PREDICTING STRESS USING SMARTPHONE USAGE DATA

BIDUS PLOMP

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

STUDENT NUMBER

763258

E-MAIL ADDRESS

b.b.plomp@tilburguniversity.edu

COHORT

Spring 2022

WORD COUNT

8669

COMMITTEE

dr. Drew Hendrickson
dr. Görkem Saygili

LOCATION

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

DATE

July 1, 2022

PREDICTING STRESS USING SMARTPHONE USAGE DATA

BIDUS PLOMP

Abstract

Stress levels have been on the rise in recent years. With a proven negative relation between stress and health, action must be taken. Therefore, this study aimed to investigate to what extent it was possible to predict stress levels using passively logged smartphone data. The data used in this study consisted of the smartphone usage logs of 227 students and their responses to a questionnaire about their mental health. The first step in this study was to investigate the best way of representing the smartphone usage data. This was done by testing different feature representations in combination with various machine learning classification models to determine which combination most accurately predicts perceived stress levels. The results showed that the best results were found for the feature representation with time and count per app category and the Random Forest model. Subsequently, it was investigated whether oversampling of the minority class by means of SMOTE, a technique not previously used in the relevant literature, produced better results. The results showed that the use of this technique indeed yielded better results. Furthermore, research that used personal information showed similar outcomes, though it scored slightly lower than the research that also used physiological sensors. In conclusion, this suggests that stress prediction using only smartphone user data did not achieve the same results as the current standard, however a step in the right direction has been made and further research is suggested.

1 DATA SOURCE/CODE/ETHICS STATEMENT

Work on this thesis did not involve collecting data from human participants or animals. The data that was used for this project was provided by the Thesis Supervisor, which remains the original owner of the data during and after completion of this thesis. I fully acknowledge that I do not have any legal claim to this data. Moreover, this data is not publicly available. Also, the code used in this thesis is not publicly available. All images in this study are self-made, or images that have been inspired by work from others have been referenced accordingly.

CONTENTS

2 PROBLEM STATEMENT & RESEARCH GOAL

2.1 *Context*

Almost all Dutch students experience stress, of these students more than half experience high levels of stress (Ministerie van Onderwijs, Cultuur en Wetenschap, 2021). According to Hudd, Dumlao, Erdmann-Sager, and Murray (2000) there is a negative relationship between perceived stress levels and healthy habits among young adults. Furthermore, high stress levels are even linked to major depressive disorder and other diseases such as AIDS (Cohen, Janicki-Deverts, & Miller, 2007). Accurately forecasting stress levels enables individuals to make suitable behavioural changes such as moderate aerobic exercise, mindfulness and deep breathing techniques (Varvogli & Darviri, 2011). Therefore, this has the potential to reduce high stress in the future.

However, to accurately predict their stress levels, individuals must consistently monitor and report on many different aspects of their lives and/or there is a need for professional medical measurement equipment (Hellhammer, Stone, & Broderick, 2010). With the growing use of technology in every area of human life, researchers have started to investigate whether technology can solve this problem. In the past decade, researchers started developing algorithms trying to accurately forecast stress using sensors and data collected from smartphones (Fukazawa et al., 2019). The possibility to forecast stress could be immensely beneficial, especially if such a forecast could be accurately done using data collected in an unintrusive and passive way. This current study will try to precisely achieve this.

2.2 *Scientific & Societal relevance*

A large worldwide population study (160,000 people in 116 countries) found that 40% of people experienced worry or stress, the highest it has been in over 15 years (GALLUP, 2021). With clear evidence that high levels of stress are related to mental health problems (Schönfeld, Brailovskaia, Bieda, Zhang, & Margraf, 2016), stress levels need to be reduced. For the very reason, that forecasting stress has the potential to combat chronic stress, predicting stress through passively logged smartphone usage data is a very cost-effective and unintrusive approach.

From a scientifical point of view, this study has a different approach to predicting stress levels. Previous research often used wearable sensors and/or data that participants needed to report themselves, such as activity level and/or social interactions. This study will only use passively logged

and privacy-sensitive data. If this study shows that this works as well as, or not much less than, the current standard, it opens up the possibility of widespread use of this technique.

2.3 *Research questions*

To investigate whether and to what extent it is possible to correctly predict stress levels, the following research question was formulated:

> *To what extent is it possible to predict perceived stress levels among Dutch students, based on their smartphone usage logs?*

For answering the main research question, four sub-questions are formulated. The scientific background for each sub-question will be further elaborated in Section 3.1. Research questions 1 and 2 were studied together, this was done by testing the different feature representations on all models. This enabled the possibility to find the best combination of model and feature representation.

RQ1 *What is the best method of the tested methods in representing the smartphone usage logs in order to realise a high Balanced Accuracy in predicting stress levels?*

The literature used different approaches in transforming the smartphone logs into a dataset, time per app category, the number of times an app is used per app category, or both are used. Also, different types and quantities of categories are used to categorise the apps. By comparing these different approaches, this sub-question aimed to investigate which yields the highest results.

RQ2 *Which of the following models most accurately predicts stress levels; Support Vector Machine, k-Nearest Neighbors, Random Forest, or Gradient Boosting Machine?*

To answer this question the models were compared to each other, see Section 4.8 for a description of the models and the reason these models are selected. The main evaluation metric for model performance was Balanced Accuracy besides that Accuracy was also used, see Section 4.6 for a further elaboration of these metrics.

RQ3 *To what extent does balancing the data using the SMOTE method improve the Balanced Accuracy?*

Because the data was imbalanced, this sub-question aimed to explore whether balancing the data using the Synthetic Minority Oversampling Technique (SMOTE) increased the overall Balanced Accuracy. This technique has not been used in related research before but is showed promising results in other research. Therefore, it was explored in this study, see Section 3.1 for further explanation. Whether SMOTE yielded better results was studied for the best feature representation and model combination of the abovementioned research question, see Section 4.2 for an overview of the research pipeline for additional clarification.

RQ4 *How are the errors distributed among the predicted classes?*

An error analysis was done to examine the errors of the predicted classes, where the main focus was to investigate the false negatives (predicted 'not stressed' instead of 'stressed') because that is worse than a false positive (predicted 'stressed' instead of 'not stressed'). The reason a false negative is worse than a false positive is that the relevance of this study is the potential of being used for stress prevention. Therefore, it is more important that high stress levels are correctly predicted than potential false positives.

2.4 *Findings*

This study demonstrated that it is possible to predict stress levels with a Balanced Accuracy of 0.466 and an Accuracy of 0.734. This result is achieved by using the dataset with time and count per app category. Moreover, the results showed that oversampling of the minority classes by means of SMOTE increased the overall scores. The best model to predict stress levels was found to be a Random Forest.

3 LITERATURE REVIEW

Predicting stress using smartphone data, personal information and physiological parameters

Previous studies have reported that stress is correlated to a person's physiological parameters such as respiratory rate, heart rate, skin temperature, pupil size and eye activity (Giannakakis et al., 2022; Vrijkotte, van Doornen, & de Geus, 2000). Several studies have used this knowledge to predict stress levels using a wristband with integrated sensors.

The two most prominent studies in this area are the studies of Umematsu, Sano, Taylor, and Picard (2019) and Jaques, Taylor, Nosakhare, Sano, and Picard (2020). Umematsu et al. (2019) used a wristband to measure skin conductance, skin temperature and it had an accelerometer in order to measure different kind of movements. In addition, the researchers used data from the participant's smartphone to calculate features about the timing, type, and duration of phone calls and SMS messages. Besides that, data from a survey was collected about the participants daily activities, social interactions and sleep. Jaques et al. (2020) used comparable data, with the addition that this study also used weather information.

To forecast binary stress levels, Umematsu et al. (2019) used a Long Short-Term Memory (LSTM), SVM and LR model, with the best performing model being the LSTM with an Accuracy of 83.6%. The researchers furthermore point out that by using only the data from the wristband and smartphone there is not a significant difference in Accuracy. This implies that a similar result could have been achieved with less data. Jaques et al. (2020) used a similar approach in predicting stress by using Multi-task Learning (MTL) to predicted binary levels for happiness, stress and health. The most striking observation from their results was that by accounting for individual differences between the participants in the MTL modelling by using Hierarchical Bayes with Dirichlet Process Priors (HBDPP) the Accuracy can be improved on and reach 86.07%. Which shows that by modelling individual traits, higher results can be obtained.

However, there are some major drawbacks to using physiological sensors. For instance, the skin conductance sensor must be held firmly to the skin, thereby restricting freedom of movement (Roh, Bong, Hong, Cho, Yoo, 2012). Furthermore, in order to predict the stress level of tomorrow these sensors have to monitor continuously, meaning they have to be worn at all times.

