# Statistical Learning Final Report
# 信用卡消費類別推薦

109024514劉之榆、109024701林承慶

January 4, 2022

## 玉山AI公開挑戰賽:聰明消費來預3-信用卡消費類別推薦

**Goal** Predict top 3 highest-amount expense categories next month of each consumer. The target 16 categories and 50000 customer ids are specified.

**Scoring** NDCG@3

# Outline

1. EDA
2. Preprocessing
3. Modeling
4. Inference

# Variables

Original data: 53 variables, 32553986 observations, 7.26 GB.

1. 消費行爲:
   - dt: 月份
   - txn_amt: 刷卡金額
   - shop_tag: 消費類別
   - txn_cnt: 刷卡筆數(負值代表有刷退次數)
   - domestic(overseas)_online(offline)_cnt(/amt_pct): 國內外線上下刷卡筆數/金額占比
   - card_k_txn_cnt(/amt_pct):卡片k的筆數/金額占比, k=1∼15
2. 個人資訊:
   - chid: 客户編號
   - educd: 教育程度
   - trdtp: 行業別
   - gender_code: 性別
   - age: 年齡

# Missing data

Missing values: all are personal infos, and we classified them as 3 types:

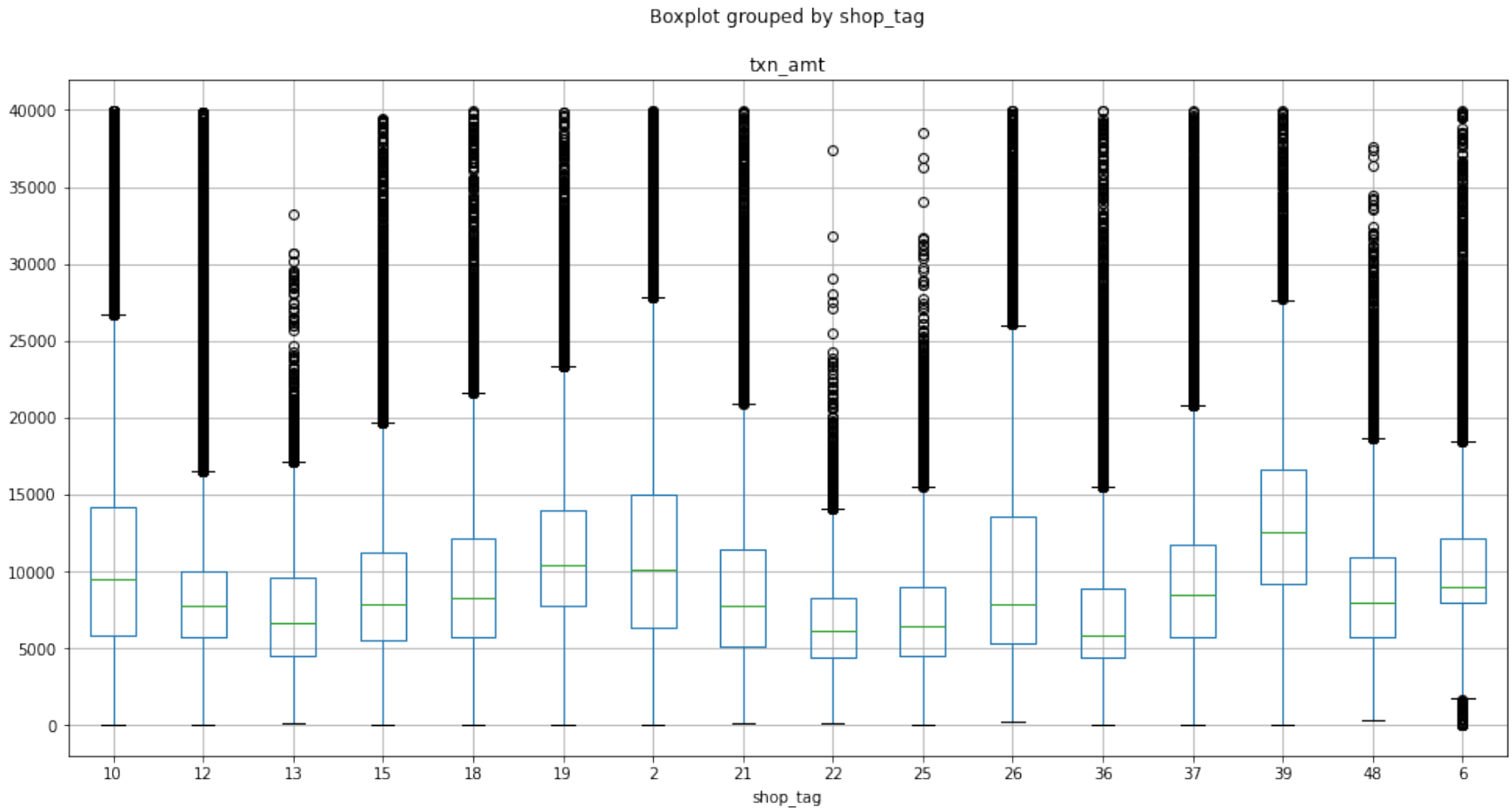
- **Stable variables:** fill with values from former month, e.g. gender_code, educd
- **Meaningful missing values:** create new level, e.g. trdtp
- **Missing at complete random:** fill with median/mode

# Thoughts

Our thought is to predict next month's txn_amt of each shop_tag, which means we will have 16 models at one time, and for each customer.
By trying to build a baseline prediction, we simply predicted the $last$ month's top 3 categories by using the top 3 categories of the nearest 24/12/6/3 months' data, and get a highest score by previous 12 months' data. Plus, we are not capable to handle such big data for now, so we used only the 11-24 months' data.

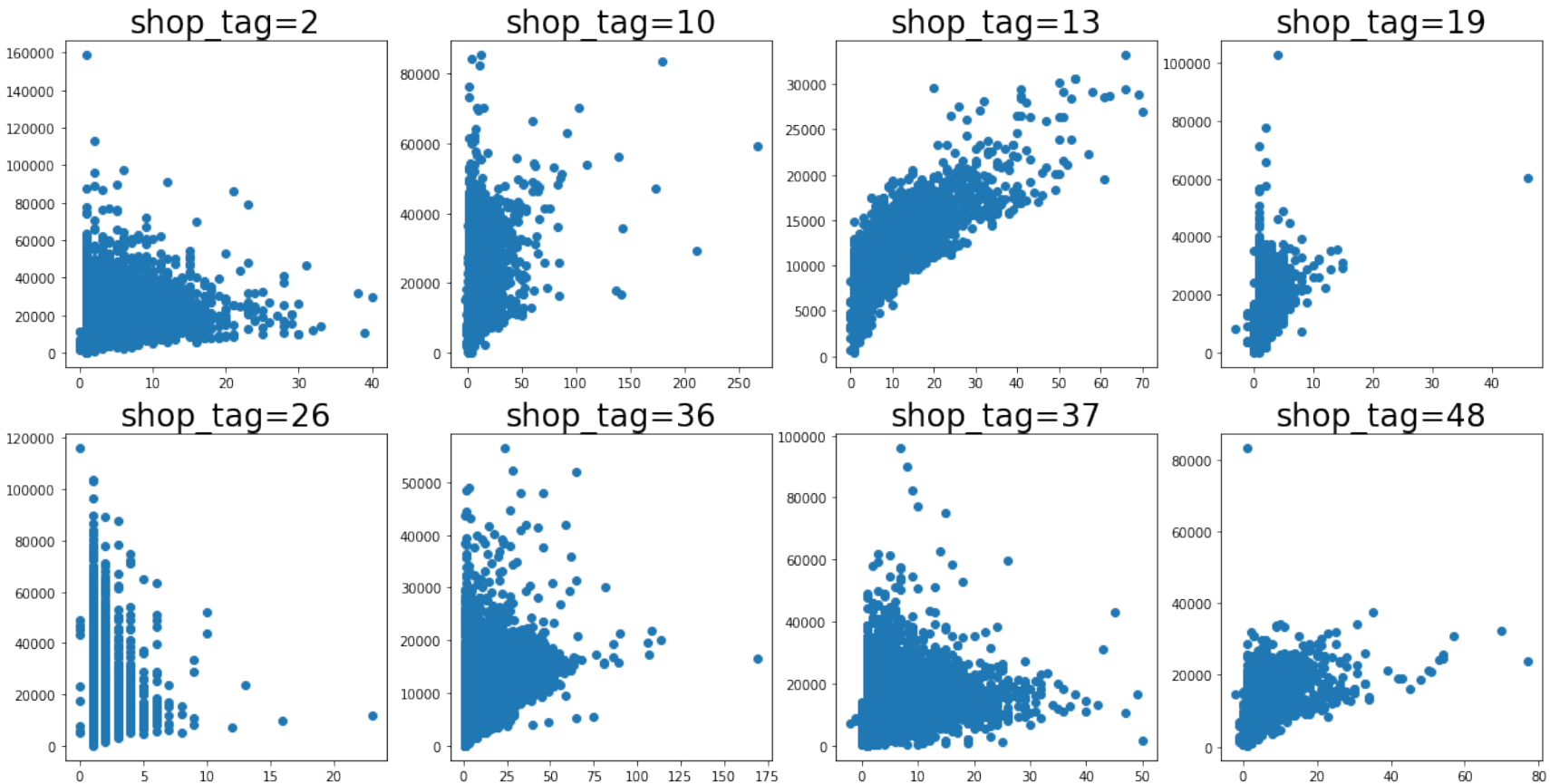
# amount & category

txn_amt(金額) distributed differently with shop_tag(消費類別).



