

理賠客戶再購與商品推薦

政大風管碩二 陳奕帆
政大風管四 何恬

台大財金所財工組碩一 周永昱
台大資工二 謝宗儒

大綱

01

研究問題
Research
question

02

探索性資料分析
EDA

03

資料預處理
Data
pre-processing

04

模型訓練
Model
Training

05

附錄
Appendix

01

研究問題

Research Question

3

Research Question	EDA	Data pre-processing	Model Training	Appendix
-------------------	-----	---------------------	----------------	----------

4

02

探索性資料分析
EDA

- A. 理賠檔EDA
- B. 再購檔EDA
- C. 客戶屬性檔EDA

5

Research Question	EDA	Data pre-processing	Model Training	Appendix
A. 理賠檔EDA				
<ul style="list-style-type: none"> 共234428筆資料，13個feature 滿期金受益人RK有69%的Missing Value，生故保險金受益人RK有75%的Missing Value。此兩者應不適合做填補，但可用來產生更多feature，如：是否具滿期金受益人、是否具滿期金受益人、任一受益人是否為被保人...等。 理賠案件型態人數極度失衡，可以注意各類的再購率是否有明顯差異。尤其是當被保人死亡或重病後，是否影響再購行為(此處須注意再購定義，如以被保人-被保人合併，那死亡件100%不會有再購行為，可能要結合客戶關係檔，如被保人的一等親作為合併條件) 有97.96%的案件被保人等於事故人，其餘可能是家庭保單，因此取一位被保人當代表，而代表人並非事故人，因此產生被保人不等於是故人情況，因此理賠再購合併時應注意此種情況，避免漏掉再購。 事故人、要保人、被保人之間關係應仔細考慮，可搭配客戶關係檔做更多Feature Engineering。 				

6

Research Question	EDA	Data pre-processing	Model Training	Appendix
C. 客戶屬性檔EDA				

03

資料預處理

Data pre-processing

- A. 理賠檔、再購檔、客戶屬性檔合併
- B. 合併檔案分析
- C. Deal with Miss Value
- D. Feature Engineering
- E. Categorical Variable Encoding
- F. Feature Scaling
- G. Deal with Imbalanced Data

9

Research Question	EDA	Data pre-processing	Model Training	Appendix
A. 理賠檔、再購檔、客戶屬性檔合併				
<ol style="list-style-type: none"> 1. 資料由要保人對要保人的方式進行合併 2. 並將理賠後120天內再購、180天內再購及360天內再購與否設為新變數，以便了解客戶再購情形 3. 本組也將合併後的資料作分析及解讀 				

10

Research Question

EDA

Data pre-processing

Model Training

Appendix

B. 合併檔案分析

不同理賠案件型態的再購情形

理賠案件型態	筆數	佔比	120內再購	180內再購	360內再購
身故給付	3841	1.64%	7.37%	7.86%	9.19%
完全失能	224	0.10%	7.59%	8.48%	9.82%
部分失能	160	0.07%	17.50%	19.38%	20.63%
重大疾病	6483	2.77%	3.44%	4.52%	7.11%
疾病醫療	132549	56.54%	4.30%	5.76%	9.67%
意外醫療	91171	38.89%	5.27%	7.22%	12.61%
	234428	100%			

11

11

Research Question	EDA	Data pre-processing	Model Training	Appendix
B. 合併檔案分析				
<p>利用Scheffé法事後比較：</p> <p>不同理賠案件型態的再購比例是否有顯著差異</p> <ul style="list-style-type: none"> 不同理賠案件型態120天內的再購情形： 部分失能 > 完全失能 = 身故給付 > 意外醫療 > 疾病醫療 = 重大疾病 不同理賠案件型態在180天內的再購情形： 部分失能 > 完全失能 = 身故給付 = 意外醫療 > 疾病醫療 > 重大疾病 不同理賠案件型態在360天內的再購情形： 部分失能 = 意外醫療 > 完全失能 = 疾病醫療 = 身故給付 > 重大疾病 				

