南山人壽

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

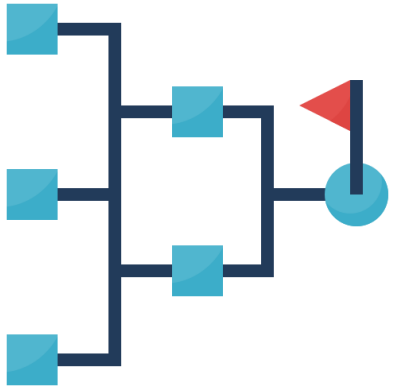
05 系統延伸發想 Future System

01

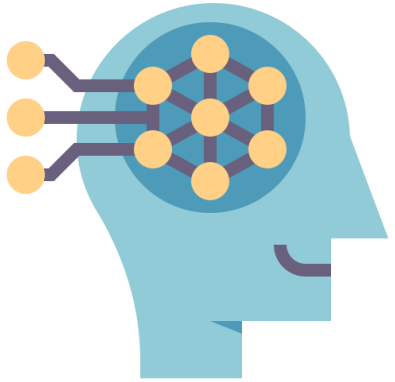
研究問題

Research Question

題目：理賠客戶再購與商品推薦



利用理賠內容、客戶屬性、再購商品等資料透過機器學習的方法來判斷未來理賠客戶是否再購及傾向再購哪類型的商品，以利未來接觸理賠客戶時能夠精準推薦商品，提高再購機會。



解讀出特徵影響再購的關聯和特性，透過結合保險知識和實際數據來做出合理的判斷和解釋，進而做出解釋性高的機器學習模型。

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

A. 理賠檔EDA

- 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事故人情況，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。

B. 再購檔EDA

- 共134472筆資料，11個feature
- 再購檔的資料是Dependent Variable，可視所需來產生對應的Y，例如是否再購、再購什麼...等。
- 產品細項欄位有多項只有一筆資料，是否該刪除此類資料?
- 大多數保單生效日在3~6月，是否有什麼經濟意義?

C. 客戶屬性檔EDA

- 共130487筆資料，28個feature
- 變數有呈現客戶屬性之變數如：現有或曾有哪種類型保單，以及人口統計變數如婚姻狀況、年收入等，亦有綜合各項屬性所組成之變數如忠誠度、客戶分群等。有助於產生更多其他特徵。
- 婚姻狀況及年收入有嚴重缺漏植

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. Deal with Imbalanced Data
F. Categorical Variable Encoding
G. Feature Scaling

A. 理賠檔、再購檔、客戶屬性檔合併

- 程式碼：step1_理賠再購屬性合併.ipynb、step1-2_理賠再購屬性合併.ipynb
- 資料分別由**要保人對要保人、被保人對被保人及其親屬**的方式進行合併

▒再購檔:其實就是你們可以整理成的Y變數的檔案，如果是再購預測模型就是Y=1 OR Y=0; 如果是再購產品預測模型Y="期繳"OR Y="躉繳 OR 其它
▒理賠檔&屬性檔:就是你們要整理X變數的檔案，其中屬性檔是包括所有的要/被保人/事故人(除了少數在分析日已經流失的客戶)
▒再購定義:之前有提過 ，你們可以用理賠檔的事故人(被保人)對映再購檔的被保人、要保人、受益人，或者其它定義; 而再購時間可以取理賠發生後的4個月/6個月/12個月，只要你們提出一個你們覺得適當的看法
▒檔案合併: 對於要用要保人或被保人合併，沒有定見，主要還是看你們就再購的定義，如果是要保人出發就用要保人合併，如果是被保人出發就用被保人合併

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%			

B. 合併檔案分析

利用Scheffé法事後比較:

不同理賠案件型態的再購比例是否有顯著差異

- **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

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

	再購AHa	再購Ahb	再購Ahc	再購Ahd	再購ILP
01.傳染病和寄生蟲病	35.2%	14.3%	36.8%	7.8%	6.0%
02.腫瘤	27.7%	12.1%	34.9%	9.8%	15.5%
03.血液相關及免疫系統的疾患	15.3%	8.7%	47.3%	20.7%	8.0%
04.內分泌營養和代謝疾病	44.4%	7.2%	35.0%	3.9%	9.4%
05.精神和行為疾患	33.0%	18.2%	30.7%	8.0%	10.2%
06.神經系統疾病	38.7%	6.3%	27.0%	10.8%	17.1%
07.眼和附器疾病	24.1%	8.8%	37.4%	14.7%	15.0%
08.耳和乳突疾病	26.5%	15.0%	39.5%	10.2%	8.8%
09.循環系統疾病	23.4%	9.2%	40.9%	13.7%	12.8%
10.呼吸系統疾病	37.0%	14.9%	35.1%	7.8%	5.2%
11.消化系統疾病	27.7%	8.6%	36.6%	12.6%	14.5%
12.皮膚和皮下組織疾病	34.0%	9.8%	41.6%	7.3%	7.3%
13.肌肉骨骼系統和結締組織疾	27.9%	7.5%	41.5%	11.2%	12.0%
14.泌尿生殖系統疾病	31.6%	12.6%	33.5%	12.7%	9.6%
15.妊娠、分娩和產褥期	35.5%	23.7%	33.1%	3.8%	3.9%
17.先天畸形變形和染色體異常	52.4%	9.5%	23.8%	14.3%	0.0%
18.症狀異常所見，不可歸類	28.2%	15.5%	39.5%	8.5%	8.3%
19.損傷中毒和外因的某些其他	19.6%	7.1%	67.9%	3.6%	1.8%
20.疾病和死亡的外因	25.0%	9.5%	43.8%	12.1%	9.6%
21.影響健康狀態與保健機構接	24.9%	12.9%	47.7%	8.4%	6.1%
99.不知道不想猜	25.5%	10.9%	43.3%	10.7%	9.6%

	再購REG	再購SIN
01.傳染病和寄生蟲病	70.9%	29.1%
02.腫瘤	51.9%	48.1%
03.血液相關及免疫系統的疾患	66.3%	33.7%
04.內分泌營養和代謝疾病	78.8%	21.2%
05.精神和行為疾患	45.7%	54.3%
06.神經系統疾病	65.6%	34.4%
07.眼和附器疾病	61.7%	38.3%
08.耳和乳突疾病	73.1%	26.9%
09.循環系統疾病	57.6%	42.4%
10.呼吸系統疾病	69.4%	30.6%
11.消化系統疾病	66.1%	33.9%
12.皮膚和皮下組織疾病	67.9%	32.1%
13.肌肉骨骼系統和結締組織疾	62.7%	37.3%
14.泌尿生殖系統疾病	68.3%	31.7%
15.妊娠、分娩和產褥期	69.9%	30.1%
17.先天畸形變形和染色體異常	72.7%	27.3%
18.症狀異常所見，不可歸類	63.5%	36.5%
19.損傷中毒和外因的某些其他	67.9%	32.1%
20.疾病和死亡的外因	71.8%	28.2%
21.影響健康狀態與保健機構接	75.1%	24.9%
99.不知道不想猜	71.9%	28.1%

欄位依序為:住院醫療、重疾癌症、意外傷害、長期照顧、投資型

期繳保單、躉繳保單

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。
- 將客戶屬性欄位為空值的列刪除。

整理後匯入資料如右圖→

	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

D. Feature Engineering

```
df['被保人是否為事故人'] = np.where(df['INJURED_RK']==df.index.get_level_values(0), 1, 0)
df['結案月份'] = list(map(lambda x:str(x)[5:7],df['claim_settle_dt']))
df['計數'] = 1
df['累積理賠金額'] = df.groupby('Policy_RK')['REIMBURSED_YR_TW'].transform('cumsum')
df['初次理賠時間'] = df.groupby('Policy_RK')['claim_settle_dt'].transform(min)
df['累積理賠次數'] = df.groupby('Policy_RK')['計數'].transform('cumsum')
df = df.drop(columns=['計數'])
df['被保人年收'] = np.where(df['CLIENT_INCOME']>=df['CLIENT_INCOME'].median(), 1, 0) #好像有直接分幾等份的method
df['被保人總資產'] = np.where(df['TOTAL_AUM']>=df['TOTAL_AUM'].median(), 1, 0)
df['具滿期金受益人'] = np.where(df['MATURITY_BENEFICIARY_RK'].isna(), 0, 1)
df['具生故保險金受益人'] = np.where(df['DEATH_BENEFICIARY_RK'].isna(), 0, 1)
```

E. Deal with Imbalanced Data

- 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。

處理資料不平衡

不平衡資料的二元分類 2：利用抽樣改善模型品質

Oversampling: SMOTE for binary and categorical data in Python

[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
```

[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)
```

[13]:
```
df1 = X_res
df1['y'] = y_res
```

[14]:
```
df1.to_excel('理賠再購屬性合併balanced_before_encoding.xlsx')
```

F. Categorical Variable Encoding

- 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中有太多類，無法使用one-hot-encoding

encoding

[16]:
```
## 要先分訓練跟測試，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)
```

G. Feature Scaling

- 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。

min_max

[17]:
```
scaler = MinMaxScaler()
scaler.fit(training_numeric_dataset[['REIMBURSED_YR_TW', '累積理賠金額', '累積理賠次數', 'ternure_m', 'recency_m','AGE']])
training_numeric_dataset[['REIMBURSED_YR_TW', '累積理賠金額', '累積理賠次數', 'ternure_m', 'recency_m','AGE']] = scaler.transform(training_numeric_dataset[['REIMBURSED_YR_TW', '累積理賠金額', '累積理賠次
testing_numeric_dataset[['REIMBURSED_YR_TW', '累積理賠金額', '累積理賠次數', 'ternure_m', 'recency_m','AGE']] = scaler.transform(testing_numeric_dataset[['REIMBURSED_YR_TW', '累積理賠金額', '累積理賠次數
```

04

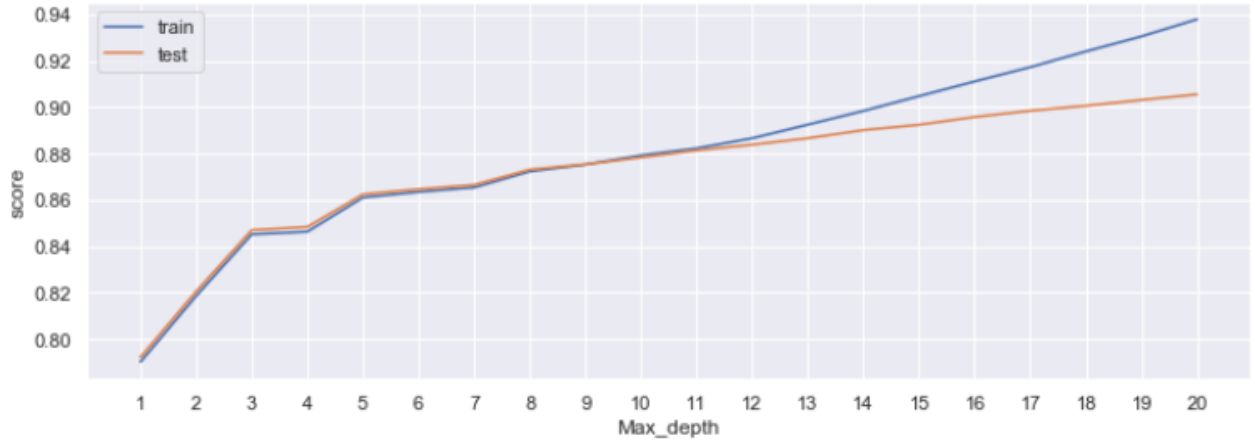
模型訓練

Model Training

A. Baseline
B. DecisionTreeClassifier Training
C. Result
D. Visualization

A. Baseline

- Data : 僅做oversampling及target_encoding、未進行特徵工程
- Model : DecisionTree、僅調整max_depth
- 程式碼：step3_Training-baseline.ipynb



	precision	recall	f1-score	support
0	0.85	0.91	0.88	69432
1	0.91	0.84	0.87	68920
accuracy			0.88	138352
macro avg	0.88	0.88	0.88	138352
weighted avg	0.88	0.88	0.88	138352

```
stick_level20.252 +/- 0.001
recency_m0.231 +/- 0.001
DiagnosisCode_DESC0.229 +/- 0.001
illness_desc0.222 +/- 0.001
cust_group20.220 +/- 0.001
BundleSubtype20.161 +/- 0.001
REIMBURSED_YR_TW0.094 +/- 0.001
WEALTH_LEVEL0.058 +/- 0.001
ternure_m0.056 +/- 0.001
REG_his 0.026 +/- 0.000
GENDER  0.024 +/- 0.000
REG     0.022 +/- 0.000
AHb_his 0.016 +/- 0.000
SIN_his 0.015 +/- 0.000
DIGI_FLG0.011 +/- 0.000
ILP     0.011 +/- 0.000
ILP_his 0.009 +/- 0.000
AHb     0.009 +/- 0.000
AHa     0.008 +/- 0.000
AHd     0.008 +/- 0.000
AHc     0.006 +/- 0.000
SIN     0.004 +/- 0.000
AHd_his 0.003 +/- 0.000
VIP_CLASS0.003 +/- 0.000
TOPCARD 0.001 +/- 0.000
AHc_his 0.001 +/- 0.000
AHa_his 0.000 +/- 0.000
VIP     0.000 +/- 0.000
```

B. DecisionTreeClassifier Training

建立Pipeline模型並自動調參數

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

Ref : 管道模型Pipeline《Python機器學習》

原文網址：https://kknews.cc/code/6bnvre3.html

B. DecisionTreeClassifier Training

```
## 需要調參數的部位
parameters = {'clf__criterion':('entropy','gini'),
              'clf__max_depth':(10,20,30,40,50),
              'clf__min_samples_split':(20,100,500),
              'clf__min_samples_leaf':(2,3,4)}
```

```
grid_search = GridSearchCV(pipeline, parameters, n_jobs=-1, verbose=1, scoring='f1')
```

```
from sklearn.model_selection import cross_val_score
# grid_search.fit(X_train, y_train)
score = cross_val_score(grid_search, X_train, y_train, cv=5)
```

```
score.mean()
```

```
0.9207439528125739
```

```
grid_search.fit(X_train, y_train)
```

B. DecisionTreeClassifier Training

```
## 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__criterion:gini
clf__max_depth:30
clf__min_samples_leaf:2
clf__min_samples_split:20
```

C. Result

混淆矩陣解讀

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

```
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	0.93	0.93	69432
1	0.93	0.92	0.92	68920
accuracy			0.92	138352
macro avg	0.92	0.92	0.92	138352
weighted avg	0.92	0.92	0.92	138352

Our model

	precision	recall	f1-score	support
0	0.85	0.91	0.88	69432
1	0.91	0.84	0.87	68920
accuracy			0.88	138352
macro avg	0.88	0.88	0.88	138352
weighted avg	0.88	0.88	0.88	138352

baseline

C. Result

```
stick_level20.196 +/- 0.001
recency_m0.180 +/- 0.001
cust_group20.170 +/- 0.001
DiagnosisCode_DESC0.116 +/- 0.001
illness_desc0.102 +/- 0.001
BundleSubtype20.093 +/- 0.001
累積理賠金額  0.063 +/- 0.001
REIMBURSED_YR_TW0.054 +/- 0.000
WEALTH_LEVEL0.047 +/- 0.000
被保人年收   0.036 +/- 0.001
ternure_m0.023 +/- 0.000
REG_his 0.017 +/- 0.000
累積理賠次數  0.010 +/- 0.000
SIN     0.010 +/- 0.000
REG     0.009 +/- 0.000
GENDER  0.008 +/- 0.000
DIGI_FLG0.008 +/- 0.000
結案月份    0.008 +/- 0.000
被保人總資產  0.008 +/- 0.000
具生故保險金受益人0.007 +/- 0.000
ILP     0.007 +/- 0.000
ILP_his 0.006 +/- 0.000
AHa     0.006 +/- 0.000
SIN_his 0.006 +/- 0.000
具滿期金受益人 0.005 +/- 0.000
AHd     0.004 +/- 0.000
AHc     0.004 +/- 0.000
AHb     0.004 +/- 0.000
AHb_his 0.003 +/- 0.000
VIP_CLASS0.002 +/- 0.000
被保人是否為事故人0.002 +/- 0.000
AHc_his 0.001 +/- 0.000
AHd_his 0.001 +/- 0.000
TOPCARD 0.000 +/- 0.000
VIP     0.000 +/- 0.000
AHa_his 0.000 +/- 0.000
```

Our model

