An analysis of bedtime procrastination using data mining techniques*

Sven van den Beukel, Erik van den Boogaard, and John (Can) Lokman**

Vrije Universiteit Amsterdam, Department of Computer Science, De Boelelaan 1105, 1081 HV Amsterdam, Netherlands
Date: 2017.05.28
Student Numbers: 2521966, 2582970, 2563162

Abstract. Bedtime procrastination – failing to go to bed at the intended time while no external circumstances prevent one from doing so [4] typically results in a lack of sleep, and it significantly affects health and well-being. From literature, we know that bedtime procrastination is influenced by many factors such as personality traits, psychological, and physiological factors. For this project, we wanted to predict *next-day bedtime procrastination* using a dataset that contains experimental data collected from surveys and activity trackers (i.e. Fitbit) of 10 participants during a period of 14 consecutive days. We used machine learning algorithms for prediction purposes, and this report details this process.

Keywords: bedtime procrastination, data mining techniques

1 Exploratory data analysis

Data was collected from two surveys using Google Forms. One *intake* survey assessed several traits (e.g. bedtime procrastination trait) and demographics. The other survey assessed several daily behaviors for 14 consecutive days. Combined goal of these surveys was to measure several psychological and physiological factors that may influence bedtime procrastination (BTP).

Although several methods exist to calculate BTP, in our study BTP is calculated as the difference between *planned bedtime* and *actual bedtime*. Figure 1 shows a scatter plot of these individual (id) bedtime procrastinations (minutes). In the scatterplot, the diverse distribution of BTP is striking. It seems that while some people adhere to their plans very accurately, the others procrastinate. In the scatterplot, an outlier is detected that is approximately 12 hours off. We reason that this could be due to a mix-up of AM/PM when reporting bedtime.

Dataset characteristics. The dataset contains experimental data collected from 10 participants who participated in a survey about personality traits and daily

* This project was carried out as part of the Data Mining Course of Dr. Mark Hoogendoorn of Vrije Universiteit Amsterdam

** The order of authors are alphabetical by last name; and each author made an equal amount of contribution to the project.

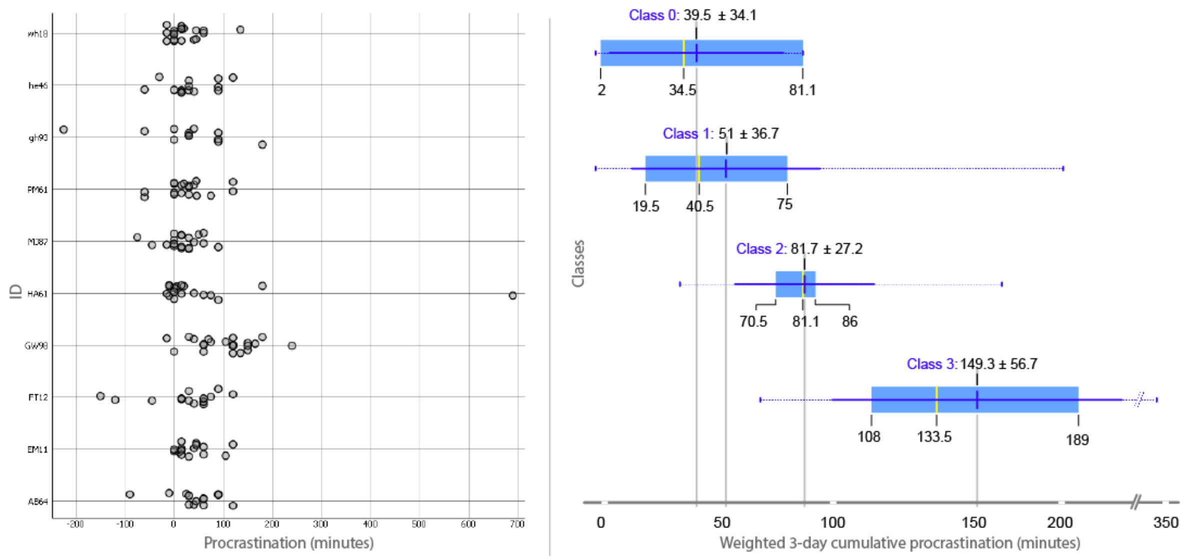

Fig. 1: *Left panel*: Scatterplot of BTP (minutes on x axis) per participant (ids on y axis) *Right panel*: Boxplot of weighted cumulative procrastination against actual procrastination classes (*Class 0*:0-15 minutes , *Class 1*:15-30 minutes, *Class 2*:30-60 minutes and *Class 3*:More than 60 minutes).

habits. Each participant filled in an intake survey and at least 14 (consecutive) daily surveys. During this period, each participant also wore an activity tracker (i.e., a Fitbit device) that recorded the amount of steps taken per day and minutes the participant was physically active. The resulting dataset contains 169 rows (10 participants x 14 days) and 100+ columns (i.e., features). A significant number of features in the dataset refer to a question such as "When you went to bed, you fell asleep within...", that could be answered using options such as *fifteen minutes, half an hour, an hour, or only after an hour.* Most of such questions were part of validated questionnaires such as Morningness-Eveningness Questionnaire [6], and were used to assess traits (e.g., *chronotype*, which refers to the tendency towards a spending one's waking hours in the morning or in the evening)[1]. After assigning scores for regular and reverse-scored items in the questionnaires (e.g. 4 points for *totally agree*, and 0 points for *totally disagree*), the scores were summed up to total questionnaire scores (e.g., an *eveningness score*, which varied between 3 or 12, with a low score indicating a *morning type* and a high score indicating an *evening type*). These scores were then added as new columns (i.e., as *new* features). Since the calculation of bedtime procrastination requires the planned bedtime of yesterday, the BTP of each participant on the first instance (first day)

[1] According to [5] chronotype, and more specifically, self-reported eveningness (evening type) is a significant predictor of bedtime procrastination, and it is an important feature in our dataset.

is unknown. No errors in data types were found and they were converted from strings to appropriate data types (i.e., floats, NaN values, and datetime objects) during parsing in Python.

Dependencies. A dependency between planned bedtime and actual bedtime can be assumed, since actual bedtime can only be later than planned bedtime. Moreover, as explained in the previous section, total questionnaire scores (e.g., chronotype) depend on other features, which are individual questions of the respective questionnaires. Finally, some dependencies can be expected to arise from operations that combine the data from previous days into new features. For instance, the *weighted 3-day cumulative BTP* in Figure 1 (left panel) depends on BTP scores of the last three days.

Exploration. During our initial exploration, we found few interesting correlations and associations. The only noteworthy correlation was between *sleep quality* and the number of times someone woke up during the night: unsurprisingly, the more one woke up during the night the less sleep quality s/he reported. The correlation between calculated BTP and self-reported procrastination (yes or no), was surprisingly low (0.12). This may indicate that people may be less aware of their bedtime procrastination. Most informative predictor for one's BTP appears to be historical BTP data of that person (e.g., cumulative BTP scores of past days).

Distributions. Figure 1 (right panel) shows the distribution of BTPs. It is remarkable that most procrastination seems to be small in duration (i.e., 15 or 30 minutes). When exploring the figure, it should be noted that we treated negative BTP scores (i.e., going to bed earlier than planned) as *zero* procrastination during the analyses[2]. The boxplot shown in Figure 1 (right panel) shows an outlier with a BTP of around 700 minutes (almost 12 hours). We believe that this is caused by a participant entering bedtime as PM instead of AM, and thus, this value was removed in the final analysis.

2 Preprocessing

Feature creation. Some features we created are based on scores of validated questionnaires used during data collection. These score-based features can be seen in Table 1.

We also created *cumulative history of last two days* and *cumulative history of last 3 days*. These *time series*-based features are listed in Table 2.

Data reduction Besides the *number of minutes of bedtime procrastination* which can be predicted using (linear) regression, we decided to also discretise this amount using 15-minute intervals (classes) in compliance with the intervals

[2] Although it could have been interesting to measure proactive or responsible behavior through negative BTP scores, this was beyond the scope of our analysis.

Feature	Description	Explanation
BTP	Bedtime procrastination	Difference between actual bedtime and planned bedtime
BPTR	Bedtime procrastination trait score	Trait, calculated questionnaire score
ATS	Aversiveness to sleep score	Trait, calculated questionnaire score
ATBR	Aversiveness to bedtime routine score	Trait, calculated questionnaire score
SQ	Sleep quality score	Self reported, calculated questionnaire score
SC	Self control score	Trait, calculated questionnaire score
TA	Task Aversiveness	Sum of ATS and ATBR
CHRON	Chronotype	Eveningness score, calculated questionnaire score
EGO	Ego depletion score	Sum of temptations experienced *and* resisted

Table 1: Features used in the analyses and their explanations.

Feature	Explanation
Bedtime procrastination (minutes)	Adds to *cumulative sleep deficit* and sleep insufficiency; these are often results from bedtime procrastination [3]
Sleep struggle	Time period (minutes) before falling asleep
Night wake	Number of times woke up during the night
Socially active	Number of social activities during the day
Daylight	Hours of daylight (based on sun hours on that day and location)
Temptation score	Experienced (and admitted) temptations for eating, coffee and total

Table 2: Features whose (cumulative) histories were calculated in the analyses.

used in questionnaires. By using discretisation, and therefore incorporating classification in our predictions —i.e., machine learning algorithms— we aimed to prevent overfitting, and more precisely predict the next-day's BTP level (or *class*).

Further preprocessing steps and creation of a preprocessing library Besides the ones mentioned above, we took a large number of additonal steps that are more fundamental in their nature (e.g., converting the dates to *datetime* objects, and then using this to calculate amount of sleep per night, and so on). For preprocessing and such initial calculations, we wrote each function from scratch in native Python. These efforts culminated in a Python library that is aimed at easy data preparation, and we encourage the reader to view this side product of the project in the Jupyter notebook at `https://github.com/clokman/DMT/blob/master/Preparation.ipynb` or at `https://goo.gl/sS8LDm` [3].

3 Methods

Benchmark. We needed a benchmark to which we can compare the performance of our pipeline. Both for classification and regression, the benchmark was an

[3] The main repository for our project can also be viewed at `https://github.com/clokman/DMT`

algorithm that predicted the next-day bedtime procrastination (BTP) to be equal to that of today. The performance measures were the same as those used in the pipeline. The results of the benchmark can be found in Table 4.

Treating N/A values and outliers. There are two types of N/A values in our dataset. Firstly, because in order to calculate the BTP, both the actual bedtime (entered today) *and* the planned bedtime (entered yesterday) are needed, the BTP for the first entry (row) of each participant could not be calculated due to lacking values from a previous day. We filtered out these first rows, discarding a total of 10 rows. Secondly, all created features that require data from previous days whose values are unknown (e.g. three-day cumulative BTP on day 2) contained N/A values. These values are replaced with the average value (or in case of a discrete variable, the most frequent one). Finally, all rows with BTP values above 4 hours were considered outliers and thus removed from the data (in practice, this led to one instance containing a BTP value of 690 being removed).

Feature ranking. Feature ranking helped minimize the number of features used by the classifier and regression algorithm. For both algorithms, the performance of several feature ranking methods were compared to each other to find out which —and how many— features were required. *Information gain* proved to select the best features for the Adaboost classifier, requiring just *three* features for optimal performance. The most informative features for regression were found using Univariate Linear Regression. The variables are listed in Table 3. Empirical testing showed that *unsupervised* approaches for data reduction (e.g. PCA, MDS) are not feasible for this dataset.

Feature	Clasification	Regression
Procrastination_minutes_three_day_weighted_cumulative_history	1	1
Procrastination_minutes_two_day_cumulative_history	2	2
Late	3	-
Procrastination_minutes_three_day_cumulative_history	-	3

Table 3: Top three selected features for each algorithm. The three-day weighted cumulative history is calculated by muliplying the BTP scores of previous 1,2, and 3 days with 1.0, 0.5, and 0.2 respectively, and then summing them. The other cumulative history variables are summed but not multiplied with weights.

Validation. Validation of our model was limited by the relatively small size of the dataset (less than 170 rows). This prevented us from creating a third validation set next to the training- and test sets. Instead, we used 10-fold stratified cross validation for validation purposes. This allowed us to use all data points for training, while limiting the required computational effort (in comparison to *leave-one-out*). The K-value (K=10) was picked because it was found to be a good balance between bias and variance [2]. Stratification is needed to ensure that the folds contain approximately the same proportions of classes as the original dataset, since the BTP values do not follow a normal distribution (as discussed in Section 2).

Prediction. Because we wanted to predict both (1) the exact amount of BTP through regression *and* (2) the class of BTP the participant would fall in each day (classification), we employed *two* prediction algorithms. To keep the results somewhat comparable, we chose to use *AdaBoost* based on a tree-classifier for both classification and regression. Adaboost is a boosting algorithm that combines weak classifiers to converge to a stronger one [1]. It first picks the best classifier. Next, the classifier that best predicts the instances that were predicted wrongly by the first classifier is added. In this manner, Adaboost builds a set of expert classifiers that show a bad individual performance, but together perform quite well. Our Adaboost algorithm uses 10 estimators. We lowered this amount to reduce the chance of overfitting. For classification, we used the *Samme.r* algorithm (which is specifically designed for classification problems with more than 2 classes) [7]. For regression, we used *linear regression*. This is because we found no significant performance gain in using an *exponential* or *square loss* function, and also due to the fact that linear regression requires the least computational effort. The learning rate was set at a fixed value of 1.

Performance measures. The performance of our classifier was measured as *accuracy*, which is the ratio of *correct* classifications. Accuracy is also reported for each class separately. For regression, we deemed RMSE (root of mean squared error) to be the most suitable performance measure, since we wanted to penalize predictions that are further off relatively more than small predictive errors.

4 Results

Table 4 shows the results of the implemented Adaboost algorithms. Column *Adaboost** shows the accuracy for the Adaboost implementation using *all* available features. The *Accuracy* row shows the *overall* accuracy of the *classification* algorithms while row *RMSE* shows the *Root Mean Squared Error* of the regression algorithms.

Class	**Benchmark**	**AdaBoost**	**Adaboost***
0-14	0.090	0.812	0.786
15-29	0.069	0.708	0.708
30-59	0.028	0.844	0.786
60-	0.083	0.870	0.825
Accuracy	**0.269**	**0.617**	**0.552**
RMSE	**64.093**	**30.865**	**31.285**

Table 4: Results of the classification (Accuracy) and regression (RMSE) algorithms

.

5 Discussion

The Adaboost algorithm performs significantly better than the benchmark. Although the features used by the algorithms depend on data that is calculated from on BTP histories, simply stating that *tomorrow's BTP* equals *todays BTP* would be an oversimplification. Our results suggest that one can predict the *class* of next-day BTP with a fair accuracy ratio of 61.7%, based on one's previous BTPs. Adding more information on the persons' characteristics surprisingly added little to no extra predictive power to both the classification- and the regression algorithm. Two out of three features used in the final classifier were created during the data processing phase. For the regression implementation of Adaboost, all three most informative features were created. The RMSE achieved (30.865) also suggests that the best predictor for today's specific behavior, is that of previous displays of the same behavior. Perhaps most notably, is that the weighted cumulation of historic BTP, which puts extra emphasis on yesterday, and less on the days before that, is the *most informative* feature for both classification and regression purposes. Feature selection successfully improved performance by removing the vast majority of features, which apparently introduced more noise than useful information to the dataset.

6 Reflection

Schedule. Table 5 shows the tasks we performed as a group in a chronological order. Exploratory data analysis and feature engineering were performed in an iterative way.

Lessons learned. Looking back on our project, we can conclude that our collaboration has been synergetic. Our collaboration took place 2-3 times a week in *scrum* sessions, during which we discussed our findings, the new features, tools, techniques and algorithms, and also interpreted results and divided tasks. Using a new tool for data visualization and analysis (i.e., Orange 3; `https://orange.biolab.si/`) posed some challenges for us because of unexpected outputs and introducing some new techniques, but me managed to overcome these problems with research. Time-series data also posed an interesting challenge but with our chosen approach - adding some historic and cumulative features - we were able to deduce some interesting new findings. We learned new skills (e.g. exploratory data analyis), new knowledge about data mining techniques (e.g. classification and K-fold cross validation) and also some domain knowledge: important and predictive factors regarding bedtime procrastination. An Improvement opportunity is that we should orient and familiarize ourselves sooner using new tools (e.g., Orange) to be able to more timely solve issues.

When	Who	Task	Observations
April w2	All	Discuss the problem and the available dataset	We concluded it is a relatively small dataset but an interesting behavioral problem, and and a challenge that involves time series data.
April w3	Erik	Compose proposal on using our own dataset and discuss approach with Mark	Our problem looks like first advanced assignment. We suggest using different techniques and feature engineering approaches.
April w4	Erik	Data pre-processing	Relatively good data quality; no missing values and only few outliers.
May w1	All	Data exploration and discussion	Discussions introduced new features e.g. cumulative values.
May w1	All	Discuss feature engineering and creation	Using new time series (cumulative) features introduced *unknown* values, which was a challenge due to recognition issues between packages.
May w2	John	Building features; creating a feature engineering and data processing library in Python	Using native Python functions to write our own data processing library, and creating new features with it was a novel and exciting challenge.
May w2	All	Select data analysis approach and tools	We decided to learn and use Anaconda's Orange package for data mining, analysis, and data visualization due to its compatibility with Python."
May w3	All	Data analysis	Using Orange was mostly intuitive but sometimes challenging because it introduced some new techniques and sometimes unclear output.
May w3	All	Select most promising features and best predictive algorithms	This was a challenging process because of different performance measuring techniques.
May w4	All	Final report (LaTeX)	LaTeX (and the 8-page limit) proved to be both fun and challenging!

Table 5: Weekly project schedule, tasks, and team members who worked on them.

References

1. Freund, Y., Schapire, R.E.: A desicion-theoretic generalization of on-line learning and an application to boosting. In: European conference on computational learning theory. pp. 23–37. Springer (1995)
2. Kohavi, R., et al.: A study of cross-validation and bootstrap for accuracy estimation and model selection. In: Ijcai. vol. 14, pp. 1137–1145. Stanford, CA (1995)
3. Kroese, F., e.a.: Bedtime procrastination: introducing a new area of procrastination. Frontiers in Psychology (2014)
4. Kroese, F., e.a.: Bedtime Procrastination - A Behavioral Perspective on Sleep Insufficiency. https://www.researchgate.net/publication/305208514 (2016)
5. Oginska, H.: Can you feel the rhythm? A short questionnaire to describe two dimensions of chronotype, Personality and Individual Differences. Volume 50, Issue 7, May 2011, Pages 1039–1043 (2011)
6. Terman, M., Terman, J.: Morningness-Eveningness Questionnaire Self-Assessment Version. Center for Environmental Therapeutics, www.cet.org (2008)
7. Zhu, J., Zou, H., Rosset, S., Hastie, T.: Multi-class adaboost. Statistics and its Interface 2(3), 349–360 (2009)