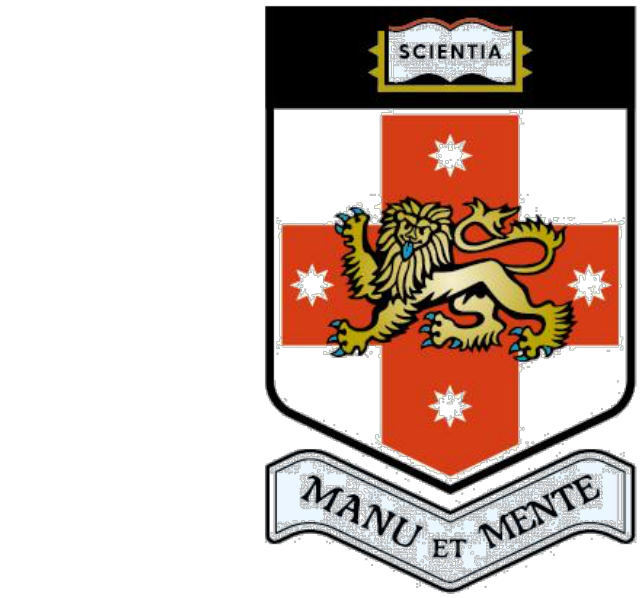

UNSW
SYDNEY

COMP9417 GROUP PROJECT - STUDENTLIFE

Group Name: 5 Nearest Neighbors

Members: Haotian Chu, Daifei Zhang, Jianlong Sun, Patrick Li and Nicholas Quinn

1.Introduction

In this project, we were tasked with predicting multiple measures of student's mental health from sensing data automatically generated from their phones. To predict these scores, we had to examine the dataset available to extract and engineer features, and then train and tune multiple different machine learning models. The aim was to find which features from the complete dataset correlate most with mental well-being, then train and tune models with these features, and then accurately classify/predict scores based on this data. The methods we used for identifying and extracting valuable features, processing them, and tuning (hyperparameters) and training models to generate predictions is covered in each of the relevant sections below.

2.Dataset

StudentLife_Dataset, consists of both the sensing inputs and the mental health score outputs.
The inputs were collected over a ten week period from 48 Dartmouth student's phones via automatic sensing. For each of the categories of input data, we chose to only use some of the data collected for them and thus we will omit the unused columns in describing the dataset. The categories of input data include: **activity** - a timestamped inferred movement type which distinguishes stationary (0) from walking (1) and running (2), **audio** - a timestamped inferred sound type which distinguishes silence (0) from voice (1) and general noise (2), **bluetooth** - a timestamped signal strength to the nearby bluetooth device, **conversation** - start-finish time periods in which conversation was taking place, **dark** - start-finish time periods in which the student phone was in a dark environment, **gps** - time stamped speed and location data about the student's phone, **phone charge** - start-finish time periods in which the student's phone was charging, **phone lock** - start-finish time periods in which the student's phone was locked, **wifi** - a timestamped signal strength to the nearby wifi access point, and **wifi location** - timestamped entries indicating the location of the connected to wifi access point.
The outputs are scores generated from surveys taken by the students, both prior and after the sensing period. There are three distinct scores, namely positive and negative affect schedule (+PANAS and -PANAS) and Flourishing scale (FS), and hence there were three classes to be created (binarization method below).
Flourishing Scale (one class): In the FS survey, students had to answer 8 questions with a number from 1 to 7, with each number reflecting a level of agreement. Higher FS scores reflect a higher quality of self-perceived success and thus greater mental health. The FS of a student is obtained by simply summing the score of each question. In the case of **missing answers, we generated an answer using the mean value** of all other students' scores of the same question. Furthermore, we chose to **use the mean value of the pre and post score** when they were present, and only the present one otherwise. After getting the FS score for each student, we calculated the sum of all scores so that we could use the **median as a threshold for classification**. If a student's FS score is greater than the threshold, it will be classified as 1, otherwise it will be classified as 0. Thus, a class label of 0 or 1 represents the FS feature for each student.
PANAS (two classes): The PANAS survey is used to generate two scores, +PANAS score and -PANAS score. The "PANAS_Scoring.pdf" given in "StudentLife_Dataset" details how to calculate both positive and negative PANAS scores, which involves dividing the questions up into two subsets and summing their scores separately. Identical to the FS scores, If **missing values** occurred we **replaced them by the mean value** of all other students responses, and **if a student has both pre and post scores, the mean value of the two will be taken** as the student's final score. It should be noted, however, that the "excited" and "ashamed" items are missing, which means that the mean score data given in this file won't be helpful (at least we cannot be sure it is). Hence, we add up all items except for these two and find the **median number for** positive **PANAS scores** and negative

PANAS scores of all students, which is **taken as a threshold**. If a student's score is greater than the threshold, a label of 1 will be given, otherwise a label of 0 will be given. Ultimately, for each student, a classified positive PANAS score and a classified negative PANAS score will be extracted as PANAS score features.

We **justify using the mean to replace values** are they appear to be **missing at random** as opposed to a specific type of answer being manually removed, and thus doesn't introduce bias. Furthermore, the **amount of students in the study is already low** and so we **do not want to use methods that discard data**. Additionally, we **justify using the mean of pre and post scores** to obtain a **more accurate scoring of the student's mental health over time** (to try to reduce outliers that result from only having one sample). We believe this is valid as the majority of the data collection was done automatically and so is not likely to have affected the students post scores (minimal observer effect[1]) and thus both pre and post are valid scores.

We also used two extra datasets from the complete dataset that were not part of the sensing data. Both of these extra datasets were EMA survey results, namely sleep and social. The sleep survey obtains information about sleep duration, quality and tiredness during the day, whilst the social survey obtains information about any type of conversational interaction via any medium.

The features that we generated from the base dataset, plus these extra datasets, is explained below.

3.Methods

3.1 Feature generation

The datasets chosen to be pre-processed were all the datasets in the "Inputs" folder in "StudentLife_Dataset", plus the "Sleep" and the "Social" dataset under the directory "dataset/EMA/response" which was obtained from the full dataset. Firstly, we extracted all the features we thought may be helpful. We then used some feature selection methods to determine which features to use in the models we developed.

The following paragraphs illustrate what features were extracted or engineered from which dataset, and how this was accomplished.

Activity: According to Table 7 of the paper given in the assignment specification, Rui Wang et al. (2014)[2] found that there are correlations between activity duration, activity duration for day, activity duration for evening and the loneliness scale. **Activity duration** for each student is extracted by calculating and adding up the period of time where the label of "activity inference" is not 0, meaning the student was not stationary. We consider **activity duration for day** to include activities between 6:00 and 18:00, and compute activity duration for day in the same way as activity duration. **Activity duration for evening** is drawn by subtracting activity duration for day from total activity duration. Besides these features, the ratio of being stationary, walking activity and running activity are extracted as well. We believe physical exercise ratio, regardless in the form of walking or running, would influence or indicate student's mental health state to some extent. These three features, **stationary ratio, walking ratio and running ratio**, are extracted by computing the ratio of the number of data samples where the label is 1 or 2 and the number of all data samples.

Audio: Socializing is known to be a significant factor in maintaining good mental health and reduces stress levels. We believe the audio feature is a good indication of an individual's level of socializing, specifically the ratios of a person being in a conversation environment or noisy environment. Using the inference labels of silence, voice and noise mentioned earlier, the ratio of each label occurring in the column is computed. Note that we intentionally ignore label 3, as we believe the unknown audio is too broad to be considered a single feature or useful for our purposes. From this dataset, the **ratio of being in a silent environment**, the **ratio of being in a conversation environment** and the **ratio of being in a noisy environment** are all obtained.

Conversation: Rui Wang et al. (2014) found some connections between **conversation duration**, **conversation duration during day**, **conversation duration during evening**, **conversation frequency**, **conversation frequency during day**, **conversation frequency during evening** and perceived stress scale (PSS). From the "conversation" dataset, conversation durations are computed by adding up all durations, and those between 6:00 and 18:00 are defined as conversations during the day, the rest are during the evening. Frequencies are calculated by counting the number of relevant data. On top of these 6 features, the numbers of conversations of different length of time were also extracted. Conversations under 5 minutes are defined as short conversations, from 5 minutes to 30 minutes are defined as medium conversations, and those above 30 minutes are defined as long conversations.

Dark: The "Dark" dataset measures the amount of time the student's phone is located in a dark environment for over one hour continuously. Timestamps between 6:00 and 23:00 are considered as daily activity time, and those between 23:00 and 6:00 are daily sleeping time. Naturally, the duration of dark periods between these times are summed to get the following features: the **total duration of time when the phone was in a dark environment**, **duration during daily activity time**, **duration during daily sleeping time**, **total frequency of the phone being at dark environment**, **frequency during daily activity time**, and **frequency during daily sleeping time**.

Phone_lock: The data in the "phone_lock" dataset can roughly reflect a student's amount of usage of their mobile phone, which could be an important indicator of mental health, since nowadays the mobile phone is the most important tool for everyone to conduct social activities. The "phone_lock" dataset is pre-processed the same as the "dark" dataset, by which 6 features are extracted: **total duration when the phone was locked**, **duration during daily activity time**, **duration during daily sleeping time**, **total frequency of phone being locked**, **frequency during daily activity time**, and **frequency during daily sleeping time**.

Phone_charge: For the "phone_charge" dataset, only **the frequency of the phone being charged** is calculated. This is done by counting the number of data in the data file for each student.

Wifi_location: Since the data in the "wifi_location" dataset is in the format of time and location the student has been to, for each student, **number of places the student has been to** is calculated. We believe the number of places a student has been to in the whole program may have some effects on student's mental health or may indicate student's mental health.

Wifi: The "wifi" dataset shows physical address (BSSID), frequency and level of wifi signal strength. Due to features about location are already extracted from the "wifi_location" dataset, we focus on the strength of wifi signal part which we think could influence student's emotion to some extent. First, the median number of the "level" column of all students is taken as a threshold. Then, the **ratio of level data below the threshold** among all data for each student is calculated.

Bluetooth: The "bluetooth" dataset is processed as the "wifi" dataset mentioned above, a **ratio of signal level below the median number of all signal level data** is computed for each student.

GPS: Traveled distance of a student correlates to the student's loneliness scale as shown in Table 7 in findings of Rui Wang et al. (2014). **Traveled distance** for each student is computed by adding up all distances the student has traveled, which could be drawn from multiplying the data in the "speed" column by the period of time. On top of this, **traveled distance during day** is also computed based on the timestamp in the dataset. Besides traveled distance, we consider the data in "network_type" column would provide some helpful information about student's physical location. Thus, ratios for cell type, wifi type and empty which implies GPS type in "provider" column are calculated.

