بسمه تعالی

دانشکده مهندسی کامپیوتر

داده کاوی

نام استاد: دکتر حسین رحمانی

پروژه اول

آرمان حیدری

شماره دانشجویی: ۹۷۵۲۱۲۵۲

فروردین ۱۴۰۱

فهرست

پیش‌پردازش داده‌ها

ابتدا تمام داده را در github خودم بارگزاری کرده ام و با url، آن را در pandas.dataframe ذخیره کرده ام. چون کتابخانه pandas برای کار با دیتا بسیار مناسب است و ایزار و سرعت عالی دارد.

بررسی‌های اولیه

- مهم است که تمام داده ها مربوط به دانش آموزان ایرانی باشد. پس چک کرده ایم که IDCNTRY همگی ۳۶۴ باشد.
- نباید دانش آموز تکراری داشته باشیم پس IDSTD های تکراری را حذف کرده ایم.(موردی نبود)
- برای بسیاری از تحلیل ها و مراحل بعدی مانند درخت تصمیم، باید تمام dataframe متشکل از اعداد باشد. پس به جای نمرات A، 5 و به جای نمرات B، ۴ و ... به جای E، 1 گذاشتیم. تا بهترین نمره، بیشترین باشد.

ستون‌های اضافه

- ابتدا تعداد مقادیر مختلف ستون‌ها را میبینیم. ستون های ID مقادیر متمایز زیادی دارند و در تعیین انتروپی درخت تصمیم را دچار مشکل میکنند. همچنین نیازی هم به آن ها برای تحلیل هایمان نداریم. پس همه آن ها را حذف کرده ایم.
- همینطور totalscore و شماره دفترچه هم برای ما مهم نیست چون طبق توضیحات داده شده، finalscore بر اساس totalscore و نمره کلی دفترچه پاسخ داده شده محاسبه شده است. پس می توانیم این دو ستون را نیز حذف کنیم.
- دو تا از ستون ها تماما Nan هستند و هیچ مقداری برایشان ثبت نشده است. (BSDGSEC و BSBGSEC)
- ستون هایی که correlation بسیار زیادی دارند، یعنی مثلا بالای ۰.۷ را اگر جزو نمرات نباشد، فقط یکی را نگه میداریم. چون در واقع آن دو متغیر کاملا وابسته به هم هستند و بعضا یک مفهوم را می رسانند. مثلا دو ستون ITSEC و BSBG01 که همواره برابر هستند و نمایانگر جنسیت هستند.

مقادیر null

- وجود خانه‌های های خالی در داده می تواند باعث بروز مشکلاتی در تحلیل های ما شود. ابتدا بررسی کردیم که به طور کلی هر ستون چند مقادیر null دارد. که اگر تعدادشان زیاد است سعی کنیم clustering و کارهای مشابه انجام دهیم. اما چون حداکثر ۳۰ تا بود و ما در کل 5900 رکورد داریم، نیازی به این کار نیست.
- دیدیم که اکثر مقادیر nan در نمرات است و آن ها را حذف کرده ایم.
- به جز آن‌ها بررسی میکنیم و میبینیم که دو سطر هم مشکلاتی دارند پس آن ها را نیز حذف کردیم و دیگر در Dataframe خانه خالی نداریم.

ستون‌های ایراددار

- با مطالعه codebook میبینیم که وقتی داده دچار مشکلی در ثبت یا سیستمی یا ... شده است، مقدار ۹ یا ۹۹ یا ۹۹۹ یا ۹۹۹۹ یا ۹۹۹۹۹ یا ۹۹۹۹۹۹ برای آن ثبت شده است. ابتدا بررسی کردیم که هر ستون جندبار دچار مشکل شده است.

```
('BCBG18', 319),
('BTBM22BD', 341),
('BSBM26AA', 365),
('BSBM42AA', 365),
('BTBM22BG', 378),
('BSBM26BA', 389),
('BSBM42BA', 389),
('BTBM22BC', 389),
('BTBM22BE', 390),
('BCBG03B', 456),
('BTBM22BF', 480),
('BCDGSBC', 518),
('BCBG21B', 778),
('BCBG21C', 842),
('BSBM27BA', 1650),
('BSBM43BA', 1650),
('BSBM27AA', 1654),
('BSBM43AA', 1654)]
```

- این مقادیر را مرتب کرده ایم و پر ایرادترین ها را در عکس بالا میبینید. مواردی که بیش از ۱۰ درصد خزا در ثبت دارند را چون کاملا میتواند تحلیل را دچار خطا کند کناز گذاشتیم. یعنی از BCBG21B به پایین.

- البته سایر خرابی ها را هم موقع تحلیل های بعدی برای هر ستون خاص کنار میگذاریم. اما در اینجا اگر بخواهیم تمام رکودهای حاوی خطا در ثبت را کنار بگذاریم، چه در ردیف ها این کار را کنیم و چه در ستون ها مجبور به حذف عمده ی داده مورد بررسی میشویم. که اصلا کار درستی نیست.

پس از این مراحل پیش پردازش، داده تمیزی داریم که به صورت زیر است :

	finalscore	finalscorealgebra	finalscoredat	finalscoregeo	finalscorenum	BSBG01	BSBG03	BSBG04	BSBG05A	BSBG05B
0	2.0	5.0	2.0	1.0	1.0	1.0	2.0	3.0	1.0	1.0
1	2.0	2.0	2.0	1.0	2.0	1.0	1.0	3.0	1.0	1.0
2	4.0	4.0	2.0	3.0	4.0	1.0	1.0	4.0	1.0	1.0
3	1.0	2.0	1.0	1.0	1.0	1.0	1.0	2.0	2.0	2.0
4	2.0	2.0	3.0	1.0	2.0	1.0	1.0	2.0	1.0	1.0
...	...	...	...	...	...	...	...	...	...	...
5975	2.0	2.0	2.0	2.0	2.0	2.0	2.0	2.0	2.0	2.0
5976	2.0	1.0	1.0	3.0	1.0	1.0	4.0	1.0	2.0	1.0
5977	1.0	1.0	1.0	2.0	1.0	1.0	3.0	2.0	1.0	2.0
5978	2.0	1.0	1.0	3.0	1.0	2.0	1.0	5.0	1.0	1.0
5979	1.0	1.0	2.0	1.0	1.0	2.0	3.0	5.0	2.0	1.0

5931 rows × 324 columns

نرمال‌سازی داده

در این بخش دو تابع مشهور را پیاده سازی کرده ام. که شامل min_max و Z_score است. البته در ادامه با به نظر نتایج z_score بهتر بود و آن ها را در نظر گرفتم.

Min_max

این نرمال سازی یکی از رایج ترین روش ها برای عادی سازی داده ها است. برای هر ویژگی، حداقل مقدار آن ویژگی به ۰، حداکثر مقدار به ۱، و هر مقدار دیگر به اعشار بین ۰ و ۱ تبدیل می شود. در اینجا تابعی پیاده سازی کرده ایم که یک dataframe را میگیرد و تمام ستون های آن را iterate میکند و با فرمول زیر هر ستون را نرمال میکند. در نهایت dataframe جدید را باز می گرداند.

$$x' = \frac{x - \min(x)}{\max(x) - \min(x)}$$

Z_score

نرمال سازی z_score به فرآیند نرمال سازی هر مقدار در یک مجموعه داده اشاره دارد به طوری که میانگین همه مقادیر ۰ و انحراف استاندارد ۱ باشد. این به مدیر داده اجازه می دهد تا احتمال وقوع یک امتیاز در توزیع عادی داده ها را درک کند. z-score یک مدیر داده را قادر می‌سازد تا دو امتیاز مختلف را که از توزیع‌های نرمال متفاوت داده‌ها هستند، مقایسه کند.

در اینجا تابعی که پیاده سازی کرده ایم یک dataframe را میگیرد، تمام ستون های iterate میکند و فرمول این نرمال سازی را برروی هرکدام از ستون ها اعمال می کند. در نهایت dataframe جدید را باز می گرداند.

$$Z = \frac{x - \mu}{\sigma}$$

درخت تصمیم

- تابع visualize_tree، درخت را میگیرد و فایل تصویر آن را می سازد. در این سوال چون لزومی نبود، و اجرایش زمان بر بود اجرای آن را کامنت کرده ام اما به راحتی میتوانید آن را اجرا کنید.
- در تابع desicion_tree، یک dataframe را به همراه ستون مقصد و تعدادی ستون که نباید در درخت بررسی کنیم، گرفته ایم.
- این تابع با استفاده از DesicionTree موجود در کتابخانه scikit_learn.tree کار میکند.
- ابتدا باید ویژگی هایی که بیش از ۸ یا کمتر از ۲ تا مقدار متفاوت دارند را از تحلیل خارج کنیم. چون در درخت تصمیم ایجاد خطا میکنند و آن را بی معنی میکنند.
- با کنار گذاشتن این ویژگی ها و همچنین ویژگی هایی مانند سایر نمرات که در finalscore آشکارا موثرند ولی تاثیرشان برای ما مهم نیست، ورودی X را میسازیم.
- ورودی y را هم برابر ویژگی مورد بررسی میگذاریم که در دو سوال اول finalscore و در سوال سوم ۴ نمره ی دیگر است.
- به این ترتیب درخت تصمیم را بر اساس x و y تشکیل داده ایم. سپس استفاده ای که از آن میکنیم، dimension reduction است. چون همانطور که میدانیم ۳۰۰ ستون بسیار زیاد است و خیلی از آن ها در ویژگی مدنظر ما تاثیر جندانی ندارند که نیاز به بررسی داشته باشد. پس ما ۳۰ ویژگی موثر (عمق ۳۰ درخت تصمیم) را به صورت یک لیست best_attributes بازگردانده ایم.
- همچنین ویژگی هایی که به علت مقادیر متمایز زیاد کنار گذاشتیم (حدودا ۲۰ تا) را نباید از تحلیل هایمان خارج کنیم. چون همگی از نوع scale هستند و مقادیر متفاوتی دارند، با قراردادنشان در correlation میتوانیم به خوبی میزان ارتباطشان با ویژگی مقصد را بفهمیم. پس در کل لیست هایی شامل بهترین متغیر ها (در اینجا ۳۰ تای اول که باعث بیشترین کاهش انتروپی در ویژگی مدنظر میشوند)، به همراه متغیرهایی که از درخت تصمیم حذف کردیم را برگردانده ایم.

تحلیل correlation

- تابع correlation_analysis یک dataframe را به همراه ویژگی مدنظر مان دریافت میکند. با استفاده از تابع ()corr موجود در pandas، matrix correlation را تشکیل میدهد.
- برای تشکیل این ماتریس، ابتدا تمام ستون های dataframe را پیموده ایم و هر کدام را با کنار گذاشتن مقادیر مشکل دار (۹ یا ۹۹ یا ... یا ۹۹۹۹۹۹) و همچنین نرمال سازی z-score وارد correlation کرده ایم و ضریب ارتباط آن با ویژگی مدنظر (اکثرا finalscore) را حساب کرده ایم. در نهایت لیستی مرتب شده از کوچک به بزرگ بازمیگرداند که میزان وابستگی ستون مدنظر با هرکدام از ستون های جدول نشان می دهد.
- باید دقت کنیم که اعداد منفی و مثبت بازگردانده شده کاملا معنی دار هستند و درواقع جون در این پروژه به دنبال میزان وابستکی هستیم، قدر مطلق این اعداد هرچقدر بیشتر باشد به معنی وابستگی بیشتر است. و علامت عدد وابستگی مستقیم یا معکوس را به ما نشان میدهد.

- تابع plot_relation، را برای بررسی چشمی رابطه بین دو ستون توسعه میدهیم. درواقع با دریافت یک جدول و نام دو تا از ستون های آن، وابستگی این دو ستون را به هم نشان میدهد. ورودی y را در محور عمودی قرار میدهد و محور x هم ورودی x است.
- تمام مقادیر جدول را بر اساس مقادیر متمایز X، تقسیم میکند و میانگین y در هر کدام از مقادیر x را در آن مقدار plot میکند. بدین ترتیب میتوانیم روند زیاد یا کم شدن متغیر y با تغییر x را به طور کلی (میانگینی) ببینیم.

پاسخ سوالات

حالا که پیش پردازش کردیم و توابع مورد نیازمان برای تحلیل را گسترش دادیم، به سادگی با ساختن dataframe مورد نیاز و تعیین ورودی صحیح توابع در هر بخش، سوالات را پاسخ میدهیم.

۱. مهم‌ترین عوامل بر finalscore

- جدول پیش پردازش شده را به صورت کامل در این بخش نیاز داریم چون محدودیتی نیست. فقط ۴ ستون نمره به جز Finalscore را حذف میکنیم چون قاعدتا وابستگی زیادی بین آن هاست و ما دنبال آن نیستیم. (به ما knowledge نمیدهد).
- ستون مقصد برای تمام توابع این بخش finalscore است.
- ابتدا درخت تصمیم میزنیم و بهترین ویژگی ها را به همراه ویژگی هایی که مقادیر متماز زیاد دارند (scale) فقط نگه میداریم:

```
features with more than 8 unique values: ['BSBGSB', 'BSBGSCM', 'BSBGSLM', 'BSBGSVM', 'BSBGICM', 'BCDGTIHY',

after remove extra features:
 BTDMGEO     8
BCBG05A     8
BSBM42BA    7
BSDGEDUP    7
BSBG07      7
           ..
BTBG02      2
BTBG05F     2
BTBG05G     2
BTBG05H     2
BSBG01      2
Length: 296, dtype: int64
```

- بر اساس درخت تصمیم پیاده سازی شده، ویژگی های مهم به این شکل هستند که بر اساس استفاده در کمترین عمق درخت مرتب شده اند:

```
the most important features (features on top of the tree):
          importance
BSBG07      0.035029
BSDGSCM     0.025975
BSDGEDUP    0.018016
BSBG04      0.016037
BCBG05A     0.013228
BSBG13D     0.013067
BSBG13E     0.012869
BSBM19H     0.012116
BSBM19G     0.012104
BSBM15      0.011853
BSBM18F     0.011030
BSBM18B     0.010888
BSBM16H     0.010612
BSBM19B     0.010417
BSBM19F     0.009758
BSBM26AA    0.009673
BSBG13B     0.009667
BSBM19E     0.009657
BSBM42BA    0.009605
```

- پس جدول جدیدمان را میسازیم. که میبینیم از سطرها چیزی حذف نکرده ایم ولی تعداد ستون ها به شدت کاهش یافته و dimension reduction داشته ایم.

	finalscore	BSBG07	BSDGSCM	BSDGEDUP	BSBG04	BCBG05A	BSBG13D	BSBG13E	BSBM19H	BSBM19G	...	BCBG18	BCBG06B	BTBM
0	2.0	5.0	3.0	2.0	3.0	1	3.0	3.0	3.0	3.0	...	20	270	
1	2.0	4.0	3.0	1.0	3.0	1	3.0	3.0	1.0	2.0	...	20	270	
2	4.0	6.0	1.0	1.0	4.0	1	4.0	4.0	4.0	1.0	...	20	270	
3	1.0	6.0	3.0	4.0	2.0	2	1.0	1.0	1.0	3.0	...	1	210	2
4	2.0	3.0	3.0	3.0	2.0	2	4.0	2.0	4.0	2.0	...	1	210	2
...	...	...	...	...	...	...	...	...	...	...	...	...	...	
5975	2.0	4.0	2.0	3.0	2.0	7	2.0	3.0	3.0	1.0	...	9	270	
5976	2.0	3.0	2.0	5.0	1.0	6	1.0	1.0	1.0	2.0	...	5	250	1
5977	1.0	6.0	2.0	2.0	2.0	6	1.0	4.0	1.0	2.0	...	5	250	1
5978	2.0	6.0	3.0	1.0	5.0	4	1.0	1.0	1.0	4.0	...	1	270	1
5979	1.0	9.0	2.0	4.0	5.0	4	1.0	4.0	1.0	1.0	...	1	270	1

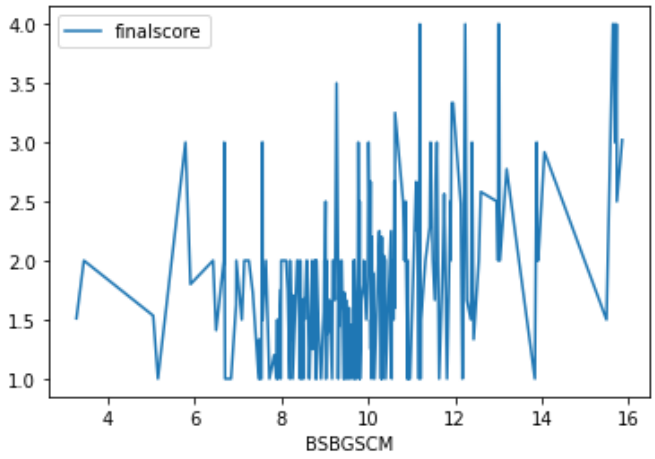
5931 rows × 54 columns

- و در نهایت جدول جدید را با ستون مقصد به تابع correlation_analysis میدهیم تا وابسته ترین ویژگی ها را بیابیم:

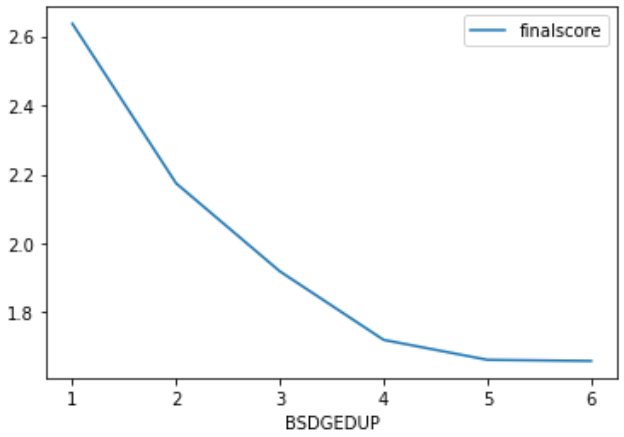
```
[('BSDGSCM', -0.3846731112089682),
 ('BSDGEDUP', -0.3326660695788252),
 ('BSBM19A', -0.2763613671274829),
 ('BCBG05A', -0.2666288236472383),
 ('BSBM19F', -0.2576769528568183),
 ('BSBM19G', -0.25701648333164256),
 ('BSBM15', -0.21303178097520667),
 ('BSBG03', -0.16457825169001686),
 ('BSBM17B', -0.1300063734650294),
 ('BSBM20E', -0.12279558847433497),
 ('BSBM16H', -0.1202182708275955),
 ('BSBG12B', -0.10289100284959549),
 ('BSDAGE', -0.10218872222579273),
 ('BSBG13E', -0.0982732434932951),
 ('BTBG11', -0.09214366192999246),
 ('BSBG13D', 0.00343318663332102),
 ('BTDMALG', 0.011785083871524446),
 ('BSBM17E', 0.013661308033310124),
 ('BSBG13B', 0.014390074896699414),
 ('BTDMDAT', 0.015092138530943105),
 ('BCBG19', 0.027867528205948018),
 ('BSBM26AA', 0.02905765925957637),
 ('BSBM42AA', 0.02905765925957637),
 ('BSBGSSB', 0.029466043340394953),
 ('BSBM18F', 0.08594176282531692),
 ('BTBG01', 0.08931340303185353),
 ('BSBGSB', 0.09503057332820554),
 ('BCBG06A', 0.10029319943704622),
 ('BCBG18', 0.11499397902003339),
 ('BSBGSVM', 0.11668288904235004),
 ('BSBGICM', 0.11685692614278986),
 ('BSBM42BA', 0.13526177718806007),
 ('BSBM26BA', 0.13526177718806007),
 ('BTBM14', 0.13640193416856228),
 ('BSBG06A', 0.18328758868882009),
 ('BCBG07', 0.18862056265384539),
 ('BSBG06B', 0.21195651816204714),
 ('BCBG06B', 0.21882354096584944),
 ('BCDGTIHY', 0.22110594185895738),
 ('BSBM19E', 0.22593052613746575),
 ('BSBGSLM', 0.24488785147771336),
 ('BSBG07', 0.27577744244065916),
 ('BSBM19B', 0.333017655412876),
 ('BSBG04', 0.33526256395504267),
 ('BSBM19H', 0.33802508511683577),
 ('BSBGHER', 0.3860257332236845),
 ('BSBGSCM', 0.4107773735563639)]
```

- عکس سمت چپ نمایانگر روابط معکوس زیاد و عکس راست نمایانگر بیشترین روابط مستقیم است. با مقایسه قدر مطلق ها، میتوانیم ۳ ویژگی بسیار مهم برای جواب این مسئله را موارد زیر بگوییم:

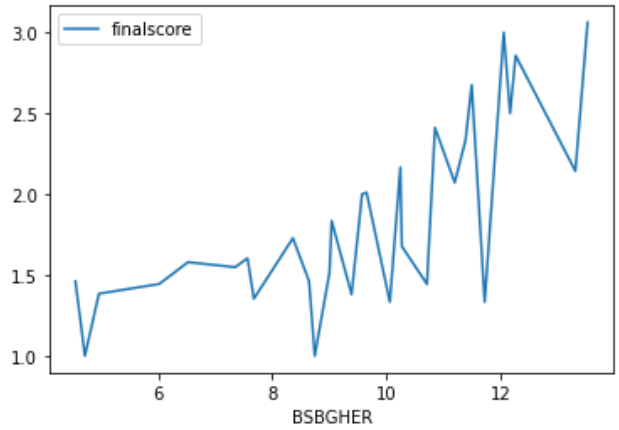
رابطه مستقیم:Student Confident in Mathematics/SCL



رابطه معکوس:Parents' Highest Education Level



رابطه مستقیم:Home Educational Resources/SCL



۲. بررسی عوامل برای دختر/پسر

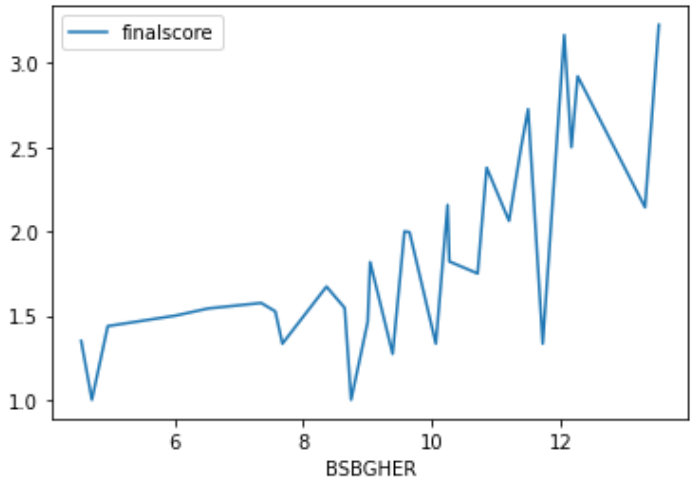
برای این بخش به راحتی همان مراحل سوال قبل را این بار برای دو جدول جداگانه انجام دادیم. که یکی برای نیمه دانش آموزان پسر و یکی برای نیمه دختر است. این کار را با یک کوئری روی ویژگی جنسیت انجام داده ایم. و ستون مقصد همان finalscore است. جدولی که برای پسرها تشکیل میدهیم:

	finalscore	finalscorealgebra	finalscoredat	finalscoregeo	finalscorenum	BSBG01	BSBG03	BSBG04	BSBG05A	BSBG05B
5	1.0	2.0	2.0	1.0	1.0	2.0	4.0	3.0	2.0	2.0
6	1.0	1.0	2.0	1.0	1.0	2.0	4.0	2.0	1.0	1.0
7	2.0	2.0	1.0	1.0	2.0	2.0	1.0	2.0	1.0	2.0
11	1.0	2.0	2.0	1.0	1.0	2.0	2.0	1.0	1.0	1.0
12	2.0	2.0	1.0	1.0	2.0	2.0	3.0	3.0	1.0	2.0
...	...	...	...	...	...	...	...	...	...	...
5971	4.0	4.0	3.0	3.0	2.0	2.0	1.0	2.0	1.0	2.0
5972	3.0	3.0	2.0	4.0	2.0	2.0	1.0	3.0	1.0	1.0
5975	2.0	2.0	2.0	2.0	2.0	2.0	2.0	2.0	2.0	2.0
5978	2.0	1.0	1.0	3.0	1.0	2.0	1.0	5.0	1.0	1.0
5979	1.0	1.0	2.0	1.0	1.0	2.0	3.0	5.0	2.0	1.0

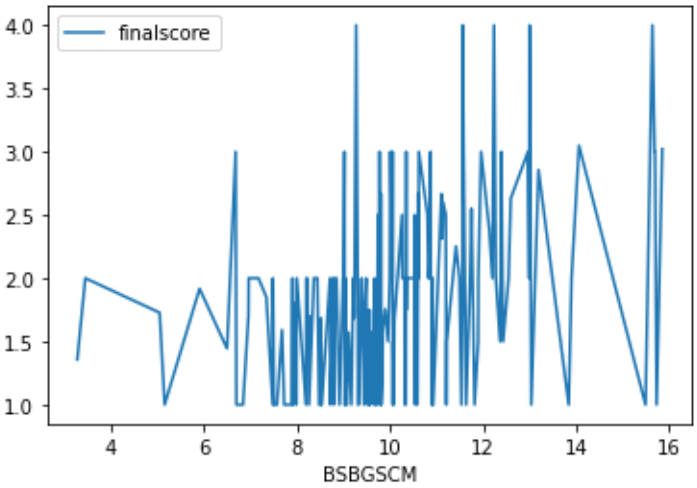
3004 rows × 245 columns

که میبینیم تعداد سطرها تقریبا نصف قبل شده است. حالا با همان مراحل قبلی، به این متغیر های موثر (به ترتیب) میرسیم:

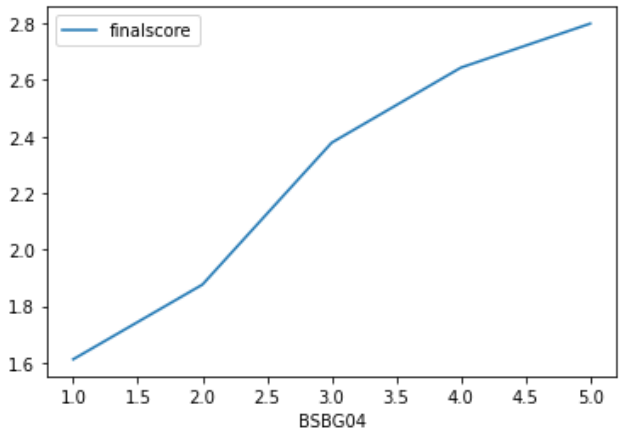
رابطه مستقیم:Home Educational Resources/SCL



رابطه مستقیم:Student Confident in Mathematics/SCL



رابطه مستقیم:GEN\AMOUNT OF BOOKS IN YOUR HOME



به طور مشابه برای دخترها را محاسبه میکنیم. و فقط کوئری اول را تغییر میدهیم. دیگر تصاویر مراحل را برای جلوگیری از شلوغی گزارش نیاورده ام. ۳ متغیر اول به ترتیب موارد زیر بودند:

رابطه مستقیم:Student Confident in Mathematics/SCL

رابطه مستقیم:MATH\AGREE\MATHEMATICS HARDER FOR ME

رابطه معکوس:MATH\AGREE\LEARN QUICKLY IN MATHEMATICS

که میبینیم عوامل زمینه ای در دو جنسیت تا حدی متفاوت عمل کرده اند. البته در هر دو مورد، این ویژگی ها جزو وابستگی های زیاد بودند اما میزان آن ها کمی متفاوت است. مثلا میزان منابع منزل برای پسرها اهمیت بیشتری دارد، و برای دخترها اعتماد به نفس در مقابله با ریاضی و همچنین سرعت یادگیری آن مهم تر است.

۳. بررسی عوامل بر حیطه‌های محتوایی

در این بخش هم مشابه بخش اول، فقط ۴ بار همان روند را برای ۴ ستون مقصد متفاوت طی میکنیم. تا برای هرکدام از محتوا حساب کرده باشیم. برای هرکدام از ستون ها، سه متغیری که وابستگی زیادی وجود داشته است را در زیر آورده ام:

- جبر: BSBGSCM(مستقیم)، BSBGHER(مستقیم)، BSDGEDUP(معکوس)
- داده‌ها: BSBGHER(مستقیم)، BSBGSCM(مستقیم)، BSBG04(مستقیم)
- هندسه: BSBGSCM(مستقیم)، BSBGHER(مستقیم)، BSBM19H(مستقیم)
- اعداد: BSBGSCM(مستقیم)، BSBGHER(مستقیم)، BSBM19H(مستقیم)

میبینیم که کمی معیارها متفاوت است. و ضرایب correlation هم بعضا بین حیطه های محتوایی مختلف اختلاف قابل توجهی دارد. که در نوتبوک با جزئیات همه ی ضرایب را مرتب شده میتوانید ببینید.