تشخیص سرطان با استفاده از الگوریتم درخت تصمیم

نام دانشجو: صالح سام پناه

نام استاد: امین دهقان

خرداد ۱۴۰۳

چکیده

امروزه سرطان به یکی از بیماری های خطرناک به شمار می رود که انواع بسیار زیادی دارد. یکی از مواردی که درباره این بیماری بسیار ضروری و دارای اهمیت است و به راحت تر شدن روند درمان کمک بسیار زیادی می کند بحث پیش بینی و تشخیص بیماری سرطان قبل از رسیدن به مراحل خطرناک بیماری است. به همین دلیل طبقه بندی سرطان ها امری مهم می باشد.

برای انجام این کار یارگیری ماشین که یک زیر شاخه از هوش مصنوعی به شمار می آید. تکنیک های آماری و احتمالی و بهینه سازی را بکار میگیرد تا کامپیوتر از این طریق بتوانند از مثال های گذشته بیاموزند تا بدین وسیله الگو هایی از مجموعه داده های پیچیده و بزرگ را بدست آورند.

موفقیت در یادگیری ماشین همیشه صد درصدی و تضمین شده نیست. اگر داده ها ورودی کیفیت پایینی را داشته باشند آنگاه به احتمال زیاد نتیجه نیز از کیفیت پایینی برخوردار خواهد بود. فهمیدن این که کدام الگوریتم برای حل هر مسئله ای بهتر است موضوع واضحی نیست بنابراین لازم است که بیش از یک روش برای یادگیری استفاده شود. این کار موجب صرفه جویی در وقت و هزینه ها می شود.

درخت تصمیم

درخت تصمیم یکی از اولین و پر کاربرد ترین روش ها در یادگیری ماشین است که صورت گسترده در حل مسائل طبقه بندی داده ها از آن استفاده می شود. در یک تعریف کلی درخت تصمیم گرافی ساختار یافته است که روند تصمیمات گره های این گراف هستند و برگ ها نشان دهنده تصمیم گیری هستند.

در این مقاله به ترتیب به داده های سرطان در یادگیری ماشین ماشین، زبان برنامه نویسی پایتون در هوش مصنوعی، پیش پردازش داده ها،طبقه بندی در یادگیری ماشین،الگوریتم های یادگیری ماشین،الگوریتم درخت تصمیم،تست کردن الگوریتم، بدست آوردن دقت و در نهایت توضیح کدها و نتیجه گیری نهایی شرح داده شده است.

فهرست

داده های سرطان در یادگیری ماشین

داده های ورودی برای الگوریتم نقش بسیار مهمی دارند و به همین دلیل کیفیت داده ها امری بسیار مهم بشمار می آید. از این رو برای انجام یک پیش بینی معقول باید پزشک معالج اطلاعات مهم مانند: اطلاعات مبتنی بر سلول و اطلاعات جمعیت شناختی و اطلاعات بالینی را با دقت بسیار زیاد جمع آوری کند. که این موارد شامل سابقه بیماری در خانواده و بستگان ، سن فرد ، رژیم غذایی ، وضعیت اضافه وزن ، عادات پر خطر (مصرف دخانیات مانند قلیان سیگار ، نوشیدن الکل) ، قرار گرفتن در محیط های پرخطر سرطان زا ، تشعشعات و غیره همی این موارد در پیش بینی سرطان نقش ایفا می کنند.

متاسفانه این موارد به تنهایی برای تشخیص سرطان کافی نمی باشد و به برخی جزئیات در مورد تومور سرطانی یا ژنتیک بیمار نیاز است با توسعه تکنولوژی های ژنومیک و پروتئمیک و تصویربرداری می توان اطلاعات مورد نیاز مولکولی را بدست آورد . برخی جهش ها در برخی ژن ها نیز می توانند ابزار بسیار مهمی برای پیش بینی سرطان باشند.

ترکیب داده های مولکولی با عواملی در مقیاس بزرگ می توانند باعث افزایش دقت و صحت پیش بینی شود اگرچه هر قدر تعداد متغیر های موثر بیشتر شود توانایی ربط دادن آنها به یکدیگر کمتر می شود .
تقریبا همه پیش بینی ها دارای 4 نوع داده ورودی هستند:

- داده های ژنومیک
- داده های پروتئومیک
- داده های بالینی
- ترکیبی از هر سه مورد

با استفاده از روش های یادگیری ماشین می توان دقت پیش بینی حساسیت سرطان، عود سرطان و بقای سرطان را ارتقا بخشید. امروزه با استفاده از این روش ها دقت پیش بینی سرطان ۱۵٪ تا ۲۰٪ افزایش پیدا کرده است.
دو دسته ی اصلی در انواع روش های یادگیری ماشین وجود دارد:

- یادگیری با نظارت
- یادگیری بدون نظارت

زبان برنامه نویسی پایتون در هوش مصنوعی

هوش مصنوعی و یادگیری ماشین از موضوعات پرطرفدار در عصر امروز هستند همین موضوع باعث اشتیاق برای یادگیری آن شده زبان پایتون به عنوان بهترین زبان برنامه نویسی در عرصه هوش مصنوعی و یادگیری ماشین مطرح میشود.پایتون را می توان یک زبان تفسیر شده توصیف کرد یا به عبارت دیگر این زبان قبل از اجرا نیازی به کامپایل (ترجمه تمام کد های اصلی از زبان سطح بالا به زبان کامپیوتر) در دستورالعمل زبان ماشین ندارد و برنامه نویس می تواند مستقیما از آن برای اجرای برنامه استفاده کند. پایتون یک زبان برنامه‌نویسی سطح بالا است که در سناریوهای پیچیده استفاده می‌شود. زبان‌های سطح بالا آرایه‌ها، متغیرها و محاسبات پیچیده، عبارات بولی و سایر مفاهیم انتزاعی در علوم کامپیوتر را به‌منظور کامل‌تر شدن و بهبود استفاده از آن‌ها مدیریت می‌کنند.

10 مزیت استفاده از پایتون در پیاده سازی هوش مصنوعی و یادگیری ماشین

1. سادگی و سازگاری

پایتون تمام مزایای یک کد بی‌عیب و نقص را فراهم می‌کند. هوش مصنوعی و یادگیری ماشین ماشین نیاز به حل الگوریتم‌های پیچیده دارند. با این حال، سادگی پایتون تضمین می‌کند که توسعه‌دهندگان می‌توانند به راحتی کدها را بنویسند. یکی از دلایل اصلی که اکثر افراد به دنبال آموزش صفر تا صد پایتون هستند و آن را انتخاب می‌کنند این است که یادگیری‌اش آسان است.همچنین توسعه‌دهندگان به راحتی کدهای پایتون را درک خواهند کرد.بسیاری از توسعه‌دهندگان معتقدند پایتون زبان بهتری نسبت به گزینه‌های دیگر است. زبان‌های دیگر قادر به ساده‌سازی مفاهیم نیستند؛ در حالی که پایتون از یک محیط مشارکتی بهره می‌برد. پایتون یک زبان همه‌منظوره اساسی است که به راحتی طیف گسترده‌ای از وظایف پیچیده را انجام می‌دهد.

2. سیستم کتابخانه ای بهتر

پایتون سیستم کتابخانه خوبی دارد که برای فرآیند توسعه بسیار مهم است. منظور از کتابخانه، گروهی از ماژول‌ها با یک مجموعه کد از پیش نوشته شده است. کاربران بر اساس این کدها روی ارتقاء عملکردها تمرکز می‌کنند. کتابخانه‌های پایتون به ارائه آیتم‌های پایه کمک می‌کنند. بنابراین توسعه‌دهندگان هنگام انتخاب توسعه پایتون، به‌طور مداوم کدها را نمی‌نویسند. یادگیری ماشینی به پردازش داده‌ها وابسته است. در نتیجه این پلتفرم، مزیت مدیریت داده‌های حیاتی را فراهم می‌کند.

3. انعطاف پذیری

زبان برنامه‌نویسی پایتون امکان انتخاب بین OOPS و اسکریپت را فراهم می‌کند و به شما اجازه می‌دهد برای ایجاد هرگونه تغییر، کد منبع را دوباره کامپایل کنید. پایتون به‌عنوان یک پلتفرم انعطاف‌پذیر، به توسعه‌دهندگان این امکان را می‌دهد تا از بین سبک‌های مختلف برنامه‌نویسی حق انتخاب داشته باشند و بسته به نیاز خود سبک‌های مختلف را با هم ترکیب کنند.

4. محبوبیت جهانی

طبق تحقیقات سال ۲۰۲۰ توسط "Stack Overflow" پایتون یکی از ۵ زبان برنامه‌نویسی محبوب و یکی از رایج‌ترین زبان‌های مورد استفاده برای توسعه وب است. شرکت‌های پیشرو در سراسر جهان از پایتون استفاده می‌کنند.

5. امکان تجسم بهتر

ما قبلاً فهمیدیم که پایتون کتابخانه‌های آنلاین مختلفی دارد و اکثر این کتابخانه‌ها دارای ابزارهای تجسم انحصاری هستند. وقتی صحبت از هوش مصنوعی به میان می‌آید، توسعه‌دهندگان باید برای جلب توجه، تصاویر را برجسته و داده‌ها را قابل خواندن کنند. کتابخانه‌هایی مانند Matplotlib این امکان را می‌دهد تا هیستوگرام و نمودارها را برای کمک به درک، نمایش و تجسم داده‌ها ایجاد کنید و گزارش‌های بهتری بسازید. آموزش پروژه‌محور پایتون انجام این کارها را برای شما ساده‌تر می‌کند.

6. خوانایی بیشتر

پایتون یکی از پلتفرم‌های پیشرو است که از خوانایی عالی بهره‌مند است. از آنجایی که آن یک زبان برنامه‌نویسی آسان برای خواندن است؛ پس مبتدیان می‌توانند به راحتی کدها را به اشتراک بگذارند و تغییر دهند. پایتون برخلاف سایر زبان‌های برنامه‌نویسی اصلاً پیچیده نیست. برای آموزش کامل پایتون ویدیو های مختلفی در یوتیوب وجود دارد. سهولت استفاده از زبان برنامه‌نویسی نقش مهمی در تبادل ایده‌ها، ابزارها و الگوریتم‌ها دارد. بنابراین افراد حرفه‌ای حوزه Ai به‌راحتی از پایتون برای ایجاد تغییرات جزئی یا عمده در پروژه‌های خود استفاده می‌کنند.

7. استقلال سکو

همه زبان‌های برنامه‌نویسی، مستقل از پلتفرم نیستند. با این حال، پایتون یک زبان همه‌کاره است که از استقلال پلتفرم سود می‌برد. این زبان به راحتی روی پلتفرم‌های مختلف مانند Windows، macOS، Linux،unix و ... کار می‌کند. یادگیری و آموزش هوش مصنوعی با پایتون این امکان را به توسعه‌دهندگان می‌دهد تا از تمام ویژگی‌هایی که روی یک ماشین پیاده کردند، مجدد روی ماشین دیگری بدون نیاز به تغییر استفاده کنند.بهترین نکته در مورد پایتون زبان مستقل آن است که توسط پلتفرم‌های مختلف از جمله Linux، Windows،macOS و ... پشتیبانی می‌شود. کد پایتون به‌عنوان یک برنامه مستقل برای اکثر سیستم عامل‌های رایج استفاده می‌شود. به عبارت دیگر، زبان پایتون بدون نیاز به مفسر آن قابل توزیع است. ویژگی استقلال پلتفرم پایتون نقش مهمی در صرفه‌جویی در زمان و هزینه دارد و کمک می‌کند تا کل فرآیند توسعه، سریع‌تر و آسان‌تر شود.

8. توسعه سریع تر

یکی از دلایل اصلی محبوبیت پایتون بین توسعه‌دهندگان، امکان نمونه‌سازی سریع با استفاده از آن است. تا زمانی که با پشته‌ها آشنا باشید، نیازی به هدر دادن زمان و یادگیری توسعه هوش مصنوعی نخواهید داشت.

بسیاری از توسعه‌دهندگان، پایتون را از نظر نوشتن و خوانایی، ساده می‌دانند. علاقه‌مندان به آموزش برنامه‌نویسی هوش مصنوعی با پایتون، نیازی به یادگیری کدهای پیچیده نخواهند داشت. زیرا به لطف در دسترس بودن کتابخانه‌های متعدد، امکان توسعه هوش مصنوعی و ML آسان‌تر شده است.

9. نیاز به کدنویسی کمتر

هوش مصنوعی به سرعت در حال توسعه است. بنابراین استفاده از آن شما را ملزم به استفاده از الگوریتم‌های متعدد می‌کند. هنگامی که از زبان پایتون برای توسعه هوش مصنوعی و ML استفاده می‌کنید، به بسته‌های از پیش تعریف شده متعددی دسترسی خواهید داشت. پایتون شما را ملزم به کدنویسی پایه نمی‌کند؛ زیرا از قبل بسته‌های از پیش تعریف شده دارد. زبان برنامه‌نویسی پایتون نقش مهمی در آسان کردن کل فرآیند دارد و با آوردن گزینه " Check your code" نیازی به آزمایش کد نیست و خودش این کار را برای شما انجام می‌دهد.

10.سرعت اجرا

از آنجایی که پایتون یک زبان برنامه‌نویسی قابل خواندن است، پس می‌توان فرمول‌های آن را به سرعت اجرا کرد. یادگیری ماشینی به‌ویژه یادگیری عمیق، به جلسات آموزشی طولانی نیاز دارد و گاهی اوقات این جلسات برای روزها ادامه خواهد داشت. با این وجود، پایتون سرعت اجرای بالاتری دارد و این بسیار مهم است. در این راستا کتاب آموزش هوش مصنوعی با پایتون مانند «راهنمای کامل برای ساخت برنامه‌های هوشمند Python 3.x و TensorFlow 2» به شما کمک خواهد کرد. با استفاده از این زبان تنها با چند خط کد می‌توانید کارهای زیادی انجام دهید.

کتابخانه های محبوب پایتون برای هوش مصنوعی و یادگیری ماشین

- TensorFlow

TensorFlow که توسط گوگل توسعه یافته، یک کتابخانه منبع باز برای ساخت و استقرار مدل‌های یادگیری ماشینی است. این کتابخانه برای مدیریت داده‌های مقیاس بزرگ و مدل‌های پیچیده طراحی شده و از یادگیری عمیق و تکنیک‌های یادگیری ماشین سنتی پشتیبانی می‌کند.

- PyTorch

PyTorch که توسط فیس‌بوک توسعه یافته، یک کتابخانه منبع‌باز برای ساخت و استقرار مدل‌های یادگیری عمیق است. این کتابخانه در کنار انعطاف‌پذیری، از نمودارهای محاسباتی پویا پشتیبانی می‌کند؛ به همین علت فرایند اشکال‌زدایی و بهینه‌سازی مدل‌ها آسان‌تر می‌شود.

- Keras

Keras یک API سطح بالا برای ساخت و آموزش مدل‌های یادگیری عمیق است و به گونه‌ای طراحی شده که کاربرپسند و انعطاف‌پذیر باشد.

- Scikit- Learn

Scikit-learn یک کتابخانه برای یادگیری ماشین در پایتون است که ابزارهای ساده و کارآمدی را برای داده کاوی و تجزیه و تحلیل داده ارائه می‌دهد. این مورد، طیف وسیعی از الگوریتم‌های یادگیری تحت نظارت و بدون نظارت و ابزارهایی برای انتخاب و ارزیابی مدل را شامل می‌شود.

Scikit-learn برای کسب‌وکارهایی مانند شرکت‌هایی که به هوش مصنوعی و یادگیری ماشینی متکی هستند، مفید است. به‌عنوان‌مثال، یک شرکت فناوری اطلاعات می‌تواند از Scikit-learn برای طبقه‌بندی کارکنان براساس مهارت‌ها و تجربه‌شان استفاده کند. یا از Keras برای ایجاد یک شبکه عصبی که پیش‌بینی می‌کند کدام متخصص برای نقشی خاص مناسب است، کمک بگیرند.

کاربردهای پایتون در هوش مصنوعی و یادگیری ماشینی چیست؟

تطبیق‌پذیری و قدرت پایتون آن را به زبانی محبوب برای توسعه هوش مصنوعی و یادگیری ماشینی در طیف وسیعی از صنایع تبدیل کرده است. در اینجا برخی از کاربردهای پایتون در هوش مصنوعی و یادگیری ماشین آورده شده است. اگر هنوز شروع به آموزش زبان برنامه‌نویسی پایتون نکردید، این اطلاعات برای شما جالب خواهند بود.

- بینایی کامپیوتر

پایتون اغلب در برنامه‌های بینایی کامپیوتری مانند تشخیص اشیا، تقسیم‌بندی تصویر و تشخیص چهره استفاده می‌شود. به‌عنوان‌مثال، یک شرکت خرده‌فروشی ممکن است از بینایی کامپیوتری که توسط پایتون پشتیبانی می‌شود، برای نظارت بر قفسه‌های فروشگاه و اطمینان از قرار گرفتن محصولات در مکان‌های صحیح استفاده کند.

- پردازش زبان طبیعی "NLP"

پایتون معمولاً در برنامه‌های "NLP" مانند تجزیه و تحلیل احساسات، ترجمه زبان و ربات‌های گفتگو استفاده می‌شود. برای مثال، یک آژانس فناوری اطلاعات ممکن است از NLP پشتیبانی‌شده توسط پایتون برای تجزیه و تحلیل شرح وظایف و تطبیق کارکنان با فرصت‌های شغلی استفاده کند.

- تجزیه و تحلیل

پایتون برای تجزیه و تحلیل و پیش‌بینی فرآیندها هم قابل استفاده است. فرضاً یک متخصص مراقبت‌های بهداشتی ممکن است از این زبان برای پیش‌بینی اینکه کدام بیماران براساس سابقه پزشکی خود در معرض خطر بالای ابتلا به بیماری‌های خاص هستند استفاده کند.

کاربرد پایتون در هوش مصنوعی برای سازمان‌هایی که به دنبال خودکارسازی و ساده‌سازی عملیات خود هستند هم مفید است. به‌عنوان‌مثال، آن‌ها می‌توانند از مدل‌های یادگیری ماشینی مبتنی بر پایتون برای تجزیه و تحلیل داده‌ها و پیش‌بینی درمورد اینکه کدام فرد در یک نقش خاص موفق‌تر است، استفاده کنند.

به‌طور خلاصه، تطبیق‌پذیری و قدرت پایتون آن را به یک زبان ایده‌آل برای ساختن سیستم‌های هوشمند تبدیل کرده است. پایتون می‌تواند مشکلات دنیای واقعی را حل کند. برای شروع آموزش پایتون مقدماتی Self-study (خودآموز) شروع کنید و برای آموزش پایتون پیشرفته از کلاس‌های آنلاین یا حضوری کمک بگیرید تا درک مفاهیم مختلف برای شما راحت‌تر شود

پیش پردازش داده ها

پیش پردازش داده بخشی از آماده سازی داده ها است که در واقع یک پردازش روی داده های خام تعریف می شود تا این داده های خام را برای پردازش های دیگر آماده کند. این کار به صورت سنتی یک مرحله مقدماتی مهم برای فرایند داده کاوی بوده است.

پیش پردازش داده ، داده ها را به قالبی تبدیل می کند که در داده کاوی یادگیری ماشین و سایر کار های علم داده پردازش را آسان تر و موثرتر باشد. این تکنیک معمولا در مراحل اولیه یادگیری ماشین و توسعه هوش مصنوعی برای اطمینان از نتایج دقیق استفاده می شوند.

ابزارهای پیش پردازش داده

نمونه گیری : یک زیرمجموعه نماینده را از جمعیت بزرگی از داده‌ها انتخاب می‌کند.
تبدیل : داده‌های خام را برای تولید یک ورودی واحد دستکاری می‌کند.
حذف نویز : نویز را از داده‌ها حذف می‌کند.
انتساب : داده های آماری مرتبط را برای مقادیر از دست رفته ترکیب می‌کند.
استخراج ویژگی : یک زیرمجموعه ویژگی مرتبط را که در یک زمینه خاص مهم است، بیرون می‌کشد.
این ابزارها و روش‌ها را می‌توان در انواع منابع داده، از جمله داده‌های ذخیره‌شده در فایل‌ها یا پایگاه‌های داده و جریان داده استفاده کرد.

پیش پردازش داده ها چرا مهم است؟

تقریباً هر نوع تجزیه و تحلیل داده، علم داده یا توسعه هوش مصنوعی به نوعی از پیش پردازش داده نیاز دارد تا نتایج قابل اعتماد، دقیق و قوی برای برنامه‌های کاربردی سازمانی ارائه دهد. داده‌های دنیای واقعی کثیف هستند و اغلب توسط انسان‌ها، فرآیندهای کسب و کار و برنامه‌های کاربردی مختلف ایجاد، پردازش و ذخیره می‌شوند. در نتیجه، یک مجموعه داده ممکن است فیلدهای جداگانه نداشته باشد، حاوی خطاهای ورودی دستی باشد، یا داده های تکراری یا نام‌های متفاوتی برای توصیف یک رکورد داشته باشد. انسان‌ها اغلب می‌توانند این مشکلات را در داده‌هایی که در مسیر کسب و کار استفاده می‌کنند شناسایی و اصلاح کنند، اما داده‌هایی که برای آموزش یادگیری ماشین یا الگوریتم‌های یادگیری عمیق استفاده می‌شوند باید به طور خودکار پیش پردازش شوند.

الگوریتم‌های یادگیری ماشین و یادگیری عمیق زمانی بهترین عملکرد را دارند که داده‌ها در قالبی ارائه شوند که جنبه‌های مرتبط مورد نیاز برای حل یک مشکل را برجسته کنند. روش‌های مهندسی ویژگی‌ها که شامل تبدیل داده‌ها، کاهش داده‌ها، انتخاب ویژگی و مقیاس‌بندی ویژگی است، به بازسازی داده‌های خام به شکلی مناسب برای انواع خاصی از الگوریتم‌ها کمک

می‌کنند. این امر می‌تواند به طور قابل توجهی قدرت پردازش و زمان مورد نیاز برای آموزش یک الگوریتم یادگیری ماشینی یا هوش مصنوعی جدید را کاهش دهد.

مراحل پیش پردازش داده

پروفایل داده: پروفایل داده‌ها فرآیند بررسی، تجزیه و تحلیل و بررسی داده‌ها برای جمع آوری آمار در مورد کیفیت آن است. این مرحله با بررسی داده‌های موجود و ویژگی‌های آن شروع می‌شود. متخصصان داده مجموعه‌های داده‌ای را شناسایی می‌کنند که مربوط به مسئله مورد نظر هستند، ویژگی‌های مهم آن را فهرست‌بندی می‌کنند و فرضیه‌ای از ویژگی‌هایی را تشکیل می‌دهند که ممکن است برای تحلیل پیشنهادی یا کار یادگیری ماشین مرتبط باشند. آن‌ها همچنین منابع داده را به مفاهیم کسب و کار مرتبط مرتبط می‌کنند و در نظر می‌گیرند که کدام کتابخانه‌های پیش پردازش پایتون می‌توانند مورد استفاده قرار گیرند.

پاکسازی داده‌ها: هدف در اینجا یافتن ساده‌ترین راه برای اصلاح مشکلات کیفیت است، مانند حذف داده‌های اضافی، پر کردن داده‌های از دست رفته یا اطمینان از مناسب بودن داده‌های خام برای مهندسی ویژگی‌ها.
کاهش داده‌ها: مجموعه داده‌های خام اغلب شامل داده‌های اضافی می‌شوند که از توصیف پدیده‌ها به روش‌های مختلف یا داده‌هایی که به یک کار خاص AI،ML یا تجزیه و تحلیل مرتبط نیستند، ناشی می‌شوند. روش کاهش داده‌ها از تکنیک‌هایی مانند تجزیه و تحلیل مؤلفه‌های اصلی برای تبدیل داده‌های خام به شکل ساده‌تر مناسب برای موارد استفاده خاص استفاده می‌کند.

تبدیل داده‌ها: در اینجا، متخصصان داده به این فکر می‌کنند که چگونه جنبه‌های مختلف داده‌ها باید سازماندهی شوند تا بیشترین معنا را برای هدف داشته باشند. این مرحله می‌تواند شامل مواردی مانند ساختار دادن به داده‌های بدون ساختار و تمرکز روی آن‌ها باشد.

غنی سازی داده‌ها: در این مرحله، متخصصان داده، کتابخانه‌های مهندسی ویژگی‌های مختلف را روی داده‌ها اعمال می‌کنند تا تبدیل‌های مورد نظر را اعمال کنند. نتیجه باید مجموعه داده‌ای باشد که برای دستیابی به تعادل بهینه بین زمان آموزش برای یک مدل جدید و محاسبات مورد نیاز سازماندهی شده است.

اعتبار سنجی داده‌ها: در این مرحله داده‌ها به دو مجموعه تقسیم می‌شوند. اولین مجموعه برای آموزش یک مدل یادگیری ماشین یا یادگیری عمیق استفاده می‌شود. مجموعه دوم داده‌های آزمایشی است که برای سنجش دقت و استحکام مدل به دست آمده استفاده می‌شود. این مرحله دوم به شناسایی هرگونه مشکل در فرضیه استفاده شده در تمیز کردن و مهندسی ویژگی داده‌ها کمک می‌کند. اگر متخصصان داده از نتایج راضی باشند، می توانند وظیفه پیش پردازش را به یک مهندس داده سوق دهند که چگونگی مقیاس بندی آن را برای تولید بیابد. در غیر این صورت، متخصصان داده می‌توانند به عقب برگردند و تغییراتی در نحوه اجرای مراحل پاکسازی داده‌ها و مهندسی ویژگی‌ها ایجاد کنند.

تکنیک های پیش پردازش داده ها

دو دسته اصلی پیش پردازش وجود دارد: تمیز کردن داده‌ها و مهندسی ویژگی داده‌ها. هر کدام شامل تکنیک های متنوعی است که در زیر توضیح داده شده است.

تمیز کردن داده ها

تکنیک‌های پاکسازی داده‌های نامرتب شامل موارد زیر است:

داده های از دست رفته را شناسایی و مرتب کنید: دلایل مختلفی وجود دارد که یک مجموعه داده ممکن است فیلدهای جداگانه داده را از دست بدهد. متخصصان داده باید تصمیم بگیرند که آیا بهتر است رکوردهای دارای فیلدهای گمشده را کنار بگذارند، آن‌ها را نادیده بگیرند یا آن‌ها را با مقدار احتمالی پر کنند. به عنوان مثال، در یک برنامه IoT که دما را ثبت می‌کند، اضافه کردن یک میانگین دمای از دست رفته بین رکورد قبلی و بعدی ممکن است راه حل مطمئنی باشد.

داده‌های نویزی را کاهش دهید: داده‌های دنیای واقعی اغلب پر از نویز هستند که می‌تواند مدل تحلیلی یا هوش مصنوعی را مخدوش کند. به عنوان مثال، یک سنسور دما که به طور مداوم دمای 75 درجه فارنهایت را گزارش می‌کند ممکن است به اشتباه دما را 250 درجه گزارش کند. انواع روش‌های آماری را می‌توان برای کاهش نویز استفاده کرد، از جمله binning، رگرسیون و خوشه بندی.

موارد تکراری را شناسایی و حذف کنید: هنگامی که دو رکورد تکرار می شوند، یک الگوریتم باید تعیین کند که آیا یک اندازه گیری دو بار ثبت شده است یا اینکه رکوردها نشان دهنده رویدادهای مختلف هستند. در برخی موارد، ممکن است تفاوت های جزئی در یک رکورد وجود داشته باشد زیرا یک فیلد به اشتباه ثبت شده است. در موارد دیگر، سوابقی که به نظر تکراری هستند ممکن است واقعاً متفاوت باشند، مانند پدر و پسری با نام مشابه که در یک خانه زندگی می‌کنند اما باید به عنوان افراد جداگانه نشان داده شوند. تکنیک‌های شناسایی و حذف یا پیوستن موارد تکراری می‌تواند به رفع خودکار این نوع مشکلات کمک کند.

