IEEE-CIS Fraud Detection

武子越 马万腾 王炳杰

Context

1. 赛题背景与数据描述

2. 探索性数据分析

3. 数据预处理

4. 特征选择与特征工程

5. 建模与结果评估

I. 赛题背景

在电子商务中，海量消费者交易数据具有巨大的商业价值。IEEE计算智能协会的研究员希望能够利用这些数据，改进银行的反欺诈系统，有效识别客户欺诈问题，又能够保证无欺诈客户的体验。

I. 数据描述

数据来源

Vesta的消费者电子商务交易数据，包括消费者的信用卡信息，使用设备，交易时间，交易类型等。

数据构成

训练集：train_transaction.csv (59w×394) train_identity.csv (59w×41)

测试集：test_transaction.csv (50w×393) test_identity.csv (50w×41)

提交数据：sample_submission.csv (50w×2)

数据导入

将transaction数据和identity数据根据共同的id进行join操作

最终训练集和测试集大小：train (59w×434) test (50w×433)

II. 探索性数据分析

数据基本描述

二分类目标分布

- 0 (not fraud): 569877
- 1 (fraud): 20663

缺失值情况检查

- TransactionAmt、ProductCD等20列没有缺失值
- 部分列的缺失值很多，甚至超过了90%
- 不同列之间数据完整性相差较大

column	missing ratio
id_24	99.1962
id_25	99.131
id_07	99.1271
id_08	99.1271
id_21	99.1264
id_26	99.1257
id_22	99.1247
id_23	99.1247
id_27	99.1247
dist2	93.6284
D7	93.4099
id_18	92.3607
D13	89.5093
D14	89.4695
…	
V317	0.002032
V318	0.002032
V319	0.002032
V320	0.002032
V321	0.002032
TransactionID	0
isFraud	0
TransactionDT	0
TransactionAmt	0
ProductCD	0
card6	0
…	