Sleep: Sleep is arguably one of the, if not most, important determiners of stress levels and general well being. From the paper of Rui Wang et al. (2014), we found that they implemented a sleep model combining the light

features, phone usage features, activity features and sound features to form a single sleep duration feature, yet we couldn't find the outputs of said model nor adequate information about how to reconstruct it. However, upon downloading the full dataset, there appeared to be a second method used for documenting sleep, which was daily sleep surveys, and this appears to be the ground truth that their sleep model was evaluated against, so we instead opted to use these. The survey answers contain information about the student's sleep duration, sleep rate (quality of sleep) and level of tiredness felt the next day. In order to deal with missing values existing in each student's file, we add up all data for each of the features and calculate a mean value as the final result. Ultimately, **mean sleep duration**, **mean sleep rate** and **mean level of tiredness** were extracted as our features.

Social: In the paper, a correlation between number of co-locations with other people and flourishing scale was discovered, yet we didn't get the method they computed this feature. Instead, we used the answers for the question "How many people did you have contact with yesterday, including anyone you said hello to, chatted, talked or discussed matters with, whether you did it face-to-face, by telephone, by mail or on the internet, and whether you personally knew the person or not ?". The survey answers are given in the form of labels, which represent different number of people contacted. In the same way we extracted mean sleep duration, we computed the mean **number of people a student contacted every day** to handle missing values.

Above are all the features we extracted from the given dataset, which includes features found helpful in the paper of Rui Wang et al. (2014) and some other features we believe might be useful, as shown in Table 1 and Table 2 below, respectively. We made three changes on the original features from Rui Wang et al. (2014). One is that we took mean sleep duration for each student instead of total duration as discussed above, again to deal with missing values/varying sample sizes. The second is that we took the ratio of the network type being wi-fi as the indicator of indoor mobility, which could imply the ratio of time a student spent inside buildings, instead of distance traveled inside buildings. The last one is that we used the mean number of people a student contacted every day instead of number of co-locations with other people.

No.	Feature	No.	Feature
1	Activity duration	2	Activity duration for day
3	Activity duration for evening	4	Conversation duration during day
5	Conversation duration during day	6	Conversation duration during evening
7	Conversation frequency	8	Conversation frequency during day
9	Conversation frequency during evening	10	Traveled distance
11	Traveled distance for day	12	Ratio of the network type being wi-fi
13	Mean sleep duration	14	Mean number of people contacted

Table 1 The original features from Rui Wang et al. (2014)

No.	Feature	No.	Feature
1	Activity duration	2	Activity duration for day
3	Activity duration for evening	4	Ratio of being stationary

5	Ratio of walking	6	Ratio of running
7	Conversation duration	8	Conversation duration during day
9	Conversation duration during evening	10	Ratio of being in a silent environment
11	Ratio of being in a conversation environment	12	Ratio of being in a noisy environment
13	Conversation frequency	14	Conversation frequency during day
15	Conversation frequency during evening	16	Number of short conversations
17	Number of medium conversations	18	Number of long conversations
19	Total duration when the phone was in a dark environment	20	Duration when the phone was in a dark environment during daily activity time
21	Duration when the phone was in a dark environment during daily sleeping time	22	Total frequency of the phone being in a dark environment
23	Frequency of the phone being in a dark environment during daily activity time	24	Duration of the phone being in a dark environment during daily sleeping time
25	Total duration when the phone was locked	26	Duration when the phone was locked during daily activity time
27	Duration when the phone was locked during daily sleeping time	28	Total frequency of the phone being locked
29	Frequency of the phone being locked during daily activity time	30	Frequency of the phone being locked during daily sleeping time
31	Total frequency of the phone being charged	32	Traveled distance
33	Traveled distance for day	34	Ratio of the network type being cell
35	Ratio of the network type being wi-fi	36	Ratio of the provider being GAS
37	Mean sleep duration	38	Mean sleep rate
39	Mean level of tiredness	40	Ratio of the wi-fi signal level below the threshold
41	Ratio of the bluetooth signal level below the threshold	42	Number of places a student has been to
43	Mean number of people contacted		

Table 2 All the features extracted

3.2 Feature Selection

We started feature selection from the two feature sets above, one contains the original features from the paper, and another is the feature set including the original features and some extra features we believe might be helpful.

First, we used the original features as the training set for all models we chose. There are three classes we need to predict as discussed in the Dataset section above, which are a class for flourishing score, a class for positive PANAS score and a class for negative PANAS score. Since we obtained a prediction for one class at one time, we ran every model three times to get all the predictions for all three classes.

In the second step, we repeated what we did in the previous step, except that we took all the features including the original ones and those we believe might be helpful as the input dataset.

Then, we compared the results obtained from the same models of the two different feature sets.

Once knowing which feature set made the models perform better, we generated three different selected feature sets by three different feature selection method from it, which are variance threshold, gradient boosting decision tree (GBDT) and chi-square test. In the variance threshold feature selection method, we eliminated those features having low variance. GBDT could choose features based on their importance, and implement interactions between multiple features which shows GBDT has a non-linear feature. As for chi-square test, it would discard features which are independent of the target value we are going to predict.

The whole progress of feature selection is shown in Figure 1.

