**OʻZBEKISTON RESPUBLIKASI OLIY TA’LIM, FAN VA** **INNOVATSIYALAR VAZIRLIGI**

**MIRZO ULUGʻBEK NOMIDAGI OʻZBEKISTON MILLIY UNIVERSITETINING JIZZAX FILIALI**

Kompyuter ilmlari va dasturlash texnalogiyalari kafedrasi

Sun’iy intellekt va neyron to‘rli tenologiyalar fanidan

**MUSTAQIL ISH**

**Mavzu:** **Sinflashning metrik algoritmlari.**

**Guruh:** 461-22 guruh talabasi

**Bajardi:** Nurmamatov Mirzabek

**Tekshirdi:** Ergashev Sirojiddin

**Mavzu:Sinflashning metrik algoritmlari.**

**Reja:**

1. Yaqin qo‘shni usuli va uni umumlashtirish.

2. k-yaqin qo‘shni algoritmi.

3. Parzen darchasi usuli.

4. Yadro funktsiyasi.

**Sinflashning metrik algoritmlari**

Sinflashning metrik algoritmlari, gunohlarning boshqa gunohlardan qanday o'xshash yoki farqli bo'lishini o'lchaydigan metriklarni ifodalaydi. Uch metriklar sinflarning o'xshashligini, aniqlanish va sinflar masofani o'lchaydi. Hisoblash metriklar sinflashning o'lchov algoritmlaridan bir necha misoldir:

1.Silhouette koeffitsienti :Barcha obrazlarni sinflash va sinflar o'rtasidagi masofani o'lchaydi.

2. Purity : Sinflarning qancha to'g'ri aniqlanganligini o'rganish uchun foydalaniladi.

3.Pair F1-Score : Sinflash dasturlarni o'rganish va sinflarni aniqlash uchun.

4.To'g'ri javob (True Positives - TP) : Sinflashning to'g'ri buni qanday to'g'ri aniqlaganligini ko'rsatadi.

5.Yolg'on javob (False Positives - FP) :Sinflashning noto'g'ri rejasini qanday to'g'ri aniqlaganligini ko'rish.

6.To'g'ri negativ javob (True Negatives - TN) :Sinflashning to'g'ri aniqlagan noto'g'ri sabablarini qanday to'g'ri aniqlaganligini ko'rish.

7.Yolg'on javob (False Negatives - FN) : Sinfning to'g'ri nima ekanligini qanday noto'g'ri aniqlaganligini ko'rsatadi.

8.Ishonchlilik (Accuracy) : Ishonchlilik metri sinflashning to'g'ri aniqlagan rasmlar soni va to'plam rasmlar soni nisbatni ifodalaydi.

9.Tugri aniqlangan qat'iylik (Precision) : Tugri aniqlangan rasmlar soni va to'g'ri aniqlangan rasmlar soni nisbatni ko'rinishi

10.Eslatib o'tamiz (sensitivlik) : Noto'g'ri aniqlangan rasmlar soni va to'g'ri aniqlangan rasmlar soniga nisbatan nisbatni ko'rsatadi.

11.F-measure (F-skor) : Preciziyani va sensitiqlikni bir baholashgan umumiy metrik, hisobot o'rtacha qiymatni hisoblaydi.

12.ROC e'gri (Receiver Operating Characteristic curve) :Tugri aniqlangan va yolg'on aniqlanganlarning nisbatini qo'llaydi.

13.AUC (Ard the ROC Curve) :ROC e'gri ostidagi quvvati. AUC o'rtacha sinflashning ishonchlilik yordamida ifodalaydi.

14.Jaccard indeksi (Jaccard Index) :Yer koordinatalarni aniqlashda olinadi. Yangi natija va sinf to'plamining ko'rishini tahlil qilishda ko'rinish.

15.Mean Absol Error (MAE) :Regressiyai uchun qo' tekshiruvi va sinfi jamlashning to'g'ridan-to'g'ri sinashga aniqlangan videoning o'rtacha uzluksizdagi qaydni ifodalaydi.

16.Rand Index :Sinflash va to'g'ri noto'g'ri birligini, o'rganish va farqli bo'lishini aniqlashda aniqlanadi.

17.Adjusted Rand Index (ARI) :Rand Indexni sinflarning darajasiga o'xshash va to'g'ri chiqishni boshqarishga ta'lim oladi.

18.Normalized Mutual Information (NMI) :Sinflash tekshirishning o'zlashtirilgan entropiyasi va gunohlarni aniqlash orqali o'zlashtirilgan ma'lumotlarni o'z ichiga oladi.

19.Fowlk-Mallows Index (FMI) : Rand Index.

20.Jakkard koeffitsienti (Tanimoto koeffitsienti) :Sinflash natija bo'ylab barcha gunohlarni hisoblash va to'plamlarni solishtirishda aniqlanadi.

21.Davies-Bouldin Index :Sinflash dasturlarini to'g'ri, noto'g'ri, o'xshash va farqli turlardagi sinflarga taqsimlanganlikni o'lchaydi.

22.Dunn Index :Sinflash tarkibi tuzilishini va to'plamlarni solishtirishda foydalaniladi

23.V-Measure :Entropiya va homogenlikning birlashtirilgan ko'rsatkichi, sinflash usullarini o'rganish va gunohlarni aniqlashda qo'shimcha

24.Bu metriklar sinflashning yordamini o'lchash va tahlil qilishda muhim ega. Ular sinflarning o'xshashligini, aniqlanish o'xshashligini va sinflar masofasiga, shuning uchun sinflashni optimallashtirish va ishlab chiqarishda juda foydali bo'ladi.

**Yaqin qo‘shni usuli va uni umumlashtirish.**

Yaqin qo'shni usuli va uni umumlashtirish, yaqin qo'shni aniqlash va uni boshqa sinflar bilan solishtirish usullarini o'z ichiga oladi. Yaqin qo'shni aniqlash uchun bir necha algoritmlar, lekin eng ko'p kuchlaridan biri K-Nearest Neighbors (K-NN) algoritmi.

K-NN algoritmi, biror obyektning yaqin qo'shini aniqlash uchun eng yaqin qo'shlar soni(K) o'lchaydi va sudlov eng ko'p sifatini tanlaydi.

**K-NN algoritmi qadamlarni o'z ichiga oladi:**

Ma'lumotlar ma'lumotlarini tayyorlash : Sizning ma' bazangizda, har bir obyektning koordinatalarini va sinfini bilish kerak.

Yangi obyektni tanlash : Yangi obyektni tanlash uchun, u obyektning koordinatalarini ma'lumotlar bazasiga qo'shish.

Yaqin qo'shlarni topish : Yangi obyektning koordinatalarini ma'lumotlar bazasida topish va eng yaqin K qo'shlarni topish.

Sinfni xususiyatlari : Topilgan yaqin qo'shlarning sinflari bo'yicha eng ko'p moddiy sinfni yangi obyektning sinfi sifatida belgilangan.

K-NN algoritmi umumiylashtirish uchun, uni optimallashtirish va boshqa algoritmlar bilan solishtirish mumkin.

Bu optimallashtirishlar o'z ichiga oladi:

K-ni belgilash : K-NN algoritmi uchun eng muhim parametrlardan biri K-ni belgilashdir.

K-ni hisoblash uchun, ma'lumotlar zahirasining hajmi, gunohlarning soni va gunohlar masofa ko'zga zararsiz.

Masofa funksiyasi : Obyektlarga masofani masofa uchun turli xil masofa funksiyalari mavjud. Masalan, Evklid masofasi (chiziqsizlik) va Manhattan masofasi (shahar masofasi) kabi.

Qo'shlarni tuzatish usuli : Yaqin qo'shlarni topish uchun turli xil usullar mavjud. Misol uchun, eng qo'shnilarni topish (Nearest Neighbors) yoki eng yaqin qo'shnilarni topish va qo'shnilar sinflarini yaqin' (Nearest Neighbors with Weighted Voting).

Dinamik ma'lumotlar bazasi : Ma'lumotlar bazasining dinamik yangilanishi mumkin.

K-NN algoritmi, yaqin qo'shni aniqlash va uni boshqa sinflar bilan solishtirishda juda samarali. Ush algoritmni optimallashtirish va boshqa algoritmlar bilan solishtirish, yaqin qo'shni aniqlashda muhim korxona ega.

**k-yaqin qo‘shni algoritmi**

K-Yaqin qo'shnilar (KNN) algoritmi tasniflash va regressiya vazifalari uchun ishlatiladigan mashhur mashinani o'rganish usulidir. Bu o'xshash ma'lumotlar nuqtalari odatda o'xshash teglar yoki qiymatlarga ega degan fikrga tayanadi.

Trening bosqichida KNN algoritmi barcha o'quv ma'lumotlar to'plamini ma'lumotnoma sifatida saqlaydi. Bashorat qilishda u Evklid masofasi kabi tanlangan masofa ko'rsatkichidan foydalanib, kiritilgan ma'lumotlar nuqtasi va barcha o'qitish misollari orasidagi masofani hisoblab chiqadi.

Keyinchalik, algoritm kirish ma'lumotlar nuqtasiga eng yaqin K qo'shnilarini ularning masofalariga qarab aniqlaydi. Tasniflash holatida algoritm K qo'shnilari orasida eng keng tarqalgan sinf yorlig'ini kirish ma'lumotlar nuqtasi uchun bashorat qilingan yorliq sifatida belgilaydi. Regressiya uchun u kirish ma'lumotlar nuqtasi uchun qiymatni taxmin qilish uchun K qo'shnilarining maqsadli qiymatlarining o'rtacha yoki og'irlikdagi o'rtacha qiymatini hisoblab chiqadi.

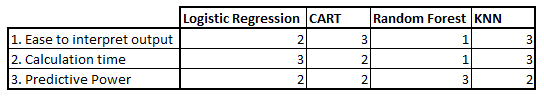
KNN algoritmi sodda va tushunarli bo'lib, uni turli sohalarda mashhur tanlovga aylantiradi. Biroq, uning ishlashiga K ni tanlash va masofa ko'rsatkichi ta'sir qilishi mumkin, shuning uchun optimal natijalar uchun parametrlarni ehtiyotkorlik bilan sozlash kerak. KNN algoritmi ham tasniflash, ham regressiyani bashorat qilish muammolari uchun ishlatilishi mumkin. Biroq, u sanoatda tasniflash muammolarida kengroq qo'llaniladi. Har qanday texnikani baholash uchun biz odatda uchta muhim jihatni ko'rib chiqamiz:

1. Chiqishni talqin qilish qulayligi

2. Hisoblash vaqti

3. Bashoratli kuch

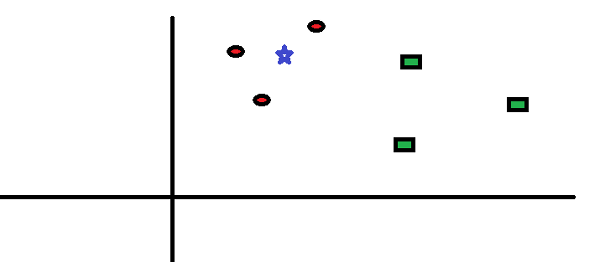
Keling, KNNni o'lchovga joylashtirish uchun bir nechta misollarni olaylik:

[](https://www.analyticsvidhya.com/wp-content/uploads/2014/10/Model-comparison.png)

*1-rasm*

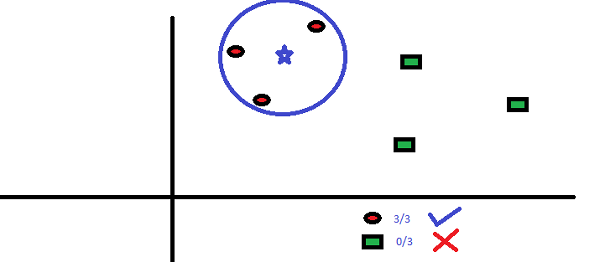
Ko'rib chiqishning barcha parametrlari bo'yicha KNN tasniflagichi yarmarkalari. U odatda talqin qilish qulayligi va hisoblash vaqtining pastligi uchun ishlatiladi.

Bu algoritmni tushunish uchun oddiy bir holatni olaylik. Quyida qizil doiralar (RC) va yashil kvadratlar (GS) yoyilgan:



*2-rasm*

Siz ko'k yulduz (BS) sinfini bilmoqchisiz. BS RC yoki GS bo'lishi mumkin va boshqa hech narsa emas. KNN algoritmidagi "K" biz ovoz berishni istagan eng yaqin qo'shnimiz. Aytaylik, K = 3. Demak, biz endi tekislikda faqat uchta ma'lumot nuqtasini o'rab oladigan darajada BS markazi bo'lgan doira yasaymiz. Batafsil ma'lumot uchun quyidagi diagrammaga qarang:

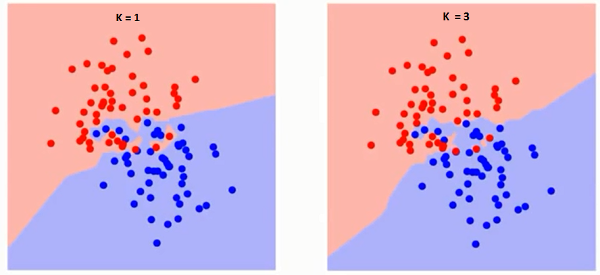
[](https://www.analyticsvidhya.com/wp-content/uploads/2014/10/scenario2.png)

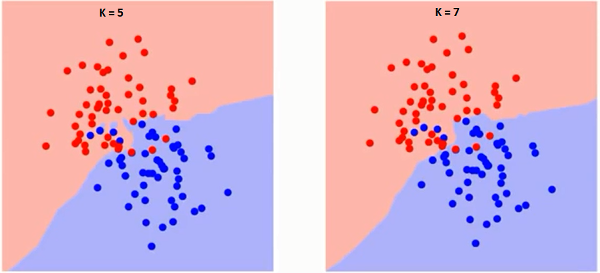
*3-rasm*

BS ga eng yaqin uchta nuqta RC hisoblanadi. Demak, yaxshi ishonch darajasi bilan biz BS RC sinfiga tegishli bo'lishi kerakligini aytishimiz mumkin. Bu erda tanlov ayon bo'ldi, chunki eng yaqin qo'shnining uchta ovozi RCga o'tdi. Ushbu algoritmda K parametrini tanlash juda muhim. Keyinchalik, eng yaxshi K ni xulosa qilish uchun hisobga olinadigan omillarni tushunamiz.

**K omilini qanday tanlaymiz?**

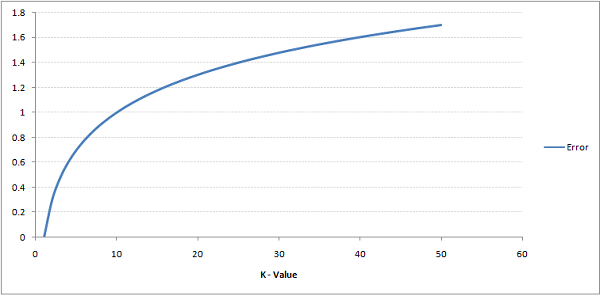
Birinchidan, algoritmdagi K-eng yaqin qo'shnilarning (KNN) ta'sirini tushunishga harakat qilaylik. Agar oxirgi misolni ko'rib chiqsak, barcha 6 ta o'quv kuzatuvlarini doimiy ravishda ushlab tursak, berilgan K qiymati har bir sinf uchun chegaralarni belgilashga imkon beradi. Ushbu qaror chegaralari, masalan, RCni GSdan samarali ravishda ajratib turadi. Xuddi shunday, "K" qiymatining ushbu sinf chegaralariga ta'sirini ko'rib chiqamiz. Quyida har biri K ning turli qiymatlariga mos keladigan ikkita sinfni ajratib turuvchi aniq chegaralar tasvirlangan.

[](https://www.analyticsvidhya.com/wp-content/uploads/2014/10/K-judgement.png)

[](https://www.analyticsvidhya.com/wp-content/uploads/2014/10/K-judgement2.png)

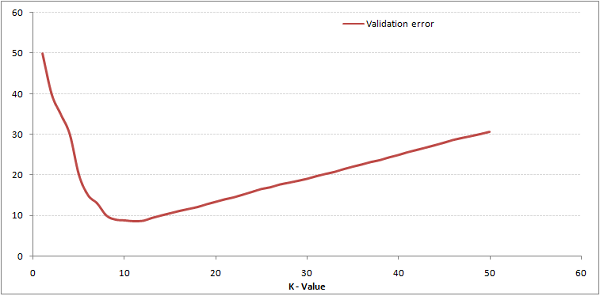
*4-rasm*

Ehtiyotkorlik bilan kuzatsangiz, chegara K qiymati ortib borishi bilan silliqroq bo'lishini ko'rishingiz mumkin. K ning cheksizgacha o'sishi bilan u umumiy ko'pchilikka qarab butunlay ko'k yoki butunlay qizil rangga aylanadi. Trening xatosi darajasi va tekshirish xatosi darajasi turli K-qiymatiga kirishimiz kerak bo'lgan ikkita parametrdir. Quyida o'zgaruvchan K qiymatiga ega bo'lgan mashq xatosi darajasining egri chizig'i keltirilgan:



*5-rasm*

Ko'rib turganingizdek, K=1 da xato darajasi o'quv namunasi uchun har doim nolga teng. Buning sababi shundaki, har qanday o'quv ma'lumot nuqtasiga eng yaqin nuqta o'zidir. Demak, K=1 bilan bashorat har doim to'g'ri bo'ladi. Agar tekshirish xatosi egri chizig'i o'xshash bo'lganida, K ni tanlashimiz 1 bo'lar edi. Quyida o'zgaruvchan K qiymatiga ega bo'lgan tekshirish xatosi egri chizig'i keltirilgan:

[](https://www.analyticsvidhya.com/wp-content/uploads/2014/10/training-error_11.png)

*6-rasm*

Bu hikoyani yanada aniqroq qiladi. K = 1 da biz chegaralarni haddan tashqari oshirib yubordik. Shunday qilib, xato darajasi dastlab pasayadi va minimal darajaga etadi. Minimal nuqtadan so'ng u K ning ortishi bilan ortadi. K ning optimal qiymatini olish uchun siz boshlang'ich ma'lumotlar to'plamidan trening va tekshirishni ajratishingiz mumkin. Endi K ning optimal qiymatini olish uchun tekshirish xatosi egri chizig‘ini chizing. K ning bu qiymati barcha bashoratlar uchun ishlatilishi kerak.

Yuqoridagi tarkibni bizning bepul kursimiz - [Python va R-da K-Yaqin qo'shnilar (KNN) algoritmi yordamida yanada intuitiv tushunish mumkin.](https://courses.analyticsvidhya.com/courses/K-Nearest-Neighbors-KNN-Algorithm?utm_source=blog&utm_medium=knn_in_python&R)

Uni buzish - KNN psevdokodi

Biz KNN modelini quyidagi amallarni bajarish orqali amalga oshirishimiz mumkin:

Ma'lumotlarni yuklang

k qiymatini boshlang

Bashorat qilingan sinfni olish uchun 1 dan o'quv ma'lumotlarining umumiy soniga qadar takrorlang

Sinov ma'lumotlari va o'quv ma'lumotlar to'plamining har bir qatori orasidagi masofani hisoblang. Bu erda biz Evklid masofasidan masofa ko'rsatkichi sifatida foydalanamiz, chunki bu eng mashhur usul. Foydalanish mumkin bo'lgan boshqa masofa funksiyasi yoki ko'rsatkichlari: Manxetten masofasi, Minkovski masofasi, Chebishev, kosinus va boshqalar. Agar kategorik o'zgaruvchilar mavjud bo'lsa, hamming masofasidan foydalanish mumkin.

Hisoblangan masofalarni masofa qiymatlari asosida o'sish tartibida tartiblang

Saralangan massivdan eng yuqori k qatorni oling

Ushbu qatorlarning eng tez-tez uchraydigan sinfini oling

Bashorat qilingan sinfni qaytaring

KNN algoritmi eng oddiy tasniflash algoritmlaridan biridir. Bunday soddaligi bilan ham u yuqori raqobatbardosh natijalar berishi mumkin. KNN algoritmidan regressiya muammolari uchun ham foydalanish mumkin. Muhokama qilingan metodologiyadan yagona farq k-eng yaqin qo'shnilardan ovoz berish o'rniga eng yaqin qo'shnilarning o'rtacha ko'rsatkichlaridan foydalanish bo'ladi. KNN R da bitta satrda kodlanishi mumkin. Men SAS da KNN algoritmidan qanday foydalanishimiz mumkinligini hali o'rganmayapman.

**Parzen darchasi usuli**

Parzen darchasi (Parzen Window usuli), yoki Parzen-Rosenblatt darchasi, ma'lumotlar bazasidan tushunarli bo'lmaydigan tuzilishlar uchun ehtimollik yo'qotish funksiyasini qo'shish uchun keng qo'llash usulisiz usulidan o'rnatish. Ushbu usul, ma'lumotlar bazasining tuzilishini aniqlash uchun eng yaqin qo'shimchalarni topish va sudlovlarni tekshirish natijasida natija beradi.

Parzen usuli qadamlarni o'z ichiga oladi:

Darcha o'lchamini tanlash : Darcha o'lchamini juda ko'p narsa, chunki ma'lumotlarning dasturiy ta'minoti uchun bu o'lchamni ishlatamiz. Darcha o'lchamini hisobga olish uchun, ma'lumotlar bazasining hajmi, gunohlarning soni va sinflar masofasi ko'zga tushadi.

Yadro funksiyasini tanlash : Yadro funksiyasi, yaqin qo'shlarni topish uchun. Eng ko'p qobiliyat yadro funktsiyalari giperbe va Gauss yadrosi. Yadro funksiyasini tuzatish uchun, ma'lumotlar bazasining tuzilishini aniqlash uchun kerak bo'lgan natijani topish uchun turli xil usullar.

Yangi qo'shni tanlash : Yangi qo'shni tanlash uchun, u ob'ektning koordinatalarini ma'lumotlar bazasida saqlash va eng yaqin qo'shnilarni topish.

Natija Berish : Topilgan yaqin qo'shlarning sinflari bo'yicha eng ko'p qobiliyatini yangi obyektning sinfi sifatida ajratib turadi.

Parzen darchasi usuli, yaqin qo'shni aniqlash va uni boshqa sinflar bilan solishtirishda juda samarali. Ush algoritmni optimallashtirish va boshqa algoritmlar bilan solishtirish, yaqin qo'shni aniqlashda muhim korxona ega.

Parzen darchasi usulining afzalligi, u tuzilishni aniqlash uchun ma'lumotlar bazasining tuzilishini nazorat qilmaydi. Bu usul, ma'lumotlar bazasining tuzilishini aniqda juda samarali bo'lib ammo darcha o'lcham va yadro funksiyasini juda mos keladi. Darcha o'lchamini hisobga olish uchun, ma'lumotlar bazasining hajmi, gunohlarning soni va sinflar masofasi ko'zga tushadi. Yadro funksiyasini tuzatish uchun, ma'lumotlar bazasining tuzilishini aniqlash uchun kerak bo'lgan natijani topish uchun turli xil usullar

# **Yadro funktsiyasi nima?**

[**Yadrolar haqida gapirish uchun biz SVM**](https://en.wikipedia.org/wiki/Support_vector_machine) (vektorli mashinalarni qo'llab-quvvatlash) - **tasniflar** - **nazorat ostida o'rganish** - **mashinani o'rganish** - blah blah ... kabi atamalarni tushunishimiz kerak . Juda ko'p atamalar to'g'rimi?, lekin bu sizni xafa qilishiga yo'l qo'ymang (men DIY mashqidan oldin bularning barchasi haqida hech narsa bilmasdim). Keling, birga yuramiz: -

Xo'sh, "mashinalarni o'rganish (ML)" nima? Ma'lum bo'lishicha, ML aslida judako'p narsa, ammo asosiy mavzu 1959 yilda Artur Samuel tomonidan aytilgan tez-tez keltiriladigan bayonot bilan jamlangan:

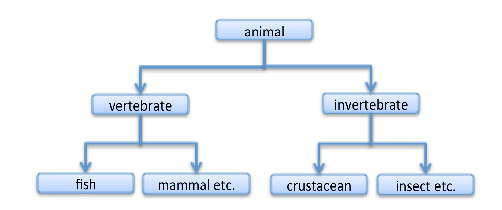
"Mashinani o'rganish - bu kompyuterlarga aniq dasturlashtirilmagan holda o'rganish qobiliyatini beradigan o'rganish sohasi ."

Kompyuter dasturi o'rganish uchun aytiladi tajriba Ba'zilarga nisbatan EvazifaT va ba'zilar ishlash ko'rsatkichi P, agar P tomonidan o'lchangan T bo'yicha uning ishlashi E tajribasi bilan yaxshilanadi. - Tom Mitchell, Karnegi Mellon universiteti;

Shunday qilib, agar siz dasturingiz, masalan, gavjum chorrahadagi (T topshiriq) tirbandlikni bashorat qilishni istasangiz, uni o'tmishdagi trafik naqshlari haqidagi ma'lumotlarga ega mashina o'rganish algoritmi orqali ishga tushirishingiz mumkin (E tajribasi), agar u muvaffaqiyatli "o'rgangan" bo'lsa. , keyinchalik u kelajakdagi trafik shakllarini bashorat qilishda yaxshiroq ishlaydi (ishlash o'lchovi P).

ML vazifalarining har xil turlari orasida biz **nazorat ostida o'rganish (SL) deb ataydigan narsa bor.** Bu siz allaqachon javoblaringiz bo'lgan ba'zi ma'lumotlarni kiritgan vaziyat (masalan, itning ma'lum bir zot ekanligini taxmin qilish uchun biz millionlab it ma'lumotlarini/xususiyatlarini yuklaymiz, masalan, turi, bo'yi, terining rangi, tana sochining uzunligi va boshqalar. ML lingoda bu xususiyatlar "xususiyatlar" deb ataladi. Ushbu xususiyatlar ro'yxatining bitta yozuvi ma'lumotlar namunasi bo'lib, hamma narsa to'plami sizning bashoratingizning asosini tashkil etuvchi Trening ma'lumotlaridir, ya'ni terining rangini bilsangiz. , ma'lum bir itning tana soch uzunligi, bo'yi va boshqalar, keyin siz, ehtimol, tegishli bo'lgan zotni taxmin qilishingiz mumkin.

Yadrolarga o'tishdan oldin, biz qo'llab-quvvatlash vektor mashinasi nima ekanligini tushunishimiz kerak. Vektor mashinasi yoki SVMni qo‘llab-quvvatlash – bu ma’lumotlarni tasniflash uchun tahlil qiluvchi, bog‘langan o‘rganish [algoritmlariga](https://en.wikipedia.org/wiki/Algorithm) ega bo‘lgan nazorat ostidagi o‘rganish modellaridir (tasniflar nimaga tegishli ekanligini bilishni anglatadi, masalan, “olma” “meva” sinfiga, “it” esa “hayvonlar” sinfiga tegishli –7-rasmga qarang. )

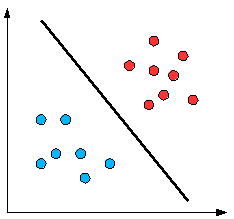


*7-rasm*

Qo'llab-quvvatlovchi vektorli mashinalarda u ko'k sharlarni qizildan ajratib turadigan quyidagi 8-rasmga o'xshaydi.

SVM - bu ajratuvchi giperplan bilan rasmiy ravishda **belgilangan tasniflagich . Giper tekislik atrof-muhit bo'shlig'idan** bir **o'lchamdagi** kichik bo'shliqdir . Matematik fazoning (yoki ob'ektning) o'lchami norasmiy ravishda uning ichidagi har qanday nuqtani (masalan, har bir ko'k va qizil nuqta) belgilash uchun zarur bo'lgan koordinatalarning minimal soni (x, y, z o'qi) sifatida belgilanadi **,** atrof-muhit maydoni esa uni o'rab turgan bo'shliqdir. matematik ob'ekt. Matematik **ob'ekt** - matematikada paydo bo'ladigan mavhum ob'ekt **Mavhum ob'ekt** - bu biron bir muayyan vaqt yoki joyda mavjud bo'lmagan, balki narsaning bir turi, ya'ni g'oya yoki abstraksiya sifatida mavjud bo'lgan ob'ektdir (vikipediya).

Shuning uchun pastda joylashgan ikki o'lchovli fazoning giper tekisligi (8-rasm) qizil va ko'k nuqtalarni ajratuvchi bir o'lchovli chiziqdir.



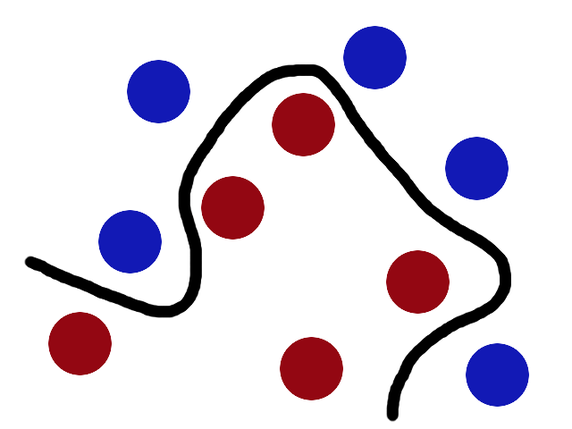
*8-rasm*

Yuqoridagi misoldan ma'lum bir itning zotini bashorat qilishga urinish shunday bo'ladi

Ma'lumotlar (itlarning barcha zotlari)→ Xususiyatlar (teri rangi, sochlari va boshqalar)→ O'rganish algoritmi

Xo'sh, nega yadrolar?

Quyidagi 9-rasmni ko'rib chiqing



*9-rasm*

Yuqoridagi masalani 8-rasmdagi kabi chiziqli hal qilishga harakat qila olasizmi?

YO'Q!

Qizil va ko'k to'plarni to'g'ri chiziq bilan ajratib bo'lmaydi, chunki ular tasodifiy taqsimlanadi va bu, aslida, haqiqiy hayot muammosi ma'lumotlarining ko'pchiligi - **tasodifiy taqsimlangan.**

Mashinani o'rganishda "yadro" odatda yadro hiylasiga murojaat qilish uchun ishlatiladi, chiziqli bo'lmagan muammoni hal qilish uchun chiziqli tasniflagichdan foydalanish usuli.Bu (9-rasm) kabi chiziqli ajralmaydigan ma'lumotlarni chiziqli bo'linadigan ma'lumotlarga (8-rasm) aylantirishni o'z ichiga oladi. Yadro funktsiyasi - bu har bir ma'lumot misolida asl chiziqli bo'lmagan kuzatishlarni ajratish mumkin bo'lgan yuqori o'lchamli bo'shliqqa joylashtirish uchun qo'llaniladigan narsa.

It zotini bashorat qilish misolidan yana foydalanib, yadrolar yaxshiroq alternativani taklif qiladi. Bir qator xususiyatlarni aniqlash o'rniga, siz it zotlari o'rtasidagi **o'xshashlikni** hisoblash uchun bitta yadro funktsiyasini belgilaysiz. Siz ushbu yadroni ma'lumotlar va teglar bilan birga o'rganish algoritmiga taqdim etasiz va klassifikator chiqadi.

**Bu qanday ishlaydi?**

Yadrolar qanday ishlashini yaxshiroq tushunish uchun keling, Lili Jiangning matematik [illyustratsiyasidan foydalanamiz](https://www.quora.com/What-are-Kernels-in-Machine-Learning-and-SVM/answer/Lili-Jiang?srid=oOgT).

**Matematik ta'rif**: K(x, y) = <f(x), f(y)>. Bu erda K - yadro funktsiyasi, x, y - n o'lchovli kirish. f - n-o'lchovdan m-o'lchovli fazoga xarita. < x,y> nuqta hosilasini bildiradi. odatda m n dan ancha katta.

**Sezgi***:* odatda <f(x), f(y)> ni hisoblash uchun avval f(x), f(y) ni hisoblashimiz kerak, keyin esa nuqta mahsulotini bajarishimiz kerak. Ushbu ikki hisoblash bosqichi ancha qimmat bo'lishi mumkin, chunki ular m o'lchovli fazoda manipulyatsiyani o'z ichiga oladi, bu erda m katta raqam bo'lishi mumkin. Ammo yuqori o'lchamli fazoga borishdagi barcha qiyinchiliklardan so'ng, nuqta mahsulotining natijasi haqiqatan ham skalerdir: biz yana bir o'lchovli fazoga qaytamiz! Endi bizda savol tug'iladi: bu bitta raqamni olish uchun barcha qiyinchiliklarni boshdan kechirishimiz kerakmi? Biz haqiqatan ham m o'lchovli fazoga borishimiz kerakmi? Agar siz aqlli yadro topsangiz, javob yo'q.

**Oddiy misol:** x = (x1, x2, x3); y = (y1, y2, y3). U holda f(x) = (x1x1, x1x2, x1x3, x2x1, x2x2, x2x3, x3x1, x3x2, x3x3) funksiya uchun yadro K(x, y ) = (<x, y>)² bo'ladi.

Buni yanada intuitiv qilish uchun ba'zi raqamlarni kiritamiz: deylik x = (1, 2, 3); y = (4, 5, 6). Keyin:  
f(x) = (1, 2, 3, 2, 4, 6, 3, 6, 9)

f(y) = (16, 20, 24, 20, 25, 30, 24, 30, 36)

<f(x), f(y)> = 16 + 40 + 72 + 40 + 100+ 180 + 72 + 180 + 324 = 1024

Ko'p algebra, asosan f 3 o'lchovli dan 9 o'lchovli fazoga xaritalash bo'lgani uchun.

Endi uning o'rniga yadrodan foydalanamiz:

K(x, y) = (4 + 10 + 18 ) ^2 = 32² = 1024

Xuddi shu natija, lekin bu hisoblash juda oson.

**Xulosa**

KNN algoritmi eng oddiy tasniflash algoritmlaridan biridir. Bunday soddaligi bilan ham u yuqori raqobatbardosh natijalar berishi mumkin. KNN algoritmidan regressiya muammolari uchun ham foydalanish mumkin. Muhokama qilingan metodologiyadan yagona farq k-eng yaqin qo'shnilardan ovoz berish o'rniga eng yaqin qo'shnilarning o'rtacha ko'rsatkichlaridan foydalanish bo'ladi. KNN R da bitta satrda kodlanishi mumkin. Men SAS da KNN algoritmidan qanday foydalanishimiz mumkinligini hali o'rganmayapman.

**Asosiy xulosalar**

KNN klassifikatori ma'lum bir ma'lumot nuqtasiga eng yaqin qo'shnilarni topish orqali ishlaydi va ma'lumotlar nuqtasini tasniflash uchun ko'pchilik ovoz oladi.

K qiymati hal qiluvchi ahamiyatga ega va modelni haddan tashqari yoki to'liq moslashtirmaslik uchun uni oqilona tanlash kerak.

k-NN algoritmi uchun k ning optimal qiymatini tanlash uchun o'zaro tekshirishdan foydalanish mumkin, bu uning ish faoliyatini yaxshilashga va ortiqcha yoki to'liq moslashishning oldini olishga yordam beradi. KNN algoritmini qo'llashdan oldin o'zaro ta'sirni aniqlash uchun ham o'zaro tekshirish qo'llaniladi.

Yuqoridagi maqola KNN ning Python va R-da amalga oshirilishini ta'minlaydi va natijani R-dagi scikit-learn va "Class" kutubxonasi bilan taqqoslaydi.