In [1]:
```python
import pandas as pd
import matplotlib.pyplot as plt
```

Machine Learning Algorithms:

```
Decision Tree
Logistic Regression
```

Experiment 1: Data Splitting with the hybrid oversampling and undersampling- In the first experiment, we split the dataset into training (80%) and testing (20%) subsets, maintaining class balance through stratification.

Experiment 2: Stratified Cross-Validation with SMOTE (Oversampling)
For the third experiment, we enhance our stratified cross-validation by incorporating
Synthetic Minority Over-sampling Technique (SMOTE). SMOTE is used to address class\ imbalance by generating synthetic samples, improving the model's ability to learn from the minority class.

In [2]:
```python
data = pd.read_csv('credit_card_churn.csv')
```

In [3]:
```python
data.head(2)
```

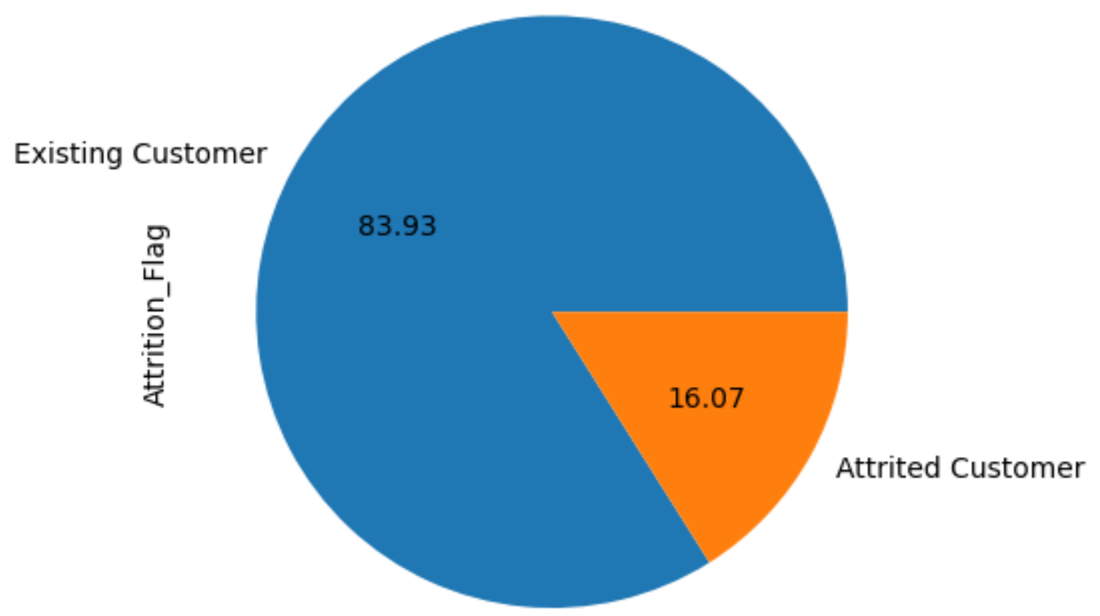
Out[3]:

	CLIENTNUM	Attrition_Flag	Customer_Age	Gender	Dependent_count	Education_Level	Marital_Status	Income_Category	Card_Category	Months_on_book	...	Months_Inactive_12_mon	Contacts_Count_12_mon	Credit_Limit	Total_Revo
0	768805383	Existing Customer	45	M	3	High School	Married	$60K-$80K	Blue	39	...	1	3	12691.0	
1	818770008	Existing Customer	49	F	5	Graduate	Single	Less than $40K	Blue	44	...	1	2	8256.0	

2 rows × 21 columns

In [4]:
```python
data['Attrition_Flag'].value_counts().plot.pie(autopct='%.2f')
#The target variable is not balance
```

Out[4]:
```
<AxesSubplot:ylabel='Attrition_Flag'>
```



In [5]:
```python
Finaldata = data[['Attrition_Flag', 'Gender', 'Income_Category', 'Total_Trans_Ct', 'Avg_Utilization_Ratio', 'Total_Revolving_Bal', 'Months_Inactive_12_mon', 'Total_Trans_Amt', 'Total_Amt_Chng
```

