- 官方总结帖
  - https://www.kaggle.com/c/ieee-fraud-detection/discussion/111284
- 1st solution
  - https://www.kaggle.com/c/ieee-fraud-detection/discussion/111284
- 1st kernel
  - https://www.kaggle.com/whitebird/ieee-internal-blend
  - https://www.kaggle.com/super13579/find-client-by-d1-and-card-leak-use-on-submit
- 1st EDA
  - https://www.kaggle.com/cdeotte/eda-for-columns-v-and-id
- 问题
  - 预测一个用户欺诈的概率，而不是一笔交易……

```
df.groupby（user）[pred].transform(mean/median...))
```

- 用户ID构建
  - D1、D2结合card、addr构建用户id
    - 其他：账号、email、退款、账单地址
    - 具体：
      - card1、addr1、D1
  - 定义好id后lgb单模型A榜0.96+……
- lag信息、去匿名化?
- 特征选择方法
  - 单特征筛选
    - 如果我的一个特征在训练集上训练完之后在验证集上的预测结果的auc低于0.5，那么这个特征很大概率是一个噪音特征。
  - Permutation importantce特征筛选

        - 我训练好了一个模型，我们对验证集中的某一个特征列进行shuffle，如果对shuffle之后的特征进行预测的准确性没有变化甚至变好了，那么这个特征可能意义不大，可以毫不犹豫的进行删除,如果预测结果变差了，那么该特征是非常重要的，不可以删除。
- 1st
    - 完整kernel
        - https://www.kaggle.com/cdeotte/xgb-fraud-with-magic-0-9600
    - EDA
        - https://www.kaggle.com/cdeotte/eda-for-columns-v-and-id
        - https://www.kaggle.com/c/ieee-fraud-detection/discussion/111284
        - https://www.kaggle.com/c/ieee-fraud-detection/discussion/111308
    - Magic work
        - 构造UID
            - Ref: https://www.kaggle.com/c/ieee-fraud-detection/discussion/111510
            - 优点:
                - 使模型更好预测诈骗

- 寻找出UID后，使用部分UID的交易数据进行训练，对未加入训练的UID的交易进行预测，更好的检测模型
- 方便进行一些后续操作，如分组聚合特征的创建

- 实现过程
  - 混合train和test，记得加个feature用于区分混合后的数据中的训练集和测试集
  - 计算当前混合数据集的特征重要性，筛选出较重要的用于组合成UID
  - 根据UID进行分组，并对重要特征生成新的聚合特征，注意观察聚合后特

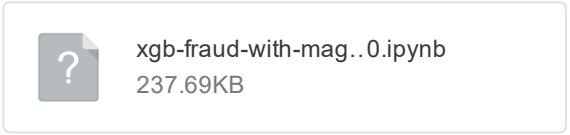
征的方差或者能否把该组进行进一步划分

- 2
- 计算特征之间的相关系数，留下尽可能多的不相关的特征用于聚合

- 创建聚合性的组特征
  - 原理：https://www.kaggle.com/c/ieee-fraud-detection/discussion/111453
  - 将数据集按UID分组后，计算每个分组的某特征值的平均值，作为新的聚合特征，在放入决策树进行划分，效果可能会更好！
  - 创建分组后的aggregated特征后，记得删除UID（原分组标准的特征）

- 构建ID特征
  - TransactionDay = TransactionDT / (24*60*60))
  - D1n = TransactionDay minus D1
  - D3n = TransactionDay minus D3
- 过拟合

        - 不直接用UID
        - 使用所有的C、M列的均值作为UID
            - new_features = df.groupby('uid')[CM_columns].agg(['mean'])
    - 模型选择
        - Catboost did well on all groups
- XGB - best for known
- LGBM - best for unknown
- 数据
    - Catboost (0.963915 public / 0.940826 private)
    - LGBM (0.961748 / 0.938359)
    - XGB (0.960205 / 0.932369)
- kenel
    - https://www.kaggle.com/kernels/scriptcontent/21610581/download



- 寻找UID
    - https://www.kaggle.com/c/ieee-fraud-detection/discussion/111510
- EDA
    - 前150个特征
        - https://www.kaggle.com/alijs1/ieee-transaction-columns-reference
    - 剩余300个特征

        - https://www.kaggle.com/cdeotte/eda-for-columns-v-and-id
    - 对于V系列的特征分组后，因为数据具有NAN结构，有3种处理方式
        - 对每个V组单独进行PCA
        - 从每个组中选择最大不相关的子集
        - 使用整个组的列平均值替换整个组
    - Feature Selection
        - forward feature selection (using single or groups of features)
        - recursive feature elimination (using single or groups of features)
        - permutation importance
        - adversarial validation
        - correlation analysis
        - time consistency
            - train a single model using a single feature (or small group of features) on the first month of train dataset and predict isFraud for the last month of train dataset.

