Bridging the Gap between Research and Production with CODE

Yiping Jin[1](✉), Dittaya Wanvarie[1], and Phu T. V. Le[2]

[1] Department of Mathematics & Computer Science
Chulalongkorn University, Thailand 10300
Dittaya.W@chula.ac.th
[2] Knorex Pte. Ltd., 8 Cross St, Singapore 048424
{jinyiping,le_phu}@knorex.com

Abstract. Despite the ever-increasing enthusiasm from the industry, artificial intelligence or machine learning is a much-hyped area where the results tend to be exaggerated or misunderstood. Many novel models proposed in research papers never end up being deployed to production. The goal of this paper is to highlight four important aspects which are often neglected in real-world machine learning projects, namely **C**ommunication, **O**bjectives, **D**eliverables, **E**valuations (CODE). By carefully considering these aspects, we can avoid common pitfalls and carry out a smoother technology transfer to real-world applications. We draw from a priori experiences and mistakes while building a real-world online advertising platform powered by machine learning technology, aiming to provide general guidelines for translating ML research results to successful industry projects.

Keywords: Machine Learning · Project Management · Online Advertising · Real-Time Bidding

1 Introduction

Modern machine learning approaches achieved impressive results on many challenging tasks, such as image recognition [21], machine translation [2] and speech recognition [3]. However, most machine learning research is concerned with building high-quality classification methods in isolation. While they are essential in advancing the research frontier, many proposed methods are never productionised and applied to real-world problems.

Having a high accuracy on the test set does not guarantee ML models can be applied in production. We need to consider multiple factors, such as latency, scalability, cost and the accuracy on real-world data. Even if all of these conditions are satisfied, the research team still need to convince the CTO and the engineering team to give their support before the model can be integrated into production systems. All of these complications will cause ML models built by the research team to remain on the shelf and do not have an impact on the real-world problem. A natural question then arises: what is the gap between research outputs and real-world applications and how can we bridge this gap?

The goal of this paper is to propose four important aspects as a framework for technology transfer that will bridge the above gap, namely, **C**ommunication, **O**bjectives, **D**eliverables, and **E**valuations (CODE). By giving careful considerations to these aspects, we can align the ML projects better with the business goals and accelerate the productionisation process. We demonstrate the CODE framework with the help of past industrial experiences building a large-scale machine learning-powered online advertising platform. While most cases studies used in this paper are related to online advertising, the framework we propose is not limited to a particular domain but is applicable to a wide range of machine learning projects in the production.

2 Related Works

Most attention to machine learning was paid to the modelling step with numerous new model architectures being proposed every year. However, applying machine learning models to real-life problems is far beyond that. The data mining community has been trying to standardise the data mining workflow and establish a common methodology since the 1990s. The resultant cross-industry standard process for data mining (CRISP-DM) [23] breaks the process of data mining into six major phrases, namely business understanding, data understanding, data preparation, modeling, evaluation and deployment. Modern large-scale ML frameworks share a similar workflow [18, 12]. While a common methodology is critical to the success of machine learning projects, most research papers only focus on the modeling part, leaving little discussion on the rest of the steps.

Sculley et al.'s seminal work [22] highlighted technical debts in machine learning systems that can incur massive maintenance cost and make future changes forbiddingly tricky. They commented that entanglement is in some sense "innate" to machine learning because it aims to mix the information sources to make a more accurate prediction. Raeder et al. [20] is also concerned with building large-scale machine learning systems. They proposed an end-to-end ML system that has been designed with maintenance and quality control in mind. They formulated three design principles for building massive and robust prediction systems, namely *yield fail-safe predictions*, *scale and easily extend* and *minimise human intervention*. They demonstrated the application of these principles with a large-scale online advertising platform.

Thomas [26] listed several scenarios where machine learning projects fail. Based on the article, none of the failure cases is due to the incompetency of machine learning practitioners. In contrary, almost all cases are related to miscommunication, either with the management team or with the engineering team. Some example scenarios are 1) the machine learning team produces models faster than engineering team can put them in production. 2) The model built by the machine learning team does not align with the business priority or logic.

Most recently, Ng [13] and Hermann and Del Balso [8] published blog articles sharing their experience executing AI projects in large Internet companies. Ng [13] drew from his experience leading the AI transformation in Google and

Baidu and provided five recommendations for large enterprises who wish to become an AI company. Hermann and Del Balso [8] reflected on the ML evolution at Uber and the process of developing Michelangelo, a platform which helped the company to scale ML services in production.

Our work is closest to [13] and [8] in that they also drew insights and recommendations from their experience working on ML projects.Both articles were published after we submitted our manuscript and they were developed independently from our work. We were not surprised to see that they share some common ideas with our work. All in all, we are not trying to propose a totally new methodology, but to draw from our success and failure working on ML projects and formalise a simple and easy-to-follow framework for the ML research community.

3 Background of Online Advertising

Real-time bidding (RTB) is an emerging business model of online advertising markets. In RTB, the Ad exchange will consolidate opportunities for showing ads, namely *impressions*, and send them to eligible demand-side platforms (DSPs). DSPs act as an agents for multiple advertisers to run ad campaigns across different platforms and ad formats. Each DSP will evaluate the value of the impression and submit its bid. The highest bidder among all DSPs will win the impression and display their ad. The whole process takes place within 100 milliseconds and hence is called "real-time bidding" [27].

Usually, advertisers want to optimise the number of clicks on their ads and the number of *conversions*, which is a specific user action they define a priori, such as user booking a hotel, purchasing a product or signing up for a newsletter. Standard metrics of online advertising are cost per 1,000 impressions (CPM), click-through rate (CTR), cost per click (CPC), conversion rate (CVR) and cost per acquisition (conversion) (CPA).

4 The CODE Framework

In this section, we present the CODE framework, which summarises four important yet often neglected aspects of machine learning projects. CODE stands for **C**ommunication, **O**bjectives, **D**eliverables, **E**xperiments. Besides illustrating each aspect, we will also provide related case studies based on our experiences while building a large-scale online advertising platform.

4.1 Communication

In ML research, people give strong emphasis on novel models and approaches. Almost no attention has been given to the human and organisational aspects of ML projects. In the real-world scenario, teams across different departments need to work together in synergy to deploy a large-scale ML model to production.

Each team has their own priorities. If we ignore the people aspect, we will almost certainly run into obstacles trying to push the progress.

The large team size and the management cost are the major causes of communication overhead in software engineering [5]. As a result, the total productivity increases sub-linearly with additional manpower added. In machine learning projects, there is another challenge: the technical knowledge. Machine learning is difficult for people outside the field to understand, even on an intuitive level [6]. In the industry, machine learning teams often need to work closely with the rest of the organisation, such as the management team, the engineering team or even the business team. For a ML model to be productionised, the rest of the organisation need to "buy" the idea and offer their support (e.g. the management team needs to allocate sufficient budget and time; the engineering team needs to build the supporting data pipeline). However, it's hard for them to be willing to support without sufficient understanding of the newly proposed model.

In a recent project, we worked on optimising clicks on online ads. The team proposed two approaches. The first one is a K-nearest neighbour approach [16], which finds the users who are most similar to the users who clicked on the ads and then displays the ads to them. The second approach is a recent state-of-the-art click-through rate prediction model based on the interaction of feature embeddings [19]. When we communicated the proposed methods to the management team, the first approach was immediately embraced because it was straightforward to understand. In fact, it is the core idea behind an advertising strategy called look-alike model [1]. However, the management team was sceptical about the second approach despite the impressive reported accuracy.

Having learned from this experience, we urge our researchers to be able to "sell" their models. Doing novel research and building highly-accurate models are important, yet being able to communicate the idea succinctly with the decision makers who have little ML background can be more critical to productionise the ML models. To this end, we initiated an internal technical blog and host knowledge sharing sessions where the researchers share the ideas behind their models in simple language. Other teams can also ask questions to better understand the models and their potential impact on the business. We believe that in machine learning projects, although external teams do not have to understand the technical details of machine learning models, it is critical to communicate to them the intuition behind the models and their implications and impact. A clear communication and a common ground will drastically boost people's trust in the ML models and the willingness to adopt them.

4.2 Objectives

A large proportion of machine learning problems can be modelled as optimisation problems [25]. Thus, the careful selection of the objective function (also known as cost function) plays an essential role in the success of machine learning projects. Machine learning researchers tend to be eager to jump into the modelling part without spending time trying to understand the business metric. The specific business metric may differ subtly from the commonly used objective functions

in machine learning. Failing to notice this difference will lead to misalignment of the produced model and makes it unable to achieve the business objective.

In the user click prediction example, the business requirement is to optimise the click-through rates (CTR) of ad campaigns. However, the final observed CTR is affected by many other factors, including the campaign budget, the market competitiveness and the bidding strategy, which takes the estimated CTR as an input and returns the bid price. After conducting a literature survey, the team decided to follow the simple linear bidding approach proposed by Perlich et al. [15]. The system consists of two components. The first component is a CTR estimation model, which predicts the likelihood that the user will click on the ad. The second component is a simple bid price logic which bids proportional to the predicted CTR. The intuition is that we bid at a higher price for the ad impressions which are more likely to be clicked.

Since only the CTR estimation part is an ML model, the team focused on improving the model performance as measured by the area under the ROC curve (AUC), which is the standard evaluation metric for CTR estimation in the literature. AUC is a number between 0 and 1 with a larger value indicating a more accurate prediction. An AUC of 1 means the model can perfectly predict clicks and non-clicks. An AUC of 0.5 means the prediction is equivalent to random. After deploying the new model, we observed that although AUC improved consistently between 26th Aug and 3rd Sep, the CTR did not show clear improvement as in Figure 1. We finally resolved the problem by building a bidding simulation environment and testing each component in the pipeline (CTR estimator, bidding strategy, budget allocation) individually. This helped us to improve the average CTR by 10%, which is a huge improvement in the context of online advertising and saves us millions of dollars of advertising cost per year. The improvement would not be likely if we focused on CTR estimation model only, because previous work showed that even by applying a very complex state-of-the-art CTR estimation model, the improvement in CTR is no more than 2.2%-6.3% [11].

This example demonstrates that although AUC is the standard evaluation metric for CTR estimation in the literature, optimising it alone does not guarantee a better business metric because the final business metric is affected by multiple other factors and components, not the ML model alone.

4.3 Deliverables

The performance indicator of an industrial R&D team is the models that it delivers and the overall impact on the business and products, not the amount of novel research which remains a proof-of-concept. In the IT industry, the business pressure is always high. Companies may lose their business if their competitors provide a feature which they do not provide. Therefore, the management often proposes a new problem to the machine learning team and expect them to deliver a solution within a short time frame, usually within six months. However, it takes considerable time to conduct an in-depth literature survey, to build the

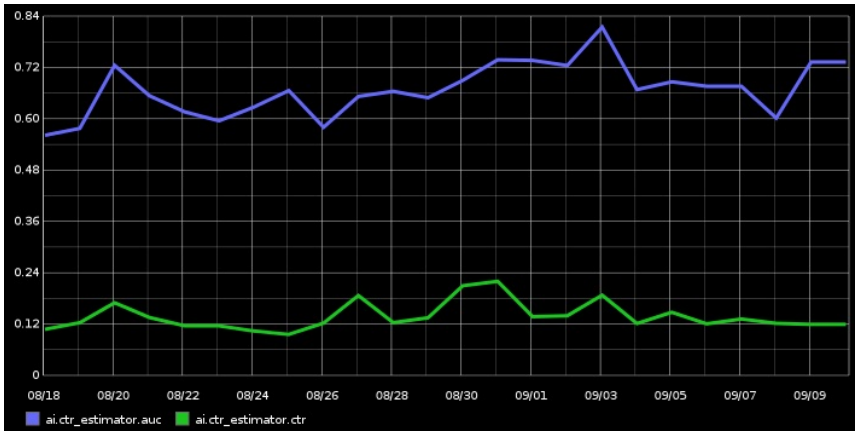

Fig. 1. Area-under-curve (AUC) and click-through rates (CTR) of an ad campaign.

data pipeline, to evaluate competitive methods and to fine-tune the model. The given time is usually not sufficient to deliver a model with a good accuracy.

Instead of delaying the delivery date, the ML team should wrap up a functioning v0.1 and deliver it first. From the business point of view, whether the ML model has a good accuracy is less critical than whether the model is in production or not. After the first deliverable, the business pressure will reduce. We will have time to refactor the model and deliver on a better v1.0. Another advantage of deploying an initial version of the ML model fast is that we can gather feedbacks from internal and external users, which will shape the directions for the subsequent effort.

As part of the global expansion effort, our team were requested to extend the text classifiers for contextual advertising [10] to ten more languages within a quarter. For each language, the number of categories is around 400, and we have millions of training documents. Even the engineering effort is tremendous to train all the models within the given time frame. Our team initially surveyed semi-supervised learning [24] or cross-lingual deep learning methods [14]. After a shallow exploration, we concluded that we would not be able to use these approaches in the first version. They require either a bi-lingual dictionary or multi-lingual word embeddings aligned to the same semantic space. Such resources may be available for high-resource languages such as German, Chinese and French, but not for low-resource languages such as Bahasa Malay and Thai. To meet the project deadline, we decided to make use of Google Translate API [3] to translate the training documents from English to the target language. The translated corpus is then used to train the text classifiers. With this, we were able to deliver the text classifiers for ten languages with an average accuracy of 2-3% lower than English, which is acceptable for the application.

[3] https://cloud.google.com/translate/

Besides the tight timeline, the management team often formulates requirements for the ML models based on the business requirements, instead of based on the state of the technology development. For example, the business may require that a chatbot to give a meaningful response 95% of the time while the state-of-the-art may only achieve 80% on a similar benchmark. While keeping pushing the research frontier is an obvious solution, it may not be feasible within the timeframe or the target may not be achievable at all. A more immediate and promising solution is to propose a new scope of the problem. While we cannot solve the general problem with 95% accuracy, it is possible that we can solve a subset of relatively easy problems accurately. In the same line, Goodfellow et al. [7] presented a case study on Street View address number transcription system. The goal was to automatically transcribe 95% of the address numbers at 98% accuracy. The rest 5% hard cases will be transcribed by human annotators.

4.4 Evaluations

While the "Experiments and Results" section is in almost every ML research paper, the result can sometimes be difficult to interpret. It is especially true for intrinsic evaluation metrics (such as mean-squared error) and artificially designed metrics (such as ROUGE score for machine translation). Even the straight-forward accuracy measure can sometimes have discrepancy between the reported figure and the accuracy the user perceives.

Although *automatic evaluations* are essential for quick experimentation, we believe that *human evaluation* cannot be neglected, especially for ML models in production systems. Automatic evaluation has certain limitations, such as it does not reflect which type of mistakes the model tends to make and it does not guarantee that the test data and the real-world data are similar enough. Exhaustive human evaluation takes a long time, but a small-scale "smoke testing" can already help us to identify most obvious problems of the ML model.

Evaluate with Simple Examples In a project detecting the language of web pages. We made use of the Optimaize library [4], which claimed to be the best open-source language detection library. The author of the library reported a 99% accuracy for 53 languages. We also evaluated the library using sampled Wikipedia pages and obtained similar results. When we delivered the project, the engineering manager randomly tried the API with a few sentences. One example input was "today is wednesday", where the library wrongly predicted as *Somali* instead of *English*. He then concluded that the accuracy of the library is bad and it cannot even classify a simple case correctly. After days' of investigation, the team found out that the problem is because the model uses a Naive Bayes classifier with character n-gram features. When the input text is short, it may not contain sufficient unique n-grams to distinguish the languages. A seemingly trivial example turned out to be the weakness of the ML model.

[4] https://github.com/optimaize/language-detector

Language	Original	After Improvement
English	0.76/0.52/0.62	0.91/0.98/0.94
French	0.67/0.58/0.62	0.97/0.95/0.96
Indonesian	0.62/0.36/0.46	0.94/0.80/0.87
Malay	0.75/0.58/0.66	0.83/0.94/0.88
Macro Avg.	0.70/0.51/0.59	**0.92**/**0.92**/**0.91**

Table 1. Short text language detection accuracy for ambiguous language pairs before and after the improvement ($P/R/F_1$).

The team then worked on improving the accuracy for short text input by extending the unigram, bigram, trigram features to 4,5,6-grams. By Adding longer n-gram features, we hope to capture n-gram features unique to each language (We do not use word dictionaries for each language because we need to know the language before we can tokenise the text into words). We selected two most ambiguous language pairs to conduct the evaluation, namely English-French and Bahasa Indonesia-Bahasa Malay [5]. The results in Table 1 shows that the simple treatment drastically improved the language detection accuracy.

From this experience, we learned that a high accuracy does not necessarily mean a good user perception. We need to analyse the actual user input and ensure that the model can deliver the promised accuracy on the real-world data.

Compare with Simple Baselines One of our data scientists recently proposed to use a multi-layer Long Short-Term Memory (LSTM) [9] model to predict the real-time bidding traffic coming to our system. LSTM model is a specific type of recurrent neural networks architecture which can model long-term dependencies effectively. The model has become the de facto approach for sequence prediction tasks such as speech recognition, and part-of-speech tagging. It is therefore natural to expect that it will yield good performance in time series prediction.

The model we used contains two layers of LSTM units, with dropout layers to prevent over-fitting. At each time step, it takes a scalar number as input, which is the traffic volume of the current minute. The last layer is a linear layer with one output unit predicts traffic volume for the following minute.

The model was trained using a single epoch consisting roughly 5,000 training sequences. The point-by-point prediction achieves a mean-squared error (MSE) of 0.0333 on the test set. The prediction of the model is also shown in Figure 2(b). At this point, we may conclude that the model is effective. However, the model may output "accurate" predictions simply because it predicts a value similar to the value of the previous time step.

To validate our assumption, we implemented a naive baseline of persistent prediction, a model which simply predicts the same value as the previous time step. The simple baseline turned out to achieve an MSE of 0.0316, which is lower

[5] Adding more languages will actually inflate the average accuracy because most other languages can be easily identified by looking at the character alone and have an accuracy close to 1 (e.g. Chinese, Korean).

than the proposed LSTM model. We can also observe clearly from Figure 2 that the predictions of the baseline are closer to the actual values. This example demonstrates that the effectiveness of a ML model cannot be validated without a meaningful benchmark and a comparison with (possibly rule-based) baselines.

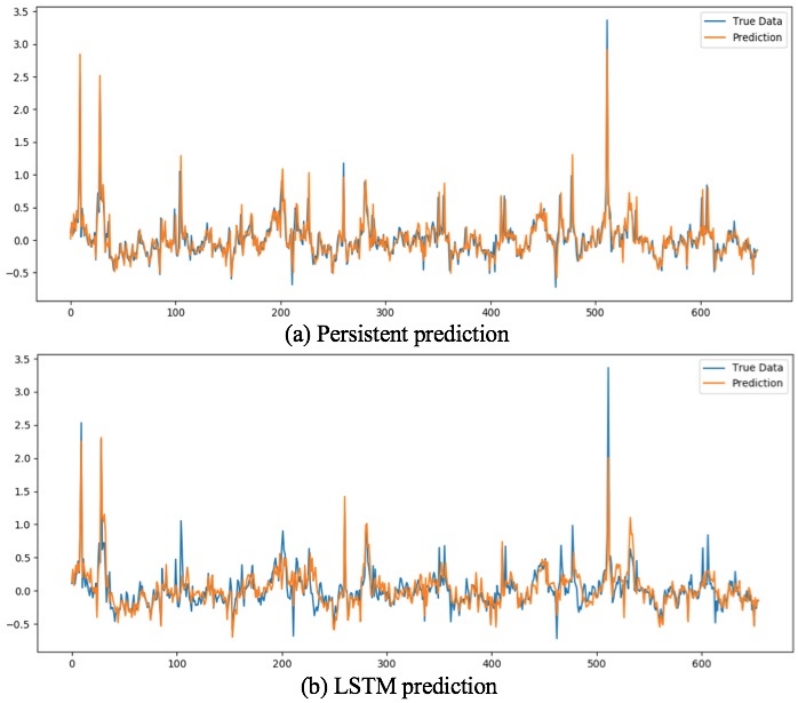

Fig. 2. Comparison of persistent prediction (loss=0.0316) and LSTM prediction (loss=0.0333).

Ensure Fair Evaluation For ML models in production, another challenge is that sometimes it is impossible to conduct a head-to-head comparison. In online advertising, the most popular way to compare two competing strategies is to perform A/B testing [6]. Advertisers will run two models with the same setting except for the different bidding strategies. After the evaluation period, they collect the performance metrics such as clicks and conversions to compare which strategy is better. However, a pitfall of this approach is that the two strategies will never display ads to the same user at the same moment. Therefore, the samples used to evaluate the two strategies are different. This problem was

[6] https://vwo.com/ab-testing/

Communication	Communicate the intuition and implication of ML models with external teams to facilitate decision-making.
Objectives	Optimise for business metrics instead of focusing on objective functions of ML models only.
Deliverables	Deliver the first version of the ML model fast without worrying too much about the accuracy. Carefully scope the problem to make it feasible and useful to the business.
Experiments	Make sure the model predicts correctly for simple examples and beats naive baselines. Try to ensure the evaluation is as fair as possible.

Table 2. Summary of the CODE Framework.

observed in [4] as well. They proposed to split the real-time bidding traffic based on geography and allocate two subpopulations for each competing strategy.

When we first deployed our ML models, our system could not serve multiple versions of models simultaneously to conduct A/B testing yet. Therefore, we had *model A* running during *period* 1 and *model B* running during *period* 2 (there is no overlap between the two periods). Nevertheless, to conclude whether *model A* and *model B* yield statistically different performance, we calculate the confidence interval for two independent samples [17]. We first compute the sample sizes (n_1 and n_2, the number of days we run each strategy), means ($\overline{x}_1$ and $\overline{x}_2$, the average click-through rates) and standard deviations (s_1 and s_2) of each sample. The pooled estimate of the common standard deviation S_p is computed as:

$$S_p = \sqrt{\frac{(n_1 - 1)s_1^2 + (n_2 - 1)s_2^2}{n_1 + n_2 - 2}} \tag{1}$$

Depending on the sample size, we use either z-table or t-table to compute the final confidence interval, which is used to evaluate whether the performance difference between *model A* and *model B* is significant.

Regardless of the evaluation metrics, we should always try to identify and eliminate potential bias and make the evaluation as fair as possible. In the case where it is not possible, we should also take note of the bias and understand its impact on the possible conclusions we can derive.

5 Conclusions and Future Work

In this work, we highlighted the gap between academic research and industry applications of machine learning technologies. We proposed the CODE framework, which summarises four essential aspects for machine learning projects to succeed, namely **C**ommunication, **O**bjectives, **D**eliverables, and **E**xperiments. We summarise the key takeaway from this paper in Table 2. We wish that the recommendations in this paper will be helpful for machine learning researchers who want to productionise their models.

In future work, we want to re-examine established frameworks in software engineering such as Agile or Scrum and adapt them for machine learning projects. We also want to establish evaluation methods to quantitatively evaluate the effectiveness of the proposed framework. This work is just a tip of the iceberg, and we believe much more effort needs to be invested to establish a general framework for machine learning projects and to help the community to translate the success in academic research into real-world applications.

Acknowledgement

The first author is supported the scholarship from "The 100^{th} Anniversary Chulalongkorn University Fund for Doctoral Scholarship" and also "The 90^{th} Anniversary Chulalongkorn University Fund (Ratchadaphiseksomphot Endowment Fund)". We would like to thank Assoc. Prof. Peraphon Sophatsathit and the anonymous reviewers for their careful reading and their insightful suggestions.

References

1. Bagherjeiran, A., Tang, R., Zhang, Z., Hatch, A., Ratnaparkhi, A., Parekh, R.: Adaptive targeting for finding look-alike users (Jul 21 2015), uS Patent 9,087,332
2. Bahdanau, D., Cho, K., Bengio, Y.: Neural machine translation by jointly learning to align and translate. arXiv preprint arXiv:1409.0473 (2014)
3. Barker, J., Watanabe, S., Vincent, E., Trmal, J.: The fifth'chime'speech separation and recognition challenge: Dataset, task and baselines. arXiv preprint arXiv:1803.10609 (2018)
4. Boyko, A., Harchaoui, Z., Nedelec, T., Perchet, V.: A protocol to reduce bias variance in head-to-head tests. Criteo Internal Report (2015)
5. Brooks, F.P.: The mythical man-month. Datamation **20**(12), 44–52 (1974)
6. Enam, S.Z.: Why is machine learning 'hard'? http://ai.stanford.edu/ zayd/why-is-machine-learning-hard.html (2016), accessed: 2018-09-10
7. Goodfellow, I., Bengio, Y., Courville, A., Bengio, Y.: Deep learning, vol. 1. MIT press Cambridge (2016)
8. Hermann, J., Del Balso, M.: Scaling machine learning at uber with michelangelo. https://eng.uber.com/scaling-michelangelo/ (2018)
9. Hochreiter, S., Schmidhuber, J.: Long short-term memory. Neural computation **9**(8), 1735–1780 (1997)
10. Jin, Y., Wanvarie, D., Le, P.: Combining lightly-supervised text classification models for accurate contextual advertising. In: Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 1: Long Papers). vol. 1, pp. 545–554 (2017)
11. Juan, Y., Lefortier, D., Chapelle, O.: Field-aware factorization machines in a real-world online advertising system. In: Proceedings of the 26th International Conference on World Wide Web Companion. pp. 680–688. International World Wide Web Conferences Steering Committee (2017)
12. Modi, A.N., Koo, C.Y., Foo, C.Y., Mewald, C., Baylor, D.M., Breck, E., Cheng, H.T., Wilkiewicz, J., Koc, L., Lew, L., Zinkevich, M.A., Wicke, M., Ispir, M., Polyzotis, N., Fiedel, N., Haykal, S.E., Whang, S., Roy, S., Ramesh, S., Jain, V., Zhang, X., Haque, Z.: Tfx: A tensorflow-based production-scale machine learning platform. In: KDD 2017 (2017)

13. Ng, A.: Ai transformation playbook: How to lead your company into the ai era. https://landing.ai/ai-transformation-playbook/ (2018)
14. Pappas, N., Popescu-Belis, A.: Multilingual hierarchical attention networks for document classification. arXiv preprint arXiv:1707.00896 (2017)
15. Perlich, C., Dalessandro, B., Hook, R., Stitelman, O., Raeder, T., Provost, F.: Bid optimizing and inventory scoring in targeted online advertising. In: Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining. pp. 804–812. ACM (2012)
16. Peterson, L.E.: K-nearest neighbor. Scholarpedia **4**(2), 1883 (2009)
17. Pfister, R., Janczyk, M.: Confidence intervals for two sample means: Calculation, interpretation, and a few simple rules. Advances in Cognitive Psychology **9**(2), 74 (2013)
18. Polyzotis, N., Roy, S., Whang, S.E., Zinkevich, M.: Data management challenges in production machine learning. In: Proceedings of the 2017 ACM International Conference on Management of Data. pp. 1723–1726. ACM (2017)
19. Qu, Y., Cai, H., Ren, K., Zhang, W., Yu, Y., Wen, Y., Wang, J.: Product-based neural networks for user response prediction. In: Data Mining (ICDM), 2016 IEEE 16th International Conference on. pp. 1149–1154. IEEE (2016)
20. Raeder, T., Stitelman, O., Dalessandro, B., Perlich, C., Provost, F.: Design principles of massive, robust prediction systems. In: Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining. pp. 1357–1365. ACM (2012)
21. Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., Berg, A.C., Fei-Fei, L.: ImageNet Large Scale Visual Recognition Challenge. International Journal of Computer Vision (IJCV) **115**(3), 211–252 (2015). https://doi.org/10.1007/s11263-015-0816-y
22. Sculley, D., Phillips, T., Ebner, D., Chaudhary, V., Young, M.: Machine learning: The high-interest credit card of technical debt (2014)
23. Shearer, C.: The crisp-dm model: the new blueprint for data mining. Journal of data warehousing **5**(4), 13–22 (2000)
24. Shi, L., Mihalcea, R., Tian, M.: Cross language text classification by model translation and semi-supervised learning. In: Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing. pp. 1057–1067. Association for Computational Linguistics (2010)
25. Sra, S., Nowozin, S., Wright, S.J.: Optimization for machine learning. Mit Press (2012)
26. Thomas, R.: What do machine learning practitioners actually do? http://www.fast.ai/2018/07/12/auto-ml-1/ (2018), accessed: 2018-09-10
27. Yuan, Y., Wang, F., Li, J., Qin, R.: A survey on real time bidding advertising. In: Service Operations and Logistics, and Informatics (SOLI), 2014 IEEE International Conference on. pp. 418–423. IEEE (2014)