Data Mining Techniques - Assignment 1

Dominic Istha (2600140), Titus Pellegrom (2594989), and Jaimie Rutgers (2598649) - DMT-2021 33

Vrije Universiteit Amsterdam

1 Introduction/problem statement

On average, students use their smartphone five hours per day according to [1]. Meanwhile, studies also show worrying results on depression amongst students. Findings of [3] state that there is accumulating evidence that depression is a significant health concern in universities, as it affects nearly a third of students. Therefore, we have investigated the relationship between the smartphone usage of a student and their mood. This research is done based on [4]. In this research we worked with a similar data set, although the feature set was not identical. Furthermore, we have extend the reserach of [4] in two ways. First, we also tested a model that is able to cope with the temporal nature of the data, namely an ARIMA model. Second, we have used more advanced machine learning models, such as a Regression Tree and Support Vector Regression. The motivation behind these models is based on the results as shown in [5].

2 Data

2.1 Source

The data that has been used in this research is based on the data collected by [4]. In [4] they collected data from a small group ($n = 27$) of Dutch students that self-monitored their mood for 6 weeks (42 days). During the same period, a custom smartphone app collected data of their social and physical activity and general phone usage. The self-monitoring was done by student's rating their mood five times per day (approximately at 09:00, 12:00, 15:00, 18:00, and 21:00).

2.2 Exploration

Table 1 shows the features that are present in the data. Note that for all duration variables, the range is any number in $\mathbb{R}^+$. The features in italic text were derived from the data and are described in section 2.3. Like [4] our data contains information of 27 students. However, the number of days is not equal to 42 and even differs per student. Figure 2 shows the number of reported days per student. Furthermore, most students have data from mid March 2014 until the beginning of May but with different start dates (Figure 1. We noted a decline in the average mood between March 21 and 28. Since the student were studying on the Vrije Universiteit in Amsterdam [4], this turned out to be an exam week.

Table 1: Data set feature description

Feature	Description	Range
mood	The mood scored by the user	$\in \{1, ..., 10\}$
circumplex.arousal	The arousal scored by the user	$\in \{-2, ..., 2\}$
circumplex.valence	The valence scored by the user	$\in \{-2, ..., 2\}$
affect	The transformed combination of circumplex.arousal and circumplex.valence	$\in [-12, 12]$
activity	Activity score of the user	$\in [0, 1]$
screen	Duration of screen activity	
call	Call made	$\in \{0, 1\}$
sms	SMS sent	$\in \{0, 1\}$
appCat.builtin	Duration of usage of builtin apps	
appCat.communication	Duration of usage of communication apps	
appCat.entertainment	Duration of usage of entertainment apps	
appCat.finance	Duration of usage of finance apps	
appCat.game	Duration of usage of game apps	
appCat.office	Duration of usage of office apps	
appCat.other	Duration of usage of other apps	
appCat.social	Duration of usage of social apps	
appCat.travel	Duration of usage of travel apps	
appCat.unknown	Duration of usage of unknown apps	
appCat.utilities	Duration of usage of utilities apps	
appCat.weather	Duration of usage of weather apps	
weekday	Day of the week (dummy per day)	$\in \{0, 1\}$
sleep	Duration of sleep	

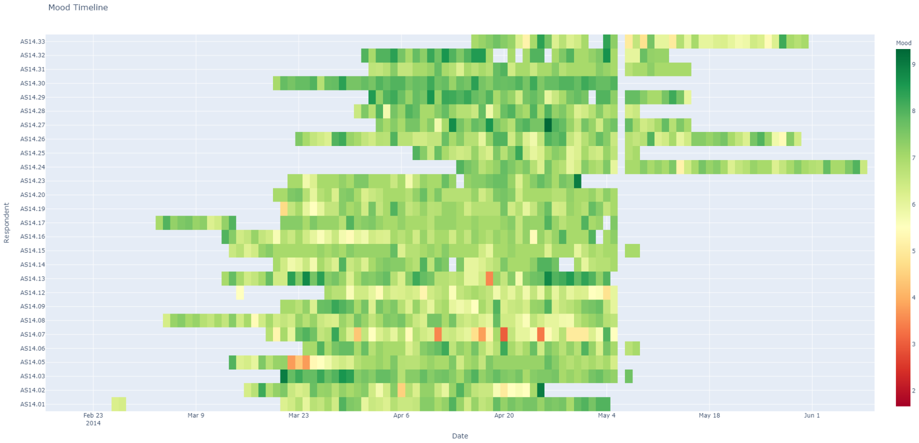

Fig. 1: Logged moods per respondent id

2.3 Feature engineering

Besides the features that were already present in the data, we added additional features based on literature. The variables *circumplex.valence* and *cir-*

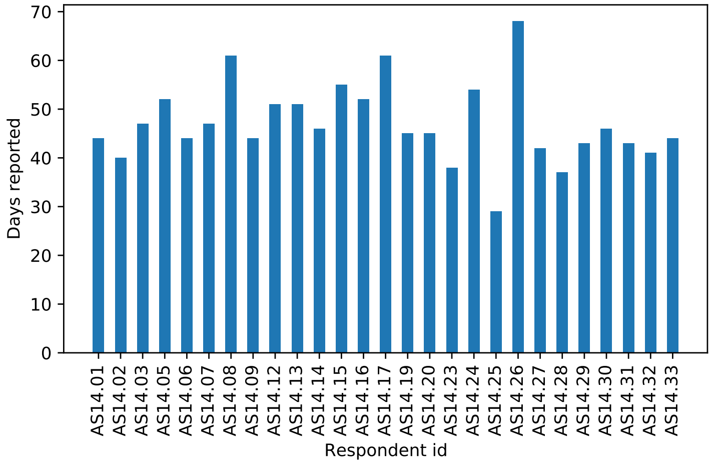

Fig. 2: Number of reported days per respondent id

cumplex.arousal form the basis of the Circumplex Model of Affect [6]. Since the Circumplex Model of Affect provides additional information about the student's mood, we created an extra variable *affect*. This variable was created as follows:

$$affect_{t,i} = circumplex.arousal_{t,i} \times 5 + circumplex.valence_{t,i} \tag{1}$$

Here, t denotes the time of the observation and i denotes the student. We multiply by 5, as there are 5 possible values for the *circumplex.arousal* variable (-2, -1, 0, 1, 2). We also include sleep information. [7] has shown a significant effect of sleep duration, bedtimes and wake up times on PHQ-8 depression scores [11]. Also, [8] have shown that sleep differs the most between people with a good and bad mental health. To detect the amount of sleep we, use a rather robust approach. We use the idle duration between two observations of the variable *screen* to determine this. Since this also occurs during the day, the second observation (the wake-up time) must be between 04:00 and 12:00. Finally, the duration sleep must be at least three hours to count as sleep. To account for potential failures in this method, we replace any missing values by the average amount of sleep per respondent. Technically, the sleep period falls on the same day as the prediction target (mood), but also precedes logging a mood. Therefore, we included the sleep duration at the night before the prediction day in the exogenous variables. The third variable we will add is a day-of-the-week variable, where we follow [9]. This variable is added as a dummy variable per weekday to the data set.

2.4 Preprocessing

Cleaning and Format The data is preprocessed to be used effectively with different models. First, we removed negative time values which occurred for the 'appCat.Builtin' and 'appCat.Entertainment' category. Secondly, we transformed the data to a wide-format. Note that this creates a sparse data set since only a few parameters are logged per timestamp. Thirdly, we added the additional features *affect*, *sleep* and *weekday*, as described in section 2.3.

