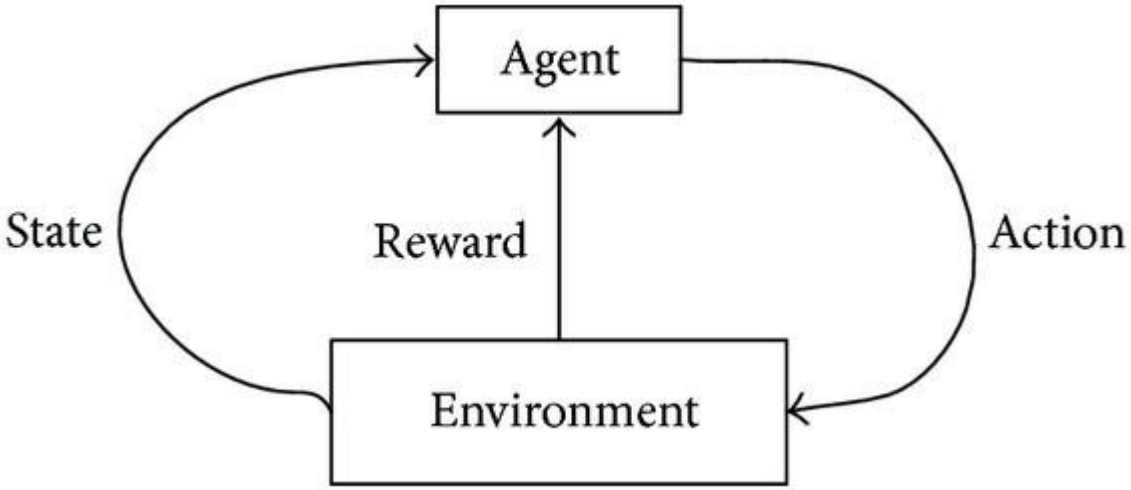
- Radi sa neoznačenim podacima.
- Agent uči obrasce u ulaznim podacima iako mu nije pružena eksplicitna povratna sprega.
- Uobičajeni zadatak nenadgledanog obučavanja je klasterizacija (grupisanje) – otkrivanje potencijalno korisnih grupa ulaznih primera.
  - K-means algoritam



Reinforcement learning

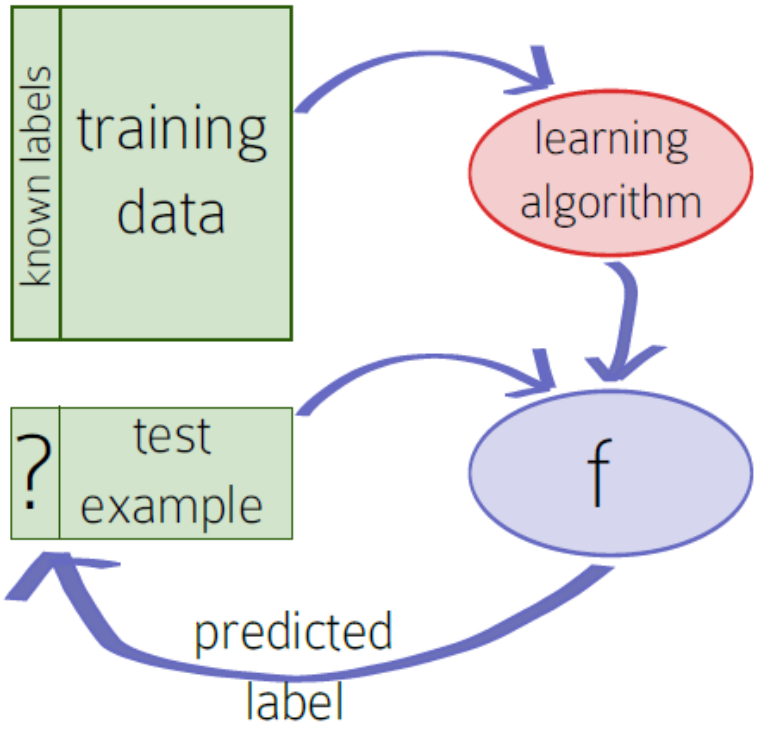
- Agent uči iz niza pojačavanja – nagrada ili kazni.

- Na primer, nedostatak napojnice na kraju vožnje daje taksi agentu indikaciju da nešto nije bilo u redu.



Obnavljanje – nadgledano obučavanje

- Cilj algoritama mašinskog učenja je određivanje (učenje) funkcije (f) koja najbolje preslikava vrednosti ulaznih promenljivih (X) u vrednosti izlaznih promenljivih (Y).
- Output = f (Input)
- f – model, hipoteza
- Algoritam uči funkciju f iz trening podataka.
- Naučena funkcija f može predvideti output za novi input koji nije korišćen tokom treniranja.



Uobičajeni problemi mašinskog učenja

Regression: trying to predict a real value. For instance, predict the value of a stock tomorrow given its past performance. Or predict Alice's score on the machine learning final exam based on her homework scores.

Binary Classification: trying to predict a simple yes/no response. For instance, predict whether Alice will enjoy a course or not. Or predict whether a user review of the newest Apple product is positive or negative about the product.

Multiclass Classification: trying to put an example into one of a number of classes. For instance, predict whether a news story is about entertainment, sports, politics, religion, etc. Or predict whether a CS course is Systems, Theory, AI or Other.

Ranking: trying to put a set of objects in order of relevance. For instance, predicting what order to put web pages in, in response to a user query. Or predict Alice's ranked preferences over courses she hasn't taken.

Stabla odlučivanja

Primer – otkrij ko je vampir

	Ima senku	Jede beli luk	Ten	Akcenat	Vampir
1	?	Da	Bled	Nema	Ne
2	Da	Da	Rumen	Nema	Ne
3	?	Ne	Rumen	Nema	Da
4	Ne	Ne	Prosečan	Težak	Da
5	?	Ne	Prosečan	Čudan	Da
6	Da	Ne	Bled	Težak	Ne
7	Da	Ne	Prosečan	Težak	Ne
8	?	da	Rumen	Čudan	Ne

Primer – otkrij ko je vampir

	Ima senku	Jede beli luk	Ten	Akcenat	Vampir
1	?	Da	Bled	Nema	Ne
2	Da	Da	Rumen	Nema	Ne
3	?	Ne	Rumen	Nema	Da
4	Ne	Ne	Prosečan	Težak	Da
5	?	Ne	Prosečan	Čudan	Da
6	Da	Ne	Bled	Težak	Ne
7	Da	Ne	Prosečan	Težak	Ne
8	?	da	Rumen	Čudan	Ne

- Nenumerički (simbolički atributi)
- Neki atributi nam nisu od značaja za odlučivanje, ili su značajni, ali ne sve vreme
- Testiramo vrednosti atributa da bismo došli do zaključka.

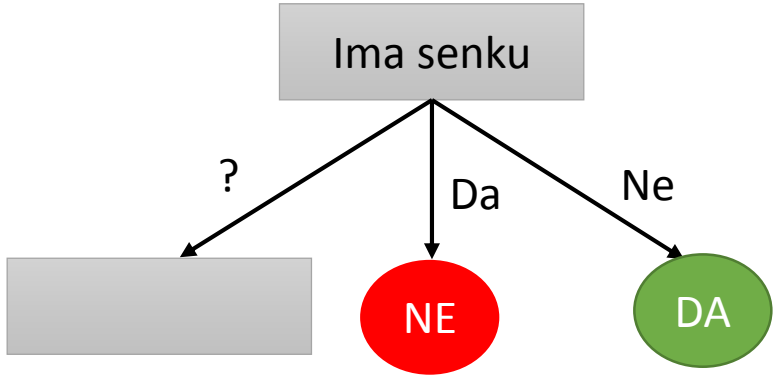
Primer – otkrij ko je vampir

	Ima senku	Jede beli luk	Ten	Akcenat	Vampir
1	?	Da	Bled	Nema	Ne
2	Da	Da	Rumen	Nema	Ne
3	?	Ne	Rumen	Nema	Da
4	Ne	Ne	Prosečan	Težak	Da
5	?	Ne	Prosečan	Čudan	Da
6	Da	Ne	Bled	Težak	Ne
7	Da	Ne	Prosečan	Težak	Ne
8	?	da	Rumen	Čudan	Ne

Da možete postaviti samo jedno pitanje, koje bi to pitanje bilo?

Primer – otkrij ko je vampir

	Ima senku	Jede beli luk	Ten	Akcenat	Vampir
1	?	Da	Bled	Nema	Ne
2	Da	Da	Rumen	Nema	Ne
3	?	Ne	Rumen	Nema	Da
4	Ne	Ne	Prosečan	Težak	Da
5	?	Ne	Prosečan	Čudan	Da
6	Da	Ne	Bled	Težak	Ne
7	Da	Ne	Prosečan	Težak	Ne
8	?	da	Rumen	Čudan	Ne

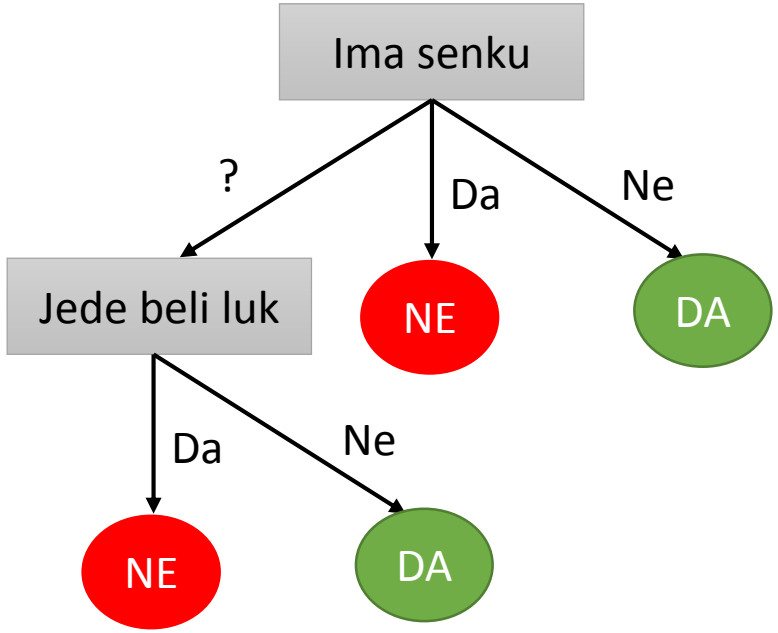


Primer – otkrij ko je vampir

	Ima senku	Jede beli luk	Ten	Akcenat	Vampir
1	?	Da	Bled	Nema	Ne
2	Da	Da	Rumen	Nema	Ne
3	?	Ne	Rumen	Nema	Da
4	Ne	Ne	Prosečan	Težak	Da
5	?	Ne	Prosečan	Čudan	Da
6	Da	Ne	Bled	Težak	Ne
7	Da	Ne	Prosečan	Težak	Ne
8	?	da	Rumen	Čudan	Ne

Primer – otkrij ko je vampir

	Ima senku	Jede beli luk	Ten	Akcenat	Vampir
1	?	Da	Bled	Nema	Ne
2	Da	Da	Rumen	Nema	Ne
3	?	Ne	Rumen	Nema	Da
4	Ne	Ne	Prosečan	Težak	Da
5	?	Ne	Prosečan	Čudan	Da
6	Da	Ne	Bled	Težak	Ne
7	Da	Ne	Prosečan	Težak	Ne
8	?	da	Rumen	Čudan	Ne



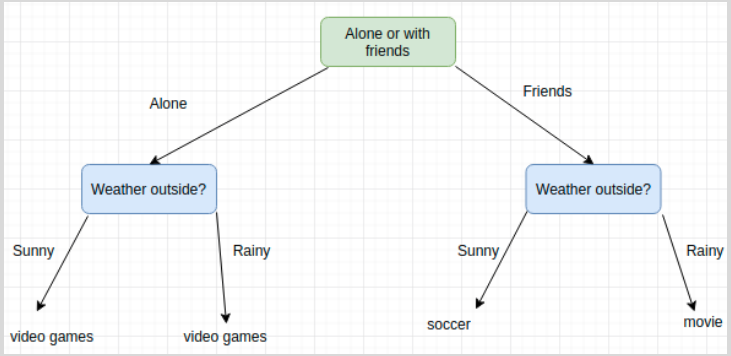
Stablo odlučivanja je klasifikator koji ima strukturu stabla

- U svakom čvoru odluke (Decision node) testira po jedan atribut
- Čvor list (Leaf node): određuje vrednost ciljne funkcije (rezultat klasifikacije)
- Svaka grana (branch) koja polazi iz nekog čvora odgovara jednoj od mogućih vrednosti atributa koji je testiran u tom čvoru.
- Putanja odlučivanja: niz čvorova i grana od korenog čvora do jednog od listova koji predstavlja konačnu odluku.

Osnovni zahtevi za primenu stabala odlučivanja

- **Opis „Atribut-vrednost“ (Attribute-value description):** primeri moraju biti izraženi u obliku konačne kolekcije svojstava ili atributa (npr. hot, mild, cold).
- **Predefinisane klase (target values):** ciljna funkcija ima diskretne izlaze (**discrete output values)**
- **Dovoljno podataka (Sufficient data):** dovoljan broj trening primera kako bi bilo moguće obučiti stablo.

Zašto stablo odlučivanja?



- Stabla odlučivanja su najjednostavniji mehanizam za klasifikaciju i predikciju.
- Na osnovu stabla odlučivanja mogu se generisati pravila, koja ljudi mogu da razumeju i koja mogu biti upotrebljena za formiranje baze znanja.
- Skup odluka koje se donose po određenoj hijerarhiji, dok se ne dođe do konačnog rezultata.

Okamova oštrica (Occam’s razor)

- Broj odluka koje se donose u stablu treba da bude što je moguće manji, a da se pritom ne naruši tačnost klasifikacije.
- Princip za rešavanje problema izborom onog rešenja koje je najjednostavnije.
- Ovo heurističko načelo, kaže da kada postoji više hipoteza, onda treba izabrati onu koja se može dokazati sa najmanje pretpostavki.
- Kada se u nauci neka pojava objašnjava na takav način da se neprekidno uvode nove i nove pretpostavke samo da bi teorija opstala, Okamova oštrica zapravo pokazuje kako nešto nije u redu sa teorijom.

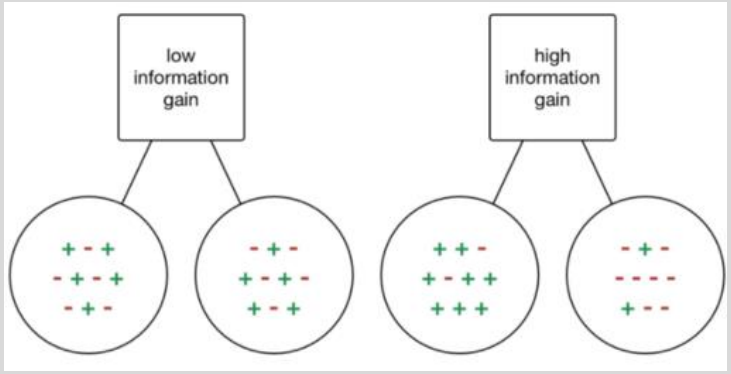
Proces učenja stabala odlučivanja

- Decision Tree model se kreira kroz učenje i orezivanje (pruning).
- Učenje je proces formiranja stabla, tj. skupa hijarahijski uređenih tačaka odluke na osnovu trening podataka.
- Pruning je proces uklanja nepotrebnih delova stabla čine se efektivno smanjuje njegova kompleksnost, povećava interpretabilnost i rešava overfitting.

Obučavanje stabla odlučivanja

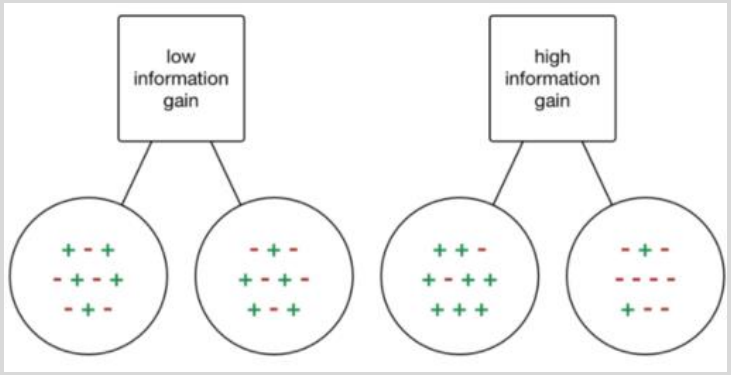
- Cilj: Formirati stablo odlučivanja koje je u saglasnosti sa trening primerima.
- Ideja: (rekurzivno) odabrati “najznačajniji” atribut kao koren (pod)stabla

Koji atribut je najbolji klasifikator?



- Selekcija atributa koji će biti testiran u svakom čvoru – biramo najkorisniji (najinformativniji) atribut za klasifikaciju primera.
- Najkorisniji atribut je onaj koji vrši najbolju podelu primera na podskupove u kojima su samo pozitivni ili samo negativni primeri.
- Ovom podelom primera po vrednosti „najinformativnijeg atributa“ se praktično kreira koreni čvor stabla.
- Novi čvorovi stabla dalje se generišu rekurzivno na osnovu sledećeg „najinformativnijeg atributa“.

Koji atribut je najbolji klasifikator?



- Selekcija trenutno najboljeg atributa u svakom koraku konstruisanja stabla odluke vrši se korišćenjem mere koja se naziva *Informaciono poboljšanje* (*Information gain*)
  - Meri koliko dobro dati atribut razdvaja trening primere prema ciljnoj klasifikaciji.
  - Ova mera se koristi kako bismo u svakom koraku izgradnje stabla, između različitih atributa odabrali onaj koji će biti naredni čvor odluke.

Entropija

- Da bismo precizno definisali information gain, moramo definisati *entropiju.*

- Entropijom merimo homogenost (uređenost) skupa podataka.

- Ako je S skup pozitivnih (onih za koje je odluka True) i negativnih primera (onih za koje je odluka false), entropija skupa S, u odnosu na ovu binarnu klasifikaciju primera, je:

$$Entropy(S) = -p_{\oplus} \log_2 p_{\oplus} - p_{\ominus} \log_2 p_{\ominus}$$

gde je $p_{\oplus}$ udeo pozitivnih primera u S a $p_{\ominus}$, je udeo negativnih primera u S

Podaci

Klasifikacija primera na pozitivne (Yes) ili negativne (No)

Oznake primera

Ne koriste se za obučavanje

Day	*Outlook*	*Temperature*	*Humidity*	*Wind*	*PlayTennis*
D1	Sunny	Hot	High	Weak	No
D2	Sunny	Hot	High	Strong	No
D3	Overcast	Hot	High	Weak	Yes
D4	Rain	Mild	High	Weak	Yes
D5	Rain	Cool	Normal	Weak	Yes
D6	Rain	Cool	Normal	Strong	No
D7	Overcast	Cool	Normal	Strong	Yes
D8	Sunny	Mild	High	Weak	No
D9	Sunny	Cool	Normal	Weak	Yes
D10	Rain	Mild	Normal	Weak	Yes
D11	Sunny	Mild	Normal	Strong	Yes
D12	Overcast	Mild	High	Strong	Yes
D13	Overcast	Hot	Normal	Weak	Yes
D14	Rain	Mild	High	Strong	No

