In [79]:
```
# Importing the libraries
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import sklearn.metrics
```

In [55]:
```
# Importing the dataset
dataset = pd.read_table('Restaurant_Reviews.tsv')
```

In [56]:
```
dataset
```

Out[56]:

	Review	Liked
0	Wow... Loved this place.	1
1	Crust is not good.	0
2	Not tasty and the texture was just nasty.	0
3	Stopped by during the late May bank holiday of...	1
4	The selection on the menu was great and so wer...	1
...	...	...
995	I think food should have flavor and texture an...	0
996	Appetite instantly gone.	0
997	Overall I was not impressed and would not go b...	0
998	The whole experience was underwhelming, and I ...	0
999	Then, as if I hadn't wasted enough of my life ...	0

1000 rows × 2 columns

In [57]:
```
dataset.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1000 entries, 0 to 999
Data columns (total 2 columns):
 #   Column  Non-Null Count  Dtype 
---  ------  --------------  ----- 
 0   Review  1000 non-null   object
 1   Liked   1000 non-null   int64 
dtypes: int64(1), object(1)
memory usage: 15.8+ KB
```

In [58]:
```
dataset.describe()
```

Out[58]:

	Liked
count	1000.00000
mean	0.50000
std	0.50025
min	0.00000
25%	0.00000
50%	0.50000
75%	1.00000
max	1.00000

In [59]:
```
dataset.columns
```

Out[59]:
```
Index(['Review', 'Liked'], dtype='object')
```

In [60]:
```
dataset['Liked'].nunique()
```

Out[60]:
```
2
```

In [61]:
```
print(dataset['Liked'].unique())
```

```
[1 0]
```

In [62]:
```
dataset['Liked'].value_counts()
```

Out[62]:
```
1    500
0    500
Name: Liked, dtype: int64
```

In [63]:
```
dataset.head()
```

Out[63]:

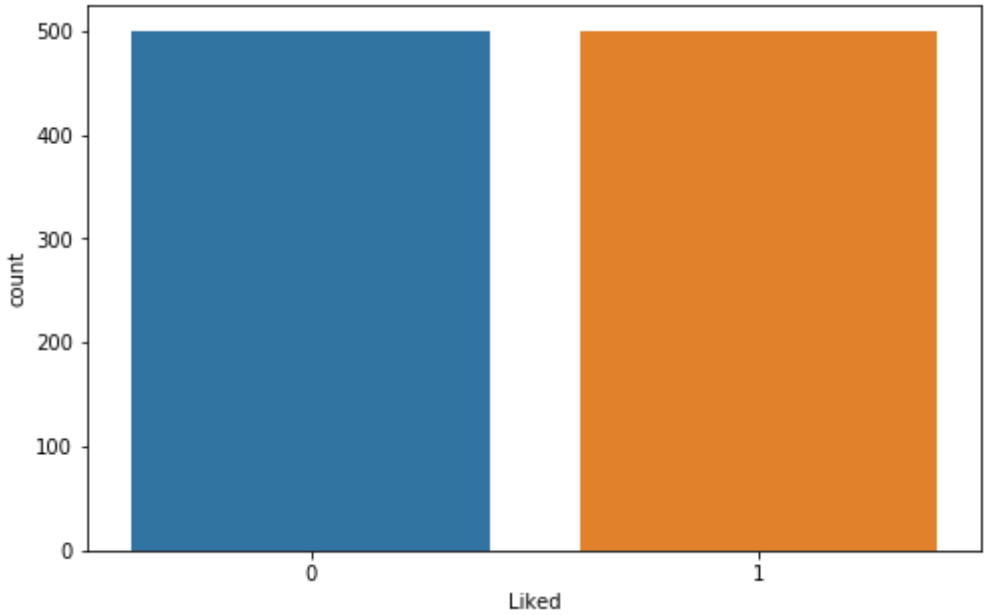
	Review	Liked
0	Wow... Loved this place.	1
1	Crust is not good.	0
2	Not tasty and the texture was just nasty.	0
3	Stopped by during the late May bank holiday of...	1
4	The selection on the menu was great and so wer...	1

In [64]:
```
dataset.tail()
```

Out[64]:

	Review	Liked
995	I think food should have flavor and texture an...	0
996	Appetite instantly gone.	0
997	Overall I was not impressed and would not go b...	0
998	The whole experience was underwhelming, and I ...	0
999	Then, as if I hadn't wasted enough of my life ...	0

In [65]:
```
plt.figure(figsize=(8,5))
sns.countplot(x=dataset.Liked);
```



In [66]:
```
x=dataset['Review'].values
y=dataset['Liked'].values
```

In [67]:
```
from sklearn.model_selection import train_test_split
x_train,x_test,y_train,y_test=train_test_split(x,y,random_state=0)
```

In [68]:
```
x_train.shape
```

Out[68]:
```
(750,)
```

In [69]:
```
x_test.shape
```

Out[69]:
```
(250,)
```

In [70]:
```
y_train.shape
```

Out[70]:
```
(750,)
```

In [71]:
```
y_test.shape
```

Out[71]:
```
(250,)
```

In [72]:
```
from sklearn.feature_extraction.text import CountVectorizer
vect=CountVectorizer(stop_words='english')
```

In [73]:
```
x_train_vect=vect.fit_transform(x_train)
x_test_vect=vect.transform(x_test)
```

In [74]:
```
from sklearn.svm import SVC
model=SVC()
```

In [75]:
```
model.fit(x_train_vect,y_train)
```

Out[75]:
```
SVC(C=1.0, break_ties=False, cache_size=200, class_weight=None, coef0=0.0,
    decision_function_shape='ovr', degree=3, gamma='scale', kernel='rbf',
    max_iter=-1, probability=False, random_state=None, shrinking=True,
    tol=0.001, verbose=False)
```

In [76]:
```
y_pred=model.predict(x_test_vect)
```

In [81]:
```
sklearn.metrics.accuracy_score(y_pred,y_test)
```

Out[81]:
```
0.72
```

In [90]:
```
from sklearn.pipeline import make_pipeline
text_model=make_pipeline(CountVectorizer(),SVC())
```

In [91]:
```
text_model.fit(x_train,y_train)
```

Out[91]:
```
Pipeline(memory=None,
         steps=[('countvectorizer',
                 CountVectorizer(analyzer='word', binary=False,
                                 decode_error='strict',
                                 dtype=<class 'numpy.int64'>, encoding='utf-8',
                                 input='content', lowercase=True, max_df=1.0,
                                 max_features=None, min_df=1,
                                 ngram_range=(1, 1), preprocessor=None,
                                 stop_words=None, strip_accents=None,
                                 token_pattern='(?u)\\b\\w\\w+\\b',
                                 tokenizer=None, vocabulary=None)),
                ('svc',
                 SVC(C=1.0, break_ties=False, cache_size=200, class_weight=None,
                     coef0=0.0, decision_function_shape='ovr', degree=3,
                     gamma='scale', kernel='rbf', max_iter=-1,
                     probability=False, random_state=None, shrinking=True,
                     tol=0.001, verbose=False))],
         verbose=False)
```

In [92]:
```
y_pred=text_model.predict(x_test)
```

In [93]:
```
y_pred
```

Out[93]:
```
array([0, 0, 0, 0, 1, 1, 1, 0, 0, 1, 1, 1, 0, 1, 1, 1, 0, 0, 0, 1, 0, 1,
       1, 0, 0, 1, 1, 1, 1, 0, 1, 0, 0, 0, 0, 0, 0, 1, 0, 1, 1, 0, 1, 0,
       0, 1, 0, 1, 0, 0, 1, 1, 1, 1, 0, 0, 0, 1, 0, 1, 1, 0, 1, 1, 0, 1,
       1, 0, 1, 0, 1, 1, 1, 0, 0, 0, 0, 1, 0, 1, 0, 1, 1, 0, 1, 1, 0, 0,
       1, 0, 0, 1, 0, 0, 0, 1, 1, 0, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 0, 0,
       0, 1, 0, 1, 1, 0, 1, 1, 1, 1, 0, 0, 1, 0, 0, 0, 1, 0, 0, 0, 0, 1,
       0, 0, 1, 1, 1, 1, 1, 0, 1, 1, 1, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1,
       0, 0, 1, 0, 1, 0, 0, 0, 1, 1, 0, 0, 1, 0, 0, 0, 0, 0, 1, 0, 0, 0,
       0, 1, 0, 0, 1, 0, 0, 1, 1, 0, 1, 1, 0, 1, 1, 0, 0, 1, 1, 0, 1, 1,
       0, 0, 1, 1, 1, 0, 1, 1, 1, 0, 0, 0, 1, 1, 0, 1, 1, 0, 1, 1, 1, 1,
       0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 1, 1, 0, 1, 0, 0, 1, 0, 0, 1, 1,
       1, 1, 0, 1, 1, 0, 0, 0], dtype=int64)
```

In [95]:
```
sklearn.metrics.accuracy_score(y_pred,y_test)
```

Out[95]:
```
0.792
```

In [96]:
```
import joblib
joblib.dump(text_model,'Project')
```

Out[96]:
```
['Project']
```

In [98]:
```
text_model.predict(['hello!!Love Your Food'])
```

Out[98]:
```
array([1], dtype=int64)
```

In [99]:
```
text_model.predict(["omg!!it was too spice and i asked you don't add too much "])
```

Out[99]:
```
array([0], dtype=int64)
```

In [ ]:
```

```