Group Project COMP9417
Machine Learning and Data Mining
T3, 2019

Group name:
Big vegetable chicken

Group members:

z5225857 Bangda Li

z5156479 Linfeng, Wang

z5048267 Lipeng, Tao

z5219640 Bang, An

z5216496 Tingning, Zhang

Introduction

Data from StudentLife contains several datasets such as Activity, Audio and Conversation, these sensor data collected by student's mobile phone through ten Weeks' life content. We want to predict the score of the flourishing scale and PANAS, then the values of them will be classification by a threshold, which may affect the accuracy of the results. In this case, we use the classifiers: random forest, k-NN and SVM to processing data, since these methods might be influenced by the data we selected, there might be different between the predict data and the result that we got from sensory data combined with survey research. This report will focus on demonstrating how to pre-process data, after that, it will describe and analysis these three methods. In the end, we will make a final statement on the results based on the accuracy.

Dataset

There are 10 datasets provided in total, and each of them have different features. For such a large amount of data information, it is necessary to integrate and group them. After analysis the features from each dataset, some of them can be classified by the same method. Generally, we divide 10 datasets into 6 categories, which are:

1. Physical activity interface & Audio: These two datasets are consisting with timestamp and interface and the numbers of ID are 4 for both of them. We can count the percentage of each ID, sort them, and then get the processed data.
2. Conversation: The features are start time and end time, we can choose a threshold to divide each conversation into three groups: long, medium, and short. Then count and sort the frequency of each group.
3. GPS Location: For the GPS location, we can split the latitude feature by 5 unit, group the divided data, and then counted them.
4. Bluetooth & WIFI: Although MAC and BSSID are not the same, they can be categorized in the same way. We can count the virtual address with high frequency of ID, and then select the top 30. This method can improve the efficiency since the it reduces the redundant data.
5. WIFI Location: Similar as the method we used in Bluetooth, we count and rank the top 30 buildings that send WIFI signals.
6. Light & Phone lock & Phone charge: These data contain start time and end time, we will classify them according to the Morning, Afternoon and Night, and then sort the data.

Methods

SVM (Support Vector Machines)

SVM is one of the most robust and accurate methods among all well-known data mining algorithms. It belongs to the two-class algorithm and can support linear and nonlinear classification. In a two-dimensional linearly separable data set, we have to find a hyperplane to separate the two sets of data. In this case, we think that linear regression or linear regression of linear regression can also do this classification. However, the best generalization ability is a hyperplane that maximizes the size of the space between the two classes.

The purpose of the support vector machine is to find the optimal hyperplane, which is the largest geometric interval. When the sample data is linearly separable, the interval maximization is called the hard interval maximization（When the training data is approximately separable, it is called soft interval.

The learning algorithm of support vector machine can be expressed as the following constraint optimization problem: max(w,b)gamma = (wx+b)/||W|| (s.t. yi(w*x+b)>=1, i = 1,2,3,...N)

k-Nearest Neighbors

The k-Nearest-Neighbors (k-NN) method of classification is one of the methods in machine learning. This method is widely used in machine learning problems and some other domains, such as recommendation systems, semantic searching, and anomaly detection. The main idea of k-Nearest Neighbors classifier is to find the most similar data points in the training data, and make a prediction based on their classifications. There is a simple way to represent k-NN method.

1. For each item in test set, computing the distance value to every item in the training set. The distance function is:

$$Dis\left(x^{(i)}, x^{(j)}\right) = \sqrt{\sum_{r=1}^{d}(x_r^{(i)} - x_r^{(j)})^2}$$

2. Picking the k closest data points which means the items with the k lowest distances.

3. Conducting a “majority vote” among those data points — the dominating classification in that pool is decided as the final classification. The formula is,

$$y = argmax \sum_{x_i \in N_k(x)} I(y_i = c_j), i = 1,2,\dots,N; j = 1,2,\dots,K$$

Random Forest description

Random Forest is an algorithm that integrates multiple trees through the idea of integrated Learning. Its basic unit is decision tree, and its essence belongs to a big branch of machine Learning -- Ensemble Learning. There are two key words in the name of random forest, one is "random", the other is "forest". "Forest" is very easy for us to understand. If one tree is called a tree, then hundreds of trees can be called a forest. In fact, this metaphor is quite appropriate. We'll talk about what "random" means in the next section

Dataset preprocessing

Data Output files:

Flourishing scale

Idea: Pre and post respectively calculate the total number, compared with median, divided into two types of low and high, pre and post state comparison, and get 4 types of results.
8 questions, each with a score range of 1-7 and a total score ranging from 8-56. According to the results can be divided into 4 categories:

Pre	Post
Low	Low
High	Low
Low	High
high	high

The four types of data are divided, classified according to the results, and the unique value of the four types of results is determined by the method of each result vote:

	low	High
Pre	0	1
Post	2	4

The output data can be processed into 4 categories corresponding to the above four types of cases:

pre	post	calc	level	grade
Low	Low	0+2	2	D
High	Low	1+2	3	C
Low	High	2+2	4	B
high	high	1+4	5	A

PANAS:

Similar processing with the flourishing scale, divided into 4 categories:

pre	post	calc	level	grade
Low	Low	0+2	2	D
High	Low	1+2	3	C
Low	High	2+2	4	B
high	high	1+4	5	A

The intersection of the participant with the flourishing scale is taken as the output data. The data of the flourishing scale is updated accordingly to remove the missing items.

Data Input files:

Activity and audio:

The activity has two columns of data, the first column is time, the second column is state, and 0-3 corresponds to Stationary, Walking, Running, Unknown. Audio structure is similar, 0-3 corresponds to Slience, Voice, Noise, Unknown.

