In [1]:
```
import numpy as np
import pandas as pd

df = pd.read_csv('BBC News Train.csv')
df.head()
```

Out[1]:

	ArticleId	Text	Category
0	1833	worldcom ex-boss launches defence lawyers defe...	business
1	154	german business confidence slides german busin...	business
2	1101	bbc poll indicates economic gloom citizens in ...	business
3	1976	lifestyle governs mobile choice faster bett...	tech
4	917	enron bosses in $168m payout eighteen former e...	business

In [2]:
```
len(df)
```

Out[2]:
```
1490
```

In [3]:
```
df.isnull().sum()
```

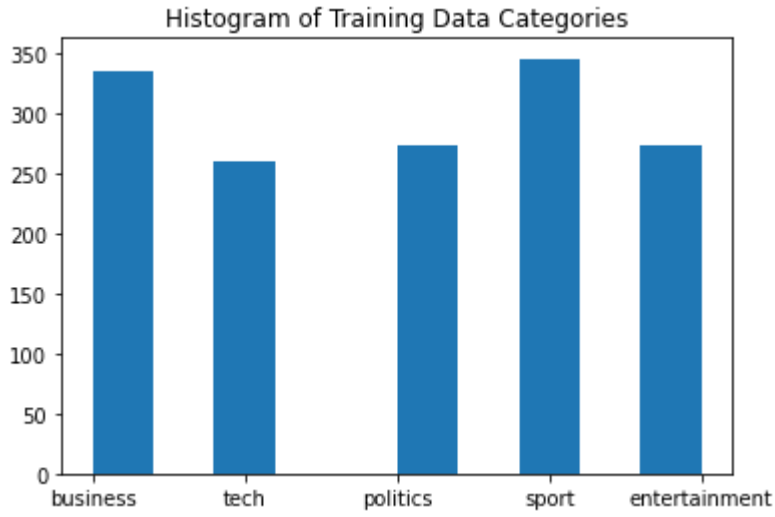
Out[3]:
```
ArticleId    0
Text         0
Category     0
dtype: int64
```

In [4]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1490 entries, 0 to 1489
Data columns (total 3 columns):
 #   Column     Non-Null Count  Dtype 
---  ------     --------------  ----- 
 0   ArticleId  1490 non-null   int64 
 1   Text       1490 non-null   object
 2   Category   1490 non-null   object
dtypes: int64(1), object(2)
memory usage: 35.0+ KB
```

In [26]:
```
import matplotlib.pyplot as plt

plt.hist(df['Category'])
plt.title("Histogram of Training Data Categories")
categories = df['Category'].unique()
```



In [27]:
```
from sklearn.model_selection import train_test_split
```

```
X = df['Text']
y = df['Category']

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.33, random_state=42)
```

In [28]:
```
X
```

Out[28]:
```
0       worldcom ex-boss launches defence lawyers defe...
1       german business confidence slides german busin...
2       bbc poll indicates economic gloom citizens in ...
3       lifestyle  governs mobile choice  faster  bett...
4       enron bosses in $168m payout eighteen former e...
                              ...
1485    double eviction from big brother model caprice...
1486    dj double act revamp chart show dj duo jk and ...
1487    weak dollar hits reuters revenues at media gro...
1488    apple ipod family expands market apple has exp...
1489    santy worm makes unwelcome visit thousands of ...
Name: Text, Length: 1490, dtype: object
```

In [29]:
```
y
```

Out[29]:
```
0            business
1            business
2            business
3                tech
4            business
              ...
1485    entertainment
1486    entertainment
1487         business
1488             tech
1489             tech
Name: Category, Length: 1490, dtype: object
```

In [30]:
```
from sklearn.pipeline import Pipeline
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.naive_bayes import MultinomialNB
from sklearn.svm import LinearSVC

# Naïve Bayes:
text_clf_nb = Pipeline([('tfidf', TfidfVectorizer()),
                     ('clf', MultinomialNB()),
])

# Linear SVC:
text_clf_lsvc = Pipeline([('tfidf', TfidfVectorizer()),
                     ('clf', LinearSVC()),
])
```

In [31]:
```
text_clf_nb.fit(X_train, y_train)
```

Out[31]:
```
Pipeline(steps=[('tfidf', TfidfVectorizer()), ('clf', MultinomialNB())])
```

In [32]:
```
predictions = text_clf_nb.predict(X_test)
```

In [33]:
```
from sklearn import metrics
print(metrics.confusion_matrix(y_test,predictions))
```

```
[[113   0   2   0   2]
 [  2  78   1   7   0]
 [  2   0  90   1   0]
 [  0   0   0 110   0]
 [  1   0   3   1  79]]
```

In [34]:
```
print(metrics.classification_report(y_test,predictions))
```

```
                precision    recall  f1-score   support

      business       0.96      0.97      0.96       117
entertainment       1.00      0.89      0.94        88
      politics       0.94      0.97      0.95        93
         sport       0.92      1.00      0.96       110
          tech       0.98      0.94      0.96        84

      accuracy                           0.96       492
     macro avg       0.96      0.95      0.95       492
  weighted avg       0.96      0.96      0.96       492
```

In [35]:
```
print(metrics.accuracy_score(y_test,predictions))
```

```
0.9552845528455285
```

In [36]:
```
text_clf_lsvc.fit(X_train, y_train)
```

Out[36]:
```
Pipeline(steps=[('tfidf', TfidfVectorizer()), ('clf', LinearSVC())])
```

In [37]:
```
predictions = text_clf_lsvc.predict(X_test)
```

In [38]:
```
from sklearn import metrics
print(metrics.confusion_matrix(y_test,predictions))
```

```
[[114   0   2   0   1]
 [  1  87   0   0   0]
 [  2   1  88   0   2]
 [  0   0   0 110   0]
 [  1   1   0   0  82]]
```

In [39]:
```
print(metrics.classification_report(y_test,predictions))
```

```
                precision    recall  f1-score   support

      business       0.97      0.97      0.97       117
entertainment       0.98      0.99      0.98        88
      politics       0.98      0.95      0.96        93
         sport       1.00      1.00      1.00       110
          tech       0.96      0.98      0.97        84

      accuracy                           0.98       492
     macro avg       0.98      0.98      0.98       492
  weighted avg       0.98      0.98      0.98       492
```

In [40]:
```
print(metrics.accuracy_score(y_test,predictions))
```

```
0.9776422764227642
```

In [ ]: