درس داده کاوی

سری سوم تمریها

دکتر هادی فراهانی

گردآورنده

مهدی شایسته

۹۹۴۲۲۱۱۴

تمام قسمت های کدنویسی شده این سری سوالات با نام های زیر بروی گیت هاب اپلود شده است و در هر قسمت از پاسخ ها به لینک گیت هاب مجددا اشاره می شود

https://github.com/mahhhdy/SBU_DataMining_Python/blob/main/exercise3_part1_mobile_price_prediction_using_SVM.ipynb

https://github.com/mahhhdy/SBU_DataMining_Python/blob/main/exercise3_part2.ipynb

https://github.com/mahhhdy/SBU_DataMining_Python/blob/main/exercise3_part3.ipynb

https://github.com/mahhhdy/SBU_DataMining_Python/blob/main/exercise3_part4_Indicators.ipynb

جواب سوال یک :

در svm هدف پیدا کردن یک ابر صفحه برای طبقه بندی است با بیشترین مرز تصمیم گیری

اما گاهی اوقات ما نمیتوانیم به صورت خطی داده ها را از هم جدا کنیم مخصوصا زمانی که داده ها بصورت تصادفی گسترده شده باشند

و راه حل برای این مساله بردن ابعاد مسئله به بعد های بالاتر است ، اما زمانی که ابعاد مسئله رو به افزایش است محاسبات پیدا کردن ابرصفحه بسیار زیاد می شوند

راه حل این استفاده کرنل تریک هست که به ما این امکان را میدهد که بدون بردن مسئله به ابعاد بالاتر ، در همان بعد ویژگی محاسبات را انجام دهیم

از پرکاربردترین کرنل ها میتوان به کرنل چند جمله ای (polynomial kernel) و radial basis function-RBF اشاره کرد

به RBF همچنین gaussian kernel نیز گفته میشود ، به دلیل اینکه از سری تیلور استفاده میکند ، امکان پشتیبانی از بینهایت بعد در فضای ویژگی را دارا می باشد

البته این مشکل احتمال دارد که با بردن مساله به فضای بالاتر احتمال overfit شدن مدل ما افزایش پیدا میکند به همین دلیل انتخاب کرنل مناسب بسیار اهمیت دارد تا از بروز overfitting جلوگیری کند

Linear kenel : ساده ترین نوع کرنل کرنل خطی هست که بهترین تابع در زمانی که تعداد زیادی ویژگی داریم ، از این نوع کرنل بیشتر در مسائل طبقه بندی متون استفاده می شود ، سریع ترین نوع کرنل ها کرنل های خطی هستند

Polynomial kernel کرنل های چند جمله ای ، زمانی که مجموعه داده های اموزشی ما نرمال سازی شده باشند بهترین نتیجه رو میده

Radial Basis Function زمانی که با مساله ای با ابعاد بالا کار میکنیم و میخواهیم داده ها رو به صورت خطی جدا کنیم

kernel Sigmoid که در شبکه های عصبی کاربرد دارند ، عملکردی مشابه مدل پرسپترون دو لایه دارد

Gaussian kernel مرسوم ترین کرنل هست و زمانی که اطلاعاتی در مورد مجموعه نداریم این کرنل میتونه مفید واقع بشه

Anova kernel به عنوان RBF هم شناخته می شود ، معمولا در مسائل رگرسیون چند وجهی شناخته شده است

جواب سوال ۲و۳و۴ در گیت هاب به ادرس زیر قرار گرفته است :

https://github.com/mahhhdy/SBU_DataMining_Python/blob/main/exercise3_part2.ipynb

جواب سوال پنج :

قسمت الف) عملیات binning بروی ستون battery power انجام گرفت و کد آن در گیت هاب به ادرس زیر قرار گرفته است

https://github.com/mahhhdy/SBU_DataMining_Python/blob/main/exercise3_part2.ipynb

قسمت ب) عملیات one hot encoding برای ستون های categorical استفاده می شوند. مقادیر categorical نوع داده ای هستند که تنها میتوانند مقادیر محدودی را در خود جای دهند ، برای مثال ممکن است برای ستون رنگی ، شامل رنگهای سبز ، ابی ، قرمز و ... باشد.

در صورت استفاده از این نوع داده ای بعضی از مدل های یادگیری ماشین امکان انجام پردازش مستقیم و بدون کدبندی بروی این نوع داده را ندارند ، یکی از روش های encoding روش one hot encoding است .

تنها ستون categorical ما همان ستونی binning_bettery_power هست که از عملیات binning بروی ستون power battery حاصل شده است و مابقی ستون ها از نوع int و float هستند

کد این قسمت در گیت هاب به ادرس زیر قرار گرفته است

https://github.com/mahhhdy/SBU_DataMining_Python/blob/main/exercise3_part2.ipynb

قسمت ج) گاهی اوقات داده های خامی که در اختیار داریم برای تحلیل هایی که در نظر داریم اماده نیستند و تخمین های ما را دچار مشکل میکنند ، ممکن است داده ها نرمال نبوده ، خطی نبوده و یا دارای پراکندگی یکسانی نبوده باشند ، در این جا ما از تبدیلاتی چون ریشه دوم، لگاریتم، وارون و مجذور کردن برای رفع مشکل نرمال نبودن ، خطی نبودن و یا توزیع پراکنده استفاده میکنیم.

به طور کلی زمانی که داده ها نرمال نیستند از روش های ریشه دوم، لگاریتم و وارونه کردن برای تبدیل داده ها استفاده می شود.

نوع مشکل	نوع تبدیل پیشنهادی
غیرخطّی بودن رابطه	مجذور کردن (به توان دو رساندن)
چولگی متوسط (مثبت یا منفی)	ریشه دوم
چولگی شدید (مثبت یا منفی)	لگاریتم (Log 10)

ابتدا ما با ازمون های نرمال بودن می بایست نرمال بودن توزیع داده هامون رو تست کنیم ، بعد میتونیم از تبدیل های log و نمایی برای نرمال کردن توزیع داده هامون استفاده کنیم و البته با انجام مجدد ازمون های نرمال بودن ، می توانیم ببینیم تبدیلاتی که انجام دادیم مفید بوده است یا خیر .

کد پیاده سازی شده این تست ها در گیت هاب بارگزاری به ادرس زیر اپلود شده است.

https://github.com/mahhhdy/SBU_DataMining_Python/blob/main/exercise3_part2.ipynb

قسمت د) اضافه کردن یک ویژگی در کد بالا که که در گیت هاب قرار گرفته است ، اماده شده است .

جواب سوال هفت)

درخت تصمیم (decision tree)یکی از پرکاربردترین الگوریتم‌ها در بین الگوریتم‌های داده‌کاوی است. درخت تصمیم دقیقا مانند یک درخت است با این تفاوت که از ریشه به سمت پایین (برگ) رشد کرده است. در الگوریتم درخت تصمیم نمونه‌ها را دسته‌بندی می‌کنیم که در واقع دسته‌ها در انتهای گره‌های برگ قرار دارد. درخت تصمیم در مسائلی کاربرد دارد که بتوان آنها را به صورتی مطرح نمود که پاسخ واحدی به صورت نام یک دسته یا کلاس ارائه دهند. مانند تشخیص های پزشکی که با داشتن اطلاعات بیماران، بر اساس اینکه فرد دارای بیماری خاصی است یا خیر طبقه بندی انجام می‌شود.

دقیقا شبیه بازی بیست سوالی که یک نفر موضوع خاصی را در ذهن خود در نظر می‌گیرد و شخص دیگری سعی می‌کند با پرسش تعدادی سوال که جواب آنها بلی و خیر است موضوع مورد نظر شخص اول را شناسایی کند.

الگوریتم های زیادی برای درخت تصمیم وجود دارند که به نام چند نمونه از انها در زیر ذکر شده است:

- ID3:Iterative Dichotomiser
- C4.5: Classifier 4.5
- CART: Classification And Regression Tree
- ID4
- ds CART: DempsterShafer Classification And Regression Tree
- ID5R
- EC4.5:Efficient Classifier 4.5
- CHAID: Chi square Automatic Interaction Detection
- RF: Random Forest
- RT: Random Tree
- DS: Decision Stump
- QUEST: Quick Unbiased Efficient Statistical Tree

اغلب تفاوت درخت های تصمیم گیری در معیار اندازه گیری عدم خلوص ، شیوه شاخه بندی و هرس کردن گره های درخت هستند.

در بعضی از الگوریتم های درخت تصمیم تنها برای داده های اسمی و متنی کاربرد دارند و بضی تنها برای داده های عددی و بعضی هر دو .

در بعضی از الگوریتم های درخت تصمیم توانایی ساخت مدل با وجود مقادیر گمشده وجود دارد و در برخی دیگر وجود ندارد .

در بعضی از الگوریتم های درخت تصمیم توانایی هرس کردن برای جلوگیری از بیش برازش وجود دارد

در بعضی از الگوریتم های درخت تصمیم توانایی وزن دهی متفاوت به ویژگی ها وجود دارد.

جواب سوال هشت)

جواب سوال ۸ در مورد پیاده سازی یک درخت تصمیم در گیت هاب به ادرس زیر بارگزاری شده است و ویژگی قیمت به عنوان ویژگی برای درخت تصمیم انتخاب شد.

https://github.com/mahhhdy/SBU_DataMining_Python/blob/main/exercise3_part2.ipynb

جواب سوال نه)

در درخت تصمیم هرچه اندازه درخت کوچکتر باشد باعث می‌شود مدل تعمیم‌پذیری بیشتری داشته باشد و واریانس کمتر می‌شود. در درخت تصمیم هرچقدر تعداد نود ها بیشتر شود به اولویت‌های ریزتر هم اجازه می‌دهیم وارد مسأله شوند.

در مواردی که تعداد دسته‌ها زیاد و تعداد نمونه‌ها کم است احتمال خطا و عملکرد نادرست در مدل زیاد است.

تنظیم عمق درخت کار مشکلی است و عملکرد مناسب درخت به طراحی بهینه آن بستگی دارد.

در صورت وجود خطا در درخت تصمیم‌گیری این خطا به برگ‌های زیرین منتقل می‌شود و بر روی عملکرد کل درخت تاثیر می‌گذارد.

جواب سوال ده)

ممکن است در ایجاد درخت تعداد زیادی شاخه به وجود آید ، دلیل آن وجود آنومالی در داده‌ها است و آنومالی به دلیل وجود نویز و داده های پرت به وجود آید ، از طرفی ممکن است درخت ایجاد شده برای داده های جدید ضعیف عمل کند و در نتیجه درخت دچار بیش برازش شود .

هدف از حرص کردن درخت تصمیم جلوگیری از بیش برازش یا overfitting است ، اینکار باعث کوچک شدن، ساده شدن و به الطبع فهم آسان درخت خواهد شد. ازجهتی برای داده های تست عملکرد بهتری خواهد داشت

برای حرص کردن درخت تصمیم دو روش Prepruning و postPrepruning وجود دارد

جواب سوال یازده)

جواب این سوال در کولب پیاده سازی و در گیت هاب بارگزاری شده است.

https://github.com/mahhhdy/SBU_DataMining_Python/blob/main/exercise3_part2.ipynb

جواب سوال دوازدهم)

جواب این سوال در کولب پیاده سازی و در گیت هاب بارگزاری شده است.

نتیجه مقایسه رندوم فارست و درخت تصمیم نشان میدهد نتیجه رندوم فارست بهتر بوده است

https://github.com/mahhhdy/SBU_DataMining_Python/blob/main/exercise3_part2.ipynb

جواب سوال سیزدهم)

روش های یادگیری عمیق و شبکه عصبی بیشتر شبیه یک جعبه سیاه هستند، اگرتصمیمی که با یک شبکه عصبی گرفته شده رو به چالش بکشید ، توجیه و توضیح اون به سختی ممکن خواهد بود ، در مقابل نتیجه یک درخت تصمیم رو میتوان براحتی توضیح و تفسیر کرد

جواب سوال چهاردهم)

روش Ripper یکی از پرکاربردترین روش های استخراج قوانین هست که از روش تقسیم و حل استفاده میکند. در این روش ابتدا برای بدست اوردن مجموعه قوانین ، هرس صورت میگیرد ، سپس یک مرحله بهینه سازی بروی هر قانون صورت میگیرد و در نتیجه دو قانون دیگر استخراج می شود ، یک قانون جایگزینی و یک قانون تجدید نظر.

پس از آن ، در مورد اینکه آیا مدل باید قاعده اصلی ، جایگزینی یا قانون تجدید نظر را بر اساس معیار حداقل طول توصیف حفظ کند ، تصمیم گیری می شود

روش CAMUR که بر اساس الگوریتم Ripper هست ، روش دیگری از استخراج قوانین است که در این روش با محاسبه تکراری یک مدل طبقه بندی مبتنی بر قاعده ، پایه های قاعده ای متعدد و معادل را استخراج می کند CAMUR .شامل یک مخزن پایگاه داده و یک ابزار پرس و جو است

روش PART یک الگوریتم درخت تصمیم گیری جزئی است که از استراتژی تقسیم و حل استفاده میکند ، در این روش مجموعه ای از قوانین ایجاد می شود و بصورت بازگشتی همه ایتم ها سپری می شوند تا هیچ ایتمی نماند ، که برای این کار از یک درخت تصمیم C4.5 جزئی استفاده میکند و برگ با بیشترین پوشش را انتخاب میکند، پس از ان درخت تصمیم گیری به همراه موارد تحت پوشش حدف می شوند تا از تعمیم اولیه جلوگیری کنند ، این روند تا زمانی که همه قوانین استخراج پوشش داده شوند ، تکرار می شود.

جواب سوال پانزدهم)

اکثرا ما برای گرفتن یک تصمیم از درخت تصمیم استفاده میکنیم ، ساختاری شبیه به درخت که هر برگ ان یک تصیمیم است و این روند تا گرفتن نتیجه نهایی ادامه پیدا میکند .

اما با تلفیق ensemble methods و درخت تصمیم گیری میتوان برای استفاده در پیش بینی سری های زمانی استفاده کرد. در واقع سری های زمانی، مجموعه ای اطلاعات از افزایش های متوالی داده ها که در یک دوره زمانی جمع آوری شده اند، می باشد .

پرکاربردترین الگوریتم در پیش بینی سری های زمانی decision tree ensemble methods رندوم فارست هست ، البته محبوبترین الگوریتم های دیگر درخت تصمیم گیری که در پیش بینی سری های زمانی مورد استفاده قرار میگیرند به شرح زیر است.

- Regularized Greedy Forest (RGF)
- Gradient Boosting
- eXtreme Gradient Boosting (XGBoost)
- LightGBM
- Combining Tree-Boosting with Gaussian Process and Mixed Effects Models (GPBoost)
- Natural Gradient Boosting (NGBoost)

البته این رو مد نظر داشته باشید که با توجه به کاری که مد نظر دارید و سرعتی که برای محاسبه نیاز دارید ، ممکن است اولویت استفاده از الگوریتم های فوق تغییر کند ، برای مثال پیش بینی بارش یا پیش بینی فروش محصول و یا پیش بینی بازار سهام ، در هر کدام ممکن از الگوریتم درخت تصمیم متفاوتی استفاده شود.

جواب سوال شانزدهم)

پاسخ این سوال ، در کولب انجام گرفته و در گیت هاب به ادرس زیر آپلود شده است

چون اعداد داخل دیتاست شامل کاما (,) بودند ، در ابتدا دیتاست رو داخل نوت پد ادیت کردم و سپس دیتا ست رو اپلود کردم ، به این شکل داده های عددی از جنس float بودند و دیگر مشکل تبدیل ستون های عددی برطرف شد .

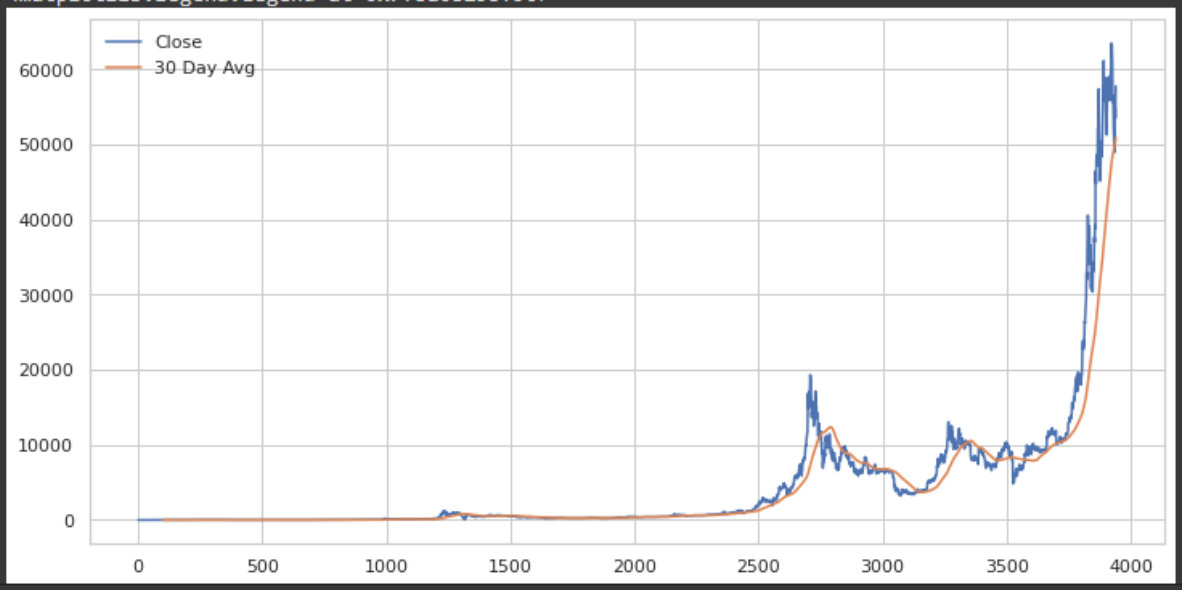
https://github.com/mahhhdy/SBU_DataMining_Python/blob/main/exercise3_part3.ipynb

جواب سوال بیست و پنجم)

در این سوال خواسته شده است که از اندیکاتور ها برای پیش بینی قیمت استفاده کنیم ، یکی از اندیکاتور ها MA یا moving average است و تمامی اندیکاتور های دیگر از فرمول های ریاضی با استفاده از MA بدست می آیند.

```
[8] sns.set()
    sns.set_style('whitegrid')
    btc['Close'].plot(figsize=(12,6),label='Close')
    btc['Close'].rolling(window=100).mean().plot(label='30 Day Avg')
    plt.legend()
```

```
<matplotlib.legend.Legend at 0x7f8a08295f90>
```



همین طور که ملاحظه می کنید MA30 ، EMA و RSI با استفاده از قیمت Close در بالا نمایش داده شده است و در گیت هاب به ادرس زیر قرار گرفته است

https://github.com/mahhhdy/SBU_DataMining_Python/blob/main/exercise3_part4_Indicators.ipynb

اما در مورد پیش بینی قیمت با استفاده از اندیکاتور ها می بایست این مطلب رو بیان کنم که تمامی اندیکاتور ها دارای تاخیر می باشند به علت این که با توجه به سابقه قیمت سهم تولید می شوند و استفاده صحیح از اندیکاتور ها در پیش بینی قیمت ها نیست ، بلکه در گرفتن تاییدیه است ، مخصوصا در بازار های ساید یا رنج که اصطلاحا گفته می شود مرگ اندیکاتور هست ، تنها اندیکاتوری که در بازار های رنج مقداری کاربرد دارد اندیکاتور RSI است .

یعنی یک تردیر با ابزارها و اطلاعاتی که در دست دارد ، یک نقطه را برای ورود به سهم (خرید یا فروش) انتخاب میکند ، در این مرحله از اندیکاتور ها برای تایید و اطمینان از نظر خود استفاده می کند .

فرایند شکل گیری نمودار های بازار سهام به این شکل است که در ابتدا بر اساس احساس و یا اطلاعاتی (رانت) که صاحبان سرمایه دارند تصمیم به خرید و یا فروش یک سهم میگیرند ، در این مرحله نواحی عرضه و تقاضا شکل میگیرند ، قیمت های کندل بسته می شوند و بعد از این هست که اندیکاتور ها نمایش پیدا میکنند ، چیزی شبیه به این که یک لغت از یک زبان به زبان دوم و سپس به زبان سوم تبدیل و ترجمه شود و سپس به زبان اول برگردانده شود ، در حالی که برای پیش بینی قیمت سهم می توان از همان نواحی عرضه و تقاضا استفاده کرد .