After analyzing such data, there is a column of data ranging from [0, 1, 2, 3], and it is intended to pre-process the proportion of the value in the entire sample data, taking one of the ratios as input, and Other tags are calculated together with the output.

Conversation, dark, phonecharge, and phonelock:

This kind of data is a time period, and it is necessary to take the difference to obtain the time period and obtain the average of the data of the type.

k-fold Validation:

Cross-validation is a resampling procedure used to evaluate machine learning models on a limited data sample. The scale of input data and output data is showing as below (Figure 1). Since the limited data sample, we separate data to several sets then use one of them as test set and the others as training set. According to the above test accuracy, different k values are selected for different methods.

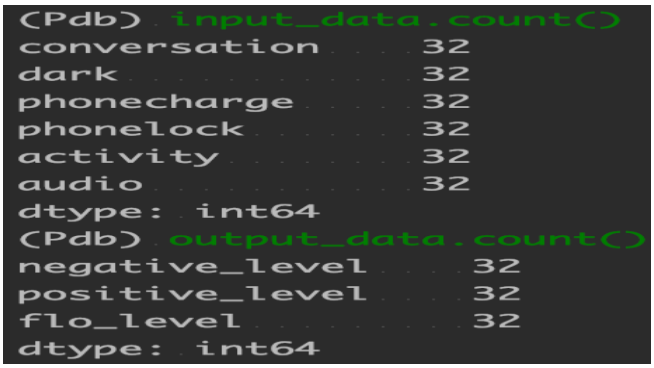


Figure 1 The scale of data set

Missing value

We find that there are some empty value data in the dataset. If we do not process the missing data, it would affect the accuracy of output result. According to Week 2 lecture, we decide to deal the empty value as below in order to minimize the influence:

1. If there are some missing value in input data, we decide to use the mean value of that column to replace the missing value.
2. There are some miss values in the output file. We did delete processing for data missing from the label. This is because we can't guess the true output

Binarization

The threshold should be giving first and convert feature to 0/1. The code as below:

```
binarizer = sklearn.preprocessing.Binarizer(threshold=1.1)
binarizer.transform(X)
```

In addition, we use binarization to process “PANAS” and “flourishing scale”. The data is compared with median and divided into 0 and 1 for low and high.

Min-max normalization

Since the dataset has a huge range of numbers, it would be hard and take a long period of time to process the data when computing the regression or result. What’s more if we use raw data which have a wide margin, there would be huge deviation. In order to deal with such problems, we decide to do a min-max normalization to make the data in a range of [0,1] using the following formula:

$$x_{new} = \frac{x - \min(x)}{\max(x) - \min(x)}$$

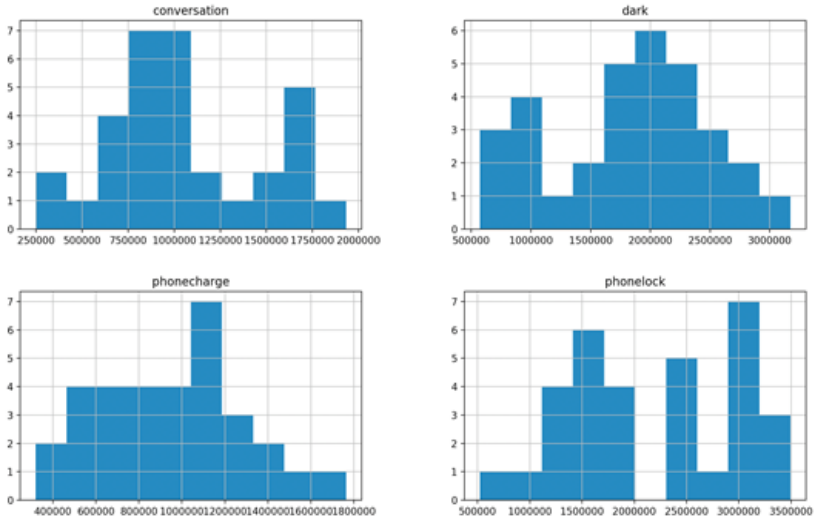


Figure 2 data distribution before *min-max normalization*

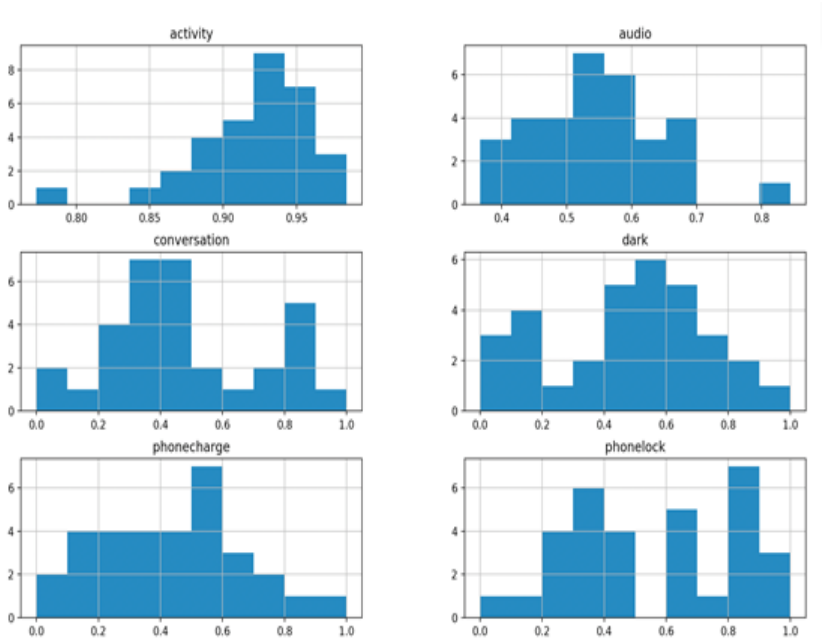


Figure 3 data distribution after *min-max normalization*

Feature extraction

Analysis of the correlation between various characteristics and the results through permutation and combination. In this work, we use ‘’activity’’, ‘’conversation’’, ‘’dark’’, ‘’phone charge’’ and ‘’phone lock’’ as input data, while ‘’flourishing score’’, ‘’PANAS positive score’’ and ‘’PANAS negative score as output data. For 'activity' and 'audio' data set, using percentage to mark different type in features; for 'conversation', 'phone charge' and 'phone lock' data set, calculating the time spending in each action, and median value is standard to divide data into 'long time' part and 'short time' part. With this, the input data sets and test data sets are ready for putting into algorithm to do machine learning.

Complex

k-Nearest Neighbours is a kind of lazy learning, it does not need to train the data set. Therefore, the running time of kNN method is the least among these three models.

For SVM, the time complexity for obtaining an analytical solution can be as low as O(Nsv3). Where Nsv is the number of support vectors. Although there is no fixed ratio, the number of support vectors is also related to the size of the training set.

Random forest need a huge number of subtrees to train the data, so it would take a period of time to get the result. If there are n samples, m features and m decision trees to process, then the time complexity is O(M(mn logn)).

Logic of method choices and design choices

Since after data pre-processing we only have about 40 valid datasets to use. In order to improve the accuracy of three models, we decide to do following method:

1. Use cross-validate to test and validate the accuracy of data
2. Find the appropriate K-fold value to training and testing
3. Adjust the hyper parameters to improve the accuracy

```
output_names = ["flo_level", "negative_level", "positive_level", ]
for on_index, on in enumerate(output_names):
    fig = plt.figure()
    index = 0
    fig_ax = {}
    for n in range(5, 11):
        # test kf from 5 to 10
        kf = KFold(n_splits=n, shuffle=True)
        logging.info("%s Fold test:" % n)
        score = []
        std = []
        for k in range(1, 20):
            clf = KNeighborsClassifier(n_neighbors=k)
            scores = cross_val_score(clf, input_data.values, output_data.values.T[on_index], cv=kf)
            logging.info("%s Fold %s neighbors KNN Accuracy: %0.2f (SD +/- %0.2f)" % (n, k, scores.mean(), scores.std() * 2))
            score.append(scores.mean())
            std.append(scores.std() * 2)
        fig_ax[index] = fig.add_subplot(6, 1, 1 + index, xlabel='neighbor', ylabel='Accuracy')
        fig_ax[index].plot(np.arange(1, 20), score, 'r-', np.arange(1, 20), std, 'b--')
        plt.ylim(top=1)
        plt.grid()
        plt.title(("%s Fold cross val" % n))
        index = index + 1
```

Figure 4 code of implementation

Result

Analyze the output data

Following is the Flourishing level, Panas positive level and Panas negative level histogram. After data process using Flouring_scale and Panas, we should have data result with following table:

pre	post	calc	level	grade
Low	Low	0+2	2	D
High	Low	1+2	3	C
Low	High	2+2	4	B
high	high	1+4	5	A

According to the bar graph (Figure 5), we found that there was no data with a score of 5.

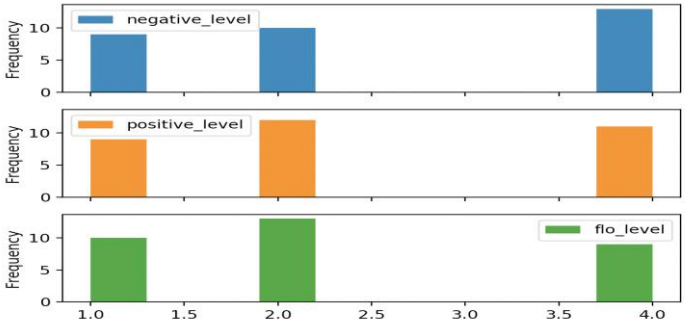


figure 5 data score result distribution

Analyze the relationship between input and output

We can analyze the relation between input and output of arbitrary combination of two-dimensional plane by drawing histogram.

The Figure 6 is the relationship between any two inputs and flourishing score:

