ClassifAI:
embedding AI tools into survey operations

Mat Weldon, Jyldyz Djumalieva ,
Edward Jackson, Andy Banks
Strategic Technologies & Techniques Team

12 December 2024

Data Science Campus

Embedding AI tools into survey operations

Advancements in Large Language Models (LLMs) make them more suitable for text classification tasks in organisations. Assigning free text to categories is a common activity for National Statistical Institutes (NSIs), and currently a combination of manual, rules-based, and machine learning techniques are employed.

We have explored a Retrieval Augmented Generation (RAG) approach, involving the latest LLMs, to classify anonymised free text from labour market surveys to a Standard Industrial Classification (SIC).

Technical Approach

Semantic search using an embedding of SIC descriptions:

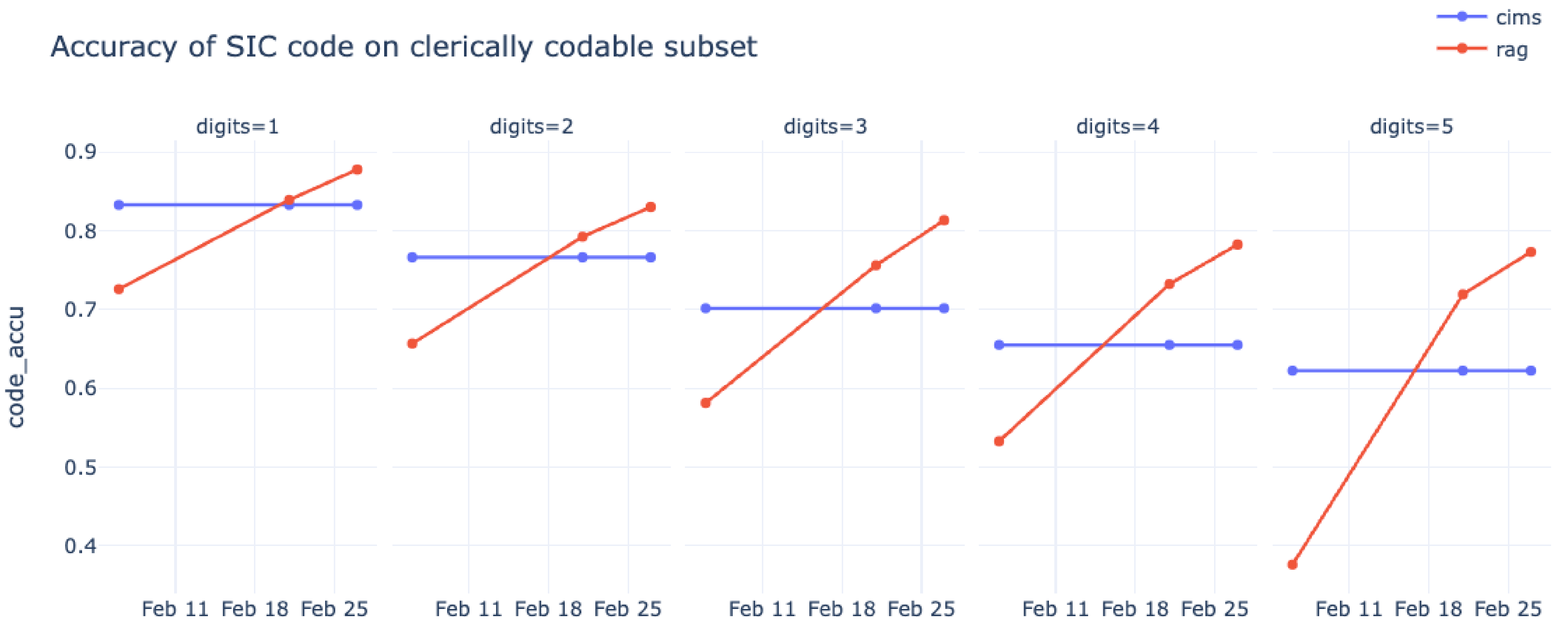
- Shortlist of candidates

Generative **large language model** to pick the best candidate:

- SIC allocation, with reasoning
- Follow-up question when unallocated

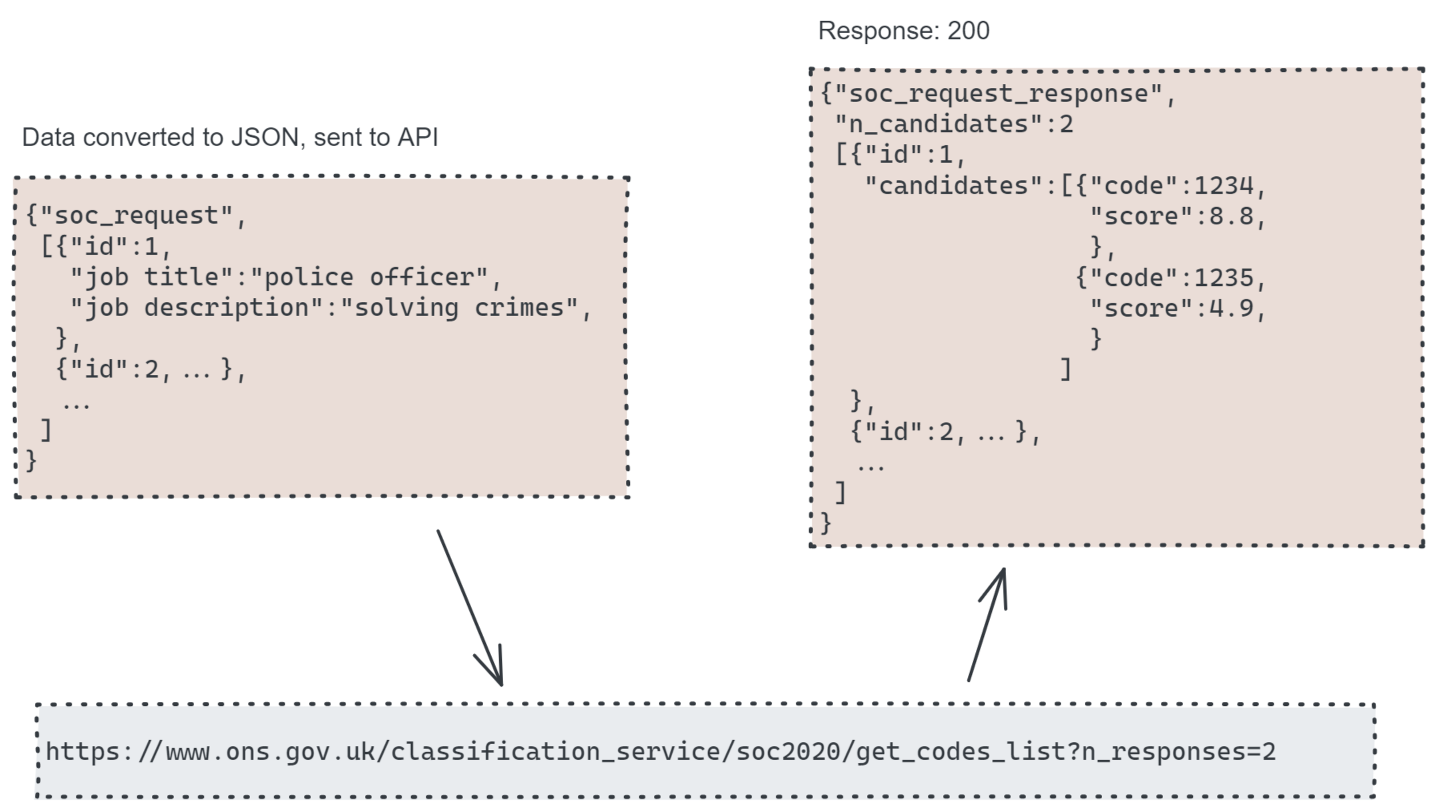
Assessing quality

- Initial results were promising, demonstrating a 5 and 11 percentage point improvement on existing approaches when classifying data to the 2-digit and 5-digit SIC levels



The idea: API framework for any classification

- Adapts to different codes and different data schemas
- Handles big workloads (millions of records)
- Production-ready (tested, documented)
- Secure (encrypted, authenticated, assured)
- **Externally available? (other NSOs, NESTA, Pensions Regulator, and DESNeZ already interested)**



Data converted to JSON, sent to API

```
{"soc_request",
 [{"id":1,
   "job title":"police officer",
   "job description":"solving crimes",
  },
  {"id":2, … },
   …
 ]
}
```