In [6]:
```
Finaldata.dtypes
```

Out[6]:
```
Attrition_Flag              object
Gender                      object
Income_Category             object
Total_Trans_Ct               int64
Avg_Utilization_Ratio      float64
Total_Revolving_Bal          int64
Months_Inactive_12_mon       int64
Total_Trans_Amt              int64
Total_Amt_Chng_Q4_Q1       float64
dtype: object
```

In [7]:
```
from sklearn.preprocessing import LabelEncoder
le = LabelEncoder()
Finaldata['Gender'] = le.fit_transform(data['Gender'])
Finaldata['Income_Category'] = le.fit_transform(data['Income_Category'])
Finaldata['Attrition_Flag'] = le.fit_transform(data['Attrition_Flag'])
```

```
C:\Users\Romelio Villar Jr\AppData\Local\Temp\ipykernel_6900\320870704.py:3: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy (https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy)
  Finaldata['Gender'] = le.fit_transform(data['Gender'])
C:\Users\Romelio Villar Jr\AppData\Local\Temp\ipykernel_6900\320870704.py:4: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy (https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy)
  Finaldata['Income_Category'] = le.fit_transform(data['Income_Category'])
C:\Users\Romelio Villar Jr\AppData\Local\Temp\ipykernel_6900\320870704.py:5: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy (https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy)
  Finaldata['Attrition_Flag'] = le.fit_transform(data['Attrition_Flag'])
```

In [8]:
```
Finaldata.head(2)
```

Out[8]:

	Attrition_Flag	Gender	Income_Category	Total_Trans_Ct	Avg_Utilization_Ratio	Total_Revolving_Bal	Months_Inactive_12_mon	Total_Trans_Amt	Total_Amt_Chng_Q4_Q1
0	1	1	2	42	0.061	777	1	1144	1.335
1	1	0	4	33	0.105	864	1	1291	1.541

In [9]:
```
Finaldata.dtypes
```

Out[9]:
```
Attrition_Flag               int32
Gender                       int32
Income_Category              int32
Total_Trans_Ct               int64
Avg_Utilization_Ratio      float64
Total_Revolving_Bal          int64
Months_Inactive_12_mon       int64
Total_Trans_Amt              int64
Total_Amt_Chng_Q4_Q1       float64
dtype: object
```

In [10]:
```
X = Finaldata.drop('Attrition_Flag', axis = 1)
y = Finaldata['Attrition_Flag']
```

Experiment 1: Data Splitting training (80%) and testing (20%) subsets, stratify=yes, hyperparameters

In [11]:
```
import numpy as np
from sklearn.model_selection import StratifiedKFold, GridSearchCV, train_test_split, cross_val_score
from sklearn.tree import DecisionTreeClassifier
from sklearn.linear_model import LogisticRegression
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import accuracy_score, roc_auc_score, precision_score, recall_score, f1_score, confusion_matrix
from imblearn.over_sampling import RandomOverSampler, SMOTE
from imblearn.under_sampling import RandomUnderSampler
```

In [12]:
```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42, stratify=y)
```

In [13]:
```
print(y_train.value_counts(normalize=True)*100)
```

```
1    83.92791
0    16.07209
Name: Attrition_Flag, dtype: float64
```

In [14]:
```
print(y_test.value_counts(normalize=True)*100)
```

```
1    83.958539
0    16.041461
Name: Attrition_Flag, dtype: float64
```

Decision Tree

Experiment 1

In [15]:
```
dt_classifier = DecisionTreeClassifier(random_state=42)
```

In [16]:
```
param_grid_dt = {
    'criterion': ['gini', 'entropy'],
    'max_depth': [None, 10, 20, 30],
    'min_samples_split': [2, 5, 10],
    'min_samples_leaf': [1, 2, 4],
    'max_features': ['auto', 'sqrt', 'log2']
}
```

In [17]:
```
clf = GridSearchCV(dt_classifier, param_grid_dt)
clf
```

Out[17]:
```
GridSearchCV(estimator=DecisionTreeClassifier(random_state=42),
             param_grid={'criterion': ['gini', 'entropy'],
                         'max_depth': [None, 10, 20, 30],
                         'max_features': ['auto', 'sqrt', 'log2'],
                         'min_samples_leaf': [1, 2, 4],
                         'min_samples_split': [2, 5, 10]})
```

