LEARN-MATCH: Online Course Recommendation System

Zhiyu Wang
University of Florida
*G*ainesville, United State
wangzhiyu@ufl.edu

Praneeth Vempati
University of Florida
Gainesville, United State
pvempati@ufl.edu

***Abstract*—During the challenging times of the COVID-19 pandemic, online courses emerged as a beacon of opportunity, enabling uninterrupted learning and professional development. These courses bridged the gap created by physical distancing, providing accessible, flexible, and diverse educational resources to learners worldwide. However, as more online courses platform pop up such as Edx, Educative and Coursera. The course information are secrete in different places which makes user hard to find the best-match. To tackle this challenge, we designed a online-course recommendation system called LEARN-MATCH using K-means clustering. It integrates all the online learning resources together and customize recommendations according to the users' preference with a user friendly interface. Moreover, to improve the stability and performance of whole AI life cycle for the recommendation. We handle the potential risk and trustworthiness at different state with a performance monitoring.**

***Index Terms*—Recommendation system, K-means clustering, AI life cycle, Trustworthy AI, User interface**

I. Introduction

The COVID-19 pandemic brought about a dramatic increase in online learning opportunities, with thousands of courses becoming available to learners worldwide. While this created new possibilities for people to enhance their skills and knowledge from the comfort of their homes, it also introduced a significant challenge: finding the right course in an overwhelming sea of options. Many learners find it difficult to decide which courses match their goals, interests, and current level of understanding, often leading to frustration or wasted time on less suitable choices. This problem is especially important to address because online education is meant to be accessible and personalized, and without proper tools to guide learners, the benefits of this new educational era cannot be fully achieved Nagarnaik and Thomas, 2015.

Modern course recommendation systems are integral to online learning platforms, offering personalized suggestions to users based on their interests, preferences, and past behaviors (Ko et al., 2022). These systems utilize a combination of data-driven algorithms and user feedback to enhance learning experiences (Isinkaye et al., 2015). The limitation of existed course recommendation system is the recommendations can favor popular courses or established instructors, marginalizing niche topics or new creators (Lü et al., 2012; Puspasari et al., 2021) and lack of context awareness (Lu et al., 2015; Resnick and Varian, 1997).

To solve this problem, this work focuses on building a smart and reliable online course recommendation system. The goal of our system is to make it easier for learners to discover courses that match their needs, saving them time and helping them achieve their learning goals more effectively. We use K-means clustering to map the user input to our integrated database and offers the Top 3 recommendations based on the Euclidean distance of user input and data-points in our dataset (Shani and Gunawardana, 2011). As K-means clustering is known for the sensitivity for initial guess of centroids, outliers and the choice of K value, we carefully handle the risk of our AI system from the data processing until the model deploy with the

focus on trustworthy AI principles, such as being fair, transparent, and feedback collecting through a user-friendly interface (Kim and Ahn, 2008). Our system not only simplifies course selection but also ensures that learners have a positive and engaging experience, ultimately helping them make the most of the vast world of online education.

This report is organized by the following section: an review of related work, AI system design and implementation, trustworthiness and risk Management, evaluation and results, discussion, future work and conclusion.

II. Related Work

The course recommendation is not a new concept. For example, the Mondal *et al* proposed a course recommendation system based on the grade (Mondal et al., 2020). This paper proposes a machine learning approach to recommend suitable courses to learners based on their learning history and past performance. The framework first classifies a new learner based on their past performance using the k-means clustering algorithm. Collaborative filtering will be applied in the cluster to recommend a few suitable courses followed by a collaborative filtering. However, this recommendation system doesn't deal with semantic data carefully as we do. Their data processing process only contains the removal of punctuation and white space stripping. This work is only focus on the AI model which ignore the maintainance of whole AI cycle. Tan *et al* proposed a E-learning course recommendation system which helps learners to make choices without sufficient personal experience of the alternatives (Tan et al., 2008). Instead of using clustering-based recommendation, the use the Learner-to-Learner correlation recommender systems to recommend courses to a learner based on the correlation between that learner and other learners who have studied courses from the E-Learning site. This recommendation system requires the learning information and the smilarity comparision between the learner which is not veru efficient.

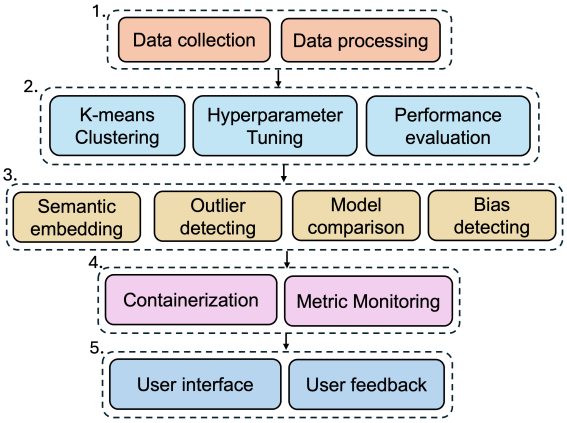

Fig. 1. Graphic illustration of AI system design: 1. Data collection and processing, 2. model design and evaluation, 3. risk management and trustworthiness, 4. Deployment and monitoring and 5. Human-computer interaction and feedback collecting.

III. AI System Design and Implementation

A. *System overview*

The course recommendation system is utilize the existed dataset in https://www.kaggle.com/. It contains the data from Udemy, Skillshare, Coursera and Edx with more then 20k data-points. We first processing the dataset by integrating the common features and discardind unimportant features to form a clean dataset for clustering purpose. The features are enumerating in different ways such as semantic embedding for the course title and label encoding for the difficulty level. Since the course recommendation doesn't hve groung truth label, we adopt the K-means clustering which is unsupervised learning algorithm. To find out the optimal K-value, we conduct the hyperparameter tuning using elbow (intertia) methods and silhoutte score. As K-means clustering is known for the sensitivity of outlier and initial centroids, we handle the risk by detecting the outlier, clustering stability with different initialization and clustering biasing detecting to improve the trustworthiness of the model. The implement of trustworthiness also realized by the semantic embedding for the text feature such as course title. To ensure the AI system running consistently, we use Docker to containerize our application and connect

it with Grafana and Prometheus for the system performance monitoring. At last, we use Streamlit to design a user-friendly interface to collect user input and display the recommendation result. The users could also provide their feedback for us for further improvement and delayed performance monitoring, as it shown in **Figure 1**.

B. AI life cycle

Data Collection and Preprocessing We collect around 20k data points from online dataset provided by Kaggle. It contains the data from Udemy, Skillshare, Coursera and Edx with different features. In current AI system, we only consider the most relevant features such as the "difficulty level", "language", "title" and "crediteligibility" 4 features for all the data-points for the basic recommendation system. To maintain the relevancy between the course title for the clustering, the one-hot encoding and label encoding are not suitable. We use the TfidfVectorizer implemented in sklearn.featureExtraction.text to vectorize the course title and use the cosin similarity to assign a value based on the similarity threshold = 0.5. For other 3 feature, we simply apply the label encoding to complete the data enumerating.

Model Development and Evaluation We select K-means as our clustering algorithm, since the recommendation system has no label and ground truth. For K-means clustering, the only most important hyperparameter is the choice of K. To select the optimal K value, we adopt two method to compare the results. The first method is the inertia method. Elbow Method examines the sum of squared distances from each point to its assigned cluster center (inertia). The idea is to find the "elbow" point where adding more clusters does not significantly reduce inertia. Another more advanced method is silhoutte score to measure the compactness of each cluster. The value of silhoutte score can be vary from -1 to 1. A value close to 1 means data point is well-matched to its own cluster and poorly matched to neighboring clusters. Therefore, we compare these two methods and choice of optimal K value as I shown in **Figure. 2a and 2b**. The inertia give the optimal K equals to 4 and the silhoutte score gives the optimal K equals to 2.

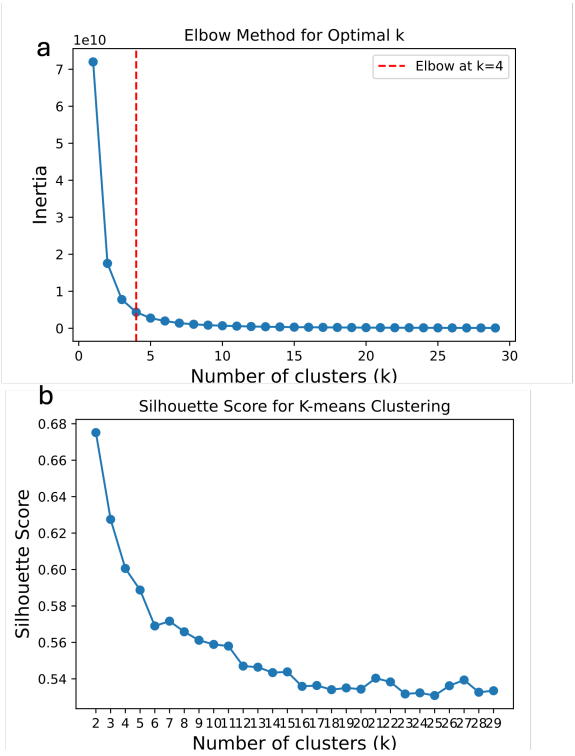

Fig. 2. Hyperparameter tuning for choice of K. a: the inertia value of k value range from 2-30. The optimal choice of K is 4. b: The silhoutte score of k value range from 2-30. The optimal choice of K is 2.

Sine the inertia methods and silhoutte score gives different optimal choice of K. We visualize the clustering result use the Principle Component Analysis to project the data to 2-dimension (2D). From the results show in **Figure 3a and 3b**, the K=2 could well-separated the 2 clustering even without using K=4. From the computational cost and overfitting considering, we decide to choose K = 2 as the optimal K for our model.

Deployment Strategy Since the K-means clustering is not a complicated model, we can easily host it in local machine. Therefore, our deployment of model is in local with Docker for the containerization. The Docker files ensure the consistent deployment of our model by including the

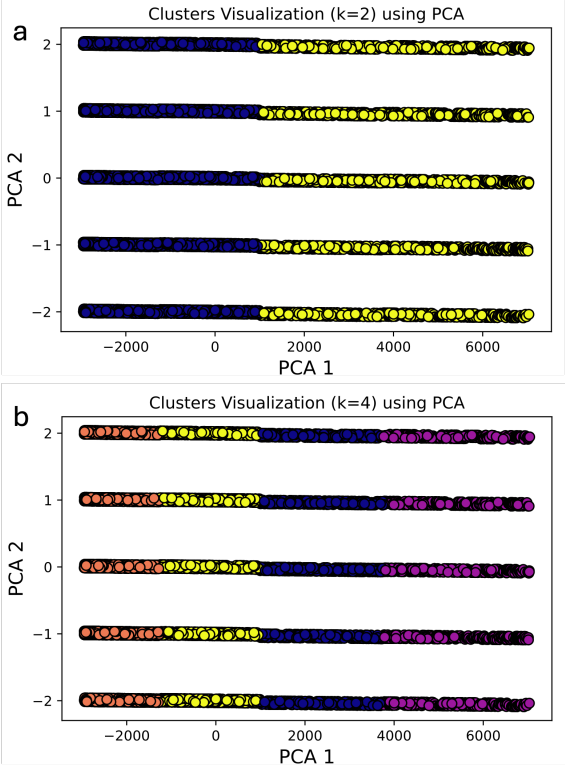

Fig. 3. Hyperparameter tuning for choice of K. a: the inertia value of k value range from 2-30. The optimal choice of K is 4. b: The silhoutte score of k value range from 2-30. The optimal choice of K is 2.

requirement.txt for all the dependencies and the integration with Grafana and Prometheus. Grafana and Prometheus are the platform to monitoring system performance and gives alert for potential model drifting.

C. *Human-computer Interaction (HCI)*

To realize the human-computer interaction, we use Streamlit to further implement our data processing, clustering and recommendation process. It takes the user input through our designed interface. For example, their interested topic, preferred languages, difficulty level and crediteligibility into text input or button selection. The user input will be added to the last row of our original dataset to complete identical processing and enumerating procedure. A two-class K-means clustering is adopted for each user input based on previous justification. Since the recommendation is based on each-time user input, the clustering has to perform every time unlike the inference on pretrain model. The top 3 datapoints which are closest to the user input data points will be selected as display with course title and the link directing to the webpage. After the display of the recommendation result, we will collect the user feedback as "satisfied" and "unsatisfied" as a record.

IV. Trustworthiness and Risk Management

Data processing stage Since the clustering algorithm is to cluster the data point based on the similarity, the "course title" is a challenge for the data enumerating. the "course title" is a long phrase or short sentence which requires the enumerating maintains the semantic relevancy. To resolve this issue, we adopt TfidfVectorizer to transforms the processed titles into numerical representations based on Term Frequency-Inverse Document Frequency (TF-IDF), which measures how relevant a word is to a document in a collection. Then using the cosine similarity, it compare the similarity pairwise and assign a unique number based on the threshold. In this way, the enumerating of "course title" won;t be some random number but the number reflect the content relevancy.

Model development stage Since the K-means is known for the sensitivity of initial guess of centroids and the outliers. I first apply the z-score ¿3 to detect the outlier in the dataset, as the Z-score tells us how many standard deviations a data point is from the mean. By apply this critria, there is no outlier has been detected.

To measure the sensitivity of initial centroids to the clustering result, I track the silhouette score with 10 different centroid initialization. With 10 different initialization, the average silhouette score is 0.6750128756787486 and standard deviation of silhouette scores is 0.00013463614078702003. It indicates even the K-means clustering is sensitive to initial centroids but it could offer a stable results. Similarly, we also monitor the label stability of 10 runs of clustering use Adjusted Rand Index

(ARI) which measures the similarity between two clustering result. From the **Figure 4**, the mean value of ARI is 0.726. It indicates the clustering result is relatively stable. To detecting the potential clustering bias, we measure the number of datapoint with K=2 and K = 4, the results indicated there is no obvious bias in the model since the number of data points are relatively similar in **Table 1**.

TABLE I
BIASING DETECTING (RANDOM SELECTED 10K DATA)

K value	Datapoints in each cluster
K=2	6423, 3527
K=4	1742, 1290, 4458, 2510

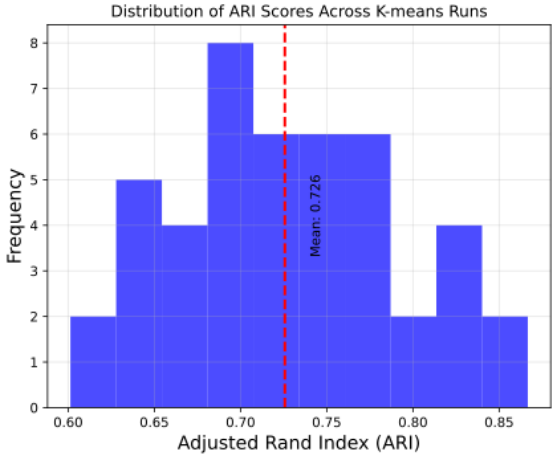

Fig. 4. The distribution of the ARI score for the 10 runs of K-means clustering. The mean value is 0.726.

Since K-means has other variants such as mini-batch K-means, we also compare the optimal number of clusters using silhouette score and the execution time along with different number of K. From the plot in **Figure 5**. From the track of silhouette score, the optimal choice of K is 2 for both of the model and the values for silhouette score are similar for both methods. For the execution time, the mini-batch Kmeans doesn't show obvious advantage over the K-means, therefore we stick to the K-means clustering in this work.

