Adversarial Validation to Select Validation Data for Evaluating Performance in E-commerce Purchase Intent Prediction

Shotaro Ishihara
Nikkei, Inc.
Chiyoda, Tokyo, Japan
shotaro.ishihara@nex.nikkei.com

Shuhei Goda
Wantedly, Inc.
Minato, Tokyo, Japan
shu@wantedly.com

Hidehisa Arai
Recruit Co., Ltd.
Chiyoda, Tokyo, Japan
hidehisa_arai@r.recruit.co.jp

ABSTRACT

In recent years, there has been a tremendous growth in research on machine learning techniques such as deep learning. However, regarding real-world problems, practitioners still face many challenges on how to prepare data and how to validate machine learning models for specific problems. This paper describes a method for validation data selection to apply adversarial validation methodology, creating a classifier for training and test data. This approach evaluates machine learning models, taking the differences between training data and test data into account. We applied it to the purchase intent predicting task in the **Coveo Data Challenge**: 2021 SIGIR Workshop on eCommerce. This approach guided us in the challenging task to achieve the third place in the leaderboard. Our codes are available at https://github.com/upura/sigir-ecom-2021/.

KEYWORDS

adversarial validation, gradient boosting decision trees, transformer, purchase intent prediction, concept drift

ACM Reference Format:
Shotaro Ishihara, Shuhei Goda, and Hidehisa Arai. 2021. Adversarial Validation to Select Validation Data for Evaluating Performance in E-commerce Purchase Intent Prediction. In *Proceedings of ACM SIGIR Workshop on eCommerce (SIGIR eCom'21).* ACM, New York, NY, USA, 5 pages.

1 INTRODUCTION

In the last few years, there has been a lot of research on machine learning. Especially in the field of deep learning, the state of the art performance is constantly updated in a wide number of areas. The challenge that still remains is how to use machine learning to solve real-world problems. "Is the problem worthy of using machine learning?" "How should we validate our machine learning models?" "How do we detect the performance degradation of machine learning models on live data?" There is no silver bullet to these questions, and practitioners are struggling to find their own way.

One of the main problems in the real-world is the concept drift in input data[18][20][21]. For example, the distribution of features may be different between training and inference due to the change in the characteristics of the data during daily operation. Not only features but also the proportion of the objective variable may change. As a result, there can be a risk that the machine learning model in operation cannot be validated correctly.

This paper proposes a framework to select validation data to evaluate machine learning models. We utilize adversarial validation methodology which enables us to extract a subset of training data that is similar to the test data. As a case study, we tackled the task of predicting user purchase intent using e-commerce browsing information in the Coveo Data Challenge: 2021 SIGIR Workshop on eCommerce[16]. The training and test data for this challenge were sampled from separate, adjacent time periods. In addition, in order to solve this task, participants had to extract training data from the given data though the test data was prepared. Under these circumstances where there were differences in timelines and processing methods, a concept drift may occur. We attempted to tackle this problem by using adversarial validation methodology.

Furthermore, regarding the e-commerce purchase intent prediction task, there are lots of discussions on class imbalance. The classification task of the competition was difficult so that most participants could not surpass the benchmark prediction, which labels all samples as negative. Therefore, it was even more important than usual to properly evaluate the machine learning models.

The main contribution of this paper is to apply the adversarial validation methodology to the e-commerce purchase intent prediction task. It is useful to share a guideline to measure the performance of the models in the task with the concern for concept drift. Furthermore, this paper is valuable as an example of the adversarial validation application as it has rarely been reported in the literature.

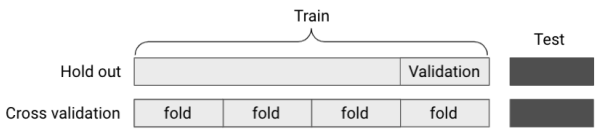
The rest of this paper is organized as follows. Section 2 explains the related work from two perspectives: 1) adversarial validation, and 2) e-commerce purchase intent prediction task. In section 3, we describe the overview of the case study challenge, the findings from exploratory data analysis, and the results of the simple baseline. Section 4 shows our methodology of applying adversarial validation to the challenge, and section 5 reports the results. The final section provides a conclusion of this paper.

2 RELATED WORK

2.1 Adversarial Validation

In the development of machine learning models, it is standard practice to divide the data into three parts: for training, validation,

Figure 1: Example of data separation



and test, as shown in Figure 1. The method of extracting part of training data for validation is called hold out and is commonly used to validate the generalization performance of machine learning models. Cross validation is also frequently used. In cross validation, the data is divided into k folds; k-1 folds are used for training and the other fold is used for validation, which is done for all combinations[1][4].

Although hold out and cross validation are effective approaches, there is a concern that if the distribution of features and target variables in training and test data are different, the models cannot be validated accurately. Adversarial validation is an approach to detect and address the difference between training and test data. In adversarial validation, a binary classifier is trained to predict whether a sample belongs to test data or not. Better classification performance than random guessing indicates that the feature distributions are different between training and test data.

Adversarial validation has been popular in recent machine learning competitions[22], and Pan et al. summarizes its methodology to concept drift problem in user targeting automation systems[12]. There are three approaches proposed; 1) automated feature selection, 2) validation data selection, and 3) inverse propensity weighting. In this paper we use the second method, which deals with differences in distribution by using the predictions of an adversarial classifier and sample training data that is highly similar to test data.

2.2 E-commerce Purchase Intent Prediction

With the growth of online retail, there has been a lot of research into recommendations and searches to improve the user experience. One of the long-standing challenges for e-commerce is the task of predicting user intent (add to cart, purchase, abandonment, etc.) from click-stream data. For example, there are some researches using traditional Markov chains[2], hand-crafted feature-based methods, and deep learning approaches[17][3]. A comprehensive survey[14] of different methods reports that deep learning achieves the best performance, beating gradient boosting decision trees (GBDT) such as XGBoost[6].

One of the issues in this task is the extreme class imbalance[16]. A simulation[14] with artificially sampled raw data shows that the model outperforms the simple baseline only up to a class imbalance of 80 % to 20 %.

3 CHALLENGE TASK AND BASELINE

3.1 Task Description

As a case study, we tackled the Coveo Data Challenge organized by a collaboration between academic researchers and Coveo, which provides a cloud-based platform of more than 1000 deployments for customer service such as e-commerce and enterprise search cases. The organizers released a new session-based data including fine-grained browsing events (detail, add, purchase, and remove), enriched by linguistic behavior (queries made by shoppers, with items clicked and items not clicked after the query) and catalog meta-data (image, text, and pricing information).

There were two tasks provided, and in the **purchase intent prediction** task, the participants were asked to create a model to predict whether the item shown in a session containing an add-to-cart (AC) event will be bought before the end of the session.

Table 1: The number of positive and negative sessions

nb	positive	negative	positive rate
0	669254	2915973	0.1866
2	640096	2517839	0.2026
4	472735	1987344	0.1921
6	406812	1665684	0.1962
8	360100	1419456	0.2023
10	318760	1230186	0.2057

Submissions must be done per session, with a value of 0 or 1. For the test data, there were several sessions with a few events after the first AC. In this paper, the number of browsing events is denoted by $nb \in \{0, 2, 4, 6, 8, 10\}$. The evaluation metric was a weighted combination of the accuracy for each *nb*. The weights were set to (1, 0.9, 0.8, 0.7, 0.6, 0.5) respectively. The maximum number of submissions is ten per day in stage 1 from April 21 to June 10, and only one per day in stage 2 from June 11 to 17.

3.2 Data Description

The Coveo Data Challenge released three main files for training. Note that the participants had to trim and label the sessions for the training models in the task.

- **browsing interactions**: a text file containing fine-grained shopping tracking for more than 4M sessions.
- **search interactions**: a text file containing more than 800k search-based interactions.
- **product meta-data**: a text file containing a mapping between the product ids (hashed) present in the entire data and their respective meta-data (when available), including a hashed representation of product categories, pricing information and vectorized representations of textual and image meta-data.

3.3 Exploratory Data Analysis

We performed an exploratory data analysis from a variety of perspectives. The most noteworthy finding was the degree of class imbalance. Table 1 lists the number of sessions with positive and negative labels for each *nb* extracted from the session data. The positive rate was around 20 % for every *nb*. The definitions of positive and negative sessions were described as follows. Note that if an action for the same product appeared after AC, the session was terminated before that. This is the reason the number of sessions are different in each *nb*.

- **positive**: Sessions contain AC and purchase action after that.
- **negative**: Sessions contain AC and do not contain purchase action after that.

3.4 Baseline Approach

When dealing with imbalanced data, a model that predicts a particular value for all samples can be used as a benchmark. In this competition, this gave us a strong result. The score of all zero submissions in stage 1 was 3.62034182210093, and none of the participants were able to outperform it significantly[1].

[1]https://sigir-ecom.github.io/data-task.html

Table 2: The reproducibility of the scores when certain samples were assigned as positive and others as negative

nb	none	0	2	4	6	8	10
leaderboard at stage 1	3.62	3.04	3.04	3.14	3.18	3.24	3.31
selected validation data	3.60	**3.00**	**3.06**	**3.12**	**3.18**	**3.24**	**3.30**
all validation data	**3.61**	2.98	**3.08**	**3.12**	3.19	3.25	**3.32**

For example, none in this table means that all samples were predicted as 0, and $nb == 0$ means that all samples except $nb == 0$ were predicted as 0.

4 METHODOLOGY

4.1 Project Overview

In this competition, it was important to detect as many positive sessions as possible. We adopted two different approaches: machine learning and rule-based. The former used GBDT and neural networks, which are expected to perform well in dealing with tabular format data[13]. Since different values of *nb* have different information about the session, six separated models were trained. We decided to judge whether each machine learning model should be used or not based on the validation score.

4.2 Validation Strategy

The overview of the validation strategy is shown in Figure 2. After obtaining the predictions for all training data through cross validation, the validation score was calculated using only the data selected by adversarial validation. In general, the average of the scores in each split of the cross validation is used as the validation score. However in this case, we believed that this was not appropriate because of the different distributions of the training and test data.

In this competition, we were able to check the positive rate in the test data was around 0.2 via all zeros submission. Therefore, similar samples were extracted until the estimated number was reached for each label. We also confirmed that there were no differences between the ratios for each *nb* in the test data. Table 2 illustrates that sampling better reproduced the leaderboard scores than using all validation data.

4.3 Machine Learning Approach

4.3.1 Gradient Boosting Decision Trees. As an implementation of GBDT, we used LightGBM[9], which has been used more frequently than XGBoost in recent machine learning competitions. About 120 features are created per session and important features are shown in Figure 3. Additional explanation of the important features can be seen in Appendix A.1. GBDT feature importance also took a role in detecting failure in feature engineering as described in Appendix A.2.

4.3.2 Neural Networks. The architecture of the neural networks contained transformer[19] and lstm[8] as shown in Figure 4. Transformer has achieved great success in natural language processing and computer vision[10], and it has been applied to tabular format data[15]. The combination of transformer and lstm is also used by the winners of the similar competition held in the AAAI 2021 Workshop on AI Education[11][15]. Implementation details and settings can be seen in Appendix B and the code.

4.4 Rule-based Approach

With a careful data analysis and insight into GBDT feature importance, we noticed that a feature `num_add_not_same_product_nb`, the number of AC for the same product in *nb* period, took positive values only in positive samples.

5 EXPERIMENTS

5.1 Training and Validation

For the adversarial validation, we trained the classifier for training and test data. LightGBM was used, and the AUC (Area Under the ROC Curve)[5] scored over 0.99 for every model. It was suggested that there was a sufficient distribution distance. There was an option of feature selection using the feature importance of adversarial validation as explained in 2.1, but this was not adopted in this time.

Table 3 shows the validation scores of GBDT model. The scores of selected data by adversarial validation told us that the performance of the models were improved as *nb* increases. This was a rational result, since the larger the *nb*, the more information the session had. On the other hand, when all validation data were used, there was no difference in the score between *nb*. We also tried random selection of validation data to match the number of positives and negatives, but the scores were not stable[2]. The baseline scores

[2]There were no trends for each *nb* and huge fluctuation due to the randomness.

Figure 2: The overview of the validation strategy

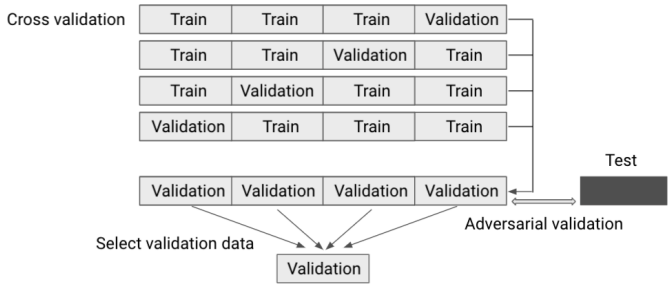


Figure 3: Feature importance of GBDT ($nb == 10$)

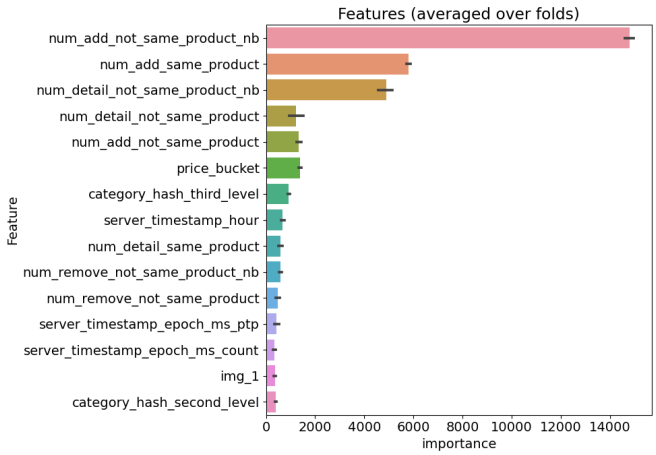


Table 3: Comparison of validation scores of GBDT

nb	all	0	2	4	6	8	10
adversarial validation	3.79	0.79	0.79	0.85	0.87	0.88	0.91
random selection	3.64	0.80	0.79	0.80	0.81	0.84	0.82
all validation data	3.56	0.79	0.77	0.78	0.79	0.80	0.80

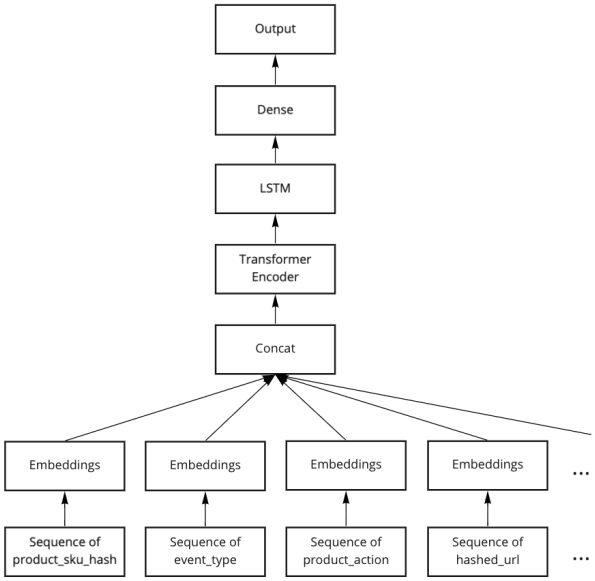
with all zeros were 0.8 for selected validation data, and around 0.8 for all validation data[3]. Evaluating the models using adversarial validation indicated that models with large *nb* have the potential to outperform the baseline.

5.2 Final Submissions and Results

In stage 1, we overcame baselines with both machine learning and rule-based approaches. For the machine learning approach, we only used the model with the largest *nb* according to the insight of the validation. Using other *nb* models did not work for us. We used the rank averaging[7] of the predictions from GBDT and neural networks, and submitted a value of positive for the sample predicted with high confidence. This gave us a score of 3.62147689588072, which was a bit greater than the baseline. Setting an extremely high threshold to predict only a few samples as positive has worked well. The same method was successful in stage 2, beating the baseline score. In the rule-based approach, predicting samples of *nb* equals ten with `num_add_not_same_product_nb` greater than three should be positive proved successful in stage 1. Unfortunately it didn't work in stage 2.

[3]To be precise, it can be calculated by (1 - positive rate) in Table 1.

Figure 4: The architecture of neural networks



6 CONCLUSION

This paper described a methodology of using adversarial validation to select validation data for the evaluation of machine learning models[4]. We tackled the e-commerce purchase intent prediction task and the insight gained by the proposed methodology enabled us to outperform the baseline.

ACKNOWLEDGMENTS

We thank the organizers of the Coveo Data Challenge for the opportunity to participate in this interesting challenge.

REFERENCES

[1] Martin Anthony and Sean B Holden. 1998. Cross-validation for binary classification by real-valued functions: theoretical analysis. In *Proceedings of the eleventh annual conference on Computational learning theory* (Madison, Wisconsin, USA) *(COLT' 98)*. Association for Computing Machinery, New York, NY, USA, 218–229.

[2] Dimitris J Bertsimas, Adam J Mersereau, and Nitin R Patel. 2003. Dynamic Classification of Online Customers. In *Proceedings of the 2003 SIAM International Conference on Data Mining (SDM)*. Society for Industrial and Applied Mathematics, 107–118.

