AIM

Natural Language Processing

- Part of Speech tagging
- N-gram and smoothening
- Chunking

Programming code:

```
import nltk
from nltk.corpus import stopwords
from nltk.tokenize import word_tokenize, sent_tokenize
nltk.download('stopwords')
nltk.download('punkt')
nltk.download('averaged_perceptron_tagg
er')
stop_words = set(stopwords.words('english'))
```

TOKENIZATION

```
#Dummy text
txt = "Hello. MCA S3 is fantastic. We learn many new concepts and implement
them in our practical exams. "\
"1st of all the data science is a new paper."
# sent_tokenize is one of instances of
# PunktSentenceTokenizer from the nltk.tokenize.punkt module
tokenized=sent_tok
enize
(txt)
for i in tokenized:
# Word tokenizers is used to find
the words # and punctuation in a
string
wordsList = nltk.word_tokenize(i)
# removing stop words from wordList
wordsList = [w for w in wordsList if not w in stop_words]
# Using a Tagger. Which is part-
of-speech # tagger or POS-tagger.
tagged=nltk.pos_tag(Wordslist)
print(tagged)
```

OUTPUT

```
[nltk_data] Downloading package stopwords to /root/nltk_data...
```

```
[nltk_data] Unzipping corpora/stopwords.zip.
[nltk_data] Downloading package punkt to /root/nltk_data...
[nltk_data]Unzipping tokenizers/punkt.zip.
[nltk_data] Downloading package averaged_perceptron_tagger to
[nltk_data]/root/nltk_data...
[nltk_data] Unzipping taggers/averaged_perceptron_tagger.zip.
 [('Hello', 'NNP'), ('.', '.')]
[('MCA', 'NNP'), ('S3', 'NNP'), ('fantastic', 'JJ'), ('.', '.')]
[('We', 'PRP'), ('learn', 'VBP'), ('many', 'JJ'), ('new', 'JJ'),
('concepts', 'NNS'), ('implement', 'JJ'), ('practical', 'JJ'),
('exams', 'NN'), ('.', '.')]
[('1st', 'CD'), ('data', 'NNS'), ('science', 'NN'), ('new', 'JJ'), ('paper', 'NN'), ('.', '.')]
```

SENTIMENTAL ANALYSIS

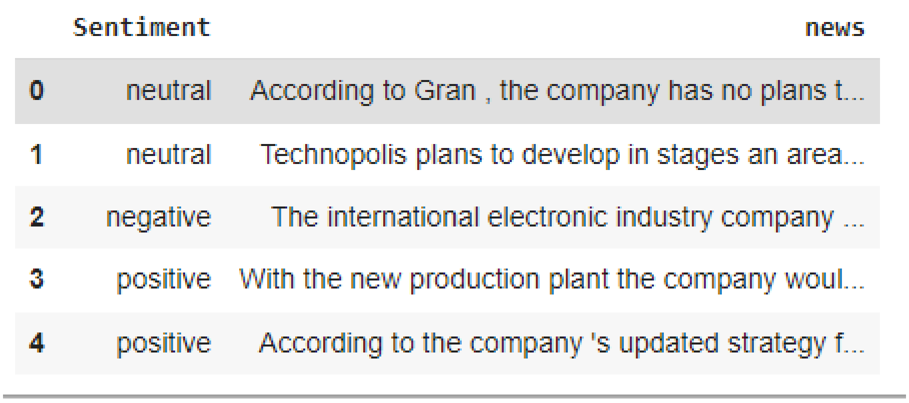
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
plt.style.use(style='seaborn')

#get the data from https://www.kaggle.com/ankurzing/sentiment-analysis-for-financial-news/version/5

colnames=['Sentiment', 'news']

df=pd.read_csv('all-data.csv',encoding = "ISO-8859-1", names=colnames, header = None)
df.head()
```

OUTPUT

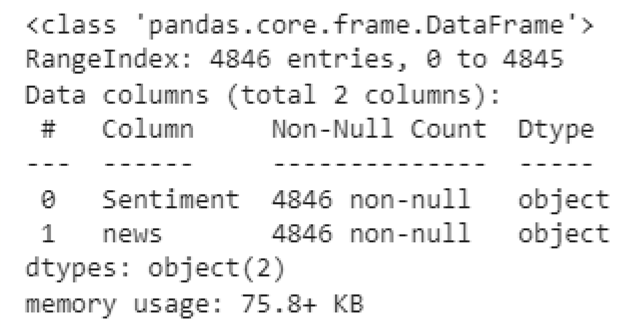


	Sentiment	news
0	neutral	According to Gran , the company has no plans t...
1	neutral	Technopolis plans to develop in stages an area...
2	negative	The international electronic industry company ...
3	positive	With the new production plant the company woul...
4	positive	According to the company 's updated strategy f...

Programming code:

```
df.info()
```

OUTPUT

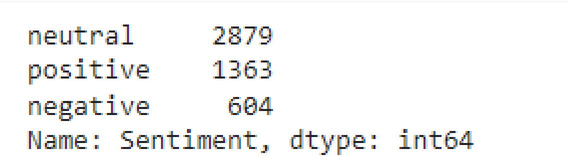


```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 4846 entries, 0 to 4845
Data columns (total 2 columns):
 #   Column     Non-Null Count  Dtype 
---  ------     --------------  ----- 
 0   Sentiment  4846 non-null   object
 1   news       4846 non-null   object
dtypes: object(2)
memory usage: 75.8+ KB
```

Programming code:

```
df['Sentiment'].value_counts()
```

OUTPUT

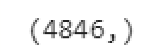


```
neutral     2879
positive    1363
negative     604
Name: Sentiment, dtype: int64
```

Programming code:

```
y=df['Sentiment'].values
y.shape
```

Output

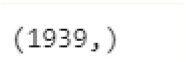


```
(4846,)
```

Programming code:

```
from sklearn.model_selection import train_test_split
(x_train,x_test,y_train,y_test)=train_test_split(x,y,test_size=0.4)
x_train.shape
y_train.shape
x_test.shape
y_test.shape
```

OUTPUT

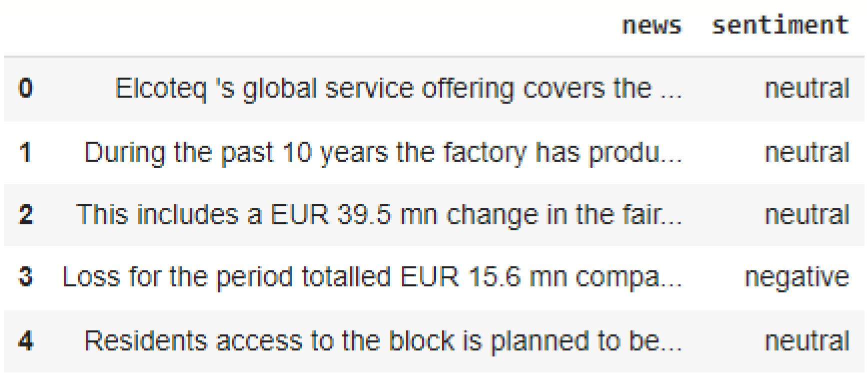


```
(1939,)
```

Programming code:

```
df1=pd.DataFrame(x_train)
df1=df1.rename(columns={0:'news'})
df2=pd.DataFrame(y_train)
df2=df2.rename(columns={0:'sentiment'})
df_train=pd.concat([df1,df2],axis=1)
df_train.head()
```

OUTPUT

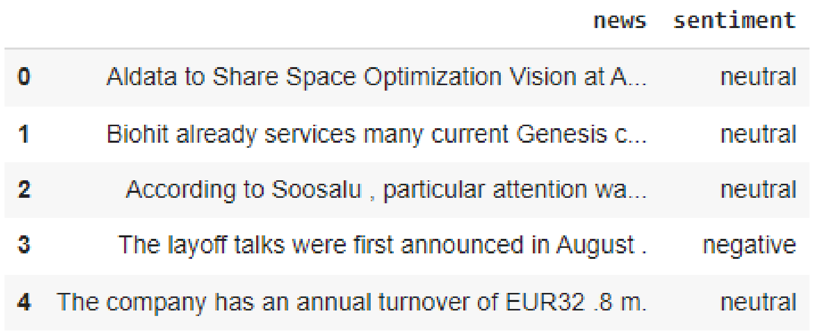


	news	sentiment
0	Elcoteq 's global service offering covers the ...	neutral
1	During the past 10 years the factory has produ...	neutral
2	This includes a EUR 39.5 mn change in the fair...	neutral
3	Loss for the period totalled EUR 15.6 mn compa...	negative
4	Residents access to the block is planned to be...	neutral

Programming code:

```
df3=pd.DataFrame(x_test)
df3=df3.rename(columns={0:'news'})
df4=pd.DataFrame(y_test)
df4=df2.rename(columns={0:'sentiment'})
df_test=pd.concat([df3,df4],axis=1)
df_test.head()
```

OUTPUT

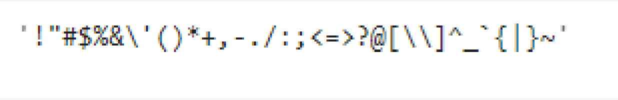


	news	sentiment
0	Aldata to Share Space Optimization Vision at A...	neutral
1	Biohit already services many current Genesis c...	neutral
2	According to Soosalu , particular attention wa...	neutral
3	The layoff talks were first announced in August .	negative
4	The company has an annual turnover of EUR32 .8 m.	neutral

Programming code:

```
#removing punctuations
#library that contains punctuation
import string
string.punctuation
```

OUTPUT



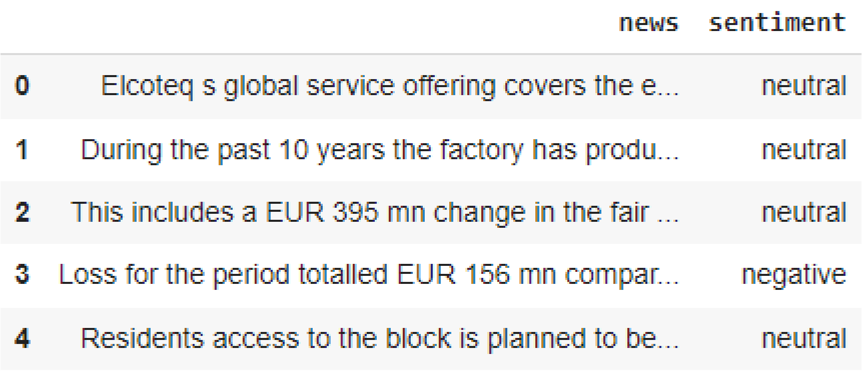
```
'!"#$%&\'()*+,-./:;<=>?@[\\]^_`{|}~'
```

Programming code:

```
#defining the function to remove punctuation
def remove_punctuation(text):
  if(type(text)==float):
    return text
  ans=""
  for i in text:
    if i not in string.punctuation:
      ans+=i
  return ans

#storing the puntuation free text in a new column called clean_msg
df_train['news']= df_train['news'].apply(lambda x:remove_punctuation(x))
df_test['news']= df_test['news'].apply(lambda x:remove_punctuation(x))
df_train.head()
#punctuations are removed from news column in train dataset
```

OUTPUT

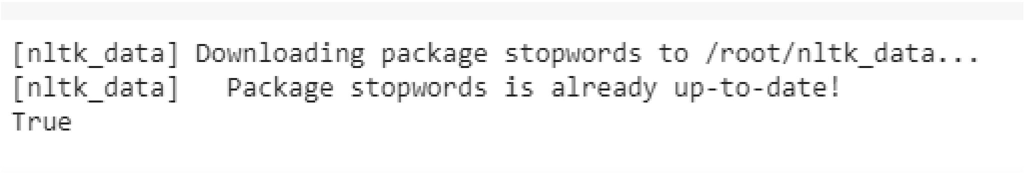


	news	sentiment
0	Elcoteq s global service offering covers the e...	neutral
1	During the past 10 years the factory has produ...	neutral
2	This includes a EUR 395 mn change in the fair ...	neutral
3	Loss for the period totalled EUR 156 mn compar...	negative
4	Residents access to the block is planned to be...	neutral

Programming code:

```
import nltk
from nltk.corpus import stopwords
nltk.download('stopwords')
```

OUTPUT



```
[nltk_data] Downloading package stopwords to /root/nltk_data...
[nltk_data]   Package stopwords is already up-to-date!
True
```

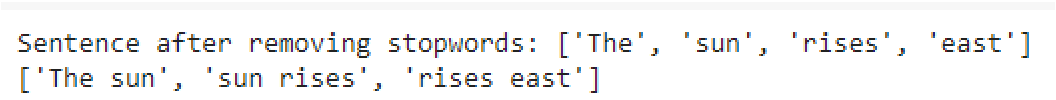
N-gram model

Programming code:

```
#method to generate n-grams:
#params:
#text-the text for which we have to generate n-grams
#ngram-number of grams to be generated from the text(1,2,3,4 etc., default value=1)
def generate_N_grams(text,ngram=1):
  words=[word for word in text.split(" ") if word not in set(stopwords.words('english'))
]
  print("Sentence after removing stopwords:",words)
  temp=zip(*[words[i:] for i in range(0,ngram)])
  ans=[' '.join(ngram) for ngram in temp]
   return ans


generate_N_grams("The sun rises in the east",2)
```

OUTPUT

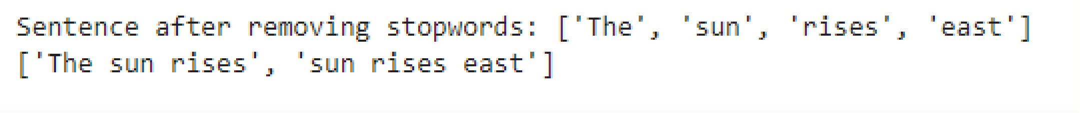


```
Sentence after removing stopwords: ['The', 'sun', 'rises', 'east']
['The sun', 'sun rises', 'rises east']
```

Programming code:

```
generate_N_grams("The sun rises in the east",3)
```

OUTPUT

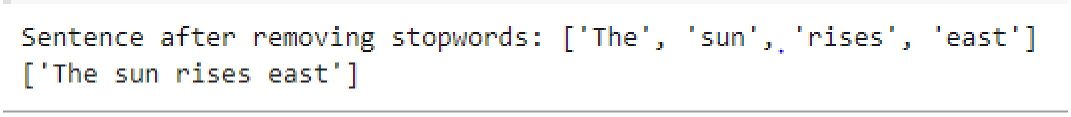


```
Sentence after removing stopwords: ['The', 'sun', 'rises', 'east']
['The sun rises', 'sun rises east']
```

Programming code:

```
generate_N_grams("The sun rises in the east",4)
```

OUTPUT



```
Sentence after removing stopwords: ['The', 'sun', 'rises', 'east']
['The sun rises east']
```