K-artimiausiųjų kaimynų algoritmas

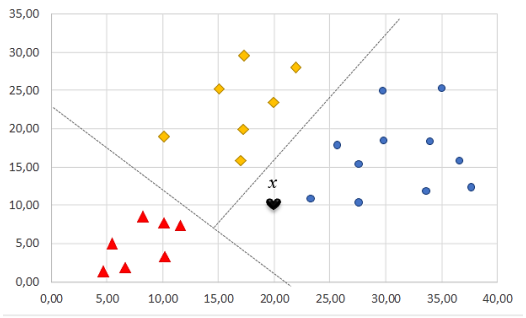
1 Kas yra K-artimiausiųjų kaimynų algoritmas ir kaip jis veikia

Panašumu grįstas mokymasis – duomenų klasifikavimas pagal tam tikrų savybių (kitaip atributų) panašumus. Patys paprasčiausi ir geriausi žinomi šio mašininio mokymosi tipo atstovai yra *K*-artimiausiųjų kaimynų ir *K*-vidurkių algoritmai. *K*-artimiausiųjų kaimynų algoritmas (KNN) yra prižiūrimojo tipo algoritmas, o *K*-vidurkių metodas – neprižiūrimojo tipo algoritmas.

KNN gali būti naudojamas tiek klasifikavimo, tiek regresinėms prognozavimo problemoms spręsti. Tačiau dažniausiai šis algoritmas naudojamas prognozuojant klasifikavimo problemas pramonėje. KNN yra tingusis mokymosi algoritmas, nes jis neturi mokymosi etapo, o tiesiog klasifikuoja naudodamas visus duomenis. Tai neparametrinis mokymosi algoritmas, todėl modelio parametrai auga didėjant duomenų kiekiui. Kiekvienas duomenų pavyzdys gali būti traktuojamas kaip parametras, kuris gali būti naudojamas klasifikuojant. Parametrinis modelis turi fiksuotą skaičių parametrų, o KNN parametrų skaičius yra begalinis. Naivusis Bajesas, dirbtiniai neuroniniai tinklai yra parametriniai algoritmai, o štai KNN ir sprendimų medžiai neparametriniai.

1 pav. *Duomenų – karoliukų pavyzdys.*

Paimkime pavyzdį, kuriame turime daug įvairiausių karoliukų (1 pav.). Tarkime, visame šitame karoliukų rinkinyje atsirado naujas elementas – širdelės formos karoliukas. Klausimas, kuriai iš trijų klasių jį priskirti? Gyvenime žmonės tokius dalykus daro intuityviai, galbūt pagal metalą arba pagal panaudojimo tikslą, bet skaitmeniniam problemos aprašui reikia skaitinių įverčių, o ne verbalinių reikšmių. Tarkime, nusprendėm visus karoliukus surūšiuoti į tris klases remdamiesi dviem panašumo savybėmis, kurios turi skaitines reikšmes. Tai gali būti svoris, ilgis ir pan. Uždavinys išlieka tas pats – prie kurių priskirti naują karoliuką, kuris diagramoje atvaizduojamas kaip taškas *x* (2 pav.).



2 pav. *Duomenų suklasifikavimo į tris klases pavyzdys.*

K-artimiausiųjų kaimynų algoritmas remiasi savybių panašumu, vadinasi, naujam x duomenų taškui bus priskirta klasė atsižvelgiant į tai, kaip tiksliai jis atitinka duomenų imties taškus arba į ką jis labiausiai panašus remiantis pateiktomis skaitinėmis savybėmis.

2 *K*-artimiausiųjų kaimynų algoritmo žingsniai

KNN algoritmas susideda iš tokių žingsnių:

1 žingsnis – duomenų rinkinio sudarymas. Vadinasi, jeigu mes turime netinkamą duomenų rinkinį (pvz., nuotraukas, verbalines reikšmes), jį reikia apdoroti ir paruošti naudoti;

2 žingsnis – *K* reikšmės parinkimas, t. y., artimiausių taškų (kaimynų) skaičius, kur *K* gali būti bet koks sveikasis skaičius;

3 žingsnis – apskaičiuoti atstumą tarp taško *x* ir visų kitų duomenų imties taškų (Euklido, Manhatano ar Hammingo atstumą). Dažniausiai naudojamas atstumo apskaičiavimo metodas yra Euklido;

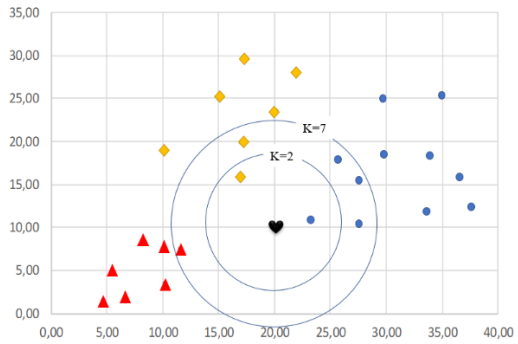
4 žingsnis – surūšiuoti gautus atstumus didėjančia tvarka;

5 žingsnis – surūšiuotam sąraše parinkti *K* elementų nuo viršaus;

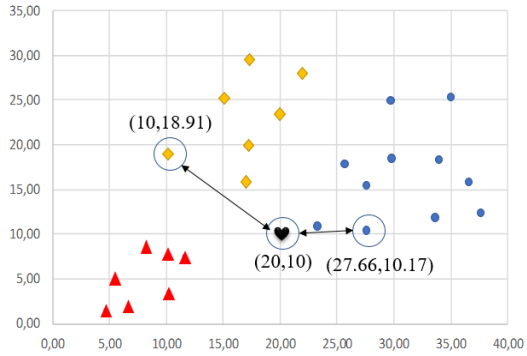
6 žingsnis – taškui *x* priskirti tą klasę, kuri atrinktame sąraše iš *K* elementų kartojasi daugiausiai kartų.

Pirmame žingsnyje reikia sudaryti duomenų rinkinio savybių sąrašą skaitine išraiška, nes neturint skaičių nebus galima paskaičiuoti panašumo tarp duomenų rinkinio elementų. Skaitinės savybių išraiškos gali būti labai įvairios: elemento svoris, ilgis, plotis, skonio ar kvapo stiprumas, kaina ir pan. Tai yra viskas, kam galima suteikti skaitinę reikšmę. Tuomet yra atrenkamos pačios reikšmingiausios savybės ir jų skaitinės reikšmės dėl patogumo atvaizduojamos grafiškai. Tačiau savybės, kurios turi žemą kardinalumą (kardinalumas – galimų skirtingų reikšmių skaičius), nėra tinkamos KNN algoritmui. Tarkime, tokios savybės (atributai) kaip lytis (kardinalumas yra du), spalva, Taip / Ne atsakymai yra tokio netinkamo tipo pavyzdžiai.

Antrame žingsnyje yra parenkama *K* reikšmė, kuri parodo, kiek artimiausių taškų reikės surasti nurodytam taškui. Tarkime, mūsų pasirinktas *K* skaičius taškui *x* yra keturi, vadinasi, ieškosime keturių artimiausių kaimynų. Grafiškai atvaizduojant *K* artimiausių kaimynų skaičių kartais iliustracijai naudojami apskritimai (3 pav.), kartais atvaizduojama rodyklėmis ir pan. Bet kuriuo atveju, tiksliausias įvertinimas remiasi atstumo skaičiavimais išmatuojant atstumą nuo naujojo taško, kurį reikia suklasifikuoti iki gretimų taškų.



3 pav. *K-artimiausiųjų kaimynų pavyzdys, kur K = 2, K = 7*

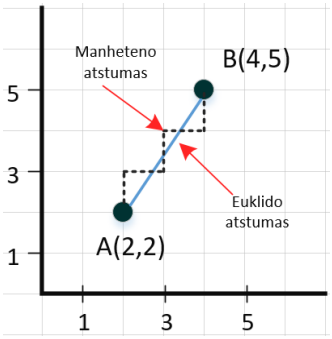


4 pav. Skirtingų taškų koordinatės

Todėl trečiajame žingsnyje reikia paskaičiuoti atstumą tarp taško *x* ir visų kitų duomenų imties taškų. Dažnai skaičiuojamas Euklido arba Manheteno atstumas (5 pav.).

Euklido atstumo formulė: $\sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$.

Manheteno atstumo formulė: $|x_2 - x_1| + |y_2 - y_1|$.



