فایل ضمیمه : HW5.ipynb

داده های سوال را از فایل HW5.csv به صورت یک دیتافریم از pandas باز می‌کنم. این داده ها مجموعا 32561 ردیف داده هست که هر کدام از ۱۵ ویژگی ساخته شده. که ۶ تا مقدار عددی و بقیه رشته رشته هستند.

روی ۳ ستون از این ها (WorkClass, Occupation, NativeCountry) هر کدام به ترتیب ۱۸۳۶ و ۱۸۴۳ و ۵۸۳ مقدار گم شده وجود دارد. برای کار کردن با مقادیر گم شده روش های مختلفی وجود دارد. هر ۳ ستون مقادیر کیفی دسته ای دارند. از جمله ناکارامد ترین (!) روش های برخورد با مقادیر گم شده حذف سطر یا (بدترا!) ستون است. یک روش بهتر که برای داده های کیفی کاربرد دارد ولی چندان هم هوشمندانه نیست استفاده از مقدار مد هر ویژگی برای پر کردن مقدار گم شده است. یعنی پرتکرارترین برچسب هر ستون را به جای مقدار خالی به کار بگیرم. نقطه ی قوت این روش ثابت ماندن مد داده هاست. چون در ادامه با دسته بندهای آماری سروکار دارم در حالت بهتر باید تلاش بر حفظ پارامترهای مرکزی مثل میانه و میانگین می بود ولی چون داده ها الان عددی نیستند استفاده از میانه و میانگین امکان پذیر نیست. از طرفی برای روش های هوشمندانه تری مثل KNN هم پیش پردازش های زیادی لازم است. مشکل دیگر این است که برای حل مشکل مقادیر گم شده در بهترین حالت باید داده های تست و آموزش از هم جدا شده باشند. در استفاده از هر دوی روش ها (KNN و مد) داده های تست از آموزش تاثیر می‌گیرند ولی چون KNN خود یک روش دسته بندی است تاثیر نتایج نهایی بیشتر خواهد بود .قصد دارم برای گزارش نتایج به جای اینکه خودم تست و آموزش را جدا کنم از cross_val_score با معیار accuracy از کتابخانه ی sklearn استفاده کنم پس برای اینکه پیش پردازش انجام شده روی کل داده ها تاثیر کمتری روی نتیجه نهایی داشته باشد، از همان روش جاگذاری بیشترین تکرار یا همان مد هر ستون استفاده می‌کنم.

روی این ۱۵ ویژگی ستون آخر یا همان Income متغیر وابسته یا هدف در در نظر می‌گیریم که یا مقدار <50K= یا >50K دارد. برای ادامه ی کار این برچسب ها را به ترتیب نظیر صفر و یک می‌گیرم.

روی ستون Education و Education-num هم در واقع یک ویژگی دارم که در ستون اولی به صورت مقدار رشته ای و در دومی به حالت عدد کد شده وجود دارد. برای ادامه ی کار با ستون کد شده ادامه می‌دهم و چون مقدار این ستون کیفی ترتیبی است، یعنی افزایش و کاهش مقدار کد معنادار است (پایین و بالاتر بودن سطح تحصیلات)، این ستون را بدون پیش پردازش های بعدی مستقیما می‌توانم برای ساخت مدل استفاده کنم.

پس کل ستون های مستقل من ۱۳ تا هست که ۶ تای آنها مقادیر عددی و یکی مقدار ترتیبی کیفی یا ordinal و ۶ تای بقیه دسته ای یا categorical دارند که نیاز به پیش پردازش های دیگری هستند و باید کد شوند. کد گذاری این ستون ها باید شامل مراحل زیر باشد :

۱. روی هر ستون تعداد کل حالت ها شناسایی شود. فرضا d حالت.
۲. نظیر هر حالت یک متغیر در نظر گرفته شود. یعنی d تا ستون جدید می‌گیریم.

۳. به ازای مقدار آن ستون، متغیری که برابر مقدار آن ستون را دارد یک و بقیه صفر مقداردهی شوند.
۴. برای از بین بردن وابستگی ستون های اضافه شده ستون اول حذف شود چون با داشتن d-1 تا ستون میتوان ۱ یا ۰ بودن ستون dام را پیدا کرد.
۵. روی داده های اصلی به جای این ستون کیفی دسته ای d-1 ستون باینری تولید شده را جایگزین کنم.

این عملیات روی تمام ۶ ستون به صورت یکجا با استفاده از OneHotEncoder از sklearn.preprocessing انجام شده و دادگان حاصل ۸۲ ستون به عنوان بردار مستقل دارد. پیش پردازش هایی که با مقیاس داده ها سرو کار دارند روی مدل رگرسیون چندان تاثیر ندارند چون ضرایب مدل رگرسیون می‌تواند مطابق مقیاس داده ها انتخاب شود. برای مدل بیزی ساده هم می خواهم از توزیع گاوسی برای تمام ستون ها استفاده کنم که توزیع مقدار میانگین و واریانس را با روش ML از خود داده ها تخمین می‌زند و به عنوان پارامتر استفاده می‌کند و نهایتا مثیاس مقادیر روی توزیع احتمالات و نتیجه ی دسته بندی تاثیرگذار نیست پس بدون انجام پیش پردازش های اضافه تر برای تمام مدل های ادامه ی کار از همین داده ها استفاده می‌کنم و مقادیر df['y'] را هم عنوان ستون وابسته ی دوحالته می‌گیرم.

به عنوان دسته بندی حالت پایه از یک دسته بند تک حالته ی clfB استفاده می‌کنم که به تمام داده ها مقدار صفر یا نظیر همان کمتر از 50k را برچسب می‌دهد. نتیجه ی تست برابر است با فراوانی نسبی کلاس صفر روی مجموعه ی تست. که از روش cross validation با cv=10 برابر به طور میانگین accuracy برابر حدود ۷۶ درصد به دست می آید.

به عنوان دسته بند دوم از clf1 استفاده می‌کنم که یک مدل رگرسیون خطی است. از LogisticRegression همان کتابخانه ی sklearn استفاده می‌کنم با پارامترهای پیش فرض penalty = 12 و حداکثر تکرار برابر ۱۰۰ . با روش cross validation با cv=10 برابر به طور میانگین accuracy برابر نزدیک ۸۰ درصد به دست می‌دهد.

آخرین مدل از clf2 استفاده می‌کنم که یک دسته بند بیز ساده است که برای تمام ستون ها از توزیع گاوسی را فرض می‌گیرد. از کتابخانه ی sklearn از مدلهای بیزی ساده از GaussianNB استفاده می کنم. این مدل هم با همان معیار و روش قبل دقت نزدیک ۸۰ درصد را می‌دهد.

اعداد دقت به دست آمده نظیر هر مدل به طور دقیق در زیر آمده. هر دو مدل clf2 و clf1 به طور واضحی از مدل تک حالته هوشمندتر عمل می‌کنند که یعنی یادگیری روی این داده ها ثمربخش است و عملکرد نزدیک به هم دارند ولی مدل رگرسیون لوجستیک کمی نتیجه ی بهتری دارد.

classifier	method	accuracy
clfB	F(x) = 0, constant	**0.7591904489970196**
clf1	Logistic regression	**0.7969043240074865**
clf2	Gaussian Naive Bayes	**0.7951538040538655**