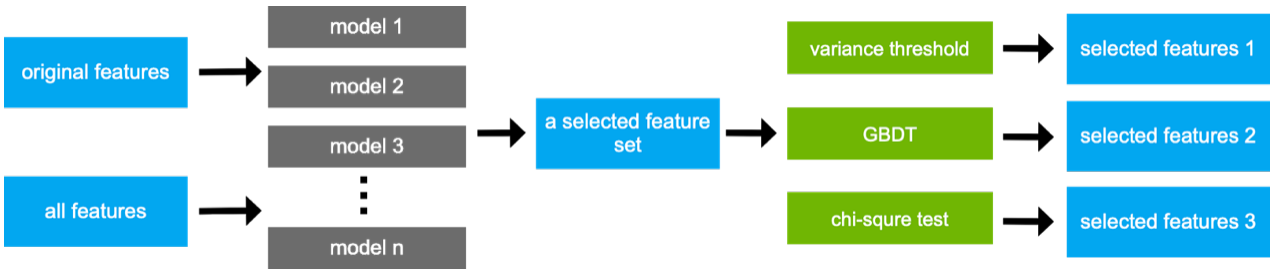

Figure 1 Feature selection

3.3 Feature Normalization

We used Sklearn's MinMaxScaler to normalize features.

3.4 Evaluation

We picked accuracy score and RMSE to evaluate our models. Accuracy score could tell how accurate our predictions are, and RMSE could indicate the absolute fit of a model to the input features.

In each model, we predicted all of the three classes (-PANAS, +PANAS and flourishing score), and computed the accuracy score and RMSE for each of them. To calculate accuracy scores, we used k-fold cross-validation due to lack of adequate amount of data, and to compute RMSE we just imported the function from Sklearn[3].

3.5 Logistic Regression

Explanation

Logistic regression is a predictive method often used for classification problems, and it's based on the concept of probability. Thus, we believe it would be suitable for our task.

We implemented the Logistic Regression algorithm by using the tools in Sklearn library (from sklearn.linear_model import LogisticRegression)[4].

Hyper-Parameters

We used GridSearchCV alongside doing k-fold for tuning the hyperparameters to tune hyper-parameters of 'penalty', 'C' (regularization coefficient) and 'solver' in our logistic regression model.

The value of 'penalty' could be 'L1' or 'L2'. These two methods both support the 'liblinear' of 'solver' hyper-parameter, yet ''newton-cg', 'sag' and 'lbfgs' are only supported by 'L2'.

For 'C' (regularization coefficient), the smaller it is, the stronger the regularization is. We chose the value of 'C' from a list of numbers: [0.01, 0.05, 0.1, 1, 5, 10, 50, 100].

3.6 Support Vector Machine

Explanation

The Support Vector Machine model is a classic way to deal with label classification problems. It is a discriminative model as a line is generated from labelled samples to be able to classify new samples, depending on which side of the line they fall on (at least for 2D space). We implemented the SVM algorithm by using the tools in the Sklearn library as well (from sklearn.svm import SVC)[5].

Hyper-Parameters

We used GridSearchCV alongside doing k-fold for tuning the hyperparameters as well. There are four hyper-parameters to be tuned, 'C' (regularization coefficient),'degree', 'kernel' and 'gamma'.

With larger 'C' (regularization coefficient), there will be less tolerance for errors.

The larger the value of 'gamma' is, the fewer the support vectors there will be. The number of support vectors could influence the training and predicting speed. We tested the value of 'gamma' from a list of numbers: [1, 0.1, 0.01, 0.001].

The 'kernel' hyper-parameter indicates the core function. There are three core functions shown as below:

1. Linear:

$$K(x_i, x_j) = x_i^T x_j$$

2. Poly:

$$K(x_i, x_j) = (\gamma x_i^T x_j + r)^d, d > 1$$

3. Rbf

$$K(x_i, x_j) = exp(-\gamma ||x_i - x_j||^2), \gamma > 0$$

We used different types of 'kernel' to iterate the value of 'C'. Like 'linear' part , 'C' value is [1, 10, 100, 1000], and 'poly' part, 'C' value is [1, 10],with 'degree' is [2, 3]. The last type 'rbf' has the same value of C with 'linear' part.

3.7 K Neighbors Classifier

Explanation

KNN model is one of the most well-known models in machine learning.

After normalizing the features, we used cross-validation to calculate the accuracy of our KNN model, and grid research to find the best K-neighbors classifier to calculate the accuracy of this model by setting different parameters of KNN.

Hyper-parameters

Three parameters of this KNN model need to be tuned.
The ‘weights’ shows if the distance has a weight. It could be ‘uniform’ or ‘distance’. The ‘uniform’ means the distance is not weighted, and the ‘distance’ means when it was calculated, we need to consider the different weights of distances. The ‘p’ shows different types of distance, but it only appear when the ‘weights’ is ‘distance’[6].
We found the best value of n_neighbors by using an iteration from 1 to 11, and when ‘weights’ is ‘distance’, we iterated ‘p’ from 1 to 6.

3.7 Decision Tree Classifier

Explanation

Decision tree is a widely used model in machine learning, and we reckon it would be a suitable model for our task. We implemented our decision tree model by Sklearn’s library as well[7].

Hyper-parameters

In decision tree classifier, we tuned ‘max_depth’ and ‘criterion’. ‘Max_depth’ indicates the max depth of a decision tree, and it was set to be a range from 1 to 21. ‘Criterion’ indicates which type of evaluation method we use during the tuning part, and it includes ‘entropy’ and ‘gini’. We used grid search[8]. and cross-validation to extract the best parameters for our decision tree classifier, and we took the mean value of the cross-validation outputs as the result.

