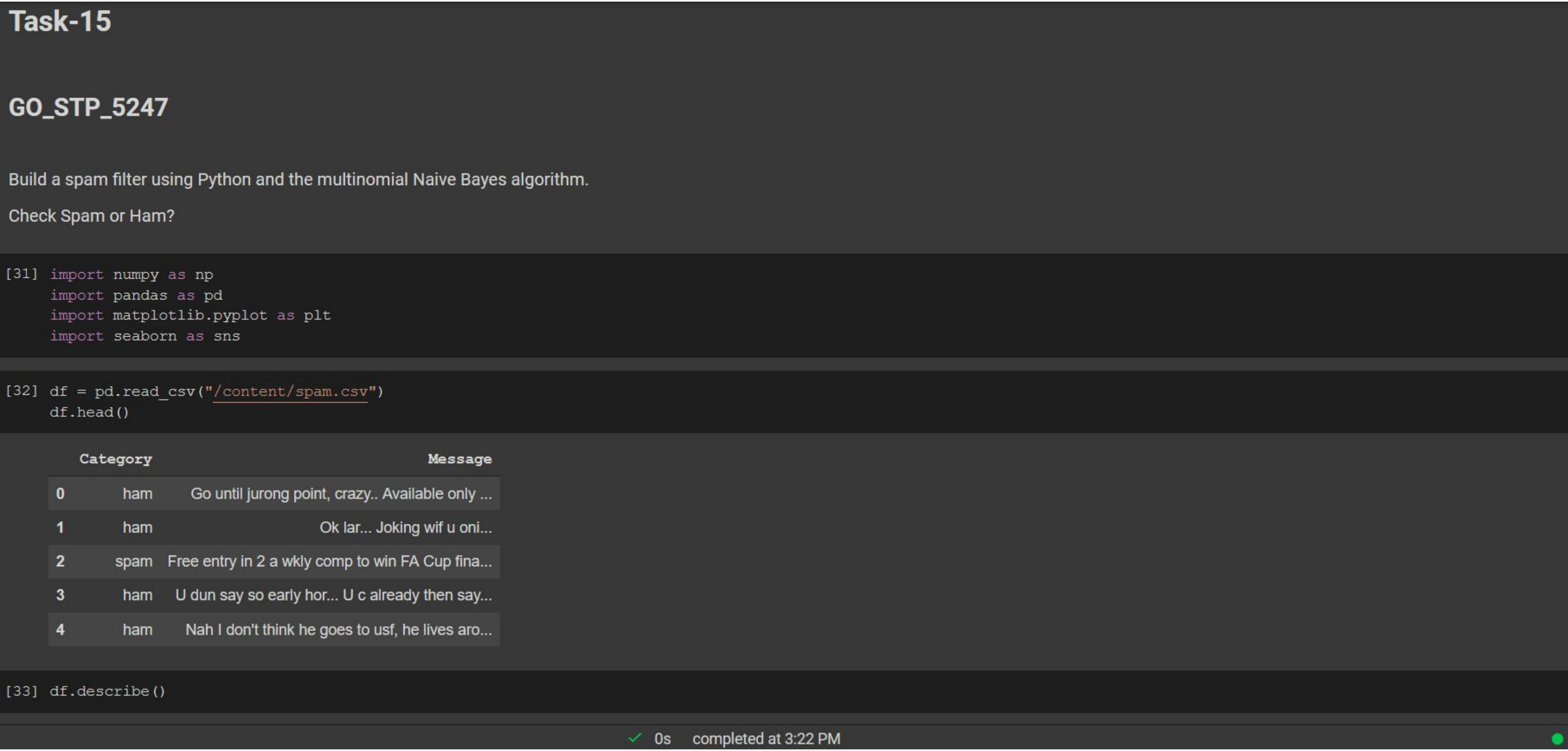


Task-15

GO_STP_5247

Build a spam filter using Python and the multinomial Naive Bayes algorithm.

Check Spam or Ham?

[31]

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

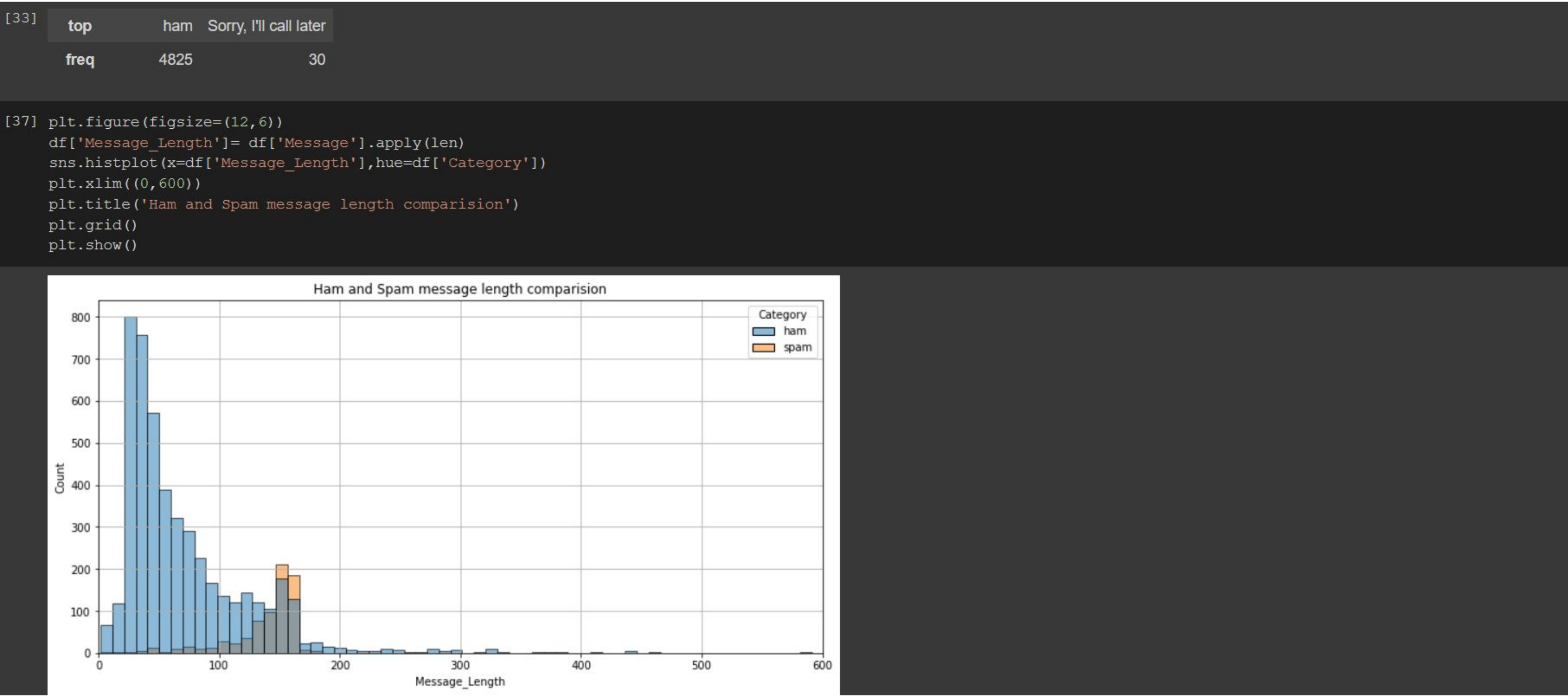
[32]

```
df = pd.read_csv("/content/spam.csv")
df.head()
```

	Category	Message
0	ham	Go until jurong point, crazy.. Available only ...
1	ham	Ok lar... Joking wif u oni...
2	spam	Free entry in 2 a wkly comp to win FA Cup fina...
3	ham	U dun say so early hor... U c already then say...
4	ham	Nah I don't think he goes to usf, he lives aro...