Predicting stress using smartphone data and personal information
Other researchers have shown that it is still possible to predict stress with smartphone data and personal information. Bogomolov, Lepri, Ferron, Pianesi, and Pentland (2014) have reported that they are able to recognise daily binary stress level reliably based on smartphone data and personality traits. These personality traits were measured using the Big Five, which divides personality into five main components: Extraversion, Neuroticism, Agreeableness, Conscientiousness, and Openness to experience. Besides that, smartphone data was used, this consisted of various features related to calling and texting. Additionally, the Bluetooth sensor in the smartphones was used to determine how often a participant encountered new or known people. According to the researchers this gave insight into how active the social life of the participant was, a good predictor of stress according to psychological research.

The models used to predict the stress levels where a Random Forest (RF), Gradient Boosted Model (GBM), SVM and Neural Networks. They concluded that a RF was the best performing model. However, it did not achieve the highest Accuracy (72.28% for RF and 84.86% for GBM). The research stated that, the RF was the best model because it was better at predicting stress. Meaning, RF correctly predicted that a participant was stressed more often. As a result of this finding, it was decided to use the same models in the current study as well.

Furthermore, Bauer and Lukowicz (2012) have used similar features to investigate whether behavioural changes can be detected. By conducting an experiment in which 7 students who were monitored during a two-week stressful period (exam sessions) and then two weeks of little to no stressful period. Conclusively, they demonstrated that there was a behavioural differentiation between the two periods. Although, their research has not been conclusive in establishing a model that can actually predict stress, it did show that there is a scientific foundation for using this kind of data.

Predicting Stress using only smartphone data
Since the following research used the same type of data as this current study, the design of the study and the methodology will be discussed in more detail. Osmani, Ferdous, and Mayora (2015) investigated to what extend it was possible to predict binary stress levels on the work floor. This was done by collecting smartphone usage data. After which, the researchers categorised the apps which the participants used in five categories: entertainment, social networking, utility, browser, and game apps. Then, for each of these categories, they calculated how often an app was opened and how much time was spent per category.

The smartphone usage data was combined with the stress levels of the participants in order to train an SVM classifier on the complete dataset. This was done by first splitting the data into train/test and tuning of the SVM is done by cross validation on the train dataset, resulting in a test Accuracy of 54%. Since the research question and data set of Osmani et al. (2015) are very comparable to this study, the research method will be similar. However, Osmani et al. (2015) concluded that a participant-specific model performed better than a generic model (88.1%). It is nonetheless not feasible to explore this within this current study and will therefore be out of the scope.

However, at the time of the study of Osmani et al. (2015), which was conducted several years ago, smartphones where not as widely used as nowadays. In that study participants used an average of 12 unique apps in total, whereas millennials nowadays use 67 unique apps per day (Nick, 2022). This enormous increase shows that people use smartphone apps for many tasks, which is why this study will investigate whether dividing the apps into more categories will yield better results. This hypothesis is supported by the work of Stütz et al. (2015), which examines the correlation between smartphone usage patterns and self-reported stress levels. Their research suggests that the more detailed the smartphone usage can be represented, the stronger the correlation with stress levels.

3.1 *Relevance research questions & research gap*

Multiclass vs binary

All of the research discussed in this section uses a binary outcome variable (stressed/not stressed), however long-term exposure to moderate stress levels has also been linked to negative physiological effects (Dalla et al., 2005). Consequently, this study used a multiclass target variable, with the objective of exploring the feasibility of differentiating between moderate stress and high stress levels.

Imbalanced data set

It is assumed that most people are not stressed most of the time, which means that there was very likely an imbalance in the dataset, what for some models is problematic (Provost, 2017). In the described literature, there was little or nothing to find about the balance between stressed and not stressed. Bogomolov et al. (2014) and Osmani et al. (2015) described that they used a Likert scale to assess whether participants were stressed or not, and that the data was skewed to the low levels of stress. Although it was unclear how they transformed the Likert-scale to a binary variable.

Considering that the literature described above did not address how to handle unbalanced data, research that examined this was explored. Research by Ren et al. (2021) that used machine learning to investigate psychological impact of COVID-19 on college students, used SMOTE to oversample the minority class. This SMOTE technique was used in diabetes prediction as well because there the data was also frequently unbalanced. To sum up, several studies in predicting diabetes showed that oversampling the minority class using SMOTE yields better results (Alghamdi et al., 2017; Nguyen et al., 2019; Shuja, Mittal, & Zaman, 2020).

For the reason that various research showed good results using SMOTE, this study examined whether a higher Balanced Accuracy could be achieved by means of Synthetic Minority Oversampling Technique (SMOTE).

Error Analysis

Bogomolov et al. (2014) concluded that a RF model was better in predicting participants correctly as stressed then a GBM model, which showed that some kind of error analysis was done. Yet, this were the only researchers who provided further insight into the model performance besides the evaluation metrics. However, it was expected that a more extensive error analysis can provide valuable insights, for example how often the model incorrectly predicts that someone is not stressed. As this was in fact a more significant error than predicting incorrectly that someone is not stressed.

4 METHODOLOGY & EXPERIMENTAL SETUP

4.1 *Data*

The dataset used for this study originates from another study which investigated the relationship between smartphone usage and mental health of the students at Tilburg University (Aalbers, vanden Abeele, Hendrickson, de Marez, & Keijsers, 2021). The data includes 227 unique participants (this is after cleaning see Section 4.3.2), with a mean age of 21 (*SD* 2.8) and the majority of participants identified as female (54.6%). The data is composed of two parts, one of which is a questionnaire. For two blocks of 30-day periods participants had to fill in the questionnaire 5 times a day, asking them about their procrastination, fatigue, stress, and happiness. This is done using a 7-point Likert scale where point 1 corresponds to 'Not at all stressed', point 4 to 'Moderately stressed and point 7 to 'Very much stressed'. Because the study focused on mental health, not all features of this questionnaire were relevant for this study. Table 1 shows the relevant features and a short description.

Feature	Description
ID	Unique ID of the participant
IssuedTime	Date and time for questionnaire notification
S1	Response on 7-point Likert scale to: "I feel rushed"
S2	Response on 7-point Likert scale to: "I feel relaxed"
S3	Response on 7-point Likert scale to: "I feel stressed"
Age	Age of the participant
Gender	Gender of participant: male (1)/female (2)/other(3)

Table 1: Features questionnaire

During the period when the participants had to answer the questionnaire, the mobileDNA app was used to log the participant's smartphone usage. For every participant, there are two CSV (Comma Separated Value) files, one containing the logs from the apps they used and the other contained the notifications they received.

The CSV files for the app events had the structure that every time an app is opened, a new row is constructed with the features of that event in the columns. Table 2 shows which information is logged every time an app was opened and the corresponding feature names.

Feature	Description
ID	Unique ID of the participant
startTime	Date and time application is opened
endTime	Date and time application is closed
Application	Name of application
Notification	True/False input whether application was opened because of notification
Battery	Battery level in percentage at time of app event

Table 2: Features app logs

The CSV files for the notifications were constructed in a similar way as the ones of the app events, meaning that every time there was a notification a row was added to the logs with the features described in Table 3.

Feature	Description
ID	Unique ID of the participant
Time	Date and time of notification
Application	Application of the notification
Posted	(True/False) whether a notification is visible by user

Table 3: Features notifications

4.2 *Pipeline Overview*

This section provides a concise overview of how this study is structured, after which key components are further elaborated upon in the sections that follow. Figure 1 shows a flowchart of the research methodology, the first step was pre-processing of the data and construction of several different datasets. Different datasets are constructed to be able to test whether more or fewer app categories give better results and test whether time per category, count per category or a combination of these will give better results. Then each dataset was split into a train and test set (see Section 4.5 for how the data was split). The test set was kept separate until it was needed at the end to test the definitive model and perform the error analysis. The train data was used to test the various dataset model combinations through cross-validation. The highest-scoring combination was used to investigate whether oversampling the minority class increases Balanced Accuracy, and prior to testing the model on the test set, RandomizedSearchCV was used to find the best hyperparameters. So only the best performing model dataset combination with and without

SMOTE will be tested on the test set after HP tuning. Finally, after testing the best model, the error analysis and feature importance were evaluated.