Deployment stage For the model deployment, we use the same K-means clustering algorithm as

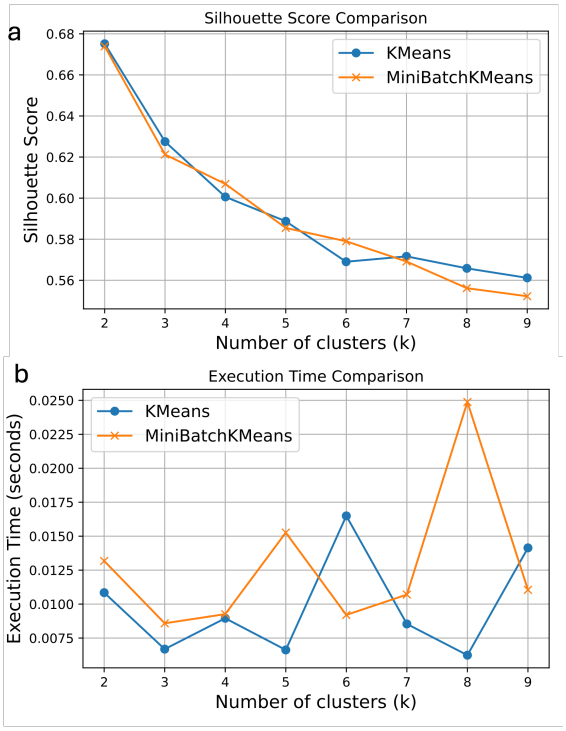

Fig. 5. The distribution of the ARI score for the 10 runs of K-means clustering. The mean value is 0.726.

we designed with K=2. As we take the user input as a part of dataset for the data processing and enumerating, to maintain a reproducible clustering result, we will delete the user input datapoint after the recommendation is completed to maintain a clean dataset for further use.

Risk Management Framework (RMF) The residual risk for the course recommendation system can be managed using the table bellow: The first step in the RMF is to identify potential risks that may affect the course recommendation system. These risks can be categorized into several areas:

- Data Privacy and Security Risks: Sensitive information such as user profiles, browsing history, and course preferences can be exposed if not properly secured.
- Algorithmic Bias: The recommendation algorithm may unintentionally favor certain courses, instructors, or content types, leading

to biased suggestions.

- System Downtime or Failures: Technical issues such as server outages or database failures could disrupt the availability of the system.
- User Experience Issues: Poor interface design, slow response times, or irrelevant recommendations can frustrate users and decrease engagement.

Risk Mitigation After assessing the risks, the next step is to implement strategies to mitigate them. Mitigation efforts may include:

- Data Encryption and Access Controls: To prevent unauthorized access to user data and comply with privacy regulations.
- Regular Audits and Testing: To identify potential biases in the recommendation algorithm and make adjustments as necessary.
- Redundancy and Backup Systems: To ensure that critical systems are available in case of technical failures.
- Improved User Interface (UI) and UX Design: To enhance user satisfaction and engagement by making the platform intuitive and responsive.
- Legal and Compliance Reviews: To ensure adherence to data protection and privacy regulations.
- Load Balancing and Server Scaling: To ensure that the system can handle increases in traffic and prevent performance degradation.

V. Evaluation and Results

Performance Metrics Since the K-means clustering has to performance each time without using any pretrain model, we plot the optimal choice of K values from using two different user input as the representation for the performance metrics user input in **Figure 6a** is "Python", "All levels", " English" and "No". The user input for in **Figure 6b** is "Wed design", "intermediate", " English" and "yes". As we can see, both of the clustering result gives the K=2 as the optimal choice of K which indicates the K=2 is correct choice of k hyperparameter which isn't altered by different user input.

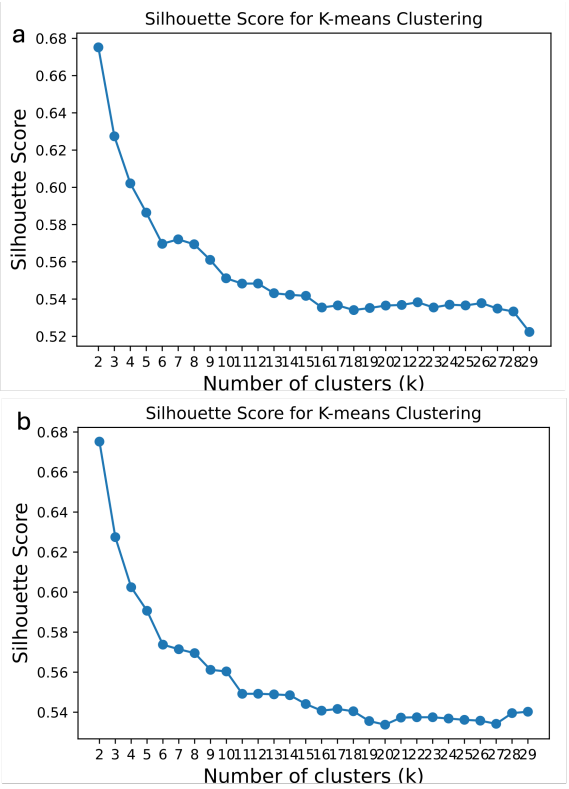

Fig. 6. silhouette score for two user input.

Monitoring and Feedback The AI system performance monitoring in this work is using the total run time of each course recommendation and the run time for the clustering part. In this way, we can evaluate if the run time has dramatic increased or decrease at each time with the dashboard in Grafana and alerting in Prometheus if the run time is deviated more than 20%. In the **Figure 7**, we plot the clustering run time for the recent 10 real world user input. From the figure we can see, the average time for the clustering is from 0.88 second to 0.92 second. Therefore, for these 10 runs, there is no obvious runtime length are monitored.

Besides the monitoring for the runtime, we also save the user feedback as a .csv file by clicking two button "satisfied" and "unsatisfied". If the system is keep getting "unsatisfied" record. That's also a allert for potential model drifting or the indication

of adding new data.

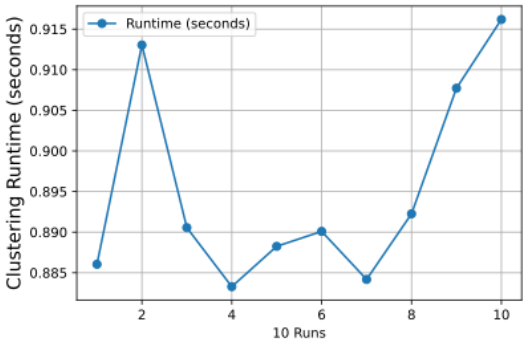

Fig. 7. The monitoring of the clustering runtime for each of 10 user input

VI. Discussion

In this work, we designed a online course recommendation system which integrates the online learning resources at one place. It greatly satisfy people's need for find the suitable course in a short time, which is hasn't achieved before. This recommendation system is powered by the K-means clustering for the unsupervised learning task. It takes user input as one data point and combine it with the dataset we obtained from Kaggle. The combined dataset will be processed and enumerated together which maintains the semantic meaning among the "course title" to ensure the clustering could reflect the relevancy between the datapoints. This is the one of the challenging part of data preprocessing step. To resolve it, we found the skit-learn has a function called TfidfVectorizer which can extract the text feature and maintain the semantic relevancy after the transformation. Therefore we combined TfidfVectorizer with Cosine similarity for the text feature enumerating. However, later we found this vectorization and pair-wise similarity calculation requires good amount of computational time that we need to improve it later. The recommendation is based on the Euclidean distance of the user input datapoint with the neighborhood datapoints. The top 3 recommendation is given by the closest three neighborhood datapoints.

Unlike the AI model development, this work stand on the AI life cycle level to manage all kind of risks at different stages and monitoring mechanism for the model deployment. We considered the potential risk from data processingc stage until the deployment stage to reduce the vulnerability of the system. We examine the model sensitivity to the outliers, to the initial guess of centroids. We also compare the K-means with mini-batch K-means with the optimal K value and the execution time. We also considered the potential clustering biasing by looking into the number of datapoints inside each cluster.

We use docker to containerize our applications and all required dependencies. It also allow the stable system monitoring by connecting with Grafana for the metric display and the essential alerting with Prometheous. The docker provide a stable environment and security for the deployed system and allow the system performance monitoring based on it. Overall, during this work, we developed a AI system for course recommendation. It's not only a AI model or application but a AI system to find the best course for you.

VII. Future Work and Improvements

Data updating from API The dataset we currently adopt is from the Kaggle. The dataset is static without any real-time updating. However, to realize the efficient and up-to-date course recommendation, we need to have the access with each online course website like API to keep getting the new courses they update and the old course they may remove. It ensures the information we have in our recommendation system is valid and up-to-date.

Advanced model In current work, we only record the user feedback as a performance monitoring metric. To improve the user experience, we plan to offer a advanced clustering by getting more user input features, if the user doesn't satisfied with the current recommendation result. The advanced clustering will including the features such as the duration of lecture, the instructor of the lecture, the course if free or not to take care the users interest carefully. At the same time, the computational cost will be also scaled due the increased fea-

tures. Therefore, we consider to make the advanced recommendation system non-free to facilitate our resource allocation.

Similarity measurement In current work, we are using the Cosine similarity to do the pair-wise comparison to assign a value for each "course title". Any pair-wise comparison algorithm requires the quadratic run time respect to the number of samples. To reduce the computational cost for the "course title" enumerating, we need to identify if there any similarity measurement algorithm could offer the time complexity near linear time. That could dramatically reduce the overall computational time for each recommendation.

VIII. Conclusion

In this work, we used clustering-based recommendation system to map the user input to the online course dataset. This recommendation system taking the user input as the a datapoints to perform the clustering then recommend the course based on the distance (smilarity) of the datapoints. It greatly considers users preference without taking redundent user information for the privacy risk. Before the clustering, the critical step is the data enumerating. To maintain the semantic relevancy between the data point for "course title" feature, we utilize the TfidfVectorizer and cosine similarity to handle the challenge. To overcome the sensitivity for the initial centroid guess and outliers, we first detect the possible outliers use the Z-score then track the clustering stability using the silhouette score. For the optimal choice of K, we compare the inertia method with the silhouette score. After the visualization, we found the K=2 is the best choice. After the model development, we deployed our model with a user-friendly interface where user can type-in or click button to provide what kind of course they want. We also handle the potential vulnerability of the system by monitoring the run time and user feedback using Grafana and Prometheus in the container to keep our AI system running stable and consistent.

References

[1] G. Eason, B. Noble, and I. N. Sneddon, "On certain integrals of Lipschitz-Hankel type involving products of Bessel functions," Phil. Trans. Roy. Soc. London, vol. A247, pp. 529–551, April 1955.

[2] J. Clerk Maxwell, A Treatise on Electricity and Magnetism, 3rd ed., vol. 2. Oxford: Clarendon, 1892, pp.68–73.

[3] I. S. Jacobs and C. P. Bean, "Fine particles, thin films and exchange anisotropy," in Magnetism, vol. III, G. T. Rado and H. Suhl, Eds. New York: Academic, 1963, pp. 271–350.

[4] K. Elissa, "Title of paper if known," unpublished.

[5] R. Nicole, "Title of paper with only first word capitalized," J. Name Stand. Abbrev., in press.

[6] Y. Yorozu, M. Hirano, K. Oka, and Y. Tagawa, "Electron spectroscopy studies on magneto-optical media and plastic substrate interface," IEEE Transl. J. Magn. Japan, vol. 2, pp. 740–741, August 1987 [Digests 9th Annual Conf. Magnetics Japan, p. 301, 1982].

[7] M. Young, The Technical Writer's Handbook. Mill Valley, CA: University Science, 1989.