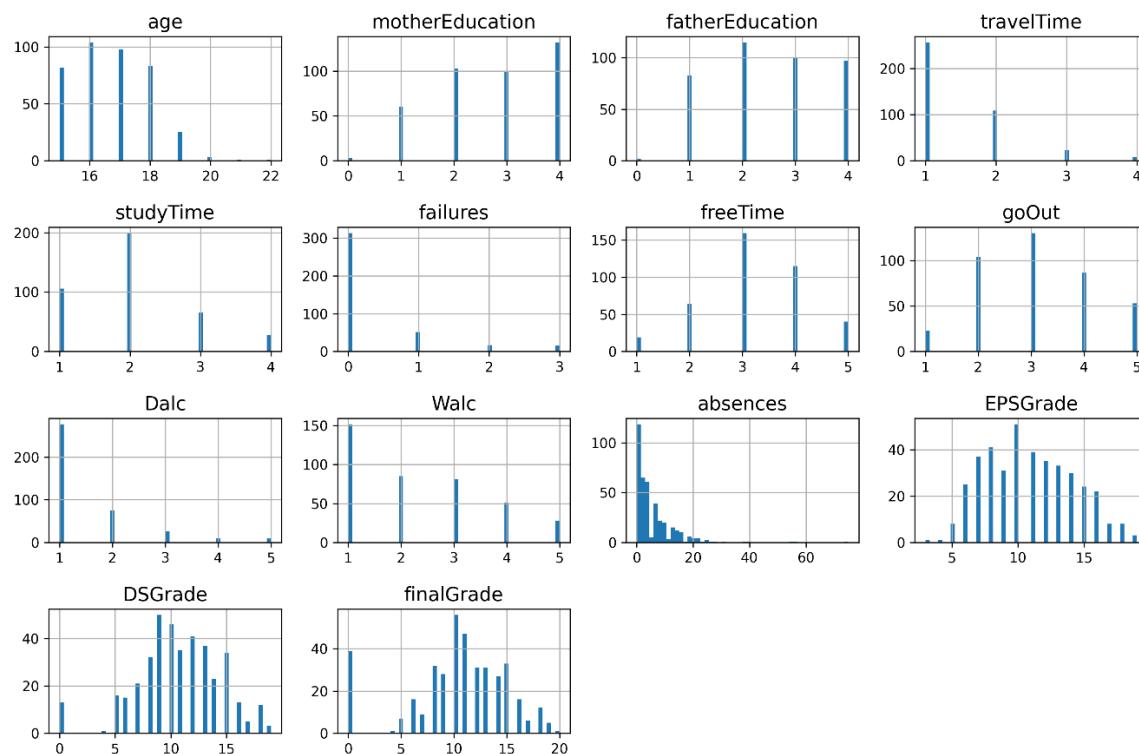


برای انجام پروژه های ماشین لرینگی میشه از راه های مختلفی استفاده کرد و رسیدن به نتیجه مطلوب رو به شیوه های متفاوتی روایت کرد. من در این پروژه از شیوه روانی آقای Aurélien Géron در کتاب Hands-on machine learning استفاده کرده ام.

so lets go! 😊

مرحله ۱) مشاهده و دریافت داده ها:



مشکلی که همین اول با مشاهده نمودار داده فیچرها متوجه میشیم اینه که بعضی فیچرها از توزیع شبه نرمالی پیروی می کنند به جز در یک bar! یعنی احتمالاً این bar جوری تعریف شده که معنی other میده یا همه دیتاهایی که در هیچ کلاس تعریف شده ای جانمی گرفتند رو به صورت فله ای برداشت و ریختن توی اون ستون! مثلاً نمونه های زیر:

- ستون ۴ در نمودار motherEducation: ستون other برای شغل پدر و مادر شامل دسته وسیعی از شغل ها به حساب می آید که دارد با سایر دسته های شغلی رقابت می کند. به نظر عادلانه نیست و می شود این ستون را حذف کرد.

- ستون ۴ در نمودار fatherEducation

- ستون ۰ در نمودارهای DSGrade and finalGrade: به نظر میرسه يه عده ای اصلاً این درس ها رو پاس نکردن. حالا یا حذف کردن که صفر ثبت شده یا نتونستن حذف کنن و نمره صفر ثبت شده.

مشکل بعدی اینه که ترجیح ما اینه که نمودارهای bell shape باشند ولی نمودارهای مربوط به age و absences چولگی به چپ دارند و این مسئله باید هندل بشه یه جوری. چون فرایند ترین شدن مدل رو کند می کنه.

در ادامه پیاده سازی مدل و feature engineering سعی می کیم این مشکلات را در دقتاست موجود حل کنیم. 🦜

جایگزینی نمره های صفر در DSGrades : برای این جایگزینی آپشن های مختلفی وجود دارد. مثلا:

۱. می تونیم با میانه نمرات جایگزینشون کنیم.
۲. می تونیم یک مدل RandomForestRegressor تولید کنیم که براساس feature های EPSGrade و failures بیاد نمره های missing رو predict کنه. من از این آپشن استفاده کردم
۳. Probability Distribution Matching: یعنی توزیع احتمال متغیر تصادفی داده های غیرصفر رو مشخص می کنیم و براساس اون ها برای نمره های صفر دیتاساینس یک نمره انتخاب می کنیم.

چون از Regressor استفاده کردیم نمرات تخمین زده شده به صورت float هستند و باید این نمرات رو به نمره صحیح تبدیل کنیم.

چند آپشن داریم:

۱. خیلی ساده نمرات رو round کنیم.

۲. این هم تقریب میزنه ولی به صورت احتمالی و دقیق تره.

```
def probabilistic_round(x):
    return int(x) + (1 if np.random.rand() < (x - int(x)) else 0)

df['DSGrade'] = df['DSGrade'].apply(probabilistic_round).clip(0, 20)
```

۳. می تونیم کلا بریم از داده های نمره به صورت کلاس استفاده کنیم و از RandomForestClassifier استفاده کنیم. احتمالاً این مورد دقت بسیاری بالایی بهمون بده ولی داره با ذات نمره که عدد است به صورت آبجکت رفتار می کنه و این درست نیست. اگر داده های واقعی به سوال بدھیم مثل نمره های ثبت شده در پهستان، دیگر مدل classifier حرفی برای گفتن ندارد. در واقع این مورد یک راهکار نیست بلکه یک فریب است.

```
from sklearn.ensemble import RandomForestClassifier

# Convert grades to integer categories
y = grading['DSGrade'].round().astype(int).clip(0, 20)

# Train classifier instead of regressor
clf = RandomForestClassifier()
clf.fit(X_train, y_train)
predicted_grades = clf.predict(X_test)
```

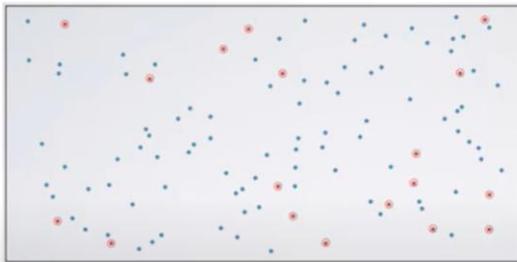
: Binned Probability option .۴

```
# Get probability distribution for each possible grade (0-20)
probabilities = model.predict_proba(X_test) # If using classifier
```

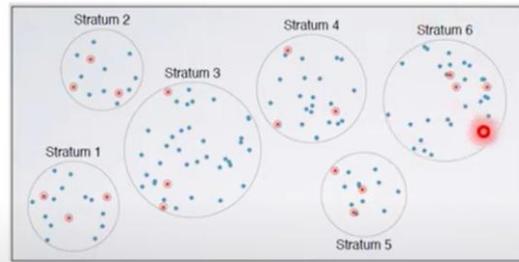
```
# Choose grade with highest probability
grading['DSGrade'] = np.argmax(probabilities, axis=1)
```

موضوع دیگری که وجود دارد در انتخاب داده تست و ترین این است که باید نمونه گیری به صورت کارآمد انجام شود. مثلاً نمونه فقط شامل دخترها یا فقط پسرها نباشد و نسبت درستی از این دخترها و پسرها در نمونه وجود داشته باشد. به همین دلیل برای نمونه گیری خود از روش StratifiedSampling استفاده می کنیم:

Methods of Sampling

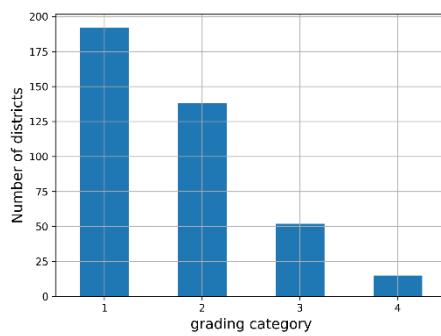


Simple Random Sampling

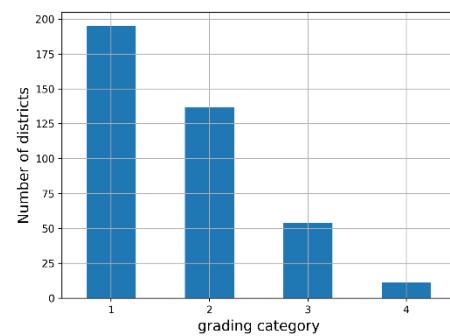


Stratified Sampling

مثلاً فیچر EPSGrade و DSGrade برای ما فیچرهای مهمی هستند و باید براساس آنها stratified sampling را انجام دهیم:



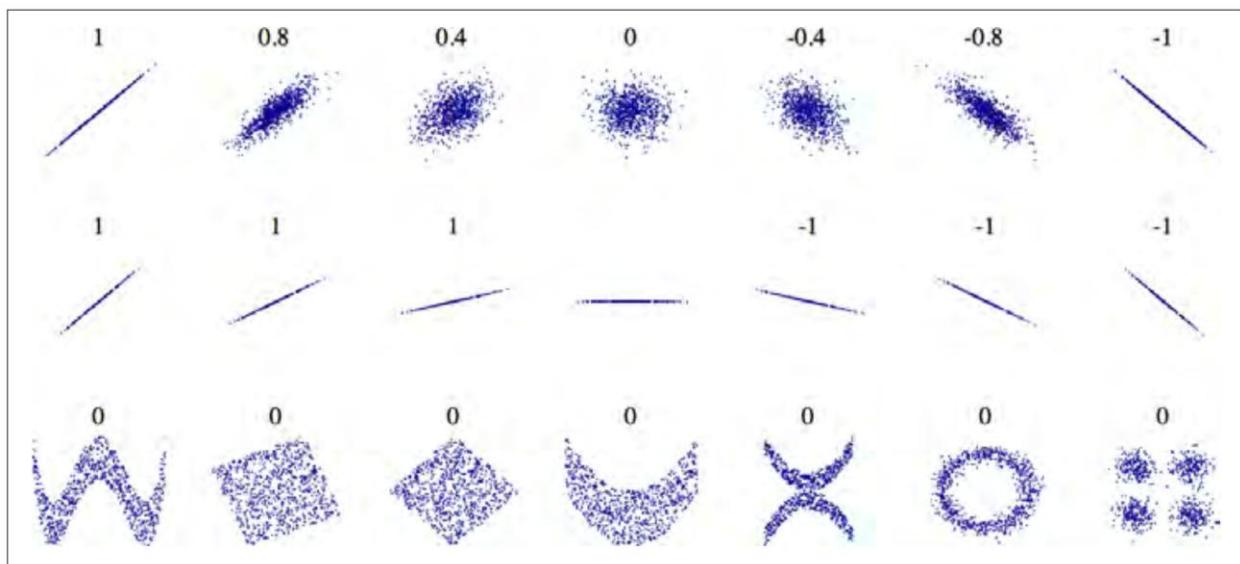
Based on DSGrade



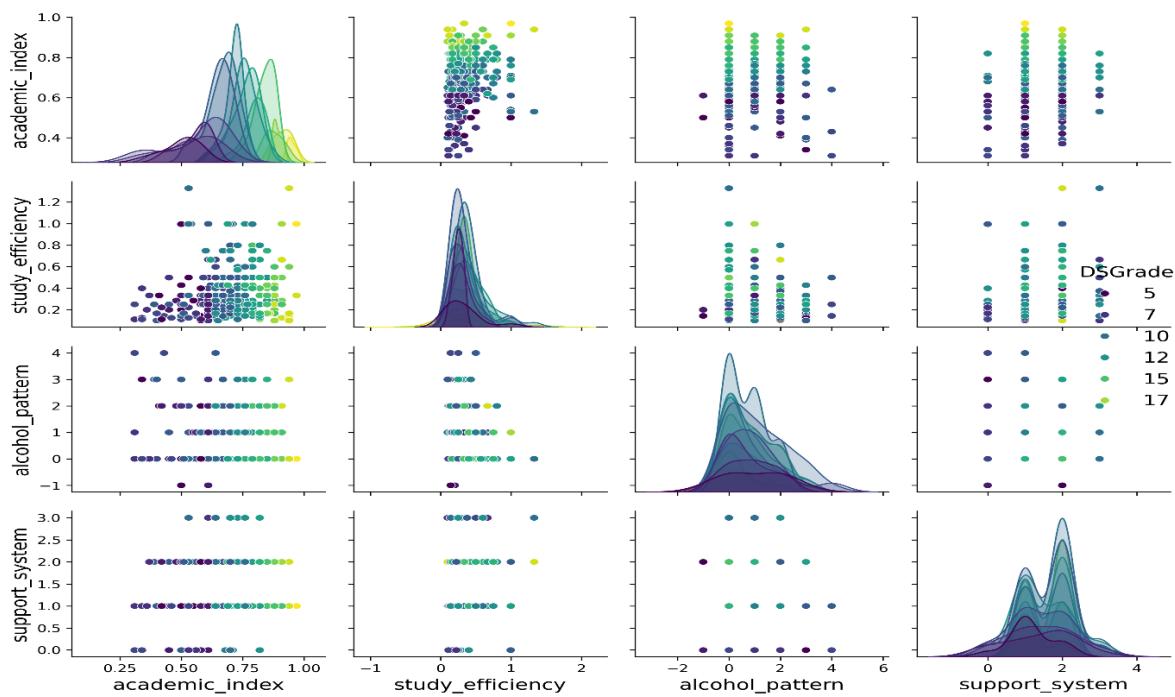
Based on EPSGrade

بنابراین باید براساس فیچرهای مهمی که داریم، stratified sampling انجام بدیم. لازم است که feature های مهمتر و با همبستگی بیشتر رو پیدا کنیم که از همه دسته های اون فیچر لایه هایی در نمونه انتخابی برای train و test داشته باشیم.

در درس های آمار و احتمال و دیتاساینس با ضریب همبستگی آشنا شدیم. ضریب همبستگی، ارتباط خطی بین دو متغیر را نشان می دهد. در کتاب *hands-on machine learning* عکس زیر آورده شده است:



توی feature engineering انجام شده ویژگی های ساختیم که همبستگی خطی بیشتری به ما می دهند، نسبت به ویژگی های خام اولیه که در دیتاست وجود داشت:



توی مرحله بعد میایم لیبل هایی که قراره پیش بینی کنیم رو از predictor ها جدا می کنیم. یعنی یک کپی می سازیم که توی اون دیگه نمره هایی که قراره پیش بینی بشن نباید به مدل مورد آموزش نمایش داده بشه. در واقع در این مرحله y , X را می سازیم و در ادامه به مدل های ماشین لرنینگی خود X و y را جدا می دهیم تا بتوانند پیش بینی لازم را انجام دهند.