

---

**Predicting therapy success of blended cognitive behavioral therapy for depression:  
application of machine learning methods**

Thesis submitted in partial fulfilment of  
the degree requirements for the Degree of  
Master of Science

Management & Data Science

Supervisor

Prof. Dr. Burkhardt Funk

Prof. Dr. Dieter Riebesehl

Leuphana Universität Lüneburg

Presented by

*Yongwoo Kim*

September 13, 2018

---

## Note

This document is divided into two parts. The framework paper, which introduces the background knowledge and outline of the research, is placed in front to supplement the research paper. The research paper following the framework paper begins on page 37.

Leuphana Universität Lüneburg  
Management & Data Science  
Yongwoo Kim [3029775]  
Wichernstr. 17  
21335, Lüneburg  
yongwoo.kim@stud.leuphana.de

# Contents - Framework Paper

<b>1</b>	<b>Introduction</b>	<b>4</b>
<b>2</b>	<b>Predictive analytics in E-mental health</b>	<b>5</b>
<b>3</b>	<b>Research questions</b>	<b>6</b>
<b>4</b>	<b>Representation of the research paper</b>	<b>6</b>
4.1	Dataset . . . . .	6
4.2	Research design . . . . .	8
4.3	Feature engineering . . . . .	10
4.4	Feature selection . . . . .	11
4.5	Classifier . . . . .	13
4.5.1	Random forest . . . . .	14
4.5.2	AdaBoost . . . . .	15
4.5.3	Gradient boosting . . . . .	16
4.5.4	KNN . . . . .	17
4.5.5	SVM . . . . .	18
4.5.6	Logistic regression . . . . .	23
4.5.7	Multilayer perceptron classifier . . . . .	25
4.5.8	Gaussian naive Bayes . . . . .	28
4.5.9	Ensemble . . . . .	29
<b>5</b>	<b>Answer to the research questions</b>	<b>30</b>

## List of Figures

1	Evolution of analytics in mental health (from Hahn et al. 2017) . . . . .	5
2	Summary of research design . . . . .	9
3	Examples of ROC curve (from Molina Arias M 2017) . . . . .	9
4	Generating aggregated features using different time windows (from van Breda et al. 2016) . . . . .	10
5	Generating aggregated features using the same time window . . . . .	11
6	Different types of feature selection methods (from Wang et al. 2016b). . . . .	12
7	Feature selection in the study: hybrid method . . . . .	13
8	Example of KNN classification: majority voting (from He et al. 2015). . . . .	17
9	Example of KNN classification: using distance measures for weighted voting . . . . .	18
10	Example of SVM classification: hyperplane of linear SVM (from Upadhyaya and Ramsankaran 2014). . . . .	19
11	Example of SVM classification: Kernel-SVM (from Zararsiz et al. 2012). . . . .	22
12	Comparison of loss functions (from James et al., 2013). . . . .	25
13	Single-layer perceptron (from Verma and Singh 2015). . . . .	26
14	Different activation functions . . . . .	27
15	Multilayer perceptron (from PEREZ-MARIN et al. 2006). . . . .	27

## List of Tables

1	Types of predictive models in e-mental health (from Becker et al. 2018) . . . . .	5
---	---	---

# 1 Introduction

Approximately 17.6% of adults have experienced a common mental disorder within the past 12 months, and 29.2% will have across their lifetime (Steel et al., 2014). Mental disorders are therefore not limited to a small group of individuals but rather affect a large fraction of the public. A larger problem is that mental disorders place a massive burden on individuals, their families, society, and the economy. Mental disorders may have adverse effects on critical developmental transitions and role performances of individuals (Kessler, 2012); these influences are subsequently transmitted throughout society. The direct costs of mental diseases derived from medication, visiting a clinic, or hospitalization are a heavy burden; however, the indirect costs of mental disorders go far beyond direct costs (Trautmann et al., 2016). The global cost of mental illness in 2010 was estimated at US\$ 2.49 trillion. Approximately 1.67 trillion (66%) of the total cost comes from indirect costs and the remainder (33%) from direct costs. High-income countries account for about 65% of the total costs, which is not expected to change over the next 20 years. The total cost is projected to rise to US\$ 6.05 trillion by 2030 (Bloom et al., 2012). Therefore, there is a consensus that the global burden of mental health problems should be more effectively addressed. Considering the high indirect costs arising from mental illness in most countries, additional investment should be made in appropriate and efficient services. The key is to allocate resources for mental health care in a cost-effective way because public budgets are under pressure in many countries (Hewlett and Moran, 2014).

As part of this initiative, some countries have established professional services for the improvement of mild and moderate mental illness in innovative forms. Many computer- and Internet-based programs are currently being used to treat and manage mental illnesses. Much research has been conducted to investigate the cost-effectiveness of online treatment. There are many opinions that e-mental health care will be cost-effective, but there are still dissenters. Therefore, more studies about cost-effectiveness should be conducted, and the effectiveness of online treatment should be increased further.

Although the cost-effectiveness of online intervention has remained controversial, online intervention can be an effective and inexpensive method to treat some mental illness, if it is carefully adjusted. To increase the effectiveness of treatment, the data generated by patients during online treatment can be used. For instance, patients' online activity can be recorded. The data collected during the online treatment provide therapists with a opportunity for better intervention. Therapists can deepen their understanding of the patients' current status and assess the patients' progress based on the data. Another example of valuable data is smartphone usage data, which can be collected during the daily life of patients. Smartphone can help to record and transmit the activity, emotional and the cognitive status of patients in near real time. The data enable therapists to understand patient-specific needs and offer a personalized treatment.

However, the persons concerned still do not know the potential of data. More research is necessary to determine to what extent data can affect treatment improvement. There are many issues that must be clarified, including which data are useful and what type of prediction is possible above a reasonable level. These undiscovered capabilities of data in the domain of mental health are now beginning to be elucidated through predictive modeling.

## 2 Predictive analytics in E-mental health

Analytics in the domain of mental health have evolved courtesy of technology and ample data. In recent years, predictive modeling has become popular. According to Hahn et al. (2017), "Analytics in mental health is moving from the description of patients (hindsight) and the investigation of statistical group differences or associations (insight) toward models capable of predicting current or future characteristics for individual patients (foresight)". Recently, there is hope to gain unprecedented opportunities by using machine learning for predictive analytics, thereby allowing better prevention, detection, diagnosis, and treatment of disease (Fogel and Kvedar, 2018).

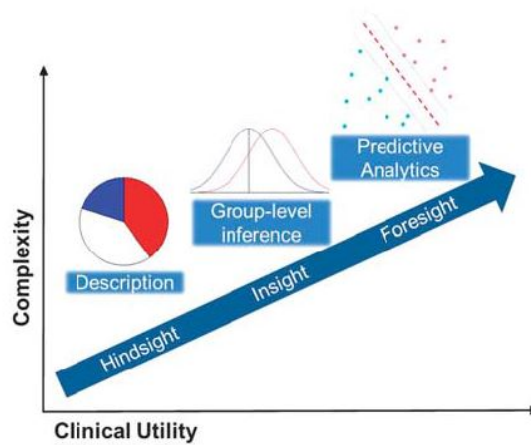


Figure 1. Evolution of analytics in mental health (from Hahn et al. 2017)

While there is growing interest in predictive modeling, Becker et al. (2018) divides predictive models into several types and suggests how the models can have a positive effect on therapy. Table 1 briefly summarizes four types of models.

Type	Predictors	Purpose	Usage
1	Preintervention	Risk assessment and diagnosis	Identify clients at risk for mental health problems Support diagnostic process Support selection of intervention
2	Preintervention and intervention	Predict short-term trends	Track therapeutic progress Identify risk of drop-out Support selection of therapy Adapt intervention to maximize short-term outcomes Facilitate personalized therapy
3	Preintervention and intervention	Predict therapy outcome	Support selection of therapy Adapt intervention to maximize treatment outcomes Facilitate personalized therapy
4	Preintervention, intervention, postintervention	Stabilize results, prevent relapse	Identifying clients with high relapse risk Facilitate personalized aftercare Adapt aftercare to maximize long-term outcomes

Table 1. Types of predictive models in e-mental health (from Becker et al. 2018)

Type 1 models mainly use baseline characteristics of patients including sociodemographic characteristics, personality traits, and illness characteristics to predict risk of mental illness. Type 2 models help to optimize treatment by predicting short-term changes in the health status of pa-

tients. Data collected both during the treatment process and prior to treatment may be used for constructing type 2 models. Type 3 models use data from the intervention and preintervention phases to predict treatment outcomes, which can help to adapt intervention to maximize treatment outcomes. Type 4 models are intended for predicting relapse. These models can be used to screen patients who require special aftercare to prevent relapse. As such, predictive models contribute differently to treatment depending on the characteristics of the model. Therefore, it is necessary to develop various models that can help to achieve various goals.

### **3 Research questions**

In the study, predictive models for predicting therapy outcome are created using the dataset from E-COMPARED project (see section 4.1), which belongs to type 3 according to Table 1. These models aim to classify patients into two groups, improved and nonimproved. Since it is important to determine whether the models contribute to improvement of treatment, research questions that can contribute to the usage of type 3 models are established. The study focuses on the following three questions:

1. How accurately can the therapy outcome be predicted by various machine learning algorithms?

Answering this question can let the people concerned obtain information about the reliability of contemporary predictive models. In addition, if the predictive power of the models is good, it is more likely to be used to assist therapists' decisions.

2. Which kind of data is more important in predicting the therapy outcome?

The answer to this question can show which dataset should be considered first to make better predictive models. Therefore, it can be helpful for researchers who want to make predictive models in the future and eventually help to facilitate personalized therapy.

3. What are the features with strong predictive power?

The answer to this question can affect the people concerned, especially therapists. Therapists can use the most influential features revealed to adjust and improve future treatments.

## **4 Representation of the research paper**

### **4.1 Dataset**

E-COMPARED is the European COMPARative Effectiveness research on Internet-based Depression treatment and evaluates the comparative effectiveness of blended cognitive behavior therapy (bCBT) for adult depression in comparison to treatment as usual (TAU) in 9 countries, consisting of Germany, Poland, Spain, Sweden, UK, Denmark, France, the Netherlands, and Switzerland

(Riper, 2018). The subjects allocated to the bCBT program took the online treatment on a platform called ICT4Depression/Moodbuster. The large number of data gathered from the online platform serve as good material for making predictive models.

The dataset consists of 6 subsets: (1) EMA, (2) Message, (3) Usage(Web), (4) Usage(Mobile), (5) Patients, and (6) PHQ-8. Each subset contains variables that are already known to be academically important for therapy or diagnosis.

First, the EMA subset contains measures of ecological momentary assessments (EMAs) and times of reply. EMA questions are periodically sent to patients via smartphone and e-mail, and answers are collected from the patients. There are seven EMA questions, which check the current state of patients: mood, sleep quality, rumination, self-esteem, enjoyment of activities, social contacts and level of (pleasant) activities. Patients can reply to EMA questions on the same day as requested. The answer has a value from 0 to 12 and can also not be recorded if a patient does not answer the question despite the request. The most frequently asked question among them is the question about the mood of the patient, which appears to be mostly related to depression than other questions. EMA measures can have advantages over traditional methods for measuring the status of patients because it reduces patient recall bias and enables more sensitive assessments by therapists (Moskowitz and Young, 2006).

The Message subset contains information about the recipient, sender, time, and length of a message exchanged via the web. Information about content of the message is not stored. Since many therapists and patients are satisfied and think that web-messaging is useful, web-messaging is perceived as a useful communication method available to therapists and patients (Liederman and Morefield, 2003).

The Usage(Web) subset contains information about the actions patients have made on the web. Patients are encouraged to complete several online sessions and exercises in sequence. The dates, times, and progress of these sessions and exercises are recorded in this subset. A study has already shown that tracking patients' progress can help to improve outcomes (Lambert et al., 2003), and therapists said that they are interested in using this type of data to improve their services (Bickman et al., 2000).

The Usage(App) subset only includes the time when the patient logs in to the mobile application. The variable log-in implies several things. The application called MoodBuster provides the user with motivational feedback to improve treatment adherence, allows therapists to communicate with their patients, and provides the user with a graphical interpretation of their mood levels over time for self-monitoring (Van de Ven et al., 2017). Studies have shown that self-monitoring is an important clinical technique used in cognitive-behavioral therapy (Cohen et al., 2013; Hoyman et al., 2013). Although this subset does not contain various types of variables, it can be a measure of treatment adherence.

The Patients subset contains information about individual patients, such as the patient's nationality, therapist, account creation time, and treatment period. The most notable of these aspects is the nationality. When therapists and patients have different backgrounds, it can be more difficult to establish a therapeutic alliance. Differences in language and communication styles, cultural values and attitudes may lead to misunderstandings and errors in a clinical process (Okasha et al., 2008). Therefore, knowing the background of patients can be beneficial to treatment.



The PHQ-8 subset has the responses of the Patient Health Questionnaire-8 from patients. This questionnaire is composed of 8 items, each item having a value from 0 to 3, so the sum of all items can have a value from 0 to 24. A total PHQ-8 score of  $\geq 10$  is usually considered major depression. The criterion has been already established as a valid diagnostic and severity measure for depression in many clinical studies (Kroenke et al., 2009), which means that PHQ-8 score can be used as a target feature indicating the degree of patients' depression.

Each subset contains information that is considered important for better outcomes or diagnosis. However, some of these information may be less important in the predictive models, or vice versa. Therefore, the importance of the variables in predictive models must also be studied.

## 4.2 Research design

A summary of the entire process is shown in Figure 2. First, patients for whom we cannot measure therapy success due to lack of pretest and posttest PHQ-8 scores or whose therapy duration is too short are excluded. Then, therapy success is defined to categorize patients into two groups, improved and nonimproved. Based on academic grounds, one of several possible classification criteria is employed for classification. In the preprocessing step, imputation for the missing values and feature engineering are performed. Then, the predictors of the dataset are grouped into three subdatasets, composed of Activity, Patient, and EMA, according to the relevant properties. Organizing the subdatasets is a preliminary task to determine which types of data are more important for predicting therapy success. The classification accuracy between the models created using each subdataset will be compared. The patient dataset has been excluded in the tasks because the number of unique patients is not sufficient to reflect the patterns of patient characteristics, such as country-specific patterns. After the subdatasets are organized, variance filtering and the Benjamini-Hochberg procedure are performed eliminate some features of each subdataset. Setting the aggregating time window is also performed using the Benjamini-Hochberg procedure.

Four different classification algorithms (random forest, Adaboost, GradientBoost, and logistic regression) are applied with the reduced feature subset of each subdataset prior to constructing the remaining four algorithms. Feature selection, parameter tuning and model fitting are performed simultaneously using recursive feature elimination in the first four models. The optimal parameter and optimal feature subset are simultaneously found via a mixture of grid search and recursive feature elimination. Models are fitted for every possible combination of parameters and features, and then, parameters and a feature set with the best results are finally selected. The union of the features selected from the first four models is used to make the remaining four models (SVM, KNN, MLP classifier, and Gaussian naive Bayes). Therefore, model fitting and parameter tuning for the remaining four models is performed with the more-reduced feature subset. Training and validation are performed via leave-one-out cross validation (LOOCV) to use the training set exhaustively. LOOCV uses a single observation for validation and the remaining observations for training. This process is repeated for each observation in the training set, such that every observation is used once for validation. This is the exactly the same as a K-fold cross-validation with K being equal to the number of observations. LOOCV is expensive from a computational point of view because of the large number of times the training process is repeated, but it is favored

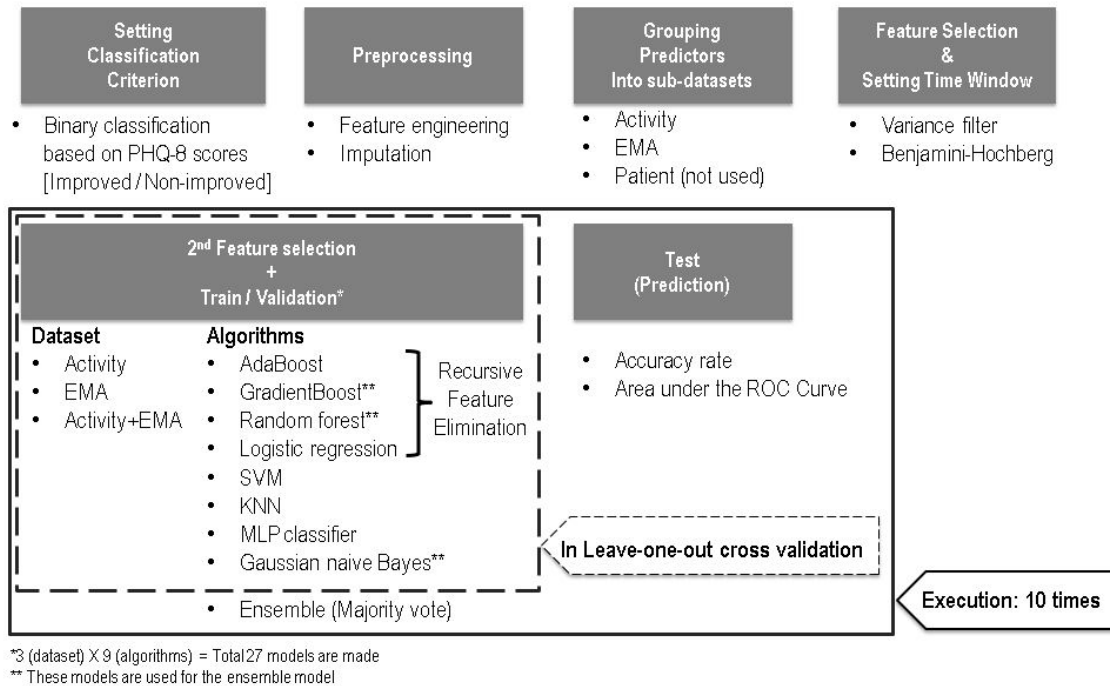


Figure 2. Summary of research design

for a small training set because LOOCV enables one to use the training set thoroughly and obtain a less-biased model (Zhang and Yang, 2015). After fitting all models, an ensemble model that combines multiple classifiers (random forest, Gaussian naive Bayes, and GradientBoost) into one classifier using majority vote is created.

