assignment-5

April 15, 2024

1 Name: Sourabh Barala

2 Course: M.Sc. Data Science

3 Year: 1st

4 Reg. No.: 23MSD7044

5 Subject: Machine Learning and its Applications

6

7

8

9

10

11

[56]:
```
import pandas as pd
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split
import numpy as np
import seaborn
from sklearn.base import BaseEstimator, TransformerMixin
from sklearn.impute import SimpleImputer
from sklearn.pipeline import Pipeline
from sklearn.preprocessing import StandardScaler,OneHotEncoder
from sklearn.compose import ColumnTransformer
```

[2]:
```
cs_churn=pd.read_csv('churn_prediction.csv')
```

[3]:
```
cs_churn.head(5)
```

[3]:
```
   customer_id  vintage  age gender  dependents     occupation    city  \
0            1     3135   66   Male         0.0  self_employed   187.0
1            2      310   35   Male         0.0  self_employed     NaN
2            4     2356   31   Male         0.0       salaried   146.0
3            5      478   90    NaN         NaN  self_employed  1020.0
4            6     2531   42   Male         2.0  self_employed  1494.0

   customer_nw_category  branch_code  days_since_last_transaction  …  \
0                     2          755                        224.0  …
1                     2         3214                         60.0  …
2                     2           41                          NaN  …
3                     2          582                        147.0  …
4                     3          388                         58.0  …

   previous_month_end_balance  average_monthly_balance_prevQ  \
0                     1458.71                        1458.71
1                     8704.66                        7799.26
2                     5815.29                        4910.17
3                     2291.91                        2084.54
4                     1401.72                        1643.31

   average_monthly_balance_prevQ2  current_month_credit  \
0                         1449.07                  0.20
1                        12419.41                  0.56
2                         2815.94                  0.61
3                         1006.54                  0.47
4                         1871.12                  0.33

   previous_month_credit  current_month_debit  previous_month_debit  \
0                   0.20                 0.20                  0.20
1                   0.56              5486.27                100.56
2                   0.61              6046.73                259.23
3                   0.47                 0.47               2143.33
4                 714.61               588.62               1538.06

   current_month_balance  previous_month_balance  churn
0                1458.71                1458.71      0
1                6496.78                8787.61      0
2                5006.28                5070.14      0
3                2291.91                1669.79      1
4                1157.15                1677.16      1

[5 rows x 21 columns]
```

[4]:
```
cs_churn.info()
```

```
<class 'pandas.core.frame.DataFrame'>
```

```
RangeIndex: 28382 entries, 0 to 28381
Data columns (total 21 columns):
 #   Column                          Non-Null Count  Dtype
---  ------                          --------------  -----
 0   customer_id                     28382 non-null  int64
 1   vintage                         28382 non-null  int64
 2   age                             28382 non-null  int64
 3   gender                          27857 non-null  object
 4   dependents                      25919 non-null  float64
 5   occupation                      28302 non-null  object
 6   city                            27579 non-null  float64
 7   customer_nw_category            28382 non-null  int64
 8   branch_code                     28382 non-null  int64
 9   days_since_last_transaction     25159 non-null  float64
 10  current_balance                 28382 non-null  float64
 11  previous_month_end_balance      28382 non-null  float64
 12  average_monthly_balance_prevQ   28382 non-null  float64
 13  average_monthly_balance_prevQ2  28382 non-null  float64
 14  current_month_credit            28382 non-null  float64
 15  previous_month_credit           28382 non-null  float64
 16  current_month_debit             28382 non-null  float64
 17  previous_month_debit            28382 non-null  float64
 18  current_month_balance           28382 non-null  float64
 19  previous_month_balance          28382 non-null  float64
 20  churn                           28382 non-null  int64
dtypes: float64(13), int64(6), object(2)
memory usage: 4.5+ MB
```

[5]:
```
cs_churn.describe()
```

[5]:

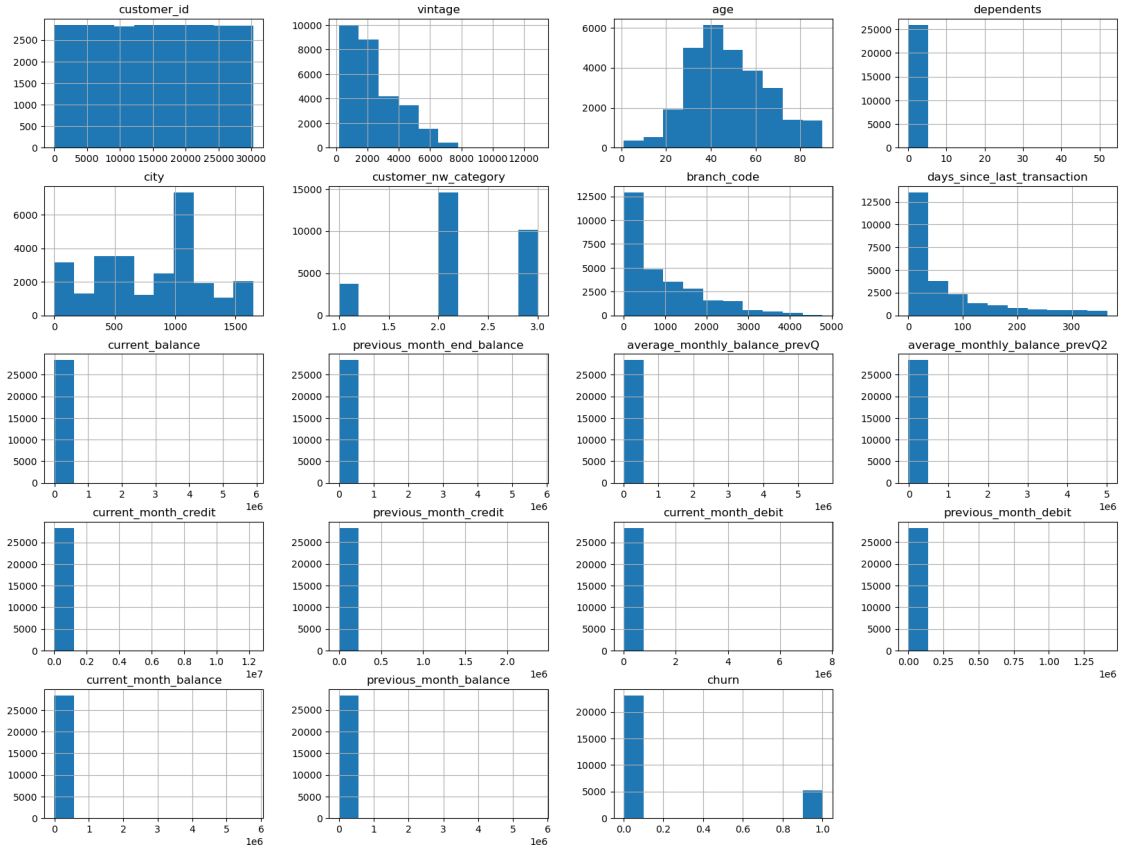
	customer_id	vintage	age	dependents	city
count	28382.000000	28382.000000	28382.000000	25919.000000	27579.000000
mean	15143.508667	2364.336446	48.208336	0.347236	796.109576
std	8746.454456	1610.124506	17.807163	0.997661	432.872102
min	1.000000	180.000000	1.000000	0.000000	0.000000
25%	7557.250000	1121.000000	36.000000	0.000000	409.000000
50%	15150.500000	2018.000000	46.000000	0.000000	834.000000
75%	22706.750000	3176.000000	60.000000	0.000000	1096.000000
max	30301.000000	12899.000000	90.000000	52.000000	1649.000000

	customer_nw_category	branch_code	days_since_last_transaction
count	28382.000000	28382.000000	25159.000000
mean	2.225530	925.975019	69.997814
std	0.660443	937.799129	86.341098
min	1.000000	1.000000	0.000000
25%	2.000000	176.000000	11.000000
50%	2.000000	572.000000	30.000000

```
75%                3.000000   1440.000000                    95.000000
max                3.000000   4782.000000                   365.000000

       current_balance  previous_month_end_balance  \
count     2.838200e+04                2.838200e+04
mean      7.380552e+03                7.495771e+03
std       4.259871e+04                4.252935e+04
min      -5.503960e+03               -3.149570e+03
25%       1.784470e+03                1.906000e+03
50%       3.281255e+03                3.379915e+03
75%       6.635820e+03                6.656535e+03
max       5.905904e+06                5.740439e+06

       average_monthly_balance_prevQ  average_monthly_balance_prevQ2  \
count                   2.838200e+04                    2.838200e+04
mean                    7.496780e+03                    7.124209e+03
std                     4.172622e+04                    4.457581e+04
min                     1.428690e+03                   -1.650610e+04
25%                     2.180945e+03                    1.832507e+03
50%                     3.542865e+03                    3.359600e+03
75%                     6.666887e+03                    6.517960e+03
max                     5.700290e+06                    5.010170e+06

       current_month_credit  previous_month_credit  current_month_debit  \
count          2.838200e+04           2.838200e+04         2.838200e+04
mean           3.433252e+03           3.261694e+03         3.658745e+03
std            7.707145e+04           2.968889e+04         5.198542e+04
min            1.000000e-02           1.000000e-02         1.000000e-02
25%            3.100000e-01           3.300000e-01         4.100000e-01
50%            6.100000e-01           6.300000e-01         9.193000e+01
75%            7.072725e+02           7.492350e+02         1.360435e+03
max            1.226985e+07           2.361808e+06         7.637857e+06

       previous_month_debit  current_month_balance  previous_month_balance  \
count          2.838200e+04           2.838200e+04            2.838200e+04
mean           3.339761e+03           7.451133e+03            7.495177e+03
std            2.430111e+04           4.203394e+04            4.243198e+04
min            1.000000e-02          -3.374180e+03           -5.171920e+03
25%            4.100000e-01           1.996765e+03            2.074407e+03
50%            1.099600e+02           3.447995e+03            3.465235e+03
75%            1.357553e+03           6.667958e+03            6.654693e+03
max            1.414168e+06           5.778185e+06            5.720144e+06

              churn
count  28382.000000
mean       0.185329
std        0.388571
```

```
min         0.000000
25%         0.000000
50%         0.000000
75%         0.000000
max         1.000000
```

[6]:
```
cs_churn.hist(figsize=(20,15))
plt.show()
```



[7]:
```
cs_churn=cs_churn.dropna(subset=['gender','occupation'])
```

[8]:
```
train_cs_churn,test_cs_churn=train_test_split(cs_churn,random_state=100,stratify=cs_churn['chu
↪2)
```

[9]:
```
cs_churn=train_cs_churn.copy()
```

[10]:
```
def give_numeric_cols(data):
    numeric_cols=[]
    for i in range(len(data.dtypes)):
        if (data.dtypes[i]==int) or (data.dtypes[i]==float):
```

```
            numeric_cols.append(data.columns[i])

    return numeric_cols
def give_object_cols(data):
    object_cols=[]
    for i in range(len(data.dtypes)):
        if (data.dtypes[i]==object):
            object_cols.append(data.columns[i])

    return object_cols
object_cols=give_object_cols(cs_churn)
numeric_cols=give_numeric_cols(cs_churn)
```

