> Agenda (10)

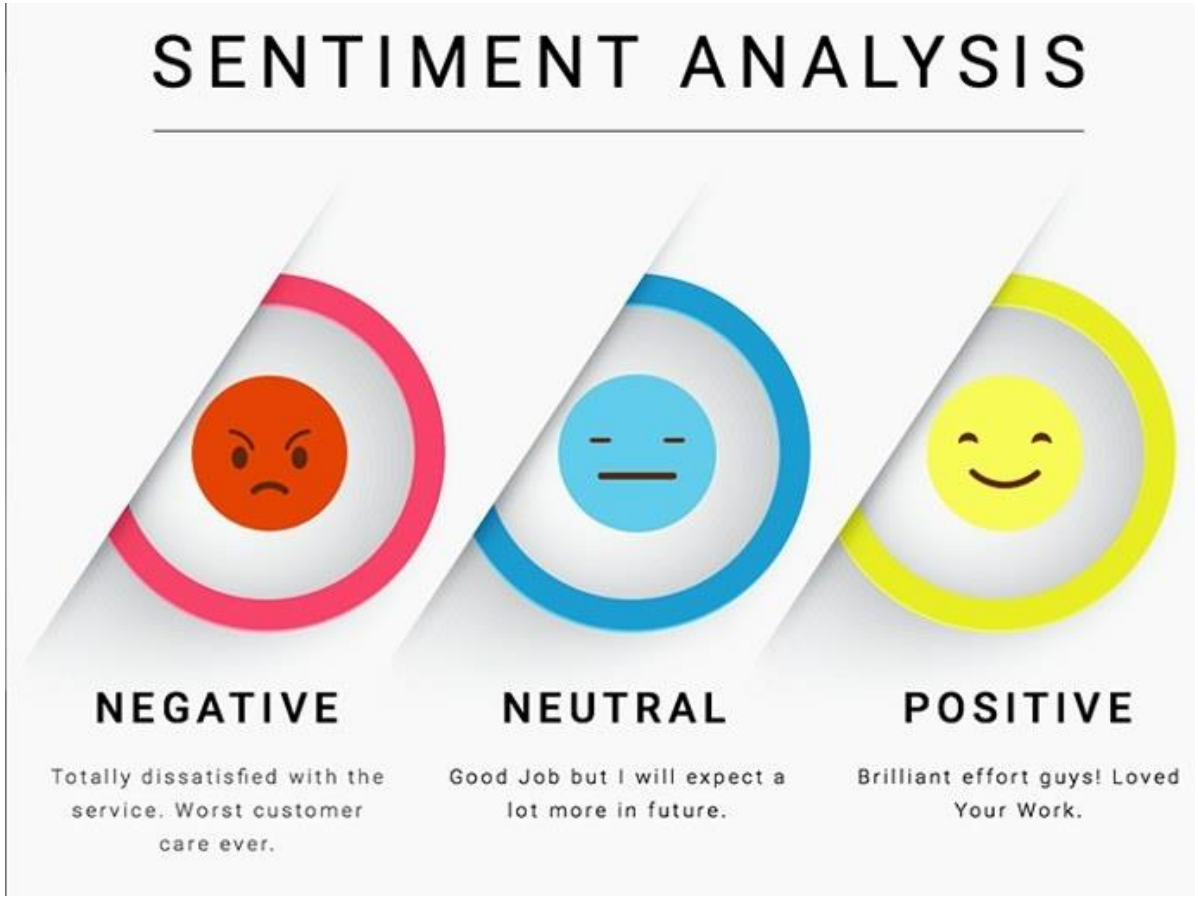
- Sentiment Analysis Definition
- Assignment #4
- Sentiment Analysis EXAMPLE

Azure Week

Microsoft

> Sentiment Analysis | Practice

Sentiment analysis (also known as **opinion mining** or **emotion AI**) refers to the use of natural language processing, text analysis, computational linguistics, and biometrics to systematically identify, extract, quantify, and study affective states and subjective information. Sentiment analysis is widely applied to voice of the customer materials such as reviews and survey responses, online and social media, and healthcare materials for applications that range from marketing to customer service to clinical medicine.



https://en.wikipedia.org/wiki/Sentiment_analysis

> Sentiment Analysis | Practice

Sentiment analysis is the process of detecting positive or negative sentiment in text. It's often used by businesses to detect sentiment in social data, gauge brand reputation, and understand customers.

The most popular applications of sentiment analysis in real life:

- Social media monitoring
- Customer support
- Customer feedback
- Brand monitoring and reputation management
- Voice of customer (VoC)
- Voice of employee
- Product analysis
- Market research and competitive research

https://monkeylearn.com/sentiment-analysis/

> #4 Sentiment Analysis Assignment

INSTRUCTIONS:

Record a video with 5-10 minutes **OR** prepare a presentation explaining how to create a **Sentiment Analysis solution (OR ALTERNATIVE SOLUTION)** and your challenges.

- Use you Azure / AWS instance (student) **OR** local (Azure is not mandatory)
- Select your data sources
- Perform all the steps to acquire data
- Select your preferred lib (python or other)
- Use ADF to ingest data to your preferred database
- Generate visuals using your preferred visualization tool (PowerBI, Tableau, etc.)
- Evaluate the results

In your video **OR** presentation explain how you performed each step and interpret the result. You can use the example below as a reference.

RUBRIC:

Mark: 20%

( ) project architecture
( ) code / lib used
( ) project presentation (video **OR** PPT) Coherence and adherence with the assignment
( ) project screenshots (solution running, evidences)
( ) explanation (level of details, clarity, quality)
( ) data presentation (dashboards, results)
( ) your code on GitHub (submit your link!)
( ) creativity – solution scope and design
( ) conclusions reached
( ) project references, idea, design, code, etc.

Are allowed groups of 2 members (max)

Due date: Week 11 class

SCREENCAST MATIC

To record your video you can use:
https://screencast-o-matic.com/

> #4 Sentiment Analysis Assignment

ALTERNATIVE SOLUTIONS: (1)

Alternatively, you can choose:

- NLP - Natural Language Processing
- Sentiment analysis for depression based on social media post
- Virtual Assistants / Chat Bots
- Face Recognition (Face API or other solution)
- Advertising and product suggestions
- Translation (Translator API or other solution)
- Handwritten equation solver (CNN)
- Social Network
- Predict Credit Default - Credit Risk Prediction Project
- Business meeting summary generation (NLP)

ALTERNATIVE SOLUTIONS: (2)

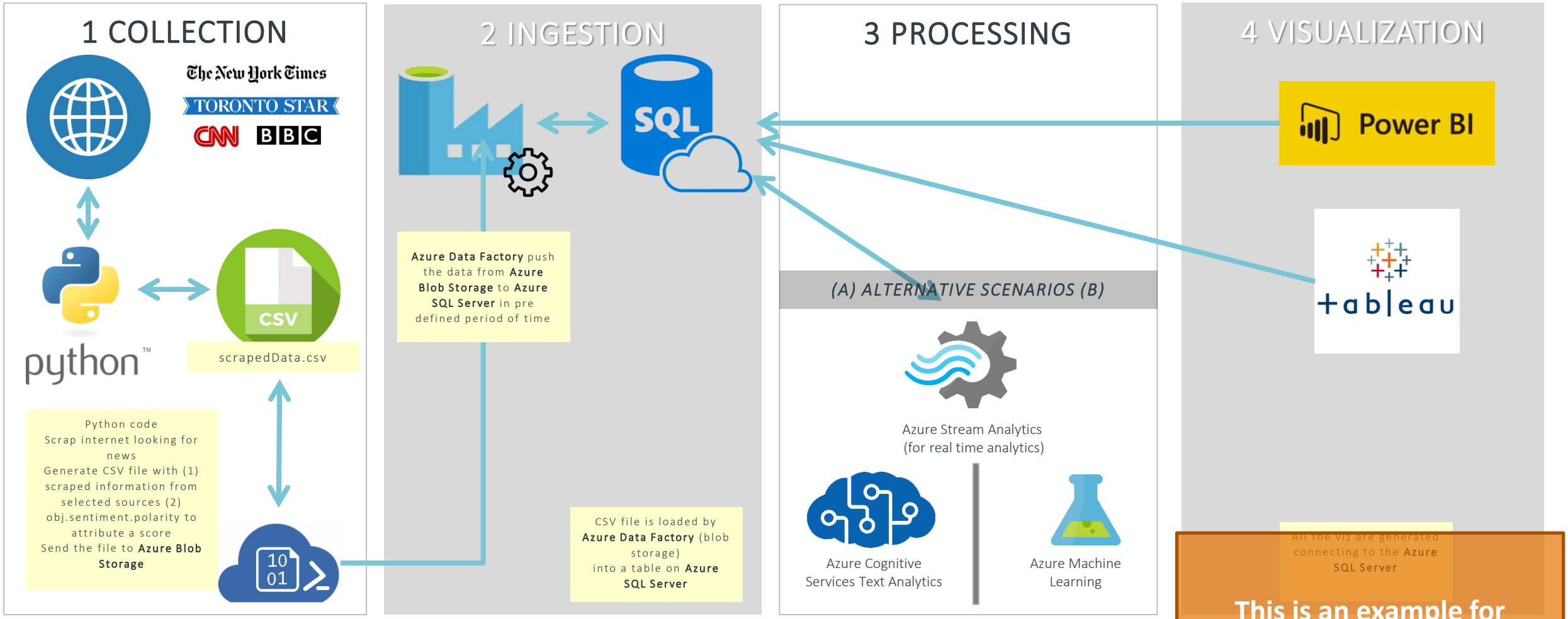
- Sales Forecasting using X Dataset
- Facial recognition to detect mood and suggest songs
- Finding out habitable exo-planet from images
- Music generation (DL)
- Iris Flowers Classification
- Stock Prices Predictor (time series)
- Build a Movie Recommender System
- Inventory Demand Forecasting
- Predicting Interest Levels of Rental Listings
- Fake News Classification
- Speech Emotion Recognition

Practice #4

Sentiment Analysis

> Sentiment Analysis | Practice

PROJECT ARCHITECTURE



This is an example for academic purposes

> Sentiment Analysis | Practice

TEXT BLOB LIBRARY

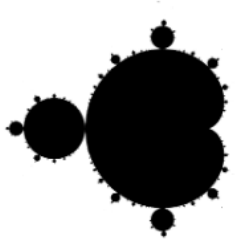
SENTIMENT ANALYSIS

```
from textblob import TextBlob
sentiment =
obj.sentiment.polarity
```

Features

- Noun phrase extraction
- Part-of-speech tagging
- Sentiment analysis
- Classification (Naive Bayes, Decision Tree)
- Language translation and detection powered by Google Translate
- Tokenization (splitting text into words and sentences)
- Word and phrase frequencies
- Parsing
- n-grams
- Word inflection (pluralization and singularization) and lemmatization
- Spelling correction
- Add new models or languages through extensions
- WordNet integration

```
$ pip install -U textblob
$ python -m textblob.download_corpora
```



TextBlob

Star 6,859

TextBlob is a Python (2 and 3) library for processing textual data. It provides a consistent API for diving into common natural language processing (NLP) tasks such as part-of-speech tagging, noun phrase extraction, sentiment analysis, and more.

