```
from google.colab import drive
drive.mount('/content/drive')
```

```
Drive already mounted at /content/drive; to attempt to forcibly remount, call drive.mount("/content/drive", force_remount=
```

```
import pandas as pd
import numpy as np
from nltk.corpus import stopwords
from nltk.stem.porter import PorterStemmer


import nltk
nltk.download('stopwords')
print(stopwords.words('english'))
```

```
['i', 'me', 'my', 'myself', 'we', 'our', 'ours', 'ourselves', 'you', "you're", "you've", "you'll", "you'd", 'your', 'yours
[nltk_data] Downloading package stopwords to /root/nltk_data...
[nltk_data]   Package stopwords is already up-to-date!
```

```
data=pd.read_csv(r"/content/drive/MyDrive/Colab Notebooks/train.csv")
data.head(10)
```

	category	sub_category	crimeaditionalinfo
0	Online and Social Media Related Crime	Cyber Bullying Stalking Sexting	I had continue received random calls and abusi...
1	Online Financial Fraud	Fraud CallVishing	The above fraudster is continuously messaging ...
2	Online Gambling Betting	Online Gambling Betting	He is acting like a police and demanding for m...
3	Online and Social Media Related Crime	Online Job Fraud	In apna Job I have applied for job interview f...
4	Online Financial Fraud	Fraud CallVishing	I received a call from lady stating that she w...
5	Online Financial Fraud	UPI Related Frauds	FRAUD \t UPI PAYTM \r\nBANK \tPunjab National ...
6	Online Financial Fraud	Fraud CallVishing	Sir I am Prabhat Singh jat An app on playstor...
7	Online Financial Fraud	Internet Banking Related Fraud	FINANCIAL FRAUD RS
8	RapeGang Rape RGRSexually Abusive Content	NaN	I got the message on Whatsapp to my number The...
9	Any Other Cyber Crime	Other	Details entered in pdf file Person posing as A...

```
test_data=pd.read_csv(r"/content/drive/MyDrive/Colab Notebooks/test.csv")
test_data
```

	category	sub_category	crimeaditionalinfo
0	RapeGang Rape RGRSexually Abusive Content	NaN	Sir namaskar mein Ranjit Kumar PatraPaise neh...
1	Online Financial Fraud	DebitCredit Card FraudSim Swap Fraud	KOTAK MAHINDRA BANK FRAUD\r\nFRAUD AMOUNT
2	Cyber Attack/ Dependent Crimes	SQL Injection	The issue actually started when I got this ema...
3	Online Financial Fraud	Fraud CallVishing	I am amit kumar from karwi chitrakoot I am tot...
4	Any Other Cyber Crime	Other	I have ordered saree and blouse from rinki s...
...	...	...	...
31224	Online and Social Media Related Crime	Online Matrimonial Fraud	A lady named Rashmi probably a fake name had c...
31225	Online Financial Fraud	Internet Banking Related Fraud	I am Mr Chokhe Ram Two pers mobile number wer...
31226	Any Other Cyber Crime	Other	Mai Bibekbraj maine pahle ki complain kar chuk...
31227	Online Financial Fraud	Internet Banking Related Fraud	received URL link for updating KYC from mobile...
31228	Any Other Cyber Crime	Other	I saw add on facebook for job placement and I ...

```
data[data["category"]=="Online and Social Media Related Crime"].sub_category.value_counts()
```

sub_category	count
Cyber Bullying Stalking Sexting	4089
FakeImpersonating Profile	2299
Profile Hacking Identity Theft	2073
Cheating by Impersonation	1988
Online Job Fraud	912
Provocative Speech for unlawful acts	417
EMail Phishing	157
Online Matrimonial Fraud	132
Impersonating Email	44
Intimidating Email	29

dtype: int64

```
data=pd.concat([data,test_data],axis=0)
data
```

	category	sub_category	crimeaditionalinfo
0	Online and Social Media Related Crime	Cyber Bullying Stalking Sexting	I had continue received random calls and abusi...
1	Online Financial Fraud	Fraud CallVishing	The above fraudster is continuously messaging ...
2	Online Gambling Betting	Online Gambling Betting	He is acting like a police and demanding for m...
3	Online and Social Media Related Crime	Online Job Fraud	In apna Job I have applied for job interview f...
4	Online Financial Fraud	Fraud CallVishing	I received a call from lady stating that she w...
...	...	...	...
31224	Online and Social Media Related Crime	Online Matrimonial Fraud	A lady named Rashmi probably a fake name had c...
31225	Online Financial Fraud	Internet Banking Related Fraud	I am Mr Chokhe Ram Two pers mobile number wer...
31226	Any Other Cyber Crime	Other	Mai Bibekbraj maine pahle ki complain kar chuk...
31227	Online Financial Fraud	Internet Banking Related Fraud	received URL link for updating KYC from mobile...
31228	Any Other Cyber Crime	Other	I saw add on facebook for job placement and I ...

124915 rows × 3 columns

```
data.isnull().sum()
```

	0
category	0
sub_category	8827
crimeaditionalinfo	28

```
data.category.value_counts()
```

category	count
Online Financial Fraud	76330
Online and Social Media Related Crime	16279
Any Other Cyber Crime	14548
Cyber Attack/ Dependent Crimes	4869
RapeGang Rape RGRSexually Abusive Content	3734
Sexually Obscene material	2504
Hacking Damage to computercomputer system etc	2302
Sexually Explicit Act	2087
Cryptocurrency Crime	646
Online Gambling Betting	578
Child Pornography CPChild Sexual Abuse Material CSAM	502
Online Cyber Trafficking	244
Cyber Terrorism	213
Ransomware	74
Crime Against Women & Children	4
Report Unlawful Content	1

```
data[data["category"] == "Online Financial Fraud"].sub_category.value_counts()
```

sub_category	count
UPI Related Frauds	35746
DebitCredit Card FraudSim Swap Fraud	14361
Internet Banking Related Fraud	11845
Fraud CallVishing	7630
EWallet Related Fraud	5385
DematDepository Fraud	983
Business Email CompromiseEmail Takeover	380

```
data.shape
```

```
(124915, 3)
```

```
data=data.dropna()
data.shape
```

```
(116061, 3)
```

```
Start coding or generate with AI.
```

```
import re
port_stem = PorterStemmer()
```

```
def clean_data(combine):
    stemmed_content = re.sub('[^a-zA-Z)]',' ', combine)
    stemmed_content = stemmed_content.lower()
    stemmed_content = stemmed_content.split()
```

```
    stemmed_content = [port_stem.stem(word) for word in stemmed_content if not word in stopwords.words('english')]
    stemmed_content = ' '.join(stemmed_content)
    return stemmed_content
```

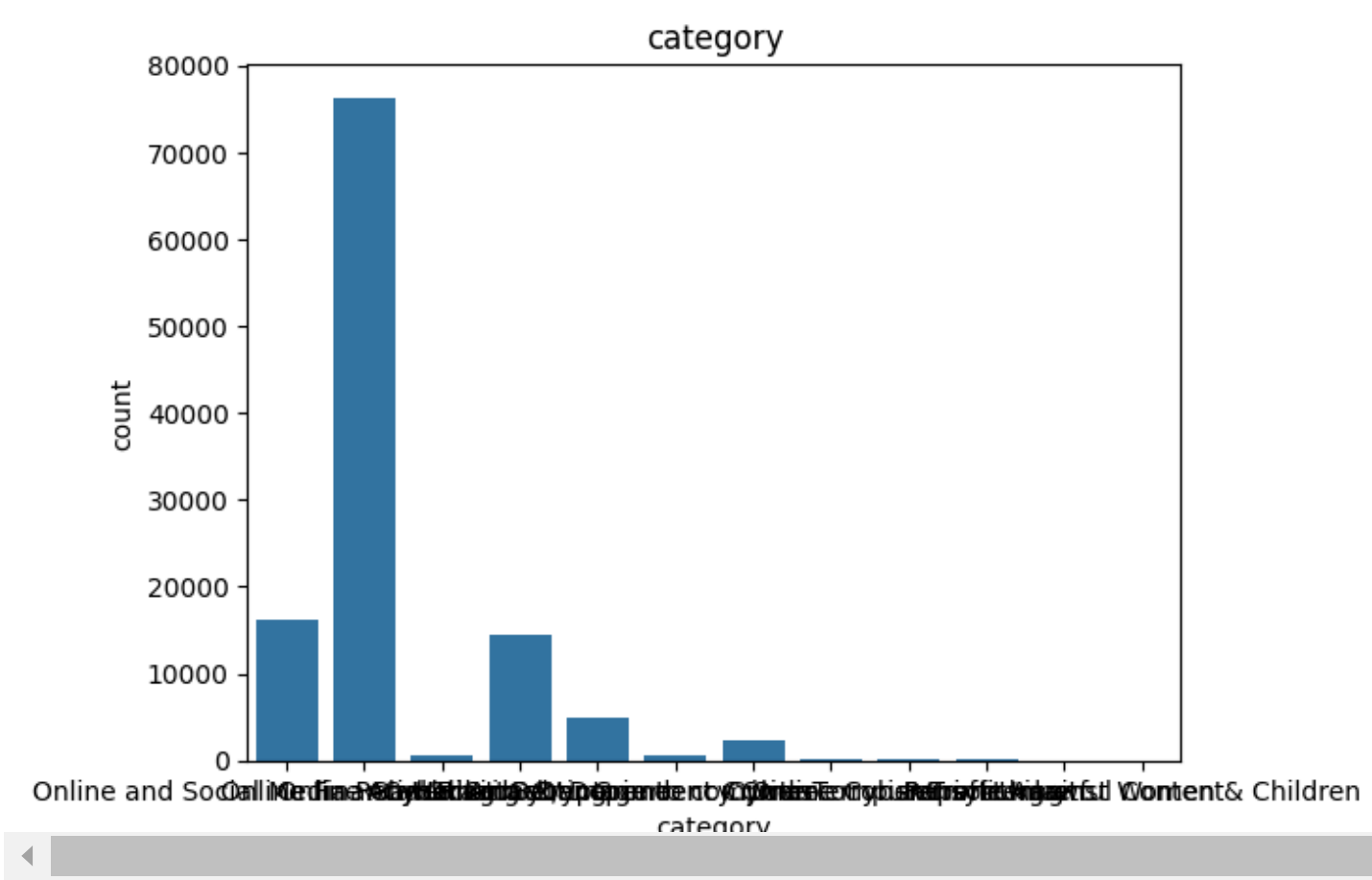
```
data['crimeaditionalinfo'] = data['crimeaditionalinfo'].apply(clean_data)
data
```

```
<ipython-input-92-4aa99fe3de90>:11: 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-v
  data['crimeaditionalinfo'] = data['crimeaditionalinfo'].apply(clean_data)
```