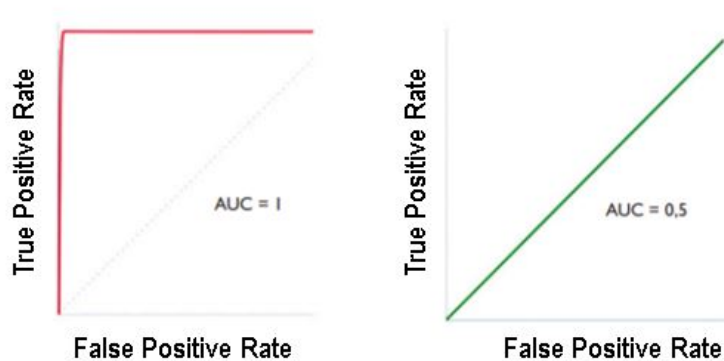


Figure 3. Examples of ROC curve (from Molina Arias M 2017)

After building the models, the performance of the models is evaluated in terms of the accuracy rate and the area under the ROC curve (AUC). The accuracy rate is defined as the percentage of correctly classified instances, which is  $(\text{True Positive} + \text{True Negative}) / \text{Total}$ . Accuracy may not be an appropriate metric for the performance of a classifier when the dataset is unbalanced. However, in the case of the dataset used, the class is fairly balanced, so the accuracy rate can be used for evaluation of the models. The proportion of the classes in the training set is 55:45; this is not a large difference, but it is not exactly fifty-fifty. Therefore, the AUC is used as another evaluation index. The AUC is a very useful metric because it can be used without considering the balance

of classes (Bradley, 1997). The AUC means the area under the ROC curve. The ROC curve is a graph in which the true positive rate is the Y-axis and the false positive rate is the X-axis. Therefore, in case of perfect classification, an ROC curve passes through the top-left corner, and the value of AUC is 1, as shown in Figure 3. When the ROC curve forms the diagonal, the AUC equals 0.5. In this case, the model has no predictive power, which means that the probability of correct classifications is same as random guessing.

Overall, 80% of the total data are used for training and validation, and 20% of the total data are used for testing. Because the size of the dataset is rather small, there may be variation in the results depending on the sampling order. Therefore, training, validation and testing are repeated 10 times with different sampling orders, and the final result is the average of 10 runs.

### 4.3 Feature engineering

Feature engineering is the process of generating new representations of data to create features that make machine learning algorithms work well. Since the performance of machine learning is heavily dependent on the representation of features, much effort is invested in the feature engineering process (Heaton, 2016). New features are created by varying existing features. New features might be sums, differences, or other mathematical transformations of existing features. For example, a market share can be seen as a new feature created with the total size of the market and revenue from that market to indicate market competitiveness. The compound annual growth rate (CAGR) of market share can be considered a new feature comparing the market share values of consecutive year.

Across datasets, there are several common mathematical operations for creating new features. van Breda et al. (2016) have shown an example of how feature engineering for temporal datasets in the domain of e-mental-health. For any existing feature  $a$ , time window size  $k$ , time point  $t$ , and mathematical operation  $O$ , an aggregated feature  $O(a, t, k)$  can be defined. For example,  $Mean(a_2, 5, 3)$  means the aggregated feature is made by averaging the existing feature  $a_2$  with window size  $k = 3$  at time point  $t = 5$ , as shown in Figure 4.

	t=1	t=2	t=3	t=4	t=5	t=6	t=7
Target							
Attribute 1			Std ( $a_1, 5, 2$ )				
Attribute 2		Mean ( $a_2, 5, 3$ )					
Attribute 3			Coef ( $a_3, 5, 2$ )				
Attribute 4	Sum ( $a_4, 5, 4$ )						

Figure 4. Generating aggregated features using different time windows (from van Breda et al. 2016)

The same time window can be used, as in Figure 5, or a different time window can be used, as shown in Figure 4. Additionally, existing features can be aggregated forward based on the time point, as shown in Figure 4, or aggregated backward, as shown in Figure 5.

Applying a different time window condition to each patient or feature may be helpful in enhancing the performances of predictive models, but it may not be helpful in finding generally

	t=1	t=2	t=3	t=4	t=5	t=6	t=7
Target							
Attribute 1	Mean ( $a_1, 1, 3$ )						
Attribute 2	Sum ( $a_2, 1, 3$ )						
Attribute 3	Min ( $a_3, 1, 3$ )						
Attribute 4	Max ( $a_4, 1, 3$ )						

Figure 5. Generating aggregated features using the same time window

influential variables for the treatment. Therefore, the time window condition was fixed for all patients and features in the study. The following conditions were considered for fixing the time window.

- ( $a_{all}, PHQ_{1st}, 7$ ): aggregate 7 days after the first PHQ-8 test for all existing features.
- ( $a_{all}, PHQ_{1st}, 14$ ): aggregate 14 days after the first PHQ-8 test for all existing features.
- ( $a_{all}, PHQ_{1st}, 30$ ): aggregate 30 days after the first PHQ-8 test for all existing features.
- ( $a_{all}, PHQ_{1st}, \text{days between two tests}$ ): aggregate all days between the first PHQ-8 test and the last PHQ-8 test for all existing features. (Same condition but actual time window is different for each patient)

#### 4.4 Feature selection

When constructing and analyzing data in high-dimensional spaces, the problem caused by the high dimensionality of data is a major problem in machine learning; this problem is called "the curse of dimensionality" (Keogh and Mueen, 2010). With many features (with high dimensionality), a learning model is likely to overfit, resulting in a degradation of performance. The dimensionality of data should be reduced to address this problem. Feature selection is considered as an essential step for dimensionality reduction in many machine learning tasks. Through the feature selection procedure, a small subset of the features is selected from the whole feature set according to an evaluation standard. By reducing dimensions through feature selection procedures, machine learning models generally can produce better outcomes, have better model interpretability and incur fewer computational costs (Wang et al., 2016a).

Feature selection methods are broadly divided into three categories: filter method, wrapper method, and embedded method. A brief comparison is depicted in Figure 6. First, the filter method select variables according to a specific measure, such as variance, information gain, similarity, dependency, and distance of the predictors. In other words, a subset of features is decided regardless of the learning algorithm. In contrast, relevant features are selected by the learning algorithm in wrapper methods. Since wrapper methods use the outcomes of a predetermined learning algorithm to assess the suitability of selected features, wrapper methods are usually much more expensive and take more time than filter methods. In contrast, because filter methods evaluate the features independently from the learning algorithm, the features selected through filter methods may cause performance degradation (Saeys et al., 2007). To address these weaknesses of filter and wrapper

methods, embedded methods have been recently proposed. Such methods select the subset with the greatest outcomes, like wrapper methods. The major difference is that embedded models perform feature selection and model fitting simultaneously. In this manner, embedded models can secure both decent accuracy and efficiency in shorter times compared to wrapper methods.

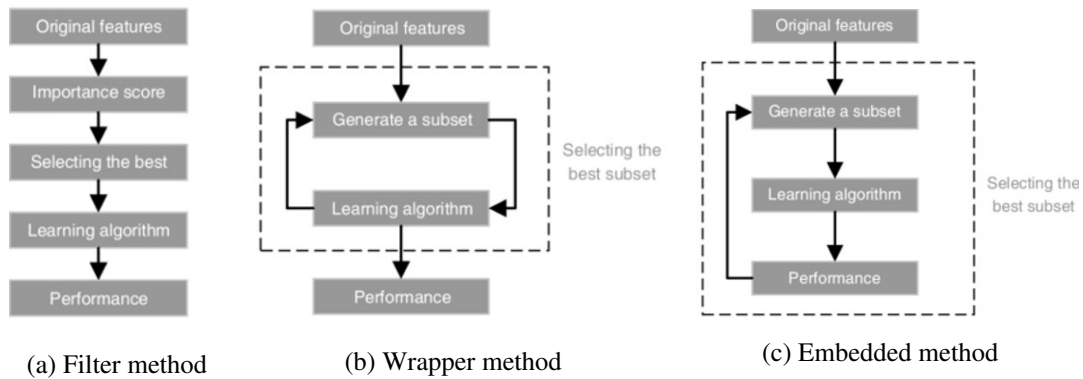


Figure 6. Different types of feature selection methods (from Wang et al. 2016b).

In this study, filter methods and embedded method are sequentially used to select an appropriate subset in a relatively short time. This multiple use of different types of filters is often referred to as hybrid methods. After the feature engineering procedure, variance filtering first eliminates the features that have insufficient variance. This approach considers only the variance of the input features without considering the target features. This primarily aims at eliminating the most meaningless features quickly. The Benjamini-Hochberg procedure then selects only those features that have significant mean differences between the two groups (improved and nonimproved groups). A large number of features should be investigated to determine whether there is a significant difference between the two groups. When we perform a large number of hypothesis tests (e.g. t-test, ANOVA), a few tests may have P-values less than the threshold by chance even if there is no significant differences in all features. Therefore, when performing many hypothesis tests on a large number of features, it is necessary to conduct hypothesis testing in a more conservative manner to reduce false positives. Several methods have been developed to reduce false positives. If strictly false detection is controlled, false discovery can be removed well, but it can be a problem because there are some cases in which it is concluded that there is no difference even if there is actually a difference (Jang, 2013). The researcher should choose the appropriate method depending on the situation and purpose. In the study, the Benjamini-Hochberg procedure, which is not very strict, was used since (1) this is not a final feature selection procedure and (2) the features that have a significant difference must be prevented from being removed before being provided to the subsequent feature selector. In the Benjamini-Hochberg procedure, similar features can be selected redundantly because features are selected without considering the correlation or similarity of features. This aspect is a disadvantage of the Benjamini-Hochberg procedure, but it is not if there is another following feature selector.

At the same time, the aggregating conditions that are presented in Section 4.3 were examined through the Benjamini-Hochberg procedure to determine how well-suited they would be for prediction. When the features were aggregated using all the periods between the first test and the

last test, significant differences were found between the groups in terms of many features. Under other conditions, the number of features with significant differences between the two groups was insufficient. Therefore, we decided to use features generated from the period between  $X_1$  and  $X_f$ .

After the reduced subset of features is generated by passing through two filters, a embedded model using recursive feature elimination is applied to select the best feature subset. Initially starting with all the features of the reduced subset, for every iteration, the worst feature (i.e., the features that have the lowest feature importance in the case of tree-based models and the features that have the lowest coefficient in the case of logistic regression) is eliminated. For every iteration, the scores for the validation sets are calculated via LOOCV. The features left at the iteration that gives the highest score on the validation sets is considered as the best feature set of data. Recursive feature elimination is applied to 4 different models (AdaBoost, random forest, GradientBoost, and logistic regression), and then, the union set of the feature subsets selected by the four models is used to construct the remaining four models (KNN, SVM, MLP classifier, and Gaussian naive Bayes). That is, the first four models function like another feature selector for the remaining four models. Therefore, the remaining four models can also generate models with an appropriate subset of features. By combining the two methods, it is possible to effectively reduce a large number of variables in a relatively short time and obtain the goodness of variable selection.

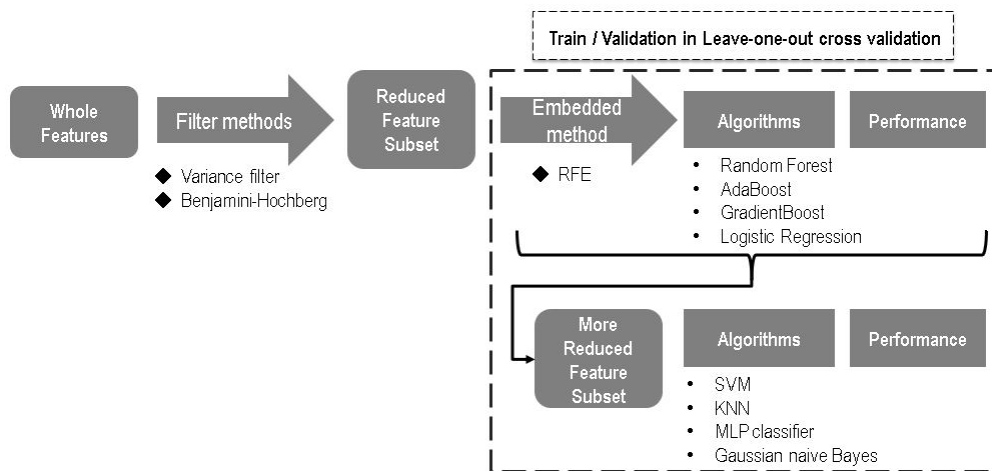


Figure 7. Feature selection in the study: hybrid method

#### 4.5 Classifier

Classification algorithms work differently, so it is important to use many algorithms to improve prediction results and assess feature importance in a comprehensive perspective. Eight different algorithms are applied for the experiment in the Python environment using scikit-learn (Pedregosa et al., 2011). The 8 different algorithms are AdaBoost, gradient boosting, random forest, KNN, SVM, logistic regression, MLP classifier, and Gaussian naive Bayes. AdaBoost, gradient boosting, and random forest are based on decision trees. These classifiers derive from a mixture of multiple decision trees through an ensemble method. K-nearest neighbors (KNN, see e.g. Peterson, 2009) is an intuitive model that considers k-nearest neighbors' voting for classification. A support vector machine (SVM, see, e.g., Cortes and Vapnik, 1995) constructs a hyperplane that has the largest

distance to the nearest data point of each class and classifies elements using the hyperplane. Logistic regression (see, e.g., Kleinbaum and Klein, 2010) is a model to estimate the probability of belonging to specific group using a logistic function. The function gives an S-shaped curve that is restricted between 0 and 1 like a probability. Multilayer perceptron (MLP, see e.g. Popescu et al., 2009) is a feedforward artificial neural network model that maps sets of input data onto a set of appropriate outputs. Gaussian naive Bayes (see e.g. Zhang, 2004) is a classifier based on Bayes' theorem with naive independence assumptions between features.

#### 4.5.1 Random forest

The random forest algorithm uses a decision tree as a basic element that makes up the model. The main disadvantage of a decision tree is that it tends to be overfit for training data. Each decision tree is good predictor, but has a tendency to be overfitted for training data. The random forest algorithm is a method to avoid this problem. A random forest derives from a bunch of different decision trees that are slightly different. If we make many trees that work well and are overfitted in different directions, we can reduce the amount of overfitting by averaging the results of many trees.

To implement this strategy, we must create many decision trees. Each tree should be good at predicting and distinct from other trees. Random forests inject randomness when creating a tree such that the trees can be differently made, as the name suggests. There are two points to make a tree random in a random forest, a random selection of data points used to create trees, and a random selection of features when splitting nodes.

To create trees, a bootstrapping process randomly extracts the same number of data points as an original dataset has (sampling with replacement). Consequentially, the size of each dataset extracted is the same as the original dataset size, but some data points may be missing, and some data points may be duplicated. Then, we create a decision tree with each dataset extracted. However, the method of making a decision tree is slightly different from a normal decision tree algorithm. Rather than trying to find the best model with entire features when making each node, the algorithm randomly selects candidate features and tries to find best cutpoint with only selected candidate features. A user can adjust how many candidate features are selected. Choosing candidate features is repeated for each node, so each node in a tree is split using different features and secures diversity. Due to bootstrapping and random feature selection, the random forest algorithm is excellent at preventing overfitting and reducing the variance of the model.

Finally, each independent tree produces each prediction result, and all predictions of trees are combined to make one prediction via majority voting. The method of combining results after bootstrapping is often called bagging (bootstrap aggregating).

A large number of trees have a positive effect on the results. However, Oshiro et al. (2012) have shown that the performance is closer to the maximum performance when the number of trees is 64 or more and suggests that 64 to 128 trees are appropriate, considering both the computational cost and the results. Therefore, the number of trees in the forest was set to 100 in the study. The maximum depth of the tree also must be set to reduce the running time and prevent overfitting. The model was actually observed to be overfitted when the maximum depth of the tree was high,

so it is defined using grid search.

## 4.5.2 AdaBoost

Another approach to supplement the weakness of a single decision tree is boosting, which is the basic idea of AdaBoost. In AdaBoost, trees are not independent each other, which is different from random forest. AdaBoost make a single strong learner by combining weak learners with adjusted coefficients through iterative learning process. AdaBoost generates several weak learners one by one. In this process, instances that are misclassified by a previous classifier are considered to be more important for a next weak learner, which improves the likelihood of correct classification by a next weak learner.

The steps below explain creation of a strong learner through combination of weak learners. First, weights for each data point  $D_1(i)$  are initialized. Then, the learning process is repeated  $T$  (number of weak learners) times to find proper  $\alpha$  (the weight applied to the strong learner). The learning process starts with finding the best weak learner with the smallest error value. The error value is calculated by the product of  $D_t(i)$ . If the error value of the best weak learner is less than 0.5,  $\alpha_t$  and the weights of each data point for next round  $D_{t+1}$  are updated. the process will repeat till AdaBoost can perfectly classify all data points or  $T$  repetitions are completed. When this running process is complete, one strong learner is created with the weak learners, and the  $\alpha$  values are finally adjusted.

---

**Algorithm: AdaBoost for binary classification** - (from Ferreira and Figueiredo 2012)

---

**H(x):** Strong learner

**h(x):** Weak learner

$\alpha$ : The weight applied to the strong learner

**T:** The number of iteration

Given:  $(x_1, y_1), (x_2, y_2), \dots, (x_m, y_m)$

$$x_i \in X, y_i \in \{+1, -1\}$$

Initialize weights for misclassified instances (normally set to  $D_1(i) = 1/m$ )

For  $t = 1$  to  $T$  :

1. Find  $h_t = \underset{h_j}{\operatorname{argmin}} \epsilon_j = \sum_{i=1}^m D_t(i)[y_i \neq h_j(x_i)]$
2. If  $\epsilon_t \geq 1/2$ , break
3.  $\alpha_t = \frac{1}{2} \log \left( \frac{1-\epsilon_t}{\epsilon_t} \right)$
4.  $D_{t+1}(i) = \frac{D_t(i) \exp(-\alpha_t y_i h_t(x_i))}{Z_t}$   
where  $Z_t$  is normalization factor

The final strong learner is  $H(x) = \operatorname{sign} \left[ \sum_{t=1}^T \alpha_t h_t(x) \right]$

---



Since AdaBoost increases the weight of data points which are classified wrongly, it can be vulnerable to noisy data or outliers (Rätsch et al., 2001). In addition, AdaBoost generates learners sequentially, which means that it cannot be executed in parallel like random forest and takes longer time.

AdaBoost is quite robust to overfitting; however, Schapire (2013) have shown that AdaBoost certainly can overfit when too many weak learners are used for training, which means that the number of weak learners should be limited. Therefore, in the study, the maximum number of weak learners is determined using grid search.

### 4.5.3 Gradient boosting

Gradient boosting also makes many weak learners consecutively and produces a single strong learner similar to AdaBoost. However, the manner in which gradient boosting makes weaker learners differs from AdaBoost. As we have seen, AdaBoost increases the importance of incorrectly categorized instances for each iteration such that a next weaker learner focuses on the errors of the previous step. Unlike AdaBoost, Gradient boosting creates weak learners in a manner such that a next weaker learner trains the remaining error itself. In other words, the intuition behind gradient boosting is to consecutively use the pattern of remaining errors (so-called pseudoresiduals) and improve following weak learners. The following explains how to make a strong learner using gradient boosting.

---

**Algorithm: Gradient boosting for binary classification** - (from Lusa et al. 2017)

---

**H(x)**: Strong learner

**h(x)**: Weak learner

$\gamma$ : The weight applied to the strong learner

**T**: The number of iteration

Given:  $(x_1, y_1), (x_2, y_2), \dots, (x_m, y_m)$

$$x_i \in X, y_i \in \{+1, -1\}$$

Initialize model with constant,  $F_0(x) = \underset{\gamma}{\operatorname{argmin}} \sum_{i=1}^m L(y_i, \gamma)$

For  $t = 1$  to **T** :

1. Calculate pseudoresiduals (gradient) :

$$r_{it} = - \left[ \frac{\partial L(y_i, F(x_i))}{\partial F(x_i)} \right], i = 1, \dots, m$$

2. Fit learner  $h_t$  to calculated pseudoresiduals

3. Compute weights  $\gamma_t$

$$\gamma_t = \underset{\gamma}{\operatorname{argmin}} \sum_{i=1}^m L(y_i, F_{t-1}(x_i) + \gamma h_t(x_i))$$

4.  $F_t(x) = F_{t-1}(x) + \gamma_t h_t(x)$

The final strong learner is  $F_M(x)$

---

In gradient boosting, the weight for the strong learner is calculated independently of each weak learner's error. However, it is derived by gradient descent optimization, which minimizes the error of the strong learner.

Similar to other boosting algorithms, GradientBoost is fairly robust to overfitting. However, too many weaker learners can cause overfitting. Therefore, the maximum number of trees (weak learners) and depth of trees are set using grid search in the study.

#### 4.5.4 KNN

K-nearest neighbors (KNN) is one of the simplest machine learning algorithms. It first projects a training set into multidimensional space and store the location of training set. Then, it classifies the new (test) data through a majority vote from the  $k$ -number of nearest data. For example, in Figure 8, test sample 'X' can be classified into either the +1 class of squares or -1 class of triangles. If  $k = 1$ , it is assigned to the -1 class because the nearest one is a triangle. If  $k = 3$ , it is assigned to the +1 class because there are two circles and one triangle. KNN differs from most other machine learning algorithms in that it does not establish a classification model in advance but rather only stores historical data and performs comparisons when performing tests. This method is called instance-based learning because the results are derived directly from training data stored in memory.

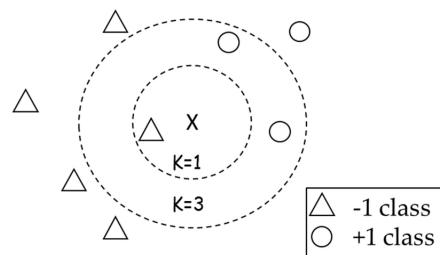


Figure 8. Example of KNN classification: majority voting (from He et al. 2015).

As we have seen, it is very important to select a suitable  $k$  value because it can produce completely different classification results depending on the value of  $k$ . Generally, the larger the value of  $k$  is, the less the effect of noise in classification, but the boundary between classes becomes unclear. Since choosing the best  $k$  value is data-dependent, hyperparameter optimization is usually performed via grid search in a training set, which is also done in the study.

A tied vote can cause a problem in majority voting. In binary classification problems, we can set the  $k$  value to an odd number to easily avoid this problem. However, we can fundamentally solve this problem by using distance measures for weighted voting instead of simple majority voting. It is possible to obtain distance measures from each  $k$ -nearest neighbor to the test point and use its inverse proportion as the weight. When this method is used, the influence of closer neighbors that have a small distance measures increases, and the influence of farther neighbors that have large distance measures decreases, such that classification is possible even if tied voting occurs. In addition, we can solve another problem of majority voting by using weights. In the case of a majority vote, all neighbors have uniform weights, so  $k$ -neighbors contribute equally to

e scaled before running[h!]

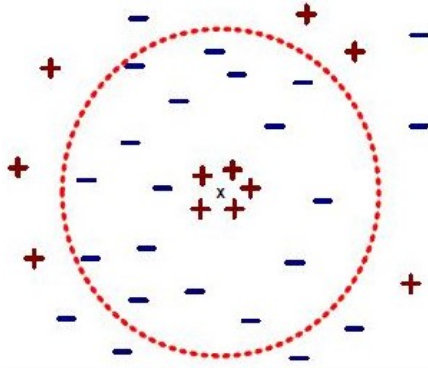


Figure 9. Example of KNN classification: using distance measures for weighted voting

classification regardless of their similarity to the test points. Distance measures can be used to classify test points in sophisticated way, for instance, the test point ‘x’ located in the middle of Figure 9 can be classified as ‘+’ despite the overwhelming number of ‘-’ entities. Because of this merit, distance measure was used instead of uniform weights in the study.

Distance measures can help overcome the disadvantages of majority voting, but KNN is still sensitive to the local structure (distribution) of data. In addition, since KNN is an algorithm that is based on a distance measure, problems may arise when the scales of features are different. Therefore, it is good to equalize the scale of variables before execution and delete unnecessary features before implementing the algorithm. Thus, in the study, the input features were scaled before running KNN.

#### 4.5.5 SVM

SVM also uses distance measures in multidimensional space for classification, similar to KNN. It places data points in  $p$ -dimensional space and classifies the data points using the hyperplane in  $p - 1$ -dimension. For example, Figure 10 shows the simplest SVM, the so-called hard-margin linear SVM. All straight lines classify two classes perfectly. However, it can be said that the red straight line is a better classification boundary than the other straight lines because the two categories are categorized by a wide gap. The two dashed lines that are parallel to the red straight line are called the minus-plane and plus-plane, respectively. The gap between the two dotted lines is called the “margin”. In other words, the purpose of SVM is to find a hyperplane with the largest margin.

This process can be mathematically expressed. If the optimal hyperplane is set to  $w^T x + b = 0$ , vector  $w$  is a normal vector perpendicular to this boundary. Based on this fact, the relationship between the plus-plane ( $x^+$ ) and the minus-plane ( $x^-$ ) can be defined as

$$\begin{aligned} x^+ &: w^T x + b = 1 \\ x^- &: w^T x + b = -1 \\ x^+ &= x^- + \lambda w \end{aligned}$$

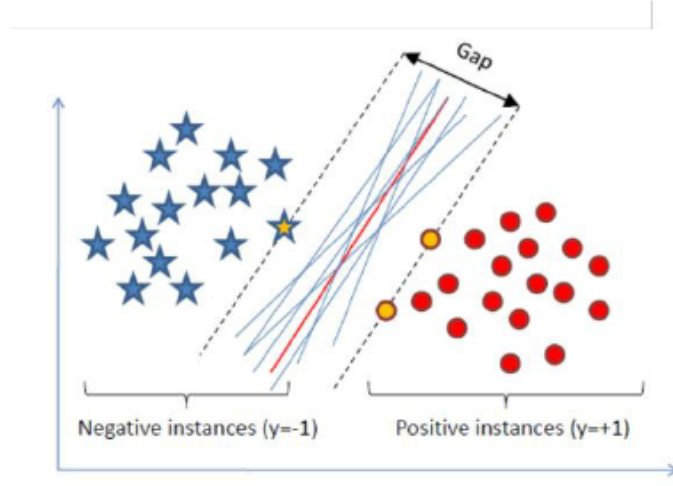


Figure 10. Example of SVM classification: hyperplane of linear SVM (from Upadhyaya and Ramsankaran 2014).

$x^-$  can be regarded as moved from  $x^+$  in parallel to the direction of  $w$ , and the movement width is scaled by  $\lambda$ . Using the above equation, we can derive  $\lambda$  as follows.

$$\begin{aligned}
 w^T x^+ + b &= 1 \\
 w^T (x^- + \lambda w) + b &= 1 \\
 w^T x^- + b + \lambda w^T w &= 1 \\
 -1 + \lambda w^T w &= 1 \\
 \therefore \lambda &= \frac{2}{w^T w}
 \end{aligned}$$

The margin is the distance between plus-plane and minus-plane, which is equal to the distance between  $x^+$  and  $x^-$ . Since we know the relation between the two and the  $\lambda$  value, we can derive the margin as follows.

$$\begin{aligned}
 \text{Margin} &= \text{distance}(x^+, x^-) \\
 &= \|x^+ - x^-\|_2 \\
 &= \|x^- + \lambda w - x^+\|_2 \\
 &= \|\lambda w\|_2 \\
 &= \lambda \sqrt{w^T w} \\
 &= \frac{2}{w^T w} \sqrt{w^T w} \\
 &= \frac{2}{\sqrt{w^T w}} \\
 &= \frac{2}{\|w\|_2}
 \end{aligned}$$

The purpose of SVM is to find a hyperplane that maximizes the margin, so the purpose can be

expressed as follows.

$$\max \frac{2}{\|w\|_2} \rightarrow \min \frac{1}{2} \|w\|_2^2 \quad (1)$$

The observations behind the plus-plane are  $w^T x + b \geq 1$ . Conversely, the points in front of the minus-plane are  $w^T x + b \leq -1$ . The constraint can be expressed as a single expression:

$$y_i(w^T x_i + b) \geq 1 \quad (2)$$

Using the objective (1) and constraints (2) defined above, we use the Lagrange multiplier method to find the optimal hyperplane.

$$\begin{aligned} \min L_p(w, b, \alpha_i) &= \frac{1}{2} \|w\|_2^2 - \sum_{i=1}^n \alpha_i (y_i(w^T x_i + b) - 1) \\ &\alpha_i \geq 0, \quad i = 1, \dots, n \end{aligned}$$

According to the Karush-Kuhn-Tucker condition (Karush, 1939),  $L_p$  has a minimum value at the point at which the partial differential equation is zero.

$$\frac{\partial L(w, b, \alpha_i)}{\partial w} = 0 \quad \rightarrow \quad w = \sum_{i=1}^n \alpha_i y_i x_i \quad (3)$$

$$\frac{\partial L(w, b, \alpha_i)}{\partial b} = 0 \quad \rightarrow \quad \sum_{i=1}^n \alpha_i y_i = 0 \quad (4)$$

Inserting equations (3) and (4) into  $L_p$ , the first term of  $L_p$  is expressed as follows.

$$\begin{aligned} \frac{1}{2} \|w\|_2^2 &= \frac{1}{2} w^T w \\ &= \frac{1}{2} w^T \sum_{j=1}^n \alpha_j y_j x_j \\ &= \frac{1}{2} \sum_{j=1}^n \alpha_j y_j (w^T x_j) \\ &= \frac{1}{2} \sum_{j=1}^n \alpha_j y_j \left( \sum_{i=1}^n \alpha_i y_i x_i^T x_j \right) \\ &= \frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n \alpha_i \alpha_j y_i y_j x_i^T x_j \end{aligned}$$

Inserting equations (3) and (4) into  $L_p$ , the second term of  $L_p$  is expressed as follows.

$$\begin{aligned}
-\sum_{i=1}^n \alpha_i (y_i (w^T x_i + b) - 1) &= -\sum_{i=1}^n \alpha_i y_i (w^T x_i + b) + \sum_{i=1}^n \alpha_i \\
&= -\sum_{i=1}^n \alpha_i y_i w^T x_i - b \sum_{i=1}^n \alpha_i y_i + \sum_{i=1}^n \alpha_i \\
&= -\sum_{i=1}^n \sum_{j=1}^n \alpha_i \alpha_j y_i y_j x_i^T x_j + \sum_{i=1}^n \alpha_i
\end{aligned}$$

Finally, the primal problem  $L_p$  changes to the dual problem, and now the purpose is to find  $\alpha$ .

$$\max L_D(\alpha_i) = \sum_{i=1}^n \alpha_i - \frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n \alpha_i \alpha_j y_i y_j x_i^T x_j$$

Through quadratic programming, values of  $w$  and  $b$  can be found efficiently. The maximum-margin hyperplane is set with the values of  $w$  and  $b$ . When test data arrive, test data are input into  $y_i (w^T x_i + b - 1)$ . If the value is greater than 0, it is classified as '+', and vice versa.

There may not be a hyperplane that accurately divides all data points. In this situation, we cannot solve it in the manner that we have seen. In this case, we use soft-margin SVM (Cortes and Vapnik, 1995) to find a hyperplane that admits the existence of outliers. This method uses a slack variable,  $\xi$ , which indicates the degree of penalties to misclassified data. Due to the slack variable, the optimization is to balance the margin and the penalty for errors. The objective function and constraint are changed as follows.

$$\min \frac{1}{2} \|w\|_2^2 + C \sum_{i=1}^n \xi_i \quad (5)$$

$$y_i (w^T x_i + b) \geq 1 - \xi_i, \quad \xi_i \geq 0 \quad (6)$$

The Lagrangian primal problem constructed using Equations (5) and (6) is as follows:

$$\begin{aligned}
\min L_p(w, b, \xi_i, \alpha_i, \mu_i) &= \frac{1}{2} \|w\|_2^2 + C \sum_{i=1}^n \xi_i - \sum_{i=1}^n \alpha_i (y_i (w^T x_i + b) - 1) - \sum_{i=1}^n \mu_i \xi_i \\
&\alpha_i \geq 0, \quad \mu_i \geq 0, \quad i = 1, \dots, n
\end{aligned}$$

At the point at which the partial differential of  $L_p$  is 0,  $L_p$  has the minimum value according to the Karush-Kuhn-Tucker conditions (Karush, 1939),

$$\begin{aligned}
\frac{\partial L_p}{\partial w} = 0 &\rightarrow w = \sum_{i=1}^n \alpha_i y_i x_i \\
\frac{\partial L_p}{\partial b} = 0 &\rightarrow \sum_{i=1}^n \alpha_i y_i = 0 \\
\frac{\partial L_p}{\partial \xi_i} = 0 &\rightarrow C - \alpha_i - \mu_i = 0
\end{aligned}$$

Putting the above equation into  $L_p$ , the Lagrangian primal problem turns into a dual problem. The dual problem is efficiently solvable using quadratic programming algorithms.

$$\max L_D(\alpha_i) = \sum_{i=1}^n \alpha_i - \frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n \alpha_i \alpha_j y_i y_j x_i^T x_j$$

This is the same as the dual problem of SVM without the slack variable, which we have seen earlier.

In case of hard-margin SVM, there is no parameter that the user must set separately. In case of soft-margin SVM, the user must set parameter  $C$ , which is the strength of penalties. Normally, low  $C$  values enable neglecting outliers, which results in a large minimum margin. Conversely, SVM with large  $C$  values tries to find a more accurate hyperplane. Since the appropriate  $C$  value changes depending on the structure of the data, it is important to conduct tests with various  $C$  values.

Even with soft-margin, there are many cases where it is impossible to accurately classify data with linear hyperplane. Kernel-SVM is a good solution to solve this problem. By introducing a kernel function, data in an input space are mapped to a high-dimensional feature space. Problems that cannot be linearly separated in input space become separable.

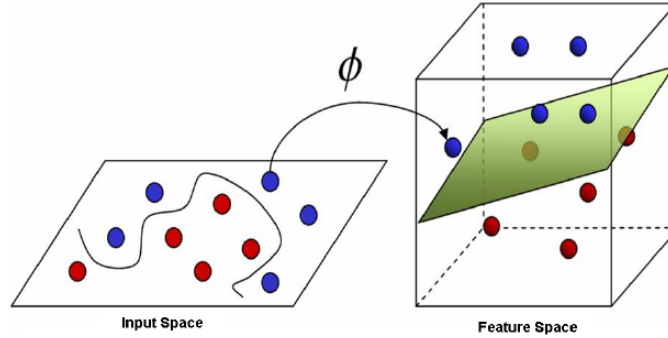


Figure 11. Example of SVM classification: Kernel-SVM (from Zararsiz et al. 2012).

The most commonly used kernel functions are the following:

$$\begin{aligned} \text{linear} & : K(x_1, x_2) = x_1^T x_2 \\ \text{polynomial} & : K(x_1, x_2) = (x_1^T x_2 + c)^d, \quad c > 0 \\ \text{sigmoid} & : K(x_1, x_2) = \tanh \{ a(x_1^T x_2) + b \}, \quad a, b \geq 0 \\ \text{gaussian} & : K(x_1, x_2) = \exp \left\{ -\frac{\|x_1 - x_2\|_2^2}{2\sigma^2} \right\}, \quad \sigma \neq 0 \end{aligned}$$

SVM with a Gaussian kernel is widely used as a reasonable first choice because it exhibits good generalization performance for many real issues and can be tuned easily. However, using a linear kernel can yield better results when a number of attributes is very large since a transformation from a large input dimension to a higher dimension does not improve the separability (Hsu et al., 2003). Namely, depending on the structure of the data, the use of the kernel should be differently decided. In the study, Gaussian kernel was used for classification. The hyperparameters  $C$  and

gamma for SVM were tuned via grid search. SVM is also a distance-based algorithm; therefore, the input features were scaled before running the model.

#### 4.5.6 Logistic regression

Logistic regression is an algorithm for classification. It is similar to linear regression in terms of describing dependent variables as linear combinations of independent variables. However, unlike linear regression, logistic regression is used for categorical dependent variables.

When the dependent variable is binary, the dependent variable can be only two cases, 0 or 1. If simple linear regression is applied, the result is out of the range [0,1]. In the equation of the linear regression, the equation can be partially adjusted such that the dependent variable has a probability between 0 and 1.

$$\begin{aligned} P(Y = 1|X = \vec{x}) &= \beta_0 + \beta_1x_1 + \beta_2x_2 + \dots + \beta_px_p \\ &= \vec{\beta}^T \vec{x} \end{aligned}$$

However, in the above equation, the range of the left-hand side is between 0 and 1. However, the right-hand side can be a positive or negative infinity. Let us change the expression once again by setting the left-hand side equal to the odds ratio.

$$\frac{P(Y = 1|X = \vec{x})}{1 - P(Y = 1|X = \vec{x})} = \vec{\beta}^T \vec{x}$$

However, the ranges of both sides still do not match. The range on the left-hand side (odds ratio) has a range from 0 to positive infinity. However, the right-hand side (regression) has the range from negative infinity to positive infinity. When the log is taken on the left-hand side, the left-hand side then has the range from negative infinity to positive infinity, like the right-hand side.

$$\log\left(\frac{P(Y = 1|X = \vec{x})}{1 - P(Y = 1|X = \vec{x})}\right) = \vec{\beta}^T \vec{x}$$

Now, the range of left-hand side matches the right-hand side. We now have an equation for logistic regression; the next step is to find the coefficients. In linear regression, coefficients can be determined by solving a closed-form equation, but it is impossible in logistic regression. Iterative reweighted least squares (IRLS, see e.g. Burrus, 2012) is usually used to optimize the coefficients and fit the model and gradient decent (see e.g. Bottou, 2010) is also a good way to find coefficients. As the above regression coefficient vector  $\beta$  increases, the probability  $P(Y = 1)$  increases. For example, suppose that the regression coefficient  $\beta_1$  corresponding to  $x_1$ , which is the first element of the input vector  $x$ , is 2.5. If  $x_1$  increases by 1, then the  $\log(\text{odds ratio})$  corresponding to  $Y = 1$  increases by 2.5. If we substitute  $P(Y = 1|X = \vec{x})$  as  $p(x)$  and the right side of the above equation as  $a$ , the above equation becomes

$$\frac{p(x)}{1 - p(x)} = e^a$$



By summing up the above equation, we can get  $P(Y = 1|X = \vec{x})$ . This is called a logistic function (also called a sigmoid function).

$$\begin{aligned} p(x) &= e^a \{1 - p(x)\} \\ &= e^a - e^a p(x) \end{aligned}$$

$$\begin{aligned} p(x)(1 + e^a) &= e^a \\ p(x) &= \frac{e^a}{1 + e^a} = \frac{1}{1 + e^{-a}} \\ \therefore P(Y = 1|X = \vec{x}) &= \frac{1}{1 + e^{-\vec{\beta}^T \vec{x}}} \end{aligned}$$

Putting the test vector  $x$  in the trained model returns the probability of belonging to the category. After comparing the probability of belonging to category 1 with the probability of belonging to 0, we classify the category of the test data into those with higher probability. That is, in the case of the below equation, the test data are classified as  $Y = 1$ .

$$P(Y = 1|X = \vec{x}) > P(Y = 0|X = \vec{x})$$

After replacing the left-hand side of the above equation with  $p(x)$ , we can summarize it as follows:

$$\begin{aligned} p(x) &> 1 - p(x) \\ \frac{p(x)}{1 - p(x)} &> 1 \\ \log \frac{p(x)}{1 - p(x)} &> 0 \\ \therefore \vec{\beta}^T \vec{x} &> 0 \end{aligned}$$

That is, if  $\beta^T x > 0$ , the category of the data is classified as 1. Therefore, the decision boundary of the model is a hyperplane with  $\beta^T x = 0$ . Logistic regression is quite similar to linear SVM in the manner in which logistic regression discovers a hyperplane. However, because there is a difference in the loss function that should be minimized, the two methods do not produce the same result. A comparison of the loss function of SVM and logistic regression is as follows:

$$\begin{aligned} L_{SVM}(X, y, \beta) &= \sum_{i=1}^n \max [0, 1 - y_i (\beta_0 + \beta_1 x_{i1} + \dots + \beta_p x_{ip})] \\ L_{logistic}(X, y, \beta) &= - \sum_{i=1}^n y_i \log (\beta_0 + \beta_1 x_{i1} + \dots + \beta_p x_{ip}) \\ &\quad - \sum_{i=1}^n (1 - y_i) \log (1 - \beta_0 - \beta_1 x_{i1} - \dots - \beta_p x_{ip}) \end{aligned}$$

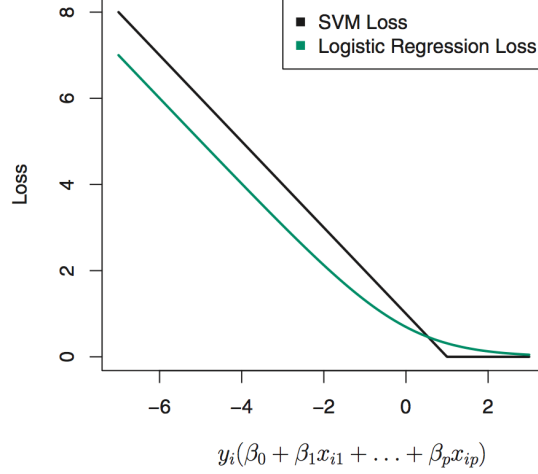


Figure 12. Comparison of loss functions (from James et al., 2013).

The goal of SVM is minimizing hinge loss instead of minimizing logistic loss. More specifically, the loss is 0 when  $y_i(\beta_0 + \beta_1 x_{i1} + \dots + \beta_p x_{ip}) \geq 1$  is satisfied in SVM. However, even in the case of logistic regression, the loss is not completely zero even if  $y_i(\beta_0 + \beta_1 x_{i1} + \dots + \beta_p x_{ip}) \geq 1$  is satisfied. As a result, logistic regression would keep reducing loss towards 0 in high dimensions. Therefore, when a regularization function is not specified, the model can become overfit in high dimensions. Usually, L1 regularized logistic regression is widely used to address this problem (Lee et al., 2006). In the study, since a variable is selected through RFE, a regularizer is not necessary. Nevertheless, a L1 norm is used for more efficient computation, and the regularization strength is tuned via grid search.

Although there is a difference in the loss function, the logistic regression produces a result that is quite similar to that of the linear SVM. However, both algorithms have quite different origins. SVM is geometrically motivated, whereas logistic regression comes from linear regression. Therefore, logistic regression has the advantage that we can interpret the coefficients and evaluate the importance of features. To correctly compare the importance of features, the input features should be scaled before running logistic regression. Therefore, in the study, the input features were scaled before running the model.

#### 4.5.7 Multilayer perceptron classifier

An early perceptron is a linear classifier that can classify only linearly separable cases. The single-layer perceptron has two layers of input and output layers. The input layer receives an input vector  $x = (x_1, x_2, \dots, x_d)^T$ , and the output layer outputs a binary classification result. The input layer has  $d + 1$  nodes and the output layer has one node. One of the input nodes ( $x_0$ ) is the bias, which always has a value of 1. The input and output layers are connected by an edge, and each edge has a weight  $\omega_n$ , which is also called the connection strength.

A simple depiction of the single-layer perceptron is shown in Figure 13. In the input layer, the input vector is received and transmitted to the right direction. In the connection line,  $\omega$  is multiplied with each value of the input vector according to the weight. Finally, in the output layer,

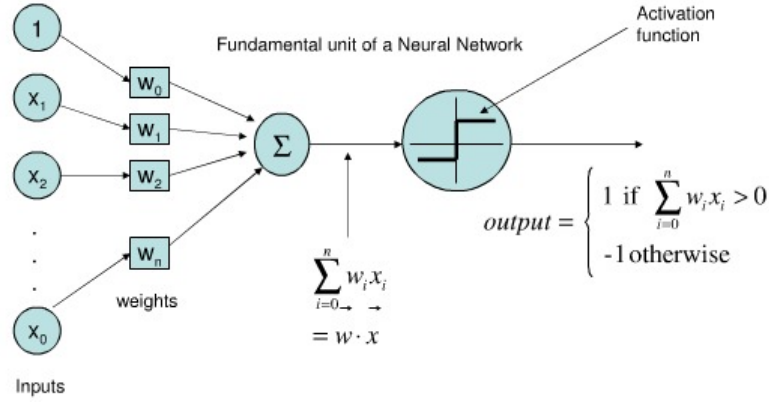


Figure 13. Single-layer perceptron (from Verma and Singh 2015).

the values are summed and passed through to the activation function  $\tau$ . This can be expressed as follows:

$$y = \tau(\sum_{i=0}^n w_i x_i)$$

$$y = +1, \tau(\sum_{i=0}^n w_i x_i) \geq 0$$

$$y = -1, \tau(\sum_{i=0}^n w_i x_i) < 0$$

Given input vectors, proper weights can be found efficiently through gradient decent (see e.g. Bottou, 2010). In the example, a step function is used, but various types of activation functions can be used as an activation function. When we use the logistic function as an activation function instead of the step function, the single perceptron model becomes identical to logistic regression. Widely used activation functions include the following:

$$Sigmoid : \sigma(x) = \frac{1}{1 + e^{-x}}$$

$$HyperbolicTangent : \tanh(x) = 2\sigma(2x) - 1$$

$$= \frac{e^x - e^{-x}}{e^x + e^{-x}}$$

$$ReLU : f(x) = \max(0, x)$$

$$LeakyReLU : f(x) = \max(0.01x, x)$$

The sigmoid function is a logistic function, as explained in section 4.5.6. A hyperbolic tangent (tanh) is a function that rescale and shifts the size and position of a sigmoid function. The hyperbolic tangent has a range of [-1,1] and the shape of a hyperbolic tangent function is symmetric with respect to zero (zero-centered). This means that the slope of the derivative is greater than that of the sigmoid function. Because of this difference, a hyperbolic tangent function has a faster learning convergence than a sigmoid function when used as an activation function. How-

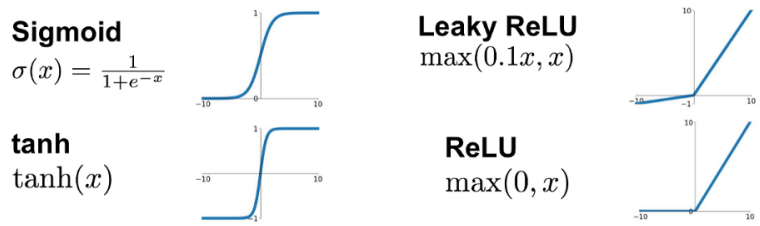


Figure 14. Different activation functions

ever, gradient vanishing problems (see e.g. Hochreiter, 1998) can occur when hyperbolic tangent or sigmoid functions are used as an activation function. By using ReLU function, we can avoid gradient vanishing because the gradient of ReLU function is constant at 1 when  $x > 0$ . For this reason, the rate of learning convergence of ReLU is much faster than the sigmoid or hyperbolic tangent function (Krizhevsky et al., 2012). However, ReLU also has a problem. If  $x$  is negative, the gradient of ReLU becomes 0. When  $x < 0$ , a node can be stuck on the negative side. This result means that the node always outputs 0 and dead, which is called dying ReLU. Once a node becomes negative, it is unlikely for it to recover. Such nodes cannot play any role in discriminating input data over time. The input vectors usually involve multiple data points. As long as not all of observations are negative, we expect that we obtain a positive slope of ReLU. The dying ReLU problem is likely to occur when learning rate is too high or there is a negative bias. We can use the LeakyReLU function to solve the dying ReLU phenomenon. It has the same properties as ReLU except that the gradient is 0.01 when  $x < 0$ .

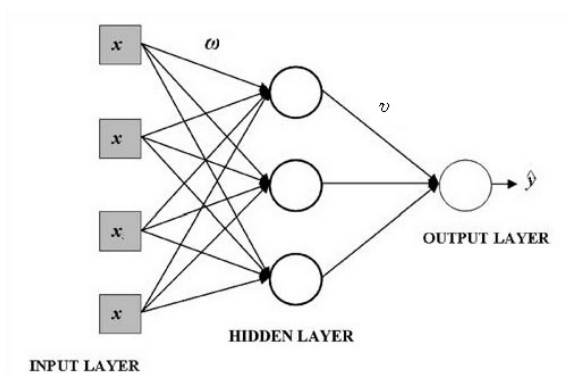


Figure 15. Multilayer perceptron (from PEREZ-MARIN et al. 2006).

Multilayer perceptrons (MLPs) are basically combinations of linear models. A hidden layer is added between the input and output layers. Several hidden layers can be also added. The structure shown in Figure 15 can be expressed as follows:

$$\begin{aligned}
h[0] &= \tanh(w[0,0]x[0] + w[1,0]x[1] + w[2,0]x[2] + w[3,0]x[3] + b[0]) \\
h[1] &= \tanh(w[0,1]x[0] + w[1,1]x[1] + w[2,1]x[2] + w[3,1]x[3] + b[1]) \\
h[2] &= \tanh(w[0,2]x[0] + w[1,2]x[1] + w[2,2]x[2] + w[3,2]x[3] + b[2]) \\
\therefore \hat{y} &= v[0]h[0] + v[1]h[1] + v[2]h[2] + b
\end{aligned}$$

$w$  is the weight between the input  $x$  and the hidden layer, and  $v$  is the weight between the hidden layer  $h$  and the output. The weights  $v$  and  $w$  are learned from the training data,  $x$  is the input characteristic,  $\hat{y}$  is the calculated output, and  $h$  is the output of hidden layer calculation. An important parameter that we must determine in person is the number of units in the hidden layer. For small datasets, the number of units in the hidden layer can be five to ten, but in a very complex dataset, it can be as large as 10,000. Furthermore, we can also add more hidden layers, resulting in a deeper model. When there are multiple layers (there is more than one weight that we have to train), we adjust the weights in a special manner called backpropagation (see, e.g., Sadowski, 2016).

Then, we should decide how many hidden layers are appropriate and how many nodes there should be. Macukow, 2016 said that one hidden layer is sufficient for the large majority of problems and that an additional layer yields the instability of gradient. Two hidden layer are necessary only if the learning process refers the function with points of discontinuity. Thus, trying to design a model with a minimal hidden layer is recommended. With respect to the number of nodes in a hidden layer, we do not have any magic formula to determine a structure that minimizes the error and produces a best result. Using insufficient nodes in the hidden layers will result in underfitting, and using excessive nodes in the hidden layers can cause overfitting. The best choice have must be search by the randomly alternatives according to the data (Maciel and Ballini, 2008). Therefore, in the study, the number of hidden layers is set to 1, and the number of nodes in a hidden layer is decided using grid search. Hyperbolic tangent is used as an activation function because there is only one hidden layer, which means that the model does not suffer from gradient vanishing.

#### 4.5.8 Gaussian naive Bayes

Naive Bayes is a kind of probabilistic model, suggesting a simple manner to solve problems through a conditional probability. For example, if there are three features and each one is binary, there must be at least  $2^3$  data points to explain all cases with a conditional probability. However, the number of features that usually describe data is often much greater. For example, if there are 10 features and each is binary, we require  $2^{10}$  data points to describe all the data properly. That is, the number of data required is exponential to the dimension of the data. In this case, classification using Bayes' rule may be difficult or can be overfitted due to the insufficient amount of data. Thus, naive Bayes uses a new assumption to solve this problem. It assumes that all features are independent and identically distributed, which is commonly abbreviated to *i.i.d.* Naturally, this assumption can be a false assumption, as features are normally related each other and have different distributions. However, if we assume that all features are *i.i.d.*, then the minimum number of data we require is linearly dependent rather than exponentially dependent for the number of

features. A simple assumption greatly reduces complexity of the model.

When the input data is represented by  $x = (x_1, \dots, x_n)$  and the class is represented by  $C_k$ , the target can be expressed using Bayes' theorem as follows:

$$p(C_k|x_1, \dots, x_n) = p(C_k|x) = \frac{p(C_k) \times p(x|C_k)}{p(x)} = \frac{\text{prior} \times \text{likelihood}}{\text{evidence}} = \text{posterior}$$

The prior and likelihood must be calculated to obtain the posterior. However, since this is a joint probability, calculating these probabilities requires much data, as mentioned earlier. However, if we assume that all  $x$  are independent, then we can simply say

$$p(C_k) \times p(x|C_k) = p(C)p(x_1|C)p(x_2|C) \dots = p(C) \prod_{i=1}^n p(x_i|C)$$

$$\therefore \text{posterior} = p(C|x) = \frac{1}{Z} p(C) \prod_{i=1}^n p(x_i|C)$$

$Z = p(\mathbf{x}) = \sum_k p(C_k) p(\mathbf{x} | C_k)$  is a constant scaling factor dependent on  $x_1, \dots, x_n$

Naive Bayes classifier refers to the probabilities of training data. If test data comes, the probabilities of being classified into each class are computed for the test data, and each data point is normally classified into the class that has higher probability (a different decision rule can be applied depending on the user's choice).

When processing data with continuous features, it is typically assumed that continuous features of each class follow a Gaussian distribution. For Gaussian naive Bayes, we divide data by classes and then calculate the mean and variance of  $x$  for each class. Next, the probability distribution of a given class can be calculated by putting the values into the equation for a normal distribution. This aspect is the only difference between Gaussian naive Bayes and naive Bayes with binary features. When the observation value is  $v$ , the probability distribution of  $v$  given a class  $C_k$  is:

$$p(x = v|C_k) = \frac{1}{\sqrt{2\pi\sigma_k^2}} e^{-\frac{(v-\mu_k)^2}{2\sigma_k^2}}$$

$\mu_k$ : the mean of the values in  $x$  associated with class  $C_k$

$\sigma_k^2$ : the variance of the values in  $x$  associated with class  $C_k$

Naive Bayes relies on an independence assumption, which can adversely affect the quality of the results. However, we can still expect good performance even if the independence assumption of variables is somewhat undermined (Zhang, 2004). In addition, it is simple to implement and computationally fast.

#### 4.5.9 Ensemble

"Ensemble learning is the process by which multiple models, such as classifiers or experts, are strategically generated and combined to solve a particular computational intelligence problem"

(Polikar, 2009). The random forest, AdaBoost, and GradientBoost algorithms discussed above are included in ensemble learning because they output a single result using a lot of trees. This method is usually used to enhance the performance of a model or escape from unexpected poor performance. This approach is often used in our human lives, also. For example, asking five doctors for a diagnosis may be more helpful for a better diagnosis than asking just one doctor. In terms of avoiding false diagnosis, the decision of five doctors may be preferable to the decision of one doctor. However, this approach does not always guarantee a better decision. If two doctors are very good and the other three are not, the other three may interfere with the two correct diagnoses. Fortunately, relatively good doctors try to correct the wrong diagnoses of those who do not, so using this method leads to a better diagnosis than the worst diagnosis of a single doctor. This analogy is identical to the situation in which machine learning classifiers are combined through majority voting. In other words, it cannot be guaranteed that the combination of several classifiers always performs better than the best individual classifier. However, the approach certainly reduces the probability of making a poor classification.

The success of an ensemble system depends heavily on the diversity of the classifiers that make up the ensemble model (Polikar, 2009). If all classifiers provide the same output, the mistake will not be corrected. Classifiers that combine weak learners and produce a single result try to increase diversity. For example, random forests makes many different trees with many times of sampling for the sake of diversity. In the cases of AdaBoost and GradientBoost, the weak learners are programmed to reduce previous errors, thus ensuring diversity between weak learners.

In this study, an ensemble model combining the final single classifier is created to attempt to improve the accuracy of the classification results. Among the classifiers, three classifiers that exhibited relatively good performance and work in a different manner are combined into one ensemble model using majority voting. The ensemble model consists of random forest (use bagging), GradientBoost (use boosting), and Gaussian naive Bayes (use Bayes' theorem).

## **5 Answer to the research questions**

The study answered the research questions listed below.

1. How accurately can the therapy outcome be predicted by various machine learning algorithms?

The study answers the above question by presenting the classification results of the eight algorithms and the classification results of the ensemble model combining three different classifiers.

2. Which kinds of data are more important in predicting the therapy outcome?

The kinds of data that have strong predictive power have been determined by comparing models made with only features of EMA, models made with features of activities and a model made with features of both. The accuracy of the models made with each subdataset is given in the research paper.

3. What are the features with strong predictive power?

It is shown how many times each feature is selected for constructing different algorithms. Factors associated with frequently selected variables regardless of the model are considered important in predicting therapy outcomes. The selected number of important features is attached to the appendix of the research paper.



## References

- Becker, D., van Breda, W., Funk, B., Hoogendoorn, M., Ruwaard, J., and Riper, H. (2018). Predictive modeling in e-mental health: A common language framework. *Internet interventions*.
- Bickman, L., Rosof-Williams, J., Salzer, M. S., Summerfelt, W., Noser, K., Wilson, S. J., Karver, M. S., et al. (2000). What information do clinicians value for monitoring adolescent client progress and outcomes? *Professional Psychology: Research and Practice*, 31(1):70.
- Bloom, D. E., Cafiero, E., Jané-Llopis, E., Abrahams-Gessel, S., Bloom, L. R., Fathima, S., Feigl, A. B., Gaziano, T., Hamandi, A., Mowafi, M., et al. (2012). The global economic burden of noncommunicable diseases. Technical report, Program on the Global Demography of Aging.
- Bottou, L. (2010). Large-scale machine learning with stochastic gradient descent. In *Proceedings of COMPSTAT'2010*, pages 177–186. Springer.
- Bradley, A. P. (1997). The use of the area under the roc curve in the evaluation of machine learning algorithms. *Pattern recognition*, 30(7):1145–1159.
- Burrus, C. S. (2012). Iterative reweighted least squares. *OpenStax CNX*. Available online: <http://cnx.org/contents/92b90377-2b34-49e4-b26f-7fe572db78a1>, 12.
- Cohen, J. S., Edmunds, J. M., Brodman, D. M., Benjamin, C. L., and Kendall, P. C. (2013). Using self-monitoring: Implementation of collaborative empiricism in cognitive-behavioral therapy. *Cognitive and Behavioral Practice*, 20(4):419–428.
- Cortes, C. and Vapnik, V. (1995). Support-vector networks. *Machine learning*, 20(3):273–297.
- Ferreira, A. J. and Figueiredo, M. A. (2012). Boosting algorithms: A review of methods, theory, and applications. In *Ensemble machine learning*, pages 35–85. Springer.
- Fogel, A. L. and Kvedar, J. C. (2018). Artificial intelligence powers digital medicine. *npj Digital Medicine*, 1(1):5.
- Hahn, T., Nierenberg, A., and Whitfield-Gabrieli, S. (2017). Predictive analytics in mental health: applications, guidelines, challenges and perspectives. *Molecular psychiatry*, 22(1):37.
- He, M., Weir, J., Wu, T., Silva, A., Zhao, D.-Y., and Qian, W. (2015). K nearest gaussian-a model fusion based framework for imbalanced classification with noisy dataset. *Artificial Intelligence Research*, 4.
- Heaton, J. (2016). An empirical analysis of feature engineering for predictive modeling. In *South-eastCon, 2016*, pages 1–6. IEEE.
- Hewlett, E. and Moran, V. (2014). Making mental health count. *OECD iLibrary*.
- Hochreiter, S. (1998). The vanishing gradient problem during learning recurrent neural nets and problem solutions. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, 6(02):107–116.

- Hoyman, L., Tamas, M., Pacholec, N., Chow, N., Friedberg, R., Poggesi, R., Schmeling, B., and Wetherbee, L. (2013). Self-monitoring in cbt with youth: Everything counts! In *Annual International Conference on Cognitive and Behavioral Psychology*.
- Hsu, C.-W., Chang, C.-C., Lin, C.-J., et al. (2003). A practical guide to support vector classification. *Taipei*.
- James, G., Witten, D., Hastie, T., and Tibshirani, R. (2013). *An introduction to statistical learning*, volume 112. Springer.
- Jang, W. (2013). Multiple testing and its applications in high-dimension. *Journal of the Korean Data and Information Science Society*, 24(5):1063–1076.
- Karush, W. (1939). Minima of functions of several variables with inequalities as side constraints. *M.Sc. Dissertation*.
- Keogh, E. and Mueen, A. (2010). Curse of dimensionality. In Sammut, C. and Webb, G. I., editors, *Encyclopedia of Machine Learning*, pages 257–258, Boston, MA. Springer US.
- Kessler, R. C. (2012). The costs of depression. *Psychiatric Clinics*, 35(1):1–14.
- Kleinbaum, D. G. and Klein, M. (2010). Introduction to logistic regression. In *Logistic regression*, pages 1–39. Springer.
- Krizhevsky, A., Sutskever, I., and E. Hinton, G. (2012). Imagenet classification with deep convolutional neural networks. *Neural Information Processing Systems*, 25.
- Kroenke, K., Strine, T. W., Spitzer, R. L., Williams, J. B., Berry, J. T., and Mokdad, A. H. (2009). The phq-8 as a measure of current depression in the general population. *Journal of affective disorders*, 114(1-3):163–173.
- Lambert, M. J., Whipple, J. L., Hawkins, E. J., Vermeersch, D. A., Nielsen, S. L., and Smart, D. W. (2003). Is it time for clinicians to routinely track patient outcome? a meta-analysis. *Clinical Psychology: Science and Practice*, 10(3):288–301.
- Lee, S.-I., Lee, H., Abbeel, P., and Ng, A. Y. (2006). Efficient  $l_1$  regularized logistic regression. In *AAAI*, volume 6, pages 401–408.
- Liederman, E. M. and Morefield, C. S. (2003). Web messaging: a new tool for patient-physician communication. *Journal of the American Medical Informatics Association*, 10(3):260–270.
- Lusa, L. et al. (2017). Gradient boosting for high-dimensional prediction of rare events. *Computational Statistics & Data Analysis*, 113:19–37.
- Maciel, L. S. and Ballini, R. (2008). Design a neural network for time series financial forecasting: Accuracy and robustness analysis. *Anales do 9º Encontro Brasileiro de Finanças, Sao Pablo, Brazil*.

- Macukow, B. (2016). Neural networks—state of art, brief history, basic models and architecture. In *IFIP International Conference on Computer Information Systems and Industrial Management*, pages 3–14. Springer.
- Molina Arias M, O. S. C. (2017). Pruebas diagnósticas con resultados continuos o politómicos. curvas roc. *Evid Peditr*, 13(1).
- Moskowitz, D. S. and Young, S. N. (2006). Ecological momentary assessment: what it is and why it is a method of the future in clinical psychopharmacology. *Journal of Psychiatry and Neuroscience*, 31(1):13.
- Okasha, A., Arboleda-Florez, J., and Sartorius, N. (2008). *Ethics, culture, and psychiatry: International perspectives*. American Psychiatric Pub.
- Oshiro, T. M., Perez, P. S., and Baranauskas, J. A. (2012). How many trees in a random forest? In *International Workshop on Machine Learning and Data Mining in Pattern Recognition*, pages 154–168. Springer.
- Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., and Duchesnay, E. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.
- PEREZ-MARIN, D., Garrido-Varo, A., Guerrero Ginel, J., and Estrada, J. (2006). Use of artificial neural networks in near-infrared reflectance spectroscopy calibrations for predicting the inclusion percentages of wheat and sunflower meal in compound feedingstuffs. *Applied spectroscopy*, 60:1062–9.
- Peterson, L. E. (2009). K-nearest neighbor. *Scholarpedia*, 4(2):1883.
- Polikar, R. (2009). Ensemble learning. *Scholarpedia*, 4(1):2776. revision #186077.
- Popescu, M.-C., Balas, V. E., Perescu-Popescu, L., and Mastorakis, N. (2009). Multilayer perceptron and neural networks. *WSEAS Transactions on Circuits and Systems*, 8(7):579–588.
- Rätsch, G., Onoda, T., and Müller, K.-R. (2001). Soft margins for adaboost. *Machine learning*, 42(3):287–320.
- Riper (2018). Final report summary - e-compared (european-comparative effectiveness research on online depression). Report, E-COMPARED.
- Sadowski, P. (2016). Notes on backpropagation. homepage: <https://www.ics.uci.edu/~pjsadows/notes.pdf> (online).
- Saeys, Y., Inza, I., and Larrañaga, P. (2007). A review of feature selection techniques in bioinformatics. *bioinformatics*, 23(19):2507–2517.
- Schapire, R. E. (2013). Explaining adaboost. In *Empirical inference*, pages 37–52. Springer.

- Steel, Z., Marnane, C., Iranpour, C., Chey, T., Jackson, J. W., Patel, V., and Silove, D. (2014). The global prevalence of common mental disorders: a systematic review and meta-analysis 1980–2013. *International journal of epidemiology*, 43(2):476–493.
- Trautmann, S., Rehm, J., and Wittchen, H.-U. (2016). The economic costs of mental disorders: Do our societies react appropriately to the burden of mental disorders? *EMBO reports*, page e201642951.
- Upadhyaya, S. and Ramsankaran, R. (2014). Support vector machine (svm) based rain area detection from kalpana-1 satellite data. *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*.
- van Breda, W., Hoogendoorn, M., Eiben, A., Andersson, G., Riper, H., Ruwaard, J., and Vernmark, K. (2016). A feature representation learning method for temporal datasets. In *Computational Intelligence (SSCI), 2016 IEEE Symposium Series on*, pages 1–8. IEEE.
- Van de Ven, P., O’Brien, H., Henriques, R., Klein, M., Msetfi, R., Nelson, J., Rocha, A., Ruwaard, J., O’Sullivan, D., Riper, H., et al. (2017). Ulteamat: A mobile framework for smart ecological momentary assessments and interventions. *Internet Interventions*, 9:74–81.
- Verma, K. and Singh, P. (2015). An insight to soft computing based defect prediction techniques in software. *International Journal of Modern Education and Computer Science*, 7:52–58.
- Wang, S., Tang, J., and Liu, H. (2016a). Feature selection. In Sammut, C. and Webb, G. I., editors, *Encyclopedia of Machine Learning and Data Mining*, pages 1–9, Boston, MA. Springer US.
- Wang, S., Zhang, Y., Zhan, T., Phillips, P., Zhang, Y., Liu, G., Lu, S., and Wu, X. (2016b). Pathological brain detection by artificial intelligence in magnetic resonance imaging scanning (invited review). *Progress in Electromagnetics Research*, 156:105–133.
- Zararsiz, G., Elmali, F., and Ozturk, A. (2012). Bagging support vector machines for leukemia classification. *IJCSI International Journal of Computer Science Issues*, 9:355–358.
- Zhang, H. (2004). The optimality of naive bayes. *AA*, 1(2):3.
- Zhang, Y. and Yang, Y. (2015). Cross-validation for selecting a model selection procedure. *Journal of Econometrics*, 187(1):95–112.

End of framework paper

Start of research paper

# Abstract

Blended CBT treatment, a combination of face-to-face CBT and online CBT, is increasingly being applied in mental health care to obtain benefit from technology. Rich data sources collected from online is opening up new possibilities of better decision-making, cost-effectiveness, and improved outcomes. Exploring complex data and making predictive models can be a valuable asset for accelerating the possibilities and eventually benefit stakeholders, including therapists, policy makers and also patients. This paper aims at constructing decent predictive models that predict treatment success in the domain of depression. The data of patients gathered from the E-COMPARED project, which contains data regarding 201 patients' usage of the system for depression treatment, are used to make predictive models. Feature engineering is used to improve classification accuracy and discover features highly correlated with the target feature. Patients are classified according to the certain criterion by diverse machine learning algorithms. A total of 8 classification algorithms are applied to these data to discover and assess variables that contribute greatly to treatment success. The most successful model obtains an accuracy of 0.7793 and AUC value of 0.7870. It is found that the features related to the EMA responses and cognitive restructuring module are influential. In particular, features related to EMA appear to have particularly strong predictive power in predicting treatment success, given the fact that the models that use only the features from EMA can produce fairly good results. More data is necessary to create more sophisticated models reflecting the patterns of baseline patient characteristics.

# Contents - Research Paper

<b>1</b>	<b>Introduction</b>	<b>41</b>
<b>2</b>	<b>Related work</b>	<b>42</b>
2.1	Predicting short-term mood . . . . .	42
2.2	Predicting treatment outcome . . . . .	43
2.3	Reflection on previous studies . . . . .	43
<b>3</b>	<b>Methods</b>	<b>44</b>
3.1	Data . . . . .	44
3.2	Data selection & Definition of treatment success . . . . .	45
3.3	Preprocessing . . . . .	47
3.4	Feature selection . . . . .	48
3.5	Training models . . . . .	50
3.6	Validation . . . . .	51
<b>4</b>	<b>Results</b>	<b>51</b>
<b>5</b>	<b>Discussion</b>	<b>52</b>
<b>6</b>	<b>Conclusions</b>	<b>54</b>
<b>A</b>	<b>Appendix</b>	<b>58</b>

## List of Figures

1	A brief summary of the dataset. Major activities of participants. . . . .	44
2	Feature engineering . . . . .	48
3	Method overview . . . . .	49
4	Distribution of the most influential features . . . . .	59

## List of Tables

1	Used dataset and examples of features (○: predictor, ●: target) . . . . .	45
2	Different cutoffs for classification . . . . .	46
3	Summary of improvements by different cutoff criteria . . . . .	47
4	Setting condition of aggregating time window . . . . .	50
5	Accuracy rate with 95% confidence intervals. . . . .	51
6	AUC values with 95% confidence intervals. . . . .	52
7	Top 3 important features that were selected by different models . . . . .	52
8	Similar questions between EMA and PHQ-8 . . . . .	54
9	Feature importance ranking in detail: combined dataset . . . . .	58
10	Feature importance ranking in detail: activity dataset . . . . .	58
11	Feature importance ranking in detail: EMA dataset . . . . .	58



## List of Abbreviation

<b>AUC</b> .....	Area Under the ROC Curve
<b>BH step-up</b> .....	Benjamini-Hochberg procedure
<b>CBT</b> .....	Cognitive Behavioral Therapy
<b>CR</b> .....	Cognitive Restructuring
<b>E-COMPARED</b> .	European Comparative Effectiveness Research on Internet-based Depression Treatment
<b>EMA</b> .....	Ecological Momentary Assessment
<b>ILP</b> .....	Inductive Logic Programming
<b>KNN</b> .....	K-Nearest Neighbors algorithm
<b>LOOCV</b> .....	Leave-One-Out Cross-Validation
<b>MLP</b> .....	Multilayer Perceptron
<b>PHQ-8</b> .....	Patient Health Questionnaire-8
<b>RC</b> .....	Reliable Change index
<b>RFE</b> .....	Recursive Feature Elimination
<b>SVM</b> .....	Support Vector Machine

# 1 Introduction

Depression is one of the prevalent mental diseases and causes a significant burden to both individuals and society. (Tylee, 2000; Sobocki et al., 2006). At least 21 million people were affected by depression in 2004, and the number has since increased; approximately 30 million citizens were affected by depression in 2010 (Sobocki et al., 2006; Gustavsson et al., 2011). On an individual level, these people feel depressed in mood or irritable and experience decreased interest or pleasure in most activities. People also experience fatigue or loss of energy, have difficulties in concentration, and experience changes in sleep. In cases of severe depression, people feel worthlessness and plan suicide (Castillo et al., 2007). Because of these symptoms, depression causes not only direct costs for treatment but also indirect social costs. On a society level, depression is one of the costliest diseases. Earlier research has estimated the cost of depression in EU to be more than €90 billion in 2010 (Gustavsson et al., 2011). Moreover, depression is projected to become the second-leading cause of disease burden in the world by the year 2030 (Mathers and Loncar, 2006). Therefore, various efforts have been made not only for efficient treatment but also to reduce the cost of dealing with depression.

Technology has created a new opportunity to psychotherapeutic services offered in person (face-to-face) by mental health professionals. Cognitive behavior therapy (CBT), which is an initial treatment choice for patients with mild-to-moderate depression, anxiety disorders and other mental health problems (Taylor and Chang, 2008), has begun to be provided to patients through the Internet. Until now, many psychotherapists have stated different opinions about the CBT offered through the Internet. Many professionals who oppose Internet-based delivery say that it is difficult to overcome the limitations of distance, which can cause loose working alliance, distraction, technical glitches, and the impossibility of detecting nonverbal cues and body language (Amichai-Hamburger et al., 2014). On the other hands, proponents assert that various disadvantages of e-therapy can be overcome to some extent through various aids. They even argue that online interventions have many other advantages including affordability, time savings, and low costs (Amichai-Hamburger et al., 2014; El Alaoui et al., 2017). Recently, blended CBT treatment, a combination of face-to-face CBT and online CBT, has been increasingly being applied in mental health care to obtain benefits from the advantages these two treatment modalities have (Wentzel et al., 2016). In addition, a plenty of data gathered from an online platform can potentially give us improvements of treatments for patients and achievements for practitioners (AX et al., 2005). For example, the patients data can be useful for therapists to confirm how patients are progressing, and accordingly, patients receive personalized encouragement to obtain better treatment outcomes.

Therefore, the possibility of predicting treatment success of blended CBT using the dataset from the E-COMPARED (see, e.g., Kleiboer et al., 2016), which conducted comparative effectiveness research in effectiveness of blended CBT for depression in comparison with standard care, is explored. Treatment success is defined based on the Patient Health Questionnaire-8 (PHQ-8), which has been widely validated as a screening instrument for depression (Kroenke et al., 2009).

Previous studies have already made some predictive models in the scope of the E-COMPARED project. They mainly focused on short-term prediction of a specific variable, such as mood (e.g., Mikus et al., 2018; van Breda et al., 2016a) or therapy outcomes for predicting treatment success (e.g., Rocha et al., 2018; van Breda et al., 2017). Of course, there have been fine outcomes

in predicting a final treatment success with respect to accuracy of prediction. However, there is still room for improvement. Hence, Feature engineering is used in order to improve the accuracy of prediction and to figure out which variables are the most influential for long-term treatment success. Candidate features are selected through feature selection procedures. Then, several classification algorithms are used to achieve the goals because classification algorithms work in a different manner. In this manner, predictions can be improved and variables can be evaluated from a wide perspective. Finally, an ensemble model that classifies patients is constructed by taking a vote of many classifiers. The possibility of improving outcomes through ensemble methods is explored.

In the following sections, related works are introduced. Then the dataset is described and experiments are explained. Finally, the result is proposed and the outcomes and limitations are discussed.

## **2 Related work**

Various predictive models have been developed. The closely related works can be classified into 2 types. First, many researchers have performed prediction of short-term changes of the patients' current status such as mood. The successful prediction of the patients' status helps the following interventions to be appropriately planned. Moreover, the short-term prediction can eventually benefit treatment outcomes because the prediction helps identify at-risk patients and provide appropriate feedback to patients, which may positively affect treatment adherence and outcomes (Lambert, 2010). Second, some researchers are more interested in predicting the treatment outcomes. They build predictive models using the observations collected during interventions to predict treatment outcome. This type of model is also useful for a subsequent treatment planning because these models allow practitioners to identify variables that positively contribute to treatment success for the whole intervention period.

### **2.1 Predicting short-term mood**

Becker et al. (2016) utilized smart-phone usage data of Dutch students to predict the mood level of the next day. The data include application usage, activity levels, history of phone calls and SMS, and screen-on frequency (see Asselbergs et al., 2016 for detail). The authors applied various methods such as linear regression, SVR, lasso regression, and Bayesian hierarchical linear regression. The authors conducted non-user-level analysis, which does not consider user-specific context, as well as user-level analysis, which considers individuals. They obtained a RMSE of 0.53 from the user-level analysis using the lasso regression, which is slightly better than their baseline model. Nevertheless, they indicated the necessity of personalized level analysis and revealed important variables for predicting mood level.

van Breda et al. (2016b) applied machine learning techniques to predict the mood of the next day. They exploited the same dataset that Becker et al. (2016) used. But the authors conducted feature engineering, which derives new features from the existing features, and used history of mood levels to predict the future mood levels. They conducted a time series analysis, including autoregressive integrated moving average and dynamic time wrapping. The authors also built

different machine learning models using random forest and SVR. They compared the results of time series analysis and machine learning techniques and consequently demonstrated that using machine learning techniques may result in better performance than time series analysis. Based on the results of time series analysis, the authors mentioned that mood generally does not appear to have intrinsic statistical properties that can explain the dynamics of mood to a large degree. Finally, they highlighted the necessity for discovering relevant features that are highly correlated with the dynamics of the target feature.

van Breda et al. (2016a) tried to predict mood for the next day using EMA data of the last few days. They conducted feature engineering procedure and also included an additional procedure for optimizing the number of days taken into account in their features to predict the mood level of the next day. The authors demonstrated that predictive accuracy can be increased by optimizing the time window for temporal attributes.

Mikus et al. (2018) applied recurrent neural networks to a dataset gathered from an online treatment platform (see section 3). They aimed at predicting the mood of the next day based on the measurements of previous days. Crucially, the authors showed that the history of mood rating is a significant input to predict short-term mood level. This finding is inconsistent with a statement in van Breda et al. (2016b). The discrepancy may have been caused by differences in the datasets used and differences in the methods applied. Therefore, the importance of EMA mood rating must be discovered further in future studies with diverse methods.

## **2.2 Predicting treatment outcome**

van Breda et al. (2017) investigated possibilities of predicting therapy success using a dataset including demographic information, ongoing treatment information, various questionnaires used to measure psychiatric disorders, and EMA. They applied random forest, K-nearest neighbors, and a generalized linear model that uses likelihood-based boosting to classify patients into two groups. The models did not achieve high sensitivity and specificity at the same time, but most of the successful patients were correctly classified (84.62%). The authors addressed the need for more data by mentioning the existence of country-specific patterns. They also found that EMA data have predictive power in predicting treatment outcomes.

Rocha et al. (2018) applied inductive logic programming (ILP) to a dataset containing anonymous data from the online platform (see section 3). They classified patients into two groups using ILP models with different combinations of subdatasets and obtained an accuracy rate of approximately 60% from the models. A different usage pattern of patients by countries was found, which is also discovered by van Breda et al. (2017). The authors suggested a possible research direction, which is to include only the most promising features in a model.

## **2.3 Reflection on previous studies**

The implications of previous studies are reflected. Feature engineering to discover relevant features highly correlated with the target feature and user-level analysis is conducted (Becker et al., 2016; van Breda et al., 2016b). The features created through feature engineering are selected so that only the optimal features are selected through selection methods (Rocha et al., 2018). Simultaneously, the proper time windows which appropriately reflect the dynamics of features

is considered (van Breda et al., 2016a). In addition, different algorithms are used because the importance of features can be varied by the algorithms applied (van Breda et al., 2016b; Mikus et al., 2018). Country-specific variable is excluded because it seems that country-specific variables are not helpful to identify common influential features in the small dataset we have (van Breda et al., 2017).

### 3 Methods

#### 3.1 Data

For the experiment, anonymous log-data of 201 patients who used the online treatment platform called ICT4Depression/MoodBuster was used. During the treatment, patients log in regularly to the treatment platform over a specified period to access, read and learn online materials organized into a series of modules. There are also exercises that the patients are expected to perform before going to the next module. They periodically respond to computer-administered questionnaires relevant to their existing problems, which allows a therapist to observe progress, status and responses. The patient and therapist can also send and receive messages through the platform. To use some of these features, the patient can log in to the mobile app. The brief statistics on the behaviors of participants are depicted in Figure 1.

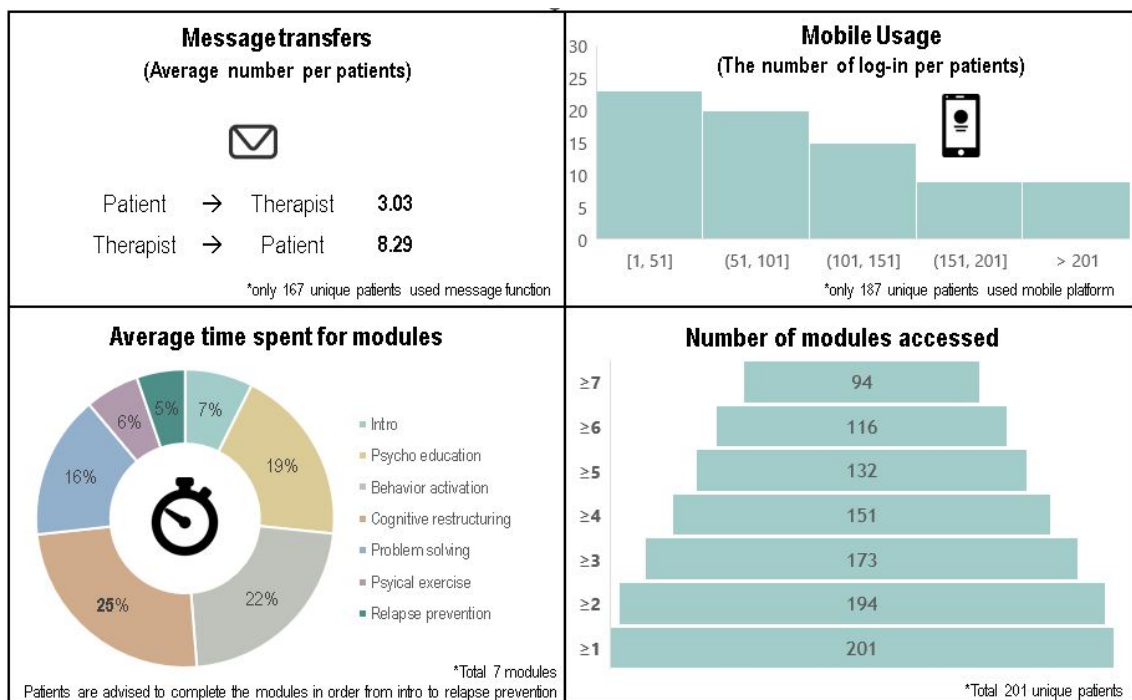


Figure 1. A brief summary of the dataset. Major activities of participants.

Ecological momentary assessment (EMA, see e.g. Shiffman et al., 2008), messages, web usage, mobile usage, and PHQ-8 test scores were collected through the system which is called ICT4D/MoodBuster. A detailed description of these data is provided in Table 1. The patient dataset has not been used as a predictor because the number of unique patients is not sufficiently

large to reflect a country-specific pattern or therapist-specific pattern.

Although patients selected in five countries (DE, FR, NL, PL, UK) had differences in treatment durations, treatment proceeded according to a common treatment protocol, which makes the dataset comparable (Kleiboer et al., 2016).

Dataset	Contents	Description	Features
EMA <sup>○</sup>	EMA responses	Repeated record of subjects' current experiences with minimum delays to minimize recall bias, maximize ecological validity	Mood Sleep Activity Worry Self-esteem
Activity <sup>○</sup>	Message	Data about messages between patient and therapist	Msg. Length Title Length Time received
	Usage(Web)	Usage data of participants (WEB)	Time spent online for modules Number of pages visited Exercise complete Modules complete
	Usage(App)	Usage data of participants (Mobile app.)	Mobile log-in time
Patient	Patient info.	Baseline characteristics about patients	Connected therapist Country
PHQ-8 <sup>●</sup>	PHQ-8 scores	8-question instrument given to patients in a primary care setting to screen for the presence and severity of depression	PHQ-8 Score

Table 1. Used dataset and examples of features (○: predictor, ●: target)

### 3.2 Data selection & Definition of treatment success

It seems that the number of subjects may not be sufficient to implement machine learning techniques. The number of subjects enrolled is 201, but only subjects who have greater than or equal to two valid PHQ-8 scores and at least one EMA response remain for the analysis. Furthermore, a patient who has adequate PHQ-8 scores is excluded if the interval of the first test and the last test is equal to or less than 7 days. For patients with a short treatment period, features that are summarized in trends through feature engineering (see section 3.3) may not be represented correctly. This filtering procedure resulted in a final dataset containing data for 144 unique patients. Because of the small number of patients, There is much attention towards the possibility of class imbalance. Under circumstance of class imbalance, undersampling methods have the potential of eliminating valuable samples from the training set, but oversampling can make a classifier to overfit to the training set (Wasikowski, 2009). These issues often represent a difficult problem in classification when we have insufficient data volume (Kondratovich et al., 2013). Thus, several cutoff conditions reported in Table 2 are reviewed to examine class imbalance and secure an academic basis.

First, a rule-of-thumb approach would be to consider a PHQ-8 score that declined 50% from the first test score as a successful treatment. It is often used by practitioners, but it is more desirable to select criterion that are more academically grounded. Jacobson and Truax (1991) suggested the method for defining a meaningful change in psychotherapy research. Through the method, a patient gets reliable change index (RC) based on one's score differences. This method can be taken

Criteria	Condition	Remarks
50% improvement	$X_f \leq X_1$	Rule of thumb
RC	$X_f - X_1 \leq -5$	Jacobson and Truax (1991)
Standard	$X_f - X_1 \leq -1$ $X_f < 10$	Kroenke et al. (2009)
<b>Clinically improved</b>	$X_f - X_1 \leq -5$ $X_f < 10$	Jacobson and Truax (1991) Kroenke et al. (2009) Rocha et al. (2018) Balanced class

Table 2. Different cutoffs for classification

advantage of and a criterion is derived by applying the method. According to this method, a RC is computed as follows.

$$\begin{aligned}
 RC &= \frac{X_f - X_i}{S_{diff}} \\
 &= \frac{X_f - X_i}{\sqrt{2 \times (S_{tot} \sqrt{1 - r_{xx}})^2}}
 \end{aligned}$$

$X_f$ : The latest PHQ-8 score

$X_1$ : The first PHQ-8 score

$r_{xx}$ : Test-retest reliability of PHQ-8 test

$S_{tot}$ : Standard deviation of subjects

$S_{diff}$ : The spread of the distribution of change scores that would be expected if no actual change had occurred.

By setting  $r_{xx} = 0.84$  (Kroenke MD et al., 2001),  $S_{diff}$  is calculated to be 2.55 with the dataset we have. RC is set as  $RC = 1.96$  (because p-value for RC is set as  $p = 0.05$ ) and statistically significant cutoff is determined. Since the value found is 4.998, a score difference greater than or equal to 5 is considered to be a significant change, which is also written as RC in Table 2.

$$\begin{aligned}
 \frac{X_f - X_i}{2.55} &= 1.96 \\
 \therefore X_f - X_i &= 4.998
 \end{aligned}$$

Another condition that we can consider is mainly not using a score difference between the two tests but rather the last PHQ-8 score itself. PHQ-8 scores of 5, 10, 15, and 20 represent mild, moderate, moderately severe, and severe depression, respectively. In particular, the PHQ-8 score of 10 or greater is commonly considered depression, which is proven that there is no significant difference from the diagnosis by the DSM-IV based diagnostic algorithm (Kroenke et al., 2009). Patients with the final PHQ-8 score of less than 10 and the improved final test result compared to the first test may be classified as an improved group, as denoted as Standard. However, the error inherent to the PHQ-8 test should be also considered. For example, if a patient's score decreased by only 2 points from 11 to 9, it would be difficult to classify the patient into the improved group in

terms of test reliability. Therefore, we can combine two conditions what we derive. Patients who have a reliable score changes between the first and last test, with no further moderate or severe depression, are classified into the improved group under the *Clinically improved* condition.

Criteria	Class	Patients from DE	Patients from FR	Patients from NL	Patients from PL	Patients from UK	Total
50% improvement	nonimproved	42	12	13	11	11	89 (62%)
	improved	32	6	4	8	5	55 (38%)
RC	nonimproved	26	11	9	7	7	60 (42%)
	improved	48	7	8	12	9	84 (58%)
Standard	nonimproved	28	8	8	6	9	59 (41%)
	improved	46	10	9	13	7	85 (59%)
<b>Clinically improved</b>	nonimproved	34	11	12	10	11	78 (54%)
	improved	40	7	5	9	5	66 (46%)
TOTAL		74	18	17	19	16	144 (100%)

Table 3. Summary of improvements by different cutoff criteria

Treatment success is finally defined with the condition Clinically improved, considering class imbalance, academic background, and comparability. The criterion shows a well-balanced class, as reported in Table 3. Because this criterion combines the method of finding reliable change index considering the distribution of score differences and the diagnostic function of the PHQ-8 questionnaire itself, using this criterion appears to be academically sounder than using other criteria. Since the same criterion is used in Rocha et al., 2018, the classification results can be compared. This criterion is used to classify patients into two groups, improved and nonimproved groups.

### 3.3 Preprocessing

In the most datasets we have, we do not have to replace any missing values. For example, missing values in the activity dataset indicate that the patient did not perform a specific action on the day. However, in the case of the EMA dataset, the valid validation is propagated back to the next valid validation. This propagation is because the absence of the EMA value does not mean that the patient’s condition does not exist but rather only that the patient did not respond to the question about the patient’s current condition.

