Research Paper Business Analytics

Analyzing and Predicting Mood of Depression Patients

Author:
Johnno Pastor

Supervisor:
Ward van Breda

Vrije Universiteit Amsterdam
Faculty of Science
De Boelelaan 1081a
1081 HV Amsterdam

September 23, 2015

Abstract

In this research paper we looked at the mood of individuals suffering from depression. We applied time series analysis techniques such as decomposition and clustering to gain insight into the historical data and measurements obtained from the participant's smartphone during this period. This data was used to construct models using machine learning algorithms to predict mood. Of the applied algorithms the Support Vector Machine algorithm performed best, correctly predicting 68.88%. The constructed models were analyzed to assess which variables were important: we found that in addition to the expected variables describing the historical mood the important variables are diverse and dependent on the individual.

Contents

1 Introduction

1.1 Goal and Research Question

The goal of this research paper is to predict mood of participants suffering from depression based on their historical mood and logged smartphone data. Besides just fitting models to predict mood, the important variables in these models will be analyzed per individual to answer the question of what variables have an effect on mood. This will be achieved by making use of time series analysis as well as data mining techniques. This will be done on the basis of the following research question. Can historical mood and logged smartphone sensor data be used to construct a model that predicts the future mood and which factors are important in this prediction?

To answer this question three subquestions are considered. Which techniques can be used to analyze and compare time series? Which algorithm performs best in predicting mood? Which variables are important in influencing mood?

1.2 Background

For GGZ Nederland (Geestelijke Gezondheidszorg - Mental Healthcare Netherlands) one of the fundamental principals is to keep mental healthcare affordable and accessible. One of the resources that is made use of is called e-Health, an emerging field that employs information and communication technology for support and improvements in healthcare [Sorbi and Riper, 2009]. As described by Sorbi and Riper [2009], E-mental health has several benefits. The main benefit is achieved by the increase of reach and accessibility of advisable preventions. The application of technology to analyze and predict is (as of yet) not widely used in e-mental Health.

There are numerous benefits that can be derived from predicting mood and finding out which factors play a role. Finding out which variables influence mood offer many possibilities, one of the possible benefits is acting on a future trend in mood.

The data used in this research paper describes participants suffering from depression, this background was kept in mind during the research process. However, the applications of this research are not only applicable to people suffering from depression, examples of this include mood sharing and mood-enhanced applications [LiKamWa et al., 2013], an example of this is the MoodScope application.

1.3 Outline

The outline of this research paper is as follows. We begin with discussing related research in Section 2, followed by a description of the data in Section 3. The applied techniques are described in Section 4, in which we cover techniques regarding time series analysis in Subsection 4.1 and machine learning algorithms in Subsection 4.2. Next we describe the method in Section 5. At last, Section 6 describes the results and interpretation of these results. We conclude with the conclusion in Section 7, followed by the discussion and future work in Section 8.

2 Related Research

In this chapter we will give an overview of the research already been done in the field of e-mental health regarding mood prediction and analysis. This chapter does not provide an extensive list but merely a short overview. In particular a study by LiKamWa et al. [2013]: "MoodScope: building a mood sensor from smartphone usage patterns" will be discussed, due to the original goal of the composed dataset of replicating the results of this study as well as the strong similarity to this research paper.

One way depression has been looked at is as a dynamic system. A dynamic system is a system that can be used to describe nonlinear behaviours over time using internal feedback loops and time delays. System dynamics offer opportunities in public health care [Homer and Hirsch, 2006]. The methodology is "well suited to address the dynamic complexity that characterizes many public health issues". Demic and Cheng [2014] have proposed such a dynamical systems model for the progression of major depressive disorder. In this study the computation modeling has been used to improve the understanding of major depressive disorder; the model is abstract and combines several major factors (mechanisms) that influence the dynamics of major depressive disorder.

Another approach that has been studied is the length of depression episodes [Tomitaka and Furukawa, 2014]. It is observed that there has not been much research on developing mathematical models describe (the durations of) depressive episodes, even though mathematical models should be especially of interest since the duration of these episodes vary widely. The study has sought to fit a mathematical model to the duration of depressive episodes and has shown the duration is best modeled by a log-normal model or a power law model.

The influence of particular variables on mood has been widely studied. Literature suggests there is a positive correlation between mood and daily activities [Clark and Watson, 1988] and social interactions [Joseph P Forgas, 1984]. At current times most of the population is in possession of a smartphone, these smartphones contain rich information about the user, including information about social interactions and activities. This information can be leveraged to predict mood, as has been attempted by LiKamWa et al. [2013]. In this study an application has been build that predicts mood based on smartphone data like browser history, application usage and location history. The task of predicting mood produces remarkable results: with out of bag sampling and using two months personal of data to construct a model in 93% of the cases the squared error was below 0.25. Mood in this study was measured in two dimensions: pleasure and activeness, both measured on a scale from 1 to 5. Some concerns regarding the validation of these results should be addressed: the validation process makes use of leave-one-out sampling and thus data "from the future" is possibly used to construct models.

3 Data

3.1 Background

The data used in this research paper is from the VU Unobtrusive Ecological Momentary Assessment Pilot Study Data. This data consists of measurements obtained through two applications installed on the smartphone of participants. The data can be split into two components: mood as assessed by the participants and logged smartphone sensor data.

The data was put together by Joost Asselbergs, Jeroen Ruwaard and Heleen Riper. The primary goal for the originally composed dataset was for replication of the study by LiKamWa et al. [2013]: "MoodScope: building a mood sensor from smartphone usage patterns".

3.2 Description

The measurement were obtained over a period of about six weeks in which data was collected by 33 users of which 27 contributed enough data for meaningful analysis. The data consists of a 76 variables and 1249 observations. The data was obtained through two applications installed on the participant's smartphone: the eMate Ecological Momentary Assessment (EMA) application and the IYOuVu application. The eMate application prompts the user to rate their mood five times per day and the IYOuVu application is a sensor logger. The duration for which the participants logged data is not the same for all participants, also some values are missing. In Figure 1 the logged mood per participants is displayed.

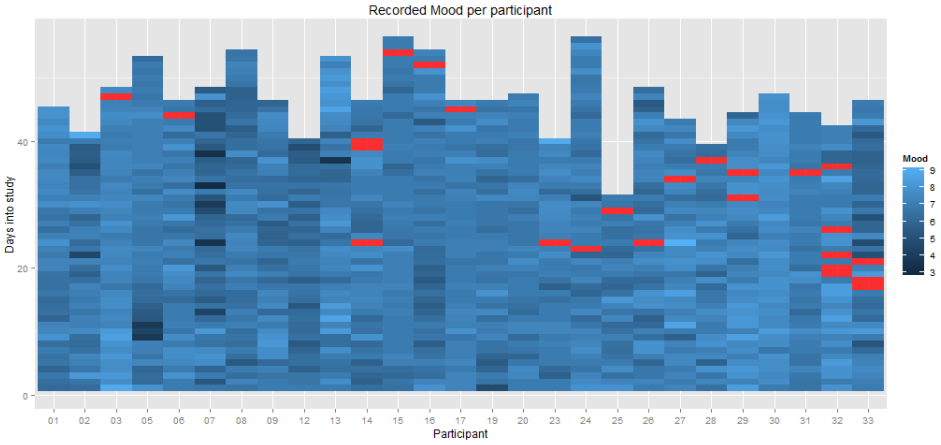


Figure 1: Mood per participant per day, red indicates missing

The data as obtained is already quite preprocessed. This preprocessing was done mainly following LiKamWa et al. [2013], who make use of normalized features. Other preprocessing has also already been applied. An example of this is the mood variable: the participant

was prompted to enter mood multiple times per day, in the dataset only the "flattened" variable per day is present.

The participants mood is expressed in three ways of which two are kept track of (almost) each day. First, the users mood is expressed unidimensional as "mood" on a scale from 1-10, second, mood is expressed following the Circumplex mood model. The Circumplex model employs two dimensions, the valence and arousal dimension, to describe mood. The values in both of these dimensions range from -2 to 2. Besides the measures logged by the participants the results of CESD depression questionnaires held during the study are available.

The variables obtained through the IYOuVu application include the number of images taken (image.n), the average screen duration per screen-on moment (screen.duration) and the screen-on frequency (screen.n) and were all standardized within the participant. Another variable obtained through the IYOuVu application is the average percentage of accelerometer data point that are classified as "high" (accelerometer.high). The appCat .n and appCat .duration are features representing the number of uses and duration for application categories. These variables are normalized per day between all categories of their respective class. The categories include: android, books, browser, business, education, entertainment, game, life style, email, music, news, productivity, social, tools, transportation, unknown and EMA/UEMA. The following features used by LiKamWa et al. [2013] are also included: the normalized frequency of the number of SMS messages and calls made to the top 5 contacts, the duration of calls made to the top 5 contacts and the number of and duration of the most / longest used application. For each day the day, participant, averaged mood, valence and arousal are available. Also one and two day lagged variables of the logged mood variables are available, as well as an interpolated CESD depression questionnaires score.

More about the processing of these variables, feature construction and missing values and how these are dealt with can be found Section 5: Method.

4 Theory of Applied Techniques

This chapter provides a background on the techniques used in this research paper. We start by describing time series and techniques for time series analysis in the first subsection. Then we describe the differing machine learning algorithms used to build models for the prediction of mood in the second subsection.

4.1 Time Series Analysis

A stochastic time series denoted X is a sequence $x_0, x_1, x_2, ..., x_n$, of random variables which represents a set of observations made sequentially in time [M.C.M. de Gunst, 2013]. The subscript t indicates the time at which the variable x_t was observed. If the x_t could be completely arbitrary, then it would be impossible to infer something about the distribution of the observations $(x_0, x_1, ..., x_t)$, let alone of that of the future $x_{t+1}, x_{t+2}, ...$, therefore one has to make some assumptions about the structure of the data. Often a time series consists of repeated observations on the same object. In this case we are dealing with a multivariate time series, the observation of more than one variable made sequentially in time. The time series variables at time t can be denoted by $x_{1t}, x_{2t}, ..., x_{nt}$. A multivariate time series then is made up of these variables recorded over time over time.

4.1.1 Goals of time series analysis

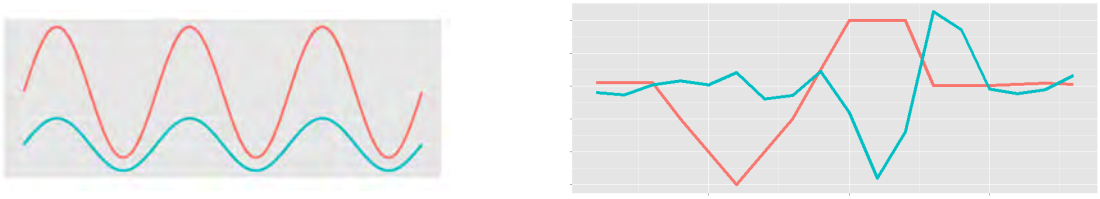
Time series analysis can be used to accomplish a number of goals. Ratanamahatana et al. [2010] describe the tasks considered by the time series data mining community.

Time series analysis can be used for **summarization** of data. Because of the nature of time series, often more than a single number is needed to summarize essential features of the times series, as opposed to a random sample from a distribution. As opposed to making use of summary statistics, time series data is more often graphically represented.

Another goal is **prediction** of future values based on observed values. In time series analysis the assumption is often made that the data consist of a systematic pattern (usually a set of identifiable components) and random noise (error). Future values can be predicted by separating the noise and identifying the systematic patterns.

Two other goals are indexing (query by content) and clustering. For some similarity measure the goal of **indexing** is to find for a given time series the most similar time series in a 'database', the goal of **clustering** is to find groupings in such a 'database' of time series.

Other major tasks included by Ratanamahatana et al. [2010] are anomaly detection, segmentation and classification. **Anomaly detection** is the goal of finding all sections of a time series which contain "surprising, interesting or unexpected" occurrences, **segmentation** is the partitioning of a time series in sections and **classification** is the assignment of a time series to a class. The primary focus in this research paper is on the task of prediction, other tasks such as summarization, clustering and indexing are also tackled.



(a) Scale and Shift y-axis

(b) Shift x-axis and acceleration

Figure 2: (Dis)Similar Times Series

4.1.2 Similarity measures

In the previous subsection we defined tasks that may be considered when regarding time series. In two of these tasks, indexing and clustering, a similarity measure is directly mentioned. This similarity measure is also often (indirectly) used in other tasks, such as prediction. Similarity measures quantify the degree to which multiple time series are similar to each other. The notion of similarity is subjective; there is no fixed expression that describes how similar two time series are, and thus different similarity measures exist.

One of the issues when dealing with time issues is deriving a similarity measure that reflects similarities. In choosing a similarity measure there are several features that are favorable. Time series can often be considered as multi-dimensional data and thus the similarity measure should be able to deal with high-dimensional data. Besides, the calculation should be fast and efficient. It is also beneficial if the similarity measure is robust to different data types and can be applied to a range of problems [Joan Serrà, 2014].

Euclidean Distance

One of the simplest similarity measures for times series is the Euclidean distance. The Euclidean distance measure is straight forward, easy to interpret and easy to compute. Given two time series $X = x_1, x_2, ..., x_n$ and $Y = y_1, y_2, ..., y_n$ with equal length n, the Euclidean distance can be defined as follows:

$$d_E(X, Y) = \sqrt{\sum_{i=1}^{n} (x_i - y_i)^2}$$

Even though two time series seem similar following some subjective notion of similarity, the Euclidean distance measure does not always reflect this in its measure. An example of this can be seen in Figure 2a. In this figure the sequences are not close to each other in an Euclidean sense. The difference in the Euclidean measure comes from a difference in fluctuation (scale) and a difference in starting values (shift).

Similarity transformation

Arguably a similarity measure should take into account the transformations of scaling and shifting. This can be done by making use of so called similarity transformations which will be briefly explained here, for a more extensive description see Goldin and Kanellakis

[1995].

Again denote a time series sequence X of n real numbers by $x_1, ..., x_n$, the average of this sequence by $\alpha(X)$ and the deviation by $\sigma(X)$. A similarity transformation denoted $T_{a,b}$ with $a, b \in \mathbb{R}$ maps each element x_i to $a * x_i + b$. Then X is similar to Y if there exists some (a, b) such that $X = T_{a,b}(Y)$. By only permitting $a > 0$ it is implied that a sequence symmetric to X with respect to the x-axis is not considered similar.

Every sequence X has a normal form denoted $\nu(X)$ in which $\alpha(X) = 1$ and $\sigma(X) = 1$. This normal form van be obtained by applying the transformation of $T_{1/\sigma,-\alpha/\sigma}(X)$. The similarity distance between two sequences X and Y is the similarity of their normal forms. For example using the Euclidean distance the similarity distance is equal to:

$$d_S(X, Y) = d_E(\nu(X), \nu(Y))$$

After applying the transformations of scaling and shifting with regard to the y-axis are accounted for. Another way in which two time series seem similar following some subjective notion of similarity, which is not reflected in the measure that compares transformed sequences is illustrated in Figure 2b. In this figure the sequence is "accelerated" and shifted with regard to the x-axis.

Dynamic Time Warping

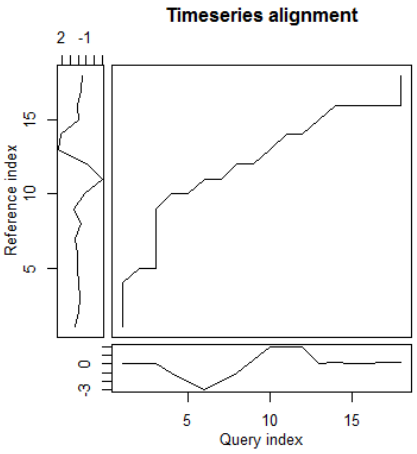
Arguably a similarity measure should take into account the acceleration and deceleration of time series, as well as a shift in time (lagg). This can be done by making use of a technique known as dynamic time warping (DTW). Dynamic time warping has been applied in many areas such and has applications in video, audio, and graphics data. It allows for non-linear "warping" of one signal to another, it allows two time series that are similar but out of phase in the time dimension to align.

Given two time series $X = x_1, x_2, ..., x_n$ and $Y = y_1, y_2, ..., y_m$ (not necessarily of the same length). These sequences can be aligned using DTW by constructing an matrix where the each element of the matrix corresponds to the distance between two points of the two time series. The best alignment between these time series can be found by finding a path through the matrix that minimizes the sum of the distances, which can be obtained using dynamic programming.

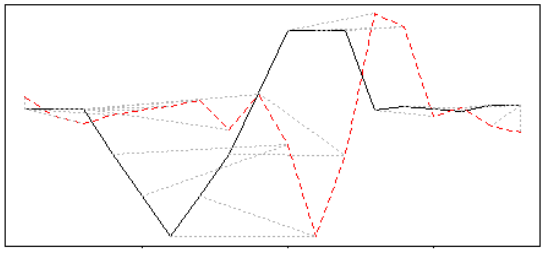
In Figure 3 the results of applying DTW to the example in Figure 2b are shown. Furthermore certain constraints can be composed to limit the number of paths to explore, such as continuity to prevent the alignment path from jumping in time and the warping window which guarantees that the alignment does not get stuck and the alignment path wanders to far from the diagonal. For a more descriptive overview of DTW and the constraints see [Müller, 2007].

Other similarity measures

There exist a range of similarity measures and many different variations. An example of different measures is the euclidean distance: also known as the L^2-norm, which may be generalised to the L^P-norm. The L^1-norm is better known as the Hamming distance, a measure that is often used in information theory when comparing strings. An exam-



(a) Point-by-point comparison



(b) Warping curve

Figure 3: Example Dynamic Time Warping

ple of different variations is dynamic time warping: by applying different constraints the similarity measure is considered to be different.

An empirical evaluation of similarity measures was carried out by Joan Serrà [2014], and concludes that: "One of the main conclusions of the study is that, even though the newly proposed measures can be theoretically attractive, the efficacy of some common and well-established measures is, in the vast majority of cases, very difficult to beat. Specifically, dynamic time warping (DTW; Berndt and Clifford, 1994) is found to be consistently superior to the other studied measures (or, at worst, for a few datasets, equivalent). In addition, the authors emphasize that the Euclidean distance remains a quite accurate, robust, simple, and efficient way of measuring the similarity between two time series."

4.1.3 Deconstruction of time series

Time series can be represented in multiple ways. The original time series can be kept as is, or it may be deconstructed to another or more simple form. An example of such a deconstruction is the derivation of a sequence of up, down, up, down, etc from an original sequence. More complicated deconstructions include the decomposition of the original time series by making use of fourier transforms. Fourier transforms are based on Fourier Series, they represent periodic time series in the frequency domain as a sum of sine and cosine components. Another example is to decompose a time series into a trend component, a cyclical component a seasonal component and a random component.

The process of destructing time series can be helpful in multiple ways. It can offer insight into what may contribute to the observed measurements, as well as offer a way to compare the (deconstructed) time series. Models that consist of deconstructed components can often also be used to predict future points by combining the components.

Another way to specify time series is by making using of an autoregressive integrated

moving average (ARIMA) model. An ARIMA model assumes the time series can be deconstructed into an autoregressive, integrated, moving average and random component. ARIMA models are generally denoted by ARIMA(p, d, q) where the parameters indicate the order of the different components: p denotes the autoregressive model order, d denotes the degree of differencing and q denotes the order of the moving-average model.

The simplest example of a sequence is white noise. White noise is a sequence of uncorrelated random variables with a mean of zero and a finite variance. White noise is often used as a building block in constructing models for time series with dependent random variables.

The moving average model, denoted $MA(q)$, refers to a model that describes each element of a sequence as a filter of finite duration applied to a white noise model/process. The moving average model of order q can be denoted as: $x_t = \beta_0 z_t + \beta_1 z_{t-1} + \cdots + \beta_q z_{t-q}$ where $\beta_0, \beta_1, \ldots, \beta_q$ are constants and z_t describes a white noise model.

The autoregressive model, denoted $AR(p)$, refers to a model that describes each element of a sequence as a filter of finite duration applied to its past values. The autoregressive model of order p can be denoted as: $x_t = \alpha_1 x_{t-1} + \alpha_2 x_{t-2} + \cdots + \alpha_q x_{t-p} + z_t$ where $\alpha_1, \alpha_2, \ldots, \alpha_q$ are constants and z_t describes a white noise model.

Another model for time series is obtained by combining AR and MA models. A autoregressive/moving average process denoted $ARMA(p, q)$ refers to a model can be denoted as: $x_t = \alpha_1 x_{t-1} + \alpha_1 x_{t-2} + \cdots + \alpha_q x_{t-p} + z_t$. The ARIMA model allows the extension towards a non-stationary stationary model through describing the differenced data by an ARMA model, to obtain the model for the non-stationary model the the differenced model has to be summed or 'integrated'. The ARIMA model can be further expanded upon. Two examples of this are (1) by allowing for seasonal influences and (2) by describing multidimensional time series (vectors, also known as VARIMA) instead of unidimensional time series.

4.2 Machine Learning Algorithms

In this section we describe the background and workings of different machine learning algorithms. Machine learning algorithms are algorithms that learn from data by constructing models that make predictions. The algorithms described here are fit for so-called supervised regression learning. Supervised means that the algorithm learns from training examples of which the correct output is known. It infers a function that can be used on new examples. Regression means that the predictions made (the output of the inferred function) are continuous.

4.2.1 Support Vector Machines

Support vector machines is a class of algorithms that build a model that represents the training data as points in space, mapped in such a way that examples that are not alike are divided as much as possible. A support vector machine constructs a so called "maxed-margin hyperplane" in a high-dimensional space which has the largest distance to the

nearest training data point. One of the gimmicks of support vector machines is that the similarity is not computed by mapping every point into high-dimensional space and then determining the distance, but instead by making use of so called "kernels". These kernels are used to implicitly determine the distance between two points, and are computational much cheaper.

Of course it is almost never possible to completely separate all data: there can simply be a "grey zone" in which there are classes that are not alike overlap. If this is the case the optimization problem that searches for the maxed-margin hyperplane can be adapted by including a cost that penalizes misclassification of training data. By introducing this cost, the optimization problem becomes a trade off between the maximizing the distance between classes and splitting the data as good as possible.

Originally, support vector machines are used to separate two classes. However, support vector machines algorithms can be extended to work for regression problems. Just as with classification the same goal of finding a maximal-margin hyperplane holds. This is done in what is called ϵ-SV regression by defining a margin of tolerance, denoted ϵ, which defines how large the error should be before it is taken into consideration. Only the points that deviate more than ϵ are seen as misclassification and thus penalized in the optimization problem.

4.2.2 Regression Tree

Another algorithm that can be used for regression are regression trees. Instead of defining a global model for all data, the regression tree algorithm sub-divides the training examples into smaller partitions by building a tree using recursive partitioning. Tree building starts at the root node, which includes all data, then beginning with this node, the algorithm chooses the variable that most effectively splits the set of samples into subsets. The "most effective" splitting variable is determined by the split function. In the case of regression trees the split function that is most often used is based on analysis of variance (ANOVA) models that are often applied in statistics to analyze the differences among group means. The split function minimizes the sum of squares between groups.

The process of building a tree goes on for as long as it is impossible to split the data any further. This point is reached when the number of observations in each of the child node is below a certain predetermined threshold, all observations within a node have the identical distribution making splitting impossible or by reaching a predetermined maximum depth. The final output of the model is the sample mean of the dependent variable of the final node that is reached when traveling down through the tree.

4.2.3 Random Forest

Random forests are an ensemble learning method, which entails that it combines multiple models. The random forest algorithm constructs multiple regression trees and gives as output the mean prediction of the individual trees. It is often noted that regression trees tend to overfit the training set. One of the benefits of random forests is that they correct

for this.

The random forest algorithm constructs trees in the forest in a way which differs in two ways from the standard regression tree approach. The first difference is that, instead of using all training data, a sample with replacement is randomly selected which is used to fit an individual tree model. This process is also known as tree bagging. The second difference is that, when deciding on a split in each node, a random subset of the features is used. This is also sometimes referred to as feature bagging. Where regression trees often have a low bias and high variance, this indicates that there is generally a lot of variability when predicting a new datapoint. The two adjustments in random forests allow for a trade off: a slight increase in bias against a reduction in variance.