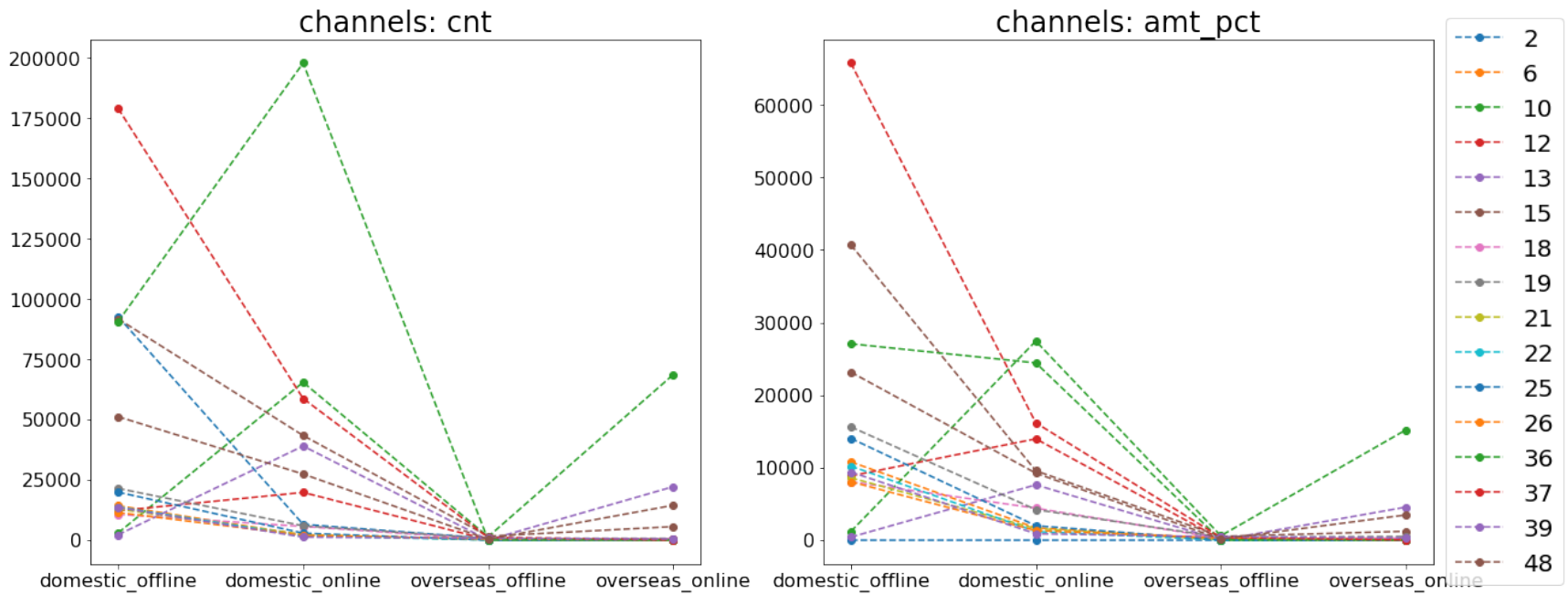
# amount & count

txn_amt & txn_cnt(刷卡筆數): correlated in some shop_tag



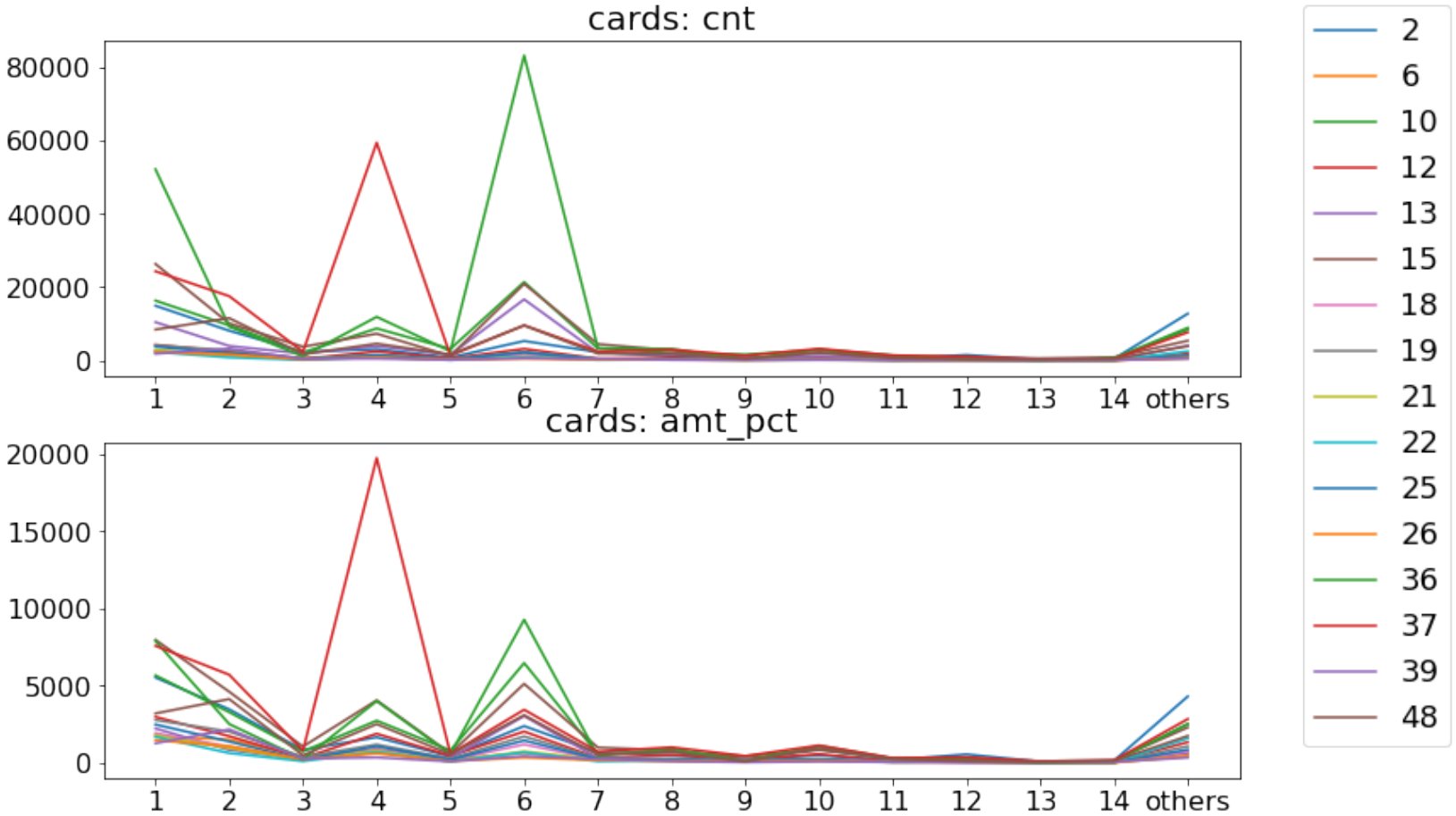
# sales channel

## domestic(/overseas)_online(/offline)_cnt(/amt_ pct)




- overseas_offline(國外線下) is the least
- seems to be a trade-off between online & offline
- the patterns of cnt(筆數) & amt_pct(金額占比) are different

