task-1

February 22, 2024

1 Churn_Modeling

[1]:
```
import pandas as pd
```

[2]:
```
import numpy as np
```

[4]:
```
pip install --upgrade pandas
```

```
Requirement already satisfied: pandas in /usr/local/lib/python3.10/dist-packages
(2.2.0)
Requirement already satisfied: numpy<2,>=1.22.4 in
/usr/local/lib/python3.10/dist-packages (from pandas) (1.25.2)
Requirement already satisfied: python-dateutil>=2.8.2 in
/usr/local/lib/python3.10/dist-packages (from pandas) (2.8.2)
Requirement already satisfied: pytz>=2020.1 in /usr/local/lib/python3.10/dist-
packages (from pandas) (2023.4)
Requirement already satisfied: tzdata>=2022.7 in /usr/local/lib/python3.10/dist-
packages (from pandas) (2024.1)
Requirement already satisfied: six>=1.5 in /usr/local/lib/python3.10/dist-
packages (from python-dateutil>=2.8.2->pandas) (1.16.0)
```

[5]:
```
pd.__version__
```

[5]: '2.2.0'

[12]:
```
data = pd.read_csv("/content/churn.csv")
```

[13]:
```
data
```

[13]:
```
      RowNumber  CustomerId   Surname  CreditScore Geography  Gender  Age  \
0             1    15634602  Hargrave          619    France  Female   42
1             2    15647311      Hill          608     Spain  Female   41
2             3    15619304      Onio          502    France  Female   42
3             4    15701354      Boni          699    France  Female   39
4             5    15737888  Mitchell          850     Spain  Female   43
…           …          …         …           …         …      …  …
9995       9996    15606229  Obijiaku          771    France    Male   39
```

```
9996        9997   15569892  Johnstone          516    France    Male   35
9997        9998   15584532        Liu          709    France  Female   36
9998        9999   15682355  Sabbatini          772   Germany    Male   42
9999       10000   15628319     Walker          792    France  Female   28

      Tenure    Balance  NumOfProducts  HasCrCard  IsActiveMember  \
0          2       0.00              1          1               1
1          1   83807.86              1          0               1
2          8  159660.80              3          1               0
3          1       0.00              2          0               0
4          2  125510.82              1          1               1
…        …         …            …         …          …
9995       5       0.00              2          1               0
9996      10   57369.61              1          1               1
9997       7       0.00              1          0               1
9998       3   75075.31              2          1               0
9999       4  130142.79              1          1               0

      EstimatedSalary  Exited
0           101348.88       1
1           112542.58       0
2           113931.57       1
3            93826.63       0
4            79084.10       0
…                 …      …
9995         96270.64       0
9996        101699.77       0
9997         42085.58       1
9998         92888.52       1
9999         38190.78       0

[10000 rows x 14 columns]
```

[14]:
```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 10000 entries, 0 to 9999
Data columns (total 14 columns):
 #   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)
memory usage: 1.1+ MB
```

[15]:
```
data.duplicated().sum()
```

[15]:
```
0
```

[16]:
```
data.head()
```

[16]:
```
   RowNumber  CustomerId   Surname  CreditScore Geography  Gender  Age  \
0          1    15634602  Hargrave          619    France  Female   42
1          2    15647311      Hill          608     Spain  Female   41
2          3    15619304      Onio          502    France  Female   42
3          4    15701354      Boni          699    France  Female   39
4          5    15737888  Mitchell          850     Spain  Female   43

   Tenure    Balance  NumOfProducts  HasCrCard  IsActiveMember  \
0       2       0.00              1          1               1
1       1   83807.86              1          0               1
2       8  159660.80              3          1               0
3       1       0.00              2          0               0
4       2  125510.82              1          1               1

   EstimatedSalary  Exited
0        101348.88       1
1        112542.58       0
2        113931.57       1
3         93826.63       0
4         79084.10       0
```

[17]:
```
data.head(2)
```

[17]:
```
   RowNumber  CustomerId   Surname  CreditScore Geography  Gender  Age  \
0          1    15634602  Hargrave          619    France  Female   42
1          2    15647311      Hill          608     Spain  Female   41

   Tenure   Balance  NumOfProducts  HasCrCard  IsActiveMember  \
0       2      0.00              1          1               1
1       1  83807.86              1          0               1
```

```
   EstimatedSalary  Exited
0        101348.88       1
1        112542.58       0
```

[18]:
```
data.shape
```

[18]:
```
(10000, 14)
```

[19]:
```
data['Gender'].unique()
```

[19]:
```
array(['Female', 'Male'], dtype=object)
```

[20]:
```
data["Geography"].unique()
```

[20]:
```
array(['France', 'Spain', 'Germany'], dtype=object)
```

[21]:
```
data['Gender']
```

[21]:
```
0       Female
1       Female
2       Female
3       Female
4       Female
         …
9995      Male
9996      Male
9997    Female
9998      Male
9999    Female
Name: Gender, Length: 10000, dtype: object
```

[22]:
```
data['Gender'].value_counts()
```

[22]:
```
Gender
Male      5457
Female    4543
Name: count, dtype: int64
```

[23]:
```
data.dtypes
```

[23]:
```
RowNumber            int64
CustomerId           int64
Surname             object
CreditScore          int64
Geography           object
Gender              object
Age                  int64
```

```
Tenure               int64
Balance            float64
NumOfProducts        int64
HasCrCard            int64
IsActiveMember       int64
EstimatedSalary    float64
Exited               int64
dtype: object
```

[24]:
```
data.describe()
```

[24]:
```
         RowNumber    CustomerId   CreditScore           Age        Tenure  \
count  10000.00000  1.000000e+04  10000.000000  10000.000000  10000.000000
mean    5000.50000  1.569094e+07    650.528800     38.921800      5.012800
std     2886.89568  7.193619e+04     96.653299     10.487806      2.892174
min        1.00000  1.556570e+07    350.000000     18.000000      0.000000
25%     2500.75000  1.562853e+07    584.000000     32.000000      3.000000
50%     5000.50000  1.569074e+07    652.000000     37.000000      5.000000
75%     7500.25000  1.575323e+07    718.000000     44.000000      7.000000
max    10000.00000  1.581569e+07    850.000000     92.000000     10.000000

             Balance  NumOfProducts    HasCrCard  IsActiveMember  \
count   10000.000000   10000.000000  10000.00000    10000.000000
mean    76485.889288       1.530200      0.70550        0.515100
std     62397.405202       0.581654      0.45584        0.499797
min         0.000000       1.000000      0.00000        0.000000
25%         0.000000       1.000000      0.00000        0.000000
50%     97198.540000       1.000000      1.00000        1.000000
75%    127644.240000       2.000000      1.00000        1.000000
max    250898.090000       4.000000      1.00000        1.000000

       EstimatedSalary        Exited
count     10000.000000  10000.000000
mean     100090.239881      0.203700
std       57510.492818      0.402769
min          11.580000      0.000000
25%       51002.110000      0.000000
50%      100193.915000      0.000000
75%      149388.247500      0.000000
max      199992.480000      1.000000
```

[25]:
```
columns_to_drop = ['RowNumber', 'CustomerId', 'Surname']
```

[26]:
```
data
```

[26]:
```
      RowNumber  CustomerId    Surname  CreditScore Geography  Gender  Age  \
0             1    15634602   Hargrave          619    France  Female   42
```

```
1              2  15647311       Hill          608     Spain  Female   41
2              3  15619304       Onio          502    France  Female   42
3              4  15701354       Boni          699    France  Female   39
4              5  15737888   Mitchell          850     Spain  Female   43
…          …        …        …           …           …       …   …
9995        9996  15606229   Obijiaku          771    France    Male   39
9996        9997  15569892  Johnstone          516    France    Male   35
9997        9998  15584532        Liu          709    France  Female   36
9998        9999  15682355  Sabbatini          772   Germany    Male   42
9999       10000  15628319     Walker          792    France  Female   28

      Tenure    Balance  NumOfProducts  HasCrCard  IsActiveMember  \
0          2       0.00              1          1               1
1          1   83807.86              1          0               1
2          8  159660.80              3          1               0
3          1       0.00              2          0               0
4          2  125510.82              1          1               1
…       …         …             …         …             …
9995       5       0.00              2          1               0
9996      10   57369.61              1          1               1
9997       7       0.00              1          0               1
9998       3   75075.31              2          1               0
9999       4  130142.79              1          1               0

      EstimatedSalary  Exited
0           101348.88       1
1           112542.58       0
2           113931.57       1
3            93826.63       0
4            79084.10       0
…                …      …
9995         96270.64       0
9996        101699.77       0
9997         42085.58       1
9998         92888.52       1
9999         38190.78       0

[10000 rows x 14 columns]
```

2 Logistic Regression

[27]:
```python
from sklearn.preprocessing import LabelEncoder
```

[28]:
```python
labelencoder = LabelEncoder()
data['Geography'] = labelencoder.fit_transform(data['Geography'])
data['Gender'] = labelencoder.fit_transform(data['Gender'])
```

[29]:
```
data
```

[29]:
```
      RowNumber  CustomerId    Surname  CreditScore  Geography  Gender  Age  \
0             1    15634602   Hargrave          619          0       0   42
1             2    15647311       Hill          608          2       0   41
2             3    15619304       Onio          502          0       0   42
3             4    15701354       Boni          699          0       0   39
4             5    15737888   Mitchell          850          2       0   43
…           …          …        …           …          …       …   …
9995       9996    15606229   Obijiaku          771          0       1   39
9996       9997    15569892  Johnstone          516          0       1   35
9997       9998    15584532        Liu          709          0       0   36
9998       9999    15682355  Sabbatini          772          1       1   42
9999      10000    15628319     Walker          792          0       0   28

      Tenure    Balance  NumOfProducts  HasCrCard  IsActiveMember  \
0          2       0.00              1          1               1
1          1   83807.86              1          0               1
2          8  159660.80              3          1               0
3          1       0.00              2          0               0
4          2  125510.82              1          1               1
…       …        …               …          …               …
9995       5       0.00              2          1               0
9996      10   57369.61              1          1               1
9997       7       0.00              1          0               1
9998       3   75075.31              2          1               0
9999       4  130142.79              1          1               0

      EstimatedSalary  Exited
0           101348.88       1
1           112542.58       0
2           113931.57       1
3            93826.63       0
4            79084.10       0
…                 …     …
9995         96270.64       0
9996        101699.77       0
9997         42085.58       1
9998         92888.52       1
9999         38190.78       0

[10000 rows x 14 columns]
```

[30]:
```
X=data.drop('Exited', axis=1)
y=data['Exited']
```

[31]:
```
y
```

[31]:
```
0       1
1       0
2       1
3       0
4       0
       ..
9995    0
9996    0
9997    1
9998    1
9999    0
Name: Exited, Length: 10000, dtype: int64
```

[34]:
```
from sklearn.model_selection import train_test_split
```

[35]:
```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2,␣
↪random_state=0)
```

[36]:
```
X_train.shape
```

[36]:
```
(8000, 13)
```

[37]:
```
y_train.shape
```

[37]:
```
(8000,)
```

[38]:
```
X_test.shape
```

[38]:
```
(2000, 13)
```

[39]:
```
y_test.shape
```

[39]:
```
(2000,)
```

[45]:
```
from sklearn.linear_model import LogisticRegression
```

[46]:
```
X_encoded = pd.get_dummies(X)
X_train_encoded, X_test_encoded, y_train, y_test = train_test_split(X_encoded,␣
↪y, test_size=0.2, random_state=0)
logreg.fit(X_train_encoded, y_train)
```

[46]:
```
LogisticRegression()
```

[54]:
```
X_test_aligned = X_test.reindex(columns=X_train_encoded.columns, fill_value=0)
y_pred = logreg.predict(X_test_aligned)
```

[55]:
```
y_pred
```

[55]: array([0, 0, 0, …, 0, 0, 0])

[56]:
```
X_test_aligned = X_test.reindex(columns=X_train_encoded.columns, fill_value=0)
print("Accuracy: ", logreg.score(X_test_aligned, y_test))
```

```
Accuracy:  0.7975
```

3 Accuracy of Logistic Regression Classifier is 0.79.

[ ]:
```
```