互联网金融算法应用介绍

内容from知乎

目前本人在某P2P公司做风控建模的工作，最近在金融风控的A卡申请评分卡的风控建模工作，目前告一段落，一方面巩固用到的技术另一方面也希望业内人士能给出一些不足的指点，现把建模过程中所用到的技术处理方法总结整理出来。

背景描述

最近几年P2P信贷行业的广泛发展，越来越多的消费者选择通过P2P贷款的方式解决资金暂时短 缺的问题。不幸的是，这虽然有利于资金的重新配置，但贷款申请人因各种原因逾期不还的现象 多次发生，这给P2P公司造成一定程度的坏账损失。机器学习算法可以实现对申请人进行甄别， 在贷款发放前拒绝有极大逾期风险的用户，达到控制达到金融风险控制的目的，降低企业潜在的 坏账风险。

在本项目中，目标就是利用申请人基本信息，信用卡使用交易，征信信息等数据构建甄别用户的机器学习模型。使用的数据是国际著名的p2p信贷平台lendingclub官网开放的借贷人数据，涉及42万条申请人样本，145个字段信息。

Data source:https://www.lendingclub.com/info/download-data.action

问题陈述

本项目是二分类的监督学习模型，旨在利用已有的数据构建能够将所有的贷款申请人划分成'好 用户'和'坏用户'的机器学习模型。

探索数据集得到数据的基本统计信息，将离散特征通过哑变量编码和编码转变成数值 型变量。之后，将全部的数据集划分成训练集和测试集，在训练集上应用机器学习分类模型进行 训练，参数调优，效果评估。将训练稳定的模型应用到测试集上进行预测。

算法和技术 我用到的工具主要是python和R，python完成数据预处理，特征筛选，模型搭建和调优的工作，R包完成数据的探索可视化以及统计计算的工作。该数据集包含连续特征和离散文字型特征，连续特征需要进行异常值处理，离散文字型特征需要进行数值转化，连续特征和离散特征都存在数据缺失，需要缺失处理。主要用到的预处理技术包括:归一化技术，哑变量编码技术，缺失填充技术，异常值检测技术;特征工程部分涉及的技术包括:方差阈值技术，lasso回归L1截断技术，vif共线性判断技术，递归特征删除技术;在模型阶段，本次项目选择逻辑回归分类模型和随机森林分类模型训练结果作为基准线;在模型调优阶段用到的技术是:网格搜索技术，交叉验证技术;模型评估用到的主要技术是:混淆矩阵，roc曲线，学习曲线，ks曲线 作为判断及调整依据。由于本次用到的数据属于样本分类失衡数据，在构建模型时会对此进行调整，将其分类比例尽可能达到平衡状态。

- 逻辑回归
  逻辑回归的预测结果是介于0和1之间的概率，可适用于连续特征和分类特征，容易使用和解释，这三点对于消费金融贷款用户风控建模具有非常好的特点，因为在风控业务实际场景中，能够知道什么因素对贷款人的还款能力进行量化判别，对业务会有非常大的帮助，同时逻辑回归的概率也可以转化为业务场景中对贷款人的评估分数。但是，逻辑回归的本质是基于线性回归模型，其前提是假设用于建模的特征之间不存在相关性，因此对共线性问题比较敏感，故在实际建模中需要考虑自变量之间的特征关系，要消除共线性问题，才能达到比较靠谱的结果;逻辑回归的预测结果是呈's'型，由log转化而来，因此是非线性的，这个's'曲线在概率的两端随着log值的变化，

概率变化很小，边际值太小，而中间概率的变化会很敏感，这导致再两端的区分度不敏感。

- 随机森林
  随机森林算法是建立在决策树基础上的集成算法，在实际应用场景中有很好的效果表现，不容易产生过拟合，能够处理高纬特征，具有很强的抗干扰能力，同时还可以处理有缺失的数据，能够输出特征的重要度得分，运算速度快，且能平衡不平衡数据的误差。但是，随机森林算法依然继承决策树的局部最优特点，特征属性划分比较多会对随机森林的结果产生明显的影响，在实际应用中对特征的属性划分要特别注意。
  在本项目中，我主要采用上述两种算法，逻辑回归中消除特征的共线性问题，主要基于特征相关系数及vif判断;在随机森林算法中，我主要采用对特征进行卡方分箱的策略，每个特征最大箱体数量不超过5，避免特征属性值过多影响随机森林的预测效果。

评价指标

该模型是二分类模型，我使用的指标是roc,auc,ks来评估模型，这几个指标对于比例失衡的数据 是相对比较客观的评估指标。

- roc 曲线

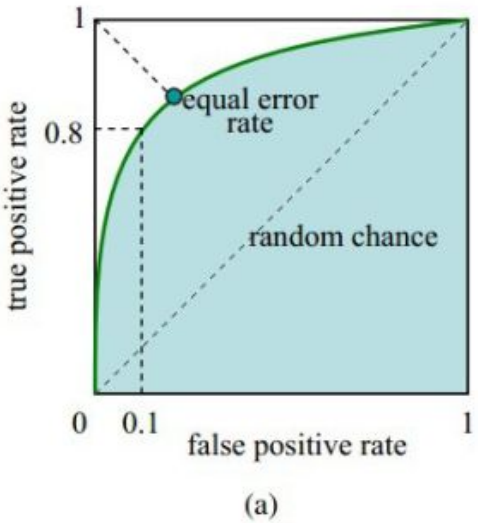
横坐标是假正率，纵坐标是真正率.roc曲线尽可能的靠近左上边(0,1)的位置，效果越好 (0,0):真正率和假正率都是0，所有样本全部预测为负样本 (1,1):真正率和假正率都是1，所有样本全部预测为正样本 (0,1):真正率为1，假正率为0，正样本全部预测正确，负样本全部预测正确最完美的 情况

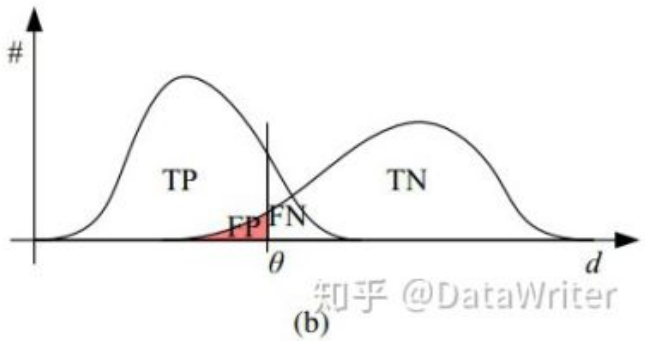
(1,0):真正率为0，假正率为1，正样本全部预测错误，负样本全部预测正确

confusion matrix

station	predict positive	predict negtive
real postive	ture positive	false negtive
real negtive	false positive	ture negtive

roc_curve





- AUC曲线

ROC曲线下的面积，常介于0.5和1之间(极端情况下低于0.5)，可以直观的评价分类器的好

坏，值越大越好。

AUC值是一个概率值，当你随机挑选一个坏样本以及好样本，当前的分类算法根据计算得

到的概率值将这个

坏样本排在好样本前面的概率就是AUC值，AUC值越大，当前分类算法越有可能将坏样本

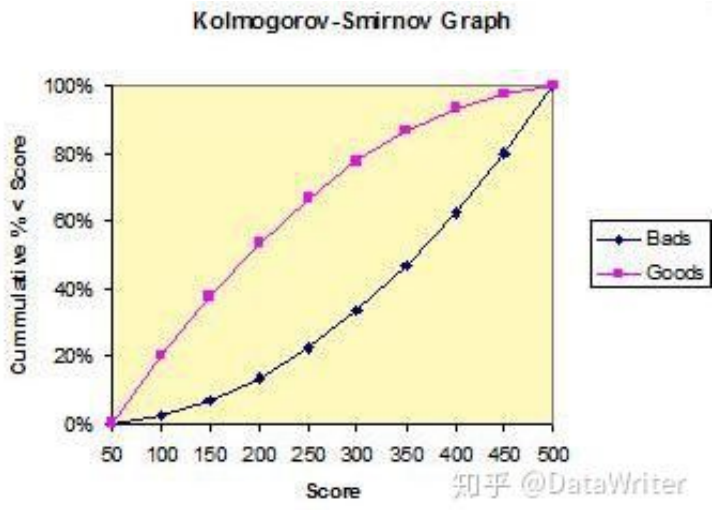
排在好样本前面，从而能够更好地分类。

AUC的常用阈值 >0.7;有很强的区分度 0.6~0.7;有一定的区分度 0.5~0.6,有较弱的区分度;

低于0.5,区分度弱于随机猜测

- KS曲线

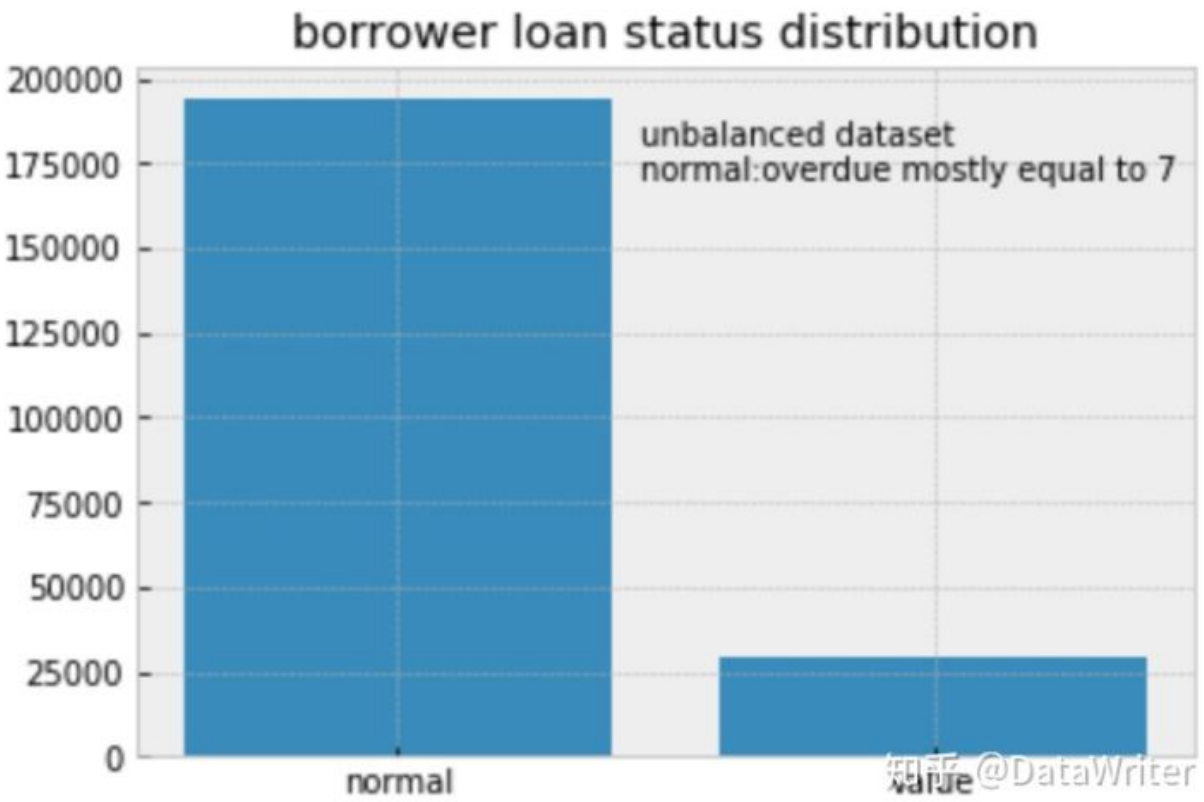
ks值大于0.3说明模型的区分里比较好，ks值大于0.2模型可用，但是区分力较差;ks值小 于0.2大于0，模型的区分力差不可用; 如果ks值为负数，说明评分与好坏程度相悖，模型出现错误。ks指标的缺点是:只能表示 区分度最好的分数的区分度，不能衡量其他分数。



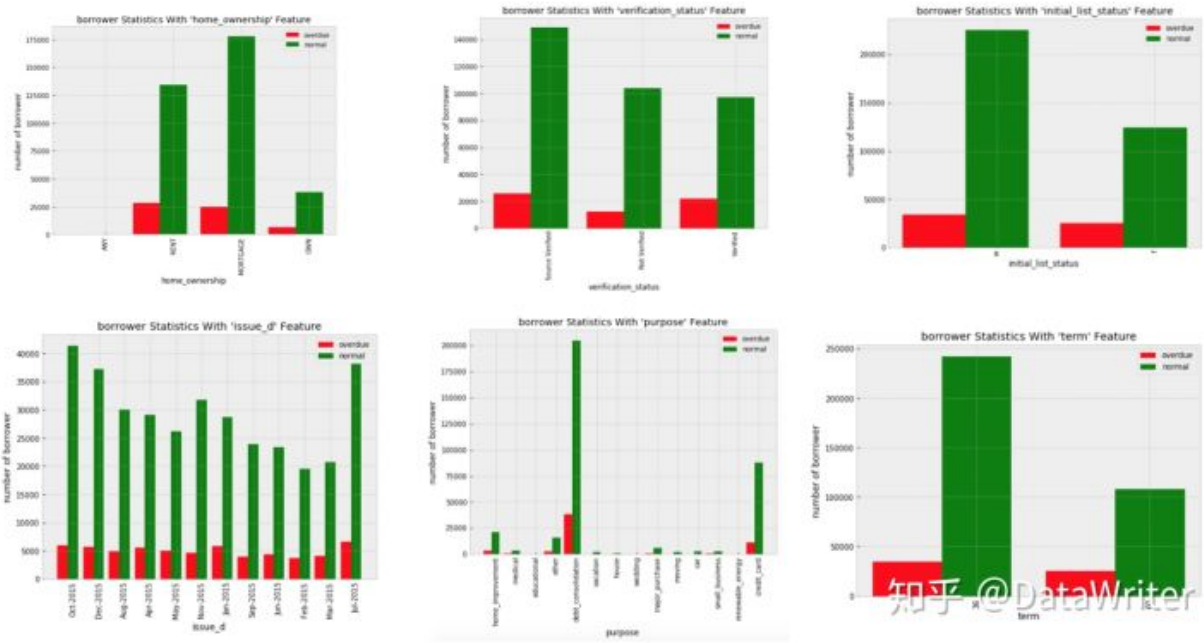
数据可视化探索 本项目主要用到的是国际著名的p2p信贷平台lendingclub官网开放的2016年第三季度借贷人数据进行分析，涉及42万条申请人样本，145个字段信息,其中好客户样本350199个，坏客户样本66175,属于样本分布不平衡数据。

根据巴塞尔协议，客户还款12个月后，还款状态会趋于稳定，在本次项目中开发模型使用到的客户数据的贷款期限是36期或60期，还款期限到了12个月。验证数据集选用2016年第四季度前两个月的数据进行验证，也会在两个月后完成相应的还款期限。