[11]:
```
numeric_cols
```

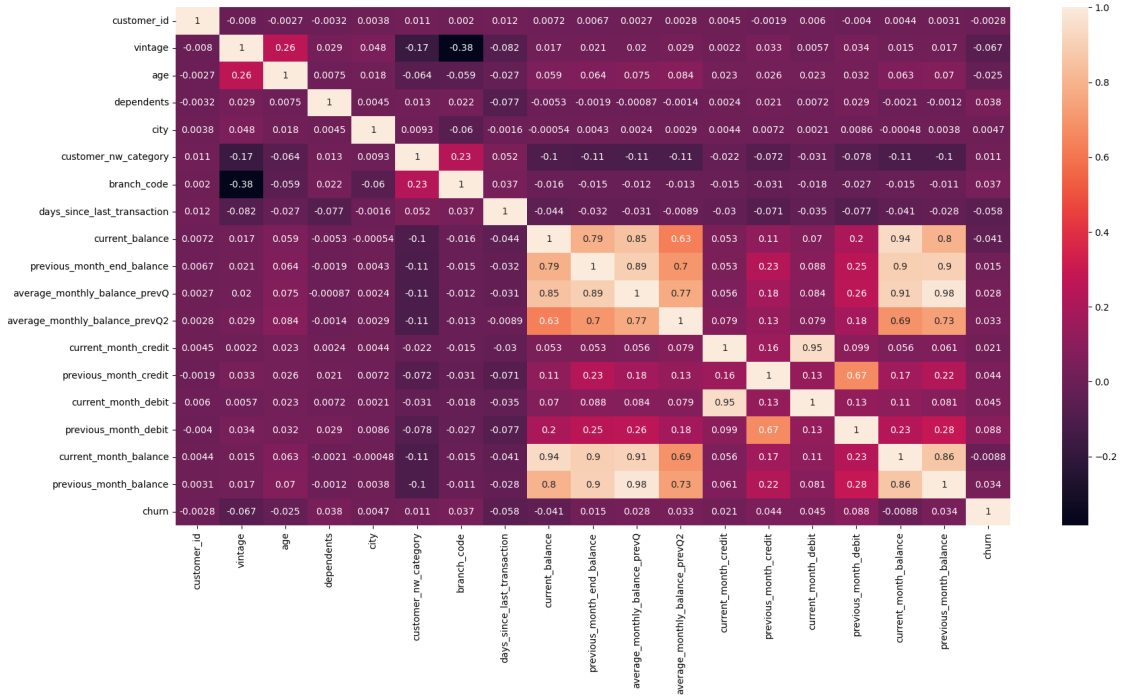
[11]:
```
['customer_id',
 'vintage',
 'age',
 'dependents',
 'city',
 'customer_nw_category',
 'branch_code',
 'days_since_last_transaction',
 'current_balance',
 'previous_month_end_balance',
 'average_monthly_balance_prevQ',
 'average_monthly_balance_prevQ2',
 'current_month_credit',
 'previous_month_credit',
 'current_month_debit',
 'previous_month_debit',
 'current_month_balance',
 'previous_month_balance',
 'churn']
```

[12]:
```
cs_churn[numeric_cols].corr()['churn']
```

[12]:
```
customer_id                    -0.002786
vintage                        -0.066560
age                            -0.024549
dependents                      0.037517
city                            0.004658
customer_nw_category            0.010780
branch_code                     0.036540
days_since_last_transaction    -0.058468
current_balance                -0.040551
previous_month_end_balance      0.015128
```

```
average_monthly_balance_prevQ     0.027654
average_monthly_balance_prevQ2    0.033217
current_month_credit              0.020538
previous_month_credit             0.044486
current_month_debit               0.044859
previous_month_debit              0.087918
current_month_balance            -0.008767
previous_month_balance            0.033667
churn                             1.000000
Name: churn, dtype: float64
```

[13]:
```
plt.figure(figsize=(20,10))
seaborn.heatmap(cs_churn[numeric_cols].corr(),annot=True)
plt.show()
```



[ ]:
```
```

[33]:
```
train_parameters=train_cs_churn.drop(['churn','customer_id'],axis=1)
train_lebels=train_cs_churn['churn']
test_parameters=test_cs_churn.drop(['churn','customer_id'],axis=1)
test_lebels=test_cs_churn['churn']
```

[15]:
```
train_parameters.isnull().sum()
```

[15]:
```
vintage                          0
age                              0
gender                           0
dependents                    1772
occupation                       0
city                           630
customer_nw_category             0
branch_code                      0
days_since_last_transaction   2515
current_balance                  0
previous_month_end_balance       0
average_monthly_balance_prevQ    0
average_monthly_balance_prevQ2   0
current_month_credit             0
previous_month_credit            0
current_month_debit              0
previous_month_debit             0
current_month_balance            0
previous_month_balance           0
dtype: int64
```

[ ]:

[65]:
```
class CleanNumericData(BaseEstimator,TransformerMixin):
    def __init__(self,method='median',fill_value=0):
        self.method=method
        self.fill_value=fill_value

    def fit(self,data):
        return self

    def transform(self,data):
        data=data.dropna(subset=['gender','occupation'])
        imputer=SimpleImputer(strategy=self.method)
        num_cols=give_numeric_cols(data)
        X=imputer.fit_transform(data[num_cols])
        X_df=pd.DataFrame(X,columns=num_cols,index=data.index)

        from sklearn.preprocessing import StandardScaler
        scaler=StandardScaler()
        X=scaler.fit_transform(X_df)

        import numpy as np

        X=np.around(X,decimals=4)
        X_df=pd.DataFrame(X,columns=num_cols,index=data.index)
        return X_df
```

```
class encode_category(BaseEstimator,TransformerMixin):
    def __init__(self,method=0):
        self.method=method

    def fit(self,data):
        return self

    def transform(self,data):
        data=data.dropna(subset=['gender','occupation'])
        encoder=OneHotEncoder()
        cat_cols=give_object_cols(data)
        X=encoder.fit_transform(data[cat_cols])
        X_df=pd.DataFrame(X.toarray(), columns=encoder.
 ↪get_feature_names_out(cat_cols),index=data.index)
        return X_df


num_pipeline=Pipeline([
    ('cleaner',CleanNumericData()),
    # ('standard_scaler',StandardScaler())
])

cat_pipeline=Pipeline([
    ('cat',encode_category())
])

full_pipeline=ColumnTransformer([
    ('num',num_pipeline, train_parameters.columns),
    ('cat',cat_pipeline,give_object_cols(train_parameters)),
])


def transform_data(data):

    indexes=num_pipeline.fit_transform(data).index

    transformed=full_pipeline.fit_transform(data)
   ␣
 ↪col_names=['vintage','age','dependents','city','customer_nw_category','branch_code','days_s
 ↪'current_balance','previous_month_end_balance','average_monthly_balance_prevQ','average_mon
 ↪'gender_Male','occupation_company','occupation_retired',␣
 ↪'occupation_salaried', 'occupation_self_employed','occupation_student']
    transformed_df=pd.DataFrame(transformed, columns=col_names,index=indexes)
    return transformed_df
```

[66]:
```
cc=cat_pipeline.fit_transform(train_parameters)
cc
```

[66]:

	gender_Female	gender_Male	occupation_company	occupation_retired	\
13401	1.0	0.0	0.0	0.0	
13998	0.0	1.0	0.0	0.0	
27572	0.0	1.0	0.0	0.0	
6449	0.0	1.0	0.0	1.0	
15774	1.0	0.0	0.0	0.0	
…	…	…	…	…	
565	0.0	1.0	0.0	0.0	
8565	1.0	0.0	0.0	0.0	
12346	1.0	0.0	0.0	0.0	
11376	1.0	0.0	0.0	0.0	
3294	1.0	0.0	0.0	0.0	

	occupation_salaried	occupation_self_employed	occupation_student
13401	0.0	1.0	0.0
13998	0.0	1.0	0.0
27572	1.0	0.0	0.0
6449	0.0	0.0	0.0
15774	0.0	1.0	0.0
…	…	…	…
565	0.0	1.0	0.0
8565	0.0	1.0	0.0
12346	0.0	1.0	0.0
11376	0.0	1.0	0.0
3294	0.0	1.0	0.0

[22224 rows x 7 columns]

