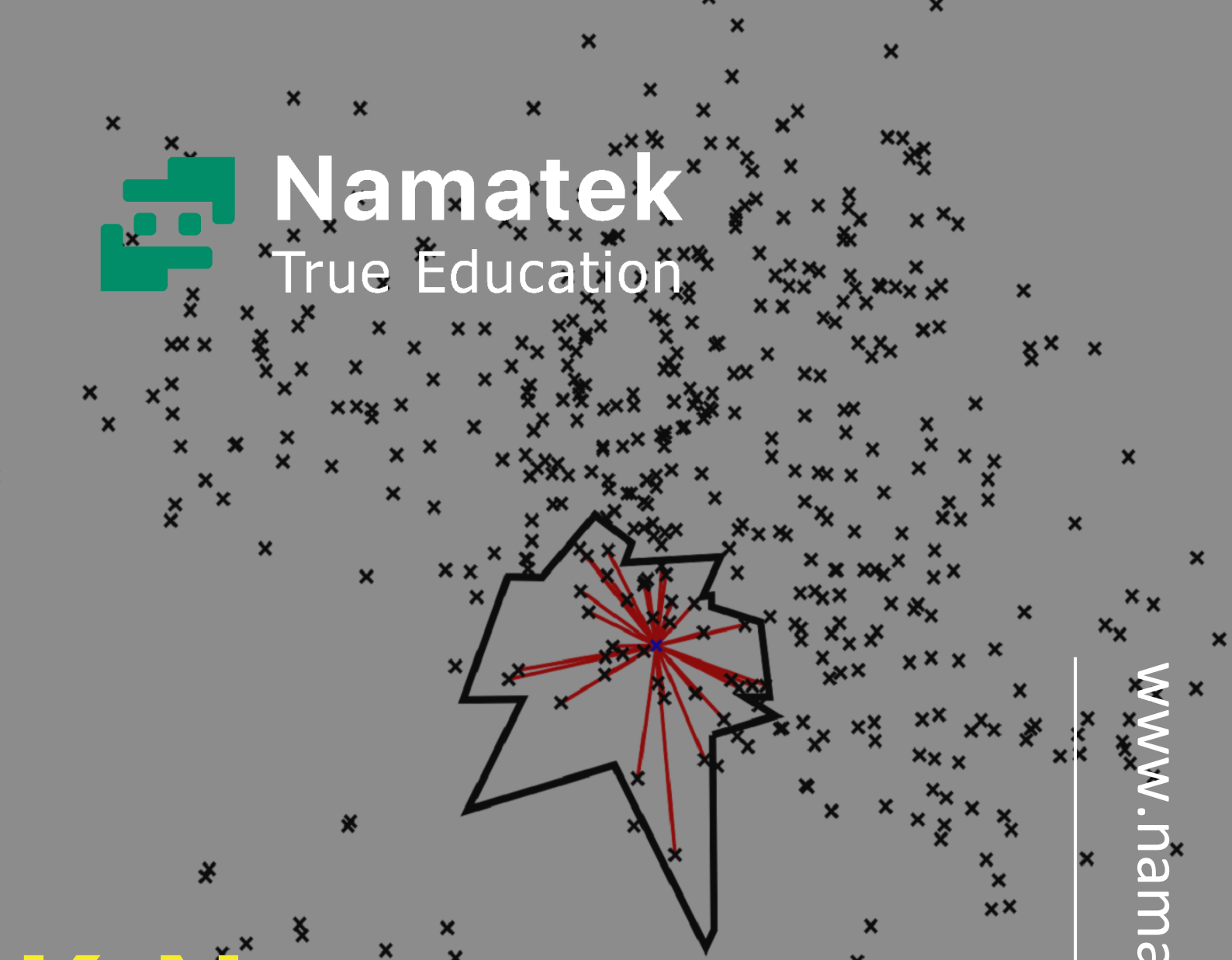
Namatek
True Education



K-Nearest Neighbors

الگوریتم KNN

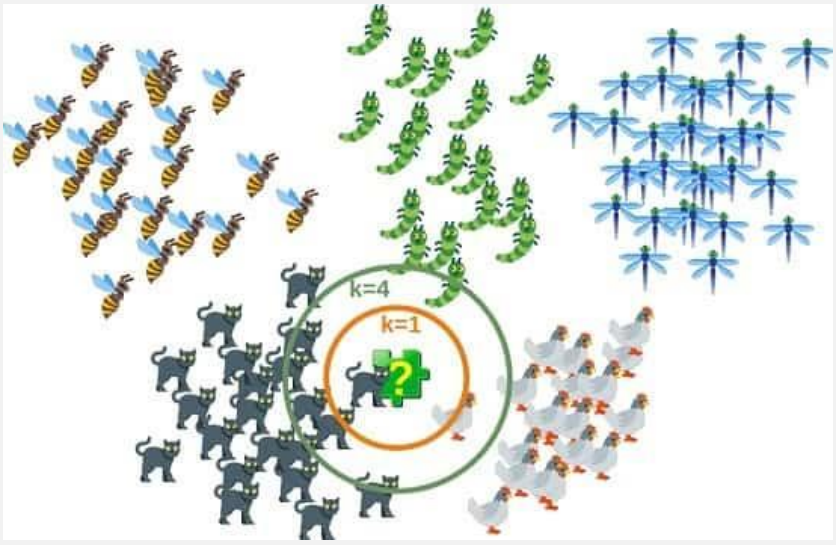
www.namatek.com

فهرست مطالب

الگوریتم KNN، یکی از ساده ترین الگوریتم های یادگیری ماشین بر اساس تکنیک یادگیری نظارت شده است. این الگوریتم شباهت بین موارد یا داده های جدید را با موارد و داده های موجود در نظر می گیرد و موارد جدید را در دسته ای قرار می دهد که بیشترین شباهت را به دسته های موجود دارد. KNN یک الگوریتم غیرپارامتریک است؛ به این معنا که هیچ فرضی در مورد داده های اساسی ایجاد نمی کند. از طرفی به آن الگوریتم یادگیرنده تنبل نیز گفته می شود؛ زیرا بلافاصله از مجموعه آموزشی نمی آموزد، در عوض مجموعه داده ها را ذخیره می کند و هنگام طبقه بندی، عملی را روی مجموعه داده ها انجام می دهد.

در این مقاله به بررسی الگوریتم KNN، علل اهمیت آن، الزامات معیارهای فاصله در آن، کاربردها، نحوه کارکرد و مزایا و معایب آن خواهیم پرداخت.