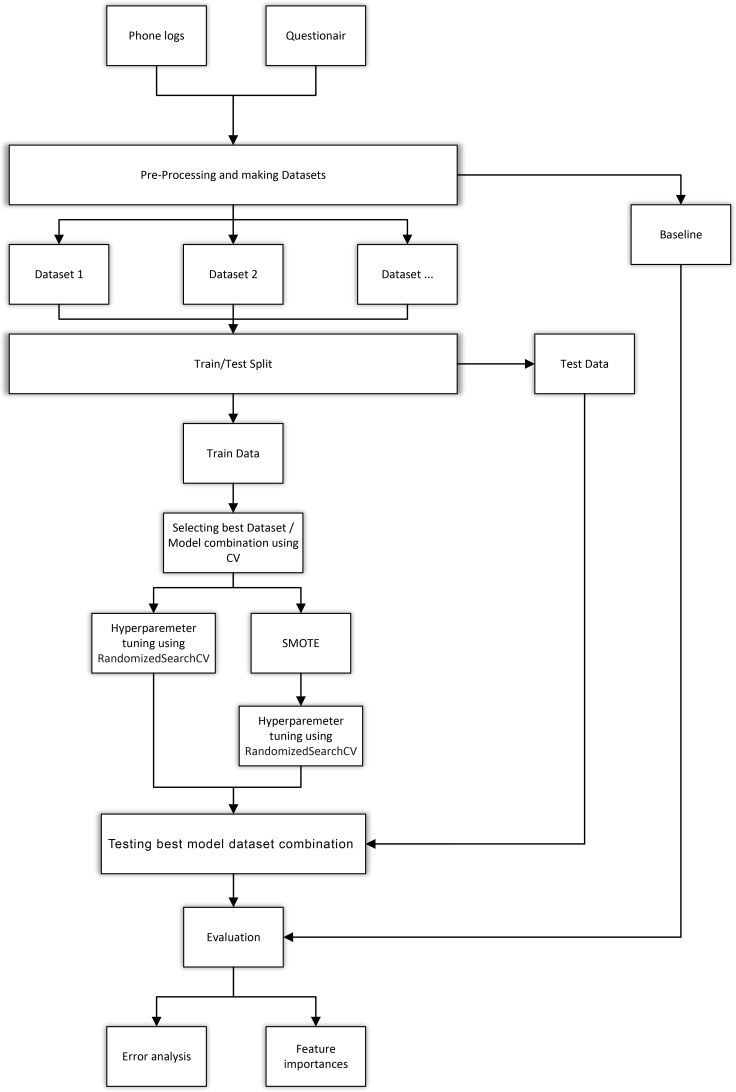

Figure 1: Pipeline overview

4.3 *Data Pre-processing*

4.3.1 *7-point Likert to multiclass stress variable*

As stated in Section 3.1, this study used a multiclass target variable ('not stressed', 'moderately stressed' and 'stressed') to achieve this the 7-point Likert scale response to the three questions was recoded into the specified target variable. This was done corresponding to the rules described in Table 4. This is done from top to bottom, so if one of the variables indicated stress, this data point was labelled as stressed. For example, a score of six for the stressed variable and a score of three for the rushed variable would result in the label stressed. This approach resulted in the final labels having a bias towards 'stressed'. But the fact that more data points are labelled stressed was not a problem because the study aimed to recognise stress.

Output	Conditions per variable		
	Stressed	Rushed	Relaxed
Stressed	7, 6	7, 6	1, 2
Moderately stressed	3, 4, 5	3, 4, 5	3, 4, 5
Not stressed	1, 2	1, 2	6, 7

Table 4: Rules used for transforming 7-point Likert scale to depended variable

Figure 2 shows the distribution of the initial variables and the constructed final multiclass stress variable.

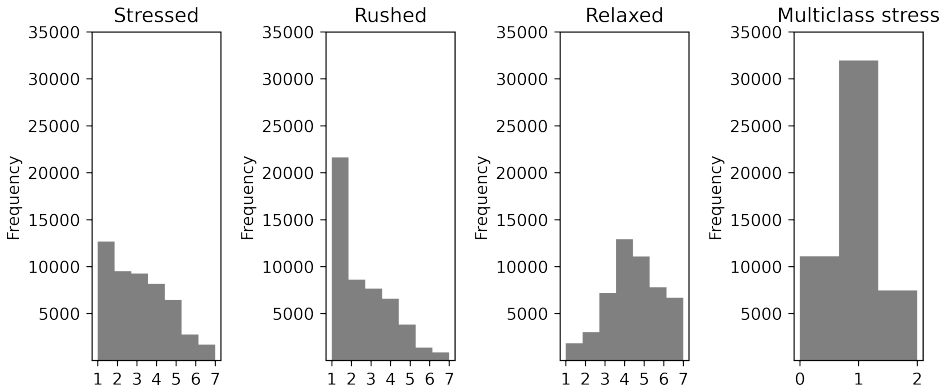

Figure 2: Variables distribution

4.3.2 *Missing data imputation*

For the constructed stress variable there was around 25% (18417 of 69880 rows) of the data points missing. The first step in solving this problem was understanding what caused the participants to complete the questionnaires. To create the dataset, the feature IssuedTime was used, which was the date and time when the participants received a notification to complete the questionnaire. However, if some participants decided to stop participating in the study, they still received these notifications some time, resulting in missing data.

Figure 3 shows for every participant how many days they filled in the questionnaire. It illustrates that most of the participants filled in the questionnaire for 60 days, which corresponds to two times 30 days of the study design as described in Section 4.1. After that there was a decline in the days of participation, there was a group that participated for 30 days. Meaning they only participated in the first 30 days of the study. The horizontal line represents the cut-off point (of 20 days) below which participants were no longer included in this study.

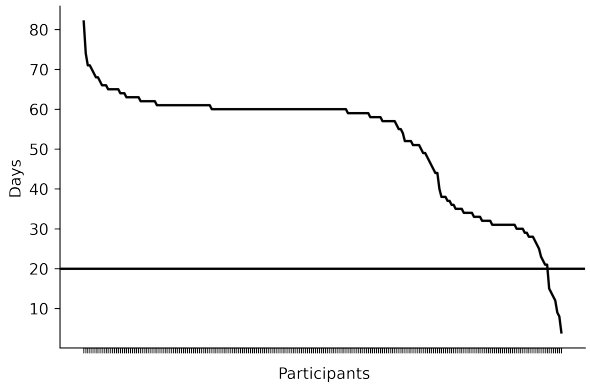

Figure 3: Questionnaire participation

Another reason for missing data is that at random moments, participants do not complete the questionnaire but then continue to participate. These missing data points are also referred to as Missing Completely At Random (MCAR), meaning that the cause of the data missing is unrelated to the data itself (van Buuren, 2021). These missing data points were calculated using spline interpolation, which fits low-degree polynomials to a subset of the data to calculate the missing data points. The reason that this interpolation was used is that the order of the data is important, meaning that if a participant was stressed in the morning and afternoon the probability of being stressed in the evening was high. Figure 4 shows an example of three different interpolation techniques; linear, polynomial and spline. In this

example, it can be seen that spline best models the relationship between the data points, which is the reason why this technique was chosen.

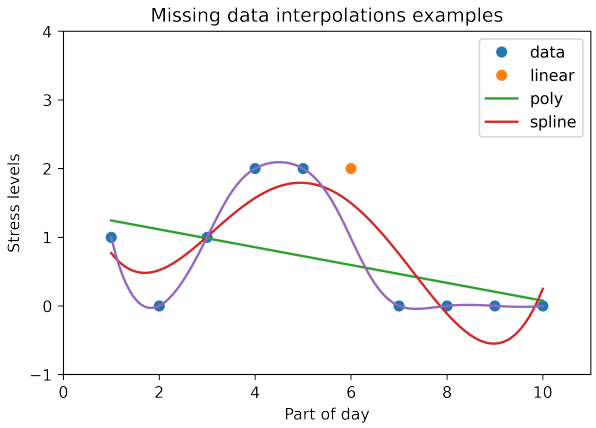

Figure 4: Missing data interpolation

The limit for the maximum number of consecutive missing data points was set to five, which means that spline interpolation was only used if less than a day's amount of data was missing, as otherwise it becomes less reliable (Kong, Siauw, & Bayen, 2020). For this interpolation, it was not possible to distinguish between the beginning and the end of the day, which means that a missing data point at the beginning of the day is interpolated from the data of the previous day.

Before the missing data interpolation for all the participants (227), there were missing data points. With a maximum of 256 missing data points for one participant and an average of 78 (*SD* 57.7). After the interpolation, only 83 participants had missing data points left with a maximum of 163 and an average of 51 (*SD* 59.0). After both techniques were applied, only 0.05 percent of the total missing data was still missing. To resolve this listwise deletion (deleting the rows with missing data) has been applied (Allison, 2000).

4.4 *Construction Datasets*

4.4.1 *Categorising apps*

To categorise the apps, a dataset containing 2.3 million of the most popular apps in the PlayStore was used. This dataset was constructed and provided by G. Prakash and is accessible under the following link `https://github.com/gauthamp10/Google-Playstore-Dataset`. The dataset contained 48 different categories of which the largest categories are Education, Music Audio and Tools. Appendix A shows the complete list of the categories and the count of apps per category.

Because this study aimed to investigate whether more or fewer categories performed better, a dataset with fewer categories is also created. This has been done by combining categories, for example, all the different types of gaming categories have been combined into one large gaming category. Table 5 shows which categories were merged to create the dataset with less categories. This dataset with less categories contains 18 distinct app categories which are shown in Appendix B.