12

Research Question

EDA

Data pre-processing

Model Training

Appendix

B. 合併檔案分析

理賠客戶中不同的疾病類別的再購比例及其再購商品之比例

	有再購比例	再購AHa	再購Ahb	再購Ahc	再購Ahd	再購ILP	再購REG	再購SIN
01.傳染病和寄生蟲病	28.1%	25.0%	10.1%	26.1%	5.5%	4.2%	20.6%	8.4%
02.腫瘤	16.0%	15.3%	6.7%	19.3%	5.4%	8.6%	23.2%	21.5%
03.血液相關及免疫系統的疾患	23.5%	9.5%	5.4%	29.3%	12.8%	5.0%	25.2%	12.8%
04.內分泌營養和代謝疾病	16.8%	25.2%	4.1%	19.9%	2.2%	5.4%	34.1%	9.1%
05.精神和行為疾患	9.7%	15.9%	8.8%	14.8%	3.8%	4.9%	23.6%	28.0%
06.神經系統疾病	14.4%	24.6%	4.0%	17.1%	6.9%	10.9%	24.0%	12.6%
07.眼和附器疾病	17.6%	13.7%	5.0%	21.4%	8.4%	8.6%	26.5%	16.4%
08.耳和乳突疾病	23.0%	18.2%	10.3%	27.1%	7.0%	6.1%	22.9%	8.4%
09.循環系統疾病	15.7%	13.8%	5.5%	24.2%	8.1%	7.6%	23.5%	17.3%
10.呼吸系統疾病	25.5%	26.4%	10.6%	25.1%	5.5%	3.7%	19.8%	8.7%
11.消化系統疾病	20.6%	17.0%	5.3%	22.5%	7.7%	8.9%	25.5%	13.1%
12.皮膚和皮下組織疾病	20.7%	23.0%	6.6%	28.1%	5.0%	5.0%	22.0%	10.4%
13.肌肉骨骼系統和結締組織疾	17.2%	16.3%	4.4%	24.2%	6.5%	7.0%	26.1%	15.5%
14.泌尿生殖系統疾病	19.8%	20.8%	8.3%	22.1%	8.4%	6.3%	23.3%	10.8%
15.妊娠、分娩和產褥期	35.1%	28.3%	18.9%	26.4%	3.0%	3.1%	14.1%	6.1%
17.先天畸形變態和染色體異常	23.9%	34.4%	6.3%	15.6%	9.4%	0.0%	25.0%	9.4%
18.症狀異常所見，不可歸類	21.7%	17.8%	9.8%	25.0%	5.4%	5.3%	23.3%	13.4%
19.損傷中毒和外因的某些其他	19.7%	13.1%	4.8%	45.2%	2.4%	1.2%	22.6%	10.7%
20.疾病和死亡的外因	27.6%	16.0%	6.1%	28.1%	7.8%	6.2%	25.7%	10.1%
21.影響健康狀態與保健機構接	26.2%	17.4%	9.0%	33.3%	5.8%	4.3%	22.8%	7.5%
99.不知道不想猜	27.8%	16.4%	7.0%	27.9%	6.9%	6.2%	25.7%	10.0%

13

Research Question

EDA

Data pre-processing

Model Training

Appendix

C. Deal with Missing Value

刪除具Missing Value的Feature刪除，如年收入、婚姻狀況和總資產等。

將要保人屬性欄位為空值的列刪除。

BundleSubtype2	illness_code	DiagnosisCode_DESC	claim_settle_dt	REIMBURSED_YR_1	
0	5.N疾病醫療	C18	02.腫瘤	2015-03-25	46987
1	5.N疾病醫療	C18	02.腫瘤	2015-08-15	7087
2	5.N疾病醫療	C18	02.腫瘤	2015-08-15	30712
3	5.N疾病醫療	C18	02.腫瘤	2015-08-15	7087
4	5.N疾病醫療	C18	02.腫瘤	2015-08-15	7087
...
210784	6.N意外醫療	Y99.8	99.不知道不想猜	2017-10-15	1050
210785	6.N意外醫療	V23	20.疾病和死亡的外因	2017-02-19	813
210786	6.N意外醫療	Y99.8	99.不知道不想猜	2017-08-23	1627
210787	5.N疾病醫療	D36	02.腫瘤	2017-08-11	546
210788	4.C重大疾病	I25.1	09.循環系統疾病	2017-04-29	16825

整理後匯入資料如右圖→

1/1

14

Research Question	EDA	Data pre-processing	Model Training	Appendix
D. Feature Engineering				
<ol style="list-style-type: none"> 1. 累計理賠金額 2. 現有／曾有AH保單種類數 3. 要保人相同與否 4. 要保人與受益人相同與否 5. 疾病發生部位 				
15				

Research Question	EDA	Data pre-processing	Model Training	Appendix
E. Categorical Variable Encoding				
<ul style="list-style-type: none"> • 問題：模型無法直接處理 Categorical Variable • 處理：匯入資料後，切割出訓練/測試集，再將文字、類別型的資料透過 target encoding 轉為數值，且在許多Feature中有太多累，無法使用one-hot-encoding 				
encoding <pre>[16]: ## 要先分割訓練跟測試，才能target encoding X_train, X_test, y_train, y_test = train_test_split(X_res, y_res, test_size=0.33, random_state=42) enc = TargetEncoder(cols=['BundleSubtype2', 'illness_code', 'DiagnosisCode_DESC', 'WEALTH_LEVEL', 'stick_level2', 'cust_group2']) training_numeric_dataset = enc.fit_transform(X_train, y_train) testing_numeric_dataset = enc.transform(X_test)</pre>				
16				

Research Question	EDA	Data pre-processing	Model Training	Appendix
F. Feature Scaling				
<ul style="list-style-type: none"> 問題：特徵的range差異太大。 處理：採用Min_Max的方法做Feature Scaling。 回饋：智星老師說可能會受outlier影響，建議使用Z-score normalization。 				
<p>min_max</p> <pre>[17]: scaler = MinMaxScaler() scaler.fit(training_numeric_dataset[['REIMBURSED_YR_TW', '累積理賠金額', '累積理賠次數', 'tenure_m', 'recency_m', 'AGE']]) training_numeric_dataset[['REIMBURSED_YR_TW', '累積理賠金額', '累積理賠次數', 'tenure_m', 'recency_m', 'AGE']] = scaler.transform(training_numeric_dataset[['REIMBURSED_YR_TW', '累積理賠金額', '累積理賠次數', 'tenure_m', 'recency_m', 'AGE']]) testing_numeric_dataset[['REIMBURSED_YR_TW', '累積理賠金額', '累積理賠次數', 'tenure_m', 'recency_m', 'AGE']] = scaler.transform(testing_numeric_dataset[['REIMBURSED_YR_TW', '累積理賠金額', '累積理賠次數', 'tenure_m', 'recency_m', 'AGE']])</pre>				
				17

Research Question	EDA	Data pre-processing	Model Training	Appendix
G. Deal with Imbalanced Data				
<ul style="list-style-type: none"> 問題：在合併後的data set中positive的比例約占5% 處理：採用Over sampling 的 SMOTE，讓 positive 和 negative 比例大約調整到1:1。 回饋：南山Mentor建議Under sampling 的方式來抽樣，減少特徵在模型裡被放大失真的可能性。 				
<p>處理資料不平衡</p> <p>不平衡資料的二元分類 2：利用抽樣改善模型品質</p> <p>Oversampling: SMOTE for binary and categorical data in Python</p> <pre>[11]: cate = [0,1,2,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27] # 這些是類別資料，使用smotenc前要先標出來</pre> <pre>[12]: sm = SMOTENC(categorical_features = cate, random_state=0) X = X.drop(columns=['claim_settle_dt', 'INSURED_DOB']) # 找不到處理timestamp的資料，決定刪除 X_res, y_res = sm.fit_resample(X, y)</pre> <pre>[13]: df1 = X_res df1['y'] = y_res</pre> <pre>[14]: df1.to_excel('理賠再購屬性合併balanced_before_encoding.xlsx')</pre>				
				18

04

模型訓練

Model Training

A. Baseline

B. Pipeline

C. Evaluation

D. Visualization

19

Research Question	EDA	Data pre-processing	Model Training	Appendix
A. Baseline				

20

Research Question	EDA	Data pre-processing	Model Training	Appendix
-------------------	-----	---------------------	----------------	----------

B. Pipeline

建立Pipeline模型並自動調參數

```
[1]: from sklearn.tree import DecisionTreeClassifier
from sklearn.pipeline import Pipeline
from sklearn.model_selection import GridSearchCV
pipeline = Pipeline([('clf', DecisionTreeClassifier(criterion='entropy'))])

[2]: ## 需要調參數的部位
parameters = {'clf__max_depth': (20, 100, 500),
              'clf__min_samples_split': (20, 100, 500),
              'clf__min_samples_leaf': (2, 3, 4)}

[3]: grid_search = GridSearchCV(pipeline, parameters, n_jobs=-1, verbose=1, scoring='f1')

[1]: grid_search.fit(X_train, y_train)

[104]: ## 回傳最好的參數
best_parameters = grid_search.best_estimator_.get_params()
for param_name in sorted(parameters.keys()):
    print('{}: {}'.format(param_name, best_parameters[param_name]))

clf__max_depth: 100
clf__min_samples_leaf: 20
clf__min_samples_split: 20

[105]: ## 最好的score
grid_search.best_score_

[105]: 0.9175343327439514
```

21

Research Question	EDA	Data pre-processing	Model Training	Appendix
-------------------	-----	---------------------	----------------	----------

C. Evaluation

混淆矩陣解讀

<https://www.libinx.com/2018/understanding-sklearn-classification-report/>

```
[96]: from sklearn.metrics import classification_report
predictions = grid_search.predict(X_test)
print(classification_report(y_test, predictions))
```

	precision	recall	f1-score	support
0	0.92	0.92	0.92	66056
1	0.92	0.92	0.92	65999
accuracy			0.92	132055
macro avg	0.92	0.92	0.92	132055
weighted avg	0.92	0.92	0.92	132055

```
[107]: print('Training data score: {}'.format(grid_search.score(X_train, y_train)))
print('Training data score: {}'.format(grid_search.score(X_test, y_test)))

Training data score: 0.9362802709885092
Training data score: 0.920046191481926
```

22

Research Question	EDA	Data pre-processing	Model Training	Appendix
D. Visualization				
<p>視覺化</p> <pre>[17]: from sklearn import tree tree.export_graphviz(clf2,out_file="tree.dot",feature_names=X_train.columns,class_names=['neg','pos']) [18]: import pydot (graph,) = pydot.graph_from_dot_file('tree.dot') [19]: graph.write_png('tree.png') []: # export_graphviz(clf, out_file="adspy_temp.dot", feature_names=feature_names, class_names=class_names, filled = True, impurity = False) from sklearn.tree import export_graphviz import graphviz with open("tree.dot") as f: dot_graph = f.read() graphviz.Source(dot_graph) []: # Alternate method using pydotplus, if installed. import pydotplus import os os.environ["PATH"] += os.pathsep + 'C:/Program Files (x86)/Graphviz2.38/bin/' graph = pydotplus.graphviz.graph_from_dot_data(dot_graph) graph.create_png()</pre> <p>Note: 太多節點或太多層無法畫出</p>				

23

Research Question	EDA	Data pre-processing	Model Training	Appendix
D. Visualization				

24

05

附錄

Appendix

A. 分工

B. 備註

C. EDA result

D. Reference

25

Research Question	EDA	Data pre-processing	Model Training	Appendix
A. 分工				

26

Research Question	EDA	Data pre-processing	Model Training	Appendix
B. 備註				
<ul style="list-style-type: none"> Github : https://github.com/teemoteemo0318/nanshan 				
27				

Research Question	EDA	Data pre-processing	Model Training	Appendix
C. EDA result				
28				

Research Question

EDA

Data pre-processing

Model Training

Appendix

C. EDA result(理賠檔)

理賠檔(CLAIM_ACCT_FIN)欄位說明			
	欄位	類型	名稱
1	INJURED_RK	字元	事故人RK
2	Claim_RK	字元	理賠案號
3	Policy_RK	字元	保單號碼
4	BundleSubtype2	字元	理賠案件型態
5	illness_code	字元	疾病代碼
6	illness_desc	字元	疾病名稱
7	DiagnosisCode_DESC	字元	疾病分類名稱
8	claim_settle_dt	日期	理賠結案日期
9	REIMBURSED_YR_TW	數值	理賠金額(歸至該結案年度)
10	INSURED_RK	字元	被保人RK
11	POLICY HOLDER_RK	字元	要保人RK
12	MATURITY_BENEFICIARY_RK	字元	滿期金受益人RK
13	DEATH_BENEFICIARY_RK	字元	身故保險金受益人RK

1. 欄位說明

29

29

Research Question	EDA	Data pre-processing	Model Training	Appendix														
C. EDA result(理賠檔)																		
<pre>: INJURED_RK 0 Claim_RK 0 Policy_RK 0 BundleSubtype2 0 illness_code 0 illness_desc 0 DiagnosisCode_DESC 0 claim_settle_dt 0 REIMBURSED_YR_TW 0 INSURED_RK 0 POLICY HOLDER_RK 0 MATURITY_BENEFICIARY_RK 162475 DEATH_BENEFICIARY_RK 175195 dtype: int64</pre>		 <table border="1"><thead><tr><th>案件型態</th><th>人數 (估計值)</th></tr></thead><tbody><tr><td>1D身故給付</td><td>~5,000</td></tr><tr><td>2T完全失能</td><td>~5,000</td></tr><tr><td>3P部份失能</td><td>~5,000</td></tr><tr><td>4C重大疾病</td><td>~10,000</td></tr><tr><td>5N疾病醫療</td><td>~110,000</td></tr><tr><td>6N意外醫療</td><td>~90,000</td></tr></tbody></table>			案件型態	人數 (估計值)	1D身故給付	~5,000	2T完全失能	~5,000	3P部份失能	~5,000	4C重大疾病	~10,000	5N疾病醫療	~110,000	6N意外醫療	~90,000
案件型態	人數 (估計值)																	
1D身故給付	~5,000																	
2T完全失能	~5,000																	
3P部份失能	~5,000																	
4C重大疾病	~10,000																	
5N疾病醫療	~110,000																	
6N意外醫療	~90,000																	
2. Missing Value		3. 理賠案件型態人數分配																

30


30

Research Question	EDA	Data pre-processing	Model Training	Appendix
C. EDA result(理賠檔)				
	<p>W18 26959</p> <p>Y99.8 22910</p> <p>V23 15598</p> <p>C50 9080</p> <p>Y93.7 8174</p> <p>...</p> <p>S92.0 1</p> <p>S71 1</p> <p>H43 1</p> <p>Q39.1 1</p> <p>I51.7 1</p>	<p>事故人 被保人 要保人重疊情形</p> <pre>[15]: # 事故人=被保人=要保人 數量 df[(df['INSURED_RK']==df['POLICY_HOLDER_RK']) & (df['INJURED_RK']==df['POLICY_HOLDER_RK'])]['Policy_RK'].count() [15]: 167860</pre> <pre>[16]: # 被保人=要保人 df[(df['INSURED_RK']==df['POLICY_HOLDER_RK'])]['Policy_RK'].count() [16]: 172115</pre> <pre>[17]: # 事故人=要保人 df[(df['INJURED_RK']==df['POLICY_HOLDER_RK'])]['Policy_RK'].count() [17]: 168112</pre> <pre>[18]: # 事故人=被保人 df[(df['INJURED_RK']==df['INSURED_RK'])]['Policy_RK'].count() [18]: 229636</pre>		
	4. 各項疾病人數	5. 事故人、被保人、要保人重複情況		

31

Research Question	EDA	Data pre-processing	Model Training	Appendix
C. EDA result				

32

Research Question	EDA	Data pre-processing	Model Training	Appendix
C. Reference				
				33