Regresia liniara

Analiza regresiei este unul din domeniile de baza ale Inteligentei Artificiale si Statisticii, regresia liniara fiind una dintre metodele simple disponibile si usor de inteles.

Regresia este un algoritm care cauta relatii intre mai multe **variabile**. Spre exemplu, poate fi studiat in ce fel pretul apartamentelor depinde de factori precum suprafata, cartierul, numarul de camere, etaj etc.

Datele legate de un apartament reprezinta o **observatie** (o inregistrare). Se pleaca de la prezumptia ca variabilele de mai sus(suprafata, cartierul etc) sunt factori independenti, pretul depinzand de acestia.

Scopul regresiei va fi asadar gasirea unei functii care sa stabileasca suficient de bine o relatie intre variabila dependenta(tinta) si variabilele independente.

In practica variabila tinta se noteaza cu y, iar variabilele de intrare se noteaza cu x.

Regresia liniara este una din cele mai cunoscute datorita simplitatii interpretarii rezultatelor.

Forma generala a regresiei:

$$y = \beta_0 + \beta_1 x_1 + \cdots + \beta_r x_r$$

Unde $\mathbf{x} = (x_1, \ldots, x_r)$

r - numarul de variabile independente

$\beta_0, \beta_1, \ldots, \beta_r$ - coeficientii regresiei

Functia de estimare a regresiei este $(\mathbf{x}) = b_0 + b_1 x_1 + \cdots + b_r x_r$, unde $b_{0,1,\ldots,} b_r$ sunt coeficientii estimati ai regresiei sau ponderi estimate.

Pentru a estima aceste ponderi, o metoda uzuala este aceea de a minimiza suma patratica a rezidurilor pentru toate observatiile $\sum_i (y_i - f(\mathbf{x}_i))^2$, denumita si metoda celor mai mici patrate.

Performanta regresiei poate fi masurata cu ajutorul coeficientului de determinare R^2, care arata in ce masura o variatie a lui y este explciata de x, aplicand modelul de regresie. Un R^2 mai mare indica un model mai robust. Valoarea $R^2=1$ corespunde unei erori (SSR) egala cu 0, adica o potrivire perfecta a modelului cu datele.

Regresia liniara cu o singura variabila

Forma generala $(x) = b_0 + b_1x$, unde scopul este de a calcula optimul valorilor estimate pentru b_0 and b_1 astfel incat SSR sa fie minim.

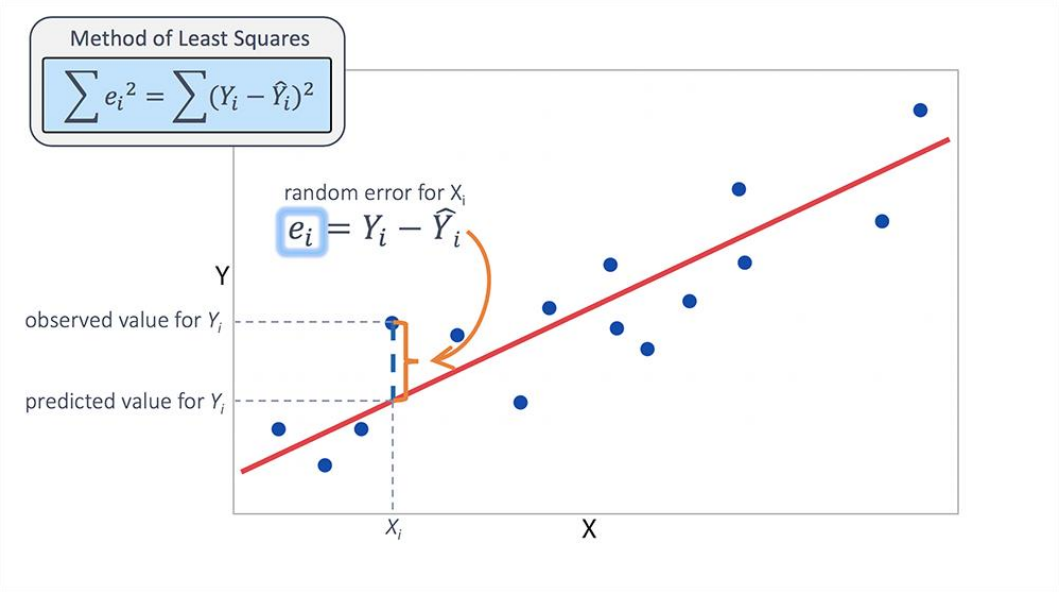

Fig.1 sursa: jmp.com

Regresia liniara multipla

Numita si regresie liniara multivariate, aceasta varianta a regresiei liniare are doua sau mai multe variabile independente. In acest caz forma generala va fi:
$(x_1, \ldots, x_r) = b_0 + b_1x_1 + \cdots + b_rx_r$, iar scopul acela de a estima ponderile $b_0, b_1, \ldots, br$.

Implementarea regresiei liniare in Python

Pentru a lucra cu implementarile in Python ale regresiilor avem nevoie de bibliotecile NumPy si scikit-learn

```
pip install --user scikit-learn
```

Nota. Pentru lucrul cu Google Colab nu este necesara instalarea bibliotecii, aceasta fiind implicit instalata.

Pasi in implementarea regresiei liniare:

1. Importarea pachetelor de lucru
2. Importarea datelor si eventual transformarea lor
3. Crearea unui model de regresie si optimzarea lui pentru datele disponibile
4. Verificarea rezultatelor pentru a testa performanta
5. Aplicarea modelului pentru predictii.

1. Importarea pachetelor de lucru

```
import numpy as np   #biblioteca numpy pt lucrul cu array
from sklearn.linear_model import LinearRegression #pt aplicarea regresiei
```

2. Importarea datelor

Se definesc cele doua variabile X – vector de date de intrare, Y- variabila tinta

```
x = np.array([5, 15, 25, 35, 45, 55]).reshape((-1, 1))
y = np.array([5, 20, 14, 32, 22, 38])
```

se apeleaza metoda *reshape* pentru a transforma array-ul X intr-o coloana.

Vizualizati forma lui X si Y in acest moment.

3. Crearea modelului de regresie.

Se va crea o instanta a clasei LinearRegression:

```
model = LinearRegression()
```

Aceasta accepta mai multi parametri, astfel:

fit_intercept este un Boolean (True by default) prin care se decide daca sa fie calculata eroarea aleatoare b_0 (True) sau sa fie considerate zero (False).

normalize este un Boolean (False by default) prin care se decide daca sa fie normalizate variabilele de intrare (True) sau nu (False).

copy_X este un Boolean (True by default) prin care se decide daca sa se copieze (True) sau sa se suprascrie variabilele de intrare (False).

n_jobs este un intreg sau None (default) si reprezinta numarul de joburi utilizate in procesarea paralela. None inseamna un job iar -1 utilizarea tuturor procesoarelor.

Pentru a crea modelul pe datele de intrare stabilite se apeleaza metoda fit():

```
model.fit(x, y)
```

Cu ajutorul fit() se calculeaza valorile optime ale ponderilor b_0 si b_1, folosind variabilele de intrare si iesire ca si argumente.

Construirea modelului poate fi realizata si direct:

```
model = LinearRegression().fit(x, y)
```

4. Verificarea rezultatelor

Odata ce modelul a fost creat, putem extrage rezultatele pentru a vedea daca acestea sunt satisfacatoare.

Coeficientul de determinare R^2 se obtine prin metoda `score()`:

```
r = model.score(x, y)
print('coeficientul de determinare R2:', r)

coeficientul de determinare R2: 0.715875613747954
```

Atributele modelului (model) sunt `.intercept_`, care reprezinta coeficientul b_0 si .coef_, care reprezinta b_1:

```
print('b0:', model.intercept_)
print('b1:', model.coef_)

b0: 5.633333333333329
b1: [0.54]
```

5. Rezultatul estimat

Modelul poate fi utilizat acum pentru realizarea de estimari pe datele existente sau pe altele noi.

Pentru aceasta utilizam metoda `predict()`.

```
y_pred = model.predict(x)
print('rezultatul estimat:', y_pred, sep='\n')

rezultatul estimat:
[ 8.33333333 13.73333333 19.13333333 24.53333333 29.93333333 35.33333333]
```

Mai multe informatii despre regresia liniara pot fi gasite pe site-ul oficial scikit-learn:
https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LinearRegression.html

Exercitii.

1. O metoda pentru a obtine rezultate similare este sa realizam estimarea folosind coeficientii `b0 si b1.` Scrieti codul aferent.

2. Realizati estimari dand alte valori pentru variabila x.

3. Realizati un grafic in care sa apara datele de intrare sub forma de puncte de coordonate(x,y) si modelul rezultat sub forma liniara(vezi Fig 1 de mai sus).

4. Folositi-va de urmatoarele date de intrare pentru a crea un model de regresie multipla, respectiv cu doua variabile de intrare. Acestea vor fi salvate intr-un array x cu doua coloane.

```
x = [[0, 1], [5, 1], [15, 2], [25, 5], [35, 11], [45, 15], [55, 34], [60, 35]]
y = [4, 5, 20, 14, 32, 22, 38, 43]
x, y = np.array(x), np.array(y)
```

5. Utilizand cunostintele acumulate, realizati un model de regresie multipla, in care datele sunt importate dintr-un fisier bidimensional m x n, in care ultima coloana este reprezentata de variabila Y, iar primele n-1 coloane vor fi salvate in variabila X.
   Dupa constuirea modelului, datele pentru noi estimari vor fi deasemenea importate dintr-un fisier, iar rezultatele vor fi scrise intr-un alt fisier de iesire.