

**15.095 Machine Learning Under a Modern
Optimization Lens**

Health Tweets Classification & Recommendation

Yuki Yu & Naiqi Zhang

December 7, 2024

Abstract

This paper addresses the classification of unlabeled text data and the development of a recommendation system. Using a dataset of health-related tweets collected in 2015 provided by the UCI Machine Learning Repository, we trained a model to classify tweets into various categories and designed a recommendation system based on the classification results. The system recommends tweets to users based on their recent viewing history of health-related content. We experimented with various models, performed hyperparameter tuning, and compared their performance to select the best approach. Ultimately, our tuned neural network performed the best, and we used it to implement a category-based recommendation system. Finally, we concluded with insight for future improvements.

1 Introduction

The rapid growth of social media platforms has generated an enormous amount of user-generated content, including text data such as tweets. This vast repository of textual information presents both opportunities and challenges for analysis. Classifying and organizing such data is essential for deriving meaningful insights and creating tools like recommendation systems. By leveraging machine learning and optimization techniques, we can develop robust models to classify unlabeled text and provide personalized content recommendations.

In this report, we focus on health-related tweets collected from a publicly available dataset ("Health News in Twitter", UCI Machine Learning Repository). We aim to address two main tasks: classifying tweets into predefined categories and designing a recommendation system that suggests content based on user preferences. Our work combines state-of-the-art machine learning methodologies with hyperparameter tuning to optimize classification performance. Furthermore, the recommendation system leverages these classification results to deliver relevant content tailored to individual users' interests.

1.1 Challenges

Developing this recommendation system presented two key challenges: defining the categories for classification and fine-tuning the hyperparameters of the models to optimize their performance.

1.1.1 Defining Categories

Given that the dataset consisted of unlabeled tweets, we determined appropriate categories for classification. Initially, we identified common themes such as viruses, diseases, public policies, and fitness. To ensure comprehensive coverage, we began by iteratively labeling the data with detailed subcategories. These subcategories were then grouped into 10 broader categories, striking a balance between granularity and manageability for classification purposes.

1.1.2 Hyperparameter Tuning

Fine-tuning hyperparameters is critical to achieving optimal performance in machine learning models. We experimented with various models, including XGBoost, Random Forest, OCTs, and Neural Networks. Among these, Neural Networks consistently delivered the best results. To optimize performance, we primarily used Bayesian Optimization to efficiently explore hyperparameter spaces across all models, ensuring a systematic and effective tuning process.

1.2 Combating Class Imbalance

The dataset exhibited significant class imbalance prior to applying any resampling techniques, as shown in the class distribution after labeling the data. Certain classes were severely underrepresented, compared to more dominant classes. This imbalance posed a challenge for the classification

model, as it could bias predictions towards the majority classes, leading to poor performance on minority classes.

To address this issue, we utilized SMOTE (Synthetic Minority Oversampling Technique), a method that generates synthetic samples for minority classes to balance the dataset. After applying SMOTE, all classes were resampled to contain an equal number of samples, creating a balanced dataset. This ensured that the classification model received sufficient representation from all classes during training, reducing bias and improving its ability to generalize across all categories.

2 Methodology

This section outlines the methodology employed in our project. We begin by detailing the pre-processing steps, including standardization, tokenization, and embedding generation. Next, we discuss the modeling approach and the hyperparameter tuning process used to optimize performance. Finally, we describe the implementation of the recommendation system, integrating the classification model to deliver personalized content recommendations.

2.1 Data Overview

The dataset, provided by the UCI Machine Learning Repository and collected in 2015 using the Twitter API, contains health-related news tweets from 16 major news agencies, including BBC, CNN, and The New York Times. It consists of 27,771 tweets with metadata such as the tweet ID, date and time, and headline.

2.2 Data Preprocessing

2.2.1 Preprocessing & Extraction

Using regular expressions, we extracted the news source, headline, and date-time information. These were then organized into a structured table, as illustrated in the example figure below.

Figure 1: Tweets in a Dataframe

	Source	Datetime	Text
0	latimes	2015-03-14 15:02:15+00:00	Five new running shoes that aim to go the extr...
1	latimes	2015-03-14 12:04:04+00:00	Gym Rat: Disq class at Crunch is intense worko...
2	latimes	2015-03-13 17:43:07+00:00	Noshing through thousands of ideas at Natural ...
3	latimes	2015-03-13 17:43:06+00:00	Natural Products Expo also explores beauty, su...
4	latimes	2015-03-13 16:02:36+00:00	Free Fitness Weekends in South Bay beach citie...

2.2.2 Tokenization & Embeddings

We utilized BERT, a transformer-based model known for its ability to capture the contextual and semantic relationships between words, to tokenize the text and generate embeddings for each tweet. BERT's tokenization process breaks down text into subword units, ensuring that even rare or unseen words can be effectively represented. This step is crucial for feeding the text into the model while preserving semantic meaning.

BERT's embedding mechanism transforms the tokenized text into dense numerical vectors, capturing both the semantic meaning and contextual relationships between words. This capability

made it a suitable choice for our classification task, as it provides more comprehensive representations of text.

Given the large volume of data, we processed the tweets in batches of size 32 to manage memory usage and prevent session crashes. This batching approach ensured efficient utilization of computational resources while maintaining model performance.

2.3 Labeling

The labeling process involved a combination of approaches to classify the headlines into various categories such as Fitness, Viruses, Mental Health, and Other.

First, we used Zero-Shot Classification, a semantic understanding method that predicts the category of headlines without requiring prior training on the dataset. This approach allowed us to assign categories based on their contextual meaning effectively.

Next, we implemented a Keywords Classification approach, where headlines were categorized based on common keywords associated with each category. For instance, headlines containing the word "gym" were classified under Fitness, while those mentioning "Ebola" were categorized under Viruses.

Finally, we iteratively applied both Zero-Shot and Keywords Classification methods specifically to refine just the "Other" category, ensuring accurate categorization of tweets that did not fit into the primary predefined groups.

Once less than 10% of the data remained in the "Other" category, we consolidated the smaller, more detailed categories into broader groups.

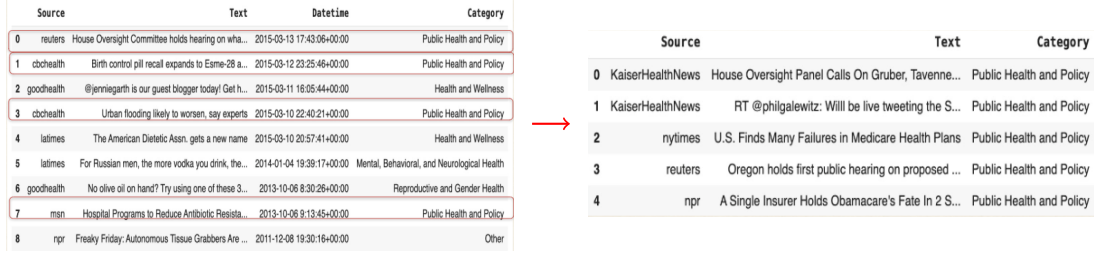
1. Health and Wellness
2. Viruses, Diseases, Pharmaceuticals
3. Public Health and Policy
4. Mental, Behavioral, and Neurological Health
5. Reproductive and Gender Health
6. Health Technology and Medical Research
7. Patient Care and Specialized Health
8. Cardiovascular and Organ Health
9. Disabilities and Diversities
10. Other

2.4 Classification Modeling

We performed a train-test split on the data, allocating 70% of the tweets for training and 30% for testing. The **baseline test accuracy**, achieved by always predicting the most frequent class, was approximately **19.74%**. Our objective was to develop a classification model that significantly outperformed this baseline. The models we tested were XGBoost, Random Forest, OCTs, and Neural Networks. This process involved hyperparameter tuning for all models and comparing and contrasting the results.

2.5 Recommendation System

To simulate user profiles, we used the trained neural network model to classify unlabeled Twitter news into predefined categories, creating a simulated viewing history with latest timestamps. This approach ensures temporal relevance while maintaining diverse content categories for further analysis.



	Source	Text	Datetime	Category
0	reuters	House Oversight Committee holds hearing on wha...	2015-03-13 17:43:06+00:00	Public Health and Policy
1	cbohealth	Birth control pill recall expands to Esme-28 a...	2015-03-12 23:25:46+00:00	Public Health and Policy
2	goodhealth	@jenniegarth is our guest blogger today! Get h...	2015-03-11 16:05:44+00:00	Health and Wellness
3	cbohealth	Urban flooding likely to worsen, say experts	2015-03-10 22:40:21+00:00	Public Health and Policy
4	latimes	The American Dietetic Assn. gets a new name	2015-03-10 20:57:41+00:00	Health and Wellness
5	latimes	For Russian men, the more vodka you drink, the...	2014-01-04 19:38:17+00:00	Mental, Behavioral, and Neurological Health
6	goodhealth	No olive oil on hand? Try using one of these 3...	2013-10-06 8:30:26+00:00	Reproductive and Gender Health
7	men	Hospital Programs to Reduce Antibiotic Resista...	2013-10-06 9:13:45+00:00	Public Health and Policy
8	npr	Freaky Friday: Autonomous Tissue Grabbers Are ...	2011-12-08 19:30:16+00:00	Other

	Source	Text	Category
0	KaiserHealthNews	House Oversight Panel Calls On Gruber, Tavenne...	Public Health and Policy
1	KaiserHealthNews	RT @philgalewitz: Will be live tweeting the S...	Public Health and Policy
2	nytimes	U.S. Finds Many Failures in Medicare Health Plans	Public Health and Policy
3	reuters	Oregon holds first public hearing on proposed ...	Public Health and Policy
4	npr	A Single Insurer Holds Obamacare's Fate In 2 S...	Public Health and Policy

Figure 2: Left: simulated user profile; Right: 5 recommended news filtered by the top category and similarity scores

As a baseline, we recommended the top 15 news articles with the highest average cosine similarity scores to the news in the user profile from the remaining test dataset. To leverage our neural network classification model, we identified the top three categories in a user profile based on frequency. We used these categories to filter the test dataset. Then, we recommended five news articles from each category in the filtered remaining test data. These are chosen based on their high average cosine similarity scores to the articles in the same category within the user’s profile. This approach ensures that the recommendation system focuses on the most relevant content for the user while significantly reducing the dataset size, making recommendation decisions more efficient. In Figure 2, we have limited the demo to a single category and 5-item recommendations, which could be generalized to 3 categories and 15-item recommendations.

For text embeddings, we experimented with two techniques. The first approach employed Scikit-learn’s TfidfVectorizer to convert textual data into TF-IDF matrices, which weight word importance within the text. However, this method yielded relatively low and unstable cosine similarity scores because it did not capture contextual relationships in the text. To address this limitation, we adopted GloVe embeddings, which incorporate contextual information, providing embeddings with more robust similarity measurements.

We evaluated our recommendation system by comparing runtime efficiency and average similarity scores with and without the category filtering step as the size of the user profile increased, which offers insights into both the computational efficiency and recommendation quality.

3 Results

3.1 Classification Model

Our neural network model ultimately had the best performance, so it was chosen as our classification model. However, our results for every model are still detailed below.

3.1.1 Random Forest

Random Forest performed the best using its default parameters, achieving a testing accuracy of approximately **49.35%**. Interestingly, attempts to improve performance through hyperparameter tuning, such as employing grid searches and adjusting the number of trees, maximum depth, and minimum samples per split, resulted in lower accuracy and F1 scores. This suggests that the default settings provided a well-balanced configuration for this dataset, likely benefiting from the

model's inherent robustness and ability to handle high-dimensional, imbalanced data effectively.

The decline in performance during hyperparameter tuning could be attributed to overfitting on the training data or a suboptimal search of the parameter space. These results underscore the strength of Random Forest's default configuration, which offers reliable and consistent performance without requiring extensive tuning. This finding highlights the practicality of Random Forest as a baseline model, especially in scenarios where computational resources or time for extensive optimization are limited.

3.1.2 Optimal Classification Tree

The Optimal Classification Tree (OCT) achieved a testing accuracy of approximately **32.44%** after hyperparameter tuning. The best parameters identified during the grid search were `max_depth=7`, `min_samples_leaf=1`, and `min_samples_split=10`. This setting allowed the model to capture complex decision boundaries while avoiding overly restrictive splits.

After applying SMOTE to address the imbalance of categories, the overall testing accuracy decreased to **26.77%**, reflecting the trade-off introduced by focusing on improving prediction for minority classes. While the model demonstrated better inclusivity for underrepresented groups, the overall performance suffered due to the altered class distributions and increased complexity introduced by SMOTE.

3.1.3 XGBoost

The XGBoost classifier achieved a testing accuracy of **35.66%** after hyperparameter tuning, with the best parameters identified as follows: `learning_rate=0.3`, `max_depth=3`, and `n_estimators=200`. Similarly, after applying SMOTE to address class imbalance, the testing accuracy dropped to **28.38%**, reflecting the trade-off introduced by oversampling the minority classes. While the overall accuracy decreased, the prediction for underrepresented classes improved, demonstrating better inclusivity for minority groups.

3.1.4 Neural Networks

The **base neural network model** used was a simple feedforward architecture designed for multi-class classification:

- **Input layer:** Accepts data with the specified input dimensions.
- **Hidden layers:**
 - The first dense layer consists of 128 neurons with ReLU activation, followed by a dropout layer with a rate of 0.3 to prevent overfitting.
 - The second dense layer has 64 neurons with ReLU activation, also followed by a dropout layer with a rate of 0.3.
- **Output layer:** A dense layer with 10 neurons, using a softmax activation function for multi-class classification.
- **Optimization and Loss Function:** The model was compiled using the Adam optimizer and categorical cross-entropy loss, with accuracy as the evaluation metric.

Before applying SMOTE, the neural network achieved a test accuracy of **58.6%**. However, the performance across classes varied significantly due to the dataset's class imbalance. Majority classes, such as class 9, achieved high F1-scores (e.g., 0.78), while minority classes, such as classes 0 and 1, had much lower F1-scores (0.19 and 0.25, respectively). This imbalance caused the model to favor majority classes, leading to poor recall for underrepresented categories and highlighting the need for resampling techniques.

After applying SMOTE to balance the dataset, the model’s performance shifted. The overall accuracy slightly decreased to **55.3%**, but there were notable improvements in the recall and F1-scores of minority classes. For example, the F1-score for class 0 increased from 0.19 to 0.38, and for class 1, it improved from 0.25 to 0.34. However, the F1-scores for majority classes, such as class 9, slightly decreased from 0.78 to 0.73, reflecting the trade-off introduced by balancing the dataset. While SMOTE helped the model achieve a more equitable performance across classes, the reduction in overall accuracy highlights the inherent trade-off between global metrics and fairness across imbalanced datasets.

Therefore, SMOTE effectively addressed the class imbalance in the dataset, improving the model’s ability to predict minority classes. Although the overall accuracy decreased slightly, the improved recall and F1-scores for underrepresented categories underscored the importance of balancing datasets for fair and comprehensive multi-class classification, which is why we proceeded with classes balanced by SMOTE to train our neural network.

Using our balanced data, we fine-tuned the hyperparameters of our neural network with Bayesian Optimization, an optimization process that efficiently searches for optimal hyperparameters by modeling the objective function and iteratively selecting the most promising values to evaluate.

The **best-performing neural network model**, optimized through hyperparameter tuning, was configured as follows:

- **Activation Function:** The hidden layers used the `tanh` activation function, which scales values between -1 and 1 to capture non-linear relationships effectively.
- **Architecture:**
 - **Number of Layers:** 3 hidden layers.
 - **Neurons in the Hidden Layer:**
 - * Layer 1: 256 neurons.
 - * Layer 2: 96 neurons.
 - * Layer 3: 192 neurons.
- **Dropout Rates:**
 - First Layer: 0.4158
 - Second Layer: 0.1109
 - Third Layer: 0.4926
- **Output Layer:** A dense layer with 10 neurons (one for each class) and a `softmax` activation function for multi-class classification.
- **Batch Size:** 32, balancing computational efficiency and convergence speed.
- **Optimizer:** `adam`, chosen for its ability to adapt learning rates and accelerate convergence.
- **Epochs:** 50, providing sufficient iterations for learning without overfitting.
- **Early Stopping:** validation accuracy.

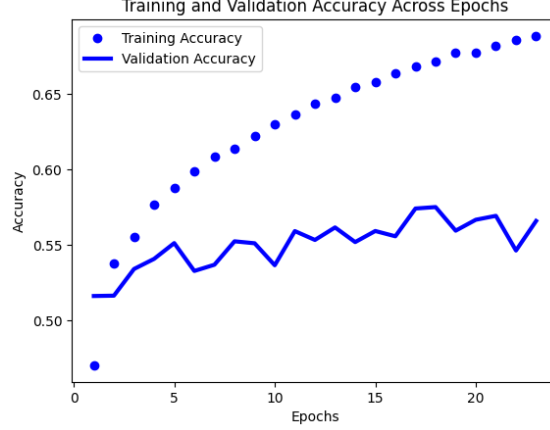
The tuned neural network model outperformed the base model after SMOTE, achieving a final test accuracy of **57.52%** compared to the base model’s **55.39%**. This improvement was reflected in better precision, recall, and F1-scores across most classes. For instance, class 0 improved in precision from 0.29 to 0.35, while class 9 maintained its strong performance with a precision of 0.79 and recall of 0.76. Minority classes like 1 and 5 also saw notable gains, demonstrating the model’s enhanced ability to handle imbalanced data.

