个人评分卡报告

徐诗雨 2168511083

1. 项目背景介绍

信用风险指的是交易对手未能履行约定合同中的义务造成经济损失的风险，即受信人不能履行还本付息的责任而使授信人的预期收益与实际收益发生偏离的可能性，它是金融风险的主要类型。在银行借贷场景中，评分卡是一种以分数形式来衡量一个客户的信用风险大小的手段，它衡量向受信人或需要融资的公司不能如期履行合同中的还本付息责任和让授信人或银行等金融机构造成经济损失的可能性的一种方式。一般来说，一个客户的评分越高，该客户违约的风险越小。主体评级和债项评级均有一系列评级模型组成，其中主体评级模型可用“四张卡”来表示，分别是 A 卡、B 卡、C 卡和 F 卡；债项评级模型通常按照主体的融资用途，分为企业融资模型、现金流融资模型和项目融资模型等。

A 卡，又称为申请者评级模型，主要应用于相关融资类业务中新用户的主体评级，适用于个人和机构融资主体。B 卡，又称为行为评级模型，主要应用于相关融资类业务中存量客户在续存期内的管理，如对客户可能出现的逾期、延期等行为进行预测，仅适用于个人融资主体。C 卡，又称为催收评级模型，主要应用于相关融资类业务中存量客户是否需要催收的预测管理，仅适用于个人融资主体。F 卡，又称为欺诈评级模型，主要应用于相关融资类业务中新客户可能存在的欺诈行为的预测管理，适用于个人和机构融资主体。

2. 信用卡评分模型开发

2.1 数据的获取及其预处理

2.1.1 数据的获取

接下来将进行模型的构建，首先进行数据的获取及其预处理。

需要获取包括存量客户包括获取存量客户及潜在客户的数据。存量客户是指已经在证券公司开展相关融资类业务的客户，包括个人客户和机构客户；潜在客户是指未来拟在证券公司开展相关融资类业务的客户，主要包括机构客户，这也是解决证券业样本较少的常用方法，这些潜在机构客户包括上市公司、公开发行债券的发债主体、新三板上市公司、区域股权交易中心挂牌公司、非标融资机构等。

本项目数据来源于 kaggle 竞赛“Give Me Some Credit”，数据当中变量的含义与定义如表 1 所示。

表 1 数据集当中各个变量的含义

Variable Name	Description	Type
SeriousDlqin2yrs	Person experienced 90 days past due delinquency or worse	Y/N
Revolving Utilization Of	Total balance on credit cards and personal lines of	percentage

Unsecured Lines	credit except real estate and no installment debt like car loans divided by the sum of credit limits	
age	Age of borrower in years	integer
Number Of Time 30-59 Days Past Due Not Worse	Number of times borrower has been 30-59 days past due but no worse in the last 2 years.	integer
Debt Ratio	Monthly debt payments, alimony,living costs divided by monthy gross income	percentage
Monthly Income	Monthly income	real
Number Of Open Credit Lines And Loans	Number of Open loans (installment like car loan or mortgage) and Lines of credit (e.g. credit cards)	integer
Number Of Times 90 Days Late	Number of times borrower has been 90 days or more past due.	integer
Number Real Estate Loans Or Lines	Number of mortgage and real estate loans including home equity lines of credit	integer
Number Of Time 60-89 Days Past Due Not Worse	Number of times borrower has been 60-89 days past due but no worse in the last 2 years.	integer
Number Of Dependents	Number of dependents in family excluding themselves (spouse, children etc.)	integer

2.1.2 数据预处理

在对数据处理之前，需要对数据的缺失值和异常值情况进行了解。通过 describe 函数，可以了解数据集的缺失值、均值和中位数等基本信息，数据整体的初步统计情况如图 1 所示：