```
stick_level20.252 +/- 0.001
recency_m0.231 +/- 0.001
DiagnosisCode_DESC0.229 +/- 0.001
illness_desc0.222 +/- 0.001
cust_group20.220 +/- 0.001
BundleSubtype20.161 +/- 0.001
REIMBURSED_YR_TW0.094 +/- 0.001
WEALTH_LEVEL0.058 +/- 0.001
ternure_m0.056 +/- 0.001
REG_his 0.026 +/- 0.000
GENDER  0.024 +/- 0.000
REG     0.022 +/- 0.000
AHb_his 0.016 +/- 0.000
SIN_his 0.015 +/- 0.000
DIGI_FLG0.011 +/- 0.000
ILP     0.011 +/- 0.000
ILP_his 0.009 +/- 0.000
AHb     0.009 +/- 0.000
AHa     0.008 +/- 0.000
AHd     0.008 +/- 0.000
AHc     0.006 +/- 0.000
SIN     0.004 +/- 0.000
AHd_his 0.003 +/- 0.000
VIP_CLASS0.003 +/- 0.000
TOPCARD 0.001 +/- 0.000
AHc_his 0.001 +/- 0.000
AHa_his 0.000 +/- 0.000
VIP     0.000 +/- 0.000
```

baseline

D. Visualization

視覺化

[17]:
```
from sklearn import tree
tree.export_graphviz(clf2,out_file="tree.dot",feature_names=X_train.columns,class_names=['neg','pos'])
```

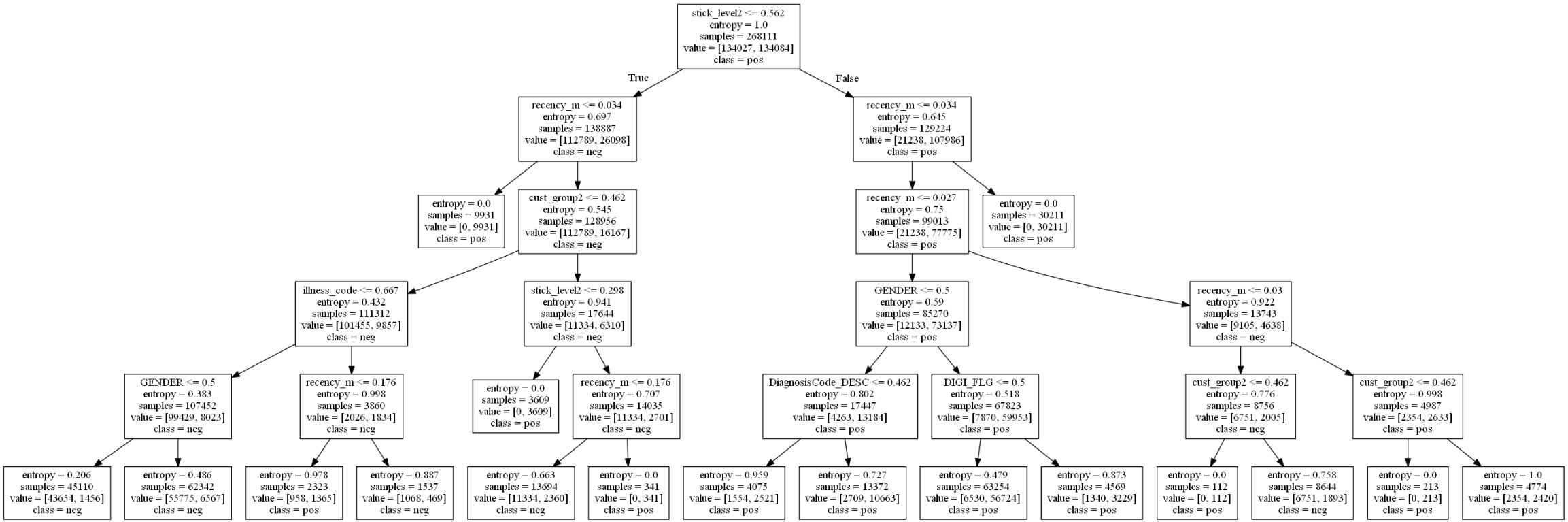
[18]:
```
import pydot
(graph, ) = pydot.graph_from_dot_file('tree.dot')
```

[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()
```

D. Visualization



05

系統延伸發想

Future System

A. 延伸發想

A. 延伸發想

若顯示保戶再購可能高，則建議業務員依該客戶傾向之商品做客製化的介紹和推薦

A. 延伸發想

若結果顯示為再購可能低，則建議業務員加強提升用戶理賠滿意度、忠誠度，提高未來再購機會


Thank You
For
Watching!
ROCKEFELLER

06

附錄
Appendix

A. 分工
B. 備註
C. EDA result

- 周永昱：EDA、資料預處理、機器學習、簡報製作
- 謝宗儒：學習了機器學習相關：KNN、回歸演算法、決策樹、隨機森林、降維演算法、貝葉斯演算法、編碼方式；保險知識、資料前處理：醫療保險、意外險、壽險等保單種類跟概況
- 何恬：理賠再購資料合併分析、資料特徵解讀及選擇、新增延伸特徵、特徵類型轉換、簡報製作
- 陳奕帆：客戶屬性變數分析、合併檔資料分析、再購情形事後比較、不同疾病的再購比例分析、尋找可增加特徵

- Github：https://github.com/teemoteemo0318/nanshan

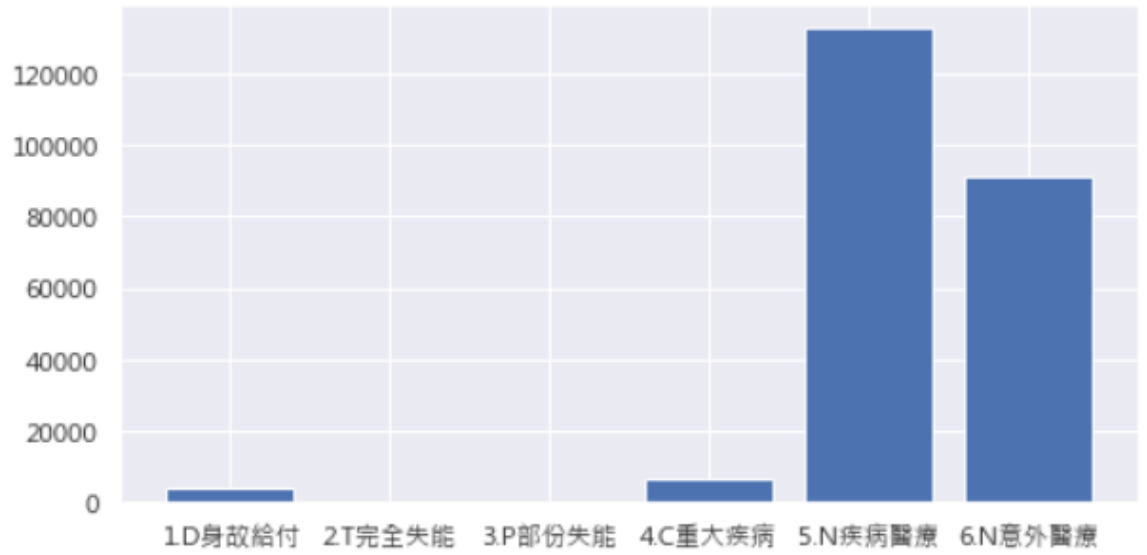
	理賠檔(CLAIM_ACCT_FIN)欄位說明		
	欄位	類型	名稱
1	INJURED_RK	字元	事故人RK
2	Claim_RK	字元	理賠案號
3	Policy_RK	字元	保單號碼
4	BundleSubtype2	字元	理賠案件型態
5	illness_code	字元	疾病代碼
6	illness_desc	字元	疾病名稱
7	DiagnosisCode_DESC	字元	疾病分類名稱
8	claim_settle_dt	日期	理賠結案日期
9	REIMBURSED_YR_TW	數值	理賠金額(歸至該結案年度)
10	INSURED_RK	字元	被保人RK
11	POLICY_HOLDER_RK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

1. 欄位說明

C. EDA result(理賠檔)

```
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
```

2. Missing Value



3. 理賠案件型態人數分配

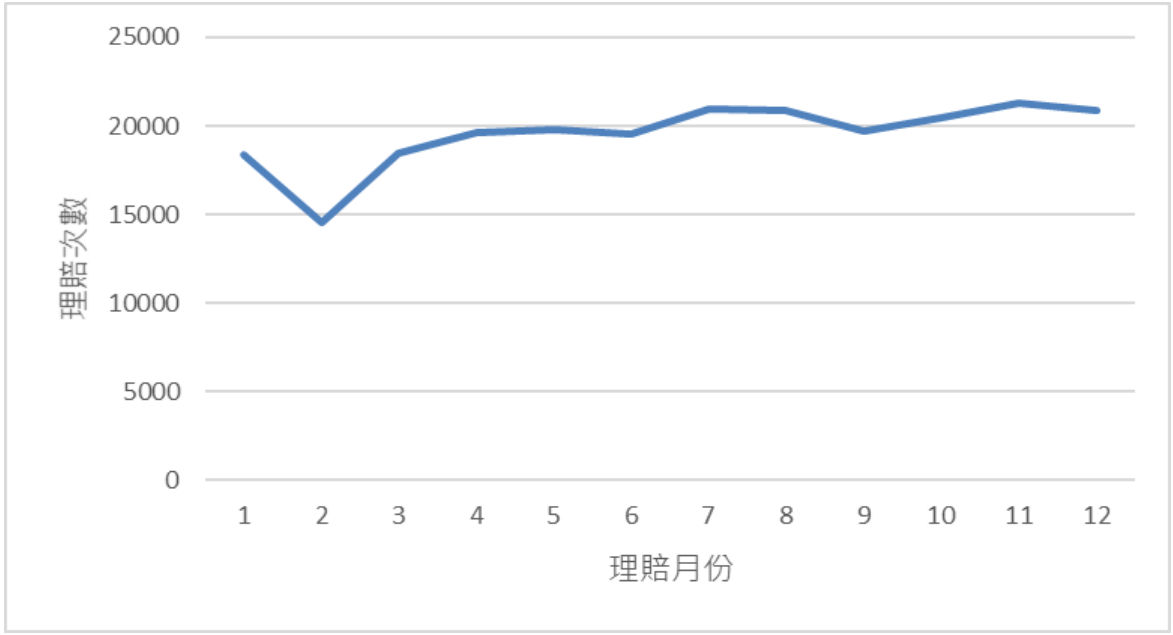
```
W18      26959
Y99.8    22910
V23      15598
C50       9080
Y93.7     8174
          ...
S92.0        1
S71          1
H43          1
Q39.1        1
I51.7        1
```

4. 各項疾病人數

事故人 被保人 要保人重疊情形

```
[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

[16]: # 被保人=要保人
      df[(df['INSURED_RK']==df['POLICY_HOLDER_RK'])]['Policy_RK'].count()

[16]: 172115

[17]: # 事故人=要保人
      df[(df['INJURED_RK']==df['POLICY_HOLDER_RK'])]['Policy_RK'].count()

[17]: 168112

[18]: # 事故人=被保人
      df[(df['INJURED_RK']==df['INSURED_RK'])]['Policy_RK'].count()

[18]: 229636
```

5. 事故人、被保人、要保人重複情況



6. 各月份理賠案件數

C. EDA result(再購明細檔)

再購明細檔(COV_ACCT_FIN)欄位說明				
	欄位	類型	名稱	說明
1	INSURED_RK	字元	被保人RK	
2	Policy_RK	字元	保單號碼	
3	RRKER_CD	數值	主附約註記	1=主約,0=附約
4	payment_period	字元	保費繳法	
5	EFFECTIVE_DT	數值	保單生效日	
6	SHORT_NAME	字元	產品細項	
7	prod_detail2	字元	產品類型	REG=期繳商品,SIN=躉繳商品,ILP=投資型商品,AHa=住院手術商品,AHb=重疾癌症,,AHc=意外保障,AHd=長期照顧
8	POLICY_HOLDER_RK	字元	要保人RK	
9	AFYP_NT	數值	保單保費	
10	MATURITY_BENEFICIARY_RK	字元	滿期金受益人RK	
11	DEATH_BENEFICIARY_RK	字元	生故保險金受益人RK	

1.欄位說明

C. EDA result(再購明細檔)

```
INSURED_RK                  0
Policy_RK                   0
RIDER_CD                    0
payment_period              0
EFFECTIVE_DT                0
SHORT_NAME                  0
prod_detail2                0
POLICY_HOLDER_RK            0
AFYP_NT                     0
MATURITY_BENEFICIARY_RK    41498
DEATH_BENEFICIARY_RK       30049
```

2.Missing Value

```
df['RIDER_CD'].value_counts(dropna=False)

# 主約:1 附約:0
```

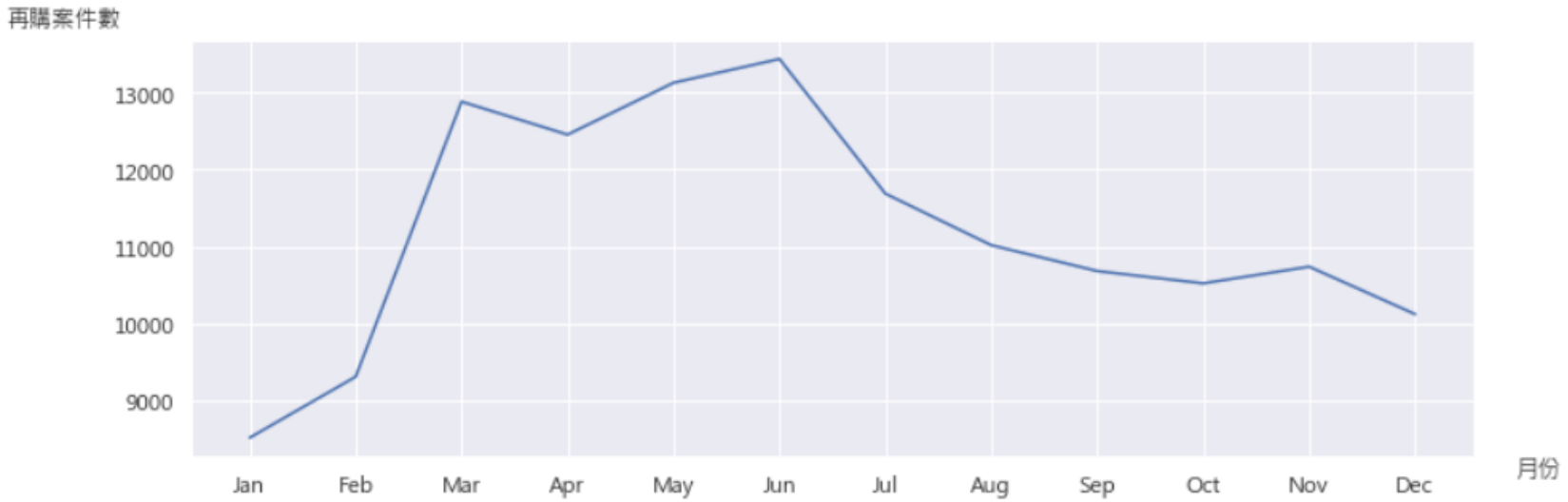
```
1    67804
0    66668
Name: RIDER_CD, dtype: int64
```

