Fakultet kemijskog inženjerstva i tehnologije

Metode umjetne inteligencije u kemijskom inženjerstvu

LINEARNA REGRESIJA

STABLO ODLUČIVANJA

SLUČAJNA ŠUMA

Željka Ujević Andrijić

zujevic@fkit.unizg.hr

UVOD U LINEARNU REGRESIJU

- **Definicija regresije:** statistička tehnika koja se koristi za modeliranje veza između zavisnih i nezavisnih varijabli.

- Cilj regresijske analize je utvrditi način na koji promjene u nezavisnim varijablama utječu na zavisnu varijablu. Omogućuje **predviđanje** vrijednosti zavisne varijable na temelju nezavisnih varijabli.

- **Jednostavna regresija**: Kada postoji jedna zavisna i jedna nezavisna varijabla.

$$y_i = \beta_0 + \beta_1 x_{i1}$$

- **Viševeličinska regresija**: Kada postoji više nezavisnih varijabli.

Viševeličinski linearni model

Što je viševeličinski linearni model?

Model koji uključuje više nezavisnih varijabli za predviđanje zavisne varijable.

Opći oblik modela:

$$y_i = \beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \cdots + \beta_K x_{iK} + e_i$$

gdje su:

- y_i je zavisna varijabla,
- x_{ij} su nezavisne varijable,
- β_j su regresijski koeficijenti,
- e_i je pogreška.

Regresijski koeficijenti viševeličinskog linearnog modela

$$y_i = \beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \cdots + \beta_K x_{iK} + e_i$$

Što su regresijski koeficijenti?

- β_0: Konstanta, vrijednost kada su nezavisne varijable nula.
- β_j: Koeficijenti koji pokazuju za koliko se zavisna varijabla mijenja kad se odgovarajuća nezavisna varijabla poveća za jedan, uz ostale varijable na konstantnoj razini.
- Npr. koeficijent β_1 pokazuje koliko se y_i mijenja kada se x_1 poveća za jedan, uz uvjet da ostale nezavisne varijable (poput x_2, x_3,…) ostanu nepromijenjene.

Procjena parametara pomoću metoda najmanjih kvadrata

- Za procjenu parametara (keoficijenata) regresijskog modela koristi se **metoda najmanjih kvadrata** koja minimizira sumu kvadrata razlika između stvarnih i predviđenih vrijednosti zavisne varijable.

- Kako bi se pronašli optimalni parametri koji minimiziraju funkciju, računaju se **parcijalne derivacije funkcije** u odnosu na svaki parametar).

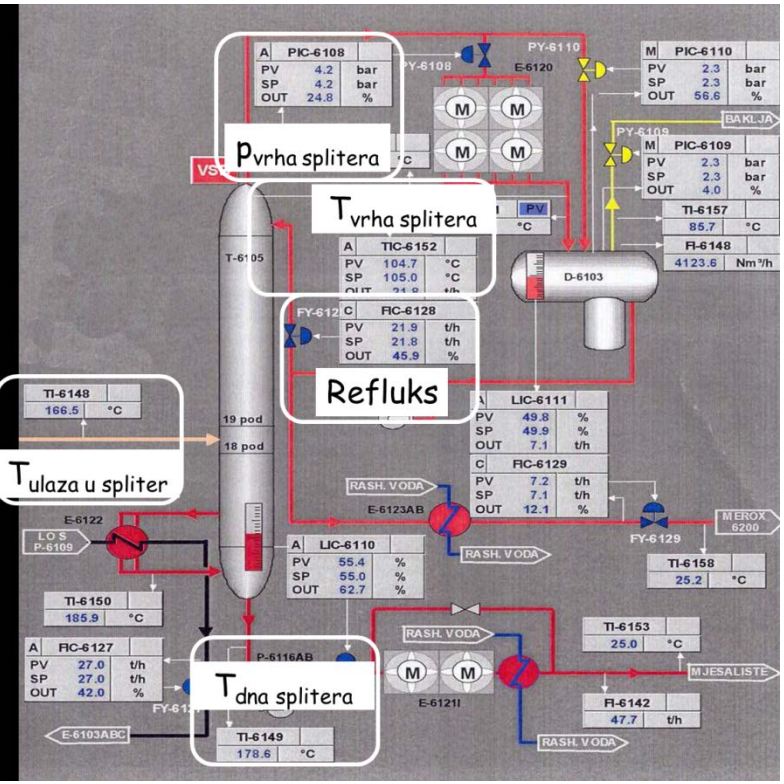
- Parcijalne derivacije regresijske funkcije daju uvid u to kako se y_i mijenja kada se promijeni samo jedna nezavisna varijabla x_j, dok su ostale varijable konstantne.

