

Continual Optimization of In-Production Machine Learning Systems Through Semantic Analysis of User Feedback

Hemadri Jayalath¹^a, Ghadeer Yassin¹^b, Lakshmish Ramaswamy¹^c and Sheng Li²^d

¹*School of Computing, University of Georgia, Athens, GA, U.S.A.*

²*School of Data Science, University of Virginia, U.S.A.*

Keywords: Machine Learning, MLOps, ML Maintenance, Clustering, Categorization, User Feedback for Learning.

Abstract: With the rapid advancement of machine learning technologies, a wide range of industries and domains have adopted machine learning in their key business processes. Because of this, it is critical to ensure the optimal performance of operationalized machine learning models. This requires machine learning models to be regularly monitored and well-maintained after deployment. In this paper, we discuss the benefits of getting human guidance during the machine learning model maintenance stage. We present a novel approach that semantically evaluates end-user feedback and identifies the sentiment of the users towards different aspects of machine learning models and provides guidance to systematize the fixes according to the priority. We also crawled the web and created a small data set containing user feedback related to language models and evaluated it using our approach and uncovered some interesting insights related to language models. Further, we compare the trade-offs of alternative techniques that can be applied in different stages in our proposed model evaluation pipeline. Finally, we have provided insights and the future work that can be done to broaden the area of machine learning maintenance with human collaboration.


1 INTRODUCTION


Over the past decades, machine learning (ML) has become one of the most successful areas which disclosed lots of undiscovered potential in the computer science field. This opened up many research possibilities, not only in the computer science field but also in many other domains. However, operationalizing different research problems and maintaining the operationalized models is not an easy task. In many cases, the complexity of the ML model grows with the complexity of the problem. This makes the maintenance of the ML model hard. Poorly maintained operationalized ML models cause many problems in the long run and can become stale (Sculley et al., 2015) due to the changes in the environment.


To effectively monitor and maintain complex ML models, collaborative research between ML, and domains which guide towards the development of better, usable applications, such as Software Engineering


(SE) and human-computer interaction (HCI) started to emerge (Sculley et al., 2015), (Inkpen et al., 2019). From these collaborations, ML models were able to gain many advantages such as better usability, reliability, and maintainability. In general SE, there are many practices to assist the maintenance stage of an operationalized application. As an example, continuous delivery, continuous testing, gathering feedback continuously, and user behavior monitoring (Amaro et al., 2022) are some of the prominent practices and they ensure a smooth, rapid and continuous delivery process that leads to a well-maintained application. These practices have been established over decades and DevOps practitioners follow these practices to monitor and maintain the deployed software application. However, many of these practices are fairly new to the ML platform, and directly mimicking the DevOps practices as it is for ML maintenance, is not successful.

In the monitoring stage of an operationalized ML model, there are several aspects that should be continuously monitored. Such as input data, the accuracy of the model predictions, model biases, infrastructure, and the condition of the upstream data sources. To explain further with an example, if the upstream data

^a <https://orcid.org/0000-0001-7073-522X>

^b <https://orcid.org/0000-0001-5135-7622>

^c <https://orcid.org/0000-0002-4567-4186>

^d <https://orcid.org/0000-0003-1205-8632>

source is not properly monitored and maintained, the sensors that collect data to feed the operationalized ML model can wear out and can feed incorrect data to the ML model. These issues cause inaccurate operationalized model outcomes and harm the credibility of the model. Continuous monitoring and updating of all these aspects accurately can help to overcome the performance degradation of the operationalized models. However, updating operationalized ML models very frequently is not realistic. Updating an ML application is far more expensive than updating a general software application. Finding relevant data and retraining a model can take days or even weeks. Hence, finding the sweet spot of retraining the model is also a critical issue. Updating too frequently is expensive and updating less frequently leads to poor performance.

