UNIVERSITY OF TORONTO

INSPIRING INCLUSIVE EXCELLENCE

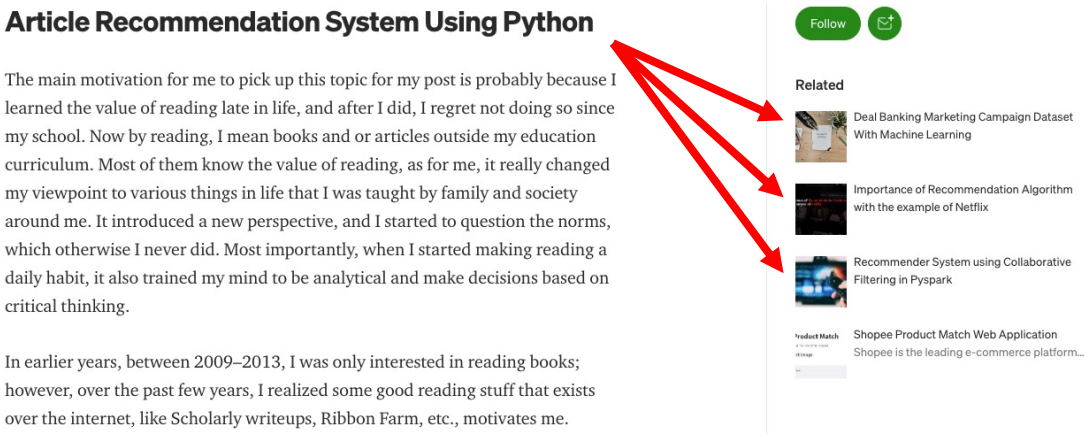
Article Recommendation System

University of Toronto 3760 Term Project | April 2022

Adon/Steven

BUSINESS CASE

- Build an article recommendation system that **EVERY** publisher can use



UNIVERSITY OF TORONTO

WHY PUBLISHER NEED IT?

- Help viewers to find desired articles quickly
  - **Better** customer satisfaction
- Increase viewer engagement
  - **Higher** impression/ads revenue
  - **More** subscribers

REVENUE AND COST ANALYSIS

- Assumptions:
  - Publisher gets 1M page views and 10M impressions on monthly basis
  - Subscriber conversion 0.002% per page views
  - RPM (**R**evenue **P**er thousand **I**mpressions) is $10
  - LTV (**L**ife**t**ime **V**alue) per subs $400

Recommendation System Revenue:

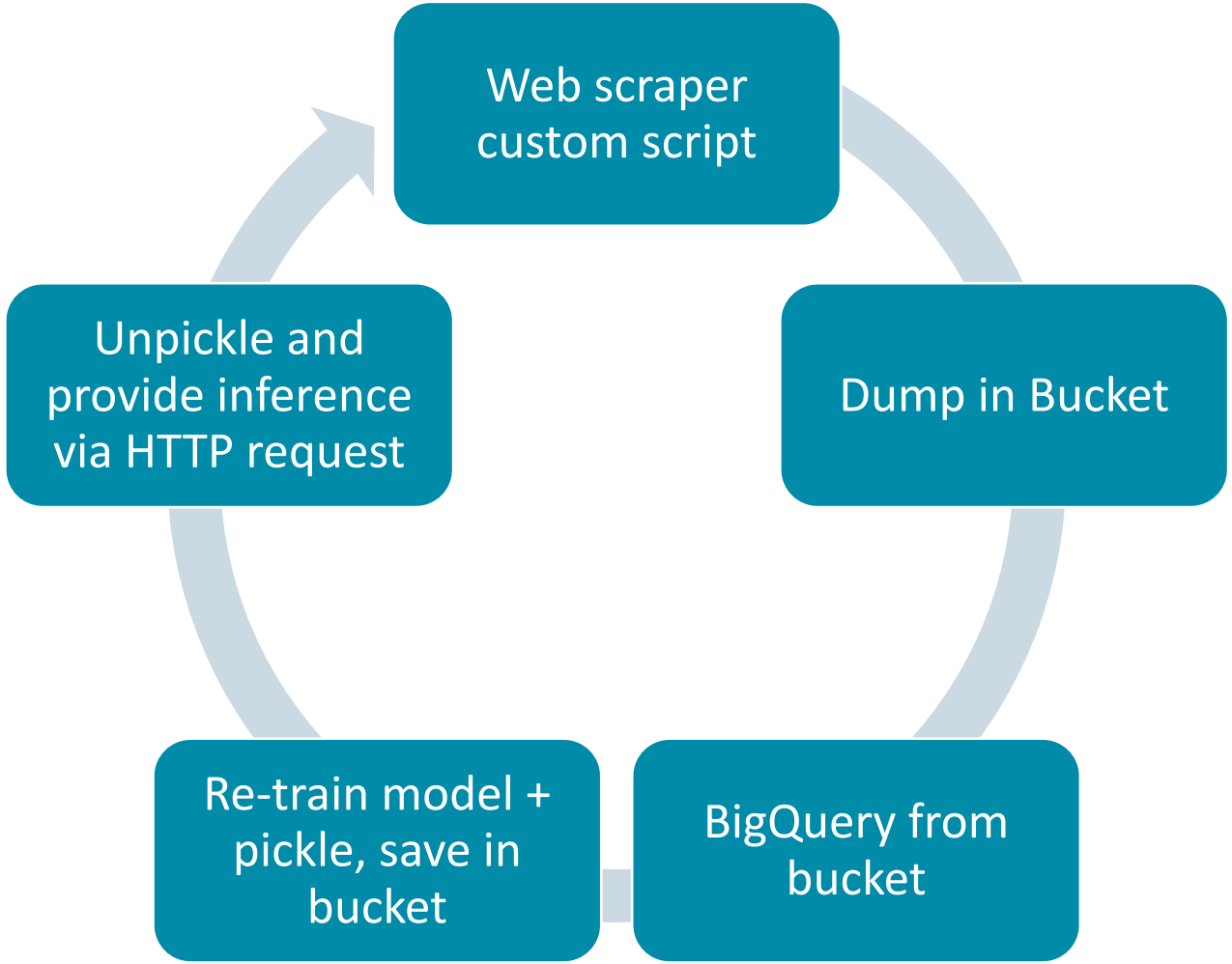
	Average Case	**Best Case**
% of User Clicks Recommended article	5%	10%
Extra Impressions	500,000	1,000,000
Extra RPM($)	$5,000	$10,000
Extra Subscribers	1	2
Extra Subs Revenue	$400	$800
Total Extra Revenue	**$5,400**	**$10,800**

WHY CLOUD SERVICE?

- Most of publishers have already moved to online and they are already using website analytic tool (eg. Google analytic) to collect data
- # of pageviews are unpredictable and it keeps changing over time, # of inference from ML model need **scale up/down** depending on pageviews
- Highly sensitive data can be **encrypted** and **protected** by VPN
- The same recommender system be **replicated** to a new publisher quickly

SOLUTION ARCHITECTURE OVERVIEW

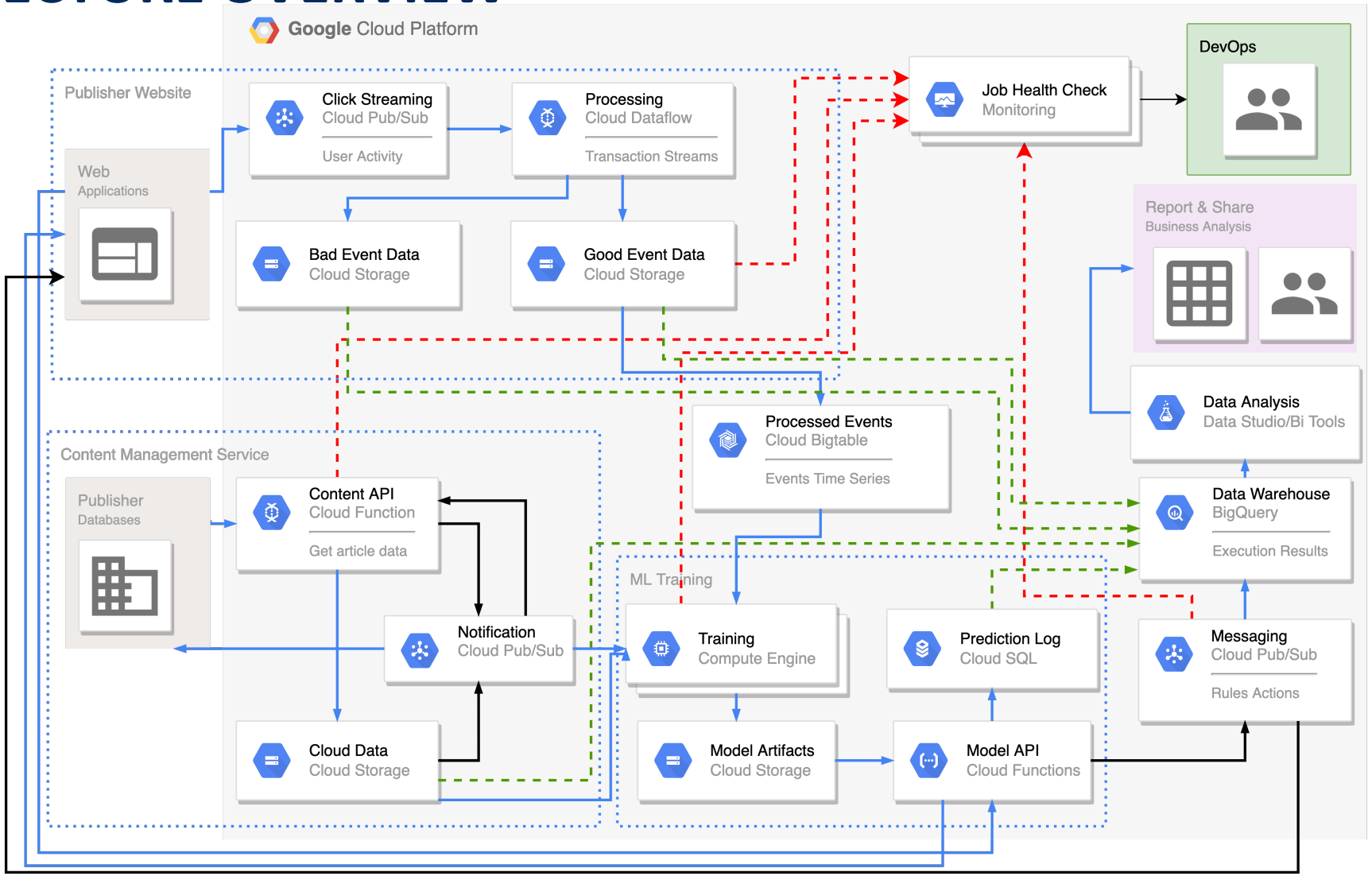
- (low resolution summary)



