In [1]:

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

In [2]:

```
train = pd.read_csv(r"train.csv")
test = pd.read_csv(r"test.csv")
```

In [3]:

```
train
```

Out[3]:

	id	label	tweet
0	1	0	@user when a father is dysfunctional and is s...
1	2	0	@user @user thanks for #lyft credit i can't us...
2	3	0	bihday your majesty
3	4	0	#model i love u take with u all the time in ...
4	5	0	factsguide: society now #motivation
...	...	...	...
31957	31958	0	ate @user isz that youuu?ð□□□ð□□□ð□□□ð□□□ð□□□ð...
31958	31959	0	to see nina turner on the airwaves trying to...
31959	31960	0	listening to sad songs on a monday morning otw...
31960	31961	1	@user #sikh #temple vandalised in in #calgary,...
31961	31962	0	thank you @user for you follow

31962 rows × 3 columns

In [4]:

```
test
```

Out[4]:

	id	tweet
0	31963	#studiolife #aislife #requires #passion #dedic...
1	31964	@user #white #supremacists want everyone to s...
2	31965	safe ways to heal your #acne!! #altwaystohe...
3	31966	is the hp and the cursed child book up for res...
4	31967	3rd #bihday to my amazing, hilarious #nephew...
...	...	...
17192	49155	thought factory: left-right polarisation! #tru...
17193	49156	feeling like a mermaid ð□□□ #hairflip #neverre...
17194	49157	#hillary #campaigned today in #ohio((omg)) &am...
17195	49158	happy, at work conference: right mindset leads...
17196	49159	my song "so glad" free download! #shoegaze ...

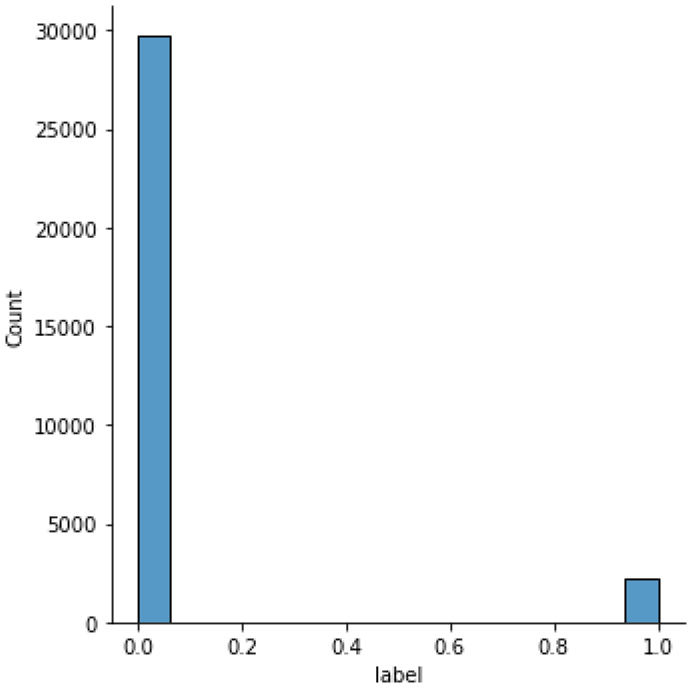
17197 rows × 2 columns

In [5]:

```
sns.displot(train['label'])
```

Out[5]:

```
<seaborn.axisgrid.FacetGrid at 0x1f961cbd700>
```



In [6]:

```
label_cnt = train['label'].value_counts()
label_cnt
```

Out[6]:

```
0    29720
1     2242
Name: label, dtype: int64
```

In [7]:

```python
label_pct = train['label'].value_counts() / len(train)
label_pct
```

Out[7]:

```
0    0.929854
1    0.070146
Name: label, dtype: float64
```

In [8]:

```python
label = train['label']

train.drop(['label'], axis=1, inplace=True)
train
```

Out[8]:

	id	tweet
0	1	@user when a father is dysfunctional and is s...
1	2	@user @user thanks for #lyft credit i can't us...
2	3	bihday your majesty
3	4	#model i love u take with u all the time in ...
4	5	factsguide: society now #motivation
...	...	...
31957	31958	ate @user isz that youuu?ð□□□ð□□□ð□□□ð□□□ð□□□ð...
31958	31959	to see nina turner on the airwaves trying to...
31959	31960	listening to sad songs on a monday morning otw...
31960	31961	@user #sikh #temple vandalised in in #calgary,...
31961	31962	thank you @user for you follow

31962 rows × 2 columns

In [9]:

```python
combi = train.append(test)
combi
```

Out[9]:

	id	tweet
0	1	@user when a father is dysfunctional and is s...
1	2	@user @user thanks for #lyft credit i can't us...
2	3	bihday your majesty
3	4	#model i love u take with u all the time in ...
4	5	factsguide: society now #motivation
...	...	...
17192	49155	thought factory: left-right polarisation! #tru...
17193	49156	feeling like a mermaid ð□□□ #hairflip #neverre...
17194	49157	#hillary #campaigned today in #ohio((omg)) &am...
17195	49158	happy, at work conference: right mindset leads...
17196	49159	my song "so glad" free download! #shoegaze ...

49159 rows × 2 columns

In [10]:
```
tweets = combi['tweet']

count_words = tweets.str.findall(r'(\w+)').str.len()
print(count_words.sum())
```

```
681137
```

In [11]:
```
import re
from nltk.corpus import stopwords

tweets = tweets.str.lower()


tweets = tweets.apply(lambda x : re.sub("[^a-z\s]","",x) )


tweets = tweets.str.replace("#", " ")


tweets = tweets.apply(lambda x: ' '.join([w for w in x.split() if len(w)>2]


stopwords = set(stopwords.words("english"))
tweets = tweets.apply(lambda x : " ".join(word for word in x.split() if wor


count_words = tweets.str.findall(r'(\w+)').str.len()
print(count_words.sum())
```

```
394674
```

In [12]:

```
most_freq_words = pd.Series(' '.join(tweets).lower().split()).value_counts(
tweets = tweets.apply(lambda x : " ".join(word for word in x.split() if wor
print(most_freq_words)

count_words = tweets.str.findall(r'(\w+)').str.len()
print(count_words.sum())
```

```
user        27008
love         4217
day          3471
happy        2630
amp          2433
time         1745
life         1719
today        1555
new          1546
like         1527
positive     1423
get          1406
thankful     1403
people       1331
bihday       1327
good         1313
cant         1239
one          1219
see          1136
fathers      1134
dont         1133
smile        1077
want          986
healthy       962
take          945
dtype: int64
328789
```

In [13]:

```
from collections import Counter
from itertools import chain

v = tweets.str.split().tolist()

c = Counter(chain.from_iterable(v))

tweets = [' '.join([j for j in i if c[j] > 1]) for i in v]

total_word = 0
for x,word in enumerate(tweets):
    num_word = len(word.split())

    total_word = total_word + num_word
print(total_word)
```

```
296750
```

In [14]:

```
X = np.array(tweets[: len(train)])
y = label
```

In [15]:

```
from sklearn.model_selection import train_test_split

X_train,X_val, y_train, y_val = train_test_split(X,y, stratify=y, test_size
X_train.shape, y_train.shape, X_val.shape,y_val.shape
```

Out[15]:

```
((22373,), (22373,), (9589,), (9589,))
```

In [16]:

```
from sklearn.feature_extraction.text import TfidfVectorizer

vectorizer_tfidf = TfidfVectorizer(stop_words='english', max_df=0.7, min_df
train_tfIdf = vectorizer_tfidf.fit_transform(X_train.astype('U'))
val_tfIdf = vectorizer_tfidf.transform(X_val.astype('U'))
print(vectorizer_tfidf.get_feature_names()[:5])
```

```
['affirmation', 'amazing', 'beautiful', 'best', 'blog']
```

In [17]:

```
train_tfIdf.shape,  val_tfIdf.shape
```

Out[17]:

```
((22373, 45), (9589, 45))
```

In [18]:

```
from sklearn.neighbors import KNeighborsClassifier

model = KNeighborsClassifier(n_neighbors=5).fit(train_tfIdf, y_train)
print(model.score(train_tfIdf, y_train))
```

```
0.9304071872346131
```

In [19]:

```
y_pred = model.predict(val_tfIdf)
print(model.score(val_tfIdf, y_val))
```

```
0.9313797059130253
```

In [20]:

```
from sklearn.metrics import confusion_matrix

print(confusion_matrix(y_val, y_pred))
```

```
[[8881   35]
 [ 623   50]]
```

In [23]:

```
from sklearn.metrics import classification_report
print(classification_report(y_val, y_pred))
```

```
              precision    recall  f1-score   support

           0       0.93      1.00      0.96      8916
           1       0.59      0.07      0.13       673

    accuracy                           0.93      9589
   macro avg       0.76      0.54      0.55      9589
weighted avg       0.91      0.93      0.91      9589
```