Primer

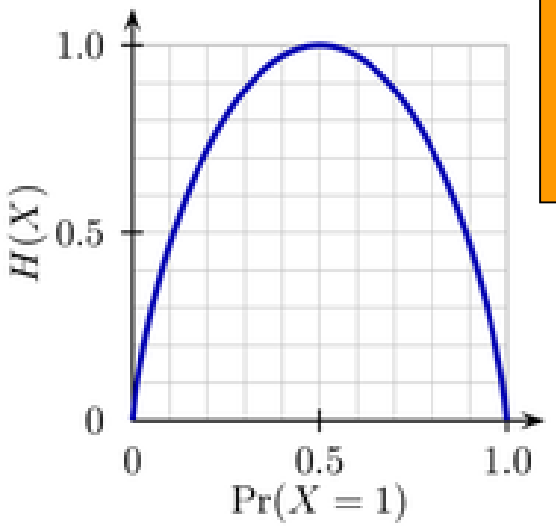
Day	*Outlook*	*Temperature*	*Humidity*	*Wind*	*PlayTennis*
D1	Sunny	Hot	High	Weak	No
D2	Sunny	Hot	High	Strong	No
D3	Overcast	Hot	High	Weak	Yes
D4	Rain	Mild	High	Weak	Yes
D5	Rain	Cool	Normal	Weak	Yes
D6	Rain	Cool	Normal	Strong	No
D7	Overcast	Cool	Normal	Strong	Yes
D8	Sunny	Mild	High	Weak	No
D9	Sunny	Cool	Normal	Weak	Yes
D10	Rain	Mild	Normal	Weak	Yes
D11	Sunny	Mild	Normal	Strong	Yes
D12	Overcast	Mild	High	Strong	Yes
D13	Overcast	Hot	Normal	Weak	Yes
D14	Rain	Mild	High	Strong	No

- 14 primera za učenje
- 9 pozitivnih
- 5 negativnih

$$Entropy(S) = -\frac{9}{14}\log_2\frac{9}{14} - \frac{5}{14}\log_2\frac{5}{14} = 0.94$$

Karakteristike entropije

- Entropija je 0 ako svi primeri iz S pripadaju istoj klasi.
- Na primer, ako su svi pozitivni ($p_{\oplus} = 1$), tada je $p_{\ominus} = 0$, i $Entropy(S) = -\log_2 1 = 0$
- Entropija je 1 ako ne posedujemo nikakvo znanje o sistemu (ili ako je svaki ishod podjednako moguć).
- Entropija je 1 ako kolekcija primera sadrži jednak broj pozitivnih i negativnih primera.
- Ako broj pozitivnih i negativnih primera nije jednak vrednost entropije je između 0 i 1.



Entropy of a 2-class problem with regard to the portion of one of the two groups

- Ako ciljna funkcija može uzeti neku od c različitih vrednosti (npr. ako klasifikujemo voće na jabuke, kruške i pomorandže, tada je c=3) , onda je entropija od S

$$Entropy(S) = \sum_{i=1}^{c} -p_i \log_2 p_i$$

Gde je p_i udeo primera iz S koji pripadaju klasi i.

- Maksimalna vrednost entropije je sada $\log_2 c$

Šta ako klasifikacija nije binarna?

Informaciono poboljšanje (Information Gain)

- Entropija je mera neizvesnosti kolekcije trening primera, koja nam omogućava da definišemo meru efikasnosti nekog atributa u klasifikovanju trening podataka. Ta mera, ***information gain,*** predstavlja očekivano umanjenje entropije nakon podele trening primera korišćenjem nekog atributa.

- Preciznije, informaciono poboljšanje, $Gain(S, A)$ koje se dobija raspodelom primera iz S na osnovu njihovih vrednosti za atribut A je:

$$Gain(S, A) = Entropy(S) - \sum_{v \in Values(A)} \frac{|S_v|}{|S|} Entropy(S_v)$$

Gde je $Values(A)$ skup svih mogućih vrednosti atributa A, a S_v, je podskup primera iz S za koje atribut A ima vrednost v, tj. $S_v = \{s \in S | A(S) = v\}$.

Informaciono poboljšanje

$$Gain(S, A) = Entropy(S) - \sum_{v \in Values(A)} \frac{|S_v|}{|S|} Entropy(S_v)$$

- Prvi deo prethodne jednakosti je entropija originalnog skupa *S,* a drugi izraz u jednakosti je očekivana vrednost entropije nakon podele primera iz *S* prema vrednosti atributa *A.*

- Ako znamo da je danas sunčano, da je umereno toplo, da je normalna vlažnost vazduha i da duva slab vetar, da li ćemo igrati tenis?

today	sunny	mild	normal	weak	?

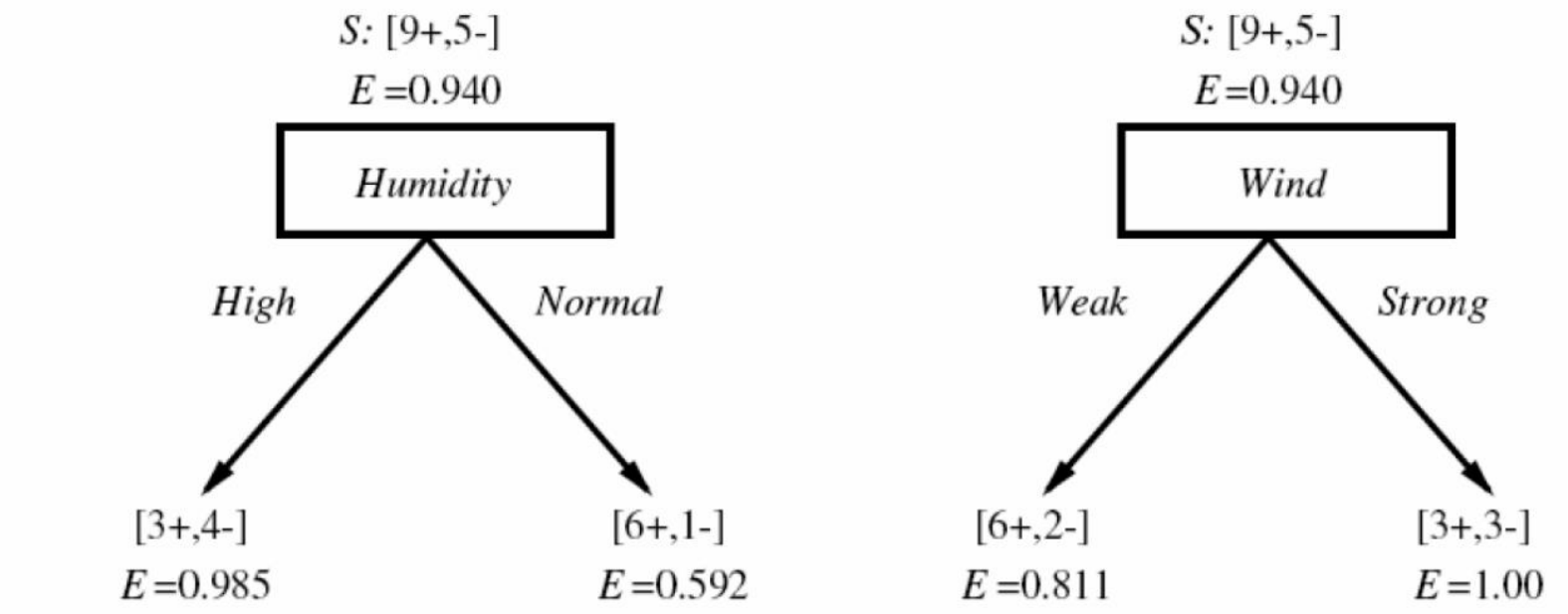
- Na ovo pitanje treba da nam da odgovor stablo odlučivanja.