To tackle these challenges, the ML monitoring and maintenance area has increasingly attracted research attention in the past few years. ML operations (MLOps) (Sculley et al., 2015) (Alla and Adari, 2021) is one of the interesting research directions that has emerged in this field. MLOps aims to assist in the maintenance of operationalized ML models. One of its main goals of it is to rapidly adapt the operationalized ML models according to the user needs (Mäkinen et al., 2021). This adaptation includes introducing new features, fixing bugs, improving the current features, etc. To make the maintenance smoother, MLOps mimics some of the DevOps practices, and it introduces ML-specific additional practices as well (Mäkinen et al., 2021).

In our research, we study the importance of incorporating human guidance during the model maintenance stage. We provide an automated guidance framework that MLOps engineers can use for monitoring the operationalized ML model. It analyzes the end-user feedback related to operationalized ML models and prioritizes the issues to be addressed. The technical contribution of this paper can be summarized as follows. 1.) We highlight the use of harnessing user feedback to identify user expectations from operationalized ML models. 2.) We provide a novel approach to identify the order of the issues of operationalized ML models by performing a semantic analysis of the end user feedback and examining how widespread the issues are. 3.) We conducted a set of experiments on harnessing user feedback using our approach. 4.) We present a manually collected and evaluated small dataset of user feedback related to the language models (LM) and utilize it in our approach to examine the insight we can get regarding the user LMs.

2 RELATED WORK

As mentioned by Sacha et al. (Sacha et al., 2017), proper and effective use of human guidance can be used to evaluate and exploit the full potential of the ML models and to give a better user experience. Human guidance can be used in different ways in different stages of the ML development life cycle. Visual Analytics (Hohman et al., 2018) and interactive ML are some interesting areas that use domain expert guidance to steer the different phases of the ML model development life cycle. In visual analytics, visualizations can be done in different phases of the software development life cycle. As an example, Arbesser et al. (Arbesser et al., 2016) introduce a visualization approach called Visplause for supporting data quality assessments. Visplause inspects the data quality problems for different time series, compares them, and summarizes the results so that experts can analyze the distributions and detect anomalies and noises.

FROTE (Alkan et al., 2022) and Amazon SageMaker Model Monitor (Nigenda et al., 2021) are two recent research works that specifically focus on maintaining deployed ML models. Both of them use user-provided rules to update deployed ML models. FROTE is focused on pre-processing the training dataset based on feedback rules. For a given tabular dataset, they produce an augmented dataset that has a better alignment with the feedback rules assigned by the domain experts. Amazon SageMaker Model Monitor continuously monitors the ML models hosted on Amazon SageMaker. This, as well, periodically analyzes the data collected from the production environment and checks if it adheres to the rules provided by the users/ domain experts.

In the existing human-involved approaches, one of the main limitations is the higher cost of manual labor. During the stage that involves human-machine collaboration, humans have to be continuously involved in the process (Jayalath and Ramaswamy, 2022). In this research, we address that limitation and effectively acquire human guidance while easing the burden of the human. We believe analyzing the user feedback of the end users is the key to achieving this goal. Also, because of the convenience of using the natural language, end users tend to give a lot of feedback regarding their experiences expecting improvements. And, with the advancement of content-sharing platforms such as blogs, YouTube, and social media platforms such as Twitter, many people tend to review and publish their user experiences. At the same time, these discussions can be found in many different forms. They can be blog posts, tweets, comments,

videos, forum posts, or example scenarios of a model failing. Doing a proper analysis of these discussions will reveal what users expect from the operationalized ML models and how to improve them.

Even though the end user and machine communication area have attracted the attention of the research community, evaluating the user feedback regarding ML models still has a lot to explore. However, evaluating user feedback has several significant challenges. User feedback can generally be chaotic and unstructured. Because of that, having a uniform way to evaluate them can be very challenging. Also, the heterogeneity of the user feedback makes the evaluation challenging as well. In addition to that, when there is a plethora of user feedback, manually examining them and finding which feedback should be addressed first can be highly time and labor-consuming.

The overall objective of our paper is to see the operationalized ML models from the perspective of the end user without having the end users involved in the process continuously. To achieve this, we analyze the feedback of the end users and find out what are the major discussion topics of the users regarding an operationalized ML model and what is the average user sentiment regarding them. We will discuss our approach in detail in the next sections.

