In [6]:

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
%matplotlib inline
```

In [7]:

```
train=pd.read_csv ('Tweet_NFT.xlsx - Sheet1.csv')
```

In [8]:

```
train .head()
```

Out[8]:

	id	tweet_text	tweet_created_at	tweet_intent
0	1212762	@crypto_brody @eCoLoGy1990 @MoonrunnersNFT @It...	2022-08-06T16:56:36.000Z	Community
1	1212763	Need Sick Character artâ “#art #artist #Artist...	2022-08-06T16:56:36.000Z	Giveaway
2	1212765	@The_Hulk_NFT @INagotchiNFT @Tesla @killabears...	2022-08-06T16:56:35.000Z	Appreciation
3	1212766	@CryptoBatzNFT @DarekBTW The first project in ...	2022-08-06T16:56:35.000Z	Community
4	1212767	@sashadysonn The first project in crypto with ...	2022-08-06T16:56:34.000Z	Community

In [4]:

```
train.isnull()
```

Out[4]:

	id	tweet_text	tweet_created_at	tweet_intent
0	False	False	False	False
1	False	False	False	False
2	False	False	False	False
3	False	False	False	False
4	False	False	False	False
...	...	...	...	...
127448	False	False	False	True
127449	False	False	False	True
127450	False	False	False	True
127451	False	False	False	True
127452	False	False	False	True

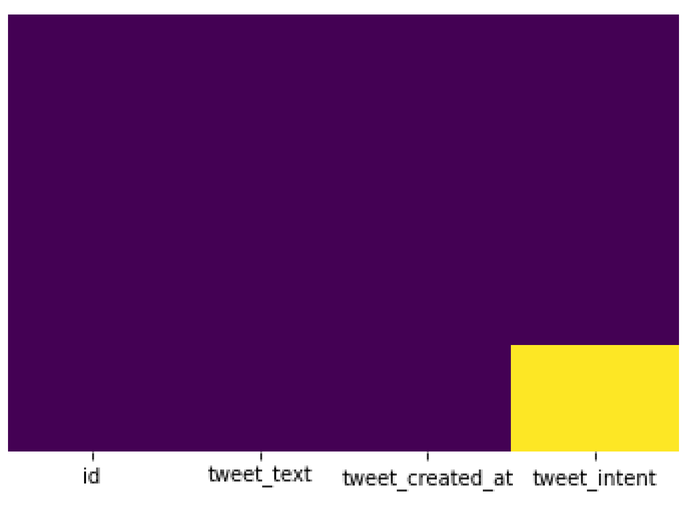
127453 rows × 4 columns

In [5]:

```
sns.heatmap(train.isnull(),yticklabels=False,cbar=False,cmap='viridis')
```

Out[5]:

```
<AxesSubplot:>
```

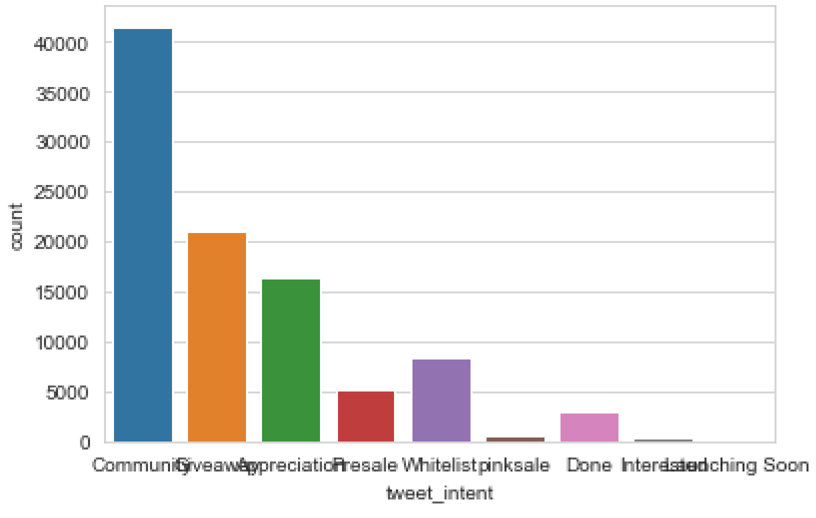


In [6]:

```
sns.set_style('whitegrid')
sns.countplot(x='tweet_intent',data=train)
```

Out[6]:

```
<AxesSubplot:xlabel='tweet_intent', ylabel='count'>
```



In [5]:

```
sns.set_style('whitegrid')
sns.countplot(x='tweet_intent',hue='tweet_created_at',data=train,palette='RdBu_r')
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
Input In [5], in <cell line: 2>()
      1 sns.set_style('whitegrid')
----> 2 sns.countplot(x='tweet_intent',hue='tweet_created_at',data=train,palette='RdBu_r')

NameError: name 'train' is not defined
```

In [3]:

```
sns.set_style('whitegrid')
sns.countplot(x='tweet_intent',hue='tweet_text',data=train,palette='rainbow')
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
Input In [3], in <cell line: 2>()
      1 sns.set_style('whitegrid')
----> 2 sns.countplot(x='tweet_intent',hue='tweet_text',data=train,palette='rainbow')

NameError: name 'train' is not defined
```

In [ ]:

```
sns.distplot(train['tweet_intent'].dropna(),kde=False,color='darkred',bins=40)
```

In [ ]:

```
train['tweet_intent'].hist(bins=30,color='darkred',alpha=0.3)
```

In [ ]:

```
sns.countplot(x='id',data=train)
```

In [ ]:

```
train['tweet_text'].hist(color='green',bins=40,figsize=(8,4))
```

In [ ]:

```
import cufflinks as cf
cf.go_offline()
```

In [ ]:

```
train['tweet_intent'].iplot(kind='hist',bins=30,color='green')
```

In [ ]:

```
plt.figure(figsize=(12, 7))
sns.boxplot(x='tweet_text',y='tweet_intent',data=train,palette='winter')
```

In [ ]:

```
def impute_tweet_intent(cols):
    tweet_intent = cols[0]
    tweet_created_at = cols[1]

    if pd.isnull(tweet_intent):

        if tweet_created_at == 2022-08-06T16:56:36.000Z:
            return community

        elif tweet_created_at == 2022-08-06T16:56:35.000Z:
            return appreciation

        else:
            return 24

    else:
        return tweet_intent
```

In [ ]:

```
train['tweet_intent'] = train[['tweet_intent','tweet_created_at']].apply(impute_tweet_intent,axis=1)
```

In [ ]:

```
sns.heatmap(train.isnull(),yticklabels=False,cbar=False,cmap='viridis')
```

In [ ]:

```
train.drop('id',axis=1,inplace=True)
```

In [ ]:

```
train.info()
```

In [ ]:

```
pd.get_dummies(train['tweet_text'],drop_first=True).head()
```

In [ ]:

```
tweet_created_at = pd.get_dummies(train['tweet_created_at'],drop_first=True)
tweet_intent = pd.get_dummies(train['tweet_intent'],drop_first=True)
```

In [ ]:

```
train.drop(['tweet_created_at','tweet_intent','id','tweet_text'],axis=1,inplace=True)
```

In [ ]:

```
train.head()
```

In [ ]:

```
train = pd.concat([train,tweet_created_at,tweet_intent],axis=1)
```

In [ ]:

```
train.head()
```

In [ ]:

```
train.drop('tweet_intent',axis=1).head()
```

In [ ]:

```
train['tweet_intent'].head()
```

In [ ]:

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

In [ ]:

```
X_train, X_test, y_train, y_test = train_test_split(train.drop('tweet_intent',axis=1),
                                                    train['tweet_intent'], test_size=0.30,
                                                    random_state=101)
```

In [ ]:

```
from sklearn.linear_model import LogisticRegression
```

In [ ]:

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

In [ ]:

```
predictions = logmodel.predict(X_test)
```

In [ ]:

```
from sklearn.metrics import confusion_matrix
```

In [ ]:

```
accuracy=confusion_matrix(y_test,predictions)
```

In [ ]:

```
accuracy
```

In [ ]:

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

In [ ]:

```
accuracy=accuracy_score(y_test,predictions)
accuracy
```

In [ ]:

```
predictions
```

In [ ]:

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

In [9]:

```
print(classification_report(y_test,predictions))
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
Input In [9], in <cell line: 1>()
----> 1 print(classification_report(y_test,predictions))

NameError: name 'classification_report' is not defined
```

In [ ]: