Predmet: "Metode i primjena vještačke inteligencije"
Laboratorijska vježba 2: Upoznavanje sa klasifikacijom i regresijom
Autor: Doc. dr Amila Akagić

1 Cilj vježbe

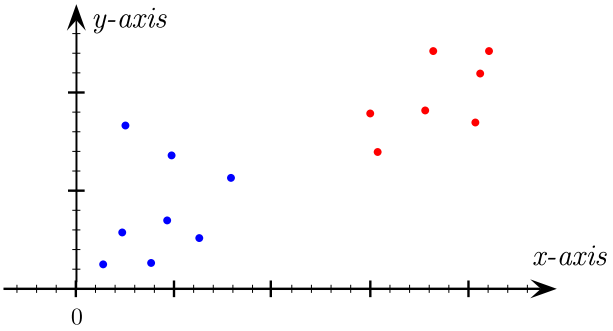
Upoznavanje sa osnovnim principima klasifikacije podataka i procesom regresije. Na predavanjima ste se upoznali sa dva najjednostavnija metoda za klasifikaciju: 1) Nearest Neighbor i 2) K-Nearest Neighbor ili KNN. U vježbi je potrebno upoznati se sa implementacijom ovih metoda i njihovom primjenom na jednostavnim primjerima, a na sljedećoj vježbi koristit će se složeniji skup podataka i naprednije metode klasifikacije.

2 KNN algoritam

KNN algoritam je jednostavan primjer algoritma superviziranog mašinskog učenja. U fazi treniranja, algoritam samo pamti podatke u memoriji, a da pri tome ne pokušava donijeti bilo kakvu pretpostavku o datim podacima (ovi podaci se nazivaju nezavisni podaci). U fazi predikcije pokušava se donijeti neki zaključak o nekom novom podatku (naziva se zavisna promjenjiva). Odnosno, bez obzira da li je riječ o klasifikaciji ili regresiji, algoritam prvo izračunava distancu između podataka korištenih u fazi treniranja i novog podatka. U samoj implementaciji algoritma distanca se vrlo jednostavno može i treba moći izabrati. Moguće je koristiti razne metode za izračunavanje distance, ali se u praksi najčešće koristi euklidska (*Euclidean*) ili menhetn (*Manhattan*) distanca.

Izračunate distance se zatim sortiraju od najmanje ka najvećoj, a potom se na osnovu pripadnosti K tačaka nekoj klasi, tj. K tačaka s prethodno izračunatom najmanjom udaljenošću u odnosu na novu tačku, vrši pridruživanje nove tačke odgovarajućoj klasi. Nakon što se izvrši pripajanje nove tačke odgovarajućoj klasi, algoritam završava sa izvršenjem.

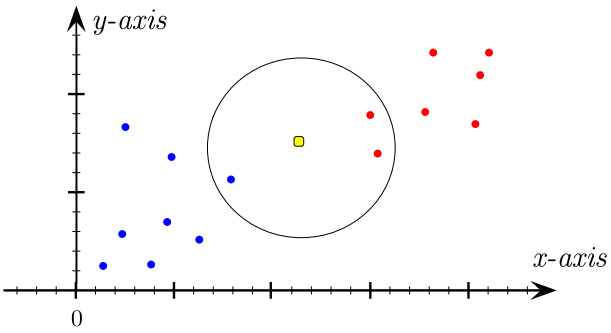
Algoritam će se demonstrirati na jednom skupu podataka s dvije promjenjive A i B. Grafički prikaz podataka dat je na slici 1. Podaci iz klase A označeni su plavom bojom, dok su podaci koji pripadaju klasi B označeni crvenom bojom.



Slika 1: Grafički prikaz hipotetskog dataset-a.

Ako se u prethodno naznačeni skup podataka nasumično doda nova tačka označena žutom bojom (slika 2.) za koju se prethodno ne zna kojoj klasi pripada, tada se postavlja pitanje pripadnosti date tačke odogovarajućoj klasi, tj. KNN algoritam treba da ustanovi kojoj klasi ova tačka pripada.

Ako se pretpostavi da je $K = 3$, onda se veoma brzo može doći do zaključka da su tri tačke sa najmanjom distancom u odnosu na žutu tačku one koje su obuhvaćene nacrtanom kružnicom. Konačno, u zadnjem koraku potrebno je odrediti kojoj klasi pripada žuta tačka. Na osnovu predhodno date slike može se vidjeti da dvije od tri tačke pripadaju B klasi, pa se iz toga može zaključiti da i nova tačka pripada B klasi.



Slika 2: Primjena KNN algoritma za određivanje pripadnosti žute tačke odogovarajućoj klasi skupa podataka kada je $K = 3$.

2.1 Implementacija KNN algoritma

U sljedećem listingu (ispisu koda) dat je dio implementacije KNN algoritma. Dijelovi za testiranje ovog algoritma ostavljeni su za samostalan rad u okviru predviđenih laboratorijskih vježbi.

```
from collections import Counter
import math

def knn(data, query, k, distance_func, choice_func):
    neighbor_distances_and_indices = []

    for index, example in enumerate(data):
        distance = distance_func(example[:-1], query)
        neighbor_distances_and_indices.append((distance, index))

    sorted_neighbor_distances_and_indices = sorted(neighbor_distances_and_indices)
    k_nearest_distances_and_indices = sorted_neighbor_distances_and_indices[:k]
    k_nearest_labels = [data[i][1] for distance, i in k_nearest_distances_and_indices]

    return k_nearest_distances_and_indices, choice_func(k_nearest_labels)

def mean(labels):
    return sum(labels) / len(labels)

def mode(labels):
    return Counter(labels).most_common(1)[0][0]

def neka_distance(point1, point2):
    sum_squared_distance = 0
    for i in range(len(point1)):
        sum_squared_distance += math.pow(point1[i] - point2[i], 2)
    return math.sqrt(sum_squared_distance)

def main():
    # Ovdje definisati podatke za klasifikaciju i regresiju po upustvima datim u vjezbi.
    # Definisati novu tacku i pozvati funkciju knn() za slucaj kada se biraju
    # razliciti parametri algoritma.

if __name__ == '__main__':
    main()
```

Obzirom da se KNN algoritam može koristiti za klasifikaciju i regresiju, za potrebe demonstracije koristiti će se veoma jednostavni primjeri u svrhu ilustracije rada algoritma. Svi koncepti iz ovih primjera mogu se primijeniti i na znatno većem skupu podataka.

U cilju testiranja algoritma definisat će se dva skupa podataka, tj. prvi koji će se koristiti za klasifikaciju i drugi skup podataka za regresiju. U slučaju klasifikacije skup podataka na izlazu daje diskretnu vrijednost. Na primjer, "voli jabuku u smoothie-ju" ili "ne voli jabuku u smoothie-ju", tj. osim ovih opcija ne postoji treća opcija. U tabeli 1. prikazani su nasumično generisani podaci koji se koriste za treniranje, a gdje se može vidjeti da je svakoj dobnoj skupini pridružen tačno jedan diskretan odgovor. U slučaju klasifikacije potrebno je za neki novi podatak napraviti predikciju na osnovu godina ispitanika (prediktor).

S druge strane, problem regresije na izlazu ima realni broj kao odgovor na zadati problem. Tako na primjer, jedan jednostavan problem koji se može pokušati realizirati u okviru ove vježbe je problem određivanja težine osobe na osnovu njene poznate visine. Podaci za ovaj problem definisani su u tabeli 2.

Godine	Voli jabuku u smoothie-ju
34	1
66	0
76	0
51	1
91	0
56	1
46	1
31	1
21	1
63	0
94	0
81	0
65	0
18	1
25	1

Tabela 1: Primjer podataka za problem klasifikacije. Podaci su generisani na nasumičan način. U prvoj koloni su prikazane godine, dok se u drugoj koloni nalazi pripadajuća labela (klasa).