Merged category	Old category
games	Puzzle, Role Playing, Adventure, Action, Arcade, Casual, Simulation, Board, Word, Racing, Card, Strategy, Trivia, Comics, Casino
music_other	Music & Audio, Music, Video Players & Editors, Photography
social	Food & Drink, Travel & Local, Social, Dating, Events, Sports
health	Health & Fitness, Beauty, Medical
business	Business, Productivity, Books & Reference

Table 5: Merging of categories

4.4.2 *Part of day*

As described in Section 4.1 the smartphone usage data were logs and they needed to be transformed into a useful dataset. For each user, there were two of these user logs, one representing the used apps and the other one the notifications. In addition, there was also the questionnaire dataset, see Section 4.1 for a description of all these datasets. In order to conduct this study, all these datasets had to be merged into one. Figure 5 shows the questionnaire response per hour, it shows that the response happened in five bursts (the different colours). For this reason, the response had been grouped into 5 parts of the day (shown in Figure 6).

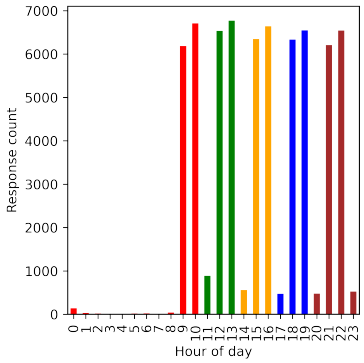

Figure 5: Questionnaire response per hour

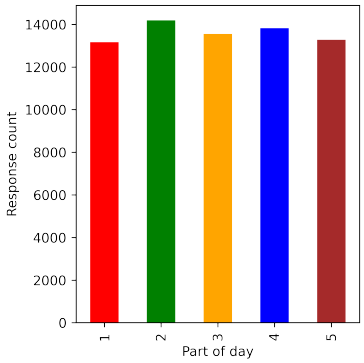

Figure 6: Questionnaire response per part of day

Each part of the day consisted of three hours, except for the ones at the beginning or end of the day. Table 6 shows when each part of day begun and ended.

Part of day	Start and end time	Total time span in hours
1	From 00:00 until 10:59	11
2	From 11:00 until 13:59	3
3	From 14:00 until 16:59	3
4	From 17:00 until 19:59	3
5	From 20:00 until 23:59	4

Table 6: Start and end time per par of day

As the smartphone logs were aggregated in the same way per part of day, this had the potential of using smartphone usage data after completion of the questionnaire to predict the stress level. Table 7 shows per part of day the average time difference to the maximum time. This shows that for the most part of days the average time of smartphone data used after

completing the questionnaire is about 35 minutes. However, part of day five was a lot higher at 86 minutes, but this can be explained by the fact that this part of day covers a larger amount of time (see Table 6).

Part of day	Mean time difference to max time (in minutes)
1	37.8
2	35.1
3	33.1
4	34.5
5	85.7

Table 7: Average time difference to the maximum time per part day

However, in the relevant literature, it is unfortunately unclear how this limitation was resolved. The study of Bogomolov et al. (2014) also used a questionnaire for the collection of stress levels but did this only once in the evening. However, the smartphone data of the whole day was used, meaning that some data after filling in the questionnaire was used.

Because the notification to fill in the questionnaire came at random moments, there was a chance that two measurement moments were allocated in the same part of the day. The start and end times as described in Table 6 resulted in 17 cases where two part of the day were the same. However, in these cases, the notification was just before or after the part of day window. Therefore, these cases were solved manually by assigning them to the part of day that was the closest.

4.4.3 *From Logs to Dataset*

The first step, to go from the logs to the final dataset, was to replace the app name with the category it belongs to. Next, the time feature was used to add an extra column in which the time was converted into a part of the day (with the start and end time of Table 6). To get to the time per app category the Pandas pivot_table function was used to calculate the time per app category for each unique participant per part-of-day. To obtain the count per app category, the column with the part-of-day was added first, after which a list of all app categories per specific part-of-day was created using the Pandas Groupby function. Then, using Sklearn's CountVectorizer, this list was transformed into a column for each app category, containing the count of how often an app in that category was opened. The reason for using this somewhat complicated technique was that in practice it turned out to be the fastest way. Also, it made sure that the columns were the same for every participant, so even if a participant

did not use a certain app category, this way of transforming still made sure that there was a column for this app category but then with a zero in it. This was essential for the later stage of merging all datasets requiring the same columns. The same approach as count per app category was used for the logs of the notifications; add a part-of-day column and then sum up the number of notifications per part-of-day. As the questionnaire questions were also asked five times a day, the app category (count and/or time) and the number of notifications were then aggregated per unique participant and per part-of-day, to create the final dataset. The process described above was repeated twice, each time for the different app categories (see Section 4.4.1 for information about these categories). This resulted in a total of six datasets, described in Table 8.

Dataset	Description
1	Count for many (48) categories
2	Time for many (48) categories
3	Time & count for many (48) categories
4	Count for less (18) categories
5	Time for less (18) categories
6	Time & count for less (18) categories

Table 8: Datasets

4.4.4 *Additional Features*

Besides the time and/or count for the app categories some additional features are either constructed or already were in the phone logs.

Total
For both the time and count dataset a total column was added, so for time that column contains the total time a participant was on their smartphone for a specific part of the day. Consequently, the dataset with both time and count had two total columns, one with the total time and another with the total count.

App notification
As described in section 4.1 the app usage logs also contain a feature about whether a participant opened an app because of a notification. This feature was also used in the final dataset as a sum of apps that are opened because of a notification.

Battery variance
The app usage logs also contained a feature about battery level, every time an app was opened the battery level at that specific time was logged. This feature was used to calculate the variance in battery levels per part-of-day.

Weekend
Based on the initial date feature an extra feature was created, this feature was a binary feature about whether a specific day is a weekend day er not.

4.5 *Out-of-sample model evaluation*

As described in Section 4.2 for the final model evaluation the data was split into two sets, a training set (70%) and a test set (30%). This dataset split was made with the stratified train-test split from sklearn. This stratification technique ensures that the distribution of the target variable is the same among the different splits. According to Kohavi (2001) this is especially important for an imbalanced dataset which is the case in this study (see Section 4.3.1) and may yield a better result than a standard train-test split.

4.6 *Evaluation Metric*

The most used evaluation metric for multiclass classification is Accuracy (Tsoumakas & Katakis, 2007). Consequently, it is mainly used in the literature. Yet, Accuracy does not work very well when data is imbalanced, which is the case for this dataset (see Section 4.3.1 about the data). Grandini, Bagli, and Visani (2020) recommend using Balanced Accuracy Score (BA) instead, it is defined as the average of the Accuracy for each class (Brodersen, Ong, Stephan, & Buhmann, 2010). Besides BA Accuracy will still be used to be able to compare the results with other research.

4.7 *Baseline*

To establish a baseline the Dummy Classifier was used, also called the ZeroR algorithm (Muhamedyev et al., 2015). This classifier follows simple rules without using the dependent variables (Pedregosa et al., 2011). In this instance, the Dummy Classifier was configured to always predict the majority class. Resulting in a Balanced Accuracy score of 0.33 and an Accuracy score of 0.63.

4.8 *Models*

The 'No Free Lunch' (NFL) theorem developed by Wolpert and Macready (1997) states that algorithms perform on average the same across all problems. Meaning that there is no single best algorithm for all problems. For this reason, there is chosen to test four different kinds of models to analyse which model performs best.

Support Vector Machine (SVM)
During training, a VSM constructs a hyperplane to do classification. And by using the so-called kernel-trick SVMs can also do non-linear classification (Noble, 2006). The reason to include this model is that it is often used in the literature and in the study of Osmani et al. (2015) it was the best performing model. The kernel used for this study was RBF, the same as used in the research of Osmani et al. (2015).

k-Nearest Neighbors (KNN)
k-NN is a relatively simple non-parametric supervised learning algorithm, that tries to predict the correct class for a new instance by calculating the distance to the training data (Zhang, 2016). Because of these characteristics, it is very different from the other models and therefore interesting to explore how the model performs

Random Forest (RF)
The Random Forest algorithm is constructed by training multiple decision trees and then combining them, either by averaging or a majority vote (Biau & Scornet, 2016). In the study of Bogomolov et al. (2014) a Random Forest was the best performing model therefore it was also used in this study.

Gradient Boosting Machine (GBM)
A GBM is similar to Random Forest, it trains multiple weaker models (usually decision trees), this is however done sequentially, by training the next model on the error of the previous (Natekin & Knoll, 2013). A GBM is used in several similar studies before, it was however not the best performing model. But since this model typically works well with imbalanced data, what is the case for this study, it was still used in this study (Noble, 2006).

4.9 *Hyperparameter Tuning*

Hyperparameter (HP) tuning is finding the optimal hyperparameters for a model, HP's are arguments for a model which are set before training. Two of the most common HP tuning methods are Grid Search and Random Search, Grid Search uses predefined parameters to test all possible combinations of HP in order to find the best combination. With a lot of HP's to tune and/or a long training time, this approach can take a considerable amount of time. The alternative is Random Search, this method searches for random combinations. This is done using a predetermined maximum number of combinations to test (Hutter, Kotthoff, & Vanschoren, 2019).