Feature engineering is performed to find optimal representations of the data. The modified representation can better represent the relationship between the target and the predictors and consequently improve the prediction accuracy (Bengio et al., 2013). The existing features are transformed into the sum, mean, minimum, maximum, trend, and standard deviation. First, each behavior item of a user is summed by day. A binary feature which means not an amount of activities but an existence of activities is also generated by day. Then, each summed/binary behavior of a day is summarized and transformed according to the measures. In Figure 2a, there is a simple depiction of how feature engineering was performed. For example, on day 40, the user logged in once and sent 300 words to a therapist and spent 4 hours on a module; these activities are summed on a daily basis in each variable indicating the amount of each activity. At the same time, each binary variables indicating the presence or absence of log-in, sending messages, and taking modules are



made and set to 1. Exceptionally, the EMA scores were averaged on a daily basis because the number of daily responses is different by users. This process is performed for all days between the first test and the last test, and daily summarized values are transformed into the sum, mean, minimum, maximum, trend, and standard deviation of the values over a set period of days.

In addition, other features were made, manually taking into account structure of the dataset according to the implications of previous researches (van Breda et al., 2016b; Becker et al., 2016). Examples include the proportion of taking a specific module out of all modules, the proportion of days when any exercise was performed, and the number of modules completed out of all 7 modules. Through feature engineering, a total 296 features was made. The features made through the foregoing procedure can represent underlying problems to the predictive models better than raw data, resulting in improved model accuracy because machine learning models benefit from a different set of synthesized features (Heaton, 2016).

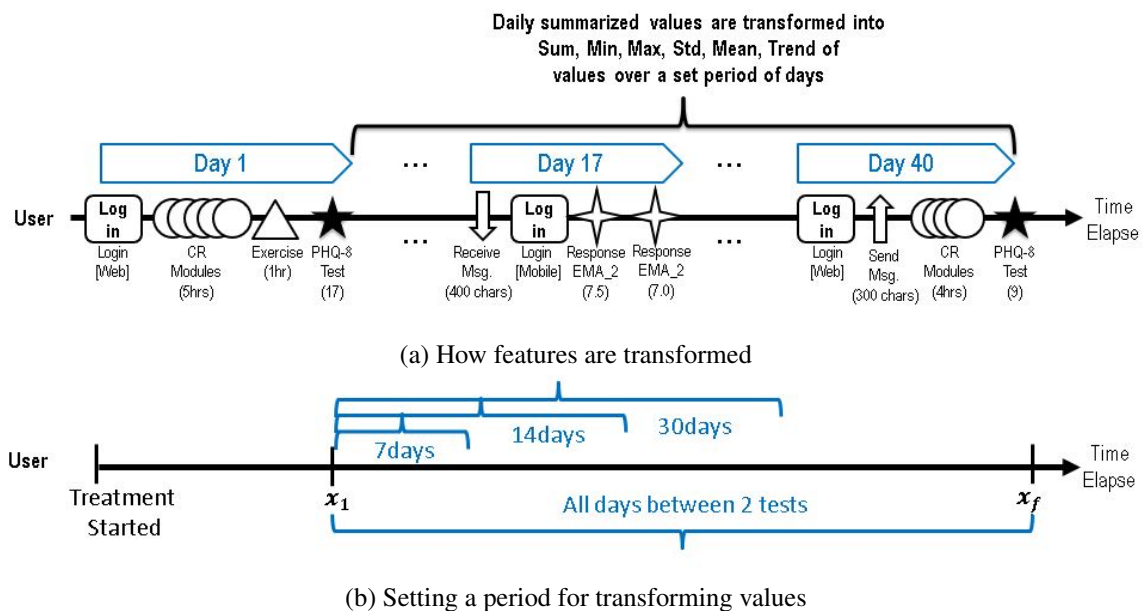


Figure 2. Feature engineering

### 3.4 Feature selection

Feature selection is a crucial component in any machine learning workflow. When presented data in high dimensions, models usually are tuned up to the best status because models have potential risk of overfitting as the number of features increase. The training time of models also increases exponentially with the number of features (Srivastava et al., 2014).

Two feature selection procedures were applied. Only filter methods were considered for feature selection without considering any wrapper method. Filter methods do not exclude correlated predictors but saves all the features that have significant differences between the two groups. Since variables will be selected adaptively in each model at the time of model training (see section 3.5), a large number of features that have a significant difference are rather fine, even though the features are correlated each other. Low-variance features whose variance did not meet a threshold (0) were first excluded. In other words, the features with zero variance were eliminated. Next, the

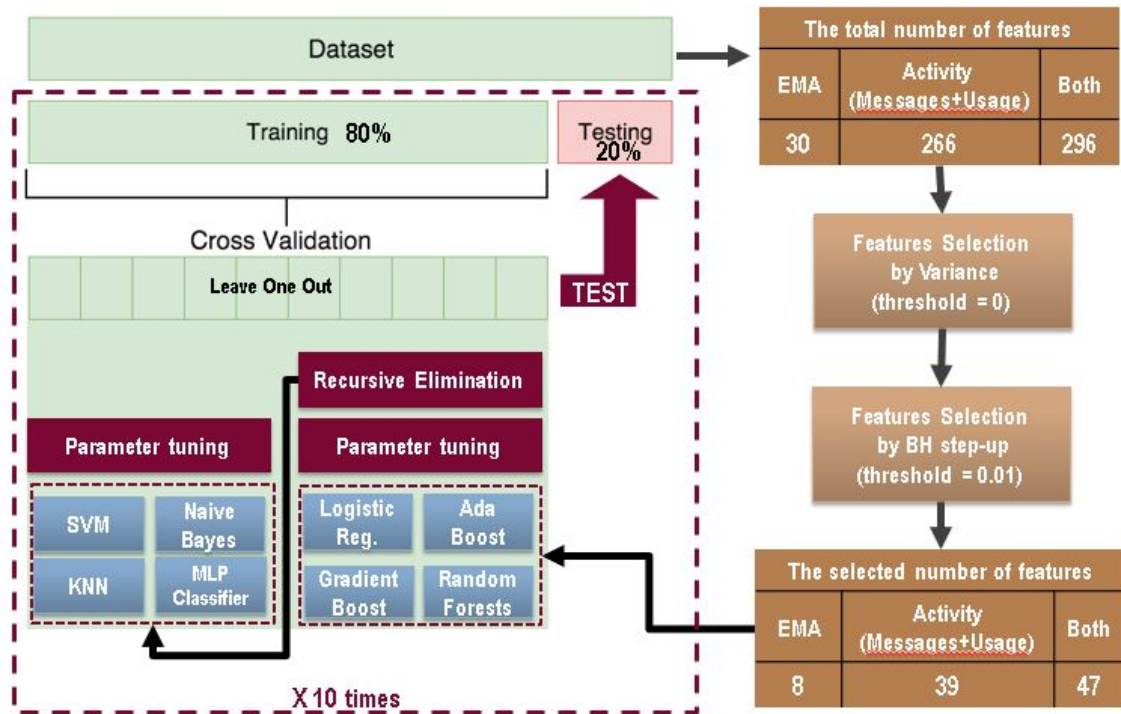


Figure 3. Method overview

Benjamini-Hochberg procedure (BH step-up, Benjamini and Hochberg, 1995) was used for feature selection. There is no universally accepted approach for dealing with the problem of multiple comparisons, and methods for multiple hypothesis testing are still evolving (McDonald, 2009; Stevens et al., 2017). Among many multiple hypothesis testing methods, the Benjamini-Hochberg procedure is commonly used and yields not-too-conservative but not-too-generous results compared to other methods (Stevens et al., 2017).

It would be more interesting if the treatment outcomes could be predicted with only the data from the initial phase of the treatment. Also, it cannot be ruled out that behavior patterns in a first few days may better predict the treatment outcome. Thus, It was attempted to perform a BH step-up for the features that are made from the combined datasets of only the first 7, 14, and 30 days as shown in Figure 2b. However, a significant difference has not been found between the groups in the features generated from the first 7, 14, and 30 days of data, as shown in Table 4. When the features were transformed by using all the periods between  $X_1$  and  $X_f$ , significant differences were found between the groups in many features. Therefore, it is decided to use features generated from the period between  $X_1$  and  $X_f$ .

Because several models predicting the treatment success are supposed to be made with the EMA dataset, activity dataset, and a combination of the two datasets, BH step-up was applied to features from the EMA dataset, activity dataset and combined dataset, respectively. As a result, only 8 features out of 30 features from the EMA dataset and 39 features out of 266 features from the activity dataset survived. In the combined dataset, 47 significant features out of 296 features are selected, as shown in Figure 3. By making models with each dataset, we can compare the results of the several models made with the different datasets to check which types of data have greater predictive power.

	Number of significant features after BH step-up (0.01)
7 days	0
14 days	0
30 days	0
All period	43

Table 4. Setting condition of aggregating time window: the number of significant features after BH step-up. If only short-term data are used, no significant difference between the two groups can be found in the features.

### 3.5 Training models

Classification algorithms work in a different manner. The classification results accordingly may vary from algorithm to algorithm, and the importance of features can also be changed by algorithms. Therefore, it is important to use many algorithms to obtain better classification results and to assess feature importance from a comprehensive perspective. 8 different algorithms (AdaBoost, GradientBoost, random forest, KNN, SVM, logistic regression, MLP classifier, and Gaussian naive Bayes) were used for the experiment in the Python environment using scikit-learn (Pedregosa et al., 2011). The models were constructed with the features selected through BH step-up. Parameters were tuned using grid search in leave-one-out cross-validation (LOOCV) to exhaustively testing all possible splits in the small training set we have. At the same time, for the models based on decision tree (AdaBoost, GradientBoost, random forest) and logistic regression, recursive feature elimination (RFE, Guyon et al., 2002) were used for adaptive feature selection. Through this process, each model chooses different features according to the variable importance in each model. That is, each model is made with an appropriate number of features that are appropriate for each model. Then, the features selected from the models were synthesized and used to create the remaining four models (KNN, SVM, MLP classifier, and Gaussian naive Bayes) that cannot compute variable importance. Only the selected features through RFE were used for fitting the rest models. Therefore, the remaining four models can be made with only the important features that have some predictive power. Of course, depending on the situation, some features that do not contribute significantly to classification results may be included. However, it is not a great hindrance to the classification performance because only significant variables were kept alive using a rather strict criterion (0.01) in the BH step-up.

It is known that ensembles are often more accurate than the individual classifiers when there is a significant diversity among the models (Hansen and Salamon, 1990; Eom et al., 2008; Kim and Kang, 2012). To overcome the limited performance of single models, the model combination methods called ‘ensemble methods’ have been attempted. After obtaining results of all models, an ensemble model was created through majority voting with three classifiers that work in a different way (random forest: bagging, Gaussian naive Bayes: Bayes’ theorem, and GradientBoost: boosting).

### 3.6 Validation

The models were trained with 80% of the data, and the remaining 20% was used to test the performance of the models. Accuracy rate and area under the ROC curve (AUC) metrics are used to evaluate the performance of models made. Because the results can be vary by sampling order, training and validation procedure are repeated 10 times with different sampling orders. The accuracy rate and AUC are calculated by averaging. Additionally, the number of times that the variables are selected by RFE during 10 times of repetitions are counted. The variable with high counting measure might be an important variable regardless of the sampling order.

## 4 Results

The results exhibit minimal differences in accuracy and AUC values among the various algorithms. Table 5 and 6 describe the average accuracy rate and average AUC values with 95% confidence intervals. In general, the random forest model worked well for the whole dataset. The majority class of the testset accounts for 55%. Therefore, the baseline (a reference measure) for accuracy rate is set as 55% and compared the accuracy rate with the baseline.

	EMA	Activity	Combined
Adaboost	<b>0.7655 (0.7356-0.7954)</b>	0.6517 (0.6167-0.6868)	<b>0.7793 (0.7433-0.8153)</b>
GradientBoost	<b>0.7655 (0.7252-0.8058)</b>	0.6552 (0.6221-0.6883)	0.7724 (0.7377-0.8071)
KNN	0.6724 (0.6505-0.6943)	0.6345 (0.5794-0.6896)	0.7241 (0.6622-0.7861)
LogisticRegression	0.6966 (0.6551-0.7380)	0.6310 (0.5795-0.6826)	0.6828 (0.646-0.7195)
MLP	0.7138 (0.6564-0.7712)	0.6103 (0.5911-0.6296)	0.6897 (0.6552-0.7241)
NaiveBayes	0.7241 (0.6847-0.7635)	<b>0.6690 (0.6317-0.7062)</b>	0.7276 (0.6676-0.7876)
RandomForests	0.7414 (0.7079-0.7748)	0.6448 (0.5999-0.6897)	0.7655 (0.7275-0.8035)
SVM	0.7000 (0.6668-0.7332)	0.6310 (0.5823-0.6798)	0.7103 (0.6596-0.7611)
Ensemble	0.7310 (0.6996-0.7624)	0.6586 (0.6236-0.6937)	0.7690 (0.7307-0.8073)

Table 5. Accuracy rate with 95% confidence intervals.

The results showed that the features made by feature engineering procedure have a certain level of predictive power. It is noteworthy that a good predictive model was made with only the features generated from the EMA dataset. The boosting models with EMA responses obtained an average accuracy rate of 0.7655, which is 21.55% above the baseline accuracy. The Gaussian naive Bayes model using features created from the activity dataset also generated 11.9% higher accuracy than the baseline accuracy. The best model was the AdaBoost model using the features from the combined dataset, which produced an average accuracy of 0.7793 and an average AUC of 0.7870. The accuracy rate is 22.93% above the baseline accuracy.

Table 7 presents the most important features that were selected by different models (AdaBoost, GradientBoost, random forest, and logistic regression) through RFE. There was a difference between algorithms in evaluating important features, but the most important features are recognized as important variables by most algorithms. First, EMA responses have considerable predictive power. Patients who answered positively to the EMA questions were more likely to belong to the improved group. Second, the feature related to the cognitive restructuring (CR) module is selected

	EMA	Activity	Combined
Adaboost	0.7738 (0.744-0.8036)	0.6635 (0.632-0.6949)	<b>0.7870 (0.7508-0.8232)</b>
GradientBoost	<b>0.7745 (0.7355-0.8135)</b>	0.6572 (0.6233-0.6911)	0.7815 (0.748-0.8149)
KNN	0.6772 (0.6510-0.7034)	0.6421 (0.5849-0.6993)	0.7305 (0.6702-0.7909)
LogisticRegression	0.6846 (0.6456-0.7237)	0.6339 (0.5804-0.6874)	0.6808 (0.6440-0.7176)
MLP	0.7161 (0.6575-0.7747)	0.6043 (0.5848-0.6238)	0.6913 (0.6560-0.7267)
NaiveBayes	0.7233 (0.6861-0.7605)	<b>0.6863 (0.6503-0.7223)</b>	0.7337 (0.6758-0.7915)
RandomForests	0.7454 (0.7119-0.7789)	0.6500 (0.6091-0.6909)	0.7724 (0.7352-0.8095)
SVM	0.6993 (0.6636-0.7349)	0.6332 (0.5849-0.6815)	0.7130 (0.6639-0.7620)
Ensemble	0.7353 (0.7042-0.7665)	0.6654 (0.6341-0.6967)	0.7762 (0.7386-0.8138)

Table 6. AUC values with 95% confidence intervals.

in the models repetitiously. Those who focused on the CR module over a certain amount were more likely to belong to the improved group. Third, the variable datediff, which indicates the date difference between the first PHQ-8 test and the last PHQ-8 test, also plays an important role in the models made with the activity dataset. Additionally, patients who had long-term valid responses to the EMA questions were more likely to be improved. Furthermore, the variables indicating the completion of the module also contributed to the models. See Table 9, 10, 11 for the detailed selection results.

Rank of importance	EMA	Activity	Combined
1	EMA_moodavg_sum	datediff	EMA_moodavg_sum
2	EMA_allavg_sum	weblog_crd_ratio	weblog_crd_ratio
3	EMA_allavgdid_sum	weblog_complete_exex_trend	EMA_allavg_sum

Table 7. Top 3 important features that were selected by different models (AdaBoost, Gradient-Boost, random forest, and logistic regression). The rank was determined by how many times the variable was selected by the models during 10 iterations.

## 5 Discussion

This work focused on applying machine learning method to predict treatment success of blended CBT intervention for depression, making use of log data collected from the system. Applying classification algorithms for predicting treatment success appears promising in the domain of depression. Many features were created through extensive feature engineering, and the generated features helped to make more accurate predictions than those from previous research (Rocha et al., 2018). The variables were comprehensively evaluated by using a variety of algorithms. Finally, several classifiers were combined to explore the possibility of improving outcomes through an ensemble.

it was observed that features related to EMA records and CR module have considerable predictive power for predicting treatment success. It seems that patients who respond to the EMA questions positively are more likely to experience improvement. This finding is parallel to the argument that measures based on EMA have the potential for assessing a wide variety of outcome variables (Moskowitz and Young, 2006). Patients who put more weight on cognitive restructuring are more likely to experience improvement than others. When patients complete most of the

CR modules, the portion of their investment in the CR module increases because the CR module contains longer contents than all other modules. Therefore, a low proportion of investments in the CR module means that the CR module has not been properly completed. Of course, not all of the patients who took most parts of the CR module were improved, but those who did not take most of the CR module were not improved. This finding is in line with the discovery of Rocha et al. (2018), who said that patients have recovered 90% of their total improvement when the patients completed the fourth module (CR module). Taken together, it can be inferred that the CR module plays a decisive role in CBT. Moreover, features that measure module completion and recompletion are important. It appears that the completion or recompletion of a particular module play a role. It is important to note that completion of modules located in the back of the sequence such as cognitive restructuring (CR), physical exercise (EX), and relapse prevention (Eval) played an important role. This result suggests that dropout management of patients is important for better outcomes. Finally, the longer the interval is between the first test and the last test, the higher the probability of the patient belonging to the improved group. It is also in the same context that patients who have a long-term valid response to an EMA question are more likely to belong to the improved group. The longer a patient has been in the treatment, the more opportunity to respond to the EMA questions. A possible explanation for the relationship between duration and treatment outcome might be that the blended CBT treatment is somewhat effective.

The features from EMA dataset and activity dataset had significant predictive power. However, combining EMA dataset with activity dataset had no significant impact on the results, which is also pointed out in Rocha et al. (2018). Apparently, the response values for EMA questions may be related to the interim results of treatment because some of the questions in PHQ-8 are very similar to the EMA questions, as indicated in Table 8. In this respect, a chain of the EMA responses can be regarded as a condensed reflection of patients' progress. Thus, it is not strange that the features from the EMA dataset have fairly strong predictive power. It is also reasonable that the combination of the activity dataset and the EMA dataset does not lead to a massive improvement in the results. This result implies that EMA responses must be gathered systematically and used more actively in the field of depression. If we can produce crucial features that complement existing EMA responses from patients' activities, combining EMA dataset and activity dataset may lead to meaningful improvement in the results.

Notwithstanding the respectable accuracy rate, there is still room for improvement regarding the models with the activity dataset. The possibility of predicting treatment outcomes using only activity dataset has been demonstrated. However, it is still not extraordinarily sufficient to say that the models with activity dataset can predict treatment success very well. More sophisticated models perhaps can be made with a deep feature engineering from richer data. Additionally, adjusting the aggregation interval of features per patient would be helpful to increase accuracy as suggested in van Breda et al. (2016a).

Considering different time windows has been attempted in a manner that explores how many variables have significant differences through the BH step-up. As a result, data between the pretest and the posttest were used to predict the treatment success. Given more abundant baseline characteristics of patients, it is expected that prediction with the data of first few days will be possible. Increased numbers of data are essential to combine log-data with baseline characteristics to pre-

dict treatment outcomes. Existing baseline characteristic such as country and therapist could not be used for the generalizability of models, as mentioned in section 3, although previous studies have confirmed that there are large differences of usage and behavior according to countries (Rocha et al., 2018). If many more participants use the system, it will be able to achieve higher accuracy by using the different baseline characteristics of patients.

The ensemble model based on majority voting had no significant impact in terms of improving the results. Several works have investigated the cause of nonimprovement and insisted that the performance of ensemble can be even degraded where multiple classifiers of an ensemble are highly correlated, which leads to performance degradation of the ensemble methods (Hansen and Salamon, 1990; Kim and Kang, 2012). In other words, it appears that the diversity of each individual classifier is not secured in our models. One possible future study is to use different algorithms that have not been used and utilize more advanced ensemble method such as weighted-voting to increase the performance.

Item	Questions
EMA Question 1. PHQ-8 Question 3.	Sleep quality How well did you sleep last night? Trouble falling or staying asleep, or sleeping too much
EMA Question 2. PHQ-8 Question 4.	Mood How is your mood right now? Feeling down, depressed, or hopeless
EMA Question 4. PHQ-8 Question 6.	Self-esteem How do you feel about yourself right now? Feeling bad about yourself, or that you are a failure, or have let yourself or your family down
EMA Question 7. PHQ-8 Question 1.	Level of pleasant activities To what extent did you accomplish pleasant activities today? Little interest or pleasure in doing things

Table 8. Similar questions that can be logically correlated

## 6 Conclusions

The models made yielded promising results. It is confirmed that the features created from the datasets have significant predictive power in predicting treatment success. Treatment guidance, personalized therapy, improved treatment outcomes, and cost reduction may be potentially benefited by the models made and the variables found. To enable these findings to be practically applied to the actual treatment field, it would be great to actively explore more influential variables and make more elaborate models with more data in the future.

## References

- Amichai-Hamburger, Y., Brunstein Klomek, A., Friedman, D., Zuckerman, O., and Shani-Sherman, T. (2014). The future of online therapy. *Computers in Human Behavior*, 41.
- Asselbergs, J., Ruwaard, J., Ejdys, M., Schrader, N., Sijbrandij, M., and Riper, H. (2016). Mobile phone-based unobtrusive ecological momentary assessment of day-to-day mood: an explorative study. *Journal of medical Internet research*, 18(3).
- AX, G., NJ, A., H, M., and et al (2005). Effects of computerized clinical decision support systems on practitioner performance and patient outcomes: A systematic review. *JAMA*, 293(10):1223–1238.
- Becker, D., Bremer, V., Funk, B., Asselbergs, J., Riper, H., and Ruwaard, J. (2016). How to predict mood? delving into features of smartphone-based data. *Twenty-second Americas Conference on Information Systems*.
- Bengio, Y., Courville, A., and Vincent, P. (2013). Representation learning: A review and new perspectives. *IEEE transactions on pattern analysis and machine intelligence*, 35(8):1798–1828.
- Benjamini, Y. and Hochberg, Y. (1995). Controlling the false discovery rate: a practical and powerful approach to multiple testing. *Journal of the royal statistical society. Series B (Methodological)*, pages 289–300.
- Castillo, R., Carlat, D., Millon, T., Millon, C., Meagher, S., Grossman, S., Rowena, R., Morrison, J., Association, A. P., et al. (2007). *Diagnostic and statistical manual of mental disorders*. Washington, DC: American Psychiatric Association Press.
- El Alaoui, S., Hedman-Lagerlöf, E., Ljótsson, B., and Lindfors, N. (2017). Does internet-based cognitive behaviour therapy reduce healthcare costs and resource use in treatment of social anxiety disorder? a cost-minimisation analysis conducted alongside a randomised controlled trial. *BMJ Open*, 7(9).
- Eom, J.-H., Kim, S.-C., and Zhang, B.-T. (2008). Aptacdss-e: A classifier ensemble-based clinical decision support system for cardiovascular disease level prediction. *Expert Systems with Applications*, 34(4):2465–2479.
- Gustavsson, A., Svensson, M., Jacobi, F., Allgulander, C., Alonso, J., Beghi, E., Dodel, R., Ekman, M., Faravelli, C., Fratiglioni, L., et al. (2011). Cost of disorders of the brain in europe 2010. *European neuropsychopharmacology*, 21(10):718–779.
- Guyon, I., Weston, J., Barnhill, S., and Vapnik, V. (2002). Gene selection for cancer classification using support vector machines. *Machine learning*, 46(1-3):389–422.
- Hansen, L. K. and Salamon, P. (1990). Neural network ensembles. *IEEE transactions on pattern analysis and machine intelligence*, 12(10):993–1001.



- Heaton, J. (2016). An empirical analysis of feature engineering for predictive modeling. In *South-eastCon, 2016*, pages 1–6. IEEE.
- Jacobson, N. S. and Truax, P. (1991). Clinical significance: a statistical approach to defining meaningful change in psychotherapy research. *Journal of consulting and clinical psychology*, 59(1):12.
- Kim, M.-J. and Kang, D.-K. (2012). Classifiers selection in ensembles using genetic algorithms for bankruptcy prediction. *Expert Systems with applications*, 39(10):9308–9314.
- Kleiboer, A., Smit, J., Bosmans, J., Ruwaard, J., Andersson, G., Topooco, N., Berger, T., Krieger, T., Botella, C., Baños, R., et al. (2016). European comparative effectiveness research on blended depression treatment versus treatment-as-usual (e-compared): study protocol for a randomized controlled, non-inferiority trial in eight european countries. *Trials*, 17(1):387.
- Kondratovich, E., Baskin, I. I., and Varnek, A. (2013). Transductive support vector machines: Promising approach to model small and unbalanced datasets. *Molecular Informatics*, 32(3):261–266.
- Kroenke, K., Strine, T. W., Spitzer, R. L., Williams, J. B., Berry, J. T., and Mokdad, A. H. (2009). The phq-8 as a measure of current depression in the general population. *Journal of affective disorders*, 114(1):163–173.
- Kroenke MD, K., L. Spitzer, R., and Williams, J. (2001). The phq-9. *Journal of General Internal Medicine*, 16:606 – 613.
- Lambert, M. J. (2010). Yes, it is time for clinicians to routinely monitor treatment outcome. *American Psychological Association*.
- Mathers, C. D. and Loncar, D. (2006). Projections of global mortality and burden of disease from 2002 to 2030. *PLoS medicine*, 3(11):e442.
- McDonald, J. H. (2009). *Handbook of biological statistics*, volume 2. sparky house publishing Baltimore, MD.
- Mikus, A., Hoogendoorn, M., Rocha, A., Gama, J., Ruwaard, J., and Riper, H. (2018). Predicting short term mood developments among depressed patients using adherence and ecological momentary assessment data. *Internet interventions*, 12:105–110.
- Moskowitz, D. S. and Young, S. N. (2006). Ecological momentary assessment: what it is and why it is a method of the future in clinical psychopharmacology. *Journal of Psychiatry and Neuroscience*, 31(1):13.
- Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., and Duchesnay, E. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.

- Rocha, A., Camacho, R., Ruwaard, J., and Riper, H. (2018). Using multi-relational data mining to discriminate blended therapy efficiency on patients based on log data. *Internet interventions*, 12:176–180.
- Shiffman, S., Stone, A. A., and Hufford, M. R. (2008). Ecological momentary assessment. *Annu. Rev. Clin. Psychol.*, 4:1–32.
- Sobocki, P., Jönsson, B., Angst, J., and Rehnberg, C. (2006). Cost of depression in europe. *Journal of Mental Health Policy and Economics*.
- Srivastava, M. S., Joshi, M. N., and Gaur, M. (2014). A review paper on feature selection methodologies and their applications. *IJCSNS*, 14(5):78.
- Stevens, J. R., Al Masud, A., and Suyundikov, A. (2017). A comparison of multiple testing adjustment methods with block-correlation positively-dependent tests. *PloS one*, 12(4):e0176124.
- Taylor, C. B. and Chang, V. Y. (2008). Issues in the dissemination of cognitive–behavior therapy. *Nordic Journal of Psychiatry*, 62(sup47):37–44.
- Tylee, A. (2000). Depression in europe: experience from the depres ii survey. *European Neuropsychopharmacology*, 10:S445–S448.
- van Breda, W., Bremer, V., Becker, D., Hoogendoorn, M., Funk, B., Ruwaard, J., and Riper, H. (2017). Predicting therapy success for treatment as usual and blended treatment in the domain of depression. *Internet Interventions*.
- van Breda, W., Hoogendoorn, M., Eiben, A., Andersson, G., Riper, H., Ruwaard, J., and Vernmark, K. (2016a). A feature representation learning method for temporal datasets. In *Computational Intelligence (SSCI), 2016 IEEE Symposium Series on*, pages 1–8. IEEE.
- van Breda, W., Pastor, J., Hoogendoorn, M., Ruwaard, J., Asselbergs, J., and Riper, H. (2016b). Exploring and comparing machine learning approaches for predicting mood over time. In *International Conference on Innovation in Medicine and Healthcare*, pages 37–47. Springer.
- Wasikowski, M. (2009). *Combating the class imbalance problem in small sample data sets*. PhD thesis, University of Kansas.
- Wentzel, J., van der Vaart, R., Bohlmeijer, T. E., and van Gemert-Pijnen, C. J. E. W. (2016). Mixing online and face-to-face therapy: How to benefit from blended care in mental health care. *JMIR Mental Health*, 3(1):e9.

## A Appendix

	AdaBoost	GradientBoost	Logistic regression	Random forest	Total
EMA_question2avg_sum	10	7	7	10	34
weblog_crd_ratio	9	4	5	8	26
EMA_allavg_sum	6	6	0	9	21
EMA_question2avg_mean	5	0	4	7	16
EMA_allavgdid_sum	6	0	1	7	14
EMA_question2did_sum	0	5	1	7	13
EMA_allavg_mean	3	1	0	7	11
datediff	1	0	2	8	11
weblog_complete_exex_trend	1	0	4	6	11
EMA_question2avg_max	2	0	3	5	10
weblog_complete_evalex_std	0	0	4	6	10

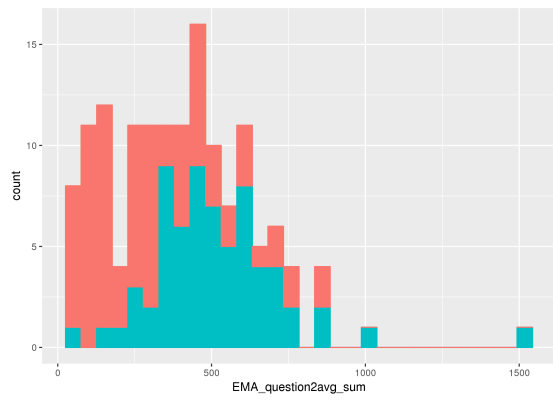
Table 9. Feature importance ranking in detail: combined dataset

	AdaBoost	GradientBoost	Logistic regression	Random forest	Total
datediff	9	9	2	9	29
weblog_crd_ratio	4	10	4	10	28
weblog_complete_exex_trend	3	4	3	8	18
weblog_complete_evalex_std	1	1	4	7	13
weblog_bad_ratio	1	1	1	10	13
weblog_complete_cr_std	0	4	1	7	12
msgreceive_trend	1	1	1	7	10
weblog_complete_evalex_mean	1	2	1	6	10

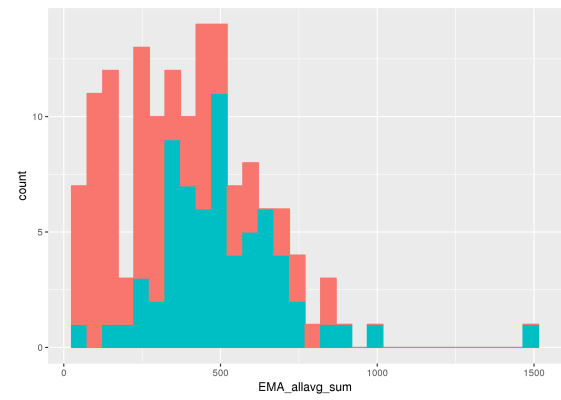
Table 10. Feature importance ranking in detail: activity dataset

	AdaBoost	GradientBoost	Logistic regression	Random forest	Total
EMA_question2avg_sum	8	9	9	10	36
EMA_allavg_sum	8	9	7	9	33
EMA_allavgdid_sum	5	5	8	9	27
EMA_question2did_sum	0	6	7	9	22
EMA_question2avg_mean	1	2	7	6	16
EMA_allavg_mean	1	2	7	6	16
EMA_question2avg_max	2	3	7	4	16
EMA_allavg_max	0	2	7	3	12

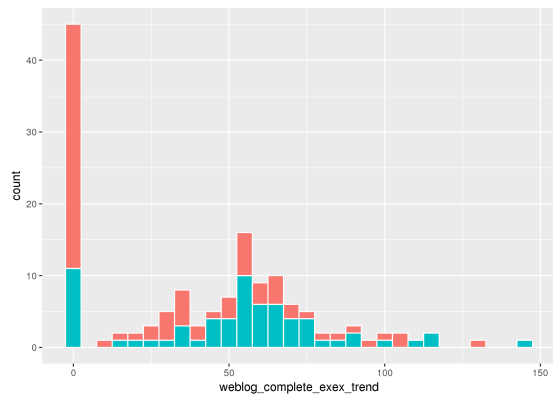
Table 11. Feature importance ranking in detail: EMA dataset



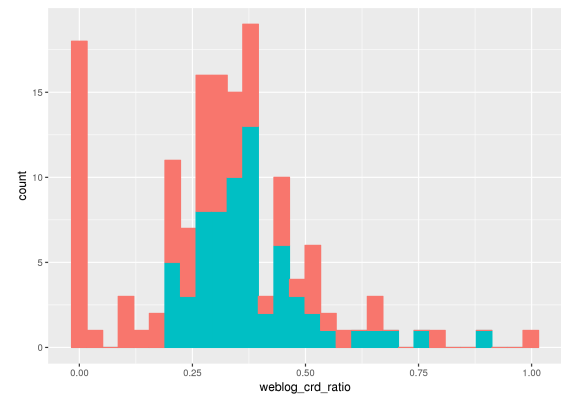
(a) Positive EMA response records (mood)  
— EMA\_question2avg\_sum



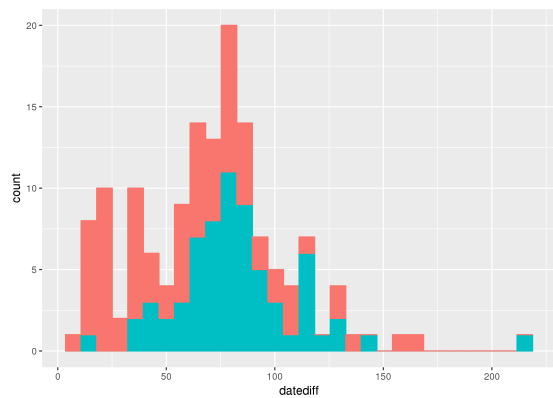
(b) Positive EMA response records (comprehensive)  
— EMA\_allavg\_sum



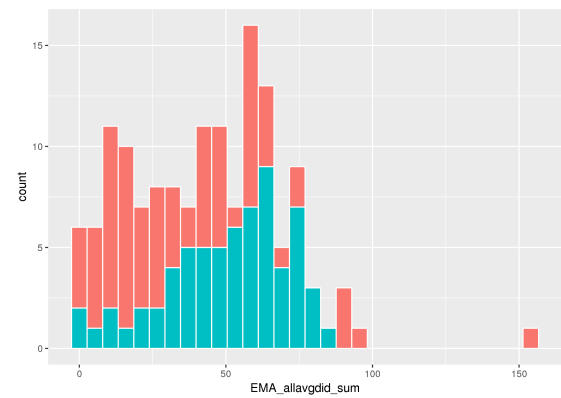
(c) Exercise trend of physical exercise module  
— weblog\_complete\_exex\_trend



(d) Ratio of cognitive restructuring module  
— weblog\_crd\_ratio



(e) Large date difference between two tests  
— datediff

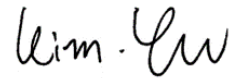


(f) Length of day that responded to EMA  
— EMA\_allavgdid\_sum

Figure 4. Distribution of the most influential features: nonimproved group(coral) and improved group(cyan).

## **Eidesstattliche Erklärung**

Ich versichere, dass ich diese Master-Arbeit selbstständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt habe. Ich versichere, alle Stellen der Arbeit, die wortwörtlich oder sinngemäß aus anderen Quellen übernommen wurden, als solche kenntlich gemacht und die Arbeit in gleicher oder ähnlicher Form noch keiner Prüfungsbehörde vorgelegt zu haben.



**Lüneburg, 13. 09. 2018**

**Yongwoo Kim**