مهندسی ویژگی، همانطور که اشاره شد، شامل تکنیک‌هایی است که توسط متخصصان داده برای سازماندهی داده‌ها به روش‌هایی که آموزش مدل‌های داده و استنتاج بر اساس آن‌ها را کارآمدتر می‌کند، استفاده می‌کند. این تکنیک ها شامل موارد زیر است:

مقیاس بندی یا نرمال سازی ویژگی: اغلب، چندین متغیر در مقیاس‌های مختلف تغییر می‌کنند، یا یکی به صورت خطی تغییر می‌کند در حالی که متغیر دیگر به صورت تصاعدی تغییر می‌کند. به عنوان مثال، حقوق ممکن است با هزاران دلار اندازه گیری شود، در حالی که سن به صورت دو رقمی نشان داده می‌شود. مقیاس‌بندی به تغییر شکل داده‌ها کمک می‌کند تا الگوریتم‌ها بتوانند رابطه معنادار بین متغیرها را از هم جدا کنند.

کاهش داده‌ها: متخصصان داده اغلب نیاز به ترکیب انواع منابع داده برای ایجاد یک مدل هوش مصنوعی یا تحلیلی جدید دارند. برخی از متغیرها ممکن است با یک نتیجه مشخص همبستگی نداشته باشند و با خیال راحت کنار گذاشته شوند. سایر متغیرها ممکن است مرتبط باشند، اما فقط از نظر رابطه – مانند نسبت بدهی به اعتبار در مورد مدلی که احتمال بازپرداخت وام را پیش‌بینی می‌کند. تکنیک‌هایی مانند تحلیل مؤلفه‌های اصلی نقش کلیدی در کاهش تعداد ابعاد در مجموعه داده‌های آموزشی به نمایش کارآمدتر دارند.

طبقه بندی داده ها در یادگیری ماشین

طبقه بندی یا Classification نوعی یادگیری تحت نظارت (Supervised Learning) در یادگیری ماشین است که هدف آن یادگیری نگاشت بین داده‌های ورودی و برچسب‌های خروجی است. داده‌های ورودی معمولاً مجموعه‌ای از ویژگی‌ها یا فیچرهایی (Feature) هستند که ورودی را توصیف می‌کنند، در حالی که برچسب خروجی یک کلاس یا دسته از پیش تعریف شده است که ورودی به آن تعلق دارد. هدف الگوریتم طبقه بندی یادگیری یک مرز تصمیم است که کلاس‌های مختلف را در فضای ویژگی از هم جدا می‌کند، به‌طوری که بتواند کلاس ورودی‌های جدید و نادیده را بر اساس ویژگی‌های آن‌ها پیش‌بینی کند؛ برای مثال اگر بخواهیم ایمیل‌های دریافتی را به دو دسته اسپم و غیر اسپم تقسیم کنیم و آن‌ها را شناسایی کنیم، در ابتدا به مدل ماشین لرنینگ مجموعه داده‌ای با برچسب اسپم و غیر اسپم می‌دهیم تا با آن آموزش ببیند و سپس از آن مدل آموزش دیده برای شناسایی یا طبقه‌بندی ایمیل‌های جدید استفاده می‌کنیم. به این کار، طبقه بندی یا Classification گفته می‌شود.

کاربردهای طبقه بندی در ماشین لرنینگ

طبقه بندی کاربردهای متعددی در زمینه‌های مختلف دارد، از جمله:

طبقه بندی تصویر: طبقه بندی تصویر برای شناسایی محتویات یک تصویر مانند وجود اشیا یا افراد استفاده می‌شود. به عنوان مثال، یک الگوریتم طبقه بندی تصویر را می‌توان برای تشخیص انواع مختلف حیوانات، گیاهان یا وسایل نقلیه آموزش داد.

تجزیه و تحلیل احساسات: تجزیه و تحلیل احساسات برای طبقه بندی احساسات یا نظر یک متن خاص، مانند بررسی محصول یا پست رسانه‌های اجتماعی استفاده می‌شود. به عنوان مثال، یک الگوریتم تحلیل احساسات را می‌توان برای طبقه‌بندی متن به عنوان مثبت، منفی یا خنثی آموزش داد.

تشخیص تقلب: تشخیص تقلب برای شناسایی فعالیت‌های متقلبانه در تراکنش‌های مالی مانند کلاهبرداری از کارت اعتباری استفاده می‌شود. به عنوان مثال، یک الگوریتم تشخیص تقلب می‌تواند آموزش داده شود تا معاملات را بر اساس ویژگی‌های آن‌ها به کلاس‌های قانونی یا تقلبی طبقه‌بندی کند.

تشخیص پزشکی: تشخیص پزشکی برای شناسایی بیماری‌ها یا شرایط بیمار بر اساس علائم و سوابق پزشکی او استفاده می‌شود. به عنوان مثال، یک الگوریتم تشخیص پزشکی می‌تواند آموزش داده شود تا بیماران را بر اساس علائم و سابقه پزشکی به عنوان مبتلا به یک بیماری خاص یا غیر مبتلا طبقه‌بندی کند.

الگوریتم‌های طبقه بندی در ماشین لرنینگ

الگوریتم‌های مختلفی برای طبقه‌بندی در یادگیری ماشین وجود دارد که هر کدام نقاط قوت و ضعف خاص خود را دارند. در این‌جا برخی از الگوریتم‌های طبقه بندی محبوب را معرفی می‌کنیم:

بیز ساده یا **Naïve Bayes**: یک الگوریتم احتمالی است که فرض می‌کند ویژگی‌ها با توجه به برچسب کلاس مستقل از هم هستند. این الگوریتم، ساده و سریع است و با مجموعه داده‌های با ابعاد بالا به خوبی کار می کند.

پیشنهاد می‌کنیم درباره الگوریتم بیز ساده هم مطالعه کنید.

درخت تصمیم: درخت تصمیم یک الگوریتم محبوب برای طبقه‌بندی هستند که می‌تواند داده‌های طبقه‌بندی (Categorical) و عددی را مدیریت کند. تفسیر آن آسان است و می‌تواند روابط غیر خطی بین ویژگی‌ها را مدیریت کند.

جنگل تصادفی: جنگل‌های تصادفی مجموعه‌ای از درخت‌های تصمیم هستند که در آن هر درخت بر روی زیرمجموعه‌ای تصادفی از ویژگی‌ها و داده‌ها آموزش داده می‌شود. آن‌ها قوی هستند، روی مجموعه داده‌های مختلف عملکرد خوبی دارند و می‌توانند مقادیر از دست رفته و داده‌های پر نویز را مدیریت کنند.

ماشین‌های بردار پشتیبان (**SVM**): یک الگوریتم محبوب برای طبقه‌بندی باینری است که با پیدا کردن یک هایپرپلین که کلاس‌ها را به بهترین شکل از هم جدا می‌کند، کار می‌کند.

پیشنهاد می‌کنیم درباره الگوریتم ماشین بردار پشتیبان هم مطالعه کنید.

K-Nearest Neighbors (KNN): یک الگوریتم غیرپارامتریک است که با یافتن k نزدیک‌ترین داده آموزشی به یک نمونه آزمایشی کار می‌کند و کلاسی را به آن اختصاص می‌دهد که بیشتر در میان k همسایه‌هایش ظاهر می‌شود. این الگوریتم ساده و انعطاف پذیر است و می‌تواند طبقه‌بندی چند کلاسه را انجام دهد.

پیشنهاد می‌کنیم درباره الگوریتم K نزدیک ترین همسایه هم مطالعه کنید.

شبکه‌های عصبی: شبکه‌های عصبی الگوریتمی قدرتمند برای طبقه بندی هستند که می‌توانند روابط پیچیده و غیرخطی بین ویژگی‌ها را مدیریت کنند. آن‌ها از چندین لایه از گره‌های به هم پیوسته تشکیل شده‌اند که هر کدام یک تبدیل غیرخطی را در ورودی انجام می‌دهند.

انتخاب الگوریتم به مسئله خاص در دست بررسی، ماهیت داده‌ها و منابع محاسباتی موجود بستگی دارد. اغلب توصیه می‌شود قبل از انتخاب بهترین الگوریتم، چندین الگوریتم را امتحان کنید و عملکرد آن‌ها را در یک مجموعه اعتبارسنجی مقایسه کنید.

خلاصه مطالب

طبقه‌بندی یا Classification یک تسک مهم در ماشین لرنینگ است که کاربردهای متعددی در زمینه‌های مختلف دارد. الگوریتم‌های طبقه‌بندی با یادگیری یک مرز تصمیم‌گیری که کلاس‌های مختلف را در فضای ویژگی‌ها از هم جدا می‌کند، می‌توانند کلاس ورودی‌های جدید و دیده نشده را بر اساس ویژگی‌هایشان پیش‌بینی کنند. با کمک الگوریتم‌های طبقه‌بندی، می‌توانیم فرآیندها را خودکار کنیم، پیش‌بینی کنیم و تصمیم‌گیری را در حوزه‌های مختلف بهبود دهیم.

