

## ▼ Financial Market News-Sentiment Analysis

This is a data (dummy) of Financial Market Top 25 News for the Day and Task is to Train and Predict Model for Overall Sentiment Analysis

## ▼ Import Library

```
import pandas as pd
```

```
import numpy as np
```

## ▼ Import DataSet

```
df= pd.read_csv('https://github.com/YBIFoundation/Dataset/raw/main/Financial%20Market%20Ne
```

```
df.head()
```

	Date	Label	News 1	News 2	News 3	N
0	01-01-2010	0	McIlroy's men catch cold from Gudjonsson	Obituary: Brian Walsh	Workplace blues leave employers in the red	CI
1	02-01-2010	0	Warning from history points to crash	Investors flee to dollar haven	Banks and tobacco in favour	F F

```
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.describe()
```

	Label
count	4101.000000
mean	0.528164
std	0.499267
min	0.000000
25%	0.000000
50%	1.000000
75%	1.000000
max	1.000000

```
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

```
s flee to dollar haven Banks and tobacco in fa  
news = []  
for rows in range(0, len(df.index)):  
    news.append(' '.join(str(x) for x in df.iloc[1,2:27]))  
    view: The Turk in Italy Deutsche spells out its  
type(news)  
  
list  
  
hit rock bottom Wilkinson out of his denth Kin  
news[0]
```

```
'Warning from history points to crash Investor  
s flee to dollar haven Banks and tobacco in fa  
vour Review: Llama Farmers War jitters lead to  
sell-off Your not-so-secret history Review: Th  
e Northern Sinfonia Review: Hysteria Review: T  
he Guardsman Opera: The Marriage of Figaro Rev  
iew: The Turk in Italy Deutsche spells out its  
X = news  
oil prices skyward TV sport chief leaves home  
type(X)
```

```
list  
s Scots rebound Battling Wales cling to lifell
```

## ▼ 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, 107)
```

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

```
y.shape
```

```
(4101,)
```

## ▼ 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 , random_state=2529, test_size=0.3, s
```

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

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

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

```
▼      RandomForestClassifier
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([[ 0, 581],
       [ 0, 650]])
```

```
accuracy_score(y_test,y_pred)
```

```
0.5280259951259139
```

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

	precision	recall	f1-score	support
0	0.00	0.00	0.00	581
1	0.53	1.00	0.69	650
accuracy			0.53	1231
macro avg	0.26	0.50	0.35	1231
weighted avg	0.28	0.53	0.36	1231

```
/usr/local/lib/python3.10/dist-packages/sklearn/metrics/_classification.py:1344
_warn_prf(average, modifier, msg_start, len(result))
/usr/local/lib/python3.10/dist-packages/sklearn/metrics/_classification.py:1344
_warn_prf(average, modifier, msg_start, len(result))
```

```
/usr/local/lib/python3.10/dist-packages/sklearn/metrics/_classification.py:1344  
_warn_prf(average, modifier, msg_start, len(result))
```