PharmaSUG 2020 - Paper AI-224

How to let Machine Learn Clinical Data Review as it can Support Reshaping the Future of Clinical Data Cleaning Process

Mirai Kikawa and Yuichi Nakajima, Novartis Pharma K.K.

ABSTRACT

Technology utilized in pharmaceutical industry has been evolving. There are a lot of innovative new technology such as Artificial Intelligence, Machine Learning, Digitization, Block chain, Big Data, Open Source Software, etc., which can build a new era of clinical drug development.

Manual data review is one of the required processes to ensure clinical data cleanness and readiness for analysis that are essential for subject safety and reliability of the submission documents. Manual data review process involves several roles of people such as Data Manager, Clinical and/or Medical Reviewers, Safety Reviewer, etc. Since it requires complicated logical thinking and clinical and medical knowledge and expertise, it has to be “manual”. That has been the common understanding, and thus the conventional approach. However, does it have to keep being true and do we have to keep manual process that requires time and efforts?

In recent years, clinical data collected during clinical trials have been structured and standardized by industrial efforts such as introduction of CDISC and standard operational process by each pharmaceutical company. The structured and standardized data across clinical trials increases compatibility of data utilization, which enables more efficient approach for data review. It can be ingested into machine to let machine learn data review using Python that has many capabilities which can easily automate and facilitate tasks, which is one of the ideas to break the traditional approach and reshape the future of clinical data cleaning process.

This paper proposes a potential way to let machine learn clinical data review using Python.

INTRODUCTION

Although innovative technologies have been evolving and increasingly sophisticated, there are many pairs of human eyes still have been involved in clinical data reviewing and cleaning processes. The roles such as Data Manager, experts with clinical scientific and medical background, safety and pharmacovigilance specialists, etc. extract clinical data listings from clinical database and manually review data, investing huge amount of time and efforts, then issue queries in EDC or take any other necessary actions. The main reasons why it has to be manual for years are:

- It requires complicated logical thinking and clinical and medical knowledge and expertise. Especially, as clinical trial protocols are getting much more complex, and therefore complexity of manual data review is increasing accordingly. There are cases that use large amount of data i.e., data from multiple domains and multiple variables, and which variables to be used may differ depending on the clinical trial protocol design and/or the status of the particular subject.

- After initiation of the clinical trial, by looking at the actual subject data, manual review criteria may need to be adjusted e.g., by considering safety data trends, the criticality of a certain lab test result and what/how to check it against Adverse Event data may be updated.

- Clinical trial data has to be very well structured and standardized to apply any checks other than manual review, but it takes time and effort. It seems to be easier for the data reviewers to just keep performing manual data review as current process than preparing well structured and standardized data and apply some other data review process.

- If the clinical trial team wants to utilize any other way than manual/programmed checks to reduce human effort such as automated data review using machine with acceptable accuracy, it is required to feed machine a certain amount of standardized data along with standardized data review criteria.

However, it is about time to say that above approach is already the “traditional way” as manual effort for data cleaning process can be minimized using machine learning as support tool for data reviewers.

The purpose of this paper is to suggest how to overcome above reasons for still taking the traditional approach for data cleaning and how to implement the new approach of automated data cleaning utilizing machine learning by Python with the careful considerations.

STANDARDIZATION - THE PREREQUISITES

One of the key success factors to move on to the new clinical data cleaning approach is the standardization – standardization of data collected in clinical trials and standardization of data review criteria.

CLINICAL DATA STANDARDIZATION

Standardized clinical data is the mandatory requirement for New Drug Applications by health authorities. For which, pharmaceutical companies have been making huge efforts to submit the data compliant to this requirement. Let us benefit from it and turn it into earned privilege, as this is the first prerequisite to let machine learn how to perform data review.

In Novartis, Clinical Data Standard team strictly controls the data collected via EDC and from external data sources such as central labs and relevant metadata that follows CDISC standards, and clinical trial teams strictly follows standard operational procedure to ensure the quality of clinical data and safety of subjects. Through those efforts, the standardized data across clinical trials are obtained.

This standardized data across clinical trials allows us to obtain the great amount of data that can be ingested into machine and can be facilitated for comparison across trials, projects, and submissions. It helps letting machine learn data review with the reliable accuracy of the results. In addition, since the first key step in supervised learning is data preprocessing, which is described in latter section, standardized data enables smooth preprocessing of data.

DATA REVIEW CRITERIA STANDARDIZATION

To let machine learn clinical data review, it is required to have clear definition of data review. In Novartis, currently there are 71 standardized manual data review criteria as of March 2020, of which, 53 criteria are performed by our Data Managers, and the other criteria are performed by other reviewers. Those standardized criteria mainly covers critical domains for safety and efficacy, such as Adverse Events Disposition, Dosing, Medical History, Vital Signs, Labs, Visits, Hospitalizations, etc., and our manual data review criteria metadata are:

- Review ID: a unique ID per criterion
- Type of Standard (Global / Project / Study)
- Primary Domain: the data domain that is being reviewed
- Cross-Domains: data domain that is checked against the primary domain. One or more than one domains can be included depending on the data review task.
- Review Task: detailed description of manual data review to be performed, what should be confirmed by this check and how
- Suggested Query Text and/or Action: query text to be used in case findings need clarification or correction by clinical trial sites and/or action to be taken as part of the data review task
- Role: who should be performing the data review task
- Suggested Frequency: how often this check should be performed at minimum (Ongoing, Monthly, Quarterly, etc.). This may vary based on the length of the trial and specific milestones.
- Review Tool: the tool to be used for each data insight e.g. dump data listings generated from EDC, data listings from other reporting tools, data visualization tools, etc. The report/output name to be specified as well

Using the above metadata, each manual review criterion is defined clearly and sufficiently, and the list of standardized criteria have been maintained by our subject matter expert team.

Although the criteria are standardized, we keep going on with fine-tuning by adapting health authority requirements, etc. In the current manual data review process, sometimes it takes time and efforts to apply the changes even minor ones; however, by taking advantage of machine learning which is introduced in the next section, it could be applied more efficiently.

DATA REVIEW AUTOMATION USING MACHINE LEARNING

As described in previous section, standardization is one of the important parts of this data review process automation. In addition to process automation, machine learning is another good situation to use standardized data. In general, the biggest challenge in using machine learning is to collect appropriate amount of structured data. That means data collection and data preprocessing determine the quality of output from machine learning. As clinical trials are conducted under the control of GCP and standardization of the clinical data is mandatory, both problems are addressed naturally.

This section introduces the basic of machine learning techniques of supervised machine learning and shows concept of data review process automation implemented by Python. Then it describes considerations and challenges for further implementations.

MACHINE LEARNING BASIC

Machine learning is to repeatedly learn from data and find patterns from it. It is an ensemble of statistical techniques that give computers the ability to learn using large amount of data, without being explicitly programmed. Generally speaking, machine learning is categorized in two major types of system, supervised and unsupervised learning (Figure 1). In addition, supervised learning has two major algorithms: classification and regression.

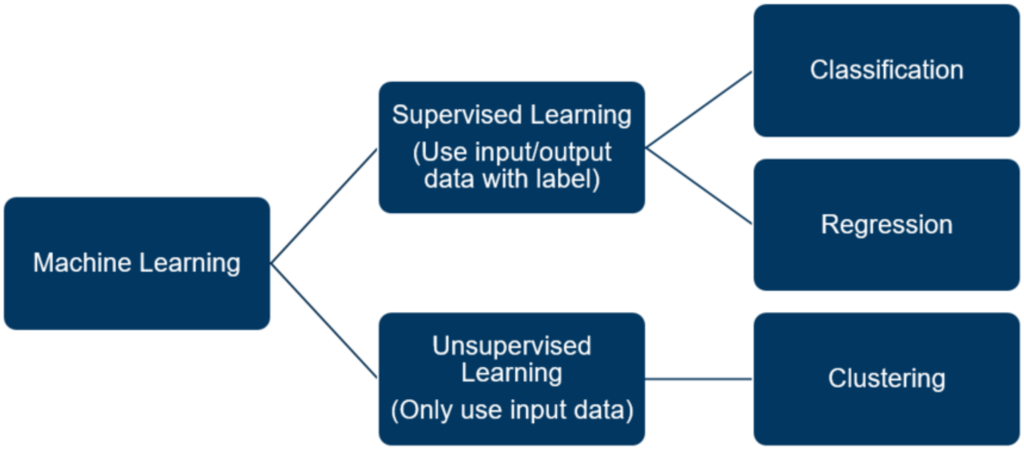

Machine Learning
Supervised Learning (Use input/output data with label)
Unsupervised Learning (Only use input data)
Classification
Regression
Clustering

Figure 1. Types of machine learning

General steps to implement machine learning

Here is general steps to implement machine learning.

- Step1: Data collection
- Step2: Data preprocessing
- Step3: Feature selection from data
- Step4: Model (Classifier) selection

- Step5: Model training and tuning / Model validation
- Step6: Model evaluation

A feature is an individual measureable property of data and is also considered as characteristic being observed in data. For example, occupation, age, place to live, family structure might be features of income. Feature selection is an important step for effective algorithms in machine learning. When it comes to data review, feature selection should be made depending on each review criterion, and this is a key step to build better model. To define feature value from data, review criteria should be discussed between data manager and data engineers. In other words, in the step for feature selection, Data Managers (and other data reviewers) specify what and how they review data so that data engineers can tell machine what to do.

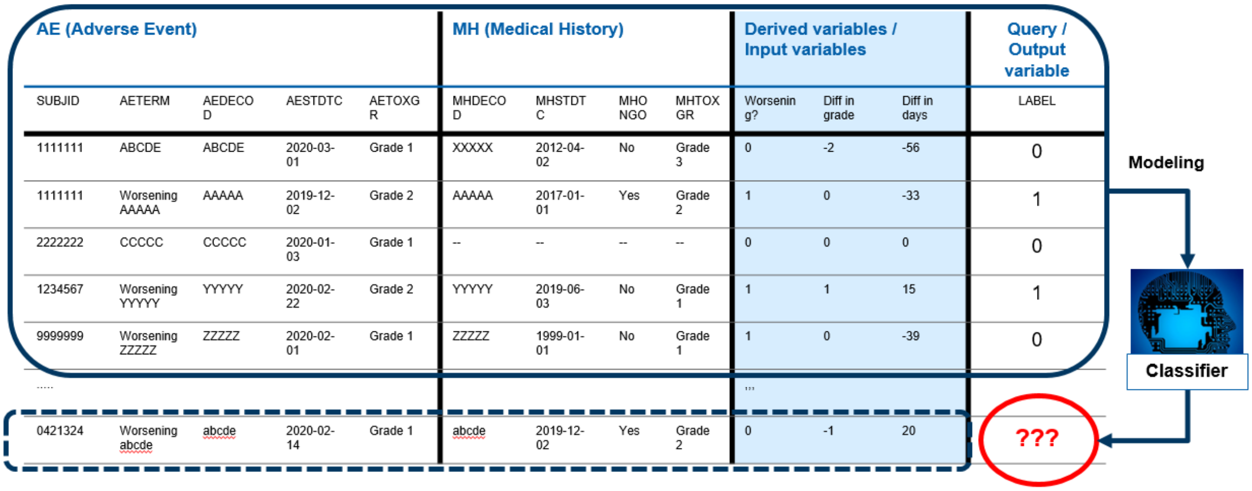
Classification

Classification is one of major algorithms in supervised learning, which categorizes some unknown items into a discrete set of classes. An output variable should be categorical variables such as Negative or Positive, Drug A or B, and it can be multiple categories like Low/Normal/High. A simple example of classification is the spam filtering of emails. Every time spam filter receives email, it makes a prediction whether it is “Spam” or “Not Spam”. To make prediction, spam filter checks the sender, IP address, and title of the email as input variables and delivers label “Spam” or “Not Spam” as an output.

There are general classification algorithms in machine learning. For example, Decision Trees, Random Forests, Naïve Bayes, K-Nearest Neighbor, Logistic Regression, Neural Networks and Support Vector Machines (SVM). Those techniques are traditional but still being widely used in machine learning systems in recent days. This paper mainly focuses on Decision Trees and Random Forests.

LABELED DATASET OF AUTOMATION PROCESS OVERVIEW

Now consider how machine learning contributes data review in clinical trials. Solid oval square in Figure 2 shows labeled dataset for automation process. Assume Query as categorical values (1: Query, 0: No query) for Output variable, and Derived variables (columns in light blue) as Input variables derived from the source variables in each domain. Labeled dataset splits into training and test set to build model. It is important that labeled dataset includes output variable, which is a variable to flag provided by data manager. Dotted oval square shows the goal of this process that it is to label (add query or not) for new data automatically by machine learning.



AE (Adverse Event)					MH (Medical History)				Derived variables / Input variables			Query / Output variable
SUBJID	AETERM	AEDECOD	AESTDTC	AETOXGR	MHDECOD	MHSTDTC	MHONGO	MHTOXGR	Worsening?	Diff in grade	Diff in days	LABEL
1111111	ABCDE	ABCDE	2020-03-01	Grade 1	XXXXX	2012-04-02	No	Grade 3	0	-2	-56	0
1111111	Worsening AAAAA	AAAAA	2019-12-02	Grade 2	AAAAA	2017-01-01	Yes	Grade 2	1	0	-33	1
2222222	CCCCC	CCCCC	2020-01-03	Grade 1	--	--	--	--	0	0	0	0
1234567	Worsening YYYYY	YYYYY	2020-02-22	Grade 2	YYYYY	2019-06-03	No	Grade 1	1	1	15	1
9999999	Worsening ZZZZZ	ZZZZZ	2020-02-01	Grade 1	ZZZZZ	1999-01-01	No	Grade 1	1	0	-39	0
.....									,,,			
0421324	Worsening abcde	abcde	2020-02-14	Grade 1	abcde	2019-12-02	Yes	Grade 2	0	-1	20	???

Figure 2. Concept of labeled dataset with input/output variable

Feature selection from data

Now select one data review criteria from global standard as an example, that is **“If an AE verbatim includes "worsening of XXX", then verify that there is a previous compatible AE or Medical History (MH) condition”**.

To develop classification model with high performance, feature selection should be carefully considered. In above criteria, below 5 additional derived variables from a) to e) from 3 features in Table 1 are included in input data during data preprocessing. Note that all input data need to be in numeric representative, character value should be converted into numeric value for machine learning in Python.

Feature	**Derivation of derived variable**	
AETERM includes text of “worsening”.	a)	Set to 1 if AETERM included “worsening”. Otherwise set to 0.
“Worsening” between AE and MH (AETOXGR / MHTOXGR)	b)	Difference from AETOXGR to MHTOXGR in numeric representative. For example, 2 when MHTOXGR = 1 and AETOXGR = 3.
	d)	Duration calculated by AESTDTC – MHSTDTC + 1.
“Worsening” between AE and previous AE within same AE term.	c)	Difference from AETOXGR and previous AETOXGR within same AE term (same AEDECOD).
	e)	Duration calculated by AESTDTC – previous AESTDTC + 1 within same AE term (same AEDECOD)

Table 1. Derived variables corresponding feature of data review criteria

After data preprocessing and feature selection, choose model for classification. In this concept, decision tree and random forest techniques are selected because those techniques focus on each input variable and determine the class by dividing the data by a certain value in input variable. For example, to divide data when AETERM contains “worsening” or not and AE occurs after event in MH or not. Especially decision tree can detect impacts to output variable from each individual input variable.

MODEL TRAINING AND TUNING / MODEL VALIDATION

To implement model training and tuning / model validation by Python, Anaconda distribution with Python version 3.7.4 is used.

Decision tree classifier

Decision Tree and Random Forest are available from **DecisionTreeClassifier()** for Tree sub module and **RandomForestClassifier()** for ensemble sub module of scikit-learn library.

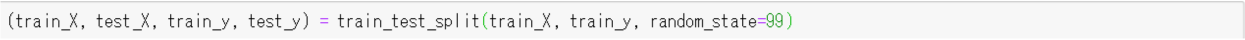
Decision Tree is a non-parametric supervised learning method used for classification and regression. The goal is to create a model that predicts the value of a target variable by learning simple decision rules inferred from the data features. This paper does not address its algorithm. If you would like to learn more about decision tree algorithm, scikit-learn user guide is very informative and useful.

Table 2 shows attribute of all labeled data to build model.

Total number of records in AE domain	740
Number of records with queries	31

Table 2. Summary of data to build model

As a first step, feature value should be derived and considered as input variables. Specifically those are a) to e) in Table 1 in previous section. After adding derived variables as features, data is to be split into train and test set. Data can be easily divided by **train_test_split()** from model selection sub module of scikit-learn (Display 1). Train_X and train_y contains input variables and output variables respectively.

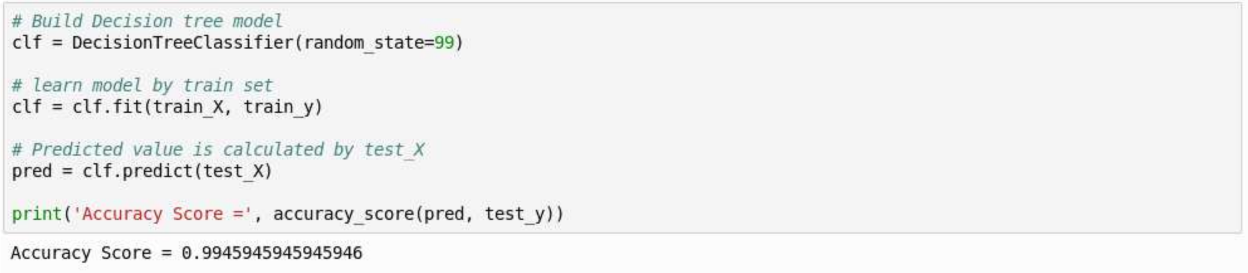


```
(train_X, test_X, train_y, test_y) = train_test_split(train_X, train_y, random_state=99)
```

Display 1. Python code to split data into train and test data

Secondary decision tree model is built by **DecisionTreeClassifier()**, and model is learned by train. By test output variable (test_y) and predicted value, accuracy score is calculated by **accuracy_score()** function. As train set is manipulated in order to add number of query, accuracy score is relatively high (0.995). Accuracy score is calculated by (number of case which the classification predicted by the model matched the actual label)/(number of test data).

Display 2 shows Python code to build decision tree model and calculate accuracy score.

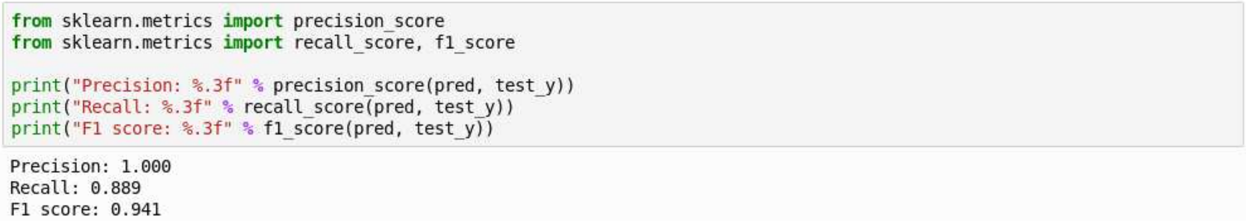


```
# Build Decision tree model
clf = DecisionTreeClassifier(random_state=99)

# learn model by train set
clf = clf.fit(train_X, train_y)

# Predicted value is calculated by test_X
pred = clf.predict(test_X)

print('Accuracy Score =', accuracy_score(pred, test_y))
```

```
Accuracy Score = 0.9945945945945946
```

Display 2. Python code to build decision tree model and calculate accuracy score

Precision and recall are also considered. Precision is True Positive Rate (TPR) defined as a rate of data predicted positive that is actually positive. Recall is False Positive Rate (FPR) defined as a rate of actual positive data that could be correctly predicted as positive. F1 score is calculated by (2 * precision * recall) / (precision + recall), in other words, F1 score is harmonic mean. When F1 score is close to 1, it can be said that model has good prediction. Display 3 shows python code to calculate precision, recall and F1 score.



```
from sklearn.metrics import precision_score
from sklearn.metrics import recall_score, f1_score

print("Precision: %.3f" % precision_score(pred, test_y))
print("Recall: %.3f" % recall_score(pred, test_y))
print("F1 score: %.3f" % f1_score(pred, test_y))
```

```
Precision: 1.000
Recall: 0.889
F1 score: 0.941
```

Display 3. Python code to display precision, recall and F1 score

Hyper parameter tuning for decision tree

Max_depth option in **DecisionTreeClassifier()** is the hyper parameter of decision tree and sometimes it needs to be tuned during learning process to improve model performance. Variable that needs tuning by human is called “Hyper parameter” and this is carefully monitored as per selected model. Figure 3 shows accuracy score when max_depth changes. This shows max_depth = 3 would be appropriate to use in this situation. Note that **random_state** option is also an important hyper parameter to keep reproducibility.

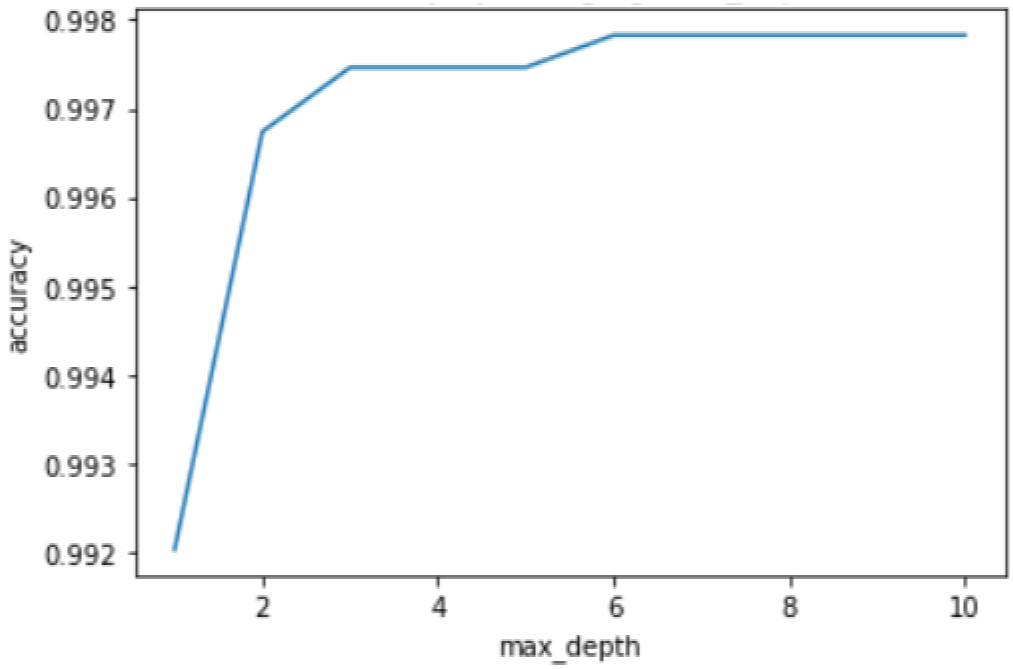


Figure 3. Accuracy rate by changing max_depth.

Now evaluate model obtained in previous section. Table 3 shows summary of new data and this is an actual case of clinical trial. Usually query are not kept open because query is supposed to be closed once it is clarified and/or data is updated.

Total number of records in AE domain	2761
Number of records with queries	4

Table 3. New data to evaluate model

As a result, number of newly labeled query is 6 and accuracy score is calculated as 0.998. However, it does not mean query mapping is successful.

Because number of records with query is very small compared to those without query (imbalanced), even accuracy score is high, model should be carefully interpreted. Figure xx shows precision, recall and F1 score. Precision = 0.500 means that model detects 2 queries out of 4. Recall = 0.333 means one third of output (records detected as query by model) is correct. More specifically, if machine newly labels 6 query and 2 (out of 4) true queries are detected. To improve these scores, model update (e.g. derivation/selection of new features, hyper parameter tuning in model, and resampling of imbalanced data) should be considered.

Random forest classifier

A random forest algorithm is constructing a multitude of decision trees at training time and outputting the class that is the mode of the classes of the individual trees. It is a meta estimator that fits a number of decision tree classifiers on various sub-samples of the dataset and uses averaging to improve the predictive accuracy and control over-fitting.

Results when new data in table 2 is applied for evaluation are,

- Number of newly labeled query: 3 (6 in decision tree)
- Accuracy Score: 0.999
- Precision: 0.500
- Recall: 0.667
- F1 score: 0.571

Hyper parameters of random forest are usually **n_estimator, max_depth**, Those results will change when hyper parameters are considered.

CONSIDERATIONS

In fact, data review criteria in the above example can be implemented by rule-based programming. Especially to detect very minor events, rule-based programming might be more effective than using machine learning. However, this does not mean machine learning is useless for data review automation. Key factor of this concept is to identify features from data which represent data review criteria. Once features are defined from data, machine learning can be more efficient and easier than rule-based programming code. In addition, data manager is not the only one responsible role of data review process. Experts with clinical scientific and medical background, safety and pharmacovigilance specialists, etc. review data from their points of view. Through the labeled data, their expertise is transferred to machine, which will be utilized for future data review. This is one of the biggest advantages of machine learning.

There are four important points to consider. Firstly, low occurrence query might be difficult to detect as described above. Secondary, machine may have a bias when data contains bias. Output variable to label queries is delivered by several reviewers possibly with biases. Thus, data manager (and/or other data reviewers) should consider to have common understanding of data review criteria to control bias in queries in training data. Thirdly, missing data should be handled carefully. It is recommended to consult appropriate function such as statisticians or data scientists. Lastly, a condition and/or threshold for when and how to stop feeding data to model should be thoroughly determined. As far as dealing with same criteria, model should be carefully updated when new data even that is an action to improve model. We should specify criteria or thresholds appropriate enough to ensure the accuracy of the model that would not be changed easily.

CONCLUSION

It is possible to let machine learn the manual review of clinical trial data by utilizing standardized data and standardized data review criteria, and this approach can be used as a support tool for clinical data cleaning. It may allow clinical trial team members to focus more on human-oriented tasks such as thorough planning of the clinical trials itself, data to be collected, and data cleaning required, and thorough evaluation of the data insights and risk assessments.

There still are uncertainties that should be clarified and further investigated, such as, what will happen to the accuracy when machine ingest much more data from multiple clinical trials, how to define a condition and/or threshold to stop machine learning, and what kind of data review criteria are optimal for application of this approach is required. In this paper, sample data review criterion used is the task of Data Manager. Most of Data Manager’s criteria could be made into rules, putting aside the complexity. Therefore, clinical, scientific, and medical review criteria which may need more complicated learnings of diseases, medical guidance, etc. will be the next target for the experiment, as it seems to be significantly beneficial if data review by machine can be utilized as their support tool.

REFERENCES

Guidance of scikit-learn, Available at https://scikit-learn.org/stable/index.html

Nicolas Dupuis, Kevin Lee. PhUSE EU Connect 2018. “Introduction to Machine Learning”. Available at https://www.lexjansen.com/phuse/2018/ml/ML01.pdf.

SAEED AGHABOZORGI, Ph.D. and Joseph Santarcangelo, Ph.D. “Machine Learning with Python by IBM”. Available at Coursera.

Akihiko Ishikawa. 2018. “あたらしいPythonで動かして学ぶ! 深層学習の教科書 機械学習の基本から深層学習まで”. Shinjuku, Tokyo: Shoeisha.

ACKNOWLEDGMENTS

The authors would like to show gratitude to Tomoyuki Kakizume, the colleague from Novartis K.K., who has provided insight and expertise that greatly assisted our research, although he may not agree with all of the interpretations and conclusions of this paper.

CONTACT INFORMATION

Your comments and questions are valued and encouraged. Contact the author at:

Mirai Kikawa
Novartis Pharma K.K.
mirai.kikawa@novartis.com

Yuichi Nakajima
Novartis Pharma K.K.
yuichi.nakajima@novartis.com