5 pav. Euklido atstumas tarp taškų A ir B yra mėlyna tiesė, o Manheteno atstumas laiptuota punktyrinė linija.

Paveiksle 5 pav. pateiktam pavyzdžiui Euklido atstumas lygus $\sqrt{(4-2)^2 + (5-2)^2}$ = 3,601, o Manheteno atstumas lygus $|4-2| + |5-2| = 5$.

Tarkime, mes skaičiuosime Euklido atstumą.

$$\sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}\ .$$

Tarkime, paskaičiuosime atstumą nuo taško x (x1 = 20 ir y1 = 10) iki geltonojo taško (rombo) su koordinatėmis x2 = 10 ir y2 = 18,91:

$$\sqrt{(10 - 20)^2 + (18{,}91 - 10)^2} = 13{,}39.$$

Euklido atstumas iki mėlynojo taško su koordinatėmis x2 = 27,66 ir y2 = 10,17 yra:

$$\sqrt{(27{,}66 - 20)^2 + (10{,}17 - 10)^2} = 7{,}66.$$

Akivaizdžiai matoma, kad mėlynas taškas yra arčiau taško *x*. Tai leidžia pagrįsti ir Euklido atstumo reikšmė, kuri yra perpus mažesnė nei iki geltonojo taško. Taigi akivaizdu, kad ieškosime keturių taškų (nes *K* = 4) su mažiausiu Euklido atstumu.

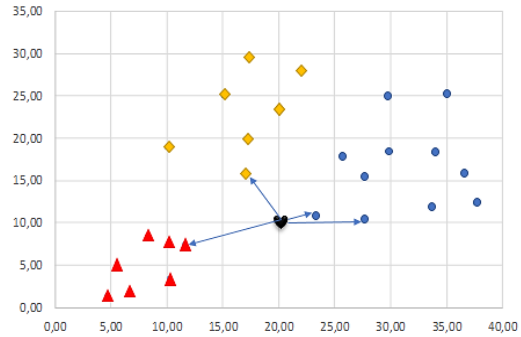
Apskaičiuotus Euklido atstumus tarp taško x ir visų kitų duomenų imties taškų reikia surašyti į vieną masyvą.

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25
13,44	17,53	11,87	15,80	11,77	8,80	10,15	15,34	13,13	17,70	17,52	3,45	16,14	7,66	13,90	9,76	15,68	6,31	19,69	17,84	18,12	21,31	9,28	13,39	10,33

Sekančiame, ketvirtajame, žingsnyje šie skaičiai yra surūšiuojami didėjančia tvarka.

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25
3,45	6,31	7,66	8,80	9,28	9,76	10,15	10,33	11,77	11,87	13,13	13,39	13,44	13,90	15,34	15,68	15,80	16,14	17,52	17,53	17,70	17,84	18,12	19,69	21,31

Penktajame žingsnyje parenkama *K* elementų nuo viršaus. Kadangi *K* yra lygu 4 šiuo atveju, tai parenkami keturi aukščiausiai esantys atstumai. Pagal koordinates yra du mėlyni, vienas geltonas ir vienas raudonas taškai (6 pav.).



6 pav. *Taško x keturi artimiausi kaimyniniai taškai*

Galiausiai taškui *x* priskiriama klasė, kuri atrinktame sąraše kartojasi daugiausiai kartų. Pagal gautus rezultatus naujam taškui x bus priskirta mėlynų skrituliukų klasė. Patarimas renkantis K skaičių yra naudotis formule: $K = n * (sk + 1)$, kur *n* yra daugiklis (1, 2, 3 ir t. t.), *sk* – klasių skaičius. Tačiau net ir naudojantis tokia formule galime gauti vienodą tam tikrų klasių skaičių. Tarkime, *K* = 7, iš viso yra trys klasės, *sk* = 3 ir atsakymas gali gautis su 3 kaimynais iš 1-osios klasės, 3 kaimynais iš 2-osios ir 1 iš 3-osios. Paties *K* skaičiaus ribos dažniausiai varijuoja tarp 3–10. Korektiškam ir patikimam KNN algoritmo atsakymui reikia atlikti kryžminę validaciją. Tačiau, jeigu ir po to gauname vienodą tam tikrų klasių kaimynų skaičių, tada viena iš šių klasių x taškui priskiriama atsitiktinai.