3. textPreProcessing

August 30, 2023

[ ]:
```
import nltk
import pandas as pd
import re
import os
from nltk.corpus import stopwords
from nltk.stem import WordNetLemmatizer
from nltk.tokenize import word_tokenize, sent_tokenize
from string import punctuation
import csv
```

[ ]:
```
# Download necessary datasets
nltk.download('punkt')
nltk.download('stopwords')
```

[ ]:
```
# Load CSV Data

# Stocks :- AAPL, MSFT, AMZN, NVDA, TSLA, GOOGL, UNH
# Sector Indices :- SSINFT (^SP500-45)

ticker = "SSINFT"

# Load the news artcile file
df = pd.read_csv(f"RawArticles/{ticker}_news_data.csv")
```

[ ]:
```
def preprocess_text(text):
    # 1. Convert to lower case
    text = str(text).lower()

    # 2. Remove hyperlinks
    text = re.sub(r'https?:\/\/\S+', '', text)

    # 3. Remove HTML tags
    text = re.sub(r'<.*?>', '', text)

    # 4. Remove special characters and symbols
    text = re.sub(r'[^a-z\s]', '', text)
```

```
    # 5. Tokenize the text
    tokens = word_tokenize(text)

    # 6. Remove stopwords
    stop_words = set(stopwords.words('english'))
    tokens = [token for token in tokens if token not in stop_words]

    # 7. Remove punctuations
    tokens = [token for token in tokens if token not in punctuation]

    # 8. Remove unnecessary spaces
    cleaned_text = ' '.join(tokens).strip()

    return cleaned_text
```

[ ]:
```
df['Headline'] = df['Headline'].apply(preprocess_text)
df['Summary'] = df['Summary'].apply(preprocess_text)
# df['Content'] = df['Content'].apply(preprocess_text)
```

[ ]:
```
df
```

[ ]:
```
directory = "PreProcessedArticles"

#If directory doesn't exist, create the directory
if not os.path.exists(directory):
    os.makedirs(directory)

filename = f"{directory}/{ticker}_news_data.csv"
```

[ ]:
```
df.to_csv(filename, index=False)
```