	category	sub_category	crimeaditionalinfo
0	Online and Social Media Related Crime	Cyber Bullying Stalking Sexting	continu receiv random call abus messag whatsap...
1	Online Financial Fraud	Fraud CallVishing	fraudster continu messag ask pay money send fa...
2	Online Gambling Betting	Online Gambling Betting	act like polic demand money ad section text me...
3	Online and Social Media Related Crime	Online Job Fraud	apna job appli job interview telecal resourc m...
4	Online Financial Fraud	Fraud CallVishing	receiv call ladi state send new phone vivo rec...
...	...	...	...
31224	Online and Social Media Related Crime	Online Matrimonial Fraud	ladi name rashmi probabl fake name call day ag...
31225	Online Financial Fraud	Internet Banking Related Fraud	mr chokh ram two per mobil number found gool i...
31226	Any Other Cyber Crime	Other	mai bibekbraj main pahl ki complain kar chuka ...
31227	Online Financial Fraud	Internet Banking Related Fraud	receiv url link updat kyc mobil open receiv ot...
31228	Any Other Cyber Crime	Other	saw add facebook job placement want job contac...

116061 rows × 3 columns

```
clean_data_from_data=data
import seaborn as sns
import matplotlib.pyplot as plt
sns.countplot(clean_data_from_data, x="category")
plt.title("category")
plt.show()
```




```
from sklearn.preprocessing import LabelEncoder
le=LabelEncoder()
def Encoder(data,i):
    data[i]=le.fit_transform(data[i])
    return data
```

```
data=Encoder(data,"category")
data
```

```
<ipython-input-94-36d756b2c2e2>: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-v
  data[i]=le.fit_transform(data[i])
```

	category	sub_category	crimeaditionalinfo
0	9	Cyber Bullying Stalking Sexting	continu receiv random call abus messag whatsap...
1	7	Fraud CallVishing	fraudster continu messag ask pay money send fa...
2	8	Online Gambling Betting	act like polic demand money ad section text me...
3	9	Online Job Fraud	apna job appli job interview telecal resourc m...
4	7	Fraud CallVishing	receiv call ladi state send new phone vivo rec...
...	...	...	...
31224	9	Online Matrimonial Fraud	ladi name rashmi probabl fake name call day ag...
31225	7	Internet Banking Related Fraud	mr chokh ram two per mobil number found gool i...
31226	0	Other	mai bibekbraj main pahl ki complain kar chuka ...
31227	7	Internet Banking Related Fraud	receiv url link updat kyc mobil open receiv ot...
31228	0	Other	saw add facebook job placement want job contac...

116061 rows × 3 columns

```
data.crimeaditionalinfo[0]
```

```
'continu receiv random call abus messag whatsapp someon ad number unknown facebook group name girl still get call unknown
```

```
data.category.unique()
values=list(le.inverse_transform(data["category"].unique()))
values
```

```
['Online and Social Media Related Crime',
 'Online Financial Fraud',
 'Online Gambling  Betting',
 'Any Other Cyber Crime',
 'Cyber Attack/ Dependent Crimes',
 'Cryptocurrency Crime',
 'Hacking  Damage to computercomputer system etc',
 'Cyber Terrorism',
 'Online Cyber Trafficking',
 'Ransomware',
 'Report Unlawful Content',
 'Crime Against Women & Children']
```

```
def value_assign(data,col):
    values=list(le.inverse_transform(data["category"].unique()))
    index=list(data[col].unique())
    d={}
    for i in range(0,len(index)):
        d[index[i]]=values[i]
    return d

d=value_assign(data,"category")
d
```

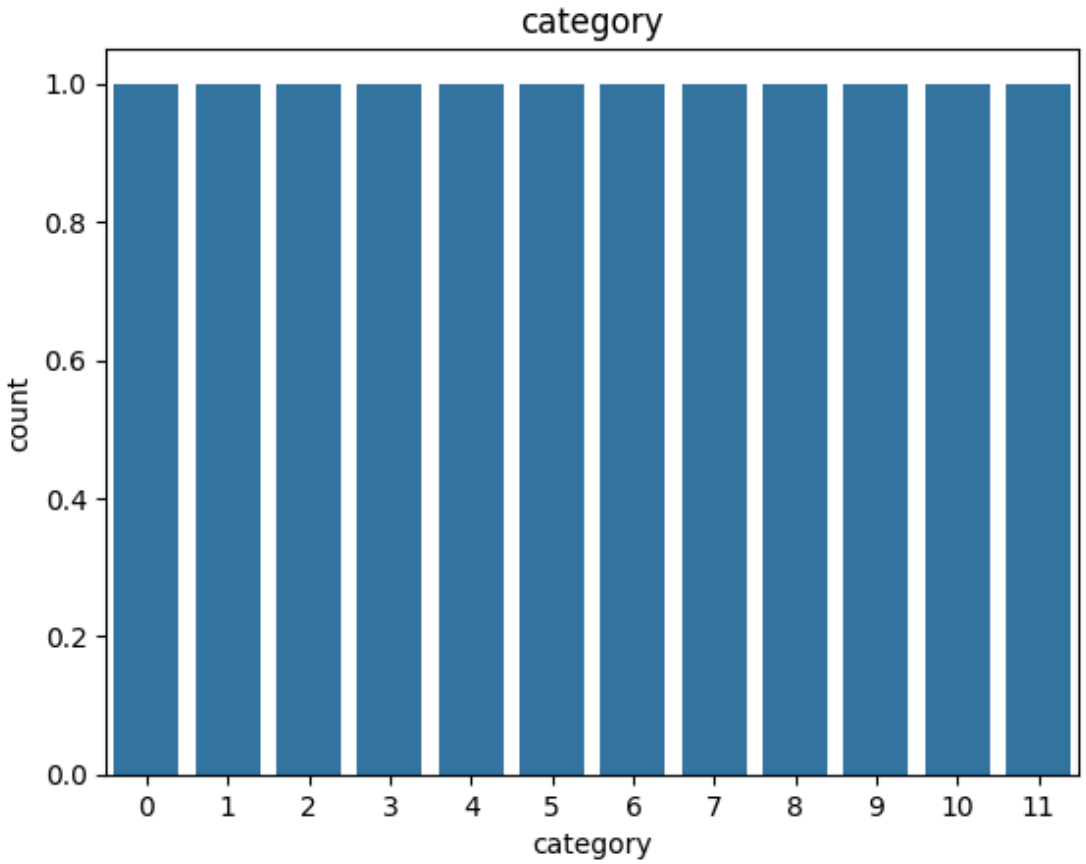
```
{9: 'Online and Social Media Related Crime',
 7: 'Online Financial Fraud',
 8: 'Online Gambling  Betting',
 0: 'Any Other Cyber Crime',
 3: 'Cyber Attack/ Dependent Crimes',
 2: 'Cryptocurrency Crime',
 5: 'Hacking  Damage to computercomputer system etc',
 4: 'Cyber Terrorism',
 6: 'Online Cyber Trafficking',
 10: 'Ransomware',
 11: 'Report Unlawful Content',
 1: 'Crime Against Women & Children'}
```

```
X=data["crimeaditionalinfo"]
Y=data["category"]
X.shape
```

```
(116061,)
```

```
from imblearn.over_sampling import RandomOverSampler
X_reshaped = pd.DataFrame(X).values.reshape(-1, 1)

oversampler = RandomOverSampler(sampling_strategy='auto', random_state=100)

X_resampled, y_resampled = oversampler.fit_resample(X_reshaped, Y)

X_resampled = pd.DataFrame(X_resampled, columns=['crimeaditionalinfo'])

print(X_resampled)
print(X_resampled.shape, y_resampled.shape)


sns.countplot(y_resampled.value_counts())
plt.title("category")
plt.show()

print(y_resampled.value_counts())
```

```
                                        crimeaditionalinfo
0       continu receiv random call abus messag whatsap...
1       fraudster continu messag ask pay money send fa...
2       act like polic demand money ad section text me...
3       apna job appli job interview telecal resourc m...
4       receiv call ladi state send new phone vivo rec...
...                                                   ...
915667  ladi attach video goe name swathi iyer social ...
915668  ladi attach video goe name swathi iyer social ...
915669  ladi attach video goe name swathi iyer social ...
915670  ladi attach video goe name swathi iyer social ...
915671  ladi attach video goe name swathi iyer social ...

[915672 rows x 1 columns]
(915672, 1) (915672,)
```




```
category
9     76306
```

```
7     76306
8     76306
0     76306
3     76306
2     76306
5     76306
4     76306
6     76306
10    76306
11    76306
1     76306
Name: count, dtype: int64
```

```
from sklearn.feature_extraction.text import TfidfVectorizer,CountVectorizer
```

```
"""tfv = TfidfVectorizer(max_features=100)
X=tfv.fit_transform(X).toarray()
X"""
```

```
tfv = TfidfVectorizer(max_features=100)
X=tfv.fit_transform(X_resampled['crimeaditionalinfo']).toarray()
X.shape
```

```
(915672, 100)
```

