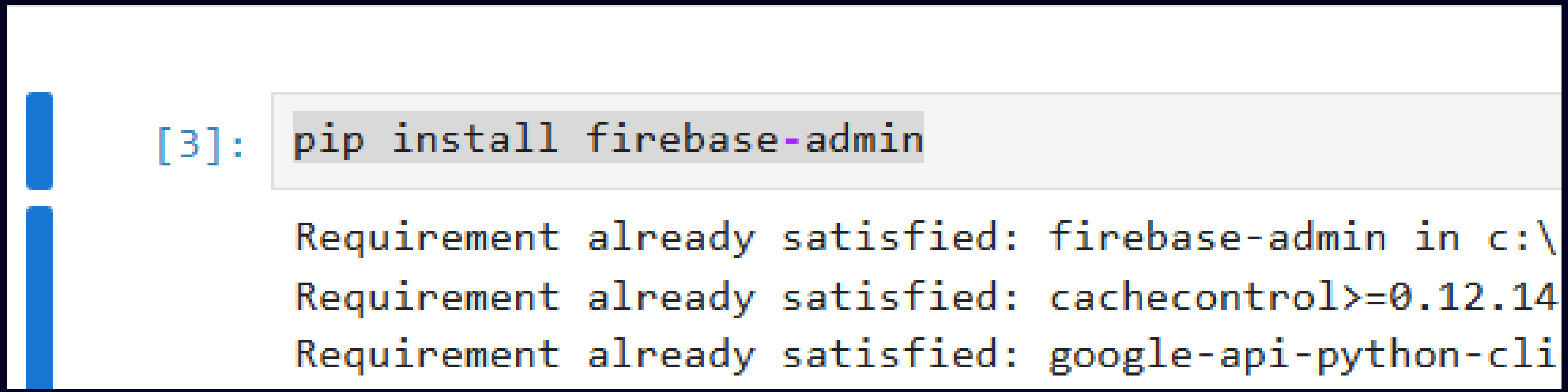
# Cloud Computing Project

# News Sentiment Analysis

Introduction : This project is the backbone of data-science as it has a mixture of both cloud integration aka (FireBase) and python code(api).

# Setting Up Your Cloud Environment with Python



```
[3]: pip install firebase-admin

     Requirement already satisfied: firebase-admin in c:\
     Requirement already satisfied: cachecontrol>=0.12.14
     Requirement already satisfied: google-api-python-cli
```

the main purpose of this project is Cloud Integration so i started with importing FireBase library ( and installing it )

# libraries Used :

1-Requests : Fetching articles using https requests( post ,gets etc..) .

