```
import numpy as np
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression
from sklearn.tree import DecisionTreeClassifier
from sklearn.svm import SVC
from sklearn.ensemble import RandomForestClassifier
from sklearn.neighbors import KNeighborsClassifier
from sklearn.preprocessing import StandardScaler, OneHotEncoder
from sklearn.impute import SimpleImputer
from sklearn.compose import ColumnTransformer
from sklearn.metrics import roc_auc_score, roc_curve, confusion_matrix
from sklearn.pipeline import Pipeline
import matplotlib.pyplot as plt
import ipaddress
from sklearn.ensemble import AdaBoostClassifier, ExtraTreesClassifier
from xgboost import XGBClassifier
from lightgbm import LGBMClassifier

data = pd.read_csv('rba-dataset.csv')

import pandas as pd

# Specify the chunk size
chunk_size = 1000000  # Adjust based on your memory limits

# Initialize an empty list to store the data
chunks = []

# Read the CSV file in chunks
for chunk in pd.read_csv('rba-dataset.csv', chunksize=chunk_size):
    chunks.append(chunk)
    print("SSs")

# Concatenate all chunks into a single DataFrame
data = pd.concat(chunks, axis=0)
```

```
SSs
SSs
SSs
SSs
SSs
SSs
SSs
SSs
SSs
SSs
SSs
SSs
SSs
```

```
SSs
SSs
SSs
SSs
SSs
SSs
SSs
SSs
SSs
SSs
SSs
SSs
SSs
SSs
SSs
SSs
SSs
SSs
SSs
```

```
data.head()
```

```
   index           Login Timestamp              User ID  Round-Trip
Time [ms]  \
0      0  2020-02-03 12:43:30.772 -4324475583306591935
NaN
1      1  2020-02-03 12:43:43.549 -4324475583306591935
NaN
2      2  2020-02-03 12:43:55.873 -3284137479262433373
NaN
3      3  2020-02-03 12:43:56.180 -4324475583306591935
NaN
4      4  2020-02-03 12:43:59.396 -4618854071942621186
NaN

      IP Address Country    Region       City     ASN  \
0    10.0.65.171      NO         -          -   29695
1   194.87.207.6      AU         -          -   60117
2  81.167.144.58      NO  Vestland  Urangsvag   29695
3  170.39.78.152      US         -          -  393398
4      10.0.0.47      US  Virginia    Ashburn  398986

                                   User Agent String  \
0  Mozilla/5.0  (iPhone; CPU iPhone OS 13_4 like ...
1  Mozilla/5.0  (Linux; Android 4.1; Galaxy Nexus...
2  Mozilla/5.0  (iPad; CPU OS 7_1 like Mac OS X) ...
3  Mozilla/5.0  (Linux; Android 4.1; Galaxy Nexus...
4  Mozilla/5.0  (Linux; U; Android 2.2) Build/NMA...

          Browser Name and Version OS Name and Version Device Type  \
```

```
0              Firefox 20.0.0.1618            iOS 13.4      mobile
1         Chrome Mobile 46.0.2490         Android 4.1      mobile
2              Android 2.3.3.2672             iOS 7.1      mobile
3  Chrome Mobile WebView 85.0.4183         Android 4.1      mobile
4  Chrome Mobile WebView 85.0.4183         Android 2.2      mobile

   Login Successful  Is Attack IP  Is Account Takeover
0             False         False                False
1             False         False                False
2              True         False                False
3             False         False                False
4             False          True                False
```

```
len(data)
```

```
31269264
```

```
data.dtypes
```

```
index                          int64
Login Timestamp               object
User ID                        int64
Round-Trip Time [ms]         float64
IP Address                    object
Country                       object
Region                        object
City                          object
ASN                            int64
User Agent String             object
Browser Name and Version      object
OS Name and Version           object
Device Type                   object
Login Successful                bool
Is Attack IP                    bool
Is Account Takeover             bool
dtype: object
```

```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 31269264 entries, 0 to 31269263
Data columns (total 16 columns):
 #   Column                    Dtype
---  ------                    -----
 0   index                     int64
 1   Login Timestamp           object
 2   User ID                   int64
 3   Round-Trip Time [ms]      float64
 4   IP Address                object
 5   Country                   object
 6   Region                    object
```

```
 7   City                      object
 8   ASN                       int64
 9   User Agent String         object
 10  Browser Name and Version  object
 11  OS Name and Version       object
 12  Device Type               object
 13  Login Successful          bool
 14  Is Attack IP              bool
 15  Is Account Takeover       bool
dtypes: bool(3), float64(1), int64(3), object(9)
memory usage: 3.1+ GB
```

```
data.describe
```

```
<bound method NDFrame.describe of              index          Login
Timestamp              User ID  \
0                0  2020-02-03 12:43:30.772 -4324475583306591935
1                1  2020-02-03 12:43:43.549 -4324475583306591935
2                2  2020-02-03 12:43:55.873 -3284137479262433373
3                3  2020-02-03 12:43:56.180 -4324475583306591935
4                4  2020-02-03 12:43:59.396 -4618854071942621186
...            ...                      ...                  ...
31269259  31269259  2021-02-28 23:59:47.766 -4324475583306591935
31269260  31269260  2021-02-28 23:59:49.956 -4324475583306591935
31269261  31269261  2021-02-28 23:59:54.233 -4324475583306591935
31269262  31269262  2021-02-28 23:59:56.343 -4324475583306591935
31269263  31269263  2021-02-28 23:59:58.756 -3863191272176615105

          Round-Trip Time [ms]     IP Address Country           Region
\
0                          NaN    10.0.65.171      NO                -

1                          NaN   194.87.207.6      AU                -

2                          NaN  81.167.144.58      NO         Vestland

3                          NaN  170.39.78.152      US                -

4                          NaN      10.0.0.47      US         Virginia

...                        ...            ...     ...              ...

31269259                   NaN  170.39.78.106      US                -

31269260                   NaN  170.39.79.123      US                -

31269261                   NaN  170.39.78.106      US                -

31269262                   NaN   10.3.205.188      RU  St.-Petersburg

31269263                   NaN  156.52.189.92      NO            Viken
```

```
                   City     ASN  \
0                     -   29695
1                     -   60117
2             Urangsvag   29695
3                     -  393398
4               Ashburn  398986
...                 ...     ...
31269259              -  393398
31269260              -  393398
31269261              -  393398
31269262  St Petersburg   15599
31269263    Fredrikstad   29695

                                          User Agent String  \
0         Mozilla/5.0  (iPhone; CPU iPhone OS 13_4 like ...
1         Mozilla/5.0  (Linux; Android 4.1; Galaxy Nexus...
2         Mozilla/5.0  (iPad; CPU OS 7_1 like Mac OS X) ...
3         Mozilla/5.0  (Linux; Android 4.1; Galaxy Nexus...
4         Mozilla/5.0  (Linux; U; Android 2.2) Build/NMA...
...                                                     ...
31269259      AwarioSmartBot/1.0  (en-us) variation/294820
31269260  Mozilla/5.0  (iPhone; CPU iPhone OS 11_2_6 lik...
31269261      AwarioSmartBot/1.0  (en-us) variation/294820
31269262  ZipppBot/0.11 (ZipppBot; https://github.com/da...
31269263  Mozilla/5.0 (Macintosh; Intel Mac OS X 10_14_6...

                  Browser Name and Version OS Name and Version Device
Type  \
0                     Firefox 20.0.0.1618            iOS 13.4
mobile
1                  Chrome Mobile 46.0.2490         Android 4.1
mobile
2                      Android 2.3.3.2672             iOS 7.1
mobile
3         Chrome Mobile WebView 85.0.4183         Android 4.1
mobile
4         Chrome Mobile WebView 85.0.4183         Android 2.2
mobile
...                                    ...                 ...
...
31269259               AwarioSmartBot 1.0              Other
unknown
31269260  Chrome Mobile WebView 80.0.3987          iOS 11.2.6
mobile
31269261               AwarioSmartBot 1.0              Other
unknown
31269262                    ZipppBot 0.11              Other
bot
```

```
31269263           Chrome 69.0.3497.17.24    Mac OS X 10.14.6
desktop

          Login Successful  Is Attack IP  Is Account Takeover
0                    False         False                False
1                    False         False                False
2                     True         False                False
3                    False         False                False
4                    False          True                False
...                    ...           ...                  ...
31269259             False         False                False
31269260             False         False                False
31269261             False         False                False
31269262             False         False                False
31269263              True         False                False

[31269264 rows x 16 columns]>
```

```
data.isna().sum()
```

```
index                             0
Login Timestamp                   0
User ID                           0
Round-Trip Time [ms]        29993329
IP Address                        0
Country                           0
Region                        47409
City                           8590
ASN                               0
User Agent String                 0
Browser Name and Version          0
OS Name and Version               0
Device Type                    1526
Login Successful                  0
Is Attack IP                      0
Is Account Takeover               0
dtype: int64
```

```
data['Login Hour'] = pd.to_datetime(data['Login Timestamp']).dt.hour
```

```
data.head()
```

```
   index          Login Timestamp              User ID  Round-Trip
Time [ms]  \
0      0  2020-02-03 12:43:30.772 -4324475583306591935
NaN
1      1  2020-02-03 12:43:43.549 -4324475583306591935
NaN
2      2  2020-02-03 12:43:55.873 -3284137479262433373
NaN
3      3  2020-02-03 12:43:56.180 -4324475583306591935
```

```
NaN
4      4  2020-02-03 12:43:59.396 -4618854071942621186
NaN

      IP Address Country    Region       City     ASN  \
0    10.0.65.171      NO         -          -   29695
1   194.87.207.6      AU         -          -   60117
2  81.167.144.58      NO  Vestland  Urangsvag   29695
3  170.39.78.152      US         -          -  393398
4     10.0.0.47      US  Virginia    Ashburn  398986

                                   User Agent String  \
0  Mozilla/5.0  (iPhone; CPU iPhone OS 13_4 like ...
1  Mozilla/5.0  (Linux; Android 4.1; Galaxy Nexus...
2  Mozilla/5.0  (iPad; CPU OS 7_1 like Mac OS X) ...
3  Mozilla/5.0  (Linux; Android 4.1; Galaxy Nexus...
4  Mozilla/5.0  (Linux; U; Android 2.2) Build/NMA...

          Browser Name and Version OS Name and Version Device Type  \
0             Firefox 20.0.0.1618           iOS 13.4      mobile
1          Chrome Mobile 46.0.2490         Android 4.1      mobile
2               Android 2.3.3.2672             iOS 7.1      mobile
3  Chrome Mobile WebView 85.0.4183         Android 4.1      mobile
4  Chrome Mobile WebView 85.0.4183         Android 2.2      mobile

   Login Successful  Is Attack IP  Is Account Takeover  Login Hour
0             False         False                False          12
1             False         False                False          12
2              True         False                False          12
3             False         False                False          12
4             False          True                False          12

data['Is Account Takeover'] = data['Is Account
Takeover'].astype(np.uint8)
data['Is Attack IP'] = data['Is Attack IP'].astype(np.uint8)
data['Login Successful'] = data['Login Successful'].astype(np.uint8)

data = data.drop(columns=["Round-Trip Time [ms]", 'Region', 'City',
'Login Timestamp', 'index'])

data.head()

                User ID     IP Address Country     ASN  \
0 -4324475583306591935    10.0.65.171      NO   29695
1 -4324475583306591935   194.87.207.6      AU   60117
2 -3284137479262433373  81.167.144.58      NO   29695
3 -4324475583306591935  170.39.78.152      US  393398
4 -4618854071942621186     10.0.0.47      US  398986

                                   User Agent String  \
0  Mozilla/5.0  (iPhone; CPU iPhone OS 13_4 like ...
```

```
1  Mozilla/5.0  (Linux; Android 4.1; Galaxy Nexus...
2  Mozilla/5.0  (iPad; CPU OS 7_1 like Mac OS X) ...
3  Mozilla/5.0  (Linux; Android 4.1; Galaxy Nexus...
4  Mozilla/5.0  (Linux; U; Android 2.2) Build/NMA...
```

```
          Browser Name and Version OS Name and Version Device Type  \
0             Firefox 20.0.0.1618           iOS 13.4      mobile
1          Chrome Mobile 46.0.2490         Android 4.1      mobile
2               Android 2.3.3.2672           iOS 7.1      mobile
3  Chrome Mobile WebView 85.0.4183         Android 4.1      mobile
4  Chrome Mobile WebView 85.0.4183         Android 2.2      mobile

   Login Successful  Is Attack IP  Is Account Takeover  Login Hour
0                 0            0                    0          12
1                 0            0                    0          12
2                 1            0                    0          12
3                 0            0                    0          12
4                 0            1                    0          12
```

```
data['User Agent String'], _ = pd.factorize(data['User Agent String'])
data['Browser Name and Version'], _ = pd.factorize(data['Browser Name
and Version'])
data['OS Name and Version'], _ = pd.factorize(data['OS Name and
Version'])
```

```
def ip_to_int(ip):
    return int(ipaddress.ip_address(ip))

data['IP Address'] = data['IP Address'].apply(ip_to_int)

data.head(20)
```

```
                User ID  IP Address Country     ASN  User Agent String
\
0  -4324475583306591935   167788971      NO   29695                  0

1  -4324475583306591935  3260534534      AU   60117                  1

2  -3284137479262433373  1369935930      NO   29695                  2

3  -4324475583306591935  2854702744      US  393398                  3

4  -4618854071942621186   167772207      US  398986                  4

5  -4324475583306591935  3521936254      US  393398                  5

6   7246533443898239661  1355474134      NO   15659                  6

7  -3243978724802435038  2854702769      US  393398                  7

8   8076000552587369902   167787988      NO   29695                  8
```

```
9  -3065936140549856249  1558015394      NO   29695                   9
10  5932501938287412564  1412447623      NO   15659                  10
11 -9080829243863829585  2620665939      NO   29695                  11
12  5729679535281970107  2854703012      US  393398                  12
13 -4324475583306591935   785658794      NO   41164                  13
14 -4324475583306591935   167789398      NO   29695                   2
15 -8296667206273764769  1385380445      NO   29492                   2
16 -4663943525943860871  1369861544      NO   29695                  10
17 -4324475583306591935  1933957922      ID   38778                  14
18 -4324475583306591935   783790079      NO  197475                  15
19  1211299018980019605   167792079      NO   29695                  16
```

	Browser Name and Version	OS Name and Version	Device Type
0	0	0	mobile
1	1	1	mobile
2	2	2	mobile
3	3	1	mobile
4	3	3	mobile
5	3	1	mobile
6	4	4	desktop
7	5	5	mobile
8	6	6	mobile
9	7	7	desktop
10	8	7	desktop
11	9	7	desktop
12	10	1	mobile
13	11	8	desktop
14	2	2	mobile
15	2	2	mobile
16	8	7	desktop
17	3	0	mobile
18	12	9	desktop
19	4	10	desktop

	Login Successful	Is Attack IP	Is Account Takeover	Login Hour
0	0	0	0	12
1	0	0	0	12
2	1	0	0	12
3	0	0	0	12

```
4                    0             1                    0          12
5                    0             1                    0          12
6                    1             0                    0          12
7                    0             0                    0          12
8                    0             0                    0          12
9                    1             0                    0          12
10                   1             0                    0          12
11                   1             0                    0          12
12                   0             0                    0          12
13                   0             0                    0          12
14                   0             0                    0          12
15                   1             0                    0          12
16                   1             0                    0          12
17                   0             0                    0          12
18                   0             0                    0          12
19                   1             0                    0          12
```

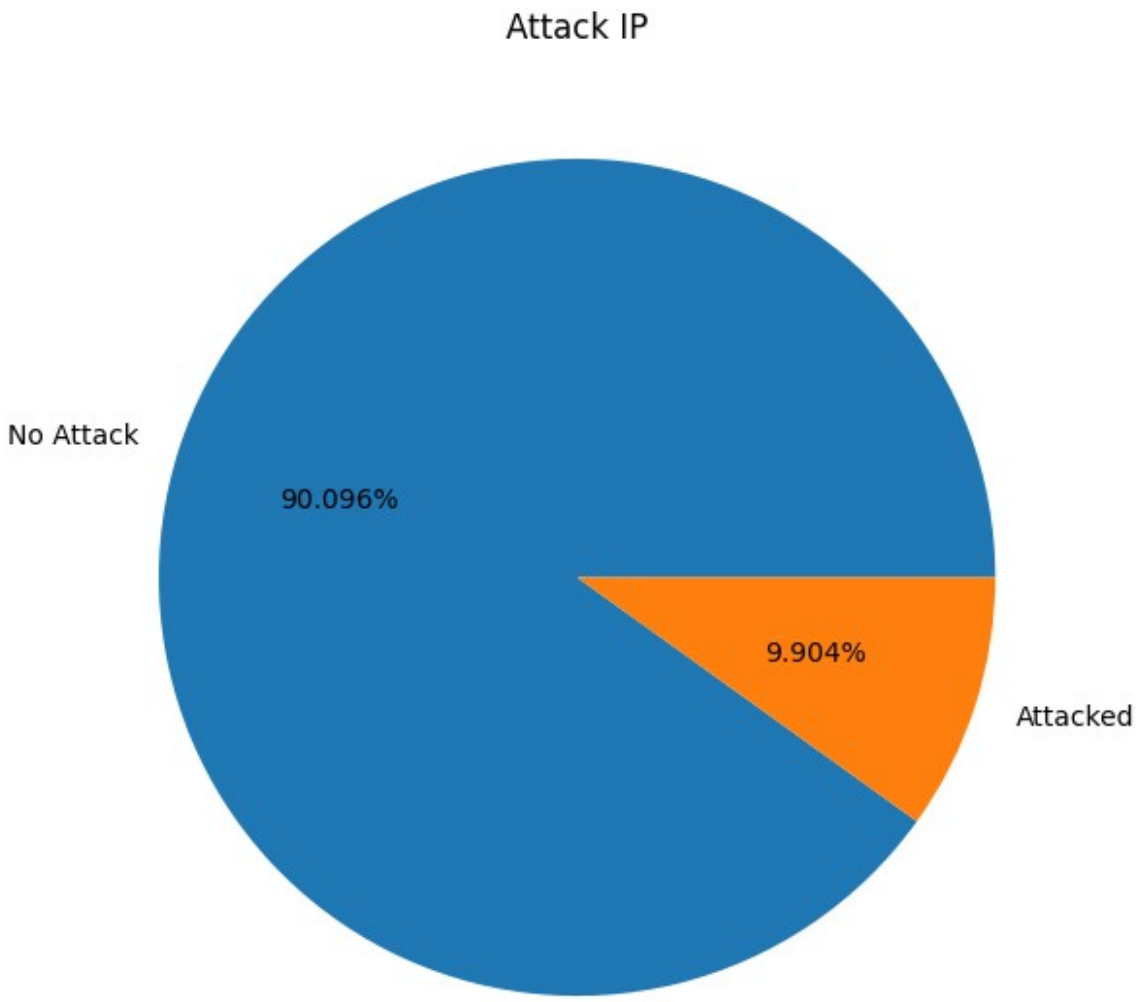
```
account_takeover_rows = data[data['Is Account Takeover'] == 1]

# Display or further process the filtered rows
account_takeover_rows
```

```
                       User ID  IP Address Country     ASN  User Agent
String  \
82873     5519106287451092780   168034722      IT  503109
2611
82947    -7654599524478640403   168034722      IT  503109
2611
100085   -6380256063165146454   528683032      RO   56851
10118
202905    4130074439166519892  3114960900      IT  206801
36
273968    -136955930917892295   167793933      NO  197475
11
...                       ...         ...     ...     ...
...
20623917  -249028206650900290  1541160943      RO  197357
324772
20700950 -1639909578889655226    95726278      RO  206801
174628
21266387 -4655135911852100550   168018399      BR  500106
2
21497310 -1999335758853878070  3000358111      RO   31028
174628
21734204   832942564942319679    37267096      RO    3280
2611

          Browser Name and Version  OS Name and Version Device Type  \
82873                           36                    7     desktop
82947                           36                    7     desktop
```

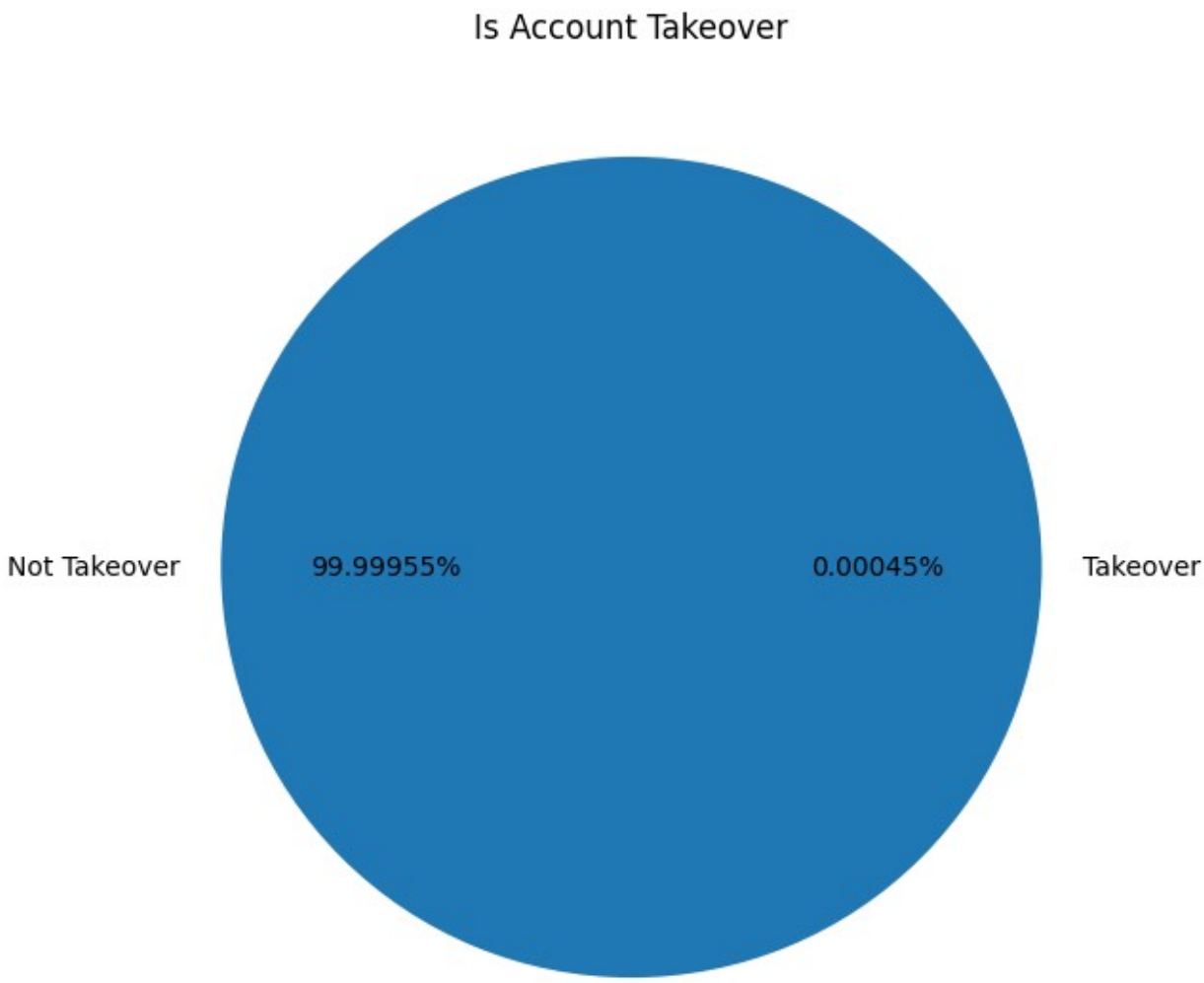
```
100085                          180                   10     desktop
202905                           24                   18      mobile
273968                            9                    7     desktop
...                             ...                  ...         ...
20623917                       3292                    0      mobile
20700950                       2316                    7     desktop
21266387                          2                    2      mobile
21497310                       2316                    7     desktop
21734204                         36                    7     desktop

          Login Successful  Is Attack IP  Is Account Takeover  Login
Hour
82873                    1             0                    1
13
82947                    1             0                    1
13
100085                   1             1                    1
17
202905                   1             0                    1
5
273968                   1             0                    1
1
...                    ...           ...                  ...
...
20623917                 1             1                    1
8
20700950                 1             1                    1
7
21266387                 1             0                    1
20
21497310                 1             1                    1
18
21734204                 1             1                    1
23

[141 rows x 12 columns]
```

```
categorical_cols = ['Country', 'Device Type']
numeric_cols = ['ASN', 'Login Hour', 'IP Address', 'User Agent
String', 'Browser Name and Version', 'OS Name and Version']

"""percentage wise calculations
"""
percentage1=data['Is Attack IP'].value_counts(normalize=True)*100
percentage2=data['Is Account
Takeover'].value_counts(normalize=True)*100
percentage3=data['Login Successful'].value_counts(normalize=True)*100

classlabels=["No Attack","Attacked"]
plt.figure(figsize=(12,7))
```

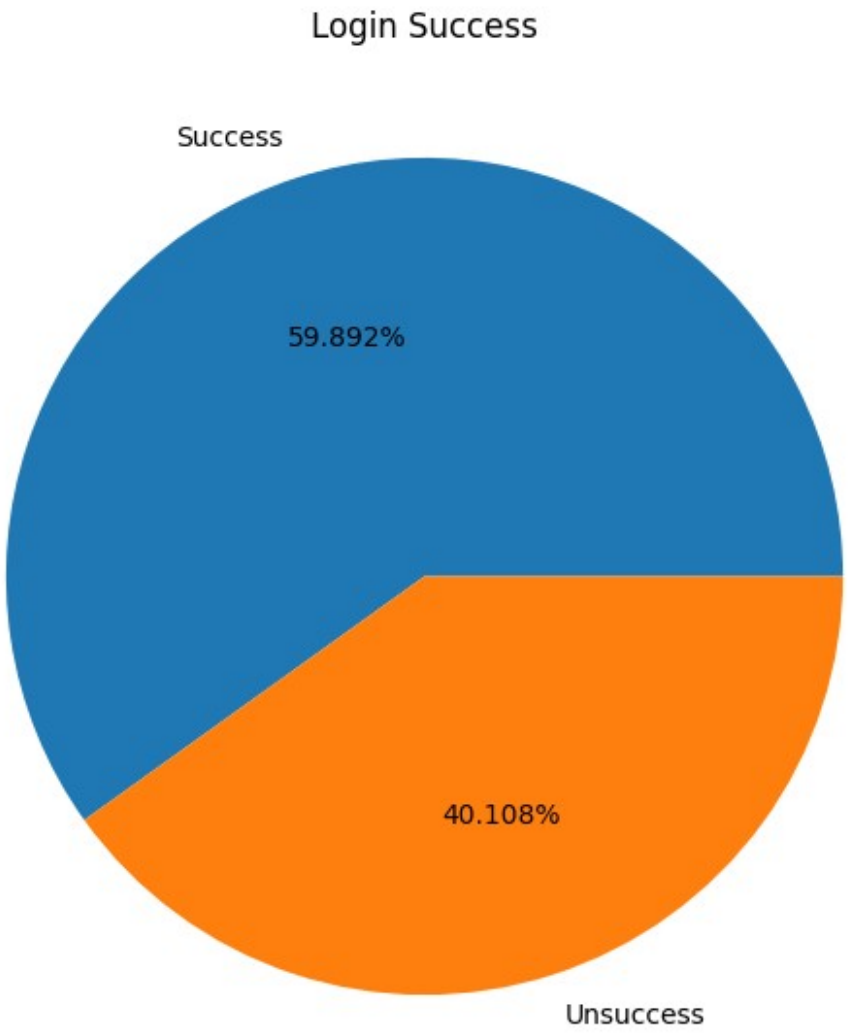
```
plt.pie(percentage1,labels=classlabels,autopct='%1.3f%%')
plt.title("Attack IP")
plt.show()
```




```
classlabels=["Not Takeover", "Takeover"]
plt.figure(figsize=(12,7))
plt.pie(percentage2,labels=classlabels,autopct='%1.5f%%')
plt.title("Is Account Takeover")
plt.show()
```




```
classlabels=["Success","Unsuccess"]
plt.figure(figsize=(12,7))
plt.pie(percentage3,labels=classlabels,autopct='%1.3f%%')
plt.title("Login Success")
plt.show()
```




```
# Splitting the dataset
features = data.drop(['Is Attack IP', 'Is Account Takeover'], axis=1)
labels = data['Is Account Takeover']

X_train, X_test, y_train, y_test = train_test_split(features, labels,
test_size=0.2, random_state=42,stratify=labels)

preprocessor = ColumnTransformer(
    transformers=[
        ('num', StandardScaler(), numeric_cols),
        ('cat', OneHotEncoder(), categorical_cols)
    ])

# Classifiers
classifiers = {
    'logistic_regression': LogisticRegression(max_iter=1000),
```

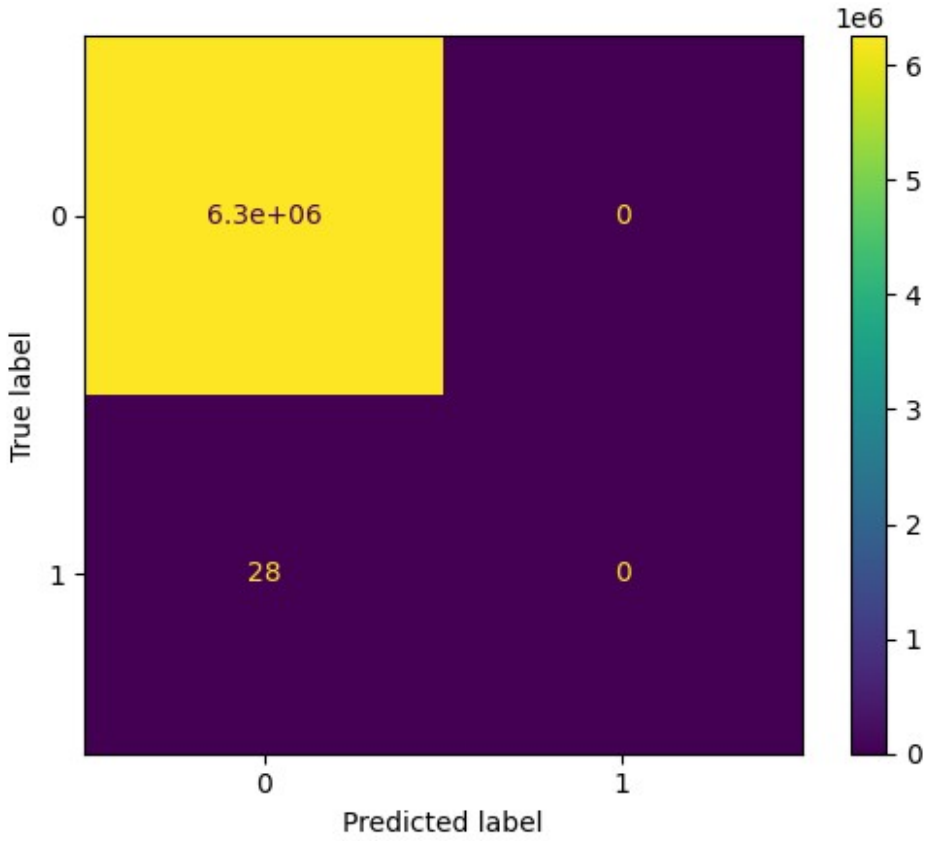
```
    'decision_tree': DecisionTreeClassifier(),
    'svm': SVC(probability=True),
    'random_forest': RandomForestClassifier(),
    'Adaboost' : AdaBoostClassifier(),
    'Extra' :  ExtraTreesClassifier(),
    'lgbm' : LGBMClassifier(),
    'XGB': XGBClassifier()
}

# A function to choose classifiers
def make_pipeline(classifier_key):
    if classifier_key in classifiers:
        clf = Pipeline(steps=[
            ('preprocessor', preprocessor),
            ('classifier', classifiers[classifier_key])
        ])
        return clf
    else:
        raise ValueError(f"Classifier {classifier_key} is not
defined")

from sklearn.metrics import ConfusionMatrixDisplay

classifier_key = 'logistic_regression'
pipeline = make_pipeline(classifier_key)
pipeline.fit(X_train, y_train)

# Evaluation
lrpredictions = pipeline.predict(X_test)
probs = pipeline.predict_proba(X_test)[:, 1]
auc_score = roc_auc_score(y_test, probs)

print(f"AUC Score: {auc_score}")
```

```
AUC Score: 0.867026585443994
```

```
from sklearn.metrics import classification_report
from sklearn.metrics import accuracy_score

Score = accuracy_score(y_test,lrpredictions)
Classification_Report = classification_report(y_test,lrpredictions)

print("Logistic Regression")
print ("Accuracy Score value: {:.4f}".format(Score))
print (Classification_Report)
```

```
/Users/kaushalkento/Desktop/GroupProject./RealTimeProject/
CaptchaRealTimeProject/.venv/lib/python3.12/site-packages/sklearn/
metrics/_classification.py:1565: UndefinedMetricWarning: Precision is
ill-defined and being set to 0.0 in labels with no predicted samples.
Use `zero_division` parameter to control this behavior.
```

```
  _warn_prf(average, modifier, f"{metric.capitalize()} is",
len(result))
/Users/kaushalkento/Desktop/GroupProject./RealTimeProject/CaptchaRealT
imeProject/.venv/lib/python3.12/site-packages/sklearn/metrics/
_classification.py:1565: UndefinedMetricWarning: Precision is ill-
defined and being set to 0.0 in labels with no predicted samples. Use
`zero_division` parameter to control this behavior.
  _warn_prf(average, modifier, f"{metric.capitalize()} is",
len(result))

Logistic Regression
Accuracy Score value: 1.0000
              precision    recall  f1-score   support

           0       1.00      1.00      1.00   6253825
           1       0.00      0.00      0.00        28

    accuracy                           1.00   6253853
   macro avg       0.50      0.50      0.50   6253853
weighted avg       1.00      1.00      1.00   6253853


/Users/kaushalkento/Desktop/GroupProject./RealTimeProject/
CaptchaRealTimeProject/.venv/lib/python3.12/site-packages/sklearn/
metrics/_classification.py:1565: UndefinedMetricWarning: Precision is
ill-defined and being set to 0.0 in labels with no predicted samples.
Use `zero_division` parameter to control this behavior.
  _warn_prf(average, modifier, f"{metric.capitalize()} is",
len(result))

Logistic_Regression_Confusion_Matrix =
ConfusionMatrixDisplay.from_estimator(pipeline, X_test, y_test)
Logistic_Regression_Confusion_Matrix
plt.show()
```




```
classifier_key = 'decision_tree'
pipeline = make_pipeline(classifier_key)
pipeline.fit(X_train, y_train)

# Evaluation
dtpredictions = pipeline.predict(X_test)
probs = pipeline.predict_proba(X_test)[:, 1]
auc_score = roc_auc_score(y_test, probs)

print(f"AUC Score: {auc_score}")
```

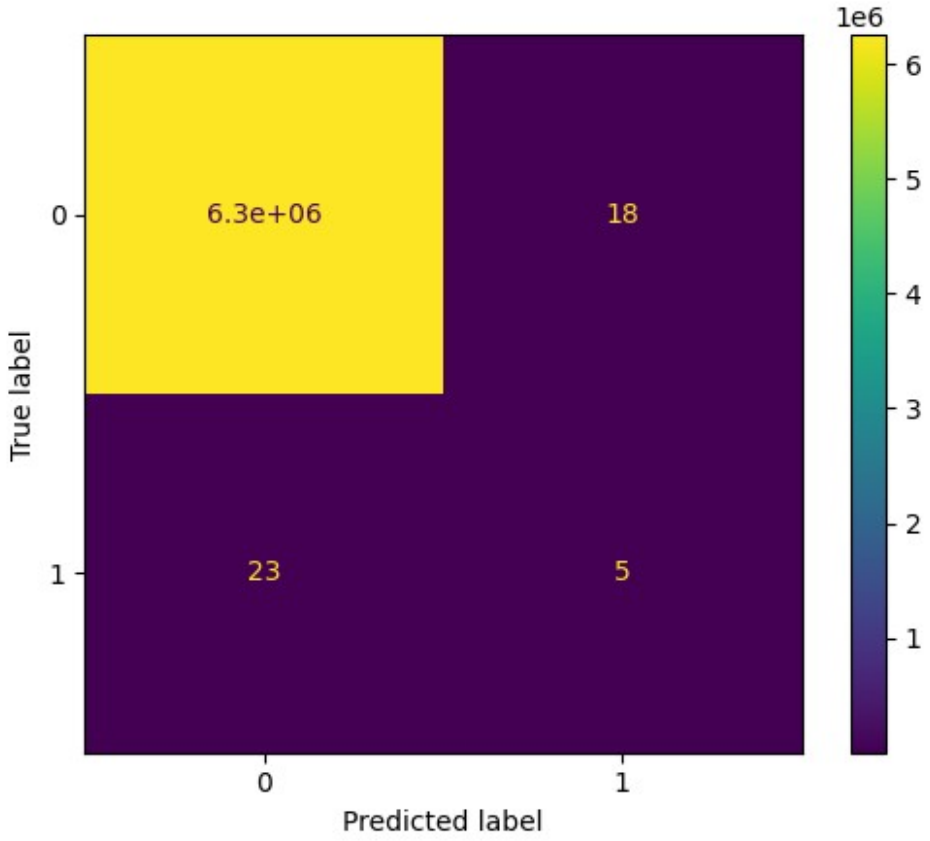
```
AUC Score: 0.6249973444823197
```

```
Score = accuracy_score(y_test,dtpredictions)
Classification_Report = classification_report(y_test,dtpredictions)

print("Decision Tree")
print ("Accuracy Score value: {:.8f}".format(Score))
print (Classification_Report)
```

```
Decision Tree
Accuracy Score value: 0.99999344
              precision    recall  f1-score   support
```

```
              0       1.00      1.00      1.00   6253825
              1       0.22      0.18      0.20        28

    accuracy                            1.00   6253853
   macro avg       0.61      0.59      0.60   6253853
weighted avg       1.00      1.00      1.00   6253853
```

```
Logistic_Regression_Confusion_Matrix =
ConfusionMatrixDisplay.from_estimator(pipeline, X_test, y_test)
Logistic_Regression_Confusion_Matrix
plt.show()
```



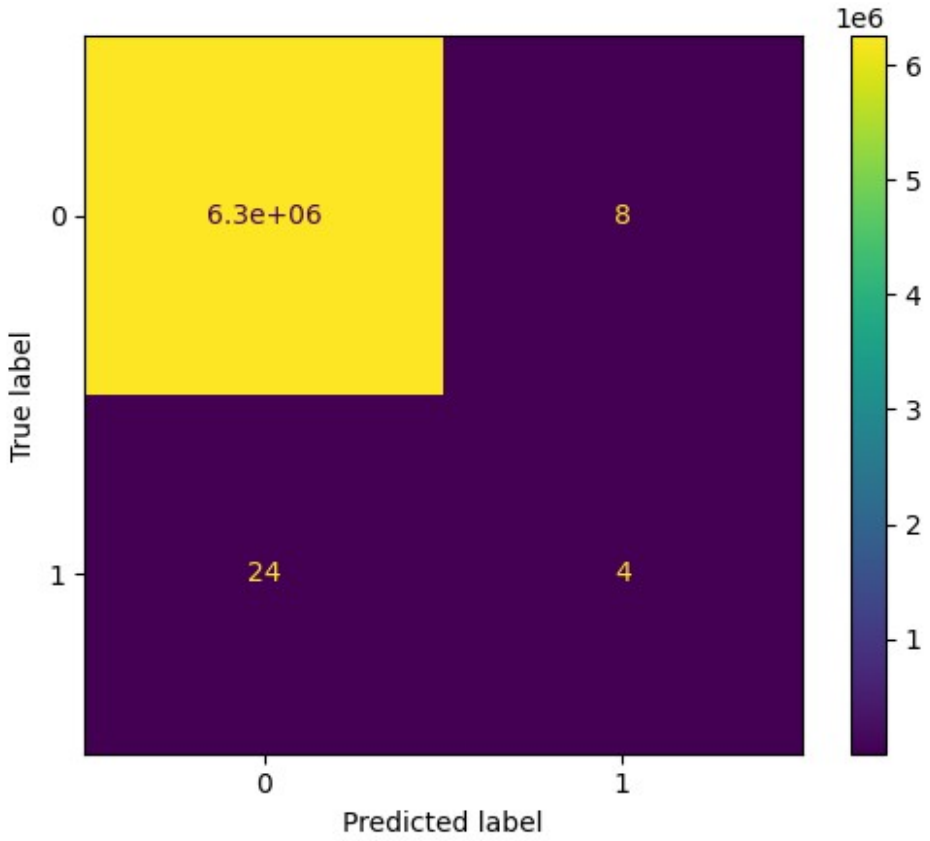

```
classifier_key = 'random_forest'
pipeline = make_pipeline(classifier_key)
pipeline.fit(X_train, y_train)

# Evaluation
predictions = pipeline.predict(X_test)
probs = pipeline.predict_proba(X_test)[:, 1]
auc_score = roc_auc_score(y_test, probs)
print(f"AUC Score: {auc_score}")
```

```
AUC Score: 0.8571237317047682
```

```
Score = accuracy_score(y_test,predictions)
Classification_Report = classification_report(y_test,predictions)

print("Random Forest")
print ("Accuracy Score value: {:.8f}".format(Score))
print (Classification_Report)
```

```
Random Forest
Accuracy Score value: 0.99999488
              precision    recall  f1-score   support

           0       1.00      1.00      1.00   6253825
           1       0.33      0.14      0.20        28

    accuracy                           1.00   6253853
   macro avg       0.67      0.57      0.60   6253853
weighted avg       1.00      1.00      1.00   6253853
```

```
Logistic_Regression_Confusion_Matrix =
ConfusionMatrixDisplay.from_estimator(pipeline, X_test, y_test)
Logistic_Regression_Confusion_Matrix
plt.show()
```



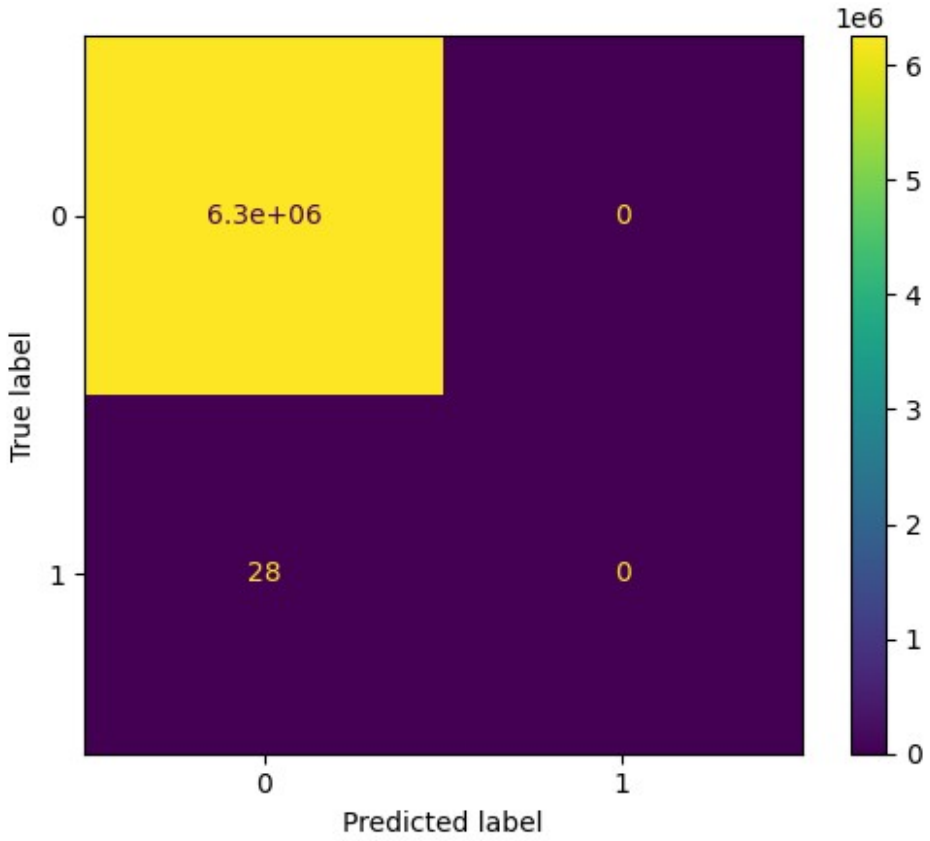

```
from sklearn.metrics import ConfusionMatrixDisplay

classifier_key = 'Adaboost'
pipeline = make_pipeline(classifier_key)
pipeline.fit(X_train, y_train)

# Evaluation
abpredictions = pipeline.predict(X_test)
probs = pipeline.predict_proba(X_test)[:, 1]
auc_score = roc_auc_score(y_test, probs)

print(f"AUC Score: {auc_score}")
```

```
AUC Score: 0.978833605262151
```

```
Score = accuracy_score(y_test,abpredictions)
Classification_Report = classification_report(y_test,abpredictions)

print("Ada Boost")
print ("Accuracy Score value: {:.8f}".format(Score))
print (Classification_Report)
```

```
/Users/kaushalkento/Desktop/GroupProject./RealTimeProject/
CaptchaRealTimeProject/.venv/lib/python3.12/site-packages/sklearn/
```

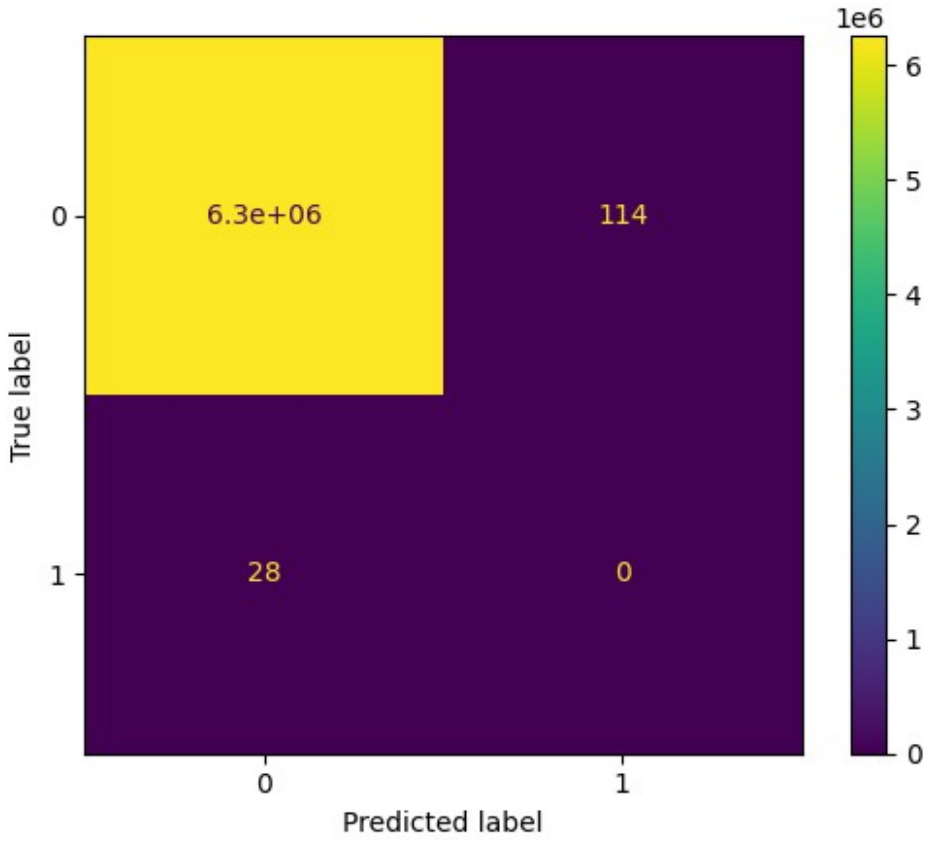
```
metrics/_classification.py:1565: UndefinedMetricWarning: Precision is
ill-defined and being set to 0.0 in labels with no predicted samples.
Use `zero_division` parameter to control this behavior.
  _warn_prf(average, modifier, f"{metric.capitalize()} is",
len(result))
/Users/kaushalkento/Desktop/GroupProject./RealTimeProject/CaptchaRealT
imeProject/.venv/lib/python3.12/site-packages/sklearn/metrics/
_classification.py:1565: UndefinedMetricWarning: Precision is ill-
defined and being set to 0.0 in labels with no predicted samples. Use
`zero_division` parameter to control this behavior.
  _warn_prf(average, modifier, f"{metric.capitalize()} is",
len(result))

Ada Boost
Accuracy Score value: 0.99999552
              precision    recall  f1-score   support

           0       1.00      1.00      1.00   6253825
           1       0.00      0.00      0.00        28

    accuracy                           1.00   6253853
   macro avg       0.50      0.50      0.50   6253853
weighted avg       1.00      1.00      1.00   6253853


/Users/kaushalkento/Desktop/GroupProject./RealTimeProject/
CaptchaRealTimeProject/.venv/lib/python3.12/site-packages/sklearn/
metrics/_classification.py:1565: UndefinedMetricWarning: Precision is
ill-defined and being set to 0.0 in labels with no predicted samples.
Use `zero_division` parameter to control this behavior.
  _warn_prf(average, modifier, f"{metric.capitalize()} is",
len(result))

Logistic_Regression_Confusion_Matrix =
ConfusionMatrixDisplay.from_estimator(pipeline, X_test, y_test)
Logistic_Regression_Confusion_Matrix
plt.show()
```




```
classifier_key = 'lgbm'
pipeline = make_pipeline(classifier_key)
pipeline.fit(X_train, y_train)

# Evaluation
lgbmpredictions = pipeline.predict(X_test)
probs = pipeline.predict_proba(X_test)[:, 1]
auc_score = roc_auc_score(y_test, probs)

print(f"AUC Score: {auc_score}")
```

```
/Users/kaushalkento/Desktop/GroupProject./RealTimeProject/
CaptchaRealTimeProject/.venv/lib/python3.12/site-packages/sklearn/
utils/deprecation.py:151: FutureWarning: 'force_all_finite' was
renamed to 'ensure_all_finite' in 1.6 and will be removed in 1.8.
  warnings.warn(

[LightGBM] [Info] Number of positive: 113, number of negative:
25015298
[LightGBM] [Info] Auto-choosing row-wise multi-threading, the overhead
of testing was 0.107935 seconds.
You can set `force_row_wise=true` to remove the overhead.
And if memory is not enough, you can set `force_col_wise=true`.
[LightGBM] [Info] Total Bins 1624
```

```
[LightGBM] [Info] Number of data points in the train set: 25015411,
number of used features: 199
[LightGBM] [Info] [binary:BoostFromScore]: pavg=0.000005 ->
initscore=-12.307610
[LightGBM] [Info] Start training from score -12.307610

/Users/kaushalkento/Desktop/GroupProject./RealTimeProject/
CaptchaRealTimeProject/.venv/lib/python3.12/site-packages/sklearn/
utils/deprecation.py:151: FutureWarning: 'force_all_finite' was
renamed to 'ensure_all_finite' in 1.6 and will be removed in 1.8.
  warnings.warn(
/Users/kaushalkento/Desktop/GroupProject./RealTimeProject/CaptchaRealT
imeProject/.venv/lib/python3.12/site-packages/sklearn/utils/
deprecation.py:151: FutureWarning: 'force_all_finite' was renamed to
'ensure_all_finite' in 1.6 and will be removed in 1.8.
  warnings.warn(

AUC Score: 0.49565553881024815

Score = accuracy_score(y_test,lgbmpredictions)
Classification_Report = classification_report(y_test,lgbmpredictions)

print("LGBM")
print ("Accuracy Score value: {:.8f}".format(Score))
print (Classification_Report)

LGBM
Accuracy Score value: 0.99997729
              precision    recall  f1-score   support

           0       1.00      1.00      1.00   6253825
           1       0.00      0.00      0.00        28

    accuracy                           1.00   6253853
   macro avg       0.50      0.50      0.50   6253853
weighted avg       1.00      1.00      1.00   6253853


Logistic_Regression_Confusion_Matrix =
ConfusionMatrixDisplay.from_estimator(pipeline, X_test, y_test)
Logistic_Regression_Confusion_Matrix
plt.show()

/Users/kaushalkento/Desktop/GroupProject./RealTimeProject/
CaptchaRealTimeProject/.venv/lib/python3.12/site-packages/sklearn/
utils/deprecation.py:151: FutureWarning: 'force_all_finite' was
renamed to 'ensure_all_finite' in 1.6 and will be removed in 1.8.
  warnings.warn(
```




```
classifier_key = 'XGB'
pipeline = make_pipeline(classifier_key)
pipeline.fit(X_train, y_train)

# Evaluation
xgbpredictions = pipeline.predict(X_test)
probs = pipeline.predict_proba(X_test)[:, 1]
auc_score = roc_auc_score(y_test, probs)

print(f"AUC Score: {auc_score}")
```

```
AUC Score: 0.9847150486759245
```

```
Score = accuracy_score(y_test,xgbpredictions)
Classification_Report = classification_report(y_test,xgbpredictions)

print("LGBM")
print ("Accuracy Score value: {:.8f}".format(Score))
print (Classification_Report)
```

```
LGBM
Accuracy Score value: 0.99999536
              precision    recall  f1-score   support
```

```
              0       1.00      1.00      1.00   6253825
              1       0.33      0.04      0.06        28

    accuracy                           1.00   6253853
   macro avg       0.67      0.52      0.53   6253853
weighted avg       1.00      1.00      1.00   6253853
```

```
Logistic_Regression_Confusion_Matrix =
ConfusionMatrixDisplay.from_estimator(pipeline, X_test, y_test)
Logistic_Regression_Confusion_Matrix
plt.show()
```