به نام خدا

# پروژه داده‌کاوی

اعضای گروه:

سارا سلطانی گردفرامرزی

بهار ۱۴۰۲-۱۴۰۱

# معرفی حوزه

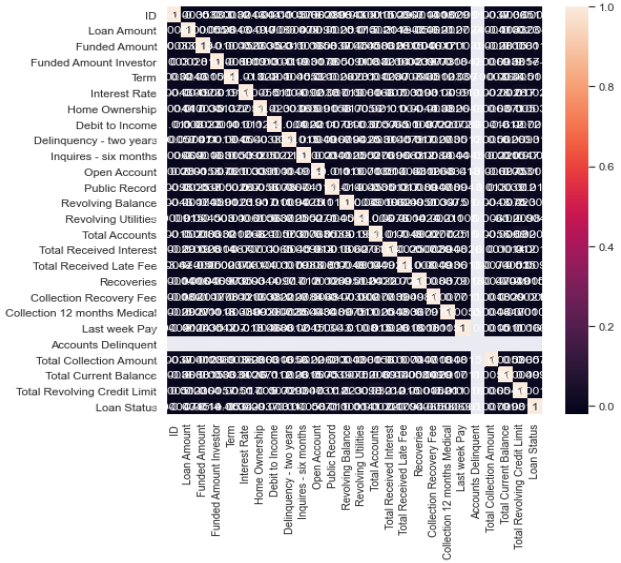
- پیش بینی تاییدیه وام بانکی:

در این مسئله پیش‌بینی کنیم که آیا یک فرد وام خود را پرداخت خواهد کرد یا خیر تا بتوان تا حدودی جلوی خسارت وارده به بانک‌ها را گرفت.

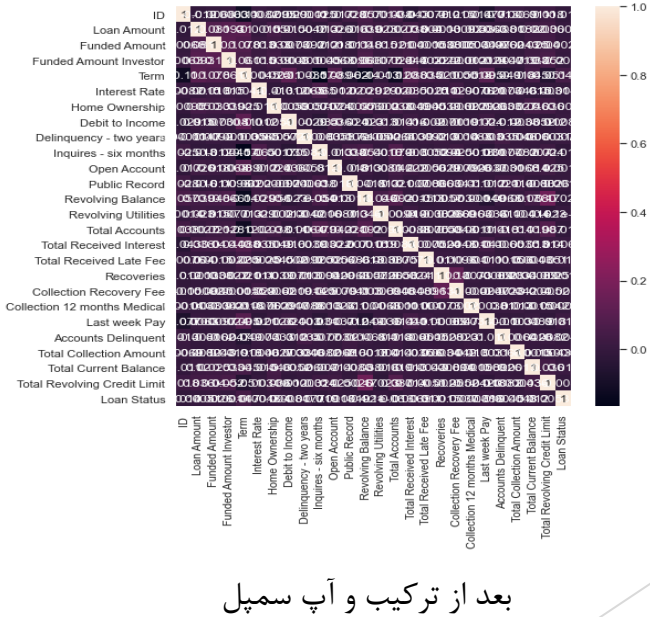
- مشخصات هر متقاضی
  - مدت زمان برگشت وام
  - مبلغ وام
  - مقدارخالص وام (مبلغ وام منهای هزینه‌های اولیه)
  - مدت زمان استخدام
  - میزان بهره
  - ...

# بالانس اولیه داده‌ها

- عدم وجود کورولیشن بین داده‌های آموزش
- صفر بودن تمامی مقادیر هدف در داده‌های تست
- ترکیب داده‌های آموزش و تست
- ۸۰٪ کل داده برای آموزش، ۲۰٪ تست
- توزیع ابتدایی داده‌های آموزش ۸۳٪ به ۱۷٪ درنتیجه up sampling




قبل از ترکیب و آپ سمپل




بعد از ترکیب و آپ سمپل

# اکتشافات داده‌ای

مجموعه‌ی دادگان شامل:

- مشخصات ۸۷۱۱۲ نفر متقاضی وام برای داده‌های آموزشی جمع‌آوری شده است
- تقاضا ۷۲۱۱۲ نفر رد و تقاضا ۱۵۰۰۰ نفر پذیرفته شده‌است
- مشخصات ۱۹۲۷۶ نفر متقاضی وام برای داده‌های تست جدا شده است
- تقاضا ۱۸۰۲۳ نفر رد و تقاضا ۱۲۵۳ نفر پذیرفته شده‌است
- ۳۴ ویژگی پیش بینی کننده و ۱ ویژگی هدف باینری داریم
- داده‌ی null یا گمشده نداریم
- حذف ویژگی‌های آیدی و برنامه پرداخت به دلیل مقادیر یکتا و تکراری برای هر رکورد

# نحوه حل مسئله

- شناسایی دادهای پرت و حذف آن‌ها
- نرمال‌سازی و استاندارسازی ویژگی‌ها
- بررسی امکان دسته بندی مجدد متغیرهای دسته‌ای
- تبدیل متغیرهای دسته‌ای به عددی برای تحلیل آسان‌تر و استفاده در آموزش مدل
- سبد بندی متغیرهای عددی و بررسی تاثیر آن روی همبستگی ویژگی‌ها
- بررسی امکان اضافه کردن متغیر جدید به دیتاست و تاثیر آن روی همبستگی ویژگی‌ها با متغیر هدف
- آموزش دادن مدل های مختلف با استفاده از داده آموزش
- ارزیابی و انتخاب مدل با توجه به مجوعه کراس ولیدیشن
- ارزیابی نتیجه پیش بینی داده‌های تست با استفاده از مدل انتخابی

# تبدیل متغیرهای دسته‌ای به عددی

- تبدیل متغیرها با روش Label Encoder
  - متغیرهای دسته ای:
  - متغیرهای اسمی:

Employment Duration

Verification Status

Loan Title

  - متغیرهای ترتیبی:

Grade

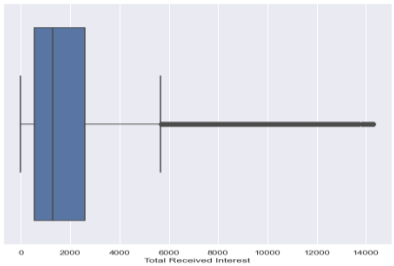
Sub Grade

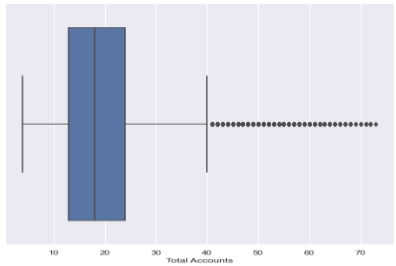
  - متغیرهای باینری:

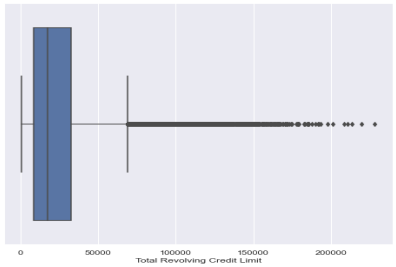
Initial List Status

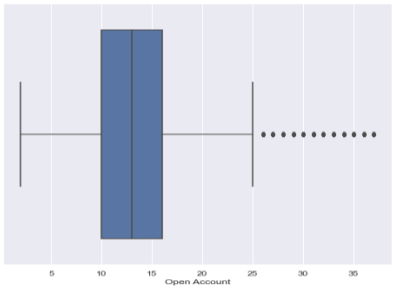
Application Type

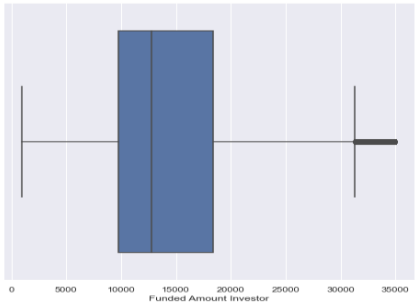
# شناسایی داده‌های پرت

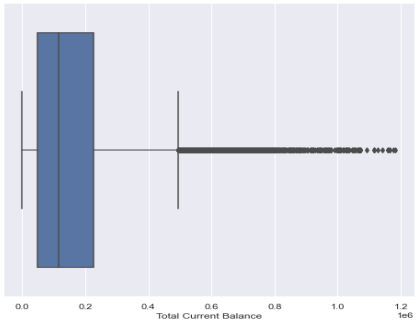


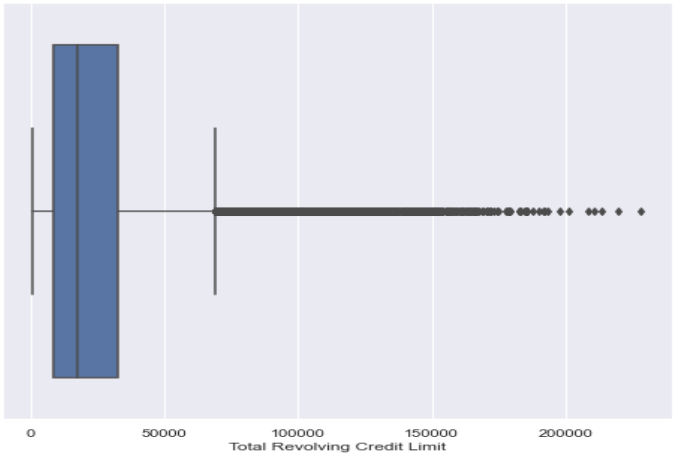



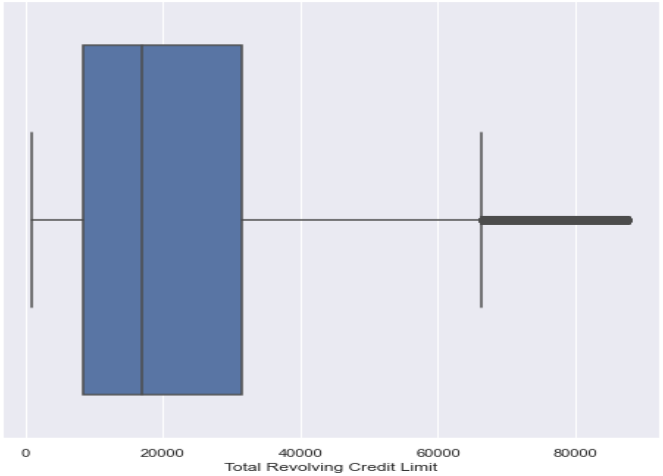



# شناسایی داده‌های پرت

- استفاده از دو روش IQR, Zscore برای شناسایی داده‌های پرت
- نمایش توزیع داده‌ها قبل وبعد از حذف داده‌های پرت
- (استفاده از دو روش و بررسی روی سه ویژگی)
- بررسی z score روی ویژگی Total Revolving Credit Limit



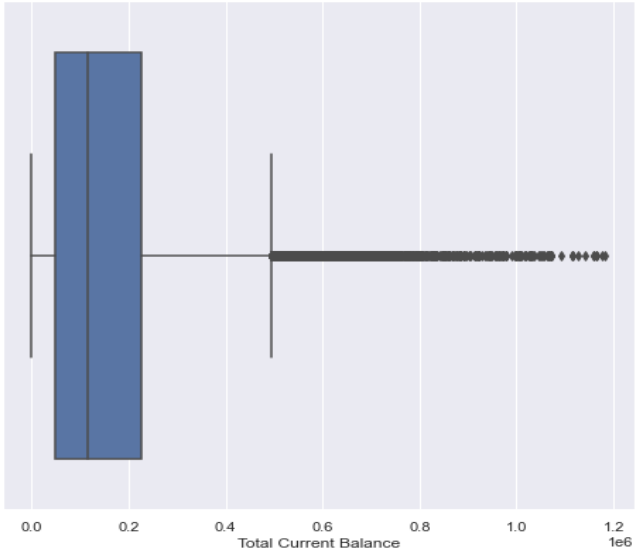

z_score قبل از اعمال



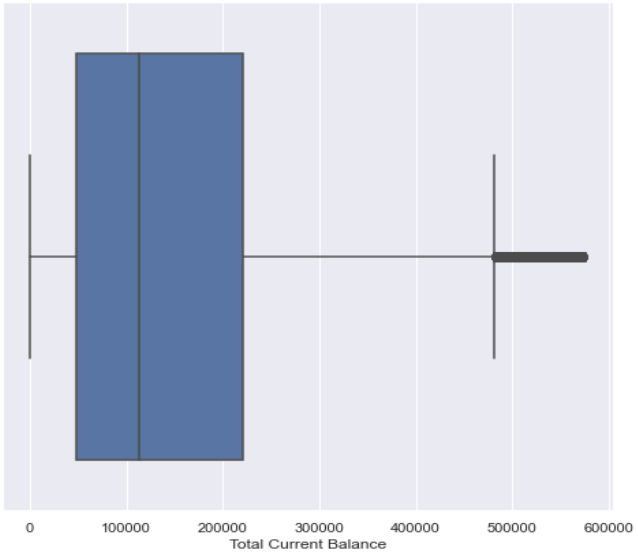

z_score بعد از اعمال

# شناسایی داده‌های پرت

- بررسی روی ویژگی total current balance



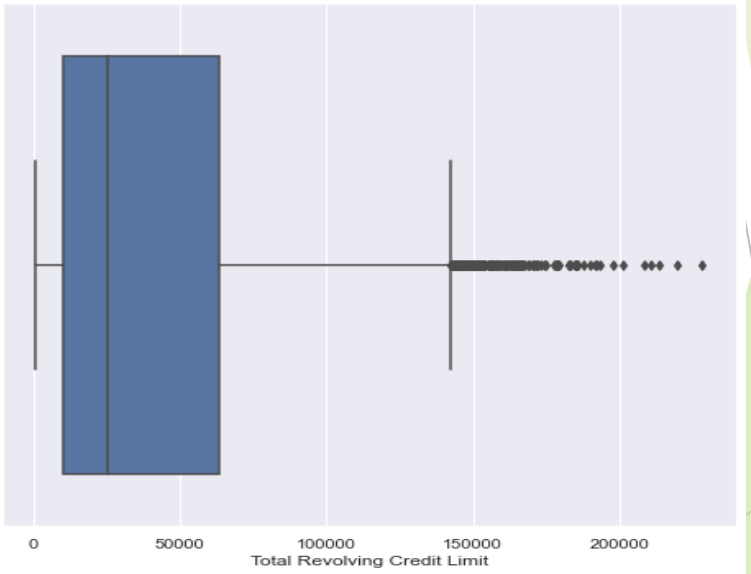

Z_SCORE قبل از اعمال



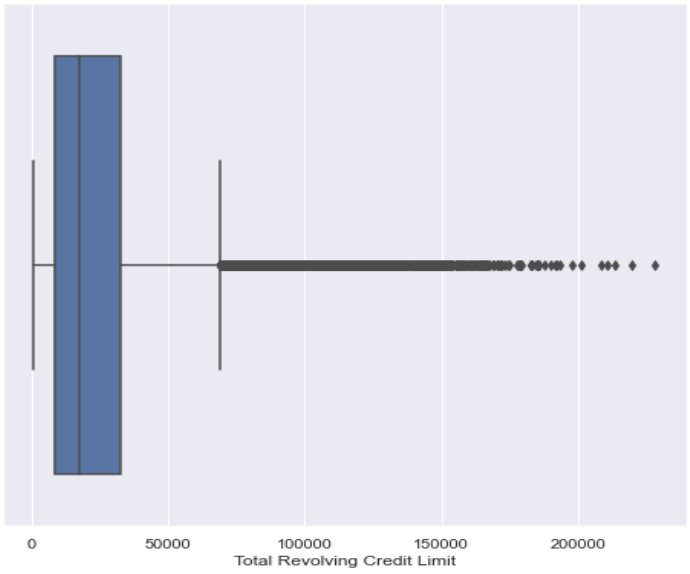

z score بعد از اعمال

# شناسایی داده‌های پرت

## Total Revolving Credit Limit روی ویژگی IQR



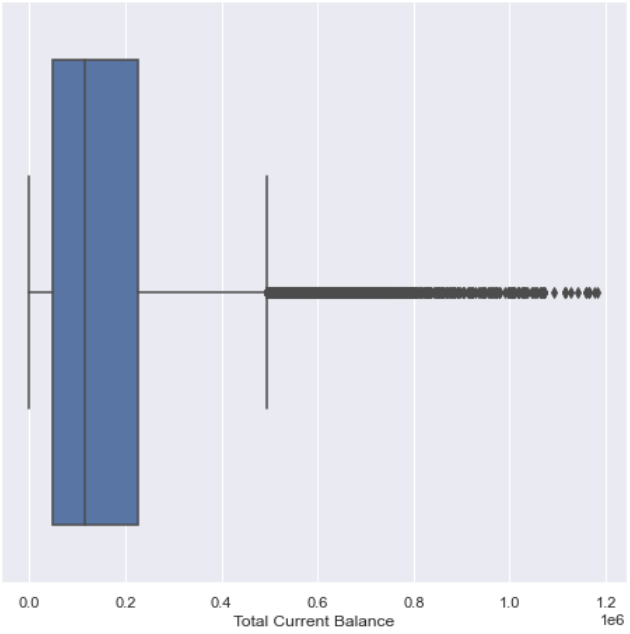

IQR بعد از اعمال



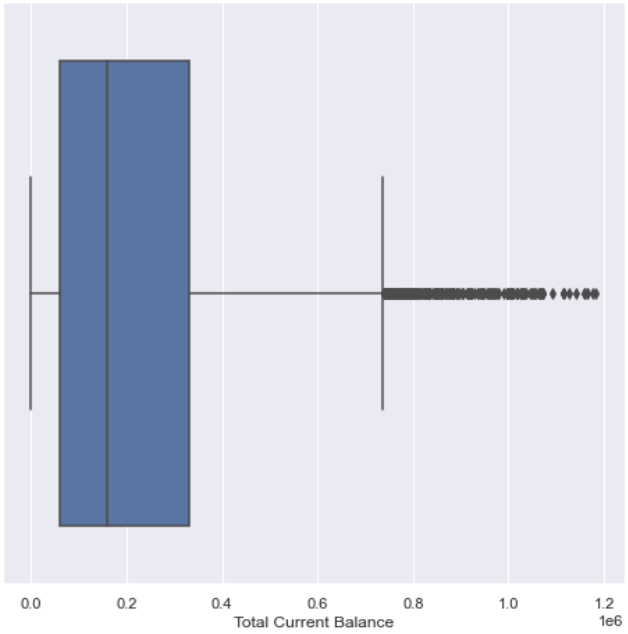

IQR قبل از اعمال

# شناسایی داده‌های پرت

## Total Current Balanceروی ویژگی IQR



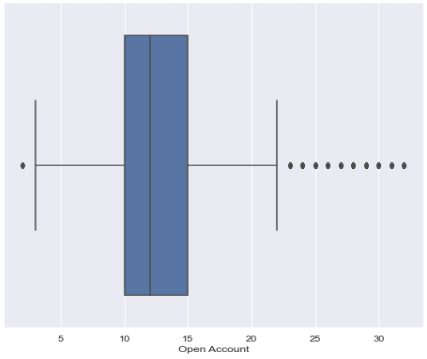

IQR قبل از اعمال

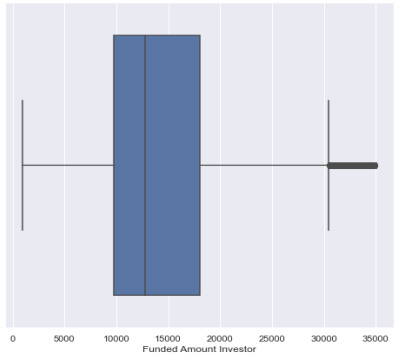


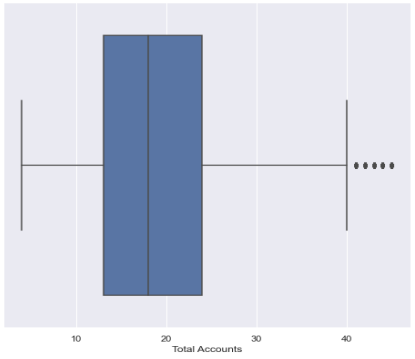

IQRبعد از اعمال

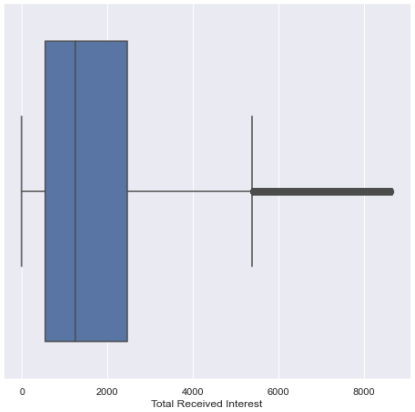
# شناسایی داده‌های پرت

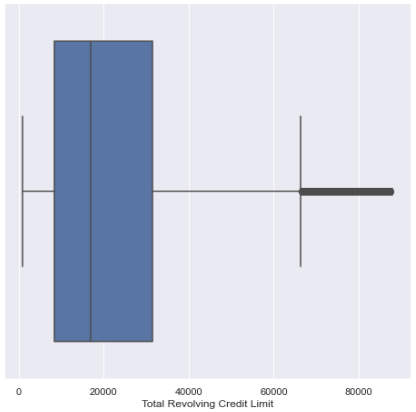
- حذف ۷۵ درصد داده ها توسط IQR
- انتخاب روش z score به عنوان روش نهایی

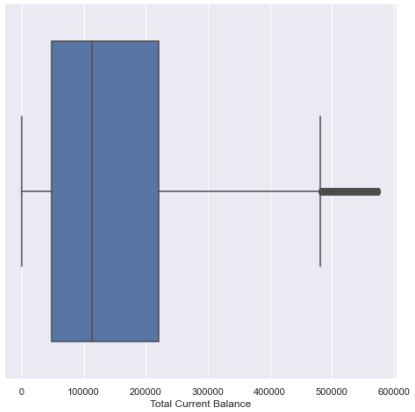


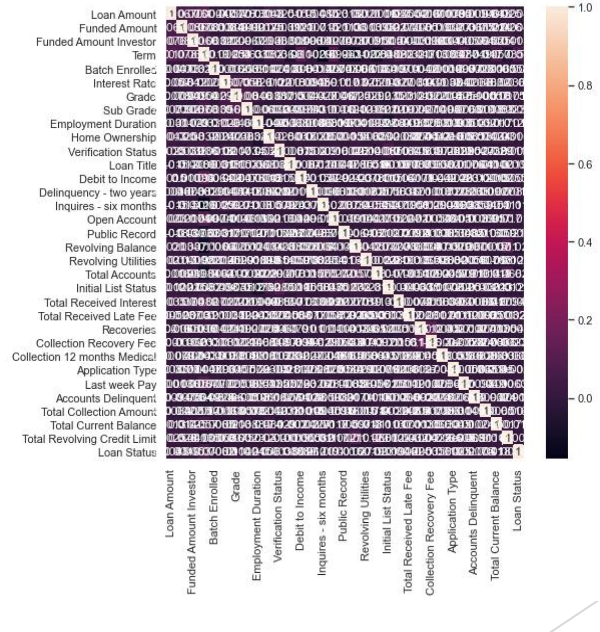



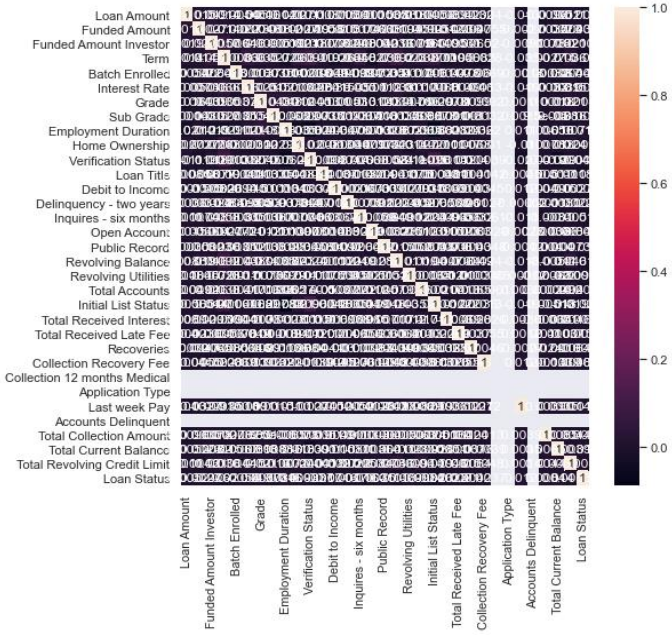

بعد از حذف داده های پرت

# شناسایی داده‌های پرت

- از بین ۳۲ ویژگی پیش‌بینی کننده موجود ۲۱ ویژگی دارای داده پرت است
- حذف ویژگی‌ها تا جایی که بهترین نتیجه از حیث کورولیشن بدست آید
- بهترین نتیجه از حذف داده‌های پرت ۶ ویژگی "Total Received Interest","Total Accounts","Open Account", “Funded Amount Investor","Total Revolving Credit Limit","Total Current Balance“ بدست آمد. که بین ۰و۳_۳و۱۲_۳۱و۱۷_۲۳و۲۴کورولیشن ایجاد می‌شود.




بعد از حذف ۶ ویژگی دارای داده پرت با z score




بعد از حذف تمامی فیچرهای دارای داده پرت با z score

# تبدیل  و استاندارد سازی داده‌ها

- شاهد تغییر مقیاس زیادی در ویژگی‌ها هستیم (با مشاهده مقادیر مینیمم و ماکزیمم و میانگین هر ویژگی)
- شاهد میانگین ۱۶۶۹۷,۷۵۸۳۵۹و ۱ هستیم در ویژگی‌های مختلف
- پس به استاندارد سازی داده‌ها نیاز است

| | Loan Amount | Funded Amount | Funded Amount Investor | Term | Batch Enrolled | Interest Rate | Grade | Sub Grade | Employment Duration | Home Ownership | ... | Recover |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| count | 79763.000000 | 79763.000000 | 79763.000000 | 79763.000000 | 7.976300e+04 | 79763.000000 | 79763.000000 | 79763.000000 | 79763.000000 | 79763.000000 | ... | 79763.000 |
| mean | 16697.758359 | 15673.711533 | 14620.063816 | 57.186628 | 3.179999e+06 | 11.935295 | 1.825483 | 1.986924 | 0.828128 | 80026.353863 | ... | 57.899 |
| std | 8357.006819 | 8126.050543 | 6877.849424 | 5.717684 | 1.523204e+06 | 3.768818 | 1.377080 | 1.488450 | 0.932119 | 44481.214913 | ... | 355.075 |
| min | 1000.000000 | 1000.000000 | 1000.000000 | 36.000000 | 2.249230e+05 | 5.320000 | 0.000000 | 0.000000 | 0.000000 | 14573.537170 | ... | 0.000 |
| 25% | 9930.000000 | 9227.000000 | 9786.753924 | 58.000000 | 1.930365e+06 | 9.329726 | 1.000000 | 1.000000 | 0.000000 | 51366.149780 | ... | 1.296 |
| 50% | 15913.000000 | 13033.000000 | 12794.921590 | 59.000000 | 2.803411e+06 | 11.459140 | 2.000000 | 2.000000 | 0.000000 | 69054.367060 | ... | 3.099 |
| 75% | 21987.500000 | 21671.500000 | 18054.801020 | 59.000000 | 4.351734e+06 | 14.322009 | 3.000000 | 3.000000 | 2.000000 | 94394.614960 | ... | 5.271 |
| max | 35000.000000 | 35000.000000 | 35000.000000 | 60.000000 | 5.924421e+06 | 27.310000 | 6.000000 | 6.000000 | 2.000000 | 406944.859000 | ... | 4354.467 |

# تبدیل و استاندارد سازی داده‌ها


قبل از استاندارد سازی

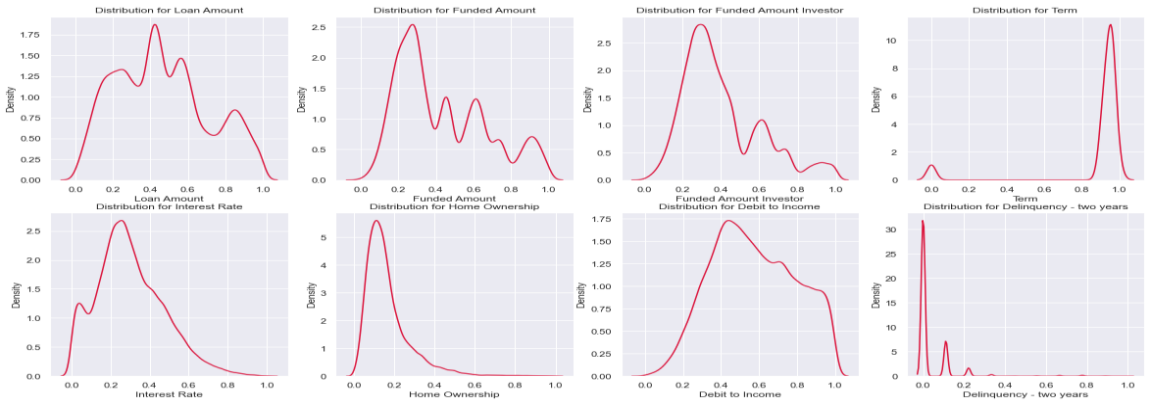
- تبدیل و استاندارد سازی داده‌ها
- استفاده از دو روش:
- z_score , min max
- loan amount: برای ویژگی


min max بعد از روش


z_score بعد از روش

# تبدیل  و استاندارد سازی داده‌ها

- انتخاب روش minmax در نهایت
- هماهنگی بیشتر مقیاس ها بعد از استانداردسازی

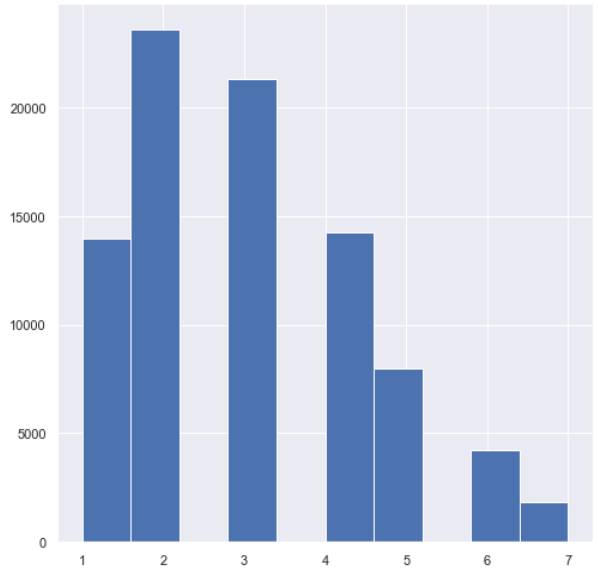



| | Loan Amount | Funded Amount | Funded Amount Investor | Term | Batch Enrolled | Interest Rate | Grade | Sub Grade | Employment Duration | Home Ownership | ... | Recoveri |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| count | 79763.000000 | 79763.000000 | 79763.000000 | 79763.000000 | 79763.000000 | 79763.000000 | 79763.000000 | 79763.000000 | 79763.000000 | 79763.000000 | ... | 79763.0000 |
| mean | 0.461699 | 0.431580 | 0.400590 | 0.882776 | 0.518480 | 0.300832 | 0.304247 | 0.331154 | 0.414064 | 0.166813 | ... | 0.0132 |
| std | 0.245794 | 0.239001 | 0.202290 | 0.238237 | 0.267252 | 0.171388 | 0.229513 | 0.248075 | 0.466059 | 0.113365 | ... | 0.0815 |
| min | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | ... | 0.0000 |
| 25% | 0.262647 | 0.241971 | 0.258434 | 0.916667 | 0.299227 | 0.182343 | 0.166667 | 0.166667 | 0.000000 | 0.093770 | ... | 0.0002 |
| 50% | 0.438618 | 0.353912 | 0.346909 | 0.958333 | 0.452406 | 0.279179 | 0.333333 | 0.333333 | 0.000000 | 0.138850 | ... | 0.0007 |
| 75% | 0.617279 | 0.607985 | 0.501612 | 0.958333 | 0.724066 | 0.409368 | 0.500000 | 0.500000 | 1.000000 | 0.203432 | ... | 0.0012 |
| max | 1.000000 | 1.000000 | 1.000000 | 1.000000 | 1.000000 | 1.000000 | 1.000000 | 1.000000 | 1.000000 | 1.000000 | ... | 1.0000 |

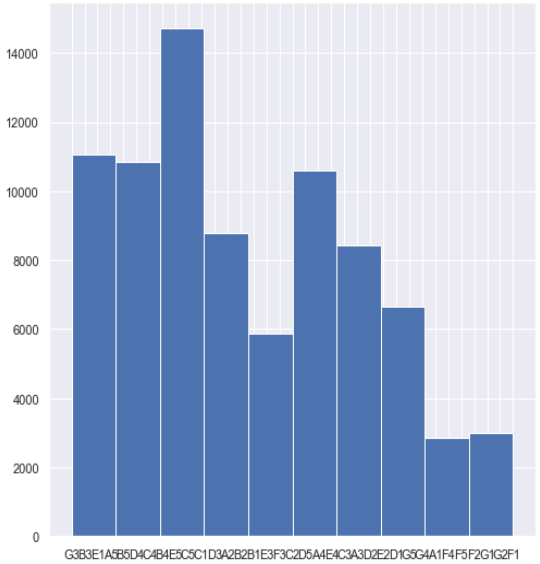
بعد از استانداردسازی

# دسته بندی مجدد متغیرهای دسته‌ای

- دسته بندی مجدد متغیر دسته ای sub grade
- بعد از بسته‌بندی مجدد متوجه شدیم کورولیشن بین ویژگی sub grade و grade به اندازه حدودا ۰/۰۱افزایش داشته که این یعنی بسته‌بندی مجدد خوب بوده




بعد از دسته بندی مجدد

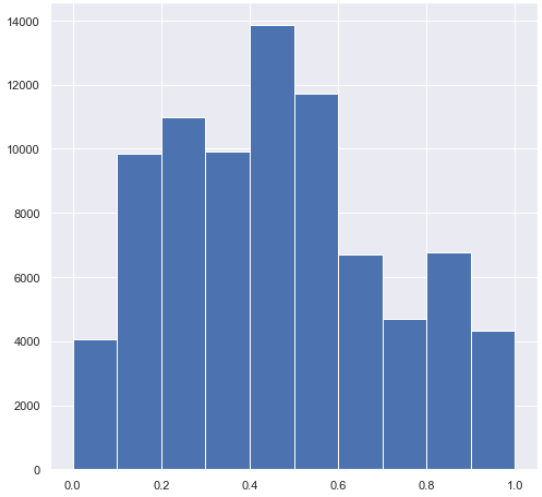



قبل از دسته بندی مجدد

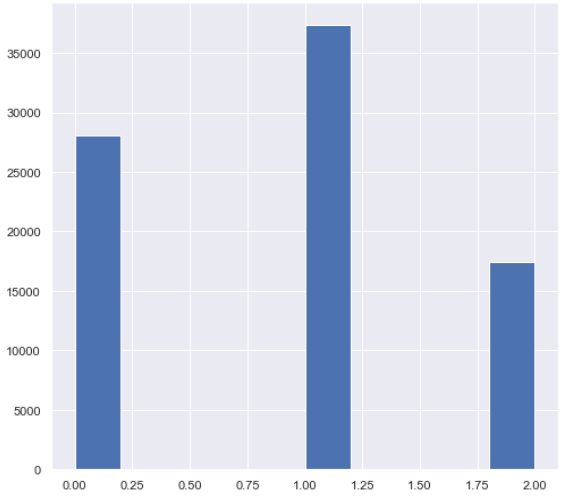
# سبد بندی متغیرهای عددی

استفاده از دو روش برای سبد بندی متغیرهای عددی:

- دسته‌های سه یا چهار یا ۵ تایی با روش KBinsDiscretizer
- دسته‌های ۳ یا ۴ یا ۵ تایی با روش Qcut
- اعمال روش KBinsDiscretizer روی ویژگی loan amount

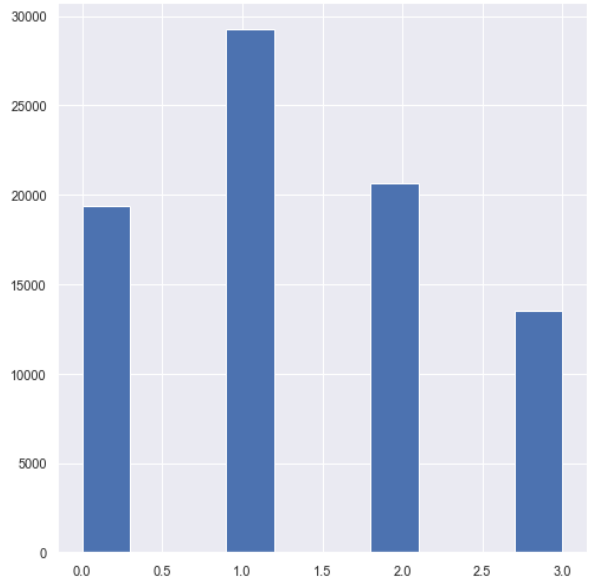



قبل از سبد بندی سه تایی

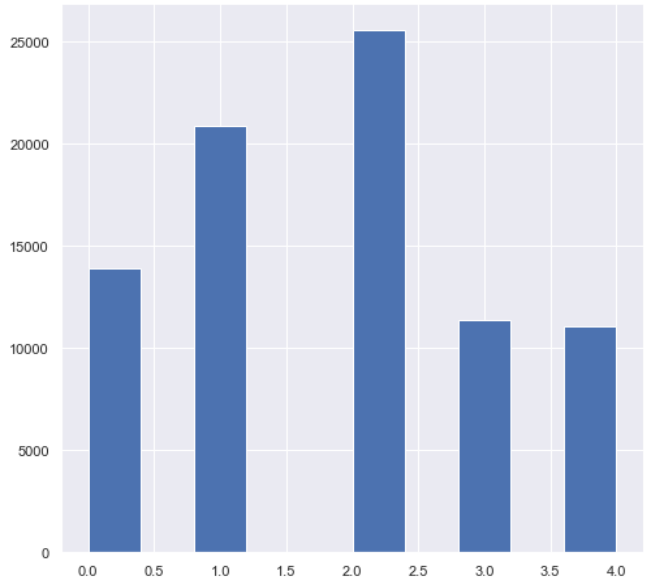



بعد از سبد بندی سه تایی

# سبد بندی متغیرهای عددی



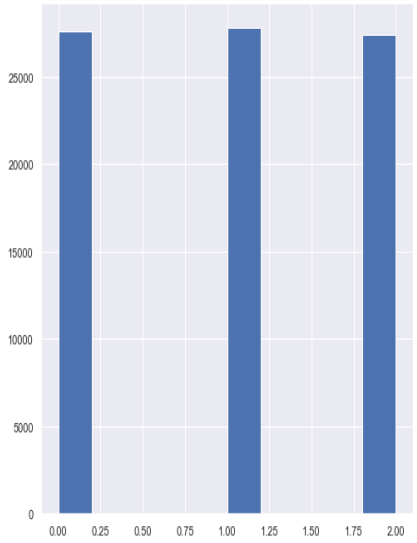

پس از سبد بندی ۴ تایی



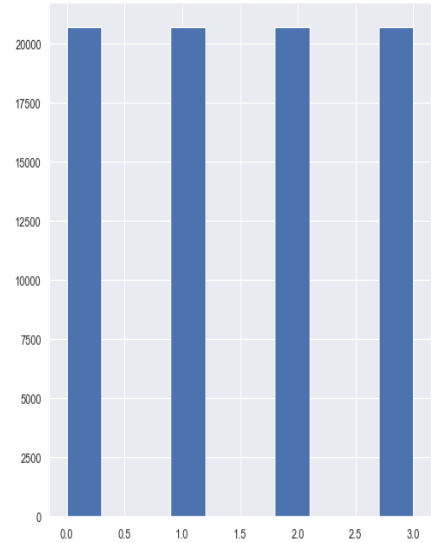

پس از سبد بندی ۵ تایی

# سبد بندی متغیرهای عددی

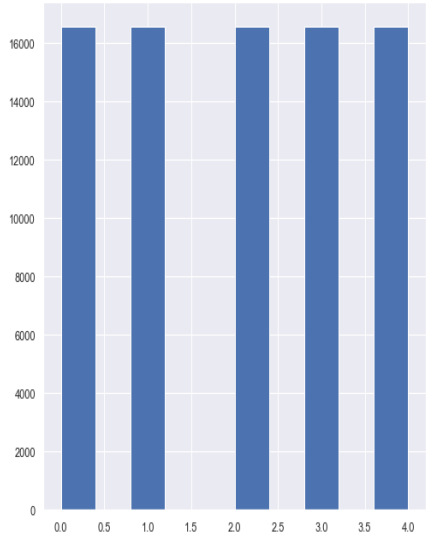
- اعمال روش qcut روی ویژگی loan amount




بعد از سبد بندی سه تایی



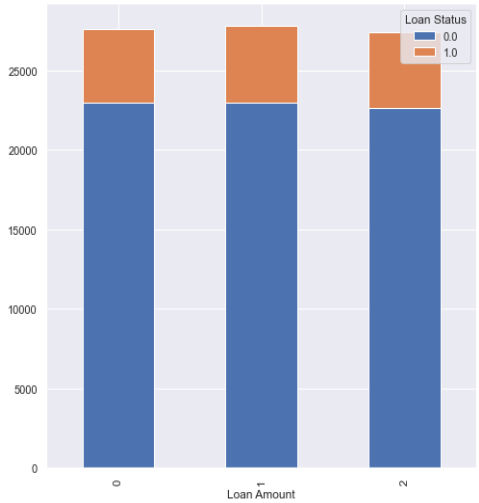

بعد از سبد بندی ۴ تایی

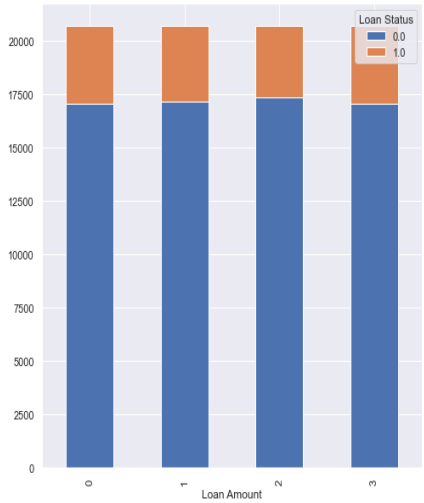


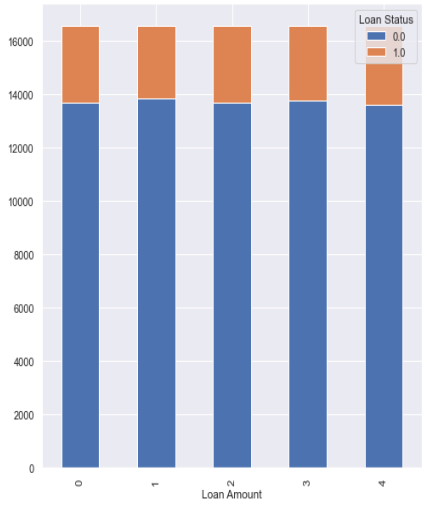

بعد از سبد بندی ۵ تایی

# سبد بندی بر مبنای مقدار پیشبینی برای بهبود مدل

- رسم نمودار overlay در حالت اسکیل شده برای هر ویژگی (term, loan amoun,...) و هدف در حالت سبدبندی شده
- محاسبه mutual information بین ویژگی‌های بالا و هدف
- سبد بندی شده توسط qcut

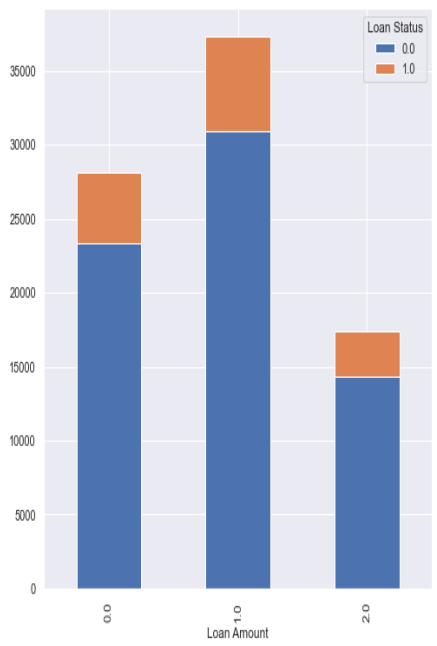


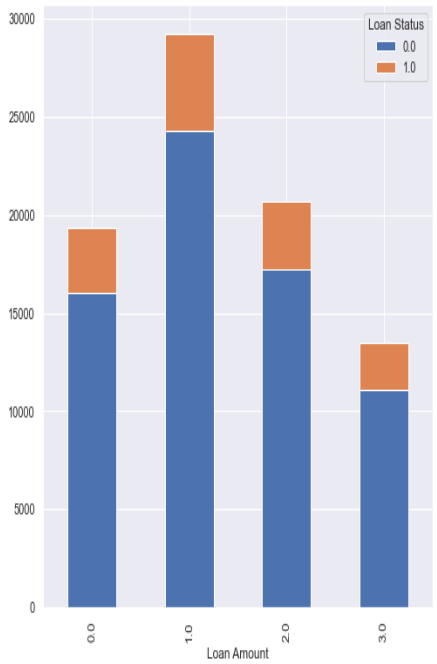



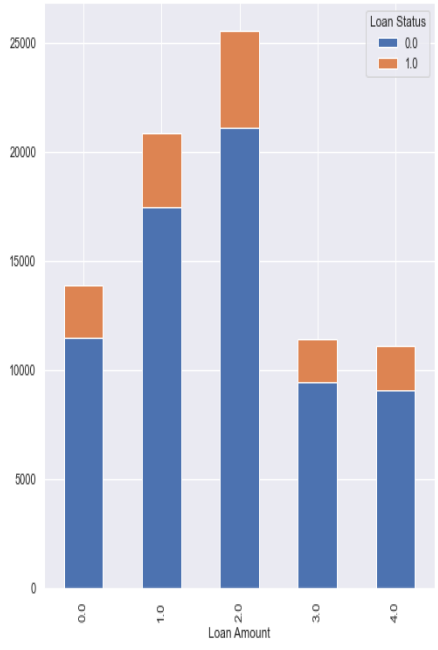



# سبد بندی بر مبنای مقدار پیشبینی برای بهبود مدل

- سبد بندی شده توسط kbins



# سبد بندی بر مبنای مقدار پیش بینی برای بهبود مدل

## Knbins:

- ستون‌های با کورولیشن بزرگتر از ۰/۱ بعد از سبد بندی سه تایی:

Loan Amount , Term

Collection Recovery Fee و Recoveries

- ستون‌های با کورولیشن بزرگتر از ۰/۱ بعد از سبد بندی ۴ تایی:

Collection Recovery Fee و Recoveries

- ستون‌های با کورولیشن بزرگتر از ۰/۱ بعد از سبد بندی ۵ تایی:

Collection Recovery Fee و Recoveries

Term و Debit to Income

- ستون‌های با کورولیشن بزرگتر از ۰/۱ قبل از سبد بندی:

Loan Amount, Term

Term و Debit to Income

Collection Recovery Fee و Recoveries

Revolving Balance, Total Revolving Credit Limit

# سبد بندی بر مبنای مقدار پیش بینی برای بهبود مدل

**Qcut:**

- ستون‌های با کورولیشن بزرگتر از ۰/۱ بعد از سبد بندی سه تایی:

Total Received Late Fee, Recoveries

- ستون‌های با کورولیشن بزرگتر از ۰/۱ بعد از سبد بندی ۴ تایی:

Recoveries و Total Received Late Fee

Collection Recovery Fee و Total Received Late Fee

Total Collection Amount و Total Received Late Fee

- ستون‌های با کورولیشن بزرگتر از ۰/۱ بعد از سبد بندی ۵ تایی:

Recoveries و Total Received Late Fee

Collection Recovery Fee و Total Received Late Fee

Total Collection Amount و Total Received Late Fee

Collection Recovery Fee و Recoveries

# سبد بندی بر مبنای مقدار پیش بینی برای بهبود مدل

مجموع mutual information ویژگی ها با ویژگی هدف:

KBinsDiscretizer method

قبل از سبد بندی: ۵/۷۸۹۹۱

بعد از سبد بندی سه تایی: ۰/۰۱۲

بعد از سبد بندی ۴ تایی: ۰/۰۱۳

بعد از سبد بندی ۵ تایی: ۰/۰۱۴

Qcut method

قبل از سبد بندی: ۵/۷۸۹۹۱

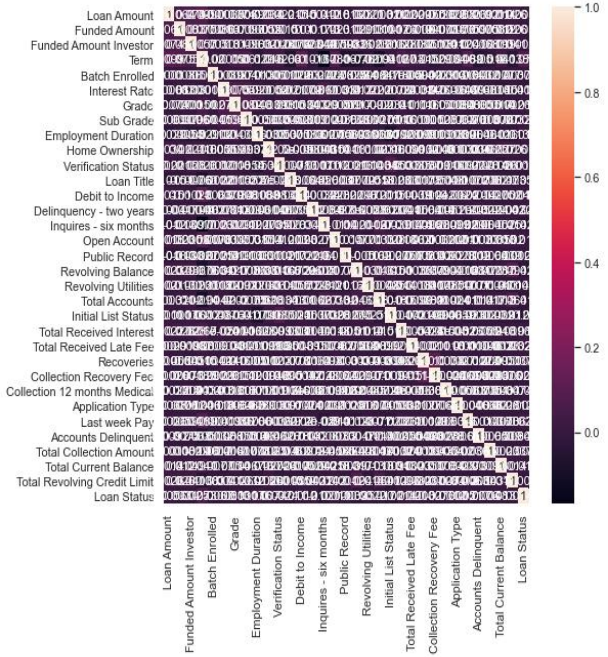
بعد از سبد بندی ۳ تایی: ۰/۰۱۵

بعد از سبد بندی ۴ تایی: ۰/۰۱۷

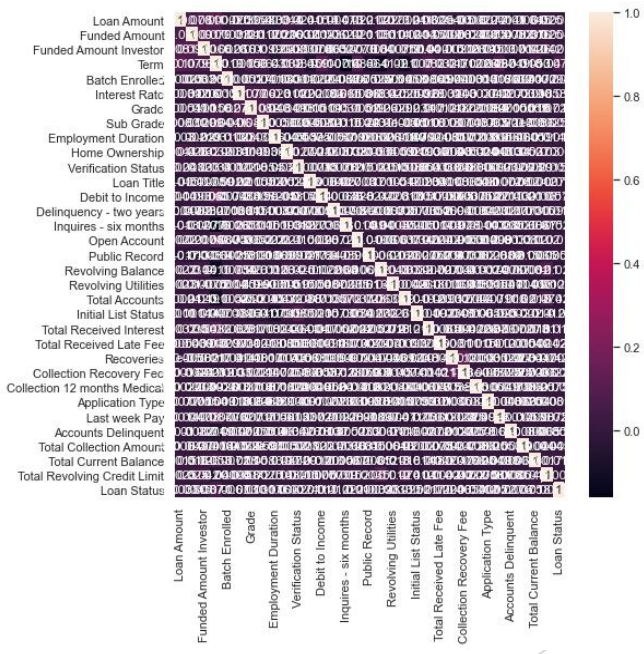
بعد از سبد بندی ۵ تایی: ۰/۰۲

# سبد بندی بر مبنای مقدار پیش بینی برای بهبود مدل

- کاهش کورولیشن بین داده‌ها بعد از سبد بندی با هر دو روش
- کاهش mutual information بین داده ها بعد از سبد بندی با هر دو روش
- در نتیجه عدم مفید بودن سبد بندی برای دیتاست



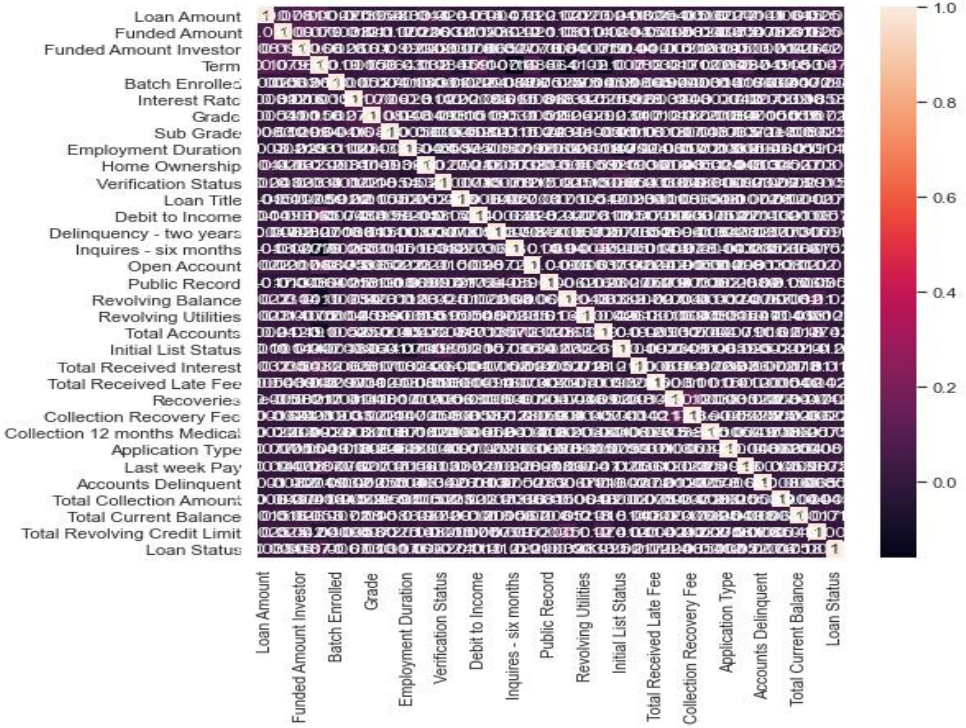

بعد از سبد بندی




قبل از سبد بندی

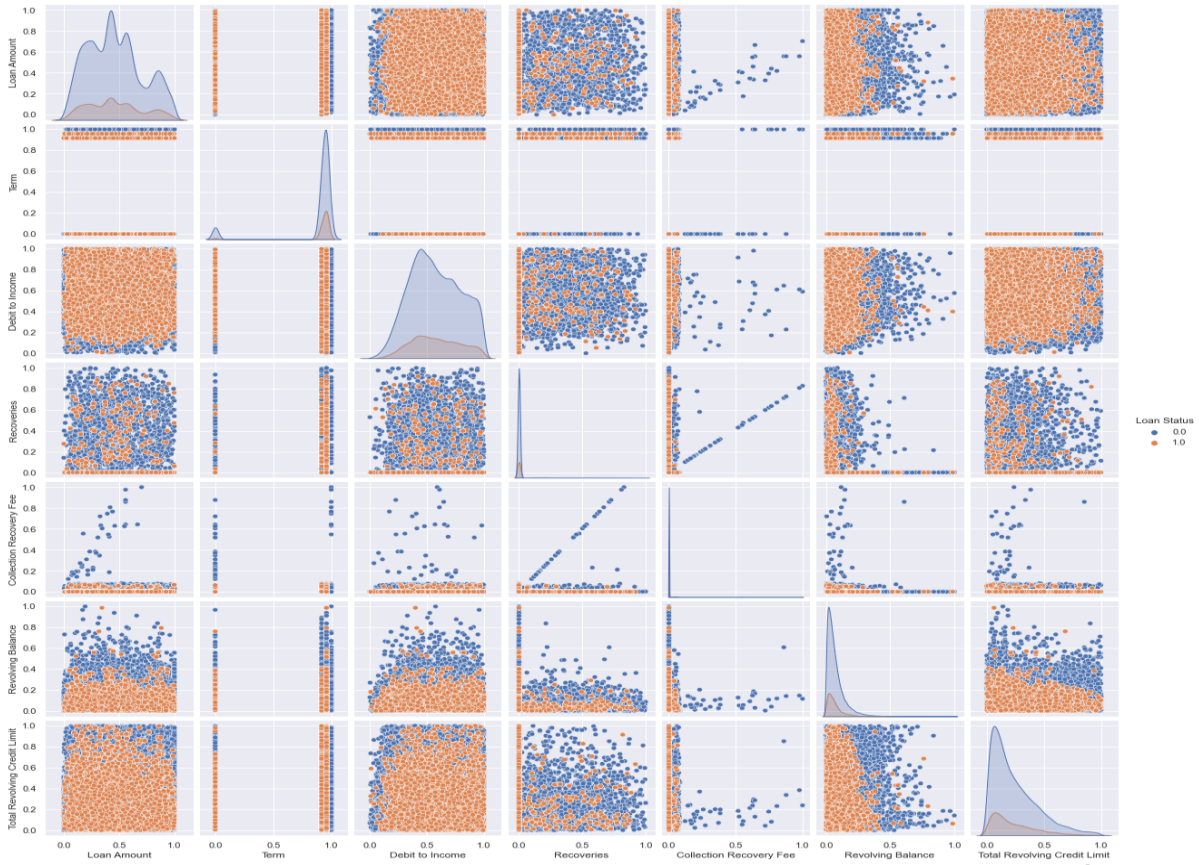
# بررسی روابط تک متغیره بین متغیرهای پیشبین و متغیر هدف

- نمایش هیت مپ مربوط به ویژگی‌های شناسایی شده و بررسی روابط
- شناسایی ویژگی‌هایی با بیشترین همبستگی با ویژگی هدف



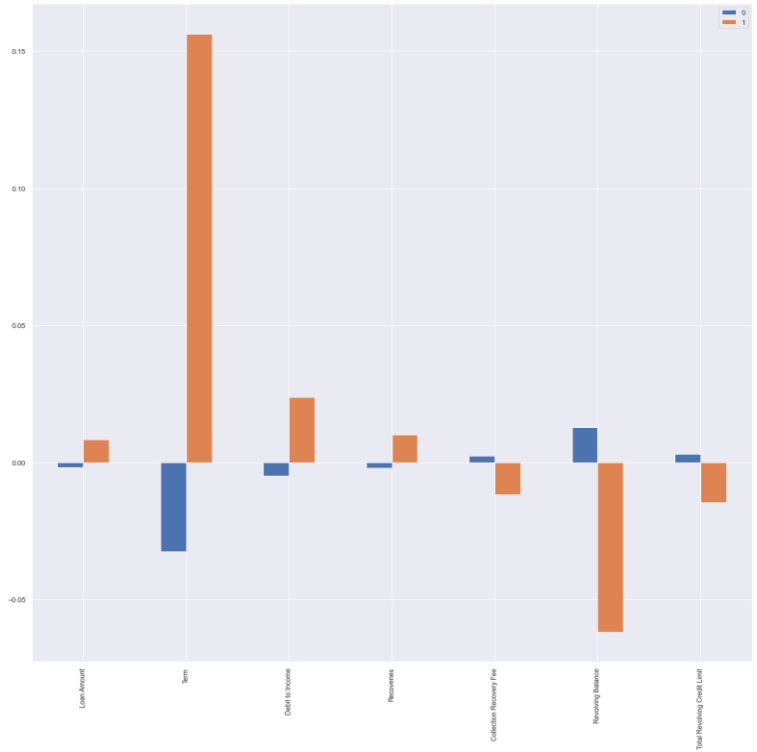
# بررسی روابط چند متغیره بین متغیرها

- بررسی روابط بین کلیه متغیرهایی که بیشترین میزان کورولیشن را با هم دارند
- شیب کم به دلیل ارتباط کم (کمتر از ۰/۲)



# استخراج متغیرهای جدید براساس ترکیب متغیرهای موجود

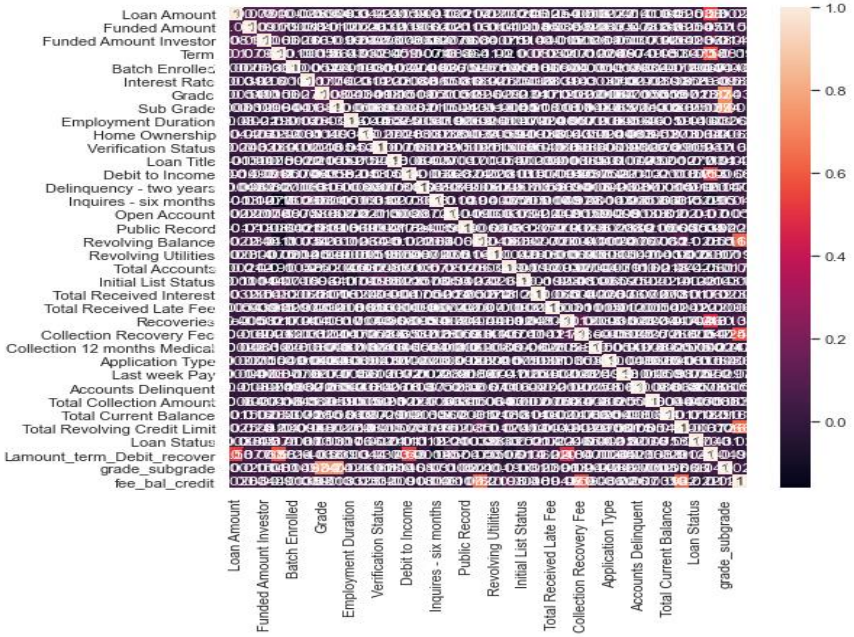
- عدم حذف فیچرها به دلیل همبستگی پایین تمامی فیچرها
- انتخاب فیچرهایی برای ترکیب با یکدیگر




| | 0 | 1 |
|---|---|---|
| Loan Amount | -0.002166 | 0.010495 |
| Term | -0.031998 | 0.155035 |
| Debit to Income | -0.004368 | 0.021163 |
| Recoveries | -0.002399 | 0.011625 |
| Collection Recovery Fee | 0.002525 | -0.012233 |
| Revolving Balance | 0.012395 | -0.060054 |
| Total Revolving Credit Limit | 0.002641 | -0.012795 |

# استخراج متغیرهای جدید براساس ترکیب متغیرهای موجود

- ترکیب خطی چندین فیچر که کمی شباهت داشتند (محاسبه مجموع و میانگین)
- مناسب نبودن این فیچرها به دلیل تغییر ندادن همبستگی ویژگی ها با متغیر هدف




پس از اضافه کردن سه ستون جدید

# چارچوب تقسیم داده‌ها
# روش انجام Cross-validation
# بالانس داده‌ها

- ۸۰٪ کل داده برای آموزش، ۲۰٪ cross validation
- توزیع یکسان داده های هدف در مجموعه validation و train(۸۳٪ به ۱۷٪)

جواب پایه با Dummy Classifier و استراتژی most frequent انجام شده که جواب تقریبا ۵ ۸۲. درصد داده است.

```
0.8261289543588505
              precision    recall  f1-score   support

           0       0.83      1.00      0.90     13684
           1       0.00      0.00      0.00      2880

    accuracy                           0.83     16564
   macro avg       0.41      0.50      0.45     16564
weighted avg       0.68      0.83      0.75     16564
```

# انتخاب و پیاده سازی الگوریتم‌های لازم

از ۴ الگوریتم زیر برای مدل کردن استفاده شد:

:knn

```
accuracy on train 0.9008074862274545
accuracy on test 0.8295097802463173
              precision    recall  f1-score   support

           0       0.90      0.89      0.90     13681
           1       0.51      0.53      0.52      2883

    accuracy                           0.83     16564
   macro avg       0.71      0.71      0.71     16564
weighted avg       0.83      0.83      0.83     16564
```

:mlp

```
accuracy on train 0.8409629461927401
accuracy on test 0.824136681960879
              precision    recall  f1-score   support

           0       0.83      0.98      0.90     13681
           1       0.47      0.07      0.13      2883

    accuracy                           0.82     16564
   macro avg       0.65      0.53      0.51     16564
weighted avg       0.77      0.82      0.77     16564
```

:naivebayes

```
accuracy on train 0.8285563353709154
accuracy on test 0.8264911857039362
              precision    recall  f1-score   support

           0       0.83      1.00      0.91     13690
           1       0.00      0.00      0.00      2874

    accuracy                           0.83     16564
   macro avg       0.41      0.50      0.45     16564
weighted avg       0.68      0.83      0.75     16564
```

:xgboost

```
accuracy on train 0.8873594445702211
accuracy on test 0.861446510504709
              precision    recall  f1-score   support

           0       0.86      1.00      0.92     13690
           1       0.93      0.22      0.35      2874

    accuracy                           0.86     16564
   macro avg       0.90      0.61      0.64     16564
weighted avg       0.87      0.86      0.82     16564
```

# انتخاب و پیاده سازی الگوریتم‌های لازم

اگر از یکی از روش‌های دیگر برای حذف داده‌های پرت (oneclasssvm) استفاده شود:

knn:

```
accuracy on train 0.8997999768089053
accuracy on test 0.8323863636363636
              precision    recall  f1-score   support

           0       0.90      0.89      0.90     14301
           1       0.51      0.54      0.52      2947

    accuracy                           0.83     17248
   macro avg       0.71      0.71      0.71     17248
weighted avg       0.84      0.83      0.83     17248
```

Mlp

```
accuracy on train 0.8433151669758813
accuracy on test 0.8276901669758813
              precision    recall  f1-score   support

           0       0.85      0.97      0.90     14301
           1       0.49      0.14      0.21      2947

    accuracy                           0.83     17248
   macro avg       0.67      0.55      0.56     17248
weighted avg       0.78      0.83      0.79     17248
```

Naivebayes

```
accuracy on train 0.8277191558441559
accuracy on test 0.8291396103896104
              precision    recall  f1-score   support

           0       0.83      1.00      0.91     14301
           1       0.00      0.00      0.00      2947

    accuracy                           0.83     17248
   macro avg       0.41      0.50      0.45     17248
weighted avg       0.69      0.83      0.75     17248
```

# انتخاب و پیاده سازی الگوریتم‌های لازم

:xgboost

```
accuracy on train 0.8837111549165121
accuracy on test 0.8603316326530612
              precision    recall  f1-score   support

           0       0.86      1.00      0.92     14301
           1       0.91      0.20      0.33      2947

    accuracy                           0.86     17248
   macro avg       0.88      0.60      0.63     17248
weighted avg       0.87      0.86      0.82     17248
```

- حاصل نشدن دقت بهتر نسبت به روش قبلی حذف داده های پرت
- Xgboost در تمامی معیارهای ارزیابی از بیس لاین بهتر بوده

# تنظیم بهینه مدل و هایپر پارامترها

:knn

```
Best parameters set:
        leaf_size: 9
        n_neighbors: 19
        weights: distance
```

:mlp

```
activation: tanh
alpha: 0.0001
hidden_layer_sizes: (50, 50, 50)
learning_rate: adaptive
solver: adam
```

:naivebayes

```
Best parameters set:
        alpha: 0.5
        class_prior: None
        fit_prior: True
```

:xgboost

```
Best parameters set:
        learning_rate: 0.1
        max_depth: 9
        n_estimators: 100
```

# تنظیم بهینه مدل و هایپر پارامترها

نتایج مدل بعد از تنظیم هایپر پارامترها

:knn

```
accuracy on train 1.0
accuracy on test 0.9645013281815986
              precision    recall  f1-score   support

           0       0.98      0.98      0.98     13690
           1       0.89      0.91      0.90      2874

    accuracy                           0.96     16564
   macro avg       0.94      0.94      0.94     16564
weighted avg       0.96      0.96      0.96     16564
```

:mlp

```
accuracy on train 0.9145573918949513
accuracy on test 0.8527529582226515
              precision    recall  f1-score   support

           0       0.90      0.93      0.91     13690
           1       0.59      0.49      0.54      2874

    accuracy                           0.85     16564
   macro avg       0.74      0.71      0.72     16564
weighted avg       0.84      0.85      0.85     16564
```

:naivebayes

```
accuracy on train 0.8286468945739944
accuracy on test 0.8261289543588505
              precision    recall  f1-score   support

           0       0.83      1.00      0.90     13684
           1       0.00      0.00      0.00      2880

    accuracy                           0.83     16564
   macro avg       0.41      0.50      0.45     16564
weighted avg       0.68      0.83      0.75     16564
```

# معرفی مجموعه معیارهای ارزیابی و محاسبه آن‌ها و تفسیر نتایج

- انتخاب مدل باتوجه به معیارهای:
- Recall
- Precision
- Accuracy
- F-score

۱. Precision = TP / (TP+FP)

۲. Recall = TP / (TP+FN)

۳. F1 Score = 2 * (Precision * Recall) / (Precision + Recall)

# مشخص کردن بهترین مدل همراه با پارامترهای تعیین شده

:Knn

بهبود دقت نسبت به بیس لاین

بهبود precision ,f1_score

بهبود recallروی کلاس یک

:mlp

بهبود دقت نسبت به بیس لاین

بهبود precision ,f1_score

بهبود recallروی کلاس یک

:Xgboost

بهبود دقت نسبت به بیس لاین

بهبود precision , recall , f1_score

**بهترین مدل با توجه به معیارها:**

xgboost

# اعمال مدل انتخابی روی داده‌ی نهایی تست

- توزیع ۹۳٪ به ۷٪ کلاس‌ها در تست
- خروجی ۹۳,۴۱٪ بیس لاین
- اعمال مدل xgboost روی داده‌های تست نهایی
- دریافت دقت حدودا ۹۳,۴۹٪(تقریبا ۸ صدم بهتر از بیس لاین)

```
accuracy on test 0.9349968873210209
              precision    recall  f1-score   support

           0       0.93      1.00      0.97     18023
           1       0.00      0.00      0.00      1253

    accuracy                           0.93     19276
   macro avg       0.47      0.50      0.48     19276
weighted avg       0.87      0.93      0.90     19276
```

# گزارش آزمایشات برگشت به فاز های قبلی برای بهبود ارزیابی

روش اول: با توجه به اینکه خیلی دقت بالاتری نسبت به بیس لاین بدست نمی آوریم به فازهای اولیه برگشته و روش های مختلف تشخیص و حذف داده های پرت(سه روش) را اعمال کرده و در هر مرحله سبد بندی مجدد متغیرهای دسته ای و همچنین میتوال اینفورمیشن میان متغیرهای عدد و تارگت با هر بار سبد بندی (سه ، ۴، ۵ تایی) بررسی میکنیم و در صورت بهبود میوتوال اینفورمیشن این سبد بندی ها را روی دیتاست خود اعمال کرده در غیر اینصورت تغییری در دیتاست اعمال نمیکنیم. همچنین در هر مرحله امکان ایجاد فیچرهای جدید را بررسی کرده و در صورت ممکن اضافه میکنیم. در نهایت با هر کدام این روش ها نتایج بهبودی نداشته و خیلی بالاتر از بیس لاین نخواهد شد.

روش دوم: به جای آپ سمپل کردن ابتدایی داده های ترین در جهت بالانس توزیع داده ها در هر کلاس از متد نیر میس استفاده کرده و مجدد مدل را ترین میکنیم اما باز هم بهبودی در دقت نهایی حاصل نمیشود.

روش سوم: به جای اینکه ابتدا داده ها را آپ سمپل کنیم(با توجه به اینکه جنریت کردن داده تا حدودی میتواند داده را دستکاری کند) از الگوریتم هایی که نسبت به بالانس نبودن کلاس ها حساسیت کمتری دارند مثل کا ان ان استفاده میکنیم. اما باز هم نتیجه بهتری حاصل نمیشود.

# تحلیل نقاط قوت و ضعف کار انجام شده و پیشنهادات برای بهبود آینده

## نقاط ضعف:

این دیتاست دارای داده‌های فیک و رندوم بوده و هیچکدام از ویژگی‌ها کورولیشنی با مقادیر هدف نداشتند. از طرف دیگر توزیع دادها در کلاس های مختلف متناسب نبوده که این عوامل منجر به این شد که نتوان هوشمند سازی در داده‌ها را به نحو احسنت انجام داد. در داده‌های واقعی قطعا الگو برای ساخت مدل وجود دارد اما وجود الگو در داده‌های رندوم احتمال خیلی کمی دارد.

## نقاط قوت:

تعداد زیاد رکوردهای دیتاست که میتواند منجر به جلوگیری از اورفیت شدن داده‌ها شود و به راحتی میتوان مجوعه کراس ولیدیشن را جدا کرد و محدودیت کمبود داده نداریم

# معرفی کارهای مرتبط

- https://rpubs.com/saramonica/loan-prediction
- https://ieeexplore.ieee.org/document/8389442
- https://www.researchgate.net/publication/325917915_Prediction_of_loan_status_in_commercial_bank_using_machine_learning_classifier