D：× blue

C：● orange

B：▲ green

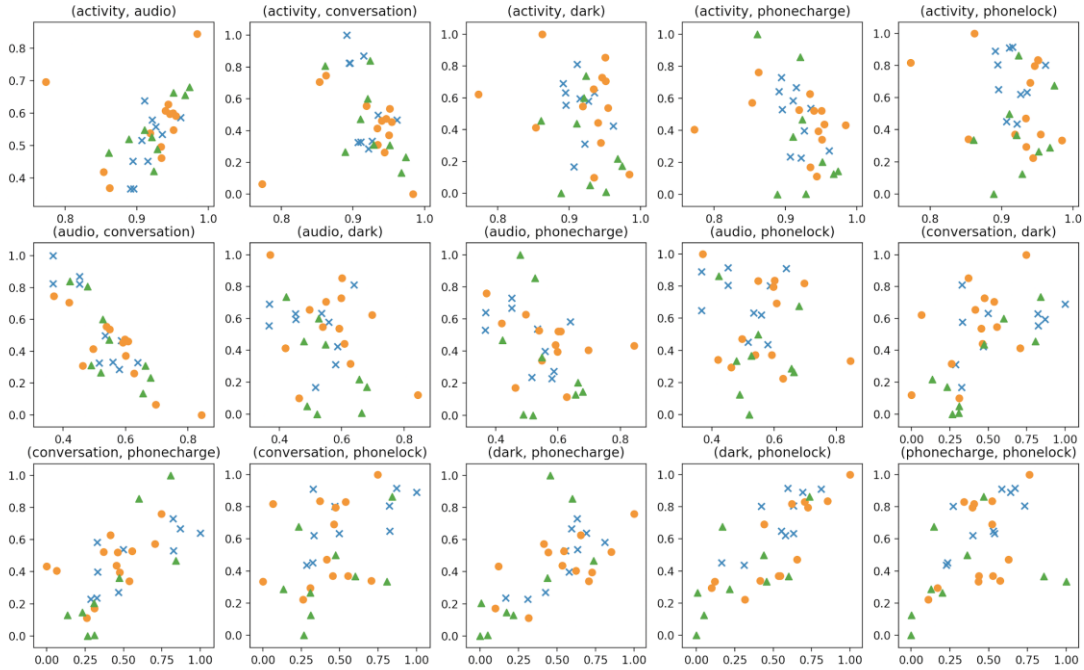


Figure 6 input and flouring scale score

Figure 7 is the relationship between any two inputs and panas positive score:

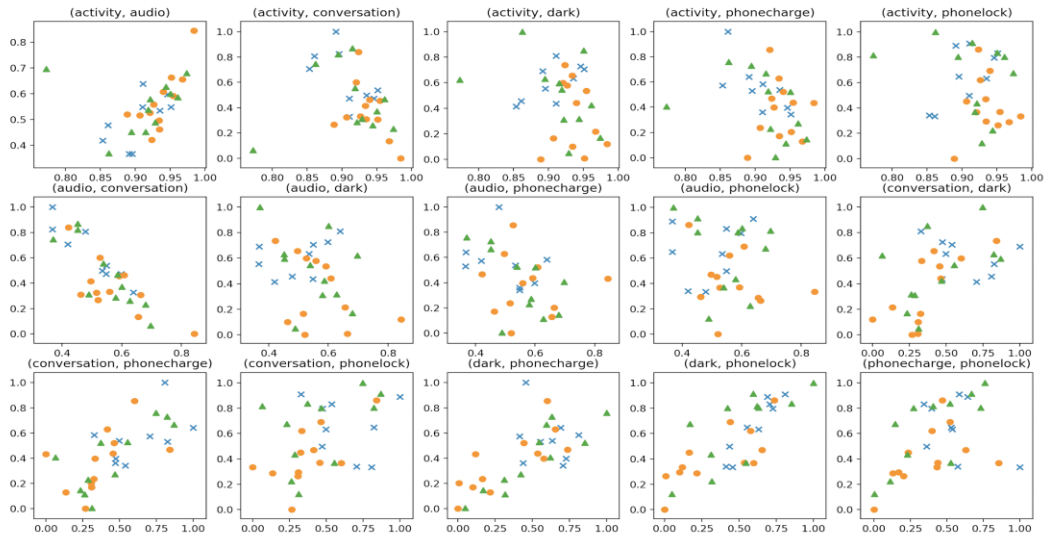


Figure 7 input and panas positive score

The Figure 8 is the relationship between any two inputs and panas negative score:

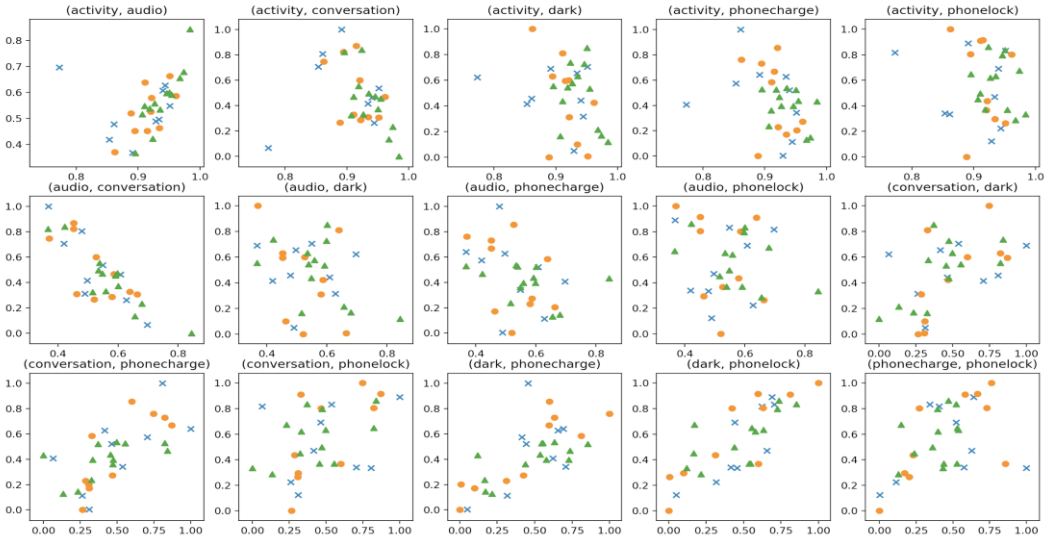


Figure 8 input and panas negative score

According to the picture, we cannot find any relation between the input and score. Maybe it is because we don't find the most suitable model for this dataset.

