Python 3.9.1 (tags/v3.9.1:1e5d33e, Dec 7 2020, 17:08:21) [MSC v.1927 64 bit (AMD64)] on win32

Type "help", "copyright", "credits" or "license()" for more information.

>>>

============= RESTART: C:\Morphy\BankChurners\Builders\BankChurners\_Build\_Models\_V2.py =============

1. Information of Raw Data:

1.1. Data columns:

Index(['Gender', 'Education\_Level', 'Marital\_Status', 'Income\_Category',

'Card\_Category', 'Customer\_Age', 'Dependent\_count', 'Months\_on\_book',

'Total\_Relationship\_Count', 'Months\_Inactive\_12\_mon',

'Contacts\_Count\_12\_mon', 'Credit\_Limit', 'Total\_Revolving\_Bal',

'Avg\_Open\_To\_Buy', 'Total\_Trans\_Amt', 'Total\_Trans\_Ct',

'Total\_Amt\_Chng\_Q4\_Q1', 'Total\_Ct\_Chng\_Q4\_Q1', 'Avg\_Utilization\_Ratio',

'Attrition\_Flag'],

dtype='object')

1.2. Data statistaics:

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

RangeIndex: 10127 entries, 0 to 10126

Data columns (total 20 columns):

# Column Non-Null Count Dtype

--- ------ -------------- -----

0 Gender 10127 non-null object

1 Education\_Level 10127 non-null object

2 Marital\_Status 10127 non-null object

3 Income\_Category 10127 non-null object

4 Card\_Category 10127 non-null object

5 Customer\_Age 10127 non-null int64

6 Dependent\_count 10127 non-null int64

7 Months\_on\_book 10127 non-null int64

8 Total\_Relationship\_Count 10127 non-null int64

9 Months\_Inactive\_12\_mon 10127 non-null int64

10 Contacts\_Count\_12\_mon 10127 non-null int64

11 Credit\_Limit 10127 non-null float64

12 Total\_Revolving\_Bal 10127 non-null int64

13 Avg\_Open\_To\_Buy 10127 non-null float64

14 Total\_Trans\_Amt 10127 non-null int64

15 Total\_Trans\_Ct 10127 non-null int64

16 Total\_Amt\_Chng\_Q4\_Q1 10127 non-null float64

17 Total\_Ct\_Chng\_Q4\_Q1 10127 non-null float64

18 Avg\_Utilization\_Ratio 10127 non-null float64

19 Attrition\_Flag 10127 non-null object

dtypes: float64(5), int64(9), object(6)

memory usage: 1.5+ MB

None

1.3. Null data:

Gender 0

Education\_Level 0

Marital\_Status 0

Income\_Category 0

Card\_Category 0

Customer\_Age 0

Dependent\_count 0

Months\_on\_book 0

Total\_Relationship\_Count 0

Months\_Inactive\_12\_mon 0

Contacts\_Count\_12\_mon 0

Credit\_Limit 0

Total\_Revolving\_Bal 0

Avg\_Open\_To\_Buy 0

Total\_Trans\_Amt 0

Total\_Trans\_Ct 0

Total\_Amt\_Chng\_Q4\_Q1 0

Total\_Ct\_Chng\_Q4\_Q1 0

Avg\_Utilization\_Ratio 0

Attrition\_Flag 0

dtype: int64

3. Feature preprocessing:

3.1. Data statistaics after encoding:

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

RangeIndex: 10127 entries, 0 to 10126

Data columns (total 20 columns):

# Column Non-Null Count Dtype

--- ------ -------------- -----

0 Gender 10127 non-null int64

1 Education\_Level 10127 non-null int64

2 Marital\_Status 10127 non-null int64

3 Income\_Category 10127 non-null int64

4 Card\_Category 10127 non-null int64

5 Customer\_Age 10127 non-null int64

6 Dependent\_count 10127 non-null int64

7 Months\_on\_book 10127 non-null int64

8 Total\_Relationship\_Count 10127 non-null int64

9 Months\_Inactive\_12\_mon 10127 non-null int64

10 Contacts\_Count\_12\_mon 10127 non-null int64

11 Credit\_Limit 10127 non-null float64

12 Total\_Revolving\_Bal 10127 non-null int64

13 Avg\_Open\_To\_Buy 10127 non-null float64

14 Total\_Trans\_Amt 10127 non-null int64

15 Total\_Trans\_Ct 10127 non-null int64

16 Total\_Amt\_Chng\_Q4\_Q1 10127 non-null float64

17 Total\_Ct\_Chng\_Q4\_Q1 10127 non-null float64

18 Avg\_Utilization\_Ratio 10127 non-null float64

19 Attrition\_Flag 10127 non-null int64

dtypes: float64(5), int64(15)

memory usage: 1.5 MB

None

2.2. Data columes:

Index(['Gender', 'Education\_Level', 'Marital\_Status', 'Income\_Category',

'Card\_Category', 'Customer\_Age', 'Dependent\_count', 'Months\_on\_book',

'Total\_Relationship\_Count', 'Months\_Inactive\_12\_mon',

'Contacts\_Count\_12\_mon', 'Credit\_Limit', 'Total\_Revolving\_Bal',

'Avg\_Open\_To\_Buy', 'Total\_Trans\_Amt', 'Total\_Trans\_Ct',

'Total\_Amt\_Chng\_Q4\_Q1', 'Total\_Ct\_Chng\_Q4\_Q1', 'Avg\_Utilization\_Ratio',

