Outstanding Project-3

By:
Name: Jatin Jindal
E-Mail Id: jatinjindal1199@gmail.com

1. Scraping News Headlines

The Hindu

```
[78]  1 page=20
      2 for i in range(1,page+1):
      3     count=50
      4     ch='https://www.thehindu.com/archive/web/2020/9/'+str(i)+'/'
      5     res=requests.get(ch)
      6     soup=bs4.BeautifulSoup(res.text,'lxml')
      7     for j in soup.select('.archive-list'):
      8         if count==0:
      9             break
     10         else:
     11             #print('2020/09/'+str(i))
     12             headline['Date'].append('2020/09/'+str(i))
     13             headline['News_Channel'].append('The Hindu')
     14             #print(j.text)
     15             news=j.text.replace('\n',"")
     16             headline['Headline'].append(news)
     17             count=count-1
     18 print(headline)
     19 len(headline['Headline'])

      {'Date': ['2020/09/1', '2020/09/1', '2020/09/1', '2020/09/1', '2020/09/1
      967
```

DD News

```
[79]  1 page=20
      2 for i in range(1,page+1):
      3     count=50
      4     ch='http://ddnews.gov.in/about/news-archive?page='+str(i)
      5     res=requests.get(ch)
      6     soup=bs4.BeautifulSoup(res.text,'lxml')
      7     for j in soup.select('.archive-title'):
      8         if count==0:
      9             break
     10         else:
     11             #print('2020/09/'+str(i))
     12             headline['Date'].append('2020/09/'+str(i))
     13             headline['News_Channel'].append('DD News')
     14             #print(j.text)
     15             news=j.text.replace('\n',"")
     16             headline['Headline'].append(news)
     17             count=count-1
     18 print(headline)
     19 len(headline['Headline'])

      {'Date': ['2020/09/1', '2020/09/1', '2020/09/1', '2020/09/1', '2020
      1567
```

Indian Express

```
page=20
for i in range(1,page+1):
    count=50
    ch='https://indianexpress.com/section/news-archive/page/'+str(i)+'/'
    res=requests.get(ch)
    soup=bs4.BeautifulSoup(res.text,'lxml')
    for j in soup.select('.articles'):
        if count==0:
            break
        else:
            #print('2020/09/'+str(i))
            headline['Date'].append('2020/09/'+str(i))
            headline['News_Channel'].append('Indian Express')
            #print(j.text)
            news=j.text.replace('\n',"")
            headline['Headline'].append(news)
            count=count-1
print(headline)
len(headline['Headline'])
```

```
{'Date': ['2020/09/1', '2020/09/1', '2020/09/1', '2020/09/1', '2020/09/1', '2
2067
```

2. Finding Positive and Negative Sentiments

```
[81]
headline['Negative']=[]
headline['Positive']=[]
headline['Neutral']=[]
headline['Total']=[]
headline['Sentiment']=[]
k=0
from nltk.sentiment.vader import SentimentIntensityAnalyzer
sia = SentimentIntensityAnalyzer()
for line in headline['Headline']:
    pol_score=sia.polarity_scores(line)
    headline['Negative'].append(pol_score['neg'])
    headline['Positive'].append(pol_score['pos'])
    headline['Neutral'].append(pol_score['neu'])
    headline['Total'].append(pol_score['compound'])
    if pol_score['compound']> 0.2:
        headline['Sentiment'].append(1)
    elif pol_score['compound']> -0.2:
        headline['Sentiment'].append(-1)
    else:
        headline['Sentiment'].append(0)
df=pd.DataFrame(headline)
df
```

	Date	News_Channel	Headline	Negative	Positive	Neutral	Total	Sentiment
0	2020/09/1	The Hindu	US Open | Fluent start for Djokovic; Osaka str...	0.000	0.000	1.000	0.0000	-1
1	2020/09/1	The Hindu	Facebook wants you to experience its virtual u...	0.043	0.048	0.909	-0.1280	-1
2	2020/09/1	The Hindu	Top news of the day: Former President Pranab M...	0.042	0.261	0.697	0.8020	1
3	2020/09/1	The Hindu	Indian Americans should get involved in U.S. P...	0.128	0.151	0.721	0.3818	1

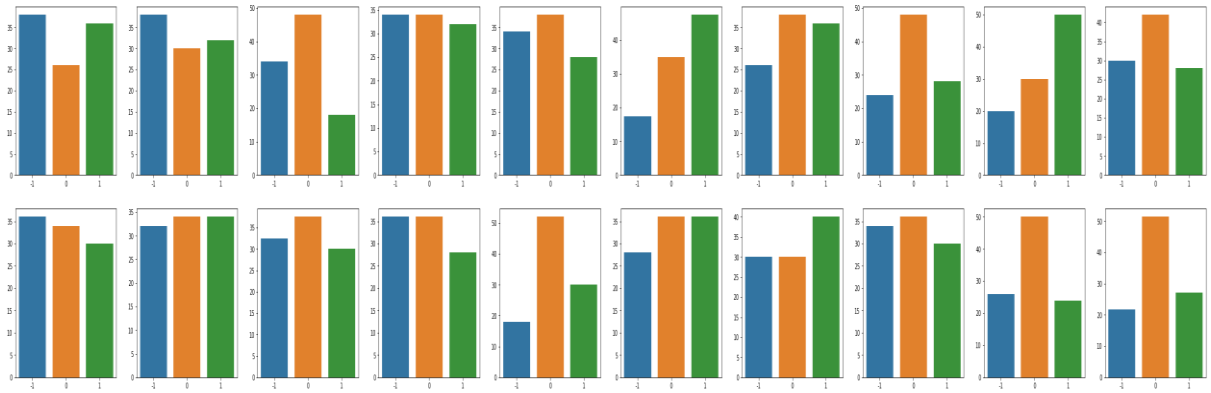
If the overall is less between -0.2 to 0.2 then it is treated as neutral
If the overall is less than -0.2 then it is treated as negative
If the overall is more than 0.2 then it is treated as Positive

3. Date Wise Sentiment%

-1:Negative(Blue)
1: Positive(Orange)
0:Neutral(Green)

The Hindu

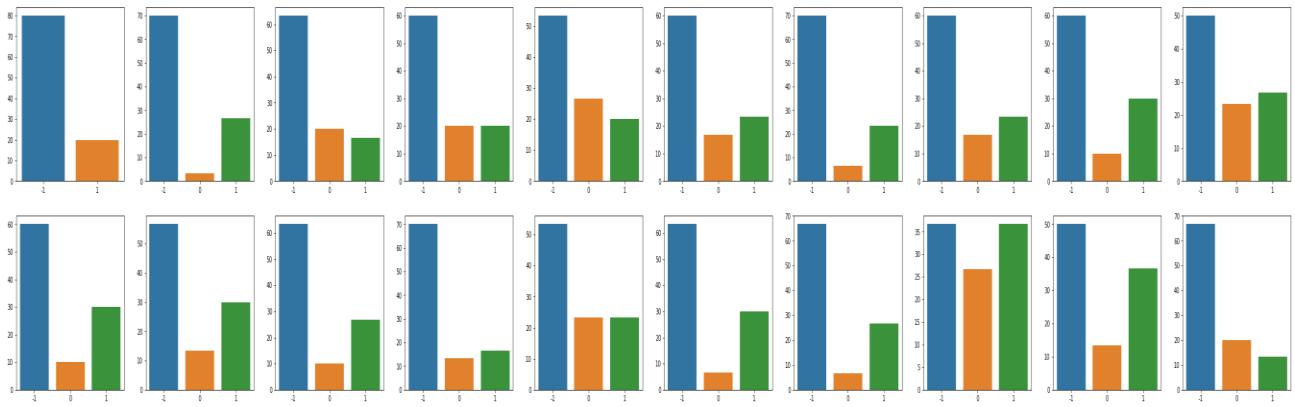
```
[120] 1 print('THe Hindu')
      2 plt.figure(figsize=(50,50))
      3 for i in range(1,page+1):
      4     plt.subplot(10,10,i)
      5     df1=df[df['Date']=='2020/09/'+str(i)]
      6     percent=df1[df1['News_Channel']=='The Hindu' ].Sentiment.value_counts()/df1[df1['News_Channel']=='The Hindu' ].Sentiment.value_counts().sum()*100
      7     sns.barplot(x=percent.index,y=percent.values)
```



We can easily see that the there is mostly a good mixture all type sentiments news on each day except on some days.

DD News

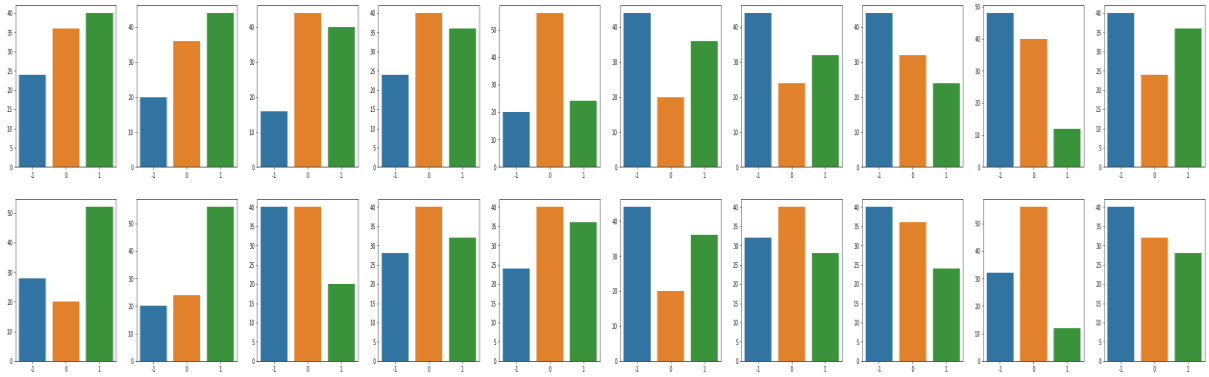
```
1 print('DD News')
2 plt.figure(figsize=(50,50))
3 for i in range(1,page+1):
4     plt.subplot(10,10,i)
5     df1=df[df['Date']=='2020/09/'+str(i)]
6     percent=df1[df1['News_Channel']=='DD News' ].Sentiment.value_counts()/df1[df1['News_Channel']=='DD News' ].Sentiment.value_counts().sum()*100
7     sns.barplot(x=percent.index,y=percent.values)
```



We can see that the Most Percent of their news are showing negative sentiment.

Indian Express

```
print('Indian Express')
plt.figure(figsize=(50,50))
for i in range(1,page+1):
    plt.subplot(10,10,i)
    df1=df[df['Date']=='2020/09/'+str(i)]
    percent=df1[df1['News_Channel']=='Indian Express' ].Sentiment.value_counts()/df1[df1['News_Channel']=='Indian Express' ].Sentiment.value_counts().sum()*100
    sns.barplot(x=percent.index,y=percent.values)
```



It can be seen as some days there it has high percent of negative news and some days it has high negative sentiment and some days high positive sentiment.

4. Cleaning News Headlines

```
import re
nltk.download('stopwords')
from nltk.corpus import stopwords
from nltk.stem.porter import PorterStemmer
clean_headlines=[]
for i in range(0,2067):
    head=re.sub('[^a-zA-Z]',' ',df['Headline'][i]) # to include only english words
    head=head.lower()
    head=head.split()  # to get each word seperated from each other
    ps=PorterStemmer()
    all_stopwords=stopwords.words('english')  # Used to get all stopwords in english language
    all_stopwords.remove('not')
    head=[ps.stem(word) for word in head if not word in set(all_stopwords)]
    # Removing the stopwords and at the same time stem the words
    head=' '.join(head)
    clean_headlines.append(head)
```

5. Creating Bag of words

```
from sklearn.feature_extraction.text import CountVectorizer
cv=CountVectorizer()
x=cv.fit_transform(df['Clean Headline'])
```

6. Splitting

```
from sklearn.model_selection import train_test_split
x_train,x_test,y_train,y_test=train_test_split(x,df['Sentiment'],random_state=0)
print(x_train.shape)
print(y_train.shape)
print(x_test.shape)
print(y_test.shape)
```

```
(1550, 12752)
(1550,)
(517, 12752)
(517,)
```

7. Model Training

a)Naive Bayes

```
[29] 1 from sklearn.naive_bayes import MultinomialNB
     2 nb=MultinomialNB()
     3 nb.fit(x_train,y_train)
     4 nb.score(x_test,y_test)

     0.5087040618955513
```

```
1 from sklearn.model_selection import cross_val_score
2 accuracy_nb=cross_val_score(estimator=nb,X=x_train,y=y_train,cv=10)
3 print('Accuracy= {:f} and Standard Deviation= {:f}'.format(accuracy_nb.mean()*100,accuracy_nb.std()*100))
4 accuracy_nb

Accuracy= 55.161290 and Standard Deviation= 3.675155
array([0.63870968, 0.59354839, 0.52258065, 0.52903226, 0.53548387,
       0.55483871, 0.52903226, 0.52903226, 0.56774194, 0.51612903])
```

b) K Nearest Neighbors

```
1 from sklearn.neighbors import KNeighborsClassifier
2 kn=KNeighborsClassifier(n_neighbors=5,metric='minkowski',p=2)
3 kn.fit(x_train,y_train)
4 kn.score(x_test,y_test)

0.38684719535783363
```

```
1 from sklearn.model_selection import cross_val_score
2 accuracy_kn=cross_val_score(estimator=kn,X=x_train,y=y_train,cv=10)
3 print('Accuracy= {:f} and Standard Deviation= {:f}'.format(accuracy_kn.mean()*100,accuracy_kn.std()*100))
4 accuracy_kn

Accuracy= 41.290323 and Standard Deviation= 0.816072
array([0.41290323, 0.40645161, 0.41290323, 0.41290323, 0.4       ,
       0.41935484, 0.40645161, 0.42580645, 0.42580645, 0.40645161])
```

C) Random Forest

```
from sklearn.ensemble import RandomForestClassifier
forest=RandomForestClassifier(n_estimators=10,criterion='entropy',random_state=0)
forest.fit(x_train,y_train)
forest.score(x_test,y_test)
```

```
0.5764023210831721
```

```
from sklearn.model_selection import cross_val_score
accuracy_forest=cross_val_score(estimator=forest,X=x_train,y=y_train,cv=10)
print('Accuracy= {:f} and Standard Deviation= {:f}'.format(accuracy_forest.mean()*100,accuracy_forest.std()*100))
accuracy_forest
```

```
Accuracy= 61.419355 and Standard Deviation= 4.686199
array([0.64516129, 0.66451613, 0.60645161, 0.61935484, 0.64516129,
       0.59354839, 0.56774194, 0.56129032, 0.69677419, 0.54193548])
```

d) Decision Tree

```
from sklearn.tree import DecisionTreeClassifier
dtree=DecisionTreeClassifier(criterion='entropy',random_state=0)
dtree.fit(x_train,y_train)
dtree.score(x_test,y_test)
```

```
0.5880077369439072
```

```
from sklearn.model_selection import cross_val_score
accuracy_dtree=cross_val_score(estimator=dtree,X=x_train,y=y_train,cv=10)
print('Accuracy= {:f} and Standard Deviation= {:f}'.format(accuracy_dtree.mean()*100,accuracy_dtree.std()*100))
accuracy_dtree
```

```
Accuracy= 60.451613 and Standard Deviation= 3.054142
array([0.67741935, 0.58709677, 0.59354839, 0.6       , 0.58064516,
       0.55483871, 0.6       , 0.61935484, 0.61935484, 0.61290323])
```

Therefore the best model is Random Forest Classification as the accuracy of this one is highest.

8. Finding Sentiment for new news channel

a) Scrapping News

```
count=4
lst=[]
ch='https://timesofindia.indiatimes.com/home/headlines'
res=requests.get(ch)
soup=bs4.BeautifulSoup(res.text,'lxml')
for j in soup.select('.w_tle'):
    if count==0:
        break
    else:
        #print(j.text)
        news=j.text.replace('\n',"")
        lst.append(news)
        count=count-1
lst
```

```
["Live: Won't contest polls with JD(U) due to ideological reasons, says LJP",
 "Live: '500m vaccine doses for 25cr by July 2021'",
 "Live: Bhim Army chief meets Hathras victim's kin",
 'LAC row: Army, IAF prepare to fight wars jointly']
```

b) Cleaning News Headlines

```
for i in range(0,len(test_lst)):
    line=lst[i]
    head=re.sub('[^a-zA-Z]',' ',line) # to include only english words
    head=head.lower()
    head=head.split()  # to get each word seperated from each other
    ps=PorterStemmer()
    all_stopwords=stopwords.words('english')  # Used to get all stopwords in eng
    all_stopwords.remove('not')
    head=[ps.stem(word) for word in head if not word in set(all_stopwords)]
    # Removing the stopwords and at the same time stem the words
    head=' '.join(head)
    lst[i]=head
lst
```

```
['live contest poll jd u due ideolog reason say ljp',
 'live vaccin dose cr juli',
 'live bhim armi chief meet hathra victim kin',
 'lac row armi iaf prepar fight war jointli']
```

c) Making Bag of words and Predicting Sentiment