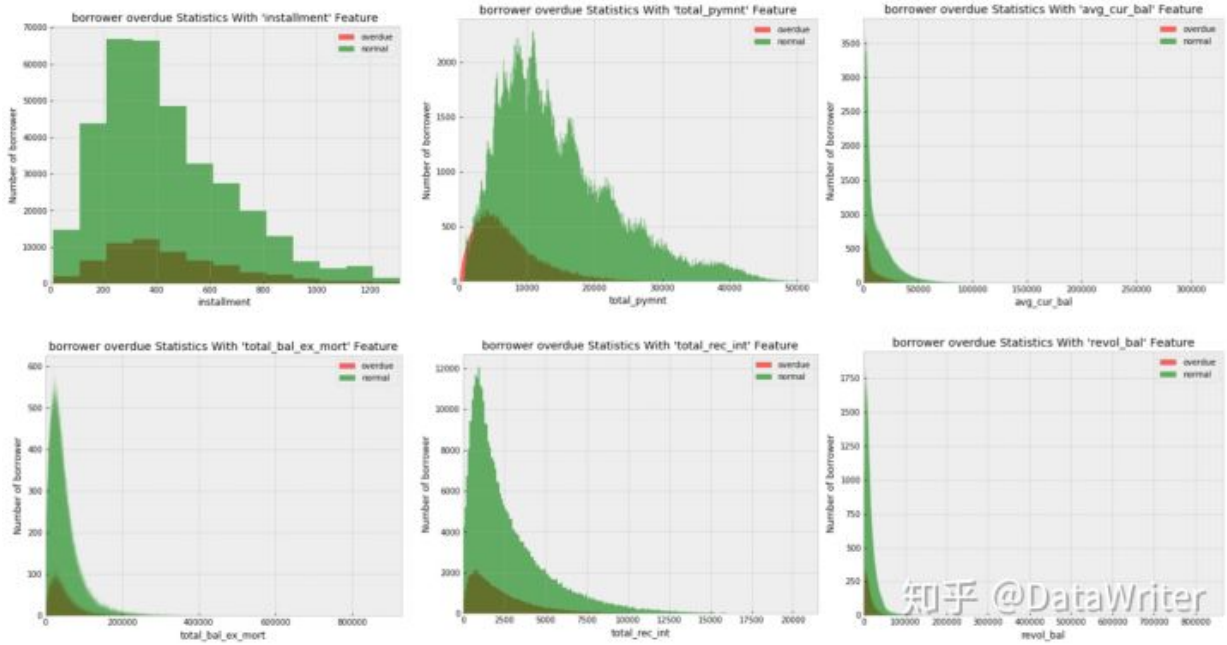
- 特征分布直方图



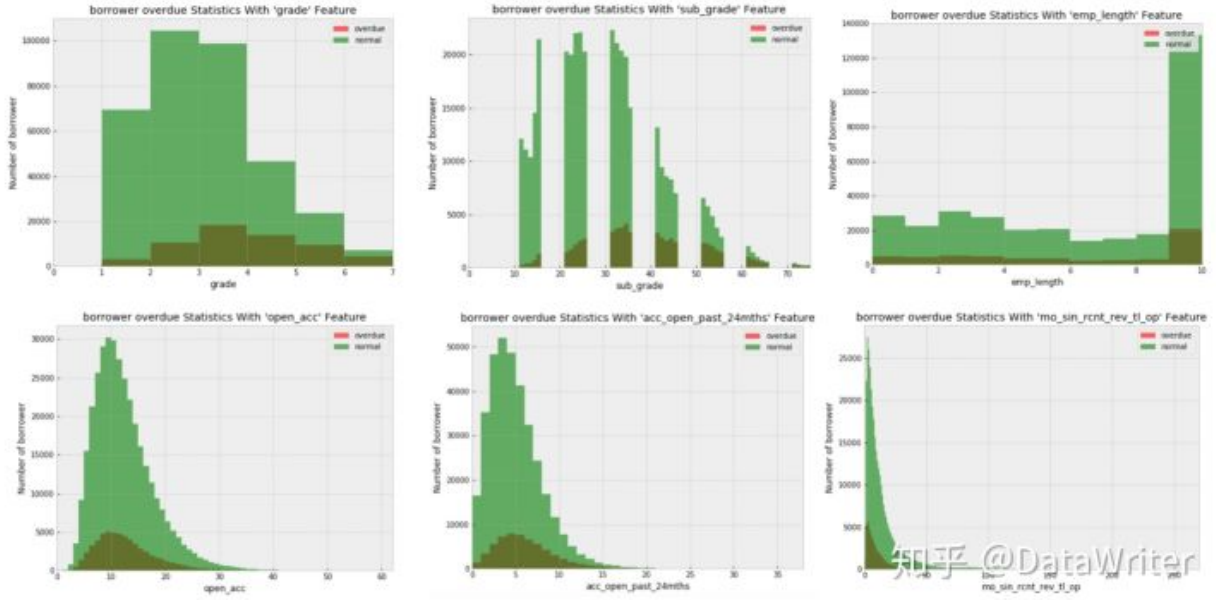
label分布直方图



分类特征分布直方图



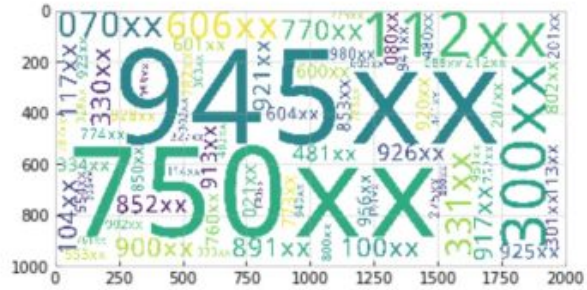
连续特征分布直方图



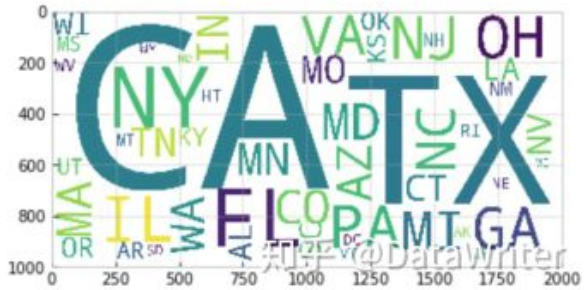
有序特征分布直方图

```
emp_title word cloud:
borrower with missing 'emp_title' values: 23196 (4381 overdue, 18815 normal)
```


```
zip_code word cloud:
```

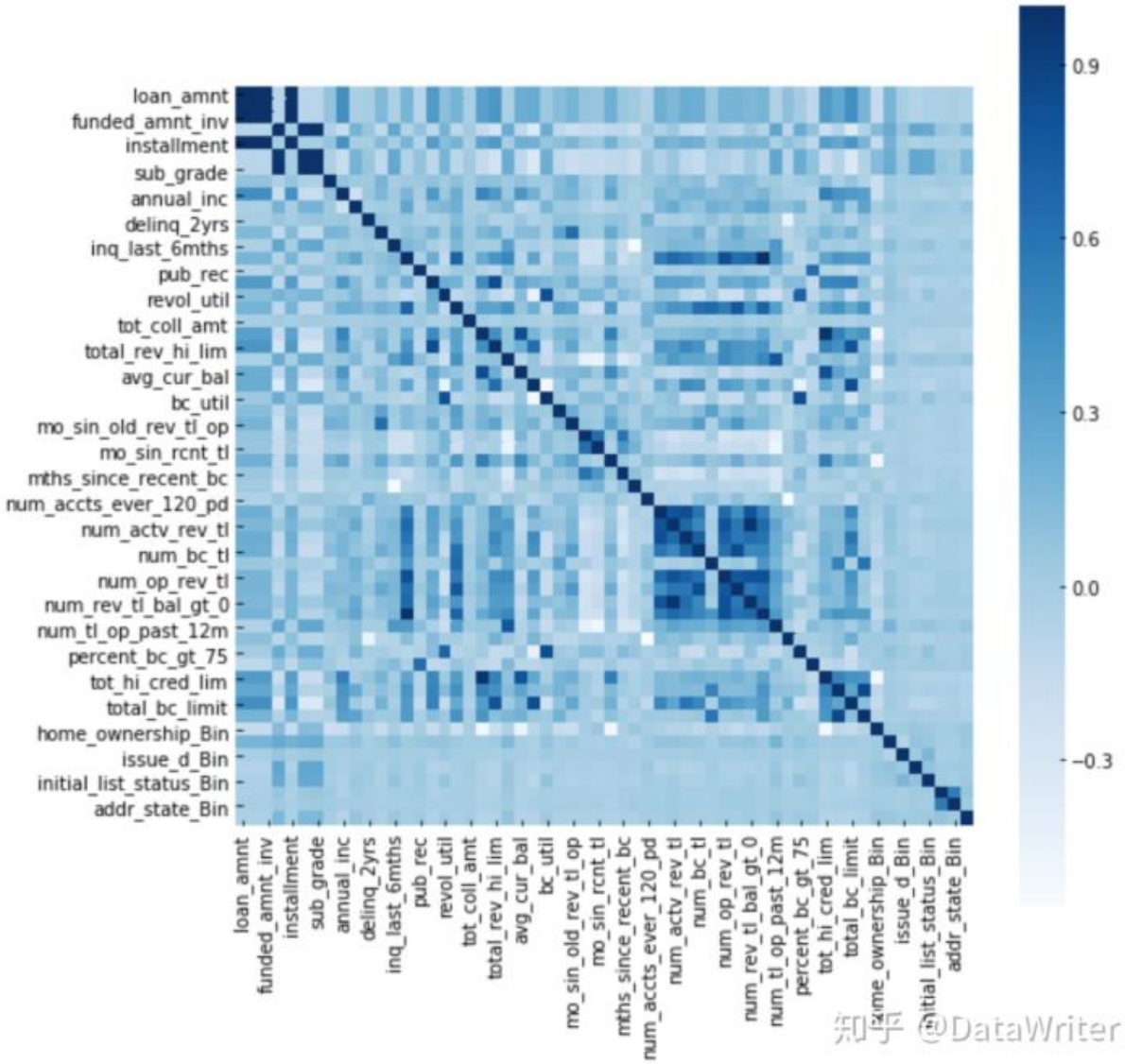



```
addr_state word cloud:
```

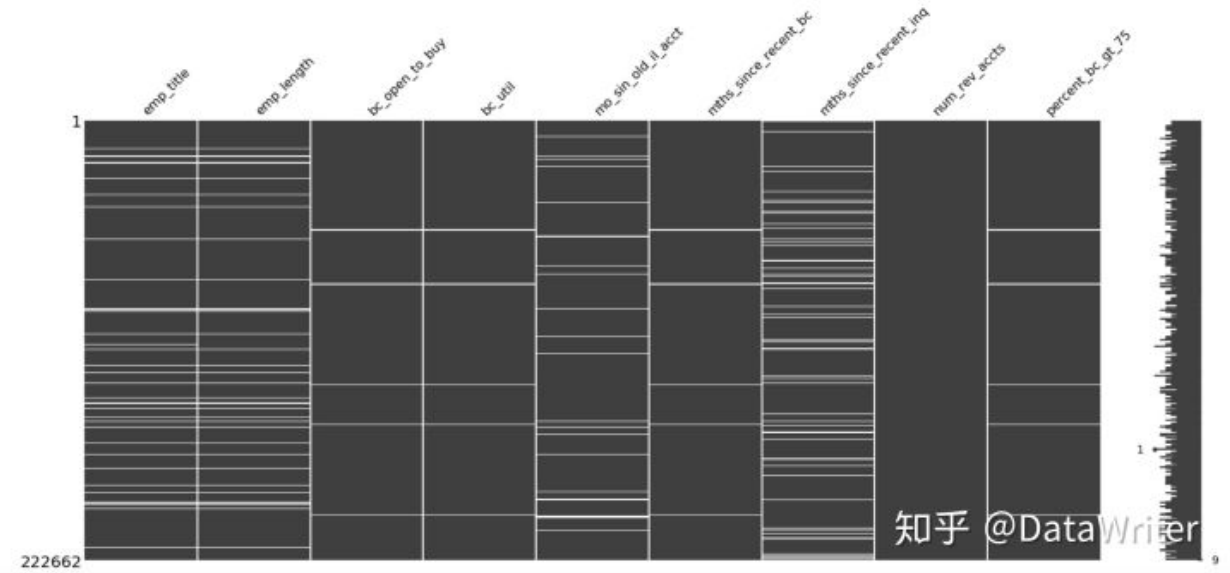


文本特征文字云图

- 特征相关关系矩阵热力图



- 特征缺失统计图



经过对数据的描述统计分析和可视化探索发现，数据集存在缺失严重，且部分特征全部缺失;部分连续型特征用离散型特征标记;部分特征之间存在共线性明显。

策略：

删除缺失比超过60%的特征:将本应是数值型的字符型特征转成数值型 异常处理: 存在明显的异常值，进行异常值调整 缺失处理:缺失填充和缺失插补 归一化:对连续特征进行min_max_scale处理 代码：

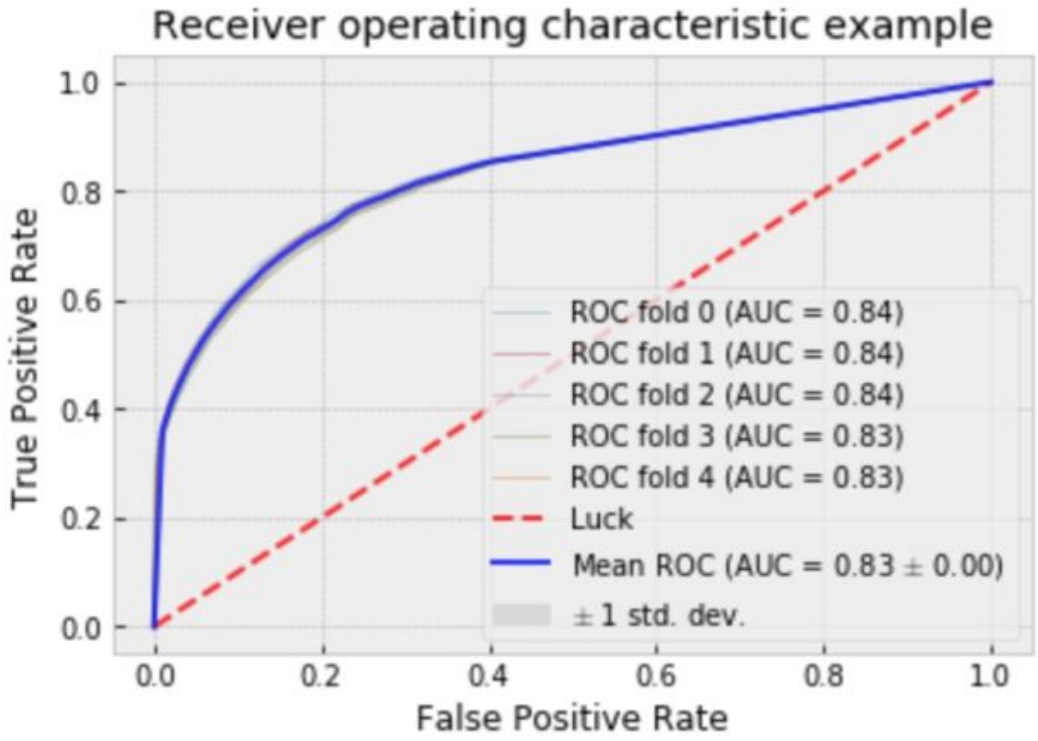
```
# drop columns if it's missing greater than 60%
def drop_missingmore60_col(data):
    miss_60_col = data.isnull().sum()
[data.isnull().sum()>=0.40*data.shape[0]].index
    df = data.drop(miss_60_col,axis=1)
    print('after drop missing greater than 60% columns, the data shape is
',df.shape)
    return df

#drop all null row and all null column
def drop_row_col_miss(data):
    data = data.dropna(how='all',axis=1)
    df = data.dropna(how='all',axis=0)
    return df

#delete 90% value same in one column
def drop_90samevalue_col(data):
    colum=data.columns
    per=pd.DataFrame(colum,index=colum)
    max_valuecounts=[]
    for col in colum:
        max_valuecounts.append(data[col].value_counts().max())
    per['mode']=max_valuecounts
    per['percentil'] =per['mode']/data.shape[0]
    same_value_col =per[per.sort_values(by='percentil',ascending=False)
['percentil']>0.9].index
    df = data.drop(same_value_col,axis=1)
    print('after delete 90% values same in one column,the data shape is',df.shape)
    return df
```

特征工程

在这个阶段首先将变量分为分类特征和连续特征，针对分类特征进行编码，对编码后的分类变量和数值变量一起进行卡方分箱，计算卡方分箱后的特征woe值和iv值，筛选iv大于0.01的特征针对筛选后的特征进行特征筛选(l1正则，特征重要度),对筛选后的特征进行建模(逻辑回归，随机森林，svm)。

基线模型结果

在这部分，我挑选了iv最大的10个特征进行训练，结果如下：




```
the ks value of base model is : 0.7557107520425962
```

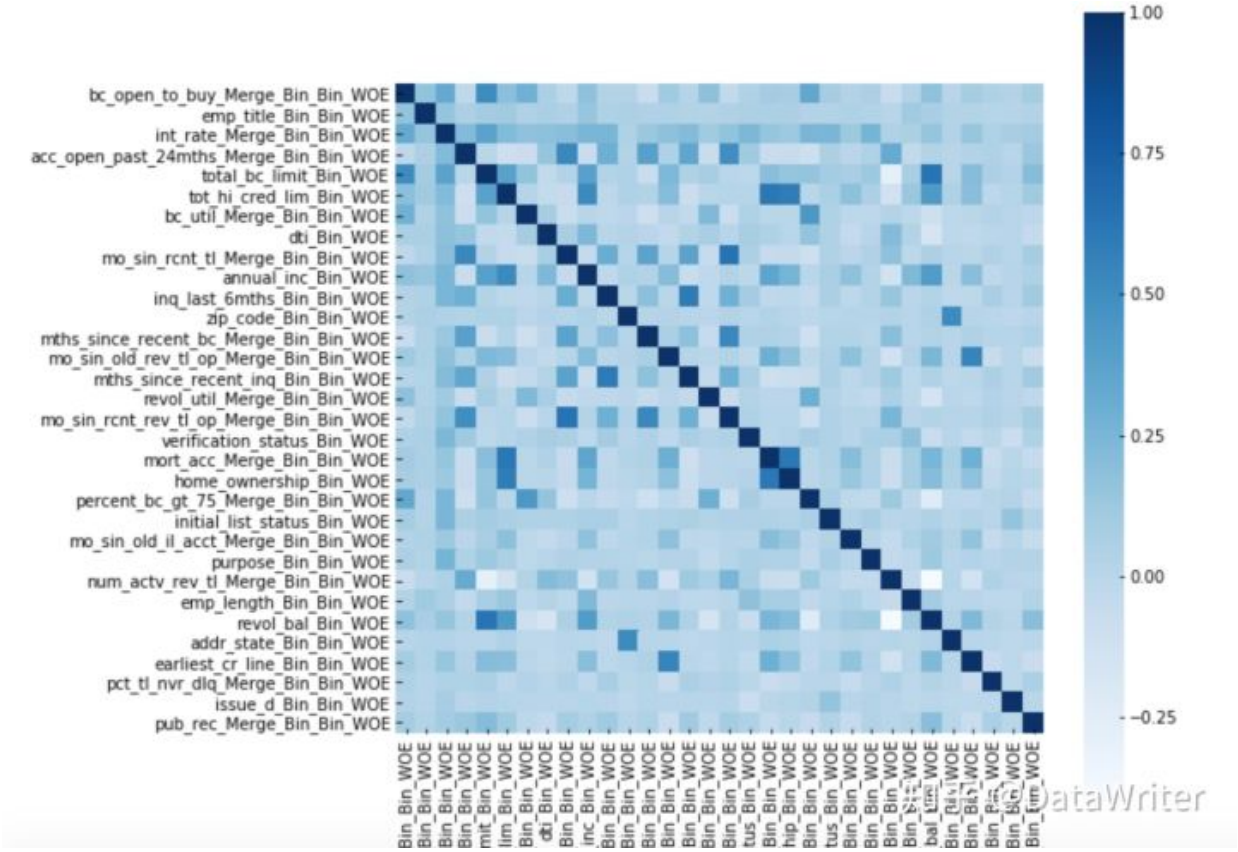

随机森林基线模型结果

```
# basic model
from sklearn.ensemble import RandomForestClassifier

# get array format data feature:X,label:y
X = np.array(base_model_data)
y = np.array(df_code['y'])

clf=RandomForestClassifier()
clf.fit(X,y)
pred = clf.predict_proba(X)[:,1]
df = pd.DataFrame([pred,y]).T
df.columns = ['score','y']
print('the 5 folds cross validation roc curve is:')
clf = RandomForestClassifier()
vs.model_roc_curve(X,y,clf)

print('\n')
print('the ks value of base model is :',lp.KS(df,'score','y'))
```

优化

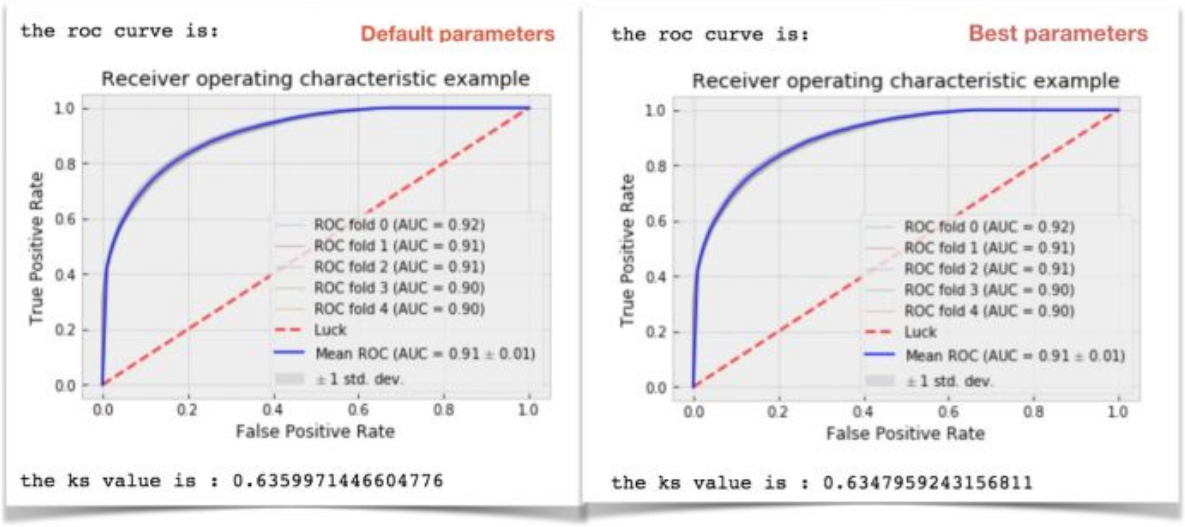
这部分我做了三步处理：

- 消除相关性：如果两个特征之间存在明显的相关性，删除iv值小的
- 构建模型：构建逻辑回归和随机森林模型
- 网格搜索：设定遍历参数范围，训练最优参数

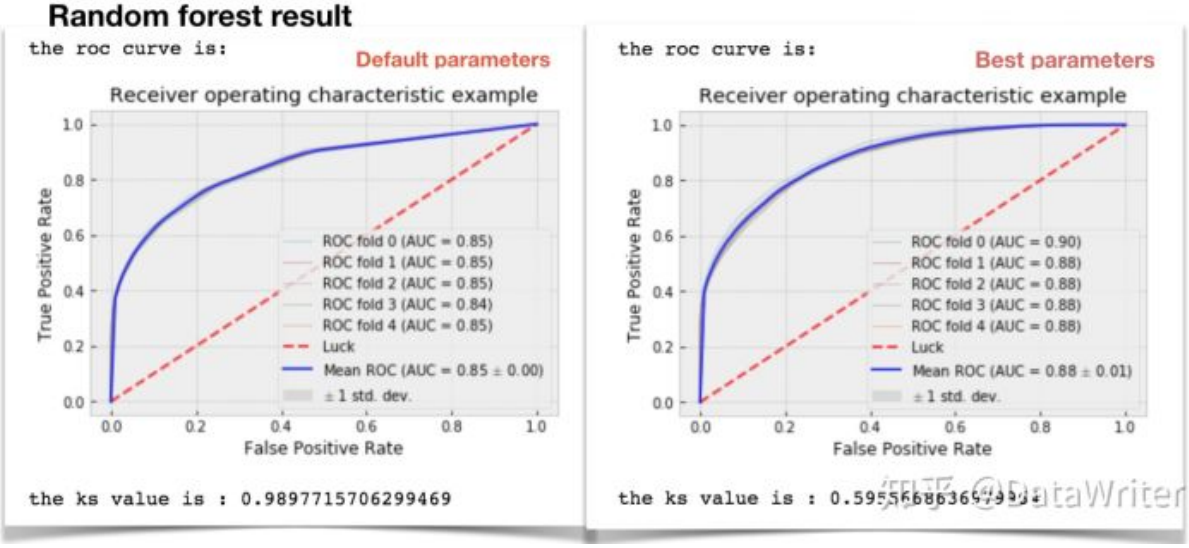


特征选择后相关系数矩阵热力图

Logistic regression result



Random forest result




```
the default parameters LogisticRegression algorithm: is:
{'C':1.0,'class_weight':None,'penalty':'l2'};the best parameters model 2:{'C':
0.01, 'class_weight': {0: 1, 1: 1}, 'penalty': 'l2'}

the default paramters RandomForestClassifier algorithm:
{'class_weight':None,'criterion':'gini','max_depth':None,'nn_estimators':10};the
best parameters model 4:{'class_weight': {0: 1, 1: 1}, 'criterion': 'gini',
'max_depth': 6, 'n_estimators': 20}
```

bc_open_to_buy_Merge_Bin_Bin_WOE	-1.030246
emp_title_Bin_Bin_WOE	-0.958251
int_rate_Merge_Bin_Bin_WOE	-0.564999
acc_open_past_24mths_Merge_Bin_Bin_WOE	-0.440555
total_bc_limit_Bin_WOE	0.651023
tot_hi_cred_lim_Bin_WOE	-0.143073
bc_util_Merge_Bin_Bin_WOE	-0.493194
dti_Bin_WOE	-0.365618
mo_sin_rcnt_tl_Merge_Bin_Bin_WOE	-0.188808
annual_inc_Bin_WOE	0.381110
inq_last_6mths_Bin_Bin_WOE	-0.144443
zip_code_Bin_Bin_WOE	-0.668091
mths_since_recent_bc_Merge_Bin_Bin_WOE	-0.495244
mo_sin_old_rev_tl_op_Merge_Bin_Bin_WOE	-0.434393
mths_since_recent_inq_Bin_Bin_WOE	-0.256700
revol_util_Merge_Bin_Bin_WOE	-0.614283
mo_sin_rcnt_rev_tl_op_Merge_Bin_Bin_WOE	0.104281
verification_status_Bin_WOE	-0.154370
mort_acc_Merge_Bin_Bin_WOE	-0.135657
home_ownership_Bin_WOE	-0.381213

入模系数

风控建模整体过程大致如上所述。