Grid search and Random Search are both capable of achieving the same results, however, Random Search generally does so slightly more effectively, in that it is able to find good HP's quicker (Bergstra & Bengio, 2012). For this reason, this study used Random Search, to be specific RandomizedSearchCV was used. It was configured to use a 5-fold cross-validation and test for 100 possible combinations.

However, to find the best HP's for the synthetically oversampled dataset, a slightly different approach was needed, because CV on the oversampled dataset means that the test part in the CV will be oversampled as well. This may lead to overfitting on the oversampled dataset and can cause poor results on the final test set. To mitigate this, the imbalanced-learn pipeline in combination with the CV pipeline was used to oversample only the train portion of the CV. Inspiration for this approach was obtained from Martin (2019).

In the literature review, it became clear that several studies have used an RF model, unfortunately it is unclear which HP's are tuned in these studies. Therefore, the study of Probst, Wright, and Boulesteix (2019) was used to select suitable HP's for optimisation, in this study the most important HP's of an RF were described. Table 9 shows the HP's which have been selected for optimisation and shows the range in which the best HP's were searched for.

Hyperparameters	**Hyperparameter range**
n_estimators	[0, 200, 400, 600, 800, 1000, 1200, 1400, 1600, 1800, 2000]
min_samples_split	[2, 5, 10, 15, 20]
min_samples_leaf	[1, 2, 4]
max_features	[auto, sqrt, log2]
max_depth	[10, 20, 30, 40, 50, 60, 70, 80, 90, 100, 110]
criterion	[gini, entropy]
bootstrap	[True, False]

Table 9: HP search range

4.10 *Software*

The majority of the study (data cleaning, feature engineering and model Training & Evolution) is done in Python (version 3.9.7) using a locally hosted JupyterLab. Table 10 shows which python packages were used.

Package	Version	Source
Pandas	1.4.1	(McKinney, 2010)
Numpy	1.22.2	(Harris et al., 2020)
Scikit-learn	1.0.2	(Pedregosa et al., 2011)
Matplotlib	3.5.1	(Hunter, 2007)
Seaborn	0.11.2	(Waskom, 2021)
Imbalanced-learn	0.9.0	(Lemaître, Nogueira, & Aridas, 2017)

Table 10: Packages to be used

4.11 *Methodology Sub-questions*

4.11.1 *Dataset model combination*

To test which was the best performing model and best performing dataset (of the datasets described in Section 4.4.3) each possible combination was tested. This was done by creating a pipeline where first a train test split (30% for test set) was made and then the test set was stored for later. After which each model on each dataset was trained and tested using Sklearn's cross_validate function (5-fold CV). The fact that only the best performing dataset was used for the rest of the study means that a large proportion of the test sets remained unused, which is not optimal. Nevertheless, this was the only approach to ensure that the definitive model can be tested on unseen data.

4.11.2 *Minority Oversampling*

The problem with imbalanced datasets can be that there are insufficient data points of the minority class for the model to effectively find the decision boundary (Provost, 2017). The most straightforward approach to solving this problem is duplicating instances of the minority class. This balances the classes but does not provide any additional information to the model.

A better approach is using Synthetic Minority Oversampling Technique (SMOTE), this technique was first described by Chawla, Bowyer, Hall, and Kegelmeyer (2002). The oversampling technique works by drawing a line between the k minority class nearest neighbours (usually k is 5) and then creating a new sample on this line.

An expansion to SMOTE is Borderline-SMOTE developed by Han, Wang, and Mao (2005). The difference is that Borderline-SMOTE only oversamples the instances of the minority class that are misclassified. Because in this way only additional data is created where it is needed, namely on the borderline between the classes.

Since this study has three classes there was the option to only oversample the minority class or to oversample all classes except the majority class. Figure 7 shows both oversampling methods in relation to the not oversampled train dataset.

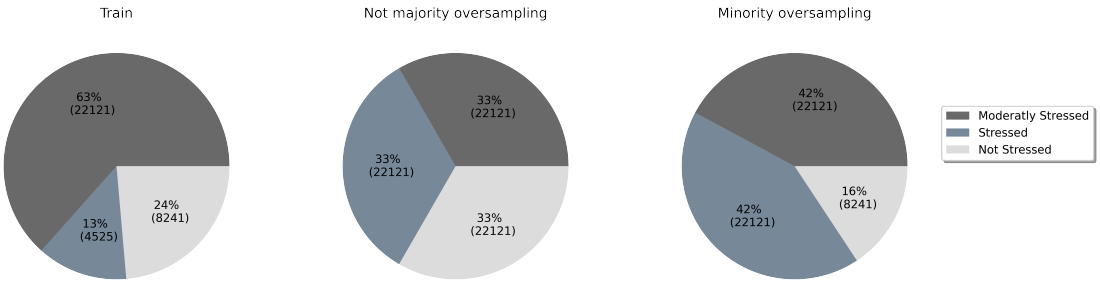

Figure 7: Comparison over different SMOTE Techniques

In order to examine which oversampling technique (SMOTE/Borderline-SMOTE) combined with which oversampling method (not majority/minority) performed best, all combinations were tested. This was done using the imbalanced-learn pipeline, where every option was tested with the best performing model/dataset combination from Section 4.11.1 using a five-fold cross-validation.

4.11.3 *Error Analysis*

To obtain a better understanding of the errors, a Confusion Matrix (CM) was used. A CM is the basis for most evaluation metrics and is consequently a useful tool to investigate errors. A CM has a size of n x n, where n stands for the number of different classes (in this study n will be three) (Visa, Ramsay, Ralescu, & Van Der Knaap, 2011).

5 RESULTS

5.1 *Dataset model combination*

In order to investigate which dataset and which model performed best, all possible combinations were investigated and the results expressed as Balanced Accuracy scores are shown in Table 11. These results are for the models without HP tuning and on the train set with the use of CV, see Section 4.2 for the research overview.

	GBM	**KNN**	**Random_forest**	**SVM**
dataset_1	0.355	0.341	0.380	0.334
dataset_2	0.360	0.342	0.386	0.336
dataset_3	0.383	0.346	0.410	0.338
dataset_4	0.354	0.344	0.384	0.340
dataset_5	0.359	0.338	0.386	0.333
dataset_6	0.358	0.339	0.378	0.333

Table 11: Balanced Accuracy scores for model dataset combination on train set

The main evolution criteria in this study was BA, the Accuracy scores however also show some interesting results, see Table 12. Here the GBM, RF and SVM scores are above the baseline of 0.63. What is striking is that dataset_3 scores the lowest for all the models. The RF is still the model with the best results, although the best score was now in combination with dataset_2. It should be noted though, that the difference with the other datasets was too small to say with certainty which of the datasets was optimal.

	GBM	**KNN**	**Random_forest**	**SVM**
dataset_1	0.706	0.638	0.713	0.700
dataset_2	0.714	0.645	0.720	0.706
dataset_3	0.656	0.559	0.667	0.634
dataset_4	0.706	0.640	0.712	0.700
dataset_5	0.708	0.636	0.716	0.700
dataset_6	0.709	0.638	0.716	0.700

Table 12: Accuracy scores for model dataset combination on train set

It can be concluded that the RF was the best performing model overall. And since the main evaluation metric was BA this means that dataset_3 was the optimal dataset. By contrast, dataset_3 had the lowest Accuracy scores. Since Accuracy is mainly used in the literature, it is interesting to com-

pare this study with the literature. Therefore, the different oversampling techniques in the next section have also been tested on dataset_2.

5.2 *SMOTE*

To investigate if, and if so which, synthetic oversampling technique worked best, different techniques as described in Section 4.11.2 have been tested. This was done with the RF model because it was the best performing model on dataset_2 and dataset_3. The results are on the train set with the use of CV (see Section 4.11.2 for further explanation). The results are shown in Table 13, and it is evident that oversampling did not increase Accuracy for either datasets. As for the Balanced Accuracy scores the best results were achieved with oversampling all classes except the majority, with the best results achieved with basis SMOTE (not majority) in combination with dataset_3. However, it should be noted that the results on the oversample dataset_2 are only slightly lower.

Oversampling technique	Dataset_2		Dataset_3	
	BA	Accuracy	BA	Accuracy
without SMOTE	0.386	0.720	0.410	0.667
SMOTE (not majority)	0.437	0.614	0.444	0.651
SMOTE (minority)	0.399	0.631	0.401	0.649
BorderlineSMOTE (not majority)	0.434	0.613	0.440	0.652
BorderlineSMOTE (minority)	0.400	0.630	0.398	0.650

Table 13: Score for different SMOTE techniques on train set

5.3 *HP Tuning & Final Results*

In Section 5.1 it became clear that RF was the best performing model, and that the highest BA score was on datset_3 and the highest Accuracy on dataset_2. Therefore, HP tuning is only done for the RF model in combination with both datasets. Furthermore, the results in Section 5.2 showed that SMOTE only increased the BA and not the Accuracy. Hence, to investigate whether SMOTE still scores higher after HP tuning, this combination was also tested. The reason that only these combinations were chosen is that these were the best performing combinations and due to time limitations it was not feasible to investigate more combinations.

Table 14 shows the score of the best performing HP's for each combination, the HP's used for these scores are in Appendix C.

Dataset	BA	Accuracy
dataset_3 (optimised for BA)	0.419	0.671
dataset_3 with SMOTE (optimised for BA)	0.442	0.662
dataset_2 (optimised for Accuracy)	0.391	0.724

Table 14: Results for RandomizedSearchCV on train set

After the HP tuning on the train set with the use of CV, the models were tested on the test set that was not used until now, the results are presented in Table 15. The results show that each tested model scores above the baseline. And it shows that dataset_3, with an RF and oversampling the data with SMOTE yields the best result regarding the main evaluation metric BA. And the best Accuracy score is 0.73 on dataset_2 without oversampling but also with an RF.

Dataset and Scoring Metric	Balanced Accuracy	Accuracy
Baseline	0.333	0.630
dataset_3 (optimised for BA)	0.426	0.682
dataset_3 with SMOTE (optimised for BA)	0.466	0.650
dataset_2 (optimised for Accuracy)	0.390	0.734

Table 15: Model results on test set

To better understand the model, RF with SMOTE in combination with dataset_3, the feature importance for the 10 most important features are shown in Figure 8 (the rest of the features can be found in Appendix D). The importance of a feature was calculated as the (normalised) total decrease of the criterion (in this case Gini) for that feature. The feature importance was not needed to answer a research question, it however provided an insight into the inner workings of the model. The feature importance did not imply a high correlation with the stress levels of the participants, but it did give an insight into which features were relevant for the model to predict the stress levels.

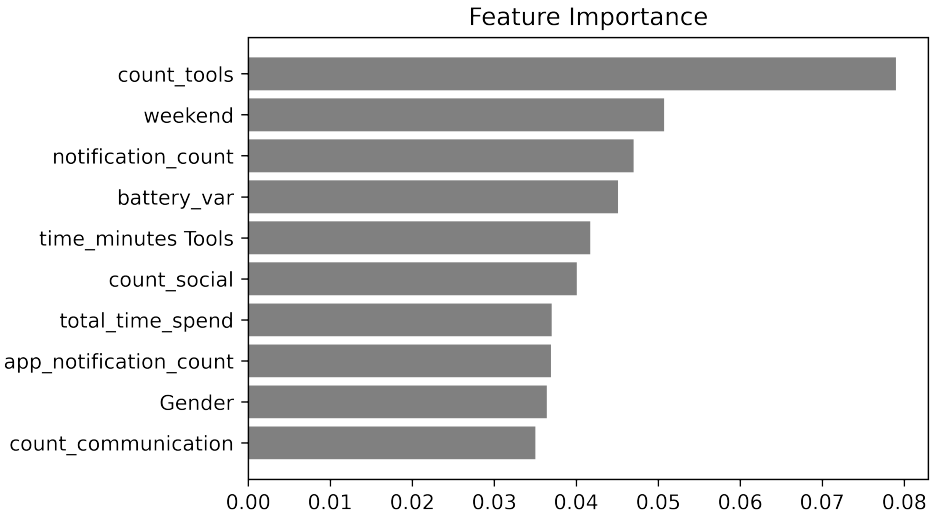

Figure 8: Feature Importance top 10

5.4 *Error Analysis*

For the error analysis, the overall best performing dataset model combination was used, thus dataset_3 with SMOTE oversampling and the RF model (with tuned HP optimised for BA). Figure 9 shows the Confusion Matrix, which illustrates the predictions on the test set. It shows that class 1 (moderately stressed) had the most mispredictions. For instance, 1561 cases which are predicted as class 1 should be predicted as class 2. The same applies to class 0, which has been wrongly predicted 2417 times as class 1. Concluding, the model is biased toward class 1. However, there are not many big errors, as for instance class 2 is not often wrong classified as class 0.

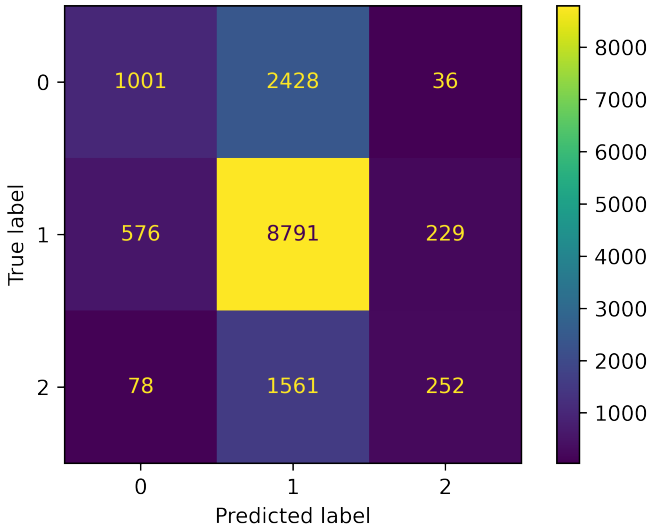

Figure 9: Confusion Matrix

Additional evaluation metrics were calculated to gain more insight into the predictive ability of the model across the different classes. Table 16 shows the Precision and Recall per class, it is noticeable that the scores for the stressed class are relatively low.

True label	Recall	Precision	F1-score
0 (not stressed)	0.36	0.38	0.37
1 (moderately stressed)	0.78	0.76	0.77
2 (stressed)	0.25	0.30	0.27

Table 16: Additional classification scores

6 DISCUSSION

Best feature representation and model combination

The goal of this study was to investigate to what extent it is possible to predict stress levels from smartphone usage data. The first step to achieve this was to explore the best approach for feature representation, see Section 5.1. The dataset with the overall highest Balanced Accuracy (BA) was the feature representation with count and time per app category. However, the feature representation with the highest Accuracy score was the one with only time per app category. Remarkably, the feature representation with the highest BA had the lowest Accuracy score and vice versa. A reasonable explanation for this difference is that the feature representation with time and count offers additional information that allows the model to make better predictions for all classes. Though, this also resulted in the total number of good predictions were slightly lower, hence the lower Accuracy score. This difference between the two scores illustrated the fact that the use of Accuracy alone can give a distorted impression of the overall model performance.

The results also showed that dividing the apps into more categories yielded better results. However, it is not certain whether this was due to the fact that it was better to use more categories or that it was due to how the categories are combined, which was done in this study.

The second part investigated the best performing model for predicting stress levels. The results showed that for both BA and for Accuracy a Random Forest (RF) model yielded the best results. These results were in line with the research from Bogomolov et al. (2014) as they also reported that an RF was the best model. The research of Osmani et al. (2015) that is very similar to this study had on the contrary concluded that an SVM was the best performing model. However, the results of this study were a lot higher than the results of Osmani et al. (2015), which suggests that an RF is the better model.

Synthetic oversampling of minority class

The third part of this study aimed to determine whether, and if so which method of, oversampling produced the best results. The results showed (see Section 5.2) that SMOTE increased the BA compared to not oversampling. However, no oversampling technique has been used in similar research yet, this is probably because it does not increase Accuracy. The reason why it does not increase the Accuracy but does increase the BA is that SMOTE causes the minority class to be predicted more accurately, which results in a higher BA. However, this did not lead to an increase in Accuracy.

Methodology
After identifying the best feature representation and model, this study continued with this combination. Continuing with only the best performing feature representation for the rest of the study had the consequence that possible improvements of other feature representations, for example in SMOTE, were not investigated. This would make it possible that if SMOTE and HP tuning were applied to all datasets, another dataset and model combination may have performed best. However, it was not feasible to do so in this study for the very reason that there would have been too many combinations to analyse. Moreover, if another dataset would score better at all, the difference would still be very limited. Because, by choosing the best dataset model combination, the differences with the other options were minimal (BA score of 0.410 versus the second best 0.386).

Results compared to the literature
The Accuracy achieved in this study is considerably higher (see Section 5.3) than the research by Osmani et al. (2015) (0.54%), which was used as a guide for this study because it had the closest resemblance. However, the findings were consistent with those of Bogomolov et al. (2014). It should be noted that Bogomolov et al. (2014) used additional features about personal traits and about the contacts of participants. The results obtained in this study are approaching those of Umematsu et al. (2019), where they used additional physiological parameters. It can be concluded that the more unintrusive approach used in this study is close to the current standard. With this, a step has been taken in the possibility of predicting stress levels for a larger group of people, supporting the societal relevance of this study.

Error analysis
The final part of this study was the error analysis, which showed that despite the relatively high Accuracy score, the model scores low in correctly predicting whether someone is stressed (see Section 5.4). However, the predictive performance of predicting whether a participant is moderately stressed is relatively higher. Which means a bias towards the moderately stressed class This study deviated from the literature because it attempted to predict stress in three classes rather than using a binary outcome variable. Consequently, the model appeared to be less able to predict high stress levels, which was the goal with the use of a different outcome variable. Whether a binary outcome variable would yield better results regarding the error analysis cannot be stated with certainty, because no error analysis has been carried out in the relevant literature. Therefore, this is something that is of interest for future research.

6.1 *Limitations and future research*

Self-reporting bias
This study used data containing a questionnaire that asked participants about their perceived stress levels and thereby measuring the participant's own perception of their stress levels. According to Althubaiti (2016), when using self-reported data of mental characteristics there is a risk of bias in the data. According to Althubaiti (2016) one type of bias that can occur with self-reported stress levels is social desirability bias. With this type of bias, participants tend to give socially desirable answers. An example of how this type of bias may occur in the data is that during exams students could think they are stressed because that is expected.

Another form of bias that is relevant to this study is measurement error bias. With this type of bias, the method of measurement is not uniform. In this study, a 7-point Likert scale was used, yet each point on the scale had a different meaning for individual participants. Where one participant will only indicate a 7 in extremely stressful situations, the other person may do so already in a demanding period. Consequently, the points on the Likert scale do not have the same meaning for every participant, which can result in measurement error bias. The extent to which these forms of bias affect this study was beyond the scope, but it is not a fact that can be overlooked when interpreting the results.

User-specific model
The research by Osmani et al. (2015) demonstrated that a user-specific model performed better than a generic model. This study did not examine a user-specific model, but this is certainly something for follow-up research to investigate further. Because this has the potential for better results. Furthermore, it would be interesting to investigate whether user-specific models are better able to recognise stress. Thus, in addition to Accuracy, the predictive ability of stress should be investigated.

A more heterogeneous group of participants
This study used a dataset that only contained data from Dutch university students. Because this is such a homogeneous group of participants, it is unclear whether the results found in this study will also apply to a broader group of people. According to Han et al. (2005) there can arise serious complications if results found for students are generalised to a broader group of people. Therefore, in future research, it may be beneficial to use data from a more heterogeneous group of participants.

7 CONCLUSION

This study aimed to discover to what extent it was possible to predict stress levels using smartphone usage logs. This was done by exploring answers to the sub-questions stated below to eventually answer the main question.

RQ1 What is the best method of the tested methods in representing the smartphone usage logs in order to realise a high Balanced Accuracy in predicting stress levels?
The results showed that the best approach to transforming the smartphone usage logs into a dataset is by calculating the time and count per app category. Furthermore, the results showed that using more app categories to categorise the apps than was done in previous research resulted in better results.

RQ2 Which of the following models most accurately predicts stress levels; Support Vector Machine, k-Nearest Neighbors, Random Forest, or Gradient Boosting Machine?
Based on the literature review, the above models were selected to explore which could best predict stress levels. By testing the different models on the feature representations of sub-question 1, the results showed that a Random Forest scored the highest on both BA and Accuracy..

RQ3 To what extent does balancing the data using the SMOTE method improve the Balanced Accuracy?
This study demonstrated that by applying the Synthetic Minority Oversampling Technique (SMOTE) technique to oversample the minority classes, a higher BA was achieved.

RQ4 How are the errors distributed among the predicted classes?
The error analysis revealed that most of the errors occurred in the misprediction of the moderately stressed class, meaning the model was biased towards that class.

The research carried out to answer the sub-questions makes it possible to answer the main question of this study.

Main RQ To what extent is it possible to predict perceived stress levels among Dutch students, based on their smartphone usage logs?
The results presented in this study showed that smartphone usage logs can be used to predict stress levels with a BA of 0.466 and an Accuracy of 0.734.

Where research commonly uses wearable sensors and/or data that subjects must report themselves, such as activity level and/or social interactions, this study only used passively logged data. This means that it is a cost-effective and unintrusive approach. Moreover, as the results are not substantially below the current standard, this study has the potential of contributing to making this technique less intrusive and therefore more accessible.

Acknowledgements
First and foremost, I would like to thank my supervisor Drew Hendrickson for providing the data, information and the help I needed to complete this thesis. In addition, I would also like to express my thanks in advance to the second reviewer.

And last but not least, my gratitude to my girlfriend, family and friends for their support over the last year.

REFERENCES

Aalbers, G., vanden Abeele, M. M. P., Hendrickson, A. T., de Marez, L., & Keijsers, L. (2021). Caught in the moment: Are there person-specific associations between momentary procrastination and passively measured smartphone use? *Mobile Media Communication*, *10*(1), 115–135. doi: 10.1177/2050157921993896

Alghamdi, M., Al-Mallah, M., Keteyian, S., Brawner, C., Ehrman, J., & Sakr, S. (2017). Predicting diabetes mellitus using SMOTE and ensemble machine learning approach: The Henry Ford ExercIse Testing (FIT) project. *PLOS ONE*, *12*(7). doi: 10.1371/journal.pone.0179805

Allison, P. (2000). Multiple Imputation for Missing Data: A Cautionary Tale. *Sage Journals*, *28*(3). doi: 10.1177/0049124100028003003

Althubaiti, A. (2016). Information bias in health research: definition, pitfalls, and adjustment methods. *Journal of Multidisciplinary Healthcare*, 211. doi: 10.2147/jmdh.s104807

Bauer, G., & Lukowicz, P. (2012). Can smartphones detect stress-related changes in the behaviour of individuals? *IEEE International Conference on Pervasive Computing and Communications Workshops*, 423–426. doi: 10.1109/PerComW.2012.6197525

Bergstra, J., & Bengio, Y. (2012). Random search for hyper-parameter optimization. *Journal of machine learning research*, *13*(2).

Biau, G., & Scornet, E. (2016). A random forest guided tour. *TEST*, *25*(2), 197–227. doi: 10.1007/s11749-016-0481-7

Bogomolov, A., Lepri, B., Ferron, M., Pianesi, F., & Pentland, A. S. (2014). Daily Stress Recognition from Mobile Phone Data, Weather Conditions and Individual Traits. *Proceedings of the 22nd ACM international conference on Multimedia*. doi: 10.1145/2647868.2654933

Brodersen, K. H., Ong, C. S., Stephan, K. E., & Buhmann, J. M. (2010). The Balanced Accuracy and Its Posterior Distribution. *2010 20th International Conference on Pattern Recognition*. doi: 10.1109/icpr.2010.764

Chawla, N. V., Bowyer, K. W., Hall, L. O., & Kegelmeyer, W. P. (2002). SMOTE: Synthetic Minority Over-sampling Technique. *Journal of Artificial Intelligence Research*, *16*, 321–357. doi: 10.1613/jair.953

Cohen, S., Janicki-Deverts, D., & Miller, G. E. (2007). Psychological Stress and Disease. *JAMA*, *298*(14), 1685. doi: 10.1001/jama.298.14.1685

Dalla, C., Antoniou, K., Drossopoulou, G., Xagoraris, M., Kokras, N., Sfikakis, A., & Papadopoulou-Daifoti, Z. (2005). Chronic mild stress impact: Are females more vulnerable? *Neuroscience*, *135*(3), 703–714. doi: 10.1016/j.neuroscience.2005.06.068

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*, 103151. doi: 10.1016/j.jbi.2019.103151

GALLUP. (2021). *Gallup Global Emotions 2021 report* (Tech. Rep.). Retrieved from https://www.gallup.com/analytics/349280/gallup-global-emotions-report.aspx

Giannakakis, G., Grigoriadis, D., Giannakaki, K., Simantiraki, O., Roniotis, A., & Tsiknakis, M. (2022). Review on psychological stress detection using biosignals. *IEEE Transactions on Affective Computing*, *13*(1), 440-460. doi: 10.1109/TAFFC.2019.2927337

Grandini, M., Bagli, E., & Visani, G. (2020). Metrics for Multi-Class Classification: an Overview. *ArXiv*. doi: 10.48550/arXiv.2008.05756

Han, H., Wang, W.-Y., & Mao, B.-H. (2005). Borderline-SMOTE: A New Over-Sampling Method in Imbalanced Data Sets Learning. *Lecture Notes in Computer Science*, 878–887. doi: 10.1007/11538059\{_}91

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

Hellhammer, D., Stone, A., & Broderick, J. (2010). Measuring Stress. *Encyclopedia of Behavioral Neuroscience*, *2*, 186–191. doi: 10.1016/b978-0-08-045396-5.00188-3

Hudd, S., Dumlao, J., Erdmann-Sager, D., & Murray, D. (2000). Stress at college: Effects on health habits, health status and self-esteem. *College Student Journal*, *34*(2), 217–227.

Hunter, J. D. (2007). Matplotlib: A 2D Graphics Environment. *Computing in Science Engineering*, *9*(3), 90–95. doi: 10.1109/mcse.2007.55

Hutter, F., Kotthoff, L., & Vanschoren, J. (2019). *Automated Machine Learning*. doi: 10.1007/978-3-030-05318-5

Jaques, N., Taylor, S., Nosakhare, E., Sano, A., & Picard, R. (2020). Personalized Multitask Learning for Predicting Tomorrow's Mood, Stress, and Health. *IEEE Transactions on Affective Computing*, *11*(2), 200–213. doi: 10.1109/TAFFC.2017.2784832

Kohavi, R. (2001). A Study of Cross-Validation and Bootstrap for Accuracy Estimation and Model Selection. *International Joint Conference on Artifficial Intelligence (IJCAI)*.