3.主、附約數量

C. EDA result(再購明細檔)

年繳	104294
基繳保費	14522
月繳	10506
季繳	3138
半年繳	2012

4. 保費繳法



5.各月份再購案件數

C. EDA result(客戶屬性檔)

客戶屬性檔(CUST_PROPERTY_FIN)欄位說明				
	欄位	類型	名稱	說明
1	CUST_RK	字元	客戶RK	
2	ternure_m	數值	客戶戶齡(month)	
3	recency_m	數值	最近生效日距今(month)	
4	SIN	數值	現在持有躉繳保單	1=現在持有、0=未持有
5	SIN_his	數值	曾經持有躉繳保單	1=曾經持有(包含現在持有)、0=未持有
6	REG	數值	現在持有期繳保單	1=現在持有、0=未持有
7	REG_his	數值	曾經持有期繳保單	1=曾經持有(包含現在持有)、0=未持有
8	ILP	數值	現在持有ILP保單	1=現在持有、0=未持有
9	ILP_his	數值	曾經持有ILP保單	1=曾經持有(包含現在持有)、0=未持有
10	AHa	數值	現在持有AH保單(住院醫療)	1=現在持有、0=未持有
11	AHa_his	數值	曾經持有AH保單(住院醫療)	1=曾經持有(包含現在持有)、0=未持有
12	AHb	數值	現在持有AH保單(重疾癌症)	1=現在持有、0=未持有
13	AHb_his	數值	曾經持有AH保單(重疾癌症)	1=曾經持有(包含現在持有)、0=未持有
14	AHc	數值	現在持有AH保單(意外傷害)	1=現在持有、0=未持有
15	AHc_his	數值	曾經持有AH保單(意外傷害)	1=曾經持有(包含現在持有)、0=未持有
16	AHd	數值	現在持有AH保單(長期照顧)	1=現在持有、0=未持有
17	AHd_his	數值	曾經持有AH保單(長期照顧)	1=曾經持有(包含現在持有)、0=未持有
18	VIP_CLASS	字元	VIP等級	VIP01最高-->VIP05最低
19	VIP	數值	VIP客戶	1=VIP客戶、0=非VIP客戶
20	WEALTH_LEVEL	字元	財富等級	W1最高-->W7最低
21	CLIENT_MARITAL	字元	婚姻狀況	M=已婚、S=單身
22	CLIENT_INCOME	數值	客戶年收入	
23	DIGI_FLG	數值	數位客戶	1=數位客戶、0=非數位客戶
24	TOPCARD	數值	頂級卡	1=頂級卡客戶、0=非頂級卡客戶
25	GENDER	數值	性別	1=女性、0=男性
26	stick_level2	字元	忠誠度	S01最高-->S10最低
27	cust_group2	字元	客戶分群	G0最高-->G4最低
28	TOTAL_AUM	數值	總資產	
29	INSURED_DOB	數值	客戶生日	

分析客戶與公司契約關係時長，一般來說若戶齡越大且最近生效日距今越小者，屬於較忠誠之客戶。但單獨看其中一個變數並無法確定其與忠誠度間的關係，例如戶齡與最近生效日距今數值同樣大表示此客戶僅買過一次公司保單。

可用來分析客戶黏著度及忠誠度及判斷未來是否有其他險種需求。

理論上應與客戶年收入、總資產兩變數有高度相關，但客戶年收入之遺漏值相當多，總資產雖然也有許多遺漏值但相較客戶年收入還算少大致能夠看出與財富等級高度相關。

遺漏值非常多

可能單純為客戶使用或接觸公司之方式，因沒有顯著與VIP或是財富等變數相關