The F1-scores for key classes improved, with the macro-average F1-score rising from 0.48 to 0.50,

and the weighted average increasing to 0.58, reflecting a better balance across all classes.

However, as seen below, our model seems to be overfitting, so this is something that needs to be improved in the future, possibly through more hyperparameter tuning or using transfer learning with other pre-trained models such as GPT-3 and XLNet.

Figure 3: Training & Validation Accuracy



3.2 Recommendation System Approach

3.2.1 Time Efficiency

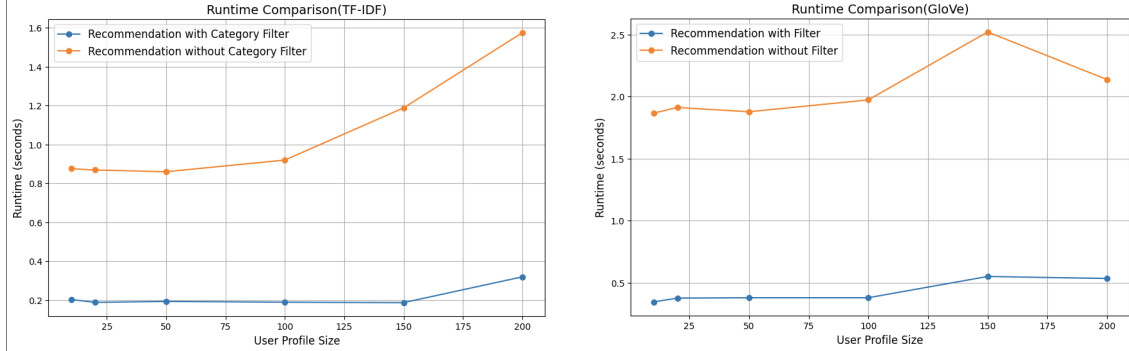


Figure 4: Runtime Analysis of a News Recommendation System with and without Category Filter

We tested the runtime of recommendation system with and without category filtering, using a test data pool of approximately **8230** news items. The evaluation measures the runtime for recommending 5 news items, comparing scenarios with a category-based filter applied to the top categories versus no filter (refer to the methodology section). As seen in Graph 4, runtime increases with larger user profile sizes due to the increased number of cosine similarity calculations required. With the TF-IDF-based embedding method, the runtime with a filter ranged from **0.2** seconds (for a profile size of 10) to **0.4** seconds (for a profile size of 200), while the runtime without a filter increased significantly, from **0.9** seconds to **1.6** seconds over the same range. Similarly, for the GloVe-based embeddings, the runtime with a filter ranged from **0.2** seconds (for a profile size of 10) to around **0.5** seconds (for a profile size of 200), whereas the runtime without a filter ranged from around **1.8** seconds to **2.5** seconds.

These results demonstrate that applying a category filter reduces the size of the data pool and improves the efficiency of the recommendation process. The filtering approach allows for more personalized recommendations with higher time efficiency, particularly evident for larger profile

sizes. The time savings with filtering become more pronounced as the computational cost of similarity score calculations increases with the dataset size. This balance between personalization and efficiency shows the advantage of incorporating a filtering step in the recommendation process, especially as the news items pool expands.

3.2.2 Recommendation Quality

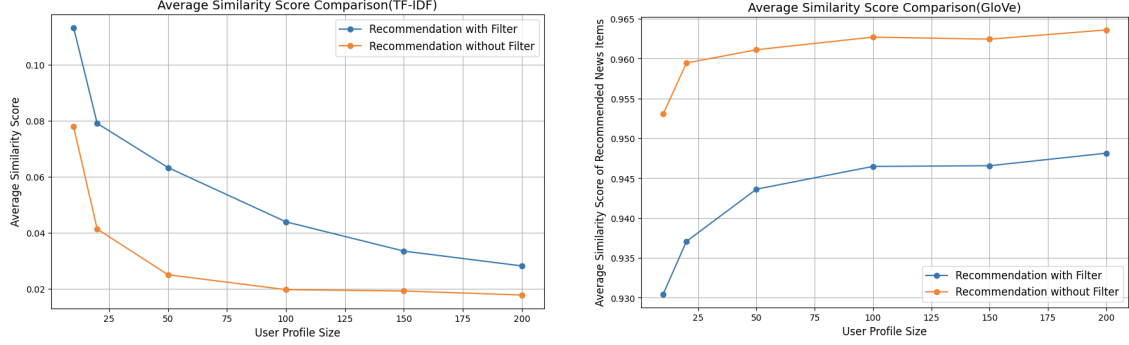


Figure 5: Similarity Scores Analysis of a News Recommendation System with and without Category Filter

To analyze the recommendation quality, we compare the average cosine similarity scores of recommended news items. As shown in Figure 5, the similarity measurements on TF-IDF embeddings demonstrate low and unstable average similarity scores, with most scores in the range from **0.02** to **0.1**. The poor performance arises likely because TF-IDF does not incorporate contextual information. To improve the recommendation quality, we transitioned to GloVe-based embeddings using the pre-trained glove.6B.300d.txt file, which provides 300-dimensional vector representations of words. This enabled us to leverage semantic and syntactic information captured from a large corpus of 6 billion tokens, generating similarity scores in the range from **0.93** to **0.96**.

Noticeably, the evaluation also highlights that the average similarity score is slightly lower when applying a filter, which is expected since the filter reduces the pool of news items to a more personalized subset. With the GloVe-based embeddings, the average similarity score without a filter remains consistently high, approximately **0.965**., compared to the filtered approach, which starts at approximately **0.93** (profile size of 10) and increases to approximately **0.945**. (profile size of 200).

While category filtering results in a slight decrease in similarity scores, the sacrifice is minimal and justified by the efficiency and focus gained through a more personalized recommendation process. This trade-off demonstrates that the filter effectively narrows the recommendation scope without significantly compromising quality.

4 Conclusion & Future Directions

Our project demonstrated the effectiveness of using machine learning and advanced embedding techniques to classify health-related tweets and develop a recommendation system. The neural network model outperformed other approaches, achieving the best accuracy for multi-class classification after addressing class imbalance. Despite slight overfitting, the model demonstrated strong performance in predicting minority classes, highlighting its potential for handling imbalanced datasets with additional refinement.

Building on the classification results, we developed a category-based recommendation system that filtered tweets by user profile categories and used cosine similarity of GloVe-based embeddings to recommend other contextually relevant health tweets. This approach significantly improved

runtime efficiency while maintaining high-quality recommendations. However, slight reductions in similarity scores with filtering suggest room for improving categorization and embeddings.

For future work, we plan to refine both components of the system. In classification, expanding the hyperparameter search space and exploring transfer learning with pre-trained models like GPT or XLNet can mitigate overfitting and enhance prediction accuracy. For recommendations, we intend to incorporate collaborative filtering, matrix completion, and reinforcement learning to enhance personalization, handle sparse user profiles, and enable a real-time feedback loop for dynamically adapting recommendations based on user interactions.

By integrating these enhancements, our project has the potential to evolve into a robust tool for classifying and recommending content efficiently, effectively tailoring results to user preferences and improving engagement with health-related information.

Contributions

With regards to the report and presentation, equal work was contributed by Yuki and Naiqi. In the report, Yuki wrote the Introduction, subsections of the methodology and results on the neural network and Random Forest, as well as designing the report structure. Naiqi wrote the subsections of the methodology, results on OCTs, XGBoost, and the recommendation system, and the Discussion and Future Directions. Both proofread the reports separately and worked on the presentation together. With regards to the technical tasks, both Yuki and Naiqi performed the data preprocessing as well as project organization. Yuki worked on the Random Forest and neural network classification models, while Naiqi worked on the XGBoost and OCTs. Both Yuki and Naiqi worked on formulating the recommendation system, and Naiqi wrote the code for it.

References

- [1] Abeed Sarker and Graciela Gonzalez. *Health News in Twitter*. 2015. UCI Machine Learning Repository. <https://archive.ics.uci.edu/ml/datasets/Health+News+in+Twitter>.
- [2] Gerard Salton and Christopher Buckley. *Term-weighting approaches in automatic text retrieval*. Information Processing Management, 24(5):513–523, 1988. [https://doi.org/10.1016/0306-4573\(88\)90021-0](https://doi.org/10.1016/0306-4573(88)90021-0).
- [3] Jeffrey Pennington, Richard Socher, and Christopher D. Manning. *GloVe: Global Vectors for Word Representation*. Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP). <https://nlp.stanford.edu/projects/glove/>.