Accepted manuscript, to appear in the 8th IEEE Asia-Pacific Conference on Computer Science and Data Engineering (CSDE), December 2021.

AgileML: A Machine Learning Project Development Pipeline Incorporating Active Consumer Engagement

Raj Mani Shukla*† and John Cartlidge*
*Department of Computer Science, University of Bristol, UK
†Claritum Limited, Bath, UK
{raj.shukla,john.cartlidge}@bristol.ac.uk

***Abstract*—Machine learning (ML) project deployments often have long lead times and may face delays or failures due to lack of data, poor data quality, and data drift. To address these problems, we introduce AgileML, a novel machine learning product development lifecycle where the end consumer and development team work collaboratively through an iterative process of development. We use AgileML to develop a commercial spend classification service and demonstrate that the earliest alpha deployment can offer users significant commercial value. User-testing with a professional spend analyst demonstrates that the system can lead to a five-fold increase in classification speed.**

***Index Terms*—Spend management, AI service, Model deployment, ML development pipeline, Support vector machine**

I. Introduction

It is no secret that Artificial Intelligence (AI) and Machine Learning (ML) have the potential to digitally transform the world and bring a plethora of avant-garde services. It has been shown that a business can experience a 40% improvement in productivity using AI by recognizing the value of data [1]; and it is estimated that more than 50% of organizations are either exploring or planning to adopt ML technology [1]. But, operationalization of ML services is a challenging problem and companies, keen on adopting ML, struggle in their journey. Challenges include the ability of ML to improve service quality assurance; enhance human-level performance; improve customer experience at scale; boost productivity in an industry; and increase revenues/profit. Burgeoning practices such as AIOps and MLOps have been proposed to solve these problems [2], [3]. Their essence is to automate various steps of the ML deployment lifecycle to create a self-adaptive system that requires a low level of human intervention.

However, the practice of automated model deployment is still at a germinal phase as a majority of AI service deployment is performed manually. One of the limitations of ML is the time taken to deployment; with many organizations reporting that it takes in the order of a year to fully implement a service in their enterprise [1]. The automated practices of AIOps/MLOps function only after initial model development and once the service has been deployed. Additionally, the success of ML projects depends upon the availability of well-annotated high quality datasets. Acquiring data can be difficult as it may involve multiple organizations, security and privacy issues, and high purchase costs. Also, collected data is frequently corrupt or incomplete. These limitations could further increase the lead time of service delivery. Furthermore, service performance can be degraded by the common problem of data drift, such that data used as input in the live system may no longer have the same pattern as data used for model training [4]. This is compounded by long lead times as data drift may have occurred as soon as a system becomes live. Finally, whilst ML models may have exceptional quantitative metrics such as low error and high accuracy and precision, for a user it is most important that they are tested against key performance indicators (KPIs) such as productivity and revenue gain.

To solve the foregoing challenges, we introduce AgileML, a novel pipeline that engages end-users throughout an iterative development cycle, beginning with initial model experimentation. The objective is to provide swift service deployment and a reduction in the quantity of data required up-front. This benefits developers and customers both, enabling users much earlier access to applications that offer time, labor, and cost savings. The main contributions of this research are as follows:

- We introduce the AgileML process of ML model development and deployment through active consumer engagement.
- We employ AgileML to deliver a commercial Spend Classification Service (SCS).
- We perform controlled user testing and demonstrate that the SCS can offer significant commercial value to users at a very early stage in the development process.

The rest of this paper is organized as follows. A literature review on ML deployment is presented in Section II. Section III introduces the AgileML development framework. In Section IV, we follow AgileML to develop a commercial spend classification service. Section V presents results from controlled user testing of the service. Finally, Section VI concludes this paper.

II. Related Work: ML Deployment

There has been much attention on ML-based service deployment for real-world applications. Aguilar et al. have proposed an ML lifecycle management platform, Ease.ML [5]. Ease.ML focuses on management and automation of the ML cycle rather

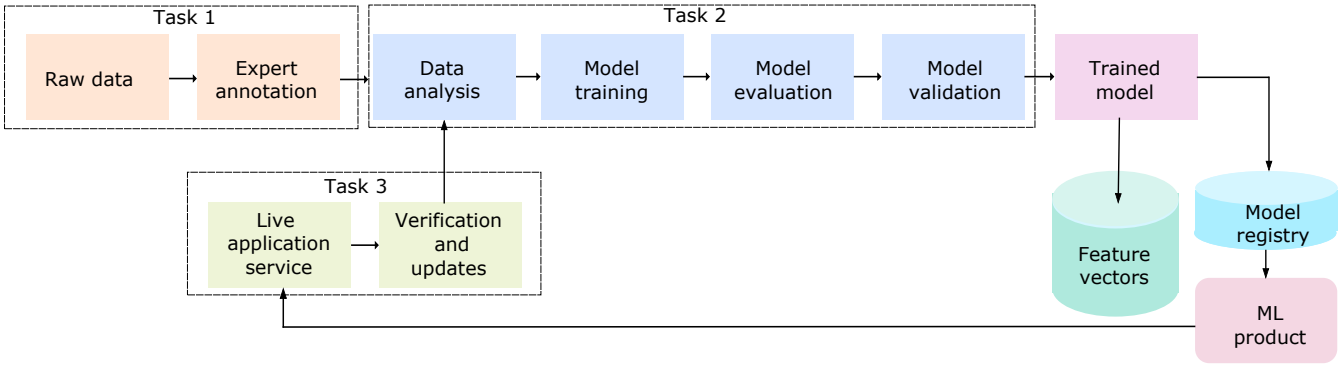


Fig. 1: AgileML continuous model upgradation and iterative human-in-the-loop testing.

than improving individual steps of the pipeline. Among the eight steps of Ease.ML, the data collection is either performed manually by the user, or data is purchased from elsewhere. In [2] the approach of *continuous integration* (CI) and *continuous delivery* (CD) has been explored. The method has three levels of automation in ML deployment to facilitate developers: (i) where the experimental phase is manual in nature; (ii) the ML training and validation steps are automated; and (iii) a full automation of training and deployment. However, MLOps primarily focuses on the automation of various processes to assist developers. It does not consider how to improve the individual steps of the data collection process itself. Zaharia et al. have proposed a framework to accelerate the ML development cycle for a company, Databricks [6]. The authors have developed a platform, MLflow, where the training code, metrics, and inference logic are brought together. In MLflow, the model is a python script that can be deployed either offline, in batch mode, or online, with real-time processing. MLflow provides flexibility and control with additional benefits of lifecycle management. Vartak et al. have proposed ModelDB, a platform to help data scientists over a period of time [7]. ModelDB consists of backend and frontend components. It helps developers track, store, and index a large number of models so that they can be easily analyzed. Schleier-Smith has implemented an architecture for agile ML for a real-time dating application [8]. The platform speeds-up the ML deployment cycle and addresses the issues of limited dataset and quick delivery of the project. The platform improves the model iteratively as data is gathered.

Often, human-in-the-loop (HITL) approaches have been used to improve the performance of ML. Yang et al. have analyzed the interaction between ML systems and HITL for text analytics applications [9]. The research first develops an ML model and then employs a domain expert to improve it. The resultant models are shown to be more interpretable. Sakata et al. have described an ML cycle that collects the data using crowd sensing for use in ML applications [10]. The study suggests that the model performs better than unsupervised learning algorithms. Li et al. have developed a HITL method for ML applications [11]. The research connects various components of the ML design project such as data collection, feature engineering, and model training together and involves HITL in the development. The given method monitors the data analysis, training, and prediction performance and if it falls below a certain threshold, the expertise of the humans are incorporated.

We propose AgileML, a development pipeline that differs to the above works in several ways. In contrast to [5]–[7], AgileML does not rely on having large quantities of data available. Indeed, AgileML assumes limited data availability and therefore takes an iterative approach to model development. In this respect, AgileML is closer to [8] and [11]. However, in these works the beneficiaries are primarily the developers; while AgileML enables consumers to also benefit much earlier in the cycle.

III. AgileML

We tackle the problem of deploying an online classification service that improves performance over time. In contrast to the traditional ML approach – involving sequential steps of data collection, data analysis, experimentation, validation, deployment, and delivery – our development process performs quick experiments on a limited dataset and then immediately deploys the model for users to utilize. Subsequently, the platform is iteratively updated through many short experiment cycles based on the accrued data and user feedback. The key difference of AgileML compared to traditional ML pipelines is that AgileML performs a large number of short experiment cycles in contrast to a single large experiment process. A schematic of this approach is presented in Figure 1. There are three main tasks. Task 1 is offline and performed only once at the beginning of the process. Subsequently, Task 2 and Task 3 are performed in short iterative cycles. The iterative tasks are important for AgileML as they achieve three concurrent objectives: (i) customer benefit; ii) incremental data acquisition; and (iii) iterative experiments involving data analysis, model selection, training, and evaluation steps. Importantly, the application is deployed immediately as an "alpha" product for customer use as soon as the first iteration of Task 2 completes; therefore enabling users to start benefiting during platform development.

TABLE I: Sample print spend descriptions, with five category labels provided by expert annotator.

Row	Specification	Project	Item	Size	Finishing	Stock
1	Supply C4 non window envelopes and overprint with PPI and return address	Direct Mail	Envelope	C4	Overprint	Paper
2	Printed pantone 5477 to face only on white 120gsm White Printspeed Laserjet Trimmed to size and boxed to suit	Stationary	Letterhead	A4	Laminated	Paper
3	Size: A4 Weight: 90gsm Material: Printspeed Offset Print: 20 Print Colours: 874 Cool Grey 11 Boxed in 500	Stationary	Letterhead	A4	Laminated	Paper

Task 1: Data collection and annotation. Task 1 consists of two sub-processes and takes place offline. The process begins with a discussion between the developers and the consumers to capture data specifications and user requirements; for example, the categories in a classification service. Often, raw data will be unlabeled or poorly labeled. Therefore, a sample of data is categorized by an expert annotator (preferably an end-user).

Task 2: Iterative model development. Task 2 involves model selection/training, model evaluation, and validation. This step performs the extensive set of short experiments in an iterative manner. Here, off-the-shelf ML algorithms and libraries are employed to develop an ML-based product, such as a classification service. The process is iterative and after a quick initial "alpha" product release, user feedback through product interaction is used to repeatedly re-train models. This quick initial release is key to the AgileML pipeline and avoids the need to commit large upfront resources pre-deployment. In successive steps, the service is updated gradually based on feedback and data obtained from user interaction (see Task 3). As the new data is acquired the ML experiments are performed in the background to improve the model. AgileML pipeline is agnostic to ML algorithms and feature section methods; in every cycle different algorithms and feature vectors can be tested and deployed. The platform is deployed as a cloud-based Software-as-a-Service (SaaS).

Task 3: HITL interaction with alpha deployment. This task involves consumers while they use the corresponding product in their day-to-day activities. The user uploads the relevant data that requires processing and the service provides the results (the prediction/classification) and their confidence levels. Subsequently, the platform allows the user to verify results sequentially. If the processed information (i.e., the predicted classification) is correct, then this is confirmed by the user. On the other hand, incorrect classifications are manually updated. The continuous review/update of the categorized data creates new training labels that are used to re-train the ML models (Task 2). In this way, as the user interacts with the data and classifications, the ML system can be considered as an expert assistant. Over time, as learning improves, users are required to commit less effort reviewing/updating categories. Through this process, the system gradually transitions from supportive assistance to fully automated. AgileML is designed to service multiple concurrent customers interacting with the system. Data from concurrent users is gathered to improve the model in each iteration. The version control is used to keep track of different ML models used by the platform at different iterations.

A. Comparison with Agile Software Development for ML

Agile development in software engineering describe beneficial practices and principles for easy collaboration between teams of software engineers, managers, and users/clients, for rapid deployment of software applications [12]. This results in better customer satisfaction, increased flexibility, and continuous improvement of the product. Agile development can also be applied to ML-based applications to ensure *continuous delivery* of the product [13]. Primarily, Agile practices for ML applications are employed to offer more efficient collaboration between hybrid teams; i.e., data scientists and software engineers. For instance, Microsoft Azure introduced a Team Data Science Process (TDSP) for managing a data science project in a systematic, collaborative, and version controlled manner [14]. This is important because each iteration step often takes longer for the data scientists than the software engineers, which means the allocation of schedules for hybrid teams must be carefully managed. However, in both [13], [14], the role of the end-user is primarily limited to providing feedback on requirements specifications each iteration. Such practices are successful when there are enough training data available at the beginning of the application development process, but they do not handle cases where data is limited at the outset.

The approaches described above help to manage collaboration between hybrid teams of data scientists and software engineers, and include methods for version control and dynamic variation in user specification. In comparison, AgileML not only includes all of these aspects, but also offers two additional benefits: (i) continuous data exchange between users and application developers; and (ii) iterative ML experimentation with users. These components are critical when developing ML applications in environments where there are little or no data available in the beginning. By including the end-user in a series of iterative experiments, AgileML enables high-quality labelled data to be captured during application development. Thus, after each iteration, AgileML not only captures updated requirements specifications, but in addition the end-user is directly involved in model improvement by confirming or rejecting model classifications (despite the user not being aware of the model details that are hidden in the back-end). This significantly reduces the need for data collection and annotation by experts. Therefore, AgileML offers a mechanism for harvesting well-annotated data, continuous data acquisition (CDA), and continuous delivery of the product.

TABLE II: Print-spend categories labeled for training.

Category	Instances
Project	5: Commercial, Direct Mail, Logistic, POS, Stationary
Item	10: Booklet, Carriage, Envelope, Fulfillment, Inkjet, Label, Leaflet, Letterhead, NCR, Poster
Size	7: A1, A2, A3, A4, A5, Double, Simplex
Finishing	4: Continuous, Laminated, Overprint, Stock supplied
Stock	4: Board, Card, Paper, PVC

IV. Spend Analytics Service

Throughout the rest of this paper, we assess the feasibility of AgileML by adopting the approach to deliver a commercial classification service for Claritum Ltd, a UK-based enterprise that offers SaaS procurement management services.

A. Spend Management

Claritum provides a set of software suites for integrated management of procurement activities, such as finding suppliers, generating quotes, submitting an order, and generating invoices. To expand its business further, Claritum is developing BRIGHT, a new software tool that provides an AI-based spend classification service to customers.

Spend management offers an organization opportunities for considerable savings. For a large organization, it is easy to lose track of purchasing. Controlled spending can lead to bulk discounts and long term relationships with preferred suppliers at preferential rates. Conversely, uncontrolled spending often results in wastage; e.g., similar items may be bought by different departments, therefore missing out on bulk discounts that would be available if purchases were consolidated. Furthermore, prices of last-minute emergency purchases of items that have run out are likely to be grossly inflated.

To overcome these challenges, it is possible to perform a spend categorization exercise, such that all purchases are classified. However, such exercises are labor intensive and expensive to perform. One leading firm charges in the region of £150,000 annually for twelve monthly classification exercises. However, this cost can be well worth it. Although the categorization process can be automated using ML [15]; the availability of the small amount of annotated data and time taken to deploy ML-based service hinders the companies adopt such an approach i.e. the categorization process is still done manually. Once procurement managers have full understanding of spend, typical savings of more than 10% are easily within reach. For a global organization, that can mean immediate annual savings of many millions of pounds [16]. BRIGHT will be offering a platform to perform spend analysis using machine learning automation. The aim is to dramatically reduce the time and cost of a spend categorization exercise, therefore enabling smaller companies to benefit from spend management opportunities.

B. Print Spend

While the BRIGHT platform is designed to be generic, for any spend category, we begin by focusing development in the area of print spend. Many large businesses spend a considerable amount on printed materials to fulfill daily activities. For instance, a supermarket chain will spend tens or hundreds of millions of pounds per year on posters, labels, letters, stickers, etc. Much of this spend data is uncategorized.[1] Typically, spend data is recorded in a company's procurement system, however, the majority of data is unstructured, free-text specifications. In addition, understanding much of this data requires expert knowledge of the industry. In Table I, we present three sample print spend specifications.

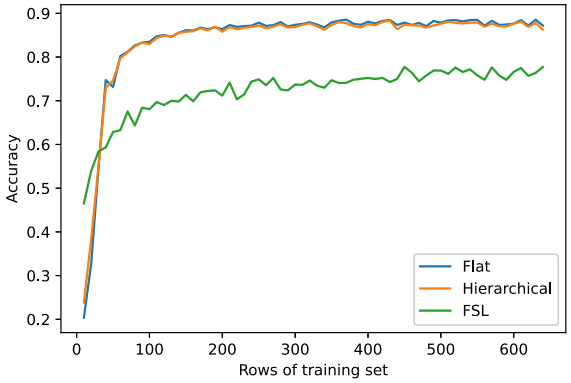


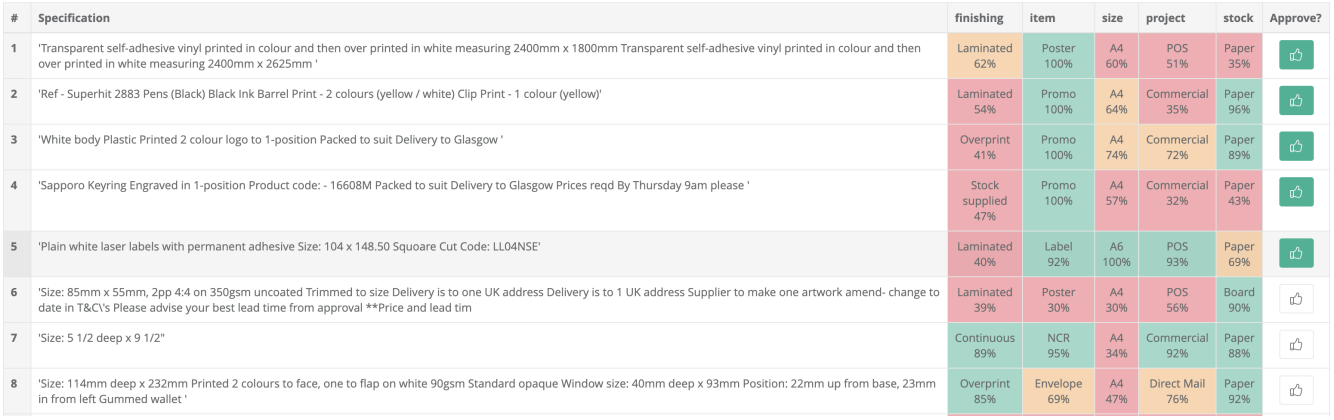
Fig. 2: Model accuracy.

A professional spend-analyst with expertise in print was employed to perform an initial labeling exercise for a sample (800 rows) of print spend data. The spend-analyst selected five category levels that would be most useful for a spend management exercise: (i) project, i.e., the general spend area, (ii) item, (iii) size, (iv) finishing, and (v) stock, i.e., the material of manufacture. Each category represents an area of interest for a procurement manager. Table I presents the five categories labeled for each example, and Table II presents a summary of all category instances that were labeled in the training set. This labeling process took the spend-analyst approximately two days' work, which equates to roughly 50 rows classified per hour. This gives an indication of how long and laborious a full manual classification exercise is, with spend data for a large company likely to contain millions of rows.

C. Model development

The labeled data was employed to develop a classification service. After testing different algorithms, an ensemble of Support Vector Machines (SVMs) with linear kernel and One-vs-Rest (OvR) heuristic was chosen for text classification. Raw text specifications are initially processed to remove redundant symbols, stop-words, and punctuation. The text is then lemmatized and converted to tokens. As the expert-labeled dataset consists of five different categories, a flat classification approach was used such that each category level is considered independently of the others. Therefore, five different SVM

[1]One major supermarket chain reported that up to half of all print spend invoices may be uncategorized (personal communication).



#	Specification	finishing	item	size	project	stock	Approve?
1	'Transparent self-adhesive vinyl printed in colour and then over printed in white measuring 2400mm x 1800mm Transparent self-adhesive vinyl printed in colour and then over printed in white measuring 2400mm x 2625mm '	Laminated 62%	Poster 100%	A4 60%	POS 51%	Paper 35%	
2	'Ref - Superhit 2883 Pens (Black) Black Ink Barrel Print - 2 colours (yellow / white) Clip Print - 1 colour (yellow)'	Laminated 54%	Promo 100%	A4 64%	Commercial 35%	Paper 96%	
3	'White body Plastic Printed 2 colour logo to 1-position Packed to suit Delivery to Glasgow '	Overprint 41%	Promo 100%	A4 74%	Commercial 72%	Paper 89%	
4	'Sapporo Keyring Engraved in 1-position Product code: - 16608M Packed to suit Delivery to Glasgow Prices reqd By Thursday 9am please '	Stock supplied 47%	Promo 100%	A4 57%	Commercial 32%	Paper 43%	
5	'Plain white laser labels with permanent adhesive Size: 104 x 148.50 Squoare Cut Code: LL04NSE'	Laminated 40%	Label 92%	A6 100%	POS 93%	Paper 69%	
6	'Size: 85mm x 55mm, 2pp 4:4 on 350gsm uncoated Trimmed to size Delivery is to one UK address Delivery is to 1 UK address Supplier to make one artwork amend- change to date in T&C\'s Please advise your best lead time from approval **Price and lead tim	Laminated 39%	Poster 30%	A4 30%	POS 56%	Board 90%	
7	'Size: 5 1/2 deep x 9 1/2"	Continuous 89%	NCR 95%	A4 34%	Commercial 92%	Paper 88%	
8	'Size: 114mm deep x 232mm Printed 2 colours to face, one to flap on white 90gsm Standard opaque Window size: 40mm deep x 93mm Position: 22mm up from base, 23mm in from left Gummed wallet '	Overprint 85%	Envelope 69%	A4 47%	Direct Mail 76%	Paper 92%	

Fig. 3: User interface of *BRIGHT Spend Analytics.AI* SaaS. Specifications are automatically classified and presented with "traffic light" coloring to indicate confidence score. Incorrect classifications can be manually corrected. Edited cells are then given a confidence value 100% (e.g., row 1 item "poster"). User approves a row by selecting "thumbs up".

models were developed; one for each level. We considered the impact of different combinations of input features, including *unigrams*, *bigrams*, and *Noun or Verb Phrases (NVP)* [17]. It was found that *unigrams* combined with top ranked *bigrams* or *NVPs* provide the highest accuracy with the least number of features. We split the training and test data in a ratio of 80:20.

We compared the Flat Classification approach with Hierarchical Classification [18] and Few Shot Learning (FSL) [19]. The accuracy of three models with the different number of training rows is shown in Figure 2. It can be observed that the Flat and Hierarchical approaches have very similar accuracy levels as both predict the same class the majority of the time. We hypothesize that this is largely due to the small training set: when more data is available, we expect that a hierarchical ensemble of models will perform better than a flat classifier. As expected, FSL performs best only when the training set is extremely small. These results demonstrate the benefits of training multiple models in the background, and then switching to the most accurate model as the training set grows.

V. User testing

Following model training, we immediately deployed a first alpha service for user testing. The application was deployed in Google App Engine as a Software as a Service (SaaS) using python-based web framework, Flask. The first iteration of the service uses pre-trained models, with classifications following the five levels previously defined by the expert.

The classification service allows a user to upload a spreadsheet of spend data. The uploaded spreadsheet is then processed and categorized to each of the five category levels. These are presented to the user on screen, along with the confidence score of each categorization. Figure 3 presents a screenshot of the user interface. We see that categories are colored using a simple traffic light scheme to help focus the user's attention. Classes with high confidence (over 80%) are colored green; classes with medium confidence (over 60%) are colored amber; and other cells are colored red. The user is able to update any cell that has an incorrect categorization; and where a row is fully correct, this can be confirmed by selecting "thumbs up".

TABLE III: Accuracy (percentage) of user testing files.

Feature	File A	File B
Project	83	100
Item	76	100
Size	84	76
Finishing	99	97
Stock	94	100
Row	51	74

A. Experiment

We performed a user experiment with the same expert that classified the initial 800 rows of training data. The experiment was conducted in two parts. In part A, the user was given a file of 100 specifications in randomized order, exactly as the original set. In part B, the user was given a file of 100 specifications that the system is able to classify with high confidence. The user was then asked to spend a maximum of 30 minutes on each file, updating incorrect cells and approving correct rows. In this way, we are able to make a direct performance comparison between human classification speed with and without the system support; and also we are able to measure the impact of confidence on classification speed.

B. Results

The user was able to classify 95 rows of file A in 30 minutes (i.e., a speed of 190 rows/hour); and all 100 rows of file B were classified in only 17 minutes (i.e., a speed of approximately 350 rows/hour). Given that the original training data was classified at only 50 rows per hour, this demonstrates a significant increase in speed of between 4 to 7 times.

However, this headline figure does not tell the whole story. The speed increase requires initial training that took 2 days

TABLE IV: Accuracy (and standard deviation) of each dataset using k=10 folds.

Set	Project	Item	Size	Finishing	Stock
D	0.88 (0.04)	0.88 (0.05)	0.84 (0.08)	0.99 (0.03)	0.86 (0.06)
D + A	0.88 (0.03)	0.90 (0.03)	0.82 (0.04)	0.92 (0.07)	0.87 (0.05)
D + B	0.90 (0.03)	0.91 (0.04)	0.83 (0.05)	0.99 (0.02)	0.89 (0.05)
D + A + B	0.90 (0.02)	0.90 (0.04)	0.82 (0.06)	0.95 (0.03)	0.89 (0.07)

of labor. Yet, it is possible for this training to be parallelized among a team of classifiers. For example, having a team of eight classifiers would reduce training to 2 hours of work. Then, immediately we see a scale-up of between 4-7 times per classifier. For a company charging £150,000 annually for a manual classification service, this equates to significant labor savings.

The accuracy of the classification for each file is shown in Table III. As expected, file B, containing classifications with high confidence values, had much higher accuracy than file A. In particular, file B classifications for *project*, *item*, and *stock* had 100% accuracy. Also, 74% of all rows in file B were correctly classified on all five categories, compared with 51% of rows in file A. These results clearly indicate the value of the confidence score in predicting class accuracy, and explain why the expert was able to complete file B checking in half the time taken to complete file A. Indeed, in both files, *all* green cells, i.e., those with confidence greater than 80%, were correct. In file A, a total of 67 edits were made, consisting of 55 red cells and 12 amber cells. This suggests that a large proportion of spend data can be auto classified with no human intervention; leaving the human to focus attention on rows that are more difficult to classify. Indeed, with overall accuracy greater than 80% across all categories, and despite having only minimal training, the system is already capable of producing a meaningful spend categorization that will allow, at the very least, reasonable estimates of spend, without requiring large teams of manual classifiers and without outlaying hundreds of thousands of pounds in labor costs.

User feedback confirmed that the first iteration provides useful assistance, resulting in increased classification speed and demonstrating immediate commercial value. Minor improvements to the interface were suggested, including: configurable threshold (a slider) for coloring cells by confidence; hiding rows above a confidence; and column sorting on confidence. These will be incorporated into the next iteration.

C. Retraining

Following the user experiment, the annotated files A and B provide an additional 200 rows of data to retrain the models. We were able to obtain these in under one hour of work for the classifier, whereas it would have taken an additional 4 hours of work without the support of the system.

We retrained the SVM using this newly labeled data and performed k-fold validation. Table IV shows the accuracy (and standard deviation) of different combinations of the original dataset (labeled D) of 800 rows, file A (labeled A), and file B (labeled B). The comparison shows that after retraining on the larger dataset (D+A+B), accuracy improved for project, item, and stock; but decreased for size and finishing. This can be explained because three new class instances of *size* (A6, C4, and C5) appeared in files A and B. Therefore, the classification task is more difficult on the larger dataset.

D. Iterative update

Having demonstrated that the initial version has some commercial value for the end-user, the next iteration will incorporate processes that we took offline for initial development. These include: (i) real-time learning, so model re-training is performed online after user edits and confirmations; and (ii) enabling users to define their own classification labels, so that the system will work with any dataset. Together, these improvements will enable users to begin training their own models on their own data. At that point, despite being early in the development pipeline, the deployed service will be ready for beta release. This is a clear advantage of AgileML.

E. Applications: Beyond Spend

Once spend is well categorized, it is possible to query the data to understand the supply chain and monitor externalities such as carbon emissions. To meet the objectives of the 2016 Paris Agreement to limit global temperature rise to 1.5 degrees Celcius, carbon emissions must be reduced by 50% within a decade [20]. It is expected that companies provide full disclosure of their performance on environmental issues. If spend data is combined with carbon emissions of suppliers in the procurement chain, it becomes possible to perform carbon "hotspotting" (e.g., discovering where in the supply chain plastics are most used), and alternative suppliers can then be selected on their carbon performance, rather than on price alone.

VI. Conclusions

We have presented AgileML, a pipeline to facilitate rapid deployment of machine learning applications. In contrast to the longer sequential development cycles typically seen, AgileML focuses on quickly deploying an initial alpha prototype for user testing that can immediately provide benefit to the user; while further deployments are then developed through iterative user engagement and experiments. We used AgileML to deploy a commercial spend classification service that automatically categorizes free-text specifications of company spend. Initial testing of the first iteration of deployment demonstrated that the service can increase the classification speed of a professional spend analyst by a factor of between 4 and 7. This significant scale-up offers immediate commercial value

to users despite the application being in an early stage of development.

Following AgileML methodology, future deployment iterations are under rapid development. Next iterations will include real-time model learning and the ability for users to define their own classification schemes through the interface. At that point, the data analysis and ML experiments will be done automatically.

ACKNOWLEDGMENT

This work was supported by Innovate UK Knowledge Transfer Partnership between University of Bristol and Claritum Limited (KTP 11952).

REFERENCES

[1] S. Bhatt, "7 machine learning challenges businesses face while implementing," 2020. [Online]. Available: https://www.business2community.com/business-innovation/7-machine-learning-challenges-businesses-face-while-implementing-02341340

[2] GoogleCloud, "MLOps: Continuous delivery and automation pipelines in machine learning," 2021. [Online]. Available: https://cloud.google.com/architecture/mlops-continuous-delivery-and-automation-pipelines-in-machine-learning

[3] Y. Dang, Q. Lin, and P. Huang, "AIOps: Real-world challenges and research innovations," in *Proc. of International Conference on Software Engineering: Companion Proceedings*, 2019, pp. 4–5.

[4] J. Lu, A. Liu, F. Dong, F. Gu, J. Gama, and G. Zhang, "Learning under concept drift: A review," *IEEE Transactions on Knowledge and Data Engineering*, vol. 31, no. 12, pp. 2346–2363, 2018.

[5] L. Aguilar, D. Dao, S. Gan, N. M. Gurel, N. Hollenstein, J. Jiang, B. Karlas, T. Lemmin, T. Li, Y. Li, S. Rao, J. Rausch, C. Renggli, L. Rimanic, M. Weber, S. Zhang, Z. Zhao, K. Schawinski, W. Wu, and C. Zhang, "Ease.ML: A lifecycle management system for MLDev and MLOps," in *Proc. of Innovative Data Systems Research*, 2021.

[6] M. Zaharia, A. Chen, A. Davidson, A. Ghodsi, S. A. Hong, A. Konwinski, S. Murching, T. Nykodym, P. Ogilvie, M. Parkhe *et al.*, "Accelerating the machine learning lifecycle with MLflow." *IEEE Data Eng. Bull.*, vol. 41, no. 4, pp. 39–45, 2018.

[7] M. Vartak, H. Subramanyam, W.-E. Lee, S. Viswanathan, S. Husnoo, S. Madden, and M. Zaharia, "ModelDB: A system for machine learning model management," in *Workshop on Human-In-the-Loop Data Analytics*, 2016.

[8] J. Schleier-Smith, "An architecture for agile machine learning in real-time applications," in *Proc. of International Conference on Knowledge Discovery and Data Mining*, 2015, p. 2059–2068.

[9] Y. Yang, E. Kandogan, Y. Li, P. Sen, and W. S. Lasecki, "A study on interaction in human-in-the-loop machine learning for text analytics," in *IUI Workshops*, 2019.

[10] Y. Sakata, Y. Baba, and H. Kashima, "Crownn: Human-in-the-loop network with crowd-generated inputs," in *Proc. of International Conference on Acoustics, Speech and Signal Processing*, 2019, pp. 7555–7559.

[11] L. Yang, M. Li, J. Ren, C. Zuo, J. Ma, and W. Kong, "A human-in-the-loop method for developing machine learning applications," in *Proc. of Int. Conference on Systems and Informatics*, 2019, pp. 492–498.

[12] P. Abrahamsson, O. Salo, J. Ronkainen, and J. Warsta, "Agile software development methods: Review and analysis," 2017.

[13] R. Hanslo and M. Tanner, "Machine learning models to predict agile methodology adoption," in *Conference on Computer Science and Information Systems*. IEEE, 2020, pp. 697–704.

[14] MarkTab, K. Sharkey, and E. Price, "Agile development of data science projects," 2020. [Online]. Available: https://docs.microsoft.com/en-us/azure/architecture/data-science-process/agile-development

[15] S. Mukherjee, D. Fradkin, and M. Roth, "Classifying spend descriptions with off-the-shelf learning components," in *IEEE International Conference on Tools with Artificial Intelligence*, vol. 1, 2008, pp. 53–60.

[16] A. Bhattacharya, "How to mint millions from your unclassified spend data?" 2018. [Online]. Available: https://www.zycus.com/blog/procurement-technology/how-to-mint-millions-from-your-unclassified-spend-data.html

[17] H. Jeong, D. Shin, and J. Choi, "FEROM: Feature extraction and refinement for opinion mining," *Etri Journal*, vol. 33, no. 5, pp. 720–730, 2011.

[18] C. N. Silla and A. A. Freitas, "A survey of hierarchical classification across different application domains," *Data Mining and Knowledge Discovery*, vol. 22, no. 1, pp. 31–72, 2011. [Online]. Available: https://doi.org/10.1007/s10618-010-0175-9

[19] Y. Wang, Q. Yao, J. T. Kwok, and L. M. Ni, "Generalizing from a few examples: A survey on few-shot learning," *ACM Comput. Surv.*, vol. 53, no. 3, Jun. 2020. [Online]. Available: https://doi.org/10.1145/3386252

[20] UNFCCC, "Paris Agreement to the United Nations Framework Convention on Climate Change," Dec. 12 2015. [Online]. Available: https://unfccc.int/sites/default/files/english_paris_agreement.pdf