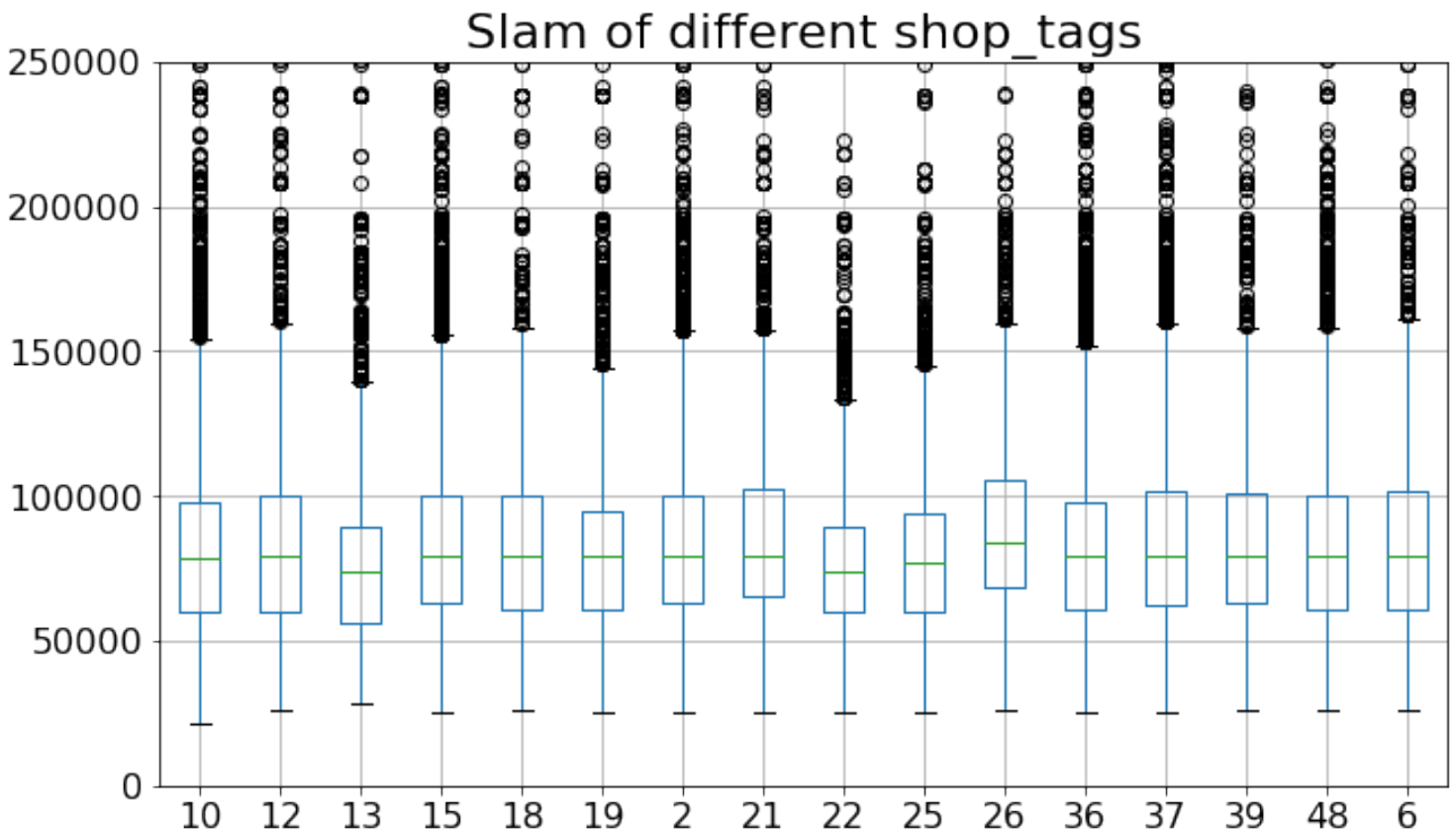
# different cards




- Card 1, 2, 4, 6 dominate: keep them and combine all others
- amt_pct has similar pattern with txn_cnt: drop amt_pct

