

Sprint Review 4

Thursday, 29th October



Hello Again 🖐️✂️



ANA



ELLIE



JAMIE

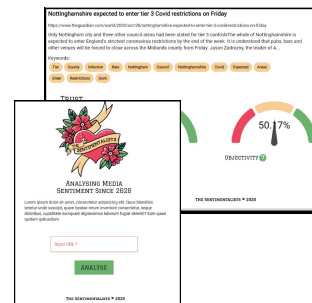


JYOTSNA



...and our first MVP was born!

- Creation of the App TheSentimentalists.github.io
 - Scans the content of a News Article and provide insights of sentiment / intent behind them
- Main Features:
 - URL input by the user
 - Article Details (Title, Summary, Keywords / Tags)
 - Sentiment Analysis (Polarity, Subjectivity)
 - Website Credibility (Score, Category, Source) via API call
- Main technology used:
 - React / JS ; AWS (lambda); Python & Rule based NLP methods



What we did ✓

- Reviewed priorities and defined the 'final features' for our MVP

The collage illustrates the development process for a sentiment analysis MVP. It includes:

- A GitHub repository for 'Simple Poll' with a list of contributors and a commit history.
- A wireframe of a news article summary titled 'UK credit risk something...' with sections for 'The summary of the article', 'Keywords', and three gauges for 'Credibility', 'Polarity', and 'Subjectivity'.
- A screenshot of a sentiment analysis tool showing a 'Credibility Score' of 82.5% on a gauge.
- A Trello board with task cards for frontend and backend development, including 'Add CORS OPTION method to analysis', 'Create Results Header component', 'Add keyboard support to URLInput component', 'Create 'PolarityScore' component', 'Create 'BiasScore' Component', and 'Return 'biascore' as part of response'.



What we did ✓

- ...and **implemented** those features!

- ✓ 'Subjectivity Score' component
- ✓ 'Polarity Score' component
- ✓ return article header/summary from API
- ✓ return article **keywords**
- ✓ frontend **error handling**
- ✓ 'Trust Score' component
- ✓ 'Results Header' component
- ✓ split 'Credibility Score' from 'Display Result'
- ✓ article 'Summary' **component**
- ✓ **refactor** 'Display Results'
- ✓ loading/processing function
- ✓ New **production** URL API (earlier test)
- ✓ return 'Trust Score' from API
- ✓ frontend **pipeline** automation



Front End

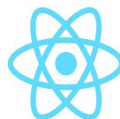
- Components we've built (tooltips, chips, etc)
- Frontend Pipe line built with Node.js on Github

Matt Hancock defends refusal to climb down on free school meals

<https://www.theguardian.com/education/2020/oct/26/matt-hancock-defends-refusal-to-climb-down-on-free-school-meals>

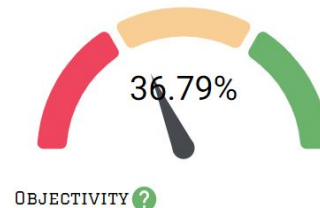
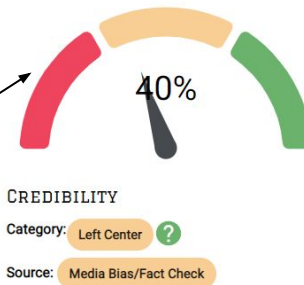
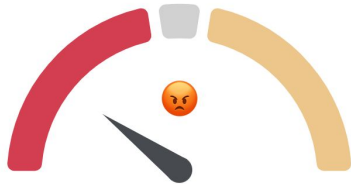
Matt Hancock has said the government does not need to extend free school meals beyond term times in England because local councils have been funded to support disadvantaged children instead. No 10 under growing pressure to U-turn over free school meals Read moreBut despite some reports predicting a partial climbdown over the refusal to fund free sc...

Keywords: Matt Hancock Support Climb Rashford Free Saying Refusal Meals Local
School Need Councils Defends

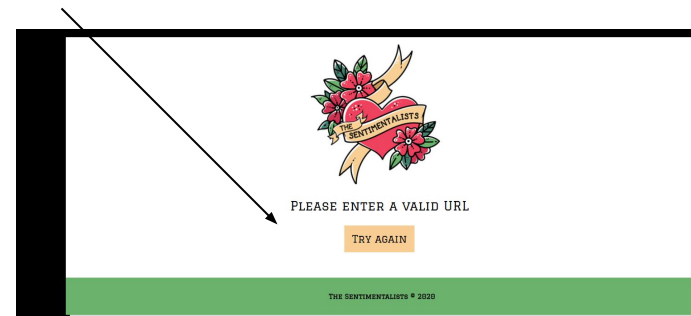


<Chip/>

<GaugeChart/>

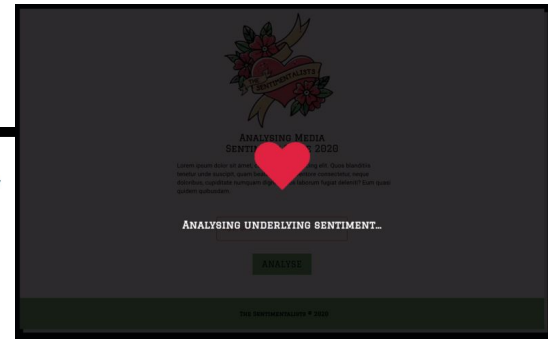
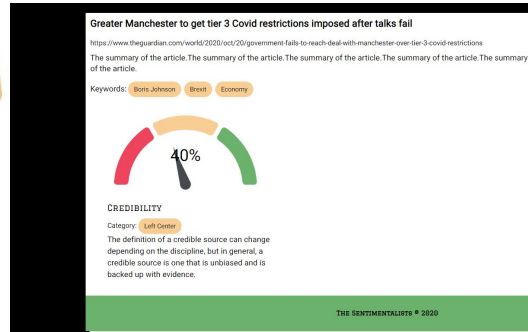
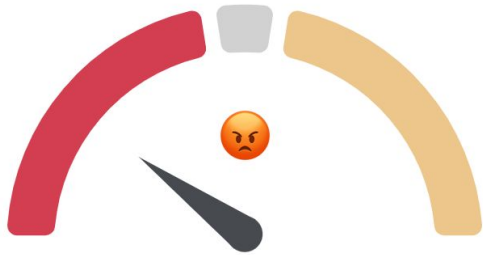


<ErrorBoundary/>

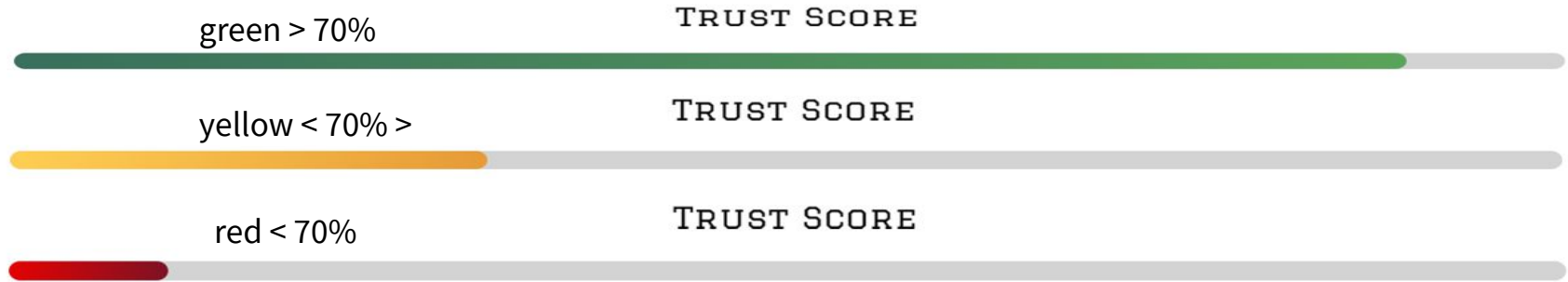


UX/UI

- Presenting each set of data in the most user-friendly format
- Creating meaningful visualisations
- Considerations about what will render in the event of an error
- Visual signifiers - what can users see while the backend is performing the analysis