الگوریتم های یادگیری ماشین

یادگیری ماشین شاخه‌ای از هوش مصنوعی و علوم کامپیوتر است که برای یادگیری انجام امور بر استفاده از داده‌ها و الگوریتم‌ها تمرکز دارد و به تدریج با گذر زمان دقت خود را بیشتر می‌کند. الگوریتم های یادگیری ماشین در بسیاری از برنامه‌ها و ابزارهایی که روزانه از آن‌ها استفاده می‌کنیم وجود دارند؛ مانند موتور جست‌وجوی گوگل. یکی از دلایل قدرت روزافزون این موتور جست‌وجو قدرت یادگیری رتبه‌بندی صفحات است که بدون برنامه‌های از پیش نوشته‌شده انجام می‌شود. این الگوریتم‌ها برای اهداف مختلفی مانند داده‌کاوی، تجزیه‌وتحلیل، پردازش تصویر و پیش‌بینی استفاده می‌شوند. مزیت اصلی استفاده از یادگیری ماشین این است که یک‌بار به ماشین آموزش می‌دهیم که چه کاری انجام دهد؛ سپس ماشین پردازش اطلاعات و انجام امور را به‌صورت اتوماتیک پیش خواهد برد.

چند نوع الگوریتم یادگیری ماشین داریم؟

به زبان ساده، الگوریتم های ماشین لرنینگ مانند دستورالعملی هستند که به رایانه‌ها اجازه می‌دهند انجام امور را یاد بگیرند و از داده‌ها و تحلیل آن‌ها به‌منظور پیش‌بینی استفاده کنند. به جای اینکه صریحا به رایانه بگوییم چه کاری انجام دهد، مقدار زیادی داده در اختیار آن قرار می‌دهیم تا الگوها و روابط را کشف کند. در حال حاضر سه نوع از انواع الگوریتم های یادگیری ماشین داریم که عبارتند از:

یادگیری تحت نظارت

یادگیری تحت نظارت نوعی از الگوریتم های یادگیری ماشین است که در آن از مجموعه داده‌های برچسب‌گذاری شده برای آموزش مدل استفاده می‌کنیم. هدف این الگوریتم یادگیری تشخیص الگوها در میان داده‌های ورودی است که به آن امکان می‌دهد پیش‌بینی‌ها یا طبقه‌بندی‌هایی را روی داده‌های جدید انجام دهد. این نوع شامل دو الگوریتم Regression و Classification می‌شود.

یادگیری بدون نظارت

یادگیری بدون نظارت نوعی از الگوریتم های ماشین لرنینگ است که در آن الگوریتم‌ها برای یافتن الگوها، ساختار یا رابطه درون مجموعه‌ای از اطلاعات، از داده‌های انبوه و بدون علامت استفاده می‌کنند. در این الگوریتم‌ها، ماشین با استفاده از تحلیل داده‌های بدون دسته یا برچسب‌های از پیش تعریف‌شده پیش‌بینی و عملکرد را بررسی می‌کند. این نوع الگوریتم یادگیری ماشین شامل دو نوع Clustering و Dimensionality Reduction نیز می‌شود.

یادگیری تقویتی

یادگیری تقویتی نوعی از الگوریتم های یادگیری ماشین است که در آن یک عامل یاد می‌گیرد با تعامل با محیط اطراف خود تصمیمات صحیح بگیرد. هدف عامل کشف تاکتیک‌های بهینه است تا در طول زمان از طریق آزمون‌وخطا پاداش‌ها را به حداکثر برساند. یادگیری تقویتی اغلب در سناریوهایی به کار می‌رود که در آن عامل باید یاد بگیرد که چگونه در یک محیط حرکت کند، بازی را انجام دهد، ربات‌ها را مدیریت یا در موقعیت‌های نامشخص قضاوت کند.

معروف ترین الگوریتم های یادگیری ماشین

معروف ترین الگوریتم های یادگیری ماشین از انواع یادگیری تحت نظارت، بدون نظارت و یادگیری تقویتی هستند که در ادامه به آن‌ها اشاره خواهیم کرد.

1.Linear Regression

این الگوریتم در زبان فارسی با نام «رگرسیون خطی» یا «پیش‌بینی خطی» خوانده می‌شود. از رگرسیون خطی برای پیش‌بینی مقادیر پیوسته مانند قیمت خانه، فروش یا حقوق استفاده می‌شود. Linear Regression توسط یافتن یک رابطه خطی بین متغیر وابسته (چیزی که می‌خواهید پیش‌بینی کنید) و یک یا چند متغیر مستقل (چیزهایی که می‌توانند بر متغیر وابسته تاثیر بگذارند) کار خود را انجام می‌دهد.

2.Logistic Regression

رگرسیون لجستیک نوعی الگوریتم Supervised Learning است که می‌تواند برای پیش‌بینی نتایج باینری استفاده شود، مانند اینکه آیا مشتری محصولی را می‌خرد یا نه، آیا فرد مبتلا به بیماری است یا نه، یا اینکه دانش‌آموز نمره قبولی را اخذ می‌کند یا نه.

رگرسیون لجستیک تصمیم‌گیری را بر اساس یافتن رابطه بین متغیر وابسته (نتیجه باینری که می‌خواهید پیش‌بینی کنید) و یک یا چند متغیر مستقل (چیزهایی که می‌توانند بر متغیر وابسته تاثیر بگذارند) انجام می‌دهد.

3.KNN (K-nearest Neighbour)

در این الگوریتم تصمیم‌گیری مدل بر اساس همسایه‌های نزدیک نقطه جدید است. K در واقع مجموعه‌ای از نزدیک‌ترین همسایه‌ها هستند که می‌توانند به مدل کمک کنند نقطه جدید را برچسب‌گذاری کند و آن را در گروه خاصی قرار دهد. در واقع مدل یاد می‌گیرد که با نگاه کردن به همسایه‌های نقطه جدید، ویژگی‌های آن را تشخیص دهد.

این مثال ساده در دنیای واقعی می‌تواند نحوه عملکرد الگوریتم KNN را نشان دهد. یک شرکت می‌خواهد سیستم توصیه محصول برای وب‌سایت خود طراحی کند. این شرکت مجموعه داده‌ای از تاریخچه خرید مشتری و اطلاعات محصول دارد. حال می‌تواند از این مجموعه داده برای آموزش یک مدل KNN استفاده کند. این مدل رابطه بین محصولاتی را که مشتریان در گذشته خریداری کرده‌اند و محصولاتی که احتمالا در آینده خواهند خرید را یاد می‌گیرد. پس از آموزش مدل، شرکت می‌تواند از آن برای توصیه محصولات به مشتریان جدید بر اساس سابقه خرید آن‌ها استفاده کند.

KNN یک الگوریتم ساده و همه کاره است که می‌تواند برای کارهای طبقه‌بندی و پیش‌بینی استفاده شود. این الگوریتم یادگیری ماشین اغلب برای دسته‌بندی تصاویر، متن و تشخیص تقلب استفاده می‌شود. KNN بهترین الگوریتم برای خوشه‌بندی مقادیر است.

4.K-Means Clustering

خوشه‌بندی K-Means یک رویکرد یادگیری بدون نظارت است که می‌تواند برای گروه‌بندی داده‌ها در خوشه‌ها استفاده شود. این الگوریتم مشابه KNN است؛ زیرا همچنان از روش نزدیک‌ترین همسایه و گروه‌بندی همسایه‌ها با یکدیگر استفاده می‌کند. خوشه‌ها به گروهی از نقاط داده می‌گویند که مشابه یکدیگر و با نقاط داده در سایر خوشه‌ها متفاوت هستند. خوشه‌بندی K-Means کار خود را با انتخاب مرکز خوشه، k شروع می‌کند. سپس به‌صورت تصادفی هر نقطه داده را به یکی از خوشه‌های k اختصاص می‌دهد. پس از آن مرکز هر خوشه را محاسبه می‌کند. هنگامی که مرکزها محاسبه شدند، K-Means Clustering هر نقطه داده را به خوشه‌ای که نزدیک‌ترین مرکز را دارد، اختصاص می‌دهد. این فرآیند تا زمانی تکرار می‌شود که هیچ نقطه داده‌ای باقی نماند. در این الگوریتم نقاط داده در هر خوشه تا حد امکان شبیه یکدیگر و در عین حال تا حد ممکن از نقاط داده در سایر خوشه‌ها متمایز هستند.

5.Decision Tree

درخت تصمیم نوعی تکنیک یادگیری با نظارت (Supervised Learning) است که برای طبقه‌بندی و همچنین رگرسیون استفاده می‌شود. این الگوریتم با تقسیم داده‌ها به گروه‌های کوچک‌تر تصمیم می‌گیرد؛ تا زمانی که دیگر داده‌ای وجود نداشته باشد. در واقع درخت تصمیم یک ساختار درخت‌مانند است که با یک گره ریشه شروع و به گره‌های فرزند منشعب می‌شود. هر گره یک شرط را بررسی می‌کند و برگ‌های درخت در Decision Tree نتایج ممکن هستند

6.Random Forest

یکی از مشکلات درخت تصمیم دشواری آن در تعمیم یک مسئله است. برای حل این مشکل، نوع جدیدی از الگوریتم درخت تصمیم با جمع‌آوری درخت‌های متعدد ایجاد شد. در این الگوریتم که جنگل تصادفی نام دارد، تصمیم‌گیری درباره بهترین نتیجه

با استفاده از یک سیستم رای‌گیری یا میانگین‌گیری از هر گروه انجام می‌شود. جنگل تصادفی نوعی روش یادگیری گروهی است که در آن درختان برای تصمیم‌گیری و پیش‌بینی با یکدیگر همکاری می‌کنند.

7.Naive Bayes

بیز ساده یک الگوریتم ماشین لرنینگ بر اساس قضیه بیز است که برای دسته‌بندی استفاده می‌شود. این الگوریتم با فرض اینکه ویژگی‌های یک نقطه داده مستقل از یکدیگر هستند کار می‌کند. عملکرد Naive Bayes به این صورت است که احتمال یک نقطه داده متعلق به یک کلاس خاص را محاسبه و بر اساس احتمال هر یک از ویژگی‌های نقطه داده متعلق به آن کلاس کار می‌کند. برای درک این موضوع مبحث را با یک مثال پیش می‌بریم.

تصور کنید یک کیسه میوه دارید و می‌خواهید بدانید که آیا این کیسه حاوی سیب است یا پرتقال. می‌توانید از الگوریتم Naive Bayes برای پیش‌بینی این موضوع با محاسبه احتمال هر یک از ویژگی‌های میوه (به‌عنوان مثال، رنگ، شکل، اندازه) متعلق به هر کلاس (سیب یا پرتقال) استفاده کنید. برای مثال، می‌دانید که سیب‌ها بیشتر گرد هستند و رنگ قرمز دارند، در حالی‌که پرتقال‌ها به احتمال زیاد نارنجی‌رنگ و بیضی‌شکل هستند. می‌توانید از این اطلاعات برای محاسبه احتمال سیب یا پرتقال بودن میوه موجود در کیسه خود استفاده کنید. بیز ساده به تمام ویژگی‌ها به‌صورت مستقل نگاه می‌کند؛ اما در نهایت آن‌ها را به‌منظور کشف گروهی که متغیر به آن تعلق دارد، با یکدیگر ترکیب می‌کند.

8.SVM (Support Vector Machine)

الگوریتم ماشین بردار پشتیبان یکی از الگوریتم‌های مفید برای دسته‌بندی و پیش‌بینی وظایف است؛ حتی زمانی که با حجم کمی از داده‌ها روبه‌رو هستیم. SVMها یکHyperplane پیدا می‌کنند که نقاط داده را در یک مجموعه به دو قسمت تقسیم می‌کند. هدف SVM ایجاد بهترین خط یا مرز تصمیم است که بتواند فضای n بعدی را به کلاس‌ها تفکیک کند تا بتوانیم به‌راحتی نقطه داده جدید را در دسته‌بندی صحیح قرار دهیم. این مرز بهترین تصمیم، ابرصفحه (Hyperplane) نامیده می‌شود. تصور کنید مجموعه داده‌ای از تصاویر گربه‌ها و سگ‌ها داریم. ما ابتدا مدل خود را با مجموعه تصاویر این دو حیوان آموزش می‌دهیم تا بتواند با ویژگی‌های مختلف گربه‌ها و سگ‌ها آشنا شود. الگوریتم SVM خطی (هایپرپلن) را در مجموعه داده پیدا می‌کند که به واسطه آن تصاویر گربه از تصاویر سگ جدا می‌شوند. حال مدل با مشاهده تصاویر هر گروه تشخیص می‌دهد که عکس جدید گربه است یا سگ.

9.Apriori

Apriori جزو الگوریتم‌های Rule Based است و به‌منظور یافتن ویژگی‌های آیتم‌های مکرر در یک مجموعه داده به‌کار می‌رود. این الگوریتم اغلب برای تحلیل سبد خرید مشتریان استفاده می‌شود؛ جایی که هدف یافتن ارتباط بین مواردی است که اغلب با هم خریداری می‌شوند. Apriori با ساخت مکرر مجموعه آیتم‌های بزرگ‌تر از مجموعه آیتم‌های کوچک‌تر کار می‌کند.

شروع کار پیش‌بینی الگوها با کوچک‌ترین مجموعه‌های آیتم‌های ممکن، که آیتم‌های مفرد هستند، انجام می‌شود. سپس، الگوریتم مجموعه آیتم‌ها را با یکدیگر ترکیب می‌کند تا مجموعه بزرگتری را تشکیل دهد. این روند تا زمانی ادامه می‌یابد که به آستانه اندازه معینی برسد یا دیگر نتواند مجموعه آیتم‌های مکرری را پیدا کند.

10.PCA (Principal Component Analysis)

الگوریتم یادگیری ماشین PCA به‌منظور کاهش ابعاد یک مجموعه و سادگی پردازش اطلاعات استفاده می‌شود. کاهش ابعاد، فرآیند کاهش تعداد ویژگی‌های یک مجموعه داده بدون از دست دادن اطلاعات زیاد است. این الگوریتم اجزای اصلی یک مجموعه داده را نگه می‌دارد و ویژگی‌های جدیدی که با این مجموعه همبستگی ندارند را حذف می‌کند. این حذف به‌گونه‌ای انجام می‌شود که تا آنجا که ممکن است واریانس داده‌ها حفظ شوند. به‌عنوان مثال، الگوریتم PCA ممکن است متوجه شود که قد و وزن حیوانات با هم ارتباط زیادی دارند. این بدان معناست که ما می‌توانیم از یک ویژگی واحد مانند قد برای نشان دادن قد و وزن استفاده کنیم.

الگوریتم درخت تصمیم

درخت تصمیم (**Decision Tree**) نوعی یادگیری ماشین نظارت‌شده (Supervised Machine Learning) است که برای طبقه‌بندی یا پیش‌بینی بر اساس پاسخ سؤالات قبلی استفاده می‌شود. این مدل، شکلی از یادگیری نظارت‌شده است؛ به این معنا که آموزش و آزمایش مدل بر روی مجموعه‌داده‌ای که شامل طبقه‌بندی موردنظر است، انجام می‌شود. ممکن است این مدل همیشه نتواند پاسخ قطعی و روشنی ارائه دهد. در عوض، گزینه‌هایی را در اختیار دانشمندان داده قرار می‌دهد تا بتوانند بر اساس آن‌ها تصمیماتی آگاهانه بگیرند. درخت‌های تصمیم از تفکر انسانی تقلید می‌کنند. بنابراین متخصصین داده معمولاً به‌راحتی می‌توانند نتایج را متوجه شده و تفسیر کنند.

عملکرد درخت تصمیم چگونه است؟

قبل از توضیح نحوه‌ی عملکرد، بیایید برخی اصطلاحات مربوط به آن را تعریف کنیم:

- گره ریشه (**Root Node**): پایه‌ی درخت تصمیم است.
- تقسیم (**Splitting**): فرایند تقسیم یک گره به چندین زیرگره را می‌گویند.
- گره تصمیم (**Decision Node**): زمانی که یک زیرگره به زیرگره‌های بیشتری تقسیم می‌شود، به آن گره‌ی تصمیم می‌گویند.
- گره برگ (**Leaf Node**): زمانی که یک زیرگره به زیرگره‌های بیشتری تقسیم نمی‌شود و در واقع نشان‌دهنده‌ی خروجی احتمالی است، به آن گره‌ی برگ می‌گویند.
- هرس (**Pruning**): فرایند حذف زیرگره‌های یک درخت تصمیم را می‌گویند.
- شاخه (**Branch**): زیرمجموعه‌ای از درخت تصمیم است که از چندین گره تشکیل شده است.

درخت تصمیم‌گیری بسیار شبیه درخت معمولی است. در ابتدای درخت، گره‌ی ریشه قرار دارد. مجموعه‌ای از گره‌های تصمیم از گره ریشه منشعب می‌شوند که نشان‌دهنده‌ی تصمیماتی هستند که باید گرفته شوند. از گره‌های تصمیم به گره‌های برگ می‌رسیم که نشان‌دهنده‌ی نتایج آن تصمیمات هستند. هر گره تصمیم نشان‌دهنده‌ی یک سؤال یا نقطه‌ی انشعاب است و گره‌های برگی که از یک گره تصمیم منشعب می‌شوند، نشان‌دهنده‌ی پاسخ‌های ممکن هستند. درست مانند رشد برگ روی شاخه، گره‌های برگ نیز از گره‌های تصمیم ایجاد می‌شوند. به همین دلیل است که به زیرمجموعه‌های این الگوریتم شاخه می‌گوییم.

انواع درخت تصمیم چیست؟

انواع اصلی درخت‌های تصمیم‌گیری عبارت‌اند از: درخت تصمیم با متغیر گسسته (Categorical Variable Decision Tree) و درخت تصمیم با متغیر پیوسته (Continuous Variable Decision Tree) که بر اساس نوع متغیر خروجیِ مورداستفاده ایجاد شده‌اند.

درخت تصمیم با متغیر گسسته: در این مدل، جواب به یک طبقه‌بندی خاص نزدیک است. سکه شیر است یا خط؟ حیوان خزنده است یا پستاندار؟ در این نوع درخت تصمیم‌گیری، داده‌ها بر اساس تصمیماتی که در گره‌های درخت گرفته شده‌اند، در یک طبقه‌بندی خاص قرار می‌گیرند.