- Obučavanjem stabla odlučivanja dobija se mehanizam za klasifikaciju koji može naučena pravila da primeni na nove primere i donese odluku.

Informaciono poboljšanje za atribut Wind

$$
\begin{aligned}
Values(Wind) &= Weak, Strong \\
S &= [9+, 5-] \\
S_{Weak} &\leftarrow [6+, 2-] \\
S_{Strong} &\leftarrow [3+, 3-] \\
Gain(S, Wind) &= Entropy(S) - \sum_{v \in \{Weak, Strong\}} \frac{|S_v|}{|S|} Entropy(S_v) \\
&= Entropy(S) - (8/14) Entropy(S_{Weak}) \\
&\quad - (6/14) Entropy(S_{Strong}) \\
&= 0.940 - (8/14)0.811 - (6/14)1.00 \\
&= 0.048
\end{aligned}
$$

Informaciono poboljšanje za svaki od atributa



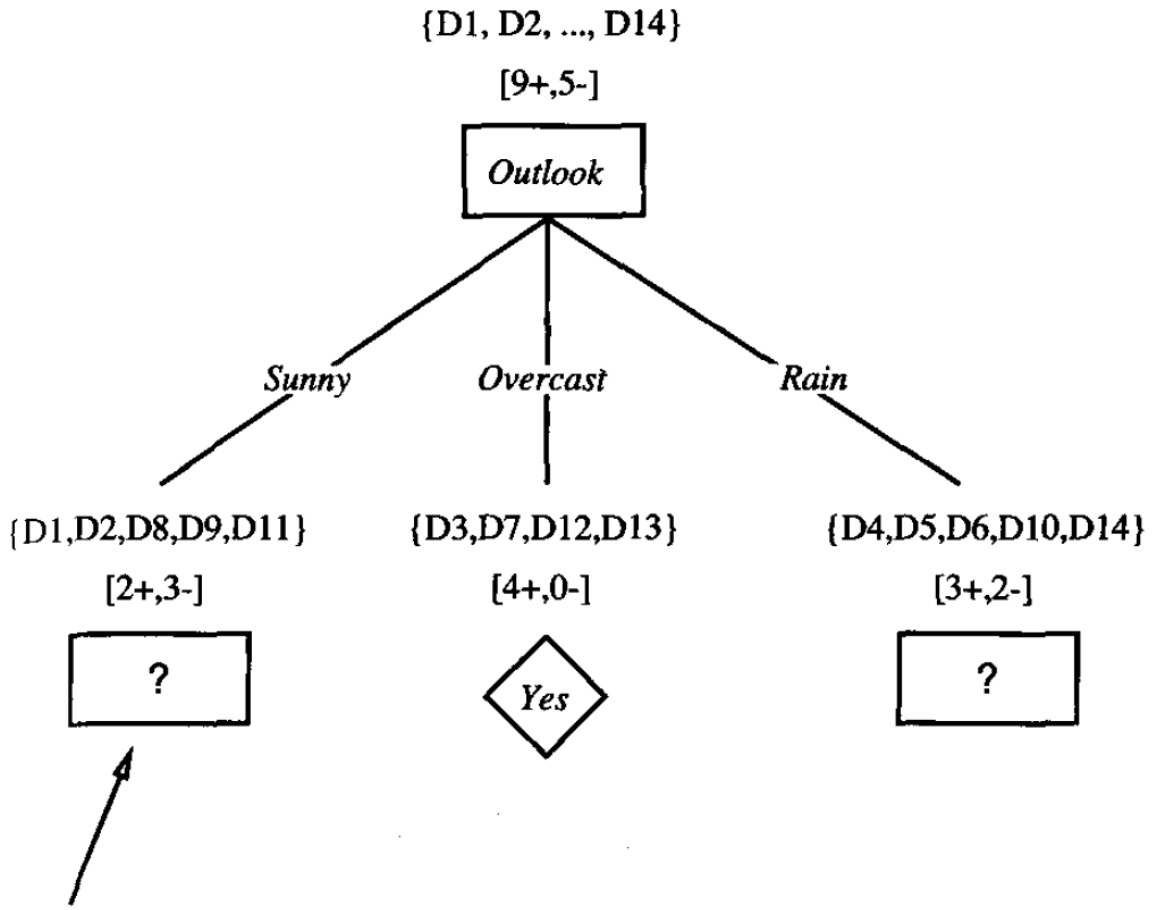
Gain (S, Humidity)
= .940 - (7/14).985 - (7/14).592
= .151

Gain (S, Wind)
= .940 - (8/14).811 - (6/14)1.0
= .048

$Gain(S, Outlook) = 0.246$

$Gain(S, Temperature) = 0.029$

Nakon što atribut Outlook postavimo u koren stabla:



Koji atribut treba testirati u ovom čvoru?