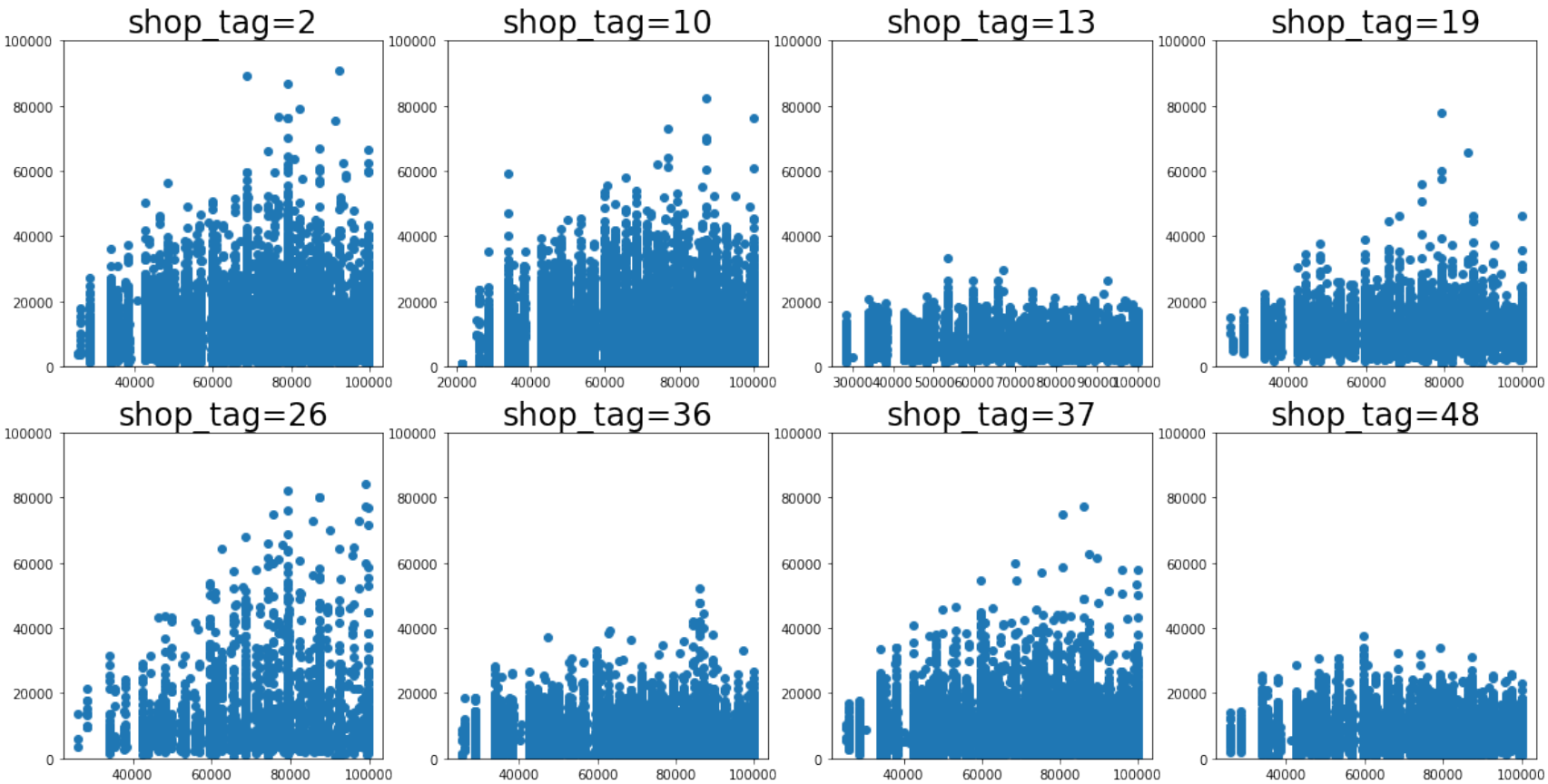
# credits

slam(信用額度) show no difference under different shop_tag.



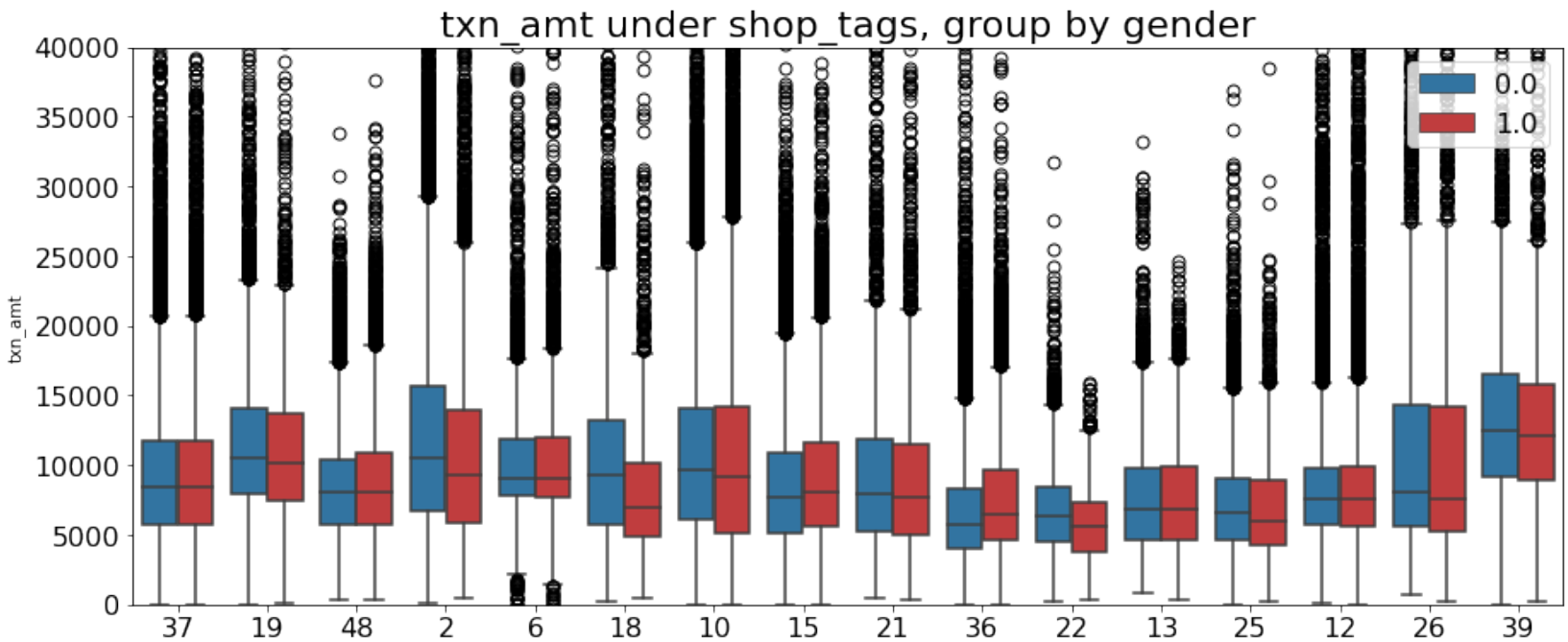
# credits

txn_amt & slam: not correlated under different shop_tag



# gender

txn_amt distributed differt under different gender_code.