درخت تصمیم با متغیر پیوسته: در این مدل، یک جواب بله یا خیر مشخص وجود ندارد. به این نوع درخت، درخت رگرسیونی هم گفته می‌شود زیرا متغیر خروجی یا همان تصمیم گرفته‌شده به تصمیمات قبلی بستگی دارد. مزیت درخت تصمیم‌گیری با متغیر پیوسته این است که می‌توان خروجی را بر اساس چندین متغیر پیش‌بینی کرد. اما در مدل با متغیر گسسته، پیش‌بینی تنها بر اساس یک متغیر انجام می‌شود. در درخت تصمیم‌گیری با متغیر پیوسته، با انتخاب الگوریتم صحیح می‌توان از هر دو روابط خطی و غیرخطی استفاده کرد.

مهم‌ترین الگوریتم‌های درخت تصمیم

ID3

الگوریتم ID3 (Iterative Dichotomiser 3) یکی از اولین الگوریتم هایی است که برای ساخت درخت تصمیم‌گیری ارائه شده است. این الگوریتم از معیار اطلاعات یا Information Gain برای انتخاب ویژگی ها برای تقسیم داده ها استفاده می کند.

C4.5

4.5C نسخه ای به روز شده از 3ID است. در مقابل 3ID که فقط با ویژگی های گسسته کار می کند، 4.5C می تواند با ویژگی های گسسته و پیوسته کار کند. علاوه بر این، 4.5C از معیار Gain Ratio که نسبت Information Gain به انتروپی ویژگی است، برای انتخاب ویژگی ها استفاده می کند.

CART

CART (Classification and Regression Trees) یک الگوریتم دیگر برای ساخت درخت تصمیم‌گیری است که برای مسائل طبقه بندی و رگرسیون قابل استفاده است. CART از معیار Gini Impurity برای انتخاب ویژگی ها استفاده می کند و درخت های باینری (دو تایی) می سازد.

CHID

CHID (Chi-square Automatic Interaction Detector) الگوریتمی است که از آزمون آماری چی دوم (Chi-square) برای ارزیابی ویژگی ها و انتخاب بهترین ویژگی برای تقسیم داده ها استفاده می کند.

اجزای درخت تصمیم

درخت‌های تصمیم می‌توانند با داده‌های پیچیده سروکار داشته باشند. با این حال، این جمله بدان معنا نیست که درک عملکرد این الگوریتم دشوار است. تمام درختان تصمیم در هسته خود، از چهار بخش کلیدی تشکیل شده‌اند:

1. گره ریشه

گره ریشه گره بالای درخت است که نقطه شروع فرآیند تصمیم‌گیری را نشان می‌دهد. این گره حاوی ویژگی است که آن را تبدیل به مهم‌ترین گره برای پیش‌بینی متغیر هدف می‌کند.

2. گره‌های داخلی

گره‌های داخلی گره‌هایی حاوی گره فرزند هستند. آن‌ها مراحل میانی در فرآیند تصمیم‌گیری را نشان می‌دهند. هر گره داخلی حاوی یک قانون تصمیم‌گیری است که داده‌ها را به دو یا چند شاخه تقسیم می‌کند. گره‌های داخلی شامل سه گره متداول می‌شوند که موارد زیر را در برمی‌گیرند:

- گره‌های تصمیم (**Decision nodes):** یک تصمیم را نشان می‌دهند (معمولا با مربع نشان داده می‌شود).
- گره‌های شانس (**Chance nodes):** نشان‌دهنده احتمال یا عدم قطعیت هستند (معمولا این گره‌ها را با یک دایره نشان می‌دهیم).
- گره‌های پایانی (**End nodes):** گره‌های پایانی یک نتیجه را در معرض دید قرار می‌دهند (معمولا با یک مثلث مشخص می‌شوند).

اتصال این گره‌های مختلف همان چیزی است که ما آن را «شاخه» (Branch) می‌نامیم. گره‌ها و شاخه‌ها را می‌توان بارهاوبارها در هر تعداد ترکیب برای ایجاد درختان با پیچیدگی‌های مختلف استفاده کرد.

3. شاخه‌ها

شاخه‌ها خطوطی هستند که گره‌ها را به یکدیگر متصل می‌کنند. آن‌ها نتایج احتمالی یک تصمیم را نشان می‌دهند. هر شاخه به یک گره فرزند منتهی می‌شود.

4. گره‌های برگ

گره‌های برگ، گره‌هایی هستند که هیچ گره فرزندی ندارند. آن‌ها نشان‌دهنده نتیجه نهایی فرآیند تصمیم‌گیری هستند. هر گره برگ حاوی یک پیش‌بینی برای متغیر هدف است.

نحوه هرس درخت تصمیم

گاهی اوقات درختان تصمیم می‌توانند بسیار پیچیده رشد کنند. در این موارد، آن‌ها معمولا به داده‌های نامربوط وزن زیادی می‌دهند. این گره‌ها مانع از رشد درخت به سمت عمق می‌شوند. برای جلوگیری از این مشکل، می‌توانیم گره‌های خاصی را با استفاده از فرآیندی به نام «هرس» حذف کنیم. هرس دقیقا همان چیزی است که به‌نظر می‌رسد: اگر درخت شاخه‌هایی را رشد دهد که به آن‌ها نیاز نداریم، باید به‌سادگی قطعشان کنیم. افزایش شاخه‌های بدون استفاده را با نام «بیش‌برازش» یا "Overfitting" می‌شناسیم. درست مانند هر الگوریتم یادگیری ماشین دیگری، آزاردهنده‌ترین اتفاقی که می‌تواند بیفتد، مشکل بیش‌برازش است. درخت تصمیم به وفور با مشکل بیش‌برازش روبه‌رو می‌شود.

دو نوع هرس Decision Tree وجود دارد: 1) قبل از هرس (Pre-pruning) و 2) پس از هرس (Post-pruning). در ادامه هر دو نوع را تشریح خواهیم کرد.

پیش هرس درخت تصمیم

پیش هرس درخت تصمیم تکنیکی برای جلوگیری از رشد بیش از حد این الگوریتم است. Decision Tree با عمق خیلی زیاد می‌تواند به خطر بیش‌برازش دچار شود؛ به این معنی که داده‌های آموزشی را به‌درستی یاد گرفته است و به‌خوبی به داده‌های جدید تعمیم نمی‌دهد. این مرحله به «توقف اولیه» مشهور است که رشد درخت تصمیم را متوقف می‌کند و مانع از رسیدن آن به عمق کامل می‌شود.

پیش هرس فرآیند درخت‌سازی را متوقف می‌کند تا از تولید برگ با نمونه‌های کوچک جلوگیری شود. در طول هر مرحله از تقسیم درخت، خطای اعتبارسنجی متقاطع پایش می‌شود. اگر مقدار خطا دیگر کاهش نیابد، رشد درخت را متوقف می‌کنیم.

هایپرپارامترهایی (Hyperparameters) که می‌توان برای توقف زودهنگام و جلوگیری از بیش‌برازش تنظیم کرد عبارتند از:

max_depth, min_samples_leaf, min_samples_split

از همین پارامترها هم می‌توان برای تنظیم کردن یک مدل قوی استفاده کرد. با این حال، باید محتاط باشید؛ زیرا توقف زودهنگام می‌تواند منجر به عدم تناسب در مدل و وقوع مشکل کم‌برازش (Underfitting) شود.

پیش هرس درخت تصمیم به دو شیوه اصلی قابل پیاده‌سازی است:

1. حداکثر عمق را برای درخت تنظیم کنید. این به این معنی است که درخت اجازه نخواهد داشت هیچ شاخه‌ای عمیق‌تر از یک سطح خاص داشته باشد.
2. حداقل تعداد نقاط داده را تنظیم کنید که باید قبل از تقسیم شدن در یک گره باشند. در این حالت درخت اجازه ندارد یک گره را تقسیم کند؛ مگر اینکه حداقل تعداد معینی از نقاط داده در آن باشد.

پیش هرس می‌تواند به بهبود دقت درخت تصمیم با جلوگیری از بیش‌برازش آن کمک کند. همچنین تفسیر درخت در مرحله پیش هرس آسان‌تر است؛ زیرا کوچک‌تر و کمتر پیچیده خواهد بود.

پس هرس درخت تصمیم

پس هرس درخت تصمیم برعکس پیش هرس عمل می‌کند و به مدل اجازه می‌دهد که تا سطح عمیق و کامل خود رشد کند. هنگامی که مدل رشد کرد و به عمق کامل خود رسید، شاخه‌های درخت برداشته می‌شوند تا از احتمال بیش‌برازش مدل جلوگیری شود.

الگوریتم به تقسیم‌بندی داده‌ها به زیرمجموعه‌های کوچک‌تر ادامه می‌دهد تا زمانی که زیرمجموعه‌های نهایی تولیدشده از نظر متغیر نتیجه مشابه باشند. زیرمجموعه نهایی درخت فقط از چند نقطه داده تشکیل شده است که به درخت اجازه می‌دهد تا داده‌ها را به شکل نمودار T یاد بگیرد. با این حال، وقتی یک نقطه داده جدید معرفی می‌شود که با داده‌های آموخته‌شده متفاوت است، احتمال خطا در پیش‌بینی نتیجه به‌وجود خواهد آمد.

هایپرپارامتری که می‌تواند برای پس هرس درخت تصمیم و جلوگیری از بیش‌برازش تنظیم شود این است:

ccp_alpha

ccp مخفف Cost Complexity Pruning است و می‌تواند به‌عنوان گزینه دیگری برای کنترل اندازه درخت استفاده شود. مقدار بالاتر ccp_alpha منجر به افزایش تعداد گره‌های هرس‌شده می‌شود.

