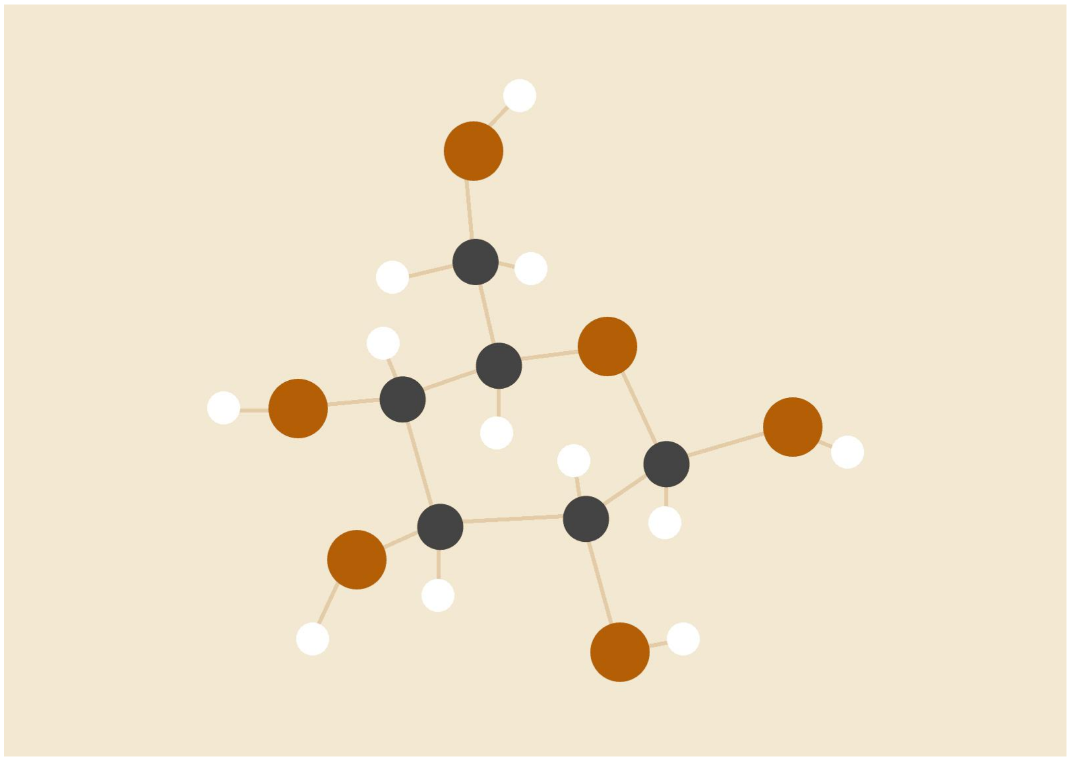
Group Project: Using Phone Sensors to Detect Student Mental Health

COMP9417 Machine Learning and Data Mining



Group Name: NewStar

Group Members:

Peiwen Lyu(Z5191791)

Shaowei Ma(Z5238010)

Wenke Yang(Z5230655)

Hao Zhong(Z5195835)

Catalogue

1. INTRODUCTION 1
2. Dataset Overview 1
2.1 Input dataset overview: 1
2.2 Output dataset overview: 2
2.3 Data grouping: 2
3. Methods 3
3.1 Input Pre-processing: 3
3.1.1 Input feature extraction: 3
3.1.2 Outlier: 4
3.1.3 Down-sample processing: 4
3.1.4 Normalization: 4
3.2 Output pre-processing 5
3.2.1 Missing value processing 5
3.2.2 Output missing questionnaire: 5
3.2.3 Missing post replacement 5
3.3 Prediction Models Selection 6
3.3.1 KNN 6
3.3.2 Linear Regression 6
3.3.3 Random Forest 6
3.3.4 Support Vector Machine 6
3.4 Feature Engineering 7
3.5 Hyper-Parameter Tuning Method 7
3.6 Cross Validation 7
3.7 Evaluation Metrics 7
4. Results 8
4.1.1 Feature selection 8
4.1.2 Evaluation 8
4.1.3 summary 9
4.2.1 Model evaluation metrics: 9
4.2.2 Feature selection: 9
4.2.3 Model fit and evaluation 10
4.4.1 Feature Selection 11
4.4.2 Parameter tuning 12
4.4.3 Summary 12
5. Discussion 12
5.1.1 Metrics comparison 12
5.1.2 Features comparison 14
(Figures attached in Appendix4) 14
5.2.1 Knn 14
5.2.2 Random forest 15
5.2.3 Linear regression 15
5.2.4 SVM 15
6. Conclusion 15
7. Reference 17
8. Appendix 1 18
9. Appendix 2 19
10. Appendix 3 20
11. Appendix 4 25
12. Appendix 5 27

1. INTRODUCTION

Nowadays, the fast-paced world not only brought an explosive growth of knowledge and information but also a significant increase in the stress of university students. According to the reports of Australia's Youth Mental Health Foundation, HeadSpace, in 2018, the university students who rated their mental health as poor or fair is more than 70%[1]. This indicates that the whole society needs to pay more attention to the mental health of university students and provide help at an early stage.

This project aims to predict the mental health conditions of university students using the data collected from their mobile phones. The data includes information about students' activities, social environments, etc. Also, another aim is to find the most significant factors that could indicate the mental health of the students. The results would be useful to monitor students' mental conditions by assessing their phone data and provide help to them as soon as there are some negative signals.

In this project, we applied several different machine learning techniques to predict the students' mental health. For pre-processing, we applied Isolation Forest for finding outliers, Max-Min for normalization. For predicting mental health, three classification methods: K-Nearest Neighbours, Random Forest, Gradient Boosting Decision Tree were applied; and two regression methods: Linear Regression, Support Vector Classification were applied. For comparison and evaluation, basic accuracy, AUC-ROC score, and F1 score were applied. These methods would be explained in detail in the following sections.

This project report is made of five sections. The first two sections are the brief introduction of the project and the dataset description. Followed by the methods section which explains the methods choices in detail and prep-processing to the dataset, as well as the application of methods to the dataset. Then, the fourth section presented the results and comparison of methods. Finally, the last section would give a summary of the finding related to students' mental health.

2. Dataset Overview

Before we start working on the data, let's take a quick look at the data set.

The StudentLife dataset consists of 2 parts: Input dataset and Output dataset. Input is collected using automatic sensors and output is collected through self-reported questionnaires.

2.1 Input dataset overview:

Input Dataset	Description
1.Activity	Periodically infer and sample physical activity type of students. Data has 2 columns, the first column is Unix timestamp and second is activity inference, id 0,1,2,3 refer to 'Stationary', 'Walking', 'Running' and 'Unknown' respectively.
2.Audio	Periodically sample the level of environmental noise. Data has 2 columns, the first column is Unix timestamp and second is noise level, level 0,1,2,3 refer to 'Silence', 'Voice', 'Noise', 'Unknown' respectively.
3.Conversation	Detect and record start/end time of each conversation.

4.GPS Location	Collect and record GPS coordinates and some other relevant information every 10 minutes.
5.Bluetooth	Scan surrounding Bluetooth devices every 10 minutes and record MAC, class_id and Signal Strength of all devices each scan
6.WIFI	Scan WiFi frequently and record BSSID, channel frequency and Signal Strength of all WiFi devices each scan.
7.WIFI Location	Data has 2 columns, the first column is Unix timestamp and the second is location inferences which can be in a building (e.g. in[kemeny]) or near some buildings (near[kemeny; cutter-north; north-main;]).
8.Light	Record start/end time when the phone was in a dark environment for a significant long time (>=1 hour).
9.Phone Lock	Record start/end time when the phone was locked for a significant long time (>=1 hour).
10.Phone Charge	Record start/end time when the phone was plugged in and charging for a significant long time (>=1 hour).

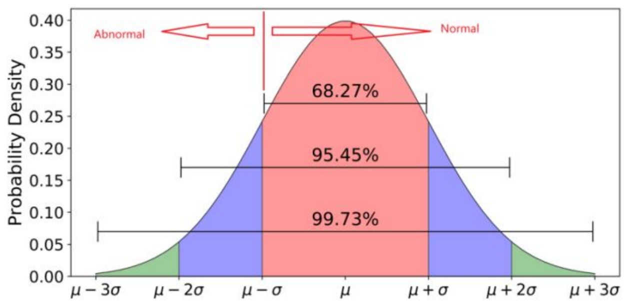
2.2 Output dataset overview:

Output Dataset	Description
1.Flourishing Scale	An 8-item summary measure of the respondent's self-perceived success in important areas. The final score can be got by adding responses varying from 1 to 7 for all eight items. Each row in the file represents 1 student.
2.Panas	10 positive items and 10 negative items, each of them has a score represent its extent. Positive affect score can be calculated by adding all scores positive items, and so for a negative score.

Note: Each student did 2 questionnaires during this project. 'pre'means the questionnaire was done before the project and 'post' means the questionnaire was done after this project.

2.3 Data grouping:

Grouping is implemented to the 2 output files with the assumption that flourishing score and panas score (positive and negative) obey normal distribution as shown below. We divide scores (Flourishing score, Panas Positive score, Panas Negative score) into 2 groups of "Normal" and "Abnormal". We set the threshold base on the assumption that 84% of students are "Normal". A score lower than the threshold will be classified as "Abnormal" and score higher than the threshold will be classified as "Normal".



However, we found this kind of grouping method will lead to a very unbalanced distribution, it's fine if the dataset is huge. While for this small dataset, it will be hard to train our model and the accuracy in the prediction will be unreliable. Hence, we finally decide to use the median as threshold although we still think grouping by normal distribution is more reasonable.

3. Methods

3.1 Input Pre-processing:

Before implementing feature extraction, we find that the timestamp in raw data is coordinated universal time, to separate data by days, we have to convert it to Dartmouth time(GMT-5) to local time.We also find some incomplete data students may have different data sizes. For activity data and audio data, students participate in this experiment are expected to keep the sampling frequency unchanged, so that their data size should be similar. However, the fact is that each student has a different number of records, even, data of some students is quite different. The missing data problem also occurs in the output.