# Feature engineering

- **Selecting variables:** 23 left
- **Categorical variables encoding**
  - dummy variables: masts, naty
  - frequence encoding: trdtp(行業), curog(客户來源)
- **Standardization**
- **Box-Cox transformation on amt**
- **New variables**
  - next_txn_amt
  - base_ans: top 3(1,2,3) or not(0)
  - count: times being top 3
  - OtherLift

# Association Rule

## Apriori algorithm:

- (支持度) Support $= P(X \cap Y)$, frequency.
- (置信度) Confidence$_{X \to Y} = P(Y|X)$, conditional probability.
- (提升度) Lift $_{X \to Y} = P(Y|X)/P(Y)$, association of X & Y.

| antecedents | consequents | antecedent support | consequent support | support | confidence | lift |
|---|---|---|---|---|---|---|
| 29 | other | 0.000010 | 0.344371 | 0.000008 | 0.880000 | 2.555381 |
| 14 | 15 | 0.008782 | 0.301077 | 0.005557 | 0.632738 | 2.101579 |

New feature OtherLift: lift of each shop_tag; or, for each tag Y,
OtherLift $=$ Lift$_{X \to Y} \times$ txn_amt$_X$

# Cutting data

- Training set: for each shop_tag, use all previous data from dt $= 11 \sim 22$ to predict the last month
- Validation set: use the $23^{th}$ month to predict the $24^{th}$ month
- Testing set: $25^{th}$ month (unknown)

# Models & Performance

For each type of method, 16 models are built, and for each customer, 16 prediction will be made. Then, we sort them and get top 3 highest amount of every shop_tags. The highest score is by randomforest.

- **Linear regression**: 0.661986
- **LASSO**: 0.381479
- **Ridge**: 0.353888
- **Elastic net**: 0.381479
- **PLS**: 0.661882
- **Randomforest**: 0.681385 (top 13%)
- **XGBoost**: 0.669545

# Models & Performance

For each type of method, 16 models are built, and for each customer, 16 prediction will be made. Then, we sort them and get top 3 highest amount of every shop_tags. The highest score is by randomforest.

- **Linear regression**: 0.661986
- **LASSO**: 0.381479
- **Ridge**: 0.353888
- **Elastic net**: 0.381479
- **PLS**: 0.661882
- **Randomforest**: 0.681385 (top 13%)
- **XGBoost**: 0.669545

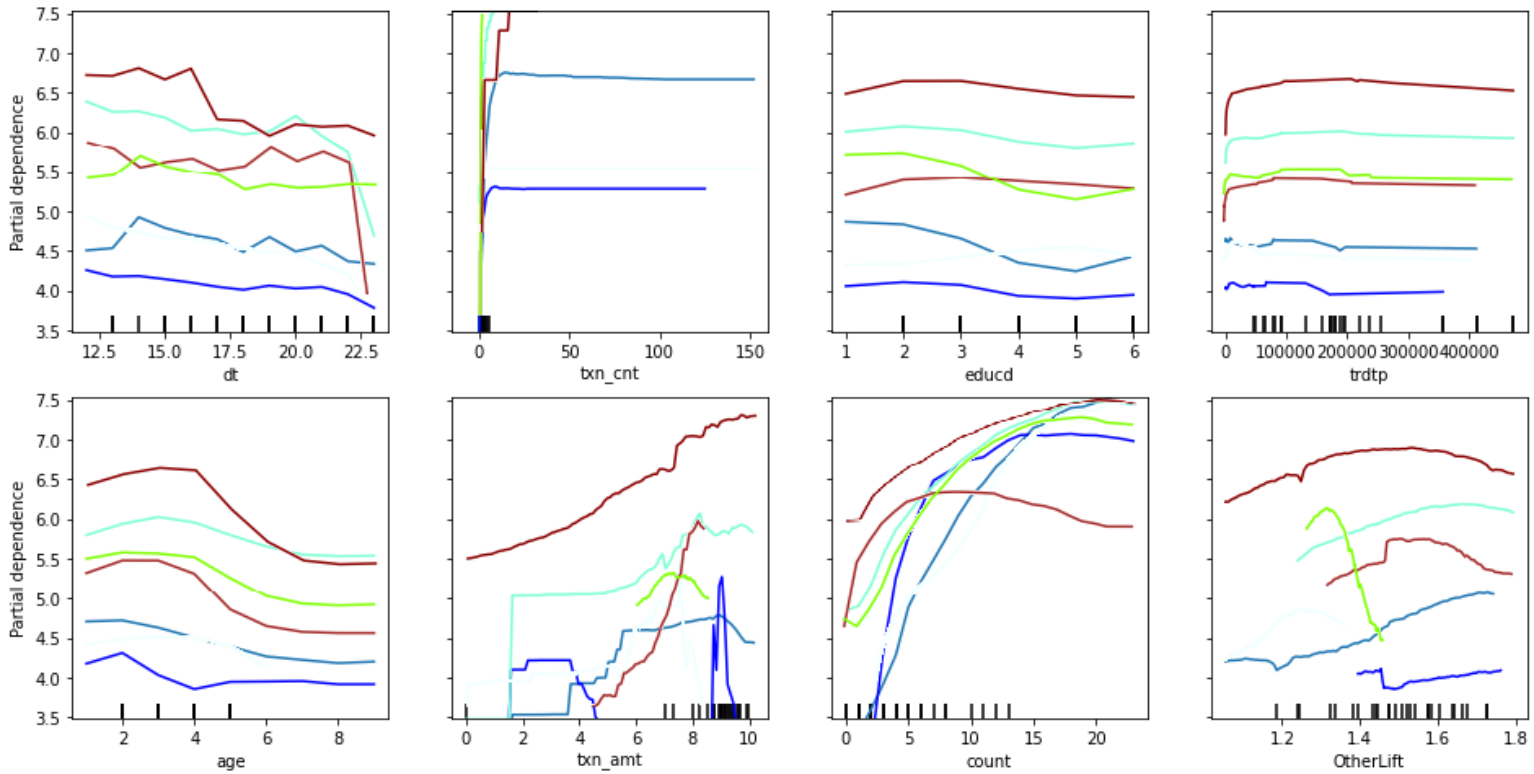
Our baseline score is 0.688609 (top 9%).

