PREDICTING PROCRASTINATION FROM SMARTPHONE USE DATA USING FREQUENT SEQUENTIAL PATTERNS

BARBORA PÍSECKÁ

THESIS SUBMITTED IN PARTIAL FULFILLMENT
OF THE REQUIREMENTS FOR THE DEGREE OF
MASTER OF SCIENCE IN DATA SCIENCE & SOCIETY
AT THE SCHOOL OF HUMANITIES AND DIGITAL SCIENCES
OF TILBURG UNIVERSITY

STUDENT NUMBER

2056938

COMMITTEE

dr. A.T. Hendrickson
dr. M. Rostami Kandroodi

LOCATION

Tilburg University
School of Humanities and Digital Sciences
Department of Cognitive Science &
Artificial Intelligence
Tilburg, The Netherlands

DATE

December 2, 2022

WORD COUNT

7299

PREDICTING PROCRASTINATION FROM SMARTPHONE USE DATA USING FREQUENT SEQUENTIAL PATTERNS

BARBORA PÍSECKÁ

Abstract

As procrastination rates keep growing, researchers strive to understand its underlying mechanisms. Such understanding could help to prevent and thus limit this behaviour and its adverse effects. Multiple authors have previously investigated the relationship between smartphone use and procrastination, however, most earlier studies on procrastination relied on surveys to capture smartphone use. This method has been questioned as it rarely reflects user smartphone behaviour accurately. Thus, this thesis implemented a different approach and used smartphone logs instead. While some other authors also used phone logs for prediction of procrastination, no one has yet explored how sequential patterns of smartphone use might contribute to the prediction. Sequential patterns have previously proven to be useful for the prediction of other psychological states, such as mood or emotions. Using a dataset that consisted of smartphone logs of 231 users, this thesis tested how well sequential and non-sequential features can predict daily procrastination. This was observed for three different classifiers, namely Decision Tree, Random Forest, and XGBoost. The best performance was achieved by a combination of sequential and non-sequential features and the XGBoost classifier. Thus, evidence was found that the sequential features complement non-sequential features. While certain limitations of the analysis were identified, the results still provide a promising starting point for future research.

1 INTRODUCTION

Procrastination describes the tendency to postpone what is necessary to reach some goal (Lay, 1986). One example of procrastinating is delaying studying until the last night before the exam, even though weeks were

given to prepare. This behaviour can then lead to various negative consequences on health and well-being, such as feelings of shame, guilt, worry, and anxiety (Sirois & Pychyl, 2013). Even though there is increasing attention to the topic of procrastination, there is still relatively little known about the relationship between smartphone usage patterns and procrastination. Specifically, no research is available on the use of sequential patterns to predict procrastination. Thus, this thesis aims to contribute to the understanding of procrastination by exploring the contribution of sequential and non-sequential features to predicting daily procrastination from individuals' smartphone logs.

From a societal perspective, being able to predict an individual's level of procrastination from their smartphone usage data could contribute to a better understanding of the mechanisms behind this undesirable behaviour, as well as to a better understanding of how to prevent it. Additionally, finding a method that allows predicting procrastination using easily collectible data, such as phone logs, would be useful for developing a cost-efficient anti-procrastination application. This application could notify its users if they are about to start procrastinating and even potentially provide some exercise that could stop them from procrastinating, based on their smartphone activity. This would allow to avoid the use of sensors or other more expensive detection methods. While some phone usage features were previously used to predict procrastination, no one has ever exploited frequent sequential patterns for the prediction of procrastination. From a scientific perspective, this thesis contributes to the existing knowledge by exploring the added value of those sequential features compared to the existing approaches.

1.1 *Research Questions*

This thesis aims to answer the following question:

MQ *To what extent is it possible to predict daily procrastination from smartphone logs using sequential and non-sequential features?*

Guided by the gap in previous literature, the main goal of this thesis is to research whether sequential features could improve the prediction of procrastination. This question will be treated as a classification task with three possible outcomes (low, medium, and high daily procrastination). For this purpose, three different classifiers will be introduced; Decision Tree, Random Forest, and XGBoost; together with three different sets of features. These sets are sequential features, non-sequential features, and a combination of both. Each pair of a classifier and a feature set will be tuned using grid search with cross-validation to find the best-performing

hyperparameters. The results will then be compared to two baseline models, Majority Class and Naive Bayes classifier.

More specifically, the following sub-questions will be addressed:

SQ1 *Which combination of a feature set and a classifier achieves the highest macro F1-score in the prediction of daily procrastination?*

To answer this sub-question, the performance of all nine possible combinations of the classifier models and feature sets will be evaluated on the testing dataset and compared to each other. The comparison between models with and without sequential features will be an area of particular interest. The sequential features will consist of sequential patterns extracted from the smartphone logs per day using the cSPADE algorithm. Instead of using their original application names, each application log will be assigned to a more general category. This way, more universal rules can be derived. For non-sequential features, the frequency of a category of application and time spent on that category per day will be considered.

SQ2 *How does the prediction quality of the best-performing model differ per class?*

To prevent procrastination, it is important to build a model that reliably predicts high and medium levels of procrastination, since these are the levels that require intervention. On the other hand, misclassifying a low level of procrastination might not present such an issue (for example, an additional phone exercise is not excessively time-consuming). A confusion matrix will be observed to understand the quality of prediction per procrastination class.

SQ3 *How does the macro F1-score of the best-performing model differ for disparate groups, namely men and women?*

This will be analysed by comparing the macro F1-score values between men and women, which will help to understand whether the best-performing model is better suited for a specific gender. If the model performs better for one gender, this might affect its generalization to the rest of the population.

The main results suggest evidence in favour of including sequential features derived from smartphone logs for procrastination prediction. It is determined that the highest F1-score can be achieved by an XGBoost classifier together with the combination of sequential and non-sequential features. This model attains a macro F1-score of 33.8%. Additionally, all nine main models outperform both baseline models. Further investigation

of the best-performing model reveals that the model predicts the medium level of procrastination the best and the high level of procrastination the worst. Additionally, the disparate group analysis suggests that the model performs better for men than for women by 7.4 percentage points.

2 RELATED WORK

2.1 *Procrastination*

The topic of procrastination has been gaining popularity in the past years, as procrastination rates keep increasing over time (Steel, 2011). In daily life, this behaviour can manifest in various ways. For example, procrastination can consist of avoiding scheduling medical appointments, which can later lead to a delay in treatments, a decrease in health, and an increase in perceived stress levels (Sirois, Melia-Gordon, & Pychyl, 2003). In a work setting, procrastination can cause postponing work-related tasks, which in turn leads to job dissatisfaction, worsened performance, and more stress (Beheshtifar, Hoseinifar, & Moghadam, 2011). Students who struggle with procrastination tend to choose other activities, such as socialization and entertainment, over studying, and because of that start studying way later than would be optimal (Schouwenburg, 1995). In general, the main effects of reoccurring procrastination can be summarized as a delay in a certain activity, as well as a decrease in mental health and well-being (Rozental & Carlbring, 2014).

2.2 *Procrastination and Smartphones*

At least a part of the interest in procrastination can be attributed to the growing availability of smartphones (Steel, 2011). While previous research suggests that there is a link between procrastination and smartphones, the direction of this relationship is not completely clear.

On the one hand, some authors proposed that the use of smartphones could lead to procrastination. Since smartphones are small and can be taken anywhere and used anytime, they can easily cause distraction and postponement of the originally intended task (Oulasvirta, Rattenbury, Ma, & Raita, 2012). Meier (2022) suggested that the urge to regularly check one's smartphone significantly contributes to procrastination. Additionally, smartphone users might be interrupted from their tasks by various notifications, such as messages (Stothart, Mitchum, & Yehnert, 2015).

On the other hand, some authors suggested that smartphone use is one of the means of procrastination. Previously, different social media and communication channels have been identified as means of procrastination.

Closson and Bond (2019) concluded that Tumblr, Twitter, Pinterest, and Instagram are all common ways to procrastinate. The results of Meier, Reinecke, and Meltzer (2016) added that students also often use Facebook to procrastinate. Since smartphones are an easy way to access social media, procrastination may manifest in increased smartphone use. Further, this behaviour can also be extended to other types of applications, such as streaming (Merrill & Rubenking, 2019), Internet browsing (Thatcher, Wretschko, & Fridjhon, 2008), and gaming (Hinsch & Sheldon, 2013).

2.3 *Smartphone Use as a Predictor of Procrastination*

Previously, smartphone use has already been utilized as a predictor of procrastination. One stream of literature chose to capture smartphone use through self-reported surveys. Yang, Asbury, and Griffiths (2019) implemented paper-based surveys to capture problematic phone use (PPU), such as constantly checking for possible messages, by using psychometric scales. Based on path analysis, they concluded that academic procrastination is positively predicted by PPU. Their reported Root Mean Square Error of Approximation equaled 0.008. PPU can also be used to predict bedtime procrastination, according to Cui et al. (2021). Their conclusion was based on a cross-lagged panel model and the results of a survey collected from university students. This research achieved a Root Mean Square Error of Approximation of 0.008. Self-reported surveys were also used by Li, Gao, and Xu (2020), who made use of correlation analysis and observed that smartphone dependence contributes to academic procrastination. These three articles are closely related to the topic of this thesis but their methodology differs strongly. Since they approached the task of procrastination prediction as a regression task, their performance was still reported but it could not be directly compared to the results of this thesis.

However, all three of the above-mentioned papers face the limitations of self-reported surveys. More specifically, this method strongly depends on assuming that its participants are able and willing to report the truth about their smartphone use behaviour, which may be debatable (Davidson, Shaw, & Ellis, 2020). As a response, a new stream of literature has advocated for a switch to activity logs to collect information about smartphone use. Aalbers, vanden Abeele, Hendrickson, de Marez, and Keijsers (2022) used smartphone logs to obtain data on the total use of smartphones, use of specific types of applications, application notifications, and smartphone use fragmentation. Their analysis, which implemented a dynamic structural equation, presented evidence for an association between smartphone use patterns and procrastination. Due to the nature of the methodology, their results could not be directly compared to the results of this thesis, either.

Nonetheless, following the arguments of the authors, this thesis also adopts smartphone logs as a substitute for surveys.

2.4 *Sequential Patterns in Prediction of Psychological States*

In a more general context of psychological state prediction, a new approach to processing sequential data (such as smartphone logs) has emerged. This approach is called sequential pattern mining, and it helps to identify frequent sequences that occur in a dataset. Sequential patterns can later be used as features in prediction. Previously, Martínez and Yannakakis (2011) used gameplay data combined with physiological data, such as blood pulse, to predict the emotions of individuals. To do so, Generalized Sequential Pattern (GSP) sequence mining algorithm was implemented. Their results suggested that introducing sequential patterns improved accuracy for two of the six studied types of emotions, namely fun and challenge. Compared to a model with non-sequential features, the accuracy of fun increased by 1.48%, while the accuracy of challenge increased by 6.48%. While these results might be slightly outdated, they are still worth mentioning in the context of this project, especially because the research in this field is very limited.

While physiological data has been proven to be informative, it remains relatively invasive and costly. This might be acceptable for high-stake situations (e.g., prediction of medical outcomes), however, it is difficult to implement on a larger scale, and especially for lower-stake scenarios (Vildjiounaite et al., 2018). Since the majority of the population in emerging economies uses smartphones daily (Silver, 2019), there is little additional cost to collecting smartphone usage data, which presents another argument in favour of using smartphone logs. Alibasa, Calvo, and Yacef (2022) introduced a model that predicted an individual's mood from their smartphone usage (specifically the opened types of applications). Frequent sequential pattern features were mined by implementing GSP and SPTC (Sequential Patterns mining with Time Constraint) together with a Random Forest classifier. They concluded that sequential patterns improved the accuracy of prediction compared to models with non-sequential features, with a final accuracy of 77.8% and a macro F1-score of 55.6%.

Yet, to my knowledge, no one has exploited sequential patterns of smartphone logs for the prediction of procrastination. Following the newest developments in the literature, this thesis aims to explore whether sequential patterns could also improve prediction accuracy for procrastination.

2.5 *Sequential Pattern Mining Methods*

As mentioned previously, sequential pattern mining allows the identification of frequent sequences in a dataset. However, the first pattern mining algorithms were introduced to discover patterns in datasets without a specified order between the individual items. Probably the best-known pattern mining algorithm, Apriori, is commonly used to determine which items are often purchased together. Apriori starts by searching for individual items in the dataset that occur frequently and then continues increasing the size of the set of items until there are no larger frequent itemsets (Agrawal & Srikant, 1994).

Eventually, the Apriori logic was also extended to ordered (i.e., sequential) data. The Generalized Sequential Pattern (GSP) algorithm takes a dataset with ordered events as an input and outputs frequent sequences found in that dataset. Like Apriori, it starts with singular items and keeps adding items to the sequence until no more frequent sequences can be found. Every time before the sequence size is increased, the algorithm eliminates all non-frequent sequences (Srikant & Agrawal, 1996).

cSPADE was introduced as an alternative to the GSP algorithm. Similar to GSP, it also builds on the Apriori algorithm. However, it works with a different format of input. GSP processes data in a horizontal format, where the input consists of a set of sequences, and each of the sequences consists of a list of items. In contrast, cSPADE requires the data to be in a vertical format, which maintains the time stamp and the object in which it occurs for every sequence. cSPADE also significantly reduces the number of scans through the database compared to GSP and minimizes computational costs, and thus outperforms the previous approaches (including GSP) with respect to runtime (Zaki, 2001). That is why, contrary to Martínez and Yannakakis (2011) and Alibasa et al. (2022), cSPADE was implemented instead of GSP.

cSPADE has been previously successfully applied to a variety of topics. Some examples include extracting the flows of taxi movements (Ibrahim & Shafiq, 2019), analyzing the behaviour of self-learners in online courses (Wong, Khalil, Baars, de Koning, & Paas, 2019), and identifying diagnosis sequences of cancer patients (Wang, Hou, & Wang, 2018). Based on its accomplishment in other fields, it was chosen as the sequential pattern mining method also for this thesis.

3 METHODOLOGY

An overview of the general pipeline for the main models used in this thesis is provided in Figure 1. Here, a general walk through the pipeline

is provided. The section Experimental Setup then provides more detailed information for each step. Starting with three separate datasets (phone logs, procrastination survey, and application categories), each of them was first explored and cleaned. This ensured that all three datasets were compatible and that any inconsistencies or illegal values were removed. Eventually, all three datasets were merged into one. Next, the data was split into training and testing data. Only the training data was utilized for mining sequential patterns using cSPADE. However, the sequential and non-sequential features were later created for both testing and training data. Since these features were created per day, there was no information bleeding between the training and testing data. Next, oversampling was performed on the training dataset with the full set of features. Then each of the three classifiers was trained on each of the three feature sets (sequential features, non-sequential features, combination of both), using hyperparameters determined by a grid search. All nine models were evaluated using test data and compared based on their macro F1-score. Lastly, disparate group analysis was presented for the best-performing model.

3.1 *Baseline Models*

In this section, two baseline models are introduced: the Majority Class model and the Naive Bayes model. Their main purpose was to provide more understanding of the classification task before the main models were introduced. The Majority Class model, also called the Zero Rate Classifier, is a common choice for classification task baselines. It simply classifies all predictions as the most frequent class. From its performance, it can be understood what F1-score can be achieved by the simplest classification method. The Majority Class model was trained on the original non-oversampled data, to make sure that the majority class identification was not distorted by the synthetic observations.

The Naive Bayes model was added to further explore how much could be predicted using only the non-linear features. It makes its predictions using the posterior probability of every class and feature available in the dataset. This model assumes that all features are independent of each other, which is not likely. However, it is still observed to perform well even on classification tasks where this assumption does not hold (Rish, 2001). A grid search with 5-fold cross-validation was applied to determine the optimal value for the hyperparameter variance smoothing, which was determined to be approximately 0.0187. The possible values were 100 equally distributed values between 0 and $1e{-}9$, which is the default value of this parameter. Such a broad scale was chosen since there was not

a lot of information available on the commonly used values. This way, the chances of completely missing the right set of possible values were minimized. The Naive Bayes model was trained on the oversampled data, to ensure that it was provided with the same data as the main models (more detailed information about the oversampling process is provided in Section 4.4).

3.2 *Main Models*

In this section, more details are introduced about the three main classifiers: Decision Tree, Random Forest, and XGBoost. All these models belong to the family of tree-based classification algorithms, but each of them provides a different level of complexity. Additionally, each of the models was introduced in three variations: one with only non-sequential features, one only with sequential features, and one with both non-sequential and sequential features. This way, it was possible to observe whether a certain set of features achieved the highest F1-score in all settings or whether the results were more model-specific. Where applicable, the random state was set to 123 to ensure that replication was possible. Previous literature did not provide a clear consensus on which hyperparameters are essential for the prediction of procrastination. Therefore, the hyperparameters to be tuned were chosen based on more general literature (Mantovani et al., 2018; Probst & Boulesteix, 2017; Sommer, Sarigiannis, & Parnell, 2019). The possible hyperparameter values were then selected based on these sources and the author's own experimentation with possible values.

Starting with the Decision Tree Classifier, this model was included for its simplicity. It has an intuitive interpretation which has made it one of the most popular algorithms in Machine Learning (Turska, Jurga, & Piskorski, 2021). The Decision Tree Classifier predicts labels by simply learning decision rules from the training data. For each feature set (non-sequential, sequential, combined) hyperparameters were tuned separately so no model was disadvantaged. The tuning was done using a grid search with 5-fold cross-validation, using macro F1-score for evaluation. The possible values of *maximum depth*, which is the depth at which the tree stops splitting, were set to 2, 4, 6, and 8. The possible values of *minimum samples split*, which is the fraction of samples needed to make a split, were set to 0.2, 0.3, 0.4, and 0.5. An overview of the optimal hyperparameter values per model is provided in Table 1. The deepest decision tree was built for the task with non-sequential features, while the task with sequential features used a depth of only 2. The minimum samples split remained relatively low for all three modifications of the model, with the maximum proportion of 0.4 used by the decision tree with sequential features.

Table 1: Hyperparameter values of each model. Learning R. stands for Learning Rate, Nr. of Estimators stands for Number of Estimators. Abbreviation Non-Seq. represents Non-Sequential set of features.

		Hyperparameters		
Model		Max Depth	Min Samples Split	
Decision Tree	Non-Seq.	8	0.2	
	Sequential	2	0.4	
	Combined	6	0.2	
		Max Depth	Nr. of Estimators	
Random Forest	Non-Seq.	4	100	
	Sequential	6	150	
	Combined	4	300	
		Max Depth	Nr. of Estimators	Learning R.
XGBoost	Non-Seq.	8	200	0.1
	Sequential	8	200	0.1
	Combined	8	100	0.1

Secondly, the Random Forest Classifier was included. This model trains multiple decision trees and then takes the majority vote of those trees to determine the predicted class. This model is included since the Random Forest Classifier proved to be the best classifier for various mood states in previous research (Alibasa & Calvo, 2019; Alibasa, Calvo, & Yacef, 2019). Three hyperparameters were tuned in this model, again separately for each feature set. The possible values of *maximum depth* were set to 2, 4, 6, and 8. For the *number of estimators*, which determines the number of trees that are built, the possible values were set to 50, 100, 150, 200, 250, 300, and 350. The actual maximum depth of the models varied between 4 and 6. The number of estimators differed per model, with the lowest value (100) belonging to the model with non-sequential features and the highest value (300) belonging to the model with combined sequential and non-sequential features.

Thirdly, the eXtreme Gradient Boosting (XGBoost) Classifier was included. This was the most complex model out of the three. Unlike the Random Forest, it does not only train many independent decision trees. In XGBoost, each tree learns from its predecessors, which helps to improve its performance. This model was included because it is generally considered a state-of-the-art method for many classification tasks, as suggested by Chen and Guestrin (2016). 3 hyperparameters were tuned using a grid search with 5-fold cross validation: *maximum depth* (2, 4, 6, 8), the *number of estimators* (50, 100, 150, 200) and *learning rate*, which determines by how

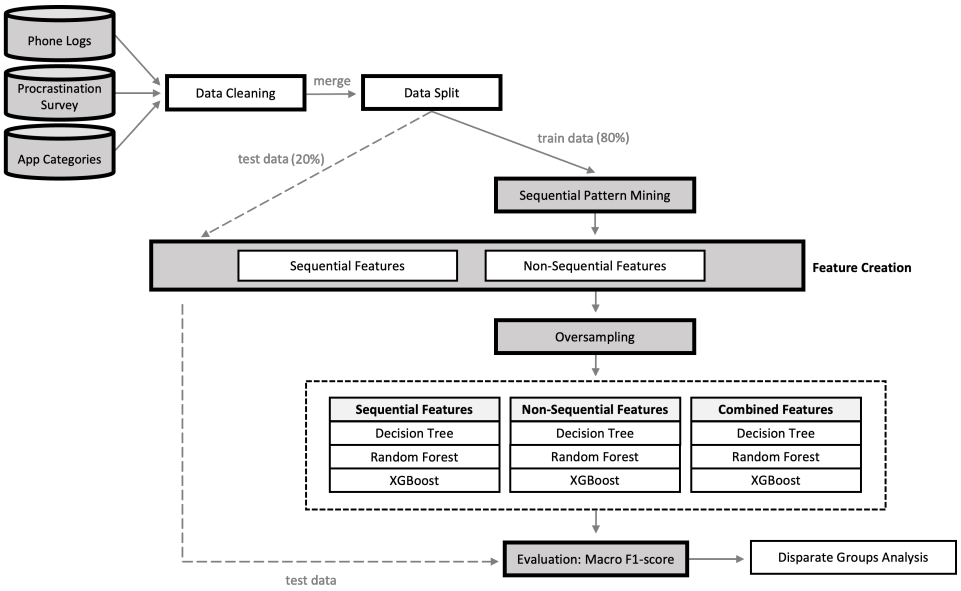


Figure 1: General data science pipeline for the main models.

much the weights of features get shrunk in each iteration (0.1, 0.01, 0.05). All three modifications of this model used very similar hyperparameter values. For all three, the maximum depth was set to 8 and the learning rate to 0.1. The only difference was in the number of estimators, which was set to 200 for the sequential and non-sequential features and 100 for the combined model.

4 EXPERIMENTAL SETUP

In this section, more details are provided about the steps introduced in Figure 1, as well as about other relevant aspects of the experimental setup.

4.1 *Dataset Description*

The data of interest was originally collected for a study of the relationship between smartphone use and mental health using the Tilburg University participation pool (for a more comprehensive description of the dataset, see Aalbers et al. (2022)). From this study, three types of data were used for this thesis. The first type was daily data on procrastination which was collected using online surveys. These surveys were distributed to the participants multiple times a day using notifications on their smartphones. Originally, 247 participants were included, however, surveys were successfully collected for only 231 of them. In total, there were 51,329 pro-

crastination answers recorded. The second type of data was smartphone use information, which was collected from participants' phones using a logging application called *mobileDNA* in the period between January 23, 2020, and June 20, 2020. For each user, there was one dataset created: this dataset contained all application logs within the given time period, including the start and end time of every application used. Altogether, there were 5,430,973 application logs collected. Thirdly, a dataset with application categories was used. This dataset contained 50 general categories; such as Social Networking, News, and Job Search; assigned to most of the application names present in the previous dataset.

4.2 *Data Cleaning and Pre-Processing*

First, all individual application logs were merged into one dataset with all participants. Next, all data was investigated and cleaned. 123,667 duplicate application logs were found and removed, as they did not carry any additional information. Additionally, one illegal application log was identified and removed. This was an application log with the end time lower than the start time. After merging application logs to the categories, 583,941 logs (belonging to 2,712 unique applications) were left without a category. The 20 most frequent missing applications (showed in Figure 2) were manually assigned to a previously existing category, which helped to regain 514,486 of the missing logs. The newly assigned categories are shown in Appendix A (page 29). The rest of the missing logs were dropped from the dataset since these applications were relatively rare and the trade-off between the information gain and the time invested in categorizing was less favourable. Moreover, observations that fell into the category Background Process were excluded from the analysis as only applications opened by the user are of interest. Variables that were not relevant to this project were dropped from the datasets and finally, procrastination survey data was merged with the application logs with categories.

4.3 *Variables*

4.3.1 *Procrastination*

The dependent variable, procrastination, was originally measured using a 7-point Likert scale. This scale reflected the extent to which participants identify with the following statement "I wasted time by doing other things than what I had intended to do." These procrastination values were first averaged per day. This step was based on Likamwa, Liu, Lane, and Zhong (2013), who suggested that daily measures better capture longer-lasting

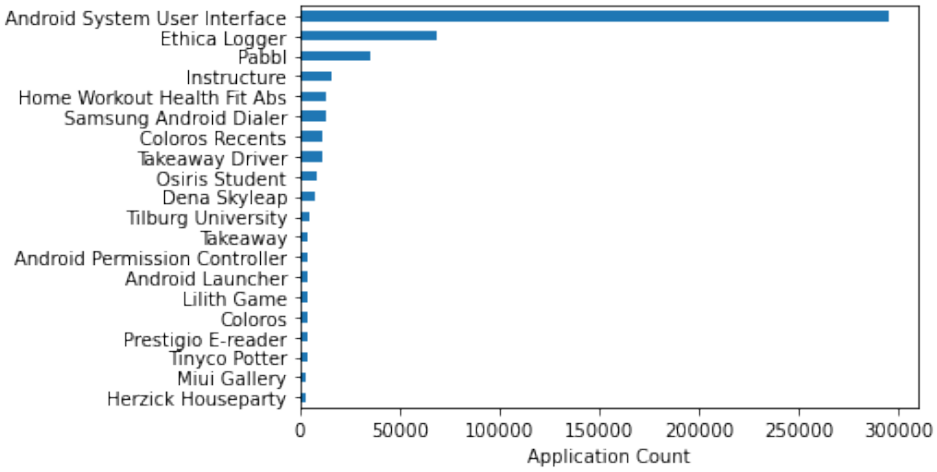


Figure 2: 20 most frequent applications with a missing category.

occurrences of psychological states. This way, the attention could shift to the more persistent cases of procrastination, instead of just isolated cases. Next, the procrastination measure was transformed into three categories: a score lower than or equal to 2 was categorized as "low procrastination"; score higher than 2 but lower than or equal to 5 was categorized as "medium procrastination"; and score higher than 5 was categorized as "high procrastination". This created classes with more intuitive interpretations and made the results easier to apply for potential procrastination detection apps. This way, major intervention could be introduced when high procrastination is predicted, medium procrastination could be treated with less urgency and no action would be required if low procrastination was predicted.

4.3.2 *Non-Sequential Features*

Since there were many unique applications included in the app logs file and some were used scarcely, the applications were previously assigned to more general categories. This allowed drawing more general conclusions considering the type of digital activity, instead of just the application name (Alibasa et al., 2022). Based on previous research on psychological state prediction, two types of non-sequential variables were constructed from the available information: duration of use per category and frequency of use per category (Alibasa & Calvo, 2019; Ciman & Wac, 2018). Including these variables helped us to understand if there was any additional value to adding sequential features and how helpful non-sequential features were by themselves.

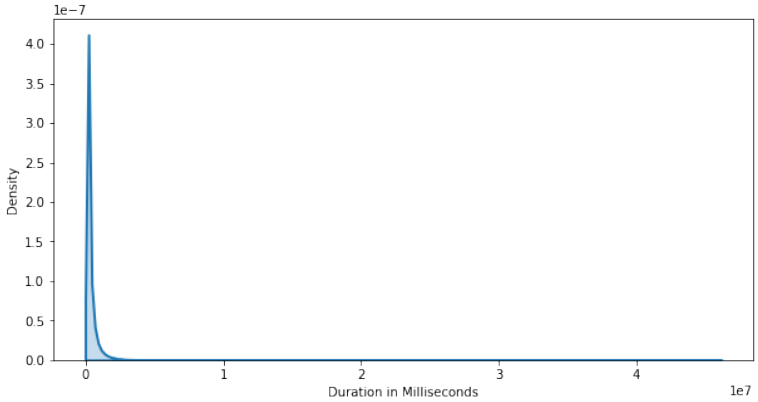


Figure 3: Density plot of duration per application log.

Firstly, the duration of use per category of application was considered. To start, duration of application per log was created. The distribution of this variable is presented in Figure 3. This figure suggested the presence of some outliers on the right side of the distribution. Further investigation suggested that most of these observations belonged to Streaming Services. It seemed plausible that some streaming service logs lasted multiple hours, and so these observations were kept in the dataset. However, four of these outlier observations belonged to Messaging and Travel Planning. While it is still possible that a participant's Messaging or Travel Planning log lasted more than nine hours, these observations were regarded as highly unlikely and so they were excluded from the analysis. Next, the durations were aggregated on application category, participant, and day. Thus, this variable showed us how much time a person spent on a certain type of activity per day.

Secondly, the frequency of use per category was created by adding up all occurrences of a certain category per participant and day. The most frequent categories were Instant Messaging, Social Networking, and Streaming Services. On the other hand, Mechanical Turk, Remote Administration, and Family Planning were among the least common ones. Since one feature was created for each application category, there were 49 features capturing the frequency of use created.

4.3.3 *Sequential Features*

Additionally, sequential features were also extracted from the data. First, frequent sequential patterns were obtained using the cSPADE algorithm on the training data. Here, the sequential patterns were observed per person per day, to match the way that procrastination was measured. Only sequences of length two or longer were considered because sequences

of length one are not different from just considering whether a certain type of application was used that day or not. There was no maximum length limitation set for the sequences. The parameter *maximum size* was set to 1, as it was assumed that applications occur sequentially, so no two observations could occur at the same time. *Support* was set to 0.95 to capture the most frequent sequences. This was the lowest support that could have been mined without running into computational constraints. These steps resulted in 187 frequent sequential patterns with a length from two to ten. Once the frequent sequences were obtained, an individual feature was created for each of the frequent sequences. This feature represented whether this specific frequent sequence was present in an observation or not.

4.4 *Oversampling*

Figure 4 illustrates the distribution of the original procrastination variable in the training dataset. The categories were noticeably imbalanced, where most observations belonged to medium procrastination and the least to high procrastination. This imbalance could lead to negligence of the least frequent classes and overemphasis on the most common classes, as the traditional classification models try to minimize error rates (Zou, Xie, Lin, Wu, & Ju, 2016). Thus, oversampling was introduced to tackle this issue: it increased the number of observations with the less common classes to reduce class imbalances. For this purpose, the SMOTE (Synthetic Minority Oversampling TEchnique) algorithm was implemented. SMOTE adds synthetic data points to the original dataset by determining the space between an observation and its nearest neighbour and then randomly selecting a point from the space in between. This way, it does not generate duplicates of the original points and introduces some variation, while maintaining the plausibility of these data points (Chawla, Bowyer, Hall, & Kegelmeyer, 2002). Only the training data was oversampled and the distribution of this newly obtained oversampled dataset is shown in Figure 4. Testing data was not oversampled at any stage, to maintain its original properties.

4.5 *Evaluation Methods*

The dataset was divided into two parts: 80 percent of the dataset was assigned to training and 20 percent to testing. In order the simulate how such an algorithm would work when used as a smartphone application where prediction is performed for the latest data, the newest data was used for testing. The validation data was not separated at this point, as cross-validation was performed later.

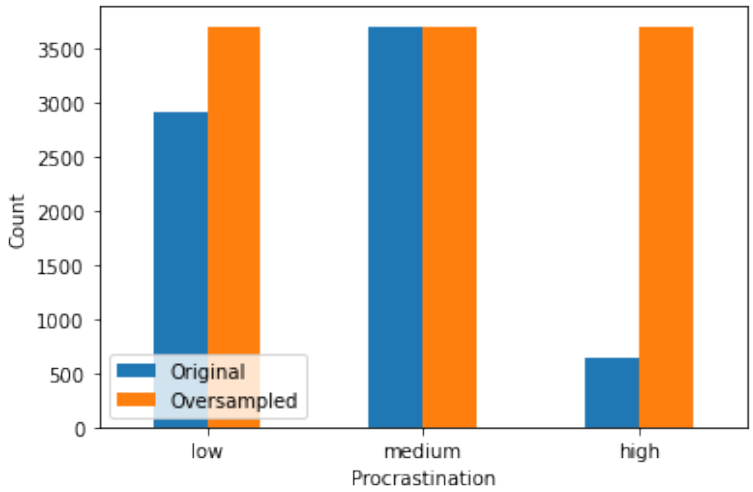


Figure 4: Comparison of the original and the oversampled training datasets.

While some related literature used accuracy as the main evaluation metric, F1-score was chosen as the main evaluation metric for this classification task. Since there was a significant imbalance in the output variable classes, a simple majority class model could still lead to relatively high accuracy. Therefore, F1-score was selected, as it is commonly used to evaluate imbalanced classification tasks (Brownlee, 2020). The macro-average was chosen since it assigns the same importance to all classes, even the more scarce ones (Narasimhan, Pan, Kar, Protopapas, & Ramaswamy, 2016).

4.6 *Algorithms and Software*

The analysis was mostly performed in a Jupyter Notebook, using Python version 3.8.8. For data cleaning, exploration, and processing Pandas version 1.2.4 (McKinney, 2010), NumPy version 1.20.1 (Harris et al., 2020), Seaborn version 0.11.1 (Waskom, 2021), and Matplotlib version 3.3.4 (Hunter, 2007) were implemented. Xgboost version 1.7.0 (Chen & Guestrin, 2016) was used for the XGBoost classifier and Sklearn version 0.24.2 (Pedregosa et al., 2011) was used for evaluation, grid search, the Naive Bayes classifier, Random Forest classifier, and Decision Tree classifier. For oversampling, imblearn version 0.9.1 (Lemaître, Nogueira, & Aridas, 2017) was applied. To identify occurrences that contained frequent sequences, re (Regular expression operations) version 2.2.1 (Van Rossum, 2020) was used. Sequential pattern mining was performed in RStudio version 4.2.1 using arulesSequences version 0.2.26 (Buchta, Hahsler, & Diaz, 2013) since R provided more support for the implementation of the cSPADE algorithm.

Table 2: Overview of the main results. The model that performed the best on the testing data is marked in bold.

			Macro F1-Score	
			Train	Test
Baselines	Majority Class		0.225	0.219
	Naive Bayes		0.383	0.270
Main Models	Decision Tree	Non-Sequential	0.451	0.293
		Sequential	0.427	0.300
		Combined	0.475	0.319
	Random Forest	Non-Sequential	0.533	0.317
		Sequential	0.562	0.308
		Combined	0.568	0.314
	XGBoost	Non-Sequential	0.568	0.304
		Sequential	0.452	0.290
		Combined	**0.534**	**0.338**

5 RESULTS

This section presents the results of the two baseline models and the nine main models. A full overview of the obtained macro F1-scores is provided in Table 2.

5.1 *Performance of the Baseline and Main Models*

The baseline models, namely the Majority Class and Naive Bayes models, achieved a relatively low performance on both the training and testing data. This illustrates the difficulty of the prediction task at hand. All variations of the Main Models achieved higher macro F1-scores than the baselines, both on the training and testing data.

Out of all the main models, the highest performance on the testing data was achieved by the XGBoost classifier with both sequential and non-sequential features. It outperformed the XGBoost classifier with non-sequential features by 3.4 percentage points and the XGBoost classifier with sequential features by 4.8 percentage points on the testing data. The second-highest macro F1-score was achieved by the Decision Tree classifier with both sequential and non-sequential features. It outperformed the non-sequential dataset by 2.6 percentage points and the sequential dataset by 1.9 percentage points. Thus, these two models provided support for the additional value of sequential patterns in procrastination prediction. Nonetheless, this result seems to be to some extent model-specific, as it was not strictly supported by the results of the Random Forest classifier. There,

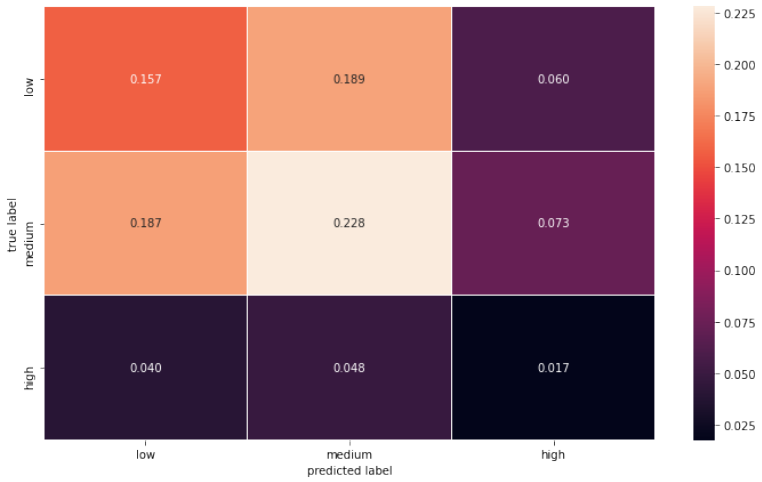


Figure 5: Multiclass confusion matrix for XGBoost classifier with sequential and non-sequential features.

the set of non-sequential features achieved a macro F1-score higher by 0.3 percentage points compared to the combined feature set. In other words, the findings might not be robust for other possible classification models. Additionally, it is also worth mentioning that the sequential feature set by itself performed worse than the non-sequential feature set on the testing data in both Random Forest and XGBoost.

5.2 *Error Analysis and Disparate Group Analysis for the Best-Performing Model*

In this section, the results of the best-performing model (XGBoost classifier with both sequential and non-sequential features) are examined more closely. Firstly, a normalized confusion matrix was observed (presented in Figure 5). The values obtained on the testing data were normalized over the total number of observations. The model succeeded the best at predicting medium procrastination, which was also the majority group in the dataset. The worst performance was observed for the high procrastination label, which was also the least common label. Thus, to some extent, the frequency of labels still affected their prediction, even after introducing oversampling. For completeness, the F1-scores per procrastination label are provided in Appendix C (page 30).

To better understand how the best-performing model made its predictions, the importance of individual features was also observed. This was done by considering the F score value for each feature used in the model. In this case, the importance of a feature was determined by calculating how many times this feature was used in a tree. The results can be found

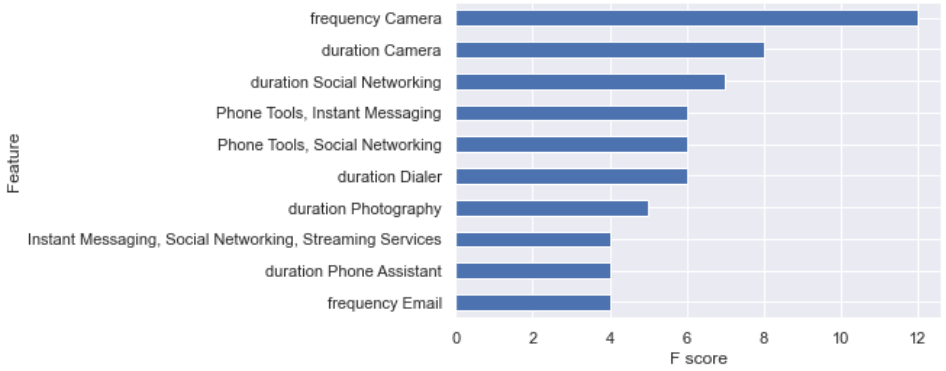
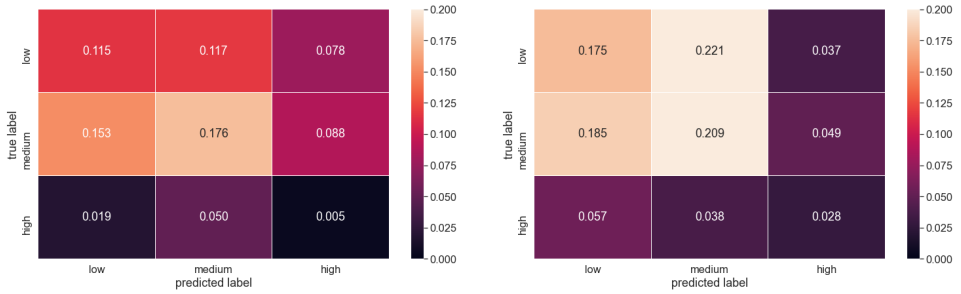


Figure 6: Feature importance for XGBoost classifier with sequential and non-sequential features.

in Figure 6. For formatting purposes, only the 10 most important features were considered. Surprisingly, the most important feature was the frequency of use of applications that fall in the Camera category, followed by the duration of use of this category. Social Networking was considered the third most important feature. Interestingly, all three types of features (frequency, duration and frequent sequences) were represented in the 10 most important features.

Next, the difference in predictions made for men and women was investigated. It is important to understand whether a model is better suited to make predictions for certain groups, as this might affect the generalization of the results. In the test set, approximately 56 percent of the observations belonged to men, while 44 percent belonged to women. The best-performing model achieved a macro F1-score of 29.6 percent for women and a macro F1-score of 37.0 percent for men. This suggests that the model is better suited for predicting the procrastination of men. This finding was also illustrated in Figure 7, which shows the confusion matrices per gender. All three labels (low, medium and high procrastination) were better predicted for men than for women.

Lastly, the results presented in Table 2 show that all models, except for the Majority Class model, achieved significantly higher macro F1-scores on the training data than on the testing data. This suggests that significant overfitting was present in those models, which was quite surprising since grid search was introduced to mitigate overfitting. One possible reason for the observed overfitting could be the chosen oversampling technique, SMOTE. Santos, Soares, Abreu, Araujo, and Santos (2018) suggest that overfitting, including the SMOTE technique, can in some cases lead to overfitting. To investigate this suspicion, the best-performing model was re-trained on the original, non-oversampled data. The hyperparameters



(a) Confusion Matrix Women

(b) Confusion Matrix Men

Figure 7: Comparison of normalized confusion matrices per gender, observed for the best-performing model (XGBoost classifier with sequential and non-sequential features).

were obtained using a grid search with the same specifications as in the original model. The specifications of this model, as well as its performance, are presented in Table 3. First, both the testing and training F1-score decreased compared to the overfitted model (the testing score decreased by approximately 10 percentage points). Secondly, the issue of overfitting disappeared. This finding indeed points to SMOTE as a possible reason for overfitting. An additional analysis of the topic of overfitting is provided in the Discussion section.

6 DISCUSSION

The main aim of this thesis was to investigate the differences in performances between models with and without sequential features for daily procrastination prediction. For sequential features, frequent sequential patterns were mined from the daily smartphone logs of the participants. Non-sequential features consisted of the frequency of use of a category per day and duration of use of a category of application. The best-performing combination of a classifier and a feature set was also determined. The

Table 3: Specifications of the non-oversampled model: XGBoost with Sequential and Non-Sequential Features. Learning R. stands for Learning Rate, Nr. of Estimators stands for Number of Estinators.

Hyperparameters	Max Depth	8
	Nr. of Estimators	200
	Learning R.	0.1
Macro F1-Score	Train	0.225
	Test	0.218

main analysis suggested promising results for the inclusion of sequential variables. While sequential patterns by themselves often achieved lower F1-scores than non-sequential features, in two of the three classifiers, the highest F1-score was achieved by a combination of the sequential and non-sequential features. The overall highest testing F1-score was observed using an XGBoost classifier with both sequential and non-sequential features. This suggests that sequential features should not be considered a substitute for non-sequential features, but they are more likely to complement each other. However, since one of the classifiers (namely the random forest classifier) achieved the highest performance with only non-sequential features, it is important to mention that the results might not extend to all types of classifiers.

Martínez and Yannakakis (2011) presented similar findings with respect to sequential and non-sequential features: for the prediction of anxiety and frustration, the combination of sequential and non-sequential features achieved the highest performance (their performance was measured in terms of accuracy). The percentage point increase in accuracy caused by using both sequential and non-sequential features was similar to the one observed in this research (in both cases it was between 1 and 7 percentage points). In addition, the authors also observed that the sequential features by themselves performed relatively worse.

The results of this thesis slightly differ from the results of Alibasa et al. (2022). Their model, which made use of sequential and non-sequential features together with a random forest classifier to predict the mood of individuals, achieved a macro F1-score of 55.6%, which was significantly higher than the macro F1-score achieved in this thesis (33.8% for the best-performing model). Their results might be better because also a more complex non-sequential feature was implemented, namely labelling applications as primary and secondary (in the case of multitasking). Indeed, once this measure was removed, the macro F1-score dropped to 34.7% which was a value really close to the F1-score achieved in this thesis. The authors also reported that the random forest generally performed the best in predicting mood. This contradicts our findings, which found the Random Forest classifier to perform the worst out of the three classifiers. It is also interesting to mention that the maximum support they found for the frequent sequential patterns was 40%, while this thesis dealt with support around 90%. These last two points might suggest that the data that was used for this thesis carries slightly different characteristics than the data used by Alibasa et al. (2022).

To answer the second sub-question, a confusion matrix was created for the best-performing model. It showed that the best performance was achieved for the medium procrastination class, while the worst perfor-

mance was achieved for the high procrastination class. In other words, the model was the most likely to miss the label that was the most important. This suggests that this specific model might not be the best option for a procrastination prevention application. This issue could be related to representation since the worst performance was observed for the least common subgroup, while the best performance was observed for the most common subgroup. Oversampling was introduced to tackle this issue; however, it was not fully successful. In the future research, this issue could be addressed in a broader way. For example, starting at the very first step, a wider sample could be considered. A larger quantity of data would also provide the possibility to employ undersampling instead of oversampling.

Additionally, it was also examined whether the best-performing model performed differently for men and women. The results suggested that the macro F1-score of the best-performing model was higher by 7.4 percentage points for men than for women. While this issue was not explicitly addressed in any of the previously mentioned papers, literature from other fields provided support for these findings. For example, Jain et al. (2018) reported that all of their nine classifier models presented in their paper perform better for emotion prediction of men than women. Their predictors consisted of various speech-derived features. In the best-performing model, the macro F1-score for men was higher by 7.4 percentage points than the one for women. One of the possible explanations could be that predicting female psychological states requires more features or more complex feature transformations. This could be addressed by creating a separate model for women; however, this is not an ideal solution for users who do not identify with the binary gender classification. Thus, the solution to this issue remains open for future research.

In conclusion, this thesis suggests that there are benefits to be gained from using sequential patterns to predict procrastination. Thus, the sequential methods used in the prediction of other psychological states seem promising for the prediction of procrastination. Nonetheless, the current model is not reliable enough for a procrastination prevention application. Thus, it serves more as a starting point for future research than a final product.

6.1 *Additional Limitations and Future Improvements*

Previously, the presence of significant overfitting was identified. One possible reason was discussed, which is that the chosen oversampling technique, SMOTE, caused oversampling. Here, an alternative (but related) explanation will be discussed in more in detail. Based on Santos et al. (2018), it is also possible that the issue was not the oversampling method

by itself, but the way the models were validated in combination with SMOTE. The data used in this thesis was split into training and testing, and later hyperparameters were selected using a grid search with 5-fold cross-validation. Thus, each time a different subsection of the training data was used as validation data. This also means that validation was performed on the oversampled data. As it was not possible to separately define validation data while using the grid search with cross-validation, this could be the cause of overfitting. This reasoning is also in line with the finding that overfitting disappeared once non-oversampled data was used. In future research, this issue could be addressed by exploring other oversampling methods (e.g., ADASYN) or different methods of validation where the validation data remains isolated for the whole time so any information transmission can be avoided.

In addition, sequential pattern mining was to some extent limited by computation and time restrictions. In future research, some additional settings of the sequential pattern mining algorithm could be explored. For example, it could be tested how different restrictions on maximum pattern lengths affect the outcomes. Alternatively, different specifications of the minimum support could be investigated. While this thesis only imposed a lower limit on the support value, potentially an upper limit could also be introduced. This would help to filter out the sequences that are so common that they are not particularly informative anymore.

7 CONCLUSION

This thesis contributed to the understanding of procrastination in the following ways. Firstly, it illustrated the possible benefits of using sequential features derived from smartphone logs as a complement to the standard non-sequential features. Secondly, it investigated which model was best suited for the prediction of daily procrastination. This was an XGBoost classifier combined with both sequential and non-sequential features, which achieved a macro F1-score of 33.8%. This model also outperformed both baseline models. Additionally, the results of the best-performing model were examined more in detail. A confusion matrix suggested that the model performed the best for medium procrastination and the worst for high procrastination. The model also showed worse performance for women than men. Lastly, even though the presented model is not reliable enough to be used for a procrastination prevention app as it is, it provides a starting point for future research. Phone logs, a less invasive and more reliable way to collect data on user phone behaviour, have a great potential for procrastination prediction in the future.

8 DATA SOURCE/CODE/ETHICS STATEMENT

Work on this thesis did not involve collecting data from human participants or animals. The original owner of the data used in this thesis retains ownership of the data during and after the completion of this thesis. The code used in this thesis will be made publicly available on GitHub. All tables and figures used in this thesis were produced by the author.

REFERENCES

Aalbers, G., vanden Abeele, M. M. P., Hendrickson, A. T., de Marez, L., & Keijsers, L. (2022). Caught in the moment: Are there person-specific associations between momentary procrastination and passively measured smartphone use? *Mobile Media & Communication*, *10*(1), 115-135. doi: 10.1177/2050157921993896

Agrawal, R., & Srikant, R. (1994). Fast algorithms for mining association rules. In *Proceedings of the 20th international conference on very large data bases* (Vol. 1215, p. 487-499).

Alibasa, M. J., & Calvo, R. A. (2019). Supporting mood introspection from digital footprints. In *8th international conference on affective computing and intelligent interaction (acii)* (p. 96-101). doi: 10.1109/ACII.2019.8925436

Alibasa, M. J., Calvo, R. A., & Yacef, K. (2019). Sequential pattern mining suggests wellbeing supportive behaviors. *IEEE Access*, *7*, 130133-130143. doi: 10.1109/ACCESS.2019.2939960

Alibasa, M. J., Calvo, R. A., & Yacef, K. (2022). Predicting mood from digital footprints using frequent sequential context patterns features. *International Journal of Human–Computer Interaction*, 1-15. doi: 10.1080/10447318.2022.2073321

Beheshtifar, M., Hoseinifar, H., & Moghadam, M. (2011). Effect procrastination on work-related stress. *European Journal of Economics, Finance and Administrative Sciences*, *38*(38), 59-64.

Brownlee, J. (2020). *Imbalanced classification with python: better metrics, balance skewed classes, cost-sensitive learning*. Machine Learning Mastery.

Buchta, C., Hahsler, M., & Diaz, D. (2013). *arulessequences: Mining frequent sequences*.

Chawla, N. V., Bowyer, K. W., Hall, L. O., & Kegelmeyer, W. P. (2002). Smote: synthetic minority over-sampling technique. *Journal of artificial intelligence research*, *16*, 321-357. doi: 10.1613/jair.953

Chen, T., & Guestrin, C. (2016). XGBoost: A scalable tree boosting system. In *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining* (pp. 785–794). ACM. doi: 10.1145/2939672

.2939785

Ciman, M., & Wac, K. (2018). Individuals' stress assessment using human-smartphone interaction analysis. *IEEE Transactions on Affective Computing*, *9*(1), 51-65. doi: 10.1109/TAFFC.2016.2592504

Closson, L. M., & Bond, T. A. (2019). Social network site use and university adjustment. *Educational Psychology*, *39*(8), 1027-1046. doi: 10.1080/01443410.2019.1618443

Cui, G., Yin, Y., Li, S., Chen, L., Liu, X., Tang, K., & Li, Y. (2021). Longitudinal relationships among problematic mobile phone use, bedtime procrastination, sleep quality and depressive symptoms in chinese college students: a cross-lagged panel analysis. *BMC Psychiatry*, *21*(1), 1-12. doi: 10.1186/s12888-021-03451-4

Davidson, B. I., Shaw, H., & Ellis, D. (2020). *Fuzzy constructs: The overlap between mental health and technology 'use'*. Retrieved from https://psyarxiv.com/6durk/

Harris, C. R., Millman, K. J., van der Walt, S. J., Gommers, R., Virtanen, P., Cournapeau, D., ... Oliphant, T. E. (2020). Array programming with NumPy. *Nature*, *585*(7825), 357–362. doi: 10.1038/s41586-020-2649-2

Hinsch, C., & Sheldon, K. M. (2013). The impact of frequent social internet consumption: Increased procrastination and lower life satisfaction. *Journal of Consumer Behaviour*, *12*(6), 496-505. doi: 10.1002/cb.1453

Hunter, J. D. (2007). Matplotlib: A 2d graphics environment. *Computing in Science & Engineering*, *9*(3), 90–95. doi: 10.1109/MCSE.2007.55

Ibrahim, R., & Shafiq, M. O. (2019). Detecting taxi movements using random swap clustering and sequential pattern mining. *Journal of Big Data*, *6*(1), 1-26.

Jain, U., Nathani, K., Ruban, N., Joseph Raj, A. N., Zhuang, Z., & G.V. Mahesh, V. (2018). Cubic svm classifier based feature extraction and emotion detection from speech signals. In *2018 international conference on sensor networks and signal processing (snsp)* (p. 386-391). doi: 10.1109/SNSP.2018.00081

Lay, C. H. (1986). At last, my research article on procrastination. *Journal of research in personality*, *20*(4), 474–495.

Lemaître, G., Nogueira, F., & Aridas, C. K. (2017). Imbalanced-learn: A python toolbox to tackle the curse of imbalanced datasets in machine learning. *Journal of Machine Learning Research*, *18*(17), 1-5. doi: 10.48550/arXiv.1609.06570

Li, L., Gao, H., & Xu, Y. (2020). The mediating and buffering effect of academic self-efficacy on the relationship between smartphone addiction and academic procrastination. *Computers & Education*, *159*, 104001. doi: 10.1016/j.compedu.2020.104001

Likamwa, R., Liu, Y., Lane, N., & Zhong, L. (2013). Moodscope: Building

a mood sensor from smartphone usage patterns.. doi: 10.1145/2462456.2464449

Mantovani, R. G., Horváth, T., Cerri, R., Junior, S. B., Vanschoren, J., & de Leon Ferreira de Carvalho, A. C. P. (2018). An empirical study on hyperparameter tuning of decision trees. *CoRR*. Retrieved from http://arxiv.org/abs/1812.02207 doi: 10.48550/arXiv.1812.02207

Martínez, H. P., & Yannakakis, G. N. (2011). Mining multimodal sequential patterns: A case study on affect detection. In *Proceedings of the 13th international conference on multimodal interfaces* (p. 3–10). doi: 10.1145/2070481.2070485

McKinney, W. (2010, 01). Data structures for statistical computing in python. In *Proceedings of the 9th python in science conference* (p. 56-61). doi: 10.25080/Majora-92bf1922-00a

Meier, A. (2022). Studying problems, not problematic usage: Do mobile checking habits increase procrastination and decrease well-being? *Mobile Media & Communication*, *10*(2), 272-293. doi: 10.1177/20501579211029326

Meier, A., Reinecke, L., & Meltzer, C. E. (2016). "facebocrastination"? predictors of using facebook for procrastination and its effects on students' well-being. *Computers in Human Behavior*, *64*, 65-76. doi: https://doi.org/10.1016/j.chb.2016.06.011

Merrill, J., K, & Rubenking, B. (2019). Go long or go often: Influences on binge watching frequency and duration among college students. *Social Sciences*, *8*(1). doi: 10.3390/socsci8010010

Narasimhan, H., Pan, W., Kar, P., Protopapas, P., & Ramaswamy, H. G. (2016). Optimizing the multiclass f-measure via biconcave programming. In *Proceedings of the 16th international conference on data mining* (p. 1101-1106). doi: 10.1109/ICDM.2016.0143

Oulasvirta, A., Rattenbury, T., Ma, L., & Raita, E. (2012). Habits make smartphone use more pervasive. *Personal and Ubiquitous computing*, *16*(1), 105-114.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., ... Duchesnay, E. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, *12*, 2825–2830. doi: 10.48550/arXiv.1201.0490

Probst, P., & Boulesteix, A.-L. (2017). To tune or not to tune the number of trees in random forest. *The Journal of Machine Learning Research*, *18*(1), 6673–6690. doi: 10.48550/arXiv.1705.05654

Rish, I. (2001). An empirical study of the naive bayes classifier. In *Ijcai 2001 workshop on empirical methods in artificial intelligence* (Vol. 3, pp. 41–46).

Rozental, A., & Carlbring, P. (2014). Understanding and treating procrastination: A review of a common self-regulatory failure. *Psychology*, *5*(13), 1488.

Santos, M. S., Soares, J. P., Abreu, P. H., Araujo, H., & Santos, J. (2018). Cross-validation for imbalanced datasets: Avoiding overoptimistic and overfitting approaches [research frontier]. *IEEE Computational Intelligence Magazine*, *13*(4), 59-76. doi: 10.1109/MCI.2018.2866730

Schouwenburg, H. C. (1995). Academic procrastination. In *Procrastination and task avoidance: Theory, research, and treatment* (p. 71-96). Boston, MA: Springer US. doi: 10.1007/978-1-4899-0227-6_4

Silver, L. (2019). In emerging economies, smartphone adoption has grown more quickly among younger generations. *Pew Research Center*.

Sirois, F., Melia-Gordon, M., & Pychyl, T. (2003). I'll look after my health, later: An investigation of procrastination and health. *Personality and Individual Differences*, *35*, 1167-1184. doi: 10.1016/S0191-8869(02)00326-4

Sirois, F., & Pychyl, T. (2013). Procrastination and the priority of short-term mood regulation: Consequences for future self. *Social and Personality Psychology Compass*, *7*(2), 115-127. doi: https://doi.org/10.1111/spc3.12011

Sommer, J., Sarigiannis, D., & Parnell, T. P. (2019). Learning to tune xgboost with xgboost. *CoRR*, *abs/1909.07218*. Retrieved from http://arxiv.org/abs/1909.07218 doi: 10.48550/arXiv.1909.07218

Srikant, R., & Agrawal, R. (1996). Mining sequential patterns: Generalizations and performance improvements. In *Proceedings of the international conference on extending database technology* (p. 1-17).

Steel, P. (2011). *The procrastination equation: How to stop putting stuff off and start getting things done*. Harper Perennial.

Stothart, C., Mitchum, A., & Yehnert, C. (2015). The attentional cost of receiving a cell phone notification. *Journal of Experimental Psychology: Human Perception and Performance*, *41*(4), 893.

Thatcher, A., Wretschko, G., & Fridjhon, P. (2008). Online flow experiences, problematic internet use and internet procrastination. *Computers in Human Behavior*, *24*(5), 2236-2254. doi: 10.1016/j.chb.2007.10.008

Turska, E., Jurga, S., & Piskorski, J. (2021). Mood disorder detection in adolescents by classification trees, random forests and xgboost in presence of missing data. *Entropy*, *23*(9), 1210. doi: 10.3390/e23091210

Van Rossum, G. (2020). *The python library reference, release 3.8.2*. Python Software Foundation.

Vildjiounaite, E., Kallio, J., Kyllönen, V., Nieminen, M., Määttänen, I., Lindholm, M., ... Gimel'farb, G. (2018). Unobtrusive stress detection

on the basis of smartphone usage data. *Personal and Ubiquitous Computing*, *22*(4), 671-688. doi: 10.1007/s00779-017-1108-z

Wang, Y., Hou, W., & Wang, F. (2018). Mining co-occurrence and sequence patterns from cancer diagnoses in new york state. *PloS one*, *13*(4), e0194407.

Waskom, M. L. (2021). seaborn: statistical data visualization. *Journal of Open Source Software*, *6*(60), 3021. doi: 10.21105/joss.03021

Wong, J., Khalil, M., Baars, M., de Koning, B. B., & Paas, F. (2019). Exploring sequences of learner activities in relation to self-regulated learning in a massive open online course. *Computers & Education*, *140*, 103595.

Yang, Z., Asbury, K., & Griffiths, M. D. (2019). An exploration of problematic smartphone use among chinese university students: Associations with academic anxiety, academic procrastination, self-regulation and subjective wellbeing. *International Journal of Mental Health and Addiction*, *17*(3), 596-614. doi: 10.1007/s11469-018-9961-1

Zaki, M. J. (2001). Spade: An efficient algorithm for mining frequent sequences. *Machine Learning*, *42*(1), 31-60.

Zou, Q., Xie, S., Lin, Z., Wu, M., & Ju, Y. (2016). Finding the best classification threshold in imbalanced classification. *Big Data Research*, *5*, 2-8. doi: 10.1016/j.bdr.2015.12.001

APPENDIX A

Table 4: 20 most frequent missing applications and the newly assigned categories.

Name of Application	Assigned Category
Herzick Houseparty	Social
Miui Gallery	Photography
Tinyco Potter	Game Singleplayer
Prestigio E-reader	Book Readers
Coloros	Phone Personalization
Lilith Game	Game Singleplayer
Android Launcher	Background Process
Android Permission Controller	Background Process
Takeaway	Food & Drinks
Tilburg University	Education
Dena Skyleap	Entertainment
Osiris Student	Education
Takeaway Driver	Productivity
Coloros Recents	Phone Personalization
Samsung Android Dialer	Background Process
Home Workout Health Fit Abs	Personal Fitness
Instructure	Education
Pabbl	Coupons
Ethica Logger	Background Process
Android System User Interface	Background Process

APPENDIX B

Table 5: 20 frequent sequences with the highest support.

Sequence	Support
Instant Messaging, Instant Messaging	0.991
Instant Messaging, Social Networking	0.991
Social Networking, Social Networking	0.991
Social Networking, Instant Messaging	0.990
Instant Messaging, Streaming Services	0.985
Streaming Services, Instant Messaging	0.984
Streaming Services, Social Networking	0.984
Social Networking, Streaming Services	0.983
Instant Messaging, Internet Browser	0.983
Social Networking, Internet Browser	0.982
Internet Browser, Social Networking	0.981
Internet Browser, Instant Messaging	0.981
Streaming Services, Streaming Services	0.976
Internet Browser, Streaming Services	0.974
Streaming Services, Internet Browser	0.974
Internet Browser, Internet Browser	0.971
Phone Tools, Instant Messaging	0.969
Instant Messaging, Phone Tools	0.968
Phone Tools, Social Networking	0.967
Social Networking, Phone Tools	0.967

APPENDIX C

Table 6: F-scores per procrastination label.

Label Procrastination	F1-Score
Low	0.40
Medium	0.48
High	0.14