

# > Agenda (10)

---

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



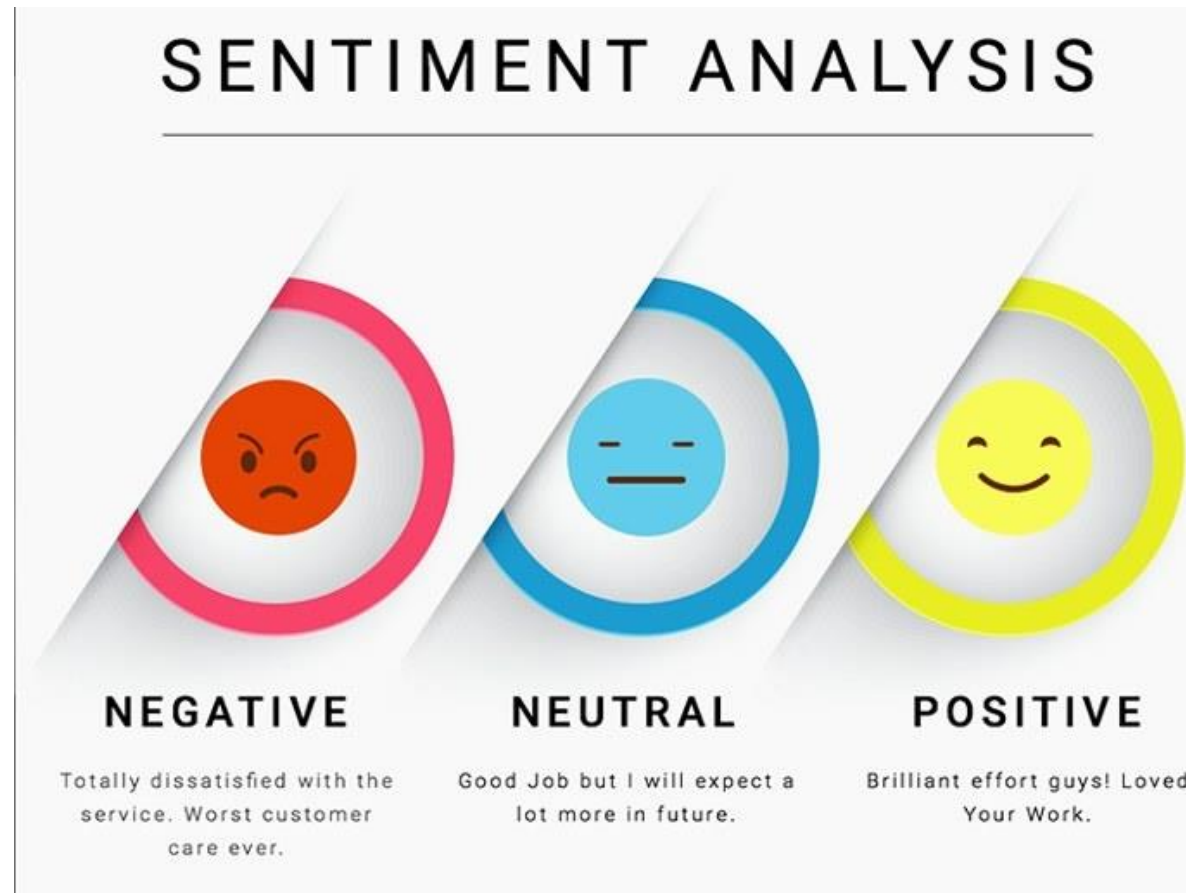
# Azure Week

## 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.



# > 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

# > #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



# > #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

SCREENCAST  MATIC

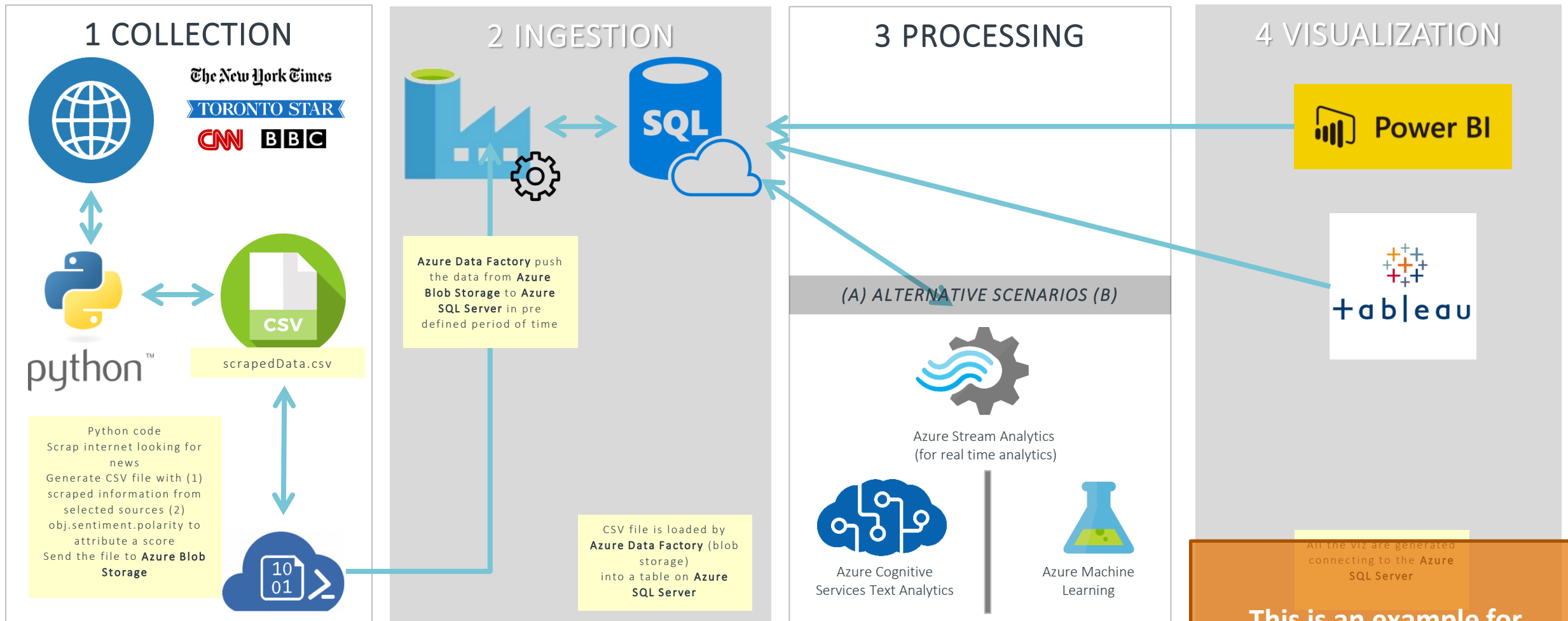
# Practice #4

## Sentiment Analysis



# > Sentiment Analysis | Practice

## PROJECT ARCHITECTURE



This is an example for academic purposes



# > Sentiment Analysis | Practice

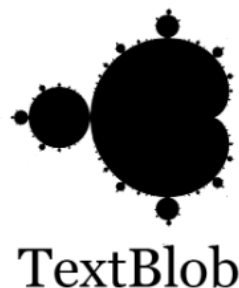
## TEXT BLOB LIBRARY

### 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
```



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

## NLTK 3.5 documentation

[NEXT](#) | [MODULES](#) | [INDEX](#)

### Natural Language Toolkit

NLTK is a leading platform for building Python programs to work with human language data. It provides easy-to-use interfaces to [over 50 corpora and lexical resources](#) such as WordNet, along with a suite of text processing libraries for classification, tokenization, stemming, tagging, parsing, and semantic reasoning, wrappers for industrial-strength NLP libraries, and an active [discussion forum](#).

Thanks to a hands-on guide introducing programming fundamentals alongside topics in computational linguistics, plus comprehensive API documentation, NLTK is suitable for linguists, engineers, students, educators, researchers, and industry users alike. NLTK is available for Windows, Mac OS X, and Linux. Best of all, NLTK is a free, open source

| TABLE OF CONTENTS    |  |
|----------------------|--|
| NLTK News            |  |
| Installing NLTK      |  |
| Installing NLTK Data |  |
| Contribute to NLTK   |  |
| FAQ                  |  |
| Wiki                 |  |
| API                  |  |

## spaCy

Out now: spaCy v3.0

[USAGE](#) [MODELS](#) [API](#) [UNIVERSE](#) [19,512](#)

# Industrial-Strength Natural Language Processing

IN PYTHON

Is done Blazing fast Awesome ecosystem

### StanfordNLP

[GitHub repo](#) [Quick links](#)

**Note**  
All development, issues, ongoing maintenance, and support have been moved to our [new GitHub repository](#) as the toolkit is being renamed as Stanza since version 1.0.0. Please visit our [new website](#) for more information. You can still download stanfordnlp via pip, but newer versions of this package will be made available as stanza. This site is kept for archival purposes.

## gensim 3.8.3

`pip install gensim`

Released: May 3, 2020

Python framework for fast Vector Space Modelling

### Navigation

- Project description
- Release history
- Download files

### Project links

- Homepage
- Download

### Project description

[build](#) [passing](#) [wheel](#) [yes](#)

Gensim is a Python library for *topic modelling*, *document indexing* and *similarity retrieval* with large corpora. Target audience is the *natural language processing* (NLP) and *information retrieval* (IR) community.

### Features

- All algorithms are **memory-independent** w.r.t. the corpus size (can process input larger than RAM, streamed, out-of-core),
- Intuitive interfaces**
  - easy to plug in your own input corpus/datastream (simple streaming API)

This is an example for academic purposes

# > Sentiment Analysis | Practice

## PYTHON (TEXT BLOB LIB) CODE RUNNING...

```
Command Prompt - python sentimental_analysis.py
6 articles from CNN @ http://rss.cnn.com/~r/rss/cnn_topstories/~3/QiHRxemDpA8/index.html
7 articles from CNN @ http://rss.cnn.com/~r/rss/cnn_topstories/~3/JJjL1CmYXo4/index.html
8 articles from CNN @ http://rss.cnn.com/~r/rss/cnn_topstories/~3/IIZ-KmkgRdM/i
9 articles from CNN @ http://rss.cnn.com/~r/rss/cnn_topstories/~3/0qZf_v-EnOQ/i
10 articles from CNN @ http://rss.cnn.com/~r/rss/cnn_topstories/~3/7s0xqgltfzw/
er-biden-nr-vpx.cnn
11 articles from CNN @ http://rss.cnn.com/~r/rss/cnn_topstories/~3/b-CZarEugn0/3
12 articles from CNN @ http://rss.cnn.com/~r/rss/cnn_topstories/~3/xgu0Pc6inAw/3
13 articles from CNN @ http://rss.cnn.com/~r/rss/cnn_topstories/~3/-OfGKZ1SHLw/3
14 articles from CNN @ http://rss.cnn.com/~r/rss/cnn_topstories/~3/UtuuitD5jEH0/3
s-starr-dnt-lead-vpx.cnn
15 articles from CNN @ http://rss.cnn.com/~r/rss/cnn_topstories/~3/vT2T03cBPsk/3
16 articles from CNN @ http://rss.cnn.com/~r/rss/cnn_topstories/~3/C2jMr8LCPDc/3
17 articles from CNN @ http://rss.cnn.com/~r/rss/cnn_topstories/~3/4e8NUM59Cv4/3
ta-nr-vpx.cnn
18 articles from CNN @ http://rss.cnn.com/~r/rss/cnn_topstories/~3/7_ILWCvh4sg/3
19 articles from CNN @ http://rss.cnn.com/~r/rss/cnn_topstories/~3/-ks27Yfkq70/3
20 articles from CNN @ http://rss.cnn.com/~r/rss/cnn_topstories/~3/iB9NYmjhbOI/3
21 articles from CNN @ http://rss.cnn.com/~r/rss/cnn_topstories/~3/4L6BczFsAfw/c
.cnn
22 articles from CNN @ http://rss.cnn.com/~r/rss/cnn_topstories/~3/qh30HcwmHVA/3
23 articles from CNN @ http://rss.cnn.com/~r/rss/cnn_topstories/~3/7Rkc6-28_Dk/3
24 articles from CNN @ http://rss.cnn.com/~r/rss/cnn_topstories/~3/V_QreGWLLSk/3
25 articles from CNN @ http://rss.cnn.com/~r/rss/cnn_topstories/~3/EfiJQCfs-i0/3
26 articles from CNN @ http://rss.cnn.com/~r/rss/cnn_topstories/~3/Nr0FH4kfR_4/c
-al-mh-orig.cnn
27 articles from CNN @ http://rss.cnn.com/~r/rss/cnn_topstories/~3/WRJHX_wI8qU/3
28 articles from CNN @ http://rss.cnn.com/~r/rss/cnn_topstories/~3/enwiid8aZq0/e
kg-ebob-vpx.cnn

Command Prompt
12 articles from Toronto Star @ https://www.thestar.com/news/world/europe/2020/03/06/asylum-seekers-coronavirus-collide-
with-complicated-results.html
13 articles from Toronto Star @ https://www.thestar.com/news/world/us/2020/03/06/washington-dc-gets-its-first-coronaviru
s-pop-up-shop.html
14 articles from Toronto Star @ https://www.thestar.com/news/world/middleeast/2020/03/05/turkey-vows-justice-for-migrant
-killed-at-border-with-greece.html
15 articles from Toronto Star @ https://www.thestar.com/news/world/europe/2020/03/06/coronavirus-eu-ministers-tackle-ris
ks-of-shortages.html
16 articles from Toronto Star @ https://www.thestar.com/news/world/asia/2020/03/06/indian-depositors-scramble-for-money-
from-stressed-bank.html
17 articles from Toronto Star @ https://www.thestar.com/news/world/asia/2020/03/06/as-virus-cases-near-100000-fear-of-de
vastation-for-poor.html
18 articles from Toronto Star @ https://www.thestar.com/news/world/asia/2020/03/06/critics-of-indias-modi-government-fac
e-sedition-charges.html
19 articles from Toronto Star @ https://www.thestar.com/news/gta/2020/03/06/ttc-has-fallen-behind-on-streetcar-maintenan
ce-and-it-may-affect-their-lifespan-and-warranty-bombardier-says.html
20 articles from Toronto Star @ https://www.thestar.com/news/world/middleeast/2020/03/06/ap-explains-militant-fighters-i
n-final-battle-in-syria.html

Uploading to Blob storage as blob -> scrapedData.csv

List blobs in the container
  Blob name: mydata.csv
  Blob name: mydata.json
  Blob name: scrapedData.csv

File SUCESSFULLY transfered to blob storage C:\sentimental_analysis\scrapedData.csv
C:\sentimental_analysis>
```

This is an example for  
academic purposes

# > Sentiment Analysis | Practice

## AZURE DATA FACTORY - DATA INGESTION PROCESS

### Copy Data (rl-datafactory-csv)

- 1 Properties  
Recurring copy
- 2 Source  
Azure Blob Storage
- 3 Destination  
Azure SQL Database
- 4 Settings  
Fault tolerance, Performance
- 5 Summary
- 6 Deployment

Azure Blob Storage  
webscraping  
Region: Brazil South

Copy Run Time Region: East US

Azure SQL Database  
1 table(s)  
Region: East US

### Deployment complete

- Validating runtime environment ✓  
Validation passed ✓
- Registering Connections ✓
- Creating Datasets ✓
- Creating Pipelines ✓

[Click here to monitor copy pipeline](#)

### Data factory

RESOURCE EXPLORER

- Data Factories
  - rl-datafactory-csv
    - Pipelines
    - Datasets
    - Linked services
      - Destination-SQLAzur...
      - Source-BlobStorage-sfj
    - Gateways

### rl-datafactory-csv / CopyPipeline-sfj

Start time (UTC): 03/06/2020 04:59 am End time (UTC): 03/10/2020 04:57 am Apply Next scheduled run at 3/7/2020, 12:00 AM UTC.

InputDataset-sfj (AZURE BLOB STOR...) → Activity-0-scrapedData\_csv->[dbo]\_[t1...] (COPY) → OutputDataset-sfj (AZURE SQL DATAB...)

ACTIVITY WINDOWS

1 filter applied.

Pipeline Activity Window Start Window End Status

There are currently no activity windows to display.

This is an example for academic purposes

# > Sentiment Analysis | Practice

## PYTHON CODE AND AZURE RESOURCES