# Importance

| | 2 | 6 | 10 | 12 | 13 | 15 | 18 | 19 | 21 | 22 | 25 | 26 | 36 | 37 | 39 | 48 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| dt | 0.097 | 0.068 | 0.088 | 0.089 | 0.104 | 0.091 | 0.104 | 0.104 | 0.099 | 0.101 | 0.105 | 0.099 | 0.077 | 0.082 | 0.102 | 0.093 |
| txn_cnt | 0.052 | 0.055 | 0.069 | 0.051 | 0.103 | 0.087 | 0.039 | 0.027 | 0.040 | 0.042 | 0.042 | 0.055 | 0.113 | 0.077 | 0.033 | 0.075 |
| domestic_offline_cnt | 0.032 | 0.024 | 0.003 | 0.011 | 0.000 | 0.038 | 0.015 | 0.009 | 0.021 | 0.022 | 0.024 | 0.029 | 0.032 | 0.029 | 0.019 | 0.024 |
| domestic_online_cnt | 0.005 | 0.015 | 0.024 | 0.014 | 0.063 | 0.015 | 0.010 | 0.009 | 0.006 | 0.006 | 0.006 | 0.010 | 0.050 | 0.017 | 0.005 | 0.011 |
| overseas_online_cnt | 0.000 | 0.000 | 0.037 | 0.001 | 0.028 | 0.003 | 0.002 | 0.002 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.001 | 0.009 |
| domestic_offline_amt_pct | 0.011 | 0.016 | 0.004 | 0.014 | 0.000 | 0.013 | 0.011 | 0.012 | 0.016 | 0.010 | 0.011 | 0.016 | 0.014 | 0.011 | 0.016 | 0.015 |
| domestic_online_amt_pct | 0.007 | 0.014 | 0.019 | 0.014 | 0.035 | 0.014 | 0.011 | 0.014 | 0.009 | 0.005 | 0.007 | 0.009 | 0.024 | 0.012 | 0.009 | 0.014 |
| overseas_online_amt_pct | 0.001 | 0.001 | 0.029 | 0.001 | 0.016 | 0.003 | 0.002 | 0.003 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.001 | 0.009 |
| card_1_txn_cnt | 0.015 | 0.011 | 0.011 | 0.010 | 0.012 | 0.023 | 0.012 | 0.011 | 0.016 | 0.008 | 0.014 | 0.014 | 0.013 | 0.009 | 0.011 | 0.011 |
| card_2_txn_cnt | 0.006 | 0.006 | 0.006 | 0.006 | 0.005 | 0.006 | 0.006 | 0.007 | 0.007 | 0.004 | 0.006 | 0.010 | 0.005 | 0.006 | 0.011 | 0.011 |
| card_4_txn_cnt | 0.006 | 0.005 | 0.006 | 0.007 | 0.005 | 0.006 | 0.007 | 0.005 | 0.007 | 0.004 | 0.005 | 0.006 | 0.008 | 0.024 | 0.005 | 0.007 |
| educd | 0.044 | 0.035 | 0.044 | 0.046 | 0.036 | 0.040 | 0.054 | 0.051 | 0.057 | 0.051 | 0.048 | 0.053 | 0.047 | 0.052 | 0.057 | 0.045 |
| trdtp | 0.091 | 0.056 | 0.083 | 0.082 | 0.065 | 0.082 | 0.094 | 0.097 | 0.097 | 0.093 | 0.097 | 0.081 | 0.081 | 0.083 | 0.082 | 0.081 |
| gender_code | 0.018 | 0.013 | 0.017 | 0.015 | 0.014 | 0.015 | 0.014 | 0.011 | 0.015 | 0.009 | 0.012 | 0.019 | 0.012 | 0.017 | 0.015 | 0.018 |
| age | 0.053 | 0.034 | 0.043 | 0.050 | 0.030 | 0.047 | 0.052 | 0.052 | 0.058 | 0.046 | 0.041 | 0.055 | 0.042 | 0.053 | 0.061 | 0.046 |
| card_others_txn_cnt | 0.020 | 0.013 | 0.023 | 0.012 | 0.029 | 0.027 | 0.014 | 0.013 | 0.013 | 0.024 | 0.014 | 0.015 | 0.030 | 0.011 | 0.015 | 0.023 |
| txn_amt | 0.218 | 0.256 | 0.220 | 0.261 | 0.241 | 0.204 | 0.236 | 0.237 | 0.242 | 0.249 | 0.251 | 0.222 | 0.172 | 0.231 | 0.233 | 0.171 |
| count | 0.119 | 0.248 | 0.099 | 0.140 | 0.069 | 0.081 | 0.107 | 0.105 | 0.070 | 0.106 | 0.083 | 0.113 | 0.091 | 0.111 | 0.105 | 0.139 |
| base_ans | 0.012 | 0.010 | 0.010 | 0.010 | 0.007 | 0.012 | 0.011 | 0.012 | 0.009 | 0.008 | 0.010 | 0.010 | 0.009 | 0.007 | 0.012 | 0.012 |
| OtherLift | 0.194 | 0.121 | 0.166 | 0.167 | 0.138 | 0.193 | 0.199 | 0.219 | 0.219 | 0.209 | 0.225 | 0.184 | 0.180 | 0.167 | 0.207 | 0.186 |