5 Method

In this chapter we describe the experimental setup. We begin by discussing the steps taken to prepare the data: the process of extraction extra features from the already preprocessed dataset and the imputation of missing values. Then we expand on the machine learning algorithms: we discuss the choice for algorithms, as well as the parameters used. We also mention which software implementation of the different algorithms we used. Lastly we discuss the setup and measurement by discussing how the results are obtained and measured.

5.1 Data Preprocessing

5.1.1 Feature extraction

In data mining features, also known as variables or attributes, are very important. Features are what make machine learning algorithms work, having the right features can improve performance predictions. Obtaining features that are informative and discriminative is an important step. Often this is accomplished by transforming raw data which is very voluminous into features that better represent the underlying problem. This job was largely already carried out: the data as obtained has already been processed and consists of a number of normalized features.

By transforming and combining old variables new features were constructed. The first added feature was the average of mood up to that point. The rationale behind adding this variable is that the naive method of using the mean already is able to make predictions reasonably well, and this feature adds this 'baseline'. Another feature that was added is the weekday, which was accomplished by transforming the already available time (day) feature. Three more features were added that are supposed to capture a summary statistic of diversity in behaviour of the individual: the number of features greater than zero of the normalized features for the number of calls, SMS messages to top contacts and number of application launches.

5.1.2 Missing value imputation

Datasets contain missing values due to various reasons, such as manual data entry procedures, equipment errors and incorrect measurements. These missing values must be dealt with since some of the algorithms applied do not work when the dataset contains missing values. There are a number of ways this can be done; observations with missing values can either be removed or the missing values can be imputed. As already mentioned in the data description, the number of observations is quite limited. Therefore, the decision was made to only remove the observations of which the variable mood was missing, and impute the other values.

In Figure 4 the combinations of variables with missing values are displayed. It shows that there are a total of 1249 observations, of which 1224 are observations where the mood variable is available. 1099 observations are without any missing values. 54 observations

have missing values that include the normalized histograms, these comprise of systematically missing values: for each participant the first two days of these variables are missing. For other combinations there is no clear pattern or reason for why these values are missing.

There are multiple options to fill in the missing values, such as the median, mean or a fixed value. Another option is impute the value based on another attribute for that observation. We chose to impute the missing values with the mean of the variable per participant, using all observations of that individual. This seemed the most natural choice. Since part of the missing data are from the first days the possibility of using only the history of that participant was not available. Besides it only concerns a limited number of missing attributes; it would be a waste to completely remove these measurements.

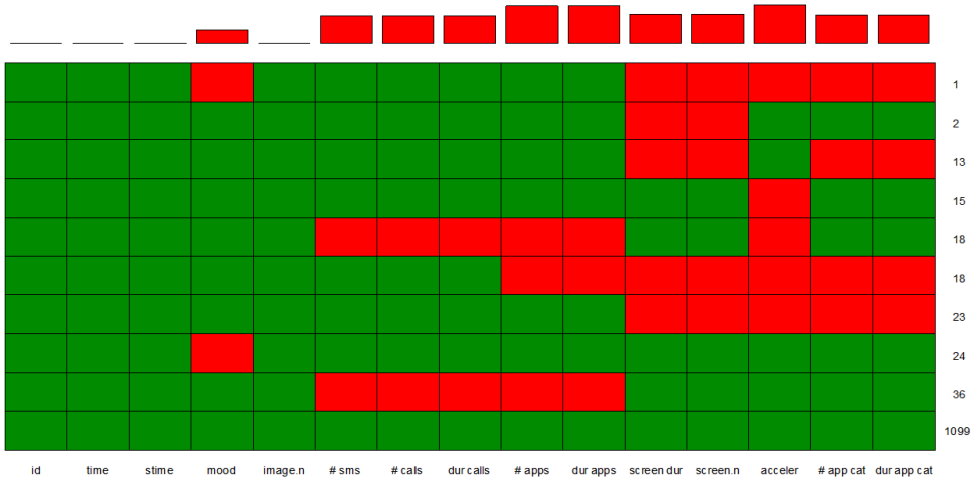


Figure 4: Combinations of missing values

5.2 Algorithms

5.2.1 Choice and parameters

Different machine learning algorithms can be used to make numerical predictions. The decision for which algorithms to implemented was made on the basis some preferred properties. Algorithms we considered preferably are widely known/supported algorithms, are not too difficult to implement, can deal with a large number of normalized features, can be used for supervised regression, do not require much parameter tuning and do not take an excessive amount of time to train models.

The machine learning algorithms that we found that fit this description are the ones previously discussed in the theory section: Regression tree, random forest and support vector machine. Besides these three algorithms the linear model with forward variable selection was implemented to compare the results with the results obtained by LiKamWa et al. [2013]. The parameters used for the algorithms were the following: For the SVM algorithm epsilon regression was applied using a radial kernel, with the cost of constraint

violation set to .5, epsilon to .1 and the sigma was determined by the built-in hyperparameter estimation heuristic function signet[Karatzoglou et al., 2004]. For the regression tree model the minimum number of observations that must exist in a node in order for a split to be attempted was set to 5, the cost complexity parameter was set to 0.5 and the minimum number of observations in any terminal leaf was set to 3. For the random forest the number of variables randomly sampled as candidates at each split was set to the number of columns divided by 3, the number of trees to grow to 500. Fitting 1170 models without parameter tuning takes a considerable amount of time, thus no further parameter tuning on a per model basis was done, as this would further increase the computation times.

For the time series analysis part the dynamic time warping algorithm was applied with the following parameters. The "Sakoe-Chiba" band was used with a maximum window size of 6. Since the DTW algorithm can not deal with missing values the missing values were imputed using the average of the variable before and after the missing value. This imputed sequence was normalised after which the algorithm was applied. The hierarchical clustering applied was done using the maximum or complete-linkage clustering method. The ARIMA models were fit based on the ACF and PACF, as well as by making use of the fitting function present the Forecast package in R using the AICc (Akaike information criterion with a correction for finite sample sizes).

5.2.2 Software

All models were constructed making use of R [R Core Team, 2014]. The implementations of the used algorithms are the following: the leaps package [Lumley, 2009] for linear models with forward selection, the rpart package [Therneau et al., 2015] for regression trees, the randomForest package [Liaw and Wiener, 2002] for random forest, for support vector machines the kernlab package[Karatzoglou et al., 2004] was used. For determining the variable importance and model training the caret package [from Jed Wing et al., 2015] was used. The time series analysis was done using the standard functionalities available in R. For fitting an ARIMA model the Forecast package[Hyndman and Khandakar, 2008] was used. For dynamic time warping the dtw package [Giorgino, 2009] was used, supported by [Tormene et al., 2008] to allow for open begin/end comparisons, the clustering of time series was done making use of the TSClust package [Montero and Vilar, 2014].

5.3 Setup and measurements

5.3.1 Performance Measures

To measure the overall performance of an algorithm the measurement that is used is the following. The algorithm is used from the third day and thus uses at least two other observations. This is to ensure there is some data to construct a model. Then, a prediction is made using this model. If in the process of fitting a model or predicting a value an error occurs (for example if no support vectors can be found) the naive method of predicting

using the mean up until that point was used. The prediction is rounded to 1 decimal, and then compared to the actual mood; if the actual mood differs with 0.5 or less (is in the interval [actual mood $-$ 0.5, actual mood + 0.5]) the prediction is considered correct, if it differs with more than 0.5 the prediction is considered incorrect. The goal is to maximize the percentage of correctly predicted moods. This matches the measure as used by LiKamWa et al. [2013]. Also the squared error was computed using the same setup of predicting from the third day of data.

The values used to make a prediction of mood that day include the measurements obtained over that day (like as the number of pictures taken). This however means that the prediction is not really 'predicting' mood of future days. This was the intended objective.

5.3.2 Variable importance

As mentioned before it would be very interesting to known which variables influence mood. For some algorithms the contribution of different variables can be estimated from constructed models. This is the case for algorithms such as the linear model, regression tree and random forest. Other algorithms however operate more in a "black-box" type manner and the constructed models are harder or even impossible to interpret. An example of this is support vector machines.

In the case of linear models there are multiple (related) ways to determine the variable importance, also depending on the way the model is constructed. The t-statistic for each model parameter can be used. This is a parameter that is often used in combination the the Akaike Information Criterion (AIC) in the stepwise construction of models. In these stepwise constructed linear models the selected variables offer a direct view on the variables containing predictive power.