Sledeći najbolji atribut je Humidity

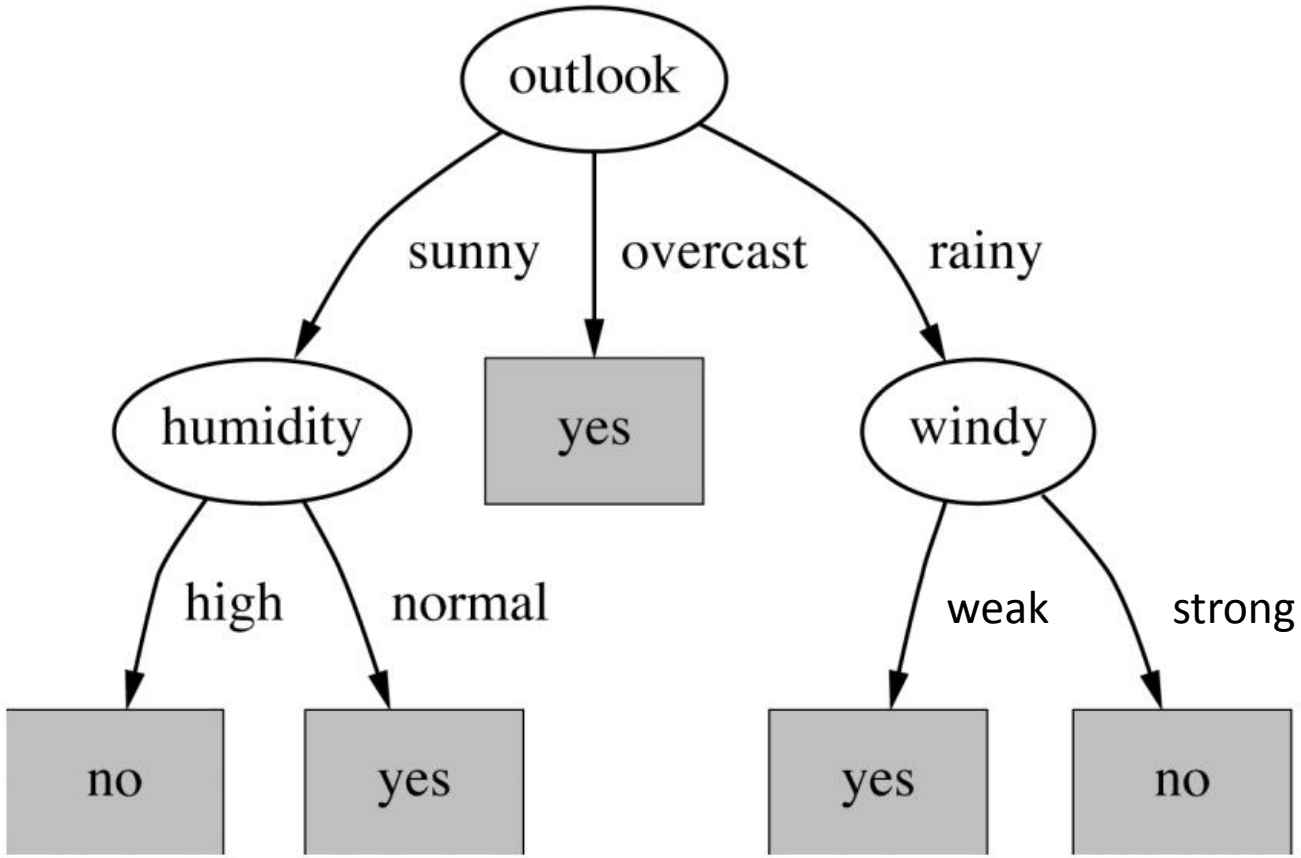
$S_{sunny} = \{D1,D2,D8,D9,D11\}$

$Gain\,(S_{sunny}, Humidity) = .970 - (3/5)\,0.0 - (2/5)\,0.0 = .970$

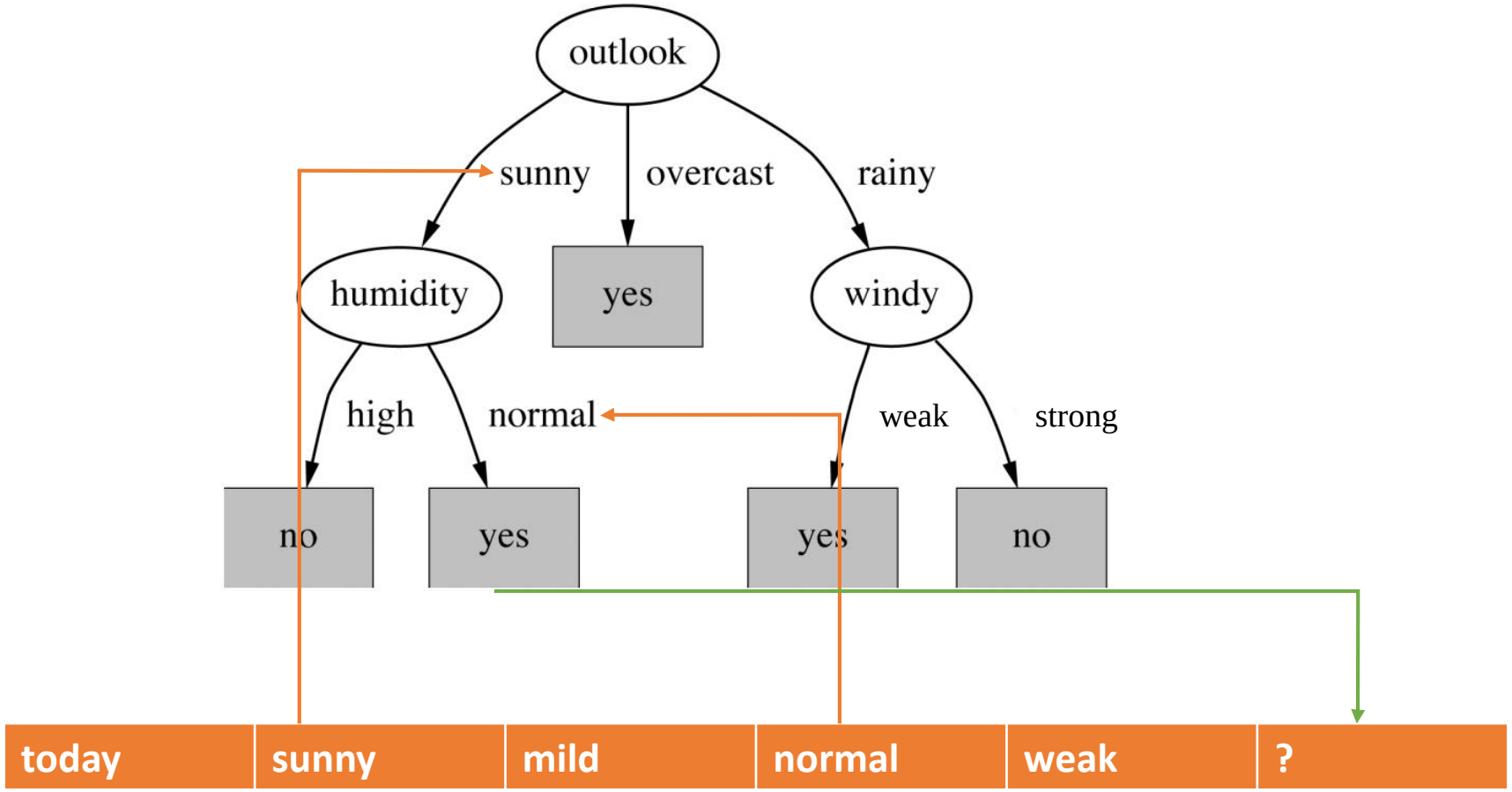
$Gain\,(S_{sunny}, Temperature) = .970 - (2/5)\,0.0 - (2/5)\,1.0 - (1/5)\,0.0 = .570$

$Gain\,(S_{sunny}, Wind) = .970 - (2/5)\,1.0 - (3/5)\,.918 = .019$

Konačno stablo odluke



Predikcija pomoću formiranog stabla



Algoritam obučavanja stabla odlučivanja

```
function DTL(examples, attributes, default) returns a decision tree

    if examples is empty then return default
    else if all examples have the same classification then return the classification
    else if attributes is empty then return MODE(examples)
    else
        best ← CHOOSE-ATTRIBUTE(attributes, examples)
        tree ← a new decision tree with root test best
        for each value v_i of best do
            examples_i ← {elements of examples with best = v_i}
            subtree ← DTL(examples_i, attributes − best, MODE(examples))
            add a branch to tree with label v_i and subtree subtree
        return tree
```

Prednosti stabala odlučivanja

- Generišu lako razumljiva pravila
- Obavljaju klasifikaciju bez mnogo izračunavanja
- Pružaju jasan uvid u značaj pojedinih atributa za klasifikaciju primera i predikciju.

Mane stabala odlučivanja

- Ne mogu se primeniti na probleme koji podrazumevaju predviđanje numeričke vrednosti (npr. predvideti cenu stana na osnovu karakteristika).
- Ne daju dobre rezultate kada postoji više izlaznih klasa, a malo primera za učenje.
- Računski skupa za obučavanje.

Šta ako vrednosti atributa nisu diskretne?

- Primena stabla odlučivanja je ograničena na atribute koji uzimaju vrednosti iz diskretnog skupa.
  1. Ciljna vrednost koju stablo treba da predvidi je diskretna.
  2. Atributi koji se testiraju u čvorovima uzimaju diskretne vrednosti.

- Uslov 2 se lako može ukloniti tako da neprekidni (kontinualni) atributi takođe budu uključeni u stablo.

Šta ako vrednosti atributa nisu diskretne?

- Definisati novi atribut sa diskretnim vrednostima koji deli neprekidni opseg vrednosti kontinualnog atributa na skup intervala.
- Za atribut A koji je kontinualan, algoritam može dinamički odrediti novi bulovski atribut A_c koji ima vrednost $true$ ako je vrednost atributa A manja od c.
- Kako odabrati prag c?

Primer

- In the current example, there are two candidate thresholds, corresponding to the values of Temperature at which the value of PlayTennis changes:

  (48 + 60)/2,

  (80 + 90)/2.

- The information gain can then be computed for each of the candidate attributes, $Temperature_{>54}$,and $Temperature_{>85}$, and the best can be selected - $Temperature_{>54}$

Temperature:	40	48	60	72	80	90
PlayTennis:	No	No	Yes	Yes	Yes	No

Primer

#	Classification	Length (L)	Beak size (B)	Vibrantly colored (V) (+1 = yes, -1 = no)
1	Finch	50	1	-1
2	Robin	80	2	+1
3	Sparrow	90	3	+1
4	Robin	55	15	+1
5	Finch	65	30	-1

- Klasifikovati pticu dužine 70, sa kljunom 10 i jarko obojenu

Primer

- Potrebno je odlučiti da li ćete čekati na slobodan sto u restoranu ili ne na osnovu prethodnih iskustava.

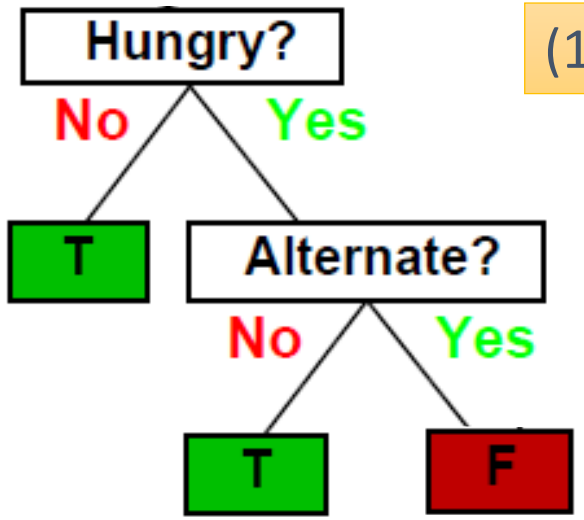
- Dostupno znanje
  - Da li u blizini postoji odgovarajući alternativni restoran
  - Da li restoran ima udoban bar gde se može čekati
  - Da li je petak/subota
  - Da li smo jako gladni
  - …..

Primer iz AIMA knjige

Klasifikacija primera na pozitivne (T) ili negativne (F)

Example	Attributes										Target
	Alt	*Bar*	*Fri*	*Hun*	*Pat*	*Price*	*Rain*	*Res*	*Type*	*Est*	*WillWait*
X_1	*T*	*F*	*F*	*T*	*Some*	*$$$*	*F*	*T*	*French*	*0–10*	*T*
X_2	*T*	*F*	*F*	*T*	*Full*	*$*	*F*	*F*	*Thai*	*30–60*	*F*
X_3	*F*	*T*	*F*	*F*	*Some*	*$*	*F*	*F*	*Burger*	*0–10*	*T*
X_4	*T*	*F*	*T*	*T*	*Full*	*$*	*F*	*F*	*Thai*	*10–30*	*T*
X_5	*T*	*F*	*T*	*F*	*Full*	*$$$*	*F*	*T*	*French*	*>60*	*F*
X_6	*F*	*T*	*F*	*T*	*Some*	*$$*	*T*	*T*	*Italian*	*0–10*	*T*
X_7	*F*	*T*	*F*	*F*	*None*	*$*	*T*	*F*	*Burger*	*0–10*	*F*
X_8	*F*	*F*	*F*	*T*	*Some*	*$$*	*T*	*T*	*Thai*	*0–10*	*T*
X_9	*F*	*T*	*T*	*F*	*Full*	*$*	*T*	*F*	*Burger*	*>60*	*F*
X_{10}	*T*	*T*	*T*	*T*	*Full*	*$$$*	*F*	*T*	*Italian*	*10–30*	*F*
X_{11}	*F*	*F*	*F*	*F*	*None*	*$*	*F*	*F*	*Thai*	*0–10*	*F*
X_{12}	*T*	*T*	*T*	*T*	*Full*	*$*	*F*	*F*	*Burger*	*30–60*	*T*

Ilustracija dela stabla

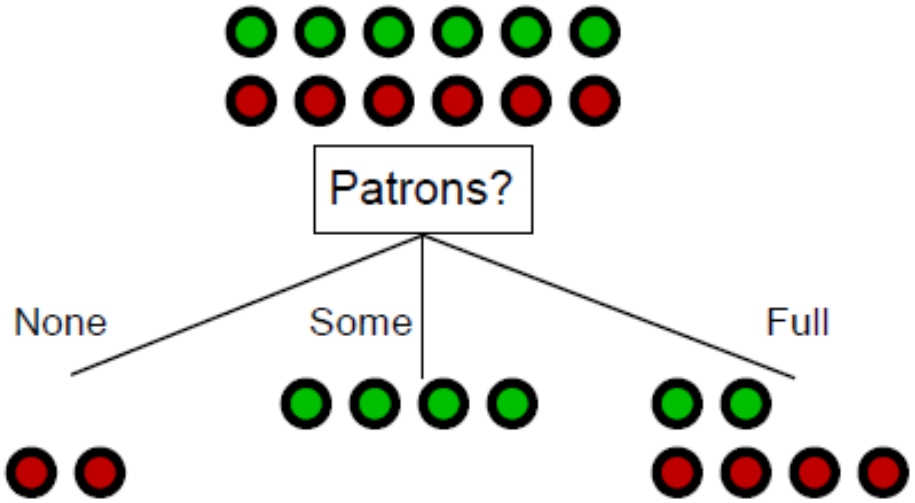


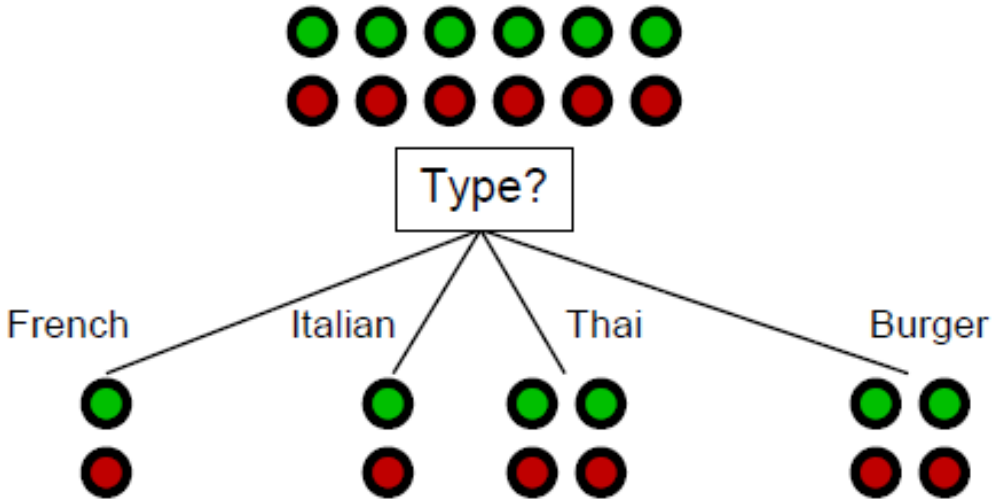
(1) Od kog čvora polazimo? (root)

(2) Naredni čvor zavisi od odabrane vrednosti atributa u prethodnom čvoru

(3) Kada smo odabirom jedne vrednosti atributa u nekom od čvorova odluke završili u čvoru koji je list došli smo do zaključka.

Koji atribut treba odabrati za koreni čvor?





- Da li je informativniji atribut Patron ili Type? Ako odaberemo Patron, imamo podelu primera na tri grupe od kojih je samo jedna mešovita (sadrži i pozitivne i negativne primere). Entropija je 0.45
- Ako odaberemo Type, imamo veću entropiju (neizvesnost) koja je 1.

https://webdocs.cs.ualberta.ca/~aixplore/learning/DecisionTrees/InterArticle/3-DecisionTree.html

Kako smo izračunali entropiju za atribut Patrons?

For "None" branch

$$-(\frac{0}{0+2}\log\frac{0}{0+2}+\frac{2}{0+2}\log\frac{2}{0+2})=0$$

For "Some" branch

$$-(\frac{4}{4+0}\log\frac{4}{4+0}+\frac{0}{4+0}\log\frac{0}{4+0})=0$$

For "Full" branch

$$-(\frac{2}{2+4}\log\frac{2}{2+4}+\frac{4}{2+4}\log\frac{4}{2+4})\approx 0.9$$

For choosing "Patrons"

weighted average of each branch: this quantity is called **conditional entropy**

$$\frac{2}{12}\times 0+\frac{4}{12}\times 0+\frac{6}{12}\times 0.9=0.45$$

Kako smo izračunali entropiju za atribut Type?

For "French" and "Italian" branch

$$-(\frac{1}{1+1}\log\frac{1}{1+1}+\frac{1}{1+1}\log\frac{1}{1+1})=1$$

For "Thai" and "Burger" branch

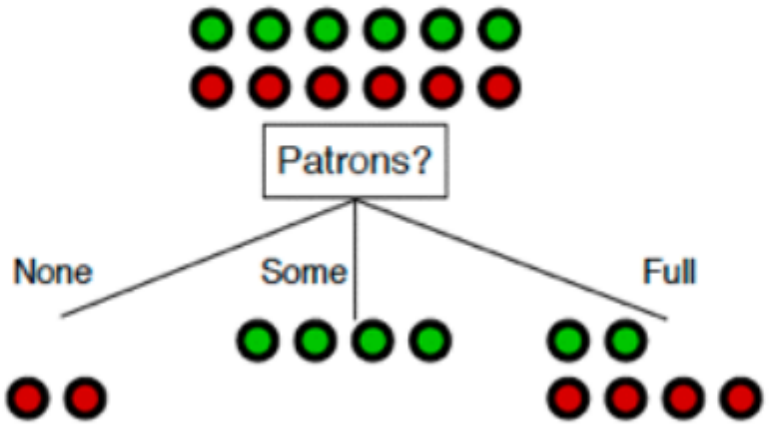
$$-(\frac{2}{2+2}\log\frac{2}{2+2}+\frac{2}{2+2}\log\frac{2}{2+2})=1$$

For choosing "Type"

weighted average of each branch (**conditional entropy**)

$$\frac{2}{12}\times 1+\frac{2}{12}\times 1+\frac{4}{12}\times 1+\frac{4}{12}\times 1=1$$

Naredni atribut za podelu?



Do we split on **"None"** or **"Some"**?

No, we do not

- The decision is **deterministic**, as seen from the training data

We will look only at the 6 instances with **Patrons** == **Full**

Konačno stablo odlučivanja: