ML Problems from CS 441 Students

CS 441 - April 25, 2024

[Problem]

by [name]

1. What you are solving and why – use visual examples if possible
2. What are the main challenges (ML and otherwise)
3. Specific questions that you would like addressed/discussed in lecture

This can be in 1-3 slides. Please be present in class on April 25 to talk through this and answer clarification questions. Practice explaining – it should take about 5 minutes for the initial explanation

Template: do not change – copy and paste

Agenda

Feedback and suggestions: tinyurl.com/441AppsSP24

1. Tinder for Ideas: Vignesh Srinivasakumar
2. Pebble (team formation platform): Matthew Lynch
3. Song Translation: Matthew Tang
4. Algophony (music generation): Ethan Chen
5. Patent text classification: Akshata Tiwari
6. Rideshare vehicle detection: Anagha Tiwari

Tinder but for ... Ideas?

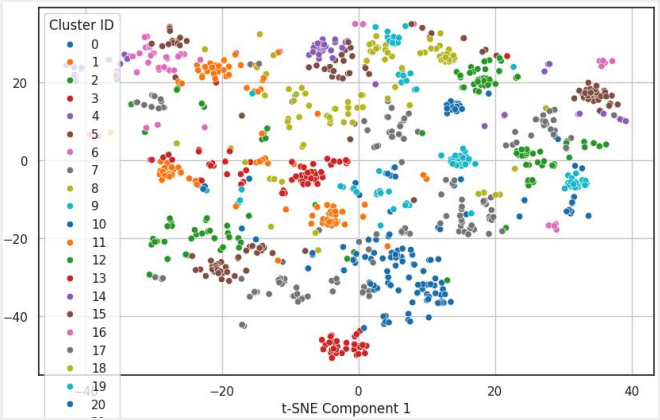
1. In an ideathon, how do you democratize voting and ranking? Build a Tinder-like interface.

2. Newly created problems :
   a. Recommendation - How do you equitably recommend ideas while keeping users engaged ?
   b. Ranking - How do you rank based on views and likes ?

~ Vignesh Srinivasakumar

Recommendation

- Show unique ideas each time.
- Use SentenceTransfomer to semantically cluster ideas.
- Pick the next idea from a random cluster



Ranking

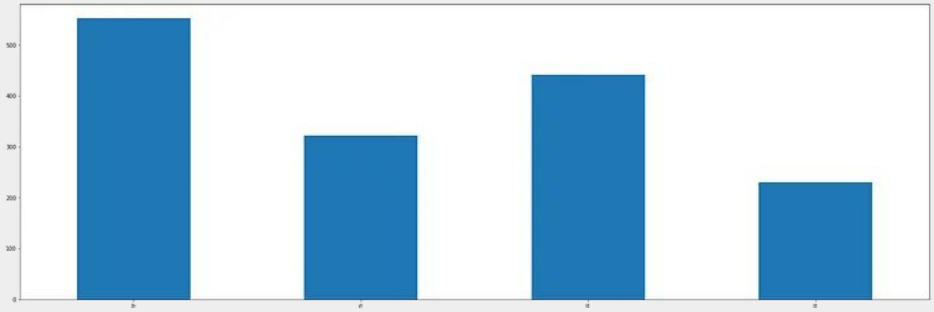
- A naive way to be rank it based on the total Likes + Superlikes,
- (Likes + 2*likes)/ Views
- Wilson's Score and Bayesian Approximation

$$S(n_1,\ldots,n_k) = \sum_{k=1}^{K} s_k \frac{n_k+1}{N+K} - z_{\alpha/2}\sqrt{\left(\left(\sum_{k=1}^{K} s_k^2 \frac{n_k+1}{N+K}\right) - \left(\sum_{k=1}^{K} s_k \frac{n_k+1}{N+K}\right)^2\right)/(N+K+1)}$$

Dismiss =	*	1 point
Like =	**	2 points
Superlike =	***	3 points

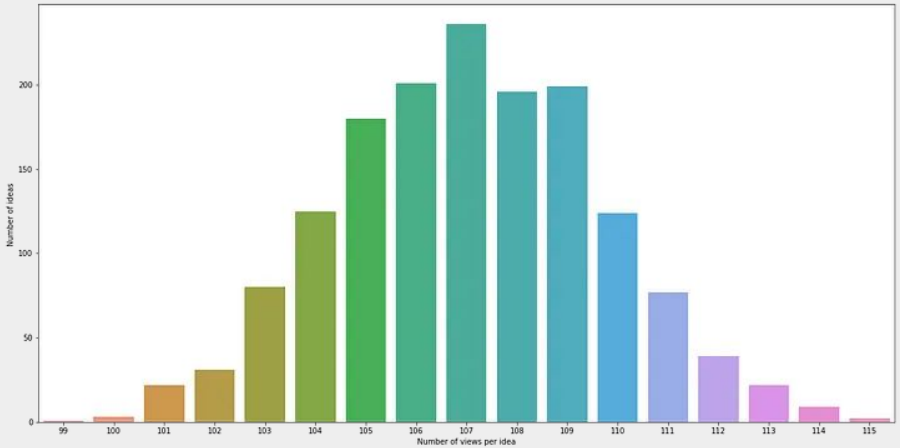
Idea	Dismiss	Like	Superlike	Score
idea1	5	40	35	2.25
idea2	40	30	10	1.51
idea3	5	5	10	1.93

Outcomes



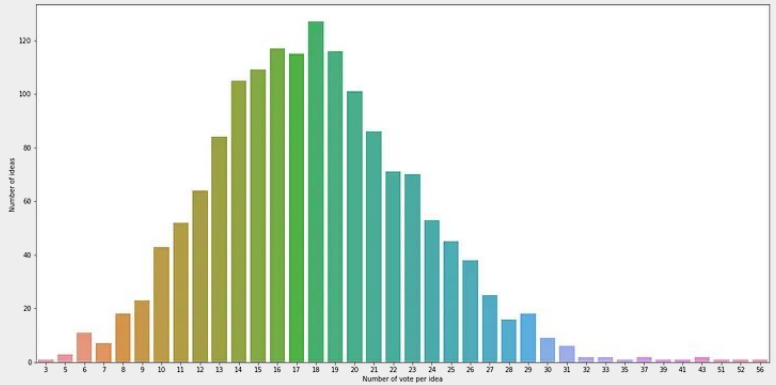
Number of ideas per problem statement

Number of ideas with N views ~ 100 +- 10

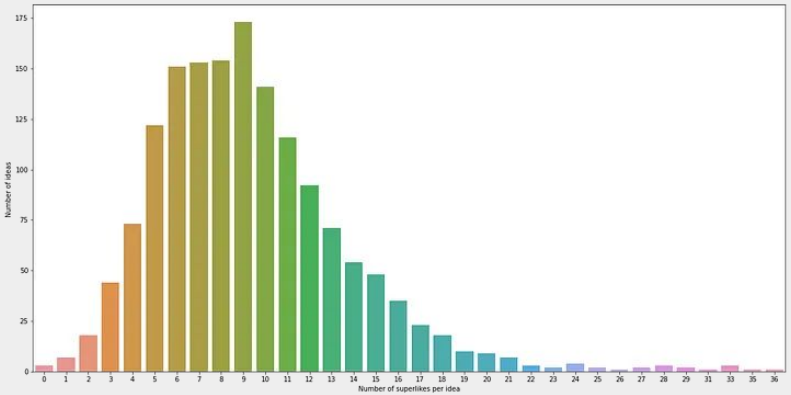


Outcomes

Number of ideas
with N ‘likes’



Number of ideas
with N ‘superlikes’



Reminder

Feedback and suggestions: tinyurl.com/441AppsSP24

Pebble (pebbleu.org) by Matthew Lynch (Soph. in CS)

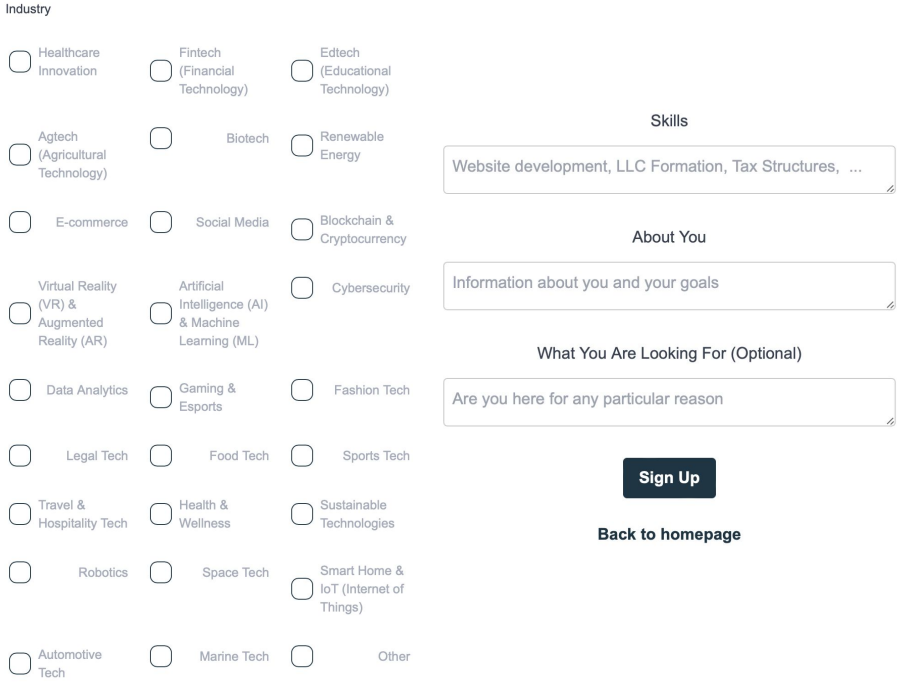
- University authorized platform for supporting co-founder tea formation and promoting student-led start-up recruitment
- User Connections:
    - Students with Students
        - Similar industries of interest
        - Similar skill sets and academic interests
    - Students with existing student-led startups
        - Students possess skills that startup is seeking
        - Similar industries of interest
- Universities Involved:
    - University of Colorado Boulder
    - Colorado School of Mines
    - University of Illinois Urbana-Champaign (awaiting Fall 2024 through TEC)
    - University of Colorado Denver (awaiting onboarding)

PEBBLE

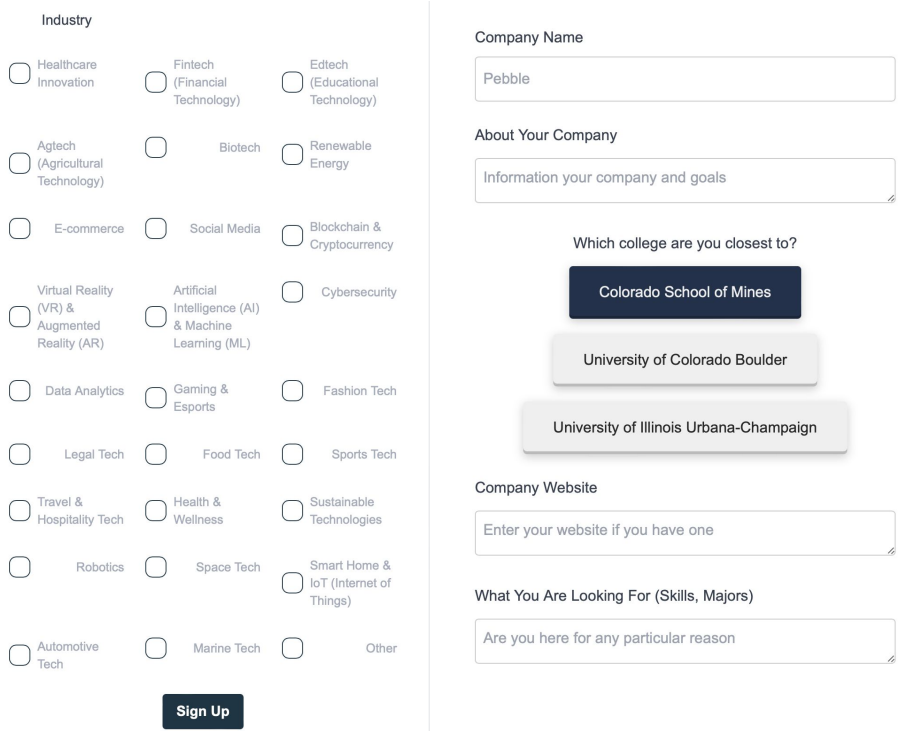
https://en.wikipedia.org/wiki/Colorado_Buffaloes

User Information

Student



Start-Up



Challenges and Guiding Questions

Current Approach:

- Similarity score between users (weighted sum of the following):
  - # of industry matches: One hot encoding industries and returning number of matches
  - Cosine similarity of BERT embeddings (using Hugging Face SentenceTransformer) of “[skills] + [information]” for user1 and “[skills looking for] + [information]” for user2

Points to Improve:

- Emphasis of skill-matching over industry matches
- Alternative approaches to BERT
- Other categories/information to match users on aside from skills, industry, and bio

Reminder

Feedback and suggestions: tinyurl.com/441AppsSP24

Song Translation For Language Learning

Matthew Tang

Motivation: Listening to music is a great way to learn a language

Problem: Sentence by sentence translations of the lyrics are not very helpful

Want a human like breakdown of individual phrases and components of the translation

Example: Which translation is more helpful?

1. 나는 읽기 쉬운 마음이야
Translation: I have an easy mind to read

2. 나는 읽기 쉬운 마음이야
   - Overall translation: I am a heart that's easy to read.
   - Word-by-word translation:
     - 나는 (naneun) = I
     - 읽기 (ilggi) = reading (noun form of "to read")
     - 쉬운 (swiun) = easy
     - 마음이야 (ma-eum-iya) = heart (is)

Song Translation For Language Learning

Matthew Tang

Workflow

System prompt: You are a translator which includes both the translation of each line as well as the translation of individual words
User prompt: Translate each line of the following lyrics, and also include the translations of each word: {lyrics}\n Include both the line by line translation as well as translations of each word

Song Translation For Language Learning

Matthew Tang

Challenges:

- Tuning level of detail
  a. Whether to translate particles as part of the word (나는 = I)
  b. Level of grammatical detail (Verb conjugations)
- Differences between ChatGPT Web and Chat API calls
  a. Likely due to hand picked system prompts in ChatGPT web

Reminder

Feedback and suggestions: tinyurl.com/441AppsSP24

Algophony: Deep Learning Music Generation

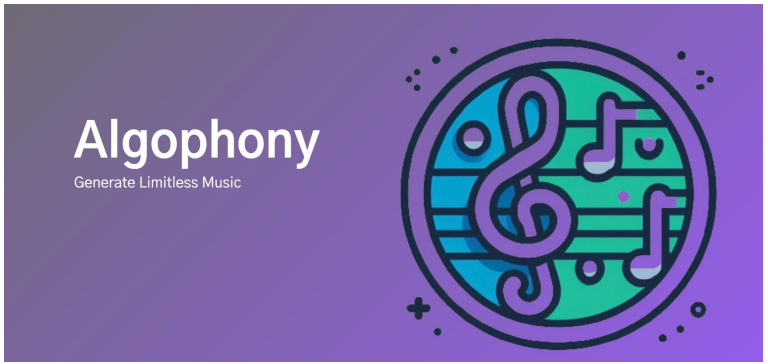
By Ethan Chen

Intuitions

Suno, Udio, Audio Craft

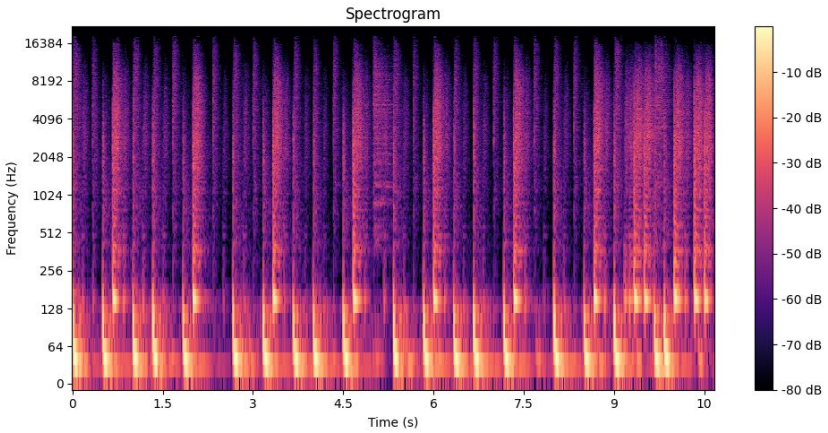
Musical “Stems”

Audio (1D) -> Spectrogram (2D)

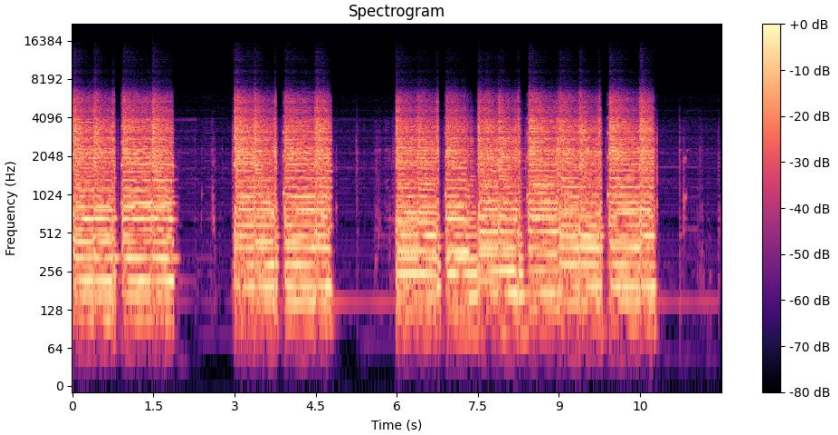


Spectrogram Examples

Drum Loop



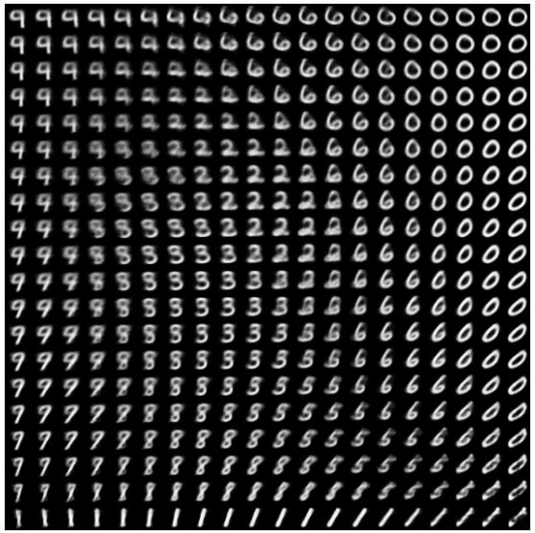
Distorted Electric Guitar

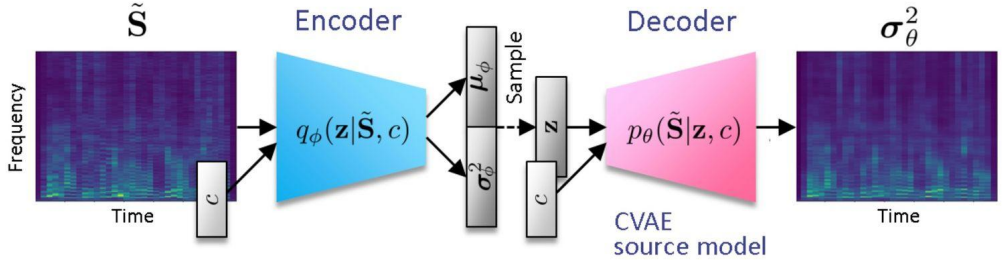


Model Architecture

Conditional Variational Auto-Encoder (CVAE)

Dataset -> X = Spectrograms, y = Labels, generated by pre-trained classification models

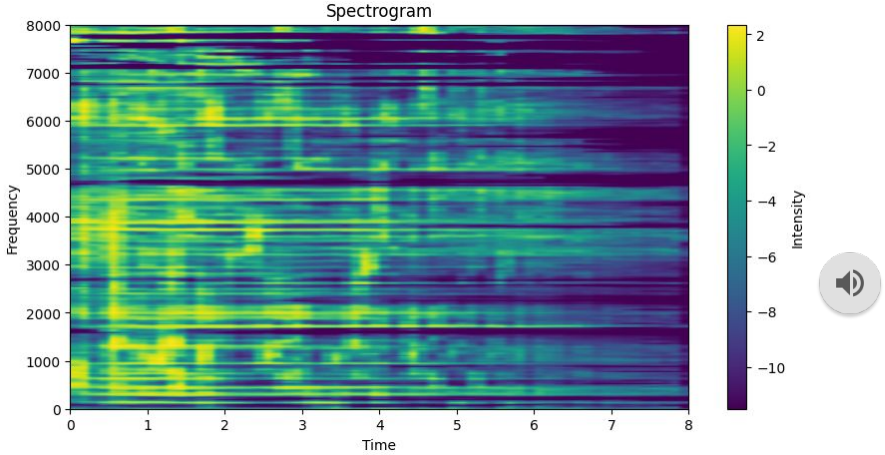


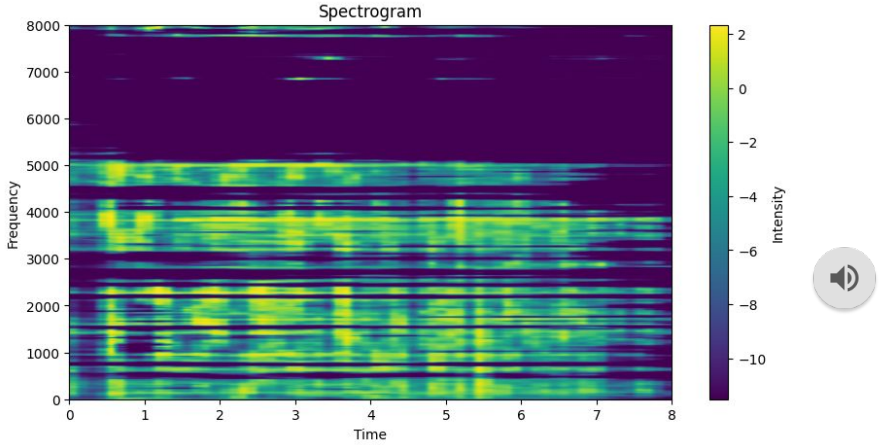


Spectrogram to Audio (output)

With a pre-trained Hifi-GAN model for speech synthesis

Results and Challenges





Dataset limitations (not enough and too diverse)

Insufficient learning of label

Reminder

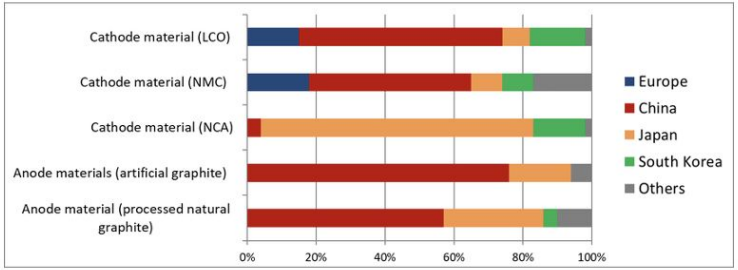
Feedback and suggestions: tinyurl.com/441AppsSP24

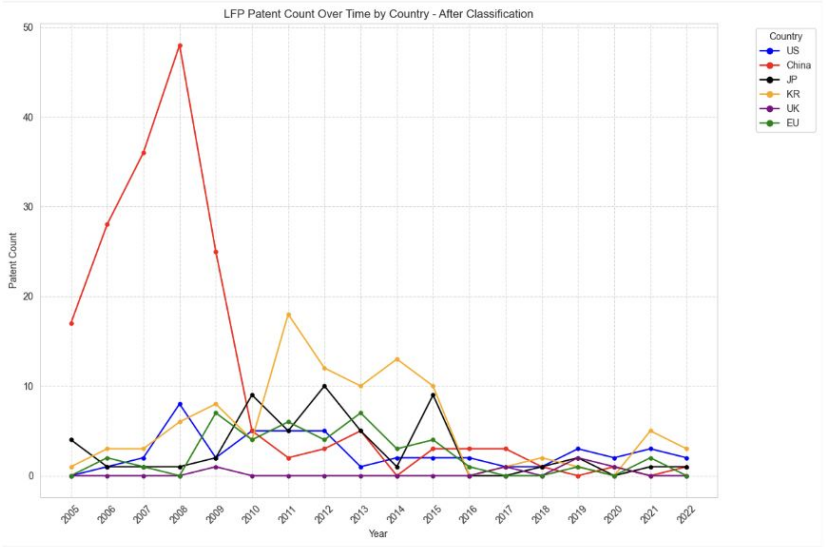
Akshata Tiwari

HOW CAN WE TRACK BATTERY TECHNOLOGY INNOVATION?

- **Battery technology** can fundamentally transform **world energy markets** and lead to birth of new industries
  - World-wide race to capture advanced battery market
  - Secure domestic supply to increase battery technology production by 2030
  - **95%** of batteries are imported from foreign nations
  - Leading patent research - **1M+ patents on battery technology**

- **RESULT:** The **Federal Consortium for Advanced Batteries** (FCAB) brings Federal agencies having a stake in establishing a **domestic supply of lithium batteries** together to accelerate the development of a robust secure domestic industrial base for advanced batteries.
  - **National collaboration** across the U.S. Government agencies to develop a healthy domestic ecosystem





TEXT CLASSIFICATION: ML APPROACH

An iterative approach for evaluating and selecting ML model based on accuracy and compute efficiency.

Model	Performance (Accuracy)	Pros & Cons
1. Preliminary Text Classification	50%	• Baseline comparison model • Dependent on keywords • Not comprehensive
2. Multinomial Naïve Bayes	71.4%	• Computationally efficient • High-dimensional data • Assumes independence
3. Support Vector Machine (SVM)	73.7%	• High dimensional data • Sensitive to noise and outliers • Flexible and interpretable
4. Logistic Regression	94.7%	• Very computationally efficient and high accuracy • Sensitive to outliers • Assumes linearity

ML and NLP Task:

- Research supervised learning algorithms and NLP techniques for text classification of patent data into 13 battery chemistries
  - Evaluate ML model results through analyzing accuracy, precision, and recall. Perform hyperparameter and fine tuning to obtain ML model with high accuracy and low bias
  - Create formal and accurate machine learning pipeline

Main challenges:

- Simple, scalable, low risk of overfitting, compute efficient
- Categorical based approach that allows for multi-label classification

MODEL RESULTS:

Successful classification of ~1 million rows of patent data into 13 battery chemistry groups

- Logistic Regression model
- Count Vectorizer*
- TF-IDF Transformer*

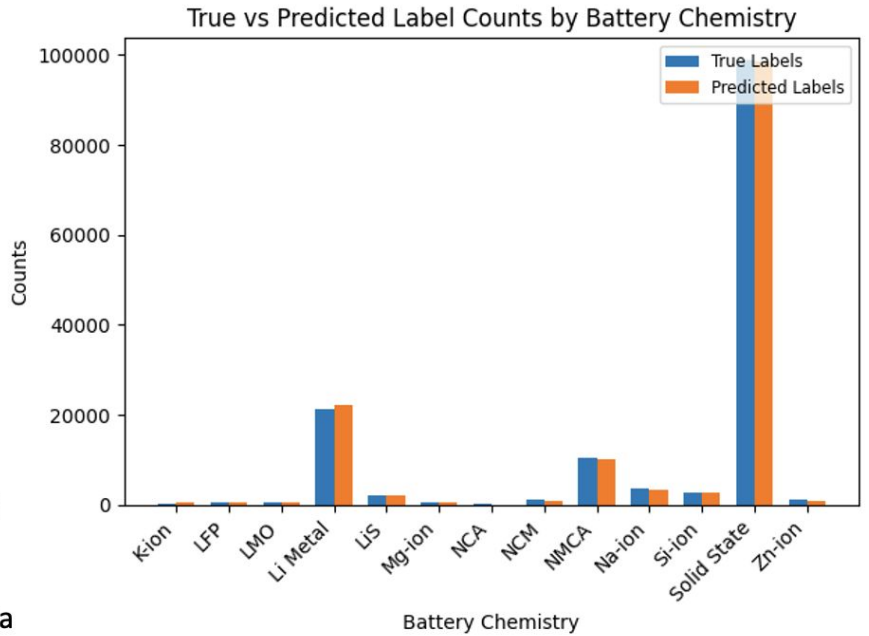
94.68% accuracy
1.3M classifications in 1.4 min
13 classes classified correctly

Would BERT for Patents perform better?

- Simultaneous ongoing efforts for patent text classification using BERT

What are the next steps?

- Begin patent data effort for next stages of the FCAB Project (data collection, cleaning, translation, and categorization using ML pipeline)



*Significant improvements in **Solid-State** and **Li Metal** chemistries*

Reminder

Feedback and suggestions: tinyurl.com/441AppsSP24

Using Edge Computing to Automate Rideshare Vehicle Detection

Rideshare Opportunity

97M
Users in 2027 in US

133%
CAGR 2022-2028

72B
US Rideshare Market in 2023

Source: Statistica

Challenges

- Lack of publicly available rideshare data
- Data is Competitive Information
- Cities don't have capabilities & resources
- Concerns with Ethics and Privacy

Developed Solution

STAGE 1 Image Augmentation Dataset

STAGE 2 Model & Algorithm Development

Two Stage Detection Framework: ~90% accuracy

Filter 1: Macro Detection → Resulting Detections → Filter 2: Micro Detection → Sticker Detected (count, time, prediction stored)

Mean Avg Precision at 95%: 0.451
46/56 identifications

Mean Avg Precision at 95%: 0.000154
0/56 identifications

STAGE 3 App Creation & Deployment

Video Stream Input

Create + Test Edge app → Publish to ECR → Rideshare count, predictions, meta-data published to Beehive Data Repo

- Unique 2-filter solution provides around 90% accuracy
- Novel rideshare image dataset published

SAGE sagecontinuum.org
Waggle
Anagha Tiwari
Argonne NATIONAL LABORATORY
U.S. DEPARTMENT OF ENERGY

Use Cases and Roadmap

First Rideshare Vehicle Detection algorithm

Development of an **API**

Transportation Development

Citizen Safety

Infrastructure Development

City Planning & Zoning

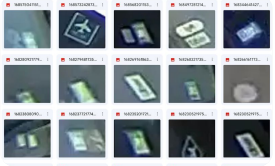


Image Dataset

Involve **metrics**: speed, direction, rush hour, waiting time in parking lots

Rideshare sticker **differentiation**

Update ML Model

Deploy nodes

Sage APIs for new devices and cloud platforms

SAGE sagecontinuum.org

Waggle

Argonne NATIONAL LABORATORY

U.S. DEPARTMENT OF ENERGY

Questions & Next Steps

Applications in **EDAs**?

Changes to model performance & accuracy?

Flexible CV algorithms @ Edge?

Forecasting
(emissions, sea level changes)

Reduction & Removal
(tracking deforestation, wildfires)

https://sagecontinuum.org,
https://sagecontinuum.org/news/sage-hawaii

Seongha Park, MCS Division, ANL & Northwestern
Pete Beckman, NAISE Co-Director

SAGE
sagecontinuum.org

Waggle

Argonne NATIONAL LABORATORY

U.S. DEPARTMENT OF ENERGY

Reminder

Feedback and suggestions: tinyurl.com/441AppsSP24

Next week

You are not required to submit a final project if you have more than 500 total points and are in the 3 credit version or more than 625 points and in the 4 credit version

- You will need to estimate interaction points and may need to estimate HW 5 if submitted late
- If you earn more than required points, it makes the 100% experience points portion of your grade count more towards final grade

Friday: will send survey for detailed feedback

Next Tuesday: wrap up with review of key concepts, trends/future of ML, and closing remarks

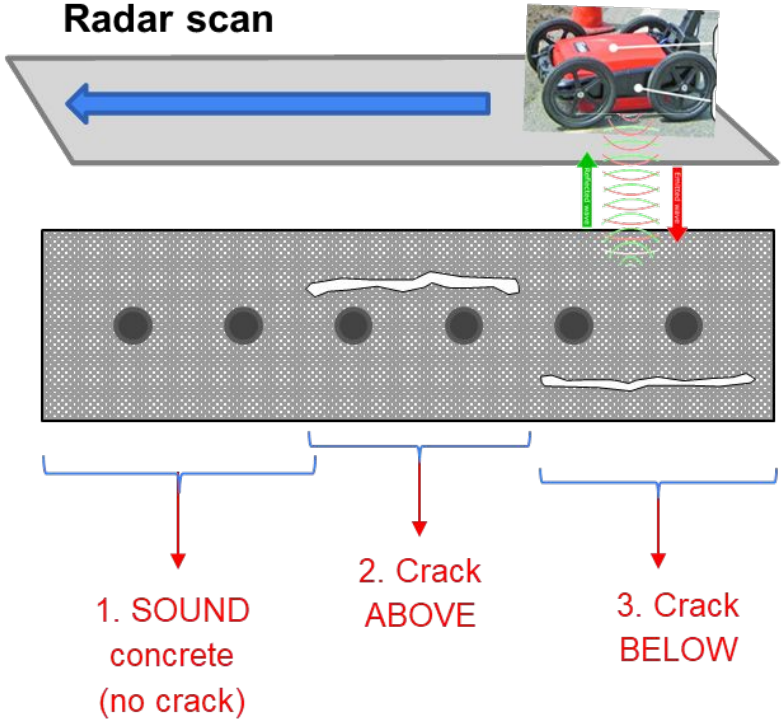
Below are examples from last year

Do not edit slides after this

*Identifying **CRACKS** in concrete bridges with Radar sensing and Machine Learning*

by Ishfaq Aziz

https://www.concrete.org/Portals/0/Files/PDF/18-JI-Paper.pdf



A radar sends EM wave and receives a reflected signal that contains information. →

Problem:

- With a given scan-signal, can we predict which class it refers to?

Dataset:

- Real data from 5 different bridges with labelled ground truth

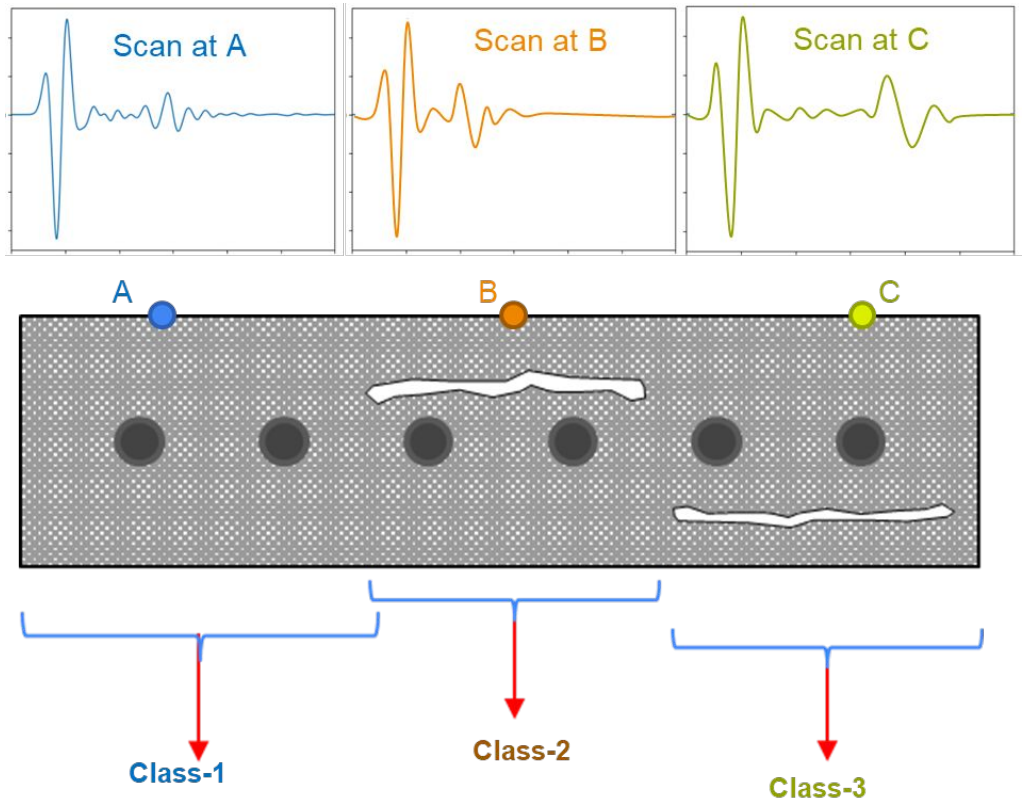
- Total data = 500,000 signals
  - Class-1 : 71 %
  - Class-2 : 24 %
  - Class-3 : 05 %

→ **Imbalanced dataset**

Target:

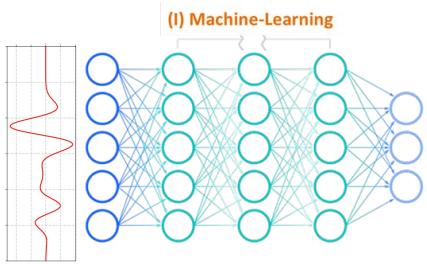
Train on data from 4 bridges and test on data from 5th (zero-shot)

Each scan-signal is a datapoint which correspond to one class



Models tried:

- Random Forest & Logistic regression with features (Freqs., PCA, etc.) as input.
- 1-D CNN with raw signal as input.

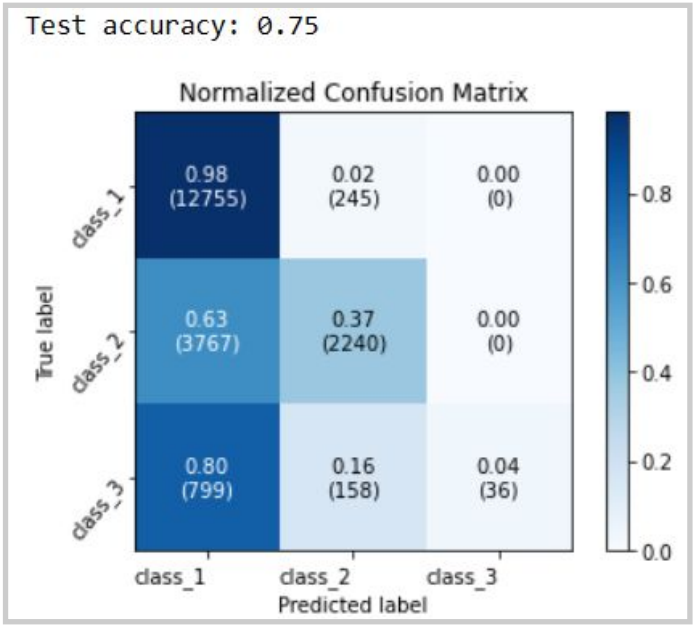


Techniques tried for class balancing:

- Oversampling minor class
- Undersampling major class
- Weighted loss function

$$(\text{weight of a class}) \; \alpha \; \frac{1}{\text{class size}}$$

Even after these, models are biased towards the major class (Class-1) →



Ideas from class

More ideas

- Amplitude and maybe phase of frequencies over time seems most relevant – use 1D Discrete Cosine Transform (DCT) as the raw representation
- Normalize for each bridge
  - Subtract median features from raw features
  - Represent the probability of seeing a set of DCT values at a particular time
- Reformulate as detection of depth of crack
- Clustering or mixture modeling may help


hatch

Recruiting Currently

1

Post a job

2

Manually filter
600-800 applicants

3

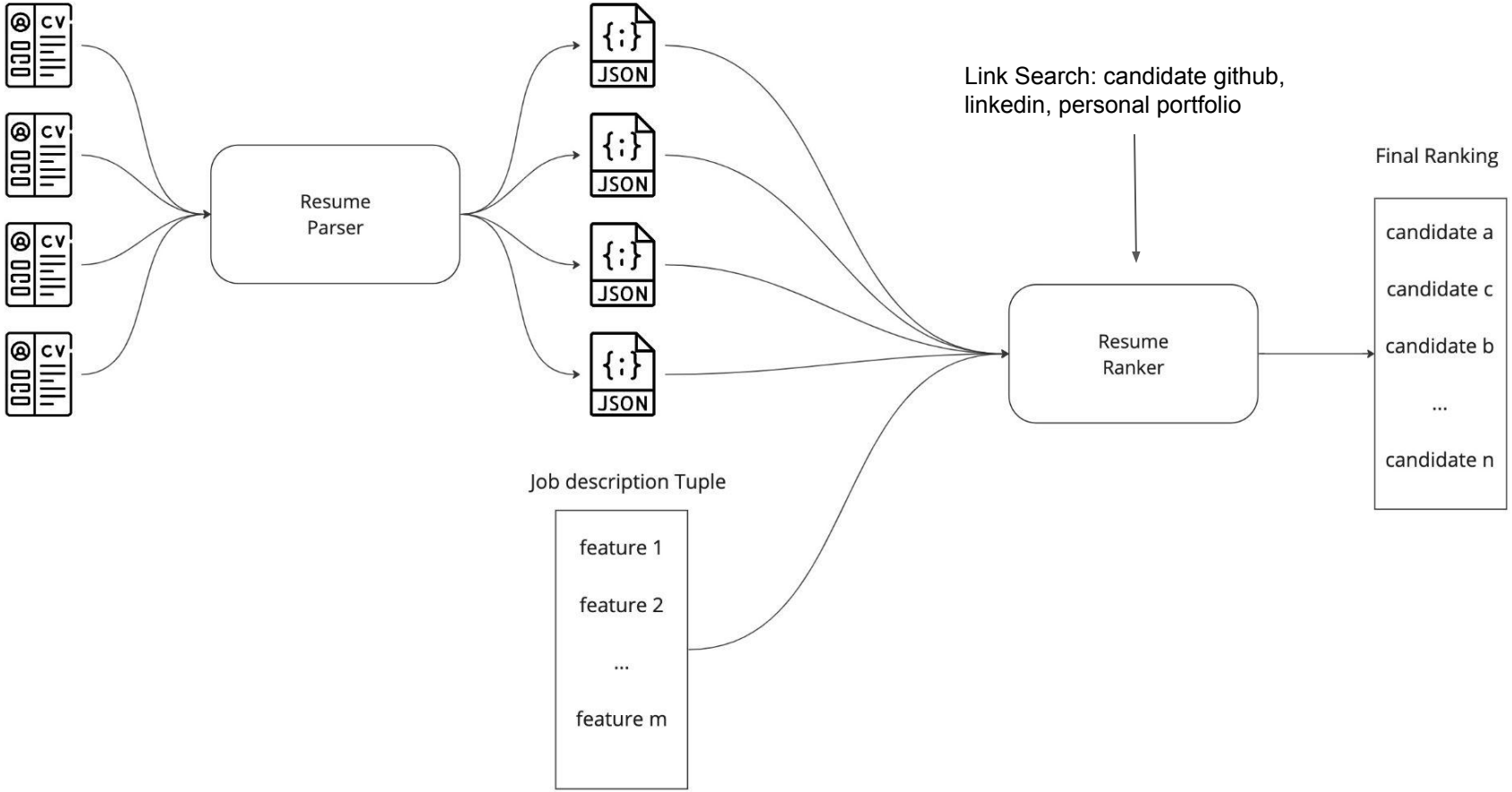
Interview top candidates

4

Extend job offer

The Solution

Hatch is an **AI recruitment platform** that allows recruiters to **rank hundreds of applicants in seconds**.


CV
CV
CV
CV
Resume
Parser
{;}
JSON
{;}
JSON
{;}
JSON
{;}
JSON
Link Search: candidate github,
linkedin, personal portfolio
Resume
Ranker
Final Ranking
candidate a
candidate c
candidate b
...
candidate n
Job description Tuple
feature 1
feature 2
...
feature m
miro

What have we considered:

- Using the Word2Vec model with either SkipGram or CBOW.
  - Which is the better choice for resume parsing?
- Then apply the model to the resumes and job descriptions:
  - We do we scale the resultant vectors equally. How?
- And apply cosine similarity between the “job vector” and “student vector” to rank the resulting resume vectors:
  - Is cosine similarity the best similarity measure? What other comparison metrics can we use?

What we are exploring:

- How do we build the dataset? We have some beta testers (startup founders, recruiters), but what sort of data should we collect from them? (ranked resumes as truth labels?)
- Are there any foundational models we can start with comparable runtimes and smaller scale?
- How do we eliminate recruiter bias?
- Anything we’re missing when considering this problem!

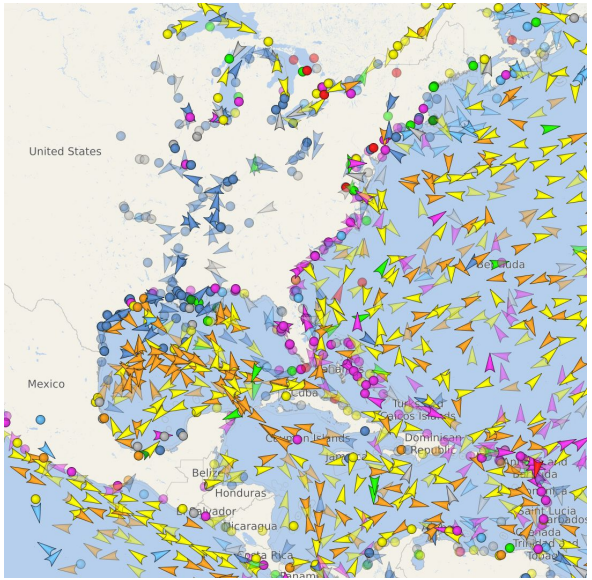
Ideas from class

More ideas

- Sounds like you have a resume parser, but just in case https://github.com/hxu296/nlp-resume-parser
  - Once you know what works, you can use it directly or to inform an approach
- BERT may be good starting point for representation: inputs are json for job description and json for resume
- Building dataset – very tricky.
  - One approach is to build a useful non-automated approach that you provide for free and gather ratings of candidate quality and job fit
  - Can use RL during deployment
  - May want to maintain both general model and client-specific model
  - Can predict whether the person is selected for interview, offered, hired – independent recruiters can offer a lot of data
- Bias – consider providing separate scores for different subgroups, various ideas in Model Cards are relevant

Vessel Route Prediction

by Marcus Kornmann



- Automated Information System (AIS)
- Vessels send dynamic and static information in fixed intervals
- Position, vessel specifications (length, width, …), status

https://www.vesselfinder.com/

Vessel Route Prediction

Goal

- Detect areas of high traffic
- Predict the route of a vessel given its previous movement
- Use information for route planning for autonomous sailing boat

Vessel Route Prediction

Challenges

- Size of dataset: > 6 million data points per day
  - How to efficiently read large datasets (even using data for a specific region requires reading all the data once)?
- Very unreliable data:
  - Some of the information is added manually => high error quotes (e.g. ship length / width)
  - Some specifications are ambiguous (e.g. status)
  - Outliers (unreasonable) values for vessel movement

Approach

- Clean data, divide tracks into subtracks, perform line-reduction (define tracks by turning points), perform clustering

Ideas from class

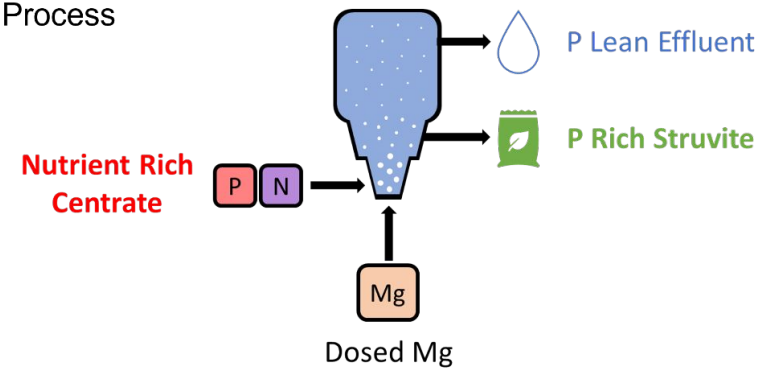
More ideas

- Track flow of traffic instead of points?
  - Model density at each tile, and probability of transitioning between tiles
  - Potentially, consider longer history, e.g. next tile given previous three tiles
-

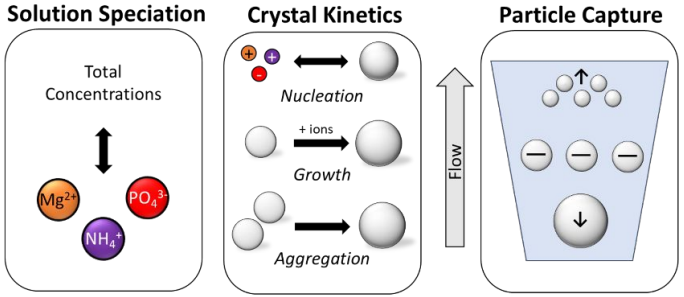
Improving Crystallizer Effluent Particle Size Distribution Predictions with Hybrid Mechanistic/ML Process Models

by Sam Aguiar

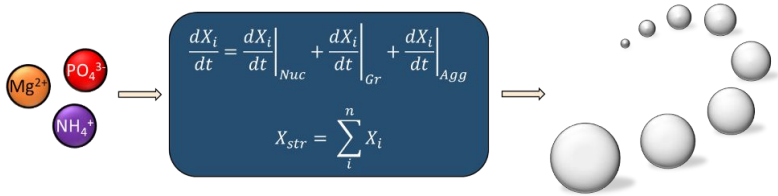
1) Struvite Crystallization Process



2) Core Crystallization Phenomena is Complex



3) Current Process Models **are not** Generalizable



4) Real world applicability – Current mechanistic understandings don't fully capture what might happen in real wastewater

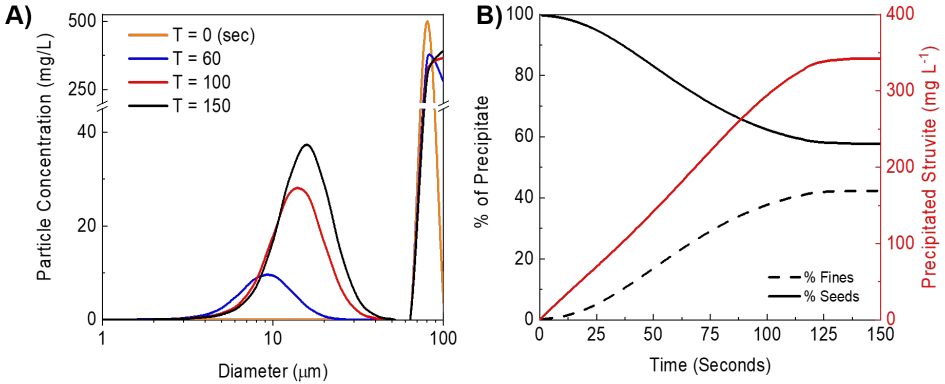
Can we combine mechanistic models with ML to help better predict changes to crystallizer effluent under non-ideal conditions?

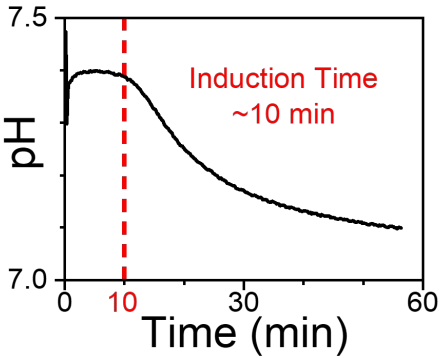
Available Data

Currently accounted for in Model
- Ion concentrations
- Particle Size Distribution in reactor

Partially included in Model
- Flow rates or Mixing Conditions
- Temperature

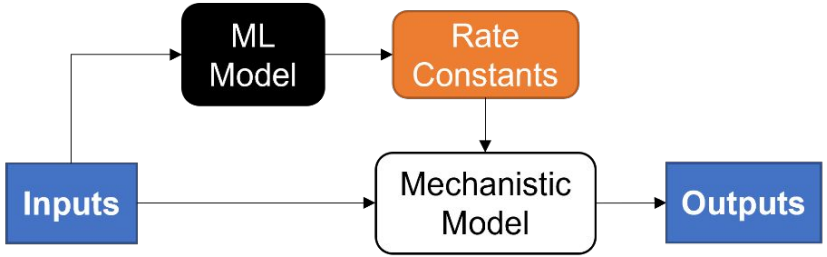
Not Included in Model
- Organic additives
- Solids content
- Coprecipitation
- Fluid dynamics



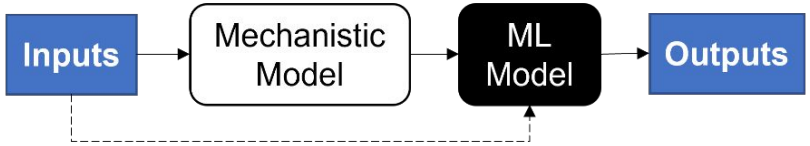


How can we implement a Hybrid Model?

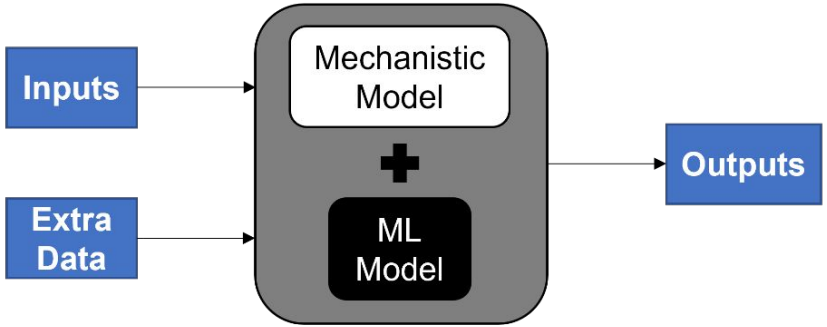
1) Adjusting Mechanistic Model Parameters



2) Adjusting Mechanistic Model Outputs



3) Accounting for Model Externalities



4) Full Replacement (Faster Simulation)

Ideas from class

More ideas

- May have some relationship to using RL to solve for kinematics e.g. this
-