KNN

As result, choosing different number of neighbours and k value in k-fold cross have different results. These are hyperparameters which need us to adjust before running the algorithm. The k-fold from 5 to 10 and k-neighbour from 1 to 19 are showing as below (Figure 4), the red line is accuracy of different k-neighbour, the blue line is standard deviation, we except to improve the accuracy while decreasing the standard deviation. According to the printed accuracy and standard deviation (Figure 9), the best accuracy is 0.46 when choose 14 nearest neighbours with 5-fold cross validation.

```
2019-11-24 16:18:02,622 5 Fold test:
2019-11-24 16:18:02,635 5 Fold 1 neighbors KNN Accuracy: 0.27 (SD +/- 0.33)
2019-11-24 16:18:02,641 5 Fold 2 neighbors KNN Accuracy: 0.25 (SD +/- 0.14)
2019-11-24 16:18:02,647 5 Fold 3 neighbors KNN Accuracy: 0.28 (SD +/- 0.24)
2019-11-24 16:18:02,653 5 Fold 4 neighbors KNN Accuracy: 0.24 (SD +/- 0.30)
2019-11-24 16:18:02,659 5 Fold 5 neighbors KNN Accuracy: 0.25 (SD +/- 0.16)
2019-11-24 16:18:02,665 5 Fold 6 neighbors KNN Accuracy: 0.26 (SD +/- 0.26)
2019-11-24 16:18:02,671 5 Fold 7 neighbors KNN Accuracy: 0.35 (SD +/- 0.34)
2019-11-24 16:18:02,677 5 Fold 8 neighbors KNN Accuracy: 0.31 (SD +/- 0.17)
2019-11-24 16:18:02,683 5 Fold 9 neighbors KNN Accuracy: 0.32 (SD +/- 0.37)
2019-11-24 16:18:02,689 5 Fold 10 neighbors KNN Accuracy: 0.28 (SD +/- 0.20)
2019-11-24 16:18:02,695 5 Fold 11 neighbors KNN Accuracy: 0.25 (SD +/- 0.37)
2019-11-24 16:18:02,702 5 Fold 12 neighbors KNN Accuracy: 0.37 (SD +/- 0.23)
2019-11-24 16:18:02,707 5 Fold 13 neighbors KNN Accuracy: 0.36 (SD +/- 0.41)
2019-11-24 16:18:02,712 5 Fold 14 neighbors KNN Accuracy: 0.46 (SD +/- 0.30)
2019-11-24 16:18:02,718 5 Fold 15 neighbors KNN Accuracy: 0.30 (SD +/- 0.36)
2019-11-24 16:18:02,724 5 Fold 16 neighbors KNN Accuracy: 0.38 (SD +/- 0.20)
2019-11-24 16:18:02,729 5 Fold 17 neighbors KNN Accuracy: 0.37 (SD +/- 0.23)
2019-11-24 16:18:02,734 5 Fold 18 neighbors KNN Accuracy: 0.32 (SD +/- 0.23)
2019-11-24 16:18:02,739 5 Fold 19 neighbors KNN Accuracy: 0.37 (SD +/- 0.23)
```

Figure 9 The accuracy of kNN with 5-fold validation

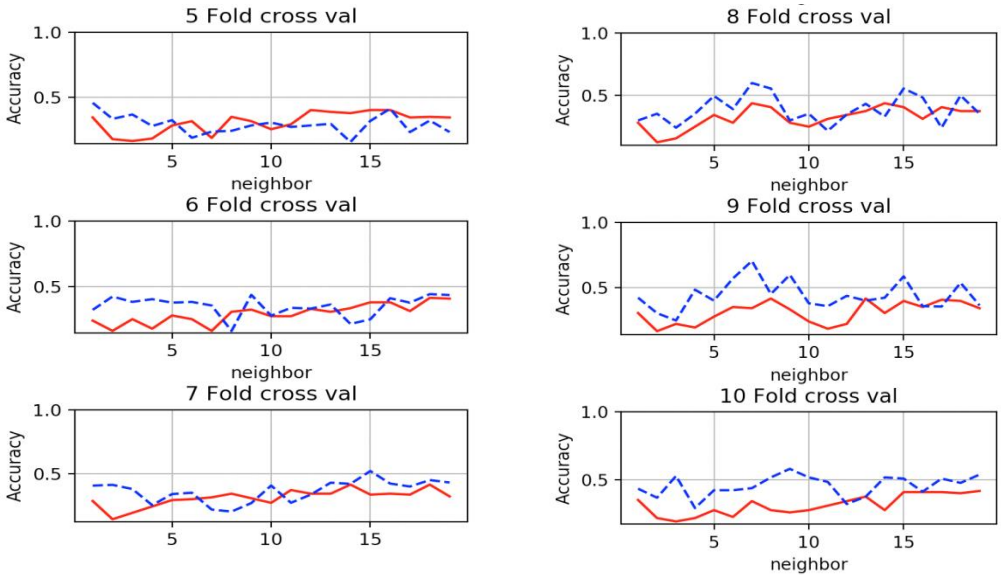


Figure 10 output Accuracy of knn

To summarize, kNN method, this method is a lazy learning algorithm, so it is an efficacy way to get results. On the other hand, the accuracy of kNN for this work is not perfect, the reason may be the input data of student data sets do not have obviously discrete relationship.