```
1 import feedparser as fp
2 import json
3 import newspaper
4 from newspaper import Article
5 from time import mktime
6 from datetime import datetime
7 import csv
8
9 LIMIT = 250
10 articles_array=[]
11
12 data={}
13 data['newspapers']={}
14 with open("NewsPapers.json") as data_file:
15     companies = json.load(data_file)
16
17 count = 1
18 for company, value in companies.items():
19     if 'rss' in value:
20         d = fp.parse(value['rss'])
21         print("Downloading articles from ", company)
22         newsPaper = {
23             "rss": value['rss'],
24             "link": value['link'],
25             "articles": []
26         }
27         for entry in d.entries:
28             # Check if publish date is provided, if no the article is skipped.
29             # This is done to keep consistency in the data and to keep the scri
30             if hasattr(entry, 'published'):
31                 if count > LIMIT:
32                     break
33                 article = {}
34                 article['link'] = entry.link
```

Microsoft Azure

Search resources, services, and docs (G+)

Home > All resources

All resources

Georgian College

+ Add Manage view Refresh Export to CSV Assign tags Delete Feedback

Filter by name... Subscription == all Resource group == all Type == all Location == all Add filter

Showing 1 to 5 of 5 records. Show hidden types No grouping

| Name ↑↓                                          | Type ↑↓         | Resource group ↑↓ | Location ↑↓  | Subscription ↑↓    |    |
|--------------------------------------------------|-----------------|-------------------|--------------|--------------------|----|
| azurestorage1979                                 | Storage account | Georgian_College  | Brazil South | Azure for Students | ** |
| ri-database-sql (ri-server-bdat/ri-database-sql) | SQL database    | Georgian_College  | East US      | Azure for Students | ** |
| ri-datafactory-csv                               | Data factory    | Georgian_College  | East US      | Azure for Students | ** |
| ri-server-bdat                                   | SQL server      | Georgian_College  | East US      | Azure for Students | ** |
| sqlvapimpp32yosiw2                               | Storage account | Georgian_College  | East US      | Azure for Students | ** |

< Previous Page 1 of 1 Next >

mysentimental...pbids

This is an example for academic purposes



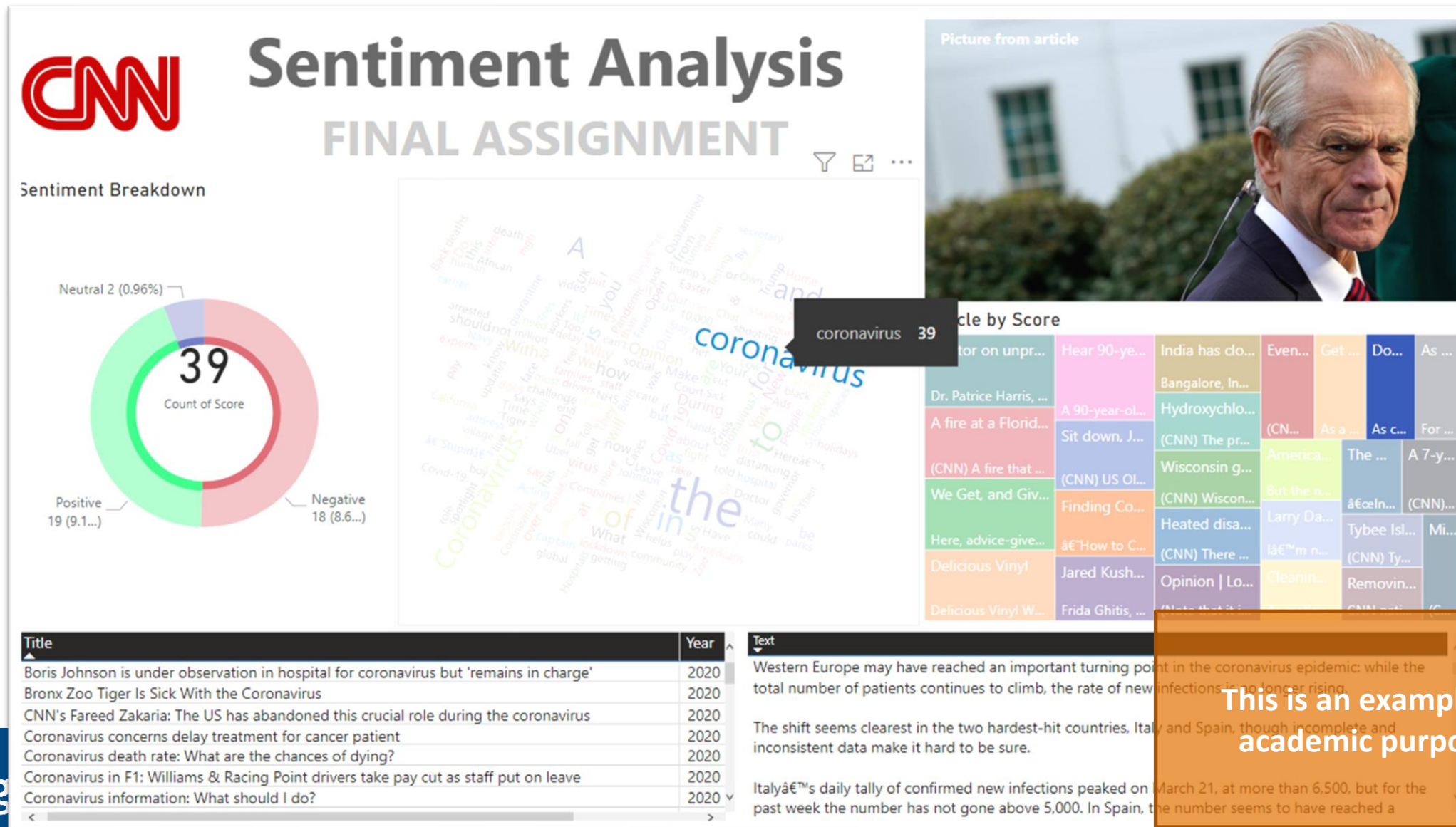
## SAMPLE DASHBOARD - POWER BI



**This is an example for academic purposes**

# > Sentiment Analysis | Practice

## SAMPLE DASHBOARD - POWER BI



This is an example for academic purposes





END OF DAY 10