ĐẠI HỌC BÁCH KHOA HÀ NỘI
HANOI UNIVERSITY OF SCIENCE AND TECHNOLOGY

VIỆN TOÁN ỨNG DỤNG VÀ TIN HỌC
School of Applied Mathematics and Informatics

Hệ hỗ trợ quyết định

Báo cáo bài tập lớn

Chủ đề: Xây dựng mô hình dự báo khách hàng rời bỏ dịch vụ

Giảng viên: TS. Lê Hải Hà

Sinh viên thực hiện: Nguyễn Quang Minh

Mã số sinh viên: 20206157

Lớp: Toán Tin 01- K65

Hà Nội - 2023

Mục lục

1 Giới thiệu bài toán Customer Churning

1.1 Tổng quan

Đối với các doanh nghiệp, việc duy trì khách hàng sử dụng dịch vụ của mình đang ngày càng cấp bách trong môi trường cạnh tranh hiện nay, có rất nhiều yếu tố ảnh hưởng đến việc này ví dụ như: chăm sóc khách hàng, chất lượng dịch vụ, áp dụng công nghệ kỹ thuật, chính sách khuyến mãi, nắm bắt tâm lý khách hàng,...

Thực tế nghiên cứu đã chứng minh, chi phí một doanh nghiệp bỏ ra để thu hút khách hàng mới nhiều gấp 5 lần chi phí để giữ chân khách hàng cũ. Hơn nữa, đối với các doanh nghiệp cung cấp sản phẩm là dịch vụ thì việc duy trì số lượng khách hàng sẽ ảnh hưởng trực tiếp đến doanh thu và lợi nhuận của công ty. Chính vì nhu cầu khách quan đó mà bài toán dự báo tỷ lệ khách hàng rời bỏ dịch vụ ra đời và được ứng dụng phổ biến hiện nay.

1.2 Định nghĩa

Khách hàng Churn: là khách hàng ngừng sử dụng dịch vụ với doanh nghiệp trong một khoảng thời gian.

Tỷ lệ Churn: là tổng số lượng khách hàng Churn trên tổng số lượng khách hàng tại một khoảng thời gian.

Trên thực tế, việc khách hàng rời bỏ, không sử dụng dịch vụ cũng xuất phát từ rất nhiều nguyên nhân, phổ biến và có thể kể đến như: dịch vụ chăm sóc khách hàng kém, phản hồi chậm, trải nghiệm không tốt như quảng cáo, nhu cầu của khách hàng đã thay đổi mà doanh nghiệp không nắm bắt kịp thời, không có chính sách quan tâm tới khách hàng lâu năm, khách hàng không thấy giá trị doanh nghiệp và chuyển sang đối thủ cạnh tranh,...

Số lượng khách hàng ngừng tất cả các hoạt động với doanh nghiệp mà lớn sẽ ảnh hưởng rất lớn đến doanh nghiệp. Do đó, doanh nghiệp cần phải phân tích các hoạt động của khách hàng với doanh nghiệp để

từ đó tìm ra các nguyên nhân khách hàng rời bỏ đồng thời dự đoán khách hàng churn. Khi đó sẽ giúp cho các nhà lãnh đạo doanh nghiệp ra những chính sách nhằm lôi kéo khách hàng quay lại và phải cải tiến bản thân doanh nghiệp để phát triển doanh nghiệp bền vững.

2 Phương pháp giải quyết bài toán

2.1 One Hot Encoding

One hot encoding là một kỹ thuật biến đổi dữ liệu phổ biến trong các bài toán học máy. Kỹ thuật này được sử dụng để biến đổi các biến đầu vào (như danh mục, loại, vị trí, v.v.) thành các biến đầu ra (các giá trị 0 hoặc 1) để thuận tiện cho việc phân tích và sử dụng trong các mô hình dự đoán.

Ý tưởng chính của one hot encoding là chuyển đổi một biến có giá trị rời rạc thành một vector có độ dài bằng với số lượng các loại và chỉ có một giá trị 1, các giá trị còn lại đều bằng 0
Ví dụ về One hot encoding:

Loại hoa	Iris Setosa	Iris Versicolour	Iris Virginica
Setosa	1	0	0
Versicolour	0	1	0
Virginica	0	0	1

2.2 Min Max Scaler

Là một phương pháp chuẩn hóa dữ liệu đơn giản bằng cách chuyển dữ liệu về cùng một khoảng, ví dụ như (0-1) hoặc (-1,1) tùy thuộc vào bài toán. Nếu muốn đưa một thành phần (feature) về khoảng [0-1] thì ta có công thức

$$x^{'} = \frac{x - min(x)}{max(x) - min(x)}$$

Trong đó x là dữ liệu ban đầu, $x^{'}$ là giá trị sau khi chuẩn hóa, $min(x)$ và $max(x)$ là giá trị nhỏ nhất và lớn nhất của trường đó.

2.3 SMOTE

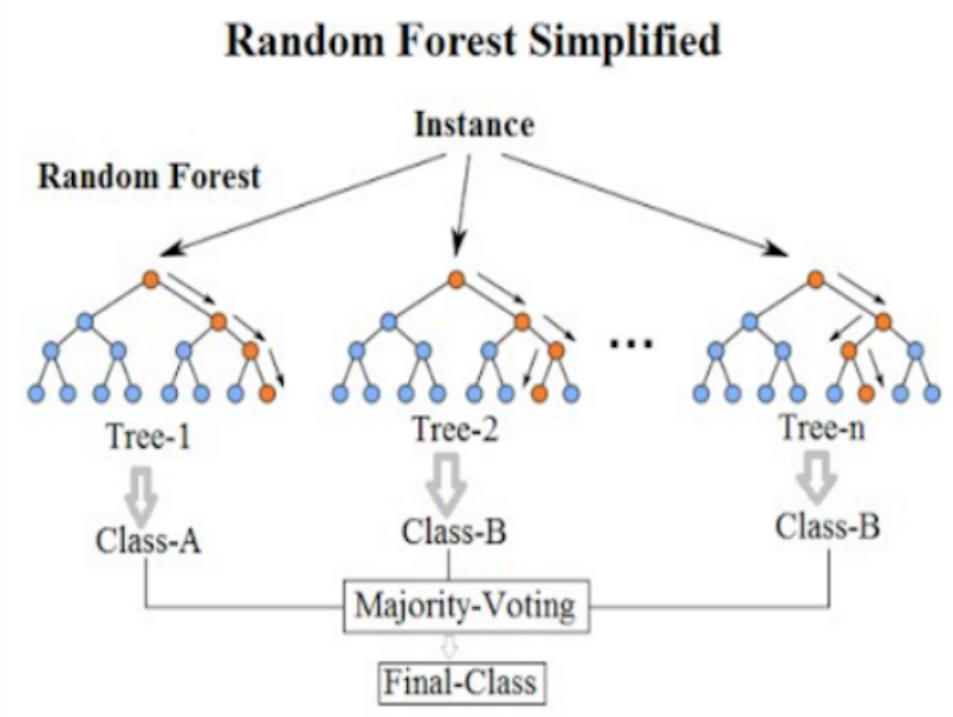
Oversampling: là phương pháp giúp giải quyết hiện tượng mất cân bằng dữ liệu bằng cách gia tăng kích thước mẫu thuộc nhóm thiểu số bằng các kĩ thuật khác nhau.

SMOTE (Synthetic Minority Over-sampling) là phương pháp sinh mẫu nhằm gia tăng kích thước mẫu của nhóm thiểu số trong trường hợp xảy ra mất cân bằng mẫu. Để gia tăng kích thước mẫu, với mỗi một mẫu thuộc nhóm thiểu số ta sẽ lựa chọn ra k mẫu láng giềng gần nhất với nó và sau đó thực hiện tổ hợp tuyến tính để tạo ra mẫu giả lập.Chawla, Bowyer, Hall, and Kegelmeyer (2002)

2.4 Random Forest

Rừng ngẫu nhiên (Random Forest) là một tập hợp của nhiều cây quyết định. Mỗi một cây quyết định sẽ trả ra một kết quả dự báo. Quyết định cuối cùng về nhãn của quan sát sẽ dựa trên nguyên tắc bầu cử đa số (Majority-Voting) trên toàn bộ các cây quyết định con.

Random Forest là thuật toán thuộc lớp mô hình kết hợp (ensemble



model). Kết quả của thuật toán dựa trên bầu cử đa số từ nhiều cây quyết định. Do đó mô hình có độ tin cậy cao hơn và độ chính xác tốt hơn so với những mô hình phân loại tuyến tính đơn giản như logistic hoặc linear regression. Rigatti (2017)

3 Các bước giải quyết bài toán

3.1 Tổng quan về bộ dữ liệu

Bộ dữ liệu phản ánh hành vi và trạng thái hiện tại của người dùng và từ đó ta sẽ dự đoán được khả năng người dùng sẽ rời đi hoặc ở lại trong 6 tháng tới.

a) Quy mô dữ liệu
Dữ liệu được sử dụng từ file Churning_Modelling.csv với kích thước là 669 Kb và được lấy trên Kaggle gồm 10000 bản ghi, 14 thuộc tính.

b) Các thuộc tính
Ta quan tâm đến đánh giá khả năng rời đi của khách hàng là giá trị cột Exited được gán giá trị:

- 1 nếu người dùng này khả năng cao sẽ rời đi.
- 0 nếu người dùng này ở lại.

Đánh giá các tiêu chí góp phần quyết định rời đi của người dùng bao gồm như một số thuộc tính sau:

- CreditScore: Điểm tín dụng của khách hàng.
- Geography: Đất nước đang sinh sống.
- Gender: Giới tính.
- Age: Tuổi.
- Tenure: Số lượng kỳ hạn khách hàng đã sử dụng dịch vụ với ngân hàng.
- Balance: Số dư tài khoản.

- NumOfProducts: Số lượng dịch vụ của ngân hàng mà khách hàng đã sử dụng.
- HasCrCard: Khách hàng có thẻ tín dụng hay không?
- IsActiveMember: Hoạt động của khách hàng (có thường xuyên hay không?)
- EstimatedSalary: Mức lương ước tính.

3.2 Khám phá dữ liệu

Bảng dữ liệu gồm tất cả các cột không tồn tại giá trị null. Trong bảng có 3 kiểu dữ liệu, 2 cột có kiểu dữ liệu là float64, 9 cột có kiểu dữ liệu int64, 3 cột có kiểu object.

```
data.info()
```

```
 #   Column           Non-Null Count  Dtype
---  ------           --------------  -----
 0   RowNumber        10000 non-null  int64
 1   CustomerId       10000 non-null  int64
 2   Surname          10000 non-null  object
 3   CreditScore      10000 non-null  int64
 4   Geography        10000 non-null  object
 5   Gender           10000 non-null  object
 6   Age              10000 non-null  int64
 7   Tenure           10000 non-null  int64
 8   Balance          10000 non-null  float64
 9   NumOfProducts    10000 non-null  int64
 10  HasCrCard        10000 non-null  int64
 11  IsActiveMember   10000 non-null  int64
 12  EstimatedSalary  10000 non-null  float64
 13  Exited           10000 non-null  int64
dtypes: float64(2), int64(9), object(3)
```

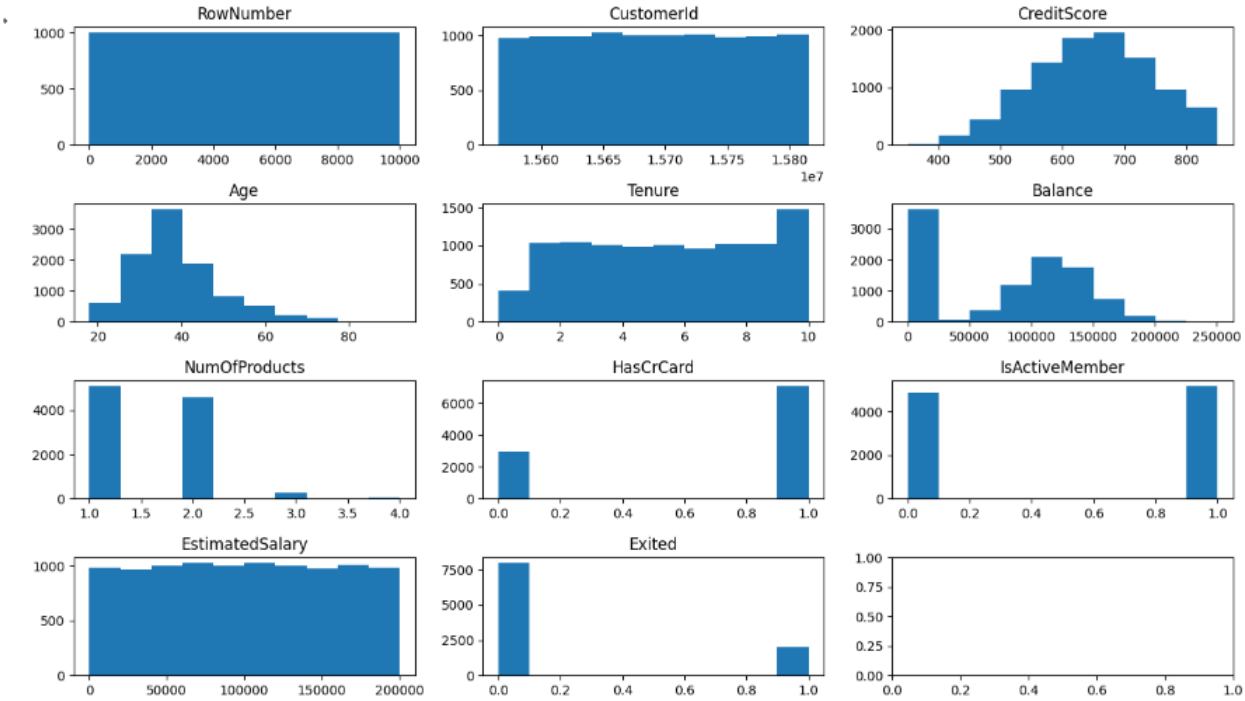
Thống kê các trường dữ liệu là numeric như trung vị (mean), phương sai (std), giá trị nhỏ nhất (min), giá trị lớn nhất (max) và các tứ phân vị.

```
data.describe()
```

	RowNumber	CustomerId	CreditScore	Age	Tenure	Balance	NumOfProducts	HasCrCard	IsActiveMember	EstimatedSalary	Exited
count	10000.00000	1.000000e+04	10000.000000	10000.000000	10000.000000	10000.000000	10000.000000	10000.00000	10000.000000	10000.000000	10000.000000
mean	5000.50000	1.569094e+07	650.528800	38.921800	5.012800	76485.889288	1.530200	0.70550	0.515100	100090.239881	0.203700
std	2886.89568	7.193619e+04	96.653299	10.487806	2.892174	62397.405202	0.581654	0.45584	0.499797	57510.492818	0.402769
min	1.00000	1.556570e+07	350.000000	18.000000	0.000000	0.000000	1.000000	0.00000	0.000000	11.580000	0.000000
25%	2500.75000	1.562853e+07	584.000000	32.000000	3.000000	0.000000	1.000000	0.00000	0.000000	51002.110000	0.000000
50%	5000.50000	1.569074e+07	652.000000	37.000000	5.000000	97198.540000	1.000000	1.00000	1.000000	100193.915000	0.000000
75%	7500.25000	1.575323e+07	718.000000	44.000000	7.000000	127644.240000	2.000000	1.00000	1.000000	149388.247500	0.000000
max	10000.00000	1.581569e+07	850.000000	92.000000	10.000000	250898.090000	4.000000	1.00000	1.000000	199992.480000	1.000000

Dùng biểu đồ Histogram để thể hiện tần suất xuất hiện các giá trị của từng thuộc tính trong tập dữ liệu. Ta thấy không có trường nào có giá trị outlier(ngoại lai)

```
fig,ax=plt.subplots(4,3,figsize=(16,9))
fig.subplots_adjust(wspace=0.2,hspace=0.5)
for i,ax in enumerate(ax.flat):
  ax.hist(data[numeric_cols[i]])
  ax.set_title(numeric_cols[i])
plt.show()
```



