Ex 4: Customer Predictive Analytics

Sự gia tăng dữ liệu xung quanh hành vi (behavior) và nhân khẩu học (demographics) của khách hàng đã mở ra rất nhiều tiềm năng cho các chiến lược tiếp thị kỹ thuật số (digital marketing strategies) sử dụng phân tích dự đoán (predictive analytics)

Cho dữ liệu Marketing-Customer-Value-Analysis.csv chứa thông tin khách hàng xung quanh việc bán bảo hiểm xe hơi. Nhiệm vụ là dự đoán liệu khách hàng có phản hồi cuộc gọi bán hàng hay không dựa trên dữ liệu nhân khẩu học và hành vi trong quá khứ của họ.

Yêu cầu: đọc dữ liệu về, chuẩn hóa dữ liệu (nếu cần) và áp dụng thuật toán SVM để thực hiện việc dự đoán khách hàng response (1 hay 0) dựa trên thông tin được cung cấp

1. Đọc dữ liệu. Tiền xử lý dữ liệu nếu cần. Trực quan hóa dữ liệu.
2. Tạo X_train, X_test, y_train, y_test từ dữ liệu đọc được với tỷ lệ dữ liệu test là 0.2
3. Áp dụng thuật toán SVM
4. Tìm kết quả. Kiểm tra độ chính xác. Nhận xét model.

In [1]:
```
# from google.colab import drive
# drive.mount("/content/gdrive", force_remount=True)
```

In [2]:
```
# %cd '/content/gdrive/My Drive/LDS6_MachineLearning/practice_2023/Chapter6_SVM/'
```

In [3]:
```
from sklearn import datasets
from sklearn import svm
from sklearn.model_selection import train_test_split
import numpy as np
import pandas as pd
```

In [4]:
```
import matplotlib.pyplot as plt
import seaborn as sns
```

In [5]:
```
data = pd.read_csv("Marketing-Customer-Value-Analysis.csv")
```

In [6]:
```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 9134 entries, 0 to 9133
Data columns (total 24 columns):
Customer                       9134 non-null object
State                          9134 non-null object
Customer Lifetime Value        9134 non-null float64
Response                       9134 non-null object
Coverage                       9134 non-null object
Education                      9134 non-null object
Effective To Date              9134 non-null object
EmploymentStatus               9134 non-null object
Gender                         9134 non-null object
Income                         9134 non-null int64
Location Code                  9134 non-null object
Marital Status                 9134 non-null object
Monthly Premium Auto           9134 non-null int64
Months Since Last Claim        9134 non-null int64
Months Since Policy Inception  9134 non-null int64
Number of Open Complaints      9134 non-null int64
Number of Policies             9134 non-null int64
Policy Type                    9134 non-null object
Policy                         9134 non-null object
Renew Offer Type               9134 non-null object
Sales Channel                  9134 non-null object
Total Claim Amount             9134 non-null float64
Vehicle Class                  9134 non-null object
Vehicle Size                   9134 non-null object
dtypes: float64(2), int64(6), object(16)
memory usage: 1.7+ MB
```

In [7]:
```
data.head()
```

Out[7]:

	Customer	State	Customer Lifetime Value	Response	Coverage	Education	Effective To Date	EmploymentStatus	Gender	Income	...	Months Since Policy Inception	Number of Open Complaints	N
0	BU79786	Washington	2763.519279	No	Basic	Bachelor	2/24/11	Employed	F	56274	...	5	0	
1	QZ44356	Arizona	6979.535903	No	Extended	Bachelor	1/31/11	Unemployed	F	0	...	42	0	
2	AI49188	Nevada	12887.431650	No	Premium	Bachelor	2/19/11	Employed	F	48767	...	38	0	
3	WW63253	California	7645.861827	No	Basic	Bachelor	1/20/11	Unemployed	M	0	...	65	0	
4	HB64268	Washington	2813.692575	No	Basic	Bachelor	2/3/11	Employed	M	43836	...	44	0	

5 rows × 24 columns

In [8]:
```
data["Response"].value_counts()
```

Out[8]:
```
No     7826
Yes    1308
Name: Response, dtype: int64
```

In [9]:
```
data["Response"] = data["Response"].apply(lambda x : 0 if x == 'No' else 1)
```

In [10]:
```
X = data.drop(['Customer','Effective To Date'], axis = 1)
```

In [11]:
```
y = data["Response"]
```

In [12]:
```
X.head()
```

Out[12]:

	State	Customer Lifetime Value	Response	Coverage	Education	EmploymentStatus	Gender	Income	Location Code	Marital Status	...	Months Since Policy Inception	Number of Open Complaints	Nu Pc
0	Washington	2763.519279	0	Basic	Bachelor	Employed	F	56274	Suburban	Married	...	5	0	
1	Arizona	6979.535903	0	Extended	Bachelor	Unemployed	F	0	Suburban	Single	...	42	0	
2	Nevada	12887.431650	0	Premium	Bachelor	Employed	F	48767	Suburban	Married	...	38	0	
3	California	7645.861827	0	Basic	Bachelor	Unemployed	M	0	Suburban	Married	...	65	0	
4	Washington	2813.692575	0	Basic	Bachelor	Employed	M	43836	Rural	Single	...	44	0	

5 rows × 22 columns

In [13]:
```
X = pd.get_dummies(X, drop_first=True)
```

In [14]:
```
X.head()
```

Out[14]:

	Customer Lifetime Value	Response	Income	Monthly Premium Auto	Months Since Last Claim	Months Since Policy Inception	Number of Open Complaints	Number of Policies	Total Claim Amount	State_California	...	Sales Channel_Branch	Sa Channel_ Cen
0	2763.519279	0	56274	69	32	5	0	1	384.811147	0	...	0	
1	6979.535903	0	0	94	13	42	0	8	1131.464935	0	...	0	
2	12887.431650	0	48767	108	18	38	0	2	566.472247	0	...	0	
3	7645.861827	0	0	106	18	65	0	7	529.881344	1	...	0	
4	2813.692575	0	43836	73	12	44	0	1	138.130879	0	...	0	

5 rows × 51 columns

In [15]:
```
y.head()
```

Out[15]:
```
0    0
1    0
2    0
3    0
4    0
Name: Response, dtype: int64
```

In [16]:
```
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.20,
                                                    random_state = 42)
```

In [17]:
```
clf = svm.SVC()
clf.fit(X_train, y_train)
```

```
c:\program files\python36\lib\site-packages\sklearn\svm\base.py:193: FutureWarning: The default value of gamma will change fr
om 'auto' to 'scale' in version 0.22 to account better for unscaled features. Set gamma explicitly to 'auto' or 'scale' to av
oid this warning.
  "avoid this warning.", FutureWarning)
```

Out[17]:
```
SVC(C=1.0, cache_size=200, class_weight=None, coef0=0.0,
    decision_function_shape='ovr', degree=3, gamma='auto_deprecated',
    kernel='rbf', max_iter=-1, probability=False, random_state=None,
    shrinking=True, tol=0.001, verbose=False)
```

In [18]:
```
y_pred = clf.predict(X_test)
```

In [19]:
```
#y_pred
```

In [20]:
```
from sklearn.metrics import accuracy_score
print("Accuracy is ", accuracy_score(y_test,y_pred)*100,"%")
```

```
Accuracy is  99.89053092501369 %
```

In [21]:
```
# Kiểm tra độ chính xác
print("The Train Score is: ",
      clf.score(X_train,y_train)*100,"%")
print("The Test Score is: ",
      clf.score(X_test,y_test)*100,"%")
```

```
The Train Score is:  100.0 %
The Test Score is:  99.89053092501369 %
```

In [22]:
```
from sklearn.metrics import classification_report, confusion_matrix
print(confusion_matrix(y_test,y_pred))
print(classification_report(y_test,y_pred))
```

```
[[1561    0]
 [   2  264]]
              precision    recall  f1-score   support

           0       1.00      1.00      1.00      1561
           1       1.00      0.99      1.00       266

    accuracy                           1.00      1827
   macro avg       1.00      1.00      1.00      1827
weighted avg       1.00      1.00      1.00      1827
```

Kết quả:

- R^2 của cả train và test đều cao và như nhau
- Precision và recall đều cao
- => Model phù hợp

In [23]:
```
# https://towardsdatascience.com/predictive-analytics-on-customer-behavior-with-support-vector-machines-svm-7e68fd2be610
```