4. Results

4.1 Feature Selection

As shown in the charts below, the results of prediction accuracy from all the features including features from paper and some other features we think might be helpful are better than the results from original features from the paper. Thus, we chose the feature set containing all the features for further feature selection.
The next step is to implement three feature selection methods to find out the most suitable features for each model, the result is shown in the four charts below.

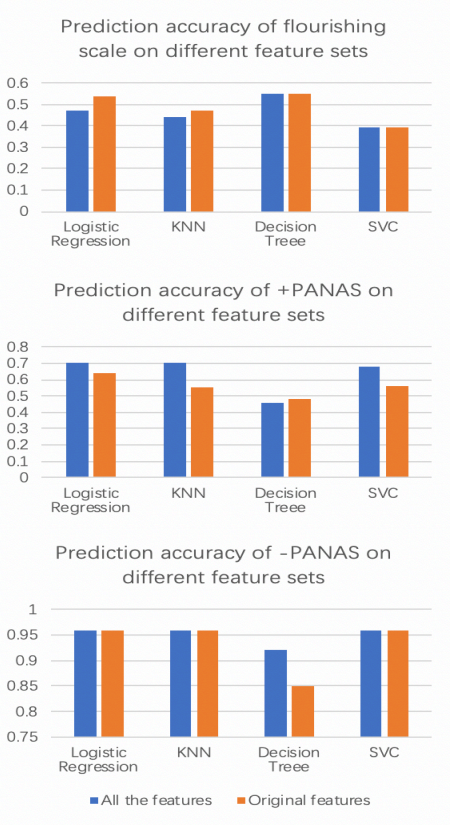

Figure 2 Accuracy score for each feature set

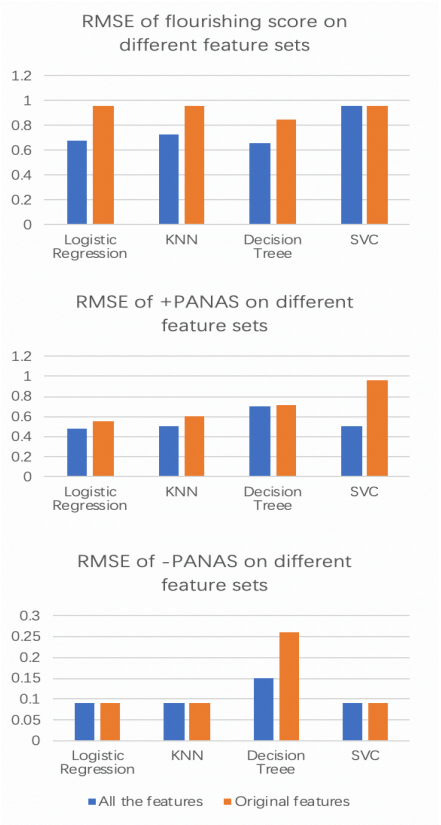

Figure 3 RMSE for each feature set

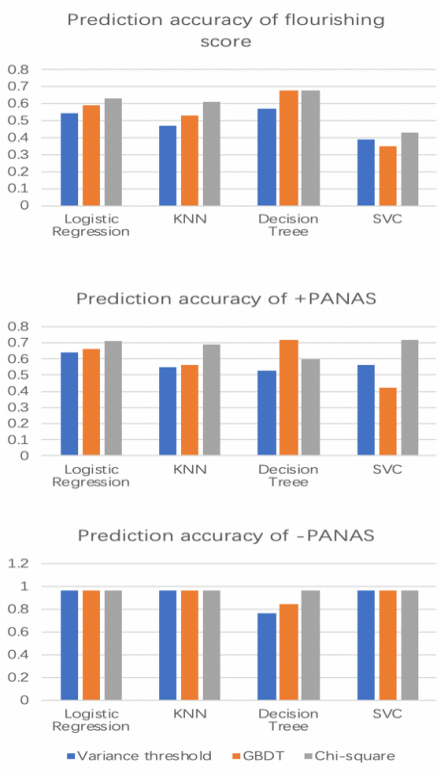

Figure 4 Accuracy score for each feature selection method

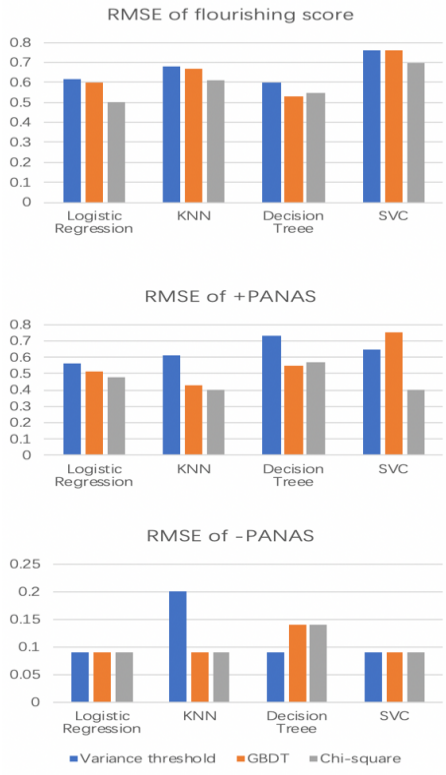

Figure 5 RMSE for each feature selection method

It can be concluded from the above figures that, for logistic regression, K neighbors classifier, and SVM, it's better to use chi-square test to select features , and for decision tree, it's better to use gradient boosting decision tree (GBDT) to select features.

4.2 Hyper-parameter Tuning

Support Vector Machine

Table 3 and Figure 6 below show the result of hyper-parameter tuning for SVM.