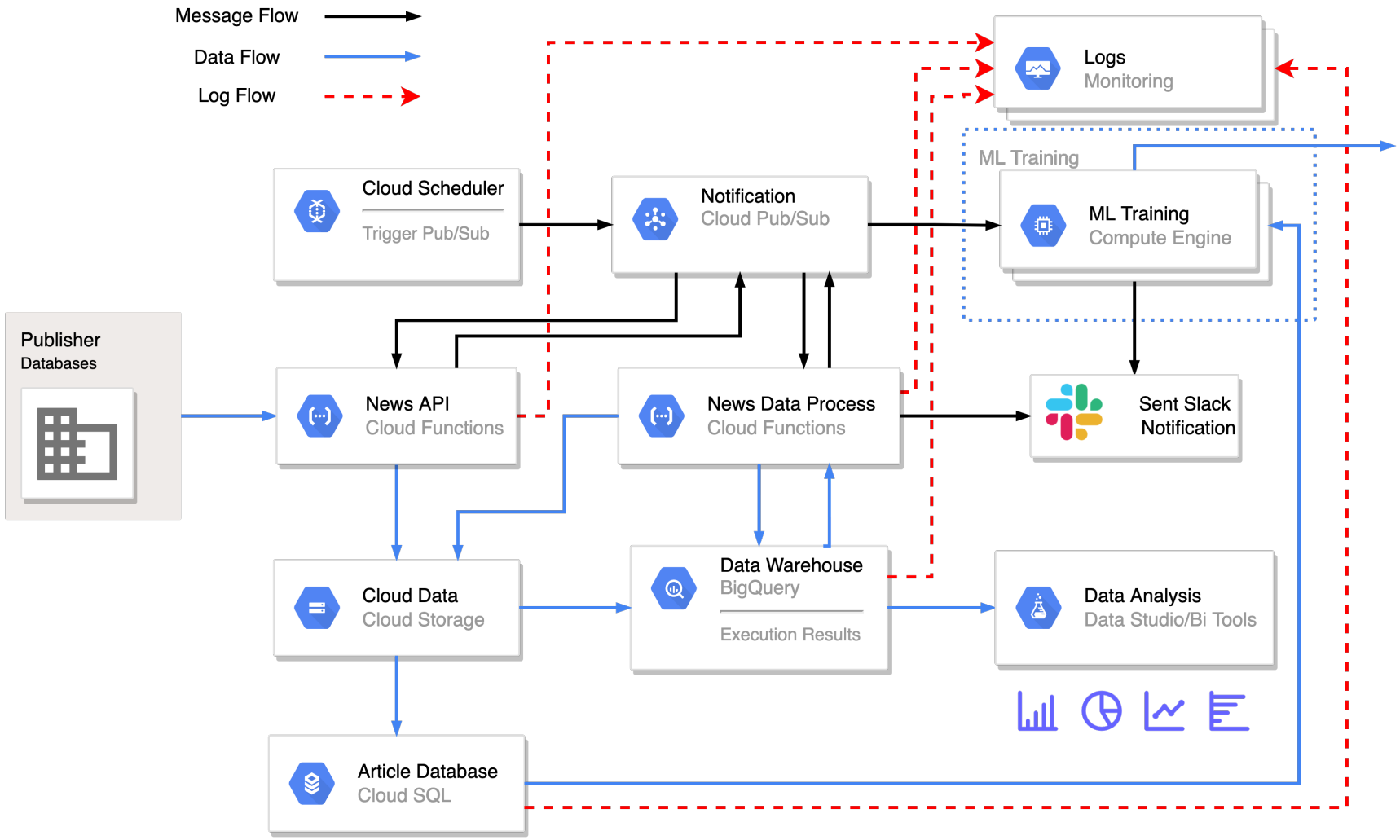
SOLUTION ARCHITECTURE OVERVIEW

The solution is implemented in GCP using:

- *Cloud Functions*
- *Cloud Storage*
- *BigQuery*
- *Cloud SQL*
- *Pub/sub scheduler*

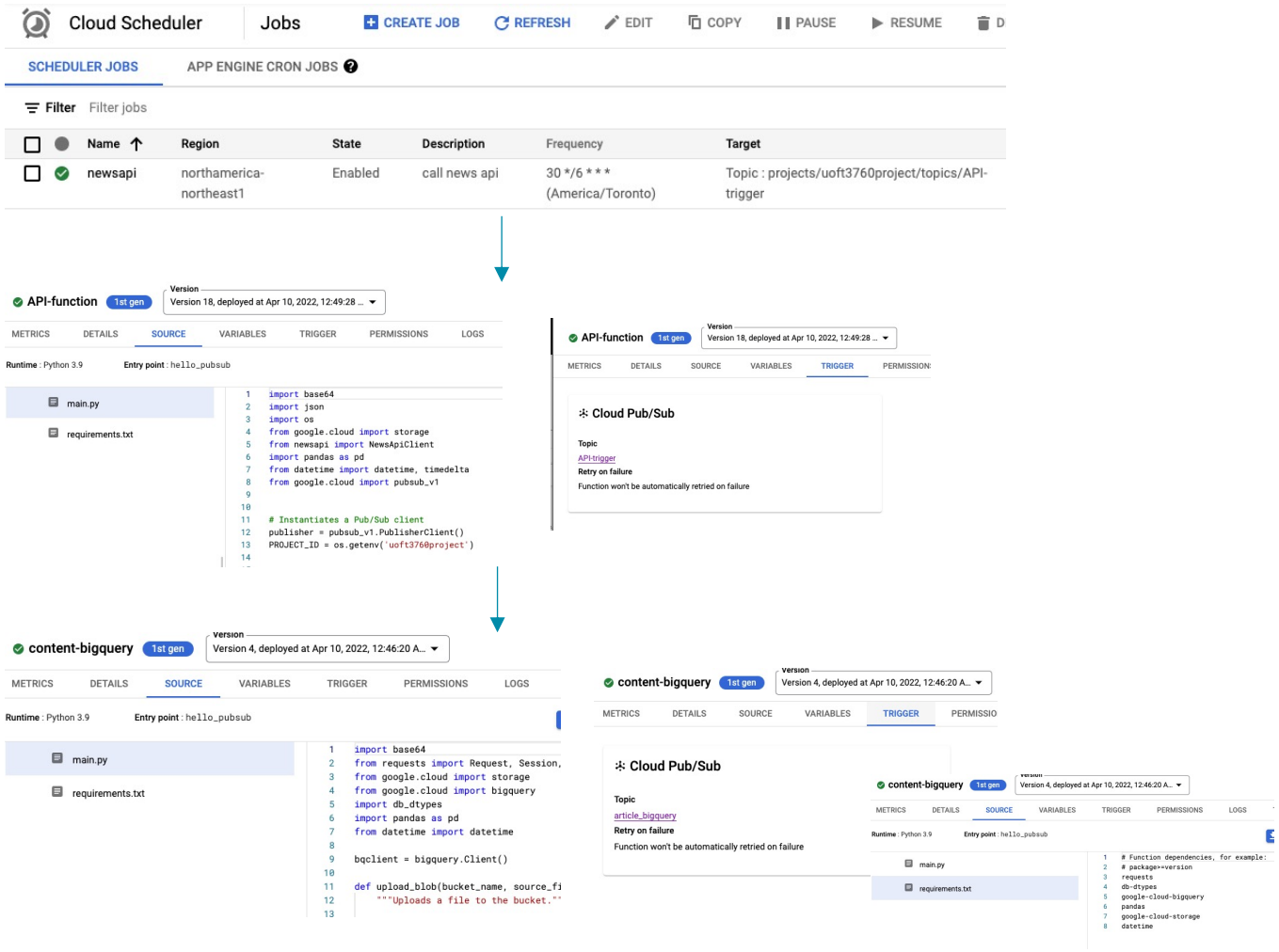


CONTENT MANAGEMENT

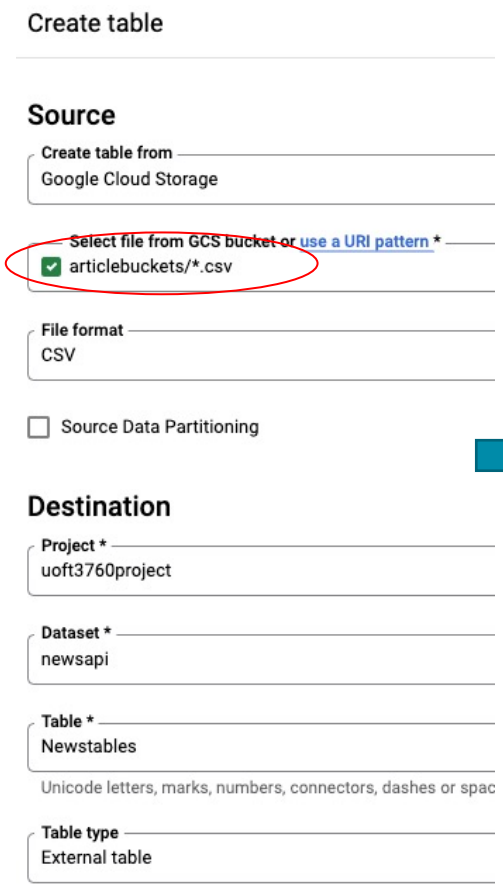


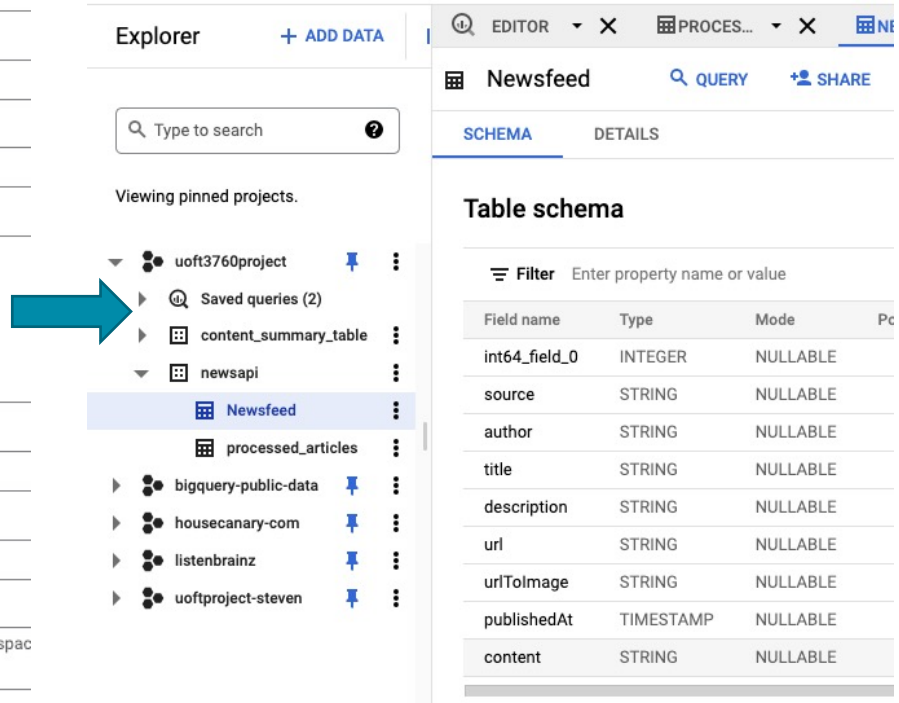
IMPLEMENTATION

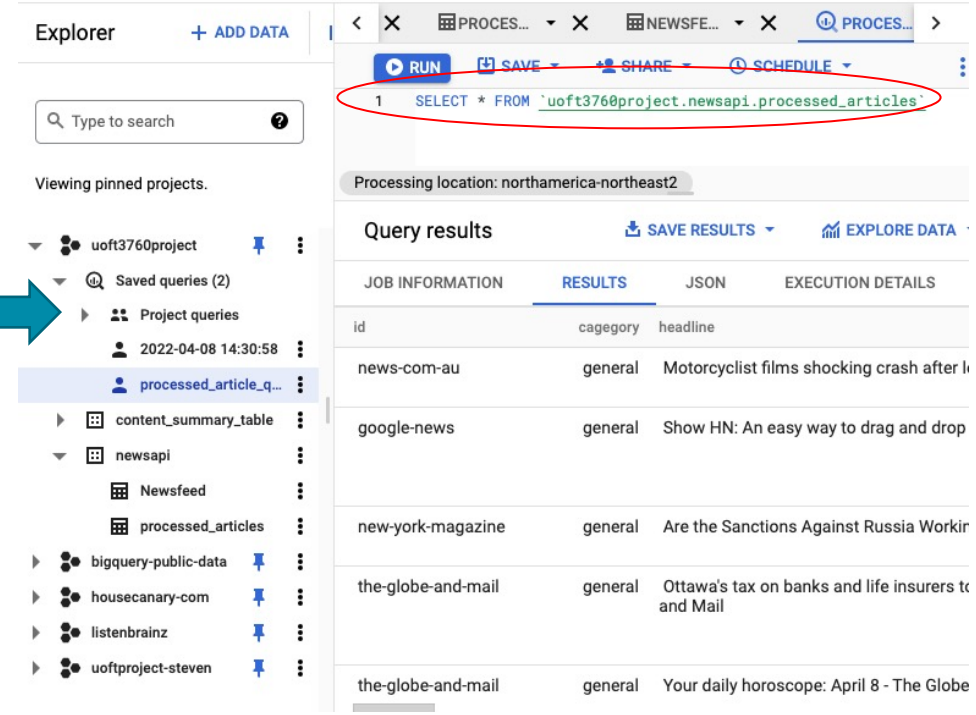
- Cloud Scheduler
- Pub/sub #1
- Cloud function #1 (News API)
  - Get News from Publisher API
  - Save csv files to Cloud Storage
- Pub/sub #2
- Cloud Function #2
  - Run BigQuery get data from Cloud Storage
  - Process Data
  - Save Processed Data to Cloud Storage
- Update data to Cloud SQL
- Pub/Sub #3
- Train model