مزایا و معایب درخت تصمیم چیست؟درخت تصمیم‌گیری نمایی از روابط علت و معلولی است که می‌تواند تصویری ساده از فرایندهای پیچیده ارائه می‌دهد. این مدل به‌راحتی می‌تواند روابط غیرخطی را ترسیم کرده و برای مسائل گسسته و رگرسیونی راه‌حل ارائه کند. با درخت تصمیم می‌توان میزان ریسک، اهداف و مزایا را مشخص کرد.

از آن‌جا که ساختار درخت تصمیم‌گیری یک فلوچارت ساده است، یکی از سریع‌ترین روش‌ها برای شناسایی متغیرهای تأثیرگذار و روابط بین دو یا چند متغیر محسوب می‌شود. اگر یک دانشمند داده روی مسئله‌ای با چندصد متغیر کار می‌کند، این مدل می‌تواند به او کمک کند تا تأثیرگذارترین آن‌ها را شناسایی کند. از آنجایی که خروجی به‌صورت بصری است، به‌راحتی می‌توان رابطه‌ی بین متغیرها را مشاهده کرد. بنابراین برای درک درخت‌های تصمیم به دانش آماری چندانی احتیاج نیست و کسانی که پیشینه‌ی تحلیلی ندارند نیز به‌راحتی می‌توانند آن را درک کنند. با همه‌ی این‌ها گاهی درخت تصمیم محدودیت‌هایی دارد. آگاهی از مزایا و معایب آن می‌تواند به شما کمک کند تا تشخیص دهید که برای چه مواردی بهتر است از آن‌ها استفاده کنید.

مزایا:

- برای داده‌ها و متغیرهای گسسته و یا عددی به خوبی کار می‌کند.
- مسائل با چندین خروجی را مدل‌سازی می‌کند.
- نسبت به سایر روش‌های مدل‌سازی داده، به پیش‌پردازش کمتری برای داده‌های ورودی نیاز دارد.
- به‌راحتی می‌توان آن را برای کسانی که پیشینه‌ی تحلیلی ندارند، شرح داد.

معایب:

- تحت‌تأثیر نویز در داده‌ها قرار می‌گیرد.
- برای مجموعه‌داده‌های بزرگ ایدئال نیست.
- می‌تواند ویژگی‌ها را به‌طور نامتناسبی ارزش‌گذاری کند.
- از آنجایی که تصمیم‌ها در گره‌ها محدود به خروجی‌های باینری هستند، نمی‌تواند پیچیدگی‌های زیاد را مدیریت کند.
- زمانی که با عدم قطعیت و خروجی‌های زیادی سروکار داریم، درخت تصمیم می‌تواند خیلی پیچیده شود.

اجرای الگوریتم درخت تصمیم

کد های برنامه

```
import numpy as np
from numpy import random
from sklearn.neighbors import KNeighborsClassifier
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import MinMaxScaler
from sklearn.tree import DecisionTreeClassifier
from sklearn.datasets import load_breast_cancer

a = load_breast_cancer()

x = a['data']

y = a['target']

scl = MinMaxScaler()

x = scl.fit_transform(x)

xtrain,xtest,ytrain,ytest,=train_test_split(x,y,test_size=0.2,rando
m_state=42)

knn = KNeighborsClassifier(n_neighbors=4)

knn.fit(xtrain,ytrain)

p = knn.predict(xtest)

np.mean(p==ytest)

knn.score(xtest,ytest)

dt = DecisionTreeClassifier()
```

```
dt.fit(xtrain,ytrain)

k = dt.predict(xtest)

np.mean(k==ytest)

dt.score(xtest,ytest)

0.9298245614035088
```

تویضحات درباره کدها

1. آماده‌سازی داده‌ها (Data Preparation):
   - ابتدا از مجموعه داده سرطان پستان (Breast Cancer) استفاده می‌کنید. این مجموعه داده شامل ویژگی‌های مختلف مرتبط با سلول‌های سرطانی و بیماری سرطان پستان است.
   - برای مقیاس‌بندی ویژگی‌ها از MinMaxScaler استفاده می‌کنید. این کار باعث می‌شود که ویژگی‌ها در بازه‌ی [0، 1] قرار گیرند.
2. الگوریتم K-Nearest Neighbors (KNN):
   - ابتدا یک مدل KNN با 4 همسایه ایجاد می‌کنید.
   - سپس مدل را روی داده‌های آموزشی (xtrain و ytrain) آموزش می‌دهید.
   - پس از آموزش، مدل روی داده‌های آزمون (xtest) پیش‌بینی انجام می‌دهد و نتایج را در p ذخیره می‌کنید.
   - دقت مدل با محاسبه میانگین دقت پیش‌بینی‌ها (تطابق پیش‌بینی‌ها با برچسب‌های واقعی ytest) محاسبه می‌شود.
3. الگوریتم درخت تصمیم (Decision Tree):
   - یک مدل درخت تصمیم ایجاد می‌کنید.
   - مدل را روی داده‌های آموزشی آموزش می‌دهید.
   - پس از آموزش، مدل روی داده‌های آزمون پیش‌بینی انجام می‌دهد و نتایج را در k ذخیره می‌کنید.
   - دقت مدل با محاسبه میانگین دقت پیش‌بینی‌ها (تطابق پیش‌بینی‌ها با برچسب‌های واقعی ytest) محاسبه می‌شود.

نتیجه گیری

در جمع بندی موضوعات می توان گفت که امروزه جهان دیگر به نقطه ای رسیده است که دیگر بازگشت به گذشته ممکن نیست و نیازمندی انسان به هوش مصنوعی روز به روز بیشتر خواهد شد این مسئله در کنار تهدیداتی که می تواند داشته باشد مزایا و فرصت های بیشماری را هم در اختیار جهان می گذارد.

الگوریتم های یادگیری ماشین بخش بسیار مهمی از این فرصت ها را شامل می شوند که در یکی از حوضه های پیش بینی سرطان می تواند آمار مرگ و میر بر اثر این بیماری مهلک را تا حد زیادی کاهش دهد و به تشخیص زودهنگام و درمان آن در مراحل اولیه کمک بسیار زیادی کند در این راستا الگوریتم درخت تصمیم یکی از این الگوریتم های مورد استفاده در این حوضه است.

در مواردی که اشاره شد داده های ورودی در دقت این الگوریتم نقش بسیار مهمی داشتند که کیفیت این داده ها در نتیجه بدست آمد بسیار موثر است البته که هیچ وقت نتایج به صورت صد درصد دقیق نمیباشند و همیشه امکان میزانی خطا وجود دارد. در مرحله پیش پردازش داده های که بروی نتایج بدست آمده نقش مهمی دارد در این مرحله داده های کثیف دادهای تکراری داده ها مبهم دادهای دارای خطا های ورودی حذف و سایر داده ها برای پردازش های بعدی به قالبی بهتر تبدیل می کند. پس از آن به طبقه بندی دادهای می رسیم هدف آن یک مرز تصمیم بر اساس ویژگی هاست که بر همین اساس برای جدا سازی استفاده میکند یعنی با ویژگی های که در داده ها می بیند آموزش دیده و موارد مختلف را از هم تشخیص میدهد.

به طور کلی الگوریتم درخت تصمیم نوعی یادگیری ماشین نظارت‌شده است که برای طبقه‌بندی یا پیش‌بینی استفاده می شود به شکل اموزش و آزمایش بر روی داده های طبقه بندی شده که این روش همیشه جواب درست و قطعی نمی دهد اما در عوض گزینه‌هایی را در اختیار دانشمندان داده قرار می‌دهد تا بتوانند بر اساس آن‌ها تصمیماتی آگاهانه بگیرند. درخت‌های تصمیم از تفکر انسانی تقلید می‌کنند. بنابراین متخصصین داده معمولاً به‌راحتی می‌توانند نتایج را متوجه شده و تفسیر کنند.

منابع

fa.wikipedia.org

hamrah.academy

behfalab.com

cafetadris.com

quera.org

Abstract

Today, cancer is one of the most dangerous diseases that has many types. One of the things that is very necessary and important about this disease and helps to make the treatment process easier is the discussion of predicting and diagnosing cancer before reaching the dangerous stages of the disease. For this reason, the classification of cancers is important.

To do this, machine learning is a sub-branch of artificial intelligence. It uses statistical and probabilistic techniques and optimization so that computers can learn from past examples in order to obtain patterns from complex and large data sets.

Success in machine learning is not always 100% guaranteed. If the input data is of low quality, then most likely the result will be of low quality. Figuring out which algorithm is best for solving any given problem is not clear-cut, so it is necessary to use more than one method for learning. This saves time and money.

decision tree
The decision tree is one of the first and most widely used methods in machine learning, which is widely used in solving data classification problems. In a general definition, a decision tree is a structured graph, where the process of decisions are the nodes of this graph, and the leaves represent the decisions.

In this article, respectively, cancer data in machine learning, Python programming language in artificial intelligence, data preprocessing, classification in machine learning, machine learning algorithms, decision tree algorithm, algorithm testing, accuracy and Finally, the explanation of the codes and the final conclusion are described.

Cancer diagnosis using decision tree algorithm

Name of the student: Saleh Sampanah

Teacher's name: Amin Dehghan

June 1403