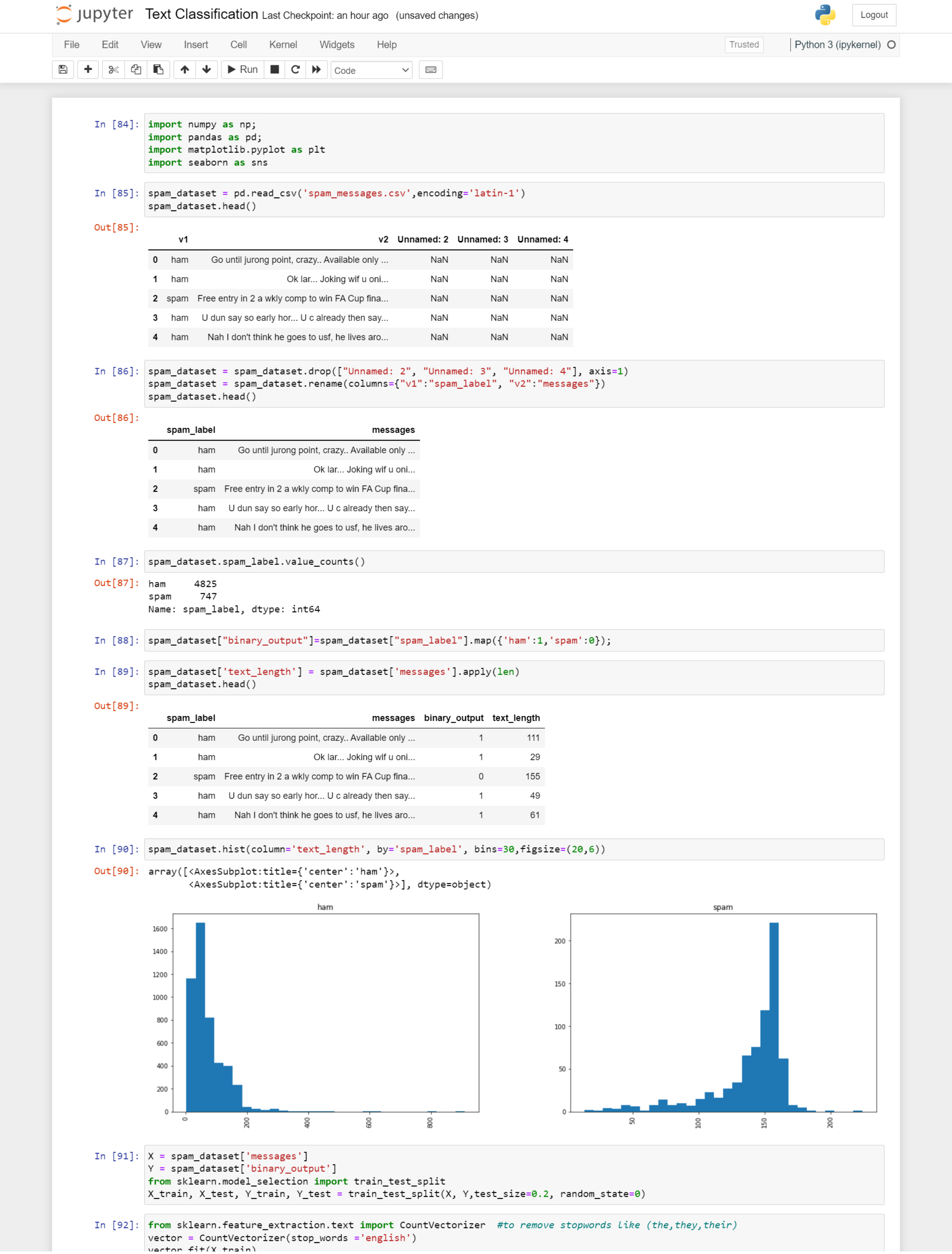


In [84]:
```python
import numpy as np;
import pandas as pd;
import matplotlib.pyplot as plt
import seaborn as sns
```

In [85]:
```python
spam_dataset = pd.read_csv('spam_messages.csv',encoding='latin-1')
spam_dataset.head()
```

Out[85]:

	v1	v2	Unnamed: 2	Unnamed: 3	Unnamed: 4
0	ham	Go until jurong point, crazy.. Available only ...	NaN	NaN	NaN
1	ham	Ok lar... Joking wif u oni...	NaN	NaN	NaN
2	spam	Free entry in 2 a wkly comp to win FA Cup fina...	NaN	NaN	NaN
3	ham	U dun say so early hor... U c already then say...	NaN	NaN	NaN
4	ham	Nah I don't think he goes to usf, he lives aro...	NaN	NaN	NaN

In [86]:
```python
spam_dataset = spam_dataset.drop(["Unnamed: 2", "Unnamed: 3", "Unnamed: 4"], axis=1)
spam_dataset = spam_dataset.rename(columns={"v1":"spam_label", "v2":"messages"})
spam_dataset.head()
```

Out[86]:

	spam_label	messages
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...

In [87]:
```python
spam_dataset.spam_label.value_counts()
```

Out[87]:
```
ham     4825
spam     747
Name: spam_label, dtype: int64
```

In [88]:
```python
spam_dataset["binary_output"]=spam_dataset["spam_label"].map({'ham':1,'spam':0});
```

In [89]:
```python
spam_dataset['text_length'] = spam_dataset['messages'].apply(len)
spam_dataset.head()
```

Out[89]:

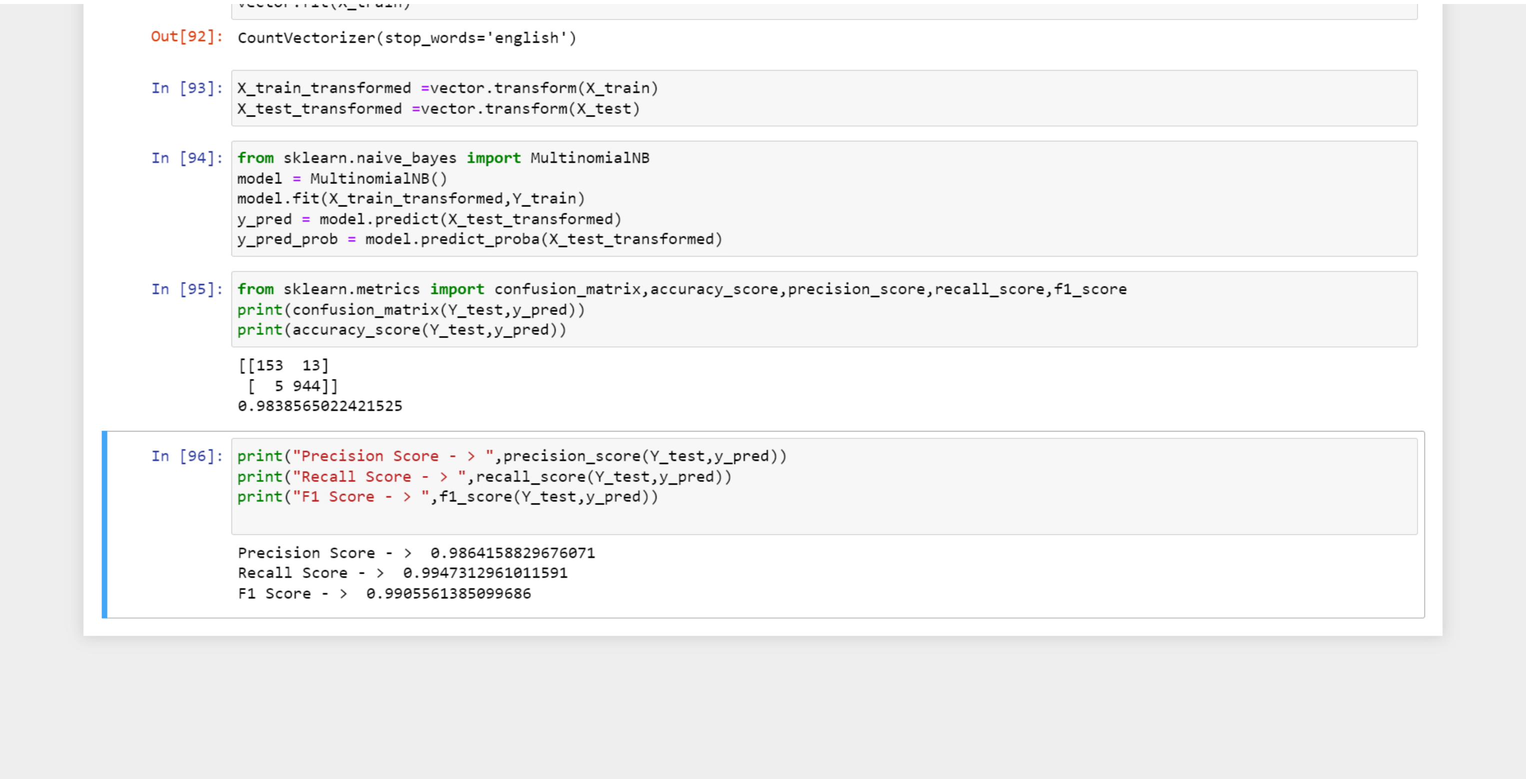
	spam_label	messages	binary_output	text_length
0	ham	Go until jurong point, crazy.. Available only ...	1	111
1	ham	Ok lar... Joking wif u oni...	1	29
2	spam	Free entry in 2 a wkly comp to win FA Cup fina...	0	155
3	ham	U dun say so early hor... U c already then say...	1	49
4	ham	Nah I don't think he goes to usf, he lives aro...	1	61

In [90]:
```python
spam_dataset.hist(column='text_length', by='spam_label', bins=30,figsize=(20,6))
```

Out[90]:
```
array([<AxesSubplot:title={'center':'ham'}>,
       <AxesSubplot:title={'center':'spam'}>], dtype=object)
```

In [91]:
```python
X = spam_dataset['messages']
Y = spam_dataset['binary_output']
from sklearn.model_selection import train_test_split
X_train, X_test, Y_train, Y_test = train_test_split(X, Y,test_size=0.2, random_state=0)
```

In [92]:
```python
from sklearn.feature_extraction.text import CountVectorizer  #to remove stopwords like (the,they,their)
vector = CountVectorizer(stop_words ='english')
vector.fit(X_train)
```



```
vector.fit(X_train)
```

Out[92]:
```
CountVectorizer(stop_words='english')
```

In [93]:
```
X_train_transformed =vector.transform(X_train)
X_test_transformed =vector.transform(X_test)
```

In [94]:
```
from sklearn.naive_bayes import MultinomialNB
model = MultinomialNB()
model.fit(X_train_transformed,Y_train)
y_pred = model.predict(X_test_transformed)
y_pred_prob = model.predict_proba(X_test_transformed)
```

In [95]:
```
from sklearn.metrics import confusion_matrix,accuracy_score,precision_score,recall_score,f1_score
print(confusion_matrix(Y_test,y_pred))
print(accuracy_score(Y_test,y_pred))
```

```
[[153  13]
 [  5 944]]
0.9838565022421525
```

In [96]:
```
print("Precision Score - > ",precision_score(Y_test,y_pred))
print("Recall Score - > ",recall_score(Y_test,y_pred))
print("F1 Score - > ",f1_score(Y_test,y_pred))
```

```
Precision Score - >  0.9864158829676071
Recall Score - >  0.9947312961011591
F1 Score - >  0.9905561385099686
```