Predicting perceived stress levels with phone application usage

Smartphone Usage as a Predictor for Perceived Stress

Name student: Gary van Koeverden

Student ANR: 299522

Email: g.c.vankoeverden@tilburguniversity.edu

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

Thesis committee:
Dr. A. Hendrickson
Dr. M. Jung

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

January, 2020

Abstract

Stress has a growing effect on society, predicting stress through smartphone usage seems a cost-effective and convenient method to measure stress. This study investigates the influence of different phone usage features on perceived stress levels and tests if these features can identify perceived stress levels. Five different models are tested, both user-specific and generic models. Three different classification algorithms were developed: Random Forest, Support Vector Machine and k-Nearest Neighbours. The data consisted of two different sets, one dataset that consisted of returned mental health surveys from a group of respondents and the other dataset was the phone usage log data of the same group. This data was merged and the aim of this study was to predict stress from small time frames of maximum two hours of phone usage data upon every returned survey. First an exploratory analysis was done on the different models to test which features have the strongest relation with the target stress levels. Afterwards the five models were tested with the classification algorithms. The classification results indicate that the classification algorithms do not perform better on the user-specific models as predictor for stress than on generic models. The different classification algorithms and models show very dissimilar results and predict in general not better than the baseline.

Contents

1. Introduction

This section will introduce the complete study. Section 1.1 provides a clear introductory context and in section 1.2 the problem statement and research questions are stated. Section 1.3 provides an outline for the rest of the thesis.

1.1 Context

These days everyone is extremely busy with life. We have an education, a job, a partner, a family, a social life, a hobby and if there is some time left even sports. It seems as if we have never been busier, there are not enough hours in a day to do all the things we want or need to do. Not only in our personal life we experience a tight schedule, the same is seen at work. Cramming too much activities and work into a short amount of time causes stress (Franke, 2003), which is the number one occupational disease in the Netherlands. In 2018 thirty-six percent of all the Dutch work related sick leave was due to work stress. In this same year over 1.3 million Dutch people showed signs of physical and emotional exhaustion. The total amount of stress related sick days exceeded 11 million, and the total costs of this absence exceeded 2.8 billion euros. This comes down to 8.100 euros annually per employee nationwide (TNO, 2019).

According to Hooftman et al. (2019) there is a trend visible, the amount of people that show stress symptoms has been growing since 2007. Every living creature experiences stress, but also needs it. A limited amount of stress aids the body and helps to cope with different situations. Experiencing too much stress over a longer period causes several health issues. Stress is associated with coronary heart diseases (Rosengren, et al., 2004), immune dysregulation and cancer (Godbout & Glaser, 2006), and reduces brain tissue volumes (Blix, Perski, Berglund, & Savic, 2006).

Early identification and prevention of extensive stress can be very beneficial on a personal level but it might also be interesting for organisations too. Kompier & Cooper (2003) showed that reducing the amount of stress in the workplace is a means to reduce employee costs. Furthermore it also improves the working environment within companies. Stress reduction can be very profitable on multiple levels, and a tool to measure stress might be closer than thought.

Smartphones have become integrated into everyday life, one in three Dutch people cannot even go to the toilet without bringing the smartphone along. With millennials this number is even higher: two out of three millennials bring their smartphone along to the toilet. Since the smartphone is always present in daily life, it seems to be an excellent tool to monitor daily

activities. In 2019 Dutch people daily spent on average two hours and fifteen minutes on their phone, this comes down to 34 full days of smartphone usage per year (Simyo, DirectResearch, 2019). The smartphone has become our small personal assistant, and collects lots of data. In this study it is tested if phone usage behaviour can predict stress. The goal of this study is to explore what phone usage features have an influence on the perceived stress levels of phone users, and tries to predict perceived stress levels based strictly on phone application usage in short time windows of maximal two hours. Another aim of this research is to test what kind of model the best predictor of stress is, a user-centric model or a generic model. This is done with the use of three different classification algorithms: Support Vector Machines, Random Forest and k-Nearest Neighbours.

In the past there already has been done research on phone usage as a predictor of stress. Multiple studies have focused solely on phone application usage (Ahn, Wijaya, & & Esmero, 2014; Ferdous, Osmani, & Mayora, 2015), some included Bluetooth interactions (Bogomolov, Lepri, Ferron, Pianesi, & Pentland, 2014), other also included call logs, Wifi and 3G connectivity and locational data (Sarker, Kayes, & Watters, 2019) (Kang, Seo, & & Hong, 2011). There has also been research on stress indentification with the use of smartphones and wearables (Zenonos, et al., 2016). These wearables collected data through physiological and movement sensors. These are all very pervasive methods that ask a lot of the privacy of the phone users.

From a societal point of view, predicting stress through phone usage is a very cost-effective and uncomplicated approach. The smartphone penetration in The Netherlands is 93% (Deloitte, 2019). More than 9 out of 10 people have a smartphone. Since the costs of stress have been rising over the years, an easy tool to measure and identify stress might help to fight these expenses. This cost reduction is very interesting for both the government and the industry. Also on a personal level there are benefits that derive from stress identification. This study can help identify the important phone usage features and might aid for future development of stress identification applications. From a scientific point of view, this study has an approach which has not been used before in similar research. The focus lies on small time windows and purely on the phone application data. The identification of the phone usage features that have a contribution to the performance of classification model helps narrowing down the scope for future research on stress detection through phone usage.

1.2 Problem Statement and Research Questions

Problem statement:

To what extent can perceived stress levels be predicted by phone application usage?

Sub questions:

Q1. Is there a distinction between the factors that influence the perceived stress levels for the user-specific models and the generic models?

Q2. What model predicts best the perceived stress levels of smartphone users?

Five different models will be developed to answer the research questions. Three of these models will be user-specific models and be based on the data of three randomly chosen respondents. Another model is based on the top30 respondents with the most completed survey responses and the last model is based on all respondents[1]. An exploratory data analysis will be done to answer research question 1. For research question 2 different machine learning classification models will be used to determine the optimal model for predicting stress.

The results are very differing for each of the models and the classification algorithms. The correlation between the different phone usage features and the perceived stress levels are moderate to strong for the user-generic models. While for the generic models there are no features that are strong linearly correlated. The balanced accuracy scores for the performance on the test sets for the user-specific models vary between 96.7% and 46.4%. While the generic models show less variance, these balanced accuracy scores vary between 50.6% and 61.2%.

1.3 Thesis Outline

The remaining of this thesis research paper is structured as follows: Section 2 reviews previous studies on stress detection and the relation of smartphone usage with stress. Section 3 describes the datasets that are used, feature selection and model development. The experimental results are shown in section 4 and discussed and concluded in section 5.

[1] These models are developed on a subset of the data, the respondents are also randomly selected from this subset. This is clarified in section3.3

2. Related work

This section will provide an overview of previous research on stress detection and phone usage.

2.1 Stress detection

A tight schedule, a car accident and a stock market crash all seem very different at first sight. But there is at least one thing that they have in common: stress. Even though the understanding of stress in these three situations is completely different, we still experience the same thing, stress. Since it is such a catchall term, it makes it very hard to define. Stress can be approached from different angles, and has already been addressed within a broad field of different sciences. The fields of biology, psychology and sociology all have their own definition of stress. (Epel, et al., 2018) This is mainly due to the fact that stress can be measured on multiple levels and through different ways.

There are several physiological measurement methods to analyse stress. Measuring cortisol levels (Kirschbaum, 1993), brain tissue volumes (Blix, Perski, Berglund, & Savic, 2006) and blood pressure or heart rates (Lebepe, Niezen, Hancke, & Ramotsoela, 2016) are ways to measure stress. All these are common practices and are highly reliable ways to measure stress but they do require the use of expensive equipment and technologies. Next to that, these technologies most often need professional guidance to be operated.

Wearables

Zenonos, et al. (2016) developed a framework in which they used smartphones and wearables to recognize mood states at the workplace. Four participants were monitored for a total of 11 workdays, between 09:00 – 17:00, with the use of a wristband and chest sensor. These devices were embedded with several physiological sensors that registered heartrate, pulse rate, body temperature and acceleration. Every two hours the respondents were asked to fill in a survey on their smartphone regarding their overall mood in that time frame. The researchers achieved a 62% accuracy on a generalized classification model and 69% accuracy on personalized classification models that predicted stress.

Smartphone usage

Ferdous, Osmani, & Mayora (2015) focused on stress recognition on the workfloor, and monitored the smartphones of 22 participants for over 6 weeks. In their study they collected the phone application usage of the respondents and prompted a survey three times a day. In this survey the respondents were asked what their perceived stress level was. The researchers categorized the used applications into 5 different categories: "Social Networking Service Applications", "Entertainment applications", "Utility applications", "Browser applications" and "Gaming Applications". The total amount of unique applications that were recorded during this study was 128. Participants used on average 12 different applications, with a standard deviation of 6.45. In this study the researchers used a metrics-based approach, where they used count data for classification. The features that were extracted from the data are the amount of unique applications used, time spent in every application category and the amount of times the applications from the different categories were used. The authors reached a 54% accuracy with a generic support vector machine (SVM) classification model that used the data of all the respondents combined. With the use of cross-validation and a user-centric model the researchers reached an average accuracy of 75%. The researchers showed with their work that the respondents use their smartphone disparate. In this study the approach will be sort-like, and the applications will be categorised into slightly more categories. This is due to the fact that applications are more widely used these days.

Bogomolov, Lepri, Ferron, Pianesi & Pentland (2014) have reported that they are able to recognize daily stress reliably based on smartphone data. In their research they monitored 117 participants for more than 8 weeks. During this time, all the respondents received an android smartphone with specialized software that registered call and sms data and social proximity data. With the use of Bluetooth the software scanned for near-by devices. The authors also collected daily surveys regarding the respondents personality ("Big Five" personality traits) and the experienced daily stress. The researchers argued that the weather conditions have impact on daily mood and included the daily weather conditions in the data. The authors approached the stress recognition task as a binary classification problem and reached an accuracy of 72.39% with a generic model that combined all the user's data. The transformation to a binary classification task will be adopted into this research.

Bauer & Lukowicz (2012) have shown that students show different mobile phone usage patterns when stressed. Their study focused on locational, call and SMS behaviour and left all other features out. Authors of another study managed to develop a similar model that reached an F-score of 74.2%. The participants in this study were asked to fill in a survey once a day and their smartphones were logged. Also the data of the sensors was logged, which gave insights into illuminance, acceleration and orientation of the phone, e.g., how parallel the smartphone was to the ground or how bright the room was. Two other studies on smartphone usage patterns show that stress can be better predicted for an individual than for a group of people (Bauer & Lukowicz, 2012) (Muaremi, Arnrich, & Tröster, 2013). The individual usage patterns of the smartphone users show a lot of variation. This suggests that it might be arduous to generalise phone usage to a larger population, and a user-centric model might suffice more than a generic model.

3. Experimental setup

This section provides information about the raw datasets that are used for this study. Furthermore this section will also provide an exploratory data analysis, giving a description on the features of the dataset. Thereafter the features are selected for the Machine Learning Experiments, along with the training of the classifier procedures and the performance criteria.

3.1 Dataset Description

For this study, two anonymized datasets (Hendrickson, Abeele, & Aalbers, (under review)) in Comma Separated Value (CSV) file extension have been provided. The datasets contain data about a group of people that volunteered in a research on phone usage. For a period of approximately four weeks the smartphones of these participants were logged with use of the MobileDNA application. This application logs phone usage and metadata. Next to that the respondents were regularly, with 4 times per day, asked to fill in a survey regarding their current activities, social interactions and different moods. To receive the surveys, the respondents had to install another application, Ethica. The phone log data is collected in the Phone Usage Dataset and the survey responses are collected in the Mood Survey Dataset. For the sake of clarity, in the rest of this thesis the phone usage data will be referred to as phone data and the mood usage data as mood data.

3.1.1 Dataset

The phone dataset consists of 124 unique user ID numbers. These user IDs used the applications for 586.792 times in 149.889 phone sessions between 21-02-19 and 26-3-19. The Phone Dataset consists of 586.792 rows with values that are separated by a comma. One row of data represents the log of single application that is used by a user.

Table 3.2 - 9 Features Phone Dataset

Feature	Description	Type
Application_ID	Unique name of application that is used	Character string
Battery_level	Battery level in percentage (%)	Ratio
End_time	Time that application was closed	Timestamp
End_time_MilliS	Conversion of End_Time into milliseconds	Characterstring
Notification	Whether a notification initiated the application	Binary
Session_ID	ID number for every unique phone usage session	Character string
Start_time	Time that application was opened	Timestamp
Start_time_MilliS	Conversion of Start_Time into milliseconds	Character string
User_ID	Unique ID number of respondent	Character string

The following string is an example of a row a values, extracted from the dataset. Nine variables, separated by a comma, combined in one string. Table 3.2 gives an overview of the different features in the dataset.

[com.ethica.logger, 96, 2019-03-12T12:09:11.390, 1552388951390, False, 1552388935, 2019-03-12T12:09:02.630, 1552388942630, 12345]

The mood dataset consists of 16.016 entries from 149 unique user id numbers, collected between 04-06-18 and 14-05-19. The id numbers from this dataset correspond to the id numbers of the phone data set. This file consists of 16.016 rows with values that are separated by a comma. One row of data represents a survey that is returned by a user.

Table 3.3 below gives a concise overview of the features in the mood dataset. The following string is an example of a row of values, extracted from the dataset. Thirty-five variables, combined in one string. *[12345,2019-02-22 10:39:41 CET, 2019-02-22 10:41:22 CET, 1, 4, 3, 3, 3, 4, 1, 2,5,2,3,1,0, Working ,Together with coworkers ,3,0,0,1,0,0,0,0,0,0,0,0,0,0,1,1]*

Table 3.3 35 Features Mood Dataset

Feature	Description	Type
User id	Unique ID number of respondent	Categorical
Sent_time	Time when survey was sent to respondent	Timestamp
Resp_time	Time when survey was returned by respondent	Timestamp
Duration	Duration – time used to fill in survey in minutes	Ratio
12 mood variables	12 different mood categorical variables	Categorical
Activity	Recent activity of respondent, categorised in 9 different options	Categorical
Social_enjoy	Whether the respondent liked current social setting	Categorical
6 social setting variables	The social feature transformed into dummy variables	Binary
9 activity variables	The activity feature transformed into dummy variables	Binary
Time_window	What part of day the survey was sent, day divided in 4 windows (1-4)	Categorical

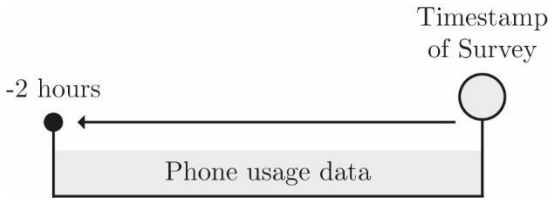
3.2 Data cleaning

The datasets both contain an amount of erroneous duplicate data. This data is removed from these datasets before the data was further pre-processed. This process contained of deleting duplicates, erroneous user ID's, expired, double, blocked and cancelled surveys. This process has decreased the total number of surveys to 6.323. The process can be found in Appendix B.

3.3 Feature engineering

The respondents were very inconsistent with returning a survey. As a result of the inconsistency the respondents frequently returned multiple completed surveys per day but also had several days of not returning a single survey. Therefor the chosen approach in this study is one that focusses on the time frame of two hours upon the surveys. For this study all the returned filled-in surveys are used as a single data measure point. A time frame of two hours was created with the survey response time as the endpoint. All the phone usage data in the two hours until the endpoint was collected. Figure shows a visualisation of this method. The surveys were approximately sent between 08:00 am and 11:00 pm. This is a total time window of 15 hours wherein the respondents can return 4 different surveys. The surveys were not sent at the exact same point of time daily, also the respondents had two hours to complete the survey. As a result of this some respondents returned multiple surveys within a very small time frame. Overlapping data is not beneficial for data analysis with relatively few observations (Harri & Brorsen, 2009). For this reason all the returned surveys from a single respondent within two hours of another survey are omitted (n=584). So there are no overlapping time frames. On the opposite, a number of respondents did not use their phone in the two hours upon replying the survey, these surveys (n=555) are also excluded since there is no corresponding phone data.

Fig 3.1 Visualisation of the process of collecting data



3.3.1 Application Categories

As a first step all the applications[2] from the phone usage dataset are categorized into 10 different categories. These categories are based on the categories in which the applications are found in the Google App Store. Application usage is based on popularity, some application categories are not used as often as others. Therefore some categories are combined, as seen in earlier research (Ferdous, Osmani, & Mayora, 2015). Table 3.4 gives an overview of the different categories.

Table 3.4 - Application Categories

Category	Feature name	Description
Entertainment	entertainment	All entertainment apps
Finance	finance	All financial and governmental apps
Games	games	All gaming apps
Lifestyle	lifestyle	All news, lifestyle and travel apps
Online shopping	onlineshopping	All online shopping apps
Process	process	All background processes
Productivity	productivity	All productivity apps
Social	social	All social media and communication[3] apps
Utility	utility	All utility apps
Wrong	wrong	Undefined/ unknown applications

3.3.2 Feature extraction

For this study there are 28 features extracted from the data. This study uses a metrics based approach, so these features are mostly based on count data. The features can be divided into three groups: application category variables, time variables and other count variables. The features are described on the following two pages.

Features

1. Total time spent in application from category.

(Ente_time ,Fina_time, Game_times, Life_time, Shop_time, Proc_time, Prod_time, Soci_time, Util_time)

These 9 features measure the total time spent in the applications from that category. The nine categories are Entertainment, Finance, Games, Lifestyle, Online shopping, Process, Productivity, Social Media and Utility. The total time is measured in seconds.

[2] The Ethica application data is included in the data and placed in the utility category.

[3] The applications from these categories are combined since there is a thin dividing line that separates the categories

2. Total count of times application used from specific category

(Ente_count, Fina_count, Game_count, Life_count, Shop_count, Proc_count ,Prod_count, Soci_count, Util_count)

These 9 features measure the total amount of times the applications from that category are used. The nine categories are Entertainment, Finance, Games, Lifestyle, Online shopping, Process, Productivity, Social Media and Utility.

3. Total amount of notification initiated sessions (noti_count)

This feature is the count of the sessions that are initiated by a notification.

4. Total time on phone (total_time_phone)

This feature is created by measuring the total amount of time that the phone has been in used. The total time is measured in seconds.

5. Session count (sess_count)

This feature is the total count of sessions in the time frame.

6. Minimum length session (min_len_sess)

This feature is the length in seconds of the shortest session in the time frame.

7. Maximum length session (max_len_sess)

This feature is the length in seconds of the longest session in the time frame.

8. Mean length session (avg_time_sess)

The mean time of a session in seconds per time frame.

9. Unique applications (uni_app_count)

Amount of unique applications that are used in a time frame.

10 Weekend (if_weekend)

A binary feature that indicates if the survey was completed during the weekend.

11 Hour of day (xhour)

Time of day in hours transformed with a trigonometric function. This is a measure to address

the cyclicity of time. The result of this transformation is that for example a value of 23 lies closer to 2 than to 20, which is the case with time. The function is formulated as:

$$xhour = \sin\left(2\,\pi\,\frac{h}{24}\right)$$

Where sin is sine, π is pi and h is the hour of the day in the 24 hours notation.

12. Hour of day (yhour)

Time of day in hours transformed with a trigonometric function. This is a measure to address the cyclicity of time. The function is formulated as:

$$yhour = \cos(2\,\pi\,\frac{h}{24})$$

Where cos is cosine, π is pi and h is the hour of the day in the 24 hours notation.

3.4 Exploratory Data Analysis

3.4.1 Target feature

Table Classes count past and present situation

Multiclass classification		Binary Classification	
Class 0	1013	Class 0	2728
Class 1	933	Class 1	1074
Class 2	782		
Class 3	607		
Class 4	374		
Class 5	93		

Binary classification task

Every person experiences and perceives stress differently. Two people can experience the same amount of stress but perceive it differently. This is also seen in the distribution of the survey answers of the respondents. Some respondents perceive stress level 1 'very slightly' as the baseline, where others perceive stress level 0 'not stressed' as their baseline.

There is only a limited amount of survey data per respondent, next to that there is a high class imbalance for a considerable amount of the respondents. Table shows the distribution of the classes. Seven respondents only perceived 1 level of stress and four respondents perceived 2 levels of stress.

There are 84 respondents that have chosen a perceived stress level category only one time in a survey, and 46 respondents that have chosen a perceived stress level two times. It is impossible for a classification model to classify with classes that have only one entry. In the case of very small classes merging is seen as a desirable solution. (Fukunaga, 2013) Classes that are similar can be merged. Therefor the six classes have been merged into two larger classes. Class 0 is 'not stressed', which is the combination of stress levels 0-1. Class 1 is 'stressed', with stress level classes 2 – 6 combined. Figure 3.4.1 displays the previous class distribution and figure 3.4.2 shows the current distribution.

Figure 3.4.1 Old class distribution

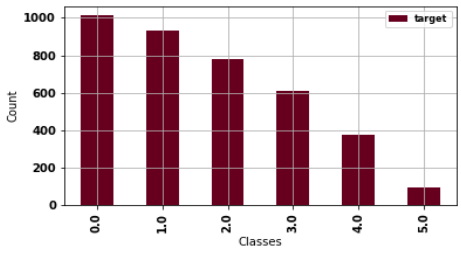
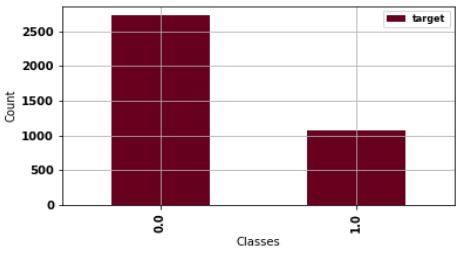


Figure.3.4.2 New class distribution



3.4.2 Application category features

The applications are placed in nine different categories and for every category there are two features that have been extracted from the data. This is the total time spent in the application and the count of the times that applications from that category have been used. The descriptive statistics of the time features are displayed in table 3.4.2.1.

Table 3.4.2.1 Descriptive statistics Categories

Feature	Count	Mean	Std	Min	25%	50%	75%	max
Ente_time	3802.00	277.96	778.13	0.00	0.00	0.00	66.00	6583.00
Fina_time	3802.00	5.39	31.94	0.00	0.00	0.00	0.00	997.00
Game_time	3802.00	47.45	273.53	0.00	0.00	0.00	0.00	4182.00
Life_time	3802.00	66.41	310.62	0.00	0.00	0.00	7.00	6334.00
Prod_time	3802.00	40.86	191.82	0.00	0.00	0.00	18.00	4976.00
Proc_time	3802.00	8.78	35.68	0.00	0.00	1.00	5.00	825.00
Shop_time	3802.00	5.93	56.65	0.00	0.00	0.00	0.00	1186.00
Soci_time	3802.00	744.69	921.18	0.00	116.00	420.00	1006.75	6749.00
Util_time	3802.00	249.40	462.00	0.00	52.25	97.00	228.00	5943.00

The descriptive statistics from table 3.4.2.1 suggest that the data is skewed. A lot of zeroes are registered. Which seems normal since the average amount of applications installed on a smartphone lies between 60 and 90 (Annie, 2017). As can be seen from table 3.4.2.2 , is the mean of uni_app_count 7.18. This indicates that that on average, around 10 percent of all applications on a phone are used in the time frames. As a result, a lot of zeroes are counted.

Table 3.4.2.2 Descriptive statistics Count Notifications & Unique Apps

Feature	Count	Mean	Std	Min	25%	50%	75%	max
Noti_count	3802.00	3.25	3.88	0.00	1.00	2.00	4.00	44.00
Uni_app_count	3802.00	7.18	3.72	1.00	4.00	7.00	10.00	25.00

As can be seen from table 3.4.2.1, the mean time spent in a category differs a lot between the categories. The mean time spent in applications from the Entertainment, Social Media and Utility categories is significantly higher than the other categories. Not only the mean time shows a lot of variation, also the total time spent in the different categories varies substantially.

Figure 3.4.2.1 Popularity category total time count

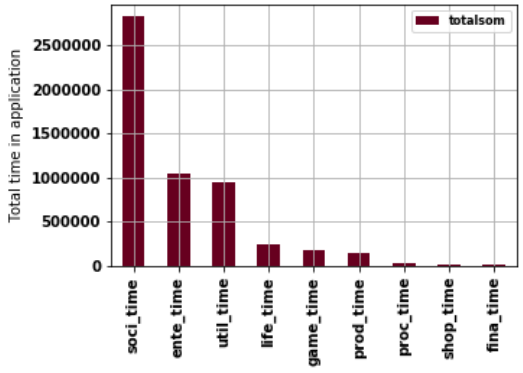


Figure 3.4.2.2 Total count application used

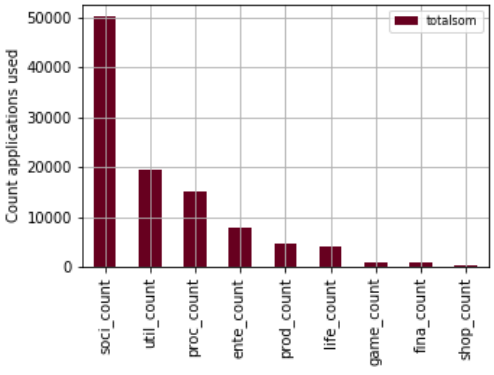


Figure 3.4.2.1 and 3.4.2.2 show the popularity of the different application categories. The social media category is the most popular one, and applications from this category are more often and longer used than the other categories combined. Social media applications are used often but for very short amount of time. This is in line with the results from the research by Oulasvirta, Rattenbury, Ma, & Raita (2012) which claims that people like to check social media regularly to receive a little informational reward.

Figure 3.4.2.3 Mean time spent per use

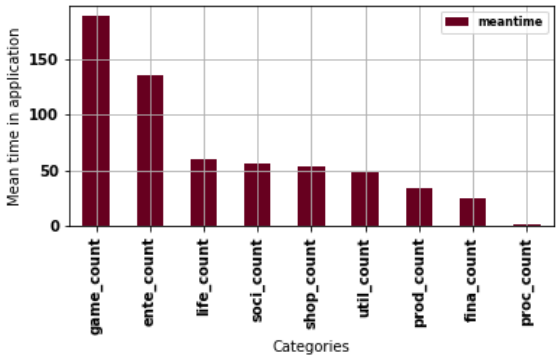


Table 3.4.2.3 displays the descriptive statistics from the features that are related to the sessions.

Table 3.4.2.3 Descriptive statistics Sessions Features

Feature	Count	Mean	Std	Min	25%	50%	75%	max
Min_len_sess	3802.00	93.35	533.89	0.00	1.00	5.00	27.00	7144.00
Max_len_sess	3802.00	782.81	1044.12	0.00	146.00	391.00	971.00	7144.00
Avg_time_sess	3802.00	285.71	624.87	0.00	55.00	107.00	246.12	7144.00
Total_time_phone	3802.00	1446.85	1444.31	0.00	317.00	963.50	2117.50	7144.00
Sess_count	3802.00	8.72	7.32	1.00	3.00	7.00	12.00	68.00

The respondents used their phone on average 8.72 times per time frame. The mean total time that a phone was used per time was 1446.31 seconds, this comes down to 24.1 minutes. After unlocking the phone, the respondents spent on average 4 minutes and 45 seconds per session.

Figure 3.4.2.4 Application category usage per model

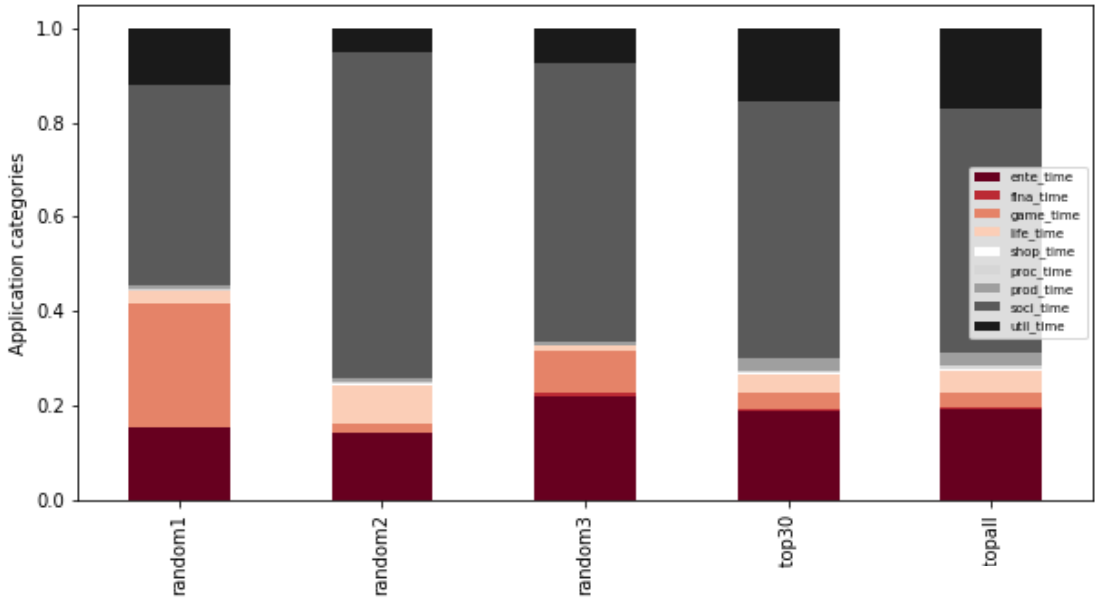


Figure 3.4.2.4 is a visualisation of the relative spent time in each application categories, based on the specific data for the earlier described models which are later used for the classification task. The distribution for the Top30 and TopAll models are fairly similar, but the individual models Random1, Random2 and Random3 show less similarity.

3.5 Data split

The performance of the different models needs to be tested. To test the performance, the researcher needs different data sets for training and testing. The creation of these datasets is done by splitting. The data for all the different models will be split into two sets, a training set (80%) and a test set (20%). This data is first randomly shuffled with the use of the shuffle function from the sklearn library, after this the data is split with the stratified sampling technique from that same library. Stratification is used to make sure that both the training and test set contain entries from both classes (Köhl, Magnussen, & Marchetti, 2006).

3.6 Imbalanced Dataset

As discussed in the previous sections, the dataset is fairly imbalanced. Class 1, the stressed class, has the lowest number of entries. There are multiple approaches to solve this inequality. Beyan & Fisher (2015) suggest that there are four approaches: on algorithmic level, data level, cost-sensitive methods and ensembles of classifiers. In this study the chosen approach is on data level. On this level there are mainly two options: undersampling and oversampling. With undersampling a subset of randomly selected entries from the majority class is created, this subset matchs the size of the minority class (Maimon & Rokach, 2005). This produces a more homogeneous dataset. The risk of undersampling, is that it may omit entries that contain useful data, and this can have a negative influence on the performance of the classification models. With oversampling, random entries from the minority class are duplicated and included in the majority class. (Tan, Steinbach, & Kumar, 2014), a downside of this is that the classification model will overfit on the training data. To address this problem, another technique has been developed: SMOTE (Chawla, Bowyer, Hall, & Kegelmeyer, 2002). This is an abbreviation of Synthetic Minority Over-Sampling Technique. As the name suggests, this method produces synthetic entries by interpolating between multiple entries from the minority class. This technique produces samples that are not exact duplicates but approximate combined copies with extra noise added. SMOTE is a powerful method that prevents overfitting, therefor this technique will be applied on the training data in this study.

3.7 Algorithms

Classification models

Wolpert, Macready & others (1997) state in their work, the 'No Free Lunch' (NFL) theorem, that there is no optimal machine learning model for every problem. Every problem is unique and has its own assumptions, and therefor the model that works best is highly dependable on the problem itself. Therefor three different classification algorithms will be used in this study: Random Forest, Support Vector Machines and K-Nearest Neighbours.

Random Forest

The Random Forest algorithm is based on a collection of decision trees. A decision tree (DT) is one of the most basic classification algorithms that is commonly used. A decision tree is a collection of decision nodes that are connected with branches. Starting from the root node, features are tested at and binary split on every decision node and branches derive from these nodes. One branch for each possible outcome, at the end of this branch there is a new decision node unless this is a leaf node. There a various measures to split a node, in this study the Gini Index is used. The Gini Index is formulated as

$$G = \sum_{k=1}^{K} \hat{P}mk\,(1 - \hat{P}\mathrm{mk})$$

$\hat{P}mk$ is the proportion of training observations in the mth region that are from the kth class. The Gini index is also seen as the measure of node 'purity'. When the Gini index is a small number, it implies that the node contains mostly observations from one specific class. (James, Witten, Hastie, & R., 2017)

Random forests are a combination of multiple decision trees. Every single tree in 'a forest' depends on the values of a random vector that is independently sampled. All these trees are sampled with the same distribution. The number of trees used in a random forest has influence on the generalization error, but converges to a limit when the number of trees becomes substantial (Breiman, 2001). The difference between a decision tree and a random forest is that a decision tree is hierarchical and chooses as the first input variable the one that explains the most variance. With a Random Forest the decision trees are fed random input variables, this is a measure to reduce the variance. (James, Witten, Hastie, & R., 2017)

K- Nearest Neighbour

K-Nearest Neighbour (k-NN) is a so-called non-parametric learning algorithm that can be used for regression and classification problems (Altman, 1992). This is an instance-based algorithm which has the training data saved in it. When a new unseen data entry is fed to this classification model, it will match with the k nearest neighbours in the feature space. A majority vote follows by its k neighbours. The entry is then classified as the class which is the most common amid the k neighbours. The value of k, which is a positive integer, has influence on the decision boundary. A low value of k can lead to overfitting, this is because the decision boundary is more complex. A high value of k can lead to underfitting, since the decision

boundary is simpler. (Larose & Larose, 2014) There are several distance measures to measure similarity. In this study the Euclidean distance is used, which is formulated as

$$d_{Euclidean}(\boldsymbol{p}, \boldsymbol{q}) = \sqrt{\sum_{i=1}^{n} (p_i - q_i)^2}$$

Where P = $p_1, p_2, p_3, \dots, p_m$ and q = $q_1, q_2, q_3, \dots, q_m$ represent the m feature values of two observations.

Support Vector Machine

The Support Vector Machine (SVM) is a robust algorithm that can be used for regression and classification. A SVM is trained to identify the maximal margin hyperplane in both non-linear and linearly dividable data (Tan, Steinbach, & Kumar, 2014). This hyperplane divides the different classes and the decision boundary is based on the training data. It is also called a soft margin classifier, which implies that the algorithm allows entries to be on the wrong side of the hyperplane. These wrong entries correspond with the training data that has been misclassified by the SVM. The observations from the training data are used as the support vectors, and help minimizing the classification error. SVM avoids the curse of dimensionality and performs successfully with high dimensional data. (James, Witten, Hastie, & R., 2017)

Pearson Correlation Coefficient

To measure the relationship between features, the Pearson Correlation coefficient(r) will be used. It is a measure to calculate the correlation between features. The value of r can vary between -1, which is a perfect negative relation, and 1, which is a perfect positive relation. Values or r near 0 indicate a very poor relation. The formula is defined as:

$$\boldsymbol{r} = \frac{\sum_{i=1}^{n}(\boldsymbol{X_i} - \bar{\boldsymbol{X}})(\boldsymbol{Y_i} - \bar{\boldsymbol{Y}})}{\sqrt{\sum_{i=1}^{n}(\boldsymbol{X_i} - \bar{\boldsymbol{X}})^2}\sqrt{\sum_{i=1}^{n}(\boldsymbol{Y_i} - \bar{\boldsymbol{Y}})^2}}$$

Where $\bar{\boldsymbol{X}}$ and $\bar{\boldsymbol{Y}}$ are the sample means for the two features. In earlier research (Evans, 1996) there has been developed a guideline for the strength of correlations. These are: a. very weak: (r=0.00-0.019), b. weak: (r=0.20-0.39), c. moderate: (r=0.40-0.59), d. strong: (r=0.60-0.79) and e. very strong: (r=0.80-1.0). These guidelines will be used in the study.

3.8 Evaluation method

Accuracy

A much used evaluation metric for classification is accuracy. Accuracy is the proportion of correctly classified true predictions among the total number of predictions.

$$Accuracy = \frac{True\ Positives + True\ Negatives}{True\ Negatives + True\ negatives + False\ Positives + False\ Negatives}$$

The definitions used in the above formula are defined below at the confusion matrix section at table 3.8.1 the bottom of the page.

Balanced Accuracy

There is a risk with using only accuracy as an evaluation metric, this risk is the so-called accuracy paradox. When data is skewed, a classification algorithm can achieve a high accuracy but still have a poor performance. Due to the imbalance of the dataset also the balanced accuracy metrics is included in this study. This metric is defined as:

$$Balanced\ Accuracy = \frac{True\ Positive\ Rate + True\ Negative\ Rate}{2}$$

Where the true positive rate is: $TPR = \frac{True\ Positives}{True\ Positives + False\ Negatives}$

The true negative rate is: $TNR = \frac{True\ Negatives}{True\ Negatives + False\ Positives}$

Confusion Matrix

A clear practice to visualise the numbers used to calculate the accuracy and balanced accuracy is the so-called error/ confusion matrix. This matrix gives a clear overview of how the algorithms have classified the data.

Table 3.8.1 – Confusion Matrix

	Class 0 prediction	Class 1 prediction
Class 0 Actual	True Negatives	False Positives
Class 1 Actual	False Negatives	True Positives

3.8 Parameter Selection and Tuning

The three different classification algorithms have several parameters which can be tuned to improve the performance. In this study the focus will lie on tuning the parameters that can be found in table 3.8.1. Every algorithm has different parameters that can be tuned. With the use of GridSearchCV, which is part of the sklearn library, the optimal parameters can be found. The function tunes the parameters of the algorithms systematically with pre-set values. The performance metrics that it seeks to optimise is accuracy. The GridSearchCV function will split the training data with the use of cross validation into five different partitions. These partitions are then alternately used as training and validation set. Having five partitions means that the GridSearchCV function will perform five different tests, where every partition will function one time as a validation set. Cross Validation prevents against over-fitting and selection bias (Cawley & & Talbot, 2010). The GridSearchCV functions tunes and tests the parameters for every partition and selects the optimal ones.

Table 3.8.1 Model parameters

Classification model	Parameters
Random Forest Classifier (RF)	Depth Number of trees Minimum samples split Minimum samples leaf
Support Vector Machine Classifier (SVM)	C: regularization parameter. Penalty error term. $\in$: epsilon Kernel: Radial Basis function (RBF)
K-Nearest Neighbours Classifier (kNN)	K : number of neighbors

The random forest has several parameters, one is depth, which is the maximum depth that a tree is allowed to have. Setting a maximum prevents the tree from overfitting on the training data (Shu, 2020). A larger number of trees improved the performance, but there is a threshold on which the performance gain is not significant anymore. From this moment an extra tree will only cost extra computational power (Oshiro, Perez, & Baranauskas, 2012). Minimum samples split and minimium samples leaf are parameters that prevent against overfitting (Brefeld, 2019). The support vector machine has parameter C, which is a regularization parameter. This parameters limits the influence that the support vectors have on the algorithm (Witten, Frank, Hall, & Pal, 2016). The kernel in this study is set on RBF since it is proven to be the most effective kernel for splitting non linearly separable data (Howlett & Jain, 2001). The epsilon is

the boundary around the classification function, all the erroneous data points that fall within this boundary are ignored, this prevents against overfitting (Hamel, 2011). The value k from the k-NN algorithm is already discussed in section. It is the amount of neighbours that the algorithms should take in consideration to predict the class.

3.9 Software

This study used Python (version 3.6.9) to build models and perform analyses. Preprocessing is done with the use of Jupyter Notebooks that are hosted by the Google Colaboratory servers. Within Python, a variety of packages has been used, these are outlined below. The 'numpy' and 'pandas' packages are used for data preprocessing. With the use of 'sklearn' package the various classification models are tuned and applied. For the data visualizations the 'Matplotlib' and 'Seaborn' packages have been employed.

Package name	Version	Source
Numpy	1.17.4	(Oliphant, 2006)
Pandas	0.25.3	(McKinney, Wes, & others, 2010)
Scikit-learn/ sklearn	0.21.3	(Pedregosa, et al., 2011)
Matplotlib	3.1.2	(Hunter, 2007)
Seaborn	0.9.0	(Waskom, et al., 2018)

3.10 Setup

There are different datasets chosen for the classification. These datasets are Random1, Random2, Random3, Top30 and TopAll, these sets will be referred to as the five different models. These five different models are used to test whether a generic classification model outperforms models that are user-specific. Respondents random 1, 2 and 3 are randomly chosen respondents from the top30 dataset. These respondent datasets all have at least 80 different data entries. The top30 is the dataset with the entries from the thirty respondents that have returned a completed survey the most often. Top30 is a dataset with 2.447 data entries. TopAll is the complete dataset with entries from all the respondents (n=64) in the sample dataset. Topall consists of 3.802 data entries.

3.10.1 Baseline

The baselines for the different models are based on the size of the majority class of these models. The classifiers will be evaluated with the accuracy and the balanced accuracy metrics. The accuracy baseline will be set to the proportional size of the majority class of the different models. All the different baselines for every model can be found in table 3.10.1. Since the dataset was split with the stratified sampling technique, both the training and test set are given the same baseline. Class 0 is for all the models the majority class

Table 3.10.1 Majority Baseline Accuracy

Respondents	Train	Test
Random1	81.7%	81.7%
Random2	63.8%	63.8%
Random3	76.3%	76.3%
Top30	69.6%	69.6%
All Respondents	71.8%	71.8%

3.10.2 Setup research question 1

Research question 1 is stated as follows: is there a distinction between the factors that influence the perceived stress levels for the user-specific models and the generic models? To test the influence of these features on the stress levels, this research has made use of the Pearson correlation coefficient. With the use of the programming language Python all the correlation coefficients of all the extracted features, which are described in section, are displayed in the correlation matrices.

3.10.2 Setup research question 2

Research question 1 is stated as follows: what model predicts best the perceived stress levels of smartphone users? Three different classification algorithms have been trained to predict perceived stress from the respondents. The six stress levels have been transformed into two classes, not stressed and stressed. With the use of the Random Forest, k-Nearest Neighbours and Support Vector Machine classifier algorithms the different models are tested. The data was split in 80% training data and 20% test data. The parameters of the various algorithms have been tuned with the use of GridSearchCV, which cross validated the training into five folds. The parameters were tested on the test dataset, and these performance scores are measured in accuracy and balanced accuracy. The scores are compared to the baseline score, which is related to the size of the majority class.

4. Results

This section will provide the results from the various analyses. Section 4.1 will give the results from the exploratory analysis and in section 4.2 the results from the classification tasks will be provided.

4.1 Exploratory analysis

The correlations between the features for respondent random1 are shown in Table 4.1.1. This respondent has not installed or used any finance or online shopping applications during the test period. Therefore the fina_time, shop_time, fina_count and shop_count features are omitted As can be seen from the table has the target feature the strongest correlation the features ente_time, max_len_sess and avg_time_sess, respectively, with moderate correlations of 0.4188, 0.5201 and 0.4916. A noteworthy addition, both the avg_time_sess and total_time_phone features are very strongly related with each other and other features. The features and correlations of respondent random1 will be further discussed in section 5.

Table 4.1.1 - Random1, Correlation matrix 10 most influential features

	soci_time	util_time	prod_count	total_time_phone	sess_count	max_len_sess	avg_time_sess	if_weekend	yhour	target
ente_time	-0.1418	-0.0431	-0.0793	0.6058	-0.1830	0.7973	0.8289	0.2179	-0.1549	-0.4188
soci_time		0.0351	0.1622	0.3654	0.3616	0.1906	0.0958	-0.0341	-0.0582	-0.1472
util_time			0.0846	0.2825	-0.0269	0.2334	0.2641	0.1497	-0.0590	-0.2230
prod_count				-0.0039	0.4255	-0.1454	-0.1334	-0.1956	0.1459	0.1777
total_time_phone					0.0702	0.8817	0.7809	0.2595	-0.0338	-0.3606
sess_count						-0.2031	-0.2922	-0.1296	0.3763	0.2638
max_len_sess							0.8986	0.2882	-0.1930	-0.5201
avg_time_sess								0.2748	-0.2376	-0.4916
if_weekend									-0.0008	-0.2199
xhour										0.1988

Table 4.1.2 shows the ten strongest correlations between the different features for respondent random2. The sess_count, xhour and proc_count are the features that have the strongest correlation with the target feature. The correlation coefficient for these features are 0.3037, 0.2420 and 0.1923 respectively, which are weak relations. Proc_count is one of the features that is stronger correlated with the target variable, but is also higly correlated with soci_count and sess_count. Also, fina_time is very strongly correlated with fina_time. A very strong relation is also seen between soci_count and sess_count (r = 0.822). This will be further discussed in section 5.

Table 4.1.2 Random2 - 10 most influential features

	fina_time	life_time	fina_count	shop_count	proc_count	soci_count	sess_count	xhour	yhour	target
ente_time	-0.0617	-0.0558	-0.0852	-0.0412	-0.0120	-0.0754	-0.1808	-0.0556	-0.0845	-0.1223
fina_time		-0.0157	0.8218	0.1512	0.0369	0.1328	0.2278	0.0790	0.0007	0.1348
life_time			-0.0003	0.0074	0.0253	-0.0910	-0.0656	-0.1113	-0.0584	0.1638
fina_count				0.1437	0.1380	0.1533	0.2812	0.1312	0.0415	0.1799
shop_count					0.2261	0.2362	-0.0043	-0.0678	-0.0299	-0.1130
proc_count						0.6993	0.4751	0.2440	0.2053	0.1923
soci_count							0.7595	0.2239	0.1601	0.1177
sess_count								0.3821	0.2569	0.3037
xhour									0.8631	0.2420
yhour										0.1726

The ten features for respondent random3 that are most correlated with the target feature are displayed in table 4.1.3. The features that are most correlated with the target feature are noti_count, util_count, life_time and ente_time, with a correlation of respectively -0.1414, -0.1339, -0.1276 and -0.1238. There are also other features that are strongly related, these will be discussed in section 5.

Table 4.1.3. Random3 - 10 most influential features

	game_time	life_time	proc_time	life_count	prod_count	util_count	noti_count	min_len_sess	if_weekend	target
ente_time	-0.0738	-0.0305	0.3659	-0.0144	0.2872	0.2196	0.3042	0.2848	-0.0277	-0.1238
game_time		0.0431	0.2641	0.0704	0.1298	0.0920	0.0415	-0.0781	-0.1188	-0.1048
life_time			0.0405	0.8917	-0.0633	0.2917	0.2813	0.1227	0.1070	-0.1276
proc_time				0.0651	0.5727	0.5074	0.2268	0.0030	-0.1685	-0.0900
life_count					-0.0574	0.3037	0.2496	-0.0178	0.1737	-0.1060
prod_count						0.6022	0.1415	0.0957	-0.0968	-0.0973
util_count							0.3036	0.0082	-0.1760	-0.1339
noti_count								0.0257	-0.0623	-0.1414
min_len_sess									0.0871	-0.0967
if_weekend										-0.1090

The correlation for the features of the top30 respondents are displayed in table. There are no strong relations between the features and the target. The feature with the strongest relation with the target feature is sess_count with a correlation of 0.0842, which is very weak. Furthermore the total_time_phone and uni_app_count are strongly related with a portion of the other features.

Table 4..1.4 Top30 - 10 most influential features

	soci_ time	util_ time	game_ count	proc_ count	soci_ count	total_ time_ phone	sess_ count	min_ len_ sess	uni_ app_ count	target
game_time	0.0283	0.0009	0.7569	-0.0315	0.0242	0.1898	0.0706	-0.0218	0.1478	0.0517
soci_time		0.0668	0.0509	0.2306	0.6032	0.7317	0.1509	0.2515	0.3800	0.0399
util_time			-0.0088	0.2281	0.0206	0.3594	0.0609	0.0272	0.2229	-0.0433
game_count				-0.0217	0.0718	0.1581	0.1340	-0.0214	0.1585	0.0376
proc_count					0.3549	0.3247	0.2457	-0.0099	0.5390	0.0675
soci_count						0.4516	0.5921	0.0654	0.5211	0.0643
total_time_phone							0.1437	0.3220	0.5138	0.0423
sess_count								-0.1718	0.5023	0.0842
min_len_sess									0.0096	-0.0531
uni_app_count										0.1015

Uni_app_count, sess_count and soci_count are the features that have strongest correlation with the target feature. These relations are all very weak. The uni_app_counts shows a moderate relation with proc_count and soci_count, which is not strange since these applications are among the most used features. Furthermore the soci_time is strongly related with the total_time_phone, this is not remarkeable as it is the most used application category.

Table 4.1.5 Topall - 10 most influential features

	soci_ time	life_ count	proc_ count	soci_ count	total_ time_ phone	sess_ count	min_ len_ sess	uni_ app_ count	xhour	target
life_time	0.0038	0.5048	0.0967	0.0312	0.2245	0.0940	0.0280	0.1698	0.0298	0.0590
soci_time		0.0268	0.2612	0.6069	0.7003	0.2243	0.2064	0.3914	0.0504	0.0419
life_count			0.1151	0.1084	0.1499	0.3338	0.0061	0.2707	0.0553	0.0440
proc_count				0.4700	0.3255	0.2938	-0.0001	0.4531	0.0310	0.0494
soci_count					0.4399	0.6600	0.0307	0.5442	0.1047	0.0725
total_time_phone						0.1876	0.2889	0.5215	0.0507	0.0412
sess_count							-0.1672	0.5574	0.1357	0.0849
min_len_sess								-0.0040	-0.0432	-0.0442
uni_app_count									0.0528	0.0994
xhour										0.0351

All the different models have their own specific features that have the most influence on the target feature. Sess_count belongs to the strongest related features for respondent random2 and this feature is also one of the stronger related features for the top30 and topAll models. This will be discussed in section 5.

4.1 Classification models

In this section the results of the second part of the research will be presented. Tests have been run to answer research question two: What model predicts best the perceived stress levels of smartphone users?

4.2.1 Random Forest

The random forest is one of the three classification algorithms that has been used in this study. The parameters of the random forest have been tuned with the use of the GridSearchCV instance. The parameters haven been tested with the values that are mentioned in section 3.8. The optimal parameters are different for each model and are displayed in table.

Table 4.2.1 – Optimal Parameters Random Forest

	N_ estimators	Max_ depth	Min_ samples_split	Min_ samples_leaf
Random1	15	25	4	1
Random2	20	50	5	3
Random3	8	15	4	2
Top30	30	10	2	1
TopAll	30	10	2	1

Four out of the five models achieved the highest accuracy with the Random Forest algorithm. Model Random1 has the best performance on the test set with the random forest with an accuracy of 94.1%, the baseline for this model was 81.7%. Model 1 scored high above the baseline. Also for model Random2 the random forest achieved the highest accuracy of 63.2% against a baseline of 63.8%. So it scored just below the baseline. The Top30 model scored an accuracy of 68.6% against a baseline of 69.6%, this model scored below the baseline. This is also the case for the TopAll model which achieved an accuracy of 67.1% against the baseline of 71.8%. Only Random1 model achieved an accuracy that was above the baseline. The table 4.2.2 shows the predictions of the random forest algorithm.

Table 4.2.2 - Confusion Matrix Random Forest

	Random1		Random2		Random3		Top30		Topall	
Actual/	Class	Class	Class	Class	Class	Class	Class	Class	Class	Class
Predicted	0	1	0	1	0	1	0	1	0	1
Class 0	2	1	9	3	8	4	283	58	450	96
Class 1	0	14	4	3	3	1	96	53	154	61

K-Nearest Neighbors (k-NN)

The second classification algorithm that was applied in this study, was the k-Nearest Neighbors. The data was normalized before analysis, since the k-NN algorithm requires normalized data. Since the K-NN algorithm is non-parametric, it is not difficult to find the optimal value of k. The different models are plotted and the optimal values of k can be noted from these plots. These optimal parameters can be found in table 4.2.3

Table 4.2.3 Optimal Parameter k-NN	
	k
Random1	15
Random2	6
Random3	2
Top30	2
TopAll	2

For model random3 the k-NN algorithm scored the highest performance. This model had an accuracy of 75% which is just below the baseline of 76.3%. Model random1 scored 76.5% accuracy against the baseline of 81.7%. Model random2 scored only 52.5% accuracy against a baseline of 63.8%. Models Top30 and TopAll scored an accuracy of 60.6% and 62.6% against a baseline of 69.6% and 71.8% respectively. Table 4.2.4 shows the predictions of the k-Nearest Neighbours algorithm.

Table 4.2.4 k-NN

	Random1		Random2		Random3		Top30		Topall	
Actual/	Class	Class	Class	Class	Class	Class	Class	Class	Class	Class
Predicted	0	1	0	1	0	1	0	1	0	1
Class 0	2	1	7	5	11	1	260	81	415	131
Class 1	3	11	4	3	3	1	112	37	157	58

Table 4.2.5 Optimal Parameters SVM		
	C	Epsilon
Random1	100	0.1
Random2	1	1
Random3	0.1	0.1
Top30	1	1
TopAll	1	0.1

Support Vector Machine Classifier

The last algorithm is the svm classifier. The parameters of this model are tuned with the use of the GridSearchCV. The optimal parameters can be found in table 4.2.5. None of the model achieved a nice accuracy score with this algorithm. Table 4.2.6 provides the predictions of the SVM algorithm. The accuracy scores for the SVM algorithm on the train and test sets can be found in table 4.2.8 on the following page.

Table 4.2.6 SVM Confusion Matrix

	Random1		Random2		Random3		Top30		Topall	
Actual/	Class	Class	Class	Class	Class	Class	Class	Class	Class	Class
Predicted	0	1	0	1	0	1	0	1	0	1
Class 0	2	1	6	6	5	7	203	138	316	230
Class 1	1	13	3	4	2	2	67	82	91	124

(Balanced) Accuracy Test

Since the data was imbalanced the accuracy score was not very useful, the score did not give a representation of the performance. Therefor the balanced accuracy is used as an extra metric. As can be seen from table 4.2.7 show the balanced accuracy scores deteriorated scores. Only model1 achieves an balanced accuracy that is above the baseline on all of the algorithms. The rest of the models only achieves an accuracy below the baseline.

Table 4.2.7 – Balanced Accuracy Test Sets				
	RandomForest	SVM	kNN	Baseline
Random1	96.7%	79.8%	65.8%	81.7%
Random2	59.6%	53.3%	50.6%	63.8%
Random3	46.4%	46.8%	64.3%	76.3%
Top30	61.2%	56.2%	50.6%	69.6%
TopAll	56.7%	56.3%	51.6%	71.8%

Table 4.2.8 Accuracy scores all models

		Random Forest		Support Vector Machine		k-Nearest Neighbours	
Dataset	Baseline	Train	Test	Train	Test	Train	Test
Random1	81.7%	100%	94.1%	98.1%	88.2%	80.2%	76.5%
Random2	63.8%	92.0%	63.2%	88.5%	52.6%	72.9%	52.6%
Random3	76.3%	96.9%	56.3%	67.3%	43.75%	98.0%	75.0%
Top30	69.6%	93.4%	68.6%	63.7%	58.2%	96.3%	60.6%
TopAll	71.8%	88.1%	67.1%	59.6%	57.8%	96.7%	62.2%

5. Discussion, Limitations & Conclusion

In this last section the discussion, limitations and conclusion can be found. Section 5.1 provides the discussion of the results. Section 5.2 discusses the limitations of the study, and Section 5.3 is the conclusion.

5.1 Discussion

The goal of this study was to analyse whether perceived stress levels could be predicted by phone application usage. This goal was split into two research questions, one that focused on the correlation of the features of the different models. The other part focussed on which model the best predictor was of the perceived stress levels of smartphone users.

The first part of the study focussed on the different features for all the different models. The aim was to explore for each model, if the stress level was strongly correlated with other features. Models random1, random2, and random3 all showed features that are moderately or strongly related with the stress features. These were smaller samples, which is also of influence but the different model also showed a lot of dissimilarities. For the three random models the most influential features were not the same. None of these three models had one of the others most influential features in their top three most influential features. This suggest that for every model other features are important. This is in line with results of another study (Ferdous, Osmani, & Mayora, 2015) (Bauer & Lukowicz, 2012) (Muaremi, Arnrich, & Tröster, 2013) that suggests that user-specific phone usage data is a better predictor for stress than generic data.

The second part of the study focussed on the predicting the stress with different algorithms. Only the model random1 achieved a balanced accuracy above the baseline on all the algorithms. The rest of the models scored below the baseline score. This suggests that the amount of data that is collected in the two hours before a survey might be too little. The performance of model Random1 on all the algorithms scored above the baseline. Model random1 seemed to be a lucky shot and is the exception on the rest. On average the algorithms scored badly. This might have to do with the minimum amount of data that is collected within those two hours upon a survey.

5.2 Limitations and future research

Two hour time frame

The decision to focus only one the two hours upon a survey might have been a little optimistic. The amount of data that is collected in this time frame is too minimal, as a result the observations have a high similarity. An idea for future research is the working with the moving time window technique. Which can be used with and without overlap.

Wasteful approach

The received survey dataset contained 16.016 entries, eventually 3.802 surveys are used for analysis. A large part of the dataset did contain empty surveys but this were only 5.372 surveys. Future researchers could make use of data imputation techniques to replace the NaNs.

The self-reporting bias

People that self-report their mood might be biased. These biases are categorised social desirability and recall bias. One might fill in on a survey that he thinks that he is stressed, but he only gives this stress level because he just almost had a car accident. With a survey this cannot be checked. For future research the application usage might be linked with a heartbeat sensor or other wearables that have physiological sensors.

5.3 Conclusion

The goal of this research was to analyse whether perceived stress levels could be predicted by phone application usage. To address this task there were two research questions developed:
Q1. Is there a distinction between the factors that influence the perceived stress levels for the user-specific models and the generic models?

Q2. What model predicts best the perceived stress levels of smartphone users?

The first question was answered through the use of Pearson correlaton matrices. Five different models were developed, 3 with user-specific data and two generic models of different amounts of data. All the user-specific models showed features that were moderately and strongly related with the stress level feature. But non of these features matched with the other models. This suggests that the impact that a phone has on the stress levels is different for each respondent. Which is in line with other studies. (Ferdous, Osmani, & Mayora, 2015) (Bauer & Lukowicz, 2012) (Muaremi, Arnrich, & Tröster, 2013)

Three different classification algorithms have been developed for the second research question: Random Forest, k-Nearest Neighbours and Support Vector Machine. The results showed that only one user-specific model scored above the baseline. The other models all scored below. The overall performance for the algorithms was poor.

The formulated problem statement for this research was:

To what extent can perceived stress levels be predicted by phone application usage?

The results showed that prediction of stress on basis of phone usage is not possible in the way that is has been addressed in this study. The models all scored below the baseline accuracy. This research has shown that the amount of data that is collected in the time frames before the survey is very little. In future research wider timeframes are suggested, that might even have overlap with other timeframes.

Bibliography

Ahn, H., Wijaya, M. E., & & Esmero, B. C. (2014). A systemic smartphone usage pattern analysis: focusing on smartphone addiction issue. *Int J Multimedia Ubiquitous Engineering*, 9-14.

Altman, N. (1992). An introduction to kernel and nearest-neighbor nonparametric regression. *The American Statistician, 46(3)*, 175-185.

American Psychological Association. (2015). *Stress in America: Paying With Our Health.* Retrieved from APA: https://www.apa.org/news/press/releases/stress/2014/stress-report.pdf

Andrews, S., Ellis, D., Shaw, H., & Piwek, L. (2015). Beyond Self-Report: Tools to Compare Estimated and Real-World Smartphone Use. *Plos One*.

Annie, A. (2017). *Retrospective: A monumental year for the app economy.* App Annie.

Bauer, G., & Lukowicz, P. (2012). Can Smartphone Detect Stress-Related Changes in Behaviour of Individuals. *IEEE International Conference on Pervasive Computing and Communications* (pp. 423 - 426). Lugano: IEEE.

Beyan, C., & Fisher, R. (2015). Classifying imbalanced data s ets using similarity based. *Pattern Recognition, 48*, 1653-1672.

Blix, E., Perski, A., Berglund, H., & Savic, I. (2006). Long-term occupational stress is associated with regional reductions in brain tissue volumes. *PLoS One, 8*.

Bogomolov, A., Lepri, B., Ferron, M., Pianesi, F., & Pentland, A. S. (2014). Pervasive stress recognition for sustainable living. *2014 IEEE International Conference on Pervasive Computing and Communication Workshops (PERCOM WORKSHOPS)* (pp. 345-350). IEEE.

Bomhold, C. R. (2013, 3 29). Educational use of smart phone. *Program: electronic library and information systems*, pp. 424-436.

Brefeld, U. (2019). *Machine Learning and Data Mining for Sports Analytics.* Springer.

Breiman, L. (2001). *Random Forests.* Berkeley: University of California.

Cawley, G. C., & & Talbot, N. L. (2010). On over-fitting in model selection and subsequent selection bias in performance evaluation. *Journal of Machine Learning Research, 11(Jul)*, 2079-2107.

Chawla, N. V., Bowyer, K. W., Hall, L. O., & Kegelmeyer, W. P. (2002). SMOTE: synthetic minority. *Journal of artificial intelligence research*, 321-357.

Clark, A. (2018, 3 26). The Mind-Expanding Ideas of Andy Clark - The tools we use to help us think—from language to smartphones—may be part of thought itself. (L. MacFarquhar, Interviewer)

Clarke, A., & Chalmers, D. (1998). The Extended Mind. *Analysis*, 7 - 19.

Deloitte. (2019). *Global Mobile Consumer Survey 2019.* Deloitte.

Epel, E. S., Crosswell, A. D., Mayer, S. E., Prather, A. A., Slavich, G. M., Puterman, E., & & Mendes, W. B. (2018). More than a feeling: A unified view of stress measurement for population science. *Frontiers in neuroendocrinology, 49*, 146-169.

Evans, J. D. (1996). *Straightforward statistics for the behavioral sciences. .* Pacific Grove: Brooks/Cole Publishing.

Ferdous, R., Osmani, V., & Mayora, O. (2015). Smartphone app usage a a predictor of perceived stress levels at workplace. *9th International Conference on Pervasive Computing Technologies for Healthcare (PervasiveHealth).* Trento: CREATE-NET.

Franke, S. (2003). Studying and working: The busy lives of students with paid employment. *Statistics Canada*.

Fukazawa, Y., Ito, T., Okimura, T., Yamashita, Y., Maeda, T., & Ota, J. (2019). Predicting anxiety state using smartphone-based passive sensing. *Journal of biomedical informatics, 93*.

Fukunaga, K. (2013). *Introduction to statistical pattern recognition.* Elsevier.

Godbout, J. P., & Glaser, R. (2006). Stress-induced immune dysregulation: implications for wound healing, infectious disease and cancer. . *Journal of Neuroimmune Pharmacology*, 421-427.

Hamel, L. H. (2011). *Knowledge discovery with support vector machines (Vol. 3).* John Wiley & Sons.

Harri, A., & Brorsen, B. (2009). The Overlapping Data Problem. *Quantitative and Qualitative Analysis in Social Sciences, 3*, pp. 78-115.

Hendrickson, A., Abeele, M. V., & Aalbers, G. ((under review)). *Phone use data.*

Hooftman, W., Mars, G., Janssen, B., de Vroome, E., Janssen, B., Pleijers, A., . . . van den Bossche, S. (2019). *Nationale Enquête Arbeidsomstandigheden 2018.* Leiden: TNO | CBS.

Howlett, R. J., & Jain, L. C. (2001). Radial basis function networks 1: recent developments in theory and applications (Vol. 66). *Springer Science & Business Media.*

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

James, G., Witten, D., Hastie, T., & R., T. (2017). *An Introduction to Statistical Learning.* New York: Springer.

Jones, E., Oliphant, T., Peterson, P., & others, &. (2001). *SciPy: Open source scientific tools for Python*. Retrieved from http://www.scipy.org/

Kang, J. M., Seo, S. S., & & Hong, J. W. (2011). Usage pattern analysis of smartphones . *13th Asia-Pacific Network Operations and Management Symposium* (pp. 1-8). IEEE.

Kirschbaum, C. P. (1993). The 'Trier Social Stress Test'–a tool for investigating psychobiological stress responses in a laboratory setting. *Neuropsychobiology*, 76-81.

Köhl, M., Magnussen, S. S., & Marchetti, M. (2006). *Sampling Methods, Remote Sensing and GIS Multiresource Forest Inventory.* Springer Science & Business Media.

Kompier, M., & Cooper, C. (2003). *Preventing Stress, Improving Productivity - European case studies in the workplace.* London: Routledge.

Kushlev, K., Proulx, J., & Dunn, E. (2016). "Silence Your Phones": Smartphone Notifications Increase. *CHI'16 - 2016 CHI Conference on Human Factors in Computing Systems* (pp. 1011 - 1020). San Jose: ACM.

Larose, D. T., & Larose, C. D. (2014). *Discovering Knowledge in Data.* New Jersey: Wiley.

Lebepe, F., Niezen, G., Hancke, G. P., & Ramotsoela, T. D. (2016). Wearable stress monitoring system using multiple sensors. *IEEE 14th International Conference on Industrial Informatics (INDIN)* (pp. 895-898). IEEE.

Maimon, O., & Rokach, L. (2005). *Data mining and knowledge discovery handbook.* Springer.

Matthews, K. A., Woodall, K. L., & & Allen, M. T. (1993). Cardiovascular reactivity to stress predicts future blood pressure status. *Hypertension, 22(4)*, 479-485.

McKinney, W., Wes, & others, &. (2010). Data structures for statistical computing in python. *9th Python in Science Conference*, (pp. 51–56).

Muaremi, A., Arnrich, B., & Tröster, G. (2013). Towards measuring stress with smartphones and wearable devices during workday and sleep. . *BioNanoScience, 3*, p. 172173.

Oliphant, T. E. (2006). *A guide to NumPy (Vol.1).* Trelgol Publishing USA.

Oshiro, T. M., Perez, P. S., & Baranauskas, J. A. (2012). How many trees in a random forest? *International workshop on machine learning and data mining in pattern recognition* (pp. 154-168). Berlin: Springer.

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

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., & ...others. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 2825 –2830.

Pielot, M., & Rello, L. (2017). Productive, Anxious, Lonely - 24 Hours Without Push Notifications. *19th International Conference on Human-Computer Interaction with Mobile Devices and Services* (p. MobileHCI '17). Austria, Vienna: ACM.

Przybylski, A. K., & Weinstein, N. (2017). A large-scale test of the Goldilocks Hypothesis: Quantifying the relations between digital-screen use and the mental well-being of adolescents. *Psychological Science*, 204-215.

Rijksoverheid. (2019, 7). *Mag ik appen, bellen en naar muziek luisteren op de fiets?* Retrieved from Rijksoverheid.nl: https://www.rijksoverheid.nl/onderwerpen/fiets/vraag-en-antwoord/mag-ik-bellen-en-naar-muziek-luisteren-op-de-fiets

Rosengren, A., Hawken, S., Ôunpuu, S., Sliwa, K., Zubaid, M., Almahmeed, W. A., & investigators., .. &. (2004). Association of psychosocial risk factors with risk of acute myocardial infarction in 11 119 cases and 13 648 controls from 52 countries (the INTERHEART study): case-control study. *The Lancet, 364*, 953-962.

Sarker, I. H., Kayes, A. S., & Watters, P. (2019). Efectiveness analysis of machine learning classifcation models for predicting personalized context-aware smartphone usage. *Journal of Big Data 6(1).*

Shu, X. (2020). *Knowledge Discovery in the Social Sciences: A Data Mining Approach.* University of California Press.

Simyo, DirectResearch. (2019, 7). *Word jij slimer of juist socialer van de smartphone?* Retrieved from Simyo: https://www.simyo.nl/onderzoek/smartphone-gebruik/

Tan, P., Steinbach, M., & Kumar, V. (2014). *Introduction to Data Mining.* Essex: Pearson.

Thomeé, S. (2018). Mobile Phone Use and Mental Health. A Review of the Research That Take a Psychological Perspective on Exposure. *International Journal of Environmental Research and Public Health, 15.*

Waskom, M., Botvinnik, O., O'Kane, D., Hobson, P., Ostblom, J., & Lukauskas, S. (2018, 07). *mwaskom/seaborn: v0.9.0.* Retrieved from https://doi.org/10.5281/zenodo.1313201

Witten, I. H., Frank, E., Hall, M. A., & Pal, C. J. (2016). *Data Mining: Practical machine learning tools and techniques.* Morgan Kaufmann.

Zenonos, A., Khan, A., Kalogridis, G., Vatsikas, S., Lewis, T., & Sooriyabandara, M. (2016). HealthyOffice: Mood recognition at work using smartphones and wearable sensors. *2016 IEEE International Conference on Pervasive Computing and Communication Workshops (PerCom Workshops)* (pp. 1-6). IEEE.

Appendix A

Correlation Matrix Random1	game_time	life_time	proc_time	prod_time	soci_time	util_time	ente_count	game_count	life_count	proc_count	prod_count	soci_count	util_count	noti_count	total_time_phone	sess_count	min_len_sess	max_len_sess	avg_time_sess	uni_app_count	if_weekend	xhour	yhour	target
ente_time	0.021418	-0.046056	-0.032529	-0.056467	-0.141830	-0.043065	0.382926	0.002012	-0.075547	-0.010710	-0.079270	-0.150215	0.006024	-0.079614	0.605748	-0.182977	-0.044460	0.797288	0.828906	-0.004162	0.217943	-0.154937	-0.082617	-0.418749
game_time		-0.090314	0.154076	-0.096295	-0.019092	-0.027160	0.024859	0.944229	-0.133936	0.099963	-0.133072	-0.156827	0.034517	-0.171748	0.562958	0.065843	0.013316	0.306545	0.147205	0.210789	0.160724	0.146311	0.115823	0.083494
life_time			-0.012409	-0.045992	0.148693	-0.027543	-0.112111	-0.099855	0.555101	-0.006725	-0.062581	0.003515	0.006977	0.098395	0.102032	-0.022528	-0.063734	0.083009	0.089831	0.003858	-0.041538	0.204242	0.237603	0.017467
proc_time				0.332707	0.020511	0.032685	-0.025548	0.108421	0.034373	0.381233	0.269210	0.041623	0.181475	0.071164	0.120659	0.029454	-0.033776	0.048761	0.054992	0.361114	-0.108767	-0.083549	0.016285	0.087487
prod_time					0.125244	0.098812	0.136643	-0.078704	-0.033317	0.567787	0.936205	0.248697	0.261236	0.410719	0.026845	0.264304	-0.086488	-0.102664	-0.088132	0.420955	-0.130620	0.190418	0.208425	0.134384
soci_time						0.035116	0.138280	-0.003419	0.206638	0.146153	0.162229	0.519261	0.192274	0.242774	0.365396	0.361600	-0.176108	0.190604	0.095782	0.323356	-0.034129	-0.058178	0.026328	-0.147209
util_time							-0.072611	-0.021618	-0.014584	0.144856	0.084553	0.015499	0.220746	-0.058119	0.282505	-0.026924	0.003078	0.233447	0.264125	0.215074	0.149677	-0.059016	-0.054290	-0.223015
ente_count								0.055264	-0.183901	0.129033	0.097963	0.058640	0.042506	-0.104744	0.300811	0.146285	-0.189764	0.303426	0.265418	0.321454	0.245897	-0.047998	0.004798	0.026769
game_count									-0.149888	0.085403	-0.106795	-0.118443	-0.010151	-0.138575	0.524414	0.110186	0.012920	0.267760	0.115505	0.207345	0.214591	0.150542	0.138071	0.069460
life_count										0.083990	-0.057778	0.164413	0.150792	0.101940	0.029800	0.117149	-0.104871	-0.004898	0.006000	0.209890	0.003741	0.122372	0.099420	-0.035061
proc_count											0.487463	0.232066	0.535837	0.206801	0.186748	0.123013	-0.018994	0.085741	0.057935	0.633893	-0.054596	0.125826	0.122332	0.116461
prod_count												0.367379	0.322001	0.398966	-0.003923	0.425524	-0.120830	-0.145379	-0.133433	0.427945	-0.195640	0.145930	0.121572	0.177671
soci_count													0.322966	0.520927	0.049483	0.827368	-0.324979	-0.150870	-0.215577	0.453013	-0.234987	0.221572	0.121961	0.129396
util_count														0.197374	0.190910	0.424995	-0.140304	0.084832	0.046109	0.672454	-0.036718	0.160820	0.173881	0.146544
noti_count															-0.036769	0.397210	-0.280701	-0.162623	-0.179849	0.218013	-0.225769	0.378981	0.311298	0.011262
total_time_phone																0.070195	-0.110546	0.881727	0.780922	0.350549	0.259528	-0.033774	0.043600	-0.360610
sess_count																	-0.411921	-0.203145	-0.292214	0.493387	-0.129560	0.376308	0.235516	0.263774
min_len_sess																		-0.058400	0.031469	-0.222681	0.156559	-0.265833	-0.134601	-0.114785
max_len_sess																			0.898569	0.155337	0.288186	-0.193041	-0.079442	-0.520098
avg_time_sess																				0.066854	0.274794	-0.237634	-0.097271	-0.491558
uni_app_count																					0.001089	0.144743	0.184794	0.119527
if_weekend																						-0.000773	-0.013034	-0.219896
xhour																							0.850833	0.198821
yhour																								0.130990

Correlation Matrix Random 2	fina_time	game_time	life_time	shop_time	proc_time	prod_time	soci_time	util_time	ente_count	fina_count	game_count	life_count	shop_count	proc_count	prod_count	soci_count	util_count	noti_count	total_time_phone	sess_count	min_len_sess	max_len_sess	avg_time_sess	uni_app_count	if_weekend	xhour	yhour	target
ente_time	-0.061716	0.004732	-0.055775	-0.036355	0.000719	-0.035170	-0.013336	-0.129264	0.554482	-0.085208	-0.006919	-0.066688	-0.041190	-0.011970	-0.026674	-0.075445	-0.207126	0.116568	0.531550	-0.180834	0.718020	0.657160	0.713801	-0.055016	0.028358	-0.055554	-0.084546	-0.122283
fina_time		-0.037548	-0.015654	0.156150	0.085352	0.111422	-0.029598	0.006608	-0.021964	0.821813	-0.030062	0.068798	0.151200	0.036880	0.074651	0.132818	0.069197	0.061601	-0.044043	0.227835	-0.025875	-0.091476	-0.050499	0.186104	-0.052590	0.079040	0.000694	0.134823
game_time			-0.038846	-0.003312	-0.007561	0.034076	0.157155	0.020312	-0.005768	-0.034057	0.943969	0.126160	-0.010119	0.070970	0.092902	0.128294	-0.003809	0.060367	0.188592	0.033372	-0.021970	0.029922	-0.020069	0.216517	-0.143225	-0.127968	-0.140465	-0.035249
life_time				0.009862	-0.003756	-0.014110	-0.104890	0.004790	-0.070497	-0.000250	-0.046529	0.192352	0.007400	0.025317	-0.039461	-0.090967	0.054159	-0.089872	0.333738	-0.065644	-0.023448	0.449815	0.342352	-0.097863	-0.057053	-0.111257	-0.058354	0.163769
shop_time					0.395473	0.120580	0.226848	-0.009486	-0.023799	0.150615	0.034195	0.124706	0.989899	0.215269	0.239132	0.235266	0.191439	0.182511	0.213181	-0.016053	-0.014605	0.040720	0.014233	0.382583	-0.080282	-0.076766	-0.028568	-0.097735
proc_time						0.106832	0.354639	-0.015546	0.097557	0.168065	0.146305	0.166139	0.383537	0.351377	0.151490	0.256115	0.085143	0.121073	0.290151	0.172976	-0.010670	0.187166	0.050367	0.357750	-0.088668	0.016826	0.020714	-0.024114
prod_time							0.069646	0.120801	-0.086205	0.176413	0.012817	0.312595	0.129113	0.207979	0.842773	0.182351	0.305019	0.307068	0.077907	0.163848	-0.032653	0.001467	-0.046399	0.422631	-0.110540	0.008810	-0.051443	-0.029062
soci_time								-0.012460	0.017520	-0.019613	0.257741	0.117475	0.224118	0.559008	0.217544	0.681583	0.172392	0.401167	0.703851	0.273041	-0.076101	0.367708	0.001510	0.508660	-0.037407	0.067399	0.140153	-0.074218
util_time									-0.001548	-0.013112	-0.007770	0.140170	0.000067	0.030070	0.102150	-0.007596	0.631792	0.083859	-0.014004	0.075328	-0.076545	-0.061282	-0.054113	0.213335	0.022863	0.036833	0.027908	-0.035492
ente_count										-0.001291	0.060168	0.053620	-0.030388	0.044881	-0.103695	0.066979	-0.019525	0.074251	0.297257	0.177659	0.419333	0.290122	0.379021	0.194284	0.102068	0.084999	0.099838	0.054319
fina_count											-0.017717	0.214380	0.143730	0.138033	0.091030	0.153229	0.092580	0.034772	-0.043512	0.281181	-0.036481	-0.109233	-0.071512	0.266548	-0.019516	0.131245	0.041507	0.179869
game_count												0.157145	0.024658	0.162535	0.059052	0.194893	-0.006249	0.070694	0.247703	0.086665	-0.031312	0.064724	-0.023153	0.294940	-0.076191	-0.029742	-0.037420	-0.008018
life_count													0.131738	0.347849	0.261769	0.245891	0.352589	0.164040	0.172218	0.311742	-0.061262	0.019545	-0.056379	0.494631	-0.210918	0.144223	0.128867	0.092215
shop_count														0.226094	0.236566	0.236191	0.181951	0.192139	0.206886	-0.004256	-0.016805	0.032310	0.010268	0.388017	-0.092832	-0.067790	-0.029894	-0.113013
proc_count															0.190576	0.699255	0.258065	0.300410	0.441697	0.475121	-0.040064	0.147457	-0.036356	0.566587	-0.155579	0.244030	0.205332	0.192345
prod_count																0.286718	0.267738	0.419802	0.184379	0.172294	-0.052055	0.018630	-0.059608	0.502318	-0.177698	-0.090549	-0.110389	-0.036699
soci_count																	0.230896	0.659597	0.445451	0.759531	-0.096556	0.008322	-0.149254	0.706516	-0.121756	0.223873	0.160057	0.117741
util_count																		0.329428	0.091668	0.297192	-0.099541	-0.036376	-0.090649	0.500687	0.032556	0.077260	0.096264	0.001885
noti_count																			0.352169	0.518484	0.103854	0.104257	0.034051	0.554927	-0.097562	0.142737	0.009736	0.055340
total_time_phone																				0.080870	0.339949	0.849244	0.559901	0.367804	-0.056885	-0.044616	0.013426	-0.062409
sess_count																					-0.172853	-0.265946	-0.291482	0.593415	-0.096654	0.382148	0.256858	0.303678
min_len_sess																						0.507069	0.890233	-0.092678	0.167190	-0.070555	-0.109247	-0.084380
max_len_sess																							0.779723	-0.025935	0.025154	-0.166197	-0.105677	-0.108710
avg_time_sess																								-0.138828	0.113662	-0.175538	-0.145019	-0.073142
uni_app_count																									-0.160753	0.200195	0.179125	0.025861
if_weekend																										-0.005070	-0.012321	0.078983
xhour																											0.863113	0.241978
yhour																												0.172580

Correlation matrix	Random3	fina_time	game_time	life_time	shop_time	proc_time	prod_time	soci_time	util_time	ente_count	fina_count	game_count	life_count	shop_count	proc_count	prod_count	soci_count	util_count	noti_count	total_time_phone	sess_count	min_len_sess	max_len_sess	avg_time_sess	uni_app_count	if_weekend	xhour	yhour	target
ente_time	-0.057881	-0.073808	-0.030549	-0.046203	0.365862	0.002804	0.268615	0.111793	0.317136	-0.037837	-0.066200	-0.014408	-0.046203	0.336821	0.287153	0.227287	0.219574	0.304188	0.622406	-0.026627	0.284779	0.590514	0.541069	0.308141	-0.027704	0.131597	0.063901	-0.123798	
fina_time		0.060165	0.079588	0.972925	0.135394	0.191846	0.054878	0.326892	0.010697	0.419386	0.028078	0.073007	0.972925	0.265600	0.561710	0.183628	0.469649	0.002336	0.099853	0.195068	-0.054642	-0.010869	-0.046657	0.253087	-0.073796	0.134949	0.123529	-0.068471	
game_time			0.043057	-0.027306	0.264065	0.071434	-0.089563	-0.001810	-0.057814	0.119200	0.615750	0.070357	-0.027306	0.019302	0.129802	0.082149	0.092039	0.041535	0.258593	0.131858	-0.078101	0.119669	-0.033962	0.137200	-0.118373	-0.009036	-0.065882	-0.104758		
life_time				-0.028419	0.040536	-0.053014	0.131102	0.461928	0.033408	0.315456	0.005894	0.891653	-0.028419	0.205765	-0.063322	0.348338	0.291749	0.281266	0.176263	0.041583	0.122692	0.147500	0.086534	0.172563	0.106954	-0.025838	-0.116184	-0.127605			
shop_time					0.082933	0.183382	0.045254	0.265365	0.027163	0.264602	-0.025744	-0.036052	1.000.000	0.261089	0.582912	0.133471	0.449662	-0.052163	0.056760	0.176115	-0.036349	-0.025827	-0.045558	0.197614	-0.073654	0.106808	0.114635	-0.062791				
proc_time						0.202545	0.237287	0.282981	0.120378	0.310736	0.098493	0.065100	0.082933	0.367715	0.572656	0.358377	0.507394	0.226849	0.464473	0.065487	0.003045	0.439332	0.357953	0.627085	-0.168513	0.165520	0.113352	-0.090027			
prod_time							0.132843	0.067737	0.032463	0.082055	-0.000516	-0.062984	0.183382	0.281077	0.500991	0.181421	0.206007	0.131860	0.145246	0.109077	-0.060066	0.084429	0.058118	0.312315	0.104705	-0.020090	0.014995	-0.011018			
soci_time								0.280037	0.089641	0.071751	-0.064898	0.082242	0.045254	0.505047	0.241809	0.753117	0.258848	0.547463	0.827770	-0.068209	0.231943	0.827737	0.724982	0.451096	-0.113702	0.222450	0.218538	0.057583			
util_time									0.053277	0.435128	-0.058244	0.452092	0.265365	0.213475	0.285059	0.313062	0.533312	0.203045	0.367676	0.134377	0.151895	0.270521	0.258910	0.383924	-0.045309	0.226962	0.210824	0.040154			
ente_count										-0.021030	-0.048338	0.239580	0.027163	0.323182	0.077944	0.173578	0.159410	0.132984	0.193195	0.770394	-0.086753	0.127690	-0.008182	0.330819	0.008049	0.073405	0.050475	0.048066			
fina_count											0.147488	0.335843	0.264602	0.014065	0.191088	0.204043	0.387218	0.255357	0.142540	0.160996	-0.110399	0.055206	-0.022903	0.372788	-0.057024	0.223537	0.154763	0.043496			
game_count												0.007096	-0.025744	-0.041170	0.032040	0.049795	-0.043318	0.000000	0.135700	0.193820	-0.071355	0.026783	-0.043823	0.157080	-0.137311	-0.060451	-0.141736	-0.087375			
life_count													-0.036052	0.198053	-0.057415	0.312773	0.303685	0.249579	0.153387	0.264608	-0.017800	0.104296	-0.014002	0.235515	0.173687	0.041842	-0.030992	-0.105967			
shop_count														0.261089	0.582912	0.133471	0.449662	-0.052163	0.056760	0.176115	-0.036349	-0.025827	-0.045558	0.197614	-0.073654	0.106808	0.114635	-0.062791			
proc_count															0.393429	0.741302	0.255232	0.366427	0.554982	0.195629	0.045006	0.400327	0.269025	0.480154	-0.064768	0.281022	0.217338	0.030311			
prod_count																0.271236	0.602193	0.141497	0.395676	0.086949	0.095735	0.369365	0.339529	0.538666	-0.096763	0.015265	0.010876	-0.097344			
soci_count																	0.391960	0.567848	0.711514	0.188318	-0.035445	0.566436	0.361310	0.598190	-0.170716	0.310030	0.249520	-0.032880			
util_count																		0.303582	0.392965	0.253981	0.008150	0.374064	0.241302	0.726796	-0.175959	0.099965	0.112157	-0.133874			
noti_count																			0.568801	0.104214	0.025687	0.556700	0.341030	0.514370	-0.062761	0.224202	0.189679	-0.141425			
total_time_phone																				0.007729	0.279833	0.922958	0.769526	0.630983	-0.135362	0.236703	0.178060	-0.055941			
sess_count																					-0.264397	-0.153869	-0.297741	0.379928	0.007520	0.181030	0.144150	0.025414			
min_len_sess																						0.269493	0.647354	0.013599	0.087120	-0.180480	-0.183300	-0.096646			
max_len_sess																							0.843361	0.547809	-0.109784	0.143837	0.108942	-0.080272			
avg_time_sess																								0.316699	-0.066477	0.042257	0.010711	-0.061811			
uni_app_count																									-0.186244	0.213998	0.206791	-0.037060			
if_weekend																										0.074727	0.048899	-0.108968			
xhour																											0.898148	0.038033		
yhour																												0.056962			

relation Matrix Top30	fina_time	game_time	life_time	shop_time	proc_time	prod_time	soci_time	util_time	ente_count	fina_count	game_count	life_count	shop_count	proc_count	prod_count	soci_count	util_count	noti_count	total_time_phone	sess_count	min_len_sess	max_len_sess	avg_time_sess	uni_app_count	if_weekend	xhour	yhour	target
ente_time	0.009872	-0.024818	-0.008830	0.014732	0.070290	0.005262	0.041109	-0.008890	0.443224	0.029484	-0.027567	-0.002523	0.048867	0.113272	0.032684	0.015624	0.055494	0.043543	0.547834	-0.047480	0.245070	0.580630	0.427811	0.177031	0.030358	0.001796	-0.007976	0.026712
fina_time		0.030434	0.031756	0.043417	0.035972	0.008069	0.029297	0.020218	0.064311	0.690017	0.003941	0.099034	0.048236	0.123178	0.075459	0.058780	0.108409	0.072691	0.062025	0.099274	-0.003783	0.013146	0.000423	0.206419	-0.062770	0.016783	-0.003319	-0.001871
game_time			-0.009088	-0.013196	0.045804	-0.003105	0.028282	0.000891	0.004610	0.018430	0.756916	0.028920	-0.016469	-0.031514	-0.000104	0.024151	0.038980	0.033015	0.189752	0.070571	-0.021821	0.104228	0.020926	0.147820	0.017968	0.014499	0.015887	0.051664
life_time				-0.002311	0.114740	0.017964	0.012592	0.038109	0.016852	0.063579	-0.002021	0.551379	-0.004963	0.119407	0.032186	0.045658	0.096472	0.074775	0.209147	0.146077	0.001502	0.190223	0.091690	0.177228	0.001524	0.004632	0.015020	0.031902
shop_time					0.036466	0.007909	0.069524	0.010540	-0.016428	0.076601	-0.008879	0.004131	0.776618	0.038202	0.014856	0.089955	0.031245	0.039112	0.083924	0.031501	-0.007245	0.030442	0.008387	0.104343	0.026193	0.010154	0.015326	-0.001535
proc_time						0.061578	0.104849	0.067318	0.058373	0.045925	0.031089	0.077492	0.031149	0.254047	0.097244	0.089989	0.173469	0.057591	0.189244	0.058829	-0.006139	0.138871	0.057550	0.258995	-0.005229	-0.003698	-0.001348	0.000704
prod_time							0.070794	0.047015	0.073285	0.011298	0.009440	0.010420	0.007597	0.112795	0.606831	0.110140	0.105488	0.064653	0.217836	0.033824	0.141431	0.190528	0.187229	0.186545	-0.030390	-0.003187	-0.000726	-0.005622
soci_time								0.066826	0.059444	0.008314	0.050883	0.014498	0.073383	0.230636	0.101260	0.603164	0.110264	0.293442	0.731696	0.150914	0.251477	0.608143	0.466484	0.379958	0.023541	0.042007	0.049382	0.039916
util_time									0.014914	0.018532	-0.008819	0.023587	0.015405	0.228118	0.080023	0.020579	0.480883	0.030006	0.359408	0.060922	0.037211	0.248307	0.127303	0.222862	0.007793	-0.009412	0.001173	-0.043311
ente_count										0.091635	0.001229	0.076830	0.002092	0.194015	0.215028	0.098279	0.166319	0.114698	0.294371	0.281384	0.028718	0.214779	0.085666	0.339390	-0.057646	0.016398	-0.000593	0.013617
fina_count											-0.003502	0.165622	0.091425	0.160393	0.069966	0.066667	0.160960	0.111844	0.058173	0.141285	-0.026403	0.008444	-0.019725	0.270921	-0.055593	0.019578	0.007700	0.013199
game_count												0.041136	-0.010794	-0.021705	0.035673	0.071840	0.022970	0.099277	0.158102	0.133955	-0.021401	0.061246	0.008464	0.158494	0.020333	0.009319	0.014295	0.037619
life_count													0.008116	0.103604	0.039509	0.099909	0.129016	0.132413	0.130498	0.370429	-0.006627	0.050863	0.011728	0.245296	-0.004934	0.047154	0.046516	0.032680
shop_count														0.094814	0.017885	0.105711	0.057172	0.046263	0.098769	0.039396	-0.012828	0.032024	0.006649	0.137344	0.008946	0.006002	0.004521	0.033296
proc_count															0.191578	0.354862	0.377645	0.165044	0.324665	0.245728	-0.009908	0.178030	0.058133	0.538989	-0.037295	0.029448	0.008652	0.067533
prod_count																0.169479	0.225967	0.141491	0.209264	0.186536	0.024106	0.135159	0.071027	0.354670	-0.073338	-0.028617	-0.041747	-0.014002
soci_count																	0.141744	0.573429	0.451643	0.592134	0.065391	0.246038	0.129619	0.521129	-0.057401	0.102131	0.054876	0.064304
util_count																		0.207743	0.296017	0.252058	0.010180	0.176830	0.073262	0.505580	-0.036777	-0.025523	0.005743	-0.001673
noti_count																			0.261164	0.479315	-0.012900	0.116413	0.010496	0.379386	-0.080754	0.031681	0.003484	0.002834
total_time_phone																				0.143672	0.322008	0.874420	0.625677	0.513761	0.032694	0.029669	0.034826	0.042278
sess_count																					-0.171766	-0.120049	-0.236507	0.502315	-0.114103	0.133080	0.078940	0.084195
min_len_sess																						0.496609	0.867414	0.009550	0.074232	-0.038440	-0.031817	-0.053134
max_len_sess																							0.809814	0.309110	0.057506	-0.026860	-0.004650	-0.007167
avg_time_sess																								0.116059	0.077629	-0.043749	-0.022837	-0.036774
uni_app_count																									-0.079824	0.032694	0.016568	0.101509
if_weekend																										0.026703	0.020770	-0.035896
xhour																											0.880488	0.036759
yhour																												0.018015

Correlation Matrix TopAll

	fina_time	game_time	life_time	shop_time	proc_time	prod_time	soci_time	util_time	ente_count	fina_count	game_count	life_count	shop_count	proc_count	prod_count	soci_count	util_count	noti_count	total_time_phone	sess_count	min_len_sess	max_len_sess	avg_time_sess	uni_app_count	if_weekend	xhour	yhour	target
fina_time	0.020526																											
game_time	-0.024018	0.006628																										
life_time	-0.017494	0.027372	-0.015646																									
shop_time	0.000607	0.022631	-0.016992	-0.004922																								
proc_time	0.064308	0.037232	0.031002	0.107516	0.004729																							
prod_time	-0.009824	0.022473	-0.009061	0.014373	0.019002	0.051952																						
soci_time	0.035905	0.026823	0.049419	0.003796	0.062426	0.086808	0.069946																					
util_time	0.040354	0.038350	-0.022440	0.045136	0.070939	0.072818	0.052938	0.057897																				
ente_count	0.441463	0.059301	-0.015140	0.017317	-0.006025	0.070893	0.046926	0.100951	0.072649																			
fina_count	0.042324	0.719531	-0.001639	0.074232	0.053090	0.056450	0.021364	0.029430	0.050162	0.105248																		
game_count	-0.029405	0.000693	0.754422	-0.009306	-0.014768	0.025978	0.006756	0.058269	-0.019935	-0.010677	-0.004253																	
life_count	-0.003056	0.085322	0.016960	0.504822	0.010262	0.094433	0.018902	0.026849	0.048551	0.084794	0.157252	0.033736																
shop_count	0.017035	0.043769	-0.021338	-0.004971	0.757099	0.012011	0.023348	0.079645	0.110652	0.017164	0.088236	-0.018233	0.015032															
proc_count	0.100376	0.065998	-0.013149	0.096691	0.006663	0.240195	0.093341	0.261179	0.207744	0.275153	0.100472	-0.003463	0.115149	0.044899														
prod_count	0.004068	0.066088	-0.016586	0.031046	0.070351	0.075514	0.596082	0.113871	0.097651	0.166809	0.083785	0.017598	0.050145	0.098609	0.158073													
soci_count	0.012477	0.058823	0.015350	0.031201	0.055703	0.110793	0.106762	0.606890	0.046737	0.201248	0.092069	0.053156	0.108391	0.088432	0.469988	0.198228												
util_count	0.065675	0.093908	-0.005228	0.089545	0.016533	0.057389	0.070633	0.302075	0.040782	0.162366	0.141526	0.000609	0.150240	0.108417	0.329111	0.222093	0.201214											
noti_count	0.039121	0.075756	0.042741	0.056788	0.100918	0.174912	0.192966	0.700312	0.396462	0.335380	0.095496	0.156792	0.149922	0.038263	0.125045	0.149762	0.514614	0.205335										
total_time_phone	0.567012	0.074479	0.196460	0.224513	0.033750	0.089392	0.070346	0.224276	0.088071	0.326111	0.142521	0.088822	0.333792	0.124319	0.325465	0.194844	0.439872	0.329616	0.260214									
sess_count	-0.042398	0.088072	0.022139	0.093983	-0.013091	-0.000445	0.113174	0.206441	0.039668	0.008212	-0.028565	0.007018	0.006137	0.063005	-0.000098	0.721798	0.659990	0.313726	0.470064	0.187600								
min_len_sess	0.308632	-0.012156	0.078178	0.027966	0.046927	0.126750	0.155526	0.561474	0.281486	0.228720	-0.016267	0.084926	0.063781	-0.019212	0.164901	0.010916	0.030712	0.005644	-0.020282	0.288936	-0.167150							
max_len_sess	0.609842	0.028365	0.158510	0.184444	0.010046	0.061251	0.148918	0.410002	0.151014	0.081628	0.030662	0.038863	0.025617	0.044834	0.061075	0.104234	0.202615	0.184831	0.115848	0.871080	-0.095719							
avg_time_sess	0.442485	-0.001429	0.115512	0.115538	0.144651	0.240498	0.203806	0.391370	0.270541	0.352996	0.276552	0.127195	0.270703	0.188982	0.453141	0.388230	0.078549	0.070427	0.003022	0.616572	-0.228096	0.847372						
uni_app_count	0.160497	0.191137	0.102615	0.168844	0.033139	0.012411	-0.001830	0.050396	0.005396	-0.042177	-0.047970	0.005747	-0.009242	-0.004210	-0.036824	-0.083265	0.544244	0.534609	-0.082838	0.521492	-0.105063	0.557424	-0.004049	0.103713				
if_weekend	0.029869	0.018611	-0.051764	0.010653	0.000057	-0.000840	-0.032497	0.011021	0.009671	0.027196	0.019074	0.004332	0.055262	0.007512	0.031011	-0.032209	-0.059822	0.034662	0.054743	0.024788	0.135679	0.075212	-0.043220	-0.035934	-0.085716			
xhour	0.018611	0.006462	-0.005105	0.029833	0.013259	0.081867	-0.008877	0.050396	0.016784	0.007087	0.005003	0.012018	0.056281	0.011047	0.009146	-0.053183	0.051586	0.003701	0.021135	0.050741	0.075212	0.084885	-0.044193	-0.016376	0.052840	0.034313		
yhour	0.006179	-0.006308	0.004304	0.037954	-0.015064	0.011468	0.010911	0.048801	-0.017315	0.060750	-0.026844	0.017468	0.044018	0.017487	0.049404	0.007550	0.072509	0.001939	0.022738	0.041153	0.084885	0.044193	-0.033684	-0.027733	0.059423	0.028139	-0.032701	
target	0.004344	-0.016836	0.019312	0.059019	-0.015064	0.011468	0.010911	0.041908	-0.017315	0.027060	-0.026844	0.017468	0.044018	0.017487	0.049404	0.007550	0.072509	0.001939	0.022738	0.041153	0.084885	0.044193	-0.033684	-0.027733	0.059423	-0.032701	0.035062	0.016710

Appendix B

3.2.1 Duplicate data

The phone dataset consists of 586.792 entries, in this data there are 19.280 erroneous duplicate entries. The mood survey dataset contained 14 duplicate entries. These duplicate entries are removed from the data.

3.2.2 Erroneous User ID's

The mood survey dataset consists of 16.002 entries from 149 different users. There are 124 unique user IDs in the phone usage dataset and 149 unique user IDs in the mood survey dataset. This indicates that there are 25 erroneous user IDs in the mood survey dataset. These 25 user IDs are responsible for another 1.949 surveys. These surveys are excluded from the data. This reduced the sample size from $n_t = 16.002$ to $n_t = 14.053$.

3.2.3 Expired, double, blocked and cancelled surveys

The survey data also consists of non-filled and erroneous filled-in surveys. The moment that the respondents received the surveys, they were given two hours to complete it. After this moment the surveys would expire. The dataset consists of 5.372 returned expired surveys that are empty and 33 expired surveys that are partly filled-in. The entries that are partly filled-in miss multiple data and are not workable for analysis. Furthermore the respondents were also given the possibility to block or cancel a survey. There are 128 surveys in the dataset that are blocked or cancelled. All these expired, cancelled and blocked surveys are removed from the dataset. This reduced the sample size from $n_t = 14.053$ to $n_t = 8.520$.

3.2.4 Failing software

Six of the respondents in the study have not returned a single survey while their phone was being logged. It is unclear what the cause of this is, but it seems that it occurred due to failing software. These users have sent 241 surveys, these surveys are omitted from the data.

Due to incorrect settings some of the participants also already send surveys before the phone was logged, but also kept receiving and replying surveys after the study was over. A total amount of 1.026 surveys was due to this useless. This brings the total amount of surveys from $n_t = 8.520$ to $n_t = 7.253$.

Another 29 respondents have returned surveys with the Ethica application at times that their phone was logged but the MobileDNA application did not register telephone use. It suggests that at least either of the two applications has malfunctioned. It was impossible to retrieve

whether the applications only malfunctioned at these moments or during the whole study period, therefore these respondents are excluded from the data. These respondents are responsible for 930 surveys, which brings the total amount of surveys from 7.253 to 6.323.

3.2.5 Outliers

Two applications were excluded from the data and placed in the 'wrong' category. One of these applications is 'com.madein.coinpotapp', which is an application that mines bitcoins. This application runs on the background and is not actively used by the phone user but seen as actively used. Only one respondent had this applications installed. The other application is 'ru.woxTGdZL.IfEhxxjCz' which is an unknown application that is used by only one user for a limited amount of times(n=5). The two applications that are placed in the wrong category are will not be included in the data for future analysis.