CITYLITICS

DATA ENGINEERING · INTERN PROJECT

18M+ URL DATA INTEGRATION PIPELINE

Cleaning, scoring, and labeling the raw crawl data behind Citylitics' Insights product

Beia · Data Engineering Intern · 2026

PROJECT OVERVIEW

From Noisy Crawl to ML-Ready Dataset

Citylitics runs a web crawler that visits **40,000+ municipal and utility websites** across North America, surfacing the documents and pages that become the "Insights" Citylitics sells to customers.

My focus sat between the raw crawl and everything downstream: turning a huge, noisy table of discovered URLs into a **clean, labeled, ML-ready dataset.**

Crawler discovers raw URLs → my pipeline cleans, scores & labels them → downstream classification model → customer Insights

40,000+

municipal & utility sites crawled across North America

Tens of millions

of raw crawled URL rows, the starting point for this project

THE PROBLEM

A Huge, Noisy Table of URLs

Before anything could be trained or trusted, the raw crawl table needed to be curated. Four issues stood in the way:

Duplicates

The same URL discovered repeatedly across crawl passes.

Nulls

Rows missing the fields a clean dataset depends on.

Dead links & redirects

Outdated URLs that no longer resolve to real content.

No real municipal owner

Social platforms and other URLs with no municipal/utility owner attached.

Curating the Gold-Standard Dataset

BEFORE

Tens of millions of raw rows

- Duplicate URLs
- Null / missing fields
- Outdated, dead links
- URLs with no real municipal owner
- Slow queries & processing downstream

AFTER

~18M curated rows

- Deduplicated, non-null, live links only
- Every row tied to a real municipal owner
- Cut to roughly a third of raw size
- Substantially faster queries & processing
- Gold standard for training & evaluating the classification model

A Weighted URL Priority Score

Every discovered URL is scored on whether it's worth crawling and whether it points to a document or a webpage, by combining five signals:

Document type	PDF vs. HTML
Domain similarity	Edit distance to known-good municipal domains
Keyword priority	Score from a gradient-boosted model
Crawl depth	How far the URL sits from the crawl's starting point
Platform exclusion flag	Filters out social media & non-owner platforms

Bias check: corrected for long URLs scoring artificially higher simply because they contain more keywords.

Human-in-the-Loop Labeling Pipeline

The scoring model needed ground truth to train against, so labels were generated in an iterative loop:

1

Sample a batch

Pull a batch of unlabeled URLs from the pool

2

Manual classification

Each URL is presented for manual review & labeling

3

Batch write-back

Labels are written back to the dataset in batches

4

Repeat

Cycle continues until the unlabeled pool is exhausted

ALONG THE WAY

Production Issues Debugged

Schema / type mismatch

A “discard” label introduced nulls into an integer column.

FIX Fixed the schema handling so discard labels no longer broke the column type.

Duplicate URL fan-out

Duplicate URLs caused a single label to fan out across multiple rows during a merge update.

FIX Deduplicated URLs before merging labels back into the dataset.

Non-deterministic split

The training/test split was generated non-deterministically between runs.

FIX Added an explicit ordering clause to the query to make sampling reproducible.

Document Classification Taxonomy

Each discovered document is classified against a taxonomy of categories relevant to municipal & utility documents, each with its own keyword set:

- Budgets
- Capital Improvement Plans
- Engineering Reports
- Rate Studies
- Meeting Minutes
- + related categories

Manual cross-verification

A manual review process catches cases where the model's predicted category doesn't match a document's actual content, keeping the taxonomy trustworthy as it scales.

IMPACT

What This Made Possible

18M

curated rows, down from tens of millions of raw, noisy crawl rows

5 signals

combined into a single weighted priority score for every URL

1 pipeline

iterative human-in-the-loop loop generating ground truth at scale

Result: faster downstream queries, a cleaner training signal, and a repeatable process for turning raw crawl noise into trustworthy municipal & utility data.