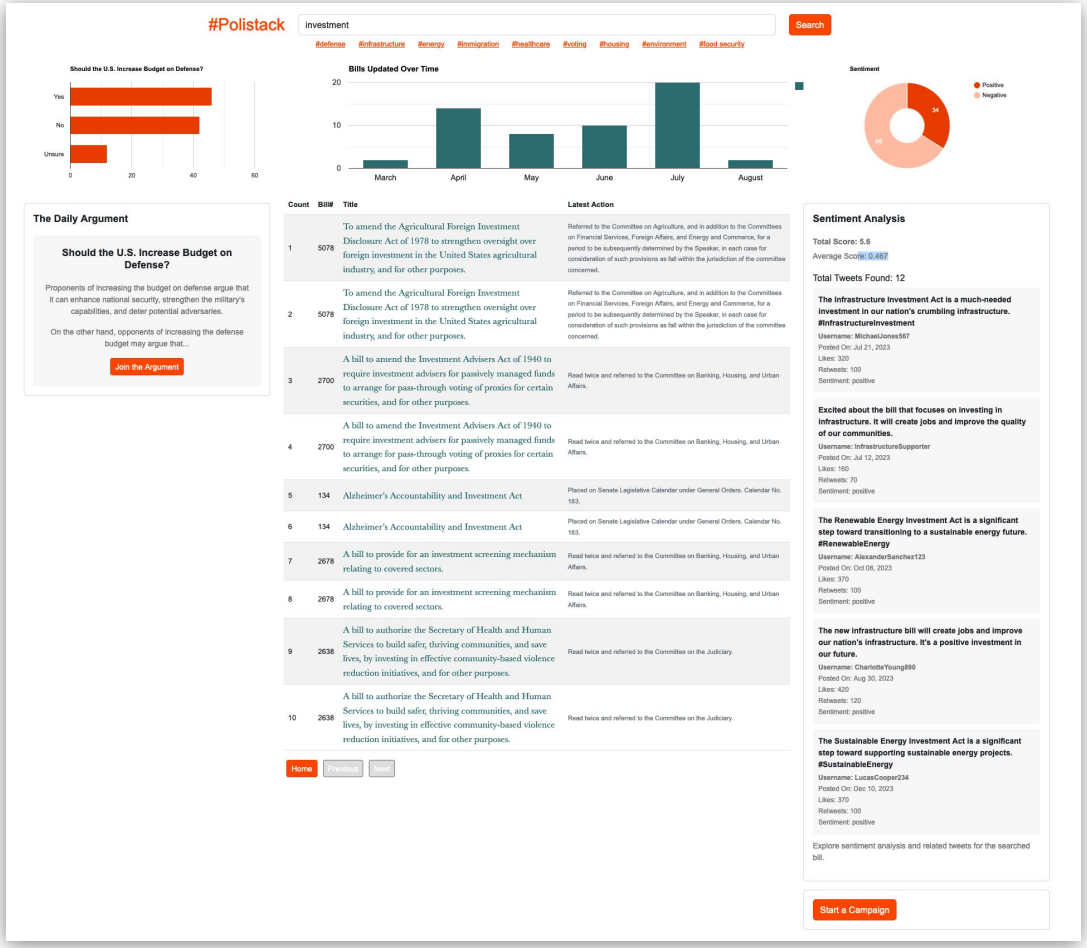
TEAM100 - Polistack: Agenda

1. Problem Description - *Liam, 1 min*
2. Data Exploration - *Liam, 2 min*
3. Data Pre-Processing and Representation - *Arjun + Rishabh, 4 min*
4. Models Used - *Arjun, 3 min*
5. Model Performance and Results - *Christian, 3 min*
6. System Architecture / Demo - *Rishabh, 10 min*

1. Problem Description

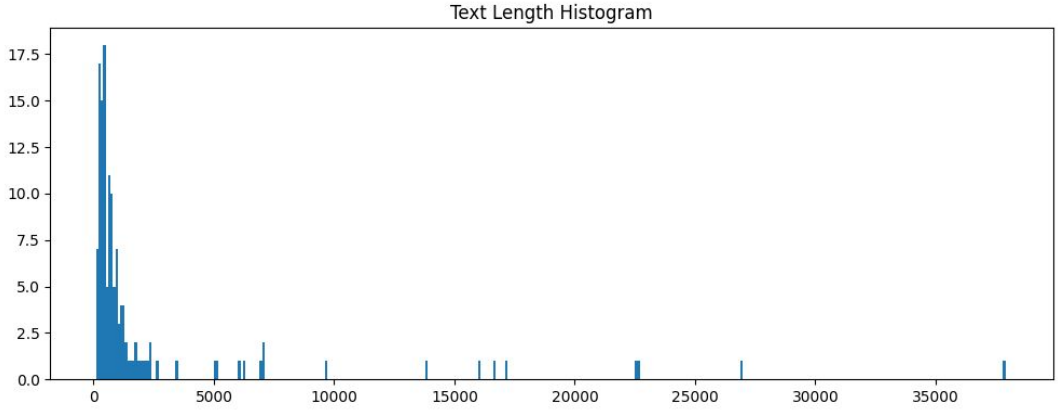
Understanding what policies impact you and your community is difficult.