3 DESIGN AND SYSTEM ARCHITECTURE

In our research, we are focused on evaluating the textual end-user feedback that is provided in the natural language. This can be an end-user discussion of the operationalized ML model, a comment from the end user, or an example scenario that the end user has provided. Our approach takes a set of unrefined end-user feedback related to an operationalized ML model and analyzes them and figures out which issues should be addressed. Figure 1 further explains how our proposed approach can be used in an operationalized setting.

When analyzing the user feedback using our approach, we specifically focus on several aspects. First, we identify the main topics of the end-user discussions. Also, we cluster the user feedback and examine the cluster size to see how widespread each identified aspect is and how many users are affected by each aspect. As well as that, we analyze the average sentiment of the end users regarding those aspects to see if the user sentiment is positive or negative.

This can be used as a guidance framework to prioritize the issues to get an understanding of which issues to be addressed first. To explain further with

an example, take an operationalized model to detect spam emails. Day by day the scammers change their strategies and try to come up with more realistic and sophisticated emails. A few years back, phishers tried to catch people by faking to offer an unrealistic amount of money to the reader. But nowadays they use more advanced methods. Such as, pretending to act like they are from a bank or a government agency. If the model is trained on past data and has not been updated, it will fail to detect the current spam emails and the end users will be frustrated. By evaluating the feedback of the end users, we can identify what kind of emails our operationalized ML model fails to identify.

In our approach, unrefined end-user feedback undergoes several stages to reach the final analysis. Figure 2 represents the high-level architecture of our approach and it exhibits the major stages of the approach. To explain the high-level architecture of our approach briefly, we first get a set of unrefined user feedback and preprocess them. After that, we simultaneously summarize each end user feedback and analyze the sentiment of each user feedback. Then, we perform feature extraction and cluster the embedded user feedback space. And then we simultaneously perform topic modeling for each cluster to identify what the users are talking about in each cluster. Also, we calculate the average sentiment for each cluster. After that, we evaluate the results to see what kind of insights we can get from our cluster space. As an example in the spam emails detecting operationalized ML model of the example that we discussed in the previous paragraph, we will get a large cluster of users that discuss about emails that pretended to be from a bank, and the average sentiment of that cluster will be low.

To experimentally see if our approach identifies the user issues accurately, we selected the LM domain. Recently, LMs have become a hype because of the various advantages they offer. However, a simple web search regarding the LMs will reveal how many LM related discussions are there regarding the LM user experiences and the challenges. Even if many researchers started to evaluate LMs in different ways, the focus of the research community on these direct end-user discussions is low. After building the pipeline of our approach, we evaluate the end user feedback related to LMs to observe if our approach can extract meaningful information from the end user feedback.

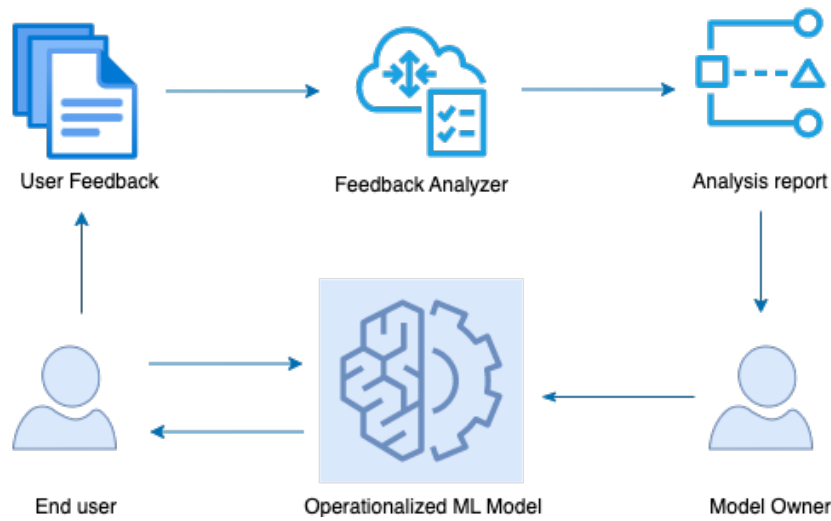


Figure 1: Proposed Approach - End users use the operationalized ML model and give textual feedback related to the model using natural language. The feedback analyzer analyzes the feedback and generates an analysis report. Model owners/developers/MLOps engineers can use the analysis report as guidance and decide which issues to be addressed first.

