دانشگاه صنعتی امیرکبیر

(پلی‌تکنیک تهران)

گزارش پروژه دوم داده کاوی

پیشبینی بارش به کمک تحلیل داده‌های اقلیمی

نگارش

امیر پارسا خادم المله

سروین باغی

استاد راهنما

دکتر شاکری

اردیبهشت ۱۴۰۳

چکیده

در این پروژه تلاش شد عملکرد ۳ مدل خواسته شده بر روی دیتاست بررسی شود، پیش پردازش های لازم برای هر کدام انجام شود و ابرپارامترهای مناسب برای هر کدام تعیین شود. در این گزارش به بررسی اجمالی کد نوشته شده می پردازیم و علل های انتخاب خود برای بخش های مختلف را شرح داده و در نهایت به بررسی نتایج می پردازیم.

فهرست مطالب

صفحه

فصل اول

مقدمه

مقدمه

در این گزارش، برای پیش‌بینی بارش به کمک تحلیل داده‌های اقلیمی از روش‌های گوناگون در مراحل مختلف کار استفاده شده است. به دلیل محدودیت اجرا در مدل‌های مختلف همه حالات تست و بررسی نشده است، اما سعی شده است تکنیک‌ها و تحلیل‌های متفاوتی انجام شود.

ابتدا به توضیح مختصری از ابزارها و مراحل انجام شده می پردازیم و در بخش های بعدی به طور دقیق به بررسی آنها می پردازیم.

هر مدل ۲ مرحله اصلی پیش پردازش و تنظیم مدل دارد. بسته به مدل استفاده شده، پیش پردازش ها می توانند با هم متفاوت باشند.

تعاریف مورد نیاز عبارت اند از:

تجزیه و تحلیل اجزای اصلی (PCA)

هدف: کاهش ابعاد مجموعه داده با حفظ بیشتر واریانس، کاهش مسائل چند خطی و بهبود سرعت آموزش مدل است

روند الگوریتم:

استانداردسازی داده ها: اطمینان حاصل می‌شود که هر ویژگی دارای میانگین ۰ و انحراف استاندارد ۱ است.

اعمال PCA: مجموعه داده را به مجموعه کوچکتری از اجزای نامرتبط تبدیل می‌کند.

انتخاب مؤلفه ها: تعداد مؤلفه های اصلی را که بخش قابل توجهی از واریانس را توضیح می دهند (مثلاً ۹۵ درصد) انتخاب می‌شوند.

تجزیه ارزش منفرد (SVD)

هدف: تکنیک دیگری برای کاهش ابعاد، ثبت مهمترین واریانس در ویژگی های کمتر است.

روند الگوریتم: ماتریس داده ها را به سه ماتریس (U، Σ و V) تجزیه می‌کند، و آن را کوتاه می‌کند تا فقط مقادیر و بردارهای تکی بالایی را حفظ کند. سپس اجزای برتر را برای کاهش ابعاد و در عین حال حفظ اطلاعات ضروری انتخاب می‌شود.

مزایای PCA و SVD:

- ساده تر شدن مدل
- بهبود سرعت و عملکرد تمرین
- با حذف نویز و ویژگی های اضافی، خطر بیش برازش را کاهش می‌دهد.

ماتریس آشفتگی

هدف: عملکرد یک مدل طبقه بندی را با مقایسه مقادیر پیش بینی شده در مقابل واقعی ارزیابی کنید.

اجزاء:

موارد مثبت واقعی (TP): موارد مثبت به درستی پیش بینی شده است.

منفی های واقعی (TN): موارد منفی که به درستی پیش بینی شده اند.

موارد مثبت کاذب (FP): موارد مثبت پیش بینی نادرست.

موارد منفی کاذب (FN): موارد منفی پیش بینی نادرست.

معیارهای به دست آمده از ماتریس آشفتگی:

Accuracy: (TP + TN) / (TP + TN + FP + FN)

Precision: TP / (TP + FP)

Recall: TP / (TP + FN)

F۱ Score: (Recall * Precision) / (Recall + Precision)

کاربرد:

- تجزیه و تحلیل دقیق عملکرد مدل را ارائه می دهد.
- به شناسایی مناطق خاصی که مدل در آنها خوب یا ضعیف عمل می کند کمک می کند.

فصل دوم

پیش پردازش اولیه

پیش پردازش اولیه

1. **بارگیری و بازرسی اولیه**: مجموعه داده از یک فایل CSV بارگیری می شود. یک ستون شاخص غیرضروری حذف می‌شود و ردیف‌های اولیه و شکل مجموعه داده برای درک ساختار آن بررسی می‌شوند.
2. **بررسی مقادیر از دست رفته و اطلاعات داده**: تعداد مقادیر از دست رفته در هر ویژگی به دست می آید.
3. **ترسیم مقادیر گمشده**: نمودار میله ای برای تجسم تعداد مقادیر گم شده در هر ستون استفاده می شود و به شناسایی ستون هایی که مقادیر قابل توجهی از داده های از دست رفته دارند کمک می کند.
4. **مدیریت مقادیر از دست رفته:** با بررسی موارد فوق در میابیم مجموعه داده مقادیر گمشده زیادی دارد و در نتیجه حذف تمامی آنها منطقی نیست. روشی ارائه می دهیم که از حذف تعداد زیاد داده جلوگیری شود. همانطور که در نمودار داخل کد قابل مشاهده است تعداد داده هایی که مقادیر گمشده بزرگتر مساوی ۶ دارند زیاد نیست و از طرفی به علت وجود تعداد زیاد گمشده حضورشان موثر نیست، پس در اولین قدم این داده ها را حذف می کنیم. با توجه به اینکه هدف نهایی پیش بینی بارش باران در روز بعد است، داده هایی که این مقدار برایشان گمشده است فاقد ارزش اند و حذف می شوند. در نهایت هم به ویژگی Rain that day می پردازیم. این ویژگی أولا مقدار ۰ یا ۱ را می تواند اختیار کند و در نتیجه استفاده از روش های مرسوم پر کردن مقادیر گمشده مثل میانگین، میانه و یا مد امکان پذیر نیست و از طرفی انتظار میرود ویژگی مهمی باشد و با توجه به دامنه محدودی که دارد پر کردن آن منطقی نیست و در نتیجه داده هایی که این ویژگی برایشان گمشده است، حذف می شود.

حال به پر کردن مقادیر گمشده باقی مانده می پردازیم. برای داده های پیوسته، ترتیبی و اسمی به ترتیب از میانگین، میانه و مد استفاده می کنیم ولی محاسبات مقادیر فوق را محدود به داده هایی می کنیم که در سال و ماه با داده مورد نظرمان اشتراک داشته باشند. توجه داشته باشید

پر کردن مقادیر مقادیر گمشده بعد از جداسازی داده های آموزشی و آزمایشی انجام می شود تا مقادیر جایگذاری شده در هر کدام مستقل از دیگری باشد.

5. تبدیل برخی ویژگی ها غیر عددی به مقادیر عددی:

1. تبدیل ویژگی Date به ۳ ویژگی سال، ماه و روز باعث می شود اولا داده ما عددی شود و ثانیا استخراج اطلاعات از آن راحتتر شود.
2. ویژگی های Rain that day و Rain that day after به شکل متغیر bool ذخیره شده اند. آنها را به فرم ۰ و ۱ در میاوریم تا عددی شوند.

فصل سوم

درخت تصمیم

درخت تصمیم

۲-۱- پیش پردازش

به غیر از پیش پردازش ها انجام شده در بخش قبل، پیش پردازش ها زیر هم برای این درخت انجام شده اند.

۲-۳-۱- حذف یا ساخت ویژگی ها:

1. حذف ویژگی با کمک ماتریس همبستگی: یک ویژگی از زوج ویژگی هایی که همبستگی بین آنها از ۰.۸ بیشتر است را حذف می کنیم.
2. ساخت ویژگی با کمک PCA: در طول الگوریتم ها از PCA استفاده می شود.

۲-۳-۲-تبدیل ویژگی های غیر عددی به عددی:

۴ تا از ویژگی ها اسمی اند و ترتیب ندارند. برای تبدیل آنها به ویژگی های عددی باید از تبدیلی استفاده شود که به داده ها ترتیب ندهد (چون ذاتا فاقد ترتیب اند). One-Hot Encoding با تبدیل هر تگ به یک ویژگی و مقداردهی ۰ و ۱ به آن تبدیل مناسبی خواهد بود.

۲-۳-۳- بالانس کردن داده ها:

با توجه به اینکه نسبت داده ها مثبت به منفی خیلی زیاد است، نمونه گیری از داده های مثبت و کوچک کردن تعداد داده های مثبت باعث می شود حجم زیادی از داده از دست برود. با نمونه گیری مداوم از داده ها منفی و بزرگ کردن آن با این روش می تواند به تکرار داده ها و کاهش دقت مدل منجر شود (انتظار می رود هر داده با برچسب منفی ۳ بار تکرار شود !!).

روش مورد استفاده SMOTE نام دارد که تلاش می کند بر مبنای داده های با جمعیت کمتر، داده های دیگری بسازد (بر اساس ترکیب خطی داده ها) و با این روش پایگاه داده را بالانس کند.

توجه کنید بالانس بعد از حذف مقادیر NaN انجام می شود و برای داده های آموزشی و آزمایشی جدا انجام می شود تا از وابستگی داده ها جلوگیری شود.

۲-۳-۴- حذف نویز:

حذف نویز از طریق الگوریتم خوشه بندی k-means انجام می شود. تعداد خوشه ها ۳ داده شده و مراکز محاسبه می شود. سپس ۲ درصد داده که بیشترین فاصله را از مراکز خود دارند حذف می شوند. توجه کنید بهتر است حذف نویز قبل بالانس انجام شود تا بالانس کردن داده سبب انتشار داده های نویز نشود.

۲-۲- توضیح مدل و کد

برای استفاده از درخت تصمیم از کتابخانه sklearn استفاده شده است و برای جلوگیری از بیش برازش ۳ روش اصلی انتخاب شد که از نتایج هر سه برای تولید درخت مناسب استفاده شد. برای استخراج پارامتر مناسب تابع decision_tree ساخته شد. در این تابع کاربر میتواند یکی از حالات زیر را انتخاب کند:

'Loop='None': در این حالت می توانید به درخت، پارامتر دلخواه بدهید: ccp_alpha مقدار خطایی است که در هرس پسین استفاده می شود، pca_number تعداد ویژگی ها را با PCA به عدد مورد نظر شما تغییر می دهد، split مینیمم تعداد داده ای که باید در هر نود باشد تا انشعاب انجام شود و depth حداکثر عمق درخت است. هر کدام که مقدار دهی نشود معادل آن است که در درخت اعمال نمی شود. خروجی کد، معیارهای Recall، Accuracy،Precision و F۱ Score به همراه ماتریس آشفتگی است.

'Loop='PCA': پارامتر pca_number به عنوان شرط حلقه در نظر گرفته می شود. یعنی به ازای تمام مقادیر از ۲ تا pca_number، PCA محاسبه می شود، درخت نظیر آن ساخته می شود، در هر حالت دقت ها برای داده های آموزشی و آزمایشی نمایش داده شده و در نهایت بهترین دقت چاپ و خروجی داده می شود.

مقادیر "'Depth' = loop و 'split' = loop هم مشابه مورد فوق عمل می کنند ولی در 'loop=Alpha'، مقادیر خطا توسط خود تابع محاسبه می شوند.

در حالت کلی ۲ مدل هرس پیشین حداکثر عمق و مینیمم داده در نود، هرس پیشین با کمک محاسبه خطا و بهینه کردن ویژگی ها با کمک PCA بررسی شد.

ابتدا درخت عادی بعد از پیش پردازش داده بررسی می شود (با کمک Loop='None')،سپس هر کدام از پارامتر های ذکر شده مستقلا با کمک پارامتر loop بررسی می شوند و بهترین هر کدام ذخیره می شود.

در نهایت بهترین های هر کدام کنار هم قرار گرفته و نتایج قبلی را بهبود می بخشند.

۲-۳- نتایج

با اجرای مکرر کد فوق، نتایج زیر بدست آمدند:

- دقت داده های آزمایشی در درخت قبل از پیش پردازش ۷۹ تا ۸۰ است که با توجه سایر معیارها در حدود ۵۰ هستند اصلا نتیجه خوبی نیست.
- دقت داده های آزمایشی در درخت فقط با پیش پردازش حدود ۸۱ تا ۸۲ بوده است، عمده تغییر آن نسبت به حالت قبل افزایش سایر معیارها به حدود ۸۰ است. شایان ذکر است که دقت داده های آموزشی ۱۰۰ بود که نشان دهنده بیش برازش است.
- دقت درخت بعد از تعیین ابرپارامترهای بهینه تا ۸۷.۹ درصد هم افزایش پیدا می کند، در حالی که دقت داده های آموزشی ۹۱ درصد است که نشان می دهد مشکل بیش برازش تا حدی حل شده است (سایر معیارها در داده های آزمایشی، حدود ۸۰ تا ۸۷ هستند)
- اگرچه اعمال PCA مستقلا بر عملکرد درخت اثر زیادی مثبتی ندارد اما در ترکیب با هرس ها اثر بسیار مثبتی دارد.
- هرس پیشین split اثر بسیار بهتری از depth دارد.
- هرس پسین عملکرد بسیار خوبی در کنار PCA دارد اما اگر زمان بسیار با ارزش باشد استفاده از هرس پیشین split به جای آن حدودا همان نتیجه را دارد.