The variable importance for regression trees can be derived from the variables used in each split. For random forest the same is hold, however, there is another way the contribution of each variable can be found. This is accomplished using data obtained in the sampling of the construction of the random forest; the prediction accuracy of a single tree on the out-of-bag portion of the data can be averaged over the trees to obtain a measure for importance.

For "black-box" type algorithms variable importance may be inferred. This can be achieved by removing one variable at the time, constructing a new model and comparing the results of this model to the results of the model that included this variable. However, because of random choices in the construction of these models, as well as dependence of variables and minor differences in performance, the picture this method provides in not always as clear.

6 Results and Interpretation

In this section we describe the results. We start with the results obtained from applying time series analysis techniques. Next we describe the performance of the applied methods to predict mood, including the results of the important variables in predicting mood.

6.1 Time Series Analysis Results

Different time series analysis techniques were applied to the data. The applied techniques are discussed in the theory of applied techniques section and include ARIMA and similarity analysis in combination with techniques such as clustering. These techniques were applied on all available data instead of subsets of the data, as was the case for the predictive algorithms. Time series analysis mainly served as a way to gain insight into the data.

6.1.1 ARIMA Results

We started time series analysis by deconstructing the univariate mood series of participants by representing them by ARIMA models. Per individual the autocorrelation function and partial autocorrelation functions were used to determine the order of the AR and MA processes. The autocorrelation function (ACF) is used to determine the order of the moving average process and can be seen as the cross-correlation of the time series with itself: it is the similarity between observations as a function of the time lag between. The partial autocorrelation function (PACF) is used to determine the order of the autoregressive process and gives the partial correlation of a time series with its own lagged values controlling for precedent laggs. We will look at this process for two participants; in Figure 5 the series as well as the ACF and PACF of the first and the fifth participant are plotted.

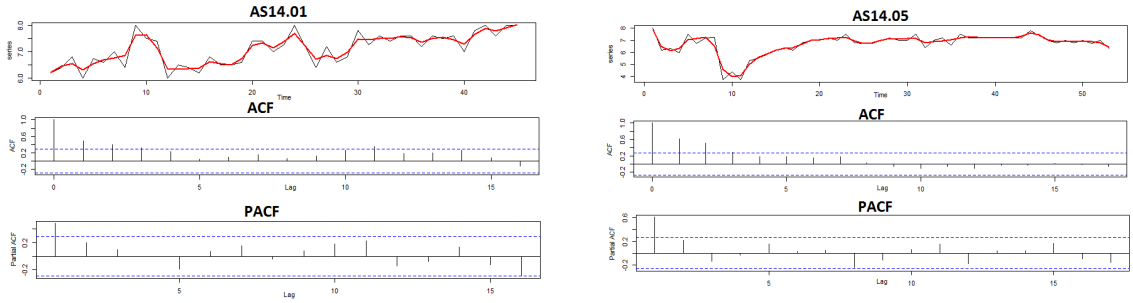


Figure 5: Participant 1 and 5 - Series with fitted spline, ACF and PACF

Figure 5 shows for participant 1 a significant value for the first ACF and a decreasing PACF, which is an indication of an MA process. For participant 5 the first lagg of the PACF shows a clear spike at the first lag, and the ACF is significant but decreasing for the first three laggs , which is an indication of an AR process. An ARIMA model was fit to both these residuals, for participant 1 this resulted in an ARIMA(0, 1, 1) model with a moving average parameter of -.6392, for participant 5 this resulted in an ARIMA(1, 1, 0)

model with an autoregressive parameter of -0.3392. The residuals (difference between the real value and the predicted value, recall that the predicted unidimensional mood ranges from 1-10) corresponding to these models can be found in Figure 6. As can be seen the residuals are quite considerable and in addition these concern individuals with quite clear cut-offs in the ACF/PACF; for other individuals the patterns of the ACF and PACF were most often not significant.

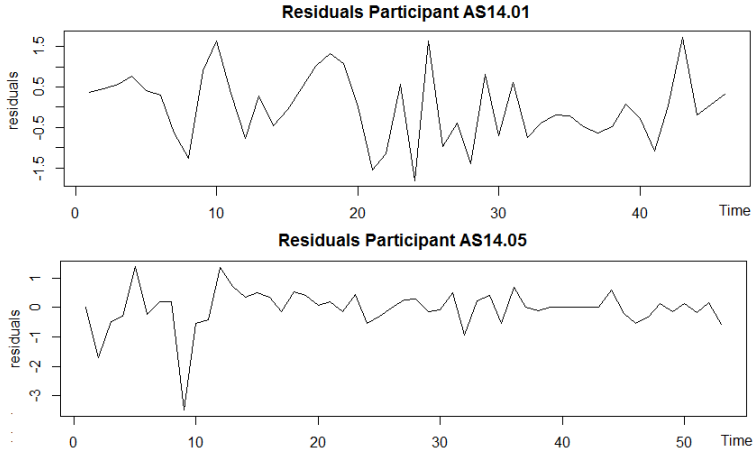


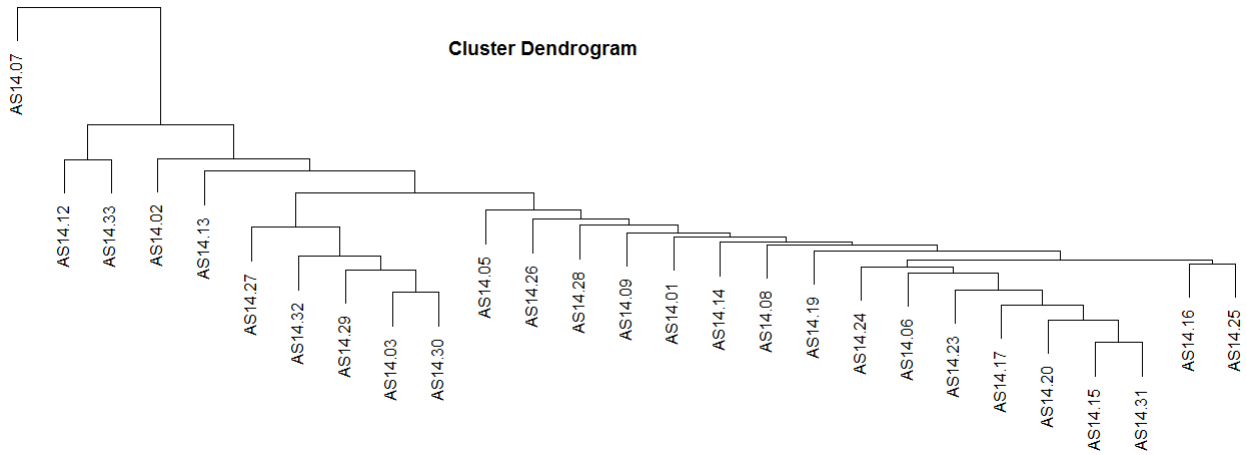
Figure 6: Residuals of the fitted ARIMA models

These models were constructed making use of all knowledge and thus offer only an indication as to whether these time series can be captured using these models. The results indicate that mood can not be represented as only as consisting of an autoregressive, moving average and integrated part.

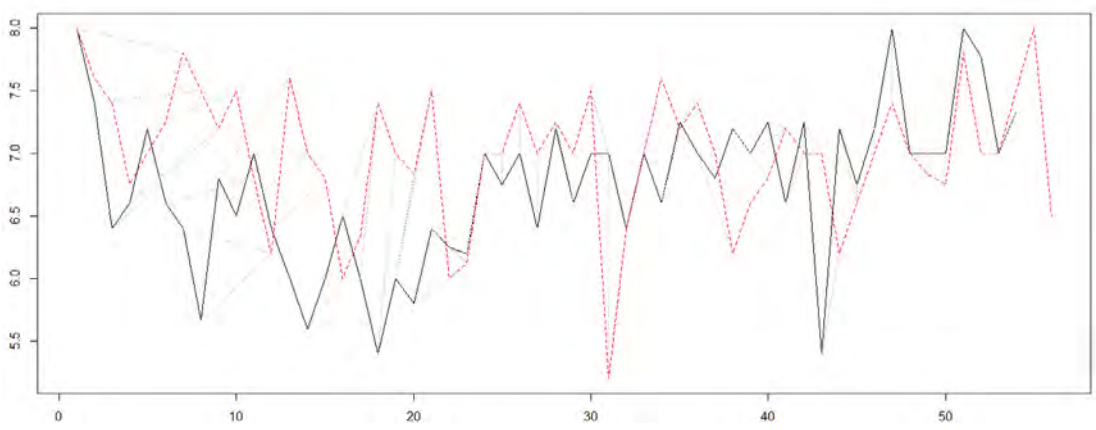
6.1.2 Similarity analysis Results

Similarity analysis was applied to the data by making use of dynamic time warping to determine the similarity of time series, for this only the univariate mood was used; subsequent results obtained with using multivariate series did not yield any clear insights and thus these results were omitted here.

As described in the method section dynamic time warping was used to determine the similarity between sequences. Using DTW the similarity among all series was evaluated, which resulted in a matrix with associated similarities. To this matrix hierarchical clustering was applied, the resulting dendogram can be found in Figure 7a. This clustering does not provide much insight. It may be concluded that the mood series of participant 7 is quite unlike the other participant. Furthermore the dendogram provides insight into how the participants are clustered and provides a quick view into how a participant compares to others participants. An example of two similar time series and the resulting dtw path is illustrated in Figure 7b, where the series of participant 16 and 24 are plotted.



(a) Dendogram of Clustered time series



(b) Similarity DTW participants 16 and 24

Figure 7: Time Series Similarity Results

6.2 Model Results

In Table 1 the results of the applied techniques to make predictions can be found. For the percentage correctly predicted the predictions are rounded before comparing them to the actual mood, this rounding is done since the mood and the measure in 0.5 in inclusive and this levels the playing field for methods that make predictions in different ways. Since here the MSE error measure is not binary the prediction is for this measure not rounded beforehand.

Technique	% Correctly predicted		Mean Squared Error	
	No added features	Added features	No added features	Added features
Mean (Naive)	64.359%	64.359%	0.441	0.441
ARIMA	63.675%	63.675%	0.475	0.475
Linear Model	60.017%	60.513%	0.890	0.903
Regression Tree	64.274%	64.615%	0.502	0.484
SVM	68.034%	68.880%	0.413	0.409
Random Forest	67.179%	68.120%	0.412	0.402

Table 1: Performance techniques

The naive predictor of the model that uses the mean can be seen as a baseline: without fitting any model already 64% is correctly predicted. The results in the second row of the table are from the extension to an ARIMA model. The ARIMA model predicts the mood on the basis of the previous moods with the assumption it consists of an autoregressive, integrated, moving average and random component. The percentage of only 63.68% show that the assumption that the mood is made up of the aforementioned components for every participant is incorrect.

The linear model that makes use of forward selection performs worse than the naive method (in contrast to the results obtained by LiKamWa et al. [2013]). The regression tree algorithm performs better than the naive predictors. The algorithm that performs best for this instance is the support vector model, which correctly predicts mood in almost 69% of the instances.

Interestingly, the features that were extracted result in an increase in performance in terms of the percentage correctly predicted for all techniques that make use of these features. The added features boost the performance of the regression tree the least with an increase of around .4% and increasing and the random forest the most with an increase close to 1%.

Considering the mean squared error of the models the results are quite similar to the results of the percentage correctly predicted. The ARIMA model performs a bit worse than the naive method, the performance of the linear model is considerably worse and both the SVM and random forest algorithms still perform notably better. The regression tree performance differs when considering the MSE: it performs worse than the naive method, also when making use of the extra features.

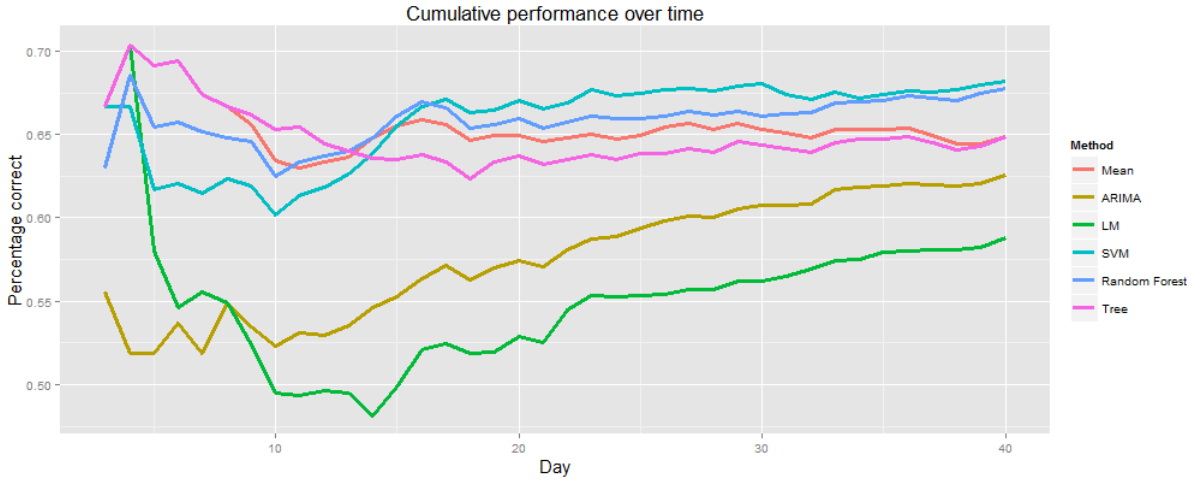


Figure 8: Performance over time

In Figure 8 the cumulative performance of the algorithms for the first 40 days is displayed. Only the first 40 days are shown since up until this day data was available for most participants. As can be seen a quite clear upward trend in performance for the all models but the regression tree that make use of the features: this is an indication that with more training data the algorithms can construct models that make better predictions. The regressions tree performs quite well in the beginning but stalls from around day 10. A possible explanation for this is that the algorithm is overfitting.

Interesting is the spike at the beginning where only minimal training data is available. We do not have a confident explanation for this, however it should be noted that performance over time is not constant and since there are not many points contributing to the cumulative performance this is likely an outlier that is more notable.

In Figure 9 the performance of some of the algorithms is compared to the naive method per participant. Everything above 0 indicates that the algorithm performs better than the naive method. As already seen in Figure 8 the linear models performs worse than the mean. For only 7 of the total of 27 participants it performs the same or better. The support vector machine performs the same or better for 20 of the participants, the Random forest 19 of the participants. For some participants (1, 5, 8, 14, 16 and 33) the models all outperform the naive method considerably, for other participants this is hardly the case (27-32) and for participant 2 and 24 the opposite holds: the algorithms perform worse than the naive method of the mean. Another interesting case is participant 26, for this participant the linear model that performs overall worse performs better than other techniques.

6.2.1 Variable importance

For the cases in which the performance of the model were better than the naive method of prediction using the mean (participants 1, 5, 8, 14, 16 and 33) the variable importance was investigated. To determine the variable importance for the models constructed by the

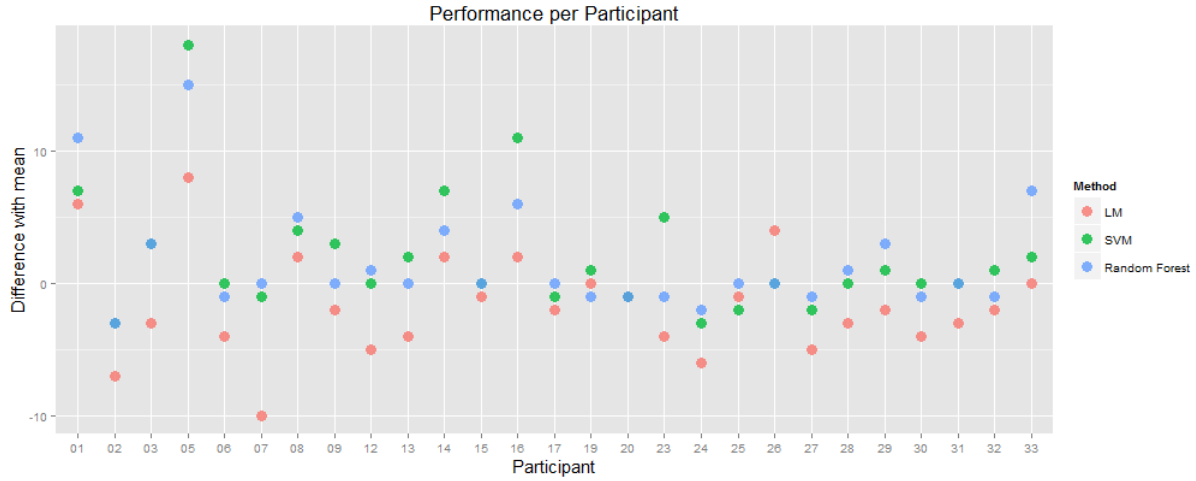


Figure 9: Performance per participant compared to mean

random forest and linear model algorithms only the model is used that uses the that uses all but the last day of data from a participant. For the SVM algorithm the performance over all days were determined as well as this performance with leaving one of the variables out.

The variable importance obtained by iterating over the variables and leaving one out to measure the difference of performance of variables in the SVM algorithm produce very unclear results. Since only a limited number of days are available per individual a difference in performance does not show very clearly. Besides, the results show that removing some variables increased the performance of the model (which indicates a possible gain to be made in feature selection), however, when removing two of these variables in many cases the performance dropped under the original level. This shows that the results are quite volatile and hence we do not take them fully into consideration.

The results of variable importance obtained for the random forest and linear model algorithms can be seen in Table 2. For the linear model the importance is based on the absolute t-statistic, for the random forest the measure is based on the difference in MSE obtained from the out-of-bag sampling. Both are scaled in such a way that the most important variable is awarded an importance of 100%.

Notable is that there is not a single prominent variable important in the models of all individuals; different variables are important for different individuals. Some variables are however often important; these are the variables average, the lagged mood variables, time and average. Interesting is that the time variable is often important; this can be interpreted that there is either a trend in mood over time or that the time can be subdivided into a parts where there is a difference in mood. This might also explain partly why the performance for the predictive algorithms in these cases is better than the naive method. Another interesting observation is the similarity of important variables per participant across methods.

Participant 1		Participant 5	
Linear Model	Random Forest	Linear Model	Random Forest
average - 100.00	average - 100	mood.l1 - 100.00	mood.l1 - 100
time - 84.54	stime - 67.06	valence.l1 - 95.34	average - 82.04
stime - 84.54	valence.l1 - 73.36	sms.c2c - 89.89	valence.l1 - 71.54
valence.l1 - 80.40	time - 65.49	screen.n - 87.50	mood.l2 - 66.65
education.n - 64.73	app.a4c - 49.49	time - 79.09	screen.n - 60.97
app.a4c - 63.88	app.a5c - 42.23	stime - 79.09	news.n - 49.68
call.c2d - 62.62	vu.n - 40.15	app.a1d - 72.86	call.c1c - 43.1
Participant 8		Participant 14	
Linear Model	Random Forest	Linear Model	Random Forest
tools.sum - 100	tools.sum - 100	time - 100	time - 100
tools.n - 98.12	tools.n - 85.48	stime - 100	average - 88.94
time - 87.73	arousal.l1 - 72.21	average - 88.91	stime - 88.67
stime - 87.73	callnumber - 59.39	valence.l1 - 71.17	app.a3c - 73.04
arousal.l1 - 86.77	app.a5c - 52.96	app.a4c - 69.1	mood.l1 - 58.36
app.a4c - 81.75	call.c1c - 29.23	mail.n - 68.35	mail.n - 54.19
arousal.l2 - 67.97	app.a5d - 26.15	android.sum - 65.55	sms.c4c - 50.87
Participant 16		Participant 33	
Linear Model	Random Forest	Linear Model	Random Forest
stime - 100	mood.l2 - 100	accelerometer - 100	average - 100
time - 100	stime - 92.24	android.sum - 96.07	android.n - 87.75
mood.l1 - 62.71	time - 86.27	app.a2d - 84.95	time - 77.2
call.c2d - 56.22	valence.l2 - 70.4	android.n - 84.26	android.sum - 72.98
call.c3c - 55.89	average - 68.19	app.a5c - 81.84	accelerometer - 70.48
sms.c4c - 54.40	entert.sum - 52.62	screen.duration - 77.89	stime - 65.06
mood.l2 - 51.20	call.c3c - 45.21	app.a3c - 75.20	app.a5c - 63.45

Table 2: Variable importance

For the first participant average is an important variable in predicting mood: it is both the most important variable for the linear model as for the random forest. This is very interesting in the fitting of an ARIMA model the best fit was found to be an integrated moving average process. A similar observation can be made for participant 5; the best ARIMA fit found consisted of autoregressive process which is in line with the importance of the lagged mood variable. For participant 8 the tools application category was found to be very important; a variable that one might expect not to play a big part in mood. For participant 14 we observe that variables that are often important for other participants are the most important variables; the lagged mood and time variables, the same holds for participant 16. For participant 14 an interesting important variable is the mail category variable. For participant 16 also the call and sms variables are important, which indicates a possible relationship between social activity and mood. For participant 33 the high accelerometer variable is important in both models; this might be interpreted as an indication that for this participant a correlation exists between his/her activity level and mood.

7 Conclusion

In this chapter we give a conclusion on the basis of the earlier posed subquestions.

Which techniques can be used to analyze and compare time series?

A range of techniques for time series analysis are discussed in the theory of applied techniques section. These techniques include the deconstruction of time series with models such as ARIMA and similarity analysis with goals such as clustering and indexing. A selection of these techniques were applied to the data.

First the unidimensional mood was represented making use of ARIMA models. Based on the autocorrelation and partial autocorrelation functions only for some participants there are hints of underlying AR and MA processes. Modelling the series for the participants for which the ACF and PACF showed significant values resulted in models with quite high residuals, indicating that the representation of the series of consisting only of autoregressive, integrated and moving average processes is too simple.

The question of how similar mood series of participants is answered; how we might view similarity is quantified by defining the normalized euclidean distance measure in combination with the dynamic time warping technique. A similarity between higher scoring 'similar' series is visible (Figure 7b) and clustering provides some insights into the dataset (Figure 7a), however, the more direct implications of these similarities are hard to put one's finger on.

Which algorithm performs best in predicting mood?

Different algorithms were used to predict mood. Before applying these algorithms some data preprocessing steps were taken. Five extra features were constructed, these are the weekday, average mood and summary variables for the normalized sms, call and application features. Missing values were removed in the cases where the mood variable was missing, other values were imputed using the mean per individual.

Machine learning algorithms were applied to construct models to make predictions. For this the support vector machine, linear model making use of forward variable selection, random forest and regression trees were used. Of these algorithms the SVM performed best predicting 68.88% correctly shortly followed by the random forest with 68.12%. This is an improvement over the 64.36% obtained by the naive method that uses the mean to predict future values. The regression tree only performs marginally better than the naive method, the linear model performs considerably worse.

When looking at the performance over time it shows that all algorithms do improve as time progresses. This is encouraging: it is an indication that with more training data the machine learning algorithms are able to make better predictions. When looking at the performance over individuals it shows that there is a varying performance. Further research is needed to improve the predictions for individuals, how this may be accomplished is discussed further in the Discussion and Future Work section.

Which variables are important in influencing mood?

To assess which variables are important in influencing mood we looked at the models that were better in predicting the future mood than the naive predictor that uses the mean. The variable importance was obtained using the out-of-bag sample for models constructed by the random forest and the t-statistic for models constructed by the linear model. There is not a single variable that is very important for all individuals; different variables play a different role depending on the individuals. Some (maybe obvious) variables are often important; these are the variables average, the lagged mood variables, time and average. Otherwise important variables differ from individual to individual: for one participant activeness and social activity seems to provide a good indication for mood (accelerometer and call variables are important) for another participant a particular application category ('tool' application category duration and frequency) was found to be important.

8 Discussion and Future Work

The aim of this study was to give insight into mood of depression patients and predict future mood. Time serie analyses techniques such as deconstruction of time series and similarity analysis resulted in interesting insights into the data, algorithms were applied to predict mood and resulted for some algorithms in a better performance than the naive predictor. The performance between algorithms was quite variable, indicating that different machine learning algorithms should be further explored. In this case the SVM performed best, however, the difference with the random forest algorithm was only marginal. The obtained results are significantly lower than the results obtained by LiKamWa et al. [2013]. In comparison the improvement over the naive predictor is very limited, possible reasons for this include the validation process: LiKamWa et al. [2013] use Leave-One-Out-Cross-Validation and therefore use future data, on top of that they describe features such as browser history and location which also come with predictive power. The performance of the machine learning algorithms do increase as more data becomes available, which is a positive sign for future research.

As already mentioned the data as obtained is already quite preprocessed; the data has been "flattened" to averages per day. This has the effect that some valuable information might be lost. With access to more 'raw' data more possibilities will be available. How this data should be preprocessed before applying the algorithms is something we believe to be important. Some of the feature engineering steps already taken give promising results, this can be built upon to further. An example is location data which may provide a huge boost in predictive power. However, this data has to be preprocessed in such a way that the algorithm can leverage this information.

Further improvements may be obtained by considering the time aspect of the data more. Presently the time aspect of the data is only considered with the lagged and average variables. Also all data is regarded with the same weight, while one would probably want to attach more value to more resent data. Leveraging knowledge of possible underlying time-dependent processes such as autoregressive or moving average processes as well as extracting predictive patterns from the data are interesting possible future challenges.

Some concerns regarding the validation of the performance of the model should be addressed. The accuracy of the model is now measured regarded as a percentage of days correctly predicted, where 'correctly' is somewhat arbitrarily defined as deviating less than 0.5. A better measure would be not make such a strong cut-off but to take into account how close the actual prediction is. Also, this measure is taken over all available data and no division of the data is made into a train and test set. This may lead to overfitting; not on the level of an individual model but over all constructed models. Out of bag sampling can not really be used here since data 'from the future' would be used. A possibility is to define a standard time window of a certain number of days and using data from these days to construct a model. This way some sampling can be done and it also ties in with the time aspect of the data, however, this introduces the problem that a big part of the data would not be used in constructing the models.

The predictions made were only in the unidimensional mood dimension for the current

day. Future research may look at the prediction on both a larger horizon (1 or even more days ahead) as well as predicting in two dimensions: valence and arousal. This might be done using two independent models (essentially in the same way as mood is predicted presently) or these might be dealt with in conjunction, looking for possible dependencies among valence, arousal and mood.

References

L. A. Clark and D. Watson. Mood and the mundane: relations between daily life events and self-reported mood. *Journal of Personality and Social Psychology*, 54(2):296–308, February 1988. ISSN 0022-3514.

Selver Demic and Sen Cheng. Modeling the Dynamics of Disease States in Depression. *PLoS ONE*, 9(10):e110358, October 2014. doi: 10.1371/journal.pone.0110358. URL `http://dx.doi.org/10.1371/journal.pone.0110358`.

Max Kuhn. Contributions from Jed Wing, Steve Weston, Andre Williams, Chris Keefer, Allan Engelhardt, Tony Cooper, Zachary Mayer, Brenton Kenkel, the R Core Team, Michael Benesty, Reynald Lescarbeau, Andrew Ziem, and Luca Scrucca. *caret: Classification and Regression Training*, 2015. URL `http://CRAN.R-project.org/package=caret`. R package version 6.0-41.

Toni Giorgino. Computing and visualizing dynamic time warping alignments in R: The dtw package. *Journal of Statistical Software*, 31(7):1–24, 2009. URL `http://www.jstatsoft.org/v31/i07/`.

Dina Q. Goldin and Paris C. Kanellakis. On similarity queries for time-series data: Constraint specification and implementation. In Ugo Montanari and Francesca Rossi, editors, *Principles and Practice of Constraint Programming — CP '95*, number 976 in Lecture Notes in Computer Science, pages 137–153. Springer Berlin Heidelberg, 1995. ISBN 978-3-540-60299-6 978-3-540-44788-7. URL `http://link.springer.com/chapter/10.1007/3-540-60299-2_9`.

Jack B. Homer and Gary B. Hirsch. System Dynamics Modeling for Public Health: Background and Opportunities. *American Journal of Public Health*, 96(3):452–458, March 2006. ISSN 0090-0036. doi: 10.2105/AJPH.2005.062059. URL `http://www.ncbi.nlm.nih.gov/pmc/articles/PMC1470525/`.

Rob J Hyndman and Yeasmin Khandakar. Automatic time series forecasting: the forecast package for R. *Journal of Statistical Software*, 26(3):1–22, 2008. URL `http://ideas.repec.org/a/jss/jstsof/27i03.html`.

Josep Lluis Arcos Joan Serrà. An Empirical Evaluation of Similarity Measures for Time Series Classification. *Knowledge-Based Systems*, 67, 2014. ISSN 0950-7051. doi: 10.1016/j.knosys.2014.04.035.

Gordon H. Bower Joseph P Forgas. The influence of mood on perceptions of social interactions. *Journal of Experimental Social Psychology*, 20(6):497–513, 1984. ISSN 0022-1031. doi: 10.1016/0022-1031(84)90040-4.

Alexandros Karatzoglou, Alex Smola, Kurt Hornik, and Achim Zeileis. kernlab – an S4 package for kernel methods in R. *Journal of Statistical Software*, 11(9):1–20, 2004. URL `http://www.jstatsoft.org/v11/i09/`.

Andy Liaw and Matthew Wiener. Classification and regression by randomforest. *R News*, 2(3):18–22, 2002. URL `http://CRAN.R-project.org/doc/Rnews/`.

Robert LiKamWa, Yunxin Liu, Nicholas D. Lane, and Lin Zhong. MoodScope: Building a Mood Sensor from Smartphone Usage Patterns. In *Proceeding of the 11th Annual International Conference on Mobile Systems, Applications, and Services*, MobiSys '13, pages 389–402, New York, NY, USA, 2013. ACM. ISBN 978-1-4503-1672-9. doi: 10.1145/2462456.2464449. URL `http://doi.acm.org/10.1145/2462456.2464449`.

Thomas Lumley. *leaps: regression subset selection*, 2009. URL `http://CRAN.R-project.org/package=leaps`. R package version 2.9.

M.C.M. de Gunst. *Statistical Models*. 1 edition, January 2013.

Pablo Montero and José A. Vilar. TSclust: An R package for time series clustering. *Journal of Statistical Software*, 62(1):1–43, 2014. URL `http://www.jstatsoft.org/v62/i01/`.

M Müller. Dynamic Time Warping. In *Information Retrieval for Music and Motion*, pages 69–84. Springer Berlin Heidelberg, 2007. ISBN 978-3-540-74047-6 978-3-540-74048-3. URL `http://link.springer.com/chapter/10.1007/978-3-540-74048-3_4`.

R Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, 2014. URL `http://www.R-project.org/`.

Chotirat Ann Ratanamahatana, Jessica Lin, Dimitrios Gunopulos, Eamonn Keogh, Michail Vlachos, and Gautam Das. Mining time series data. In *Data Mining and Knowledge Discovery Handbook*, pages 1049–1077. Springer US, 2010. URL `http://link.springer.com/chapter/10.1007/978-0-387-09823-4_56`.

Marjolijn J. Sorbi and Heleen Riper. e-Health – gezondheidszorg via internet. *Psychologie en Gezondheid*, 37(4):191–201, August 2009. ISSN 1873-1791, 1876-8741. doi: 10.1007/BF03080400. URL `http://link.springer.com/article/10.1007/BF03080400`.

Terry Therneau, Beth Atkinson, and Brian Ripley. *rpart: Recursive Partitioning and Regression Trees*, 2015. URL `http://CRAN.R-project.org/package=rpart`. R package version 4.1-9.

Shinichiro Tomitaka and Toshiaki A. Furukawa. Mathematical model for the distribution of major depressive episode durations. *BMC Research Notes*, 7(1):636, September 2014. ISSN 1756-0500. doi: 10.1186/1756-0500-7-636. URL `http://www.biomedcentral.com/1756-0500/7/636/abstract`.

Paolo Tormene, Toni Giorgino, Silvana Quaglini, and Mario Stefanelli. Matching incomplete time series with dynamic time warping: An algorithm and an application to post-stroke rehabilitation. *Artificial Intelligence in Medicine*, 45(1):11–34, 2008. doi: 10.1016/j.artmed.2008.11.007.