الگوریتم KNN چیست؟



KNN مخفف عبارت K – nearest neighbors و به معنای نزدیک ترین همسایه K است. الگوریتم KNN، یک طبقه بندی کننده یادگیری غیرپارامتریک و نظارت شده است که از نزدیک برای انجام طبقه بندی یا

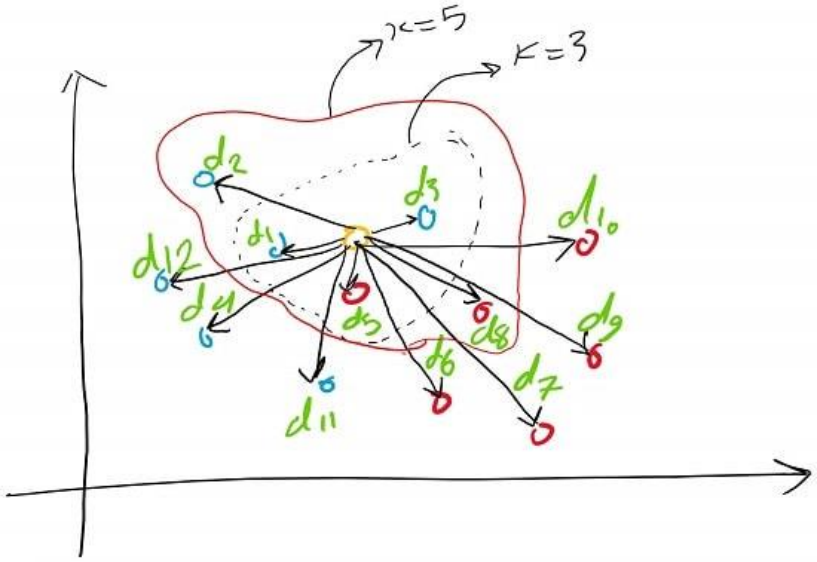
پیش بینی در مورد گروه بندی یک نقطه از داده های فردی استفاده می شود. این امر مبتنی بر این ایده است که نقاط داده مشابه تمایل دارند، برچسب یا مقادیر مشابه یکدیگر داشته باشند. الگوریتم KNN، یکی از محبوب ترین طبقه بندی کننده های دسته بندی و رگرسیون است که امروزه در یادگیری ماشین استفاده می شود. در طول مرحله آموزش، الگوریتم KNN کل داده های آموزشی را به عنوان یک مرجع ذخیره می کند. هنگام پیش بینی، فاصله بین نقطه داده های ورودی و تمامی مثال های آموزشی را با استفاده از یک متریک فاصله انتخابی مانند فاصله اقلیدسی محاسبه می کند.

