In [78]:
```
import numpy as np
import pandas as pd
train = pd.read_csv('UNSW_NB15_training-set.csv')
test = pd.read_csv('UNSW_NB15_testing-set.csv')
from sklearn.preprocessing import OrdinalEncoder
ord_enc = OrdinalEncoder()
```

In [79]:
```
train['proto_code'] = ord_enc.fit_transform(train[['proto']])
train[['proto','proto_code']].head(175341)
```

Out[79]:

	proto	proto_code
0	tcp	113.0
1	tcp	113.0
2	tcp	113.0
3	tcp	113.0
4	tcp	113.0
...	...	...
175336	udp	119.0
175337	tcp	113.0
175338	udp	119.0
175339	udp	119.0
175340	udp	119.0

175341 rows × 2 columns

In [80]:
```
train['state_code'] = ord_enc.fit_transform(train[['state']])
train[['state','state_code']].head(175341)
```

Out[80]:

	state	state_code
0	FIN	2.0
1	FIN	2.0
2	FIN	2.0
3	FIN	2.0
4	FIN	2.0
...	...	...
175336	INT	3.0
175337	FIN	2.0
175338	INT	3.0
175339	INT	3.0
175340	INT	3.0

175341 rows × 2 columns

In [81]:
```
train_updated = train.replace('-',np.nan)
print(train_updated)
```

```
            id       dur proto service state  spkts  dpkts  sbytes  dbytes  \
0            1  0.121478   tcp     NaN   FIN      6      4     258     172
1            2  0.649902   tcp     NaN   FIN     14     38     734   42014
2            3  1.623129   tcp     NaN   FIN      8     16     364   13186
3            4  1.681642   tcp     ftp   FIN     12     12     628     770
4            5  0.449454   tcp     NaN   FIN     10      6     534     268
...        ...       ...   ...     ...   ...    ...    ...     ...     ...
175336  175337  0.000009   udp     dns   INT      2      0     114       0
175337  175338  0.505762   tcp     NaN   FIN     10      8     620     354
175338  175339  0.000009   udp     dns   INT      2      0     114       0
175339  175340  0.000009   udp     dns   INT      2      0     114       0
175340  175341  0.000009   udp     dns   INT      2      0     114       0

                rate  ...  is_ftp_login  ct_ftp_cmd  ct_flw_http_mthd  \
0          74.087490  ...             0           0                 0
1          78.473372  ...             0           0                 0
2          14.170161  ...             0           0                 0
3          13.677108  ...             1           1                 0
4          33.373826  ...             0           0                 0
...              ...  ...           ...         ...               ...
175336  111111.107200  ...             0           0                 0
175337     33.612649  ...             0           0                 0
175338  111111.107200  ...             0           0                 0
175339  111111.107200  ...             0           0                 0
175340  111111.107200  ...             0           0                 0

        ct_src_ltm  ct_srv_dst  is_sm_ips_ports  attack_cat  label  \
```

```
0                1           1                0      Normal      0
1                1           6                0      Normal      0
2                2           6                0      Normal      0
3                2           1                0      Normal      0
4                2          39                0      Normal      0
...            ...         ...              ...         ...    ...
175336          24          24                0     Generic      1
175337           1           1                0   Shellcode      1
175338           3          12                0     Generic      1
175339          30          30                0     Generic      1
175340          30          30                0     Generic      1

        proto_code  state_code
0            113.0         2.0
1            113.0         2.0
2            113.0         2.0
3            113.0         2.0
4            113.0         2.0
...            ...         ...
175336       119.0         3.0
175337       113.0         2.0
175338       119.0         3.0
175339       119.0         3.0
175340       119.0         3.0

[175341 rows x 47 columns]
```

In [82]:

```
final_train = train_updated.fillna("nodata")
print(final_train)
```

```
            id       dur proto service state  spkts  dpkts  sbytes  dbytes  \
0            1  0.121478   tcp  nodata   FIN      6      4     258     172
1            2  0.649902   tcp  nodata   FIN     14     38     734   42014
2            3  1.623129   tcp  nodata   FIN      8     16     364   13186
3            4  1.681642   tcp     ftp   FIN     12     12     628     770
4            5  0.449454   tcp  nodata   FIN     10      6     534     268
...        ...       ...   ...     ...   ...    ...    ...     ...     ...
175336  175337  0.000009   udp     dns   INT      2      0     114       0
175337  175338  0.505762   tcp  nodata   FIN     10      8     620     354
175338  175339  0.000009   udp     dns   INT      2      0     114       0
175339  175340  0.000009   udp     dns   INT      2      0     114       0
175340  175341  0.000009   udp     dns   INT      2      0     114       0

                rate  ...  is_ftp_login  ct_ftp_cmd  ct_flw_http_mthd  \
0          74.087490  ...             0           0                 0
1          78.473372  ...             0           0                 0
2          14.170161  ...             0           0                 0
3          13.677108  ...             1           1                 0
4          33.373826  ...             0           0                 0
...              ...  ...           ...         ...               ...
175336  111111.107200  ...             0           0                 0
175337     33.612649  ...             0           0                 0
175338  111111.107200  ...             0           0                 0
175339  111111.107200  ...             0           0                 0
175340  111111.107200  ...             0           0                 0

        ct_src_ltm  ct_srv_dst  is_sm_ips_ports  attack_cat  label  \
0                1           1                0      Normal      0
1                1           6                0      Normal      0
2                2           6                0      Normal      0
3                2           1                0      Normal      0
4                2          39                0      Normal      0
...            ...         ...              ...         ...    ...
175336          24          24                0     Generic      1
175337           1           1                0   Shellcode      1
175338           3          12                0     Generic      1
175339          30          30                0     Generic      1
175340          30          30                0     Generic      1

        proto_code  state_code
0            113.0         2.0
1            113.0         2.0
2            113.0         2.0
3            113.0         2.0
4            113.0         2.0
...            ...         ...
175336       119.0         3.0
175337       113.0         2.0
175338       119.0         3.0
175339       119.0         3.0
175340       119.0         3.0

[175341 rows x 47 columns]
```

In [83]:

```
final_train['service_code'] = ord_enc.fit_transform(final_train[['service']])
final_train[['service','service_code']].head(175341)
```

Out[83]:

	service	service_code
0	nodata	6.0
1	nodata	6.0
2	nodata	6.0
3	ftp	2.0
4	nodata	6.0
...	...	...

	service	service_code
175336	dns	1.0
175337	nodata	6.0
175338	dns	1.0
175339	dns	1.0
175340	dns	1.0

175341 rows × 2 columns

In [84]:
```
test['proto_code'] = ord_enc.fit_transform(test[['proto']])
test[['proto','proto_code']].head(175341)
```

Out[84]:

	proto	proto_code
0	udp	117.0
1	udp	117.0
2	udp	117.0
3	udp	117.0
4	udp	117.0
...	...	...
82327	udp	117.0
82328	tcp	111.0
82329	arp	6.0
82330	arp	6.0
82331	udp	117.0

82332 rows × 2 columns

In [85]:
```
test['state_code'] = ord_enc.fit_transform(test[['state']])
test[['state','state_code']].head(175341)
```

Out[85]:

	state	state_code
0	INT	4.0
1	INT	4.0
2	INT	4.0
3	INT	4.0
4	INT	4.0
...	...	...
82327	INT	4.0
82328	FIN	3.0
82329	INT	4.0
82330	INT	4.0
82331	INT	4.0

82332 rows × 2 columns

In [86]:
```
test_updated = test.replace('-',np.nan)
print(test_updated)
```

```
          id       dur proto service state  spkts  dpkts  sbytes  dbytes  \
0          1  0.000011   udp     NaN   INT      2      0     496       0
1          2  0.000008   udp     NaN   INT      2      0    1762       0
2          3  0.000005   udp     NaN   INT      2      0    1068       0
3          4  0.000006   udp     NaN   INT      2      0     900       0
4          5  0.000010   udp     NaN   INT      2      0    2126       0
...      ...       ...   ...     ...   ...    ...    ...     ...     ...
82327  82328  0.000005   udp     NaN   INT      2      0     104       0
82328  82329  1.106101   tcp     NaN   FIN     20      8   18062     354
82329  82330  0.000000   arp     NaN   INT      1      0      46       0
82330  82331  0.000000   arp     NaN   INT      1      0      46       0
82331  82332  0.000009   udp     NaN   INT      2      0     104       0

                rate  ...  is_ftp_login  ct_ftp_cmd  ct_flw_http_mthd  \
0       90909.090200  ...             0           0                 0
1      125000.000300  ...             0           0                 0
2      200000.005100  ...             0           0                 0
3      166666.660800  ...             0           0                 0
4      100000.002500  ...             0           0                 0
...              ...  ...           ...         ...               ...
82327  200000.005100  ...             0           0                 0
82328      24.410067  ...             0           0                 0
```

```
82329       0.000000  ...             0          0                0
82330       0.000000  ...             0          0                0
82331  111111.107200  ...             0          0                0

       ct_src_ltm  ct_srv_dst  is_sm_ips_ports  attack_cat  label  proto_code  \
0               1           2                0      Normal      0       117.0
1               1           2                0      Normal      0       117.0
2               1           3                0      Normal      0       117.0
3               2           3                0      Normal      0       117.0
4               2           3                0      Normal      0       117.0
...           ...         ...              ...         ...    ...         ...
82327           2           1                0      Normal      0       117.0
82328           3           2                0      Normal      0       111.0
82329           1           1                1      Normal      0         6.0
82330           1           1                1      Normal      0         6.0
82331           1           1                0      Normal      0       117.0

       state_code
0             4.0
1             4.0
2             4.0
3             4.0
4             4.0
...           ...
82327         4.0
82328         3.0
82329         4.0
82330         4.0
82331         4.0

[82332 rows x 47 columns]
```

In [87]:

```
final_test = test_updated.fillna("nodata")
print(final_test)
```

```
          id       dur proto service state  spkts  dpkts  sbytes  dbytes  \
0          1  0.000011   udp  nodata   INT      2      0     496       0
1          2  0.000008   udp  nodata   INT      2      0    1762       0
2          3  0.000005   udp  nodata   INT      2      0    1068       0
3          4  0.000006   udp  nodata   INT      2      0     900       0
4          5  0.000010   udp  nodata   INT      2      0    2126       0
...      ...       ...   ...     ...   ...    ...    ...     ...     ...
82327  82328  0.000005   udp  nodata   INT      2      0     104       0
82328  82329  1.106101   tcp  nodata   FIN     20      8   18062     354
82329  82330  0.000000   arp  nodata   INT      1      0      46       0
82330  82331  0.000000   arp  nodata   INT      1      0      46       0
82331  82332  0.000009   udp  nodata   INT      2      0     104       0

                rate  ...  is_ftp_login  ct_ftp_cmd  ct_flw_http_mthd  \
0       90909.090200  ...             0           0                 0
1      125000.000300  ...             0           0                 0
2      200000.005100  ...             0           0                 0
3      166666.660800  ...             0           0                 0
4      100000.002500  ...             0           0                 0
...              ...  ...           ...         ...               ...
82327  200000.005100  ...             0           0                 0
82328      24.410067  ...             0           0                 0
82329       0.000000  ...             0           0                 0
82330       0.000000  ...             0           0                 0
82331  111111.107200  ...             0           0                 0

       ct_src_ltm  ct_srv_dst  is_sm_ips_ports  attack_cat  label  proto_code  \
0               1           2                0      Normal      0       117.0
1               1           2                0      Normal      0       117.0
2               1           3                0      Normal      0       117.0
3               2           3                0      Normal      0       117.0
4               2           3                0      Normal      0       117.0
...           ...         ...              ...         ...    ...         ...
82327           2           1                0      Normal      0       117.0
82328           3           2                0      Normal      0       111.0
82329           1           1                1      Normal      0         6.0
82330           1           1                1      Normal      0         6.0
82331           1           1                0      Normal      0       117.0

       state_code
0             4.0
1             4.0
2             4.0
3             4.0
4             4.0
...           ...
82327         4.0
82328         3.0
82329         4.0
82330         4.0
82331         4.0

[82332 rows x 47 columns]
```

In [88]:

```
final_test['service_code'] = ord_enc.fit_transform(final_test[['service']])
final_test[['service','service_code']].head(175341)
```

Out[88]:

	service	service_code
0	nodata	6.0
1	nodata	6.0
2	nodata	6.0
3	nodata	6.0

	service	service_code
4	nodata	6.0
...	...	...
82327	nodata	6.0
82328	nodata	6.0
82329	nodata	6.0
82330	nodata	6.0
82331	nodata	6.0

82332 rows × 2 columns

In [89]:
```
x1 = final_train[['id','dur','proto_code','state_code','spkts','dpkts',
        'sbytes','dbytes','rate','sttl','dttl','sload','dload','sloss','dloss',
       'sinpkt','dinpkt','sjit','djit','swin','stcpb',
       'dtcpb','dwin','tcprtt','synack','ackdat','smean','dmean','trans_depth','response_body_len','ct_srv_src','ct_stat
      ,'ct_dst_ltm','ct_src_dport_ltm','ct_dst_sport_ltm','ct_dst_src_ltm','is_ftp_login',
       'ct_ftp_cmd','ct_flw_http_mthd','ct_src_ltm','ct_srv_dst','is_sm_ips_ports']]
y1 = final_train['label']
x2 = final_test[['id','dur','proto_code','state_code','spkts','dpkts',
        'sbytes','dbytes','rate','sttl','dttl','sload','dload','sloss','dloss',
       'sinpkt','dinpkt','sjit','djit','swin','stcpb',
       'dtcpb','dwin','tcprtt','synack','ackdat','smean','dmean','trans_depth','response_body_len','ct_srv_src','ct_stat
      ,'ct_dst_ltm','ct_src_dport_ltm','ct_dst_sport_ltm','ct_dst_src_ltm','is_ftp_login',
       'ct_ftp_cmd','ct_flw_http_mthd','ct_src_ltm','ct_srv_dst','is_sm_ips_ports']]
y2 = final_test['label']
```

In [90]:
```
from sklearn.preprocessing import MinMaxScaler
model = MinMaxScaler()
model.fit(x1)
x1 = model.transform(x1)
x2 = model.transform(x2)
```

In [123…
```
from xgboost import XGBClassifier
import xgboost as xgb
params = {
            'objective':'binary:logistic',
            'max_depth': 4,
            'alpha': 10,
            'learning_rate': 0.1,
            'n_estimators':100
        }
xgb_clf = XGBClassifier(**params)
xgb_clf.fit(x1, y1)
```

```
C:\Users\admin\anaconda3\lib\site-packages\xgboost\sklearn.py:1224: UserWarning: The use of label encoder in XGBClassifie
r is deprecated and will be removed in a future release. To remove this warning, do the following: 1) Pass option use_lab
el_encoder=False when constructing XGBClassifier object; and 2) Encode your labels (y) as integers starting with 0, i.e.
0, 1, 2, ..., [num_class - 1].
  warnings.warn(label_encoder_deprecation_msg, UserWarning)
[11:03:18] WARNING: C:/Users/Administrator/workspace/xgboost-win64_release_1.5.1/src/learner.cc:1115: Starting in XGBoost
1.3.0, the default evaluation metric used with the objective 'binary:logistic' was changed from 'error' to 'logloss'. Exp
licitly set eval_metric if you'd like to restore the old behavior.
```

Out[123…
```
XGBClassifier(alpha=10, base_score=0.5, booster='gbtree', colsample_bylevel=1,
              colsample_bynode=1, colsample_bytree=1, enable_categorical=False,
              gamma=0, gpu_id=-1, importance_type=None,
              interaction_constraints='', learning_rate=0.1, max_delta_step=0,
              max_depth=4, min_child_weight=1, missing=nan,
              monotone_constraints='()', n_estimators=100, n_jobs=4,
              num_parallel_tree=1, predictor='auto', random_state=0,
              reg_alpha=10, reg_lambda=1, scale_pos_weight=1, subsample=1,
              tree_method='exact', validate_parameters=1, verbosity=None)
```

In [124…
```
y_pred=xgb_clf.predict(x2)
print(y_pred)
```

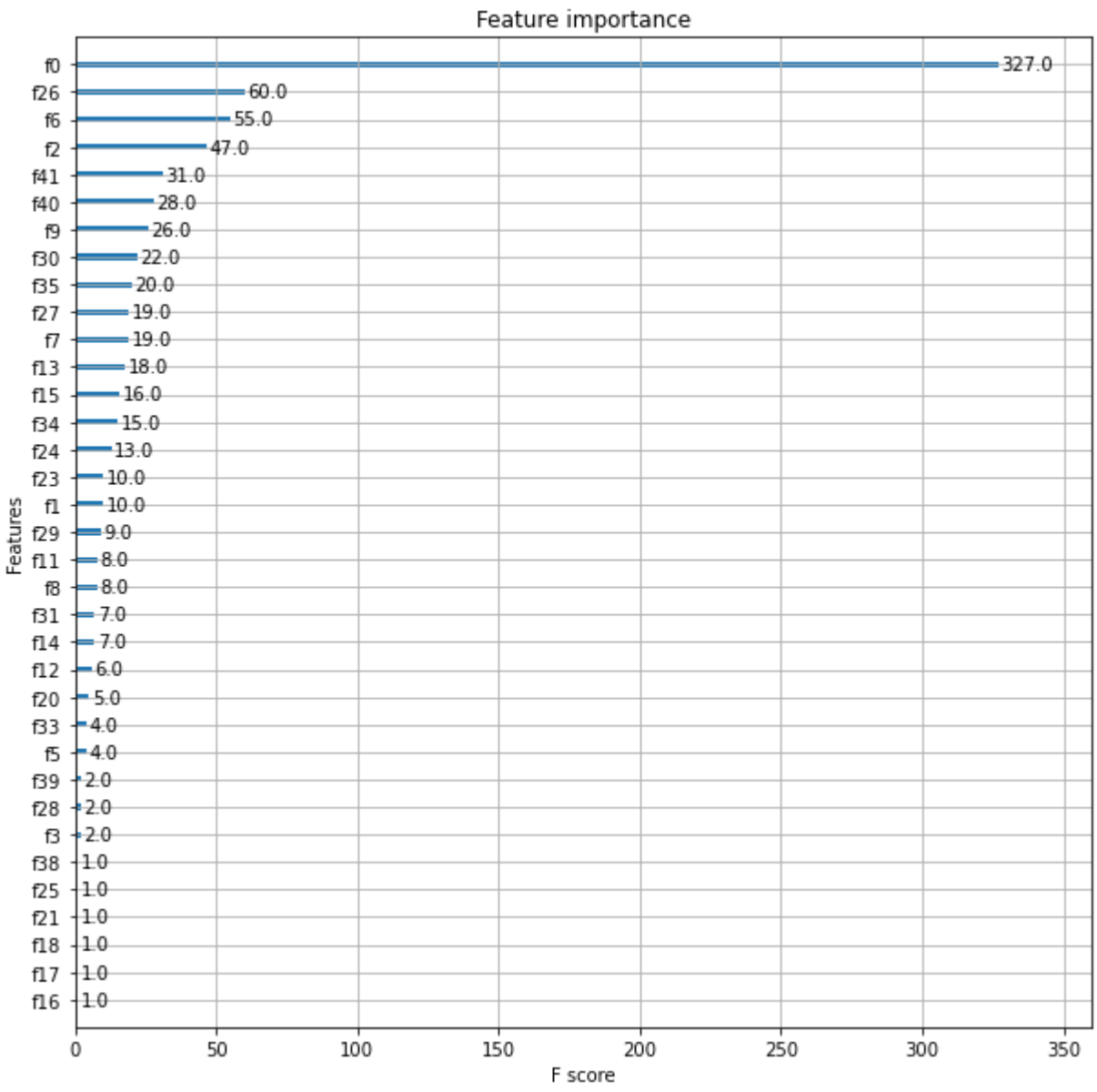
```
[0 0 0 ... 0 0 1]
```

In [125…
```
from sklearn import metrics
from sklearn.metrics import f1_score

print('Accuracy = ', metrics.accuracy_score(y2, y_pred)*100)
print("Confusion Matrix =", metrics.confusion_matrix(y2, y_pred, labels=None,
                                                     sample_weight=None))
print("Recall =", metrics.recall_score(y2, y_pred, labels=None,
                                       pos_label=1, average='weighted',
                                       sample_weight=None))
print("Classification Report =\n", metrics.classification_report(y2, y_pred,
                                                                 labels=None,
                                                                 target_names=None,
                                                                 sample_weight=None,
                                                                 digits=2,
                                                                 output_dict=False))

print("F1 Score = ",f1_score(y2, y_pred, average='macro'))
```

```
Accuracy =  49.46557838993344
Confusion Matrix = [[22795 14205]
 [27401 17931]]
Recall = 0.4946557838993344
Classification Report =
              precision    recall  f1-score   support

           0       0.45      0.62      0.52     37000
           1       0.56      0.40      0.46     45332

    accuracy                           0.49     82332
   macro avg       0.51      0.51      0.49     82332
weighted avg       0.51      0.49      0.49     82332

F1 Score =  0.4928858562028976
```

In [126…

```
import matplotlib.pyplot as plt
xgb.plot_importance(xgb_clf)
plt.rcParams['figure.figsize'] = [10, 10]
plt.show()
```



In [143…

```
x1_new = final_train[['id','sbytes','smean','ct_srv_src','sload','ct_dst_src_ltm','ct_srv_dst','synack','ackdat','dur'
                      ,'dload','tcprtt','dtcpb','stcpb']]
y1_new = final_train['label']
x2_new = final_test[['id','sbytes','smean','ct_srv_src','sload','ct_dst_src_ltm','ct_srv_dst','synack','ackdat','dur','dl
                     ,'tcprtt','dtcpb','stcpb']]
y2_new = final_test['label']
```

In [144…

```
model_new = MinMaxScaler()
model_new.fit(x1_new)
x1_new = model_new.transform(x1_new)
x2_new = model_new.transform(x2_new)
```

In [145…

```
from xgboost import XGBClassifier
import xgboost as xgb
params = {
            'objective':'binary:logistic',
            'max_depth': 4,
            'alpha': 10,
            'learning_rate': 0.1,
            'n_estimators':100
        }
xgb_clf = XGBClassifier(**params)
xgb_clf.fit(x1_new, y1_new)
```

```
C:\Users\admin\anaconda3\lib\site-packages\xgboost\sklearn.py:1224: UserWarning: The use of label encoder in XGBClassifie
r is deprecated and will be removed in a future release. To remove this warning, do the following: 1) Pass option use_lab
el_encoder=False when constructing XGBClassifier object; and 2) Encode your labels (y) as integers starting with 0, i.e.
0, 1, 2, ..., [num_class - 1].
  warnings.warn(label_encoder_deprecation_msg, UserWarning)
[11:13:31] WARNING: C:/Users/Administrator/workspace/xgboost-win64_release_1.5.1/src/learner.cc:1115: Starting in XGBoost
1.3.0, the default evaluation metric used with the objective 'binary:logistic' was changed from 'error' to 'logloss'. Exp
licitly set eval_metric if you'd like to restore the old behavior.
```

Out[145… XGBClassifier(alpha=10, base_score=0.5, booster='gbtree', colsample_bylevel=1,

```
                colsample_bynode=1, colsample_bytree=1, enable_categorical=False,
                gamma=0, gpu_id=-1, importance_type=None,
                interaction_constraints='', learning_rate=0.1, max_delta_step=0,
                max_depth=4, min_child_weight=1, missing=nan,
                monotone_constraints='()', n_estimators=100, n_jobs=4,
                num_parallel_tree=1, predictor='auto', random_state=0,
                reg_alpha=10, reg_lambda=1, scale_pos_weight=1, subsample=1,
                tree_method='exact', validate_parameters=1, verbosity=None)
```

In [146…

```
y_pred=xgb_clf.predict(x2_new)
print(y_pred)
```

```
[0 0 0 ... 0 0 1]
```

In [147…

```
from sklearn import metrics
from sklearn.metrics import f1_score

print('Accuracy = ', metrics.accuracy_score(y2, y_pred)*100)
print("Confusion Matrix =", metrics.confusion_matrix(y2, y_pred, labels=None,
                                                     sample_weight=None))
print("Recall =", metrics.recall_score(y2, y_pred, labels=None,
                                       pos_label=1, average='weighted',
                                       sample_weight=None))
print("Classification Report =\n", metrics.classification_report(y2, y_pred,
                                                                 labels=None,
                                                                 target_names=None,
                                                                 sample_weight=None,
                                                                 digits=2,
                                                                 output_dict=False))

print("F1 Score = ",f1_score(y2, y_pred, average='macro'))
```

```
Accuracy =  51.67128212602633
Confusion Matrix = [[24611 12389]
 [27401 17931]]
Recall = 0.5167128212602633
Classification Report =
               precision    recall  f1-score   support

           0       0.47      0.67      0.55     37000
           1       0.59      0.40      0.47     45332

    accuracy                           0.52     82332
   macro avg       0.53      0.53      0.51     82332
weighted avg       0.54      0.52      0.51     82332

F1 Score =  0.5135103206175136
```

In [ ]: