ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH
TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN
KHOA CÔNG NGHỆ THÔNG TIN

BÁO CÁO LAB 02

- Giảng viên hướng dẫn -

Nguyễn Thái Vũ

Thành phố Hồ Chí Minh, tháng năm 2022

ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH

TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN

KHOA CÔNG NGHỆ THÔNG TIN

Kiều Hải Đăng

19127347

Thành phố Hồ Chí Minh, tháng năm 2022

I. Tiền Xử Lý Dữ Liệu

TABLE 3.1 Summary of Exploratory Findings Thus Far

Variable	Disposition
State	Anomalous. Omitted from model.
Account length	No obvious relation with churn, but retained.
Area code	Anomalous. Omitted from model.
Phone number	Surrogate for ID. Omitted from model.
International Plan	Predictive of churn. Retained.
VoiceMail Plan	Predictive of churn. Retained.
Number of voice mail messages	No obvious relation with churn, but retained.
Total day minutes	Predictive of churn. Retained.
Total day calls	No obvious relation with churn, but retained.
Total day charge	Function of *minutes*. Omitted from model.
Total evening minutes	May be predictive of churn. Retained.
Total evening calls	No obvious relation with churn, but retained.
Total evening charge	Function of *minutes*. Omitted from model.
Total night minutes	No obvious relation with churn, but retained.
Total night calls	No obvious relation with churn, but retained.
Total night charge	Function of *minutes*. Omitted from model.
Total international minutes	No obvious relation with churn, but retained.
Total international calls	No obvious relation with churn, but retained.
Total international charge	Function of *minutes*. Omitted from model.
Customer service calls	Predictive of churn. Retained.

Figure 1: Mối quan hệ giữa các biến với churn

Đối với vấn đề tiền xử lý dữ liệu thì theo thông tin trong tài liệu churn_description.pdf, tác giả có chia sẻ về các biến tương quan.

```
# Drop columns: 'Day Charge', 'Eve Charge', 'Night Charge', 'Intl Charge'
df = df.drop(['Day Charge', 'Eve Charge', 'Night Charge', 'Intl Charge', 'State', 'Area Code', 'Phone'], axis=1)
```

Figure 2: Loại bỏ các biến không cần thiết với mô hình

Dựa trên thông tin đó, em tiến hành loại bỏ các cột: Day Charge, Eve Charge, Night Charge, Intl Charge khỏi tập dữ liệu để giảm bớt sự nhấn mạnh quá mức (overemphasize) hoặc gây ảnh hưởng đến tính ổn định của mô hình sau này. Tránh việc khiến mô hình trở nên vô nghĩa.

II. Khám Phá Dữ Liệu

```
State             False
Account Length    False
Area Code         False
Phone             False
Int'l Plan        False
VMail Plan        False
VMail Message     False
Day Mins          False
Day Calls         False
Day Charge        False
Eve Mins          False
Eve Calls         False
Eve Charge        False
Night Mins        False
Night Calls       False
Night Charge      False
Intl Mins         False
Intl Calls        False
Intl Charge       False
CustServ Calls    False
Churn?            False
dtype: bool
```

```
 #   Column          Non-Null Count  Dtype
---  ------          --------------  -----
 0   State           3333 non-null   object
 1   Account Length  3333 non-null   int64
 2   Area Code       3333 non-null   int64
 3   Phone           3333 non-null   object
 4   Int'l Plan      3333 non-null   object
 5   VMail Plan      3333 non-null   object
 6   VMail Message   3333 non-null   int64
 7   Day Mins        3333 non-null   float64
 8   Day Calls       3333 non-null   int64
 9   Day Charge      3333 non-null   float64
 10  Eve Mins        3333 non-null   float64
 11  Eve Calls       3333 non-null   int64
 12  Eve Charge      3333 non-null   float64
 13  Night Mins      3333 non-null   float64
 14  Night Calls     3333 non-null   int64
 15  Night Charge    3333 non-null   float64
 16  Intl Mins       3333 non-null   float64
 17  Intl Calls      3333 non-null   int64
 18  Intl Charge     3333 non-null   float64
 19  CustServ Calls  3333 non-null   int64
 20  Churn?          3333 non-null   object
```

Figure 3: Missing value (left) and data type (right) of dataset

Qua một số bước khám phá cơ bản em nhận thấy dữ liệu phân bố khá ổn 3333 dòng, 21 cột. Missing value là 0% và kiểu dữ liệu của mỗi cột là ổn. Do bước này làm trước bước tiền xử lý nên vẫn còn các cột chưa được loại bỏ.