[3] Luca Bigon, Giovanni Cassani, Ciro Greco, Lucas Lacasa, Mattia Pavoni, Andrea Polonioli, and Jacopo Tagliabue. 2019. Prediction is very hard, especially about conversion. Predicting user purchases from clickstream data in fashion e-commerce. (June 2019). arXiv:1907.00400 [cs.IR]

[4] Avrim Blum, Adam Kalai, and John Langford. 1999. Beating the hold-out: bounds for K-fold and progressive cross-validation. In *Proceedings of the twelfth annual conference on Computational learning theory* (Santa Cruz, California, USA) *(COLT '99)*. Association for Computing Machinery, New York, NY, USA, 203–208.

[5] Andrew P Bradley. 1997. The use of the area under the ROC curve in the evaluation of machine learning algorithms. *Pattern Recognit.* 30, 7 (July 1997), 1145–1159.

[6] Tianqi Chen and Carlos Guestrin. 2016. XGBoost: A Scalable Tree Boosting System. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (San Francisco, California, USA) *(KDD '16)*. Association for Computing Machinery, New York, NY, USA, 785–794.

[7] Armando Segnini Hendrik Jacob van Veen, Le Nguyen The Dat. 2015. Kaggle Ensembling Guide. https://mlwave.com/kaggle-ensembling-guide/. Accessed: 2021-6-28.

[8] Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long Short-Term Memory. *Neural Comput* 9, 8 (1997), 1735––1780.

[9] Guolin Ke, Qi Meng, Thomas Finley, Taifeng Wang, Wei Chen, Weidong Ma, Qiwei Ye, and Tie-Yan Liu. 2017. LightGBM: A Highly Efficient Gradient Boosting Decision Tree. In *Advances in Neural Information Processing Systems*, I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett (Eds.), Vol. 30. Curran Associates, Inc. https://proceedings.neurips.cc/paper/2017/file/6449f44a102fde848669bdd9eb6b76fa-Paper.pdf

[10] Tianyang Lin, Yuxin Wang, Xiangyang Liu, and Xipeng Qiu. 2021. A Survey of Transformers. (June 2021). arXiv:2106.04554 [cs.LG]

[11] Takashi Oya and Shigeo Morishima. 2021. LSTM-SAKT: LSTM-Encoded SAKT-like Transformer for Knowledge Tracing. (Jan. 2021). arXiv:2102.00845 [cs.CL]

[12] Jing Pan, Vincent Pham, Mohan Dorairaj, Huigang Chen, and Jeong-Yoon Lee. 2020. Adversarial Validation Approach to Concept Drift Problem in Automated Machine Learning Systems. In *AdKDD 2020*. San Diego, CA. https://doi.org/10.1145

[13] Amitai Armon Ravid Shwartz-Ziv. 2021. Tabular Data: Deep Learning is Not All You Need. (June 2021). arXiv:2106.03253 [cs.LG]

[14] Borja Requena, Giovanni Cassani, Jacopo Tagliabue, Ciro Greco, and Lucas Lacasa. 2020. Shopper intent prediction from clickstream e-commerce data with minimal browsing information. *Sci. Rep.* 10, 1 (Oct. 2020), 16983.

[15] Gowthami Somepalli, Micah Goldblum, Avi Schwarzschild, C Bayan Bruss, and Tom Goldstein. 2021. SAINT: Improved Neural Networks for Tabular Data via Row Attention and Contrastive Pre-Training. (June 2021). arXiv:2106.01342 [cs.LG]

[16] Jacopo Tagliabue, Ciro Greco, Jean-Francis Roy, Federico Bianchi, Giovanni Cassani, Bingqing Yu, and Patrick John Chia. 2021. SIGIR 2021 E-Commerce Workshop Data Challenge. In *SIGIR eCom 2021*.

[17] Arthur Toth, L Tan, G Fabbrizio, and Ankur Datta. 2017. Predicting Shopping Behavior with Mixture of RNNs. In *SIGIR eCom 2017*.

[18] Alexey Tsymbal. 2004. The Problem of Concept Drift: Definitions and Related Work. (May 2004).

[4]Another possible use case except this competition is shown in Appendix C.

[19] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł Ukasz Kaiser, and Illia Polosukhin. 2017. Attention is All you Need. In *Advances in Neural Information Processing Systems*, I Guyon, U V Luxburg, S Bengio, H Wallach, R Fergus, S Vishwanathan, and R Garnett (Eds.), Vol. 30. Curran Associates, Inc.
[20] Heng Wang and Zubin Abraham. 2015. Concept drift detection for streaming data. In *2015 International Joint Conference on Neural Networks (IJCNN)*. 1–9.
[21] Gerhard Widmer and Miroslav Kubat. 1996. Learning in the presence of concept drift and hidden contexts. *Mach. Learn.* 23, 1 (April 1996), 69–101.
[22] Zygmunt Zając. 2016. Adversarial validation, part one. http://fastml.com/adversarial-validation-part-one/. Accessed: 2021-6-28.

A GBDT FEATURES

A.1 Top Fifteen Important Features

The following shows the top fifteen important features for GDBT ($nb == 10$). The other *nb* models didn't make much difference.

- `num_add_not_same_product_nb`: The number of AC for the same product in *nb* period.
- `num_add_same_product`: The number of AC for the same product in all periods.
- `num_detail_not_same_product_nb`: The number of detail actions for the different products in *nb* period.
- `price_bucket`: The product price.
- `num_add_not_same_product`: The number of AC for the different products in all periods.
- `num_detail_not_same_product`: The number of detail actions for the different products in all periods.
- `category_hash_third_level`: The third level category hash.
- `server_timestamp_hour`: The hour of server timestamp at the end of the session.
- `num_remove_not_same_product_nb`: The number of remove actions for the different products in *nb* period.
- `num_detail_same_product`: The number of detail actions for the same product in all periods.
- `num_remove_not_same_product`: The number of remove actions for the different products in all periods.
- `server_timestamp_epoch_ms_ptp`: The duration of the session.
- `category_hash_second_level`: The second level category hash.
- `img_1`: The second value of the image vectors.
- `server_timestamp_epoch_ms_count`: The length of the session.

A.2 Detecting Failures in Feature Engineering

In the early stages of the project, there were some errors in the generated features. Figure 5 shows the GBDT feature importance at that time. The top feature was `num_add_not_same_product_nb`, the number of AC for the same product in *nb* period. However, this feature had no contribution to the prediction of the test data because all values were zeros. In order to replicate the way of cutting the session for the test data, the way for training data was also fixed as described in section 3.3.

B NEURAL NETWORKS ARCHITECTURE

Figure 4 shows the architecture of the neural networks. Sequences of categories and elapsed time in the input session were concatenated and sequentially passed through transformer and lstm. This architecture was optimized for the recommendation task[16], and a certain level of performance was achieved in that task as shown in Table 4. Since there was a certain degree of similarity between the two different tasks (e.g., the prediction unit was the session and the input was in the middle of the session), we considered that this model was also useful for purchase intent prediction task.

Table 4: Scores of the neural networks in the recommendation task

Model type	Transformer	Transformer & lstm
validation (Next)	0.084	0.100
validation (Subsequent)	0.195	0.245
stage 1 leaderboard (Next)	0.044	0.052
stage 1 leaderboard (Subsequent)	0.121	0.150

The recommendation task was evaluated for the prediction of the next item and for all subsequent items in that session[16].

C POSSIBLE USE CASES

A use case for applying adversarial validation beyond this challenge is shown in Figure 6. All machine learning models are created at some point in time. After a certain period of operation, the methodology can be used to evaluate the performance of the model on data from a different period than training. On the other hand, it is not a concept that can be applied in advance when immediate inferences are required.

Figure 5: Failures in Feature Engineering

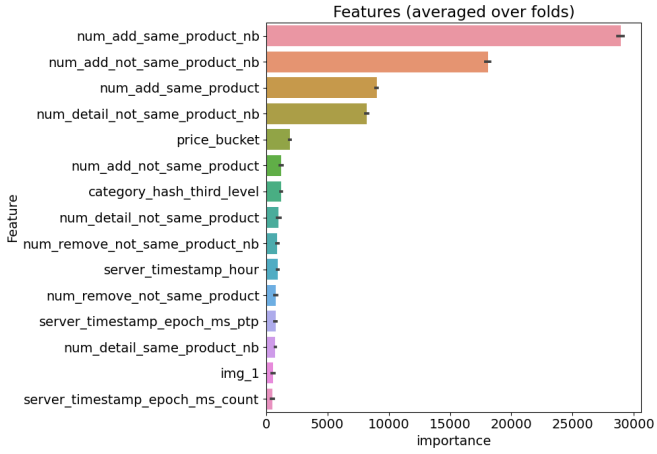


Figure 6: Timeline to create and evaluate a model