2-JSON: Printing the articles Fetched in a nicely formatted manner

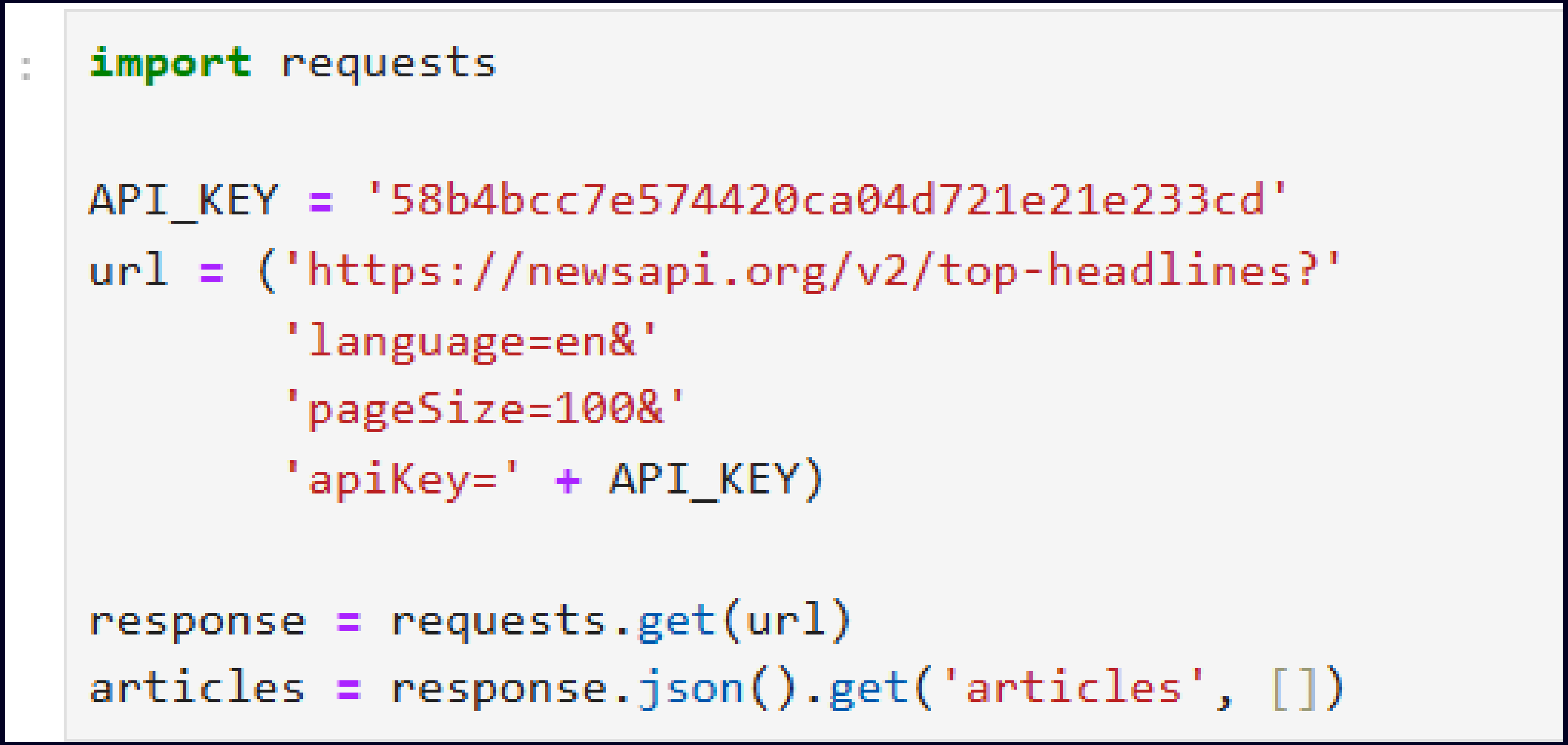
3-NLTK: Data Cleaning , Tokenizing ,Lemmatizing for sentiment analysis .

Pyspark( WAS NOT USED ) it demanded an old python so a better approach like NLTK was present.

4-Pandas(JSON to Dataframe)