In a Jupyter environment, please rerun this cell to show the HTML representation or trust the notebook.
On GitHub, the HTML representation is unable to render, please try loading this page with nbviewer.org.

```
In [18]: clf.fit(X_train, y_train)
```

```
C:\Users\Romelio Villar Jr\anaconda3\lib\site-packages\sklearn\tree\_classes.py:269: FutureWarning: `max_features='auto'` has been deprecated in 1.1 and will be removed in 1.3. To keep the
past behaviour, explicitly set `max_features='sqrt'`.
  warnings.warn(
C:\Users\Romelio Villar Jr\anaconda3\lib\site-packages\sklearn\tree\_classes.py:269: FutureWarning: `max_features='auto'` has been deprecated in 1.1 and will be removed in 1.3. To keep the
past behaviour, explicitly set `max_features='sqrt'`.
  warnings.warn(
C:\Users\Romelio Villar Jr\anaconda3\lib\site-packages\sklearn\tree\_classes.py:269: FutureWarning: `max_features='auto'` has been deprecated in 1.1 and will be removed in 1.3. To keep the
past behaviour, explicitly set `max_features='sqrt'`.
  warnings.warn(
C:\Users\Romelio Villar Jr\anaconda3\lib\site-packages\sklearn\tree\_classes.py:269: FutureWarning: `max_features='auto'` has been deprecated in 1.1 and will be removed in 1.3. To keep the
past behaviour, explicitly set `max_features='sqrt'`.
  warnings.warn(
C:\Users\Romelio Villar Jr\anaconda3\lib\site-packages\sklearn\tree\_classes.py:269: FutureWarning: `max_features='auto'` has been deprecated in 1.1 and will be removed in 1.3. To keep the
past behaviour, explicitly set `max_features='sqrt'`.
  warnings.warn(
C:\Users\Romelio Villar Jr\anaconda3\lib\site-packages\sklearn\tree\_classes.py:269: FutureWarning: `max_features='auto'` has been deprecated in 1.1 and will be removed in 1.3. To keep the
past behaviour, explicitly set `max_features='sqrt'`.
  warnings.warn(
C:\Users\Romelio Villar Jr\anaconda3\lib\site-packages\sklearn\tree\_classes.py:269: FutureWarning: `max_features='auto'` has been deprecated in 1.1 and will be removed in 1.3. To keep the
past behaviour, explicitly set `max_features='sqrt'`.
```

```
In [19]: #Get the best Decision Tree classifier with optimized Hyperparameters
         best_dt_classifier = clf.best_estimator_
         best_dt_classifier
```

```
Out[19]: DecisionTreeClassifier(max_depth=10, max_features='log2', min_samples_leaf=2,
                                min_samples_split=10, random_state=42)
```

In a Jupyter environment, please rerun this cell to show the HTML representation or trust the notebook.
On GitHub, the HTML representation is unable to render, please try loading this page with nbviewer.org.

```
In [20]: # Now, apply RandomOverSampler to the training data
         ros = RandomOverSampler(sampling_strategy=0.5)  # Adjust the sampling strategy as needed
         X_train_resampled, y_train_resampled = ros.fit_resample(X_train, y_train)
```

```
In [21]: best_dt_classifier.fit(X_train_resampled, y_train_resampled)
```

```
Out[21]: DecisionTreeClassifier(max_depth=10, max_features='log2', min_samples_leaf=2,
                                min_samples_split=10, random_state=42)
```

In a Jupyter environment, please rerun this cell to show the HTML representation or trust the notebook.
On GitHub, the HTML representation is unable to render, please try loading this page with nbviewer.org.

```
In [22]: y_pred_d1 = best_dt_classifier.predict(X_test)
```

In [26]:

```
accuracy = accuracy_score(y_test, y_pred_d1)
precision = precision_score(y_test, y_pred_d1)
recall = recall_score(y_test,y_pred_d1)
f1 = f1_score(y_test, y_pred_d1)
confusion = confusion_matrix(y_test, y_pred_d1)


print("Accuracy:", accuracy)
print("Precision:", precision)
print("Recall:", recall)
print("f1_score:", f1_score)
print("confusion matrix\n", confusion)
```

```
Accuracy: 0.914116485686081
Precision: 0.9607724803862402
Recall: 0.9359200470311582
f1_score: <function f1_score at 0x000001C107307CA0>
confusion matrix
 [[ 260   65]
 [ 109 1592]]
```

Undersampling

In [27]:

```
# Now, apply RandomOverSampler to the training data
rus = RandomUnderSampler(sampling_strategy=0.5)  # Adjust the sampling strategy as needed
X_train_resampled_U, y_train_resampled_U = rus.fit_resample(X_train, y_train)
```

In [28]:

```
best_dt_classifier.fit(X_train_resampled_U, y_train_resampled_U)
```

Out[28]:

```
DecisionTreeClassifier(max_depth=10, max_features='log2', min_samples_leaf=2,
                       min_samples_split=10, random_state=42)
```

In a Jupyter environment, please rerun this cell to show the HTML representation or trust the notebook.
On GitHub, the HTML representation is unable to render, please try loading this page with nbviewer.org.

In [29]:

```
y_pred_d2 = best_dt_classifier.predict(X_test)
```

In [31]:

```
accuracy = accuracy_score(y_test, y_pred_d2)
precision = precision_score(y_test, y_pred_d2)
recall = recall_score(y_test,y_pred_d2)
f1 = f1_score(y_test, y_pred_d2)
confusion = confusion_matrix(y_test, y_pred_d2)


print("Accuracy:", accuracy)
print("Precision:", precision)
print("Recall:", recall)
print("f1_score:", f1_score)
print("confusion matrix\n", confusion)
```

```
Accuracy: 0.9106614017769002
Precision: 0.9679802955665024
Recall: 0.9241622574955908
f1_score: <function f1_score at 0x000001C107307CA0>
confusion matrix
 [[ 273   52]
 [ 129 1572]]
```

Experiment 2: Decision Tree with Stratified Cross Validation

In [32]:

```
cv = StratifiedKFold(n_splits=5, shuffle=True, random_state=42)
```

In [33]:

```
clf_S = GridSearchCV(dt_classifier, param_grid_dt, cv=cv)
clf_S
```

Out[33]:

```
GridSearchCV(cv=StratifiedKFold(n_splits=5, random_state=42, shuffle=True),
             estimator=DecisionTreeClassifier(random_state=42),
             param_grid={'criterion': ['gini', 'entropy'],
                         'max_depth': [None, 10, 20, 30],
                         'max_features': ['auto', 'sqrt', 'log2'],
                         'min_samples_leaf': [1, 2, 4],
                         'min_samples_split': [2, 5, 10]})
```

In a Jupyter environment, please rerun this cell to show the HTML representation or trust the notebook.
On GitHub, the HTML representation is unable to render, please try loading this page with nbviewer.org.

In [34]:

```
clf_S.fit(X_train, y_train)
```

```
C:\Users\Romelio Villar Jr\anaconda3\lib\site-packages\sklearn\tree\_classes.py:269: FutureWarning: `max_features='auto'` has been deprecated in 1.1 and will be removed in 1.3. To keep the
past behaviour, explicitly set `max_features='sqrt'`.
  warnings.warn(
C:\Users\Romelio Villar Jr\anaconda3\lib\site-packages\sklearn\tree\_classes.py:269: FutureWarning: `max_features='auto'` has been deprecated in 1.1 and will be removed in 1.3. To keep the
past behaviour, explicitly set `max_features='sqrt'`.
  warnings.warn(
C:\Users\Romelio Villar Jr\anaconda3\lib\site-packages\sklearn\tree\_classes.py:269: FutureWarning: `max_features='auto'` has been deprecated in 1.1 and will be removed in 1.3. To keep the
past behaviour, explicitly set `max_features='sqrt'`.
  warnings.warn(
C:\Users\Romelio Villar Jr\anaconda3\lib\site-packages\sklearn\tree\_classes.py:269: FutureWarning: `max_features='auto'` has been deprecated in 1.1 and will be removed in 1.3. To keep the
past behaviour, explicitly set `max_features='sqrt'`.
  warnings.warn(
C:\Users\Romelio Villar Jr\anaconda3\lib\site-packages\sklearn\tree\_classes.py:269: FutureWarning: `max_features='auto'` has been deprecated in 1.1 and will be removed in 1.3. To keep the
past behaviour, explicitly set `max_features='sqrt'`.
  warnings.warn(
C:\Users\Romelio Villar Jr\anaconda3\lib\site-packages\sklearn\tree\_classes.py:269: FutureWarning: `max_features='auto'` has been deprecated in 1.1 and will be removed in 1.3. To keep the
past behaviour, explicitly set `max_features='sqrt'`.
  warnings.warn(
C:\Users\Romelio Villar Jr\anaconda3\lib\site-packages\sklearn\tree\_classes.py:269: FutureWarning: `max_features='auto'` has been deprecated in 1.1 and will be removed in 1.3. To keep the
past behaviour, explicitly set `max_features='sqrt'`.
```