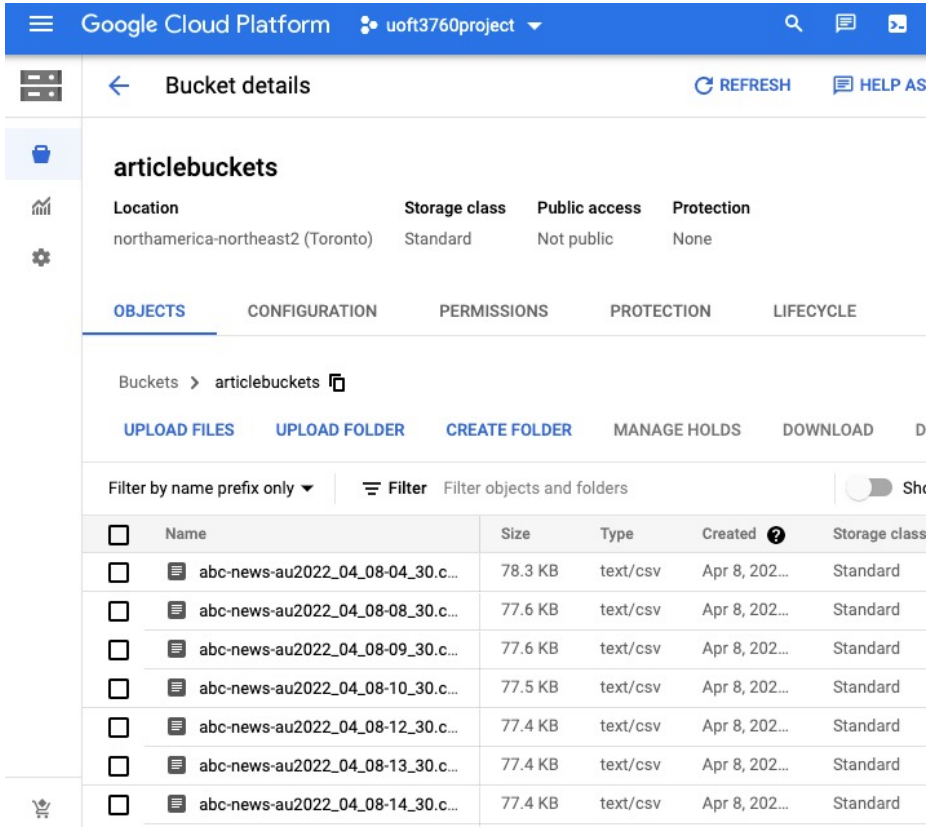
IMPLEMENTATION CONT. (BIGQUERY)







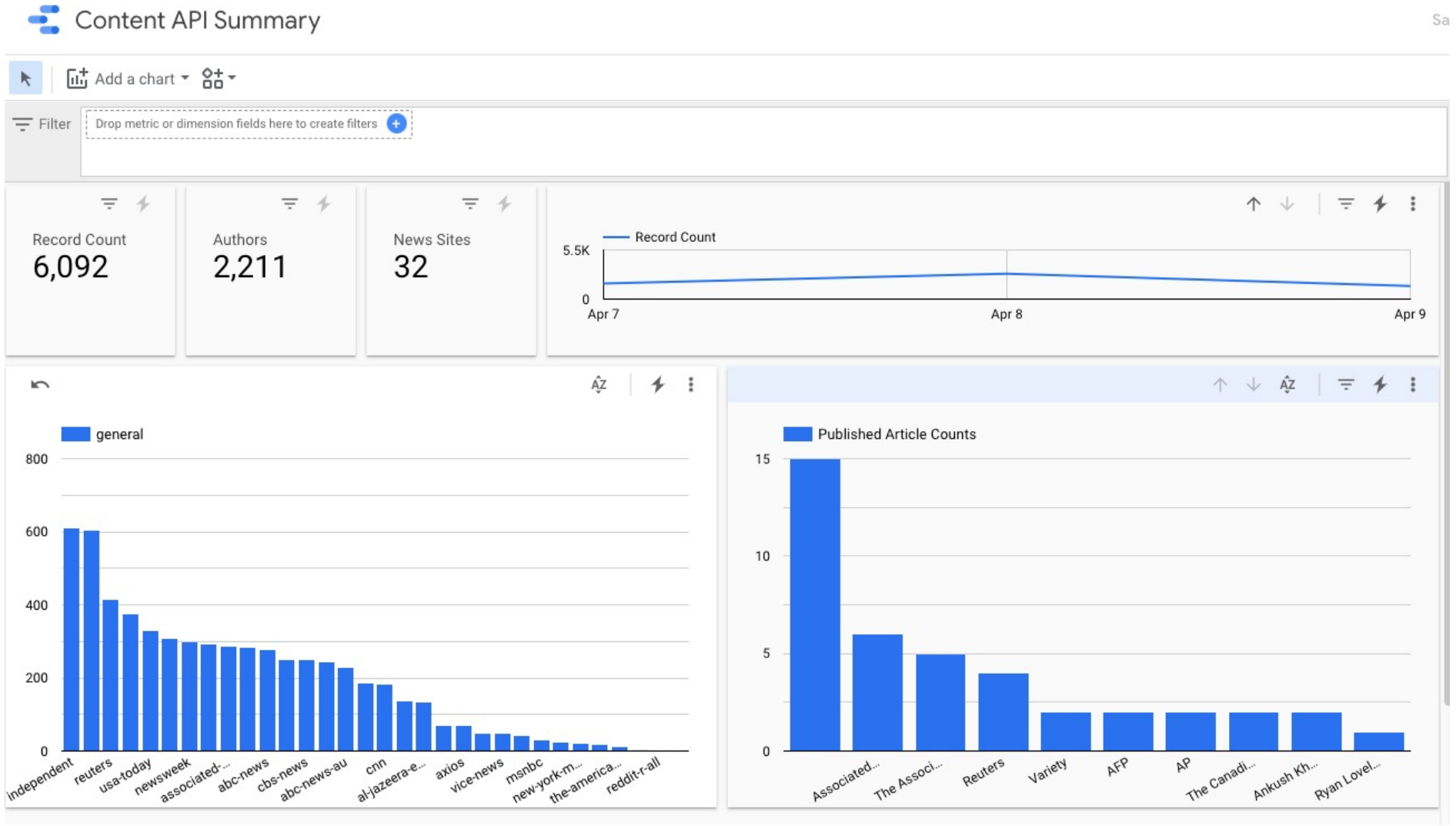
IMPLEMENTATION CONT. (CLOUD STORAGE)



WORD TO VECTOR MODEL

- We are using sklearn.feature_extraction.text.TfidfVectorizer and sklearn.metrics.pairwise_distances
  - TfidfVectorizer takes all the “words” from each article to create a matrix of term frequency
  - Pairwise distances is used to calculate the Euclidian distance between two matrices
- The final model is weighted:
  - 0.6x headline and short description
  - 0.2x category
  - 0.1x author
  - 0.1x publish day

DATA ANALYSIS



INSPIRING INCLUSIVE EXCELLENCE

Thank you!

Any Questions?

UNIVERSITY OF TORONTO