4. 特征工程

特征工程指的是把原始数据转变为模型的训练数据的过程，它的目的就是获取更好的训练数据特征，使得模型的性能得到提升，有时甚至在简单的模型上也能取得不错的效果。特征工程在机器学习中占有非常重要的作用，一般认为包括特征构建、特征提取、特征选择三个部分。特征构建比较麻烦，需要一定的经验。 特征提取与特征选择都是为了从原始特征中找出最有效的特征。它们之间的区别是特征提取强调通过特征转换的方式得到一组具有明显物理或统计意义的特征；而特征选择是从特征集合中挑选一组具有明显物理或统计意义的特征子集。两者都能帮助减少特征的维度、数据冗余，特征提取有时能发现更有意义的特征属性，特征选择的过程经常能表示出每个特征的重要性对于模型构建的重要性。

4.1 缺失值填充

首先可以查询数据中的对象特征和数值特征：

```
numerical feature
 ['id', 'loanAmnt', 'term', 'interestRate', 'installment', 'employment
Title', 'homeOwnership', 'annualIncome', 'verificationStatus', 'purpos
e', 'postCode', 'regionCode', 'dti', 'delinquency_2years', 'ficoRangeL
ow', 'ficoRangeHigh', 'openAcc', 'pubRec', 'pubRecBankruptcies', 'revo
lBal', 'revolUtil', 'totalAcc', 'initialListStatus', 'applicationTyp
e', 'title', 'policyCode', 'n0', 'n1', 'n2', 'n3', 'n4', 'n5', 'n6',
'n7', 'n8', 'n9', 'n10', 'n11', 'n12', 'n13', 'n14']
category feature
 ['grade', 'subGrade', 'employmentLength', 'issueDate', 'earliesCredit
Line']
```

图 4.1 对象特征与数值特征

- 缺失值替换为 0：data_train = data_train.fillna(0)；
- 缺失值上方值复制：data_train = data_train.fillna(axis=0,method='ffill')
- 缺失值下方值复制：data_train = data_train.fillna(axis=0,method='bfill',limit=2)

```
employmentTitle          1
employmentLength     46799
postCode                 1
regionCode               0
dti                    239
pubRecBankruptcies     405
revolBal                 0
revolUtil              531
title                    1
policyCode               0
n0                   40270
n1                   40270
n2                   40270
n3                   40270
n4                   33239
n5                   40270
n6                   40270
n7                   40270
n8                   40271
n9                   40270
n10                  33239
n11                  69752
n12                  40270
n13                  40270
n14                  40270
```

图 4.2 存在缺失值的特征

employmentLength	46799
0	2 years
1	5 years
2	8 years
3	10+ years
4	NaN
...	
799995	7 years
799996	10+ years
799997	10+ years
799998	10+ years
799999	5 years

图 4.3 ‘employmentLength’未被填充

对数值特征使用中位数进行填充，对对象特征使用众数进行填充后，发现对象特征‘employmentLength’未完成填充，且需要对时间特征‘issueDate’和‘earliestCreditLine’进行数据类型转换操作：将‘issueDate’转换成时间间隔天数，将‘earliestCreditLine’转换成后四位代表的年份。

issueDateDT		employmentLength		earliesCreditLine	
0	2587	0.0	15989	0	1974
1	2952	1.0	13182	1	2001
2	3410	2.0	18207	2	2006
3	2710	3.0	16011	3	2002
4	3775	4.0	11833	4	2000
...		5.0	12543	...	
199995	1949	6.0	9328	199995	2005
199996	3044	7.0	8823	199996	2006
199997	2222	8.0	8976	199997	2001
199998	3775	9.0	7594	199998	2005
199999	2802	10.0	65772	199999	2002
		NaN	11742		

图 4.4 特殊特征处理结果

对于其余的类别特征也可进一步处理，对于等级类特征，可以进行 label encode 或自映射：

grade		subGrade		homeOwnership	
0	5	0	E2	0	2
1	4	1	D2	1	0
2	4	2	D3	2	0
3	1	3	A4	3	1
4	3	4	C2	4	1
..		..		..	
799995	3	799995	C4	799995	1
799996	1	799996	A4	799996	0
799997	3	799997	C3	799997	1
799998	1	799998	A4	799998	0
799999	2	799999	B3	799999	0

图 4.5 类别特征处理结果

4.2 异常值处理

检测异常的方法有：均方差及箱型图：

1. 均方差

依据 3-sigma 原则，对于近似正态分布的数据，数据约 68% 在$\mu \pm \delta$内，约 95% 在$\mu \pm 2\delta$内，约 99.7% 在$\mu \pm 3\delta$内。

```
Name: homeOwnership_outliers, dtype: int64
homeOwnership_outliers
异常值         62
正常值     159548
```

图 4.6 均方差检验样例

4.3 数据分桶

数据存在极值过大的特征，对于 KNN 之类使用欧氏距离作为相似度衡量的算法来说，这些特征都会造成大值覆盖小值的结果。而特征分桶的目的就是为了降低变量复杂度，减少噪音对算法和模型的影响，提高自变量和因变量之间的相关度，提高模型鲁棒性。数据分桶的内容包括连续变量离散化和多状态离散变量合并为少状态。数据分箱具有很多优势：

- 处理缺失值：若特征存在缺失值，则可以把 null 作为单独一个分箱。
- 处理异常值：对于数据中的 outlier，可以把他通过分箱进行离散化处理，从而提高鲁棒性。
- 可解释性：x，y 之间常常存在非线性关系，此时可利用 WOE（Weight of Evidence）变换。

分箱有如下基本原则：

- 最小分箱占比不低于 5%。
- 箱内不能全是 y = 0。
- 连续箱单调。

1. 固定宽度分箱

对于跨度较大（横跨多个量级）的特征，可以按照幂级数来分组。如果计数值中有比较大的缺口时，会出现很多空箱子。

2. 分位数分箱

按照数据的分位数对特征数据进行分箱，更符合统计分布。

	除法映射分箱		对数映射分箱		分位数分箱
0	14.0	0	4.0	0	5
1	20.0	1	4.0	1	7
2	12.0	2	4.0	2	4
3	17.0	3	4.0	3	6
4	35.0	4	4.0	4	9
	...		...		..
199995	7.0	199995	3.0	199995	2
199996	6.0	199996	3.0	199996	1
199997	14.0	199997	4.0	199997	5
199998	8.0	199998	3.0	199998	2
199999	8.0	199999	3.0	199999	2

图 4.7 数据分箱结果

4.4 特征交互

想要丰富特征，特别是对于线性模型而言，除了分箱外，另一种方法是添加原始数据的交互特征和多项式特征。例如对于每一个 grade 都对应着 default 的个数，由此可以生成新的特征每个 grade 中 default 的均值：

	grade_target_mean
0	0.386464
1	0.304525
2	0.304525
3	0.059887
4	0.224647
	...
614775	0.386464
614776	0.224647
614777	0.224647
614778	0.059887
614779	0.131377

图 4.8 生成交互特征‘grade_target_mean’

或是对于匿名特征 n*，可以生成每个值对应的 grade 水平：

	grade_to_mean_n0
0	0.689391
1	1.032531
2	1.475766
3	1.106825
4	1.475766
	...
199995	0.737883
199996	0.368942
199997	1.106825
199998	1.475766
199999	0.368942

图 4.9 生成衍生特征‘grade_to_mean_n0’

4.5 特征编码

对于离散型数据，可以借助 LabelEncoder 将其转换成 0 至 n-1 之间的数。

	subGrade
0	21
1	16
2	17
3	3
4	12
	..
614775	21
614776	13
614777	12
614778	3
614779	7

图 4.10 ‘subGrade’ labelencode 编码

而对于 Logistic Regression 等模型还需要额外操作：特征归一化，去除相关性高的特征。归一化使得训练过程收敛速度更快，避免大吃小问题；去除相关性使得模型可解释性提高，并加快了预测过程。

4.6 特征选择

特征选择是特征工程里的一个重要问题，其目标是寻找最优特征子集。特征选择能剔除不相关(irrelevant)或冗余(redundant)的特征，从而达到减少特征个数，提高模型精确度，减少运行时间的目的。另一方面，选取出真正相关的特征简化模型，可以协助理解数据产生的过程。

特征选择方法有：

- Filter
  - 方差选择法
  - 相关系数（Pearson 相关系数）法
  - 卡方检验
  - 互信息法
- Wrapper（RFE）
  - 递归特征消除法
- Embedded
  - 基于惩罚项的特征选择法
  - 基于树模型的特征选择

4.6.1 Filter

Filter 是基于特征间的关系进行筛选：

1. 方差选择法：计算特征方差，然后根据阈值选择方差大于阈值的特征

2. 相关系数法：皮尔森相关系数是一种最简单的可以帮助理解特征和响应变量之间关系的方法，衡量的是变量之间的线性相关性。值域为-1 到 1，+1 表示完全正相关，-1 表示完全负相关，0 变时无线性相关。

3. 卡方检验：用于检验自变量对因变量的相关性。自变量有 N 种取值，因变量有 M 种取值，考虑自变量等于 i 且因变量等于 j 的样本频数的观察值与期望的差距。统计量为：$\chi^2 = \sum(A-T)2T$，其中 A 为实际值，T 为理论值。

4. 互信息法：用于评价自变量对因变量的相关性。在 feature_selection 库的 SelectBest 类结合最大信息系数法可以用于选择特征。

4.6.2 Wrapper

递归特征消除法：使用一个基模型来进行多轮训练，每轮训练之后消除若干权值系数的特征，再基于新特征进行下一轮训练。在 feature_selection 库的 RFE 类可以用于选择特征。

4.6.3 Embedded

1. 基于惩罚项的特征选择法：使用带惩罚项的基模型，除了筛选特征外，同时进行降维。在 feature_selection 库的 SelectFromModel 类结合逻辑回归模型可以用于选择特征。

2. 基于树模型的特征选择法：树模型中的 GBDT 也可以用来作为基模型进行特征选择。在 feature_selection 库的 SelectFromModel 类结合 GBDT 模型可以用于选择特征。

本数据经过非入模特征剔除和缺失值填充后，可以通过计算协方差矩阵来观察特征间的相关性：

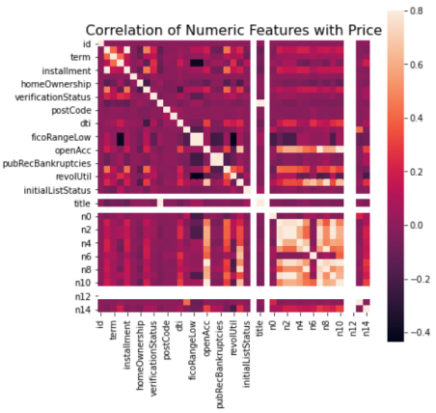

图 4.10 ‘subGrade’ labelencode 编码