II. 探索性数据分析: 基本描述

特征描述

TransactionDT/Amt：交易时间/金额，连续数值型

ProductCD：类别变量，5个类

card1/2/3/5：card的信息，整数特征编码的类别　card4/6：类别变量，4个类（极其不平衡）

addr1/2：地址，整数特征（无缺失）

dist1/2：距离，整数特征（不平衡，缺失较多）

P_emaildomain：类别变量　R_emaildomain：类别变量（缺失较多）

C1-14：counting，整数特征，有的很不平衡，但是没有缺失

D1-15：timedelta，整数特征，有的很不平衡，有的缺失很多

M1-9：match的特征，比如姓名和地址的联系，类别特征，True/False或很少的类，有的很不平衡，基本上缺失一半

V1-339：已经被Vesta处理过的特征，整数特征，有的很不平衡，有的缺失很多

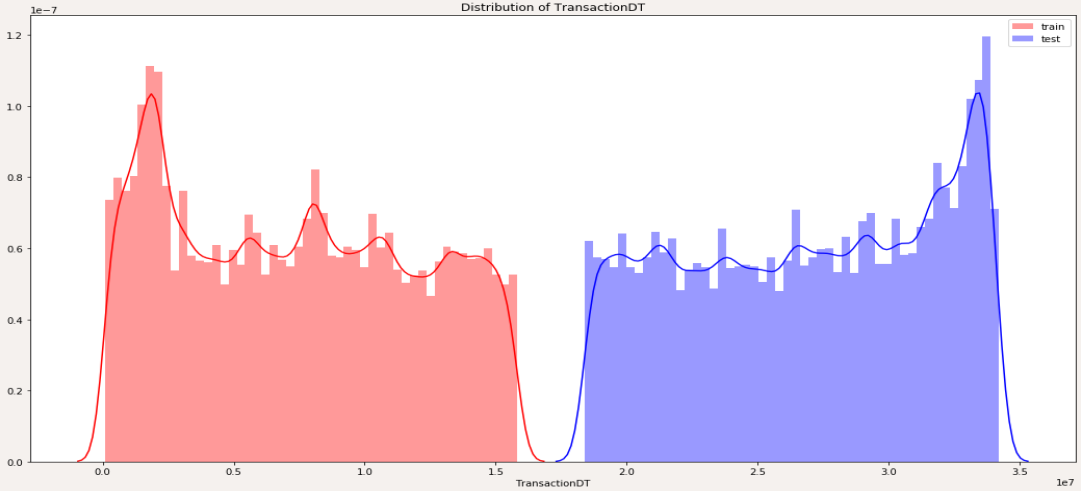
id_01-38：各种类型的特征都有

Device Type/Info：类别特征，代表设备名称

II. 探索性数据分析: 可视化

TransactionDT

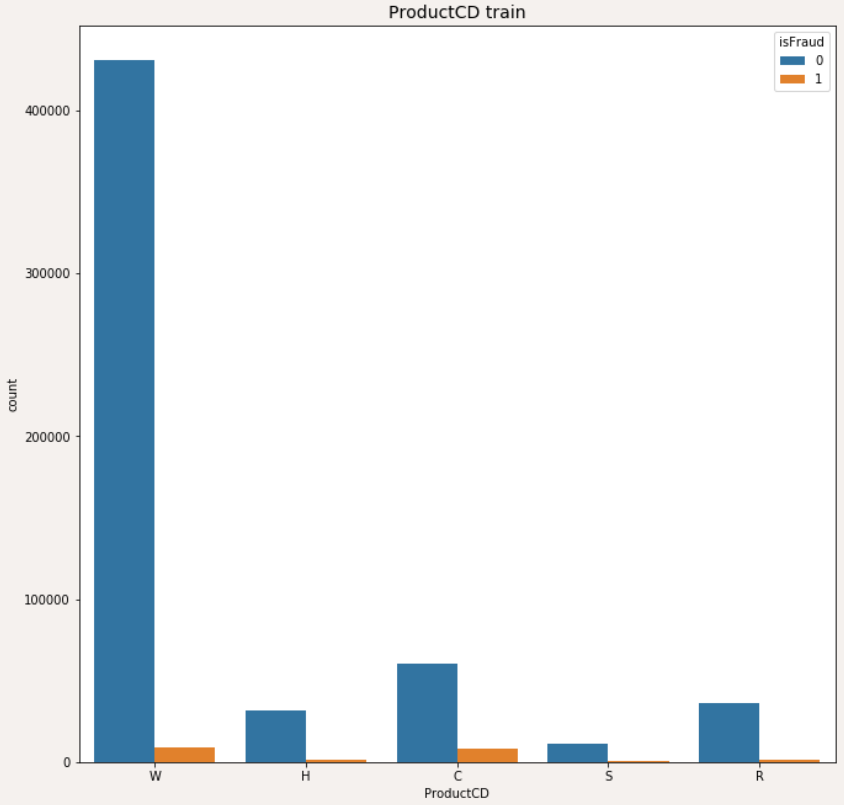
- 每一条交易的时间精确到秒跨度一年多
- 可用来查看训练集和测试集的基本分布
- 通过数据发现训练数据和测试数据有30天的gap



II. 探索性数据分析: 可视化

类别变量：ProductCD

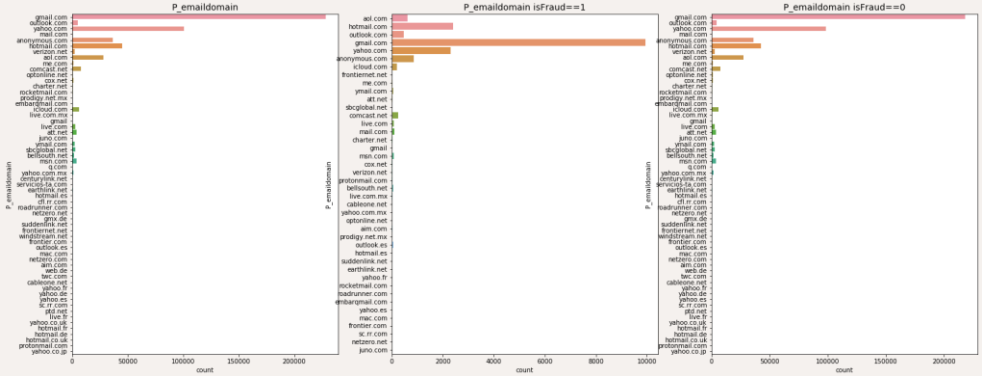
- 五种Product: W/H/C/S/R
- W的产品数量最多
- C的欺诈率从直观上来看比较高
- H/S/R无论是否欺诈总量都比较少



II. 探索性数据分析: 可视化

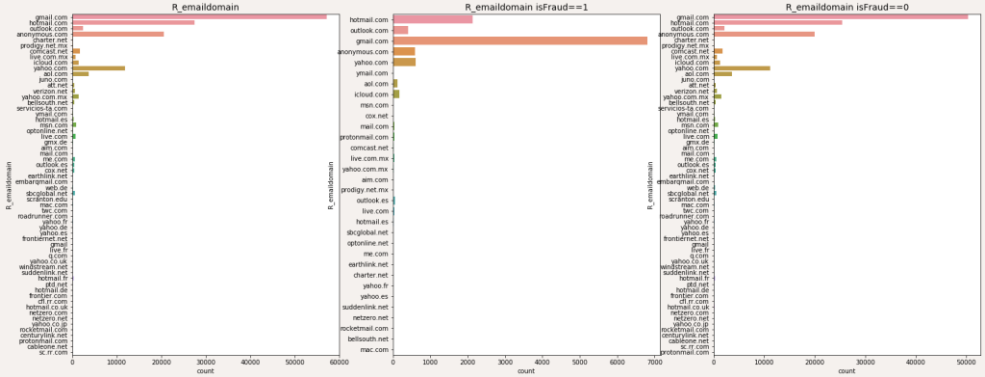
类别变量：P_emaildomain

- 顾客信用卡绑定邮箱的信息
- 可以看到使用gmail邮箱的相对来说欺诈的就比hotmail多等，email是比较有用的信息。



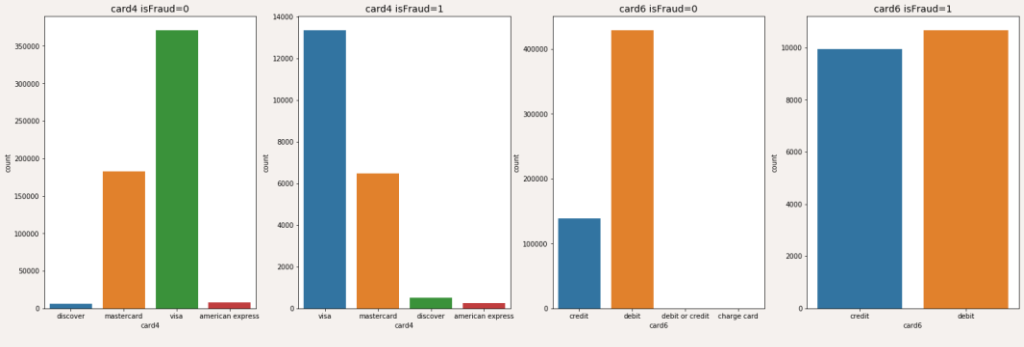
类别变量：R_emaildomain

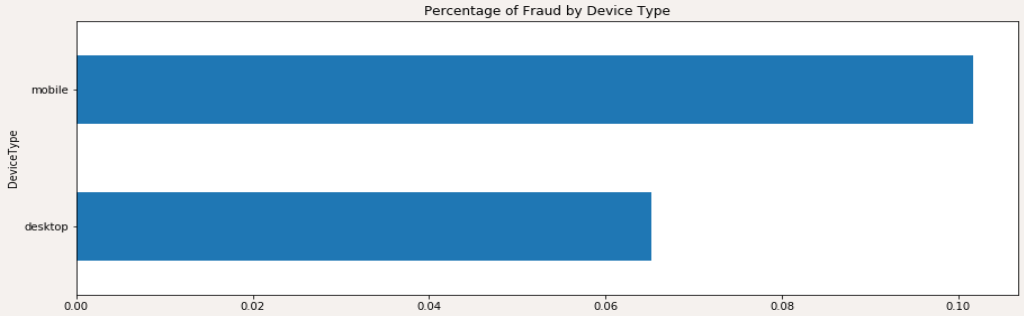
- 同样是邮箱信息，但是有缺失。
- 同样可以看到gmail的欺诈占比普遍比hotmail多。

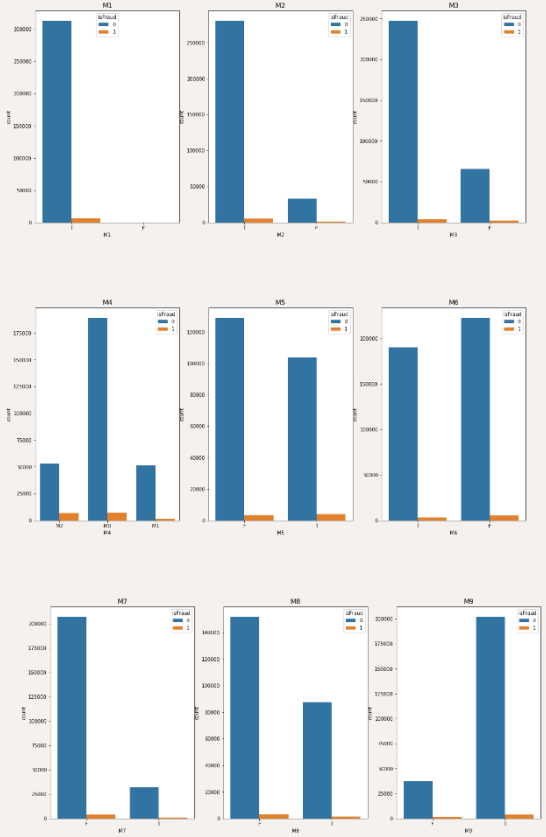


II. 探索性数据分析: 可视化

其他类别变量：Card4/6、M1-9、Device Type等



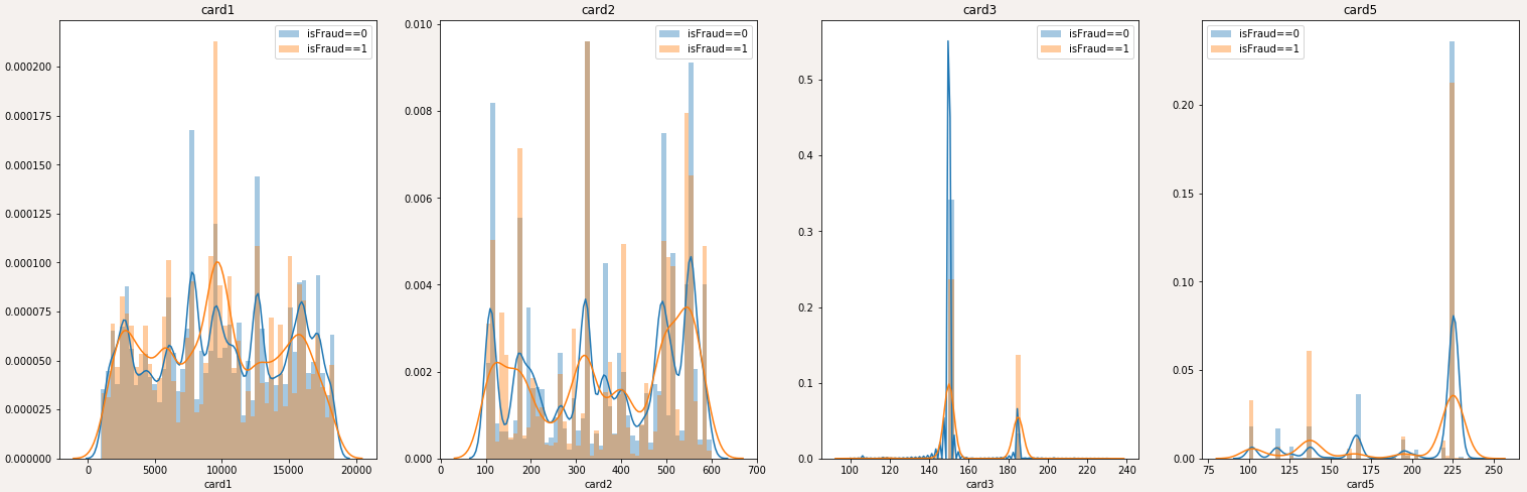




II. 探索性数据分析: 可视化

数值型变量：card1/2/3/5

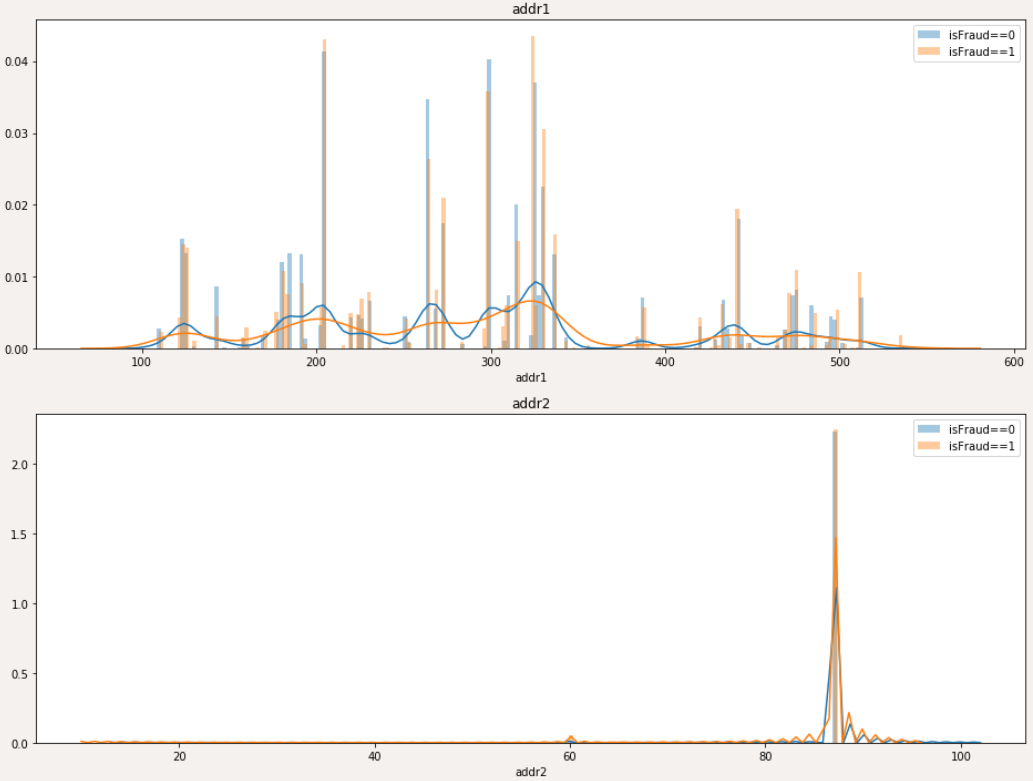
- 顾客信用卡信息
- 不同card数据分布相差较大，当做类别变量处理
- 有的card在某些数据的位置欺诈非常集中