برای دسترسی به سایر معیارها و ماتریس آشفتگی به کد مراجعه شود.

فصل چهارم

KNN و SVM

KNN و SVM

۲-۴- پیش پردازش

به غیر از پیش پردازش ها انجام شده در بخش قبل، پیش پردازش ها زیر هم برای این مدل‌ها انجام شـده اند.

۲-۳-۱- حذف یا ساخت ویژگی ها:

1. ویژگی جدید "تفاوت دما" با کم کردن دمای حداقل از حداکثر دما ایجاد می شود که به طور بالقوه یک پیش بینی مفید برای مدل ارائه می دهد.
2. حذف ویژگی با کمک ماتریس همبستگی: یک ویژگی از زوج ویژگی هایی که همبستگی بین آنها از حد آستانه تعیین شده بیشتر است را حذف می کنیم.
3. استفاده از PCA و SVD برای کاهش ابعاد داده ها با توجه به حفظ اطلاعات ضروری که سبب بهبود عملکرد مدل می شود.

۲-۳-۲-تبدیل ویژگی های غیر عددی به عددی:

استخراج بخش عددی از رشته ها: برای تبدیل متغیر های غیر عددی باقی مانده از روش Label Encoder استفاده می شود (برای این مدل نتیجه بهتری داده است)

۲-۳-۳- بالانس کردن داده ها:

در بسیاری از مجموعه داده های دنیای واقعی، کلاس ها می توانند نامتعادل باشند. برای مثال، در زمینه پیش‌بینی باران، ممکن است روزهای بدون باران در مقایسه با روزهای بارانی بسیار بیشتر باشد. این عدم تعادل می تواند باعث شود که مدل به سمت طبقه اکثریت سوگیری کند و توانایی آن در پیش بینی صحیح طبقه اقلیت (روزهای بارانی) را کاهش دهد. برای رفع این مشکل از نمونه برداری تصادفی استفاده می شود. این تکنیک شامل کاهش تصادفی تعداد نمونه ها در کلاس اکثریت (روزهای بدون باران) برای مطابقت با تعداد نمونه ها در کلاس اقلیت (روزهای همراه با باران) است. این یک مجموعه داده متعادل

ایجاد می کند که در آن هر دو کلاس دارای تعداد مساوی نمونه هستند. پس از اعمال کم‌نمونه‌سازی تصادفی، کد تعداد نمونه‌ها را در هر کلاس چاپ می‌کند تا تأیید کند که مجموعه داده متعادل است.

۲-۵- توضیح مدل و کد

KNN -۴-۳-۲

PCA تعداد ویژگی مجموعه داده را به ۱۰ (برای مثال) کاهش می دهد. داده های آموزشی (پس از نمونه گیری کم) و داده های آزمون هر دو با استفاده از این ۱۰ جزء اصلی تبدیل می شوند.

از K-Nearest Neighbors (KNN) برای طبقه بندی استفاده می شود. KNN یک الگوریتم طبقه‌بندی ساده و در عین حال مؤثر است که یک کلاس را به یک نمونه بر اساس کلاس اکثریت در میان k-نزدیک‌ترین همسایگانش اختصاص می‌دهد. طبقه بندی کننده با استفاده از داده های آموزشی تبدیل شده توسط PCA آموزش داده می شود و سپس از مدل آموزش‌دیده KNN برای پیش‌بینی برچسب‌های کلاس داده‌های آزمایش تبدیل‌شده توسط PCA استفاده می‌شود.

دقت مدل با مقایسه برچسب های پیش بینی شده با برچسب های واقعی داده های آزمون محاسبه می شود. این معیار، نسبت پیش بینی های صحیح انجام شده توسط مدل را نشان می دهد. همچنین یک گزارش نیز تولید می شود که معیارهای دیگری مانند Precision،Recall و F۱ Score را برای هر کلاس ارائه می دهد. این معیارها به ارزیابی عملکرد مدل کمک می کند.

برای درک بهتر نتیجه از بصری سازی استفاده می کنیم. بصری سازی به ۲ شکل انجام شده است:

۱. تجسم PCA: دو جزء اصلی داده های آموزشی و آزمایشی تبدیل شده توسط PCA رسم شده است. این نمودارهای پراکنده نمایشی بصری از نحوه توزیع داده ها در فضای ویژگی کاهش یافته را ارائه می دهند، با نقاط رنگ آمیزی بر اساس کلاس آنها (باران یا بدون باران).

۲. نمایش ماتریس آشفتگی: ماتریس آشفتگی با استفاده از نقشه حرارتی تجسم می شود و تفسیر آن را آسان تر می کند. این تجسم به درک انواع خاصی از خطاهای طبقه بندی که مدل انجام می‌دهد کمک

می‌کند، مانند اینکه چقدر باران را به اشتباه پیش‌بینی می‌کند یا باران را زمانی که واقعاً رخ می‌دهد پیش‌بینی نمی‌کند.

SVM -۵-۳-۲

یک ابرصفحه را پیدا می کنیم که به بهترین وجه کلاس ها را در فضای ویژگی ها از هم جدا کند. هسته های خطی مناسب برای داده های قابل جداسازی خطی، هسته چند جمله ای تعامل بین ویژگی ها را ثبت می کند، هسته تابع پایه شعاعی (RBF) روابط غیر خطی را با نگاشت داده ها به فضایی با ابعاد بالاتر مدیریت می کند و Sigmoid Kernel گاهی اوقات برای شبکه های عصبی استفاده می شود و کمتر برای SVM رایج است.

یک طبقه بندی کننده SVM با یک هسته خطی (`'kernel='linear`) مقداردهی اولیه می شود. سایر گزینه‌های هسته مانند "rbf، poly" و غیره را نیز می‌توان امتحان کرد. سپس طبقه‌بندی کننده بر روی داده‌های آموزشی تبدیل‌شده توسط PCA آموزش داده می‌شود.

مصور سازی، دقت و گزارشات مشابه PCA است.

۲-۶- نتایج

نتایج ارزیابی طبقه‌بندی‌کننده KNN را با ترکیب‌های مختلف پارامترها، به ویژه تعداد مؤلفه‌های اصلی (n_components) مورد استفاده برای PCA و تعداد همسایگان (k) در نظر گرفته شده در الگوریتم KNN ارائه می‌کنیم.

در اینجا توضیحی در مورد جدول داخل کد آورده شده است:

n_components: این ستون تعداد مولفه های اصلی مورد استفاده برای کاهش ابعاد از طریق PCA را نشان می دهد. از ۲ تا ۲۰ متغیر است.

k: این ستون تعداد همسایگان در نظر گرفته شده در الگوریتم KNN را نشان می دهد. همچنین از ۳ تا ۹ متغیر است.

accuracy: این ستون دقت طبقه بندی کننده KNN را برای هر ترکیبی از پارامترها نمایش می دهد. دقت نشان‌دهنده نسبت نمونه‌های پیش‌بینی‌شده درست از کل نمونه‌های مجموعه تست است.

مشاهدات:

اثر n_components: با افزایش تعداد اجزای اصلی، دقت به طور کلی بهبود می یابد. به عنوان مثال، دقت تمایل دارد از ۲ به ۱۰ جزء افزایش یابد و سپس کمی فراتر از آن کاهش یابد. این نشان می‌دهد که افزایش تعداد مؤلفه‌ها واریانس بیشتری را در داده‌ها ثبت می‌کند، که منجر به عملکرد طبقه‌بندی بهتر تا یک نقطه خاص می‌شود.

اثر k: تعداد همسایگان (k) نیز بر عملکرد طبقه بندی کننده تأثیر می گذارد. به طور کلی، افزایش k تمایل به بهبود دقت دارد، اما ممکن است نوساناتی وجود داشته باشد. به عنوان مثال، دقت به طور کلی با افزایش k از ۳ به ۹ افزایش می یابد، که نشان می دهد در نظر گرفتن همسایگان بیشتر می تواند به پیش بینی های قوی تر منجر شود.

پارامترهای بهینه: ترکیب پارامترهایی که بالاترین دقت را به دست می دهد n_components ۱۰ و k = ۹ با دقت تقریباً ۰.۸۴۲ است. این نشان می دهد که استفاده از ۱۰ مؤلفه اصلی و در نظر گرفتن ۹ همسایه در الگوریتم KNN بهترین عملکرد را برای این مجموعه داده به همراه دارد.

این نتایج اهمیت فراپارامترهای مدل تنظیم مانند تعداد اجزای اصلی و تعداد همسایگان را برای دستیابی به بهترین عملکرد نشان می دهد. با ارزیابی سیستماتیک ترکیبات پارامترهای مختلف، می‌توانیم پیکربندی بهینه را شناسایی کنیم که دقت پیش‌بینی مدل را به حداکثر می‌رساند.

به طور کلی، این تجزیه و تحلیل بینش هایی را در مورد اینکه چگونه انتخاب پارامترها بر عملکرد طبقه بندی کننده KNN تأثیر می گذارد و به انتخاب مناسب ترین پیکربندی برای پیش بینی های دقیق روی مجموعه داده کمک می کند، ارائه می دهد.