سپس، الگوریتم K نزدیکترین همسایه به نقطه داده ورودی را بر اساس فاصله آن ها شناسایی می کند. در مورد طبقه بندی هم این الگوریتم، رایج ترین برچسب کلاس را در بین همسایگان K به عنوان برچسب پیش بینی شده برای نقطه داده ورودی اختصاص می دهد. برای رگرسیون هم میانگین یا میانگین وزنی مقادیر هدف همسایگان K را محاسبه می کند تا مقدار نقطه داده ورودی را پیش بینی کند.

چرا به الگوریتم KNN نیاز داریم؟

برای درک بیشتر موضوع، بهتر است علت نیاز به الگوریتم KNN را با یک مثال بیان کنیم. فرض کنید دو دسته وجود دارند، دسته A و دستهB ، ما یک نقطه داده ای جدید با نام x_1 داریم که این نقطه می تواند در هر یک از دسته های مذبور قرار گیرد. برای حل این نوع مسائل به الگوریتم KNN نیاز داریم. با کمک الگوریتم KNN ما به راحتی می توانیم، دسته یا کلاس یک مجموعه خاص از داده ها را شناسایی کنیم.

علل اهمیت الگوریتم KNN

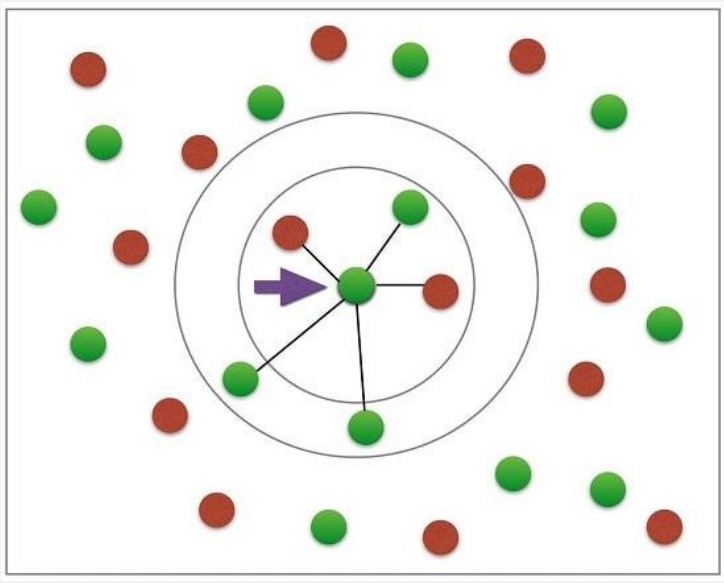


برخلاف بسیاری از الگوریتم های یادگیری ماشین دیگری که وجود دارند، KNN برای پیاده سازی به صورتی ساده و شهودی کار می کند.

- این الگوریتم، انعطاف پذیر است؛ یعنی دارای معیارهای فاصله زیادی برای انتخاب است.
- با آشکار شدن داده های جدید تکامل پیدا می کند.
- دارای یک فراپارامتر واحد برای تنظیم) مقدار (k است.

این الگوریتم می تواند داده های توزیع شده خطی یا غیرخطی را تشخیص دهد و از آنجایی که غیرپارامتریک است، هیچگونه فرضی به منظور پیاده سازی آن وجود ندارد. یعنی برخلاف انواع رگرسیون خطی که دارای مفروضات زیادی هستند، باید توسط داده ای و قبل از این که بتوان آن ها را برآورده کرد، به کار گرفته شوند.

الزامات محاسبه معیارهای فاصله در الگوریتم KNN



به صورت خلاصه، هدف الگوریتم KNN، شناسایی نزدیک ترین همسایه یک نقطه پرس و جو است؛ به گونه ای که بتوان یک برچسب کلاس به آن نقطه اختصاص داد. به همین منظور، KNNدارای یکسری الزامات است که در ادامه با آن ها آشنا خواهیم شد.

تعیین معیار فاصله در الگوریتم KNN

برای تعیین این که کدام یک از نقاط داده به یک نقطه پرس و جو نزدیک تر هستند، فاصله بین نقطه پرس و جو و سایر نقاط داده باید محاسبه شود. این معیارهای فاصله به شکل گیری مرزهای تصمیم گیری کمک می کنند؛ به نحوی که نقاط پرس و جو را به مناطق مختلف تقسیم می کنند. معمولاً مرزهای تصمیم را با نمودارهای ورونوی (Voronoi) مشخص می کنند.

در ادامه به بررسی تعدادی از معیارهای فاصله می پردازیم.

1) فاصله اقلیدسی

یکی از رایج ترین معیارهای اندازه گیری فاصله است و به بردارهای با ارزش واقعی محدود می شود. با استفاده از فرمول زیر، یک خط مستقیم بین نقطه پرس و جو و نقاط دیگر اندازه گیری می شوند.

$$d(x,y) = \sqrt{\sum_{i=1}^{n} (y_i - x_i)^2}$$

2) فاصله منهتن

فاصله منهتن (Manhattan) نیز یکی دیگر از معیارهای محبوب فاصله است که قدر مطلق بین دو نقطه را اندازه گیری می کند. همچنین به عنوان فاصله بلوک شهری یا فاصله تاکسی شناخته می شود؛ زیرا معمولاً با یک شبکه شناخته تجسم می شود و نشان می دهد که چگونه می توان از یک آدرس به آدرس دیگر از طریق خیابان های شهر حرکت کرد.

$$\text{فاصله منهتن} = d(x,y) = \left(\sum_{i=1}^{m} |x_i - y_i| \right)$$

3) فاصله مینکوفسکی

فاصله مینکوفسکی(Minkowski) ، شکل تعمیم یافته فاصله اقلیدسی و منهتن است. در فرمول زیر امکان ایجاد سایر معیارهای فاصله فراهم می شود. فاصله اقلیدسی با این فرمول نشان داده می شود که p برابر با ۲ است و فاصله منهتن با p برابر با یک نشان داده می شود.

$$\text{فاصله مینکوفسکی} = \left(\sum_{i=1}^{n} \left|x_i - y_i\right|\right)^{1/p}$$

4) فاصله همینگ

این تکنیک معمولاً با بردارهای بولی یا رشته ای استفاده می شود و نقاطی را که با بردارها مطابقت ندارند، مشخص می کند. در نتیجه از آن به عنوان متریک همپوشانی نیز یاد می شود. این مسئله را می توان با فرمول زیر نشان داد.

$$\text{فاصله همینگ} = D_H = \left(\sum_{i=1}^{k} \left|x_i - y_i\right|\right)$$

$$x=y \qquad D=0$$
$$x\neq y \qquad D\neq 1$$

مقدار K در الگوریتم KNN

در الگوریتم KNN، برای انجام یک کار طبقه بندی یا رگرسیون باید تعدادی همسایه تعریف کنید و این عدد با پارامتر K نشان داده می شود. به عبارت دیگر، Kتعداد همسایه هایی را که الگوریتم از آن ها استفاده می کند، هنگام تخصیص یک ارزش به هر مشاهده جدید، تعیین می کند. این عدد می تواند از یک (در این حالت الگوریتم، برای هر پیش بینی تنها نزدیک ترین همسایه را به کار می برد) به تعداد کل نقاط داده های مجموعه داده (در این حالت نیز، الگوریتم کلاس اکثریت مجموعه داده را به صورت کامل پیش بینی خواهد کرد.) برود. تعریف K می تواند یک عمل متعادل کننده باشد؛ زیرا مقادیر مختلف می توانند منجر به برازش بیش از حد یا عدم تناسب شود. مقادیر کمتر K می توانند، واریانس بالا، اما بایاس (Bias) کم داشته باشند و مقادیر زیاد K ممکن است منجر به بایاس بالا و واریانس کم شود.