Aggregation To reduce sparsity of the input data, we aggregated all variables per user, per day. We adhered to the split-apply-groupby doctrine during all operations [14]. All variables were aggregated by taking the daily minimum, maximum, mean, median, standard deviation and number of occurrences. The call and SMS variables were exempted, since only the count per day is relevant. Besides, for screen and application time, the daily sum was included. Lastly, the sleep variable was not aggregated since we register at most one sleep duration per day.

Missing Values Some users had periods where no mood was reported, while the smartphone use was logged. Those periods occur at the start of the experiment, where the monitoring app was likely being tested. We decided to remove those periods from the data set, both for training as for the benchmark model. The daily aggregations still contained missing observations. Especially the app usage, since some app categories are not used every day. Hence, we inserted a zero value for missing app, call or SMS data. In case of gaps in reported values such as mood, valence or arousal, we used forward-filling. Lastly, we did not manage to extract sleep duration for all nights, at which point we used the average sleep for the given user. Note that the average sleep introduces information-leakage since the average contains observations from future days. We could avoid leakage by calculating the average sleep on an expanding window. We decided not to apply this method by reason of two arguments. First, the average sleep was around eight hours, which makes intuitively sense for students. Second, an expanding average increases variance at the beginning of the time series. The resulting data set is used for the temporal model and the naive benchmark.

2.5 Time window

The Regression Tree and Support Vector Regression cannot deal with the temporal nature of the data. To cope with the temporal nature of the data, the data of k successive days are aggregated and used as predictive features for the mood of a user on the next day. It is expected that the day of the week will have a significant affect on the mood of students. Therefore, initially a time-window of 7 days is used, to average any effects caused by the day of the week. To be precise, to predict the mood on the 8th day of a user, the data of the week prior is aggregated and used as predictive features. Once feature selection has been performed and an appropriate model has been developed, the effect of aggregating the data over different time-windows, where $k = 2, 3, 4, 7$, will be investigated. We choose $k = 7$ as mentioned above, the values 2, 3 and 4 are based on the approach of [5].

2.6 Feature selection

Correlation To get an idea about the relation of the features with the next-day mood and to investigate potential important features, we can look at the correlations of our aggregated data. Figure 3 shows the top 20 features according to their Pearson coefficient. Note that most features have weak relations, except the mood itself. Thus, the average mood over the past days strongly correlates with the mood on the next day. We suspect that the benchmark model will thus be fairly strong. Some correlations can be intuitively explained. For example, the standard deviation of the mood and sleep duration in the past days are negatively correlated with the next day mood. In addition, the median of entertainment app usage is negatively correlated with the mood.

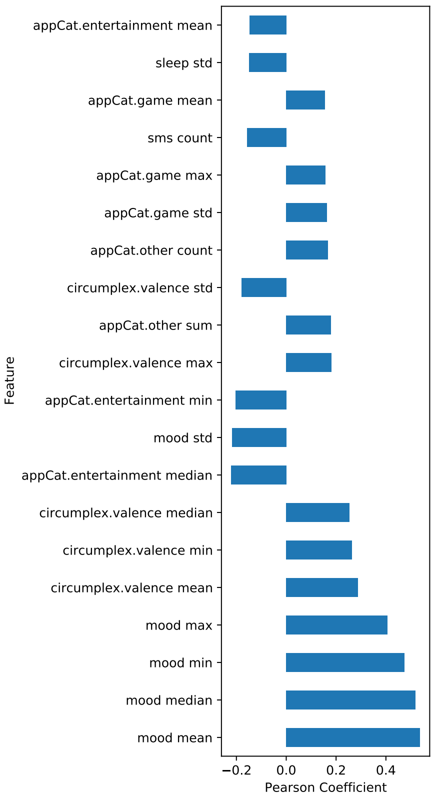

Fig. 3: Top 20 important features according to their Pearson coefficients.

Table 2: Top 20 important features according to Univariate Selection

Feature	**Score**
screen min	415.30
activity count	375.15
activity mean	297.81
activity std	202.04
appCat.builtin mean	92.81
appCat.builtin min	76.22
appCat.builtin median	70.51
circumplex.valence std	56.89
appCat.other max	52.95
screen max	51.11
appCat.office sum	44.30
mood max	42.67
appCat.builtin max	34.60
appCat.utilities count	34.16
appCat.builtin std	33.79
appCat.utilities std	29.28
appCat.travel mean	28.02
appCat.travel min	25.80
appCat.finance count	25.74

Univariate Selection The third method we will use to investigate feature importance is a univariate selection method which is based on statistical tests. For our data we will make use of the `SelectKBest` function from the scikit library. This function selects the k best features according to the F regression score. The function first computes the correlation between the feature and the target variable and converts this to an F score and then to a p-value.

Extra Trees Next, we will use a more sophisticated method to detect important features. To do so, we model an extremely randomized tree (Extra Tree) regression on all of the data. We prefer an ensemble method over a decision tree as we want to have randomness in this feature selection process. This also explains why we chose an Extra Tree over a Random Forest as the Extra Tree adds even more randomization but still has optimization. Besides, Extra Trees use the whole data set in contrast to a bootstrap sample for Random Forests. The importance of the feature is determined by it's Gini importance. The results are shown in Figure 4.

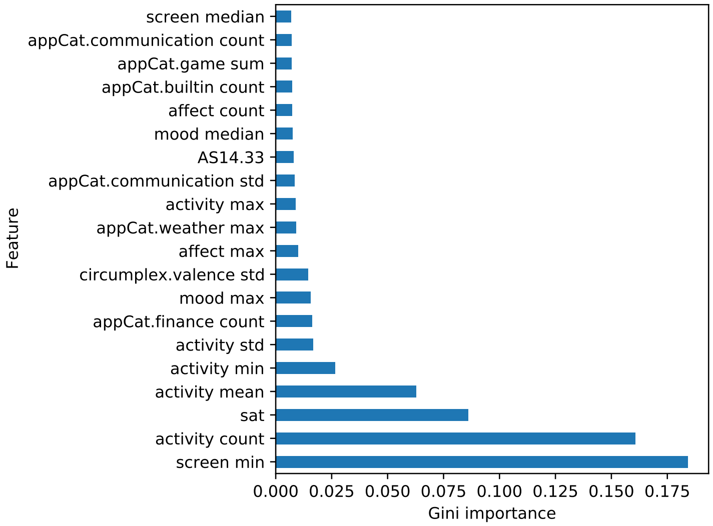

Fig. 4: Top 20 important features according to Extra Tree Regression

Final selection Based on the feature importances of the aforementioned methods we will make a final selection. We will select the following features for our models: *screen min*, *activity mean* as these are (one of) the most important features for both the Extra Tree and Univariate Selection results; *sat*, *affect max* and *sleep std* as these are all derived features; *circumplex.valence std* since this feature occurs in the results of all three methods; *sms count* and *appCat.communication count* to account for social interactions; *appCat.finance count* as we suspect that the amount of times you check your bank account affects your mood. Furthermore, we included *appCat.entertainment median, appCat.travel mean, appCat.weather max, appCat.builtin mean, appCat.game sum, appCat.office sum, appCat.other sum, appCat.utilities count* because we aimed to include every app category that was present in the feature importance results. We then selected the feature with the highest importance compared over all three methods.

3 Models

3.1 Baseline

To be able to make fair comparisons and to have valid results, we will make use of a fairly simple baseline. Namely, we use the respondent's mood from the previous day to predict the mood of the next day. Research has shown that not every study applies a meaningful baseline [13].

3.2 Temporal model

For the temporal model we will use an ARIMA model, which is commonly applied in time-series forecasting and rather simple to implement. The ARIMA model purely uses past observations of the mood to predict the future mood. It consists of three parts: (i) An auto-regressive (AR) part, which is a regression on lagged values of the target variable, (ii) an integrated part (I) to eliminate a trend in the data, and (iii) a moving average (MA) part to reduce the effect of potential errors in lagged values. To test the benefit of combining these three components, the AR-model and MA-model will also be tested on their own.

3.3 Non-temporal model

Regression tree The forst non-temporal model is a Regression Tree. A Regression Tree is similar to a Decision Tree, a supervised machine learning model, but outputs a regression instead of a class. We choose a regression tree, since we want to predict the average mood. While the mood as given by the user might be discrete, and thus a category, the average mood is not. The goal of the Regression Tree is to find groups in the data with similar values of variables, such that it can make clear distinctions between these groups based on the variables. To find these groups, the Regression Tree algorithm makes use of Gini importance. For a full explanation see [12].

Support Vector Regression The second non-temporal model we will use is the Support Vector Regression (SVR). The SVR model is a supervised regression algorithm like the Regression Tree. However, instead of using Gini importance to distinct groups, SVR tries to fit an N-dimensional hyper-plane to the training data. The hyper-plane is allowed to have a certain width ϵ and is supported by data points lying outside of the band (the support vectors). The hyper-plane is created using a kernel function that can be linear, but non-linear as well. We found the Radial Basis Function (RBF) kernel to be used most often on similar data sets [7], [2] and therefore decided to follow these approaches.

4 Experimental setup

4.1 Temporal model

To develop an ARIMA model, 3 parameters need to be determined: the order (p) of the auto-regressive (AR) term, the order (q) of the moving-average term and the number of differencing (d) required to make the time-series stationary. To determine these parameters, both a visual and statistical analysis will be performed. To keep the experiments conducted of a reasonable size and duration, the same parameter set will be used for all individual time-series. The parameters are determined by analyzing the 1st and 2nd order differenced time-series, as well as the corresponding (partial) auto-correlation plots. Furthermore, the Augmented-Dickey Fuller test, KPSS test and Phillips-Perron test are conducted to determine the order of differencing. The visual and statistical analysis suggest that the terms q and d could potentially best be 0 or 1. The visual analysis is inconclusive for the appropriate value of the term p. Even though the visual and statistical analysis provide some insight into appropriate parameters, the choice has been made to do a more extensive test of different parameter sets.

Initially, a total of 27 different parameter sets will be tested, where the parameters p, d and q may take the values 0, 1 and 2. To test the different parameter sets, the ARIMA model is trained on the first 70% of the data of each user and evaluated on the following 15% in the time-series. The best model selected from the 27, is trained on the first 85% of each time-series and evaluated on the last 15%, to be compared to the other methods described in this report.

4.2 Non temporal model

For the SVR and the Regression Tree we do not need to take the ordering of the data along the time dimension into account. Therefore, a standard train, validation and test split (70/15/15) can be applied. This will be applied to an aggregation window of 7 days. Our initial approach was to fit a model per respondent and use all features. This would cause a curse of dimensionality as we have many features versus very few observations per respondent. As a solution we implemented a Principal Component Analysis (PCA) to reduce the number of dimensions. However, we later decided to encode the respondent id as a feature, meaning we can use all of the data at once. Besides, we used feature selection strategies to come up with a final set of features (Section 2.6). Therefore, our experimental setup for these two models is as follows. First, we fit both models on the train set with only the selected features and evaluate on the validation set. We repeat the process on the data set where all features are included, but with PCA applied. For the evaluation we use the Mean Absolute Error (MAE) in a 10-fold cross validation. Using the optimal method (selected features vs. PCA) we look at the effect of using 2, 3, 4 and 7 days as aggregation period. Once we obtained these results we report the performance of the optimal model on the test set.