Polistack helps you discover policies that impact your life, and gives you tools to influence outcomes.

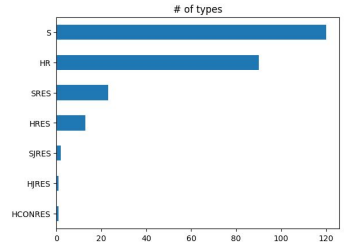


2. Data Exploration

- **Bills** from Congress.gov
- **Sentiment** from Twitter
- Data is interesting, clean, complete.
- Opens the door to more interesting application development.
- **MEAN:** 2,292
- **STD_DEV:** 5,351
- **MIN:** 57
- **MAX:** 37,856




```
RangeIndex: 250 entries, 0 to 249
Data columns (total 10 columns):
 #   Column                  Non-Null Count  Dtype
---  ------                  --------------  -----
 0   congress                250 non-null    int64
 1   latestAction            250 non-null    object
 2   number                  250 non-null    object
 3   originChamber           250 non-null    object
 4   originChamberCode       250 non-null    object
 5   title                   250 non-null    object
 6   type                    250 non-null    object
 7   updateDate              250 non-null    object
 8   updateDateIncludingText 250 non-null    object
 9   url                     250 non-null    object
```



Up next: Rishabh

3. Data Preprocessing & Representation

Project Overview:

- Django project: congress.gov API, MongoDB storage.
- 8000 bills, associated tweets.

Data Handling:

- Preprocessing ensures quality, analysis efficiency.
- Clean, transform, prepare JSON data.

Efficient Retrieval:

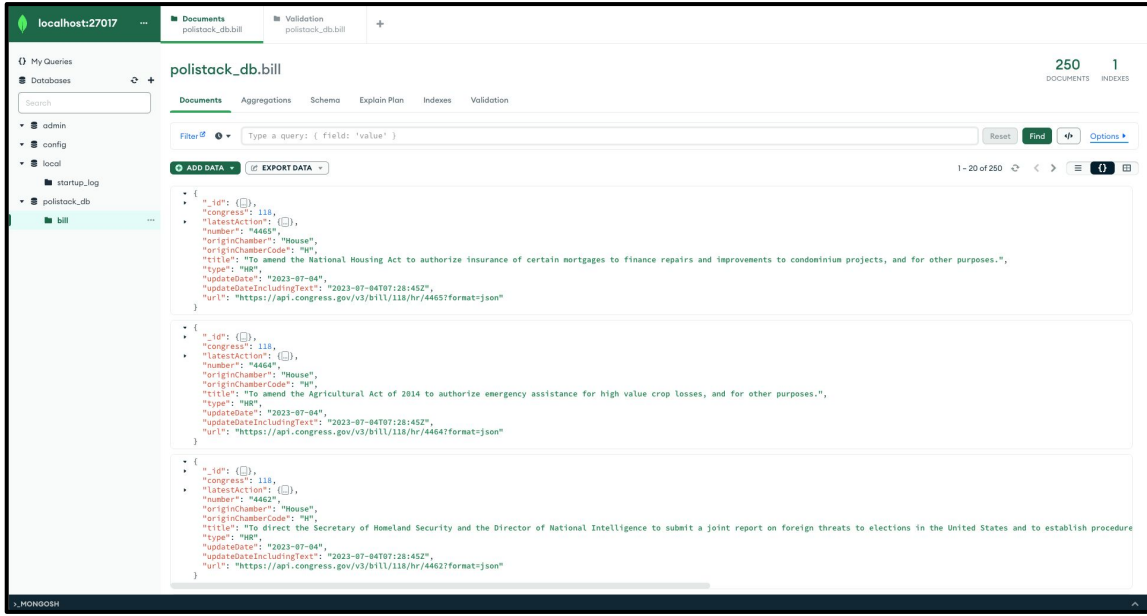
- Indexing accelerates search, retrieval.
- Applied to bills, tweets for optimized queries.

MongoDB Advantage:

- MongoDB for JSON, scalability.
- Schema designed for performance.

User Engagement:

- Intuitive data display on frontend.
- Users explore bills, linked tweets seamlessly.



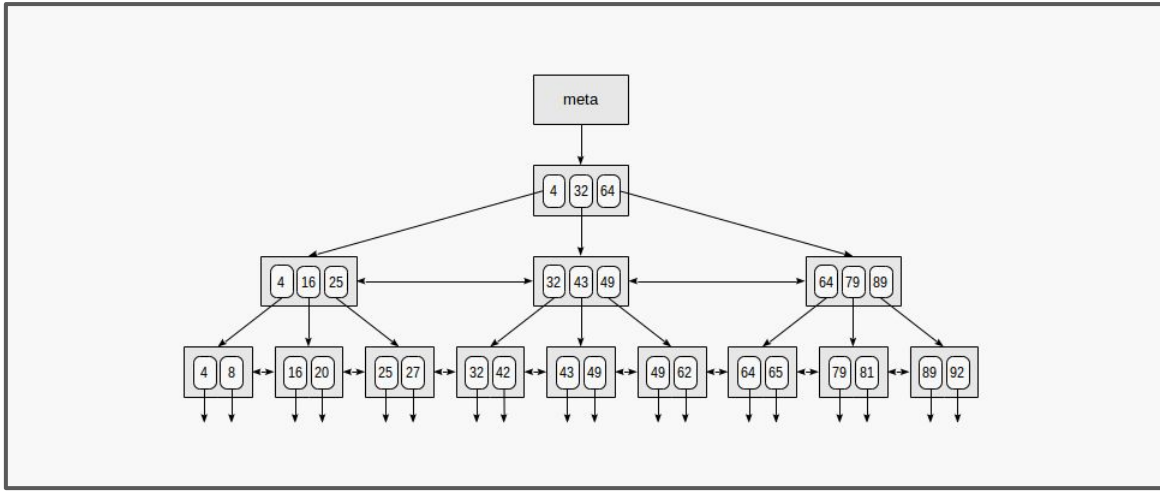
Query Optimization:

- Handle extensive data with techniques.
- Aggregation pipelines, optimized queries, caching.

Up next: Arjun

4. Models Used

- Native text indexing
- B-Tree
- Case folding
- Tokenization
- Language-specific stemming
- Language-specific stop word removal
- Weighting fields
- Document scoring
  - $totalScore_{document} = \sum_{f} matches_f * weight_f$



Up next: Christian

5. Model Performance and Results

- Starting Small
- Showcase Queries on www.polistack.com

- List of Found Stop Words:
  - Act
  - Department
  - Infrastructure
  - Defense

Discounted Cumulative Gain

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q
1	Nation Defense	Expected	3	3	1.892789261	1	0.8613531161	0.7737056145	0.7124143742	0.6666666667	0.6309297536	0.3010299957	12.83888878		NDCG		# Returned
2		Actual	3	3	1.892789261	1	0.4306765581	0.7737056145	0.3562071871	0	0.3154648768	0.3010299957	11.06987349		0.8622142992		617
3																	
4	Health Infrastructure	Expected	3	3	1.892789261	1.5	1.292029674	0.7737056145	0.7124143742	0.6666666667	0.6309297536	0.6020599913	14.07059534				
5		Actual	3	3	1.892789261	1	0.4306765581	0.7737056145	0.3562071871	0.3333333333	0	0.3010299957	11.08774195		0.7880080185		520
6																	
7	Voting Rights	Expected	3	3	1.892789261	1.5	0.8613531161	0.7737056145	0.7124143742	0.6666666667	0.3154648768	0.3010299957	13.0234239				
8		Actual	3	3	0.6309297536	0	0	0.3868528072	0	0	0.3154648768	0	7.333247438		0.5630813749		193
9																	
10	Green Energy	Expected	3	3	1.892789261	1	0.8613531161	0.7737056145	0.7124143742	0.6666666667	0.6309297536	0.6020599913	13.13991878				
11		Actual	3	3	1.892789261	1	0.8613531161	0.3868528072	0.3562071871	0	0.3154648768	0.6020599913	11.41472724		0.8687060729		126

Up next: Rishabh

6. System Architecture / Demo

Backend:

- Django
- Django Debug Toolbar
- Requests
- PyMongo
- Gunicorn
- MongoDB

Frontend:

- jQuery
- JavaScript
- HTML
- CSS

UI Framework:

- Django Bootstrap 5

Deployment:

- Vercel (Used for deploying the Django app)
- GitHub