	Unnamed: 0	SeriousDlqin2yrs	RevolvingUtilizationOfUnsecuredLines	age	NumberOfTime30-59DaysPastDueNotWorse	DebtRatio	MonthlyIncome	NumberOfOpenCreditLinesAndLoans
count	150000.000000	150000.000000	150000.000000	150000.000000	150000.000000	150000.000000	120269.000000	150000.000000
mean	75000.500000	0.066840	6.048438	52.295207	0.421033	353.005076	6670.221237	8.452760
std	43301.414527	0.249746	249.755371	14.771866	4.192781	2037.818523	14384.674215	5.145951
min	1.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
25%	37500.750000	0.000000	0.029867	41.000000	0.000000	0.175074	3400.000000	5.000000

NumberOfTimes90DaysLate	NumberRealEstateLoansOrLines	NumberOfTime60-89DaysPastDueNotWorse	NumberOfDependents
150000.000000	150000.000000	150000.000000	146076.000000
0.265973	1.018240	0.240387	0.757222
4.169304	1.129771	4.155179	1.115086
0.000000	0.000000	0.000000	0.000000
0.000000	0.000000	0.000000	0.000000

图 1 数据整体的初步统计情况

从上图可知，变量 Monthly Income 缺失 29731 个数据，Number Of Dependents 缺失 3924 个数据，数据确实量非常大，较为影响接下来的数据分析工作，因此接下来要进行数据缺失值的预处理。

2.1.3 缺失值处理

由图 1 可知 Monthly Income 及 Number Of Dependents 两个变量出现了缺失值。由于 Monthly Income 缺失值达到 29731 条数据，比例较大，因此不能直接将缺失值删除，所以选择随机森林法对 Monthly Income 缺失值预测填充函数。而 Number Of Dependents 的缺失较少，对数据影响不大，因此可以选择直接删除。使用 dropna 函数删除空值，使用 drop_duplicates 删除重复值，最后将预测的值往原有数据集当

中进行填补。

```python
# 缺失值预处理
import pandas as pd
import matplotlib.pyplot as plt #导入图像库
from sklearn.ensemble import RandomForestRegressor

# 用随机森林对缺失值预测填充函数
def set_missing(df):
    # 把已有的数值型特征取出来
    process_df = df.iloc[:,[5,0,1,2,3,4,6,7,8,9]]
    # 分成已知该特征和未知该特征两部分
    known = np.array(process_df[process_df.MonthlyIncome.notnull()])
    unknown =np.array( process_df[process_df.MonthlyIncome.isnull()])
    X = known[:, 1:]
    y = known[:, 0]

    rfr = RandomForestRegressor(random_state=0, n_estimators=200,max_depth=3)
    rfr.fit(X, y)
    # 用得到的模型进行未知特征值预测
    unknown=pd.DataFrame(unknown)
    unknown=unknown.fillna(value=pd.DataFrame(known).mean())
    unknown=np.array(unknown)
    predicted = rfr.predict(unknown[:, 1:]).round(0)
    print(predicted)
    # 用得到的预测结果填补原缺失数据
    df.loc[(df.MonthlyIncome.isnull()), 'MonthlyIncome'] = predicted
    return df

data=set_missing(data)  # 用随机森林填补比较多的缺失值
data=data.dropna()
data = data.drop_duplicates()  # 删除重复项
data.to_csv('MissingData.csv', index=False)
data.describe().to_csv('MissingDataDescribe.csv')
```

图 2 缺失值预处理

2.1.4 异常值处理

本数据中有一些客户的年龄大于 100 或小于 0，这在现实生活中几乎不可能出现，也不会有物理意义，因此接下来需要找出样本总体中的异常值，通常采用离群值检测的方法。离群值检测的方法有单变量离群值检测、局部离群值因子检测、基于聚类方法的离群值检测等方法，在本数据集中，将要采用单变量离群值检测来判断异常值，生成的箱线图如图 3 所示。

对于 age 变量而言，我们认为大于 100 岁和小于等于 0 岁的为异常值，由箱线图可知，异常值样本不多，故可以直接删除。对于“Number Of Time 30-59 Days Past Due Not Worse”、“Number Of Times 90 Days Late”、“Number Of Time 60-89 Days Past Due Not Worse”这三个变量均存在部分接近 100 的异常值，因此予以剔除，结果如图 4 所示。