3.1.1 Input feature extraction:

Dataset	features	Processing
Activity	activity_mean, activity_sum, activity_var	The data format of the 'physical activity inferences' file and the 'physical audio inferences' file are the same, so these two files can be processed in the same way. The activity inference value varies from 0- 3, indicating stationary, walking, running and unknown respectively. The sum of the data can represent the amount of exercise intensity, but value 3 cannot be directly used, it needs to be processed first. The student's state of motion is continuous, so the data item before the unknown can be used to replace the unknown. After handling the unknown state, we calculate the total amount of exercise for each student. However, we need more information to represent each specific student. Hence, we divide data by day and calculate the daily exercise amount. Also, we calculate the variance of these daily data as a measurement of the regularity of exercise amount.
Audio	audio_mean, audio_sum, audio_var	
Conversati on	conver_sum, conver_var, conver_mean,	We use the same feature extraction method with conversation data, dark data, phone_charge data, and phone_lock data. Similar to the processing of 'Activity', we divide the raw data of each student by day based on timestamp. For example, 'phone_lock' dataset record start/end time when the phone was locked for a significant long time (>=1 hour). We extract the sum of duration when the phone is in lock statue every day. Hence, we can get daily data of all the students. Then, we calculate sum of lock time during the experiment and divide by the number of days to get average lock time based on the day. Also, we calculate the variance based on the day because we want to extract information as much as we can. It can be a good measurement of the fluctuation of the lock time per day.
Dark	dark_sum, dark_var, dark_mean	
Phone charge	phcha_sum, phcha_var, phcha_mean,	
Phone lock	phlock_sum, phlock_var, phlock_mean	
Bluetooth	Bt _sum, Bt _var, Bt _mean	For Bluetooth, we also divide the raw data into a day. We calculate how many Bluetooth are recorded each day, and calculate the sum, mean and variance. We remove the duplicate Bluetooth address in each day.
Wifi Location	in_time, near_time in_all_percent age	Raw data has 2 columns, the first column is Unix timestamp and the second is location inferences which can be in a building (e.g. in[kemeny]) or near some buildings (near[kemeny; cutter-north; north-main;]). We find this data set can be a good measure of students' indoor time and outdoor time. Hence, we ignore these different building names and only focus on whether a student is in a building or near a building. Then we calculate the total "in" time and total "near" time(seconds), as well as the "in"/("in"+" near") percentage.

3.1.2 Outlier:

After a comprehensive exploration of the given dataset, we found some problems with the phone data, for example, some of the students do not attend the experiment from beginning to end, and mobile phones have different sampling rates, etc. Some of these behaviors were smoothed at the feature extraction stage, however, there are still some missing values in features that cannot be completed. Therefore, before training the model, we decided to treat the students with anomalous data as outliers and finally remove them.

To find outliers, we chose the model Isolation Forest, an unsupervised ensemble method that is based on the decision tree model. It partitions the trees by selecting a random feature, then, selecting a random split value between the maxima and minima of this feature. The outliers would be separated in a few splits, while segregation of normal data points would take a significant more split.

There are four main reasons why we choose this model. First of all, the distribution of the phone data features is not natural (not Gaussian), and the data size is too small, we cannot make any assumptions of the distribution, hence statistics methods cannot be used. Secondly, the correct outliers are not given, so we can only choose from unsupervised learning methods. Among the unsupervised learning methods, one of the most promising anomaly detection algorithms is Isolation Forest. Besides, it also support high dimensional datasets, in our case, we have approx. 30 features.

After applying the model from open-source package sklearn.ensemble.IsolationForest to the features, we found that the phone data from the students with uid u30, u39, u49, u59 has the most abnormal behavior. After examining these outliers, we found that they either have excessive samples or insufficient samples. This is believed to be the cause of machine error (e.g. problem with phone sensors), so we decided to not taking account of these data points while training the prediction model. We believe that this could increase the accuracy of predicting the mental health of the students with normal data.

3.1.3 Down-sample processing:

After the calculations of outliers, it shows that u30, u39, u49, and u59 are outliers. By observing the data detail of outliers, it can be found that the data size of u39 is much smaller than others, and the data size of u59 is much larger than others. After analyzing the data, we find that the student u39 not only has missing data(the number of days of participating in the experiment was about half of that of the normal students) but also has a low survey score.

Therefore, the data of u39 cannot provide useful information. Moreover, the reason why the data size of u59 is much larger than others is that his data collection frequency is higher(4 times) than the others. The data of u59 is continuous, so the data processing method of u59 is to divide the amount of data by 4. Then, based on the new data, the variance will be updated. The u30 and u39 exceptions could not be analyzed from the data, so no additional operation is performed to these two students. Processing by this method, u59 is no longer an outlier, finally, we select u30, u39, u49 as the outliers.

3.1.4 Normalization:

Normalization is used to pre-process the extracted feature because the measurement scales of the different attributes would be different. We use the following formula to process the data.

It means that the value of the feature would be normalized between 0 and 1. The max value is 1 and the min value is 0.

$$\mathrm{x}_{\mathrm{r}}^{'} = \frac{x_r - \min(x_r)}{\max(x_r) - \min(x_r)}$$

3.2 Output pre-processing

3.2.1 Missing value processing

There are some missing value in flourishing dataset and panas dataset, we replace the missing value by median in that column.

3.2.2 Output missing questionnaire:

In the input dataset, there are 49 students provided their data, we record uid of these students and compare it to the output dataset (FlourishingScale.csv and panas.csv).

In the output dataset, we find 4 kinds of combination of whether the student completed the 'pre' questionnaire and 'post' questionnaire.

Type	Example	Pre questionnaire	Post questionnaire
1	u00, u01...	Yes	Yes
2	u12, u13...	Yes	No
3	u54	No	Yes
4	u25, u41...	No	No

For students of type 1 and 3, we can use their 'post' data directly, while for students of type 2, we have to handle these missing 'post' values. For students of type 4, we discard their data because we can do nothing with them, but we can predict their score using our model.

To predict these missing 'post' value, we have to make an assumption that there has some relevance between pre-score and post-score so that we can predict post-score by pre-score by using some machine learning model.

3.2.3 Missing post replacement

By examining the output, we found that only 35 and 36 out of 46 (originally 49, 3 are removed as outliers) students have completed post questionnaire of PANAS and Flourishing respectively. This indicates that more than 20% of the students cannot be used for training the prediction model, whereas the original dataset is already too small to train. After further counting, it was found that among the students who do not have post labels, 80% of them had pre labels. From a logical perspective, we made an assumption that some students attend the pre questionnaire but did not attend the post questionnaire is because they believe their mental condition did not change much from pre questionnaire (i.e. the students with missing post label can be filled in with their pre label if pre is available).

In order to prove our assumption, we did some experiments on the dataset. We decided to use available pre and post labels and sensor data for training, and predict the missing post labels to compare with the pre labels. If the missing post and pre labels highly matched each other, this could prove that our assumption is correct. In our experiments, we separated students into three groups:

1st group: the students have pre and post labels.

2nd group: the students have pre labels but do not have post labels

3rd group: the students do not have pre and post labels

The 1st group is separated into 3 folds, we trained some selected models (model details introduced in Prediction Models section) using 2 folds, and do cross validations on these 2 folds. The results of cross validations would be used for comparing of our models pair-wisely. Then, use the rest 3rd fold to evaluate if the models are overfitting with the previous

2 folds (i.e. if a model has the highest accuracy in all cross validations but a poor accuracy in fold 3, then the model overfits the previous 2 folds, hence cannot be selected as the best). After selecting the model with the best performance, we trained a prediction model using this model with all data points in 1st group, and compare the predicted missing post labels with the pre labels. The experiment details of missing post are attached in appendix5

The conclusion of experiments is that the predicted missing post labels are similar to the pre labels. In addition, after examining the activity, social and other sensor data collected from these students with missing posts, we did not find any abnormal behaviour or missing data. This proves that our assumption is reasonable. Therefore, we filled in the missing post labels with the corresponding pre labels for further processing.

3.3 Prediction Models Selection

3.3.1 KNN

K-nearest-neighbors algorithm is a non-parametric method used for classification and regression. K-NN is a supervised classification algorithm and it uses distance to classify the test set. For continuous variables, Euclidean distance and Manhattan distance are commonly used as the distance metric.
At first, it requires a training set, then the user can input a test set. For each sample in the test set, the distances to all points in the train set will be calculated. And the nearest K points will be selected to count the number of points of each class. The test sample point will be classified as the class with most points in these K points.

3.3.2 Linear Regression

Linear regression is a basic and commonly used machine learning approach. The key idea of linear regression is to model the relationship between a dependent variable and one or more independent variables. The learned relationships are linear and can be written as follows:

$$\hat{y} = \theta_0 + \theta_1 + \theta_2 + \ldots + \theta_n x_n$$

The biggest advantage of linear regression is linearity which makes the estimation procedure simple. Most importantly, these linear equations have an easy understand interpretation on a modular level and this is the mean reason why we choose it as one of our model.

3.3.3 Random Forest

Random forest is a classifier that contains multiple decision trees, and it is capable of regression and classification. Random forest is suitable for classification because it is done in a random state during the process of selecting training data and classification features. Therefore, the probability of random fitting in Random Forest is smaller, and it has good generalization ability. When dealing with multi-dimensional features, feature selection may not be required due to the random selection of features.
The model is highly adaptable to data, applicable to both continuous and non-continuous data, and data does not need to be normalized. The random forest can get the importance of each feature, using this result to facilitate feature selection. To prevent Random Forest from overfitting, one way is to control the depth of each tree, because deep trees are likely to overfit; another method is to cross-validate the model.
For this project, it is reasonable to choose Random Forest as one of the models, because there are many features as input, each feature has a different proportion of influence on the result, and random forest can handle this problem well.

3.3.4 Support Vector Machine

Support Vector Machines (SVM) are supervised machine learning models that can classify data points by generating the best separating hyperplane for the given dataset. This best separating hyperplane is defined as a hyperplane among all hyperplanes which can classify data points correctly that has the maximum margin between the two classes. In our example, the value to be predicted is the post labels(high or low) for PANAS and Flourishing. The SVM would be used to train a prediction model for post labels by learning the sensor data as features, then, this model can be used for further classification of the post labels.

The most important reason that we chose SVM as one of the prediction models is that the volume of our dataset is very small. The fatal flaw of small data volume is that most machine learning methods can result in overfitting with the training dataset. However, linear models are comparatively not easy to overfit. Besides, the SVM is interpretable and the parameter tuning is simple.

3.4 Feature Engineering

To find the best feature combination, two main algorithms are applied.

The first one is the brute force algorithm. First of all, we calculated the time complexity to try all combinations for all features which are O((2^#features) * training time), in our case, it is O((2^27) * training time). Then, we measured the computation capability of the personal computer we use. It takes approximately 0.01 seconds to train a model with the given features and get the test results. Since we would like to control the training time within 5 minutes, we decided only to choose 15 most important features and try all combinations with them.

Hence, we sorted the features according to the feature importance. For example, the coefficients of linear SVM can be interpreted as the feature importance here. Finally, we selected the combination of features that gives the best performance on most metrics. The metrics we use would be introduced later.

The second one is a greedy algorithm. We started with an empty feature list, then, iteratively add one to the list if any of the metrics are improved, and the rest remains the same. Finally, if none of the metrics can be improved, the iteration stops. The final best feature combination would be chosen from the final feature list and the results from the brute force algorithm. Both algorithms have the cross-validation results as one of the metrics to reduce the overfitting.

3.5 Hyper-Parameter Tuning Method

For hyper-parameter tuning, we chose to apply the Grid Search algorithm with cross-validation. The general idea of Grid Search is to firstly define a large range of parameters, try all combinations within this range, then, iteratively narrow down the range to find the best parameter set. The combination of Grid Search and cross-validation can prevent the parameters tuned to be overfitted on the training dataset to some extent.

3.6 Cross Validation

For cross-validation, two methods were applied. The first one is three and four folds cross-validation. This cross-validation method divides the training dataset into 3 or 4 chunks with the same size, each time one chunk is left for validating the performance, and the rest chunks are used for training the model. The chance of overfitting could be effectively reduced by comparing the results of cross-validation. As for training the final post label, four-fold cross-validation is used. While training the missing post in the pre-processing stage, three folds cross-validation is applied since the volume of the dataset is even smaller.

The second one is Leave One Out (LOO) cross-validation. This is a special cross-validation that uses only one example for validation and all rest(n-1) examples for training. LOO can cause a huge amount of time for training but it is believed to perform well on small datasets. Since our dataset only has 49 students in total, LOO can be used as one of the references.

3.7 Evaluation Metrics

Metrics	Description
Precision	Ratio of the number of positive samples correctly predicted to the number of positive samples for all predictions, that is, how many of the samples with positive predictions are true positive samples. Precision only focuses on the part of the prediction that is positive, while accuracy considers all samples.
Recall	Ratio of the number of positive samples correctly predicted to the total number of true positive samples, that is, how many positive samples can be correctly obtained from these samples.
Accuracy	Ratio of the number of correctly predicted samples to the total number of predicted samples. It does not consider whether the predicted samples are positive or negative.
F1 Score	Harmonic mean of precision and recall, used to refer to two indicators. Any value of recall and precision is reduced, and F-score is also reduced.
AUC	AUC is the area under the ROC curve. The ROC curve can remain unchanged as the distribution of positive and negative samples in the test set changes. AUC considers the classification ability of the classifier for positive and negative cases, and it can still make a reasonable evaluation of the classifier in the case of sample imbalance.

4. Results

4.1 KNN

4.1.1 Feature selection

In the feature selection part, we use the combination method provided by python itetools to generate our feature set. At first, we use one feature to train our model, and we find the accuracy is close to 50%. Then we try the number of features from 2 to 10, and the accuracy is the highest when we use three features.

After determining feature number, we do the feature selection and find the following feature combination have higher accuracy than others.

[1] ['activity_mean', 'activity_var', 'audio_mean'],

[2] ['activity_mean', 'activity_var', 'bluetooth_mean'],

[3] ['activity_mean', 'activity_var', 'conversation_sum'],

[4] ['activity_mean', 'activity_var', ''phonelock'],

[5] ['activity_mean', 'bluetooth_sum', 'phonelock_var']

[6] ['activity_var', 'phonecharge_var', ' phonecharge_mean '],

[7] ['conversation_day_num', 'phonecharge_mean', ' phonelock ']

We find the 3rd,4th, 5th and 6th combination can achieve the highest accuracy of feature combination set. Then, we vary the neighbor number from 2 to 8, and get the max_ accuracy from those feature combinations.

We find that when k =3, most of these models perform well. Thus, we use k=3 in K-NN and the four feature combinations to predict our test set.

4.1.2 Evaluation

In this part, accuracy is also used as evaluation metrics.

For flourishing

We use some data to test our module, we find then, the feature combination ['activity_mean', 'bluetooth_sum', 'phonelock_var'] is better than other feature combination.

For training set,

auc_score	accuracy	F_Measure	precision	recall'
0.82	0.85	0.89	0.89	0.89

CV = [0.64, 0.9, 0.67]

For test set

auc_score	accuracy	F_Measure	precision	recall'
0.17	0.23	0.375	0.43	0.33

For panas positive and negative we do the same thing as flourishing

Positive

The best feature combine is ['activity_mean', 'activity_var', 'audio_mean'] k = 3

CV = [0.35, 0.92, 0.23]

auc_score	accuracy	F_Measure	precision	recall'
0.9375	0.928	0.933	1.0	0.875

Negative

The best feature combine is ['activity_mean', 'conversation_sum', 'phonelock_mean'] k = 3

CV = [0.3, 0.85, 0.6]

auc_score	accuracy	F_Measure	precision	recall'
0.9375	0.928	0.933	1.0	0.875

4.1.3 summary

For K-NN, we use accuracy as evaluation metrics to select the feature and neighbor number. However, through cross validation and the evaluation for test set, we find that in a small data set, K-NN would overfitting for the training set. Its accuracy for panas positive and negative is high, but due to the scale of data set, we think it may still overfitting.

4.2 Linear Regression

We have 3 processed datasets:

1. All features with the target of flourishing score
2. All features with the target of panas positive score
3. All features with the target of panas negative score

All of these 3 datasets are similar, so we choose the first one as an example to illustrate the development of our method. Our development contains 3 steps: 1. Model metrics, 2. feature selection, 3. Model fit, 4. Model evaluation.

4.2.1 Model evaluation metrics:

For linear regression, we choose Mean Squared Error (MSE) as metrics to compare models trained by different feature combinations.

4.2.2 Feature selection:

We implement 2 methods for feature selection:

1) SelectKBest: All features in the dataset are normalized and non-negative, so we use the chi-squared statistical test for features to select 20 of the best features and preserve them in descending order.

2) Correlation: We use the corr() method provided by pandas dataframe to calculate the correlation between feature and target. We select 20 features with the highest correlation and preserve them in descending order.

Now, we have 20 best features selected by SelectKBest and another 20 best features selected by correlation. Let’s call them SKBest_20 and CorrBest_20. For each of these 2 lists, we use the same method to get the best feature combination.

First, choose the feature with the highest score and train the model with this feature and get corresponding evaluation metrics. Then add the feature with the second-highest score and get evaluation metrics. Then add the feature with the third-highest score and so on and so forth. Finally, we will get a set of metrics. We will choose the feature combination with the best evaluation metrics as my features to train the model.

4.2.3 Model fit and evaluation

We choose the first 30 samples as the training set and the remaining as a test set. I fit my model with features selected by the previous step and predict the last 13 samples. The predicted values are: [44.73, 40.61, 45.84, 39.54, 36.11, 36.00, 46.32, 43.18, 46.09, 41.16, 44.30, 45.50,39.95]

In order to compare my model with other 3 models, I group my result into 2 classed based on the threshold(median) which is 44, the result is: [1, 0, 1, 0, 0, 0, 1, 0, 1, 0, 1, 1, 0]

Compare the result with the actual label, some metrics can be got as shown below:

auc_score	accuracy	F_Measure	precision	recall'
0.653	0.615	0.667	0.833	0.556

For panas positive and panas negative, the result and metrics are:

Panas positive: [0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 1, 0, 1]

auc_score	accuracy	F_Measure	precision	recall'
0.687	0.642	0.545	1.0	0.375

Panas negative: [0, 0, 1, 1, 1, 0, 0, 0, 0, 0, 0, 1, 1, 1]

auc_score	accuracy	F_Measure	precision	recall'
0.583	0.571	0.571	0.667	0.5

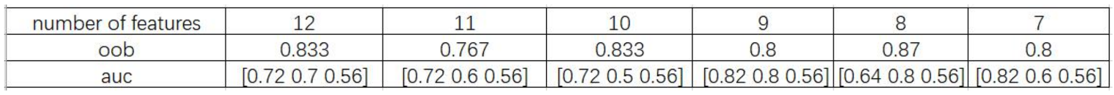
4.3 Random Forest

On account of the randomness of random forests, the model is randomly generated even if the parameters and input features are not changed. Therefore, it is necessary to set the random_state parameter and keep it unchanged each time. Consider the condition that input data are not well recorded and data of some days' is missing, the even worse condition is that different student has different missing days.

So, I only choose the mean features(e.g. activity_mean) and variance features(e.g. activity_var) because they can describe the original information to the greatest extent.

Use the combination method provided by python itertools to try out a combination of features to maximizes the model's oob which is an unbiased estimate of the random forest generalization error, and its result is similar to the k-fold cross-validation that requires a large number of calculations.

Iterate through all the combined results from the total number of input features from 12 features to 7 features, and preserve the best features combination as well as its oob score and auc score. The difference between 3 times auc score should not be too large, otherwise, the model is over-fitting. Feature reduction can also be done by comparing the feature importance (in appendix2).



number of features	12	11	10	9	8	7
oob	0.833	0.767	0.833	0.8	0.87	0.8
auc	[0.72 0.7 0.56]	[0.72 0.6 0.56]	[0.72 0.5 0.56]	[0.82 0.8 0.56]	[0.64 0.8 0.56]	[0.82 0.6 0.56]

Two scores are used to evaluate the quality of the model, oob and AUC. The importance of each feature is shown in the figure above. As can be seen, the model performs better when the total number of features is 8, but the result of the

cross-validation shows that the model is over-fitting. Hence, we still choose the first feature combination as the input to train the model.The next step is to tune the hyperparameters.
An important parameter of random forests is n_estimators, which means the maximum number of decision trees in a random forest. The most important parameters of the decision tree include the maximum number of features, the maximum depth, the minimum number of samples required for internal node subdivision, and the minimum number of samples of leaf nodes. Because the sample size is small in this project, we find that many parameters do not need to be tuned and the default value can be used directly. Generally, if n_estimators is too small, it is easy to underfit. If n_estimators is too large, the amount of calculation will be too large. After n_estimators reaches a certain number, the model's performance will not be better anymore, so choose a moderate value is important. The default value of n_estimators is 100. Find the value that optimizes the model with gradient of 50. Use the grid search method to tune the parameters, tune n_estimators and max_features together. Set max_features to 7, n_estimators to 114, get the new random forest model and the scores.

n_estimators	max_features	oob	auc(cv mean)		auc	precision	recall	accuracy	F1
defult	defult	0.833	0.66		0.472	0.5	0.11	0.3	0.18
114	7	0.633	0.691		0.167	0.25	0.11	0.15	0.15

Comparing these scores to those before tuning, the new model fits the training data better but the AUC of the test data decreases. This proves that the model is over-fitting after the tuning, so we use the default parameter finally.
Feature selection and parameter tuning in the same way, the results of the panas are obtained.
Panas_positive

n_estimators	max_features	oob	auc(cv mean)		auc	precision	recall	accuracy	F1
defult	defult	0.8	0.6		0.604	0.6	0.375	0.5	0.462
146	6	0.8	0.7		0.625	0.667	0.5	0.57	0.57

Panas_negative

n_estimators	max_features	oob	auc(cv mean)		auc	precision	recall	accuracy	F1
defult	defult	0.833	0.667		0.24	0.25	0.125	0.286	0.167
106	2	0.633	0.7		0.448	0.6	0.375	0.5	0.462

4.4 Support Vector Machine (all tables and figures of the section are attached in the appendix 3)

In this section, the results of feature selection and hyper parameter tuning using SVM model will be presented with figures and a detailed analysis. The measuring metrics for feature selection and hyper parameter tuning are training accuracy and minimum training data cross validation accuracy. The rest metrics in the tables below will be used for results analysis. This includes testing set AUC, accuracy, precision, recall and F-measure. At the beginning, the results of Flourishing post label training will be presented. This is followed by the PANAS positive and PANAS negative post label training results. Finally, a summary of performance of SVM model on the given dataset is given. Since the feature selection and hyper parameter tuning processes for Flourishing and Panas are similar, we will only present the Flourishing process as an illustration.

4.4.1 Feature Selection

According to the feature engineering methods mentioned in the Methodology section, we will apply two methods: brute force and greedy. For the brute force method, we started with selecting all 27 features and got a list of features with sorted importance. Then, experiments are done with all combinations of the top 15 features, the best result of Flourishing is presented in the Final column of the table - Flourishing post label Feature Selection. In addition, greedy algorithm was also performed. First, we performed the single best feature selection and found the feature dark_var gives the best performance on the metrics. Then, we iteratively added more features if the performance improved. However, the final greedy results does not perform as well as the brute force results in testing dataset, the greedy testing AUC is 0.61, whereas the brute force testing AUC can reach 0.67, all rest metrics are similar. This might because the greedy algorithm

does not consider the some of the combinations. For example, the best individual feature is f1 and it iteratively added f2, but actually the combination of f2 and f3 might give a better result than f1 and f2. Therefore, we only recorded the best result in the table which is the brute force one.
By comparing the final best results with the initial results, the cross validation results has a significant improvement. For the initial result of Flourishing, although the training accuracy is 0.83, higher than the final result, the minimum cross validation accuracy is only 0.43. This is a clear sign of overfitting which indicates that some of the features in the 27 features overfit the training data but does not generalise well on other dataset. After feature selection, the minimum cross validation accuracy is improved to 0.71, and the testing accuracy and AUC are both increasing, this illustrates that the accuracy of testing data and training data is closer to each other which is a sign of reduced overfitting. For Panas positive and negative, the cross validation results has also improved a lot after feature selection. Therefore, this proves that our feature selection method can effectively reduce the overfitting while increasing the testing accuracy at the mean time.

4.4.2 Parameter tuning
As to the hyper parameter tuning, two of them can be tuned to improve the model performance: C - the penalty parameter and kernel - linear or rbf. The linear kernel represents the linear model, the rbf kernel is a radial basis function model that can support classification of non-linear data. For both Flourishing and PANAS predictions, linear kernel performs better than rbf kernel. This is because our dataset is extremely small which can easily lead to overfitting, however, linear model usually does not fit the data exactly due to the linear characteristics. This makes the linear model performs better in this dataset. For the penalty parameter, we have performed grid search on the value. At the beginning, we started with a large range, from 0.1 to 1000 in 10 times interval. Finally, we narrow the range from 1.0 to 2.0 since we found that 1.0 and 2.0 both gives the best performance. However, after throughout testing, the default penalty parameter 1.0 is still the best.

4.4.3 Summary
In conclusion, the feature selection improves the training and testing accuracy but since the dataset is small, the parameter tuning did not make any significant difference. Among Flourishing, Panas positive and Panas negative, the processes are all similar, but the accuracy of Panas negative is the highest. The reason might be the sensor data collected are most related with the Panas negative. This is probably because when the students are filling in the questionnaires, the Panas negative can strongly show their emotions if they have negative mood.

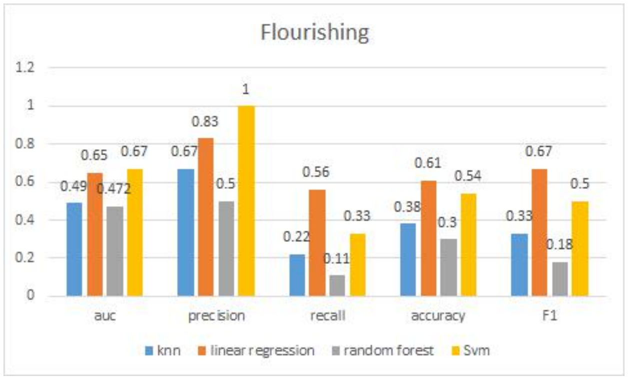
5. Discussion

5.1 Comparison of performance among different methods
After gathering the prediction results from all models, we summarised the important factors into the tables and figures shown below.

5.1.1 Metrics comparison
As to Flourishing, Linear Regression has the best performance on most metrics, including F1 and accuracy of the test dataset. Besides, it has the lowest variance on cross validations and test results compared to all other models which shows that the linear regression model has the best generalisation ability. The SVM model also illustrated a competitive results on all the metrics, however, the difference between the cross validation accuracy and test accuracy is more than 20% which indicates an obvious overfitting. For KNN and Random Forest, the performance on training the testing results

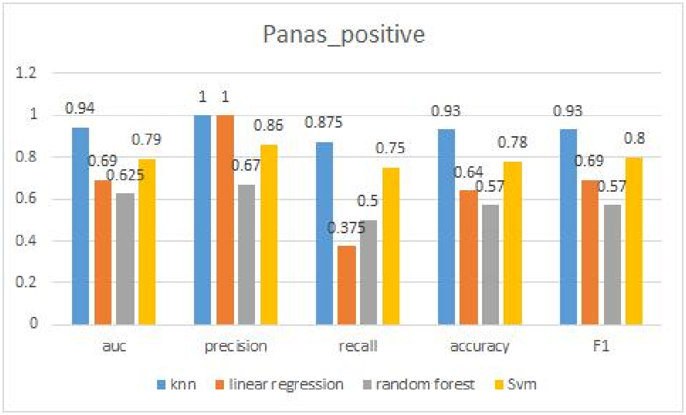
are both poor.

Method	Score(cv_accuracy)	auc	precision	recall	accuracy	F1
knn	[0.36 0.8 0.67]	0.49	0.67	0.22	0.38	0.33
linear regression	[0.73 0.8 0.7]	0.65	0.83	0.56	0.61	0.67
random forest	[0.73 0.7 0.56]	0.472	0.5	0.11	0.3	0.18
Svm	[0.78 0.71 0.71 0.86]	0.67	1	0.33	0.54	0.5



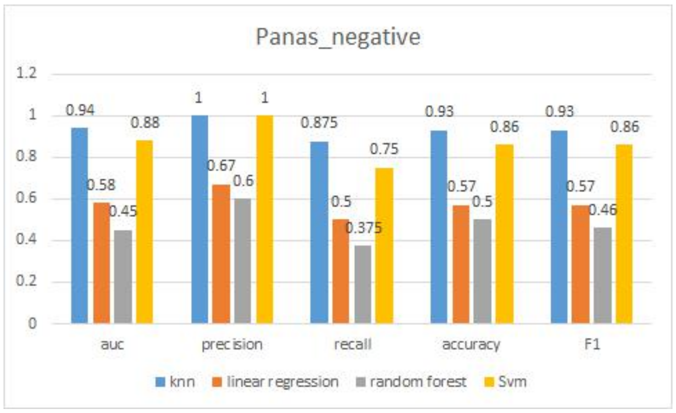
For Panas positive, KNN has the best performance on test dataset but an extremely high variance on cross validation results. This is because we used the average of cross validation accuracy as one of the standards to select features and tune hyper parameters. This indicates that for the unbalanced folding, a minimum of cross validation accuracy is more persuasive than the average. In general, KNN for Panas positive illustrated a severe overfitting on test dataset and one of the training folds. Random Forest has the worst performance on most of the metrics and shows a sign of overfitting from the testing and training accuracies. For the two linear models, Linear Regression(LR) and SVM, both of them shows a lower change of overfitting because the training cv results and testing results are similar. However, SVM has a generally higher accuracy on all metrics than LR. In conclusion, SVM has the best general performance for Panas positive.

Method	Score(cv_accuracy)	auc	precision	recall	accuracy	F1
knn	[0.35 0.92 0.23]	0.94	1	0.875	0.93	0.93
linear regression	[0.6 0.63 0.67]	0.69	1	0.375	0.64	0.69
random forest	[0.8 0.8 0.5]	0.625	0.67	0.5	0.57	0.57
Svm	[0.63, 0.75, 0.63, 0.67]	0.79	0.86	0.75	0.78	0.8



For Panas negative, the results are similar with Panas positive. From the scores for test data, KNN is the best performing model, followed by SVM. But for KNN, its cross-validation accuracy reflects overfitting. Therefore, SVM performs better.

Method	Score(cv_accuracy)	auc	precision	recall	accuracy	F1
knn	[0.3 0.85 0.6]	0.94	1	0.875	0.93	0.93
linear regression	[0.6 0.6 0.55]	0.58	0.67	0.5	0.57	0.57
random forest	[0.7 0.8 0.6]	0.45	0.6	0.375	0.5	0.46
Svm	[0.75, 0.75, 0.75, 1.00]	0.88	1	0.75	0.86	0.86



In conclusion, both linear models perform better than the Random Forest and KNN models. Among the linear models, LR has the better performance for Flourishing and SVM has the better performance for PANAS. As to the metrics, both linear models has high results in the AUC, F1 and accuracy of test dataset and all three of them show similar results. This is because the threshold for the two classes high and low are divided according to the median, so the two classes are perfectly balanced. If the classes are divided unevenly, AUC and F1 might give more information than accuracy. Therefore, in our case, all three metrics are consistent and informative. For the training cross validation results, since taking the minima into account while training rather than average can effectively avoid overfitting, we selected SVM as the best model for both positive and negative Panas predictions.

5.1.2 Features comparison

(Figures attached in Appendix4)

In addition, we will compare the features selected. Firstly, we discussed the selected feature difference for different models. Followed by the general important feature discussion for different mental health indicators.

For all indicators, KNN model has the smallest number of features selected. When the number of features increases, the dimension increases. This could result in hard calculation for distances between data points for KNN. Linear regression and SVM are simple linear models which only considers the linear correlation between the features and the post values. Hence, the number of features selected for them is also relatively small. As to the Random Forest model, the general performance on the given dataset is the worst. The reason is probably because the decision tree needs high dimensional data to learn the patterns and the accuracy is high depending on the entropy. However, the small data volume and some unbalanced folds makes the decision tree hard to calculate entropy and hence not able to improve accuracy.

For Flourishing, the most popular feature is phonelock_var, it represents the variance in phone locking time during the whole experiment period. It is believed that the phone locking time can implicitly indicates the sleeping time of students. A large variance of phone lock time can show that the differences in sleeping time during the experiment period has big fluctuations. This could be regarded as a sign of high stress, hence related to flourishing. For both Panas positive and negative, the features related to phone using time, conversation durations and activity are most are most relevant.

5.2 Discussion of advantages and disadvantages of various methods

5.2.1 Knn

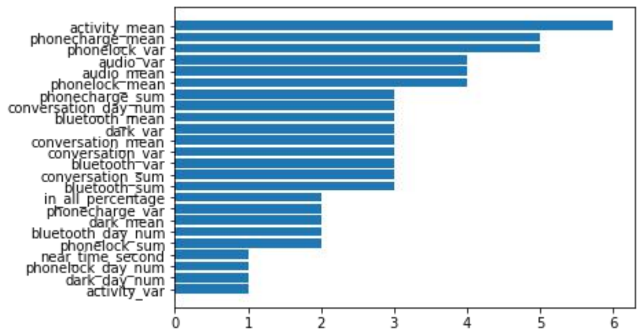
K-NN is a lazy learner, which means it does not learn anything from the training data. through analyzing the model result, we find that, for an unbalanced or small scale dataset, K-NN would be easily overfitting, but for a balanced dataset or a large scale dataset, K-NN could have high accuracy.

5.2.2 Random forest
Random Forest can process high-dimensional data and provide feature importance, so selecting feature would be more feasible. However, Random Forest will be overfitting in some cases with high noise.

5.2.3 Linear regression
For linear regression, finding the correlation between features and labels is feasible, so it would be beneficial to determine the model feature. However, for a non-linear correlation, linear would be useless.

5.2.4 SVM
Due to the robustness of SVM, in high dimensional spaces, SVM can still provide high accuracy, but when there are more than two classes, it would become a low accuracy classifier.

5.3 Discussion of findings based on results



From the table, we can know that the mental health status of students in university could be reflected by their phone usage. The daily activity time, daily phone charge time and change of daily phone lock time may play an important role in their mental health status. The other phone usage data also contribute to affect student mental health. In conclusion, the phone usage data can be used to analyze the mental health status of the university student. We implement four machine learning methods to analyze the data. We find the linear model, such as SVM and linear regression, performs better in this problem. But, for tree model and K-NN, due to the scale of dataset, the model would be easily overfitting. Moreover, we summary feature selection. And it shows that daily activity time, daily phone charge time and change of daily phone lock time may play an important role in student mental health.

6. Conclusion

From a holistic perspective, what we want to do is to predict students' mental health levels by the data recorded by sensors in the smartphone. While we encounter many problems. The first challenge is raw data processing, there are so many incomplete data in both input dataset and output dataset. We have to handle these problems to get more available samples to train our models later. What we do is extracting information as much as we can. We calculate the daily means to offset the influence of different day numbers in every single file. We also calculate the variance based on the day to measure the fluctuation of daily data. We believe it can represent the characteristics of each student and it can improve our predictions performance. The second challenge is to predict missing post value, we assume that for each student, its pre-score can represent its post-score to some extent. Hence, we use the pre-score as a post-score directly and use machine learning methods to verify it. Fortunately, it works well.

Eventually, we get the final well-processed dataset, what we have to do is select proper learning models and appropriate features to train our models. We implement many feature selection methods and select the best features for each learning model. The next step is tuning parameter to get the best model. We divide this work and every group member is in charge of one model. After that, we discuss and compare the results of our models and get the best model for each dataset. We find for different datasets, there are different learning methods to get the best performance. For a specific machine learning problem, we have to implement different learning methods to find the best one, which is a systematic work.

7. Reference

[1] Headspace (2018). Majority of Aussie students stressed, depressed. At: https://headspace.org.au/blog/majority-of-aussie-students-stressed-depressed/

[2] Lewinson, E. (2019). Outlier Detection with Isolation Forest. [Blog] Towards Data Science. Available at: https://towardsdatascience.com/outlier-detection-with-isolation-forest-3d190448d45e

[3] Krishnan, A (2019) Anomaly Detection with Isolation Forest & Visualization [Blog] Towards Data Science. Available at: https://towardsdatascience.com/anomaly-detection-with-isolation-forest-visualization-23cd75c281e2

[4] Patel, S.(2017) SVM (Support Vector Machine) Theory [Blog] Medium. Available at: https://medium.com/machine-learning-101/chapter-2-svm-support-vector-machine-theory-f0812effc72

[5] Feature Selection Techniques in Machine Learning with Python at: https://towardsdatascience.com/feature-selection-techniques-in-machine-learning-with-python-f24e7da3f36e

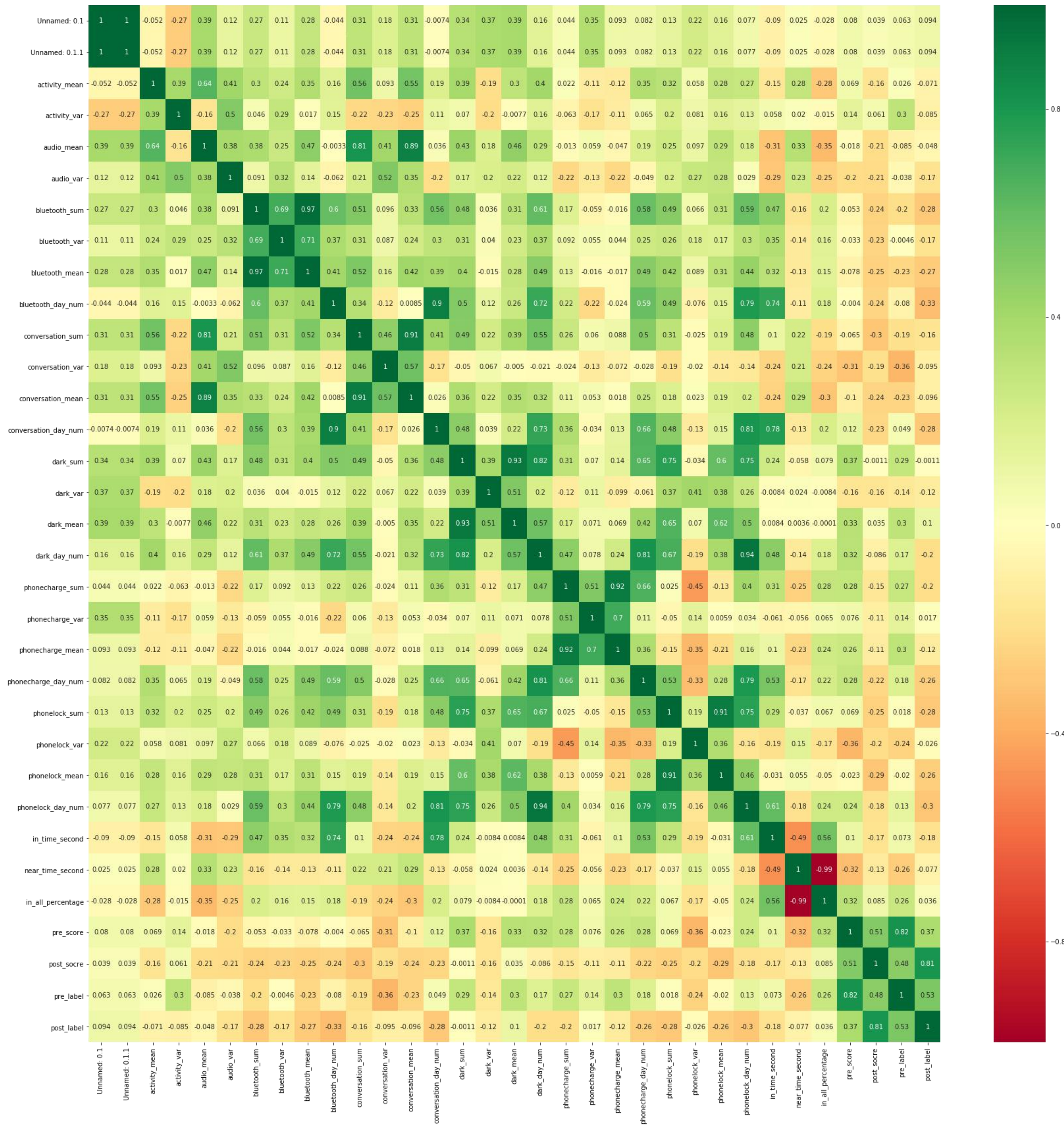
[6] Example of Multiple Linear Regression in Python at: https://datatofish.com/multiple-linear-regression-python/

[7] Selecting good features – Part II: linear models and regularization at: https://blog.datadive.net/selecting-good-features-part-ii-linear-models-and-regularization/

[8] Evaluating a Linear Regression Model at: https://www.ritchieng.com/machine-learning-evaluate-linear-regression-model/#10.-Confidence-in-our-Model

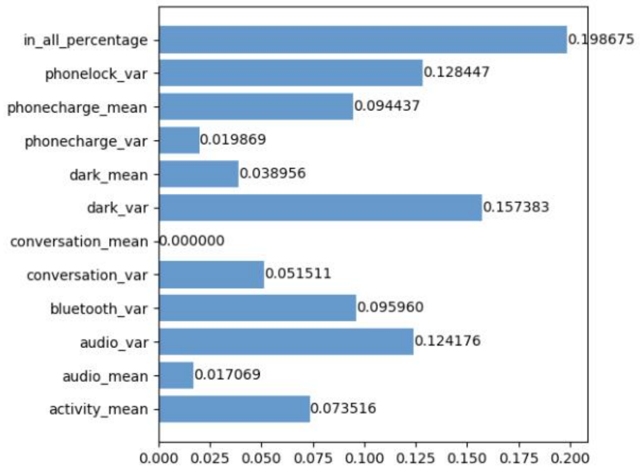
[9] A beginner’s guide to Linear Regression in Python with Scikit-Learn at: https://towardsdatascience.com/a-beginners-guide-to-linear-regression-in-python-with-scikit-learn-83a8f7ae2b4f

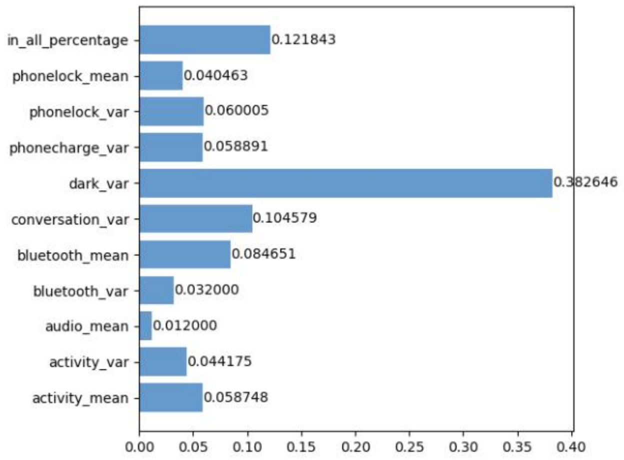
8. Appendix 1

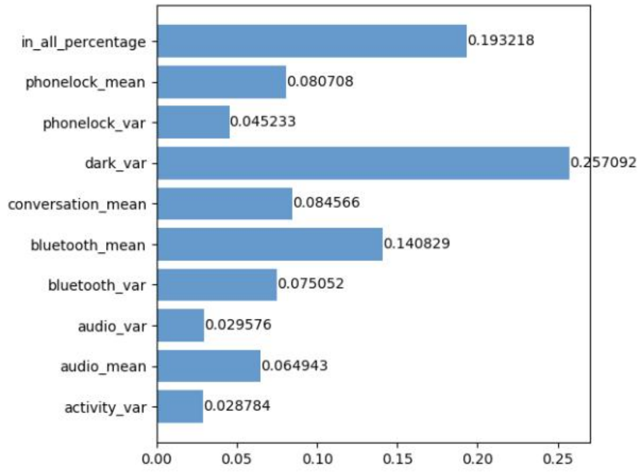


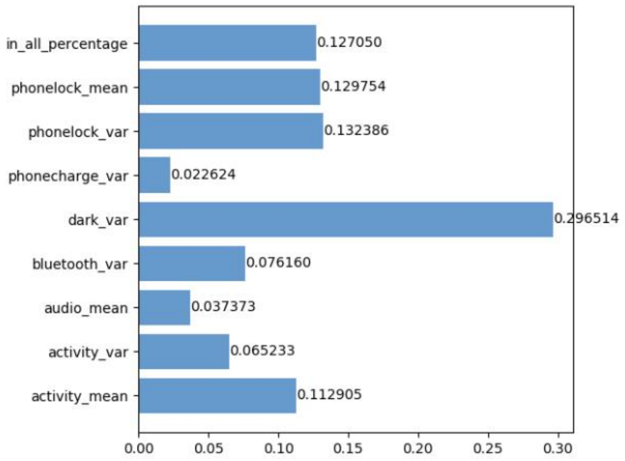
Correlation heatmap for feature selection.

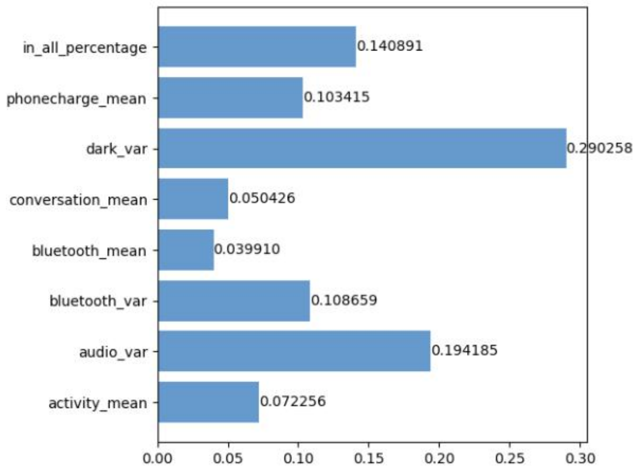
9. Appendix 2

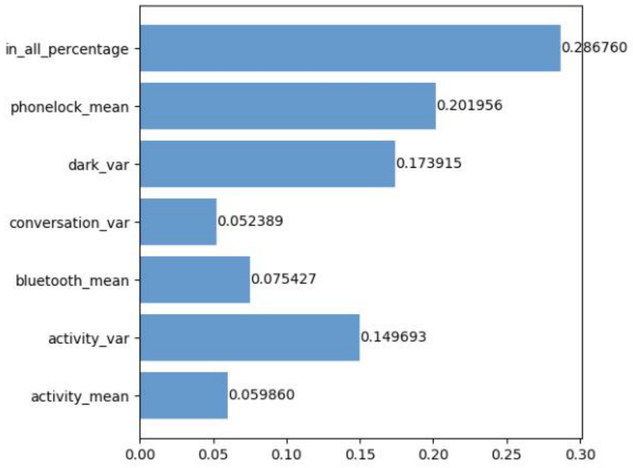

in_all_percentage 0.198675
phonelock_var 0.128447
phonecharge_mean 0.094437
phonecharge_var 0.019869
dark_mean 0.038956
dark_var 0.157383
conversation_mean 0.000000
conversation_var 0.051511
bluetooth_var 0.095960
audio_var 0.124176
audio_mean 0.017069
activity_mean 0.073516
0.000 0.025 0.050 0.075 0.100 0.125 0.150 0.175 0.200


in_all_percentage 0.121843
phonelock_mean 0.040463
phonelock_var 0.060005
phonecharge_var 0.058891
dark_var 0.382646
conversation_var 0.104579
bluetooth_mean 0.084651
bluetooth_var 0.032000
audio_mean 0.012000
activity_var 0.044175
activity_mean 0.058748
0.00 0.05 0.10 0.15 0.20 0.25 0.30 0.35 0.40


in_all_percentage 0.193218
phonelock_mean 0.080708
phonelock_var 0.045233
dark_var 0.257092
conversation_mean 0.084566
bluetooth_mean 0.140829
bluetooth_var 0.075052
audio_var 0.029576
audio_mean 0.064943
activity_var 0.028784
0.00 0.05 0.10 0.15 0.20 0.25


in_all_percentage 0.127050
phonelock_mean 0.129754
phonelock_var 0.132386
phonecharge_var 0.022624
dark_var 0.296514
bluetooth_var 0.076160
audio_mean 0.037373
activity_var 0.065233
activity_mean 0.112905
0.00 0.05 0.10 0.15 0.20 0.25 0.30


in_all_percentage 0.140891
phonecharge_mean 0.103415
dark_var 0.290258
conversation_mean 0.050426
bluetooth_mean 0.039910
bluetooth_var 0.108659
audio_var 0.194185
activity_mean 0.072256
0.00 0.05 0.10 0.15 0.20 0.25 0.30

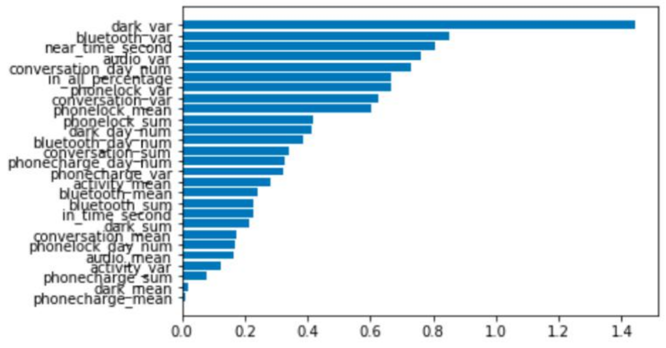

in_all_percentage 0.286760
phonelock_mean 0.201956
dark_var 0.173915
conversation_var 0.052389
bluetooth_mean 0.075427
activity_var 0.149693
activity_mean 0.059860
0.00 0.05 0.10 0.15 0.20 0.25 0.30

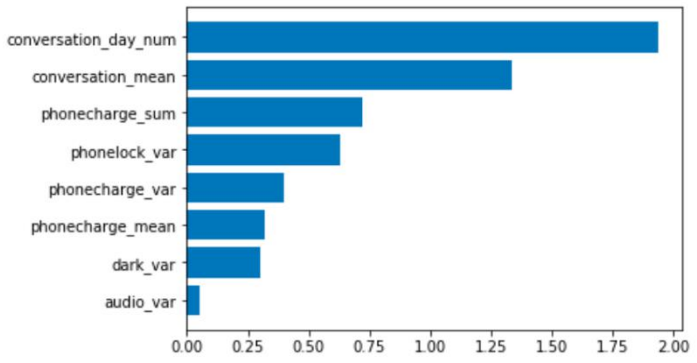
10. Appendix 3

Flourishing post label Feature Selection (default hyperparameters)

	Initial	Individual Best Feature	Final
Features selected	All 27 features	‘dark_var’	'phonelock_var', ‘phonelock_sum','bluetooth_mean', 'bluetooth_sum',‘conversation_day_num’,'conversation_mean', ‘phonelock_mean',‘phonecharge_sum','dark_var'
AUC-ROC	0.47	0.56	0.67
Training Accuracy	0.83	0.77	0.77
Testing Accuracy	0.46	0.39	0.54
Training set Cross validation (4 folds)	[0.67, 0.71, 0.71, 0.43]	[0.67, 0.86, 0.71, 0.71]	[0.78, 0.71, 0.71, 0.86]
Min CV accuracy	0.43	0.67	0.71
F1	0.53	0.20	0.50
Precision	0.67	1.00	1.00
Recall	0.44	0.11	0.33

Flourishing Feature importance(left is initial, right is final)





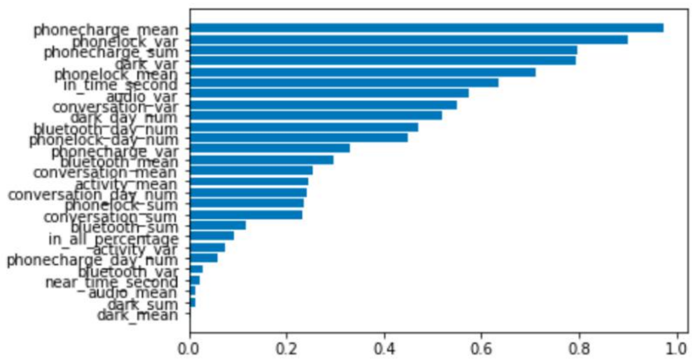
SVM Flourishing Parameter selection

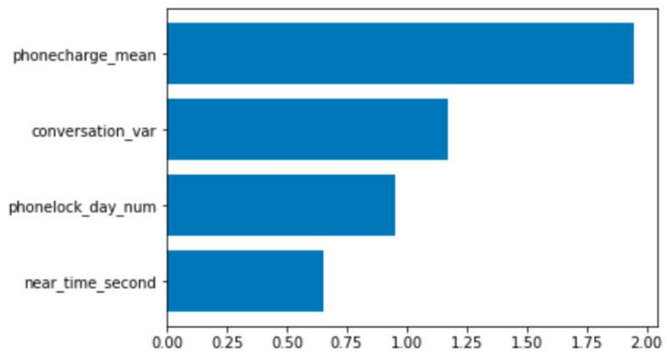
	Kernel	C (penalty parameter)	
	1st Round	1st Round	Last Round
parameters tried	[‘linear’, ‘rbf’]	[0.1, 1, 10, 100, 1000]	[1.0, 1.2, 1.4, 1.6, 1.8, 2.0]
best parameter	linear	1	1, 2 (same)
AUC-ROC	0.67	0.67	0.67
Training Accuracy	0.77	0.77	0.77
Testing Accuracy	0.54	0.54	0.54
Training set Cross validation (4 folds)	[0.78, 0.71, 0.71, 0.86]	[0.78, 0.71, 0.71, 0.86]	[0.78, 0.71, 0.71, 0.86]
Min CV accuracy	0.71	0.71	0.71
F1	0.50	0.50	0.50
Precision	1.00	1.00	1.00
Recall	0.33	0.33	0.33

Panas negative post label Feature Selection (default hyper parameters)

	Initial	Individual Best Feature	Final
Features selected	All 27 features	‘phonecharge_sum’	‘conversation_var','phonecharge_mean',' phonelock_day_num','near_time_second'
AUC-ROC	0.67	0.63	0.88
Training Accuracy	0.8	0.77	0.83
Testing Accuracy	0.64	0.57	0.86
Training set Cross validation (4 folds)	[0.38, 0.63, 0.63, 0.5]	[0.63, 0.63, 0.75, 0.67]	[0.75, 0.75, 0.75, 1.00]
Min CV accuracy	0.38	0.63	0.75
F1	0.62	0.40	0.86
Precision	0.80	1.00	1.00
Recall	0.50	0.25	0.75

PANAS negative initial feature importance(left is initial, right is final)





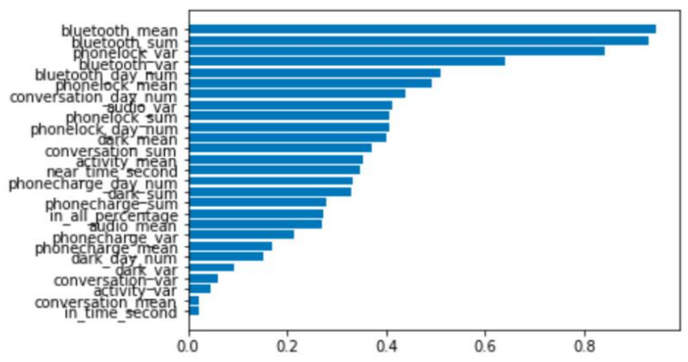
SVM PANAS negative Parameter selection

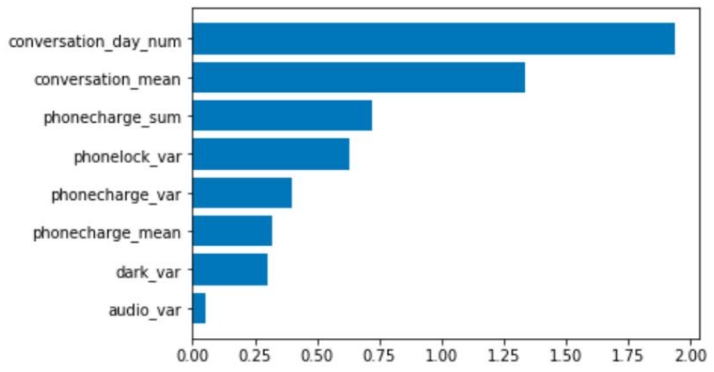
	Kernel	**C (penalty parameter)**	
	1st Round	**1st Round**	**Last Round**
parameters tried	**['linear', 'rbf']**	**[0.1, 1, 10, 100, 1000]**	**[1.0, 1.2, 1.4, 1.6, 1.8, 2.0]**
best parameter	**linear**	**1**	**1, 2 (same)**
AUC-ROC	0.88	0.88	0.88
Training Accuracy	0.83	0.83	0.83
Testing Accuracy	0.86	0.86	0.86
Training set Cross validation (4 folds)	[0.75, 0.75, 0.75, 1.00]	[0.75, 0.75, 0.75, 1.00]	[0.75, 0.75, 0.75, 1.00]
Min CV accuracy	0.75	0.75	0.75
F1	0.86	0.86	0.86
Precision	1.00	1.00	1.00
Recall	0.75	0.75	0.75

Panas positive post label Feature Selection (default hyper parameters)

	Initial	Individual Best Feature	Final
Features selected	All 27 features	'conversation_day_num'	'phonelock_var','conversation_day_num', 'phonecharge_var','conversation_mean','dark_var','phonecharge_sum','audio_var','phonecharge_mean'
AUC-ROC	0.50	0.58	0.79
Training Accuracy	0.73	0.73	0.67
Testing Accuracy	0.50	0.57	0.78
Training set Cross validation (4 folds)	[0.75, 0.75, 0.5, 0.83]	[0.88, 0.75, 0.5, 0.67]	[0.63, 0.75, 0.63, 0.67]
Min CV accuracy	0.50	0.50	0.63
F1	0.53	0.57	0.80
Precision	0.57	0.67	0.86
Recall	0.50	0.50	0.75

PANAS positive feature importance (left is initial, right is final)



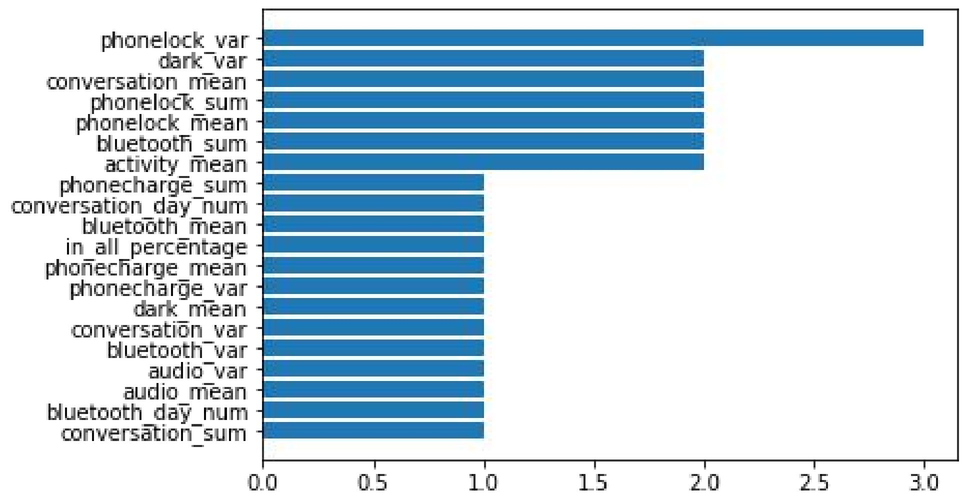


SVM PANAS positive Parameter selection

	Kernel	C (penalty parameter)	
	1st Round	**1st Round**	**Last Round**
parameters tried	**['linear', 'rbf']**	**[0.1, 1, 10, 100, 1000]**	**[1.0, 1.2, 1.4, 1.6, 1.8, 2.0]**
best parameter	**linear**	**1**	**1, 2 (same)**
AUC-ROC	0.79	0.79	0.79
Training Accuracy	0.67	0.67	0.67
Testing Accuracy	0.78	0.78	0.78
Training set Cross validation (4 folds)	[0.63, 0.75, 0.63, 0.67]	[0.63, 0.75, 0.63, 0.67]	[0.63, 0.75, 0.63, 0.67]
Min CV accuracy	0.63	0.63	0.63
F1	0.80	0.80	0.80
Precision	0.86	0.86	0.86
Recall	0.75	0.75	0.75

11. Appendix 4

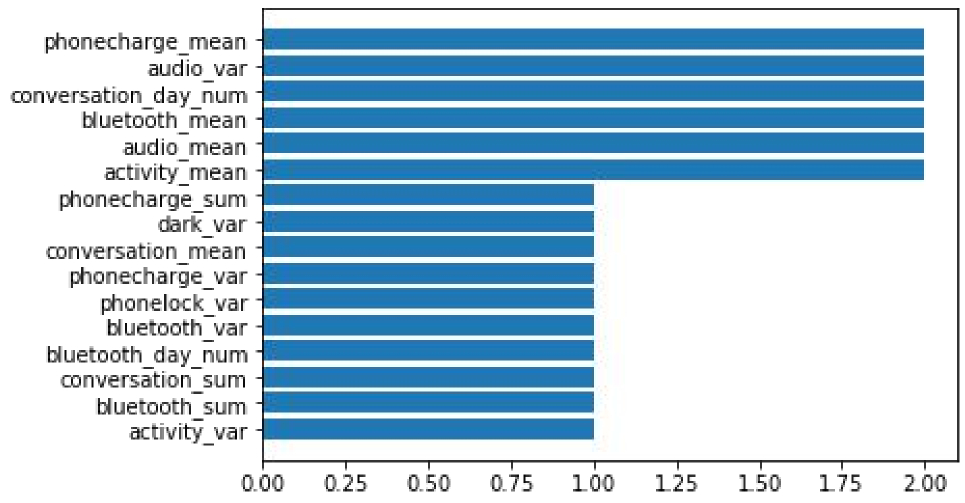
Flourishing feature selection frequency summary



Flourishing feature selected by different models

methods	number of features	features
knn	3	['activity_mean', 'bluetooth_sum', 'phonelock_var']
linear regression	4	['phonelock_mean', 'phonelock_sum', 'conversation_sum', 'bluetooth_day_num']
random forest	12	['activity_mean', 'audio_mean', 'audio_var', 'bluetooth_var', 'conversation_var', 'conversation_mean', dark_var', 'dark_mean', 'phonecharge_var', 'phonecharge_mean', 'phonelock_var', 'in_all_percentage']
Svm	9	['phonelock_var','phonelock_sum','bluetooth_mean', 'bluetooth_sum', 'conversation_day_num','conversation_mean', 'phonelock_mean', 'phonecharge_sum', 'dark_var']

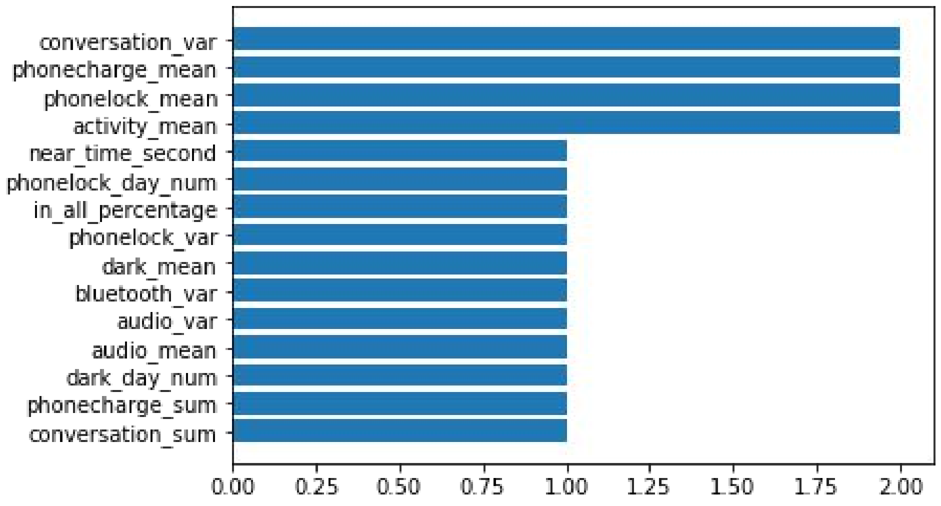
Panas positive feature selection frequency summary



Panas positive feature selected by different models

methods	number of features	features
knn	3	['activity_mean', 'activity_var', 'audio_mean']
linear regression	5	['bluetooth_sum', 'bluetooth_mean', 'conversation_sum', 'conversation_day_num', 'bluetooth_day_num']
random forest	6	['activity_mean', 'audio_mean', 'audio_var', 'bluetooth_var', 'bluetooth_mean', 'phonecharge_mean']
Svm	8	['phonelock_var', 'conversation_day_num', 'phonecharge_var', 'conversation_mean', 'dark_var', 'phonecharge_sum', 'audio_var', 'phonecharge_mean']

Panas negative feature selection frequency summary



Panas negative feature selected by different models

methods	number of features	features
knn	3	['activity_mean', 'conversation_sum', 'phonelock_mean']
linear regression	3	['phonecharge_sum', 'phonecharge_mean', 'dark_day_num']
random forest	9	['activity_mean', 'audio_mean', 'audio_var', 'bluetooth_var', 'conversation_var', 'dark_mean', 'phonelock_var', 'phonelock_mean', 'in_all_percentage']
Svm	4	['conversation_var', 'phonecharge_mean', 'phonelock_day_num', 'near_time_second']

12. Appendix 5

Missing post prediction model compare

For predicting the missing value in the post dataset, we implement four machine learning algorithms, including K-NN, linear regression, SVM and random forest. After then, accuracy would be used as evaluation metric for selecting the predicted result.

For missing value in flourishing post:

model	Min cross validation accuracy	evaluation accuracy	Predicted result
K-NN	0.625	0.8	[1, 1, 0, 1, 1, 1, 1, 1]
Random forest	0.625	0.7	[1, 1, 1, 1, 0, 1, 0, 1]
SVM	0.75	1	[0, 1, 1, 0, 1, 1, 1, 1]
Linear regression		0.3	[1, 1, 1, 1, 0, 1, 1, 1]

The table shows that SVN preform high in evaluation accuracy and min cross validation accuracy Thus, the post flourishing label of the student u08, u12, u13, u18, u22, u50, u57, u58 will be labeled as0, 1, 1, 0, 1, 1, 1, 1 in following part of this project.

For missing value in panas positive post:

model	Min cross validation accuracy	evaluation accuracy	Predicted result
K-NN	0.75	0.91	[0 1 0 0 1 1 0]
Random forest	0.625	0.73	[1 0 0 1 1 1 0]
SVM	0.75	0.73	[1, 1, 1, 0, 1, 1, 1]
Linear regression		0.82	[1, 0, 1, 0, 1, 1, 1]

The table shows that K-NN preform the highest in evaluation and highest min cross validation accuracy of those four models. Thus, the panas positive post of the student u08, u12, u13, u22, u50, u57, u58 will be labeled as 1,1,0,1,1,1,0.

For missing value in panas negative post:

model	Min cross validation accuracy	evaluation accuracy	Predicted result
K-NN	0.875	0.82	[1 1 1 1 1 1 1]
Random forest	0.875	0.82	[1 1 1 1 1 1 1]
SVM	0.875	0.82	[1, 1, 1, 1, 1, 1, 1]
Linear regression		0.55	[1, 1, 1, 1, 1, 1, 1]

The table shows that the evaluation accuracy and min cross validation accuracy of K-NN, random forest and SVM are same. And the four model also predict the same result, so the panas positive post of the student u08, u12, u13, u22, u50, u57, u58 will be labeled as 1(high).