چگونه مقدار مناسب برای K انتخاب کنیم؟

به منظور انتخاب K مناسب، الگوریتم KNN را چندین بار و با مقادیر مختلف K اجرا می کنیم. در نهایت K را به گونه تعیین می کنیم که تعداد خطاها را کاهش دهد و در عین حال توانایی الگوریتم برای پیش بینی دقیق داده های جدیدی که برای آن تعیین می کنید، حفظ شود.

چگونه مقدار K را بهینه کنیم؟

برای تعیین مقدار K هیچ روش آماری از پیش تعیین شده ای وجود ندارد؛ اما چند قرارداد رایج وجود دارد که بهتر است با آن ها آشنا شویم:

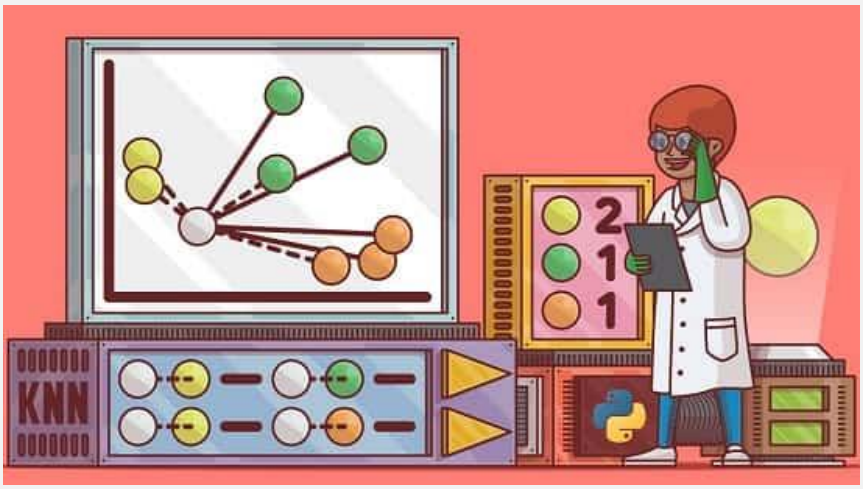
۱. مقدار K تصادفی را در نظر بگیرید و محاسبات را شروع کنید.

۲. انتخاب یک مقدار بسیار پایین برای K ممکن است به احتمال زیاد منجر به پیش بینی های نادرست و ایجاد مرزهای تصمیم ناپایداری شود.

۳. نموداری بین میزان خطا و K که مقادیر را در یک محدوده تعریف شده نشان می دهد، استخراج کنید. سپس میزان K را به صورتی که کمترین میزان خطا را داشته باشد، تعیین کنید.

اکنون و با اجرای این مدل، ایده ای به منظور انتخاب مقدار بهینه K خواهید داشت.

کاربردهای الگوریتم KNN



الگوریتم KNN را می توان برای مسائل طبقه بندی و پیش بینی رگرسیون استفاده کرد. با این حال، بیشتر در مورد مسائل طبقه بندی در صنعت مورد استفاده قرار می گیرد.

در بررسی کاربردهای الگوریتم KNN باید ۳ جنبه را مد نظر قرار داد:

- سهولت در تفسیر خروجی ها
- زمان محاسبه
- قدرت پیش بینی

از جمله کاربردهای الگوریتم KNN می توان به موارد زیر اشاره کرد.

پیش پردازش داده ها

مجموعه داده ها اغلب دارای مقادیر گم شده ای هستند؛ اما الگوریتم KNN می تواند این مقادیر را در فرآیندی با نام داده های گمشده، تخمین بزند.

موتورهای پیشنهادی

با استفاده از داده های جریان کلیک از وبسایت ها، الگوریتم KNN برای ارائه توصیه های خودکار به کاربران در مورد محتوای اضافی استفاده می شود. این تحقیق نشان می دهد که کاربر به یک گروه خاص تعلق دارد و بر اساس رفتارهای کاربر در آن گروه، به آن ها پیشنهاداتی ارائه می شود. با این حال، با توجه به مسائل مقیاس بندی با KNN، این چنین رویکردی ممکن است برای مجموعه داده های بزرگتر بهینه نباشد.

امور مالی و اقتصادی

این الگوریتم، همچنین در انواع موارد مالی و اقتصادی نیز استفاده می شود. برای مثال استفاده از KNN، بر روی داده های اعتباری می تواند به بانک ها در ارزیابی ریسک وامی که به یک فرد یا سازمان داده می شود، کمک کند. همچنین، به منظور تعیین اعتبار متقاضی وام استفاده نیز می شود.

الگوریتم KNN در موارد زیر نیز کاربرد دارد:

- پیش بینی بازار سهام
- نرخ ارز
- معاملات آتی
- تحلیل های پولشویی

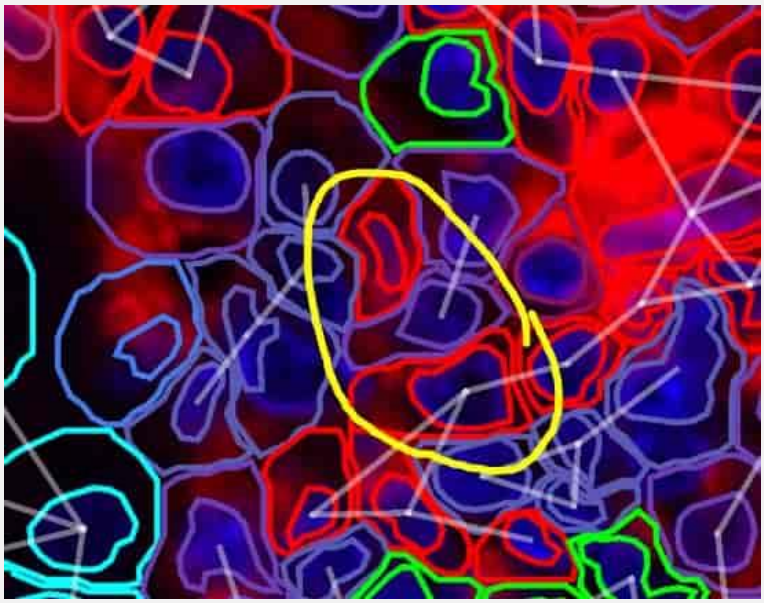
مراقبت های بهداشتی

الگوریتم KNN در صنعت مراقبت های بهداشتی نیز کاربرد دارد و احتمال خطر حملات قلبی و سرطان پروستات را پیش بینی می کند. این الگوریتم با محاسبه محتمل ترین عبارات ژنی کار می کند.

تشخیص الگو

از الگوریتم KNN در شناسایی الگوها، مانند طبقه بندی متن و رقم استفاده می شود. این امر به ویژه در شناسایی شماره های دست نویسی که ممکن است در فرم یا پاکت های پستی وجود داشته باشد، بسیار مفید خواهد بود.

چگونگی کارکرد الگوریتم KNN



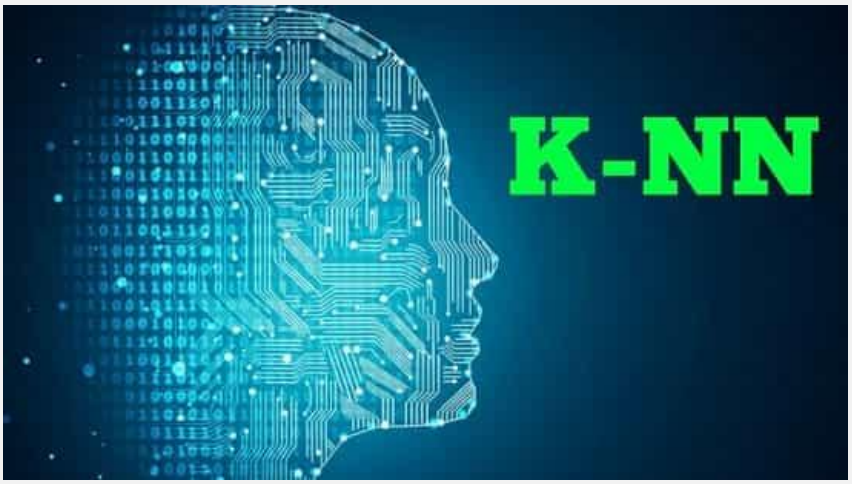
الگوریتم KNN، یک ورودی داده جدید را با مقادیر موجود در یک مجموعه داده معین (با کلاس ها یا دسته های مختلف) مقایسه می کند. این الگوریتم، بر اساس نزدیکی یا شباهت آن در یک محدوده معین (K) از

همسایگان، داده های جدید را به یک کلاس یا دسته در مجموعه داده (داده های آموزشی) اختصاص می دهد.

برای درک بهتر، آن را به مراحل زیر تقسیم می کنیم:

۱. یک مقدار به K اختصاص دهید.
۲. فاصله بین داده ورودی جدید و سایر ورودی های داده موجود را محاسبه کنید. سپس آن ها را به ترتیب صعودی مرتب کنید.
۳. نزدیک ترین همسایه K به ورودی جدید را بر اساس فواصل محاسبه شده، پیدا کنید.
٤. ورودی داده جدید را به کلاس اکثریت در نزدیک ترین همسایگان اختصاص دهید.

مزایا و معایب الگوریتم KNN



الگوریتمKNN ، درست مانند هر الگوریتم یادگیری ماشین، نقاط ضعف و قدرت مخصوص به خود را دارد و بسته به پروژه و برنامه، ممکن است انتخاب مناسبی باشد و در مواردی نیز انتخاب خوبی نباشد. در ادامه به بررسی مزایا و معایب الگوریتم KNN می پردازیم.

مزایای الگوریتم KNN

از جمله مزایای الگوریتم KNN می توان به موارد زیر اشاره کرد:

- **پیاده سازی آسان:** با توجه به سادگی و میزان دقت این الگوریتم، یکی از اولین طبقه بندی کننده هایی است که یک دانشمند داده جدید باید آن را فراگیرد.
- **تطبیق پذیری آسان:** با اضافه شدن نمونه های آموزشی جدید، این الگوریتم به منظور محاسبه داده های جدید تنظیم می شود؛ زیرا تمامی داده های آموزشی در حافظه آن ذخیره خواهند شد.
- **نیاز به تعداد کم هایپر پارامتر (Few Hyperparameters):** KNN تنها به یک مقدار برای K و یک فاصله متریک نیاز دارد که در مقایسه با سایر الگوریتم های یادگیری ماشین تعداد کمی است.

معایب الگوریتم KNN

معایب الگوریتم های KNN عبارت اند از:

- **مقیاس بد:** KNN در مقایسه با سایر طبقه بندی کننده ها، حافظه و ذخیره داده بیشتری را اشغال می کند که می تواند از نظر مالی و زمانی، هزینه بر باشد. حافظه و فضای ذخیره سازی بیشتر، باعث افزایش هزینه های کسب و کار می شود و قاعدتاً محاسبه داده های بیشتر، زمان بیشتری می برد. در حالی که ساختارهای داده متفاوتی مانند Ball – Tree به منظور رسیدگی به ناکارآمدی های محاسباتی ایجاد شده است؛ یک طبقه بندی متفاوت ممکن است، بسته به مشکل تجاری ایجاد شده، ایده آل باشد.

- **عملکرد نامناسب در ابعاد بالاتر:** الگوریتم KNN با ورودی داده هایی با ابعاد بالا عملکرد خوبی ندارد. گاهی اوقات به این پدیده، پیکینگ (Peaking) یا اوج گیری نیز گفته می شود که در آن پس از دستیابی الگوریتم به تعدادی بهینه ویژگی، ویژگی های اضافی موجود، میزان خطاهای طبقه بندی را افزایش می دهند، به خصوص زمانی که حجم نمونه کوچکتر باشد.
- **برازش بیش از حد:** به دلیل عملکرد نامناسب الگوریتم KNN در ابعاد بالا، این الگوریتم مستعد برازش بیش از حد است. در حالی که تکنیک های انتخاب ویژگی و کاهش ابعاد به منظور جلوگیری از بروز این اتفاق استفاده می شوند، مقدار K نیز می تواند بر رفتار مدل تأثیر بگذارد. مقادیر پایین تر K می توانند بیش از حد بر داده ها منطبق باشند، این در حالی است که مقادیر بالاتر K تمایل به هموار کردن مقادیر دارند؛ زیرا میانگین مقادیر را در یک منطقه یا همسایگی بیشتر می کند.