II. 探索性数据分析: 可视化

数值型变量：addr1/2

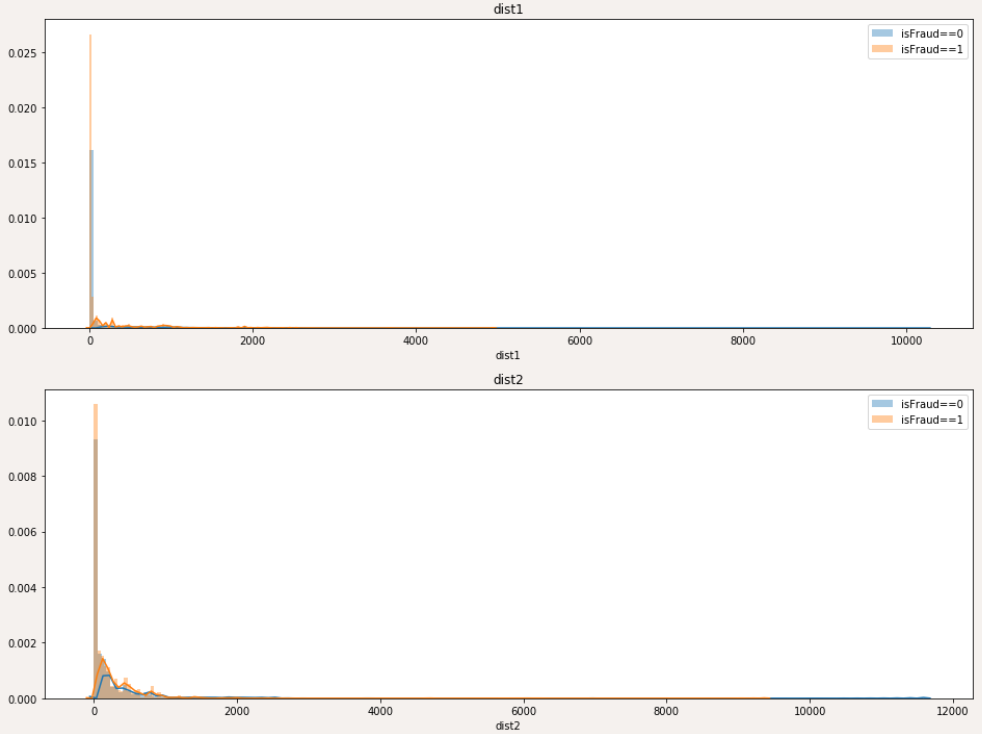
- 根据名称判断可能和地址有关
- Addr1分布较为分散，主要集中在几个数值，也可以当做类别变量处理
- Addr2分布非常奇怪，主要集中在某个数值附近



II. 探索性数据分析: 可视化

数值型变量：dist1/2

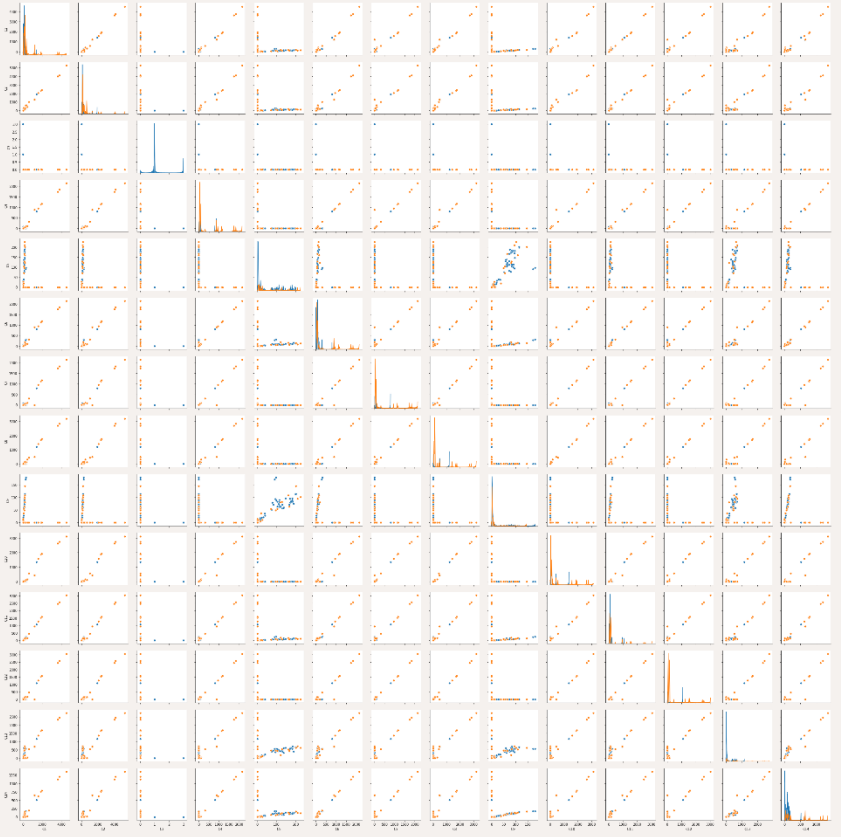
- 绝大多数为零，有少数的非零变量
- 分布很不均衡
- 可以考虑把零和非零各作为一列来处理