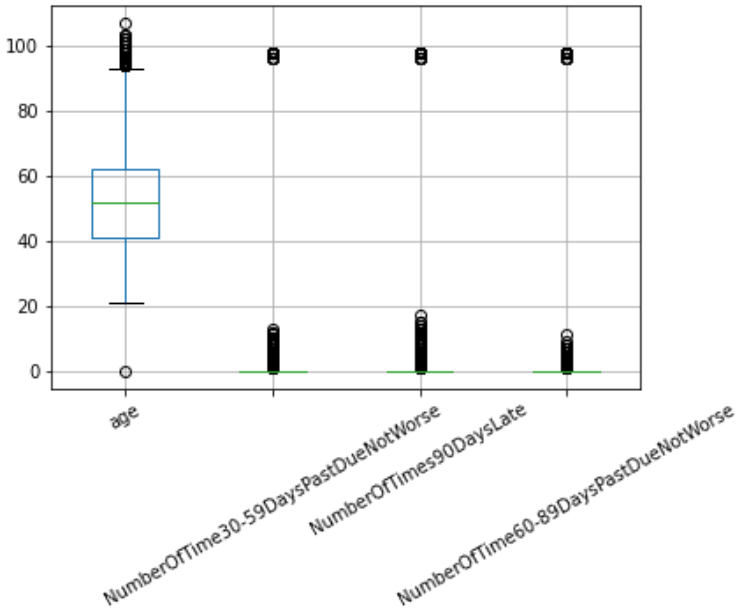

图 3 剔除异常值前数据箱线图

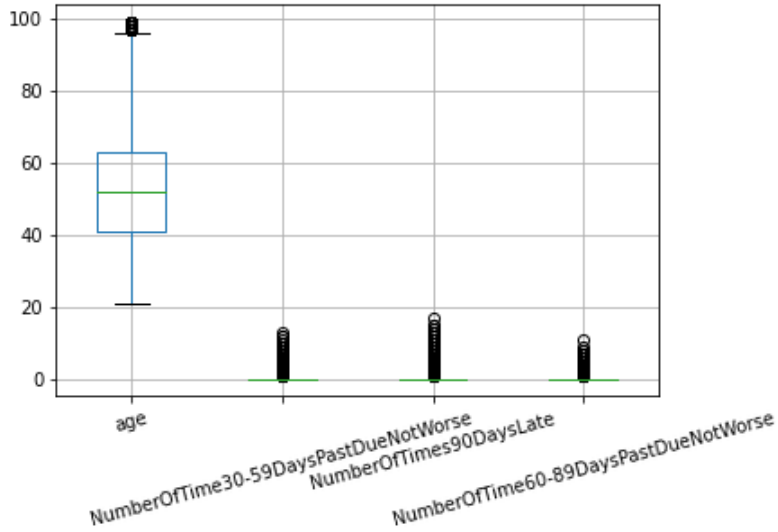

图 4 剔除异常值后数据箱线图

此外，数据集中能够正常履行合约的客户被记录为 0，不能正常履约的客户表示为 1，这里按照个人理解对其做反向赋值，即能正常履约并支付利息的客户为 1，所以我们将其取反。

2.1.5 数据集划分

为了验证模型的拟合效果，我们需要对训练集数据进行划分，将其中的 70%用于模型的训练，剩下的 30%用于验证拟合效果。

```
X_train, X_test, Y_train, Y_test = train_test_split(X, Y, test_size=0.3, random_state=0)
print(Y_train)
train = pd.concat([Y_train, X_train], axis=1)
test = pd.concat([Y_test, X_test], axis=1)

train.to_csv('TrainData.csv',index=False)
test.to_csv('TestData.csv',index=False)
print(train.shape)
print(test.shape)
```

