دانشگاه صنعتی امیرکبیر

دانشکده مهندسی کامپیوتر و فناوری اطلاعات

تمرین پنجم درس مبانی داده کاوی

احسان مهرعلیان(۹۳۳۱۸۰۲)
ehsan.mehralian@gmail.com

استاد:
دکتر احسان ناظرفرد

بهار ۹۶

مساله اول: Fraud Detection

این دیتافریم شامل ۱۰۰,۰۰۰ داده تراکنش مختلف می باشد که از این تعداد ۷۳۷۲۹ تراکنش مربوط به حساب های منحصر به فرد می باشد و ۷۰۱۲۴ مربوط به ایمیل های متفاوت می باشد

مقدار zip مربوط به zipcode یک منطقه بوده و state بیانگر شهر مورد نظر است. با داشتن zip code می توان شهر مربوط به آن را پیدا کرد پس feature مربوط به state را حذف میکنیم

همچنین feature های amount و total یکسان هستند. بنابراین یکی از آن ها را (total) را حذف میکنیم

فیلد a، چهار تا مقدار ۰و۱و۲و۳ را میگیره

فیلد b، باینری ۰و۱ است

فیلد های hour_a و hour_b مربوط به اعمالی پشت سر هم هستند چرا که حداکثر یک ساعت با هم اختلاف دارند (چون مقادیر عددی بین ۰ تا ۲۳ هستند می توان نتیجه گرفت که ساعت انجام عملی را نشان می دهند) ولی از آنجایی که در بعضی از ردیف ها hour_a یکی بیشتر و در بعضی دیگر hour_b بیشتر است پس این دو عمل وابستگی ای به یکدیگر ندارند و به هر ترتیبی می توانند انجام شوند.

برای غلبه بر این مشکل، داده هایی که fraud تشخیص داده شده اند را ۵ برابر کردیم تا تعداد کم آن ها منجر به understimate شدن این نوع داده ها نشود و مدل تشخیص داده شده به سمت برچسب صفر بایاس نگردد.

برای بخش validation این مساله از ۱۰kold استفاده کردیم.

و برای ارزیابی مدل ها از دو پارامتر TPR که نشان دهنده میزان موفقیت مدل در تشخیص تقلب (نسبت تعداد داده هایی که تقلب تشخیص داده شده اند به کل داده های تقلب) می باشد و در پارامتر مناسبی برای ارزیابی در مساله تشخیص fraud می باشد. چراکه اهمیت داده های با کلاس + را بیشتر در نظر می گیرد. همچنین از روش های RMSE و f-score نیز برای مقایسه مدل ها با یکدیگر استفاده شده است.

در مدل Decision Tree با پارامتر های max_depth=۱۵, min_samples_split=۲ مقدار TPR محاسبه شده برابر ۰.۶۵۶۶۰۱ و مقدار خطای RMSE محاسبه شده برابر ۰.۲۳۰۳۶۶ شده است و مقدار پارامتر f-score برابر ۰.۷۳۷۱۹۹ شد.

تعیین عمق درخت و حداقل تعداد گره هایی که در یک برگ از درخت تصمیم ترسیم شده قرار می گیرند منجر می شود که درخت تصمیم به دست آمده بر روی داده های train بیش برازش (overfit) نکند و حداکثر عمق در از max_depth تعیین شده بیشتر نشود (میدانیم که با افزایش عمق، مدل به دست آمده overfit می کند.)ولی با توجه به کمتر بودن داده های fraud عمق کم نیز منجر به underfit شدن مدل می شود پس یک مقدار میانی برای آن باید در نظر گرفت.

در مدل Random forest با پارامتر های max_depth=۱۰, min_samples_split=۲, n_estimators=۱۰۰

مقدار TPR محاسبه شده برابر ۰.۳۹۰۳۹۴ و مقدار خطای RMSE محاسبه شده برابر ۰.۲۲۹۳۶۵ می باشد و مقدار f score برابر ۰.۷۰۰۰۵۵ می باشد.
پارامتر های این مدل مشابه مدل درخت تصمیم می باشد ولی با توجه به اینکه تعداد درخت های ایجاد شده الزاما یکی نمی باشد عمق هر درخت را می توان کمتر از مدل DT انتخاب کرد.

در مدل Neural Network که در واقع MLPClassifier می باشد. در این مدل از دو لایه perceptron که در هر لایه ۵ تا قرار دارد (در مجموع ۱۰ تا) قرار داده شده است. که مقدار TPR محاسبه شده برابر ۰.۲۳۷۲۳۲ و مقدار خطای RMSE محاسبه شده برابر ۰.۲۷۳۸۷۳ شده است و مقدار پارامتر f-score برابر ۰.۴۳۹۲۵۷ شد.

همچنین در مدل LogisticRegression با پارامتر L۱ به عنوان خطا استفاده شده است که مقدار TPR محاسبه شده برابر ۰.۲۰۹۲۲۳ و مقدار خطای RMSE محاسبه شده برابر ۰.۲۴۸۹۵۰ شده است و مقدار پارامتر f-score برابر ۰.۶۰۷۰۹۷ شد.

و اما در نهایت برای ترکیب مدل های فوق و ایجاد یک مدل ensumble، از تابع VotingClassifier استفاده شده است. در صورتی ترکیب چند مدل نتیجه بهتری برای ما ایجاد میکند که ناحیه های ایجاد خطا در ان ها با یکدیگر متفاوت باشد در غیر این صورت حتی نتیجه بدتری نیز به دنبال خواهد داشت. بنابراین بایستی با تغییر مدل های به کار گرفته شده در مدل ensumble مدل هایی که ترکیب آن ها معیار های ارزیابی را ارتقا می بخشد استفاده میکنیم.

برای مثال وقتی از سه مدل (forest), (DT), (NN) استفاده می کنیم
مقدادیر پارامتر ها برابر ۰.۲۲۷۷۷۱ :eclf_RMSE و ۰.۳۸۷۳۲۷ :eclf_TPR و ۰.۷۰۲۵۹۶ :fECLF
و وقتی از تمام مدل های استفاده می کنیم برابر ۰.۲۴۸۰۴۵ :eclf_RMSE و ۰.۲۰۸۹۲۰ :eclf_TPR و ۰.۶۰۸۵۰۹ :fECLF می باشد. که بیانگر عملکرد بدتر مدل در این حالت است. پس الزاما استفاده از تعداد مدل های بیشتر به نتیجه بهتری نمی انجامد، بلکه باید مدل های مورد استفاده با یکدیگر سازگاری داشته باشند.
برای مثال دیگر اگر تنها از دو مدل random forest و neual network استفاده کنیم، با توجه به اینکه برای تصمیم گیری در مورد مدل نهایی از نظر سنجی بین مدل های داخلی آن استفاده میکند، در این صورت استفاده از دو مدل هیچ گونه توجیهی ندارد و با مشاهده نتایج آن نیز این نتیجه به وضوح روشن است:

eclf_RMSE: ۰.۲۷۳۸۷۳
eclf_TPR: ۰.۰۰۰۰۰۰
fECLF: ۰.۴۳۹۲۵۷

البته همان طور که مشاهده می شود، برای مثال مقدار recall(TPR) برای مدل Decision Tree برابر حدود ۰,۶ شد که از مقدار به دست آمده در تمام مدل های دیگر و مدل unsumble هم بهتر است ولی باید توجه کنیم که مقدار RMSE آن کمتر است. درست مثل این است که مدل ها تمام داده ها را تقلب تشخیص دهد که در این صورت مقدار recall یک می شود ولی مدل فوق هیچ گونه هوشمندی ندارد. بنابراین باید مدلی انتخاب کنیم که مجموعه پارامتر های ارزیابی مختلف در آن بهتر باشد نه صرفا یک پارامتر.

مساله دوم: حل مساله کشتی تایتانیک با استفاده از مدل unsumble

پس از اعمال پیش پردازش های یکسان با تمرین اول، علاوه بر دسته بند random forest که در تمرین اول مورد استفاده قرار گرفت، از دسته بند های MLPClassifier, LogisticRegression , decision tree استفاده می کنیم.
نتایج آن ها به صورت زیر است:

DT: 0.661903
DTRMSE: 0.441294

NNTPR: 0.531483
NNRMSE: 0.616627

forest: 0.707110
forestRMSE: 0.410616

LR_TPR: 0.688330
LRRMSE: 0.454187

پارامتر های این مربوط به این مدل ها را همانند همان پارامتر های مساله قبل انتخاب کرده ایم.

حال به سراغ ترکیب مدل های فوق با استفاده از دسته بند VotingClassifier می پردازیم. همان طور که در مساله قبلی نیز گفته شد الزاما ترکیب مدل ها با یکدیگر ملزم به ایجاد مدل بهتری نمی شود، بلکه در صورتی این چنین است که نواحی خطای مدل های از یکدگیر مجزا باشد. پس به این صورت
در صورت ترکیب مدل های Decision Tree , forest fandom داریم:

eclf_RMSE: 0.431149
eclf_TPR: 0.618181

و با اضافه کردن LR به آن ها، نتایج به این صورت می شود:
eclf_RMSE: 0.444315
eclf_TPR: 0.575438

[(LR',LR') ,(DT', DT') ,(forest', forest')]
eclf_RMSE: 0.406590
eclf_TPR: 0.703769

هر چهار مدل :
eclf_RMSE: 0.449528
eclf_TPR: 0.555935

به نظر می رسد ترکیب سه مدل [(forest', forest'), (DT', DT'), (LR',LR')] نتایج بهتری در بر دارد.