

```
In [1]: import pandas as pd
```

```
import numpy as np
```

```
import matplotlib.pyplot as plt
```

```
import seaborn as sns
```

```
In [37]: from sklearn.model_selection import train_test_split, GridSearchCV
from sklearn.ensemble import RandomForestClassifier, GradientBoostingClassifier, AdaBo
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import classification_report, confusion_matrix, roc_curve, auc
from sklearn.metrics import accuracy_score
```

```
In [3]: # Load your dataset
```

```
data = pd.read_csv('Churn_Modelling.csv')
```

```
In [5]: data
```

```
Out[5]:
```

	RowNumber	CustomerId	Surname	CreditScore	Geography	Gender	Age	Tenure	Balance
0	1	15634602	Hargrave	619	France	Female	42	2	0.00
1	2	15647311	Hill	608	Spain	Female	41	1	83807.86
2	3	15619304	Onio	502	France	Female	42	8	159660.80
3	4	15701354	Boni	699	France	Female	39	1	0.00
4	5	15737888	Mitchell	850	Spain	Female	43	2	125510.82
...	...	...	...	...	...	...	...	...	...
9995	9996	15606229	Obijjaku	771	France	Male	39	5	0.00
9996	9997	15569892	Johnstone	516	France	Male	35	10	57369.61
9997	9998	15584532	Liu	709	France	Female	36	7	0.00
9998	9999	15682355	Sabbatini	772	Germany	Male	42	3	75075.31
9999	10000	15628319	Walker	792	France	Female	28	4	130142.79

10000 rows × 14 columns

```
In [6]: # Display the first few rows of the dataset
print(data.head())
```

```

      RowNumber CustomerId Surname CreditScore Geography Gender Age \
0            1    15634602 Hargrave       619   France Female  42
1            2    15647311     Hill        608   Spain Female  41
2            3    15619304    Onio         502   France Female  42
3            4    15701354    Boni         699   France Female  39
4            5    15737888 Mitchell       850   Spain Female  43

      Tenure Balance NumOfProducts HasCrCard IsActiveMember \
0       2    0.00             1          1              1
1       1  83807.86             1          0              1
2       8  159660.80             3          1              0
3       1    0.00             2          0              0
4       2  125510.82             1          1              1

      EstimatedSalary Exited
0      101348.88      1
1      112542.58      0
2      113931.57      1
3      93826.63       0
4      79084.10       0

```

In [7]: `# Check for missing values  
print(data.isnull().sum())`

```

RowNumber      0
CustomerId     0
Surname        0
CreditScore    0
Geography      0
Gender         0
Age            0
Tenure         0
Balance        0
NumOfProducts  0
HasCrCard      0
IsActiveMember 0
EstimatedSalary 0
Exited         0
dtype: int64

```

In [8]: `data.info()`

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 10000 entries, 0 to 9999
Data columns (total 14 columns):
 #   Column           Non-Null Count  Dtype  
--- 
 0   RowNumber        10000 non-null   int64  
 1   CustomerId       10000 non-null   int64  
 2   Surname          10000 non-null   object  
 3   CreditScore      10000 non-null   int64  
 4   Geography         Geography        object  
 5   Gender            10000 non-null   object  
 6   Age               10000 non-null   int64  
 7   Tenure            10000 non-null   int64  
 8   Balance           10000 non-null   float64 
 9   NumOfProducts     10000 non-null   int64  
 10  HasCrCard        10000 non-null   int64  
 11  IsActiveMember    IsActiveMember  int64  
 12  EstimatedSalary   EstimatedSalary float64 
 13  Exited           10000 non-null   int64  
dtypes: float64(2), int64(9), object(3)
memory usage: 1.1+ MB
```

In [9]:

```
# Checking for null values in each column
null_values_per_column = data.isnull().sum()
print("Null values in each column:\n", null_values_per_column)
```

Null values in each column:

RowNumber	0
CustomerId	0
Surname	0
CreditScore	0
Geography	0
Gender	0
Age	0
Tenure	0
Balance	0
NumOfProducts	0
HasCrCard	0
IsActiveMember	0
EstimatedSalary	0
Exited	0

dtype: int64

In [10]:

```
# Define the target variable and features
X = data.drop('CreditScore', axis=1)
y = data['CreditScore']
```

In [11]:

```
# Split the dataset into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=42)
```

In [12]:

```
# Align the train and test sets to ensure they have the same columns
X_train, X_test = X_train.align(X_test, join='left', axis=1, fill_value=0)
```

In [13]:

```
print(X_train.isnull().sum().sum())
print(y_train.isnull().sum().sum())
```

0  
0

```
In [14]: print(X_train.dtypes)
```

```
RowNumber           int64
CustomerId         int64
Surname            object
Geography          object
Gender              object
Age                int64
Tenure              int64
Balance             float64
NumOfProducts       int64
HasCrCard           int64
IsActiveMember     int64
EstimatedSalary    float64
Exited              int64
dtype: object
```

```
In [15]: X_train = pd.get_dummies(X_train, drop_first=True)
X_test = pd.get_dummies(X_test, drop_first=True)

# Align columns in case train and test sets have different columns
X_train, X_test = X_train.align(X_test, join='left', axis=1, fill_value=0)
```