III. Đặt Câu Hỏi

a) Thời gian active của account (account length) có ảnh hưởng đến quyết định rời bỏ (churn) của khách hàng hay không?

Để trả lời cho câu hỏi này, em nhận thấy rằng dữ liệu ở cột (account length) hiện đang khác định dạng với cột (churn) cụ thể là có dạng numerical. Chính vì thế, em quyết định sử dụng phương pháp **Binning** để chia dữ liệu thành các bin, phục vụ cho việc xây dựng mô hình.

```
count    3333.000000
mean      101.064806
std        39.822106
min         1.000000
25%        74.000000
50%       101.000000
75%       127.000000
max       243.000000
Name: Account Length, dtype: float64
```

Figure 4: Mô tả của cột Account Length

Vậy chia bin sao cho hợp lý? Em nhận thấy rằng tứ phân vị có thể giúp em nhiều trong việc chia bin nên em đã quyết định sử dụng tứ phân vị để phục vụ cho mục đích của mình. Cụ thể, em chia dữ liệu thành bin: very_low, low, high, very_high lần lượt đại diện cho Q1, Q2, Q3 và Q4. Cụ thể hơn là:
[min-25%), [25%-50%), [50%-75%), [75%-max].

```
# Binning to 4 categories prepresent for Q1, Q2, Q3, Q4
very_low = temp.loc[(temp['Account Length'] < _25) & (temp['Account Length'] >= mn)]
low = temp.loc[(temp['Account Length'] < _50) & (temp['Account Length'] >= _25)]
high = temp.loc[(temp['Account Length'] < _75) & (temp['Account Length'] >= _50)]
very_high = temp.loc[(temp['Account Length'] <= mx) & (temp['Account Length'] >= _75)]
```

Figure 5: Binning dữ liệu

Sau khi đã chia bin xong em thu được 4 bin very_low, low, high, very_high và em tiến hành đánh nhãn cho dữ liệu.

```
# Replace value of account length by class name
very_low.loc[:, ['label']] = 'very_low'
low.loc[:, ['label']] = 'low'
high.loc[:, ['label']] = 'high'
very_high.loc[:, ['label']] = 'very_high'
```

Figure 6: Gắn nhãn cho dữ liệu

Sau khi đã gắn nhãn xong, em tiến hành chạy model trên các biến label và churn. Cho cols = ['label', 'Churn?'], khi chạy FP-Growth, em cho minsupp và minconf đều là 0.5 vì ở thời điểm hiện tại, em nghĩ nên cho là 0.5 trước để xem mô hình có chạy qua kết quả không, nếu ra tức là mô hình của mình ổn vì độ chính xác trên 50% thì thực sự rất tuyệt, còn nếu không em sẽ xem xét hạ tỉ lệ xuống.

Đối với từng loại nhãn, em thu được kết quả lần lượt như sau:

```
[[{'False.'}, {'very_low'}, 1.0],
 [{'very_low'}, {'False.'}, 0.8690909090909091]]

[[{'low'}, {'False.'}, 0.8498212157330155], [{'False.'}, {'low'}, 1.0]]

[[{'high'}, {'False.'}, 0.844059405940594], [{'False.'}, {'high'}, 1.0]]

[[{'very_high'}, {'False.'}, 0.8571428571428571],
 [{'False.'}, {'very_high'}, 1.0]]
```

Figure 7: Kết quả thu được sau khi chạy thuật toán FP-Growth

Kết quả có vẻ khả quan nhưng có gì đó hơi lấn cấn ở đây vì confident rất cao. Nhìn chung thì hầu hết dữ liệu thuộc vào các nhãn thì tỉ lệ người dùng không rời đi có độ tin tưởng là trên 80% cho mỗi nhãn. Đây là một độ chính xác rất cao, và có vẻ như việc active ít hay nhiều không ảnh hưởng quá nhiều đến việc rời đi của người dùng. Có lẽ vấn đề này cũng dễ hiểu vì ở đây ta có tỉ lệ người dùng rời khi so với ở lại là khá chênh lệch. Cụ thể là 85.5% - 14.5%.

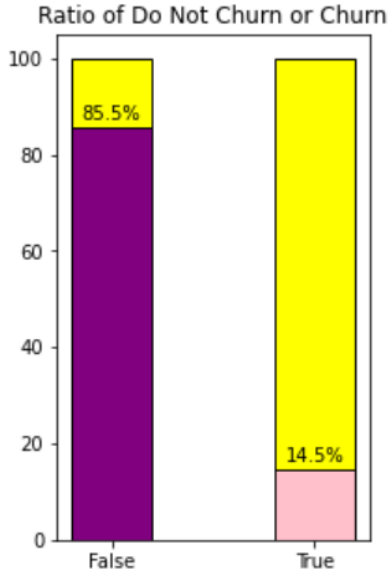


Figure 8: Trực quan hóa cho giá trị của biến churn

Hình bên em trực quan hóa lên để thấy sự phân bố các giá trị rời bỏ hoặc không.

Tiếp theo đó em quyết định concat dữ liệu và chạy lại trên toàn bộ data set.

```
itemSetList = pd.concat([high,very_high,low,very_low])[cols].to_numpy()
freqItemSet, rules = fpgrowth(itemSetList, minSupRatio=0.5, minConf=0.5)
rules
```

Figure 9: Gộp dữ liệu

Với minsup và độ tin cậy là 0.5 thì mô hình không phát hiện ra bất kì quy luật nào tồn tại trong dữ liệu cả. Em quyết định hạ minsup xuống dần và đến mức minsup=0.2, conf=0.5 thì mô hình đã cho ra kết quả.

```
# Run on full data minsup = 0.2
itemSetList = pd.concat([high,very_high,low,very_low])[cols].to_numpy()
freqItemSet, rules = fpgrowth(itemSetList, minSupRatio=0.2, minConf=0.5)
rules
```

```
[[{'high'}, {'False.'}, 0.844059405940594],
 [{'very_low'}, {'False.'}, 0.8690909090909091],
 [{'low'}, {'False.'}, 0.8498212157330155],
 [{'very_high'}, {'False.'}, 0.8571428571428571]]
```

Figure 10: Kết quả của mô hình với minsup=0.2

Qua kết quả này, em nhận thấy rằng chạy riêng hay chạy trên toàn bộ tập dữ liệu thì kết quả vẫn như nhau về confident và từ đó em rút ra kết luận rằng với độ tin cậy trên 80% thì khách hàng sẽ không rời bỏ công ty dù họ active nhiều hay ít.

b) Liệu việc chọn sử dụng gói Voice Mail có ảnh hưởng đến quyết định chọn sử dụng gói International không và ngược lại?

Đối với 2 biến VMail Plan và Int'l Plan vì chúng là biến categorical nhưng bản chất là biến hệ nhị phân, từ đó em quy định ngầm là 1 và 0 chứ không tiền xử lý chỗ này. Giải thích thêm là vì phải chạy luật kết hợp nên em giữ nguyên tên, chỉ xử lý một chút tức là no và yes của VMail Plan em sẽ thêm chứ V đằng sau đó và tương tự cho Int'l Plan, nếu chỉ là 1 và 0 thì khi chạy ra kết quả không thể phân biệt được ví dụ: [{0}, {1}, 0.8] ta không biết vế nào với vế nào cả.

```python
# Preprocess for ['VMail Plan', "Int'l Plan"] values
k = df[['VMail Plan', "Int'l Plan"]]
k["Int'l Plan"].loc[:] = k["Int'l Plan"].apply(lambda x: 'no I' if x == 'no' else 'yes I')
k["VMail Plan"].loc[:] = k["VMail Plan"].apply(lambda x: 'no V' if x == 'no' else 'yes V')
```

Figure 11: Tiền xử lý các giá trị của VMail Plan và Int'l Plan

Tiếp theo đó em tiến hành thống kê tỉ lệ tương tác giữa các cặp biến với nhau. Và thu được kết quả như sau.

```
VMail Plan  Int'l Plan
no V        no I          0.654065
yes V       no I          0.249025
no V        yes I         0.069307
yes V       yes I         0.027603
dtype: float64
```

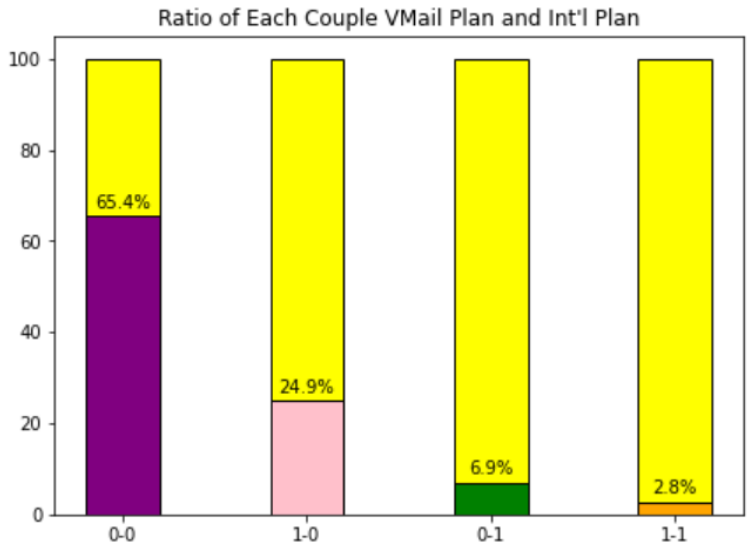


Figure 12: Tỉ lệ tương tác giữa các cặp biến

Qua thống kê, em rút ra một số điểm như sau:

VMail Plan ➔ Int'l Plan	%
No ➔ No	65
Yes ➔ No	24.9
No ➔ Yes	6.9
Yes ➔ Yes	2.8

Trong đó:

Tỉ lệ người không dùng gói Vmail dẫn đến không dùng gói Int'l là 65%.
Tỉ lệ người dùng gói Vmail dẫn đến không dùng gói Int'l là 24.9%.
Tỉ lệ người không dùng gói Vmail dẫn đến dùng gói Int'l là 6.9%.

Tỉ lệ người dùng gói Vmail dẫn đến dùng gói Int'l là 2.8%.

Và tất nhiên là chiều ngược lại của Int'l với Vmail kết quả cho ra tương tự. Qua thống kê cơ bản đó, chúng ta thấy được sự phụ thuộc của 2 biến này với nhau nhưng vẫn chưa đủ để kết luận vấn đề đặt ra ban đầu. Tiếp theo đó em chạy thuật toán FP-Growth để hy vọng rút ra được một luật nào đó. Với thử nghiệm ở mức minsup=0.5 em thu được kết quả như sau:

```
[[{'no V'}, {'no I'}, 0.904189133139776],
 [{'no I'}, {'no V'}, 0.7242524916943521]]
```

Figure 13: FP-Growth với minsup=0.5

Ta có thể nhận xét một vài điểm như sau thông qua kết quả. Với độ tin cậy 90.4% thì người không dùng Vmail sẽ không dùng Int'l. Và ở độ tin cậy 72.4% thì người không dùng gói Int'l sẽ dẫn đến không dùng gói VMail. Nhưng kết quả này vẫn chưa thỏa mãn em nên em muốn thử hạ minsup để xem xét một chút. Sau một vài thử nghiệm thì ở mức minsup=0.2, em chạy thuật toán và thu được kết quả:

```
[[{'yes V'}, {'no I'}, 0.9002169197396963],
 [{'no V'}, {'no I'}, 0.904189133139776],
 [{'no I'}, {'no V'}, 0.7242524916943521]]
```

