A Study on Childcare Information Extraction from Twitter Text

Doctoral Thesis

MAYINUER ZIPAER

March 2024

A Study on Childcare Information Extraction from Twitter Text

By

MAYINUER ZIPAER

Doctoral course Thesis approved by the Graduated School of Advanced Technology and Science for the Degree of

Doctoral of Engineering

in

System Innovation Engineering

March 2024

Department of Information Science and Intelligent Systems
Tokushima University, Tokushima 770-8506, Japan

List of Figures and Tables

Contents

Abstract

It is difficult to obtain necessary information accurately from the Social Networking Service (SNS) while raising children, and it is thought that there is a certain demand for the development of a system that presents appropriate information to users according to the child's developmental stage.

There are still a few examples of research on knowledge extraction that focuses on childcare. This research aims to develop a system that extracts and presents useful knowledge for people who are raising children, using texts about childcare posted on Twitter. In many systems, numbers in text data are just strings like words and are normalized to zero or simply ignored.

In this paper, we created a set of tweet texts and a set of profiles according to the developmental stages of infants from "0-year-old child" to "6-year-old child". For each set, we used ML algorithms such as NB (Naive Bayes), LR (Logistic Regression), ANN (Approximate Nearest Neighbor algorithms search), XGboost, RF (Random Forest), decision trees, and SVM (Support Vector Machine) to compare with BERT (Bidirectional Encoder Representations from Transformers), a neural language model, to construct a classification model that predicts numbers from "0" to "6" from sentences. The accuracy rate predicted by the BERT classifier was slightly higher than that of the NB, LR, ANN, XGboost, RF, decision trees, and SVM classifiers, indicating that the BERT classification method was better.

Chapter 1
Introduction

The information necessary for childcare changes constantly depending on the child's developmental stage. Although traditional media, such as books and magazines, provide a great deal of information on childcare, it is difficult to obtain the information needed at any given time, and a system is needed to present appropriate information to users. To address this issue, to extract information for childcare from SNS, especially Twitter, it is necessary to find information that is appropriate for the child's developmental stage.

The purpose of this research is to develop a system that extracts and presents useful information for people who are raising children from text related to childcare posted on Twitter. We proposed a method for classifying childcare texts based on the age in the text.

1.1 Motivation

There are still few examples of research on knowledge extraction focusing on childcare, for example, there is research to extract life events from SNS [1]. They contributed a codebook to identify life event disclosures and build regression models on the factors that explain life event disclosures for self-reported life events in a Facebook dataset of 14,000 posts [2]. Choudhury et al. collected the users posted about their "engagement" on Twitter and analyzed the changes in words and posts used [3]. Burke et al. also analyzed users who have experienced "unemployment" by advertising or email on Facebook and analyzed their activities on Facebook before changing stress and taking new jobs. However, in this research, we collect texts specialized in childcare [4]. but in this research, we collect texts specialized in childcare and specialize in childcare. We aim to develop a more accurate method by conducting the analysis. As a method, we mainly use natural language processing technology using neural networks, which has been rapidly developing in recent years, especially.

1.2 Problem Description

The relationship between numerals and words in text data has received less attention than other areas of natural language processing. Both words and numerals are tokens found in almost every document, but each has different characteristics. However, less attention is paid to numbers in texts. In many systems, numbers are treated in an ad hoc way, documents are just strings like words, normalized to zero, or simply ignored.

Information extraction (IE) is a question-answering task that asks the a priori question, "Extract all dates and event locations in a given document [5]." Because most of the information extracted is numeric, special treatment of numbers often improves the performance of IE systems. For example, Bakalov and Fuxman proposed a system to extract numerical attributes of objects given attribute names, seed entities, and related Web pages and properly distinguish attributes having similar values [6]

Those related papers described a study of the Nepali language, but this study focuses on the Japanese language. Chiranjibi et al proposed a new hybrid feature extraction method that combines both syntax (word bags) and semantics (domain-specific and fast Text-based) in the Nepali context [7]. For further improvement in natural language processing (NLP), research works in the Nepali language. Tej Bahadur et al. served different NLP research works with associated resources in the Nepali language [8].

1.3 Contribution

Our contribution can be summarized as follows.

(1) To extract information for childcare, it is necessary to find information according to the child's developmental stage. In this study, we proposed a method to classify childcare texts based on the age in the text ("2-year-old child", "for how many children a person has", etc.). To the best of our knowledge, this is the first research on predicting the age of children appearing in text using the surrounding words.

(2) By using the profile information of the user who posted the text together with the text and grasping the attribute information of the poster, we aim to develop a method that emphasizes the text that is closer to the user's situation.

(3) We used a BERT-based neural algorithm as well as several non-neural algorithms including NB (Naive Bayes), LR (Logistic Regression),

ANN (Approximate Nearest Neighbor algorithms search), XGboost (eXtreme Gradient Boosting), RF (Random Forest), decision trees, and SVM (Support Vector Machine) providing exhaustive evaluation on this task. We obtained parenting texts from two types of services, Twitter and mamari, and analyzed the differences between them. We tried learning using mamari's data as training data and Twitter data as test data.

In these three points, we think that the research will be highly novel in terms of method.

1.4 Outline

This chapter gives the motivation, problem description, contribution, and outline of the thesis. Chapter 2 goes over the prerequisite concepts of the classification system. It explains the basic concepts of neural language processing, machine learning, classifications, and numerical classification. Chapter 3 describes related research. Chapter 4 explains the analysis using the proposed methods of SVM and BERT and their elemental technologies. Chapter 5 describes experiments using the proposed method and discussion of the experimental results of Twitter data. Chapter 6 describes experiments on mamari data using the proposed method and a discussion of the experimental results. Finally, in Chapter 7 conclusion and future work are discussed.

Chapter 2
Prerequisite Concepts of Classification

2.1 Natural Language Processing

Natural language processing (NLP) [33] is an interdisciplinary field of computer science and linguistics. It is concerned with supporting human language and giving the computer operational capabilities. Machine learning can be used to process natural language datasets such as text corpora and speech corpora. You can also accurately extract, categorize, and organize the information and insights contained in your documents. Among the challenges in natural language processing are natural language understanding and natural language generation.

Natural language understanding (NLU) [34] or natural language interpretation (NLI) [35] is a difficult problem for AI [36].

The field of natural language understanding has received commercial interest due to its applications in analytics such as automated inference [37], [38] question answering [39], news gathering, and text classification.

Document classification [40] is a problem in library science, information science, and computer science. You can do this "by hand" or algorithmically. Manual classification of documents has mainly been performed in the field of library science, and algorithmic classification of documents has been mainly performed in information science. Classified into text, images, music, etc. If you do not specify a document type, it will be indicated as text classification.

The various processes performed in text classification are shown in the figure below. (see Figure 2.1) [41].

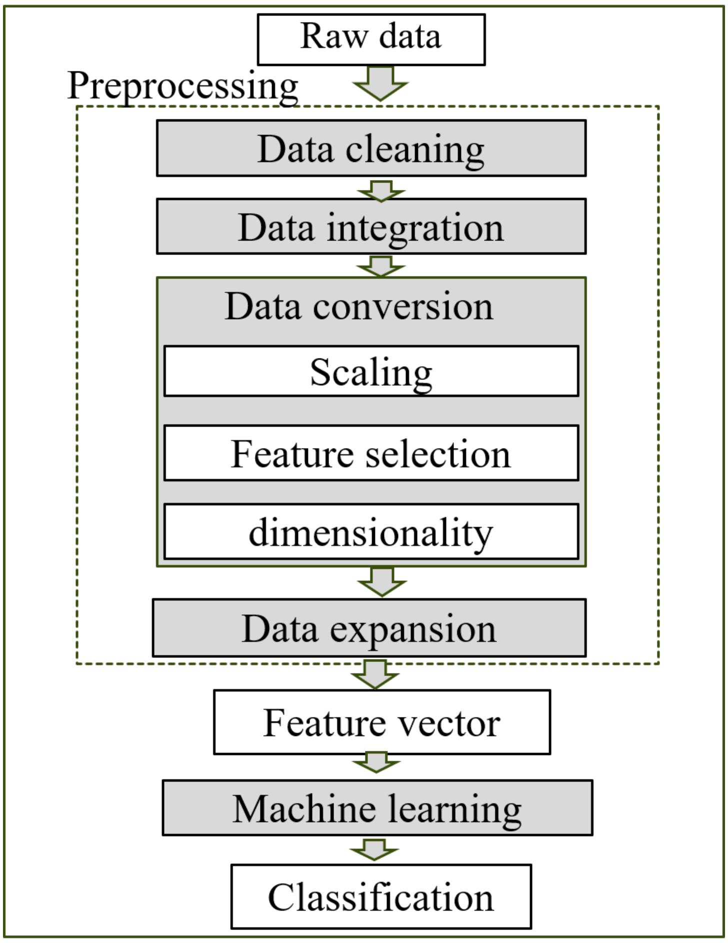

Fig 2.1 Flow of machine learning

2.2 Machine Learning

Machine Learning (ML) [42] is a research field in artificial intelligence that is concerned with the development and study of statistical algorithms that can learn from data and perform tasks without instructions [43]. Generative artificial neural networks have been able to outperform many approaches [44][45]. Machine learning has three broad categories: supervised learning, unsupervised learning, and reinforcement learning.

Supervised learning (SL) [46] trains a model with input objects and desired output values. Training data is processed to construct a function that maps new data to expected output values [47]. In unsupervised learning [48], the algorithm learns only from unlabeled data.

Chapter 3
Related Works

3.1 Related Research on Information Retrieval from Twitter

Those related papers described a study of the Nepali language, but this study focuses on the Japanese language. Chiranjibi et al proposed a new hybrid feature extraction method that combines both syntax (word bags) and semantics (domain-specific and fast Text-based) in the Nepali context [7]. For further improvement in natural language processing (NLP), research works in the Nepali language. Tej Bahadur et al. served different NLP research works with associated resources in the Nepali language [8].

Forss et al. [28] investigated a method to analyze people's social media profiles and extract information about their hobbies and interests. We proposed a baseline method for extracting user interests through profile analysis that uses heuristic rules and TF-IDF and analyzed English-speaking users. This is because by combining keyword extraction with predefined dictionaries and named entity recognition, it is possible to cover a wider range, and we have explained the advantages of keyword extraction. This research shows that machine learning is effective and that high accuracy can be achieved by narrowing down the categories that serve as user attributes. However, this method estimates the user's attributes themselves, not the user's childcare information.

Kawashima et al. [30] focused on requests that directly indicate consumer needs among reviews posted on Twitter, and proposed a method to extract posts that include requests from Twitter. Supervised machine learning algorithm to extract posts that include requests in addition to applying the SVM system, we also attempted to apply " Distant Supervision" a type of semi-supervised learning, to collect training data. The effectiveness of the proposed method, which aims to extract requests with low cost and high accuracy, has been confirmed and reported.

Kato et al. [31] estimated user attributes and habitual behavior on Twitter. Their method used not only posted content and user profile text but also users' lifestyle information. To extract opinions about commercially available products and TV programs, Ikeda et al. [32] analyzed opinions posted on Twitter and estimated user profiles such as age, gender, and region.

In this research, we analyze the data of Twitter texts posted by users and profile data.

3.2 Related Work of Numerical Classification

The relationship between numbers and words in text data has received little attention compared to other areas of natural language processing. Words and numbers are both tokens that appear in almost every document, but each has different characteristics. However, less attention is paid to numbers in text, and many systems treat numbers in documents as just strings of characters, such as words, and either normalize the numbers to zero or simply ignore them. The recent growth of the natural language processing (NLP) research field has changed this situation by increasing the focus on the computational power of documents [5].

Natural language processing (NLP) [5] is a field of research that helps machines understand the meaning of textual data (usually lists of words). In some cases, text cannot be understood in closed form, i.e. non-linguistic data. Numbers are an important data format for non-word data, and many documents have associated metadata, such as a numerical publication date, as well as the document itself. This is because it contains numbers such as "3 years old". There are many variations in the collaborative mining of text and its associated numerical metadata, and many studies have been proposed.

3.2.1 Related Research on Information Retrieval

Yoshida et al. [25] proposed a suffix array-based text mining system enhanced with numerical processing that accepts range queries such as "[1,000 - 10,000] feet." The system allows what they call "numerical similarity" to be introduced into this contextual information. By string search using a suffix array and numerical clustering using a Dirichlet process mixture model, it was possible to use a range of numbers as a query, and it was realized that the numbers appearing in the search would match.

Kenzo Kurotsuchi et al. [27] explored information extraction techniques aimed at supporting business decision-making. In this report, we will work on extracting numerical information on physical property values from papers, to create a system that supports decisions regarding technological trends. When extracting physical property values, numerical values, and item names were combined and extracted. Regarding numerical values, we considered the extraction of numerical values without units. They used a method to extract pairs of item names and numbers using pattern matching based on a dependency structure. They evaluated English papers and confirmed the effectiveness of their method.

Sarawagi and Chakrabarti [29] proposed a system to answer quantity queries on Web tables such as "escape velocity Jupiter". Their system contains the modules to interpret the numbers presented in the table cells to improve accuracy. On the contrary, queries also can be numbers.

3.2.2 Related Research on Predicting Numbers in Sentences

Some researchers have tried to acquire numerical common sense as a parameter for language modeling. In the study of predicting numbers in sentences, the system can directly predict the numbers that fill in the blanks in the text, or estimate the feasibility of the numbers presented in the text, without explicitly collecting the knowledge base described above. For example, given the sentence "My [MASK] year old son's height is he is 110 cm", the system will answer the possible values entered in the " [MASK] " position. is needed [25].

Zhang et al. [11] investigated how pre-trained language models like BERT can predict (the discretized version of) the attribute with continuous numeric values such as MASS or PRICE with evaluation with DoQ (Distributions over Quantities). Identify contextual information in pre-training and numeracy as two key factors affecting their performance, the simple method of canonicalizing numbers can have a significant effect on the results.

Recent research has shown that pre-trained language models (PTLMs) such as BERT have a certain amount of common sense and factual knowledge. On the other hand, Lin et al. [26] is considered to be a more difficult task to predict Fill in the blanks in the text with exact numbers, such as "Birds usually have a [mask] feet". They say that the current pre-trained model has his BERT and RoBERTa's performance was poor. Language models that do not use an encoder-decoder model have also been proposed [25].

They investigated whether numerical common-sense knowledge can be induced from PTLM and to what extent the process is robust. The analysis revealed the following three points. (1) BERT and its powerful variant Roberta perform poorly on diagnostic datasets before fine-tuning. (2) Fine-tuning through remote monitoring provides some improvement. (3) Even the best-supervised models still perform poorly compared to human performance [11].

Spithourakis and Riedel [27] proposed a language model for a set of words. and numbers. You can find the probability of representing a word and a number simultaneously. For example, the probability that the number "50,000" appears immediately after. The word sequence "He's the number of video game consoles I have ". introduced model the probability that each token is a word or a number, number probabilities free of word probabilities, with some variations contain a mixture of number-based RNNs and Gaussians [25].

Chapter 4
Proposed Method

The method proposed in this research classifies tweets about child-rearing posted on Twitter into classes.

The procedure is shown below.

- Acquire a large amount of text data from Twitter.
- Terms related to child care are selected, and two types of texts (tweets) containing those terms and profiles are formed. Create a set of texts divided into children's developmental stages.
- Using SVMs, etc. (the NB, LR, ANN, XGboost, RF, decision trees, and SVM) and the neural language model BERT, we build a classification model that predicts numbers from "0" to "6" from sentences.

Fig. 4.1 shows a simple flow for classifying tweets related to child-rearing.

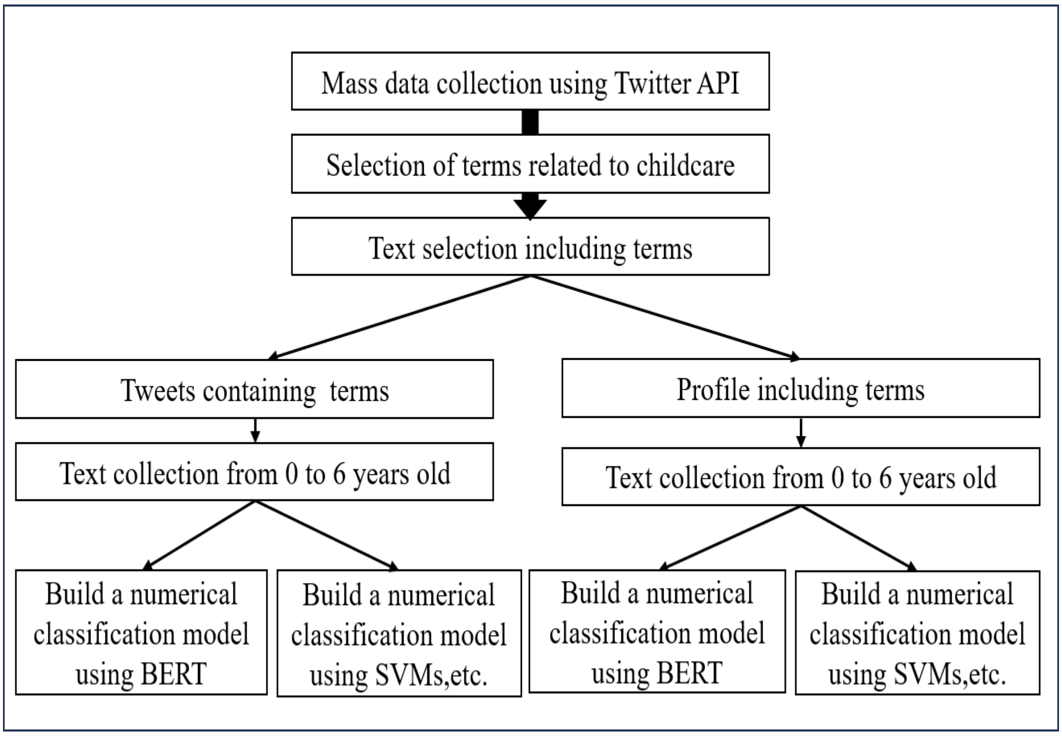

Fig. 4.1 Flow of the proposed method

4.1 Data Collection

In this research, we use Twitter's API to obtain texts that target child-rearing information from Twitter. Next, we select terms related to parenting and screen texts (tweets) containing those terms. Parenting terms are shown in Table 1.

Table 1 Parenting terms

Search words used to collect text	Search terms used to collect profiles
zero-year-old ,0-year-old child, One year old,1-year-old, Two years old,2-years old, Three years old,3-years old, Four years old,4-years old, Five years old,5-years old, Six years old,6 -years old, child	raising children, Dad, Mum Mother, Father, my dad, my mum, raising children, child

Two types of text collections are formed, one for containing terms in tweets and the other for containing terms in profiles. Search for the character string "*-year-old child" and create a set of texts classified by the developmental stage of infants from "0-year-old child" to "6-year-old child". Table 2 and Table 3 show part of the tweet text set and profile set created according to the developmental stages of infants.

Table 2 Part of the tweet text set collection from 0 to 6 years old

age	Part of the original tweet texts
4	I'm coming to Tokyo Disney, and when I saw the Coconut *year old boy said, "Mom! This is Hawaii!" and I laughed.
5	*Year-old boy often shakes his head. Isn't the ex-nurse mother-in-law a tic ... https:\/\/t.co\/KstYS2lcJ1
0	My daughter's album has stopped until *-year-old 5 months ...that's bad... After all, I haven't had time since I returned to work.
3	@0246monpe in May become *-year-old! In the pitch-black room, I can hear the *year-olds humming (groaning?) (´－｀)\n I'm depressed that it will be crowded, but I'm looking forward to it and I have to take him (^◇^;)
1	@Alice_ssni*-year-old \n started crawling. he can get over my mother's body too. \n Even when my mother was lying on her back, I was able to come up to π and drink. \n, thank you for the meal. Itadakimasu has a high probability of being done by yourself (when you wearing an apron then understand).
2	RT @FururiMama98: Parenting is hard and painful. Now, I have a *-year-old daughter, but it's really hard. It's really hard to kill myself and keep looking at others and hugging them. Even after chasing, cuteness: frustration = about 1:9. He is also an unbalanced eater, and is always overturned with toys around the house. How can I rest and comfort myself...
6	RT @unikunmama: 🎉happy birthday unnie🎂\n I turned *-year-old today (o^^o) Let's have fun together from now on, https:\/\/t.co\/ybW9AcS4oP

Table 3 Part of the profile set collection from 0 to 6 years old

age	Part of the original profile
0	I am a housewife raising a *-year-old child.
6	Full support for the Hiroshima Cup in Tokyo. Takuya Kimura was my all-time favorite player. Mother of a * -year-old child. She is from Hiroshima Prefecture. She lives in Tokyo.
2	around four working mothers who are raising a *-year-old child. I like music, Hanshin Tigers, Daichi Miura, and animals.
4	I live with Sunhora and Ayanogo. Adult. I'm quietly doing cosplay. I'm a mom of a *-year-old child. RT is too much.\n Archive ID: *52993
1	An adventurer who explores cognitive science and sign language with a focus on linguistics. Specializes in cognitive linguistics. Postdoc in R. With *-year-old child. Working mama first grader. http:\/\/ask.fm\/rhetoric
3	With * year-old daughter, Shinma who is taking care of her family.\nTwitter is still not working.
5	These daily mumblings of an unfortunate rotten adult who likes anime, manga, and sometimes games. Childcare (*Year-old child) Mutters here and there. Currently pregnant with a second child. (Scheduled for the second half of October) Gestational diabetes, hospitalization for threatened premature labor, etc. I'm already full_ (:3" ∠) _

4.2 Preprocessing

For the data, the tweet text and profile are used as they are. Therefore, information such as retweets were used as is. We have not done any processing such as removing hashtags.

Data pre-processing is performed to perform machine learning on the set of created tweet texts and profiles. In preprocessing, morphological analysis is performed with MeCab, a tool specialized for Japanese language analysis, and the document is vectorized using the library TfidfVectorizer of Scikit-learn. We used unigram features. Unigrams are standard features for text processing, especially in the case where the data size is small. We did not use bigram or trigram features as we did not have enough data to provide meaningful values for these advanced features.

Morphological analysis is the task of breaking down the words that make up a sentence into the smallest units and dividing and writing the sentences. Given a set of documents, TfidfVectorizer converts each document into a vector based on the TF-IDF values. TF (Term Frequency) represents the frequency of occurrence of words in a document. IDF (Inverse Document Frequency) is a score that lowers the importance of a word that appears in a large number of documents. TF-IDF is a metric that is a product of TF and IDF [2].

4.3 Classification Method

BERT+fine tuning is used in this research, so it is neural supervised learning. We used ML algorithms such as NB, LR, ANN, XGboost, RF, decision trees, and SVM model to compare with the pre-trained model with fine-tuning the BERT model. We used these algorithms because it is a standard non-neural algorithm.

4.4 Baseline Method

As baseline methods, we used the NB, LR, ANN, XGboost, RF, decision trees, and SVM algorithm and the implementation used the Python machine learning library Scikit-learn to predict the numbers "0" to "6" from the sentences to build a classifier. Classifiers using NB, LR, ANN, XGboost, RF, decision trees, and SVM create two types: Task Ts (tweet), which uses tweet sentences as data, and Task Ps (profile), which uses profiles.

4.4.1 Naive Bayes Classifier

Naive Bayes classifier [12] is a typical classifier based on probabilistic models. Although learning is just counting the frequency of features, it is extremely fast. Although it is generally believed that its classification performance is somewhat lower than the latest classifiers, it is still commonly used, perhaps because it is easy to implement. Although it is a simple classifier, there is much to be learned from understanding it correctly.

The Naive Bayes classifier is a classic classifier that is still used today, and when used properly it often shows high performance.

The Naive Bayes classifier is probability-based, and for example, d outputs the class $c \in C$ with the largest $P(c|d)$. This probability can be determined in various ways to find $P(c|d)$. First, we use the following property called Bayes' theorem:

$$P(c|d) = \frac{P(c)P(d|c)}{P(d)} \tag{4.1}$$

Class c which has the maximum right-hand side will be output, but since the denominator $P(d)$ does not depend on the class, it is not necessary when determining the maximum class. That is, output classmax_cthat maximizes $P(c)$ $P(d|c)$ of the molecule:

$$c_{max} = \text{arg}max_c \frac{P(c)P(d|c)}{P(d)} \tag{4.2}$$

$$=\text{arg}max_c P(c)P(d|c). \tag{4.3}$$

It would be nice if we could find the right-hand side of this, but calculating $P(d|c)$ is not easy.

Since d is a document, considering the number of types of words and their combinations, the number of possible ds is enormous. It is unrealistic to find $P(d|c)$ by maximum likelihood estimation by checking how many times each d occurs in the data. The naive Bayes classifier assumes a simplified model for document d and calculates the value of $P(d|c)$ [12].

4.4.2 Logistic Regression

Logistic regression is a type of statistical regression model for variables that follow a Bernoulli distribution. It is also a type of generalized linear model (GLM) that uses logit as the connectivity function. It was published by David Cox in 1958.[16] It is a probability regression and is mainly used for classification in statistics. Although it is called "regression", logistic regression is a classifier algorithm.

If regression was used as is for classification, the regression results would not fall within the range 0 to 1.

Therefore, we use something called a sigmoid function. The sigmoid function is also called the logistic function. For the sigmoid function, no matter what value x is input, the result z will be between 0 and 1.

The sigmoid function is

$$z = \frac{1}{1+e^{-x}} \tag{4.4}$$

If we insert $P(x)$, which is the result of linear regression, into x of the sigmoid function mentioned above, we can successfully keep the result of linear regression within 0 to 1.

$$P(x) = \frac{1}{1+e^{-(\theta 0+\theta 1x)}} \tag{4.5}$$

The result of logistic regression is a value between 0 and 1, but this can be treated as a probability (label is 1) as is, so the probability $p(x)$ when x is input is the output of logistic regression. It can be the result.

Logistic regression is only for two-class problems. Two-class classification uses a sigmoid function, but multi-class classification uses a SoftMax function. The SoftMax function is a multiclass version of the sigmoid function, if we transform the formula of the sigmoid function, z, which was used as the probability of belonging to one class in binary classification, will look like this.

$$z = \frac{e^{x1}}{e^{x1}+e^{x2}} \tag{4.6}$$

Multinomial logistic regression looks like this:

$$p_k(x) = \frac{e^{\theta_0^k+\theta_1^k x}}{\sum_{k=1}^{K} e^{\theta_0^k+\theta_1^k x}} \tag{4.7}$$

4.4.3 Decision Trees

A decision tree [13] is a simple flowchart for selecting labels for input data. This label consists of a decision node that checks the value of the feature and a leaf node that assigns the label. To select the inputted direct label, processing starts from the first decision node in the flowchart, which is called the root node.

This node contains a conditional branch that examines one of the features of the input value and uses the value of the feature of the input value to select the branch to proceed. Following the branch chosen based on the input value, there is a new decision node that contains a new conditional branch that tests one of the features of the input value. In this way, branches are selected based on the conditional branches placed at each node until finally reaching the leaf node for which the label assigned to the input value is specified.

4.4.4 Random Forest

Random forest (Random Forest, randomized trees) is a machine learning algorithm proposed by Leo Breiman in 2001 [14] that is used for classification, regression, and clustering. It is an ensemble learning algorithm that uses decision trees as weak learners, and its name comes from the use of a large number of decision trees learned from randomly sampled training data [15].

4.4.5 XGboost

XGBoost [17], which stands for eXtreme Gradient Boosting, is a decision tree gradient boosting algorithm. XGBoost uses a Bayesian boosting algorithm. Boosting is a method that weights data that the model predicts incorrectly, repeatedly learns, and ultimately weights multiple models to make predictions.

Gradient Boosting [18] is a machine-learning technique for tasks such as regression and classification, in the form of an ensemble of weak prediction models (usually decision trees). Generate a predictive model [19][20].

Ensemble [21] refers to learning multiple models and producing predicted values by majority vote (or average). If the decision tree is a weak learner, the resulting predictor is called a gradient-boosted tree and is usually better than a random forest [22]. It builds the model step-by-step like other boosting methods, but it is generalized by allowing optimization of arbitrary differentiable loss functions.

4.4.6 Approximate Nearest Neighbor Search

Nearest neighbor search [23] is a problem in which, when there is a set of many data and one piece of data that does not belong to the set is given, the closest data is found in the set. In cases where the set itself represents a class (category), such as in the nearest neighbor identification problem where the data given during this search is called a "query" (question), the accumulated data is called a "prototype", but in normal nearest neighbor search problems, it is more often simply called "data".

To explain formally, a distance d ($\boldsymbol{x, y}$) is defined between data x and y, The problem of finding the closest data (nearest neighbor solution) NN (q, s) for query q from the data set S= $\{x_i\}$ is nearest neighbor search, and the nearest neighbor solution is defined by the following equation.

$$NN(q\ , S)= \arg\min d(q\ , x)\ , x \in S \tag{4.8}$$

ANN (Approximate Nearest Neighbor) [24] is a method that uses a binary tree to perform an approximate nearest neighbor search at high speed. Tree nodes correspond to hyperrectangles that divide the feature space, and leaf nodes are associated with a single feature vector. In ANN, feature vectors for distance calculation are collected by searching a tree structure, and the one with the shortest distance among them is output as the result of an approximate nearest neighbor search. ANN has a tolerance ϵ as a parameter that represents the degree of approximation. If ϵ is large, a larger approximation is performed to narrow down the target feature vectors, reducing processing time.

4.4.7 Support Vector Machine

Support Vector Machine (SVM) [12] is a linear binary classifier that has been explosively used in natural language processing since the late 1990s and is known for its extremely high classification performance. When used in combination with kernel methods, SVM can also perform nonlinear classification. SVM is a linear binary classifier and is used for problems with two classes. The two classes are called a positive class and a negative class, respectively. Normally, we consider the class we are more interested in to be the positive class, but mathematically there is no essential difference between positive and negative classes. An example belonging to the positive class is called a positive example, or a positive instance, and an example belonging to the negative class is called a negative example, or negative instance. The training data *D* is:

$$D = \{\left(x^{(1)}, y^{(1)}\right), \left(x^{(2)}, y^{(2)}\right), \dots, \left(x^{(|D|)}, y^{(|D|)}\right)\} \quad (4.9)$$

Suppose that it is given. $x^{(1)}, x^{(2)}, \dots, x^{(|D|)}$ is the feature vector (case vector) of the case, and $y^{(1)}, y^{(2)}, \dots, y^{(|D|)}$ is the class label of each example. The class label of a positive example is +1, and the class label of a negative example is -1. Since it is a linear classifier, using the direction vector ω and intercept b of the separation plane as parameters, $f(x) = \omega \cdot x - b$ (4.10)

It is expressed using the function. Classify *x* into a positive class if $f(x) \geqq 0$, and into a negative class if $f(x) < 0$ [12].

To train the SVM classifier using the preprocessing results, first convert the sentence list and labels to np. array objects and split them into training data and test data so that they can be used in Scikit-learn. do. In this research, we do not specify what percentage of the data is used as test data, and the division is performed randomly. Next, create a pipeline that connects preprocessing and SVM. A pipeline is a series of processes from input to output. From the outside, it can be viewed as a kind of black box that inputs data, performs internal learning, and outputs the results. Perform SVM training with the function of the created pipeline.

4.5 Proposed Method

Since the input is a sequence of words, an encoder/decoder model applies to this task. In particular, the BERT language model provides a good fit for predicting numbers in sentences. BERT is a deep neural network model that consists of modules called transformers. It is trained on a task where the input is a sequence of words with a special "[MASK]" token and one of the outputs is the estimated original word at the "[MASK]" position [25].

As a language model, the neural language model BERT has been actively studied in recent years. For constructing classification model B, we use neural language model BERT to classify from "0" to "6".

Two types of classifiers of the proposed method using BERT are created: classifier Tb (tweet), which uses tweet text as data, and classifier Pb (Profile), which uses profiles. A BERT model is required to create a classification model. However, it is difficult to prepare a sufficient amount of data set for pre-training to create a model specialized for numerical classification from "0" to "6", so it needs to fine-tune the pre-trained model. A classifier is created by fine-tuning a pre-trained model.

In BERT, a model specialized for a specific task can be configured by fine-tuning using supervised data for each task, so performance improvement can be expected compared to applying a pre-trained model as it is. A trained model uses a BERT-Base model with 110M parameters and a large model size. The data used for fine-tuning are a set of tweet texts and profile sets created by the developmental stages of infants from 0 to 6 years old. The classifier S uses the tweet text set, and the classifier P uses the profile set for fine-tuning. A distinctive feature of BERT is its unified architecture across different tasks.

Chapter 5
Experiment and Results of Twitter Data

5.1 Experimental Settings

Using Twitter 1.1 Streaming API, we get real-time tweets randomly. There is data (16.19MB) of tweets acquired from July 5, 2017 to October 31, 2017. From here on, only tweets containing keywords are used. At that time, we collected tweet texts and profiles containing any of the words shown in Table 1.

Regarding the data size, the profile data is 1575 items and the size is 376 KB, The vocabulary size of the profile is 78720 words. The data of the tweet text is 953 items, and the size is 238 KB, The vocabulary size of the tweet is 55378 words. Items of each category of Tweet data profile data are shown in Table 4.

Table 4 Items of each category of Tweet data and profile data

Category / Data	age_0	age_1	age_2	age_3	age_4	age_5	age_6	Total
Profile-data	206	572	255	170	80	85	207	1575
Tweet-data	46	200	212	205	105	129	56	953

5.2 Experiment Details

In the classification experiment, Task Tb and Task Pb described in Section 4.5 are used to classify tweet texts or profiles described in Section 5.1, and the performance is compared with classifiers other than BERT.

The classifier of the proposed method, which uses BERT, classifies the data as described in the classifier. The Task Pb is fine-tuned on the profile set and classifies the profiles of users who are raising infants aged 0 to 6 years. One of the eight parts of the training data is used for verification and the rest for fine-tuning. We checked the classification results in each case.

For each classifier output, the result output by the BERT classifier is the normalized probabilities ranging from 0 to 1 for each label, summing to 1. As for the classification results, the one with the highest probability of each label for the input is taken as the output.

In baseline classifiers, we also use ML algorithms such as NB, LR, ANN, XGboost, RF, decision trees, and SVM for performance comparison. Nonlinear SVM is to map nonlinear data to a space that becomes linearly separable and linearly separable on a hyperplane. To process the Japanese sentences of the tweets to be classified, the input text is converted to vectors. When we classify with SVM modules, we first normalized with StandardScaler and then used the default parameters. The nonlinear SVM classifies each label from "0" to "6" in the identification space according to which region it belongs.

We performed a 5-fold cross-validation on the train set using the train-test split on the profile data and the tweet body data. 80% of the tweets and profiles used in the experiment were used as learning/verification data, and 20% as test data. Then, using stratified 5-fold cross-validation, the training data is divided into 5 so that the ratio of labels in each division is the same as the overall ratio, and 4 of the 5 are training data, 1 is used as validation data. to train and evaluate the model.

Once evaluation and training are completed, one of the four training data and validation data are replaced, and the model is trained and evaluated again. By doing this five times obtaining the average classification accuracy of the five times, and using that value as the classification accuracy, it is possible to perform a robust evaluation that does not depend on the division of the data.

Regarding the adjustment of the parameters of each machine learning method, the above-mentioned cross-validation is performed for all combinations of parameters specified in advance using grid search, and the parameter model that shows the best classification accuracy is generated. Finally, we tested how well each generated model could classify the test data.

5.3 Experimental Results of Twitter Data

5.3.1 The Classification Results of the Baseline Method.

We performed a 5-fold cross-validation on the train set using the train-test split on the profile data and the tweet body data. Finally, here are the test set results:

Profile data test set results:
Best score on validation set: 0.58
Best parameters: {'gamma': 0.01, 'C': 100}
Test set score with best parameters: 0.57

Results from a test set of tweet body data:
Best score on validation set: 0.28
Best parameters: {'gamma': 0.01, 'C': 100}
Test set score with best parameters: 0.31
The confusion matrix is shown in Table 5.

Table 5 The confusion matrix

		True label	
		Positive	Negative
Prediction label by SVM	positive	(***A***)True positive	(***B***)False positive
	negative	(***C***)False negative	(***D***)True negative

True positive (***A***): True label positive and prediction positive (correct answer).

False positive (***B***): True label negative and prediction positive (wrong answer).

False Negative (***C***): True label positive and prediction negative (wrong answer).

True negative (***D***): True label negative and prediction negative (correct answer).

Regarding the evaluation of test data, we used three values: accuracy, precision, recall, and F1 score.

$$\text{accuracy}=\frac{\boldsymbol{A+D}}{\boldsymbol{A+B+C+D}} \tag{5.1}$$

$$\text{recall}=\frac{A}{A+C} \tag{5.2}$$

$$\text{precision}=\frac{\boldsymbol{A}}{\boldsymbol{A+B}} \tag{5.3}$$

$$\text{F1}=2*\frac{\boldsymbol{precision*recall}}{\boldsymbol{precision+recall}} \tag{5.4}$$

The classification results for each class from 0 to 6 years old by classifier of Baseline methods are shown in Tables 6 to 12 below.

Table 6 Classification results of 0-years-old class

Age_0 / Classifier	Result of profile data			Result of tweet data		
	Precision	Recall	F1	Precision	Recall	F1
NB	0.62	0.40	0.48	0.00	0.00	0.00
LR	0.90	0.26	0.40	0.00	0.00	0.00
ANN	0.53	0.46	0.49	**0.11**	**0.05**	**0.07**
XGBoost	0.31	0.53	0.39	0.00	0.00	0.00
RF	0.35	**0.81**	0.49	0.00	0.00	0.00
Decision Tree	0.19	0.27	0.22	0.00	0.00	0.00
SVM	**0.94**	0.35	**0.52**	0.00	0.00	0.00

Regarding the results obtained from the classification results of profile text data regarding 0-year-old children, these are the results of classifying 206 profile data.

- RF showed high recall at 81%. RF was able to correctly predict 81% of the positive example data, but there were many false recognitions for the negative examples (***B***). That's why precision and F1 are low.
- Precision and F1 had the highest SVM at 94%.and 52%. In SVM, 94% of the predicted positive examples were positive, but there were many incorrect recognitions of positive examples as negative examples (***C***). So, recall and F1 were low.

For analysis of classification results of tweet text data regarding 0-year-old children, this is the results of classifying 46 tweet data.

・ It was lower overall than the profile. Some results could be classified only by ANN. Classifiers other than ANN could not predict anything at all.

Table 7 Classification results of 1-years-old class

Classifier \ Age_1	Result of profile data			Result of tweet data		
	Precision	Recall	F1	Precision	Recall	F1
NB	0.56	0.79	0.65	**0.54**	0.33	**0.41**
LR	0.45	**0.98**	0.62	0.50	0.05	0.10
ANN	0.62	0.62	0.62	0.32	0.45	0.38
XGBoost	0.78	0.57	**0.66**	0.32	0.31	0.32
RF	**0.96**	0.48	0.64	0.30	0.31	0.31
Decision Tree	0.78	0.49	0.60	0.16	**0.69**	0.26
SVM	0.48	0.94	0.64	0.47	0.16	0.24

About the result obtained from the classification result of Profile text data regarding 1-year-old children, this is the result of classifying 572 profile data.

・ The Recall rate had the highest LR at 98%. LR was able to correctly predict 98% of the positive example data, but there were many false recognitions for the negative examples (***B***). That's why precision and F1 are low.

・ RF showed the highest Precision at 96%. In RF, 94% of the predicted positive examples were positive, but there were many incorrect recognitions of positive examples as negative examples (***C***). So, recall and F1 were low.

・ F1 had the highest XGBoost at 66%. XGBoost has a better balance of Precision and Recall than other classifiers.

For analysis of Tweet text data classification results regarding 1-year-old children, this is the results of classifying 200 tweet data.

・ Precision and F1 had the highest NB at 54%.and 41%. NB has a better balance of Precision and Recall than other classifiers.

・ The Recall rate had the highest Decision Tree at 69%. NB was able to correctly predict 69% of the positive example data.

Table 8 Classification results of 2-years-old class

Age_2 / Classifier	Result of profile data			Result of tweet data		
	Precision	Recall	F1	Precision	Recall	F1
NB	0.53	0.41	0.46	0.15	0.20	0.17
LR	**0.93**	0.25	0.40	0.19	0.95	0.32
ANN	0.50	0.43	0.46	0.34	0.31	0.32
XGBoost	0.56	0.50	**0.53**	0.55	0.27	0.36
RF	0.30	**0.75**	0.43	0.64	0.26	0.37
Decision Tree	0.47	0.33	0.39	**0.96**	0.24	0.38
SVM	0.86	0.30	0.44	0.26	**0.77**	**0.39**

About the result obtained from the classification result of Profile text data regarding 2-year-old children, this is the result of classifying 255 profile data.

- LR showed the highest precision at 93%. In LR, 94% of the predicted positive examples were positive, but there were many incorrect recognitions of positive examples as negative examples (***C***). So, recall and F1 were low.
- F1 had the highest XGBoost at 53%. XGBoost has a better balance of Precision and Recall than other classifiers. However, there is a lot of omission of positive example data, which shows that improvement is necessary.
- Recall had the highest RF at 75%. RF was able to correctly predict 69% of the positive example data.

For analysis of Tweet text data classification results regarding 2-year-old children, this is the results of classifying 212 tweet data.

- Precision in Decision Tree was the highest at 93%. In Decision Tree, 93% of the predicted positive examples were positive, but there were many incorrect recognitions of positive examples as negative examples (***C***). So, recall and F1 were low.
- Recall and F1 had the highest SVM at 77%.and 39%. SVM was able to correctly predict 77% of the positive example data. but there were many false recognitions for the negative examples (***B***). That's why precision and F1 are low.

Table 9 Classification results of 3-years-old class

Age _3 / Classifier	Result of profile data			Result of tweet data		
	Precision	Recall	F1	Precision	Recall	F1
NB	0.62	0.27	0.37	0.24	**0.37**	0.29
LR	**1.00**	0.20	0.33	**0.50**	0.02	0.04
ANN	0.45	0.51	**0.48**	0.21	0.24	0.23
XGBoost	0.33	0.41	0.36	0.14	0.21	0.17
RF	0.34	**0.74**	0.47	0.21	0.27	0.24
Decision Tree	0.05	0.29	0.08	0.05	0.33	0.09
SVM	0.68	0.37	**0.48**	0.38	0.36	**0.37**

The result obtained from the classification result of Profile text data regarding 3-year-old children is the result of classifying 170 profile data.

- all rates for Decision Tree were low at 5%,29%and 8%. but there was not much difference for the others. The Decision Tree classifier was not able to predict correctly at all.
- ANN and SVM showed the same high F1 at 48%. ANN and SVM have a better balance of Precision and Recall than other classifiers. However, there is a lot of omission of positive example data, which shows that improvement is necessary.
- LR had the highest precision at 100%. In LR, all of the predicted positive examples were positive, but there were many incorrect recognitions of positive examples as negative examples (***C***). So, recall and F1 were low.

For analysis of Tweet text data classification results regarding 3-year-old children, this is the results of classifying 205 tweet data.

- All rates were generally lower than the Profile. Here, SVM showed stable and high rates at 38%,36%and37%. SVM has a better balance of Precision and Recall than other classifiers. But it needs improvement.

Table 10 Classification results of 4-years-old class

Age _4 / Classifier	Result of profile data			Result of tweet data		
	Precision	Recall	F1	Precision	Recall	F1
NB	0.33	0.21	0.26	0.05	0.04	0.04
LR	**0.67**	0.12	0.21	0.00	0.00	0.00
ANN	0.26	0.28	0.27	0.20	0.15	0.17
XGBoost	0.11	0.50	0.17	0.12	0.23	0.16
RF	0.25	0.62	**0.36**	0.12	**0.75**	**0.21**
Decision Tree	0.01	**1.00**	0.18	0.00	0.00	0.00
SVM	0.62	0.25	**0.36**	**0.50**	0.04	0.07

The result obtained from the classification result of Profile text data regarding 4-year-old children is the result of classifying 80 profile data.

- The Decision Tree had the highest Recall at 100%. Decision Tree was able to correctly predict all of the positive example data. but there were many false recognitions for the negative examples (***B***). That's why precision and F1 are low.
- RF and SVM showed stable and high F1 at 36%. RF and SVM have a better balance of Precision and Recall than other classifiers. But it needs improvement.
- LR had the highest precision at 67%. In LR, 67% of the predicted positive examples were positive, but there were many incorrect recognitions of positive examples as negative examples (***C***). So, recall and F1 were low.

For analysis of Tweet text data classification results regarding 4-year-old children, this is the results of classifying 105 tweet data.

- All rates were generally lower than the Profile. Here, Recall and F1 in RF were the highest at 75% and 21%. RF was able to correctly predict all of the positive example data. but there were many false recognitions for the negative examples (***B***).

Table 11 Classification results of 5-years-old class

Age _5 / Classifier	Result of profile data			Result of tweet data		
	Precision	Recall	F1	Precision	Recall	F1
NB	0.46	0.63	**0.53**	0.26	**0.19**	**022**
LR	**1.00**	0.14	0.25	**1.00**	0.04	0.07
ANN	0.56	0.40	0.47	0.13	0.09	0.11
GXBoost	0.33	0.46	0.39	0.09	0.11	0.10
RF	0.29	**1.00**	0.44	0.00	0.00	0.00
Decision Tree	0.14	0.43	0.21	0.00	0.00	0.00
SVM	**1.00**	0.29	0.44	0.43	0.13	0.20

The result obtained from the classification result of Profile text data regarding 5-year-old children is the result of classifying 85 profile data.

- NB was highest F1 at 53%. NB has a better balance of Precision and Recall than other classifiers. But it needs improvement.
- RF was the highest recall at 100%. RF was able to correctly predict all of the positive example data. but there were many false recognitions for the negative examples (***B***). That's why precision and F1 are low.
- LR and SVM had the highest precision at 100%. In LR, all of the predicted positive examples were positive, but there were many incorrect recognitions of positive examples as negative examples (***C***). So, recall and F1 were low.

For analysis of Tweet text data classification results regarding 5-year-old children, this is the results of classifying 129 tweet data.

- All rates were generally lower than the Profile.

Table 12 Classification results of 6-years-old class

Age _6 / Classifier	Result of profile data			Result of tweet data		
	Precision	Recall	F1	Precision	Recall	F1
NB	0.82	0.55	0.65	**0.50**	0.11	0.18
LR	**0.91**	0.36	0.51	0.00	0.00	0.00
ANN	0.54	0.78	0.64	0.06	0.17	0.09
GXBoost	0.53	0.62	0.57	0.12	**1.00**	**0.21**
RF	0.46	0.76	0.58	0.00	0.00	0.00
Decision Tree	0.10	**1.00**	0.18	0.00	0.00	0.00
SVM	0.80	0.59	**0.68**	0.00	0.00	0.00

About the result obtained from the classification result of Profile text data regarding 6-year-old children, this is the results of classifying 207 profile data.

- LR was the highest Precision at 91%. In LR, 91% of the predicted positive examples were positive, but there were many incorrect recognitions of positive examples as negative examples (***C***). So, recall and F1 were low.
- SVM had the highest F1 at 68%. SVM has a better balance of Precision and Recall than other classifiers. But it needs improvement.

For analysis of Tweet text data classification results regarding 6-year-old children, this is the results of classifying 56 tweet data.

- All rates were generally lower than the Profile. LR, RF, Decision Tree, and SVM showed the lowest in all rates at 0%.

In the 1575 profile data for analyzing classification results by age group, F1 for age 1 was high, rising to 60% of the total, while F1 for age 4 was low, less than 36% of the total. It is thought that characteristic texts appear more in children around the age of 1 than in children around the age of 4. The largest number of profile data for the 1-year-old class used for all classifiers is 572 items. Compared to that, the profile data of the 4-year-old class is less than 80 items. Consider that the classification result is affected by the amount of data.

There were 953 profile data items for analyzing the classification results by age group, which was lower than profile data overall. Among them, the F1 of 1-year-old children was the highest, and the F1 of 0-year-old children was the lowest. It is thought that characteristic texts appear more in children around the age of 1 than in children around the age of 0. The second largest number of profile data for the 1-year-old class used for all classifiers is 200 items. Compared to that, the profile data of the 0-year-old class is less at 46 items. Consider that the classification result is affected by the amount of data.

5.3.2 Results of Proposed Method Using BERT

The classifier of the Proposed method training data, validation data, and evaluation data are randomly divided, so the results change each time the program is run. Table 13 shows the results obtained after many runs. Table 13 shows the classification results of the created classifier of the proposed method.

Table 13 Results of the proposed method

BERT / Class	Result of profile data			Result of tweet data		
	Precision	Recall	F1	Precision	Recall	F1
Age_0	0.35	0.58	0.44	0.25	0.14	0.18
Age_1	**0.81**	0.55	0.66	0.25	**0.50**	0.33
Age_2	0.69	0.64	**0.67**	0.36	0.26	0.30
Age_3	0.65	0.69	**0.67**	0.24	0.28	0.26
Age_4	0.38	**1.00**	0.55	**0.64**	0.44	**0.52**
Age_5	0.12	0.50	0.20	0.31	0.44	0.36
Age_6	0.48	0.83	0.61	0.00	0.00	0.00

This is the result of classifying 1575 profile data regarding the results obtained from the classification of profile text data for children 0 to 6 years old.

- BERT showed the highest precision of 81% in the 1-year-old class. 81% of the predicted positive examples were positive.
- In F1, the 2-year-old class and the 3-year-old class had the highest rate at 68%. It has a better balance between precision and recall than other classes.

The result of classifying 953 tweet data to analyze the classification results of tweet text data for children aged 0 to 6.

- Results other than the 4-year-old's class were lower than the profile. The 4-year-old class has a better balance between precision and recall than other classes. But it needs improvement.

5.3.3 Last Results by All Classifiers

We have compared using more ML algorithms such as NB, LR, ANN, XGboost, RF, decision tree, and SVM. The classification results for each classifier are shown in Table 14 below. The result is a weighted average of the values by class.

Table 14 The classification results for each classifier

All class	Results of profile data				Result of the tweet data			
Classifier	Precision	Recall	F1	Accuracy	Precision	Recall	F1	Accuracy
NB	0.62	0.54	0.57	0.54	0.28	0.24	0.24	0.24
LR	0.74	0.53	0.47	0.53	0.40	0.20	0.10	0.20
ANN	0.54	0.54	0.54	0.54	0.24	0.24	0.24	0.24
XGBoost	0.62	0.54	0.57	0.54	0.36	0.26	0.28	0.26
RF	**0.79**	0.56	**0.60**	0.56	0.46	0.28	0.32	0.28
Decision Tree	0.61	0.43	0.49	0.43	**0.88**	0.26	**0.36**	0.26
SVM	0.70	**0.57**	0.55	**0.57**	0.35	**0.31**	0.26	**0.31**
BERT	0.70	**0.61**	**0.63**	**0.61**	0.34	**0.31**	0.31	**0.31**

About the result obtained from the classification result of Profile text data,

- The accuracy rate of BERT classification was 61%, which was slightly higher than the 57% of the SVM. The accuracy rate for Decision Tree was low at 43%, but there was not much difference for the others.
- RF and SVM showed stable and high accuracy.
- F1 had the highest RF at 60%. However, it is slightly lower than BERT's 63%.

The results of the BERT classification of profile text data showed overall more stability than the classifier of the Baseline method. BERT knows the language through pre-training, which is thought to have helped improve classification accuracy.

For analysis of Tweet text data classification results,

・ The accuracy rate was generally lower than the accuracy rate of Profile. Here, F1 in the Decision Tree was the highest at 36%.

From these results of Tweet text data, it can be seen that the accuracy rate of the classifier of the Proposed method is slightly higher than that of the classifier of the Baseline method. The results for this tweet text data show that the accuracy rate of the proposed method's classifier is slightly higher than that of the Baseline method's classifier. We found that when the amount of data is small, the accuracy rate of BERT is not very different from SVM.

It seems that the larger the amount of Finetuning data, the more effective BERT's pre-training becomes.

Chapter 6
Experiment and Results of mamari Data

To adjust the data to be more specific to childcare content, we considered using childcare data from the Connehito Inc. Q&A app for moms, "mamari". mamari is an internet service for women used by one in three mothers as of 2021 (*1). It has accumulated data of 4 million searches per month and 1.1 million posts by users from 300,000 households.

6.1 Three Characteristics of mamari Data

6.1.1 Uniqueness of Data

Search behavior and question/answer data from your date of birth are retained. By looking at the date of birth, you can statistically find out what the mother is worried about and what she is looking for.

6.1.2 Data that[1] Reveals the True Feelings of Child-rearing

Households

As it is a service that specializes in pregnancy-hunting, pregnancy, and parenting mothers, it has text data on a variety of consultations, including minor daily concerns that occur at home, education, shopping, and living. This data differs from data obtained through market research or questionnaires and is data that reveals the true feelings of users, which can only be obtained by building a relationship of trust with the service.

---

*1 Calculated from the number of users who are planning to give birth in 2021 on mamari and the number of births from the 「Vital Statistics」 published by the Ministry of Health, Labor and Welfare.

6.1.3 Data Amount

mamari started with about 300,000 new users, and that number has grown to about 300,000 per year. Data about people's lives and worries is collected. It has accumulated data of 4 million searches per month and 1.1 million posts by users from 300,000 households.

6.2 Experimental Settings of mamari Data

In this experiment, we used question data from the Connehito Inc. Q&A app for mothers mamari. We used the part of the question data. The size of the used data was 16.19 MB. mamari's question data containing any of the words shown in Table 1. The classification model uses question data that targets only those aged 0 to 6.

The question data size for ages 0 to 6 is 8344KB and the number of questions is 17365. Table 15 shows the[2] items for each category of question data.

Table 15 Items of each category of mamari data

Category	age_0	age_1	age_2	age_3	age_4	age_5	age_6	Total
mamari data	644	5165	4250	4000	1881	1094	331	17365

6.3 Experiment Details

In the classification experiment, we will classify the question data described in Section 6.4 and compare the performance of BERT and classifiers other than BERT.

The proposed method using BERT classifies the data according to the classifier's description. Classify user question data whose questions are aimed at infants aged 0 to 6. One of the eight parts of the training data is used for validation and the rest for fine-tuning. We checked the classification results for each case. For each classifier output, the results output by the

BERT classifier are normalized probabilities ranging from 0 to 1 for each label that sum to 1. For the classification results, the highest probability of each label for the input is obtained as follows: output.

The classifier of the Baseline method also uses ML algorithms such as NB, LR, ANN, XGboost, RF, decision tree, and SVM for performance comparison. Nonlinear SVM is the mapping of nonlinear data to a space that is linearly separable and linearly separable on a hyperplane. To process the Japanese text of the tweet to be classified, we convert the input text into a vector. When classifying with the SVM module, we first normalized with StandardScaler and then used the default parameters. Nonlinear SVM classifies each label in the discriminative space from "0" to "6" depending on the region to which each label belongs.

We performed 5-fold cross-validation on the train set using the train test split of the question data. We used 80% of the tweets and profiles used in the experiment as training and validation data, and 20% as test data. Next, use stratified 5-fold cross-validation to split the training data into 5 parts such that the ratio of labels in each split is the same as the overall ratio, with 4 of the 5 being the training data and 1 being used as validation data. Train and evaluate the model.

Once evaluation and training are complete, the four-training data and one of the validation data are replaced and the model is trained and evaluated again. By doing this five times, finding the average classification accuracy for the five times, and using that value as the classification accuracy, it becomes possible to perform a robust evaluation that does not depend on data division.

To adjust the parameters of each machine learning method, we perform the above cross-validation for all combinations of pre-specified parameters using grid search to generate a parametric model that shows the best classification accuracy. Finally, we tested how well each generated model could classify the test data.

6.4 Experimental Results

6.4.1 The Classification Results of Baseline Method

We performed 5-fold cross-validation on the train set using a train test split of mamari data (5.3.1).

The classification results for each class from 0 to 6 years old by the Baseline method are shown in Tables 6 to 12 below.

Table 16 Classification results of 0-years-old class

Age_0 / Classifier	mamari data		
	Precision	Recall	F1
NB	0.06	0.11	0.08
LR	0.00	0.00	0.00
ANN	0.09	0.10	0.10
XGBoost	0.09	0.35	0.14
RF	0.00	0.00	0.00
Decision Tree	0.12	**0.36**	**0.19**
SVM	**0.44**	0.09	0.15

About the result obtained from the classification results of mamari data regarding 0-year-old children, this is the results of classifying 644 question data.

・Decision Tree was highest F1 at 19%. Decision Tree has a better balance of Precision and Recall than other classifiers. But it needs improvement.

Table 17 Classification results of 1-years-old class

Age_1 / Classifier	mamari data		
	Precision	Recall	F1
NB	0.42	0.29	0.34
LR	0.30	**1.00**	0.46
ANN	0.48	0.43	0.45
XGBoost	0.71	0.53	0.61
RF	0.81	0.46	0.59
Decision Tree	**0.86**	0.35	0.50
SVM	0.57	0.77	**0.65**

About the result obtained from the classification results of mamari data regarding 1-year-old children, this is the results of classifying 5165 question data.

・ SVM was highest F1 at 65%. SVM has a better balance of Precision and Recall than other classifiers.

Table 18 Classification results of 2-years-old class

Age_2 / Classifier	mamari data		
	Precision	Recall	F1
NB	0.29	0.24	0.26
LR	0.00	0.00	0.00
ANN	0.26	0.31	0.28
XGBoost	0.39	0.37	0.38
RF	0.33	0.39	0.35
Decision Tree	0.14	0.32	0.20
SVM	**0.46**	**0.52**	**0.49**

About the result obtained from the classification results of mamari data regarding 2-year-old children, this is the results of classifying 4250 question data.

・ SVM was highest F1 at 49%. SVM has a better balance of Precision and Recall than other classifiers. But it needs improvement.

Table 19 Classification results of 3-years-old class

Age_3 / Classifier	mamari data		
	Precision	Recall	F1
NB	0.29	0.29	0.29
LR	0.00	0.00	0.00
ANN	0.29	0.29	0.29
GXBoost	**0.47**	0.37	0.41
RF	0.45	0.38	0.41
Decision Tree	0.26	0.39	0.31
SVM	0.40	**0.49**	**0.44**

About the result obtained from the classification results of mamari data regarding 3-year-old children, this is the results of classifying 4000 question data.

- GXBoost was the highest Precision at 47%. In GXBoost, 47% of the predicted positive examples were positive.
- SVM was highest F1 at 44%. SVM has a better balance of Precision and Recall than other classifiers. But it needs improvement.

Table 20 Classification results of 4-years-old class

Age_4 / Classifier	mamari data		
	Precision	Recall	F1
NB	0.14	0.19	0.16
LR	0.00	0.00	0.00
ANN	0.18	0.17	**0.17**
XGBoost	0.08	**0.31**	0.12
RF	0.00	0.00	0.00
Decision Tree	0.00	0.00	0.00
SVM	**0.35**	0.09	0.14

About the result obtained from the classification results of mamari data regarding 4-year-old children, this is the results of classifying 1881 question data.

・ SVM was the highest Precision at 35%. In SVM, 35% of the predicted positive examples were positive. but there were many incorrect recognitions of positive examples as negative examples (***C***). So, recall and F1 were low.

Table 21 Classification results of 5-years-old class

Age_5 / Classifier	mamari data		
	Precision	Recall	F1
NB	0.07	0.11	0.08
LR	0.00	0.00	0.00
ANN	0.07	0.07	0.07
XGBoost	0.06	0.23	**0.09**
RF	0.00	**0.33**	0.01
Decision Tree	0.00	0.00	0.00
SVM	**0.29**	0.03	0.06

About the result obtained from the classification results of mamari data regarding 5-year-old children, this is the results of classifying 1094 question data.

・ SVM was the highest Precision at 29%. In SVM, 29% of the predicted positive examples were positive. but there were many incorrect recognitions of positive examples as negative examples (***C***). So, recall and F1 were low.

Table 22 Classification results of 6-years-old class

Age_6 / Classifier	mamari data		
	Precision	Recall	F1
NB	0.0	**0.09**	**0.05**
LR	0.00	0.00	0.00
ANN	**0.02**	0.02	0.02
GXBoost	0.00	0.00	0.00
RF	0.00	0.00	0.00
Decision Tree	0.00	0.00	0.00
SVM	0.00	0.00	0.00

About the result obtained from the classification results of mamari data regarding 6-year-old children, this is the results of classifying 331 question data.

・ Overall, not classified correctly.

6.4.2 The Classification Results of the Proposed Method for mamari Data

The results of the proposed method are shown in Table 23.

Table 23 The results of the Proposed method for mamari data

Class	Result of mamari data		
	Precision	Recall	F1
Age_0	0.50	0.57	0.53
Age_1	**0.69**	0.75	**0.72**
Age_2	0.59	0.50	0.54
Age_3	0.57	0.48	0.53
Age_4	0.31	0.38	0.35
Age_5	0.25	0.39	0.31
Age_6	0.03	**1.00**	0.06

This is the result of classifying 17365 question data regarding the results obtained from the classification of mamari data for children 0 to 6 years old.

・ BERT showed the highest precision and F1 of 69% and 72% in the 1-year-old class. 69% of the predicted positive examples were positive. It has a better balance between precision and recall than other classes. It is thought that characteristic texts appear more in children around the age of 1 than in children around the other age. The largest number of profile data for the 1-year-old class used for all classifiers is 5165 items.

6.4.3 The Classification Results of All Classifiers for mamari Data.

We also have compared using more ML algorithms such as NB, LR, ANN, XGboost, RF, decision tree, and SVM for mamari data. The classification results for each classifier are shown in Table 24 below. The result is a weighted average of the values by class.

Table 24 The classification results for each classifier for mamari data

All class	Results of mamari data			
Classifier	Precision	Recall	F1	Accuracy
NB	0.28	0.25	0.26	0.25
LR	0.09	0.30	0.14	0.30
ANN	0.31	0.30	0.30	0.30
XGBoost	0.52	0.43	0.47	0.43
RF	0.61	0.42	**0.49**	0.42
Decision Tree	**0.67**	0.35	0.43	0.35
SVM	0.45	**0.48**	0.44	**0.48**
BERT	0.57	**0.55**	**0.56**	**0.55**

About the result obtained from the classification result of mamari data for each classifier,

- The accuracy rate of BERT classification was 55%, which was higher than the 48% of the SVM. The accuracy rate for NB was low at 25%, but there was not much difference for the others.
- Decision Tree and RF showed high precision.
- F1 had the highest RF at 49%. However, it is lower than BERT's 56%.
- The results of the BERT classification of mamari data showed overall more stability than the classifier of the Baseline method. BERT knows the language through pre-training, which is thought to have helped improve classification accuracy.

6.4.4 The Result of Classify Tweet Data Using mamari Data as Training Data by BERT

We used question data specialized in childcare information from the mamari app as training data for classifying Twitter data.

We trained mamari data to classify Twitter tweet data and profile data. The classification results are shown in Table 25 below. The result is a weighted average of the values by class.

Table 25 Result of classifying tweet data using mamari data as training data by BERT

t Train data	Test data	Precision	recall	F1	accuracy
mamari data	Tweet data	**0.47**	0.30	**0.33**	0.30
Tweet data	Tweet data	0.34	**0.31**	0.31	**0.31**
mamari data	Profile data	0.38	0.24	0.25	0.24
Profile data	Profile data	**0.70**	**0.61**	**0.63**	**0.61**

As a result, the characteristics of mamari's data and the tweet body were the same, and the results were not significantly different when learning and classifying mamari's data and classifying the tweet data. mamari's data and Twitter's text are similar, and it is thought that the F1 score for classification has increased somewhat.

On the other hand, since the nature of the data was different from that of the profile data, it seems that it was not useful for classification. when we trained mamari's data and then classified the profile data, the accuracy rate dropped from 61% to 24%. We found that using mamari's training data was effective for tweet data, but not for profile data.

Chapter 7
Conclusions and Future Works

7.1 Conclusions

In this paper, we created a set of tweet texts and a set of profiles according to the developmental stages of infants from "0-year-old child" to "6-year-old child". For each set, we used ML algorithms such as NB (Naive Bayes), LR (Logistic Regression), ANN (Approximate Nearest Neighbor algorithms search), XGboost, RF (Random Forest), decision trees, and SVM (Support Vector Machine) to compare with BERT (Bidirectional Encoder Representations from Transformers), a neural language model, to construct a classification model that predicts numbers from "0" to "6" from sentences.

In this study, we find that the results obtained by classifying the set of tweet texts with BERT are higher than the results obtained by classifying with SVM. The results obtained by classifying the set of profiles with SVM are higher than the results obtained by classifying with BERT.

Furthermore, the Connehito Inc. Q&A app for mother's mamari APP data was classified using the baseline method and the proposed method.

We also used question data specialized in childcare information from the mamari app as training data for Twitter data. We trained mamari data to classify Twitter tweet data and profile data. Finally, we found that mamari's data and Twitter's text are similar, and it is thought that the F1 score for classification has increased somewhat. the use mamari's training data was effective for tweet data, but not for profile data.

7.2 Future Works

The plans are as follows. To improve the accuracy rate, we increase the amount of data and remove noise through data preprocessing. Collects information about infants and young children from Twitter and increases the data size. Data preprocessing refers to denoising text data and removing stop words. In the future, we would like to apply and compare more advanced models such as GPT.

- Try predicting age using a generative model (such as GPT-2).
- Classify Twitter data using mamari + Twitter data as learning data.
- Predict how many children a person will have based on their profile data.
- In recent years, to solve the problem of how to present the obtained language model to users, research on visualizing the learning results of neural network models (research on model explainability) has been actively conducted. We plan to present learning results that incorporate these research results to users.

Acknowledgment

I would like to express my gratitude and greetings to my former supervisor, Professor Kenji Kita, who believed in me, and to Associate Professor Kazuyuki Matsumoto, who provided strong support. I would like to express my gratitude and greetings to Associate Professor Minoru Yoshida, who has devoted his precious time to supervising my research over the past seven years and has provided me with a lot of very useful advice. I dedicate it to my father and mother, whom I haven't seen for eight years, and my younger brother, who always supported my study abroad. Thank you to my little family, my beloved husband Abdulrahman, and the apple of my eye, Abdugheni, Rena, Abdullah, my entire extended family, and my best friends for many years.

This article was written using data provided by the Connehito Inc. Q&A app for mothers, mamari.

References

[1] Maryam Khodabakhsh, Fattane Zarrinkalam, Hossein Fani, Ebrahim Bagheri, Predicting Personal Life Events from Streaming Social Content, the 27th ACM International Conference on Information and Knowledge Management, Torino Italy, Proceedings of NAACL-HLT 2019, pages 4171–4186 (2018).

[2] Koustuv Saha, Jordyn Seybolt, Stephen M Mattingly, Talayeh Aledavood, Chaitanya Konjeti, Gonzalo Martinez, Ted Grover, Gloria Mark, Munmun De Choudhury, What Life Events are Disclosed on social media, How, When, and By Whom?, CHI '21: CHI Conference on Human Factors in Computing Systems, May 8–13, 2021, Yokohama, Japan. ACM, New York, NY, USA, 22 pages. https://doi.org/10.1145/3411764.3445405.

[3] Choudhury M.D, and Massimi M, "She said yes!" Liminality and Engagement Announcements on Twitter, Proc. iConference 2015, Newport Beach, CA. USA, pp.1-13 (2015).

[4] Burke M, and Kraut R, Using Facebook after Losing a Job: Differential Benefits of Strong and Weak Ties, Proc. 2013 Conf. Computer Supported Cooperative Work and Social Computing (CSCW 2013), San Antonio, TX, USA, pp.1419-1430 (2013).

[5] Minoru Yoshida, Kenji Kita, Mining Numbers in Text: A Survey;2021; IntechOpen.
DOI: http://dx.doi.org/10.5772/intechopen.98540.

[6] Anton Bakalov, Ariel Fuxman, Partha Pratim Talukdar, Soumen Chakrabarti, SCAD: collective discovery of attribute values. WWW 2011: 447-456.

[7] Chiranjibi Sitaula, Tej Bahadur Shahi, Multi-channel CNN to classify Nepali COVID-19 related tweets using hybrid features; arXiv:2203.10286v1 [cs.CL] 19 Mar 2022.

[8] Tej Bahadur Shahi, Chiranjibi Sitaula, Natural language processing for Nepali text: a review; Artificial Intelligence Review (2022) 55:3401–3429; https://doi.org/10.1007/s10462-021-10093-1.

[9] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova, Bert: Pre-training of deep bidirectional transformers for language understanding, arXiv preprint arXiv:1810.04805, 2018.

[10] Kohei Yamamoto, Kazutaka Shimada, Acquisition of Periodic Events with Person Attributes, IALP 2020, Kuala Lumpur, Dec 4-6, 2020.
[11] Xikun Zhang, Deepak Ramachandran, Ian Tenney, Yan Elazar, Dan Roth, Do Language Embeddings capture Scales? EMNLP(Findings) 2020: 4889-4896.
[12] Hiroya Takamura (2010), Introduction to Machine Learning for Natural Language Processing (Japanese)
[13] Steven Bird (2009), Ewan Klein, and Edward Loper, Natural language Processing with Python (Japanese translation (2010)).
[14] Breiman Leo (2001), Random Forests. Machine Learning 45 (1): 5–32. doi:10.1023/A:1010933404324.
[15] https://ja.wikipedia.org/wiki/Random Forest.
[16] Cox DR (1958), The regression analysis of binary sequences (with discussion). J Roy Stat Soc B 20: 215–242.
[17] https://datawokagaku.com/xgboost/
[18] https://ja.wikipedia.org/wiki/Gradient Boosting
[19] Piryonesi S. Madeh, El-Diraby Tamer E. (2020-03-01), Data Analytics in Asset Management: Cost-Effective Prediction of the Pavement Condition Index. Journal of Infrastructure Systems 26 (1): 04019036. doi:10.1061/(ASCE)IS.1943-555X.0000512. ISSN 1943-555X.
[20] Hastie T, Tibshirani R, Friedman J. H. (2009), 10. Boosting and Additive Trees. The Elements of Statistical Learning (2nd ed.). New York: Springer. pp. 337 –384. ISBN 978-0-387-84857-0. Original archive as of 2009-11-10.
[21] https://datawokagaku.com/ensemble/#google_vignette
[22] Piryonesi S. Madeh, El-Diraby Tamer E. (2021-02-01), Using Machine Learning to Examine Impact of Type of Performance Indicator on Flexible Pavement Deterioration Modeling. Journal of Infrastructure Systems 27 (2): 04021005. doi:10.1061/(ASCE)IS.1943-555X.0000602. ISSN 1076-0342.
[23] Toshikazu Wada: Advances in approximate nearest neighbor search technology in high-dimensional spaces and its prospects. Journal of Japanese Society for Artificial Intelligence: AI. 25(6) (No. 144) 2010.11, p.761-768. https://ndlsearch.ndl.go.jp/books/R000000004-I10893821.
[24] S. Arya, D. Mount, R. Silverman, and A. Y. Wu: An optimal algorithm for approximate nearest neighbor searching, Journal of the ACM, 45, 6, pp. 891–923 (1998).

[25] Minoru Yoshida, Issei Sato, Hiroshi Nakagawa, Akira Terada: Mining Numbers in Text Using Suffix Arrays and Clustering Based on Dirichlet Process Mixture Models. Information Processing Society of Japan Information Science Basic Research Group (F1) 96th Annual Paper Presentation, November 19, 2009.
[26] Bill Yuchen Lin, Seyeon Lee, Rahul Khanna, Xiang Ren: Birds have four legs?! NumerSense: Probing Numerical Commonsense Knowledge of Pre-Trained Language Models. EMNLP (1) 2020: 6862-6868.
[27] Georgios P. Spithourakis, Sebastian Riedel: Numeracy for Language Models: Evaluating and Improving their Ability to Predict Numbers. ACL (1) 2018: 2104-2115.
[28] T. Forss, S. Liu, and K.-M. Bjork: Extracting people's hobby and interest information from social media content, Terminology and Knowledge Engineering 2014, Berlin, Germany, 2014.
[29] Sunita Sarawagi, Soumen Chakrabarti: Open-domain quantity queries on web tables: annotation, response, and consensus models. KDD 2014: 711-720.
[30] Kawashima Takahide, Satō Tetsuji, Mikado Noriko: Proposal on methods for extracting consumer needs from Twitter. DEIM Forum 2016 B5-1.
[31] R. Kato, K. Nakamura, Y. Yamamoto, S. Tanaka, and K. Sakamoto: Research for reasoning users' attributes and habitual behavior of microblog, IPSJ Journal, vol. 57, no. 5, pp. 1421-1435, 2016.
[32] K. Ikeda, G. Hattori, K. Matsumoto, C. Ono, and T. Higashino: Demographic estimation of Twitter users for marketing analysis, IPSJ Journal Consumer Device & System (CDS), vol. 2, no. 1, pp. 82-93, 2012.
[33] https://en.wikipedia.org/wiki/Natural_language_processing.
[34] https://en.wikipedia.org/wiki/Natural-language_understanding.
[35] Semaan P. (2012): Natural Language Generation: An Overview. Journal of Computer Science & Research (JCSCR)-ISSN, 50-57.
[36] Roman V. Yampolskiy: Turing Test as a Defining Feature of AI-Completeness. In Artificial Intelligence, Evolutionary Computation and Metaheuristics (AIECM) --In the footsteps of Alan Turing. Xin-She Yang (Ed.). pp. 3-17. (Chapter 1). Springer, London. 2013. http://cecs.louisville.edu/ry/TuringTestasaDefiningFeature04270003.pdf

[37] Van Harmelen, Frank, Vladimir Lifschitz, and Bruce Porter eds: Handbook of knowledge representation. Vol. 1. Elsevier, 2008.
[38] Macherey Klaus, Franz Josef Och, and Hermann Ney: Natural language understanding using statistical machine translation. Seventh European Conference on Speech Communication and Technology. 2001.
[39] Hirschman Lynette, and Robert Gaizauskas: Natural language question answering: the view from here. natural language engineering 7.4 (2001): 275-300.
[40] https://en.wikipedia.org/wiki/Document_classification.
[41] Kenji Kita, Kazuyuki Matsumoto, Minoru Yoshida, Masaki Shijibori, Masaki Ohno (Tokushima University): Design technology series - Data Science with Python - Data preprocessing for AI and machine learning [Practical edition], 2021/8/ 31, 978-4-910558-01-1 (Japanese)
[42] https://en.wikipedia.org/wiki/Machine_learning.
[43] Koza John R, Bennett Forrest H, Andre David, Keane Martin A. (1996): Automated Design of Both the Topology and Sizing of Analog Electrical Circuits Using Genetic Programming. Artificial Intelligence in Design '96. Artificial Intelligence in Design '96. Springer, Dordrecht. pp. 151–170. doi:10.1007/978-94-009-0279-4_9. ISBN 978-94-010-6610-5.
[44] What is Machine Learning? IBM. Retrieved 2023-06-27.
[45] Zhou Victor (2019-12-20): Machine Learning for Beginners: An Introduction to Neural Networks. Medium. Archived from the original on 2022-03-09. Retrieved 2021-08-15.
[46] https://en.wikipedia.org/wiki/Supervised_learning.
[47] Mehryar Mohri, Afshin Rostamizadeh, Ameet Talwalkar (2012) Foundations of Machine Learning, MIT Press ISBN 9780262018258。
[48] https://en.wikipedia.org/wiki/Unsupervised_learning.