II. 探索性数据分析: 可视化

数值型变量：C1-14

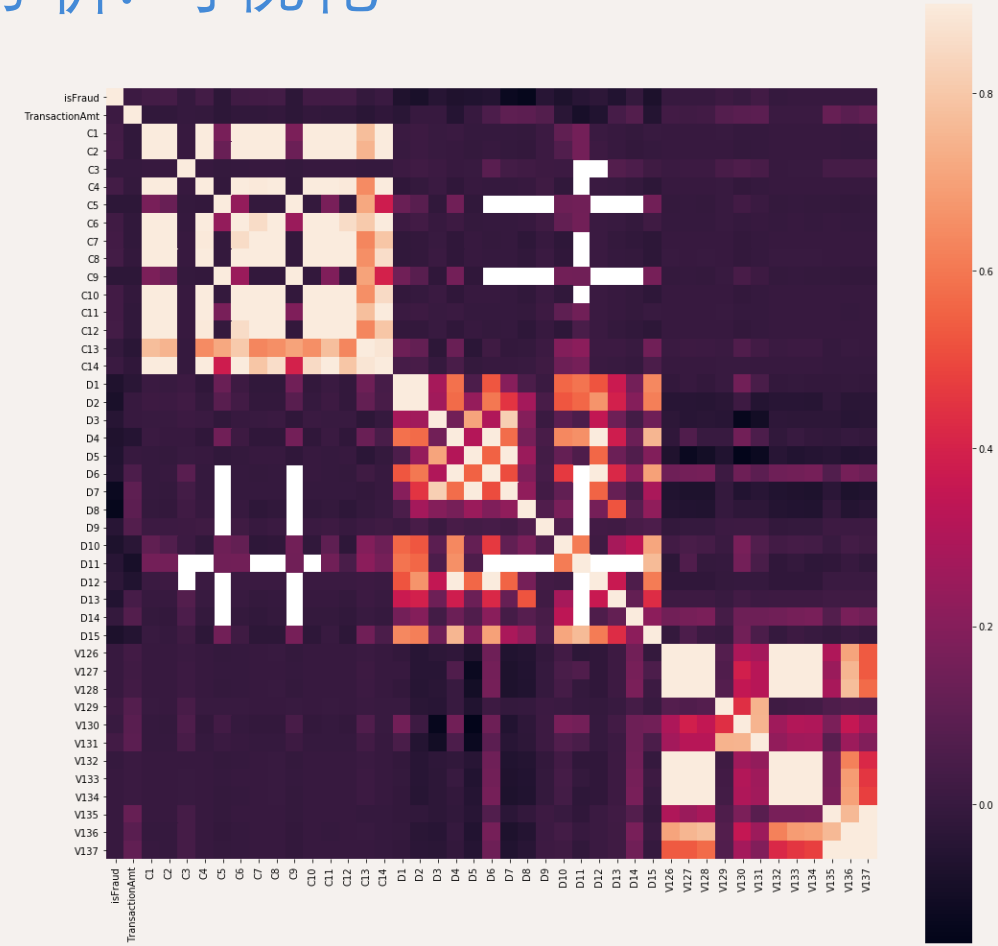
- 分布的差别较大，而且不同类之间可能存在相关性
- 可以进行主成分分析，降维等操作



II. 探索性数据分析: 可视化

变量间的相关性

- 同一类特征之间相关性较大，比如C、D、M开头的特征
- 一些特征的相关性接近1，应该是相同信息的不同形式表达
- 部分特征和目标isFraud存在一定的相关性