5 Results

In this section, the best non-temporal and best temporal model are compared to each other and the baseline. Performance is judged in terms of the mean-absolute error. Figure 5 shows the results of the different non temporal models. Here, we observe two things. (i) Regression Tree outperforms the SVR and (ii) the PCA does not perform as well as the models with pre-selected features. The latter is most likely caused by an overfitting problem, as we still include all of our features and the PCA only reduces the dimensionality of the data. Next, we test for the effect of the window size of the data aggregation on the Regression Tree with selected features. We find that a window size of $k = 7$ gives the lowest MAE of 0.022 ± 0.0219 vs. 0.056 ± 0.0756 for $k = 2$, 0.028 ± 0.0282 for $k = 3$ and 0.03 ± 0.0191 for $k = 4$. For the temporal ARIMA model, out of the 27 different parameter sets, 9 were closely matched, but the best set of parameters for p, d and q is 0, 1, 2, respectively.

The performance of the non-temporal model (Regression Tree with feature selection and window size of 7) and temporal model ARIMA(0,1,2) model, compared to the baseline, can be seen in Figure 6. In this figure we observe that the Regression Tree is better than the baseline, but also outperforms the ARIMA model. The ARIMA model is better than the baseline in terms of the mean of all folds, but is certainly not as stable as the Regression Tree as the MAE is sometimes even worse than the baseline. Based on these outcomes we selected the Regression Tree with feature selection and window size of 7 as our final model. This model yields an MAE of 0.011 on the test set. Note that this is even lower than the mean of 10-fold cross validation.

6 Conclusion and discussion

In this research we have looked at several models to predict the average mood of a person based on the historical smartphone usage data. We have seen that a Regression Tree with selected features trained on aggregated data of the past 7 days was the optimal model. The MAE of the Regression Tree is lower by a large margin, which may cause suspicion about its validity. However, we think there are two explanations for this. First, some of the features that ended up in our final feature set were selected based on the results of the Extra Trees model. Despite its randomness, it still uses the same measure to determine feature importance as the Regression Tree which is the Gini importance. Therefore these models are closely related. Second, the target variable only consists of 59 unique values,

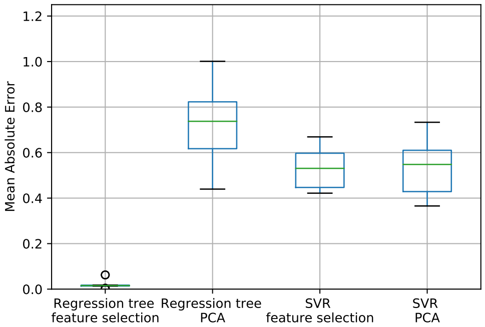

Fig. 5: MAE (10 fold) for non temporal models

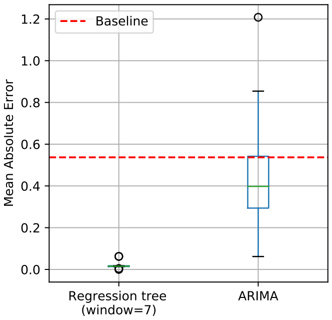

Fig. 6: MAE (10 fold) for final models

due to averaging on the 10 possible values for the mood score. This makes it relatively easy for the Regression Tree to predict, because there are few splits to be made in the tree. As a consequence, the Regression Tree is almost always correct resulting in a low MAE. For this specific application, average mood prediction, the Regression Tree is therefore a very powerful model.

During our research, we came up with several features which could have improved our model. We found multiple academic papers where GPS data was included in the data set ([7], [15], [16], [17], [2]). Furthermore, [2] used on-campus WiFi-tracking to pinpoint the indoor location of students. We suspect that exam weeks have strong impact on student's mood, hence the time spent in the university's library could serve as a proxy for the studying intensity. In hindsight, we wish we had included Dutch weather and holiday data in the feature set. Sunny days could very well lead to improved moods. Dutch national holidays drew our attention close to the deadline, hence we were not able to incorporate it into our models. Nevertheless, we noted that several holidays occur in April and May. Namely, Easter (20th and 21st of April), Kings Day/Night (25th and 26th of April) and Second World War Memorial and Liberation Day (4th and 5th of May). Kings Night and Day had strong impact on the mood on Kings Night/Day. For example, respondent 27 reported the highest daily mood on Kings Day (9.3).

References

1. Atas, A. H., Celik, B. (2019). Smartphone use of university students: patterns, purposes and situations. Malaysian Online Journal of Educational Technology, volume 7 - issue 2 (2019).
2. Ware, S., Yue, C., Morillo, R., Lu, J., Shang, C., Bi, J., Kamath, J., Russel, A., Bamis, A., Wang, B., (2020). Predicting depressive symptoms using smartphone data. Elsevier Smart Health vol. 15 (2020).
3. Ibrahim, A. K., Kelly, S. J., Adams, C. E., Glazebrook, C. (2013). A systematic review of studies of depression prevalence in university students. Journal of psychiatric research, 47(3), 391-400.
4. Asselbergs, J., Ruwaard, J., Ejdys, M., Schrader, N., Sijbrandij, M., Riper, H. (2016). Mobile phone-based unobtrusive ecological momentary assessment of day-to-day mood: an explorative study. Journal of medical Internet research, 18(3), e72.
5. van Breda, W., Pastor, J., Hoogendoorn, M., Ruwaard, J., Asselbergs, J., Riper, H. (2016, June). Exploring and comparing machine learning approaches for predicting mood over time. In International Conference on Innovation in Medicine and Healthcare (pp. 37-47). Springer, Cham.
6. Russell, J. A. (1980). A circumplex model of affect. Journal of personality and social psychology, 39(6), 1161.
7. Wang, R., Wang, W., DaSilva, A., Huckins, J. F., Kelley, W. M., Heatherton, T. F., Campbell, A. T. (2018). Tracking depression dynamics in college students using mobile phone and wearable sensing. Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies, 2(1), 1-26.
8. Ghandeharioun, A., Fedor, S., Sangermano, L., Ionescu, D., Alpert, J., Dale, C., ... Picard, R. (2017, October). Objective assessment of depressive symptoms with machine learning and wearable sensors data. In 2017 seventh international conference on affective computing and intelligent interaction (ACII) (pp. 325-332). IEEE.
9. Suhara, Y., Xu, Y., Pentland, A. S. (2017, April). Deepmood: Forecasting depressed mood based on self-reported histories via recurrent neural networks. In Proceedings of the 26th International Conference on World Wide Web (pp. 715-724).
10. Box, G. E., Jenkins, G. M., Reinsel, G. C., & Ljung, G. M. (2015). Time series analysis: forecasting and control. John Wiley Sons.
11. Kroenke, K., Strine, T. W., Spitzer, R. L., Williams, J. B., Berry, J. T., Mokdad, A. H. (2009). The PHQ-8 as a measure of current depression in the general population. Journal of affective disorders, 114(1-3), 163-173.
12. Loh, W. Y. (2011). Classification and regression trees. Wiley interdisciplinary reviews: data mining and knowledge discovery, 1(1), 14-23.
13. DeMasi, Orianna, Konrad Kording, and Benjamin Recht. ”Meaningless comparisons lead to false optimism in medical machine learning.” PloS one 12.9 (2017): e0184604.
14. Wickham, H. The Split-Apply-Combine Strategy for Data Analysis. Journal of Statistical Software vol. 40 (2011).
15. Cao, J., Truong, A. L., Banu, S., Shah, A. A., Sabharwal, A., Moukaddam, N. (2020). Tracking and predicting depressive symptoms of adolescents using smartphone-based self-reports, parental evaluations, and passive phone sensor data: development and usability study. JMIR mental health, 7(1), e14045.
16. Canzian, L., Musolesi, M. (2015, September). Trajectories of depression: unobtrusive monitoring of depressive states by means of smartphone mobility traces analysis. In Proceedings of the 2015 ACM international joint conference on pervasive and ubiquitous computing (pp. 1293-1304).

17. Thakur, S. (2018). Predicting Depressive Symptoms in Students using Smartphone-based Sensor Data.