Start coding or generate with AI.

```
from sklearn.model_selection import train_test_split
```

```
x_train,x_test,y_train,y_test=train_test_split(X,y_resampled,test_size=0.4,random_state=1000)
```

```
Start coding or generate with AI.
```

```
x_train.shape,y_train.shape
```

```
((549403, 100), (549403,))
```

```
from sklearn.naive_bayes import GaussianNB,MultinomialNB
model=GaussianNB()
model.fit(x_train,y_train)
```

```
GaussianNB
GaussianNB()
```

```
model.predict(X[40000].reshape(1, -1))
```

```
array([10])
```

```
y_prediction=model.predict(x_test)
y_prediction
```

```
array([ 3,  9, 10, ...,  5, 10,  4])
```

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

```
print(confusion_matrix(y_test,y_prediction))
```

```
[[   26     0   446     0  3985   153  5543    18  1294    39 19015     0]
 [    0 30726     0     0     0     0     0     0     0     0     0     0]
 [    0     0  9999     0  1711   133  2426     0  6818    62  9551     0]
 [    0  7268     0 23383     0     0     0     0     0     0     0     0]
 [    0     0     0     0  8479     0  5220     0   116   142 16599     0]
 [   22     0    78     0  5146   557  6763     0   278    53 17290     0]
 [    0     0   120     0  3943   268  6105     0   518     0 19603     0]
 [    6     0   249     2  2222    73  8678   101   910    21 18179     0]
 [    0     0   989     0  3457   108  5049     0  3539    76 17386     0]
```

```
[    8     0   230     0  3120   182  4404     9   687   122 21624     0]
[    0     0     0     0   405     0   736     0     0     0 29354     0]
[    0     0     0     0     0     0     0     0     0     0     0 30447]]
```

```
print(classification_report(y_test,y_prediction))
```

```
              precision    recall  f1-score   support

           0       0.42      0.00      0.00     30519
           1       0.81      1.00      0.89     30726
           2       0.83      0.33      0.47     30700
           3       1.00      0.76      0.87     30651
           4       0.26      0.28      0.27     30556
           5       0.38      0.02      0.04     30187
           6       0.14      0.20      0.16     30557
           7       0.79      0.00      0.01     30441
           8       0.25      0.12      0.16     30604
           9       0.24      0.00      0.01     30386
          10       0.17      0.96      0.29     30495
          11       1.00      1.00      1.00     30447

    accuracy                           0.39    366269
   macro avg       0.52      0.39      0.35    366269
weighted avg       0.52      0.39      0.35    366269
```

```
mnb=MultinomialNB()
mnb.fit(x_train,y_train)
```

```
MultinomialNB
MultinomialNB()
```

```
y_prediction=mnb.predict(x_test)
y_prediction
```

```
array([3, 2, 1, ..., 5, 8, 5])
```

```
print(confusion_matrix(y_test,y_prediction))
```

```
[[ 5908    77  2259    45  3888  5349   917  5226  2822  2714  1265    49]
 [    0 15328     0 15398     0     0     0     0     0     0     0     0]
 [  823     0 21610    60   338  1413   252  1366  3380   967   491     0]
 [    0 11163     0 19488     0     0     0     0     0     0     0     0]
 [ 2154     0  1122   317  9657  5350  1299  3267  2785  3852   753     0]
 [ 1350    55   542    88  4353 16977   608   651   527  2450  2541    45]
 [ 1744     0  1960   152  3295  5929  1908  5361  2723  6237  1119   129]
 [ 3013    50  1219    20  1263  2735   781 17606  2336   910   489    19]
 [ 2149   104  5726    49  2738  4812   655  3294  8261  1861   955     0]
 [ 2339    33  1250    48  2875  5913   990  1366  1880 12089  1487   116]
 [  786     0   899     0   367  1620   407   844  2455  3210 19907     0]
 [    0     0     0     0     0     0     0     0     0     0     0 30447]]
```

```
print(classification_report(y_test,y_prediction))
```

```
              precision    recall  f1-score   support

           0       0.29      0.19      0.23     30519
           1       0.57      0.50      0.53     30726
           2       0.59      0.70      0.64     30700
           3       0.55      0.64      0.59     30651
           4       0.34      0.32      0.33     30556
           5       0.34      0.56      0.42     30187
           6       0.24      0.06      0.10     30557
           7       0.45      0.58      0.51     30441
           8       0.30      0.27      0.29     30604
           9       0.35      0.40      0.37     30386
          10       0.69      0.65      0.67     30495
          11       0.99      1.00      0.99     30447

    accuracy                           0.49    366269
   macro avg       0.48      0.49      0.47    366269
weighted avg       0.48      0.49      0.47    366269
```

```
from sklearn.metrics import f1_score, precision_score,recall_score
```

```
from sklearn.ensemble import RandomForestClassifier


RFC=RandomForestClassifier()
RFC.fit(x_train,y_train)
```

```
RandomForestClassifier
RandomForestClassifier()
```

```
RFC.score(x_test,y_test)
```

```
0.966418124383991
```

```
y_prediction=RFC.predict(x_test)
y_prediction
```

```
array([3, 2, 0, ..., 5, 0, 8])
```

```
precision_score(y_test,y_prediction, average=None)
```

```
array([0.91512031, 0.99212141, 0.99159006, 1.        , 0.99823574,
       0.98766774, 0.98942259, 0.94881464, 0.99781542, 0.80812122,
       0.99980329, 1.        ])
```

```
recall_score(y_test,y_prediction,average=None)
```

```
array([0.925915  , 1.        , 0.99087948, 0.99203941, 0.98141118,
       0.96306357, 0.96733973, 0.83486088, 0.97010195, 0.97064438,
       1.        , 1.        ])
```

```
f1_score(y_test,y_prediction,average=None)
```

```
array([0.92048601, 0.99604512, 0.99123464, 0.9960038 , 0.98975197,
       0.97521049, 0.97825655, 0.88819767, 0.98376354, 0.88195805,
       0.99990163, 1.        ])
```

```
print(confusion_matrix(y_test,y_prediction))
```

```
[[28258     0    54     0     0   102    57   884     5  1154     5     0]
 [    0 30726     0     0     0     0     0     0     0     0     0     0]
 [    0     0 30420     0     0     0    39     0     0   241     0     0]
 [    0   244     0 30407     0     0     0     0     0     0     0     0]
 [    0     0     0     0 29988     0     0     0     0   568     0     0]
 [    0     0    61     0    45 29072    43    30    15   921     0     0]
 [    0     0     0     0     0     0 29559     0     0   998     0     0]
 [ 2361     0    91     0     1   188    82 25414    42  2261     1     0]
 [    0     0     0     0     0     0    55     0 29689   860     0     0]
 [  260     0    52     0     7    73    40   457     3 29494     0     0]
 [    0     0     0     0     0     0     0     0     0     0 30495     0]
 [    0     0     0     0     0     0     0     0     0     0     0 30447]]
```

```
print(classification_report(y_test,y_prediction))
```

```
              precision    recall  f1-score   support

           0       0.92      0.93      0.92     30519
           1       0.99      1.00      1.00     30726
           2       0.99      0.99      0.99     30700
           3       1.00      0.99      1.00     30651
           4       1.00      0.98      0.99     30556
           5       0.99      0.96      0.98     30187
           6       0.99      0.97      0.98     30557
           7       0.95      0.83      0.89     30441
           8       1.00      0.97      0.98     30604
           9       0.81      0.97      0.88     30386
          10       1.00      1.00      1.00     30495
          11       1.00      1.00      1.00     30447

    accuracy                           0.97    366269
   macro avg       0.97      0.97      0.97    366269
weighted avg       0.97      0.97      0.97    366269
```

```
RFC.predict(crimeaditionalinfo[2].reshape(1, -1))
```

```
array([9])
```

```
from sklearn.tree import DecisionTreeClassifier


DT=DecisionTreeClassifier()
DT.fit(x_train,y_train)
```

```
DecisionTreeClassifier()
```

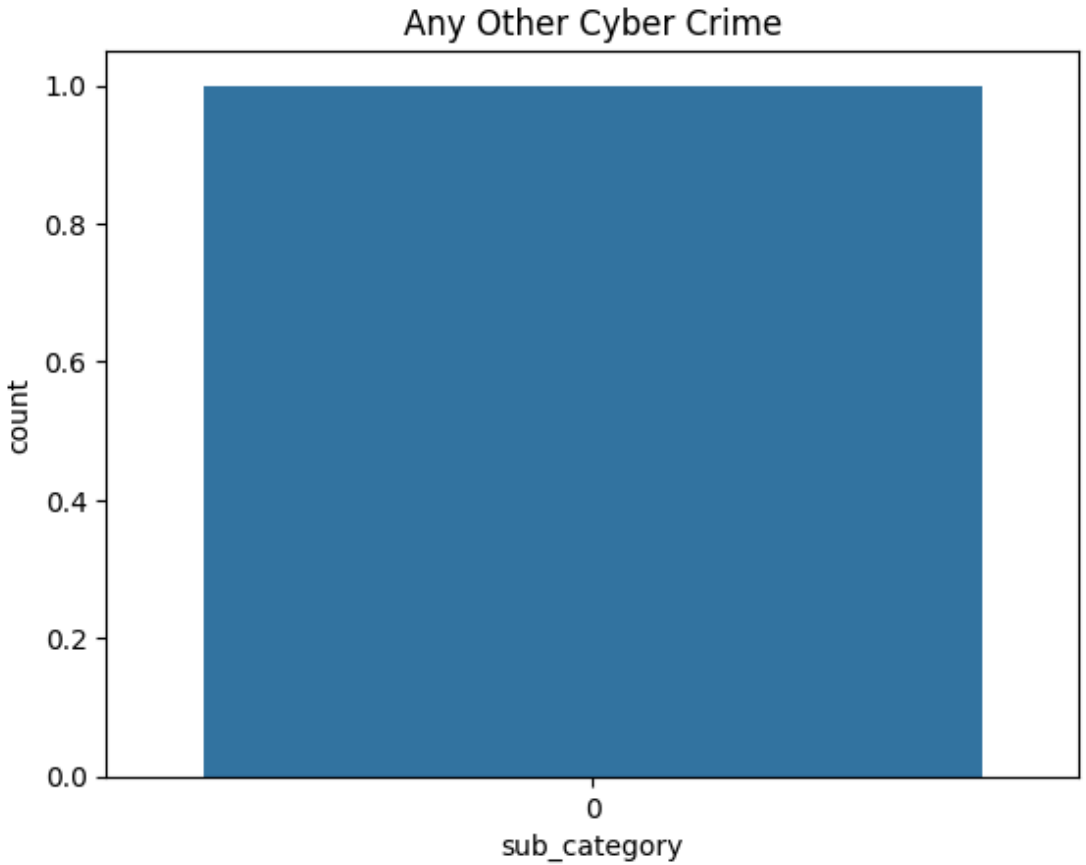
```
DT.score(x_test,y_test)
```

```
0.9548282819457831
```

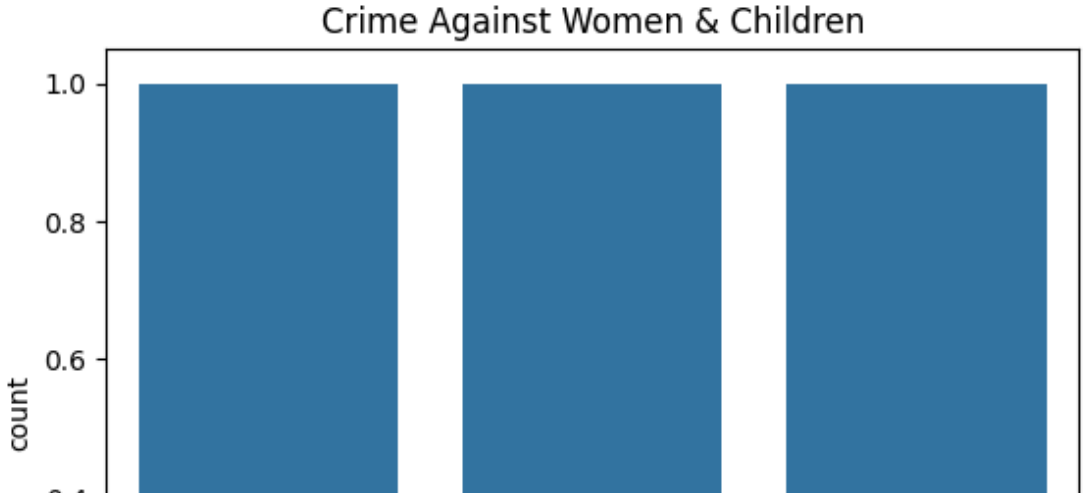
```
from sklearn.ensemble import GradientBoostingClassifier


data0=data[data["category"]==0]
data1=data[data["category"]==1]
data2=data[data["category"]==2]
data3=data[data["category"]==3]
data4=data[data["category"]==4]
data5=data[data["category"]==5]
data6=data[data["category"]==6]
data7=data[data["category"]==7]
data8=data[data["category"]==8]
data9=data[data["category"]==9]
data10=data[data["category"]==10]
data11=data[data["category"]==11]



i=0
def model_process(data_sub):
    data_sub = data_sub.drop("category", axis=1)
    data_sub = Encoder(data_sub, "sub_category")
    if (len(data_sub) == 1):
        data_sub = pd.concat([data_sub, data_sub, data_sub, data_sub, data_sub, data_sub, data_sub], axis=0)
```

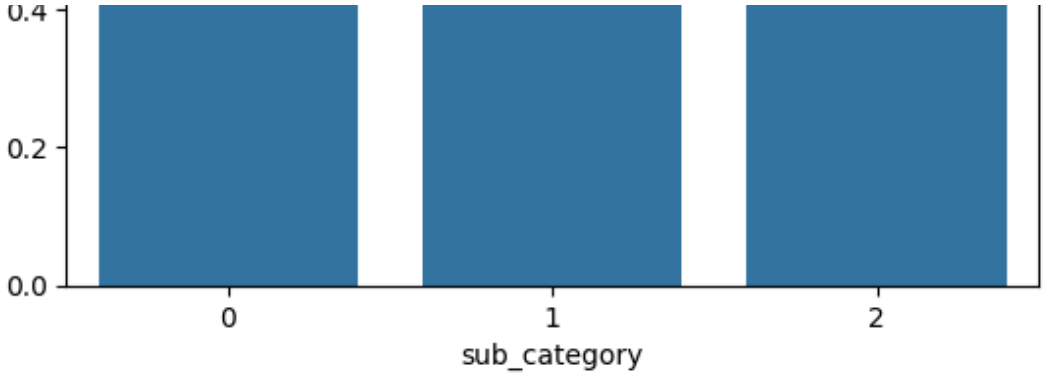
```
        X = data_sub["crimeaditionalinfo"]
        Y = data_sub["sub_category"]
        sns.countplot(Y.value_counts())
        plt.title(d[i])
        plt.show()
        print("Unique classes in Y:", Y.unique())

        if Y.nunique() <= 1:
            tfv = TfidfVectorizer(max_features=100)
            X = tfv.fit_transform(X).toarray()
            x_train, x_test, y_train, y_test = train_test_split(X, Y, test_size=0.2)
            RFC = RandomForestClassifier()
            RFC.fit(x_train, y_train)
            print(RFC.score(x_test, y_test))
            return RFC
        from imblearn.over_sampling import RandomOverSampler
        X_reshaped = pd.DataFrame(X).values.reshape(-1, 1)

        oversampler = RandomOverSampler(sampling_strategy='auto', random_state=100)

        X_resampled, y_resampled = oversampler.fit_resample(X_reshaped, Y)

        X_resampled = pd.DataFrame(X_resampled, columns=['crimeaditionalinfo'])
        sns.countplot(y_resampled.value_counts())
        plt.title(d[i])
        plt.show()
        tfv = TfidfVectorizer(max_features=100)
        X = tfv.fit_transform(X_resampled['crimeaditionalinfo']).toarray()
        x_train, x_test, y_train, y_test = train_test_split(X, y_resampled, test_size=0.2)
        RFC = RandomForestClassifier()
        RFC.fit(x_train, y_train)
        print(RFC.score(x_test, y_test))
        return RFC


    model0=model_process(data0)
    i+=1
    model1=model_process(data1)
    i+=1
```

```
    model2=model_process(data2)
    i+=1
    model3=model_process(data3)
    i+=1
    model4=model_process(data4)
    i+=1
    model5=model_process(data5)
    i+=1
    model6=model_process(data6)
    i+=1
    model7=model_process(data7)
    i+=1
    model8=model_process(data8)
    i+=1
    model9=model_process(data9)
    i+=1
    model10=model_process(data10)
    i+=1
    model11=model_process(data11)
```

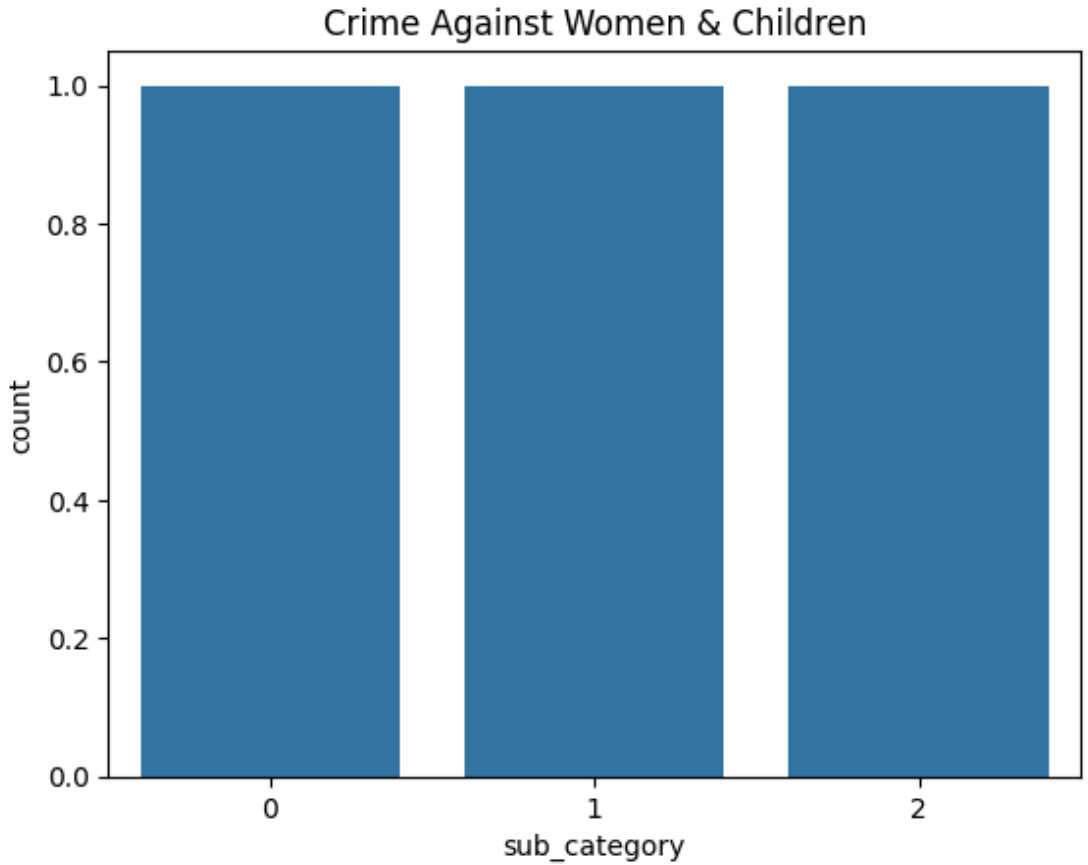



```
Unique classes in Y: [0]
1.0
```

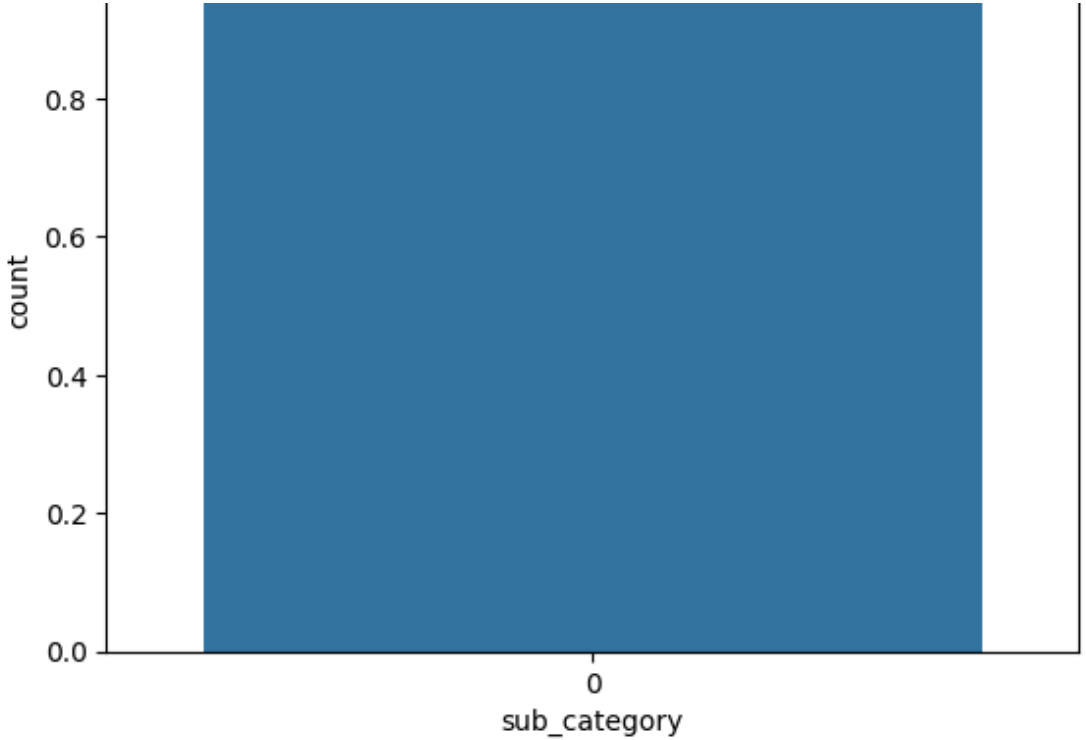




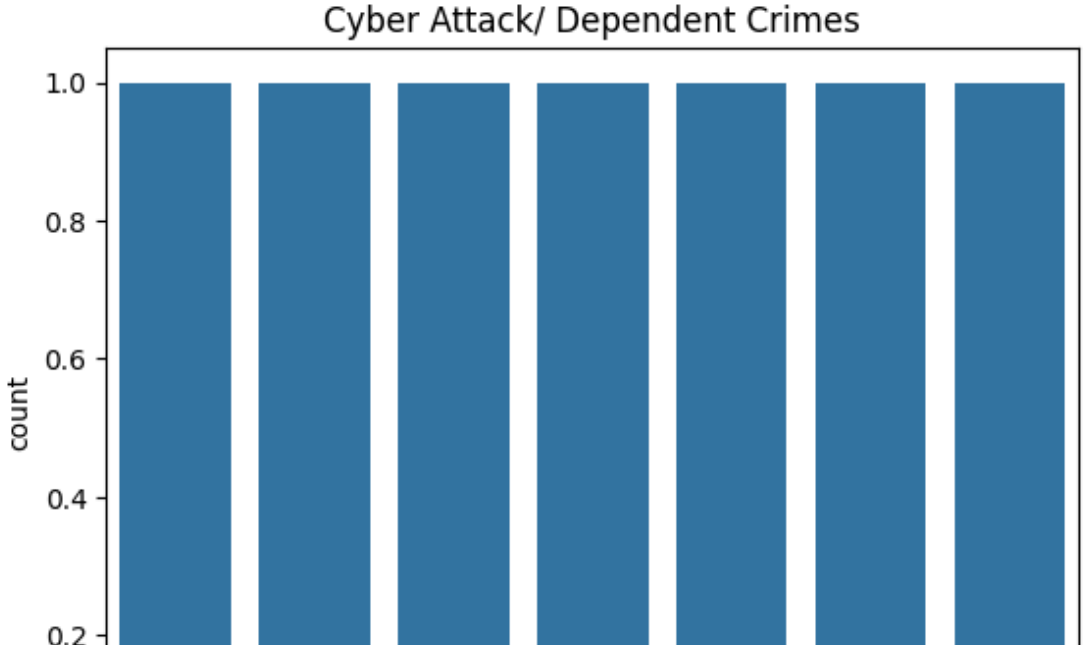

```
Unique classes in Y: [0 1 2]
```

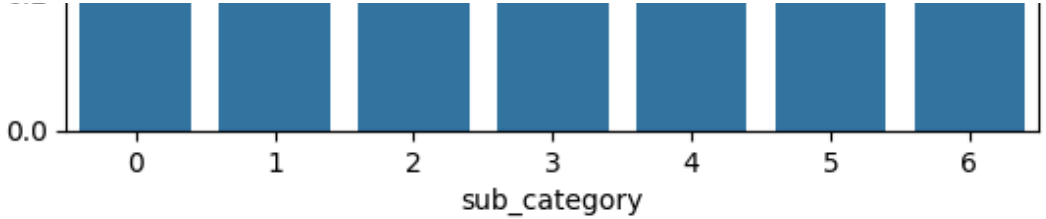



```
0.0
```

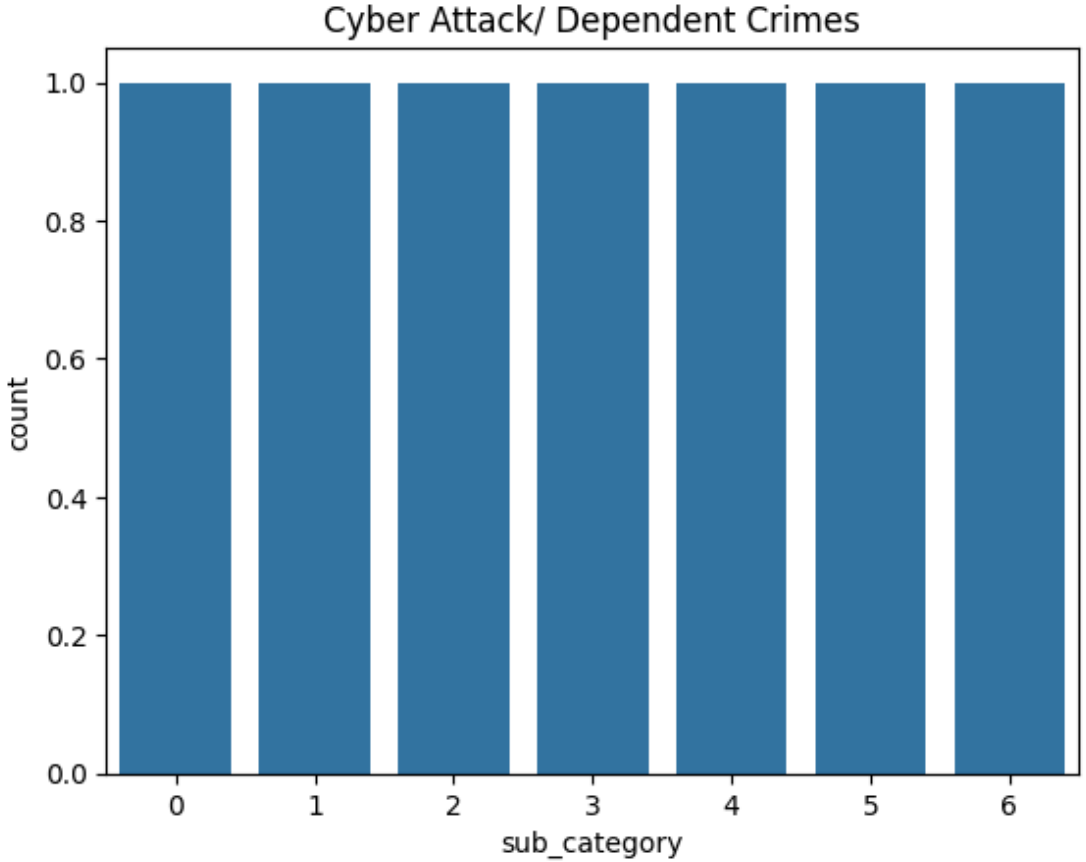



```
Unique classes in Y: [0]
1.0
```

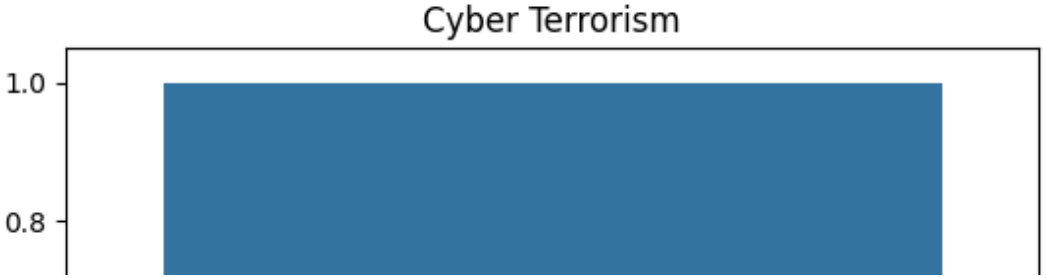


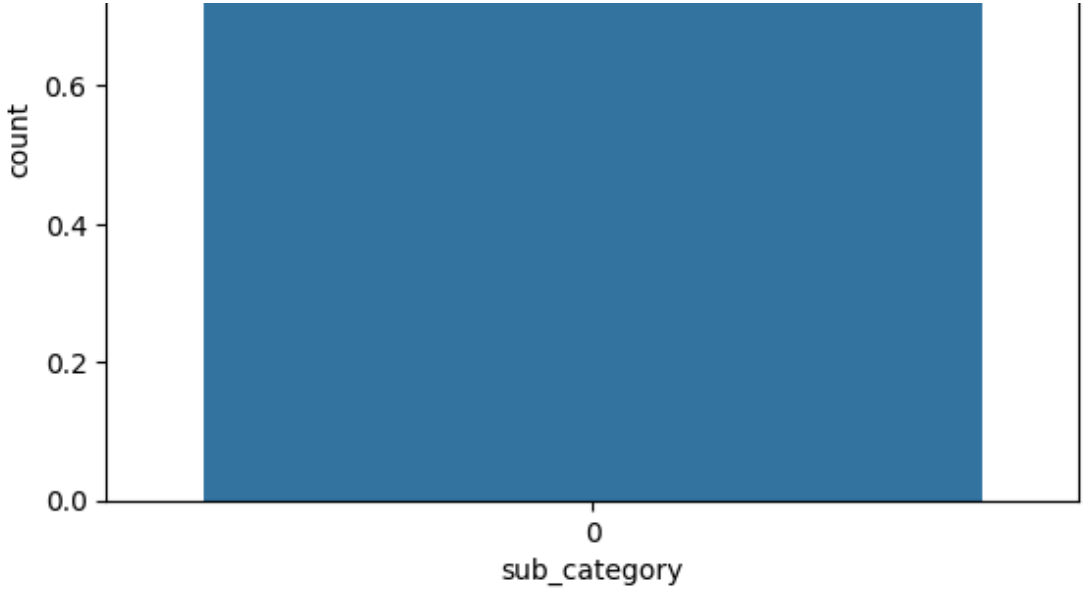



```
Unique classes in Y: [0 1 3 2 5 4 6]
```