Primjer Python koda za višestruku linearnu regresiju

Paket: scikit-learn (sklearn)
Naredbe: model = LinearRegression()
model.fit(X, y)

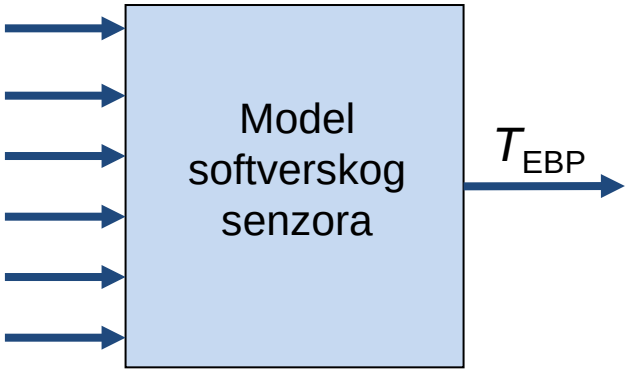
```python
from sklearn.linear_model import LinearRegression
import pandas as pd

# Učitavanje podataka iz Excel datoteke
data = pd.read_excel('data.xlsx')

# Definiranje nezavisnih varijabli i zavisne varijable
X = data[['x1', 'x2', 'x3']]  # nezavisne varijable
y = data['y']  # zavisna varijabla

# Kreiranje i treniranje modela
model = LinearRegression()
model.fit(X, y)

# Prikaz koeficijenata i konstante
print('Koeficijenti:', model.coef_)
print('Konstanta:', model.intercept_)
```

Primjer razvoja viševeličinskog linearnog modela



Predviđanje temperature kraja destilacije proizvoda dna splitera

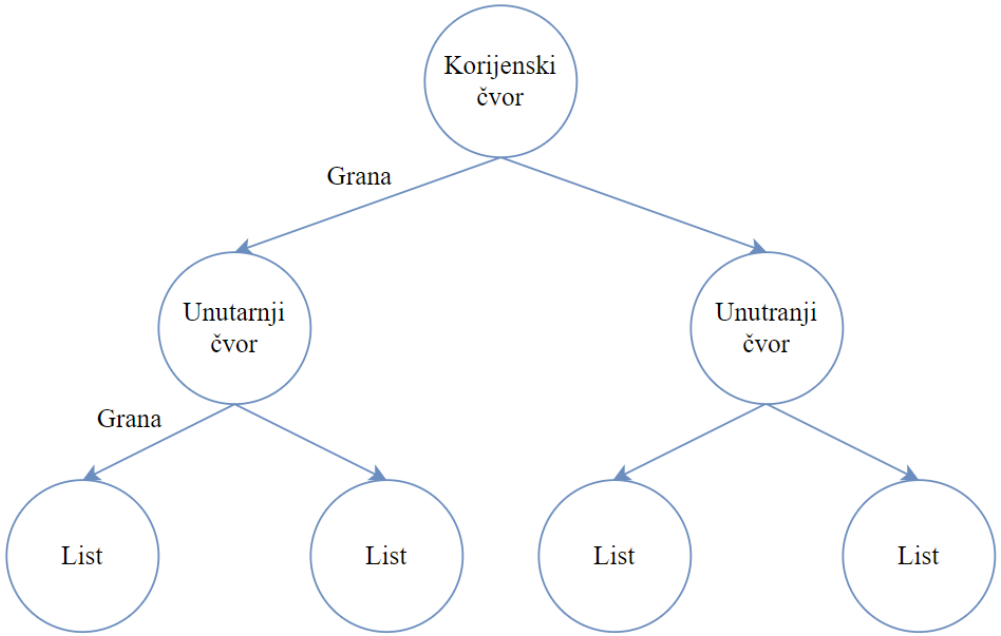
Temperatura vrha kolone
Temperatura ulaza u spliter
Temperatura vrha splitera
Temperatura dna splitera
Tlak vrha splitera
Protok refluksa (t/h)



$$T_{EBP} = -28{,}747 + 1{,}135 \cdot T_{vrhkol} - 6{,}084 \cdot p_{vrhsp} - 0{,}068 \cdot T_{vrhsp} - 0{,}035 \cdot \text{Refluks} + 0{,}190 \cdot T_{ulsp} + 0{,}354 \cdot T_{dnosp}$$

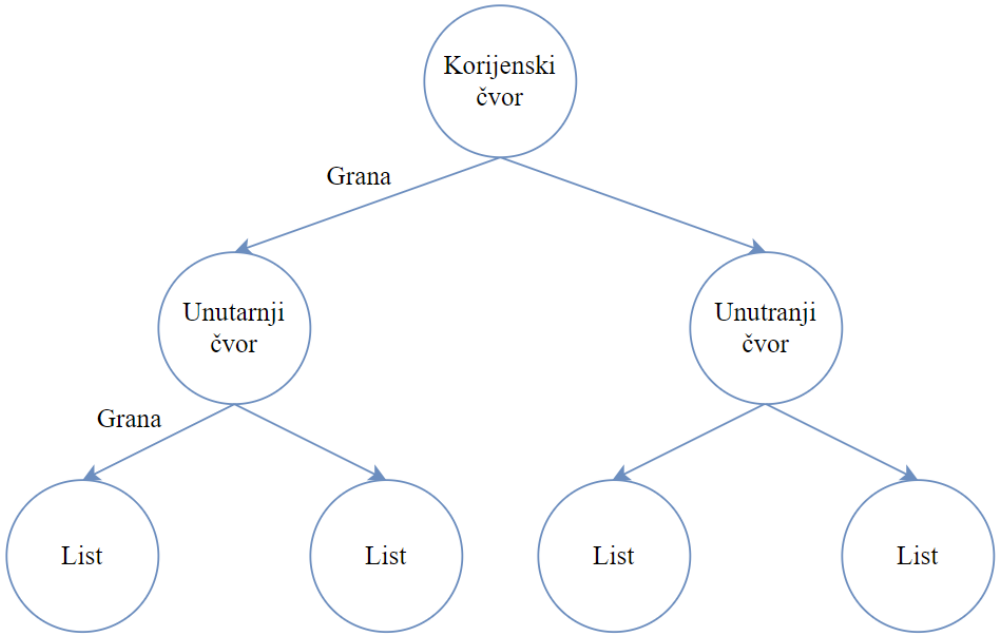
STABLO ODLUČIVANJA

- **Stablo odlučivanja** (engl. ***decision tree***) je nadzirani model strojnog učenja koji ima strukturu nalik stablu.
- Njegova glavna prednost je što omogućuje jednostavno razumijevanje i interpretaciju odluka, jer jasno prikazuje kako se donose odluke kroz različite faze analize podataka.
- Model organizira podatke u hijerarhijsku strukturu sastavljenu od korijenskog čvora (engl. *root node)*, unutarnjih čvorova (engl. *internal nodes*), grana (engl. *branches*) i listova (engl. *leaf nodes*).



STABLO ODLUČIVANJA

- **Korijenski čvor**: početni čvor u stablu odlučivanja, gdje se skup za učenje počinje dijeliti na temelju atributa podataka. Atributi su karakteristike ili svojstva koja se koriste za podjelu podataka.
- **Unutarnji čvorovi** provode testove ili donose odluke temeljem atributa podataka te predstavljaju mjesta daljnjeg razdvajanja podataka.
- **Grane** predstavljaju ishod testa ili odluke te vode do sljedećeg čvora ili **lista** koji predstavlja konačnu predikciju ili odluku.



STABLO ODLUČIVANJA

Izgradnja stabla odlučivanja sastoji se od sljedećih koraka:

- **Odabir najboljeg atributa:** Prvi korak je odabrati atribut koji najbolje razdvaja podatke prema nekom kriteriju.

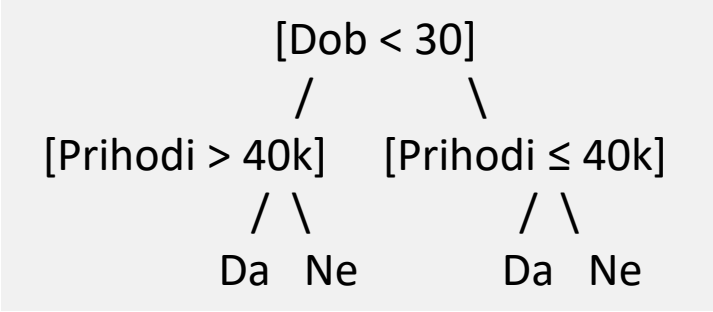
- **Podjela skupa podataka:** Na temelju odabranog atributa, podaci se dijele na podskupove.

- **Ponavljanje procesa:** Proces se ponavlja rekurzivno za svaki podskup, stvarajući novi unutarnji čvor ili list dok se ne ispune određeni kriteriji za zaustavljanje, poput unaprijed određenih dubina stabla ili kada podaci u čvoru pripadaju istoj klasi.

Primjena stabla odlučivanja - hoće li osoba kupiti proizvod

Imamo podatke o kupovini proizvoda na temelju dva atributa: **dob** i **prihodi** na temelju kojih stablo odlučivanja može donijeti odluku hoće li osoba kupiti proizvod ili ne.

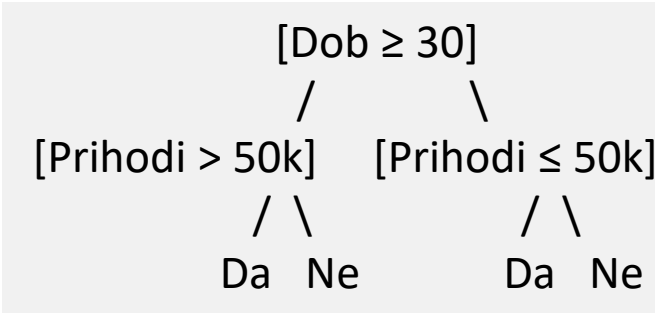
Korak 1: Odabir atributa koji najbolje razdvaja podatke
Može se koristiti **dob** kao prvi atribut. Podaci se dijele u dvije grane:
Dob < 30
Dob ≥ 30



Korak 2: Podjela skupa podataka
Na temelju odabrane granice za dob, podaci se dijele u dva skupa. Zatim, za svaki skup podataka, može se koristiti drugi atribut — **prihodi**.

Ako je Dob < 30:
Ako su prihodi > 40.000: Osoba će kupiti proizvod (**List: Da**).
Ako su prihodi ≤ 40.000: Osoba neće kupiti proizvod (**List: Ne**).

Ako je Dob ≥ 30:
Ako su prihodi > 50.000: Osoba će kupiti proizvod (List: Da).
Ako su prihodi ≤ 50.000: Osoba neće kupiti proizvod (List: Ne).

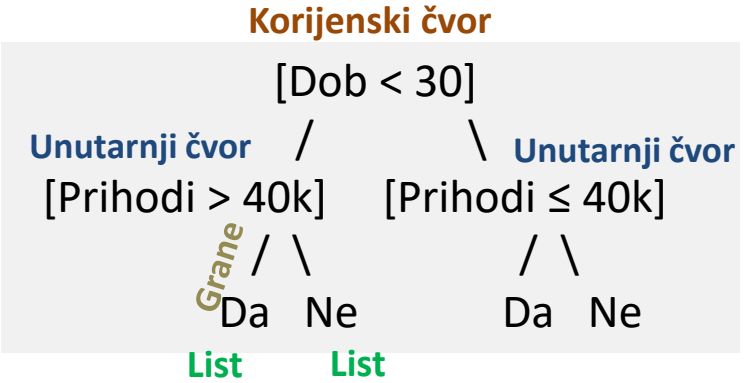


Primjena stabla odlučivanja - hoće li osoba kupiti proizvod

- **Korak 3: Ponavljanje procesa**

Proces se ponavlja za svaki podskup podataka, stvarajući nove unutarnje čvorove dok se ne ispune uvjeti za zaustavljanje. Npr. može se zaustaviti kad svi podaci u čvoru pripadaju istoj klasi (npr. svi će kupiti proizvod) ili kad se postigne unaprijed zadana dubina stabla.

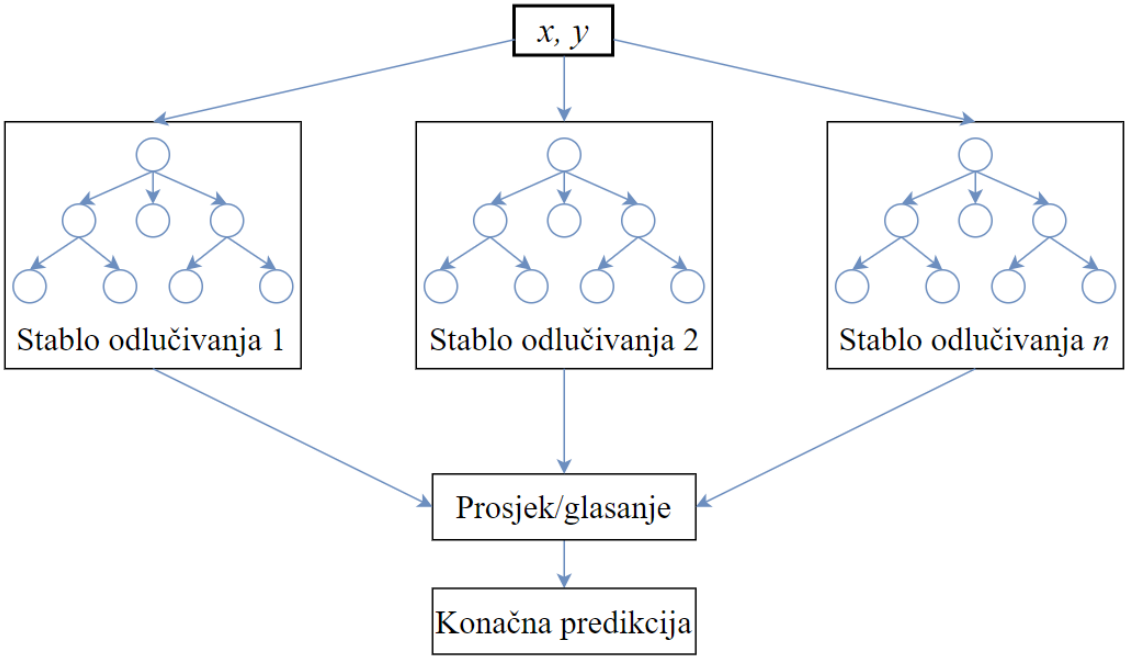
- **Korijenski čvor**: Početni čvor u stablu gdje dijelimo podatke prema atributu "dob".
- **Unutarnji čvorovi**: Mjesta gdje se podaci dalje razdvajaju prema drugim atributima, u ovom slučaju prema prihodima.
- **Grane**: Ove linije povezuju čvorove i predstavljaju mogući ishod testa, tj. ako su njeni prihodi veći od 40.000.
- **Listovi**: Predstavljaju konačnu odluku (kupnja ili ne kupnja proizvoda).



SLUČAJNA ŠUMA

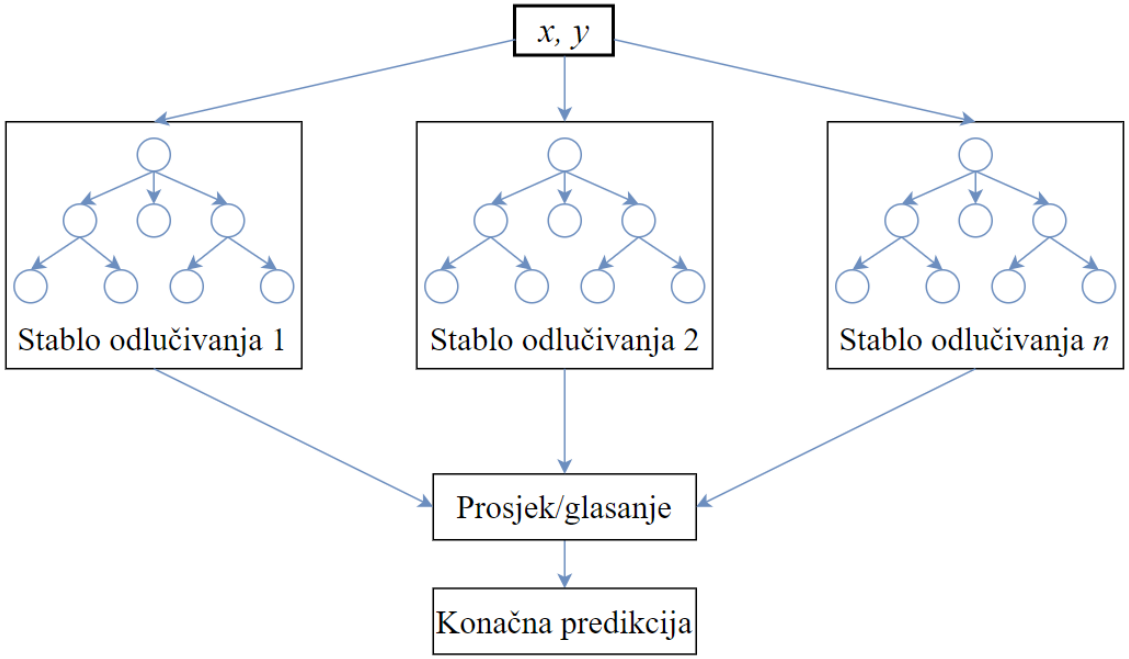
- **Slučajna šuma** (engl. ***random forest***) je ansambl metoda strojnog učenja koja koristi velik broj stabala odlučivanja s ciljem poboljšanja točnosti i robusnosti modela.
- Ključne karakteristike slučajne šume: **pakiranje** (engl. ***bootstraping***) i **nasumičan odabir značajki**.
- **Pakiranje** je ansambl metoda koja koristi više modela temeljenih na istom algoritmu učenja.

- Svaki model se uči na različitim **nasumično odabranim** podskupovima originalnog skupa podataka.
- Kombiniranjem rezultata putem računanja **prosjeka** u regresiji ili **glasanja** (engl. *Majority voting*) u klasifikaciji, dobiva se konačna predikcija.



SLUČAJNA ŠUMA

- Svako stablo u slučajnoj šumi gradi se koristeći **nasumičan podskup podataka** i **nasumičan odabir značajki**. Za svaki unutarnji čvor, nasumično se odabire manji skup značajki iz ukupnog skupa značajki. Ovaj pristup povećava raznolikost među stablima.
- Na temelju tog nasumično odabranog skupa značajki, bira se najbolja značajka za testiranje u svakom čvoru, a taj proces se ponavlja za sve čvorove u stablu.
- Rezultat svakog stabla odlučivanja je jedna predikcija, a konačna predikcija modela slučajne šume dobiva se računanjem **prosjeka** rezultata u regresiji ili **glasanjem** u klasifikaciji.



Primjer Python koda za stablo odlučivanja i slučajne šume

Paket: scikit-learn (sklearn)

```python
import pandas as pd
# Učitavanje podataka iz Excel datoteke
data = pd.read_excel('data.xlsx')

# Definiranje nezavisnih varijabli i zavisne varijable
X = data[['x1', 'x2', 'x3']]  # nezavisne varijable
y = data['y']  # zavisna varijabla


# Stablo odlučivanja / Slučajna šuma (Decision Tree / Random Forest)

from sklearn.tree import DecisionTreeRegressor

DTRmodel = DecisionTreeRegressor()

DTRmodel.fit(X, y)

y_pred = DTRmodel.predict(X)



from sklearn.ensemble import RandomForestRegressor

RFRmodel = RandomForestRegressor()

RFRmodel.fit(X, y)

y_pred = RFRmodel.predict(X)
```