III. 数据预处理

缺失值和异常值处理

- 从434个特征中选择了160余个作为有用的特征
- 删去交易数据中的异常值
- 根据列的实际意义用进行缺失值填充
  - 交易数据等用平均值填充
  - 不平衡的类别特征用众数或数量最多的特征
- 训练集和测试集的处理相同

```
useful_cols=['isFraud','TransactionAmt','ProductCD','card1','card2','card3','card4','card5','card
             'addr1','addr2','dist1','P_emaildomain','R_emaildomain','C1','C2','C3','C4','C5','C6'
             'C7','C8','C9','C10','C11','C12','C13','C14','V95','V96','V97','V98','V99','V100','V1
             'V102','V103','V104','V105','V106','V107','V108','V109','V110','V111','V112','V113',''
             'V115','V116','V117','V118','V119','V120','V121','V122','V123','V124','V125','V126',''
             'V128','V129','V130','V131','V132','V133','V134','V135','V136','V137','V279','V280',''
             'V282','V283','V284','V285','V286','V287','V288','V289','V290','V291','V292','V293',''
             'V295','V296','V297','V298','V299','V300','V301','V302','V303','V304','V305','V306',''
             'V308','V309','V310','V311','V312','V313','V314','V315','V316','V317','V318','V319',''
             'DeviceType','DeviceInfo']
```

```
def fill_na_mean(data):
    col=['addr1','card2','card3','card5','V95','V96','V97','V98','V99','V100','V101',
             'V102','V103','V104','V105','V106','V107','V108','V109','V110','V111','V112','V113',''
             'V115','V116','V117','V118','V119','V120','V121','V122','V123','V124','V125','V126',''
             'V128','V129','V130','V131','V132','V133','V134','V135','V136','V137','V279','V280',''
             'V282','V283','V284','V285','V286','V287','V288','V289','V290','V291','V292','V293',''
             'V295','V296','V297','V298','V299','V300','V301','V302','V303','V304','V305','V306',''
             'V308','V309','V310','V311','V312','V313','V314','V315','V316','V317','V318','V319',''
    for c in col:
        data[c].fillna(data[c].mean(),inplace=True)

def fill_na_mean_test(data):
    col=['C1','C2','C3','C4','C5','C6','C7','C8','C9','C10','C11','C12','C13','C14']
    for c in col:
        data[c].fillna(data[c].mean(),inplace=True)
    # C has missing values in test set only
def fill_na_mode(data):
    col=['addr2','dist1']
    for c in col:
        data[c].fillna(data[c].mode()[0],inplace=True)
```

```
fill_na_mean(train)
fill_na_mean(test)
fill_na_mean_test(test)
fill_na_mode(train)
fill_na_mode(test)
```

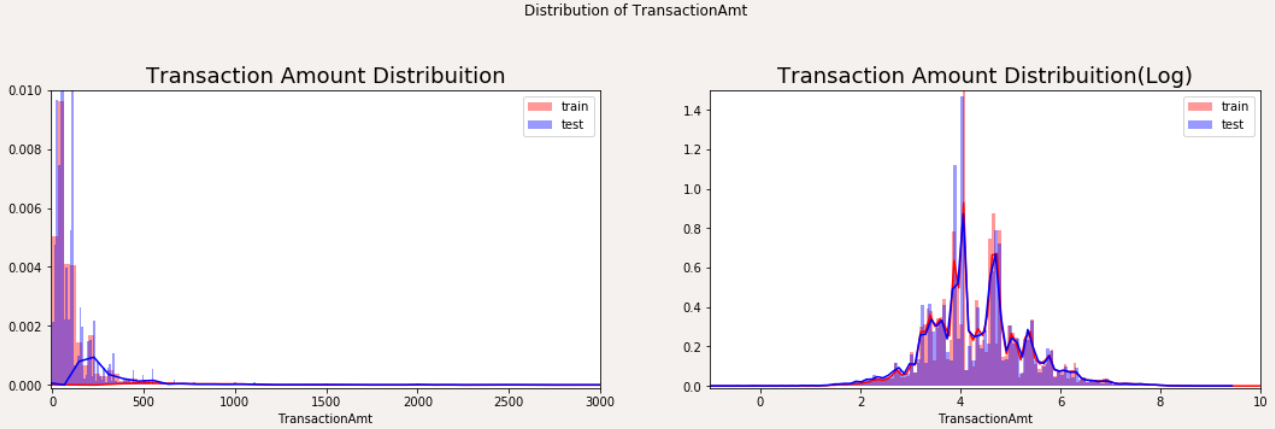
III. 数据预处理

对数变换与分箱

- 将TransactionAmt进行对数转换
  - 实际模型中对数变换效果较好
- 将TransactionAmt进行分箱处理
  - 指定分箱数量，如20个箱
  - 指定分箱中数据个数，如每个箱中一万条数据
  - 实际模型中使用原始连续型数据效果较好

```
def data_transform(data):
    log_trans_col=['TransactionAmt']
    for c in log_trans_col:
        data[c]=np.log(data[c]+1)

data_transform(train)
data_transform(test)
```



III. 数据预处理

One-hot变换与数据降维

- 类别数据进行one-hot编码转化为0-1向量
  - ProductCD、card、DeviceType等分类变量进行进行LabelEncoder转化为整数变量
  - 使用OnehotEncoder转化为0-1向量
- 数据利用PCA进行降维
- 最终训练数据由434个特征减少为125个

```
# label encoding
# convert the categorical features to integer code
for col in test.columns:
    if train[col].dtype=='object' or test[col].dtype=='object':
        lb=LabelEncoder()
        lb.fit(list(train[col].values)+list(test[col].values))
        train[col]=lb.transform(list(train[col].values))
        test[col]=lb.transform(list(test[col].values))
print(f'Train: {train.shape[0]} rows {train.shape[1]} columns.')
print(f'Test: {test.shape[0]} rows {test.shape[1]} columns.')
```

```
Train: 590540 rows 434 columns.
Test: 506691 rows 433 columns.
```

```
train=One_hot(train)
test=One_hot(test)
print(f'Train: {train.shape[0]} rows {train.shape[1]} columns.')
print(f'Test: {test.shape[0]} rows {test.shape[1]} columns.')
```

```
<class 'pandas.core.frame.DataFrame'>
<class 'pandas.core.frame.DataFrame'>
Train: 590540 rows 125 columns.
Test: 506691 rows 124 columns.
```