Visina (cm)	Težina (kg)
159.2	64.32
189.1	96.92
148.8	51.58
201.8	103.49
191.1	150.22
169.2	85.54
153.2	45.99
170.3	69.43
199.5	200.43
188.7	97.7
165.5	87.2
134.6	80.7

Tabela 2: Primjer podataka za problem regresije. Podaci su generisani na nasumičan način. U prvoj koloni je prikazana visina, dok se u drujoj koloni nalaze pripadajuće težine osobe izražene u kilogramima.

Query/K	1	3	5
33			
50			
80			

Tabela 3: Za zadatak 2.1.10.

Query/K	1	3	5
150.4			
173.4			
190.3			

Tabela 4: Za zadatak 2.1.13.

Odgovorite na sljedeća pitanja ili dopunite kod po zadatim specifikacijama:

1. Objasnite koji parametri se šalju funkciji knn() na ulazu. O kojoj strukturi i vrsti podataka je riječ (skalar, vektor, int, float, itd.)?
2. Objasnite šta funkcija knn() vraća na izlazu?
3. Kako se izračunava distanca u datom primjeru? Identifikujte o kojoj distanci je riječ (na osnovu informacija datim na predavanjima).
4. Koja struktura se koristi za `neighbor_distances_and_indices`?
5. Kako su podaci organizovani u strukturi `neighbor_distances_and_indices`?
6. Koliko elemenata će imati struktura `neighbor_distances_and_indices`?
7. Koji algoritam se koristi za sortiranje elemenata unutar ove strukture?
8. Koja vrijednost će biti zapisana u varijablu `k_nearest_distances_and_indices`?
9. Definisati strukturu podataka `classification_data`, a potom u nju spremiti podatke iz tabele 1. Svaki red u ovoj strukturi će imati dva elementa, na primjer: [34, 1], [66, 0], itd.
10. Sekvencijalno definisati sljedeće tri vrijednosti: 1) 33, 2) 50 i 3) 80 u varijablu `classification_query`, a potom sekvencijalno izvršiti testiranje algoritama na izlazu. Za sva tri slučaja, definisati da je K: a) 1, b) 3 i c) 5, kao što je prikazano u tabeli 3.
11. Za jedan od prethodnih slučaja (npr. Query=33 i K=3) matematičkim putem pokazati kako se izračunava izlazna vrijednost.
12. Definisati strukturu podataka `regression_data` i u nju spremiti podatke iz tabele 2.
13. Sekvencijalno definisati sljedeće tri 1) 150.4, 2)173.4 i 3) 190.3 vrijednosti u varijablu `regression_query`, a potom sekvencijalno testirati šta će algoritam dati na izlazu. Za sva tri slučaja definisati da je K: a) 1, b) 3 i c) 5, kao što je prikazano u tabeli 4.
14. Za jedan od prethodnih slučaja (npr. Query=150.4 i K=3) matematičkim putem pokazati kako se izračunava izlazna vrijednost.
15. Analizirati šta se dešava za slučaj izlaznih vrijednosti KNN algoritma kada se izvrše sva testiranja nad prethodno spomenutim skupovima podataka s drugom metrikom za izračunavanje udaljenosti između podataka. Da li je algoritam i u ovom slučaju vratio iste rezultate kao i za prethodni slučaj uzete metrike (distance) za izračunavanje udaljenosti? Šta se dešava s rezultatima algoritma ako se upotrijebi neka treća metrika za izračunavanje udaljenosti između tačaka? (Napomena: Za svaku uzetu metriku za izračunavanje udaljenosti potrebno je izvršiti sva potrebna testiranja).
16. Uporedite testove u odnosu na dosad izmjerene vrijednosti. Kakve zaključke možete izvesti? Da li algoritam u svim slučajevima vratio iste rezultate? Zašto?

Napomena: Format laboratorijskih vježbi na predmetu MPVI ne nalaže korištenje samo jednog alata za izradu vježbe. Možete koristiti običan tekstualni editor, ili neki Python IDE ili Jupyter Notebook. Student može izabrati bilo koji način izrade vježbe i izvještaja.