```
In [16]: print(X_train.shape)
print(y_train.shape)
```

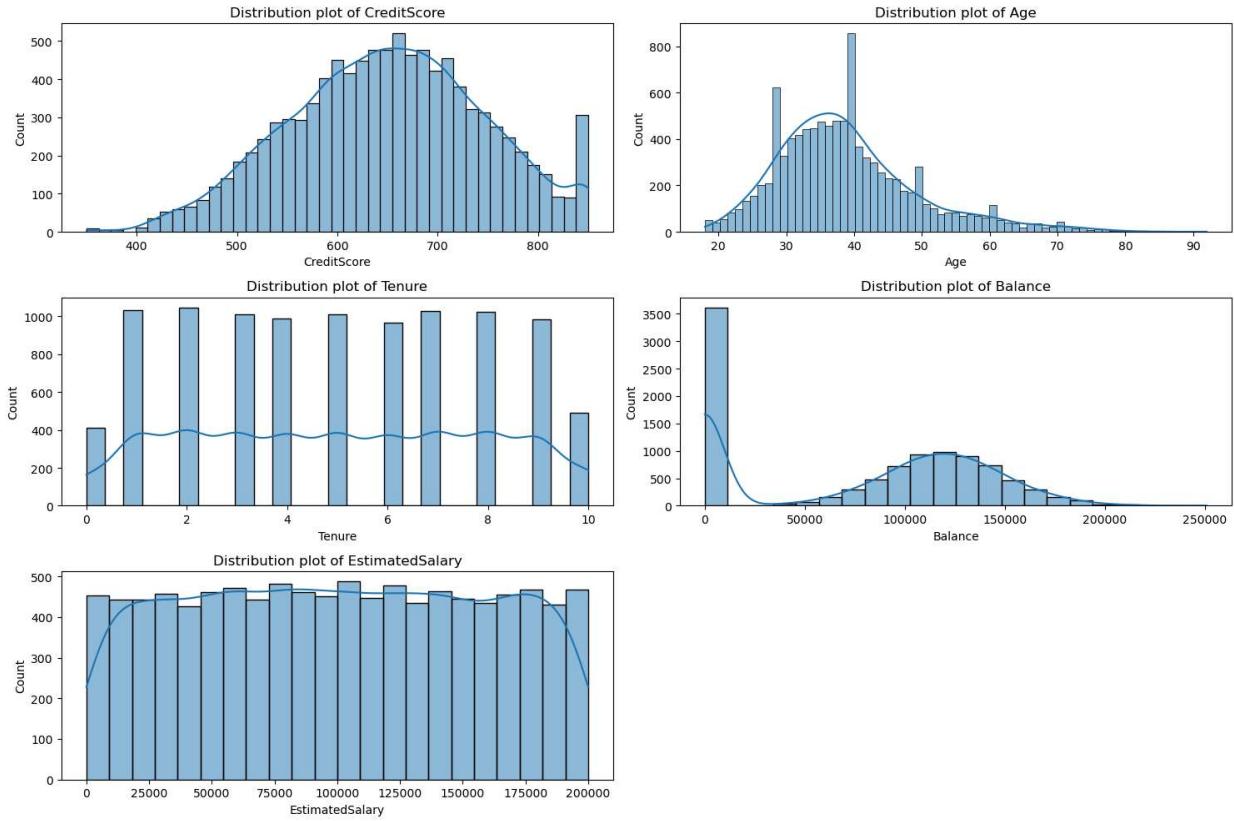
```
(7000, 2442)
(7000,)
```

```
# Generate plots for the specified columns
columns_to_plot = ['CreditScore', 'Age', 'Tenure', 'Balance', 'EstimatedSalary']

plt.figure(figsize=(15, 10))

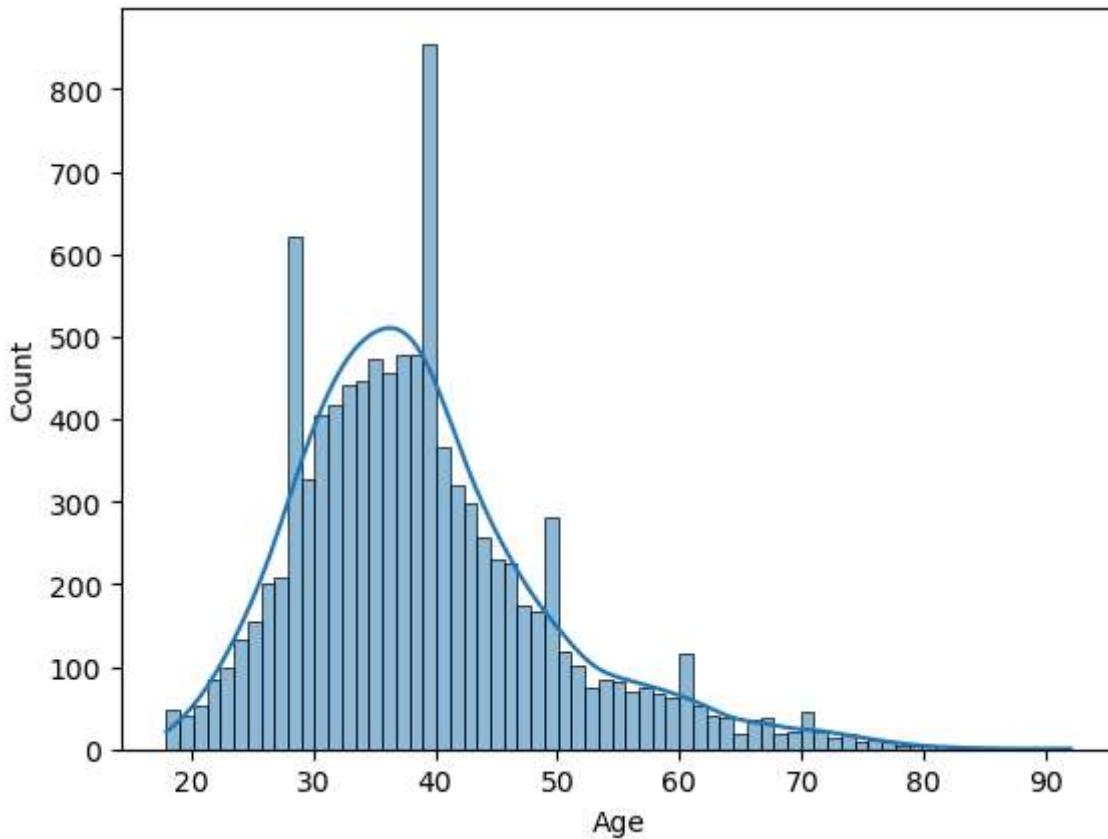
for i, column in enumerate(columns_to_plot, 1):
    plt.subplot(3, 2, i) # Adjust subplot grid if needed
    sns.histplot(data[column], kde=True)
    plt.title(f'Distribution plot of {column}')

plt.tight_layout()
plt.show()
```



```
In [18]: # Visualize data distributions
sns.histplot(data['Age'], kde=True)
plt.show()

sns.countplot(x='Churn', data=df)
plt.show()
```



**NameError**

Traceback (most recent call last)

```
Cell In[18], line 5
    2 sns.histplot(data['Age'], kde=True)
    3 plt.show()
----> 5 sns.countplot(x='Churn', data=df)
    6 plt.show()
```

**NameError: name 'df' is not defined**

```
In [19]: print("Columns in DataFrame:", data.columns)
```

```
# Print the first few rows of the DataFrame
print("DataFrame Head:\n", data.head())
```

```
# Ensure there are no Leading/trailing spaces in column names
data.columns = data.columns.str.strip()
```

```
# Verify if 'Churn' is in the columns
if 'Churn' not in data.columns:
    print('Churn')
else:
    # Proceed with your existing code
    features = ['Age', 'Gender', 'CustomerId', 'CreditScore', 'HasCrCard', 'Exited']
    X = data[features]
    y = data['Churn']
```

```
Columns in DataFrame: Index(['RowNumber', 'CustomerId', 'Surname', 'CreditScore', 'Geography',
   'Gender', 'Age', 'Tenure', 'Balance', 'NumOfProducts', 'HasCrCard',
   'IsActiveMember', 'EstimatedSalary', 'Exited'],
  dtype='object')
```

DataFrame Head:

	RowNumber	CustomerId	Surname	CreditScore	Geography	Gender	Age
0	1	15634602	Hargrave	619	France	Female	42
1	2	15647311	Hill	608	Spain	Female	41
2	3	15619304	Onio	502	France	Female	42
3	4	15701354	Boni	699	France	Female	39
4	5	15737888	Mitchell	850	Spain	Female	43

	Tenure	Balance	NumOfProducts	HasCrCard	IsActiveMember
0	2	0.00	1	1	1
1	1	83807.86	1	0	1
2	8	159660.80	3	1	0
3	1	0.00	2	0	0
4	2	125510.82	1	1	1

	EstimatedSalary	Exited
0	101348.88	1
1	112542.58	0
2	113931.57	1
3	93826.63	0
4	79084.10	0

Churn

```
In [20]: # Initialize and train the Random Forest classifier
rf = RandomForestClassifier(random_state=42)
rf.fit(X_train, y_train)
```

```
Out[20]: ▾ RandomForestClassifier
```

```
RandomForestClassifier(random_state=42)
```

```
In [21]: # Make predictions on the test set  
y_pred = rf.predict(X_test)
```

```
In [23]: # Print the classification report  
print(classification_report(y_test, y_pred))
```

	precision	recall	f1-score	support
350	0.00	0.00	0.00	2
351	0.00	0.00	0.00	1
358	0.00	0.00	0.00	1
359	0.00	0.00	0.00	1
363	0.00	0.00	0.00	1
373	0.00	0.00	0.00	1
376	0.00	0.00	0.00	0
383	0.00	0.00	0.00	0
399	0.00	0.00	0.00	1
401	0.00	0.00	0.00	1
404	0.00	0.00	0.00	1
405	0.00	0.00	0.00	1
408	0.00	0.00	0.00	0
410	0.00	0.00	0.00	1
411	0.00	0.00	0.00	2
412	0.00	0.00	0.00	1
413	0.00	0.00	0.00	2
414	0.00	0.00	0.00	1
415	0.00	0.00	0.00	1
416	0.00	0.00	0.00	2
418	0.00	0.00	0.00	1
420	0.00	0.00	0.00	1
421	0.00	0.00	0.00	2
422	0.00	0.00	0.00	3
423	0.00	0.00	0.00	1
425	0.00	0.00	0.00	1
426	0.00	0.00	0.00	2
427	0.00	0.00	0.00	3
429	0.00	0.00	0.00	2
430	0.00	0.00	0.00	2
431	0.00	0.00	0.00	1
432	0.00	0.00	0.00	0
433	0.00	0.00	0.00	1
434	0.00	0.00	0.00	1
435	0.00	0.00	0.00	0
437	0.00	0.00	0.00	1
438	0.00	0.00	0.00	0
439	0.00	0.00	0.00	1
440	0.00	0.00	0.00	1
441	0.00	0.00	0.00	0
443	0.00	0.00	0.00	1
444	0.00	0.00	0.00	1
445	0.00	0.00	0.00	7
446	0.00	0.00	0.00	0
447	0.00	0.00	0.00	2
448	0.00	0.00	0.00	2
449	0.00	0.00	0.00	3
450	0.00	0.00	0.00	2
451	0.00	0.00	0.00	1
452	0.00	0.00	0.00	3
453	0.00	0.00	0.00	2
454	0.00	0.00	0.00	1
455	0.00	0.00	0.00	1
456	0.00	0.00	0.00	2
457	0.00	0.00	0.00	1
458	0.00	0.00	0.00	2
459	0.00	0.00	0.00	1
460	0.00	0.00	0.00	2

461	0.00	0.00	0.00	4
462	0.00	0.00	0.00	1
463	0.00	0.00	0.00	0
464	0.00	0.00	0.00	2
465	0.00	0.00	0.00	4
466	0.00	0.00	0.00	1
467	0.00	0.00	0.00	2
468	0.00	0.00	0.00	2
469	0.00	0.00	0.00	3
470	0.00	0.00	0.00	3
471	0.00	0.00	0.00	1
472	0.00	0.00	0.00	2
473	0.00	0.00	0.00	3
474	0.00	0.00	0.00	6
475	0.00	0.00	0.00	1
476	0.00	0.00	0.00	4
477	0.00	0.00	0.00	5
478	0.00	0.00	0.00	3
479	0.00	0.00	0.00	7
480	0.00	0.00	0.00	6
481	0.00	0.00	0.00	7
482	0.00	0.00	0.00	3
483	0.00	0.00	0.00	1
484	0.00	0.00	0.00	2
485	0.00	0.00	0.00	4
486	0.00	0.00	0.00	5
487	0.00	0.00	0.00	4
488	0.00	0.00	0.00	2
489	0.00	0.00	0.00	4
490	0.00	0.00	0.00	4
491	0.00	0.00	0.00	2
492	0.00	0.00	0.00	4
493	0.00	0.00	0.00	4
494	0.00	0.00	0.00	6
495	0.00	0.00	0.00	3
496	0.00	0.00	0.00	5
497	0.00	0.00	0.00	4
498	0.00	0.00	0.00	6
499	0.00	0.00	0.00	2
500	0.00	0.00	0.00	4
501	0.00	0.00	0.00	6
502	0.00	0.00	0.00	5
503	0.00	0.00	0.00	4
504	0.00	0.00	0.00	6
505	0.00	0.00	0.00	4
506	0.00	0.00	0.00	5
507	0.00	0.00	0.00	6
508	0.00	0.00	0.00	3
509	0.00	0.00	0.00	5
510	0.00	0.00	0.00	5
511	0.00	0.00	0.00	4
512	0.00	0.00	0.00	5
513	0.00	0.00	0.00	5
514	0.00	0.00	0.00	6
515	0.00	0.00	0.00	7
516	0.00	0.00	0.00	8
517	0.00	0.00	0.00	5
518	0.00	0.00	0.00	4
519	0.00	0.00	0.00	3
520	0.00	0.00	0.00	9

521	0.00	0.00	0.00	11
522	0.00	0.00	0.00	6
523	0.00	0.00	0.00	6
524	0.00	0.00	0.00	6
525	0.00	0.00	0.00	6
526	0.00	0.00	0.00	8
527	0.00	0.00	0.00	6
528	0.00	0.00	0.00	13
529	0.00	0.00	0.00	10
530	0.00	0.00	0.00	5
531	0.00	0.00	0.00	8
532	0.00	0.00	0.00	2
533	0.00	0.00	0.00	3
534	0.00	0.00	0.00	3
535	0.00	0.00	0.00	11
536	0.00	0.00	0.00	4
537	0.00	0.00	0.00	8
538	0.00	0.00	0.00	6
539	0.00	0.00	0.00	8
540	0.17	0.12	0.14	8
541	0.00	0.00	0.00	4
542	0.00	0.00	0.00	5
543	0.00	0.00	0.00	7
544	0.00	0.00	0.00	4
545	0.00	0.00	0.00	8
546	0.00	0.00	0.00	7
547	0.00	0.00	0.00	8
548	0.00	0.00	0.00	3
549	0.00	0.00	0.00	9
550	0.00	0.00	0.00	13
551	0.00	0.00	0.00	5
552	0.00	0.00	0.00	3
553	0.00	0.00	0.00	3
554	0.00	0.00	0.00	14
555	0.00	0.00	0.00	8
556	0.00	0.00	0.00	7
557	0.00	0.00	0.00	7
558	0.00	0.00	0.00	11
559	0.00	0.00	0.00	9
560	0.00	0.00	0.00	8
561	0.00	0.00	0.00	5
562	0.00	0.00	0.00	6
563	0.00	0.00	0.00	3
564	0.00	0.00	0.00	8
565	0.00	0.00	0.00	9
566	0.00	0.00	0.00	8
567	0.00	0.00	0.00	8
568	0.00	0.00	0.00	7
569	0.00	0.00	0.00	9
570	0.00	0.00	0.00	4
571	0.00	0.00	0.00	7
572	0.00	0.00	0.00	7
573	0.00	0.00	0.00	9
574	0.00	0.00	0.00	9
575	0.00	0.00	0.00	8
576	0.00	0.00	0.00	7
577	0.00	0.00	0.00	11
578	0.00	0.00	0.00	4
579	0.00	0.00	0.00	15
580	0.00	0.00	0.00	11

581	0.00	0.00	0.00	13
582	0.00	0.00	0.00	9
583	0.00	0.00	0.00	9
584	0.00	0.00	0.00	14
585	0.00	0.00	0.00	12
586	0.00	0.00	0.00	9
587	0.00	0.00	0.00	10
588	0.00	0.00	0.00	12
589	0.00	0.00	0.00	11
590	0.00	0.00	0.00	14
591	0.00	0.00	0.00	12
592	0.00	0.00	0.00	9
593	0.00	0.00	0.00	11
594	0.00	0.00	0.00	11
595	0.00	0.00	0.00	9
596	0.00	0.00	0.00	9
597	0.00	0.00	0.00	13
598	0.00	0.00	0.00	6
599	0.00	0.00	0.00	6
600	0.00	0.00	0.00	9
601	0.00	0.00	0.00	14
602	0.00	0.00	0.00	10
603	0.00	0.00	0.00	10
604	0.00	0.00	0.00	10
605	0.00	0.00	0.00	14
606	0.00	0.00	0.00	13
607	0.00	0.00	0.00	17
608	0.00	0.00	0.00	9
609	0.00	0.00	0.00	12
610	0.00	0.00	0.00	9
611	0.00	0.00	0.00	12
612	0.00	0.00	0.00	9
613	0.00	0.00	0.00	14
614	0.00	0.00	0.00	8
615	0.00	0.00	0.00	12
616	0.00	0.00	0.00	8
617	0.00	0.00	0.00	11
618	0.00	0.00	0.00	9
619	0.00	0.00	0.00	12
620	0.00	0.00	0.00	8
621	0.00	0.00	0.00	7
622	0.00	0.00	0.00	8
623	0.00	0.00	0.00	14
624	0.00	0.00	0.00	7
625	0.00	0.00	0.00	9
626	0.00	0.00	0.00	13
627	0.00	0.00	0.00	10
628	0.00	0.00	0.00	12
629	0.00	0.00	0.00	17
630	0.00	0.00	0.00	10
631	0.00	0.00	0.00	7
632	0.00	0.00	0.00	12
633	0.00	0.00	0.00	11
634	0.00	0.00	0.00	14
635	0.00	0.00	0.00	11
636	0.00	0.00	0.00	17
637	0.00	0.00	0.00	14
638	0.00	0.00	0.00	9
639	0.00	0.00	0.00	14
640	0.00	0.00	0.00	10

641	0.00	0.00	0.00	7
642	0.00	0.00	0.00	9
643	0.00	0.00	0.00	9
644	0.00	0.00	0.00	12
645	0.00	0.00	0.00	11
646	0.00	0.00	0.00	17
647	0.00	0.00	0.00	5
648	0.00	0.00	0.00	11
649	0.00	0.00	0.00	13
650	0.00	0.00	0.00	9
651	0.00	0.00	0.00	13
652	0.00	0.00	0.00	14
653	0.00	0.00	0.00	5
654	0.00	0.00	0.00	8
655	0.00	0.00	0.00	11
656	0.00	0.00	0.00	17
657	0.00	0.00	0.00	13
658	0.00	0.00	0.00	6
659	0.00	0.00	0.00	10
660	0.00	0.00	0.00	10
661	0.00	0.00	0.00	15
662	0.00	0.00	0.00	10
663	0.00	0.00	0.00	15
664	0.00	0.00	0.00	7
665	0.00	0.00	0.00	7
666	0.00	0.00	0.00	13
667	0.00	0.00	0.00	15
668	0.00	0.00	0.00	11
669	0.00	0.00	0.00	12
670	0.00	0.00	0.00	19
671	0.00	0.00	0.00	13
672	0.09	0.17	0.12	6
673	0.00	0.00	0.00	11
674	0.00	0.00	0.00	6
675	0.00	0.00	0.00	13
676	0.00	0.00	0.00	12
677	0.00	0.00	0.00	13
678	0.00	0.00	0.00	17
679	0.00	0.00	0.00	17
680	0.00	0.00	0.00	15
681	0.00	0.00	0.00	14
682	0.00	0.00	0.00	17
683	0.00	0.00	0.00	17
684	0.00	0.00	0.00	11
685	0.00	0.00	0.00	10
686	0.00	0.00	0.00	17
687	0.00	0.00	0.00	15
688	0.00	0.00	0.00	14
689	0.00	0.00	0.00	8
690	0.00	0.00	0.00	14
691	0.08	0.08	0.08	12
692	0.00	0.00	0.00	11
693	0.00	0.00	0.00	9
694	0.00	0.00	0.00	12
695	0.00	0.00	0.00	8
696	0.00	0.00	0.00	9
697	0.00	0.00	0.00	7
698	0.00	0.00	0.00	15
699	0.00	0.00	0.00	9
700	0.00	0.00	0.00	11

701	0.00	0.00	0.00	10
702	0.00	0.00	0.00	11
703	0.00	0.00	0.00	11
704	0.00	0.00	0.00	13
705	0.00	0.00	0.00	12
706	0.00	0.00	0.00	10
707	0.00	0.00	0.00	11
708	0.00	0.00	0.00	8
709	0.00	0.00	0.00	17
710	0.00	0.00	0.00	16
711	0.00	0.00	0.00	12
712	0.00	0.00	0.00	10
713	0.00	0.00	0.00	7
714	0.00	0.00	0.00	11
715	0.00	0.00	0.00	9
716	0.00	0.00	0.00	9
717	0.00	0.00	0.00	9
718	0.00	0.00	0.00	13
719	0.00	0.00	0.00	7
720	0.00	0.00	0.00	5
721	0.00	0.00	0.00	10
722	0.00	0.00	0.00	3
723	0.00	0.00	0.00	6
724	0.00	0.00	0.00	8
725	0.00	0.00	0.00	9
726	0.00	0.00	0.00	9
727	0.00	0.00	0.00	7
728	0.00	0.00	0.00	3
729	0.00	0.00	0.00	12
730	0.00	0.00	0.00	6
731	0.00	0.00	0.00	8
732	0.00	0.00	0.00	13
733	0.00	0.00	0.00	7
734	0.00	0.00	0.00	7
735	0.00	0.00	0.00	10
736	0.00	0.00	0.00	5
737	0.00	0.00	0.00	8
738	0.00	0.00	0.00	6
739	0.00	0.00	0.00	8
740	0.00	0.00	0.00	5
741	0.00	0.00	0.00	6
742	0.00	0.00	0.00	6
743	0.00	0.00	0.00	12
744	0.17	0.07	0.10	15
745	0.00	0.00	0.00	8
746	0.00	0.00	0.00	7
747	0.00	0.00	0.00	9
748	0.00	0.00	0.00	9
749	0.00	0.00	0.00	11
750	0.00	0.00	0.00	7
751	0.00	0.00	0.00	6
752	0.00	0.00	0.00	4
753	0.00	0.00	0.00	7
754	0.00	0.00	0.00	6
755	0.00	0.00	0.00	7
756	0.00	0.00	0.00	9
757	0.00	0.00	0.00	5
758	0.00	0.00	0.00	9
759	0.00	0.00	0.00	4
760	0.00	0.00	0.00	2

761	0.00	0.00	0.00	4
762	0.00	0.00	0.00	7
763	0.00	0.00	0.00	8
764	0.00	0.00	0.00	3
765	0.00	0.00	0.00	7
766	0.00	0.00	0.00	2
767	0.00	0.00	0.00	7
768	0.00	0.00	0.00	4
769	0.00	0.00	0.00	5
770	0.00	0.00	0.00	5
771	0.00	0.00	0.00	7
772	0.00	0.00	0.00	4
773	0.00	0.00	0.00	7
774	0.00	0.00	0.00	9
775	0.00	0.00	0.00	6
776	0.00	0.00	0.00	5
777	0.00	0.00	0.00	8
778	0.00	0.00	0.00	7
779	0.00	0.00	0.00	5
780	0.00	0.00	0.00	2
781	0.00	0.00	0.00	2
782	0.00	0.00	0.00	1
783	0.00	0.00	0.00	4
784	0.00	0.00	0.00	4
785	0.00	0.00	0.00	5
786	0.00	0.00	0.00	7
787	0.00	0.00	0.00	5
788	0.00	0.00	0.00	4
789	0.00	0.00	0.00	3
790	0.00	0.00	0.00	7
791	0.00	0.00	0.00	1
792	0.00	0.00	0.00	5
793	0.14	0.17	0.15	6
794	0.00	0.00	0.00	3
795	0.00	0.00	0.00	4
796	0.25	0.14	0.18	7
797	0.00	0.00	0.00	5
798	0.00	0.00	0.00	4
799	0.00	0.00	0.00	3
800	0.00	0.00	0.00	2
801	0.00	0.00	0.00	1
802	0.00	0.00	0.00	4
803	0.00	0.00	0.00	5
804	0.00	0.00	0.00	3
805	0.00	0.00	0.00	3
806	0.00	0.00	0.00	1
807	0.00	0.00	0.00	8
808	0.00	0.00	0.00	2
809	0.00	0.00	0.00	5
810	0.00	0.00	0.00	3
811	0.00	0.00	0.00	3
812	0.00	0.00	0.00	5
814	0.00	0.00	0.00	4
815	0.00	0.00	0.00	7
816	0.00	0.00	0.00	2
817	0.00	0.00	0.00	2
818	0.00	0.00	0.00	4
819	0.00	0.00	0.00	2
820	0.00	0.00	0.00	2
821	0.00	0.00	0.00	3

822	0.00	0.00	0.00	0
823	0.00	0.00	0.00	3
824	0.00	0.00	0.00	1
825	0.00	0.00	0.00	3
826	0.00	0.00	0.00	1
827	0.00	0.00	0.00	1
828	0.00	0.00	0.00	1
829	0.00	0.00	0.00	2
831	0.00	0.00	0.00	4
832	0.00	0.00	0.00	4
833	0.00	0.00	0.00	5
834	0.00	0.00	0.00	2
835	0.00	0.00	0.00	1
836	0.00	0.00	0.00	4
837	0.00	0.00	0.00	0
838	0.00	0.00	0.00	5
839	0.00	0.00	0.00	1
840	0.00	0.00	0.00	2
841	0.00	0.00	0.00	5
842	0.00	0.00	0.00	1
843	0.00	0.00	0.00	1
844	0.00	0.00	0.00	2
845	0.00	0.00	0.00	1
846	0.00	0.00	0.00	1
847	0.00	0.00	0.00	0
848	0.00	0.00	0.00	0
849	0.00	0.00	0.00	5
850	0.01	0.03	0.02	63
accuracy			0.00	3000
macro avg	0.00	0.00	0.00	3000
weighted avg	0.00	0.00	0.00	3000

```
C:\Users\PoojaAarti\anaconda3\Lib\site-packages\sklearn\metrics\_classification.py:14
69: UndefinedMetricWarning: Precision and F-score are 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, msg_start, len(result))
C:\Users\PoojaAarti\anaconda3\Lib\site-packages\sklearn\metrics\_classification.py:14
69: UndefinedMetricWarning: Recall and F-score are ill-defined and being set to 0.0 i
n labels with no true samples. Use `zero_division` parameter to control this behavio
r.

    _warn_prf(average, modifier, msg_start, len(result))
C:\Users\PoojaAarti\anaconda3\Lib\site-packages\sklearn\metrics\_classification.py:14
69: UndefinedMetricWarning: Precision and F-score are 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, msg_start, len(result))
C:\Users\PoojaAarti\anaconda3\Lib\site-packages\sklearn\metrics\_classification.py:14
69: UndefinedMetricWarning: Recall and F-score are ill-defined and being set to 0.0 i
n labels with no true samples. Use `zero_division` parameter to control this behavio
r.

    _warn_prf(average, modifier, msg_start, len(result))
C:\Users\PoojaAarti\anaconda3\Lib\site-packages\sklearn\metrics\_classification.py:14
69: UndefinedMetricWarning: Precision and F-score are 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, msg_start, len(result))
C:\Users\PoojaAarti\anaconda3\Lib\site-packages\sklearn\metrics\_classification.py:14
69: UndefinedMetricWarning: Recall and F-score are ill-defined and being set to 0.0 i
n labels with no true samples. Use `zero_division` parameter to control this behavio
r.

    _warn_prf(average, modifier, msg_start, len(result))
```

In [30]:

```
log_reg = LogisticRegression()
log_reg.fit(X_train, y_train)
```

```
C:\Users\PoojaAarti\anaconda3\Lib\site-packages\sklearn\linear_model\_logistic.py:46
0: 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>

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()
```

Out[30]:

```
▼ LogisticRegression
  LogisticRegression()
```

In [34]:

```
# Logistic Regression
y_pred_log_reg = log_reg.predict(X_test)
print(f"Accuracy: {accuracy_score(y_test, y_pred_log_reg)}")
print(classification_report(y_test, y_pred_log_reg))
```

Accuracy: 0.02133333333333333

precision recall f1-score support

350	0.00	0.00	0.00	2
351	0.00	0.00	0.00	1
358	0.00	0.00	0.00	1
359	0.00	0.00	0.00	1
363	0.00	0.00	0.00	1
373	0.00	0.00	0.00	1
399	0.00	0.00	0.00	1
401	0.00	0.00	0.00	1
404	0.00	0.00	0.00	1
405	0.00	0.00	0.00	1
410	0.00	0.00	0.00	1
411	0.00	0.00	0.00	2
412	0.00	0.00	0.00	1
413	0.00	0.00	0.00	2
414	0.00	0.00	0.00	1
415	0.00	0.00	0.00	1
416	0.00	0.00	0.00	2
418	0.00	0.00	0.00	1
420	0.00	0.00	0.00	1
421	0.00	0.00	0.00	2
422	0.00	0.00	0.00	3
423	0.00	0.00	0.00	1
425	0.00	0.00	0.00	1
426	0.00	0.00	0.00	2
427	0.00	0.00	0.00	3
429	0.00	0.00	0.00	2
430	0.00	0.00	0.00	2
431	0.00	0.00	0.00	1
433	0.00	0.00	0.00	1
434	0.00	0.00	0.00	1
437	0.00	0.00	0.00	1
439	0.00	0.00	0.00	1
440	0.00	0.00	0.00	1
443	0.00	0.00	0.00	1
444	0.00	0.00	0.00	1
445	0.00	0.00	0.00	7
447	0.00	0.00	0.00	2
448	0.00	0.00	0.00	2
449	0.00	0.00	0.00	3
450	0.00	0.00	0.00	2
451	0.00	0.00	0.00	1
452	0.00	0.00	0.00	3
453	0.00	0.00	0.00	2
454	0.00	0.00	0.00	1
455	0.00	0.00	0.00	1
456	0.00	0.00	0.00	2
457	0.00	0.00	0.00	1
458	0.00	0.00	0.00	2
459	0.00	0.00	0.00	1
460	0.00	0.00	0.00	2
461	0.00	0.00	0.00	4
462	0.00	0.00	0.00	1
464	0.00	0.00	0.00	2
465	0.00	0.00	0.00	4
466	0.00	0.00	0.00	1
467	0.00	0.00	0.00	2
468	0.00	0.00	0.00	2

469	0.00	0.00	0.00	3
470	0.00	0.00	0.00	3
471	0.00	0.00	0.00	1
472	0.00	0.00	0.00	2
473	0.00	0.00	0.00	3
474	0.00	0.00	0.00	6
475	0.00	0.00	0.00	1
476	0.00	0.00	0.00	4
477	0.00	0.00	0.00	5
478	0.00	0.00	0.00	3
479	0.00	0.00	0.00	7
480	0.00	0.00	0.00	6
481	0.00	0.00	0.00	7
482	0.00	0.00	0.00	3
483	0.00	0.00	0.00	1
484	0.00	0.00	0.00	2
485	0.00	0.00	0.00	4
486	0.00	0.00	0.00	5
487	0.00	0.00	0.00	4
488	0.00	0.00	0.00	2
489	0.00	0.00	0.00	4
490	0.00	0.00	0.00	4
491	0.00	0.00	0.00	2
492	0.00	0.00	0.00	4
493	0.00	0.00	0.00	4
494	0.00	0.00	0.00	6
495	0.00	0.00	0.00	3
496	0.00	0.00	0.00	5
497	0.00	0.00	0.00	4
498	0.00	0.00	0.00	6
499	0.00	0.00	0.00	2
500	0.00	0.00	0.00	4
501	0.00	0.00	0.00	6
502	0.00	0.00	0.00	5
503	0.00	0.00	0.00	4
504	0.00	0.00	0.00	6
505	0.00	0.00	0.00	4
506	0.00	0.00	0.00	5
507	0.00	0.00	0.00	6
508	0.00	0.00	0.00	3
509	0.00	0.00	0.00	5
510	0.00	0.00	0.00	5
511	0.00	0.00	0.00	4
512	0.00	0.00	0.00	5
513	0.00	0.00	0.00	5
514	0.00	0.00	0.00	6
515	0.00	0.00	0.00	7
516	0.00	0.00	0.00	8
517	0.00	0.00	0.00	5
518	0.00	0.00	0.00	4
519	0.00	0.00	0.00	3
520	0.00	0.00	0.00	9
521	0.00	0.00	0.00	11
522	0.00	0.00	0.00	6
523	0.00	0.00	0.00	6
524	0.00	0.00	0.00	6
525	0.00	0.00	0.00	6
526	0.00	0.00	0.00	8
527	0.00	0.00	0.00	6
528	0.00	0.00	0.00	13

529	0.00	0.00	0.00	10
530	0.00	0.00	0.00	5
531	0.00	0.00	0.00	8
532	0.00	0.00	0.00	2
533	0.00	0.00	0.00	3
534	0.00	0.00	0.00	3
535	0.00	0.00	0.00	11
536	0.00	0.00	0.00	4
537	0.00	0.00	0.00	8
538	0.00	0.00	0.00	6
539	0.00	0.00	0.00	8
540	0.00	0.00	0.00	8
541	0.00	0.00	0.00	4
542	0.00	0.00	0.00	5
543	0.00	0.00	0.00	7
544	0.00	0.00	0.00	4
545	0.00	0.00	0.00	8
546	0.00	0.00	0.00	7
547	0.00	0.00	0.00	8
548	0.00	0.00	0.00	3
549	0.00	0.00	0.00	9
550	0.00	0.00	0.00	13
551	0.00	0.00	0.00	5
552	0.00	0.00	0.00	3
553	0.00	0.00	0.00	3
554	0.00	0.00	0.00	14
555	0.00	0.00	0.00	8
556	0.00	0.00	0.00	7
557	0.00	0.00	0.00	7
558	0.00	0.00	0.00	11
559	0.00	0.00	0.00	9
560	0.00	0.00	0.00	8
561	0.00	0.00	0.00	5
562	0.00	0.00	0.00	6
563	0.00	0.00	0.00	3
564	0.00	0.00	0.00	8
565	0.00	0.00	0.00	9
566	0.00	0.00	0.00	8
567	0.00	0.00	0.00	8
568	0.00	0.00	0.00	7
569	0.00	0.00	0.00	9
570	0.00	0.00	0.00	4
571	0.00	0.00	0.00	7
572	0.00	0.00	0.00	7
573	0.00	0.00	0.00	9
574	0.00	0.00	0.00	9
575	0.00	0.00	0.00	8
576	0.00	0.00	0.00	7
577	0.00	0.00	0.00	11
578	0.00	0.00	0.00	4
579	0.00	0.00	0.00	15
580	0.00	0.00	0.00	11
581	0.00	0.00	0.00	13
582	0.00	0.00	0.00	9
583	0.00	0.00	0.00	9
584	0.00	0.00	0.00	14
585	0.00	0.00	0.00	12
586	0.00	0.00	0.00	9
587	0.00	0.00	0.00	10
588	0.00	0.00	0.00	12

589	0.00	0.00	0.00	11
590	0.00	0.00	0.00	14
591	0.00	0.00	0.00	12
592	0.00	0.00	0.00	9
593	0.00	0.00	0.00	11
594	0.00	0.00	0.00	11
595	0.00	0.00	0.00	9
596	0.00	0.00	0.00	9
597	0.00	0.00	0.00	13
598	0.00	0.00	0.00	6
599	0.00	0.00	0.00	6
600	0.00	0.00	0.00	9
601	0.00	0.00	0.00	14
602	0.00	0.00	0.00	10
603	0.00	0.00	0.00	10
604	0.00	0.00	0.00	10
605	0.00	0.00	0.00	14
606	0.00	0.00	0.00	13
607	0.00	0.00	0.00	17
608	0.00	0.00	0.00	9
609	0.00	0.00	0.00	12
610	0.00	0.00	0.00	9
611	0.00	0.00	0.00	12
612	0.00	0.00	0.00	9
613	0.00	0.00	0.00	14
614	0.00	0.00	0.00	8
615	0.00	0.00	0.00	12
616	0.00	0.00	0.00	8
617	0.00	0.00	0.00	11
618	0.00	0.00	0.00	9
619	0.00	0.00	0.00	12
620	0.00	0.00	0.00	8
621	0.00	0.00	0.00	7
622	0.00	0.00	0.00	8
623	0.00	0.00	0.00	14
624	0.00	0.00	0.00	7
625	0.00	0.00	0.00	9
626	0.00	0.00	0.00	13
627	0.00	0.00	0.00	10
628	0.00	0.00	0.00	12
629	0.00	0.00	0.00	17
630	0.00	0.00	0.00	10
631	0.00	0.00	0.00	7
632	0.00	0.00	0.00	12
633	0.00	0.00	0.00	11
634	0.00	0.00	0.00	14
635	0.00	0.00	0.00	11
636	0.00	0.00	0.00	17
637	0.00	0.00	0.00	14
638	0.00	0.00	0.00	9
639	0.00	0.00	0.00	14
640	0.00	0.00	0.00	10
641	0.00	0.00	0.00	7
642	0.00	0.00	0.00	9
643	0.00	0.00	0.00	9
644	0.00	0.00	0.00	12
645	0.00	0.00	0.00	11
646	0.00	0.00	0.00	17
647	0.00	0.00	0.00	5
648	0.00	0.00	0.00	11

649	0.00	0.00	0.00	13
650	0.00	0.00	0.00	9
651	0.00	0.00	0.00	13
652	0.00	0.00	0.00	14
653	0.00	0.00	0.00	5
654	0.00	0.00	0.00	8
655	0.00	0.00	0.00	11
656	0.00	0.00	0.00	17
657	0.00	0.00	0.00	13
658	0.00	0.00	0.00	6
659	0.00	0.00	0.00	10
660	0.00	0.00	0.00	10
661	0.00	0.00	0.00	15
662	0.00	0.00	0.00	10
663	0.00	0.00	0.00	15
664	0.00	0.00	0.00	7
665	0.00	0.00	0.00	7
666	0.00	0.00	0.00	13
667	0.00	0.00	0.00	15
668	0.00	0.00	0.00	11
669	0.00	0.00	0.00	12
670	0.00	0.00	0.00	19
671	0.00	0.00	0.00	13
672	0.00	0.00	0.00	6
673	0.00	0.00	0.00	11
674	0.00	0.00	0.00	6
675	0.00	0.00	0.00	13
676	0.00	0.00	0.00	12
677	0.00	0.00	0.00	13
678	0.00	0.00	0.00	17
679	0.00	0.00	0.00	17
680	0.00	0.00	0.00	15
681	0.00	0.00	0.00	14
682	0.00	0.00	0.00	17
683	0.00	0.00	0.00	17
684	0.00	0.00	0.00	11
685	0.00	0.00	0.00	10
686	0.00	0.00	0.00	17
687	0.00	0.00	0.00	15
688	0.00	0.00	0.00	14
689	0.00	0.00	0.00	8
690	0.00	0.00	0.00	14
691	0.00	0.00	0.00	12
692	0.00	0.00	0.00	11
693	0.00	0.00	0.00	9
694	0.00	0.00	0.00	12
695	0.00	0.00	0.00	8
696	0.00	0.00	0.00	9
697	0.00	0.00	0.00	7
698	0.00	0.00	0.00	15
699	0.00	0.00	0.00	9
700	0.00	0.00	0.00	11
701	0.00	0.00	0.00	10
702	0.00	0.00	0.00	11
703	0.00	0.00	0.00	11
704	0.00	0.00	0.00	13
705	0.00	0.00	0.00	12
706	0.00	0.00	0.00	10
707	0.00	0.00	0.00	11
708	0.00	0.00	0.00	8

709	0.00	0.00	0.00	17
710	0.00	0.00	0.00	16
711	0.00	0.00	0.00	12
712	0.00	0.00	0.00	10
713	0.00	0.00	0.00	7
714	0.00	0.00	0.00	11
715	0.00	0.00	0.00	9
716	0.00	0.00	0.00	9
717	0.00	0.00	0.00	9
718	0.00	0.00	0.00	13
719	0.00	0.00	0.00	7
720	0.00	0.00	0.00	5
721	0.00	0.00	0.00	10
722	0.00	0.00	0.00	3
723	0.00	0.00	0.00	6
724	0.00	0.00	0.00	8
725	0.00	0.00	0.00	9
726	0.00	0.00	0.00	9
727	0.00	0.00	0.00	7
728	0.00	0.00	0.00	3
729	0.00	0.00	0.00	12
730	0.00	0.00	0.00	6
731	0.02	0.12	0.03	8
732	0.00	0.00	0.00	13
733	0.00	0.00	0.00	7
734	0.00	0.00	0.00	7
735	0.00	0.00	0.00	10
736	0.00	0.00	0.00	5
737	0.00	0.00	0.00	8
738	0.00	0.00	0.00	6
739	0.00	0.00	0.00	8
740	0.00	0.00	0.00	5
741	0.00	0.00	0.00	6
742	0.00	0.00	0.00	6
743	0.00	0.00	0.00	12
744	0.00	0.00	0.00	15
745	0.00	0.00	0.00	8
746	0.00	0.00	0.00	7
747	0.00	0.00	0.00	9
748	0.00	0.00	0.00	9
749	0.00	0.00	0.00	11
750	0.00	0.00	0.00	7
751	0.00	0.00	0.00	6
752	0.00	0.00	0.00	4
753	0.00	0.00	0.00	7
754	0.00	0.00	0.00	6
755	0.00	0.00	0.00	7
756	0.00	0.00	0.00	9
757	0.00	0.00	0.00	5
758	0.00	0.00	0.00	9
759	0.00	0.00	0.00	4
760	0.00	0.00	0.00	2
761	0.00	0.00	0.00	4
762	0.00	0.00	0.00	7
763	0.00	0.00	0.00	8
764	0.00	0.00	0.00	3
765	0.00	0.00	0.00	7
766	0.00	0.00	0.00	2
767	0.00	0.00	0.00	7
768	0.00	0.00	0.00	4

769	0.00	0.00	0.00	5
770	0.00	0.00	0.00	5
771	0.00	0.00	0.00	7
772	0.00	0.00	0.00	4
773	0.00	0.00	0.00	7
774	0.00	0.00	0.00	9
775	0.00	0.00	0.00	6
776	0.00	0.00	0.00	5
777	0.00	0.00	0.00	8
778	0.00	0.00	0.00	7
779	0.00	0.00	0.00	5
780	0.00	0.00	0.00	2
781	0.00	0.00	0.00	2
782	0.00	0.00	0.00	1
783	0.00	0.00	0.00	4
784	0.00	0.00	0.00	4
785	0.00	0.00	0.00	5
786	0.00	0.00	0.00	7
787	0.00	0.00	0.00	5
788	0.00	0.00	0.00	4
789	0.00	0.00	0.00	3
790	0.00	0.00	0.00	7
791	0.00	0.00	0.00	1
792	0.00	0.00	0.00	5
793	0.00	0.00	0.00	6
794	0.00	0.00	0.00	3
795	0.00	0.00	0.00	4
796	0.00	0.00	0.00	7
797	0.00	0.00	0.00	5
798	0.00	0.00	0.00	4
799	0.00	0.00	0.00	3
800	0.00	0.00	0.00	2
801	0.00	0.00	0.00	1
802	0.00	0.00	0.00	4
803	0.00	0.00	0.00	5
804	0.00	0.00	0.00	3
805	0.00	0.00	0.00	3
806	0.00	0.00	0.00	1
807	0.00	0.00	0.00	8
808	0.00	0.00	0.00	2
809	0.00	0.00	0.00	5
810	0.00	0.00	0.00	3
811	0.00	0.00	0.00	3
812	0.00	0.00	0.00	5
814	0.00	0.00	0.00	4
815	0.00	0.00	0.00	7
816	0.00	0.00	0.00	2
817	0.00	0.00	0.00	2
818	0.00	0.00	0.00	4
819	0.00	0.00	0.00	2
820	0.00	0.00	0.00	2
821	0.00	0.00	0.00	3
823	0.00	0.00	0.00	3
824	0.00	0.00	0.00	1
825	0.00	0.00	0.00	3
826	0.00	0.00	0.00	1
827	0.00	0.00	0.00	1
828	0.00	0.00	0.00	1
829	0.00	0.00	0.00	2
831	0.00	0.00	0.00	4

832	0.00	0.00	0.00	4
833	0.00	0.00	0.00	5
834	0.00	0.00	0.00	2
835	0.00	0.00	0.00	1
836	0.00	0.00	0.00	4
838	0.00	0.00	0.00	5
839	0.00	0.00	0.00	1
840	0.00	0.00	0.00	2
841	0.00	0.00	0.00	5
842	0.00	0.00	0.00	1
843	0.00	0.00	0.00	1
844	0.00	0.00	0.00	2
845	0.00	0.00	0.00	1
846	0.00	0.00	0.00	1
849	0.00	0.00	0.00	5
850	0.02	1.00	0.04	63
accuracy			0.02	3000
macro avg	0.00	0.00	0.00	3000
weighted avg	0.00	0.02	0.00	3000

```
C:\Users\PoojaAarti\anaconda3\Lib\site-packages\sklearn\metrics\_classification.py:14
69: UndefinedMetricWarning: Precision and F-score are 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, msg_start, len(result))
```

```
C:\Users\PoojaAarti\anaconda3\Lib\site-packages\sklearn\metrics\_classification.py:14
69: UndefinedMetricWarning: Precision and F-score are 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, msg_start, len(result))
```

```
C:\Users\PoojaAarti\anaconda3\Lib\site-packages\sklearn\metrics\_classification.py:14
69: UndefinedMetricWarning: Precision and F-score are 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, msg_start, len(result))
```

In [38]: classifier = AdaBoostClassifier()  
classifier.fit(X\_train, y\_train)

Out[38]: ▾ AdaBoostClassifier

AdaBoostClassifier()