Figure 14: FP-Growth với minsup=0.2

Ở mức minsup này, mô hình bổ sung thêm một kết quả đó là ở độ tin cậy 90% thì người dùng VMail sẽ không dùng gói Int'l. Có vẻ gói Int'l thực sự có vấn đề và cần cải thiện gấp lại chất lượng dịch vụ để giữ chân khách hàng. Điều này trong file mô tả cũng đã nói rất kỹ. Ban đầu theo file mô tả thì em nghĩ gói Voice Mail tốt hơn gói International nên nếu người dùng sử dụng gói Voice Mail thì sẽ không muốn sang sử dụng gói International nữa hoặc cũng có thể giả định người ta thấy sài gói Voice Mail tốt nên mới đăng ký gói International vì chất lượng dịch vụ của gói trước đó. Nhưng sau khi chạy thực tế thì em xin được kết luận rằng việc sử dụng gói Voice Mail hay không không ảnh hưởng đến việc chọn sử dụng gói International bởi vì nội tại gói International là không tốt, có lẽ vì một vài lý do nào đó nhưng thực tế em thấy là người dùng

dù Yes hay No với Voice Mail nhưng vẫn say No với gói International.

c) Theo như file mô tả thì tác giả có nhấn mạnh về số lượng cuộc gọi customer trong ngày trên 3 thì tỉ lệ khách hàng rời đi sẽ rất cao. Vậy liệu rằng lượng khách hàng rời đi đó họ có sử dụng dịch vụ Voice Mail không? Bởi vì như chúng ta đã biết thì gói Voice Mail hiện đang là gói có tỉ lệ giữ chân khách hàng tốt nhất.

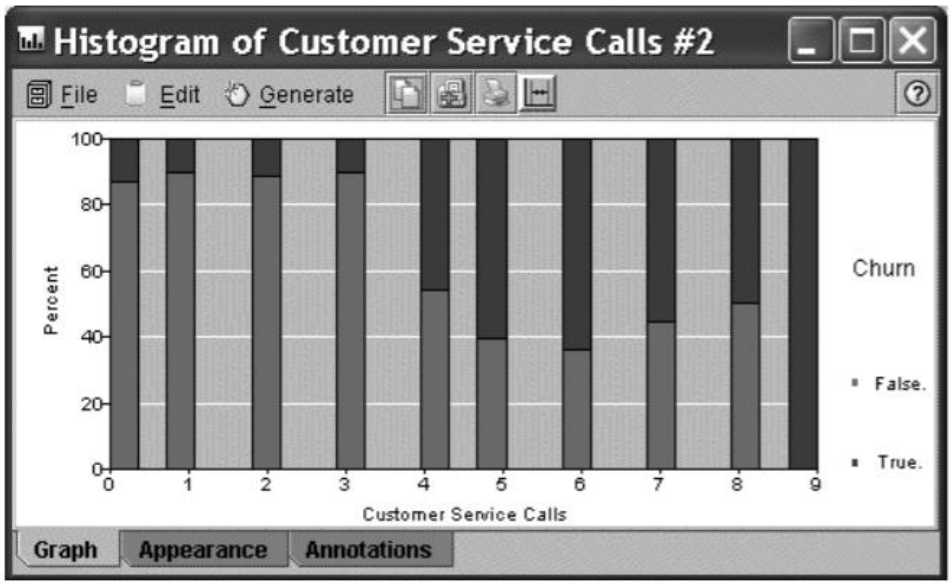


Figure 15: Mô tả của tác giả về dữ liệu theo như đề bài

Đây là tập khách hàng có xu hướng muốn rời đi. Trước tiên em sẽ chuẩn bị tập dữ liệu và gắn nhãn dữ liệu theo trình tự như sau:

	VMail Plan	CustServ Calls	label
10	no	4	high_rate_of_churn
14	no	4	high_rate_of_churn
15	no	4	high_rate_of_churn
21	no	5	high_rate_of_churn
48	no	5	high_rate_of_churn
...	...	...	...
3307	no	4	high_rate_of_churn
3309	no	4	high_rate_of_churn
3320	no	4	high_rate_of_churn
3322	no	4	high_rate_of_churn
3323	no	5	high_rate_of_churn

