

Financial Market News Sentiment Analysis

▼ Import Library

```
import pandas as pd  
import numpy as np
```

▼ Import Dataset

```
df = pd.read_csv(r'https://raw.githubusercontent.com/YBI-Foundation/Dataset/main/Financial%20  
  
df.head()
```

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 4101 entries, 0 to 4100
Data columns (total 27 columns):
#   Column      Non-Null Count  Dtype
---  -
0   Date        4101 non-null   object
1   Label       4101 non-null   int64
2   News 1      4101 non-null   object
3   News 2      4101 non-null   object
4   News 3      4101 non-null   object
5   News 4      4101 non-null   object
6   News 5      4101 non-null   object
7   News 6      4101 non-null   object
8   News 7      4101 non-null   object
9   News 8      4101 non-null   object
10  News 9      4101 non-null   object
11  News 10     4101 non-null   object
12  News 11     4101 non-null   object
13  News 12     4101 non-null   object
14  News 13     4101 non-null   object
15  News 14     4101 non-null   object
16  News 15     4101 non-null   object
17  News 16     4101 non-null   object
18  News 17     4101 non-null   object
19  News 18     4101 non-null   object
20  News 19     4101 non-null   object
21  News 20     4101 non-null   object
22  News 21     4101 non-null   object
23  News 22     4101 non-null   object
24  News 23     4100 non-null   object
25  News 24     4098 non-null   object
26  News 25     4098 non-null   object
dtypes: int64(1), object(26)
memory usage: 865.2+ KB
```

```
df.shape
```

```
(4101, 27)
```

```
df.columns
```

```
Index(['Date', 'Label', 'News 1', 'News 2', 'News 3', 'News 4', 'News 5',  
      'News 6', 'News 7', 'News 8', 'News 9', 'News 10', 'News 11', 'News 12',  
      'News 13', 'News 14', 'News 15', 'News 16', 'News 17', 'News 18',  
      'News 19', 'News 20', 'News 21', 'News 22', 'News 23', 'News 24',  
      'News 25'],  
      dtype='object')
```

▼ Get Feature Selection

```
' '.join(str(x) for x in df.iloc[1,2:27])
```

```
df.index
```

```
RangeIndex(start=0, stop=4101, step=1)
```

```
len(df.index)
```

```
4101
```

```
news = []  
for row in range(0,len(df.index)):  
    news.append(' '.join(str(x) for x in df.iloc[row,2:27]))
```

```
type(news)
```

```
list
```

```
news[0]
```

```
X = news
```

```
type(X)
```

```
list
```

▼ Get Feature text conversion to bag of words

```
from sklearn.feature_extraction.text import CountVectorizer
```

```
cv = CountVectorizer(lowercase=True,ngram_range=(1,1))
```

```
X = cv.fit_transform(X)
```

```
X.shape
```

```
(4101, 48527)
```

```
y = df['Label']
```

```
y.shape
```

```
(4101,)
```

```
y
```

```
0      0
1      0
2      0
3      1
4      1
```

```
..
4096    0
4097    1
4098    1
4099    1
4100    1
```

```
Name: Label, Length: 4101, dtype: int64
```

▼ Get train test split

```
from sklearn.model_selection import train_test_split
```

```
X_train,X_test,y_train,y_test = train_test_split(X,y,test_size=0.3,stratify =y,random_state=2
```

```
X_train.shape,X_test.shape,y_train.shape,y_test.shape
```

```
((2870, 48527), (1231, 48527), (2870,), (1231,))
```

▼ Random Forest Classifier

```
from sklearn.ensemble import RandomForestClassifier
```

```
rf = RandomForestClassifier(n_estimators=200)
```

```
rf.fit(X_train,y_train)
```

```
RandomForestClassifier(n_estimators=200)
```

```
y_pred = rf.predict(X_test)
```

```
from sklearn.metrics import classification_report,confusion_matrix,accuracy_score
```

```
confusion_matrix(y_test,y_pred)
```

```
array([[151, 430],
       [174, 476]])
```

```
print(classification_report(y_test,y_pred))
```

	precision	recall	f1-score	support
0	0.46	0.26	0.33	581
1	0.53	0.73	0.61	650
accuracy			0.51	1231
macro avg	0.50	0.50	0.47	1231
weighted avg	0.50	0.51	0.48	1231

✓ 0s completed at 2:09 PM

● ✕