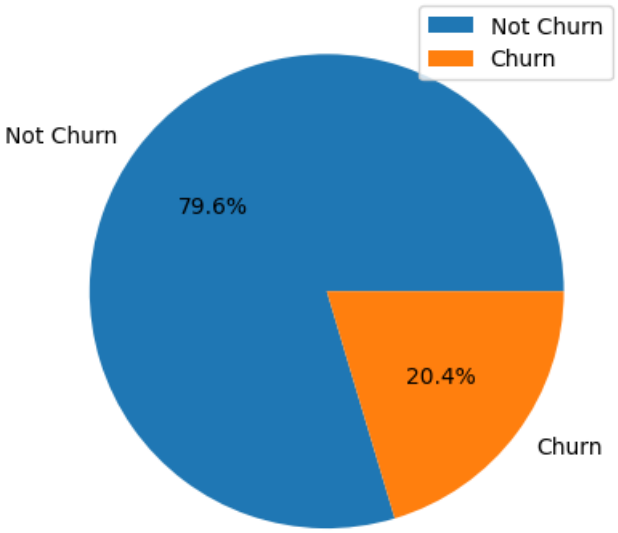
Phân tích tương quan để xem mối quan hệ tuyến tính giữa các biến. Ta thấy các biến độc lập và biến phụ thuộc (Exited) không tương quan với nhau nên không cần phải loại bỏ chúng trong mô hình dự báo.

```
corr=data.corr()
corr.style.background_gradient(cmap='coolwarm').set_precision(2)
```

	RowNumber	CustomerId	CreditScore	Age	Tenure	Balance	NumOfProducts	HasCrCard	IsActiveMember	EstimatedSalary	Exited
RowNumber	1.00	0.00	0.01	0.00	-0.01	-0.01	0.01	0.00	0.01	-0.01	-0.02
CustomerId	0.00	1.00	0.01	0.01	-0.01	-0.01	0.02	-0.01	0.00	0.02	-0.01
CreditScore	0.01	0.01	1.00	-0.00	0.00	0.01	0.01	-0.01	0.03	-0.00	-0.03
Age	0.00	0.01	-0.00	1.00	-0.01	0.03	-0.03	-0.01	0.09	-0.01	0.29
Tenure	-0.01	-0.01	0.00	-0.01	1.00	-0.01	0.01	0.02	-0.03	0.01	-0.01
Balance	-0.01	-0.01	0.01	0.03	-0.01	1.00	-0.30	-0.01	-0.01	0.01	0.12
NumOfProducts	0.01	0.02	0.01	-0.03	0.01	-0.30	1.00	0.00	0.01	0.01	-0.05
HasCrCard	0.00	-0.01	-0.01	-0.01	0.02	-0.01	0.00	1.00	-0.01	-0.01	-0.01
IsActiveMember	0.01	0.00	0.03	0.09	-0.03	-0.01	0.01	-0.01	1.00	-0.01	-0.16
EstimatedSalary	-0.01	0.02	-0.00	-0.01	0.01	0.01	0.01	-0.01	-0.01	1.00	0.01
Exited	-0.02	-0.01	-0.03	0.29	-0.01	0.12	-0.05	-0.01	-0.16	0.01	1.00

Sử dụng biểu đồ tròn (Pie plot) để hiển thị tỷ lệ khách hàng có khả năng rời bỏ dịch vụ thông qua trường **Exited**. Ta có thể thấy tỉ lệ khách hàng rời bỏ là 20.4% thấp hơn nhiều so với 79.6% của các khách hàng còn lại. Dữ liệu này bị mất cân bằng vì vậy ta cần phải có những giải pháp để cân bằng lại dữ liệu để mô hình dự báo đạt kết quả tốt nhất.

```
labels=['Not Churn','Churn']
values=data['Exited'].value_counts()
plt.pie(values,labels = labels,autopct='%1.1f%%')
plt.legend()
plt.show()
```



3.3 Tiền xử lý dữ liệu

Loại bỏ những trường không liên quan gì tới mô hình như: RowNumber(số lượng dòng), CustomerId(mã khách hàng), Surname(tên khách

hàng)

```
data.drop(['RowNumber','CustomerId','Surname'],axis=1,inplace=True)
data.head()
```

	CreditScore	Geography	Gender	Age	Tenure	Balance	NumOfProducts	HasCrCard	IsActiveMember	EstimatedSalary	Exited
0	619	France	Female	42	2	0.00	1	1	1	101348.88	1
1	608	Spain	Female	41	1	83807.86	1	0	1	112542.58	0
2	502	France	Female	42	8	159660.80	3	1	0	113931.57	1
3	699	France	Female	39	1	0.00	2	0	0	93826.63	0
4	850	Spain	Female	43	2	125510.82	1	1	1	79084.10	0

Mã hóa One hot các trường dữ liệu là category như: Geography, Gender để chuyển tất cả các trường dữ liệu về dạng số vì các thuật toán học máy chỉ học được từ dữ liệu dạng này.

```
data=pd.get_dummies(data,drop_first=True)
data.head()
```

	CreditScore	Age	Tenure	Balance	NumOfProducts	HasCrCard	IsActiveMember	EstimatedSalary	Exited	Geography_Germany	Geography_Spain	Gender_Male
0	619	42	2	0.00	1	1	1	101348.88	1	0	0	0
1	608	41	1	83807.86	1	0	1	112542.58	0	0	1	0
2	502	42	8	159660.80	3	1	0	113931.57	1	0	0	0
3	699	39	1	0.00	2	0	0	93826.63	0	0	0	0
4	850	43	2	125510.82	1	1	1	79084.10	0	0	1	0

Chuẩn hóa dữ liệu đưa hết thuộc tính về cùng một khoảng (0-1) để cho mô hình hội tụ nhanh hơn.

```
scale_cols=['CreditScore','Age','Tenure','Balance','NumOfProducts'
,'EstimatedSalary']
scaler=MinMaxScaler()
data[scale_cols]=scaler.fit_transform(data[scale_cols])
data.head()
```

	CreditScore	Age	Tenure	Balance	NumOfProducts	HasCrCard	IsActiveMember	EstimatedSalary	Exited	Geography_Germany	Geography_Spain	Gender_Male
0	0.538	0.324324	0.2	0.000000	0.000000	1	1	0.506735	1	0	0	0
1	0.516	0.310811	0.1	0.334031	0.000000	0	1	0.562709	0	0	1	0
2	0.304	0.324324	0.8	0.636357	0.666667	1	0	0.569654	1	0	0	0
3	0.698	0.283784	0.1	0.000000	0.333333	0	0	0.469120	0	0	0	0
4	1.000	0.337838	0.2	0.500246	0.000000	1	1	0.395400	0	0	1	0

3.4 Xây dựng mô hình

Tách biến độc lập và biến phụ thuộc

```
X=data.drop(['Exited'],axis=1)
```

```
y=data['Exited']
```

Sử dụng SMOTE để oversampling dữ liệu ít nhãn hơn giúp cho 2 nhãn trở nên cân bằng

```
sm = SMOTE()
X_train_resample,y_train_resample=sm.fit_resample(X, y)
```

Chia tập dữ liệu ra làm tập train và test với tỉ lệ 80%-20%. Tập train dùng để huấn luyện mô hình và tập test dùng để kiểm tra độ chính xác của mô hình huấn luyện.

```
X_train,X_test,y_train,y_test=train_test_split(X_train_resample
,y_train_resample,test_size=0.2,random_state=42)
```

Sử dụng mô hình Random Forest với số lượng cây là 200

```
model_RFC=RandomForestClassifier(n_estimators=200,random_state=42)
model_RFC.fit(X_train, y_train)
```

Dự đoán kết quả huấn luyện trên tập test

```
y_pred=model_RFC.predict(X_test)
print(classification_report(y_test,y_pred))
```

```
              precision    recall  f1-score   support

           0       0.91      0.89      0.90      1633
           1       0.89      0.91      0.90      1553

    accuracy                           0.90      3186
   macro avg       0.90      0.90      0.90      3186
weighted avg       0.90      0.90      0.90      3186
```

3.5 Đánh giá kết quả

Ta có thể thấy mô hình đạt độ chính xác (Accuracy) là 90% khá cao, F1 score cả hai lớp đều đạt là 0.9 cũng rất ấn tượng. Điều này cho thấy việc Oversampling + Ensemble learning đạt hiệu quả tốt cho các bộ dữ liệu bị mất cân bằng.

Tài liệu

Chawla, N. V., Bowyer, K. W., Hall, L. O., & Kegelmeyer, W. P. (2002). Smote: synthetic minority over-sampling technique. *Journal of artificial intelligence research*, *16*, 321–357.

Rigatti, S. J. (2017). Random forest. *Journal of Insurance Medicine*, *47*(1), 31–39.