'Attrition\_Flag'],

dtype='object')

2.3. Data info:

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

RangeIndex: 10127 entries, 0 to 10126

Data columns (total 20 columns):

# Column Non-Null Count Dtype

--- ------ -------------- -----

0 Gender 10127 non-null int64

1 Education\_Level 10127 non-null int64

2 Marital\_Status 10127 non-null int64

3 Income\_Category 10127 non-null int64

4 Card\_Category 10127 non-null int64

5 Customer\_Age 10127 non-null int64

6 Dependent\_count 10127 non-null int64

7 Months\_on\_book 10127 non-null int64

8 Total\_Relationship\_Count 10127 non-null int64

9 Months\_Inactive\_12\_mon 10127 non-null int64

10 Contacts\_Count\_12\_mon 10127 non-null int64

11 Credit\_Limit 10127 non-null float64

12 Total\_Revolving\_Bal 10127 non-null int64

13 Avg\_Open\_To\_Buy 10127 non-null float64

14 Total\_Trans\_Amt 10127 non-null int64

15 Total\_Trans\_Ct 10127 non-null int64

16 Total\_Amt\_Chng\_Q4\_Q1 10127 non-null float64

17 Total\_Ct\_Chng\_Q4\_Q1 10127 non-null float64

18 Avg\_Utilization\_Ratio 10127 non-null float64

19 Attrition\_Flag 10127 non-null int64

dtypes: float64(5), int64(15)

memory usage: 1.5 MB

2.3. Null data:

Gender 0

Education\_Level 0

Marital\_Status 0

Income\_Category 0

Card\_Category 0

Customer\_Age 0

Dependent\_count 0

Months\_on\_book 0

Total\_Relationship\_Count 0

Months\_Inactive\_12\_mon 0

Contacts\_Count\_12\_mon 0

Credit\_Limit 0

Total\_Revolving\_Bal 0

Avg\_Open\_To\_Buy 0

Total\_Trans\_Amt 0

Total\_Trans\_Ct 0

Total\_Amt\_Chng\_Q4\_Q1 0

Total\_Ct\_Chng\_Q4\_Q1 0

Avg\_Utilization\_Ratio 0

Attrition\_Flag 0

dtype: int64

3. Set Global Parameters

3.1. Split data set into train and test sets:

3.2. Set training and model parameters:

4. Build Models

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Model Name | Model Type | Model Parameter | Accuracy | Cross-Validation Score | Accuracy on test data | Time |
| LS | Logistic Regression | Trained on whole data set | 89.405% | 88.891% | 0.8923988153998026 | 6.251536130905151 |
| LS2 | Logistic Regression | -With Train Test Split | 89.280% | 88.863% | 0.8943731490621916 | 5.980744123458862 |
| KNN | K-Nearest Neighbors | Trained on whole data set | 100.000% | 88.022% | 0.983218163869694 | 10.795233964920044 |
| KNN2 | K-Nearest Neighbors | with Train Test Split | 100.000% | 88.084% | 0.8756169792694966 | 8.581111669540405 |
| DT | Decision Tree | Trained on whole data set | 97.413% | 94.697% | 0.9684106614017769 | 1.0219228267669678 |
| DT2 | Decision Tree | with Train Test Split | 100.000% | 93.406% | 0.9496544916090819 | 1.3281207084655762 |
| RF | Random Forest | Trained on whole data set | 100.000% | 96.396% | 0.9970384995064165 | 46.09694957733154 |
| RF2 | Random Forest | Using Train Test Split | 100.000% | 96.204% | 0.9644619940769991 | 41.78449773788452 |
| NB | Naive Bayes | Trained on whole data set | 89.612% | 89.553% | 0.9032576505429417 | 0.3125007152557373 |
| NB2 | Naive Bayes | Using Train Test split | 89.598% | 89.302% | 0.8963474827245804 | 0.29515743255615234 |
| SVM | Supportting vector machine | Trained on whole data set | 83.934% | 83.934% | 0.8420533070088845 | 61.05700063705444 |
| SVM2 | Surportting vector machine | Using Train Test split | 83.904% | 83.904% | 0.8420533070088845 | 52.801997423172 |
| SV | Soft voting | Trained on whole data set; LR+KNN+DT | 98.647% | 93.621% | 0.9842053307008884 | 21.265832662582397 |
| SV2 | Soft Voting | LR+KNN+DT; using Train Test Split | 98.782% | 93.307% | 0.9397828232971372 | 16.950594186782837 |
| HV | Hard Voting | Trained on whole data set | 98.460% | 92.090% | 0.9772951628825272 | 19.379376649856567 |
| HV2 | Hard Voting | Using Train Test Split | 98.552% | 92.001% | 0.9210266535044422 | 16.326966047286987 |
| Stack1 | Stacking | Using KNN as Meta; Estimators = lr2, knn2, dtree2, hard\_vote2 | 100.000% | 93.976% | 0.9466929911154985 | 160.24689269065857 |
| Stack2 | Stacking | Using Logistic Regression as Meta; Estimators = knn, dtree, soft\_vote, rf | 58.349% | 83.490% | 0.5923000987166831 | 382.2018871307373 |
| Stack3 | Stacking | Using Logistic Regression as Meta: estimators = knn2, dtree, soft\_vote2, hard\_vote2 | 80.546% | 91.068% | 0.9200394866732478 | 204.43393087387085 |