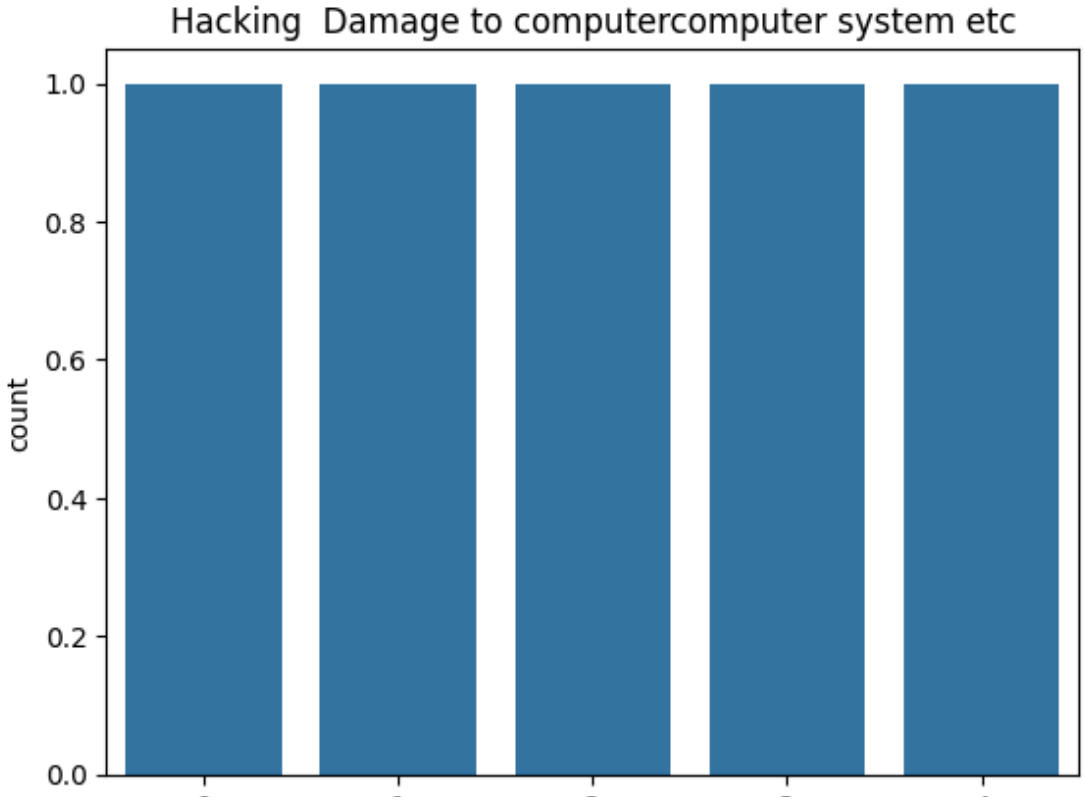


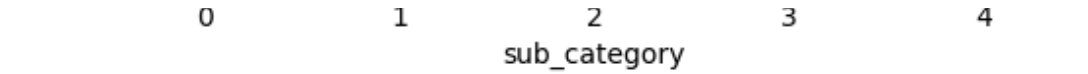

```
0.19498069498069498
```



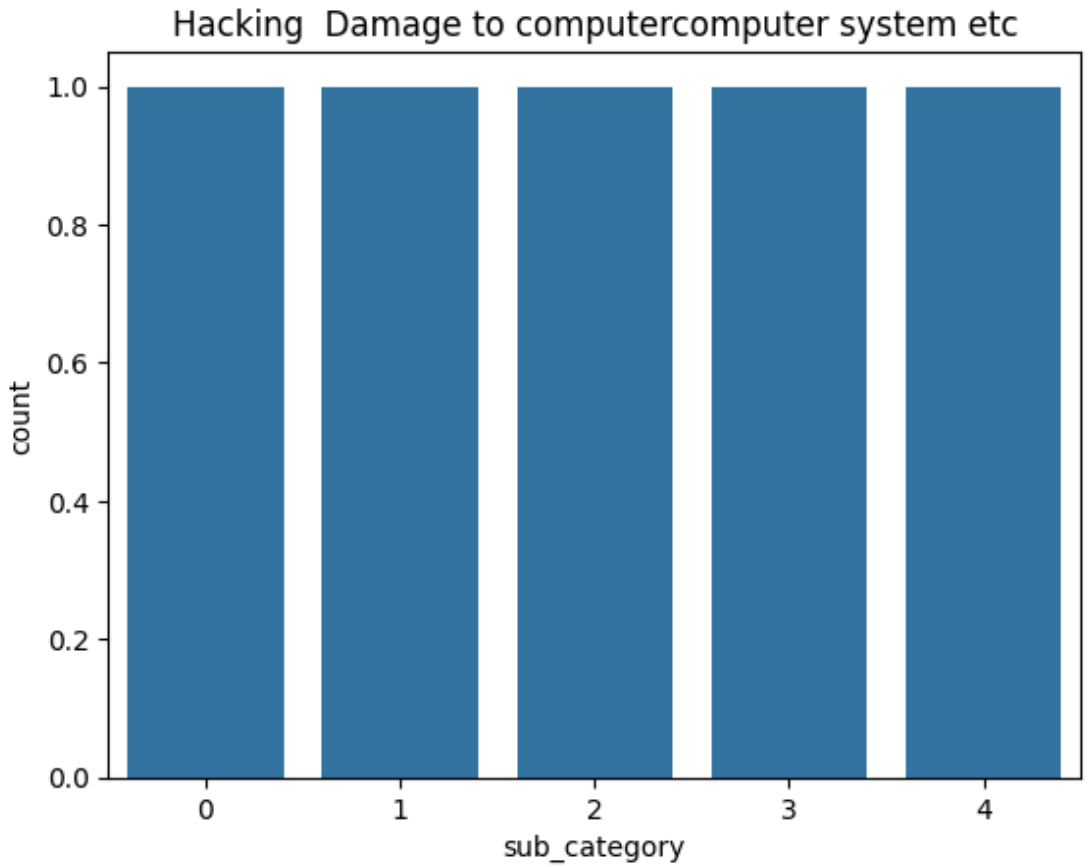



```
Unique classes in Y: [0]
1.0
```

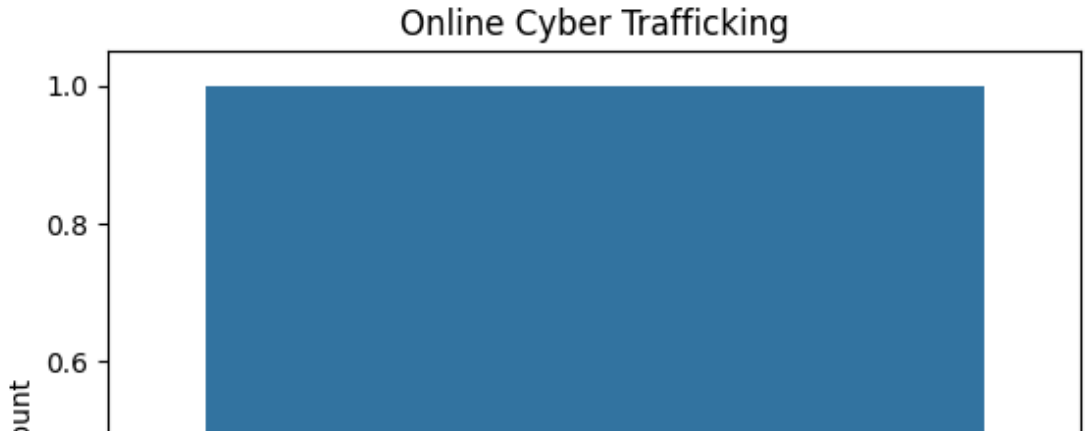


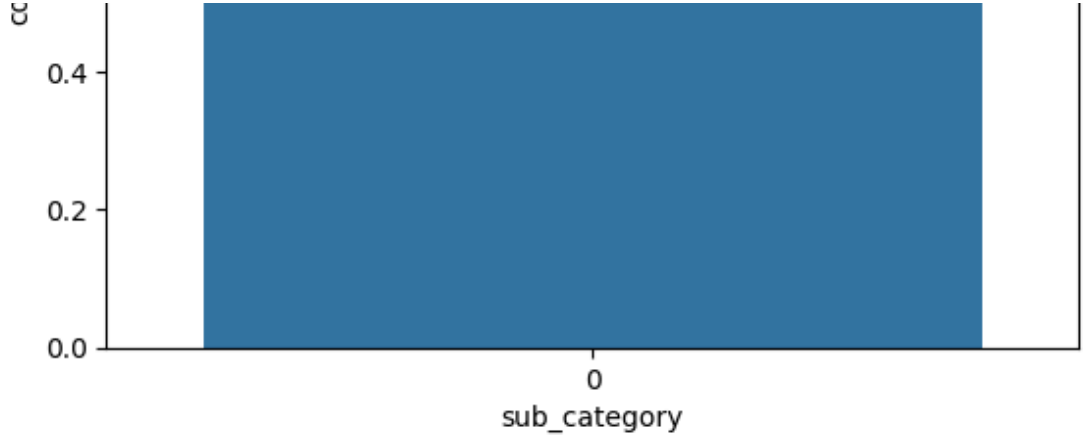



```
Unique classes in Y: [1 3 4 0 2]
```

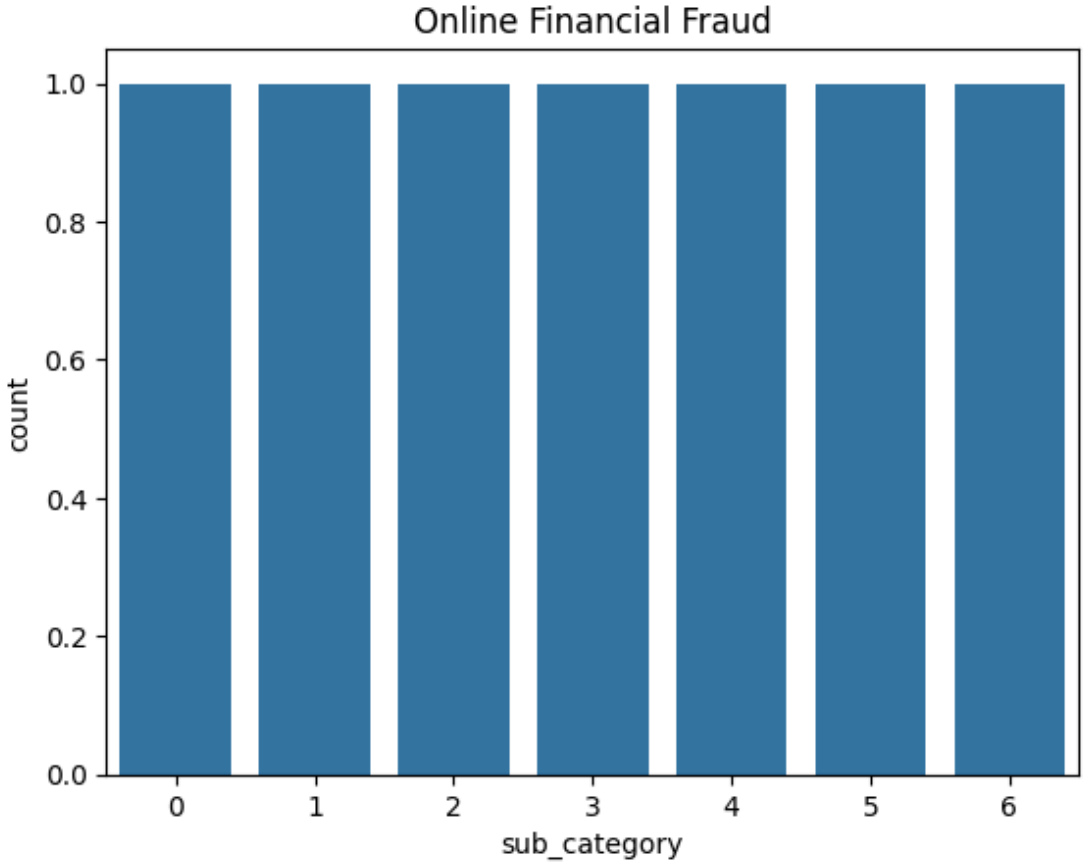



```
0.9649595687331537
```

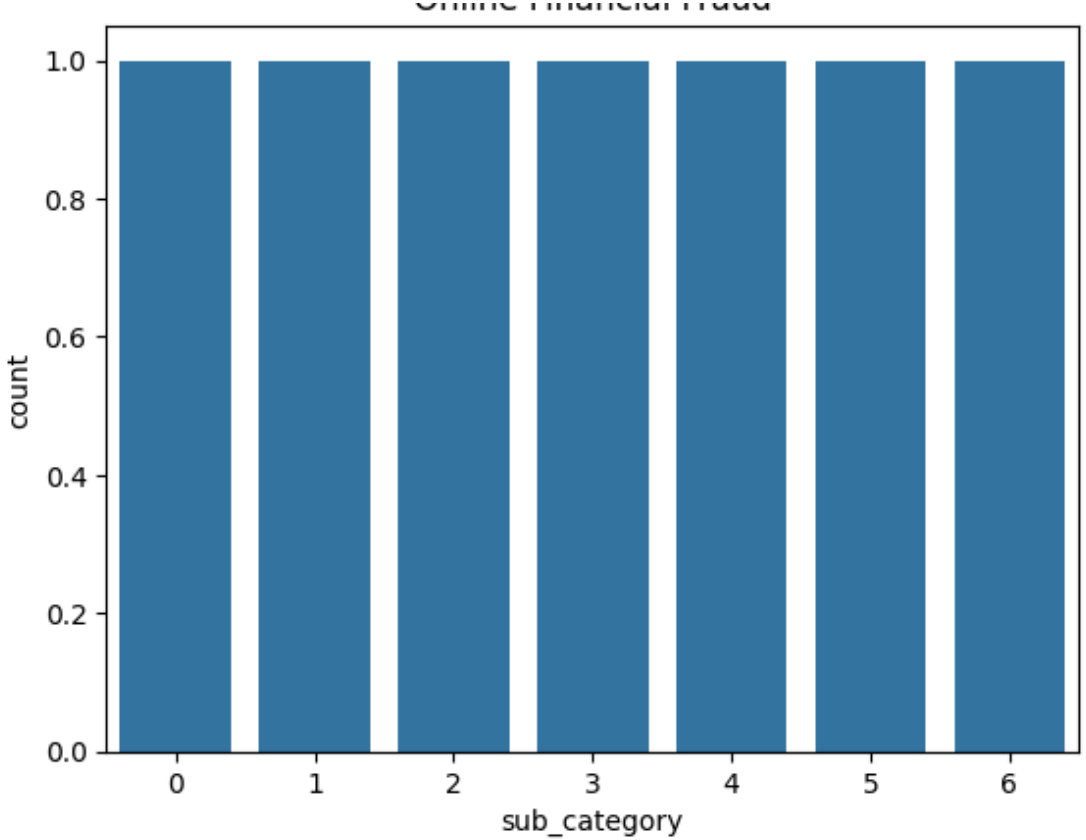




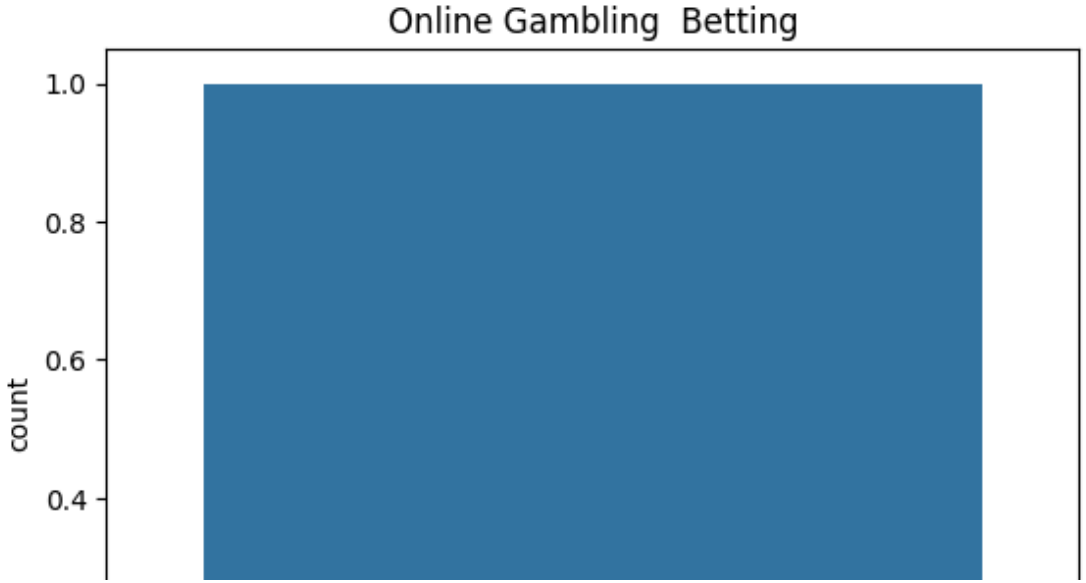

```
Unique classes in Y: [0]
1.0
```

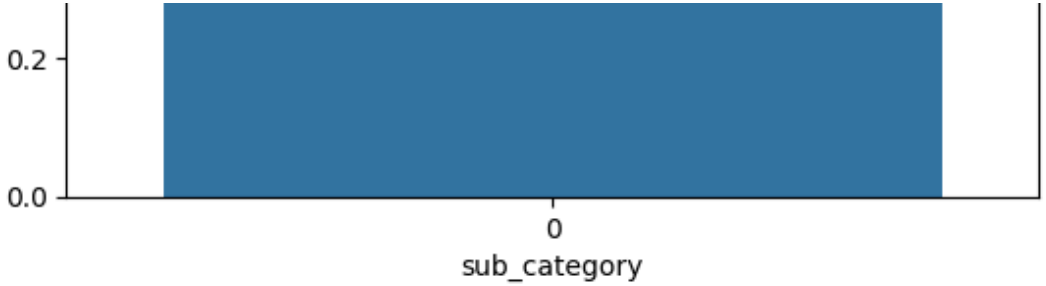



```
Unique classes in Y: [4 6 5 1 3 0 2]
```

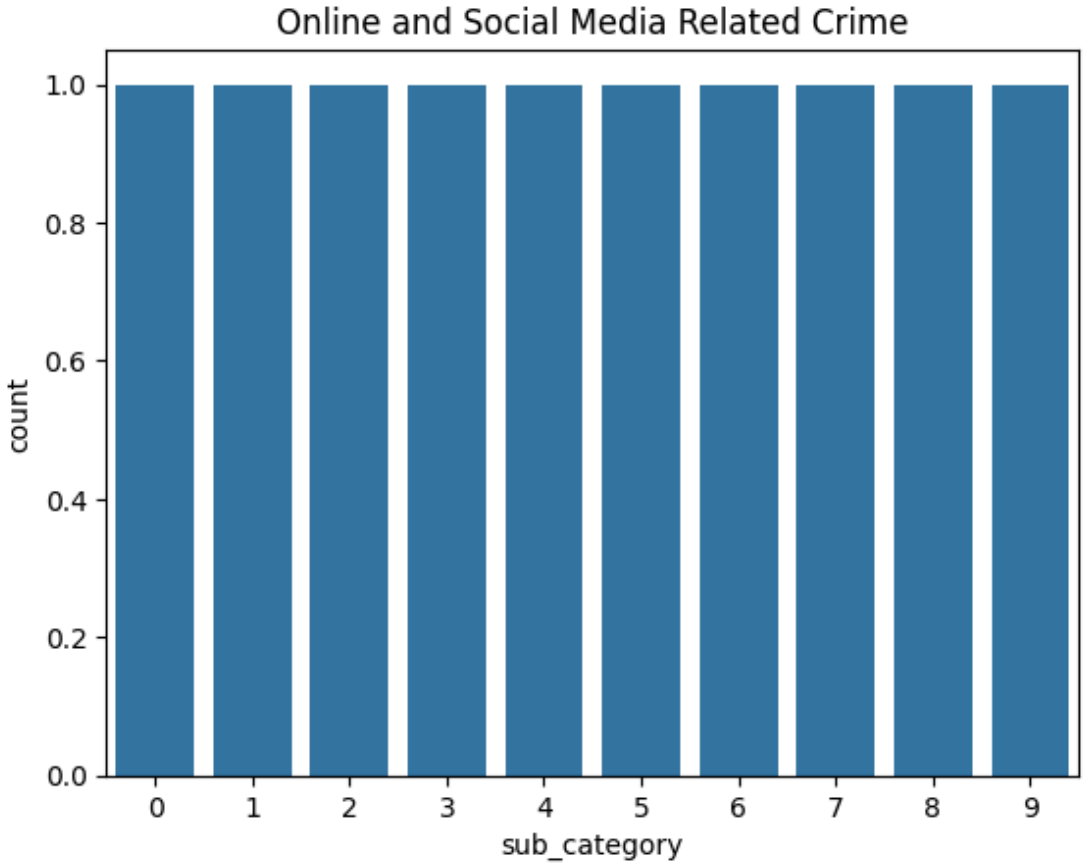



```
0.9134963315407529
```

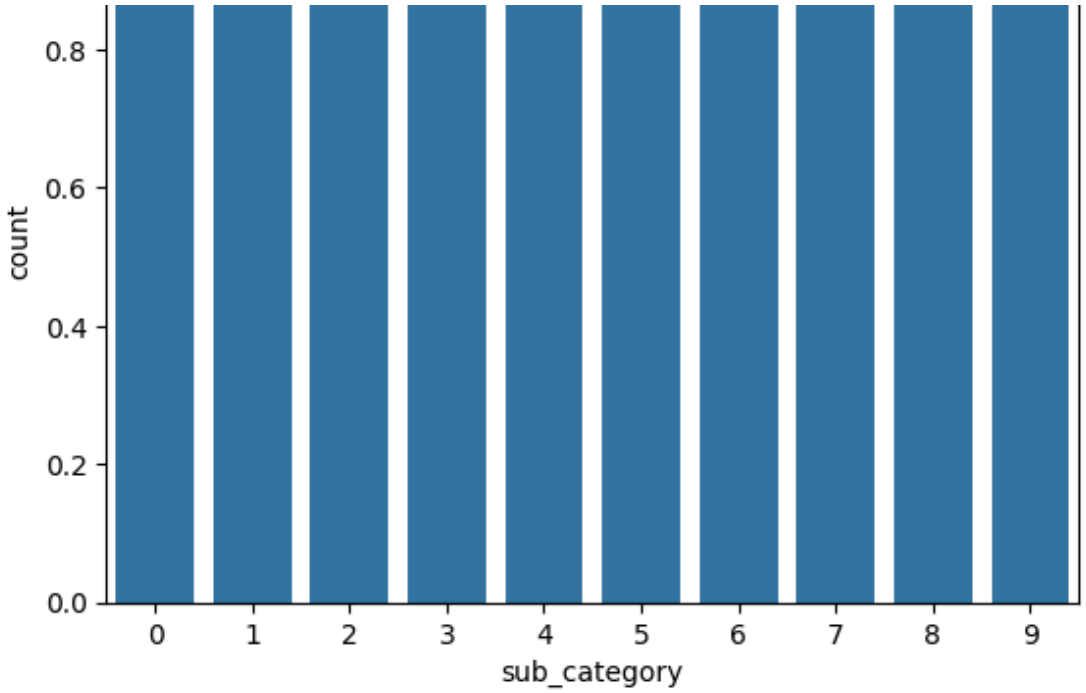




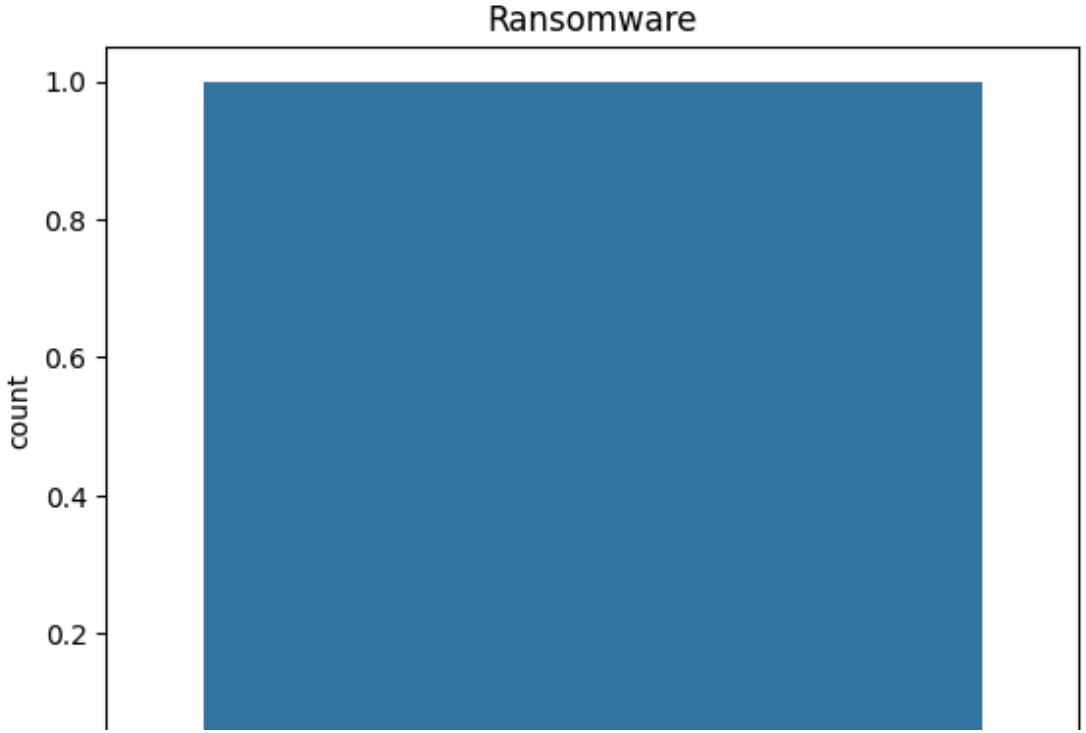

```
Unique classes in Y: [0]
1.0
```

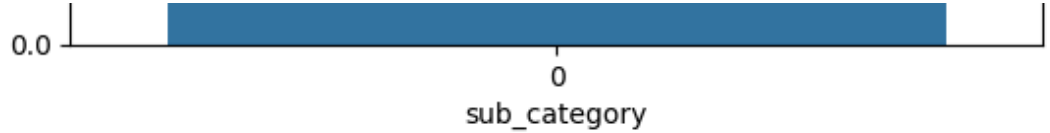



```
Unique classes in Y: [1 6 8 0 3 9 7 4 2 5]
```




```
0.8994500458295142
```






```
Unique classes in Y: [0]
1.0
```