Useful Links

TextBlob @ PyPI
TextBlob @ GitHub
Issue Tracker

Stay Informed

Follow @sloria

TextBlob: Simplified Text Processing

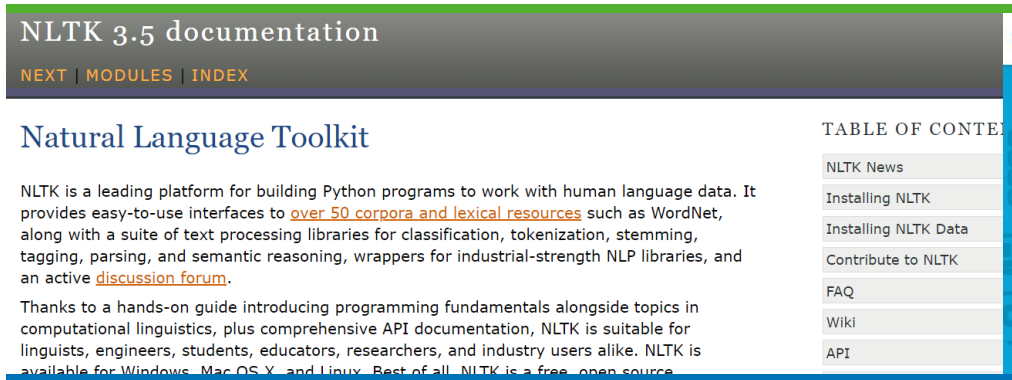
Release v0.15.2. (Changelog)

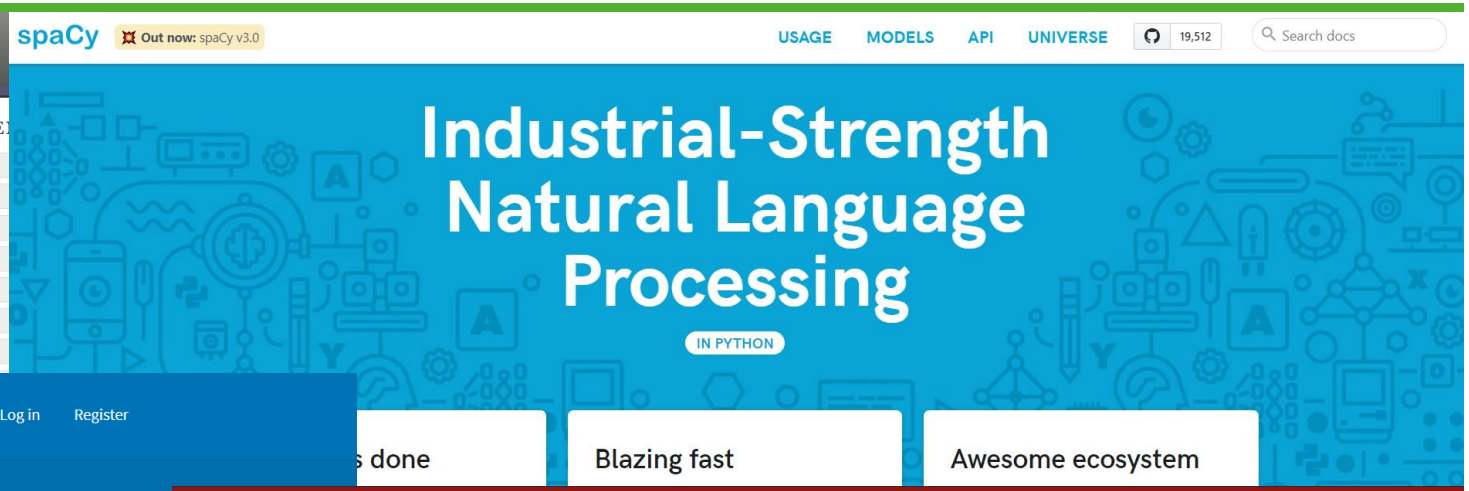
TextBlob is a Python (2 and 3) library for processing textual data. It provides a simple API for diving into common natural language processing (NLP) tasks such as part-of-speech tagging, noun phrase extraction, sentiment analysis, classification, translation, and more.

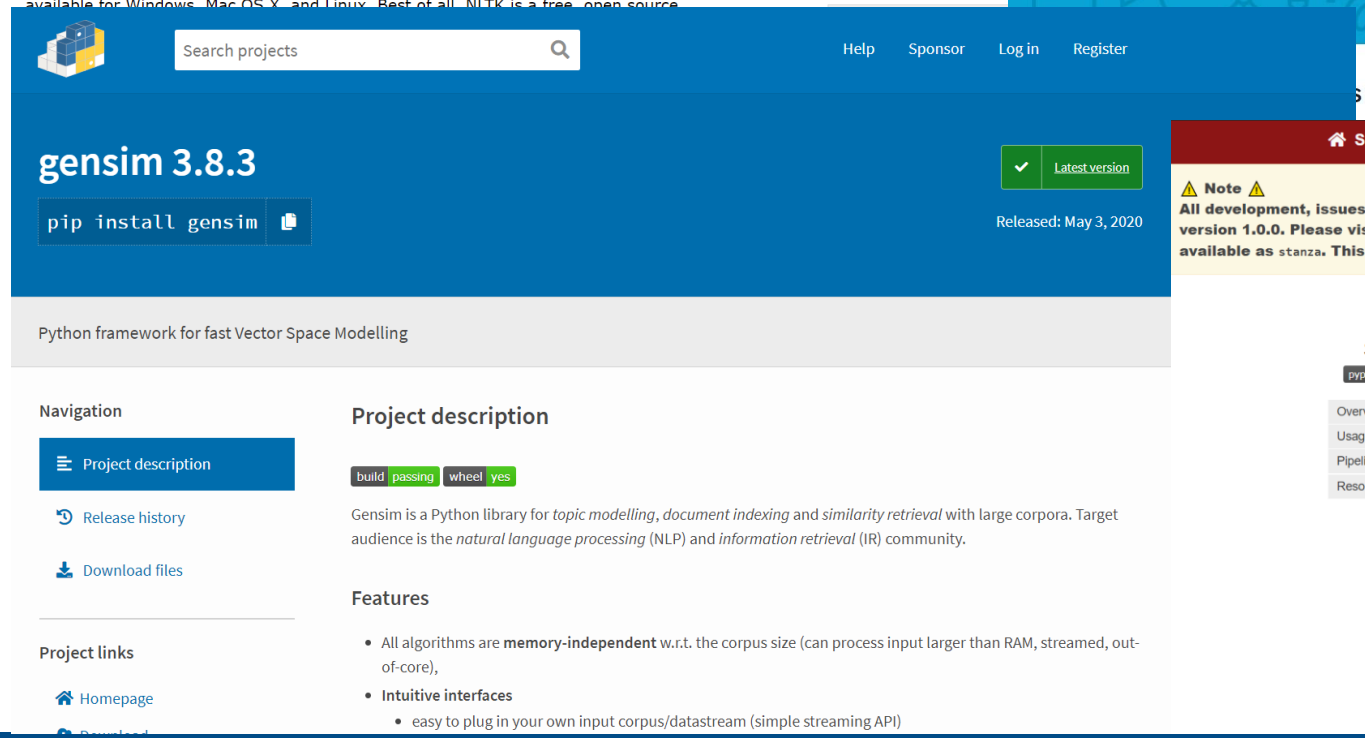
```
from textblob import TextBlob

text = '''
The titular threat of The Blob has always struck me as the ultimate movie
monster: an insatiably hungry, amoeba-like mass able to penetrate
virtually any safeguard, capable of--as a doomed doctor chillingly
describes it--"assimilating flesh on contact.
Snide comparisons to gelatin be damned, it's a concept with the most
devastating of potential consequences, not unlike the grey goo scenario
proposed by technological theorists fearful of
artificial intelligence run rampant.
'''

blob = TextBlob(text)
blob.tags           # [('The', 'DT'), ('titular', 'JJ'),
                    #  ('threat', 'NN'), ('of', 'IN'), ...]

blob.noun_phrases   # WordList(['titular threat', 'blob',
                    #           'ultimate movie monster',
                    #           'amoeba-like mass', ...])

for sentence in blob.sentences:
    print(sentence.sentiment.polarity)
```

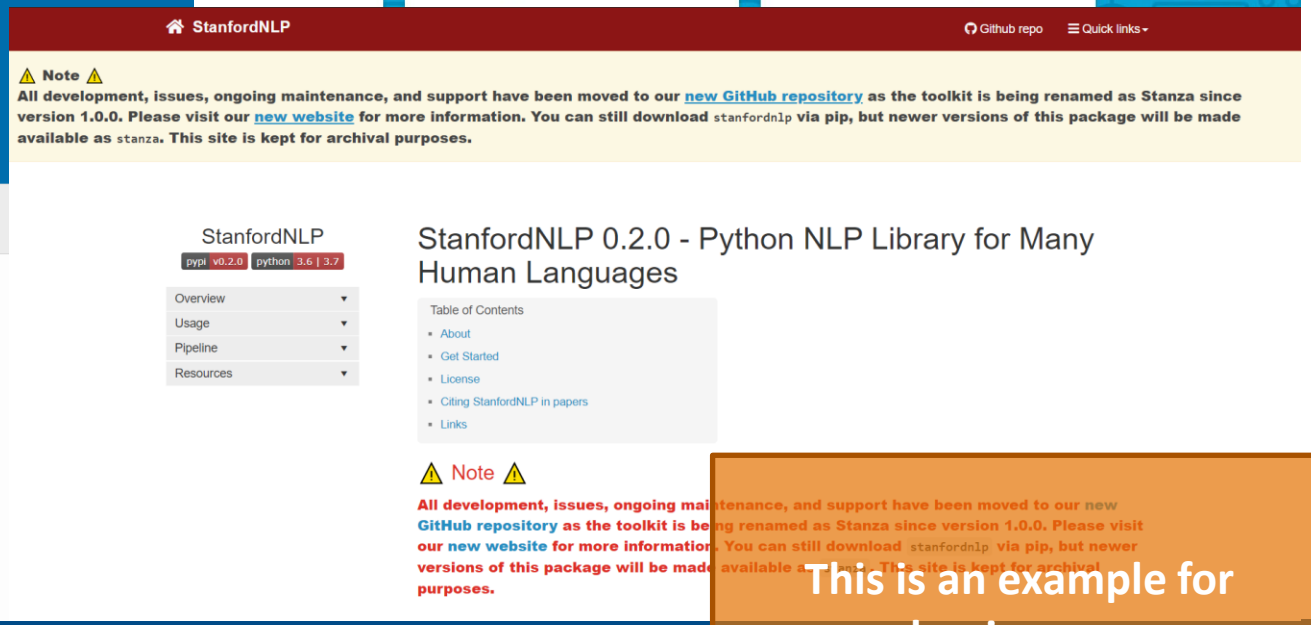
This is an example for academic purposes

> Sentiment Analysis | Practice







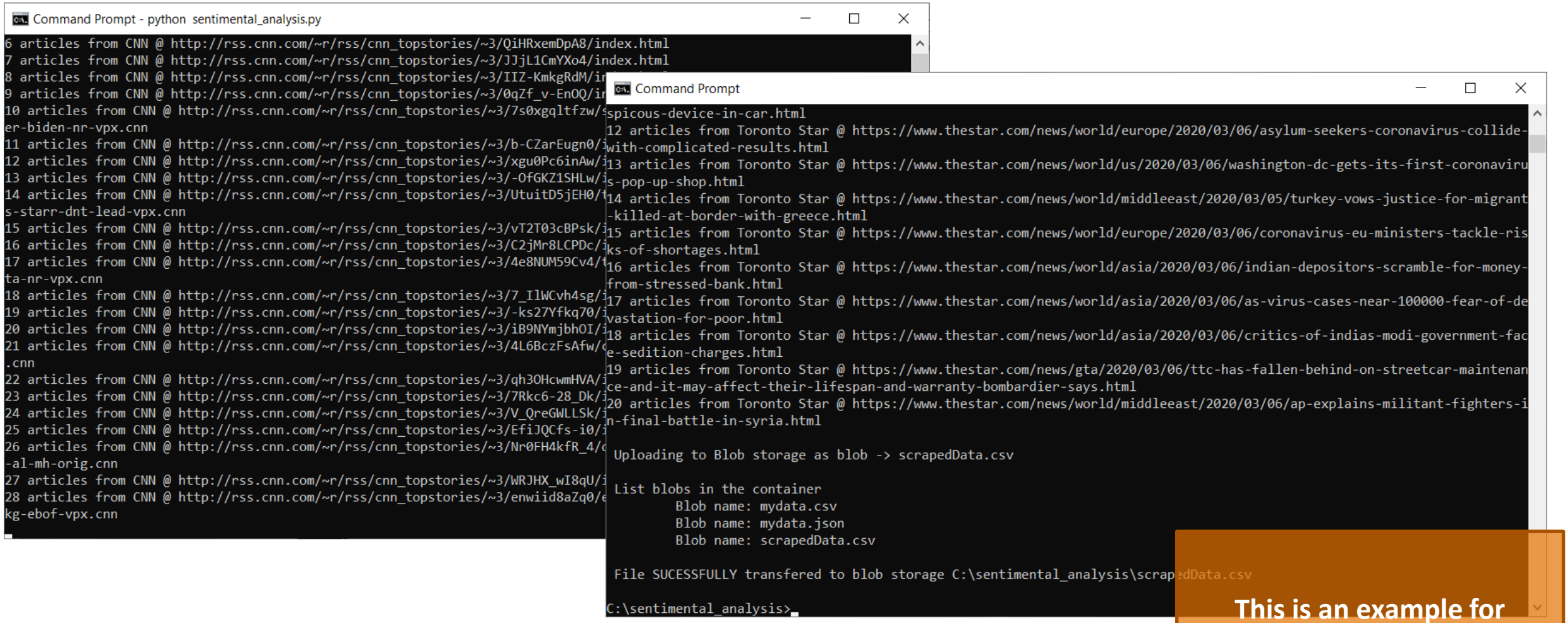


This is an example for academic purposes

Georgian

> Sentiment Analysis | Practice

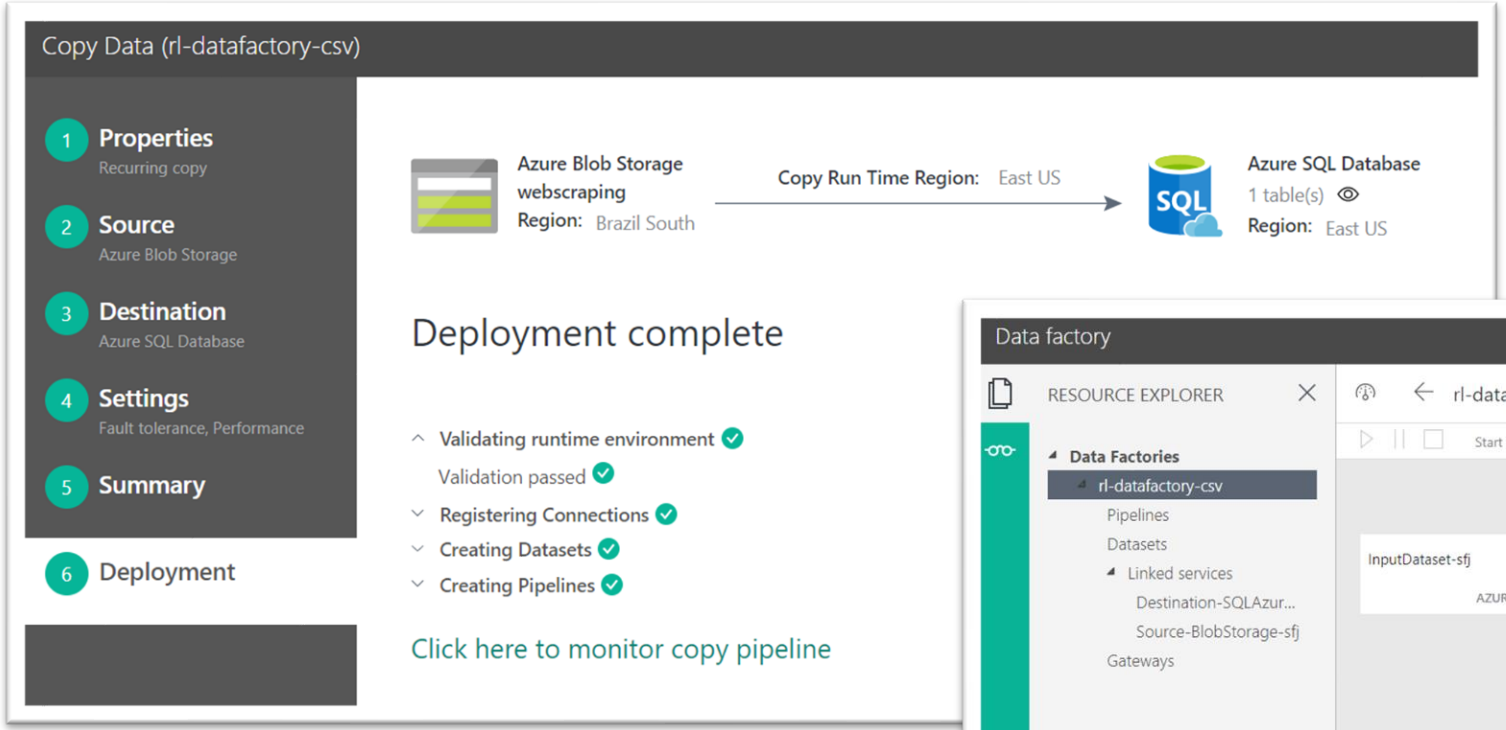
PYTHON (TEXT BLOB LIB) CODE RUNNING…

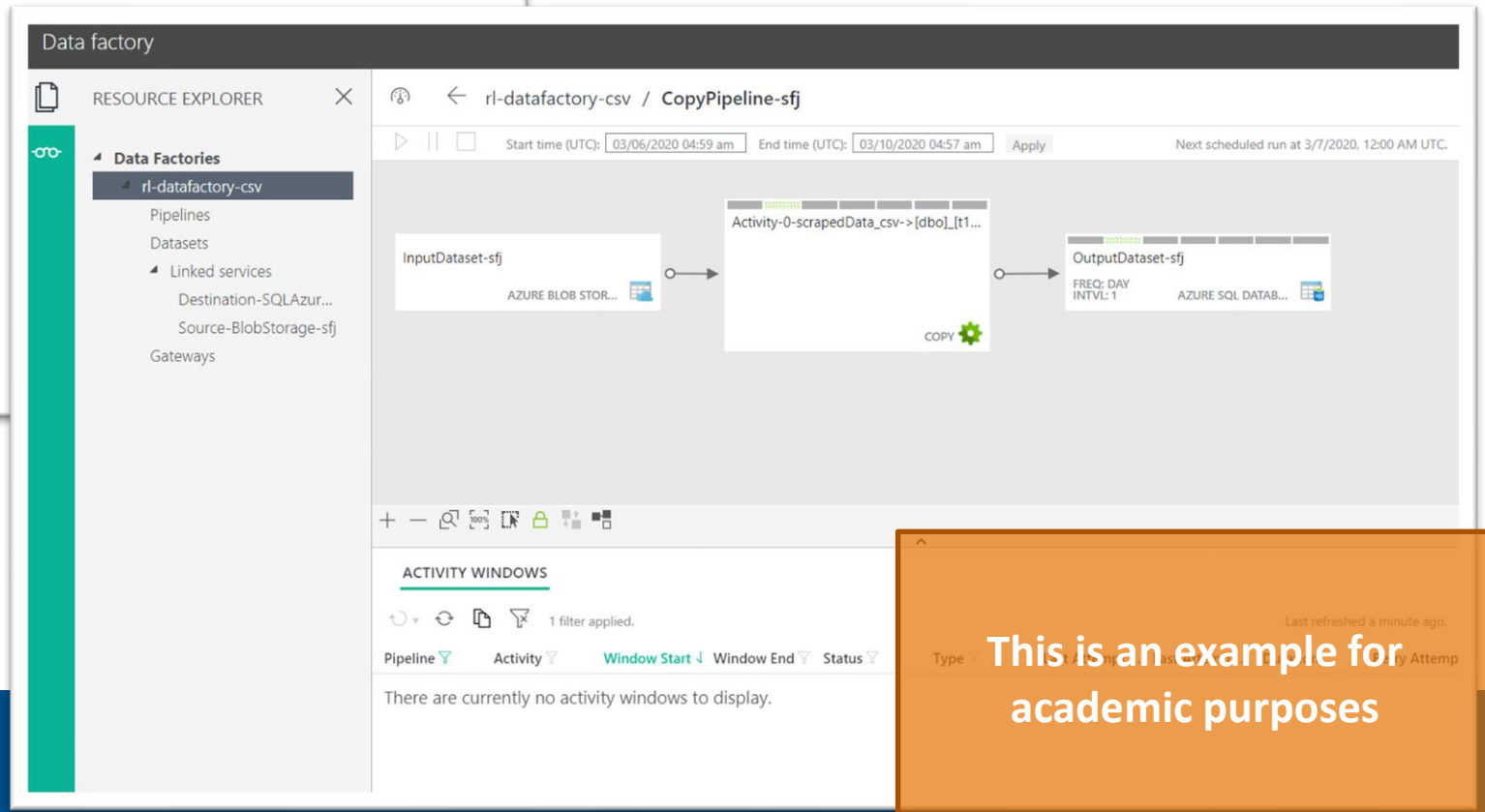


This is an example for academic purposes

> Sentiment Analysis | Practice

AZURE DATA FACTORY - DATA INGESTION PROCESS





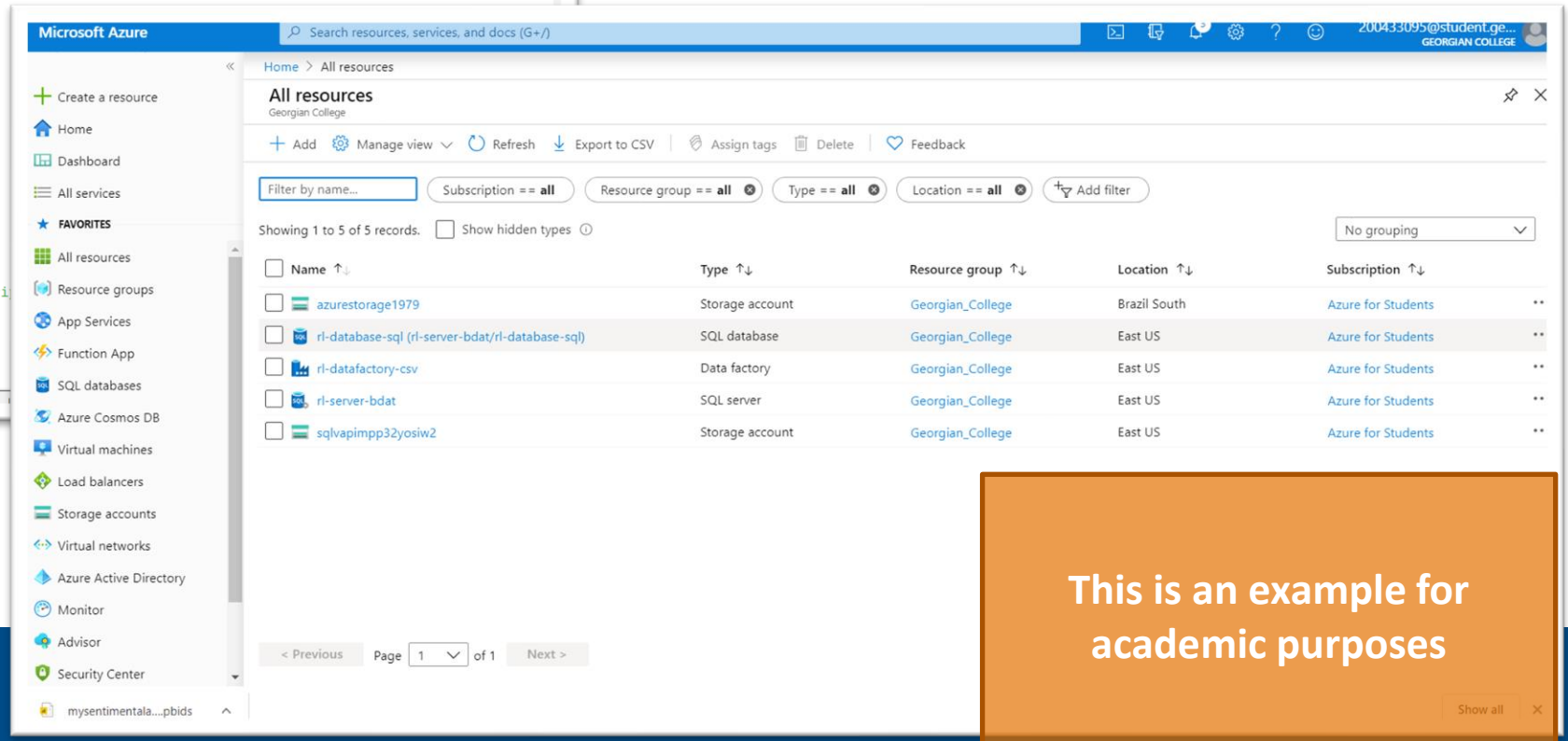
This is an example for academic purposes

Georgian

> Sentiment Analysis | Practice

PYTHON CODE AND AZURE RESOURCES

```python
import feedparser as fp
import json
import newspaper
from newspaper import Article
from time import mktime
from datetime import datetime
import csv

LIMIT = 250
articles_array=[]

data={}
data['newspapers']={}
with open("NewsPapers.json") as data_file:
    companies = json.load(data_file)

count = 1
for company, value in companies.items():
    if 'rss' in value:
        d = fp.parse(value['rss'])
        print("Downloading articles from ", company)
        newsPaper = {
            "rss": value['rss'],
            "link": value['link'],
            "articles": []
        }
        for entry in d.entries:
            # Check if publish date is provided, if no the article is skipped.
            # This is done to keep consistency in the data and to keep the scri
            if hasattr(entry, 'published'):
                if count > LIMIT:
                    break
                article = {}
                article['link'] = entry.link
```

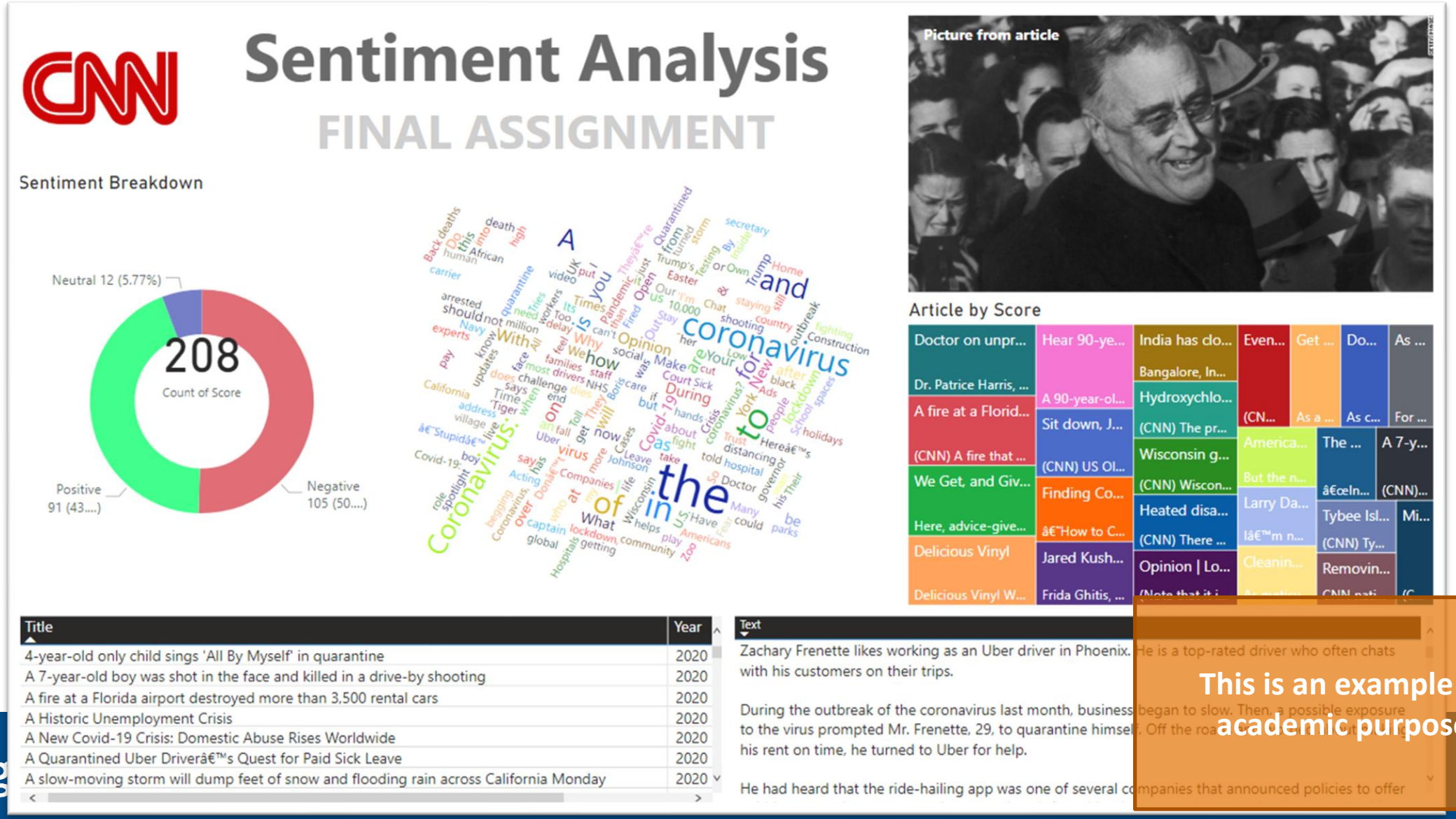


This is an example for academic purposes

Georgian

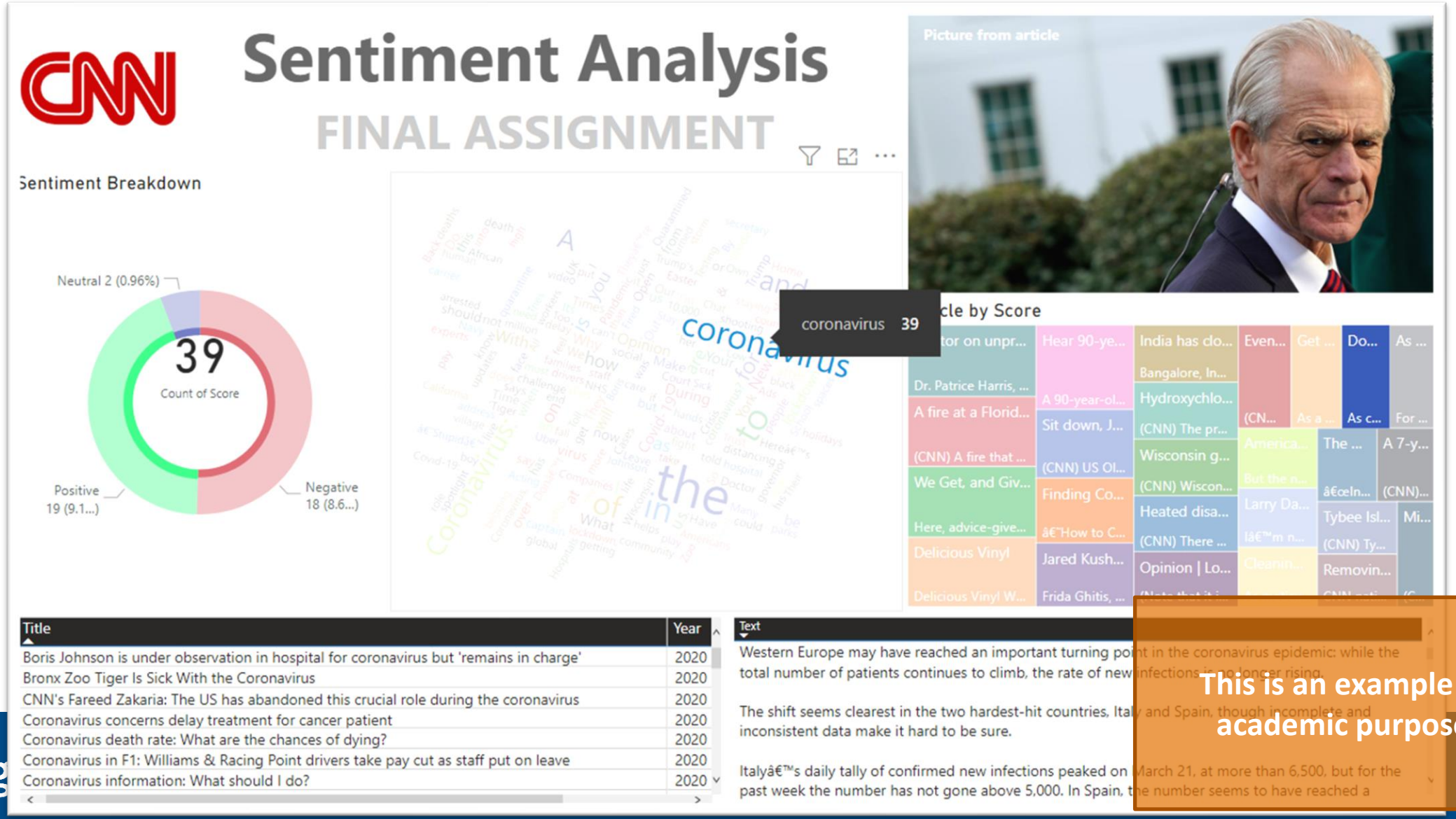
> Sentiment Analysis | Practice

SAMPLE DASHBOARD - POWER BI



> Sentiment Analysis | Practice

SAMPLE DASHBOARD - POWER BI



END OF DAY 10