图 5 数据划分

2.1.6 探索性分析

在建立模型之前，我们需要对现有的数据进行探索性数据分析（Exploratory Data Analysis），EDA 是指对已有的数据进行初步探索探索。接下来将分析客户的年龄分布。客户年龄分布如图 6 所示，客户月收入分布如图 7 所示，可以看到这两个变量都呈正态分布，呈现一定的偏态分布特征，符合统计规律。

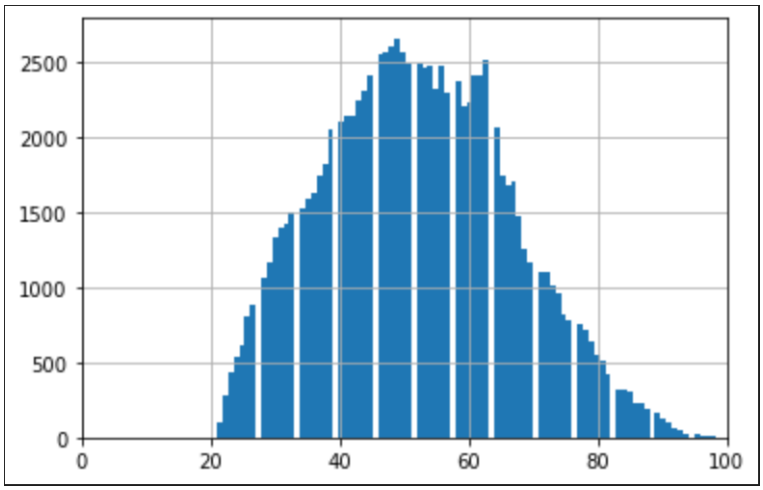

图 6 客户年龄分布

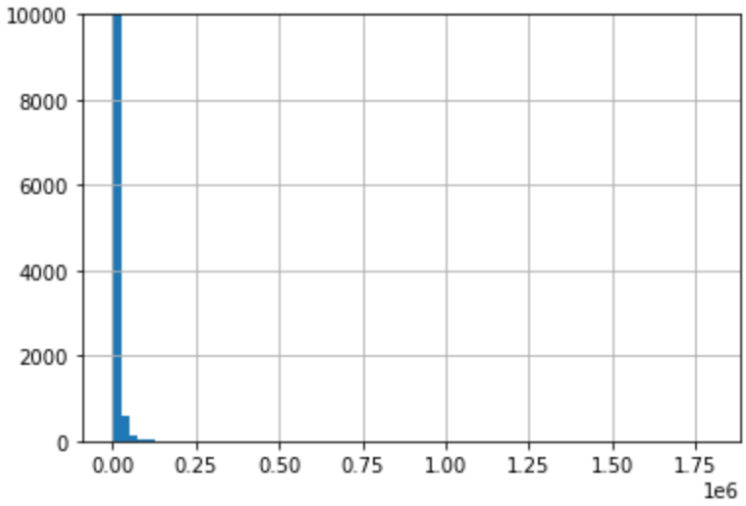

图 7 客户收入分布

2.2 单变量分析

分析客户当中能够正常履约的以及不能正常履约的各自占总客户的人数比例，结果如图 8 所示，大约有 7.13%的客户不能正常履约。

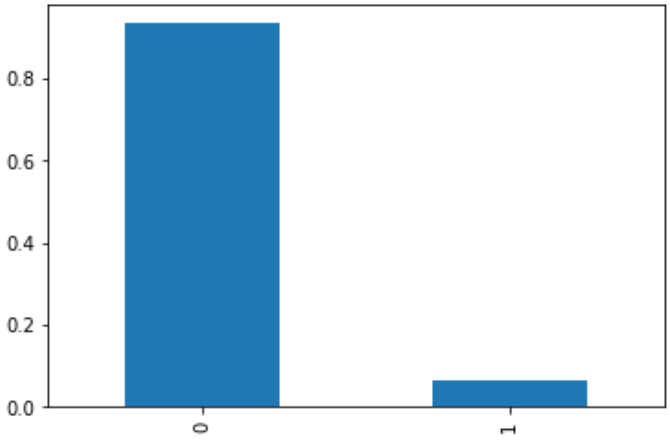

图 8 不同客户占比分析

2.3 多变量分析

多变量分析主要用于分析变量之间的相关的程度，采用 corr 函数统计各变量间的相关性。由图 9 知，各变量间相关性较小，几乎不具有共线性。

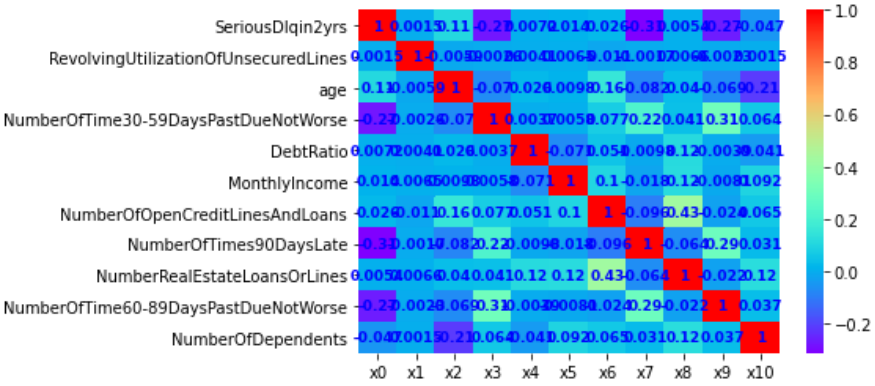

图 9 多变量分析

3. 模型构建

3.1 特征的选择

在机器学习模型构建的过程中特征的选择非常重要，选用较好的特征能够构造出较好的模型，参考 Scikit-learn 介绍几种常用的特征选择方法。在此，我们采用信用卡评分模型常用的 IV 值筛选，接下来首先进行特征分箱。

3.1.1 特征的分箱

特征的分箱指的是离散化是指将连续属性，特征或变量转换或划分为离散或标称属性/特征/变量/间隔的过程。这在创建概率质量函数时非常有用 – 正式地，

在密度估计中。它是一种离散化的形式，也可以是分组，如制作直方图。每当连续数据离散化时，总会存在一定程度的离散化误差。目标是将数量减少到手头的建模目的可忽略不计的水平。

特征的分箱有许多种方法，例如 Best-KS，ChiMerge、等频、等距、聚类等方法。在使用时根据不同的数据特征需要采用不同分箱方式。

```
if __name__ == '__main__':
    data = pd.read_csv('TrainData.csv')
    pinf = float('inf')  # 正无穷大
    ninf = float('-inf')  # 负无穷大
    dfx1, ivx1,cutx1,woex1=mono_bin1(data.SeriousDlqin2yrs,data.RevolvingUtilizationOfUnsecuredLines,n=10)
    dfx2, ivx2,cutx2,woex2=mono_bin1(data.SeriousDlqin2yrs, data.age, n=10)
    dfx4, ivx4,cutx4,woex4 =mono_bin1(data.SeriousDlqin2yrs, data.DebtRatio, n=20)
    dfx5, ivx5,cutx5,woex5 =mono_bin1(data.SeriousDlqin2yrs, data.MonthlyIncome, n=10)
    # 连续变量离散化
    cutx3 = [ninf, 0, 1, 3, 5, pinf]
    cutx6 = [ninf, 1, 2, 3, 5, pinf]
    cutx7 = [ninf, 0, 1, 3, 5, pinf]
    cutx8 = [ninf, 0,1,2, 3, pinf]
    cutx9 = [ninf, 0, 1, 3, pinf]
    cutx10 = [ninf, 0, 1, 2, 3, 5, pinf]
```

图 10 特征的分箱

3.1.2 WOE 值计算

在数据完成分箱以后，接下来需要计算各个档位的 WOE 值，观察 WOE 值随指标变化的趋势。其中正向指标越大，WOE 值越小，反向指标越大，WOE 值越大。正向指标的 WOE 值负斜率越大，反响指标的正斜率越大，则说明指标区分能力好。WOE 值趋近于直线，则意味指标判断能力较弱。若正向指标和 WOE 正相关趋势、反向指标同 WOE 出现负相关趋势，则说明此指标不符合经济意义，则应当予以去除。WOE 的定义如式(1)所示。

$$WOE = \ln \frac{good\ attribute}{bad\ attribute} \tag{1}$$

```
def self_bin(Y,X,cat):
    good=Y.sum()
    bad=Y.count()-good
    d1=pd.DataFrame({'X':X,'Y':Y,'Bucket':pd.cut(X,cat)})
    d2=d1.groupby('Bucket', as_index = True)
    d3 = pd.DataFrame(d2.X.min(), columns=['min'])
    d3['min'] = d2.min().X
    d3['max'] = d2.max().X
    d3['sum'] = d2.sum().Y
    d3['total'] = d2.count().Y
    d3['rate'] = d2.mean().Y
    d3['woe'] = np.log((d3['rate'] / (1 - d3['rate'])) / (good / bad))
    d3['goodattribute'] = d3['sum'] / good
    d3['badattribute'] = (d3['total'] - d3['sum']) / bad
    iv = ((d3['goodattribute'] - d3['badattribute']) * d3['woe']).sum()
    d4 = (d3.sort_index())
    print("=" * 60)
    print(d4)
    woe = list(d4['woe'].round(3))
    return d4, iv,woe
```

图 11 计算 WOE 值

3.1.3 IV 值的计算

IV（Information Value）为信息价值。经过分析，数据当中各变量的 IV 值如图 12 所示，在该图当中，定义 IV 值低于 0.2 的特征为预测能力较弱或无关特征，可以看出，“Debt Ratio”、“Monthly Income”、“Number Of Open Credit Lines And Loans”、“Number Real Estate Loans Or Lines”、“Number Of Dependents”五个变量的 IV 值较低，予以删除。

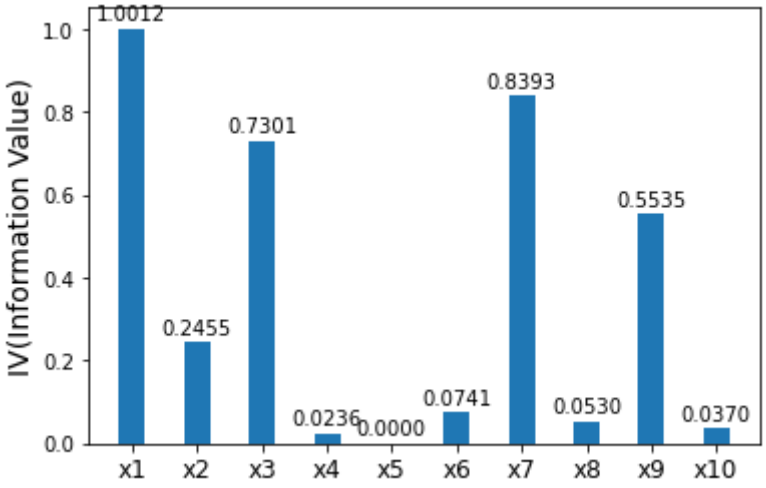

图 12 各变量 IV 值

3.2 模型的构造

3.2.1 WOE 值替换

将筛选后的特征变量进行 WOE 转换，以减少逻辑回归的自变量处理量。

```
#测试集的替换成woe
test['RevolvingUtilizationOfUnsecuredLines'] = pd.Series(replace_woe(test['RevolvingUtilizationOfUnsecuredLines'
test['age'] = pd.Series(replace_woe(test['age'], tcutx2, twoex2))
test['NumberOfTime30-59DaysPastDueNotWorse'] = pd.Series(replace_woe(test['NumberOfTime30-59DaysPastDueNotWorse'
test['DebtRatio'] = pd.Series(replace_woe(test['DebtRatio'], tcutx4, twoex4))
test['MonthlyIncome'] = pd.Series(replace_woe(test['MonthlyIncome'], tcutx5, twoex5))
test['NumberOfOpenCreditLinesAndLoans'] = pd.Series(replace_woe(test['NumberOfOpenCreditLinesAndLoans'], tcutx6,
test['NumberOfTimes90DaysLate'] = pd.Series(replace_woe(test['NumberOfTimes90DaysLate'], tcutx7, twoex7))
test['NumberRealEstateLoansOrLines'] = pd.Series(replace_woe(test['NumberRealEstateLoansOrLines'], tcutx8, twoex
test['NumberOfTime60-89DaysPastDueNotWorse'] = pd.Series(replace_woe(test['NumberOfTime60-89DaysPastDueNotWorse'
test['NumberOfDependents'] = pd.Series(replace_woe(test['NumberOfDependents'], tcutx10, twoex10))
test.to_csv('TestWoeData.csv', index=False)
```

图 13 WOE 值替换

3.2.2 LR 建模

采用 Logistic Regression 来进行模型的构建。

```
data = pd.read_csv('WoeData.csv')

df=data.drop(['DebtRatio','MonthlyIncome','NumberOfOpenCreditLinesAndLoans','NumberRealEstateLoansOrLines','Num
df.rename(columns={'NumberOfTime30-59DaysPastDueNotWorse':'NumberOfTimedays3059PastDueNotWorse'},inplace=True)
df.rename(columns={'NumberOfTime60-89DaysPastDueNotWorse':'NumberOfTimedays6089PastDueNotWorse'},inplace=True)
df.rename(columns={'SeriousDlqin2yrs':'y'},inplace=True)
#应变量
Y=df["y"]
#自变量，剔除对因变量影响不明显的变量
X=df.drop(['y'],axis=1)
X1=sm.add_constant(X)
logit1=sm.Logit(np.array(Y),np.array(X))
result1=logit1.fit()
print(result1.summary())
logit1.fit().aic#获取随机函数的AIC值
```

图 14 模型的建立

输出结果为：

```
                 coef    std err          z      P>|z|      [0.025      0.975]
------------------------------------------------------------------------------
x1             1.0329      0.028     36.530      0.000       0.977       1.088
x2             0.5436      0.027     19.799      0.000       0.490       0.597
x3            -1.4392      0.050    -29.029      0.000      -1.536      -1.342
x4             0.8644      0.014     62.920      0.000       0.837       0.891
x5            -0.4929      0.045    -10.875      0.000      -0.582      -0.404
==============================================================================
```

图 15 输出的结果

4. 模型的检验

模型建立以后，需要对模型的预测能力进行验证，这里考虑使用在建模开始阶段预留的 test 数据，通过 ROC 曲线和 AUC 来评估模型的拟合能力。利用 sklearn.metrics 计算，它能方便比较两个分类器，自动计算 ROC 和 AUC。

```
from sklearn.metrics import roc_curve, auc
resu = lr.fit().predict(X_train)

#进行预测
fpr, tpr, threshold = roc_curve(Y_train, resu)
rocauc = auc(fpr, tpr)#计算AUC
plt.plot(fpr, tpr, 'b', label='AUC = %0.2f' % rocauc)#生成ROC曲线
plt.legend(loc='lower right')
plt.plot([0, 1], [0, 1], 'r--')
plt.xlim([0, 1])
plt.ylim([0, 1])
plt.ylabel('真正率')
plt.xlabel('假正率')
plt.show()
```

图 16 模型的检验

从图 17 可知，AUC 值为 0.74，说明该模型的预测正确率较高，效果较好。

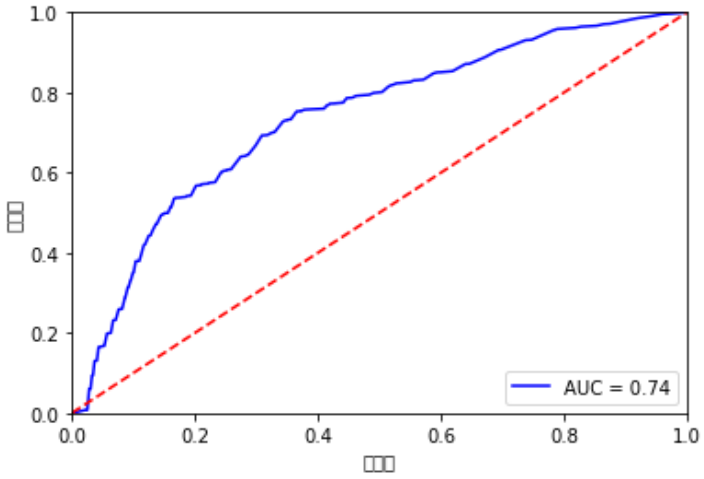

图 17 训练集 AUC

测试集下的 AUC 结果如图 18 所示。

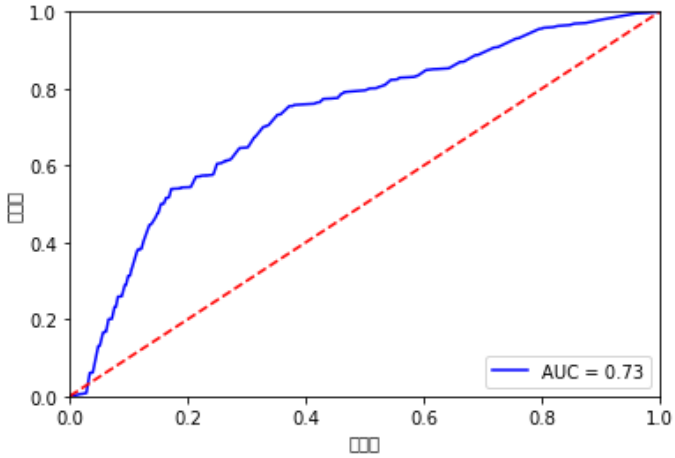

图 18 测试集 AUC

模型经过优化后重新计算 AUC，其结果如所示，可知经过优化后的模型效果更好。

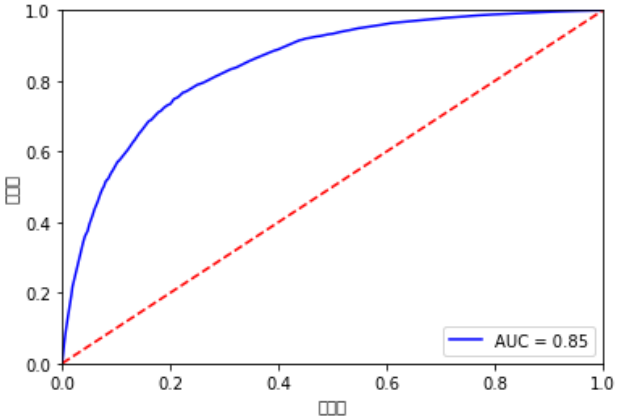

图 19 模型优化后 AUC

5. 信用评分

上面的模型可以预测出每个客户能否正常履约，也可以计算出每个客户能否正常履约的概率值，此时功能基本完成，但是结果却不够直观，接下来还需要将输出结果转换成相应的分数，以获得更加直观的结果，转换结果保存在“ScoreData.csv”当中。

6. 总结

本文选择 kaggle 上的 Give Me Some Credit 数据作为原始数据进行分析，结合信用评分卡的建立原理，首先利用随机森林模型进行了数据的预处理，随后进行数据的预分析，选择了变量进行模型的建立，最后创建了一个简单的信用评分系统。本项目还有许多不足之处，比如分箱应当使用最优分箱或卡方分箱，减少人为分箱的随机性，此外模型采用的是逻辑回归算法

7. 参考文献

信用卡评分模型
信用标准评分卡模型开发及实现
Python 异常值处理与检测
结合 Scikit-learn 介绍几种常用的特征选择方法
https://cloud.tencent.com/developer/article/1092230