267 rows × 3 columns

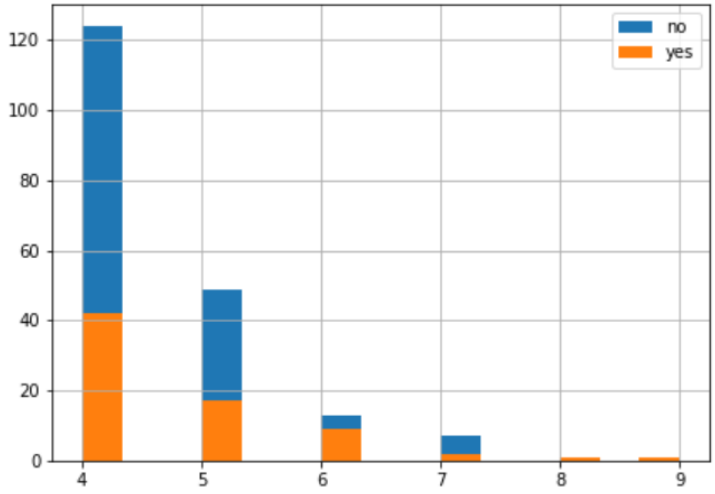
Figure 16: Bảng kết quả

Lọc ra các khách hàng có số cuộc gọi chăm sóc khách hàng > 3, sau đó tiến hành đánh nhãn cho tập khách hàng này để tìm ra luật trên 2 biến categorical. Cụ thể như hình bên dưới:

```
# Get data set of High rate of churn base on custserv_calls
CustServ_Calls_High_Rate_Of_Churn =\
df[['VMail Plan', 'CustServ Calls']].loc[df['CustServ Calls'] >= 4]
CustServ_Calls_High_Rate_Of_Churn.loc[:, 'label'] = 'high_rate_of_churn'
CustServ_Calls_High_Rate_Of_Churn
```

Figure 17: Source code lọc, đánh nhãn dữ liệu

Sau khi đã có được tập dữ liệu, em tiến hành một vài bước thống kê cơ bản. Em chọn groupby theo biến VMail Plan để xem liệu rangwftir lệ phân bố của việc dùng và không dùng gói là như thế nào.




```
VMail Plan  CustServ Calls
no          4                124
            5                 49
            6                 13
            7                  7
            8                  1
            9                  1
yes         4                 42
            5                 17
            6                  9
            7                  2
            8                  1
            9                  1
Name: CustServ Calls, dtype: int64
```

Figure 18: Phân bố của tập dữ liệu

Nhìn vào lược đồ phân bố và số liệu thống kê, ta nhận thấy rằng số cuộc gọi trong ngày chiến từ 4 – 5 cuộc gọi là nhiều nhất. ^ cuộc gọi trong ngày thì nghiêng về phía yes hơn nhưng không đáng kể, còn lại vẫn nghiêng về no và bảo hòa (50-50). Từ đó em rút ra kết luận ban đầu rằng có vẻ như những người có xu hướng gọi chăm sóc khách hàng nhiều lần trong ngày (Xu hướng rời đi)

thường thì không đăng ký sử dụng gói VMail thì phải! Tiếp theo đó, em chạy thuật toán FP-Growth và thu được kết quả như sau:

Vì đây là sample nhỏ chỉ khoảng 267 khách hàng nên em sẽ để chắc chắn về kết quả của mình, em chọn minsup=0.5. Dựa trên kết quả thu được từ tập sample, em xin rút ra kết luận với độ tin

```
itemSetList = CustServ_Calls_High_Rate_Of_Churn[['label', 'VMail Plan']].to_numpy()
freqItemSet, rules = fpgrowth(itemSetList, minSupRatio=0.5, minConf=0.5)
rules
# freqItemSet
```

```
[[{'high_rate_of_churn'}, {'no'}, 0.7303370786516854],
 [{'no'}, {'high_rate_of_churn'}, 1.0]]
```

Figure 19: Kết quả chạy mô hình

cậy 73% thì hầu hết những người có số cuộc gọi trong ngày cao, churn cao thì thường không sử dụng gói VMail. Và với độ tin cậy 100% thì những người không đăng ký gói VMail thì đều có số cuộc gọi trong ngày cao, churn cao.

IV. Tài Liệu Tham Khảo

1. churn_description.pdf
2. churn.txt
3. Wikipedia (3 tháng 1 năm 2022) Hệ Số Tương Quan
4. Pypi.org (Oct 26, 2020) Python implementation of FP Growth algorithm
5. Matplotlib, Matplotlib 3.5.1 documentation