Kong, Q., Siauw, T., & Bayen, A. (2020). *Python Programming and Numerical Methods* (1st ed.). Retrieved from https://pythonnumericalmethods.berkeley.edu/notebooks/chapter17.03-Cubic-Spline-Interpolation.html

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

from http://jmlr.org/papers/v18/16-365.html

Martin, D. (2019, 05). *How to do cross-validation when upsampling data.* Retrieved from https://kiwidamien.github.io/how-to-do-cross-validation-when-upsampling-data.html

McKinney, W. (2010). Data Structures for Statistical Computing in Python. *Proceedings of the Python in Science Conference*, *445*, 51–56. doi: 10.25080/majora-92bf1922-00a

Ministerie van Onderwijs, Cultuur en Wetenschap. (2021, 11). *Mentale gezondheid studenten onder druk.* Retrieved from https://www.rijksoverheid.nl/actueel/nieuws/2021/11/11/mentale-gezondheid-studenten-onder-druk

Muhamedyev, R., Yakunin, K., Iskakov, S., Sainova, S., Abdilmanova, A., & Kuchin, Y. (2015). Comparative analysis of classification algorithms. *2015 9th International Conference on Application of Information and Communication Technologies (AICT)*, *2012*(Volume 5), 239–243. doi: 10.1109/icaict.2015.7338525

Natekin, A., & Knoll, A. (2013). Gradient boosting machines, a tutorial. *Frontiers in Neurorobotics*, *7*. doi: 10.3389/fnbot.2013.00021

Nguyen, B. P., Pham, H. N., Tran, H., Nghiem, N., Nguyen, Q. H., Do, T. T., ... Simpson, C. R. (2019). Predicting the onset of type 2 diabetes using wide and deep learning with electronic health records. *Computer Methods and Programs in Biomedicine*, *182*, 105055. doi: 10.1016/j.cmpb.2019.105055

Nick, G. (2022, 04). *55+ Jaw Dropping App Usage Statistics in 2022 [Infographic].* Retrieved from https://techjury.net/blog/app-usage-statistics/

Noble, W. S. (2006). What is a support vector machine? *Nature Biotechnology*, *24*(12), 1565–1567. doi: 10.1038/nbt1206-1565

Osmani, V., Ferdous, R., & Mayora, O. (2015). Smartphone app usage as a predictor of perceived stress levels at workplace. *Proceedings of the 9th International Conference on Pervasive Computing Technologies for Healthcare*. doi: 10.4108/icst.pervasivehealth.2015.260192

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

Probst, P., Wright, M. N., & Boulesteix, A. (2019). Hyperparameters and tuning strategies for random forest. *WIREs Data Mining and Knowledge Discovery*, *9*(3). doi: 10.1002/widm.1301

Provost, F. (2017). Machine Learning from Imbalanced Data Sets 101. *Proceedings of the AAAI'2000 workshop on imbalanced data sets*, *68*(2000), 1–3.

Ren, Z., Xin, Y., Ge, J., Zhao, Z., Liu, D., Ho, R. C. M., & Ho, C. S. H. (2021).

Psychological Impact of COVID-19 on College Students After School Reopening: A Cross-Sectional Study Based on Machine Learning. *Frontiers in Psychology*, *12*. doi: 10.3389/fpsyg.2021.641806

Schönfeld, P., Brailovskaia, J., Bieda, A., Zhang, X. C., & Margraf, J. (2016). The effects of daily stress on positive and negative mental health: Mediation through self-efficacy. *International Journal of Clinical and Health Psychology*, *16*(1), 1–10. doi: 10.1016/j.ijchp.2015.08.005

Shuja, M., Mittal, S., & Zaman, M. (2020). Effective Prediction of Type II Diabetes Mellitus Using Data Mining Classifiers and SMOTE. *Advances in Computing and Intelligent Systems*, 195–221. Retrieved from https://link.springer.com/chapter/10.1007/978-981-15-0222-4_17

Stütz, T., Kowar, T., Kager, M., Tiefengrabner, M., Stuppner, M., Blechert, J., … Ginzinger, S. (2015). Smartphone Based Stress Prediction. *Lecture Notes in Computer Science*, 240–251. doi: 10.1007/978-3-319-20267-9\{_}20

Tsoumakas, G., & Katakis, I. (2007). Multi-Label Classification: An Overview. *International Journal of Data Warehousing and Mining*, *3*(3), 1–13. doi: 10.4018/jdwm.2007070101

Umematsu, T., Sano, A., Taylor, S., & Picard, R. W. (2019). Improving Students' Daily Life Stress Forecasting using LSTM Neural Networks. *2019 IEEE EMBS International Conference on Biomedical Health Informatics (BHI)*. doi: 10.1109/bhi.2019.8834624

van Buuren, S. (2021). *Flexible Imputation of Missing Data, Second Edition* (2nd edition ed.). Retrieved from https://stefvanbuuren.name/fimd/

Varvogli, L., & Darviri, C. (2011). Stress Management Techniques: evidence-based procedures that reduce stress and promote health. *HEALTH SCIENCE JOURNAL*, *5*(2). Retrieved from https://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.851.7680&rep=rep1&type=pdf

Visa, S., Ramsay, B., Ralescu, A., & Van Der Knaap, E. (2011). Confusion Matrix-based Feature Selection. *MAICS*, *710*, 120–127.

Vrijkotte, T., van Doornen, L., & de Geus, E. (2000). Effects of Work Stress on Ambulatory Blood Pressure, Heart Rate, and Heart Rate Variability. *Hypertension*, *35*(4), 880–886. doi: 10.1161/01.hyp.35.4.880

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

Wolpert, D., & Macready, W. (1997). No free lunch theorems for optimization. *IEEE Transactions on Evolutionary Computation*, *1*(1), 67–82. doi: 10.1109/4235.585893

Zhang, Z. (2016). Introduction to machine learning: k-nearest neighbors. *Annals of Translational Medicine*, *4*(11), 218–218. doi: 10.21037/atm

.2016.03.37

Appendices

A APPENDIX A

Category	App Count
Education	241090
Music & Audio	154906
Tools	143988
Business	143771
Entertainment	138276
Lifestyle	118331
Books & Reference	116728
Personalization	89210
Health & Fitness	83510
Productivity	79698
Shopping	75256
Food & Drink	73927
Travel & Local	67288
Finance	65466
Arcade	53792
Puzzle	51168
Casual	50813
Communication	48167
Sports	47483
Social	44734
News & Magazines	42807
Photography	35552
Medical	32065
Action	27555
Maps & Navigation	26722
Simulation	23282
Adventure	23203
Educational	21308
Art & Design	18539
Auto & Vehicles	18280
House & Home	14369
Video Players & Editors	14015
Events	12841
Trivia	11795
Beauty	11772
Board	10588

Racing	10362
Role Playing	10034
Word	8630
Strategy	8526
Card	8179
Weather	7246
Dating	6524
Libraries & Demo	5198
Casino	5076
Music	4202
Parenting	3810
Comics	2862

B APPENDIX B

Category	App count
Productivity & Business	340197
Games & Other	305865
Social	252797
Education	241090
Music & Audio & Video	208675
Tools	143988
Entertainment	138276
Health & Fitness & Medical	127347
Lifestyle	118331
Personalization	89210
Shopping	75256
Finance	65466
Other	60196
Communication	48167
News & Magazines	42807
Maps & Navigation	26722
Educational	21308
Weather	7246

C APPENDIX C

Hyperparameters	**Best Hyperparameter**		
	dataset_3 (optimised for BA)	dataset_3 with SMOTE (optimised for BA)	dataset_2 (optimised for Accuracy)
n_estimators	1400	2000	400
min_samples_split	5	15	5
min_samples_leaf	1	1	2
max_features	sqrt	sqrt	auto
max_depth	110	50	80
criterion	gini	gini	gini
bootstrap	False	False	False

D APPENDIX D

Feature importance top 30

Feature	Importance
count_tools	0.05697
weekend	0.05071
notification_count	0.04700
battery_var	0.04508
time_minutes Tools	0.04170
count_social	0.04008
total_time_spend	0.03700
app_notification_count	0.03692
Gender	0.03640
count_communication	0.03502
time_minutes Communication	0.03459
time_minutes Social	0.03414
count_music	0.02872
part_day	0.02389
time_minutes Video Players & Editors	0.02351
count_productivity	0.02157
time_minutes Music & Audio	0.01935
count_photography	0.01715
count_finance	0.01671
time_minutes Health & Fitness	0.01518
time_minutes Productivity	0.01471
count_education	0.01435
count_lifestyle	0.01381
count_entertainment	0.01322
time_minutes Travel & Local	0.01233
time_minutes Photography	0.01226
time_minutes Finance	0.01120
time_minutes Entertainment	0.01102
time_minutes Lifestyle	0.01094