Trust Analysis













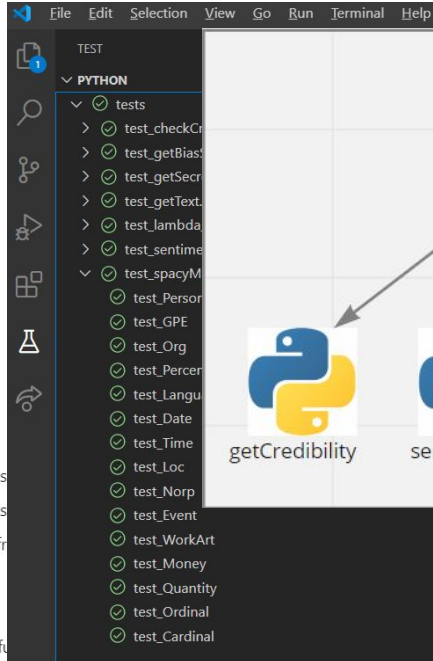
- **Progress bar** to indicate **Trustworthiness** of an article
- **Trust Score :**
 - **calculated** considering credibility, objectivity and polarity score
 - **Error message** in a scenario where value/s are unavailable
 - Python , Pytested code with Material-ui , React at the front end



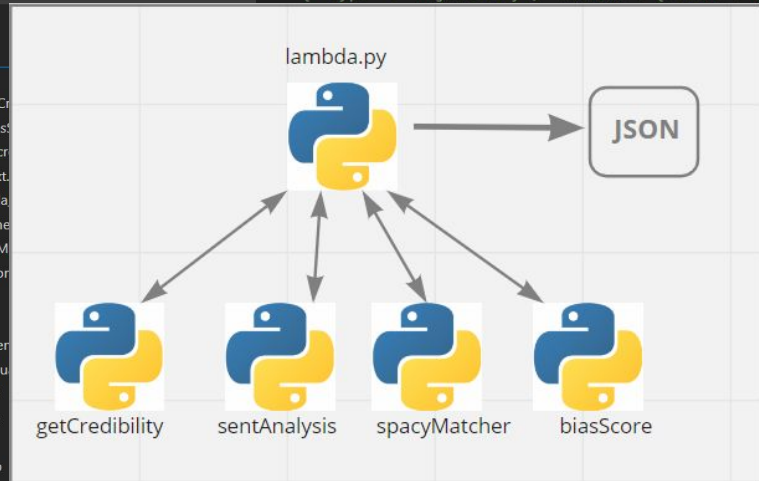
Back End

- Finalized Lambda_Function.py
- JSON returned to frontend
- Pytests validated
- Backend README updated

 Opened 22 other pull requests in 1 repository
[TheSentimentalists/SentimentalistsApp-Backend](#)
 Call to SpacyMatcher into Lambda_function.py
 getBiasScore wont be called if no POL or SUBJ
 merged bias score into lambda
 README updated + removed non code related files
 reversed subjectivity to objectivity score = $ABS(1 - s)$
 Changes in the JSON file sent from lambda to the fr
 commented test_spacyMatcher
 Spacy fixing en core web sm
 removed valURL and testValURL - done by lambda fu

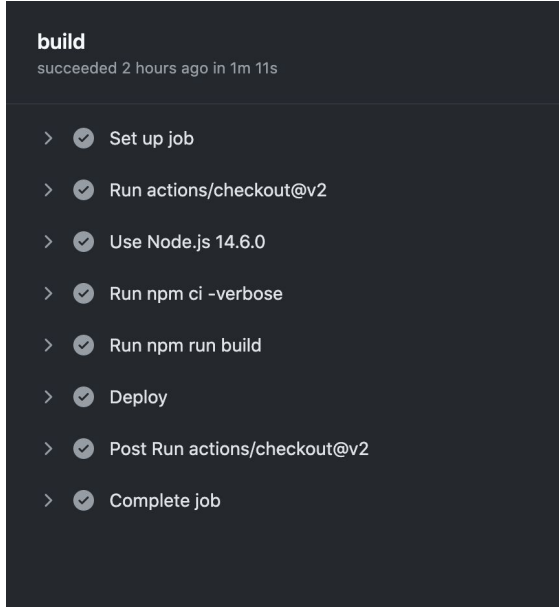


```
### JSON to return:
{
  'url': 'https://www.theguardian.com/world/2020/',
  'article': {
    'header': 'PM admits failings as England's Covid contact',
    'summary': 'Boris Johnson and his chief scientific ...',
    'keywords': ['Boris Johnson', 'Brexit']
  },
  'results': [
    { 'type': 'credibility', 'outcome': {'score': 100.0, 'source ...
    { 'type': 'polarity', 'outcome': {'score': 0.108126295001 ...
    { 'type': 'objectivity', 'outcome': {'score': 0.487846736596 ...
    { 'type': 'subjectivity', 'outcome': {'score': 0.67598528015 ...
    { 'type': 'PERSON', 'outcome': {'score': 0.0, 'source ...
  ]
}
```



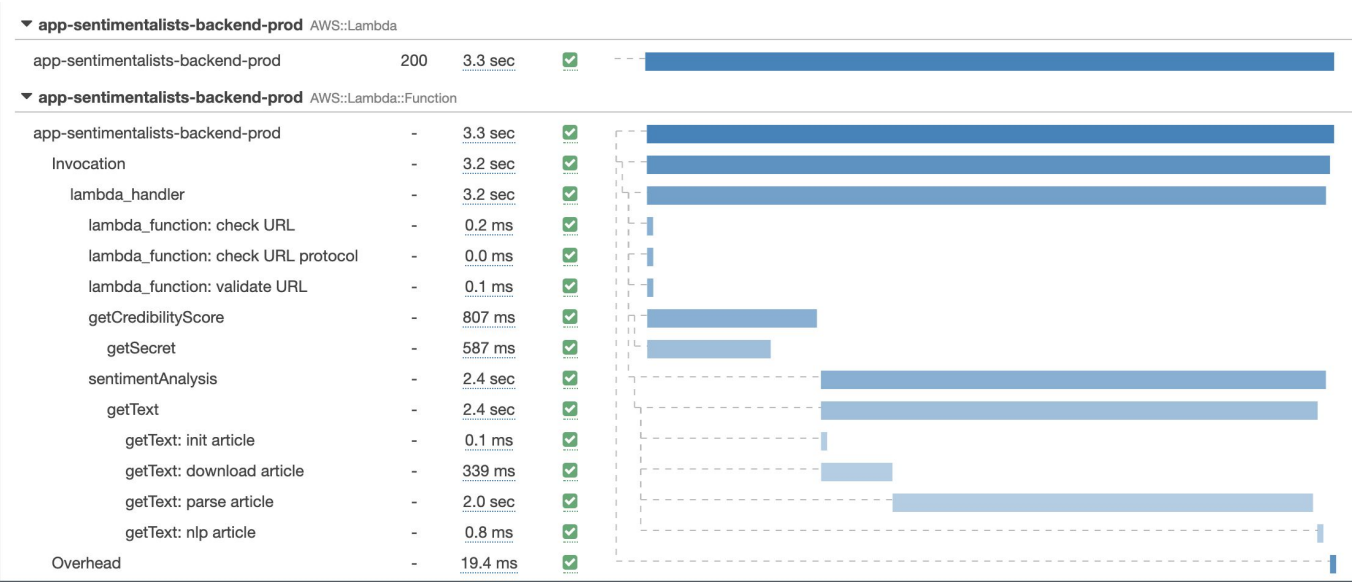
Pipeline Fun and Games

- Built the frontend pipeline!
- Regex Module and Docker Containers!
- NLTK Assets, and Lambda upload limits!
- ...and X-Ray Profiling!



Request Profiling

Using AWS X-Ray to profile requests and identify opportunities for improvement:



Final Touches 🏆

- Stay tuned for a few fixes and surprises... you'll see more on Tuesday!



Questions?