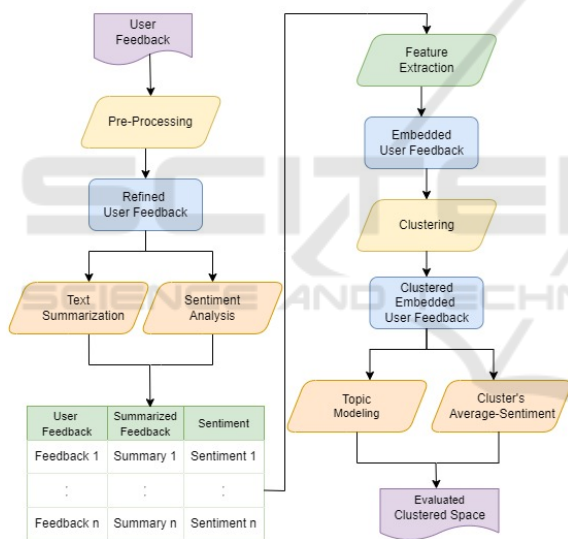


Figure 2: High level architecture of the proposed approach.

4 DATASET

Despite the availability of a wide range of user feedback on the ML models on the internet, finding a dataset that includes those feedback is very challenging. Therefore, we use different datasets that include user feedback on different domains to evaluate our approach.

20NewsGroup is a collection of approximately 20,000 newsgroup documents which are partitioned almost evenly across 20 different newsgroups.

The amazon-Alexa-Reviews dataset contains

nearly 3000 Amazon customer reviews, star ratings, date of review, variation, and feedback of various amazon Alexa products like Alexa Echo, Alexa Firesticks, etc.

After conducting the experiments with existing datasets, we evaluate our pipeline for a dataset related to the user feedback for the LMs. For this, we crawled the web that contains user reviews about the different LMs and manually created a dataset of end-user feedback related to the LMs. The dataset was also manually evaluated to get the sentiment of each user feedback. We call this data set the LM user feedback dataset. As major classes, the LM user feedback dataset includes the end-user reviews, manually identified topics, and manually evaluated sentiment.

5 EVALUATING THE LM DATASET

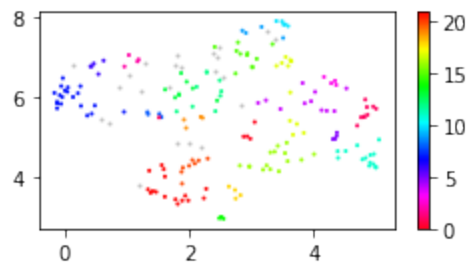


Figure 3: LM dataset clusters.

When manually evaluating the LM dataset, we evaluated the main topic of each end user feedback along

with the sentiment of the end user. Then, we evaluated the dataset using our approach. Figure 3 shows the clusters created by our approach for the LM dataset. Altogether there are 22 clusters. Since the LM dataset is very small, the clusters are very sparse.

After clustering, the next step is getting insights about the clusters. In table 1 we have summarized insights about the clusters related to the LM dataset. Some example insights we can get related to LMs through the analysis we performed using our approach are, 1.) Many people are discussing the writing skills of LMs. And on average users have a positive sentiment regarding this. 2.) Many people are talking about the racism related to LMs and they have on average a lower sentiment regarding this. 3.) The sentiment of the users related to the abilities of LMs regarding the biomedical domain is also low. 4.) Many people had bad experiences with LMs being toxic. After that, we took the cluster data points and the ground truth category and tried to name the categories manually. And the manual topics were very similar to the predicted topics.

Table 1: LM dataset insights for first 5 largest clusters.

Size	Predicted topics	Sentiment
20	writing, articles, generate, write, completion	4
16	racist, black, white, stereotypes, people	2.5625
15	toxic, stereotypical, bitter	2.4
14	biomedical, corpora, domain	2.78
12	common, sense, knowledge, understanding	3.25

6 BUILDING THE PIPELINE

In this section, we discuss comparisons of different techniques we used in each stage, reasons behind picking specific techniques and turning down the others, the evaluation techniques we performed, and how our approach performed in each stage.

6.1 Text Summarization

After preprocessing, the first step we performed on refined user feedback is applying a text summarization technique. For that, we used Huggingface transformers (Wolf et al., 2020). Hugging Face is a platform that contains various models and libraries that can be used for NLP tasks. It has BERT based models that

Table 2: Intrinsic measures for the cluster approaches.

	K-Means	HDBSCAN
Silhouette	0.41431	1.0
Calinski	5096.20	11735639222069.18
Davies	1.17167	1.92258

perform NLP tasks like summarizing very well.

6.2 Experimental Comparison of Clustering

As the first clustering technique, K-Means was selected to perform nearest neighbor based clustering. The second technique we use is Density-based spatial clustering of applications with noise, DBSCAN (Ester et al., 1996). This is a clustering algorithm that can discover clusters of any arbitrary size or shape in datasets that even contain noise and outliers. Hierarchical Density-Based Spatial Clustering of Applications with Noise (HDBSCAN) (Campello et al., 2013) is the third technique. This clustering algorithm extends the DBSCAN algorithm by converting it into a hierarchical clustering algorithm and then extracts a flat clustering based on the stability of clusters.

To determine the quality of the clusters we incorporated three Intrinsic Measures that do not require ground truth. Silhouette Coefficient (Rousseeuw, 1987), Calinski-Harabasz Index (Caliński and Harabasz, 1974) and Davies-Bouldin Index (Davies and Bouldin, 1979). All three evaluations were performed using the metrics package in sklearn on the 20NewsGroup dataset. Table 2 shows that HDBSCAN outperforms K-Means in both Silhouette and Calinski Measures. DBSCAN gave very similar results to HDBSCAN. However, DBSCAN takes the distance threshold as a user defined parameter. But in our research how user feedback scatters in the vector space is not predefined. Hence we wanted to systematically ascertain the distance threshold. HDBSCAN finds clusters of variable densities without having to choose a suitable distance threshold first. Because of that, we choose HDBSCAN as the most suitable clustering technique for our approach.

6.3 Semantic Evaluation of Clusters

The next step is evaluating if the clusters are semantically accurate. For this, we use the topic column in our 20 newsgroup dataset as the ground truth. Figure 4 represents the flow of our semantic analysis process. In this process, we group the data points of each cluster by topic and find the topic of the majority. If several topics have many different data points

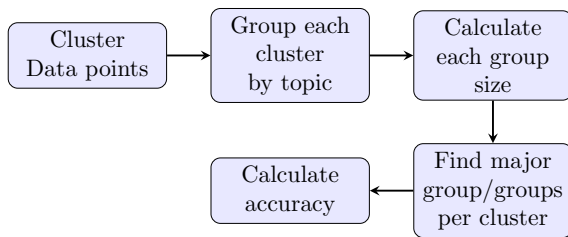


Figure 4: Semantic accuracy calculation.

in one cluster and if those topics are very similar, we consider all such topics as the major topic of the cluster. As an example in the 20 newsgroup data set there are similar topics like soc.religion.christian and talk.religion.misc in one cluster, both are related to news regarding religion. After that, we consider the data points related to these categories as the correctly classified points and the rest as the misclassified points. Then the accuracy of the predictions is calculated considering all data points and the correctly classified data points. Table 3 shows the results of the semantic accuracy analysis of the three largest clusters of the 20 newsgroup data set.

Table 3: Semantic Accuracy Analysis of the largest 3 clusters per major groups.

Major Group/s	Cluster Size	Accuracy
comp.windows.x, misc.forsale, comp.sys.mac.hardware comp.sys.ibm.pc.hardware, comp.os.ms- windows.misc, comp.graphics, misc.forsale, sci.electronics	5781	93%
soc.religion.christian, alt.atheism, talk.religion.misc	1514	92.4%
rec.motorcycles, rec.autos	1200	93.8%

6.4 Feature Extraction

Feature extraction is an essential part of the approach to derive a good cluster-friendly representation of user feedback. For this, we use several data transformation techniques. Namely, Doc2Vec (Le and Mikolov, 2014), RoBERTa (Liu et al., 2019) and msMarco-bert-base-dot-v5 sentence transformers model (Reimers and Gurevych, 2019).

To compare the performance of feature extraction techniques, we mainly consider two aspects. Computational time and semantic accuracy. Figure 5 shows

the computational time comparison between the three feature extraction techniques. Flair embedding took a significantly high training time compared to the other two techniques.

After that, we compare the semantic accuracy between the doc2vec and the hugging face techniques to pick the techniques that perform well in terms of time and accuracy. For this, we performed feature extraction from both techniques separately. Since 20news-group has 20 classes we tune the hyperparameters when clustering, to get an optimum cluster number closer to 20. Then we performed clustering using the HDBSCAN clustering. And checked the average semantic accuracy of all the clusters created after using these two feature extraction methods.

For Doc2Vec, the optimal number of clusters we can achieve by tuning the hyperparameters was 17. For Hugging face the optimum number we achieved was 18. Doc2vec clusters were mostly mixed with data points belonging to different categories. But, hugging face clusters had clear separations. And also when evaluating the semantic accuracy, hugging face has an accuracy of 0.90% and doc2vec had a very low score which is 0.22%. In terms of computational time and accuracy, we picked a hugging face transformer to conduct the feature experiment of our pipeline. Figure 6 shows the visualization difference after performing two feature extraction methods.

6.5 Topic Modeling Results

In this stage of the pipeline, our goal is to get an understanding of what each cluster represents. We conducted experiments with three different topic modeling techniques in this stage. Namely, Latent Dirichlet allocation (LDA) (Blei et al., 2003), Non-negative Matrix Factorization (NMF), and c-TF-IDF .

To evaluate the topic modeling, we use the 20NewsGroup dataset. After conducting our pipeline until the topic modeling phase, we do comparisons to see if the topics we create for our approach are relevant to the topics that are given in the dataset. The topic modeling techniques we compared are LDA, NMF, and c-TF-IDF. Table 4 shows the comparison of the topics generated by LDA, NMF, and c-TF-IDF of the largest three clusters. The performance of all three techniques was good. However, for some clusters, c-TF-IDF performed slightly better and also since c-TF-IDF is specifically adapted for multiple classes. Based on that and because of the simplicity of c-TF-IDF, we decided to stick with c-TF-IDF to perform topic modeling in our pipeline.

Table 4: Topic Modeling results for the largest three clusters.

Actual Categories	Majority	LDA First five suggested topics	NMF First five suggested topics	c-TF-IDF First five suggested topics
comp.windows.x, comp.windows.x, misc.forsale, ibm.pc.hardware		graphics, hardware, windows, card, driver	card, video, window, screen, program	windows, dos, drive, scsi, software
soc.religion.christian, alt.atheism, talk.religion.misc		mormon, religion, church, organ, subject	mormon, church, subject, organ, line	god, jesus, christian, bible, church
rec.motorcycles, rec.autos		detector, radar, rec, autos, law	radar, car, owner, speed, detect	car, bike, dod, cars, engine

6.6 Sentiment Analysis Evaluation

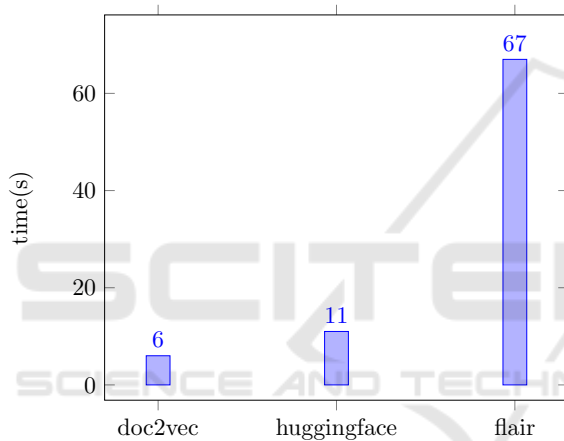


Figure 5: Computational time comparison.

To evaluate the sentiment analysis, we use the Amazon-Alexa-Reviews data set that we mentioned above. The bert-base-multilingual-uncased-sentiment model was used to perform the analysis. The amazon-Alexa-Reviews dataset contains the sentiment that was given by the users. So we compare the predicted sentiment to the actual sentiment that was given by the user. As an example, if the predicted score is 5, and the user score is 5, we give 1 point. If not 0.

Even though our scoring system was very stringent, the sentiment analyzer performed very well. For the Amazon Alexa data set, the strict score is 70%. After that, we checked a more lenient score. In this score, we gave leeway to the predicted score. If the predicted score is exactly matching or 1 lesser or 1 higher than the user sentiment, we gave a point. The sentiment analyzer scored 95% for this lenient evaluation.

After that, we evaluate the average sentiment of each cluster to identify the sentiment of the reviews

per each topic individually. We used Amazon-Alexa-Reviews and obtained different sentiments for each cluster. For example, on a score from 1 to 5 Cluster-4 average sentiment was 4.07 while cluster-6 average sentiment was 4.78. Despite that both scores are close enough, they indicate that users were more dissatisfied with the topic of cluster-4 than the topic associated with cluster-6. Therefore, the topics associated with cluster-4 are prioritized to be solved before the topic associated with cluster-6, etc.

7 CONCLUSION

In this paper, we highlight the importance of evaluating user feedback related to operationalized ML models. We presented a novel approach to getting insights about operationalized ML models through user feedback. Also, we created a data set related to user feedback regarding LMs. Then, we evaluated the dataset using our approach, and encountered interesting insights regarding the LMs. We identify several research directions stemming from our work. 1.) Evaluating user feedback is an interesting research direction to monitor and maintain operationalized ML models. 2.) In this work we only evaluate textual feedback. But users provide feedback in different forms. More studies need to be conducted to evaluate them. 3.) Collecting user feedback is also an interesting research area that should be further studied.

REFERENCES

- Alkan, O., Wei, D., Mattetti, M., Nair, R., Daly, E., and Saha, D. (2022). Frote: Feedback rule-driven oversampling for editing models. *Proceedings of Machine Learning and Systems*, 4:276–301.

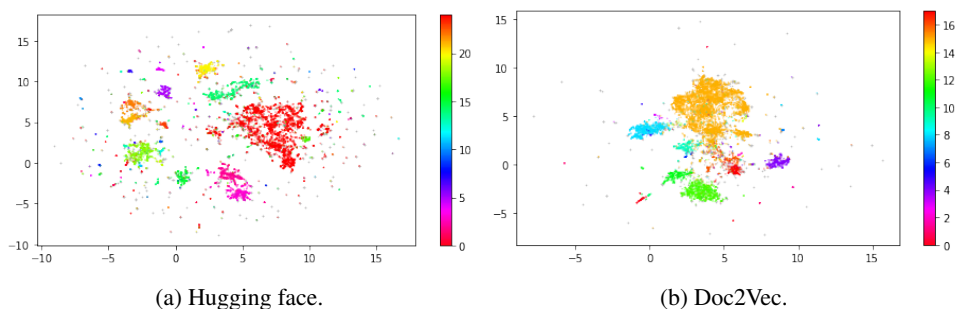


Figure 6: HBDSCAN cluster visualizations for different feature extraction methods.

- Alla, S. and Adari, S. K. (2021). What is mlops? In *Beginning MLOps with MLFlow*, pages 79–124. Springer.
- Amaro, R. M. D., Pereira, R., and da Silva, M. M. (2022). Capabilities and practices in devops: A multivocal literature review. *IEEE Transactions on Software Engineering*.
- Arbesser, C., Spechtenhauser, F., Mühlbacher, T., and Piringer, H. (2016). Visplause: Visual data quality assessment of many time series using plausibility checks. *IEEE transactions on visualization and computer graphics*, 23(1):641–650.
- Blei, D. M., Ng, A. Y., and Jordan, M. I. (2003). Latent dirichlet allocation. *the Journal of machine Learning research*, 3:993–1022.
- Caliński, T. and Harabasz, J. (1974). A dendrite method for cluster analysis. *Communications in Statistics-theory and Methods*, 3(1):1–27.
- Campello, R. J., Moulavi, D., and Sander, J. (2013). Density-based clustering based on hierarchical density estimates. In *Pacific-Asia conference on knowledge discovery and data mining*, pages 160–172. Springer.
- Davies, D. L. and Bouldin, D. W. (1979). A cluster separation measure. *IEEE transactions on pattern analysis and machine intelligence*, (2):224–227.
- Ester, M., Kriegel, H.-P., Sander, J., Xu, X., et al. (1996). A density-based algorithm for discovering clusters in large spatial databases with noise. In *kdd*, volume 96, pages 226–231.
- Hohman, F., Kahng, M., Pienta, R., and Chau, D. H. (2018). Visual analytics in deep learning: An interrogative survey for the next frontiers. *IEEE transactions on visualization and computer graphics*, 25(8):2674–2693.
- Inkpen, K., Chancellor, S., De Choudhury, M., Veale, M., and Baumer, E. P. (2019). Where is the human? bridging the gap between ai and hci. In *Extended abstracts of the 2019 chi conference on human factors in computing systems*, pages 1–9.
- Jayalath, H. and Ramaswamy, L. (2022). Enhancing performance of operationalized machine learning models by analyzing user feedback. In *2022 4th International Conference on Image, Video and Signal Processing*, pages 197–203.
- Le, Q. and Mikolov, T. (2014). Distributed representations of sentences and documents. In *International conference on machine learning*, pages 1188–1196. PMLR.
- Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L., and Stoyanov, V. (2019). Roberta: A robustly optimized BERT pre-training approach. *CoRR*, abs/1907.11692.
- Mäkinen, S., Skogström, H., Laaksonen, E., and Mikkonen, T. (2021). Who needs mlops: What data scientists seek to accomplish and how can mlops help? In *2021 IEEE/ACM 1st Workshop on AI Engineering-Software Engineering for AI (WAIN)*, pages 109–112. IEEE.
- Nigenda, D., Karnin, Z., Zafar, M. B., Ramesha, R., Tan, A., Donini, M., and Kenthapadi, K. (2021). Amazon sagemaker model monitor: A system for real-time insights into deployed machine learning models. *arXiv preprint arXiv:2111.13657*.
- Reimers, N. and Gurevych, I. (2019). Sentence-bert: Sentence embeddings using siamese bert-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics.
- Rousseeuw, P. J. (1987). Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. *Journal of computational and applied mathematics*, 20:53–65.
- Sacha, D., Sedlmair, M., Zhang, L., Lee, J. A., Peltonen, J., Weiskopf, D., North, S. C., and Keim, D. A. (2017). What you see is what you can change: Human-centered machine learning by interactive visualization. *Neurocomputing*, 268:164–175.
- Sculley, D., Holt, G., Golovin, D., Davydov, E., Phillips, T., Ebner, D., Chaudhary, V., Young, M., Crespo, J.-F., and Dennison, D. (2015). Hidden technical debt in machine learning systems. *Advances in neural information processing systems*, 28.
- Wolf, T., Chaumond, J., Debut, L., Sanh, V., Delangue, C., Moi, A., Cistac, P., Funtowicz, M., Davison, J., Shleifer, S., et al. (2020). Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45.