- 用于观察一个特征是否与时间有一致性关系
- 结论：95%的特征与时间相关，5%的特征在前一个月有某种pattern，在最后一个月不存在数据

  - client consistency
  - train/test distribution analysis

- 验证策略
  - 为验证特征与时间的关系，不像一般实验随机按比例划分训练、测试集
  - 按交易的月份分组，训练前5个月，跳1个月，测试最后1个月（类似可以改变占比）
  - 最后的结论
    - XGB model did best predicting known UIDs with AUC = 0.99723
    - LGBM model did best predicting unknown UIDs with AUC = 0.92117
    - CAT model did best predicting questionable UIDs with AUC = 0.98834

        - 其中questionable 是指预测结果没有强烈欺诈倾向的UID，根据不同模型对不同情况的UID预测结果，进行融合，效果好于单个model
- XGBoost一些过程
    - 混合训练集+测试集，删除train的isFraud标签
    - 使用公式“D15n = Transaction_Day - D15 and Transaction_Day = TransactionDT/(24*60*60)”，正则化D列，可以把D列原表示相对于过去时间点的增量，转换成准确的过去时间点，这样，D就和时间无关了
    - Encoding Functions
        - (1) encode_FE does frequency encoding where it combines train and test first and then encodes.
        - (2) encode_LE is a label encoded for categorical features
        - (3) encode_AG makes aggregated features such as aggregated mean and std
        - (4) encode_CB combines two columns
        - (5) encode_AG2 makes aggregated features where it counts how many unique values of one feature is within a group.
    - Feature Engineering

        - 当有想法增加新特征时，要把结果在本地计算AUC值，如果有提升，就保留，否则，舍弃
    - Feature Selection
        - 计算每个特征与之间的一致性，为控制变量，每个特征都应训练一个model，然后使用前几个月的数据训练，预测后几个月的情况，保留AUC值在0.5以上的
        - 删除了一些大部分为NAN的特征
    - Local Validation
        - 在训练集中按比例（3:1）划分训练集和测试集
        - model参数

```
clf = xgb.XGBClassifier(
n_estimators=2000,
max_depth=12,
learning_rate=0.02,
subsample=0.8,
colsample_bytree=0.4,
missing=-1,
eval_metric='auc',
# USE CPU
 #nthread=4,
 #tree_method='hist'
 # USE GPU
 tree_method='gpu_hist'
 )
```

        - catboost参数

```
############################ Model params
cat_params = {
 'n_estimators':5000,
 'learning_rate': 0.07,
 'eval_metric':'AUC',
```

```
 'loss_function':'Logloss',
 'random_seed':SEED,
 'metric_period':500,
 'od_wait':500,
  'task_type':'GPU',
  'depth': 8,
  #'colsample_bylevel':0.7,
  }
estimator = CatBoostClassifier(**cat_params)
estimator.fit(
  X.iloc[trn_idx,:],y[trn_idx],
  eval_set=(X.iloc[val_idx,:], y[val_idx]),
  cat_features=categorical_features,
  use_best_model=True,
  verbose=True)
```

- 可对训练结果查看特征重要性，“clf.feature_importances_”
- 使用GroupKFold进行test预测

```
if BUILD95:
  oof = np.zeros(len(X_train))
  preds = np.zeros(len(X_test))

  skf = GroupKFold(n_splits=6)
  for i, (idxT, idxV) in enumerate( skf.split(X_train, y_train, groups=X_train['DT_M']) ):
  month = X_train.iloc[idxV]['DT_M'].iloc[0]
  print('Fold',i,'withholding month',month)
  print(' rows of train =',len(idxT),'rows of holdout =',len(idxV))
  clf = xgb.XGBClassifier(
  n_estimators=5000,
  max_depth=12,
  learning_rate=0.02,
  subsample=0.8,
  colsample_bytree=0.4,
  missing=-1,
  eval_metric='auc',
  # USE CPU
  #nthread=4,
  #tree_method='hist'
```

```
# USE GPU
tree_method='gpu_hist'
)
h = clf.fit(X_train[cols].iloc[idxT], y_train.iloc[idxT],
eval_set=[(X_train[cols].iloc[idxV],y_train.iloc[idxV])],
verbose=100, early_stopping_rounds=200)

oof[idxV] += clf.predict_proba(X_train[cols].iloc[idxV])[:,1]
preds += clf.predict_proba(X_test[cols])[:,1]/skf.n_splits
del h, clf
x=gc.collect()
print('#'*20)
print ('XGB95 OOF CV=',roc_auc_score(y_train,oof))
```

- 使用UID
- 模型融合

- 总结
  - 赛题内容一定要分析好
  - 对于使用xgboost、catboost等决策树模型，具有一般决策树的特点——每次按某个特征进行划分数据集，是划分后的数据集的熵尽可能小，所以使用这些模型进行训练的时候，数据不需要做one-hot处理，模型会自动计算，或者在dataframe对象中用astype指定为类别特征，如：

```
df['P_emaildomain'] = df['P_emaildomain'].astype('category')
```

  - 对于训练集和测试集的划分，除了随机按比划分，还可以考虑使用如时间进行月份分组的方法（如果数据提供了时间信息，但预测内容又和时间无关，这个是很好的思路）
  -