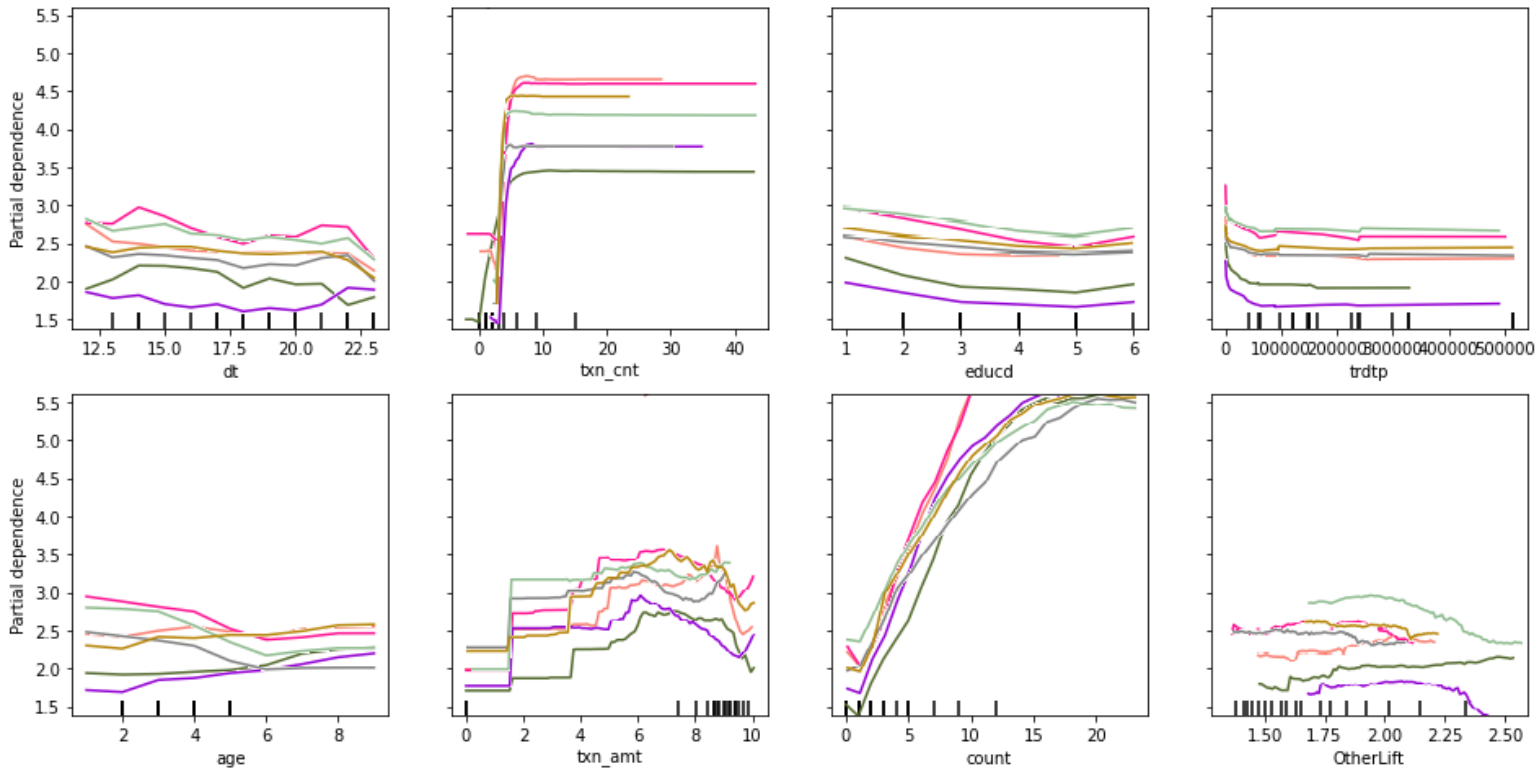
# Partial dependence plot

Tags including 2, 6, 10, 12, 13, 15, 36, 37, 48 have higher predicted value on average, and their dependence plot are similar.



# Partial dependence plot

Tags including 18, 19, 21, 22, 25, 26, 39 have smaller predicted value on average.



# Reasoning

- txn_amt: 若前一月消費金額小於7單位，則下月消費和其成正比；若大於7單位，則下月消費會隨之增加而減少。
- count: 愈常是top3的類別，下月金額愈高。
- OtherLift: 取決於不同的類別。
- dt: 若最後一月消費是在前1 ∼ 2個月，則下一月的消費金額會較少；若在3個月之前則差異不大。
- age: 取決於不同的類別，大致有隨年紀增加而增加(21,26,39,..)和減少兩類(2,10,12,13,15,48,..)。
- txn_cnt: 隨著上月消費的次數增加而增加(即使有刷退紀錄也是退愈多次下月金額愈高)，當次數到10次之後就持平(較少超過十次的)。

# Division of work

- EDA: 之榆
- Pre-processing:
  1. Missing value: 之榆
  2. Feature engineering: 承慶
- Modeling: 承慶
- Presentation: 之榆