5-Transformers: Using pre-existing sentiment analyzer

# Api Code Example:

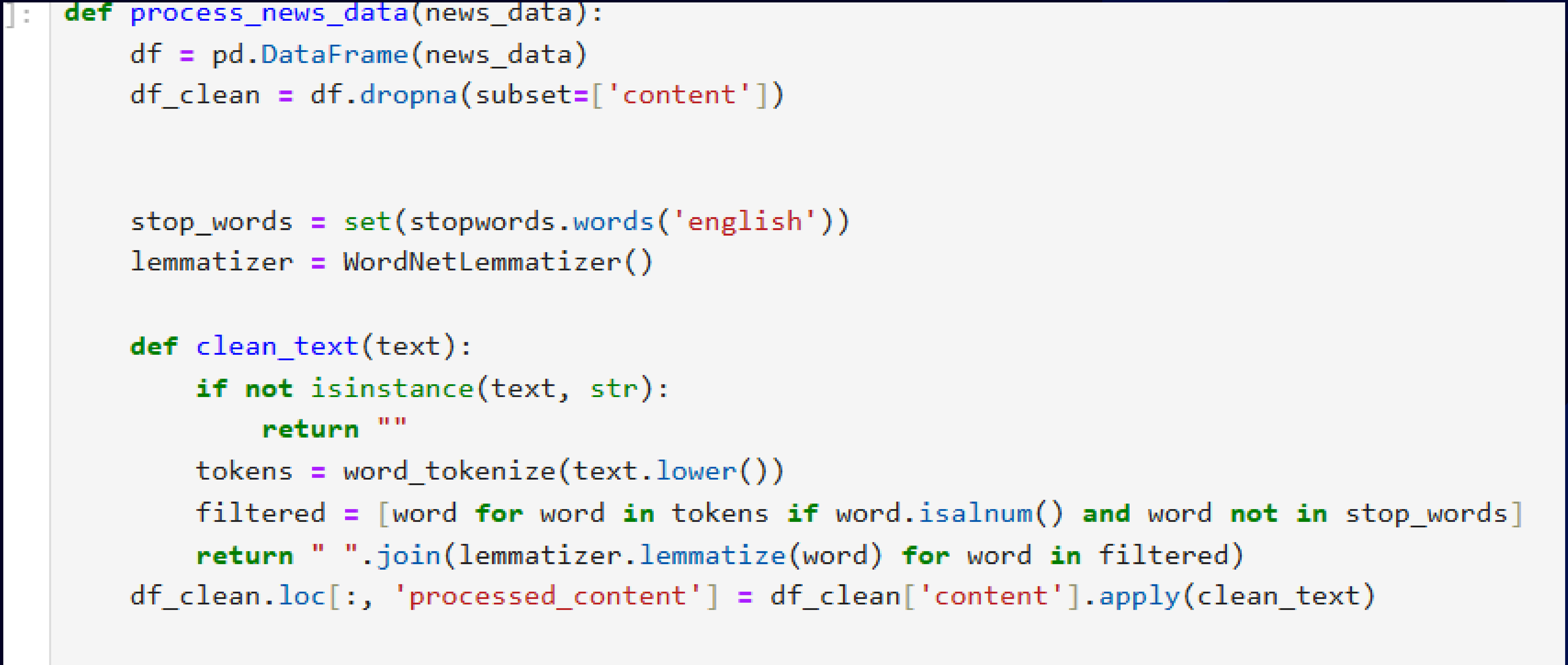


```
import requests

API_KEY = '58b4bcc7e574420ca04d721e21e233cd'
url = ('https://newsapi.org/v2/top-headlines?'
       'language=en&'
       'pageSize=100&'
       'apiKey=' + API_KEY)

response = requests.get(url)
articles = response.json().get('articles', [])
```

# Data Cleaning Section

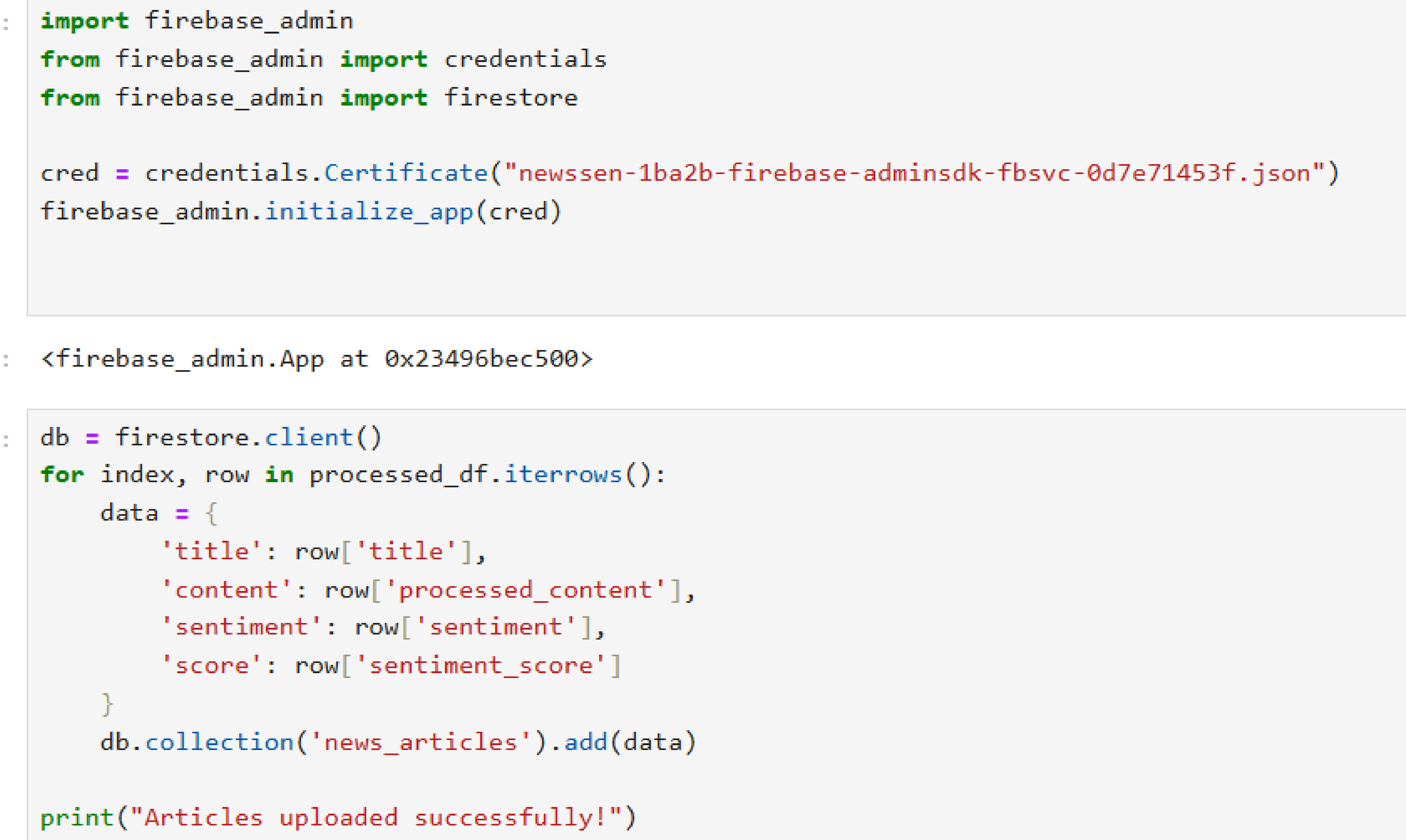


```python
def process_news_data(news_data):
    df = pd.DataFrame(news_data)
    df_clean = df.dropna(subset=['content'])


    stop_words = set(stopwords.words('english'))
    lemmatizer = WordNetLemmatizer()

    def clean_text(text):
        if not isinstance(text, str):
            return ""
        tokens = word_tokenize(text.lower())
        filtered = [word for word in tokens if word.isalnum() and word not in stop_words]
        return " ".join(lemmatizer.lemmatize(word) for word in filtered)
    df_clean.loc[:, 'processed_content'] = df_clean['content'].apply(clean_text)
```

# FireBase Integration Section:



```python
import firebase_admin
from firebase_admin import credentials
from firebase_admin import firestore

cred = credentials.Certificate("newssen-1ba2b-firebase-adminsdk-fbsvc-0d7e71453f.json")
firebase_admin.initialize_app(cred)
```

```
<firebase_admin.App at 0x23496bec500>
```

```python
db = firestore.client()
for index, row in processed_df.iterrows():
    data = {
        'title': row['title'],
        'content': row['processed_content'],
        'sentiment': row['sentiment'],
        'score': row['sentiment_score']
    }
    db.collection('news_articles').add(data)

print("Articles uploaded successfully!")
```

# Sentiment:

Many Models have been tried in the emoji library and in the transformers library the best fit one was the Uncased one for my news that i fetched. The sentiment results were less accurate when news sentiment analyzer was used.

# Goals Achieved :

1-Learned how to use apis to get web information.
2-The use of public data to benefit from it ( Analysis)
3-Using cloud services for the first time in a project such as firebase database
4-learned how to compare between multiple ready sentiment analyzers
5-the use of vast libraries to support many upcoming project

# Thanks For Listening