	-PANAS	+PANAS	FS
C	1	10	10
Degree	3	2	2
Kernel	linear	poly	poly
Gamma	auto-deprecated	auto-deprecated	auto-deprecated

Table 3 Hyper-parameter tuning for SVM

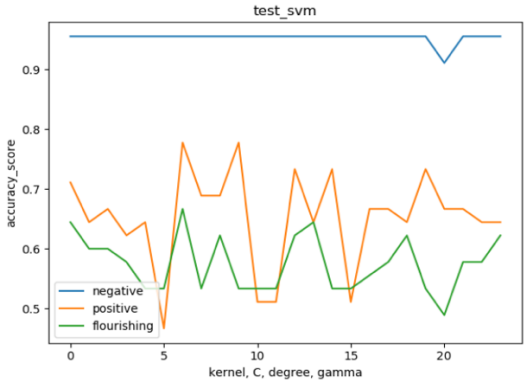

Figure 6 Accuracy score

Logistic Regression

Table 4 and Figure 7 below show the result of hyper-parameter tuning for logistic regression.

	-PANAS	+PANAS	FS
Penalty	l2	l2	l2
C	0.01	1	0.1
solver	liblinear	liblinear	liblinear

Table 4 Hyper-tuning for logistic regression

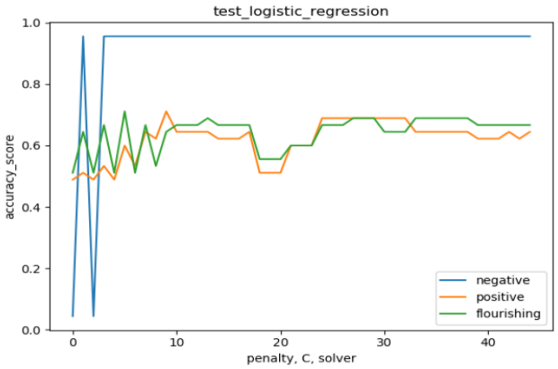

Figure 7 Accuracy score

K Neighbors Classifier

Table 5 and Figure 8 below show the result of hyper-parameter tuning for K neighbors classifier.

	-PANAS	+PANAS	FS
n_neighbors	3	3	10
weights	uniform	distance	uniform
p	/	1	/

Table 5 Hyper-parameter tuning for KNN

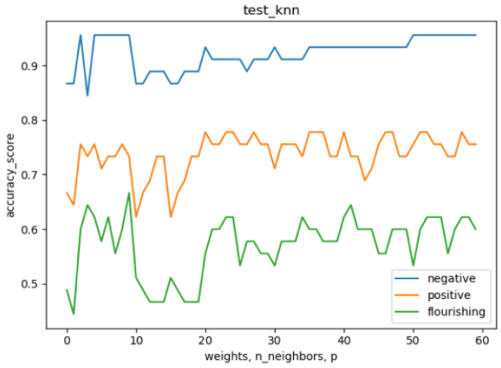

Figure 8 Accuracy score

Decision Tree Classifier

For the decision tree, we got different trees for different target values to predict (-PANAS score, +PANAS score and flourishing score). For each target value, there are two trees demonstrated below, from Figure 9 to Figure 11. The one on the left is the tree of the final result, and the one on the right is the tree pruned from the tree on the left. The result of hyper-parameter tuning is shown in Table 6.

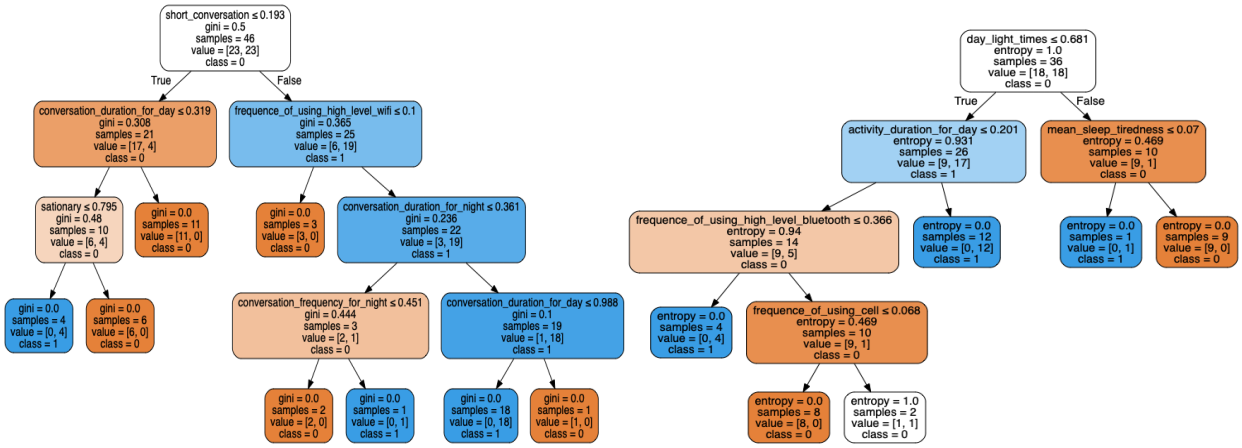

Figure 9 Decision tree for flourishing tree

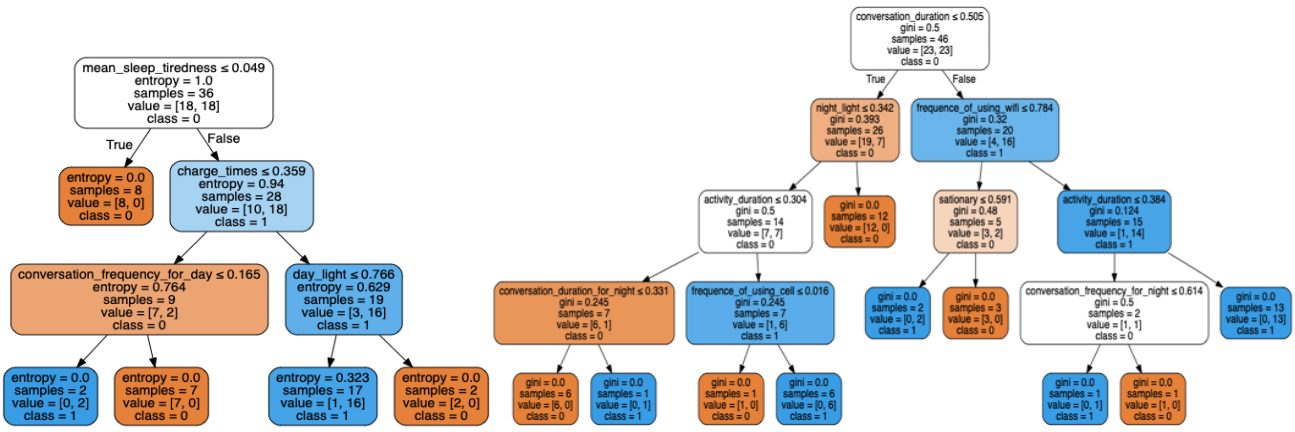

Figure 10 Decision tree for -PANAS

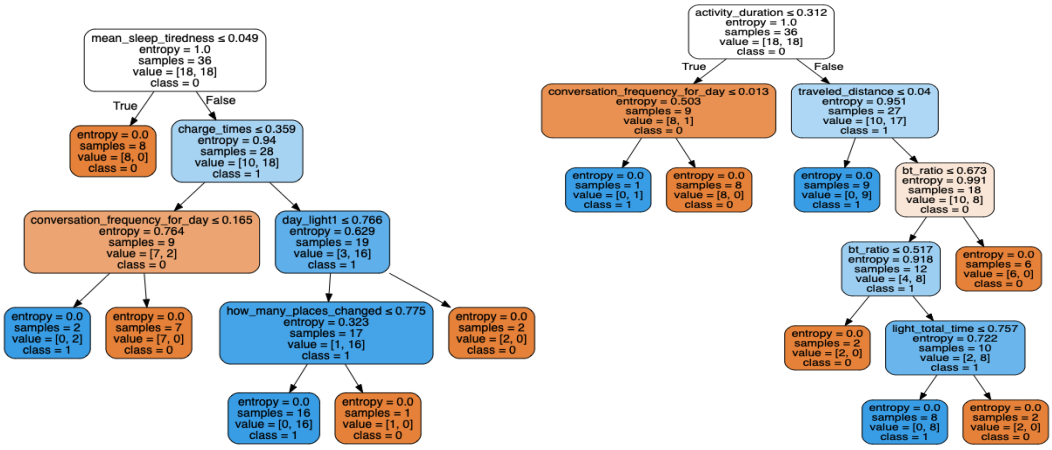

Figure 11 Decision tree for +PANAS

		-PANAS	+PANAS	FS
The tree on the left	max_depth	4	4	4
	criterion	entropy	entropy	gini
The tree on the right	max_depth	4	5	4
	criterion	gini	entropy	entropy

Table 6 Hyper-parameter tuning for decision tree

4.3 Evaluation

After tuning hyper-parameters, we used the feature set generated by feature selection part to get the final results which are shown in Table 7 below.

		-PANAS	+PANAS	FS
Accuracy	LogisticRegression	0.96	0.72	0.72
	SVM	0.96	0.78	0.67
	KNeighborsClassifier	0.96	0.78	0.66
	DecisionTreeClassifier	0.56	0.61	0.55
RMSE	LogisticRegression	0.21	0.54	0.54
	SVM	0.21	0.48	0.58
	KNeighborsClassifier	0.21	0.47	0.58
	DecisionTreeClassifier	0.53	0.62	0.54

Table 7 Evaluation results

5. Discussion

5.1 Comparison based on features

The different models select different features as most important for determining the students mental health statuses, i.e. the PANAS and flourishing scores. We chose the top 3 of them for this chart below. The most

important features for the first three models are the same because they all used Sklearn's Chi2 method[9] for finding the most important features. For the Decision-tree, we used the first 3 nodes to identify the most important features as these intuitively give the best splits between classes (and these node placements were decided with gradient boosting as mentioned previously).

Model	Flourishing scale	+PANAS	-PANAS
Support vector machine Logistic regression KNeighborsClassifier	Silence ratio	Activity duration	Noise ratio
	Noise ratio	Activity duration at night	Travel distance
	Travel distance	Walking ratio	Long conversations
Design-Tree Classifier	Day Light times	Conversation duration	Activity duration
	Activity duration for day	Night light time	Conversation frequency for day
	Mean sleep tiredness	Frequency of using wifi	Traveled distance

Table 8 Feature selected

		Baseline			Final Result (tuned, hyperparameters, features, selected)		
		-PANAS	+PANAS	FS	-PANAS	+PANAS	FS
Accuracy	LogisticRegression	0.96	0.7	0.47	0.96	0.72	0.72
	SVM	0.96	0.68	0.39	0.96	0.78	0.67
	KNeighborsClassifier	0.96	0.7	0.44	0.96	0.78	0.66
	DecisionTreeClassifier	0.92	0.46	0.55	0.96	0.82	0.73
RMSE	LogisticRegression	0.09	0.48	0.68	0.21	0.54	0.54
	SVM	0.09	0.5	0.77	0.21	0.48	0.58
	KNeighborsClassifier	0.09	0.5	0.73	0.21	0.47	0.58
	DecisionTreeClassifier	0.15	0.7	0.66	0.21	0.47	0.52

Table 9 Baseline and final results

5.2 Comparison Based on Flourishing Scale

After our own feature selection, the models have a higher accuracy, which could mean the features we selected have a higher relevance to the student's positive self esteem. The RMSE also decreases, suggesting the model has improved but could be slightly overfit.

5.3 Comparison Based on Positive PANAS Score

The performance of the first three models didn't change a lot with respect to positive PANAS. The accuracies between the default features and the features we selected by feature selection are similar, which could mean the features we selected are at fault because those three models share the same most important features. It is

also possible that none of the features correlate well to positive PANAS. The data set is small and so this is quite possible.
The decision tree model is however an outlier, with its accuracy almost doubling. This model uses a separate feature set and so we believe these are more correlated to positive PANAS/student emotion.

5.4 Comparison Based on Negative PANAS Score

In the baseline, the accuracies are very high while the RMSEs are very low compared to the results of flourishing scale and +PANAS score, which could mean there might be high overfitting for our models for this output. In the final results, the accuracies remain high while the RMSEs increased. It shows that after hyper-parameters tuning and features selection our models have improved, especially for logistic regression, k neighbors classifier and SVM, thus reducing overfitting whilst maintaining accuracy.

5.5 Small Dataset and Overfitting

We believe that it is possible our models overfit to the data. We considered splitting the data up into distinct training and test sets, and then performing k-fold cross validation on the training set before finally comparing the RMSE of the final model on both the training set and test set. Ideally, this would have allowed us to determine whether the models are overfit. However, the data set is really small in terms of the number of students, and so splitting further into a training and test set would lead to either a small testing set, which is prone to giving invalid conclusions, or a large test set but small cross validation sets, which could lead to a poorly trained model. For this reason, we decided to keep the data set whole and accept that we may have slightly overfit models. It is worth noting however that the drop in RMSE between the baseline and final result is significantly less (and sometimes increases) than the gains in accuracy, and so we deduce the models would only have a small amount of overfitting.

5.6 Advantages and Disadvantages of Various Methods

According to Table 9 above, after hyper-parameter tuning and feature selection, the decision tree model achieved the best accuracies for -PANAS, +PANAS and flourishing score. Also, compared with the results before of its own, the accuracies have improved greatly, yet it could be a disadvantage for the decision tree model, which indicates this model depends on hyper-parameter tuning and feature selection a lot. Another disadvantage is that it's hard for decision tree model to find correlations between features if there exists.
It's always easy to understand and implement a KNN model, and its performance is stable. Since our data is of a small size, the accuracies of prediction are decent. However, it is not suitable for high-dimensional data and imbalanced data, and there are not many hyper-parameters to be tuned to optimize the model.
The pros of logistic regression is that it works well for predicting categorical outcomes just like our task, yet it won't be efficient on predicting continuous values. A logistic regression model could be overfitting to some extent as well, which just happened to ours.
SVM also reached good accuracies according to Table 9, yet different extent of overfitting would happen to SVM, especially when the number of features is much greater than the number of target values.

6. Conclusion

The logical regression model was most performant for the baseline dataset, whilst the decision tree is best for the post-processing dataset (largest improvement).

The Decision tree appears to be the best model overall, but is highly dependent on hyperparameter tuning. This is evident by the significant increases in accuracy and improvements in RMSE (namely increases in RMSE when the baseline model was overfit and decreased RMSE when the baseline model was underfit). We believe that the main reason for this is that decision trees are more well suited to small datasets, whilst regression models require a large dataset. We believe we would need a much larger dataset (amount of students) to properly train the regression models because currently it is hard to determine whether the most important features for regression are correct or not (they may have only been chosen as important due to a small data size).

As the decision tree is most performant, we conclude that its features are most correlated with a students mental health. Finally, we believe our decision tree model could generalize well to unseen data and successfully determine student's mental health scores (PANAS, FS) to a high degree of accuracy. We do accept that our regression models may not perform as well as the decision tree unless they were given more data to train on, but nevertheless, they still appear to outperform the baseline results.

7. References

[1]. https://en.wikipedia.org/wiki/Observer_effect_(physics)

[2] Wang, Rui, et al. "StudentLife: assessing mental health, academic performance and behavioral trends of college students using smartphones." Proceedings of the 2014 ACM international joint conference on pervasive and ubiquitous computing. ACM, 2014.

[3] https://scikit-learn.org/stable/modules/cross_validation.html

[4] https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LogisticRegression.html

[5] https://scikit-learn.org/stable/modules/generated/sklearn.svm.SVC.html

[6]. https://scikit-learn.org/stable/modules/generated/sklearn.neighbors.KNeighborsClassifier.html

[7]. https://scikit-learn.org/stable/modules/generated/sklearn.tree.DecisionTreeClassifier.html

[8]. https://scikit-learn.org/stable/tutorial/statistical_inference/model_selection.html

[9]. https://scikit-learn.org/stable/modules/generated/sklearn.feature_selection.chi2.html