



گزارش پروژه ی اول

یافتن متغیرهای مربوط به نمره ی ریاضی دانش آموزان کلاس هشتم

فاطمه کریمی (۴۰۰۷۲۳۱۳۱)

بیستم اردیبهشت ۱۴۰۱

## چکیده

در این پروژه قصد داریم با در دست داشتن اطلاعات حدود ۵۸۰۰ دانش آموز، عوامل موثر بر نمره ی ریاضی آنان را بیاییم. برای این منظور، نمره های این دانش آموزان به همراه پاسخ آنان به شرایط محیطی آنان (مانند جنسیت، میزان تحصیلات پدر و مادر، سابقه ی کاری آموزگار و ...) به ما داده شده است. برای پاسخ به این پرسش، پس از تمیز کردن داده های ورودی، با حساب کردن میزان همبستگی بین هریک از متغیرها با متغیر خروجی نمره، موثر ترین عوامل بر نمره ی دانش آموز را می یابیم.

## ۱ راه اندازی

برای انجام این پروژه، از زبان برنامه نویسی پایتون و کتابخانه های مربوط به پردازش داده مانند pandas، و یادگیری ماشین مانند sklearn و scipy استفاده می کنیم. برای کشیدن نمودارها، کتابخانه ی matplotlib به کار رفته است. پروژه ی پیاده سازی شده در [Github](#) می باشد.

## ۲ شناسایی داده ها

گام نخست، شناسایی داده ها داده شده است. اطلاعات مربوط به هر متغیر در فایلی به نام T2019\_G8\_Codebook.xlsx آمده است. در این فایل به ازای هر متغیر، نام آن، سطح آن، بازه ای که مقادیر متغیر و جزییات آن آمده است. متغیرهای ورودی در سه نوع Ordinal، Nominal و Scale تعریف شده اند. برای متغیرهای دسته ای، توضیحات بیانگر گزینه های موجود برای انتخاب بوده است. مثلاً برای متغیر جنسیت، در توضیحات عبارت 1.Girl 2.Boy آمده است که بیان می کند برای دانش آموزان دختر در متغیر جنسیت، عدد ۱ و برای دانش آموزان پسر عدد ۲ قرار داده شده است. مورد دیگر، مقدارهای بیشینه و کمینه برای متغیرهاست که تنها برای برخی این مقدارها قرار داده شده است. برای مثال، برای متغیر سن دانش آموز بازه ی ۹ تا ۱۹ آمده است که بدین معناست هر عدد خارج از این بازه یک مقدار نامعتبر خواهد بود.

برای دسترسی به اطلاعات متغیرها، در پیاده سازی کلاسی با نام Codebook تعریف شده است. این کلاس اطلاعات مربوط به ویژگی ها را از فایل اکسل مربوط به آن بیرون می کشد و برای هر ویژگی، آبجکتی از کلاس Attribute می سازد و اطلاعات مربوط به ویژگی را در این آبجکت ذخیره می کند. به کمک کلاس Codebook، می توان به فهرستی از همه ی ویژگی ها دسترسی داشت. در دست داشتن این اطلاعات، به ما در پیش پردازش داده ها کمک می کند.

### ۳ پیش پردازش داده ها

پیش پردازش داده ها گام باارزشی در پردازش داده ها می باشد. این فاز از سه بخش تشکیل می شود: یکم پر کردن مقدارهای null و داده های پرت، دوم یافتن متغیرهای تکراری با حساب کردن همبستگی بین داده ها و سوم جایگزین کردن نمره های دانش آموزان با اعداد. در ادامه هر بخش به طور جداگانه توضیح داده می شود.

#### ۳-۱ پر کردن داده های خالی و داده های پرت

برای جایگزین کردن مقدارهای null برای یک ویژگی، ابتدا تعداد مقدارهای null را می شماریم. در صورتی که نسبت تعداد مقادیر null به تعداد کل مقادیر از ۵۰٪ بیشتر باشد، آنگاه کل آن متغیر را از داده های ورودی حذف می کنیم. علت این کار آنست که اگر درصد زیادی از داده ها null باشد، آنگاه می بایست این مقادیر خالی را با داده ای مانند میانه پر کنیم که این کار غیر منطقی ست؛ چرا که در واقع میانه ی کمتر از ۵۰٪ داده ها را برای آنان قرار می دهیم. بنابراین، در چنین شرایطی کل متغیر را از داده های ورودی حذف می کنیم. برای ویژگی هایی که میزان داده های null آنها از مقدار آستانه کمتر است، بنابر نوع ویژگی، چگونگی پر کردن آنها را بیان می کنیم.

#### پر کردن داده های null برای متغیرهای از نوع Nominal

همانگونه که می دانید، متغیرهای nominal از نوع متغیرهای دسته ای هستند و تفاوت آنها با متغیرهای ordinal آنست که هیچ دسته نسبت به دسته ی دیگر برتری ندارد. این موضوع باعث می شود که برای این دسته از متغیرها، میانگین و میانه بی معنی باشد. بنابراین ما از مقدار مد(mod) برای پر کردن مقادیر null استفاده می کنیم. مد داده ایست که بیشترین تکرار را در ورودی داشته است. گاهی در ورودی، مقدار unknown به جای آنکه با null پر شود، با مقدارهای عددی بسیار بزرگ مانند ۹۹۹۹ پر شده است. برای متغیرهای nominal، می توان بازه ی مجاز را از روی توضیحات آنها که در فایل Codebook آمده است، به دست آورد. سپس، همه ی مقادیر غیر مجاز را با مد پر کرد.

شایان ذکر است که پر کردن مقادیر غیر مجاز و null با مد، در صورتی انجام می شود که نسبت این داده ها به کل داده ها کمتر از ۰.۱ باشد. در صورتی که بیشتر بود، گزینه ی جدیدی به مجموعه گزینه های متغیر با نام UNKNOWN اضافه می شود و برای این دسته از داده ها مقدار (بیشترین شماره دسته ی موجود + ۱) را قرار می دهیم. متغیرهای BCBG10B، BCBG21B، BCBG21C، BSBM27AA و BSDG05S متغیرهایی هستند که به آنها گزینه ی UNKKNOWN اضافه شده است.

## پر کردن داده های null برای متغیرهای از نوع Ordinal

از آنجایی که در متغیرهای Ordinal ترتیب معنادار است، برای جایگزین کردن مقادیر null و غیر مجاز (خارج از بازه) از میانه استفاده می کنیم. نکته ی قابل توجه آنست که برای متغیرهای Ordinal امکان اضافه کردن دسته ی جدید وجود ندارد؛ بنابراین همه ی مقادیر null با داده ی میانه پر می شوند.

## پر کردن داده های null برای متغیرهای از نوع Scale

برای متغیرهای از نوع scale، ابتدا می بایست داده های پرت را بیابیم. برای یافتن داده های پرت، از روش whisker box استفاده می کنیم. بدین شکل که داده هایی که خارج از بازه ی  $[q1 - 1.5 * iqr, q3 + 1.5 * iqr]$  قرار دارند را داده ی پرت در نظر می گیریم و این داده ها را با مقدار null جایگزین می کنیم. حال، درصد داده های null به کل داده ها را در نظر می گیریم؛ اگر بیش از ۵۰٪ بود، کل متغیر را حذف می کنیم. در غیر این صورت، متغیر را نگه می داریم. اگر متغیر حذف نشد، توزیع آن را بررسی می کنیم. اگر از توزیع نرمال پیروی می کرد، مقادیر null را با میانگین و در غیر این صورت با میانه پر می کنیم. برای بررسی توزیع داده ی متغیر، از کتابخانه ی scipy بهره می بریم. این کتابخانه تابعی به نام normaltest دارد و یکی از مقادیری که این تابع بر می گرداند مقدار p-value است. اگر مقدار p-value کمتر از ۰.۰۵ بود، آنگاه توزیع را نرمال در نظر می گیریم. البته هیچ یک از متغیرهای این مساله توزیع نرمال نداشتند و بنابراین در نهایت مقادیر null برای آنها با میانه پر شد.

## ۲-۳ حذف متغیرهای تکراری

پس از پر کردن متغیرهای خالی، نوبت به حذف متغیرهای تکراری ست. برای انجام این کار، مقدار همبستگی را به صورت دو به دو برای متغیرهای هر نوع حساب می کنیم. سپس، آنهایی که قدرمطلق همبستگی آنها بیشتر از ۰.۸ است را در نظر می گیریم از بین آن دو، یکی را حذف می کنیم. از آنجایی که روش محاسبه ی همبستگی برای هر نوع متفاوت است، در ادامه جزئیات این کار را بیان می داریم.

## محاسبه ی همبستگی بین دو متغیر از نوع scale

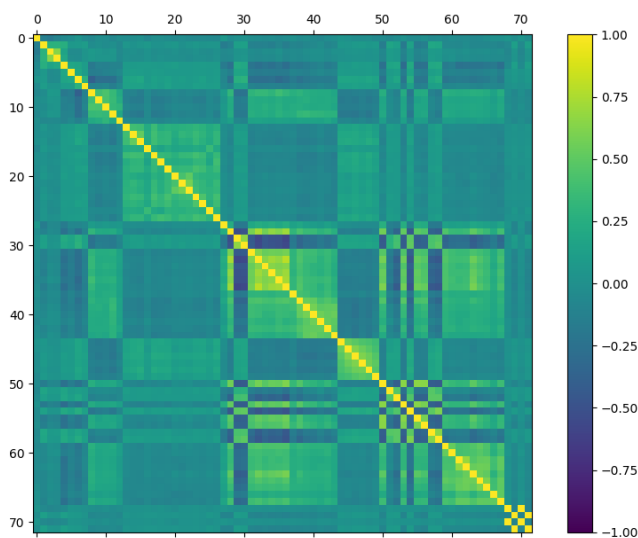
برای محاسبه ی همبستگی بین متغیر scale، از روش pearson استفاده می کنیم. برای این منظور، از تابع corr در کتابخانه ی pandas بهره می بریم. سپس، ماتریس همبستگی حاصل را به کمک کتابخانه ی matplotlib در قالب heatmap نمایش می دهیم. شکل ۱، نمودار همبستگی را نشان می دهد. همانگونه که از نمودار پیداست، هیچ کدام از دو متغیر همبستگی بسیار قوی وجود ندارد؛ بنابراین هیچ یک از متغیرهای از نوع scale به دلیل همبستگی حذف نمی شوند.



شکل 1- نمودار همبستگی میان دو متغیر از نوع scale

## محاسبه همبستگی بین دو متغیر از نوع Ordinal

برای محاسبه ی میزان همبستگی بین دو متغیر ordinal، از روش spearman rank correlation استفاده می کنیم. همانند قسمت قبل، تابع corr از کتابخانه ی pandas را فرا می خوانیم و روش محاسبه ی همبستگی را spearman تعیین می کنیم. شکل ۲، نمودار heatmap را نشان می دهد.

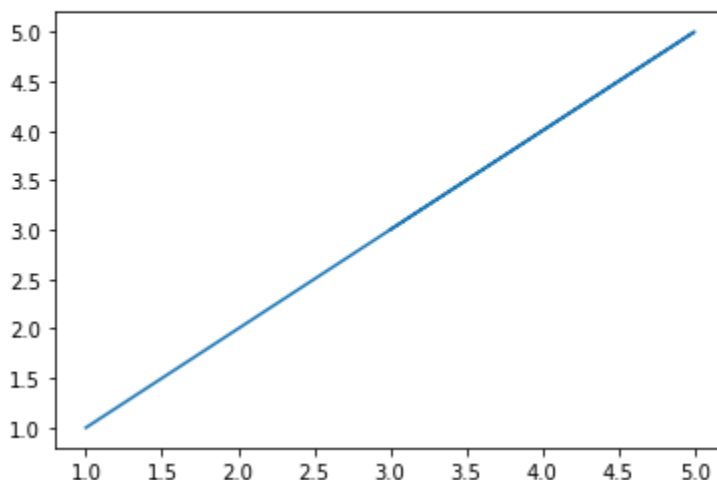


شکل 2- نمودار همبستگی میان دو متغیر از نوع ordinal

پس از محاسبه ی ماتریس همبستگی، سه ویژگی BSBM42AA،BSBM42BA و BSDGSLM به دلیل همبستگی قوی با متغیرهای دیگر حذف می شوند. برای نمونه، متغیر BSBM42AA با متغیر BSBM26AA همبسته است و نمودار آنها به شکل زیر است:

```
In [8]: plt.plot(df['BSBM42AA'], df['BSBM26AA'])
```

```
Out[8]: [<matplotlib.lines.Line2D at 0x7f1eb905a3d0>]
```

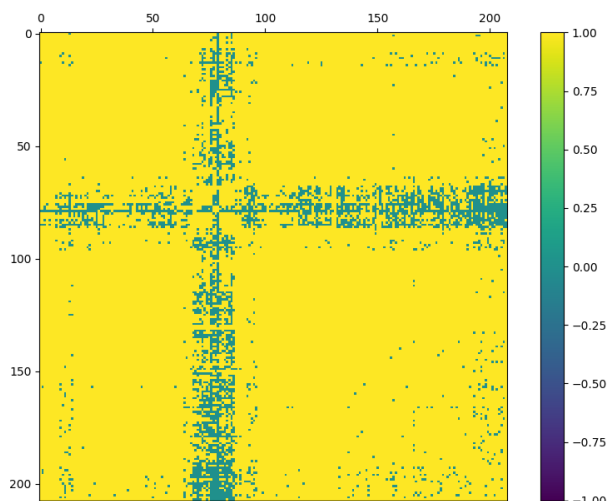


شکل 3- همبستگی دو متغیر BSBM42AA و BSBM26AA

### محاسبه ی همبستگی بین دو متغیر از نوع Nominal

برای محاسبه ی همبستگی بین دو متغیر nominal از chi-square استفاده می کنیم. برای محاسبه ی chi-square، از تابع chi2\_contingency در کتابخانه ی scipy.stats کمک می گیریم. یکی از مقادیر برگشتی این تابع مقدار p\_value است. اگر p\_value کمتر از ۰.۰۵ باشد یعنی دو متغیر همبسته هستند. نمودار heatmap در شکل ۴ آمده است.

با توجه به نمودار، به نظر می رسد درصد زیادی از متغیرها با یکدیگر همبستگی دارند. به همین دلیل، تصمیم بر آن شد تا متغیرهای nominal همبسته حذف نشوند(!).



شکل 4- نمودار همبستگی میان دو متغیر nominal

### ۳-۳ جایگزین کردن نمره ی دانش آموزان با مقادیر عددی

در پایان فرایند پاکسازی داده ها، متغیرهای نمره شامل `finalscore`، `finalscorealgebra`، `finalscoredat`، `finalscoregeo` و `finalscorenum` را با نمره های عددی جایگزین می کنیم. برای این کار، نمره ی A را با ۵، نمره ی B را با ۴، نمره ی C را با ۳، نمره ی D را با ۲ و نمره ی E را با ۱ جایگزین می کنیم. در آخر نمره های null را با میانه پر می کنیم.

### ۴ گزینش متغیرهای موثر بر نمره ی دانش آموزان

برای گزینش متغیرهای موثر بر نمره ی دانش آموزان، از کتابخانه ی `sklearn.feature_selection` استفاده می کنیم. گام نخست آنست که ستون هایی که یک مقدار ثابت دارند (واریانس داده های آنها صفر است) را حذف کنیم. حذف نکردن این متغیرها موجب دریافت هشدارهایی از طرف توابع این کتابخانه می شود. در گام دوم به کمک تابع `SelectKBest`، `k` تا از موثرترین متغیرها بر نمره را انتخاب می کنیم. در اینجا مقدار `k` را برابر یک سوم تعداد متغیرها قرار می دهیم. تابع `SelectKBest`، متدی را برای انتخاب ویژگی ها به عنوان پارامتر دریافت می کند. برای متغیرهای `categorical` (`ordinal` و `nominal`) این متد `chi2` و برای متغیرهای `scale` این متد `ANOVA` (که در کتابخانه ی `sklearn.feature_selection`، `f_classif` نام دارد) می باشد. کار این تابع آنست که میزان همبستگی میان متغیرهای ورودی و متغیر خروجی (در اینجا نمره دانش آموزان) را محاسبه کند و سپس `K` تا از متغیرهایی که بیشترین همبستگی را داشتند، گزارش کند.

## ۵ نتایج

نتایج مربوط به پیاده سازی با فرمت JSON در پوشه ی docs/results به تفکیک سوال آمده است.

## ۶ منابع

[1] Json Browlee, 2020, *How to choose a feature selection method for machine learning*, Access 2 April 2022, <https://machinelearningmastery.com/feature-selection-with-real-and-categorical-data/>

[2]Json Browlee, 2020, *Feature selection for machine learning in python*, Access 2 April 2022, <https://machinelearningmastery.com/feature-selection-machine-learning-python/>