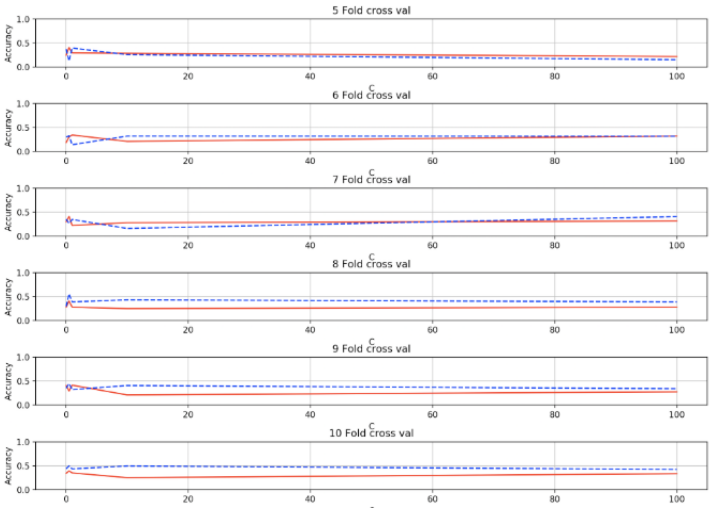
SVM hyperparameter interpretation

C: the penalty coefficient C of the objective function, used to balance the classification interval margin and error samples, the default C = 1.0;

Kernel: The parameters are RBF, Linear, Poly, Sigmoid, and the default is "RBF";

Gamma: the coefficient of the kernel function ('Poly', 'RBF' and 'Sigmoid'), the default is gamma = 1 / n_features;

Random_state : The seed of the pseudo-random number generator when the data is rearranged for probability estimation.The result as figure 11:(c=0.5)



)

Figure 11 SVM probability estimation

```
2019-11-24 16:11:54,665 5 Fold test:
2019-11-24 16:11:54,692 5 Fold 0.025 C SVM Accuracy: 0.32 (SD +/- 0.23)
2019-11-24 16:11:54,697 5 Fold 0.5 C SVM Accuracy: 0.31 (SD +/- 0.17)
2019-11-24 16:11:54,701 5 Fold 1 C SVM Accuracy: 0.27 (SD +/- 0.45)
2019-11-24 16:11:54,719 6 Fold test:
2019-11-24 16:11:54,725 6 Fold 0.025 C SVM Accuracy: 0.27 (SD +/- 0.40)
2019-11-24 16:11:54,730 6 Fold 0.5 C SVM Accuracy: 0.26 (SD +/- 0.30)
2019-11-24 16:11:54,735 6 Fold 1 C SVM Accuracy: 0.26 (SD +/- 0.45)
2019-11-24 16:11:54,746 7 Fold test:
2019-11-24 16:11:54,752 7 Fold 0.025 C SVM Accuracy: 0.40 (SD +/- 0.41)
2019-11-24 16:11:54,758 7 Fold 0.5 C SVM Accuracy: 0.41 (SD +/- 0.26)
2019-11-24 16:11:54,763 7 Fold 1 C SVM Accuracy: 0.38 (SD +/- 0.37)
2019-11-24 16:11:54,773 8 Fold test:
2019-11-24 16:11:54,780 8 Fold 0.025 C SVM Accuracy: 0.41 (SD +/- 0.43)
2019-11-24 16:11:54,787 8 Fold 0.5 C SVM Accuracy: 0.34 (SD +/- 0.56)
2019-11-24 16:11:54,793 8 Fold 1 C SVM Accuracy: 0.34 (SD +/- 0.50)
2019-11-24 16:11:54,803 9 Fold test:
2019-11-24 16:11:54,810 9 Fold 0.025 C SVM Accuracy: 0.38 (SD +/- 0.48)
2019-11-24 16:11:54,818 9 Fold 0.5 C SVM Accuracy: 0.37 (SD +/- 0.31)
2019-11-24 16:11:54,825 9 Fold 1 C SVM Accuracy: 0.19 (SD +/- 0.49)
2019-11-24 16:11:54,835 10 Fold test:
2019-11-24 16:11:54,843 10 Fold 0.025 C SVM Accuracy: 0.35 (SD +/- 0.43)
2019-11-24 16:11:54,851 10 Fold 0.5 C SVM Accuracy: 0.34 (SD +/- 0.53)
2019-11-24 16:11:54,859 10 Fold 1 C SVM Accuracy: 0.27 (SD +/- 0.52)
```

Figure 12 The accuracy of SVM with 5-fold validation

In summary, The larger the C, the worse the generalization ability and the problem of overfitting. Otherwise, it would be better with generalization ability, but the problem of under-fitting is easy to occur.

Random Forest Classifier

Following are the main hyper parameters to decide the random forest classifier, we can adjust them to get a better result:

max_features: The largest number of characteristic that random forest allows one decision tree to own. Default: None. Following is some examples:

Auto/None: Simply select all the features, and each tree can use them. In this case, there are no restrictions on each tree.

sqrt：This option is that each subtree can take the square root of the total number of features. For example, if the total number of variables (features) is 100, then each subtree can only take 10 of them. "log2" is another option of a similar type.

0.2： This option allows subtrees of each random forest to take advantage of 20% of the number of variables (characteristics). If we want to examine the function of the feature x %, we can use the format "0.x".

n_estimators: You need to build the number of subtrees before using the maximum number to vote or predict the result. More subtrees will give the model better performance, but it will make your code slower. You should choose value as high as possible, as long as your processor can handle it, because it makes your prediction better and more stable. Default: 10.

n_jobs: This parameter decides how many processors it can be used to compute. "-1" means there is no limitation. Default: None.

In order to choose the suitable parameter, we need to debug it repeatedly. And finally, we decide to use parameters: n_estimators=100, n_jobs=-1

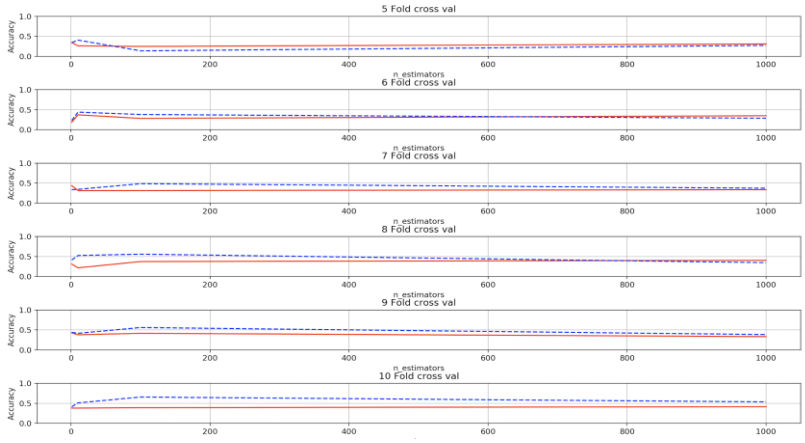


Figure 13 Random Forest probability estimation

```
2019-11-24 16:39:20,580 6 Fold test:
2019-11-24 16:39:20,648 6 Fold 1 n RandomForest Accuracy: 0.34 (SD +/- 0.52)
2019-11-24 16:39:21,548 6 Fold 10 n RandomForest Accuracy: 0.35 (SD +/- 0.54)
2019-11-24 16:39:23,228 6 Fold 100 n RandomForest Accuracy: 0.32 (SD +/- 0.20)
2019-11-24 16:39:30,670 6 Fold 1000 n RandomForest Accuracy: 0.28 (SD +/- 0.20)
2019-11-24 16:39:30,696 7 Fold test:
2019-11-24 16:39:30,779 7 Fold 1 n RandomForest Accuracy: 0.19 (SD +/- 0.38)
2019-11-24 16:39:31,800 7 Fold 10 n RandomForest Accuracy: 0.29 (SD +/- 0.44)
2019-11-24 16:39:33,857 7 Fold 100 n RandomForest Accuracy: 0.29 (SD +/- 0.31)
2019-11-24 16:39:43,288 7 Fold 1000 n RandomForest Accuracy: 0.26 (SD +/- 0.33)
2019-11-24 16:39:43,311 8 Fold test:
2019-11-24 16:39:43,404 8 Fold 1 n RandomForest Accuracy: 0.44 (SD +/- 0.33)
2019-11-24 16:39:44,652 8 Fold 10 n RandomForest Accuracy: 0.41 (SD +/- 0.24)
2019-11-24 16:39:46,869 8 Fold 100 n RandomForest Accuracy: 0.34 (SD +/- 0.43)
2019-11-24 16:39:56,937 8 Fold 1000 n RandomForest Accuracy: 0.28 (SD +/- 0.46)
2019-11-24 16:39:56,964 9 Fold test:
2019-11-24 16:39:57,038 9 Fold 1 n RandomForest Accuracy: 0.30 (SD +/- 0.45)
2019-11-24 16:39:58,371 9 Fold 10 n RandomForest Accuracy: 0.24 (SD +/- 0.45)
2019-11-24 16:40:00,792 9 Fold 100 n RandomForest Accuracy: 0.32 (SD +/- 0.52)
2019-11-24 16:40:12,141 9 Fold 1000 n RandomForest Accuracy: 0.24 (SD +/- 0.30)
2019-11-24 16:40:12,166 10 Fold test:
2019-11-24 16:40:12,284 10 Fold 1 n RandomForest Accuracy: 0.38 (SD +/- 0.37)
2019-11-24 16:40:13,790 10 Fold 10 n RandomForest Accuracy: 0.36 (SD +/- 0.35)
2019-11-24 16:40:16,595 10 Fold 100 n RandomForest Accuracy: 0.22 (SD +/- 0.37)
2019-11-24 16:40:29,263 10 Fold 1000 n RandomForest Accuracy: 0.23 (SD +/- 0.42)
```

Figure 14 The accuracy of Random Forest with 5-fold validation

To summarize the random forest, this is a model can be used in complex data environment, and it can valid to decrease data standard deviation and avoid noise. What's more, at the same time it can predict the data with a high accuracy. But it could take a longer time to do the procession when there is a large amount of data. In this project, the performance of random forest is not good enough, may because of the pre-procession of data generate excessive fitting.

Discussion

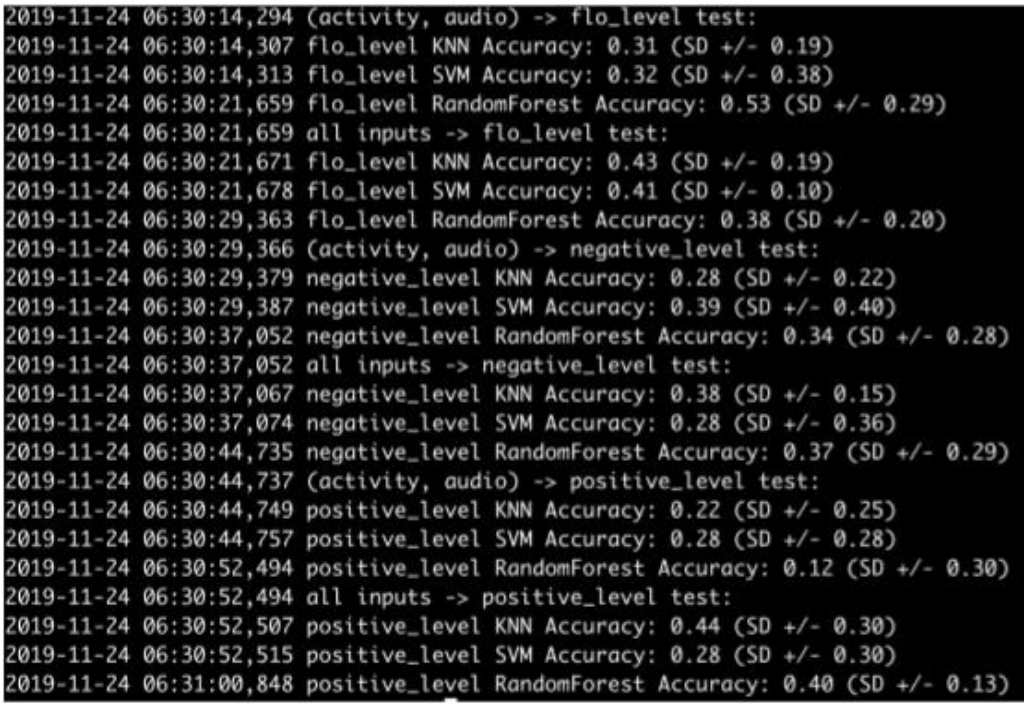


Figure 15 The Time Log

The differences between models

From Figure 15, the image of time log, the execution result of each model, each model illustrates different results, the accuracy and standard deviation. In addition, different models require different lengths of time to compute the results.

To compare the time complexity, the 'random forest' has the slowest speed of calculation, requiring the most time. In contrast, the 'KNN' takes the least amount of time then the 'SVM' follows.

k-NN is a type of lazy learning, where the function is only approximated locally and all computation is deferred until classification, which is one of the main differences compared with other models. While SVM uses Lagrangian multipliers to train input data, and random forest is based on decision tree to maximize the information gain, then classifying the input data.

The advantages and disadvantages of each model

Models	Advantages	Disadvantages
KNN	1.it is simple and mature, suitable for classification and regression. 2.It is suitable for non-linear regression. 3.The time complexity is O(n).	1. It requires large amount of calculation. 2. it requires significant interior memory.
SVM	1.It is capable of non-linear classification. 2.It provides a better result of classification. 3. It can maximize the margin between the classes.	1. It is not ideal for large dataset classification. 2. It does not support multiple classification.
The random forest	1. It is insensitive to missing values and informal values. 2. It produces high training accuracy.	1. It may cause overfitting. 2. The data attributes with more values will have a greater impact on random forests.

The findings

For the larger scale of input data, the 'KNN' would have the best accuracy in all tests. In contrast, if the input data is more significant, the accuracy of the 'SVM' would be improved in the flourishing test, decline in the negative test, remain unchanged in the positive test. Lastly, the 'random forest' would decrease the accuracy in the flourishing test but enhance the accuracy in both negative and positive tests.

Simultaneously, when the scale of the input data is changed, the 'random forest' has the most fluctuation of its standard deviation whereas the fluctuations of rest two models' are far less than it.

Conclusion

In this wok, we use StudentLife dataset to apply machine learning techniques to predict flourishing scale and panas score. According to these scores, we can evaluate the mental health of students. It is difficult to estimate each student's mental health, academic performance and behavior trends as the effect of the data classification is not ideal. Consequently, it also causes the relatively lower accuracy of each model. The 3 models are generally effective models, which have unique advantages but are not ideal for all situations. As a result of unsatisfactory data categories, it causes the relatively poor results of models. Although the accuracy could be improved, it is still far from the expected outcomes. For a larger dataset, the 'knn' would get a better accuracy whereas the other models' accuracy could rise, decline or remain unchanged. Simultaneously, when the amount of data is changed, the standard deviations among three models would fluctuate within different intervals.
The quality of this project can be improved by applying more effective models, since there are several input datasets not using in this case, the accuracy of the results might be under expected. Another factor that affect the outcome may be because the method implemented processing of missing data is not effective.

Reference

1. A Liaw, M Wiener, 2002, *Classification and regression by randomForest, R news*, Vol. 2/3, https://www.researchgate.net/profile/Andy_Liaw/publication/228451484_Classification_and_Regression_by_RandomForest/links/53fb24cc0cf20a45497047ab/Classification-and-Regression-by-RandomForest.pdf
2. Segal, Mark R, 2004, *Machine Learning Benchmarks and Random Forest Regression*, https://escholarship.org/uc/item/35x3v9t4
3. Naresh Kumar, 2019, Advantages and Disadvantages of Random Forest Algorithm in Machine Learning, The Professional Point, http://theprofessionalspoint.blogspot.com/2019/02/advantages-and-disadvantages-of-random.html
4. Weinberger, K.Q., Blitzer, J. and Saul, L.K., 2006. Distance metric learning for large margin nearest neighbor classification. In *Advances in neural information processing systems* (pp. 1473-1480).
5. Peterson, L.E., 2009. K-nearest neighbor. *Scholarpedia*, *4*(2), p.1883.
6. Naresh Kumar, 2019, Advantages and Disadvantages of SVM (Support Vector Machine) in Machine Learning, The Professional Point, http://theprofessionalspoint.blogspot.com/2019/03/advantages-and-disadvantages-of-svm.html
7. Naresh Kumar, 2019, Advantages and Disadvantages of KNN Algorithm in Machine Learning, The Professional Point, http://theprofessionalspoint.blogspot.com/2019/02/advantages-and-disadvantages-of-knn.html