[67]:
```
nn=num_pipeline.fit_transform(train_parameters)
nn
```

[67]:

	vintage	age	dependents	city	customer_nw_category	branch_code	\
13401	-1.1129	-0.2605	-0.3238	-1.7727	1.1671	0.5720	
13998	-1.2970	0.0326	-0.3238	0.2502	-0.3509	1.3938	
27572	1.1742	-1.0226	-0.3238	1.0131	-0.3509	-0.8775	
6449	-0.1898	1.1464	-0.3238	0.5792	-0.3509	-0.8796	
15774	-0.5941	0.9119	-0.3238	-1.5207	-0.3509	1.5633	
…	…	…	…	…	…	…	
565	0.9379	0.0326	-0.3238	0.4625	-0.3509	0.4047	
8565	0.1330	-0.7881	-0.3238	1.8461	-0.3509	-0.5791	
12346	0.4589	-0.2019	-0.3238	-1.1731	-1.8688	-0.2732	
11376	0.3239	-1.0812	-0.3238	-0.9071	-1.8688	-0.7571	
3294	0.4757	0.2085	-0.3238	0.8941	-0.3509	-0.4799	

```
       days_since_last_transaction  current_balance  \
13401                      -0.7477          -0.0336
13998                       1.7324          -0.1936
27572                      -0.7111          -0.2879
6449                       -0.4301          -0.1997
15774                      -0.6256          -0.0509
…                            …              …
565                        -0.1980           0.1318
8565                       -0.5645          -0.2905
12346                      -0.4301          -0.2410
11376                      -0.5889           2.7519
3294                       -0.4301          -0.2466

       previous_month_end_balance  average_monthly_balance_prevQ  \
13401                     -0.0219                        -0.0589
13998                     -0.1838                        -0.1892
27572                     -0.2708                        -0.2227
6449                      -0.1920                        -0.2037
15774                     -0.0993                         0.0649
…                           …                            …
565                        0.2713                         0.2639
8565                      -0.2450                        -0.2346
12346                     -0.2370                        -0.2519
11376                      2.6231                         2.7842
3294                      -0.2424                        -0.2581

       average_monthly_balance_prevQ2  current_month_credit  \
13401                         -0.1728                0.0559
13998                         -0.1593               -0.0402
27572                         -0.2624               -0.0269
6449                          -0.1931               -0.0402
15774                         -0.1992               -0.0069
…                               …                   …
565                            0.0513               -0.0402
8565                          -0.1210               -0.0235
12346                         -0.2554               -0.0402
11376                          4.5911               -0.0339
3294                          -0.2633               -0.0402

       previous_month_credit  current_month_debit  previous_month_debit  \
13401                -0.0599               0.1141               -0.1294
13998                -0.1106              -0.0631               -0.1543
27572                -0.0423              -0.0333                0.2107
6449                 -0.1106              -0.0590               -0.1487
15774                -0.1106              -0.0302                0.1260
…                      …                  …                        …
565                  -0.0802              -0.0538               -0.1473
```

```
8565                  0.0416              -0.0195                0.0699
12346                -0.1106              -0.0631               -0.1543
11376                -0.0916              -0.0631               -0.1543
3294                 -0.1106              -0.0631               -0.1544

       current_month_balance  previous_month_balance
13401                 0.0098                -0.0757
13998                -0.1931                -0.1780
27572                -0.3000                -0.2607
6449                 -0.2017                -0.1910
15774                -0.1121                 0.1162
…                      …                    …
565                   0.2429                 0.2569
8565                 -0.2799                -0.1510
12346                -0.2477                -0.2364
11376                 2.7802                 2.5942
3294                 -0.2534                -0.2417

[22224 rows x 17 columns]
```

[68]:
```
transformed_train=transform_data(train_parameters)
```

[69]:
```
transformed_test=transform_data(test_parameters)
```

[70]:
```
from sklearn.linear_model import LogisticRegression
```

[71]:
```
model=LogisticRegression(max_iter=10000)
```

[72]:
```
model.fit(transformed_train,train_lebels)
```

[72]:
```
LogisticRegression(max_iter=10000)
```

[74]:
```
pred=model.predict(transformed_test)
```

[78]:
```
from sklearn.metrics import recall_score,accuracy_score
```

11.1 Recall

[79]:
```
recall_score(y_true=test_lebels,y_pred=pred)
```

[79]:
```
0.13702623906705538
```

11.2 Accuracy

[80]:
```
accuracy_score(y_true=test_lebels,y_pred=pred)
```

[80]: 0.8294043548677343

[ ]: