hearsay

Challenges of Building a Domain-Specific Recommender System

Case Study

06/08/2023

Agenda

01 Introduction

02 Problem Statement & Background

03 Topic Modeling

04 Zero-Shot Text Classification

05 Imbalanced Dataset

06 Business Impact & Insights

Introduction

About Us

Zoltán Balogh

Senior Data Scientist

Veronika Palotai

Data Scientist

About Hearsay Systems

SaaS products for clients in the **financial services industry**

compliant communication across all social media platforms between financial services providers and their audience

ensure compliance

heavily regularized (U.S. market)

Problem Statement & Background

100+ leading financial firms, more than 200,000 users

Quality customer data from the past 10 years

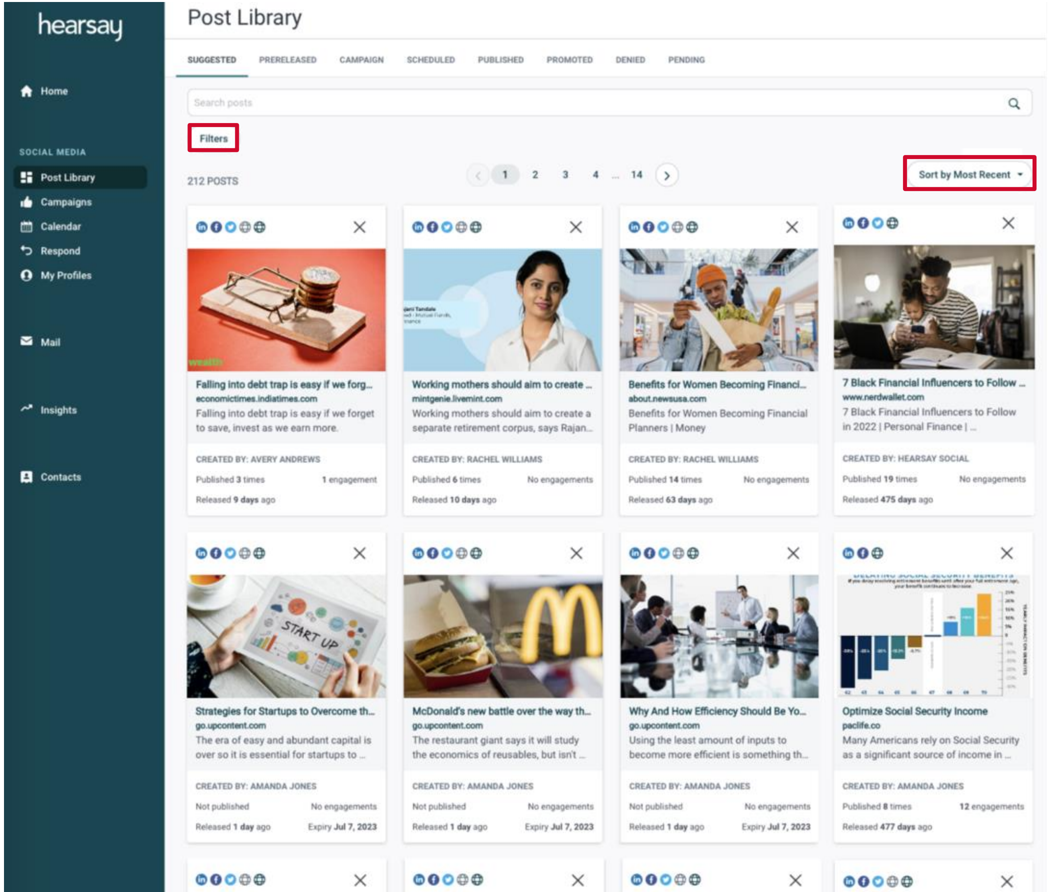
Data science use cases identified

Business Requirements

- Personalize the **Post Library** based on agents' interest and their network's preferences

⬇

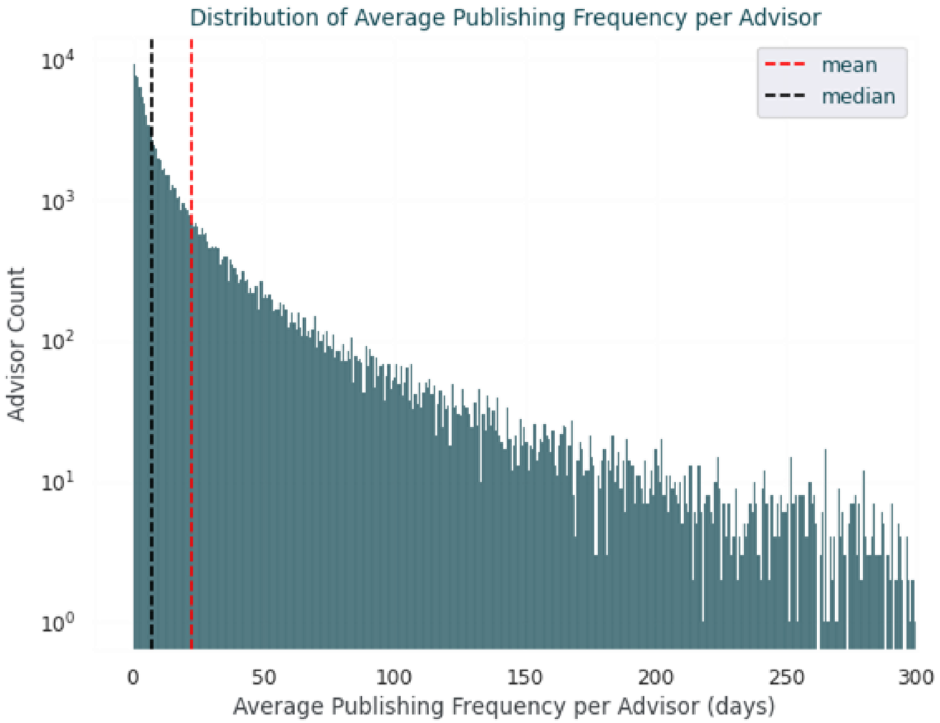
- ***Advisor interest*** - recommend posts similar to what the advisor has published recently.
- ***Audience interest*** - recommend posts similar to what the advisor's audience liked



Exploratory Data Analysis Highlights

How do advisors interact with the Post Library?

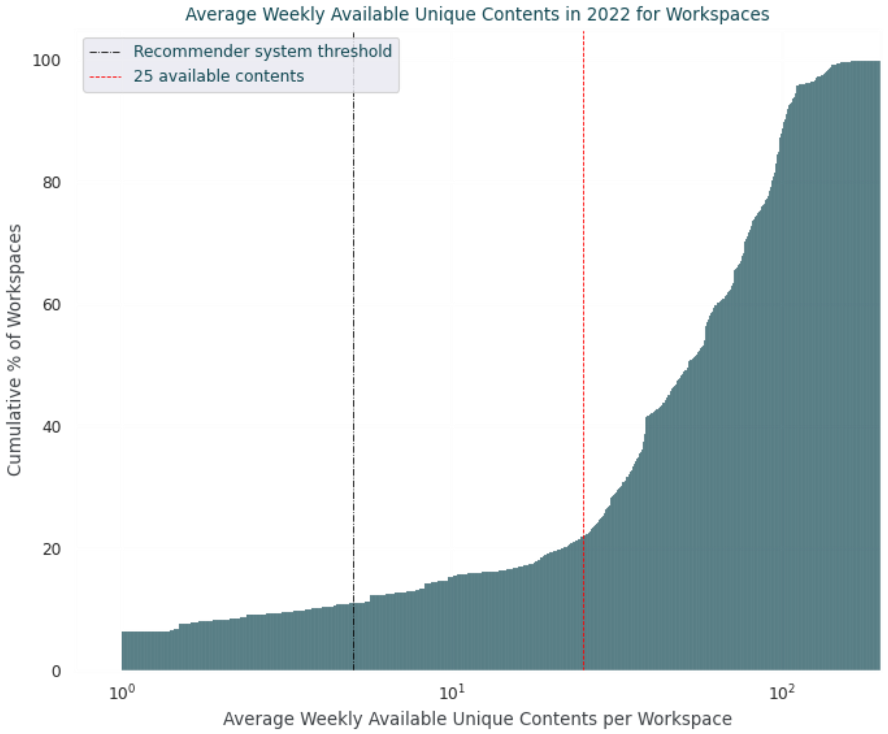
- 50% of advisors publish articles once a week on average



Exploratory Data Analysis Highlights

Are there enough articles to recommend from?

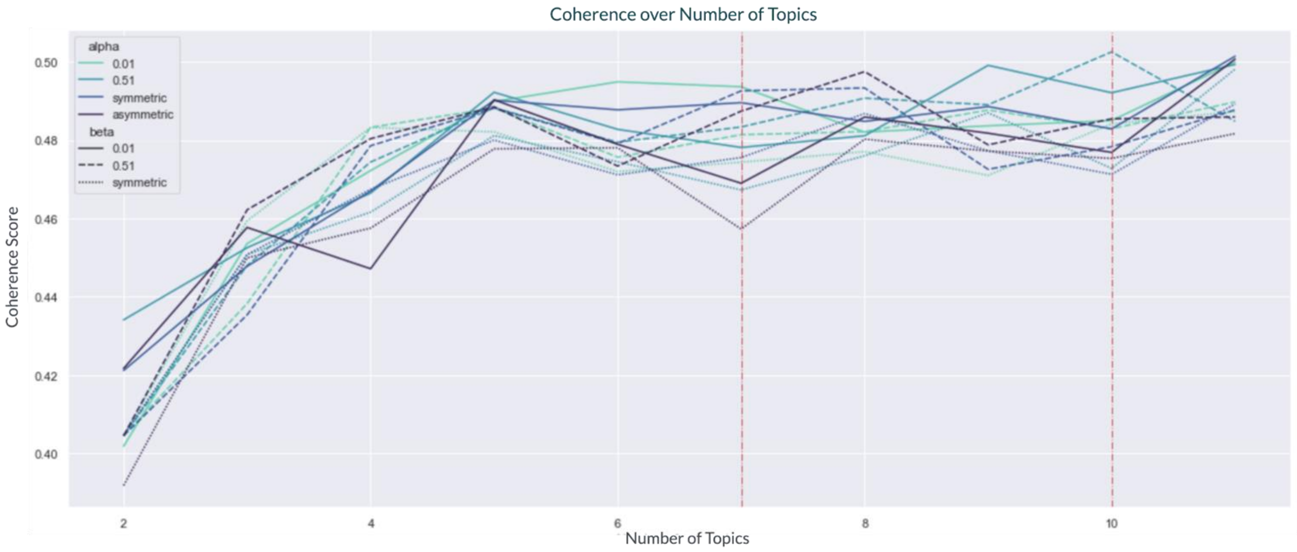
- ~12% of workspaces had on average 5 articles available on a weekly basis



Topic Modeling

LDA - A Probabilistic Approach

- Assumptions:
  - Document ➡ distribution of topics
  - Topic ➡ distribution of words
- Topic coherence to decide the optimal number of topics
- Tweaking hyperparameters ***alpha*** and ***beta***



Interpreting the Results

- Topics with the 10 most probable words in each

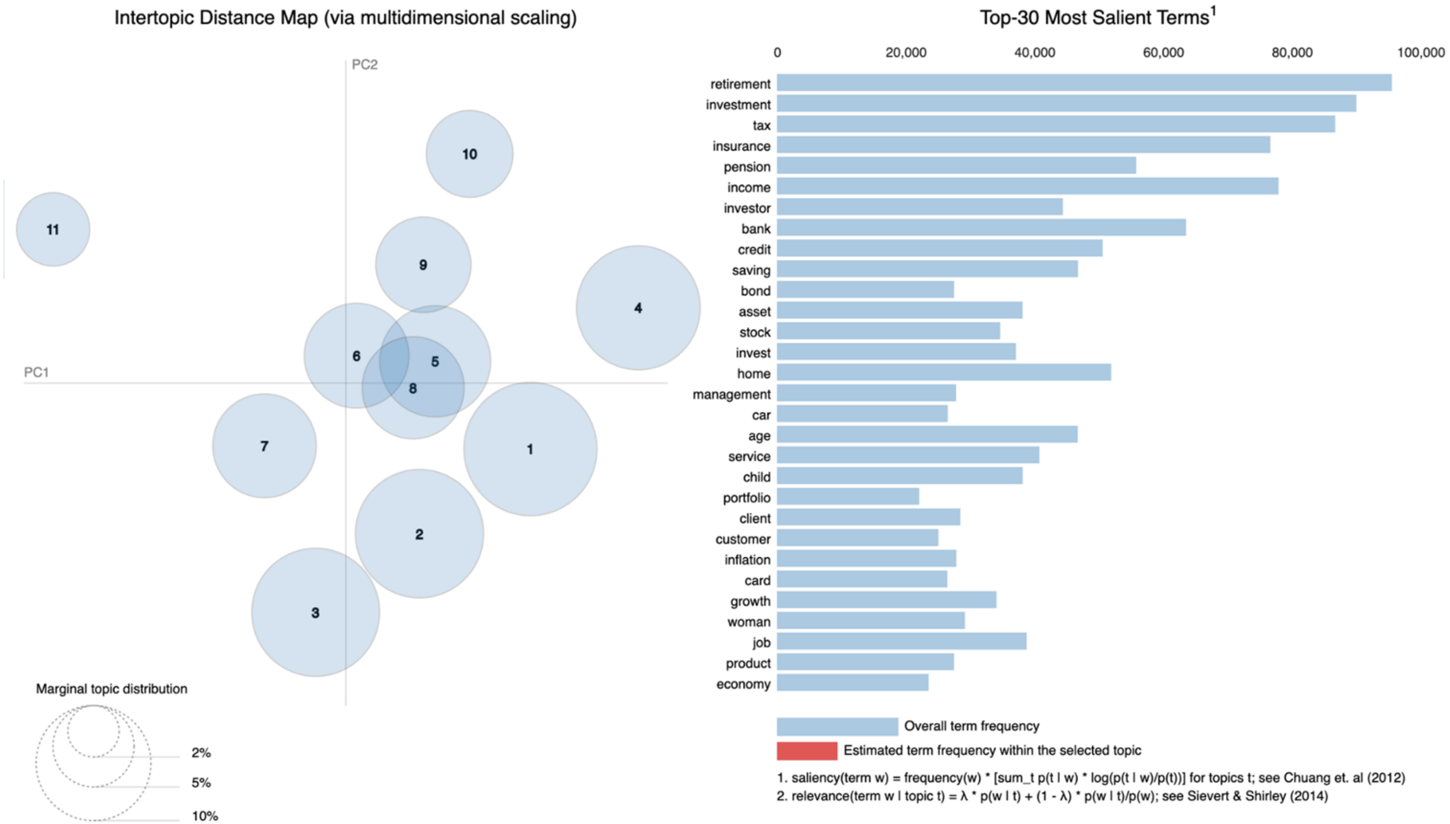
```
[(0,
  '0.015*"technology" + 0.015*"insurance" + 0.012*"news" + 0.009*"datum" + 0.009*"industry" + 0.008*"climate" + 0.006*"tech" + 0.006*"follow" + 0.006*"story" + 0.006
*"system"'),

 (1,
  '0.014*"job" + 0.010*"employee" + 0.010*"team" + 0.010*"learn" + 0.009*"career" + 0.009*"client" + 0.008*"opportunity" + 0.008*"experience" + 0.007*"ask" + 0.007*"s
upport"'),
```

⋮

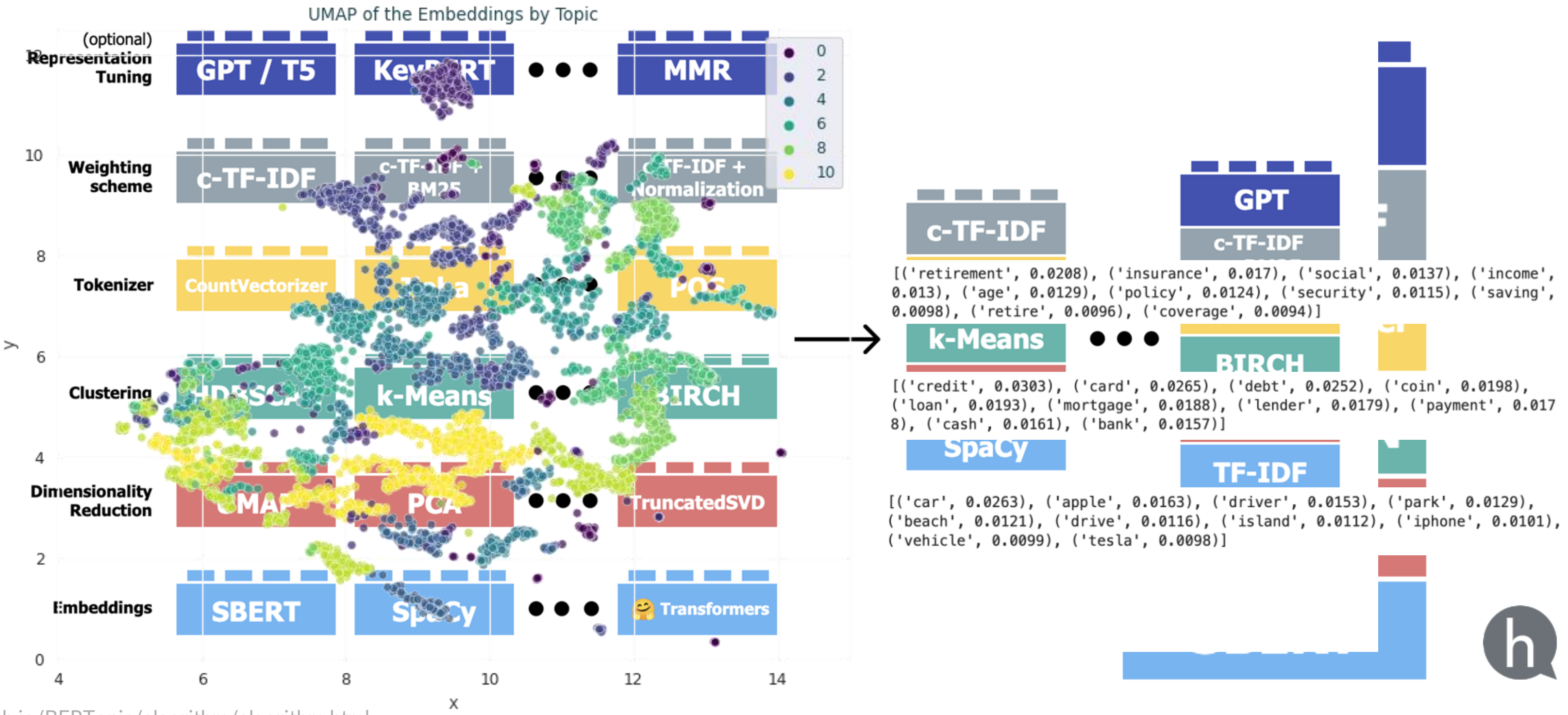
```
 (10,
  '0.058*"investment" + 0.038*"investor" + 0.026*"bond" + 0.025*"stock" + 0.024*"invest" + 0.020*"portfolio" + 0.020*"asset" + 0.012*"income" + 0.012*"equity" + 0.011
*"interest"'),
```

Visualizing the Output with PyLDAvis



Leveraging BERT for Topic Modeling

- Embeddings-based ➡ capture semantic information
- Modular, no preprocessing required

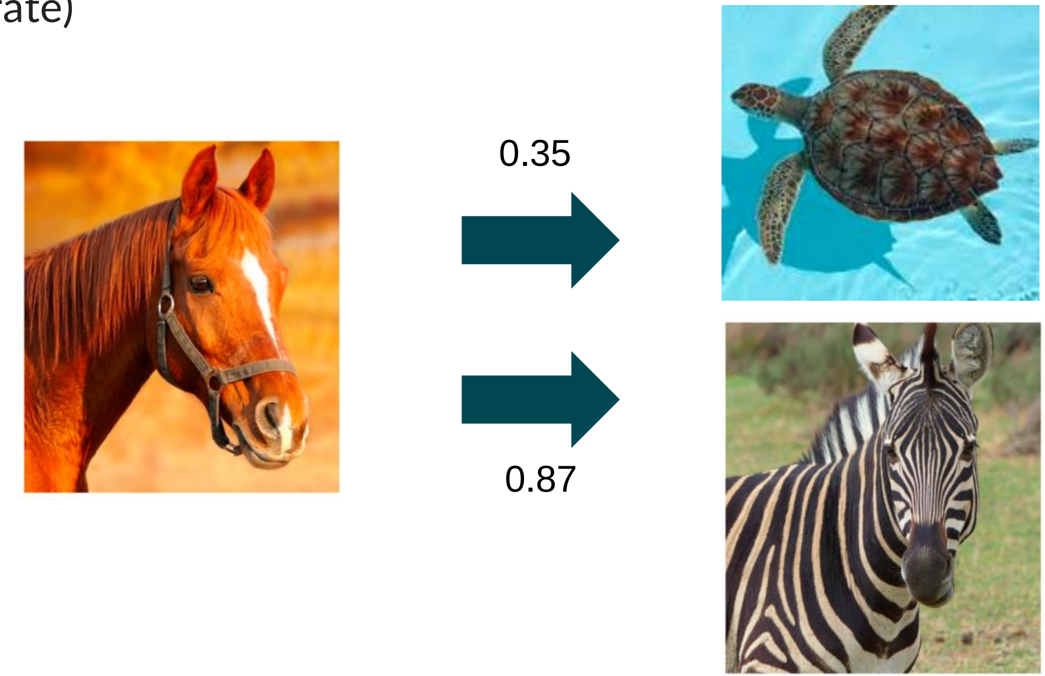


https://maartengr.github.io/BERTopic/algorithm/algorithm.html

Zero-Shot Text Classification

Zero-Shot Learning

- **Zero-Shot Learning (ZSL)** is a *“heterogeneous transfer learning”*, where a pre-trained deep learning model is used to generalize on a novel category of samples (feature and label spaces are disparate)



https://www.v7labs.com/blog/zero-shot-learning-guide

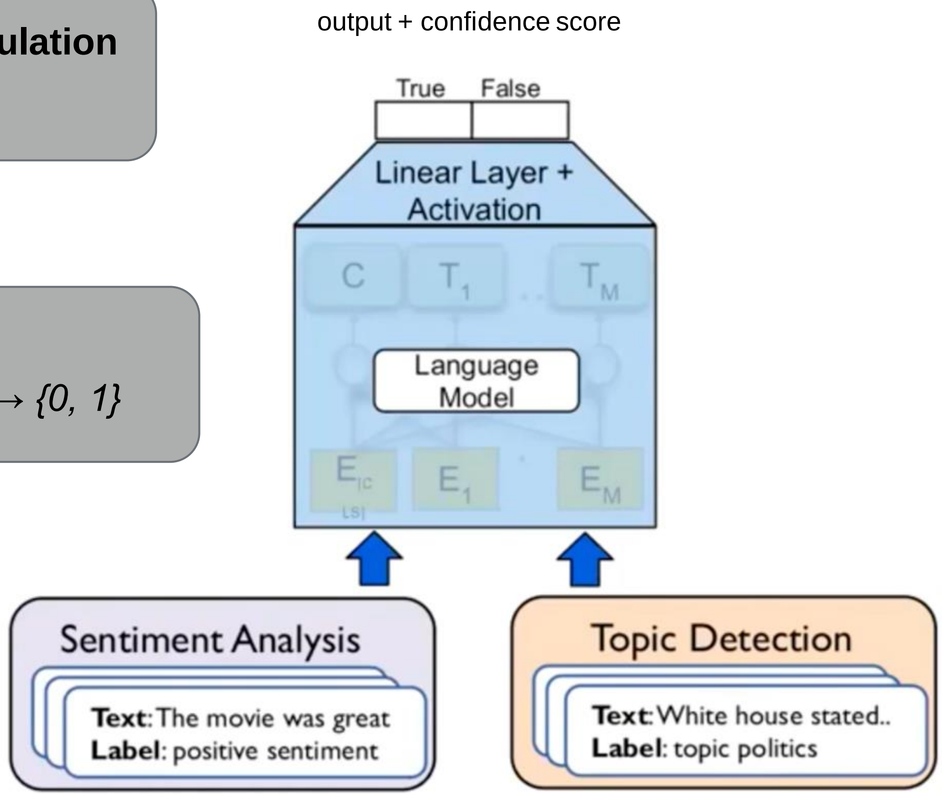
Task-Aware Representation of Sentences

Standard formulation

$f: \text{text} \rightarrow \{0, 1\}^P$

Our formulation

$f: \langle \text{task label}, \text{text} \rangle \rightarrow \{0, 1\}$



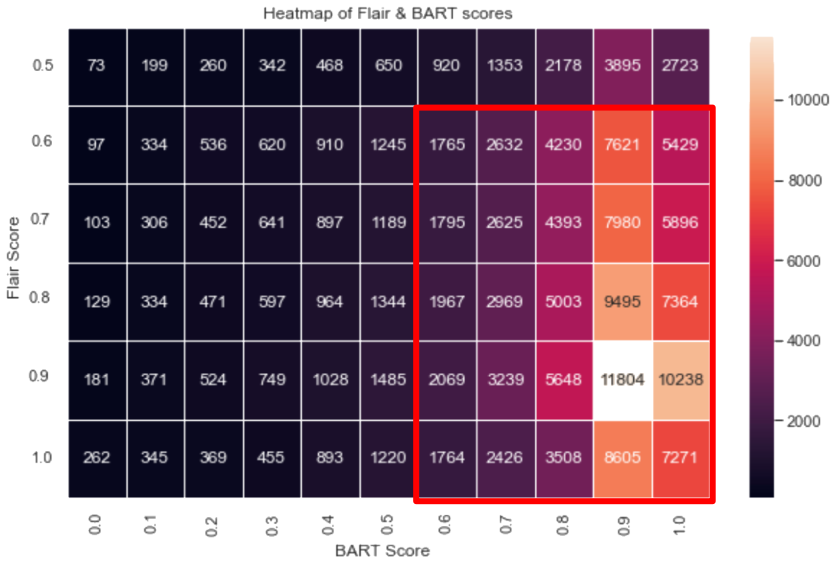
https://kishaloyhalder.github.io/pdfs/tars_coling2020.pdf

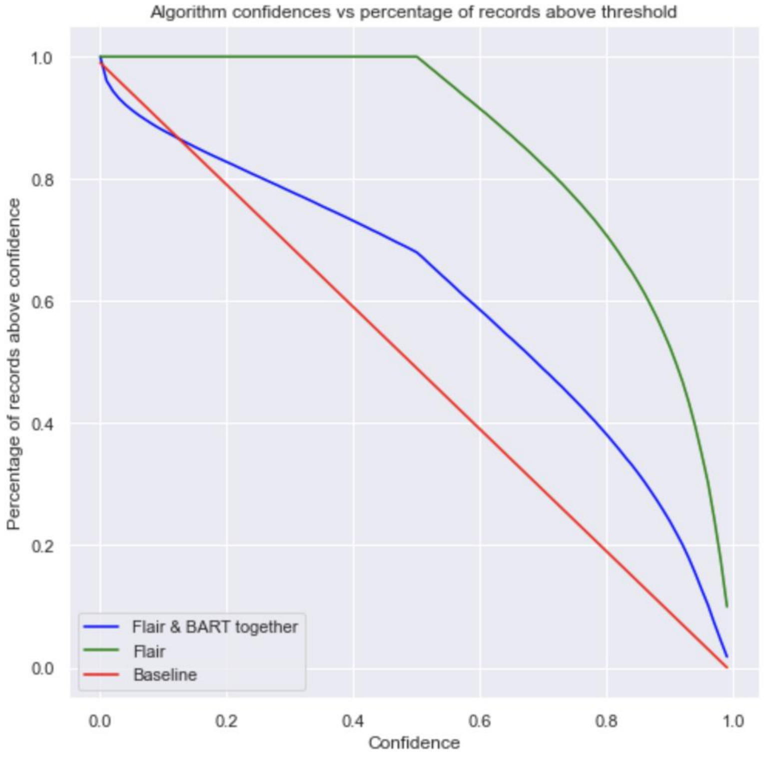
Zero-Shot Text Classification

- PoC locally on 12 cores, then on AWS
- **FlairNLP**: pretty fast and accurate result
  - Inference speed (12 categories, local): ~0.5s/article
  - Rate of successful predictions: ~65%
- **Facebook's BART** (Hugging Face transformers)
  - Inference speed (12 categories, local): ~20s/article
  - Inference speed (validating the output of FlairNLP, 1 category, local): ~0.5s/article

ZSTC Results

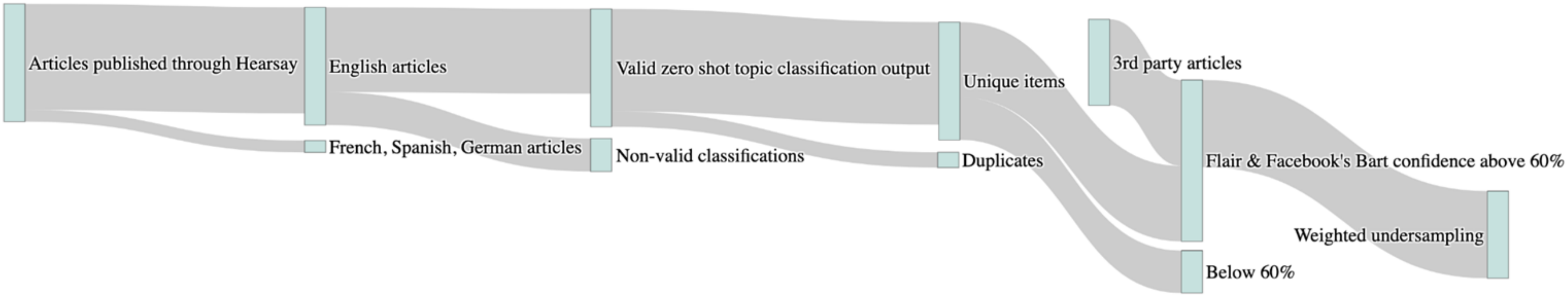
- Some of the articles fall into more than one category
  - The algorithms struggle to agree





The Data Requirement of the Project

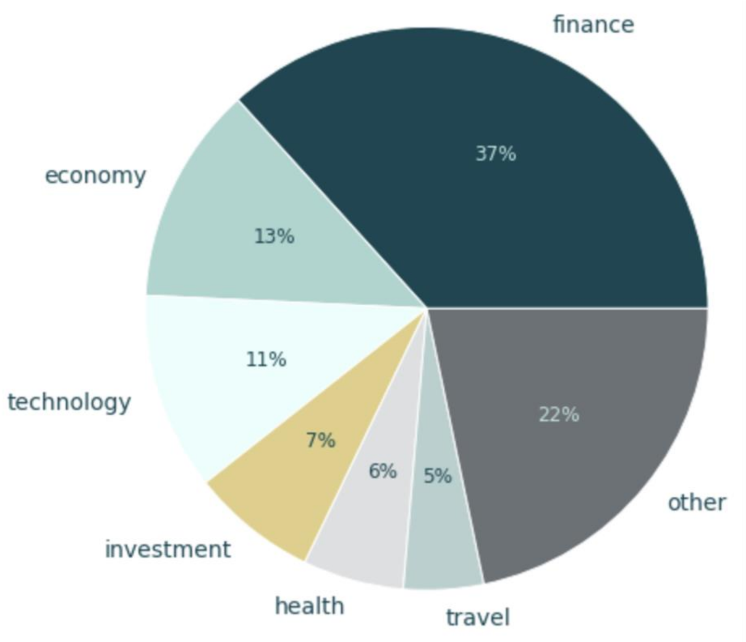
- Only approximately ⅓ of the articles were kept in the training set from the available articles
- A huge proportion of the raw data is unusable



Imbalanced Dataset

Imbalanced Dataset

- To reduce imbalances, we tried
  - Adding articles from 3rd parties
  - Text generation
  - Article crawling
  - Undersampling



Evaluation Metrics

- **Threshold metrics** (accuracy, F-measure etc.): Quantify the classification prediction errors
  - Specificity
  - G-mean
- **Ranking metrics** (ROC, PR etc.): Focuses on how effective the algorithms are at separating classes
  - ROC (Receiver Operating Characteristic) curve
  - PR (Precision-Recall) curve

$$\text{Specificity} = \frac{\text{TrueNegative}}{\text{FalsePositive} + \text{TrueNegative}}$$

$$\text{G-mean} = \sqrt{\text{Sensitivity} \times \text{Specificity}}$$

$$\text{Precision} = \frac{\text{TruePositive}}{\text{TruePositive} + \text{FalsePositive}}$$

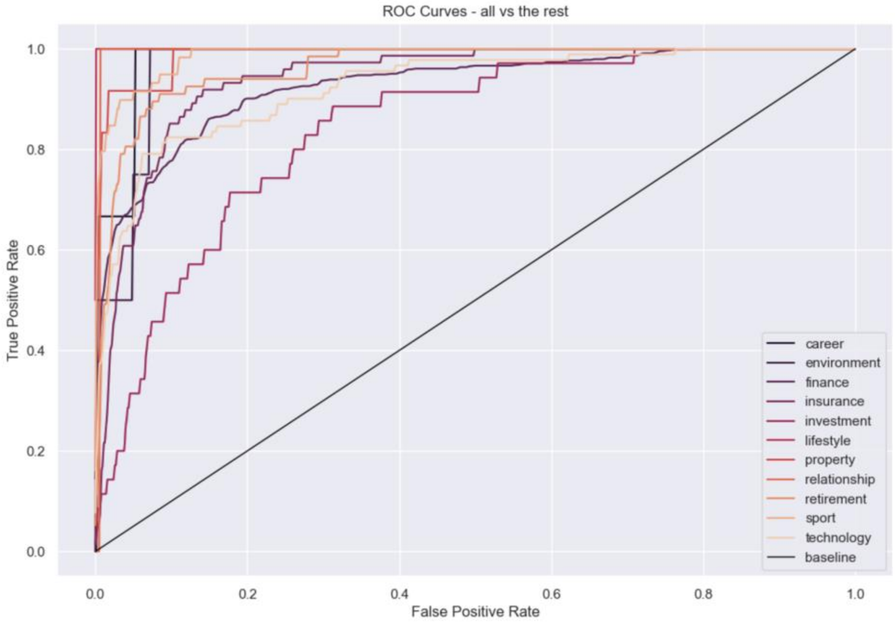
$$\text{Recall} = \frac{\text{TruePositive}}{\text{TruePositive} + \text{FalseNegative}}$$

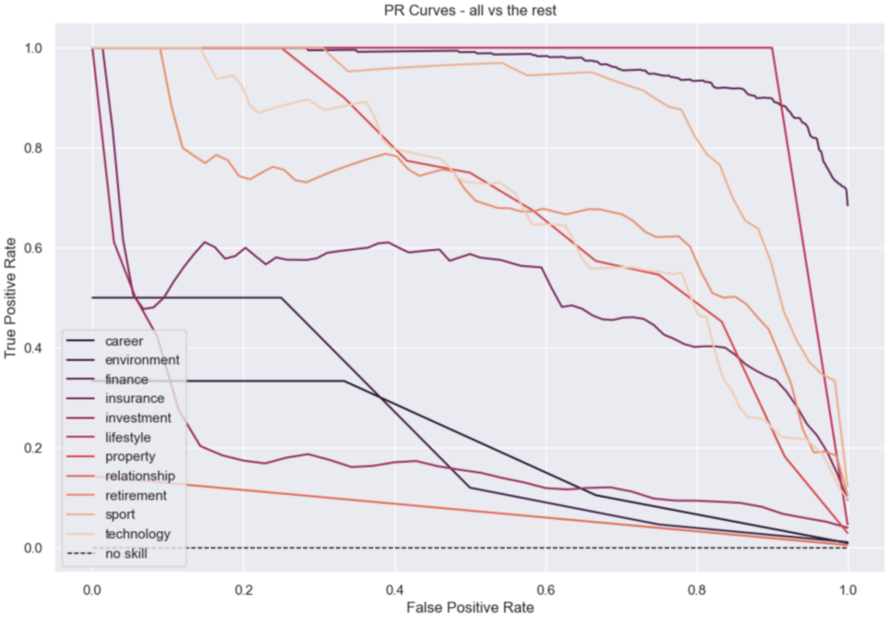
Metrics & Evaluation

XGBoost model:

- Accuracy: 0.7799
- ROC AUC: 0.7834
- PR AUC: 0.2992

- Train/test split
- K-fold cross-validation with stratification

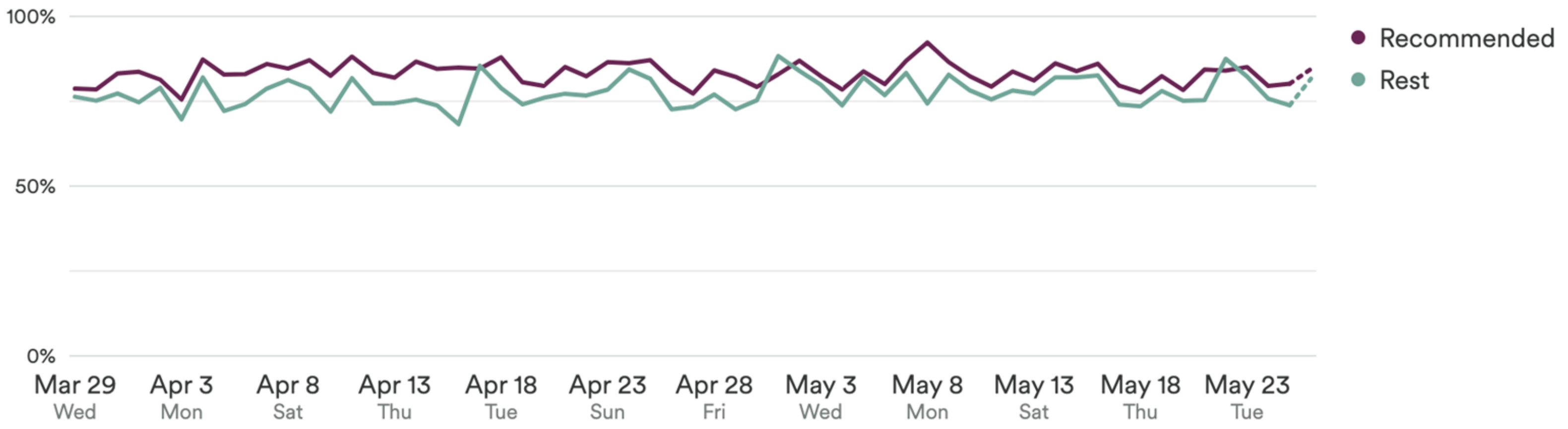




Business Impact & Insights

Post Publication Rate

- The rate of publishing a post after clicking on it is **~8% higher** for the recommended contents



Engagement Rate

- The engagement rate* of advisor and audience interest-based recommendations was **9% higher** than that of posts published from the library

$$^{*}Engagement\ rate\ of\ a\ post = \frac{\sum Inbound\ Engagement}{\sum Publishes}$$

hearsay

Questions?