IV. 特征工程

难点

- 特征较多，特征的筛选非常繁琐复杂
- 由于隐私问题，列名重新进行编码，导致列的实际意义不明确

特征工程的尝试

- 标准化，MinMax区间缩放
- 特征叠加
- 构造多项式特征，如数值特征取二次项，或者两者的交叉乘积项
- 二值化，设定阈值转化为0-1变量

IV. 特征工程

columns	operation
device_info	(percentage<10%)->(class='others')
P_emaildomain	(data==NaN)->(class='others')
V95,V96,V97	(colA,colB)->(colA+colB)
V297	(col)->(col^2)
C1,C2	(colA,colB)->(colA+colB)
C5,C6	(colA,colB)->(colA+colB)
C1,…,C14	mean(C1,…,C14)
addr1,addr2	(data==NaN)->(class='others')
C1,…C14	max(C1,…,C14),min(C1,…,C14)
V301,V302,V305	sum(columns)
card1-6	sum(group)
V95,…,V137	mean(V95,…,V137)
V279,…,V320	mean(V279,…,V320)

V. 模型建立与评估

采用模型

- Decision Tree
- Random Forest
- KNN
- SVM Classifier
- Logistic Regression
- Gradient Boosting Tree
- MLP
- XGBoost
- LightGBM

```
import sklearn.metrics as metric
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestClassifier
from sklearn.neighbors import KNeighborsClassifier
from sklearn.linear_model import LogisticRegression
from sklearn.svm import LinearSVC
from sklearn.tree import DecisionTreeClassifier
from sklearn.ensemble import GradientBoostingClassifier
from sklearn.neural_network import MLPClassifier
from sklearn.preprocessing import LabelEncoder
from sklearn.preprocessing import OneHotEncoder
import xgboost as xgb
import lightgbm as lgb
```

V. 模型建立与评估

评估指标：利用混淆矩阵

准确率：预测无欺诈中真正无欺诈的比例

召回率：真正无欺诈中被判为无欺诈的比例

F1-score：综合考虑准确率和召回率的指标

		Actual	
		Positive	Negative
Predicted	Positive	TP	FP
	Negative	FN	TN

- **查准率（percision）** $P = \frac{TP}{TP + FP}$
- **查全率（recall）** $R = \frac{TP}{TP + FN}$
- **F1分数** $F1 = \frac{2PR}{P + R}$

V. 模型建立与评估

评估指标：利用混淆矩阵

识别欺诈率

$$C = \frac{TN}{TN + FP}$$

		Actual	
		Positive	Negative
Predicted	Positive	TP	FP
	Negative	FN	TN

- 对于该反欺诈问题在实际意义上的评价指标
- 意义为真实的反例中有多少被查出来
- 对于降低银行资金风险，规范消费者行为有重要的意义

V. 模型建立与评估

Model	Confusion Matrix		P	R	C	F1-score	Time(s)
RF	142452	132	0.99907423	0.97253456	0.8862069	0.98562577	86.9
	4023	1028					
KNN	141658	926	0.99350558	0.9758144	0.62449311	0.98458053	1646.5
	3511	1540					
Logistic	142527	57	0.99960024	0.96617339	0.51694915	0.98260261	101.2
	4990	61					
SVM	142418	166	0.99883577	0.96744786	0.60941176	0.9828913	226.1
	4792	259					
Tree	142229	355	0.99751024	0.9758355	0.81157113	0.98655383	6.9
	3522	1529					
GBDT	142342	242	0.99830275	0.97499863	0.85270846	0.98651308	290.9
	3650	1401					
MLP	140823	1761	0.98764939	0.96792219	0.17902098	0.97768629	373.1
	4667	384					
XGB	142354	230	0.99838692	0.98383474	0.92182189	0.99105741	1444.1
	2339	2712					
LGBM	142354	230	0.99838692	0.98588564	0.92907801	0.9920969	280
	2038	3013					

V. 模型建立与评估

结论总结

- **大数据量提高建模的准确性**
  - 50w条数据能够快速发现特征之间的相关关系
  - 400+个特征能够保证预测一定的准确性
- **特征工程是提高精度的最有效方式**
  - 根据数据分布特征进行预处理
  - 根据特征的实际意义考虑不同的特征筛选、组合方式
  - 特征没有明确意义情况下需要反复尝试
- **模型选择需要根据实际情况考虑**
  - 反欺诈问题上，主流采用的还是决策树及一系列基于树的延伸模型
  - 实验证明树模型的效果平均意义上确实要好
- **评价模型准则需要根据数据集特点和实际需求决定**
  - 二分类且不平衡目标问题中，单一使用准确率不再合适

Thank you