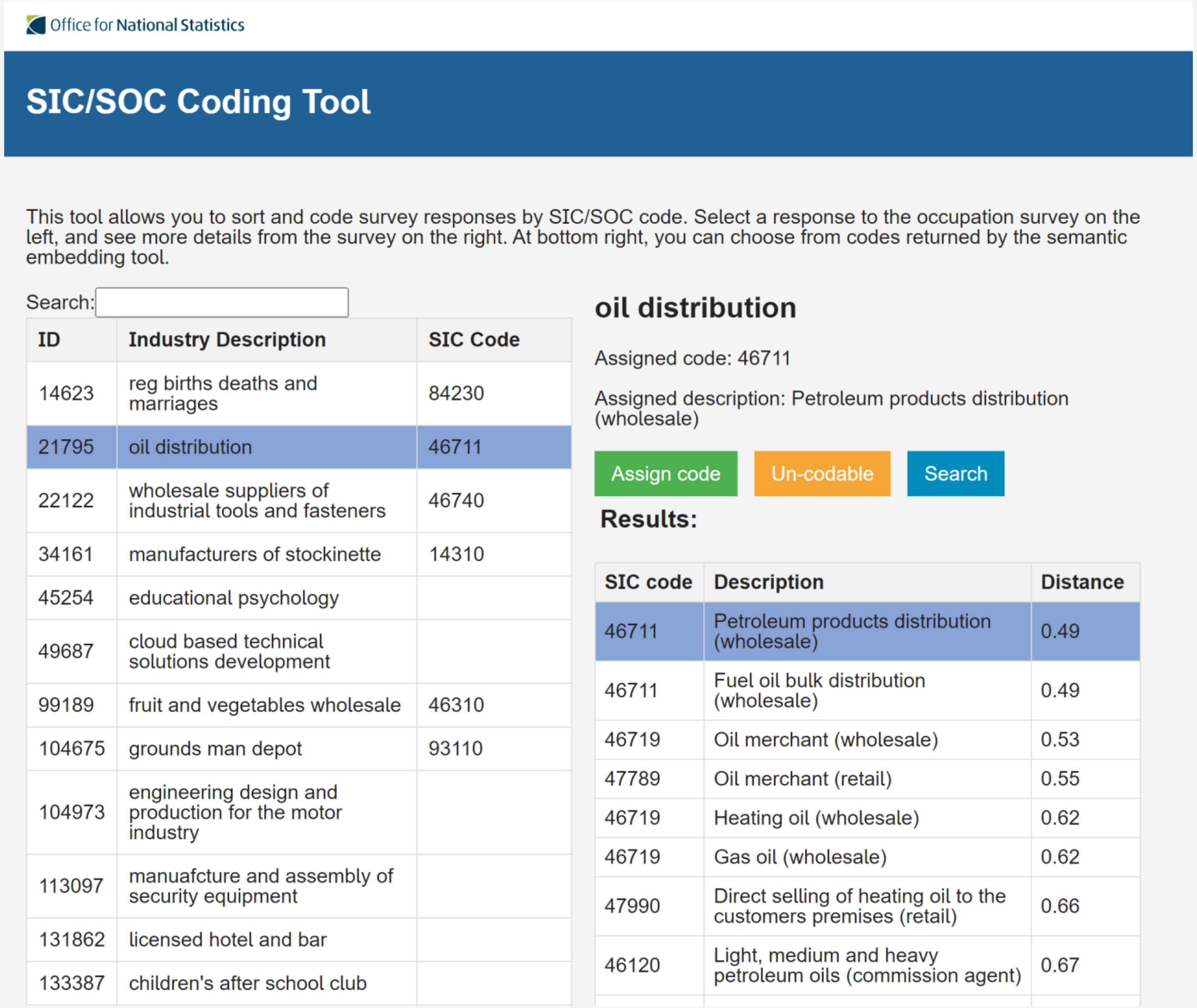
Response: 200

```
{"soc_request_response",
 "n_candidates":2
 [{"id":1,
   "candidates":[{"code":1234,
                  "score":8.8,
                 },
                {"code":1235,
                  "score":4.9,
                 }
               ]
   },
   {"id":2, … },
    …
 ]
}
```

```
https://www.ons.gov.uk/classification_service/soc2020/get_codes_list?n_responses=2
```

Free text coding tool

- Functioning web app being developed.
- Will reinforce model automatically



SIC/SOC Coding Tool

This tool allows you to sort and code survey responses by SIC/SOC code. Select a response to the occupation survey on the left, and see more details from the survey on the right. At bottom right, you can choose from codes returned by the semantic embedding tool.

Search:

ID	Industry Description	SIC Code
14623	reg births deaths and marriages	84230
21795	oil distribution	46711
22122	wholesale suppliers of industrial tools and fasteners	46740
34161	manufacturers of stockinette	14310
45254	educational psychology	
49687	cloud based technical solutions development	
99189	fruit and vegetables wholesale	46310
104675	grounds man depot	93110
104973	engineering design and production for the motor industry	
113097	manuafcture and assembly of security equipment	
131862	licensed hotel and bar	
133387	children's after school club	

oil distribution

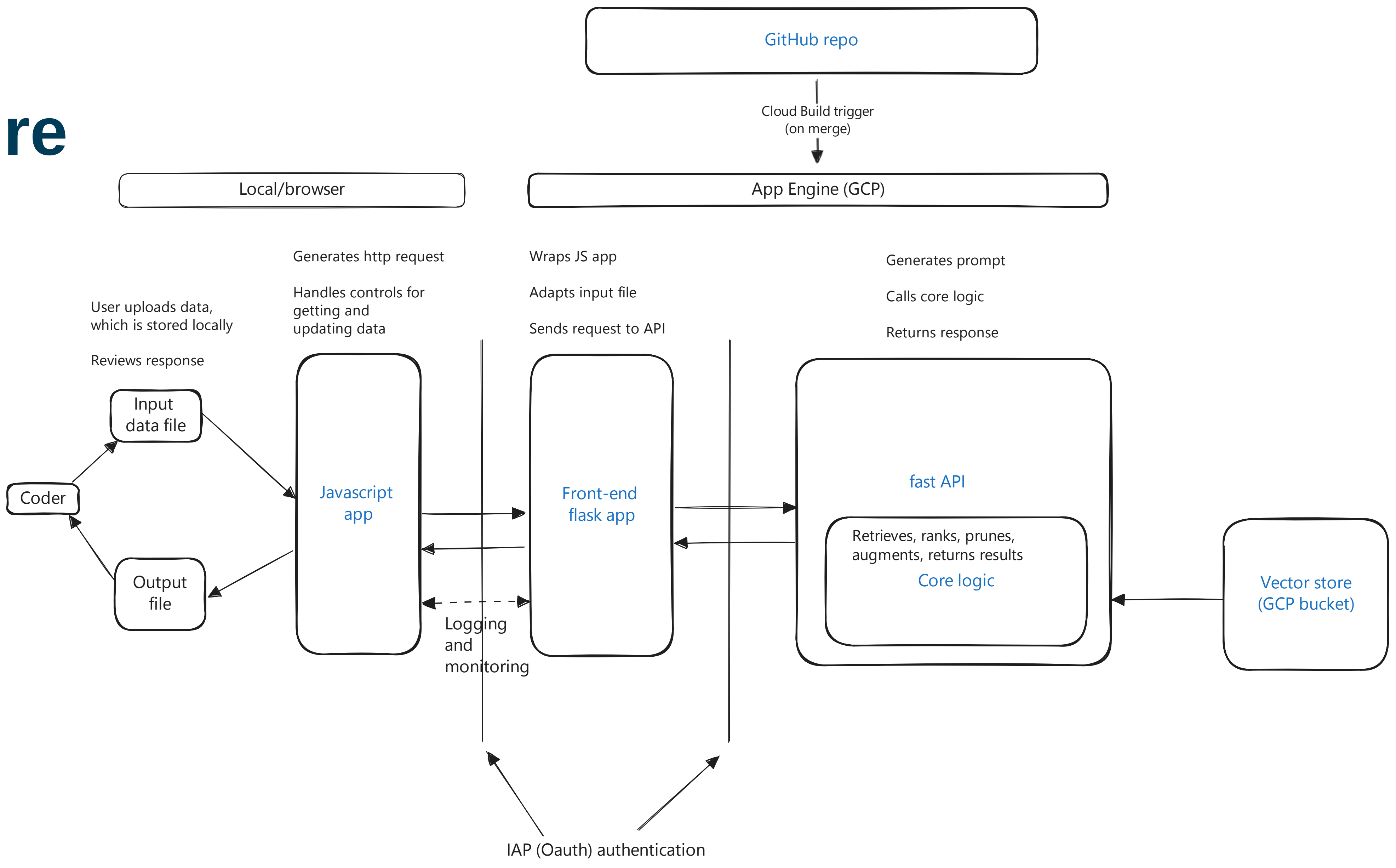
Assigned code: 46711

Assigned description: Petroleum products distribution (wholesale)

Results:

SIC code	Description	Distance
46711	Petroleum products distribution (wholesale)	0.49
46711	Fuel oil bulk distribution (wholesale)	0.49
46719	Oil merchant (wholesale)	0.53
47789	Oil merchant (retail)	0.55
46719	Heating oil (wholesale)	0.62
46719	Gas oil (wholesale)	0.62
47990	Direct selling of heating oil to the customers premises (retail)	0.66
46120	Light, medium and heavy petroleum oils (commission agent)	0.67

Service architecture



Uncertainty quantification

LLM methods give answers with good accuracy (and can be improved even more), but how can we ensure the answers are not overconfident?

Conformal prediction

- We have labelled data from a clerical coding feedback system
- For each prediction from the model, we can obtain the model's confidence score
- Confidence scores may be poor measures of uncertainty: the model may be overconfident
- Conformal prediction corrects for model overconfidence (or underconfidence) to obtain calibrated confidence sets with good coverage

Example

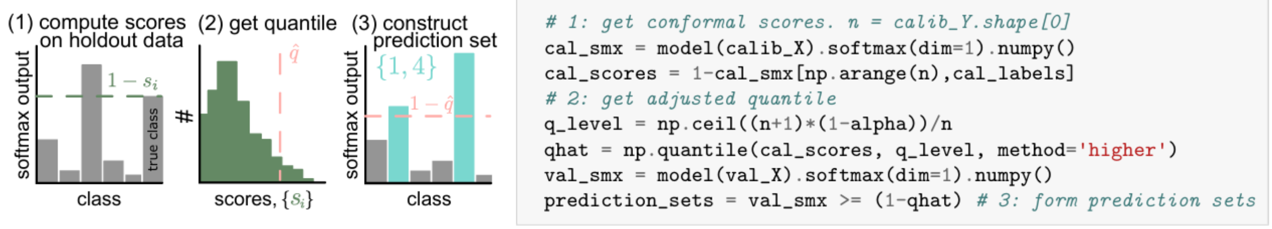
- The app requests a SOC code for "Administrator"
- The model gives several possible answers with confidence scores

  `{A: 0.5, B: 0.4, C: 0.09, D: 0.01, ...}`

- The calibrated model returns a 95% confidence set

  `{A: 0.5, B: 0.4, C: 0.09}`

- The set is sent to clerical coders to choose



```python
# 1: get conformal scores. n = calib_Y.shape[0]
cal_smx = model(calib_X).softmax(dim=1).numpy()
cal_scores = 1-cal_smx[np.arange(n),cal_labels]
# 2: get adjusted quantile
q_level = np.ceil((n+1)*(1-alpha))/n
qhat = np.quantile(cal_scores, q_level, method='higher')
val_smx = model(val_X).softmax(dim=1).numpy()
prediction_sets = val_smx >= (1-qhat) # 3: form prediction sets
```

Figure 2: Illustration of conformal prediction *with matching Python code.*

Principles

Security: classifAI will use best practices in auth, encryption, software assurance and continuous monitoring to address realistic threats and provide confidence

Efficiency: classifAI will save users time and effort to help them to be more productive, and will use cost effective services and computing resources to provide value for money

Quality: classifAI will produce accurate, trustworthy results with continuously monitored distributional metrics, and will have well-calibrated uncertainty to fail safely when data is insufficient.

Sustainability: classifAI will be designed to be easy to maintain, adapt, deploy, upgrade or replace to meet future needs

Challenges

- **Data:** Ensuring safety of sensitive data
- **Platforms**: Negotiating access to secure cloud platforms for web deployment
- **Consistency**: Even an improvement has downstream effects on statistics we have to understand before it can be used in production
- **Stakeholders**: Our immediate customers (Surveys) are enthusiastic, but NSO's are "tightly coupled" – every change needs approval from multiple business areas