[33]

```
df.describe()
```

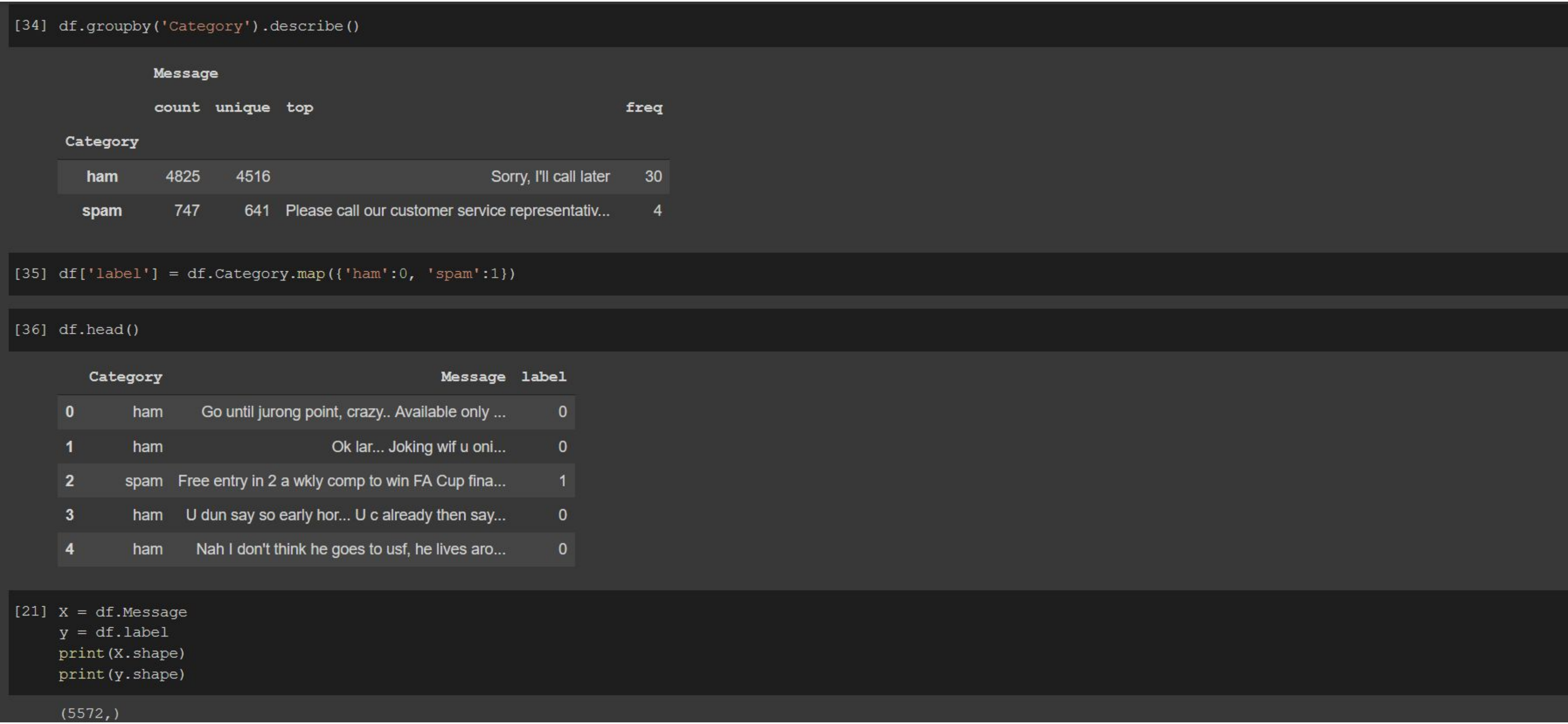


[33]

top	ham	Sorry, I'll call later
freq	4825	30

[37]

```
plt.figure(figsize=(12,6))
df['Message_Length']= df['Message'].apply(len)
sns.histplot(x=df['Message_Length'],hue=df['Category'])
plt.xlim((0,600))
plt.title('Ham and Spam message length comparision')
plt.grid()
plt.show()
```



```
[34] df.groupby('Category').describe()
```

	Message			
	count	unique	top	freq
Category				
ham	4825	4516	Sorry, I'll call later	30
spam	747	641	Please call our customer service representativ...	4

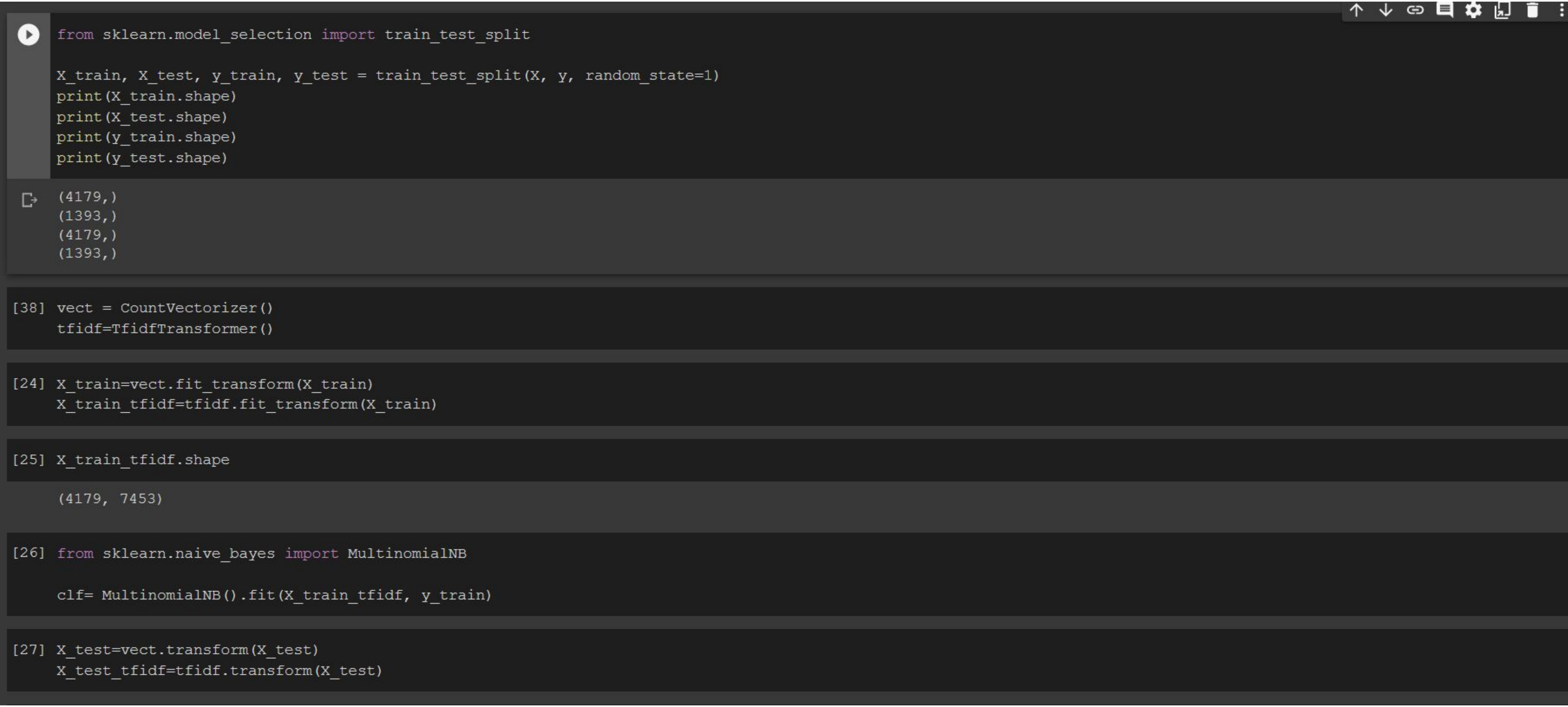
```
[35] df['label'] = df.Category.map({'ham':0, 'spam':1})
```

```
[36] df.head()
```

	Category	Message	label
0	ham	Go until jurong point, crazy.. Available only ...	0
1	ham	Ok lar... Joking wif u oni...	0
2	spam	Free entry in 2 a wkly comp to win FA Cup fina...	1
3	ham	U dun say so early hor... U c already then say...	0
4	ham	Nah I don't think he goes to usf, he lives aro...	0

```
[21] X = df.Message
     y = df.label
     print(X.shape)
     print(y.shape)
```

```
(5572,)
```



```
from sklearn.model_selection import train_test_split

X_train, X_test, y_train, y_test = train_test_split(X, y, random_state=1)
print(X_train.shape)
print(X_test.shape)
print(y_train.shape)
print(y_test.shape)
```

```
(4179,)
(1393,)
(4179,)
(1393,)
```

```
[38] vect = CountVectorizer()
     tfidf=TfidfTransformer()
```

```
[24] X_train=vect.fit_transform(X_train)
     X_train_tfidf=tfidf.fit_transform(X_train)
```

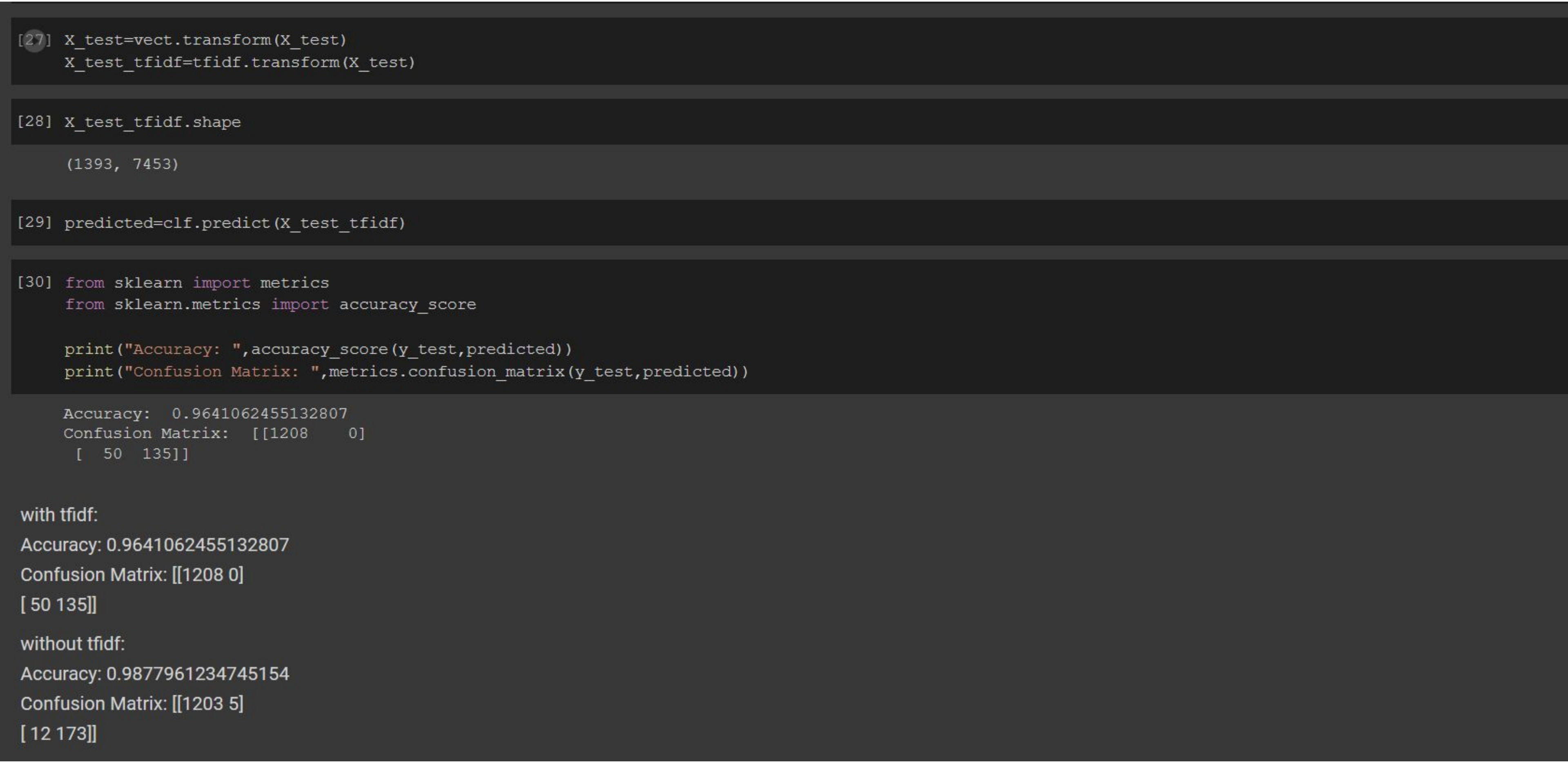
```
[25] X_train_tfidf.shape
```

```
(4179, 7453)
```

```
[26] from sklearn.naive_bayes import MultinomialNB

     clf= MultinomialNB().fit(X_train_tfidf, y_train)
```

```
[27] X_test=vect.transform(X_test)
     X_test_tfidf=tfidf.transform(X_test)
```



```
[27] X_test=vect.transform(X_test)
     X_test_tfidf=tfidf.transform(X_test)
```

```
[28] X_test_tfidf.shape
```

```
(1393, 7453)
```

```
[29] predicted=clf.predict(X_test_tfidf)
```

```
[30] from sklearn import metrics
     from sklearn.metrics import accuracy_score

     print("Accuracy: ",accuracy_score(y_test,predicted))
     print("Confusion Matrix: ",metrics.confusion_matrix(y_test,predicted))
```

```
Accuracy:  0.9641062455132807
Confusion Matrix:  [[1208    0]
 [  50  135]]
```

with tfidf:

Accuracy: 0.9641062455132807

Confusion Matrix: [[1208 0]

[50 135]]

without tfidf:

Accuracy: 0.9877961234745154

Confusion Matrix: [[1203 5]

[12 173]]