In [38]:

```
best_dt_classifier = clf_S.best_estimator_
best_dt_classifier
```

Out[38]:

```
DecisionTreeClassifier(criterion='entropy', max_depth=10, max_features='log2',
                       random_state=42)
```

In a Jupyter environment, please rerun this cell to show the HTML representation or trust the notebook.
On GitHub, the HTML representation is unable to render, please try loading this page with nbviewer.org.

In [39]:

```
smote = SMOTE(sampling_strategy='auto')
X_train_smote, y_train_smote = smote.fit_resample(X_train, y_train)
```

In [40]:

```
best_dt_classifier.fit(X_train_smote, y_train_smote)
```

Out[40]:

```
DecisionTreeClassifier(criterion='entropy', max_depth=10, max_features='log2',
                       random_state=42)
```

In a Jupyter environment, please rerun this cell to show the HTML representation or trust the notebook.
On GitHub, the HTML representation is unable to render, please try loading this page with nbviewer.org.

In [41]:

```
y_pred_d3 = best_dt_classifier.predict(X_test)
```

In [44]:

```
cv_metrics_P = cross_val_score(best_dt_classifier, X_train_smote, y_train_smote, cv=cv, scoring='precision')
cv_metrics_P
```

Out[44]:

```
array([0.92709867, 0.93966817, 0.9314759 , 0.93436578, 0.92547529])
```

```
In [45]: cv_metrics_R = cross_val_score(best_dt_classifier, X_train_smote, y_train_smote, cv=cv, scoring='recall')
         cv_metrics_R
```

```
Out[45]: array([0.92573529, 0.91617647, 0.90955882, 0.93230316, 0.89485294])
```

```
In [46]: print('mean_precision:', np.mean(cv_metrics_P))
         print('mean_recall:',np.mean(cv_metrics_R))
```

```
mean_precision: 0.9316167639960244
mean_recall: 0.9157253387006016
```

Logistic Regression

Experiment 1

```
In [47]: LR_classifier = LogisticRegression()
```

```
In [48]: param_grid_LR = {
             'penalty' : ['l1', 'l2'],
             'C' : [0.001, 0.01, 0.1, 1.0],
             'solver' : ['newton-cg', 'lbfgs', 'liblinear', 'sag', 'saga'],
             'max_iter':[100, 200, 300]
         }
```

```
In [49]: R_clf = GridSearchCV(LR_classifier, param_grid_LR)
```

```
In [50]: R_clf.fit(X_train, y_train)
```

```
C:\Users\Romelio Villar Jr\anaconda3\lib\site-packages\sklearn\linear_model\_sag.py:350: ConvergenceWarning: The max_iter was reached which means the coef_ did not converge
  warnings.warn(
C:\Users\Romelio Villar Jr\anaconda3\lib\site-packages\sklearn\linear_model\_sag.py:350: ConvergenceWarning: The max_iter was reached which means the coef_ did not converge
  warnings.warn(
C:\Users\Romelio Villar Jr\anaconda3\lib\site-packages\sklearn\linear_model\_sag.py:350: ConvergenceWarning: The max_iter was reached which means the coef_ did not converge
  warnings.warn(
C:\Users\Romelio Villar Jr\anaconda3\lib\site-packages\sklearn\linear_model\_sag.py:350: ConvergenceWarning: The max_iter was reached which means the coef_ did not converge
  warnings.warn(
C:\Users\Romelio Villar Jr\anaconda3\lib\site-packages\sklearn\linear_model\_sag.py:350: ConvergenceWarning: The max_iter was reached which means the coef_ did not converge
  warnings.warn(
C:\Users\Romelio Villar Jr\anaconda3\lib\site-packages\sklearn\linear_model\_logistic.py:458: ConvergenceWarning: lbfgs failed to converge (status=1):
STOP: TOTAL NO. of ITERATIONS REACHED LIMIT.

Increase the number of iterations (max_iter) or scale the data as shown in:
    https://scikit-learn.org/stable/modules/preprocessing.html (https://scikit-learn.org/stable/modules/preprocessing.html)
Please also refer to the documentation for alternative solver options:
    https://scikit-learn.org/stable/modules/linear_model.html#logistic-regression (https://scikit-learn.org/stable/modules/linear_model.html#logistic-regression)
  n_iter_i = _check_optimize_result(
C:\Users\Romelio Villar Jr\anaconda3\lib\site-packages\sklearn\linear_model\_logistic.py:458: ConvergenceWarning: lbfgs failed to converge (status=1):
STOP: TOTAL NO. of ITERATIONS REACHED LIMIT.
```

```
In [51]: LR_best_parameter = R_clf.best_estimator_
         LR_best_parameter
```

```
Out[51]: LogisticRegression(solver='newton-cg')
```

In a Jupyter environment, please rerun this cell to show the HTML representation or trust the notebook.
On GitHub, the HTML representation is unable to render, please try loading this page with nbviewer.org.

```
In [52]: LR_ros = RandomOverSampler(sampling_strategy = 'auto')
         X_train_L, y_train_L = LR_ros.fit_resample(X_train, y_train)
```

In [53]:

```
LR_best_parameter.fit(X_train_L, y_train_L)
```

Out[53]:

```
LogisticRegression(solver='newton-cg')
```

In a Jupyter environment, please rerun this cell to show the HTML representation or trust the notebook.
On GitHub, the HTML representation is unable to render, please try loading this page with nbviewer.org.

In [54]:

```
y_pred_LR = LR_best_parameter.predict(X_test)
```

In [56]:

```
accuracy = accuracy_score(y_test, y_pred_LR)
precision = precision_score(y_test, y_pred_LR)
recall = recall_score(y_test,y_pred_LR)
f1 = f1_score(y_test, y_pred_LR)
confusion = confusion_matrix(y_test, y_pred_LR)


print("Accuracy:", accuracy)
print("Precision:", precision)
print("Recall:", recall)
print("f1_score:", f1_score)
print("confusion matrix\n", confusion)
```

```
Accuracy: 0.7966436327739388
Precision: 0.9593727726300784
Recall: 0.7912992357436802
f1_score: <function f1_score at 0x000001C107307CA0>
confusion matrix
 [[ 268   57]
 [ 355 1346]]
```

Undersampling

In [57]:

```
LR_rus = RandomUnderSampler(sampling_strategy = 'auto')
X_train_LR, y_train_LR = LR_rus.fit_resample(X_train, y_train)
```

In [58]:

```
LR_best_parameter.fit(X_train_LR, y_train_LR)
```

Out[58]:

```
LogisticRegression(solver='newton-cg')
```

In a Jupyter environment, please rerun this cell to show the HTML representation or trust the notebook.
On GitHub, the HTML representation is unable to render, please try loading this page with nbviewer.org.

In [59]:

```
y_pred_LR2 = LR_best_parameter.predict(X_test)
```

In [61]:

```
accuracy = accuracy_score(y_test, y_pred_LR2)
precision = precision_score(y_test, y_pred_LR2)
recall = recall_score(y_test,y_pred_LR2)
f1 = f1_score(y_test, y_pred_LR2)
confusion = confusion_matrix(y_test, y_pred_LR2)

print("Accuracy:", accuracy)
print("Precision:", precision)
print("Recall:", recall)
print("f1_score:", f1_score)
print("confusion matrix\n", confusion)
```

```
Accuracy: 0.7951628825271471
Precision: 0.9599427753934192
Recall: 0.7889476778365667
f1_score: <function f1_score at 0x000001C107307CA0>
confusion matrix
 [[ 269   56]
 [ 359 1342]]
```

Experiment 2

In [62]:
```
smote_LR = SMOTE(sampling_strategy='auto')
```

In [63]:
```
R_clf = GridSearchCV(LR_classifier, param_grid_LR, cv=cv)
```

In [64]:
```
R_clf.fit(X_train, y_train)
```

```
C:\Users\Romelio Villar Jr\anaconda3\lib\site-packages\sklearn\linear_model\_sag.py:350: ConvergenceWarning: The max_iter was reached which means the coef_ did not converge
  warnings.warn(
C:\Users\Romelio Villar Jr\anaconda3\lib\site-packages\sklearn\linear_model\_sag.py:350: ConvergenceWarning: The max_iter was reached which means the coef_ did not converge
  warnings.warn(
C:\Users\Romelio Villar Jr\anaconda3\lib\site-packages\sklearn\linear_model\_sag.py:350: ConvergenceWarning: The max_iter was reached which means the coef_ did not converge
  warnings.warn(
C:\Users\Romelio Villar Jr\anaconda3\lib\site-packages\sklearn\linear_model\_sag.py:350: ConvergenceWarning: The max_iter was reached which means the coef_ did not converge
  warnings.warn(
C:\Users\Romelio Villar Jr\anaconda3\lib\site-packages\sklearn\linear_model\_sag.py:350: ConvergenceWarning: The max_iter was reached which means the coef_ did not converge
  warnings.warn(
C:\Users\Romelio Villar Jr\anaconda3\lib\site-packages\scipy\optimize\_linesearch.py:416: LineSearchWarning: Rounding errors prevent the line search from converging
  warn(msg, LineSearchWarning)
C:\Users\Romelio Villar Jr\anaconda3\lib\site-packages\scipy\optimize\_linesearch.py:306: LineSearchWarning: The line search algorithm did not converge
  warn('The line search algorithm did not converge', LineSearchWarning)
C:\Users\Romelio Villar Jr\anaconda3\lib\site-packages\sklearn\utils\optimize.py:203: UserWarning: Line Search failed
  warnings.warn("Line Search failed")
C:\Users\Romelio Villar Jr\anaconda3\lib\site-packages\sklearn\utils\optimize.py:210: ConvergenceWarning: newton-cg failed to converge. Increase the number of iterations.
  warnings.warn(
C:\Users\Romelio Villar Jr\anaconda3\lib\site-packages\sklearn\linear_model\_logistic.py:458: ConvergenceWarning: lbfgs failed to converge (status=1):
STOP: TOTAL NO. of ITERATIONS REACHED LIMIT
```

In [117]:
```
X_train_LRsmote, y_train_LRsmote = smote_LR.fit_resample(X_train, y_train)
```

In [118]:
```
LR_best_parameter.fit(X_train_LRsmote, y_train_LRsmote)
```

Out[118]:
```
LogisticRegression(solver='newton-cg')
```

In a Jupyter environment, please rerun this cell to show the HTML representation or trust the notebook.
On GitHub, the HTML representation is unable to render, please try loading this page with nbviewer.org.

In [119]:
```
y_pred_smlr = LR_best_parameter.predict(X_test)
```

In [114]:
```
LR_metrics_P = cross_val_score(LR_best_parameter, X_train_LRsmote, y_train_LRsmote, cv=cv, scoring='precision')
LR_metrics_P
```

```
C:\Users\Romelio Villar Jr\anaconda3\lib\site-packages\scipy\optimize\_linesearch.py:306: LineSearchWarning: The line search algorithm did not converge
  warn('The line search algorithm did not converge', LineSearchWarning)
C:\Users\Romelio Villar Jr\anaconda3\lib\site-packages\sklearn\utils\optimize.py:203: UserWarning: Line Search failed
  warnings.warn("Line Search failed")
```

Out[114]:
```
array([0.8515219 , 0.86712225, 0.87372549, 0.8453997 , 0.85352761])
```

In [115]:
```
LR_metrics_R = cross_val_score(LR_best_parameter, X_train_LRsmote, y_train_LRsmote, cv=cv, scoring='recall')
LR_metrics_R
```

```
C:\Users\Romelio Villar Jr\anaconda3\lib\site-packages\scipy\optimize\_linesearch.py:306: LineSearchWarning: The line search algorithm did not converge
  warn('The line search algorithm did not converge', LineSearchWarning)
C:\Users\Romelio Villar Jr\anaconda3\lib\site-packages\sklearn\utils\optimize.py:203: UserWarning: Line Search failed
  warnings.warn("Line Search failed")
```

Out[115]:
```
array([0.84338235, 0.83970588, 0.81911765, 0.82487123, 0.81838235])
```

In [116]:

```
print('Precision_Logistic Regression:', np.mean(LR_metrics_P))
print('Recall_Logistic Regression:', np.mean(LR_metrics_R))
```

```
Precision_Logistic Regression: 0.8582593887901722
Recall_Logistic Regression: 0.8290918928277712
```

In [ ]:

```
```