APPLY MECHINE LEARNING TO PREDICT TWO PSYCHOLOGICAL PHENOMENA BASED ON STUDETLIFE DATASET

West Journey Group

GROUP MEMBERS:
Z5231733-ZHENHANG SUN
Z5223292-SHIYI YANG
Z5173767-XINJIE ZHI
Z5200177-YU ZHANG

Introduction

Many investigations indicate that students' life may be affected by various aspects like their psychological health, academic performance, life habits and so on. In order to find the relationships behind the Students' life and lots of different factors, in this project, we use the StudentLife dataset to conduct experiments. This dataset includes two folders: Inputs and Outputs. Inputs are factors that may affect students' life and data are from a mobile app. Outputs are from two self-reported questionnaires which are separately represent the self-perceived and emotion(positive and negative affect) over the past weeks. From questionnaires, we can acquire different scores of students, and what we want is to use the information from the Inputs and scores from Outputs to find relationships between them.

In this project, we preliminarily decide this task into two parts: classification (mainly focus) and regression. For classification, we will implement logistic regression, perceptron and SVM. With respect to the regression task, we will use linear regression and neural network methods. We will evaluate the performance of different models in different datasets. Finally, we will acquire the best model separately in classification and regression.

Dataset

In this project, we use StudentLife dataset. It contains two folders: Inputs and Outputs. Inputs records 10 different sensing data species, some of which will be used as the features of each student. And Outputs collects two kinds of methods to measure the performance of every student's life. Before we implement our experiments, we must pre-process the raw data and select some useful features for the following techniques.

For the Inputs, we will choose some meaningful features from these 10 sensing data:

- Activity: records student's status in every timestamp, which includes stationary(0), walking(1), running(2) and unknown(3). The number represents the degree of the status. Our group decides to use the mean value of all the statuses degree as the first attribute, the larger number implies this student prefers moving.

- Audio: makes audio inferences and it has four description: silence(0), voice(1), noise(2) and unknown(3). Like “Activity”, we use the mean value as our second attribute.
- Conversation: records the duration for each call. We assume the duration obeys Gaussian distribution thus we use its mean value as the third feature.
- GPS Location, WiFi: these two data can be used to inference the WiFi Location, so we decide to only use the WiFi Location data.
- WiFi Location: records the location of the student at each timestamp. Each student may go to different places and stay there for a various duration. We decide to pick up the top N high frequency locations of each student and then consider every location as a feature. In our experiment, we will set N to 5, 8 and 10. For example, if N is 5, then we will get another 51 new features(because some locations will be repeated), every feature stands for a specific location.
- Bluetooth: records the MAC address of surrounding Bluetooth device and signal strength at every timestamp. We think this data is relatively useless, so we decided not to use it.
- Light: records the duration for each dark environment period (that is sleep period). We assume the duration obeys Gaussian distribution and use its mean value as the feature.
- Phone Lock: records the duration for each phone locked period. We assume the duration obeys Gaussian distribution and use its mean value as the feature.
- Phone Charge: records the duration for each charging period. We assume the duration obeys Gaussian distribution and use its mean value as the feature.

In conclusion, we will use Activity, Audio, Conversation, Light, Phone Lock, Phone Charge as our features and use WiFi Location to create some other new features (depends on the value of N, here we set N to 5). Now we obtain the input data.

For the Outputs, there are two evaluation methods: Flourishing Scale and PANAS. Flourishing Scale is calculated by adding eight items’ scores, and the range of this scale is from 8 to 56. Higher score means a person with many psychological resources and strengths. PANAS consists of positive affect score and negative affect score. Higher positive score represents a higher level of positive affect and lower negative score shows lower levels of negative affect. Therefore, we get three scores datasets. We consider these as our classification labels.

For regression task, we consider the scores as the ground truth. As for binary classification, we use binarization method to divide the scores into two balanced groups, 1 represents high scores and 0 stands for low values. The threshold we choose is the median value of each scores dataset. We consider these as our regression labels.

In the end, we decide to predict the Flourishing Scale and PANAS scores for post students.

NOTE:

In Outputs, there are some missing values for some students, in this case, we will ignore these students and not use them as our training set or test.

Methods

We will present classification and regression separately and evaluate whether they work well.

Pre-processing and feature extraction

Before training, we found that different features have different sizes, so we need to do pre-processing on the input data. First, we apply z-score standardization to scale the feature, which can speed up the convergence during the training and improve the performance of the model. Then, we also found that the number of features is larger than that of the instances. So we need to do feature selection. Here, we focus on the Filter method to choose more useful features. Filter has several approaches: Anova test, Chi-Square test and Pearson's Correlation. After comparing each approach, we finally decide to use Pearson's Correlation. Besides, we will choose 10 most meaningful features for our training. Now, we finally determine which features will be used for training.

First, we consider the classification method.

Classification

Evaluation metric selection

To evaluate our model, we choose mean accuracy as metric because of the following 2 reasons:

1. The number of all samples is fewer.
2. The scores are divided into 2 groups and the numbers of samples in the two groups are nearly equal, which means the computed result will not be affected by data stew.

We calculate mean accuracy score based on 5-fold cross validation for each dataset. And use this metric to find the best hyperparameters.

Models

We start with a simple model: Logistic Regression.

For the training part, we add regularization term to decrease the influence from overfitting, and it will help us to set some relatively unimportant features' coefficients to small values during the training. In order to find the best penalty value of the regularization term, we use grid search method, that is to set some different penalty values previously.

In experiment, we use LogisticRegression model, which is from sklearn library, as our training model. And set some relative parameters as below: penalty = 'l2', max_iter = 100, solver = 'liblinear', random_state = 0. Another relative parameter C is the inverse of regularization strength, our aim is to find the optimal C to fit the dataset. We choose the value of C from a list including 7 values which are 0.001, 0.01, 0.1, 1, 10, 30, 100, 300, 1000. After training three datasets, we get best C which can make each model has the highest mean accuracy on test data.

The following tables show the mean accuracy of training set and test data:

Table 1-1: Result of the Flourishing Scale(post)

C	0.001	0.01	0.1	1	10	30	100	300	1000
Accuracy of training data	0.824	0.838	0.851	**0.872**	0.851	0.845	0.845	0.851	0.858
Accuracy of test data	0.811	0.811	**0.811**	0.779	0.807	0.807	0.807	0.807	0.807

For the Flourishing Scale(post) dataset, the optimal parameters are: penalty = ‘l2’, max_iter = 100, solver = ‘liblinear’, random_state = 0, C=0.1 and the accuracy is 0.811.

Table 1-2: Result of the PANAS Negative Score(post)

C	0.001	0.01	0.1	1	10	30	100	300	1000
Accuracy of training data	0.751	0.757	**0.757**	0.751	0.757	0.757	0.757	0.757	0.757
Accuracy of test data	0.557	0.557	**0.636**	0.636	0.636	0.611	0.611	0.611	0.611

For the PANAS Negative Score(post) dataset, the optimal parameters are: penalty = ‘l2’, max_iter = 100, solver = ‘liblinear’, random_state = 0, C=0.1 and the accuracy is 0.636.

Table 1-3: Result of the PANAS Positive Score(post)

C	0.001	0.01	0.1	1	10	30	100	300	1000
Accuracy of training data	0.796	0.796	0.770	0.789	0.796	0.803	0.803	**0.809**	0.809
Accuracy of test data	**0.714**	0.686	0.682	0.682	0.657	0.632	0.604	0.604	0.604

For the PANAS Positive Score(post) dataset, the optimal parameters are: penalty = ‘l2’, max_iter = 100, solver = ‘liblinear’, random_state = 0, C=0.001 and the accuracy is 0.714.

Furthermore, it is easy to conclude that LogisticRegression model performs better on Flourishing scale than PANAS.

Secondly, we use perceptron.

Perceptron is a linear regression with supervised learning to solve dichotomies. We use Perceptron model, which is from sklearn library, as our training model. In this case, we choose l2 regularization to reduce the complexity of the model to prevent overfitting, because we have already used filters to get the features we need, we might get fewer features if we use l1 regularization. Obviously, this is not what we want. In addition, we choose to train the perceptron model 1000 times, tol equals to '1e-3' and random_state equals to 0 to increase the accuracy of the perceptron model. Moreover, in the perceptron model, alpha is the coefficient of the regularization term. So, we need to find the most appropriate alpha for each dataset to improve the generalization ability of the perceptron model, that is, the accuracy of the perceptron model. We choose the value of alpha between [0.0001, 0.0010] with a step size of 0.0001. After training three datasets, we get best C for each dataset.

The following tables show the mean accuracy of training set and test set:

Table 2-1: Result of the Flourishing Scale(post)

alpha	0.0001	0.0002	0.0003	0.0004	0.0005	0.0006	0.0007	0.0008	0.0009	0.0010
Accuracy of training data	0.810	0.810	0.837	0.831	0.831	**0.858**	0.844	0.837	0.837	0.851
Accuracy of test data	0.721	0.721	0.696	0.721	0.746	0.721	0.696	0.721	**0.750**	0.746

For the Flourishing Scale(post) dataset, the optimal parameters are: penalty = 'l2', max_iter = 1000, tol = '1e-3', alpha = 0.0009 and the accuracy is 0.750.

Table 2-2: Result of the PANAS Negative Score(post)

alpha	0.0001	0.0002	0.0003	0.0004	0.0005	0.0006	0.0007	0.0008	0.0009	0.0010
Accuracy of training data	0.671	0.651	0.652	0.638	0.672	0.665	0.639	0.626	**0.691**	0.671
Accuracy of test data	0.546	0.496	0.550	0.421	0.521	**0.575**	0.489	0.518	0.575	0.571

For the PANAS Negative Score(post) dataset, the optimal parameters are: penalty = 'l2', max_iter = 1000, tol = '1e-3', alpha = 0.0006 and the accuracy is 0.575.

Table 2-3: Result of the PANAS Positive Score(post)

alpha	0.0001	0.0002	0.0003	0.0004	0.0005	0.0006	0.0007	0.0008	0.0009	0.0010
Accuracy of training data	0.731	0.738	0.764	0.738	0.751	**0.784**	0.738	0.757	0.751	0.718
Accuracy of test data	0.664	**0.793**	0.668	0.693	0.693	0.668	0.743	0.614	0.643	0.536

For the PANAS Positive Score(post) dataset, the optimal parameters are: penalty = 'l2', max_iter = 1000, tol = '1e-3', alpha = 0.0002 and the accuracy is 0.793.

In conclusion, we can find that the perceptron model performs better on the PANAS Positive Score than PANAS Negative Score and Flourishing Scale.

Thirdly, we use Support Vector Machine.

Support Vector Machine (SVM) is a Supervised learning algorithm and can be used in classification and regression. Here, we use classification method. SVM has three models to support classification method: SVC, LinearSVC and NuSVC. We choose LinearSVC model because it is more flexible to choose penalty and loss function.

We use LinearSVC model, which is from sklearn library, as our training model. In the LinearSVC model, we also choose l2 regularization to prevent overfitting for the same reason we mentioned before. And then, we choose 'squared_hinge' as loss function and choose 10000 as the max iteration because we need more than enough times of iteration to determine the optimal value of C. We choose the value of C from a list including 7 values which are 0.001, 0.01, 0.1, 1, 10, 100, 1000. After training three datasets, we get best C for each dataset.

The following tables show the mean accuracy of train set and test set:

Table 3-1: Result of the Flourishing Scale(post)

C	0.001	0.01	0.1	1	10	100	1000
Accuracy of training data	0.845	0.845	0.865	**0.872**	0.865	0.872	0.845
Accuracy of test data	0.811	**0.839**	0.807	0.779	0.807	0.725	0.704

For the Flourishing Scale(post) dataset, the optimal parameters are: penalty='l2', loss='squared_hinge', dual=True, C=0.01, max_iter=10000 and the accuracy is 0.839.

Table 3-2: Result of the PANAS Negative Score(post)

C	0.001	0.01	0.1	1	10	100	1000
Accuracy of training data	0.731	0.738	**0.744**	0.770	0.770	0.757	0.718
Accuracy of test data	0.557	0.611	0.611	0.611	0.611	**0.636**	0.582

For the PANAS Negative Score(post) dataset, the optimal parameters are: penalty='l2', loss='squared_hinge', dual=True, C=100, max_iter=10000 and the accuracy is 0.636.

Table 3-3: Result of the PANAS Positive Score(post)

C	0.001	0.01	0.1	1	10	100	1000
Accuracy of training data	0.763	0.750	0.789	0.809	**0.836**	0.829	0.797
Accuracy of test data	**0.686**	0.682	0.607	0.554	0.554	0.529	0.582

For the PANAS Positive Score(post) dataset, the optimal parameters are: penalty='l2', loss='squared_hinge', dual=True, C=0.001, max_iter=10000 and the accuracy is 0.686.

From the above comparison, we can conclude that LinearSVC model performs better on Flourishing scale than PANAS.

Next, we move to the regression method.

Regression

Evaluation metric selection

Here, we choose mean squared error (MSE) to evaluate each model. A better model should have a small MSE value. We calculate mean MSE based on 5-fold cross validation for each dataset. And use this metric to find the best hyperparameters.

Models

Also, we start with a simple model: Linear regression.

Linear regression is often introductory algorithm for supervised learning because it is easy to use and interpretable. Standard linear regression will fail in the case of high collinearity between feature variables. Collinearity is an approximate linear relationship between independent variables, which will have a huge impact on regression analysis. Here we consider two regression methods replacing standard linear regression, namely Lasso and Ridge. Both of them add an offset term in the regression optimization function to reduce the influence of collinearity and thus reduce the variance of the model. We use Linear Regression model, which is from sklearn library, as our training model.

The following tables show the mean MSE of train set and test set:

Table 4-1: Result of the Flourishing Scale(post)

alpha	0.001	0.01	0.1	1	10	100	1000	10000
MSE of training data (Ridge)	**50.156**	**50.156**	50.163	50.276	51.293	62.126	74.206	76.493
MSE of test data (Ridge)	122.547	122.348	120.798	114.405	81.337	**65.170**	73.874	75.455
MSE of training data (Lasso)	**50.156**	50.162	50.315	55.831	76.769	76.769	76.769	76.769
MSE of test data (Lasso)	122.197	119.006	99.216	**68.456**	75.643	75.643	75.643	75.643

For the Flourishing Scale(post) dataset, L2 regularization does better, which means in this dataset, all features are been chosen, there is no feature weighted 0. The optimal method is Ridge, max_iter = 20,000 and MSE is 65.170.

Table 4-2: Result of the PANAS Negative Score(post)

alpha	0.001	0.01	0.1	1	10	100	1000	10000	100000	800000
MSE of training data (Ridge)	**28.714**	29.185	29.315	29.977	33.286	44.70	55.971	58.63	58.938	58.969
MSE of test data (Ridge)	1815.17	622.428	594.696	536.552	280.39	87.881	65.333	63.272	63.068	**63.049**
MSE of training data (Lasso)	**28.466**	29.295	30.23	36.399	58.973	58.973	58.973	58.973	-	-
MSE of test data (Lasso)	5389.4	593.427	505.09	85.701	**63.046**	**63.046**	**63.046**	**63.046**	-	-

For the PANAS Negative Score(post) dataset, the two methods are performed similarly. Choosing all features with L2 regularization or choosing some of them with L1 regularization seems no significantly difference. Either of them can be chose, max_iter = 150,000 and MSE is 63.05.

Table 4-3: Result of the PANAS Positive Score(post)

alpha	0.001	0.01	0.1	1	10	100	1000	10000
MSE of training data (Ridge)	**18.56**	**18.56**	18.561	18.593	19.961	30.695	40.351	42.163
MSE of test data (Ridge)	516.518	516.192	512.939	481.51	277.148	52.323	**43.096**	44.803
MSE of training data (Lasso)	**18.56**	18.562	18.63	24.533	42.383	42.383	42.383	42.383
MSE of test data (Lasso)	515.356	504.536	399.319	**36.747**	45.045	45.045	45.045	45.045

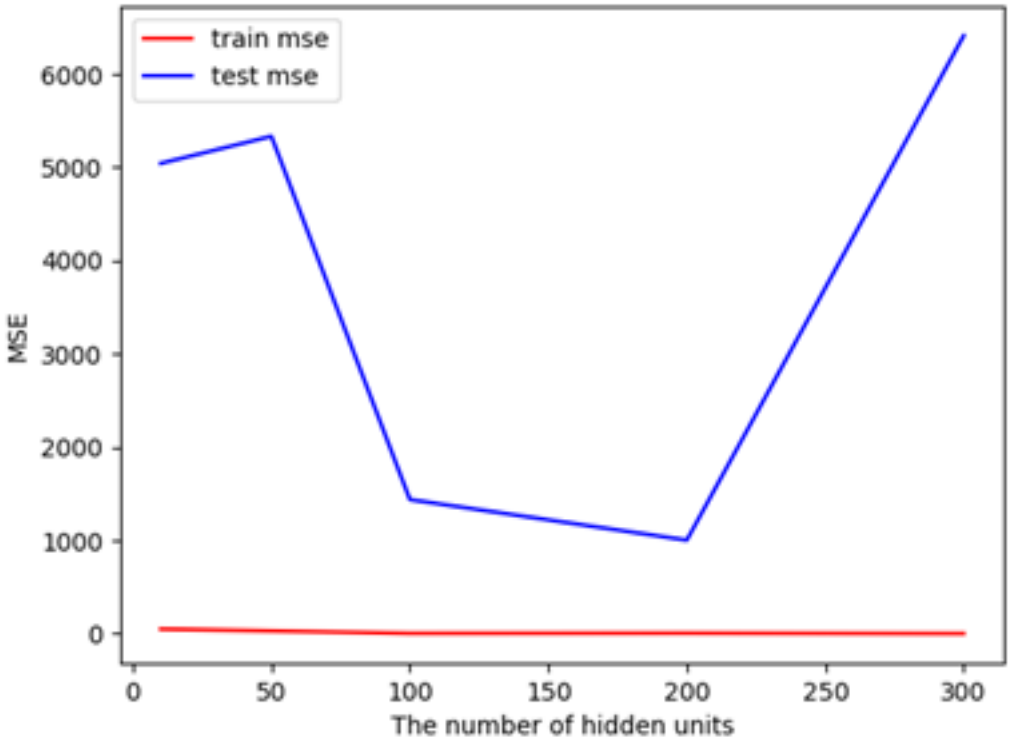
For the PANAS Positive Score(post) dataset, L1 regularization does better, which means there are several unimportant features, those weights are set to 0. The optimal method is Lasso, max_iter=20,000 and MSE is 36.747.

Secondly, we consider the neural network.

After implementing Linear Regression model for the Regression Task, we see that the MSE value is relatively large, one possible reason is that the data is not separated linearly well. So now we try to use Neural Network model to see if it can decrease the MSE value. Here, we only use single one hidden layer.

For the experiments, we will use MLPRegressor model from the sklearn library. Before training, we need to set some hyperparameters. First, we set some fixed hyperparameters: activation = ‘relu’ (which can speed up the convergence during the training), solver = ‘adam’, max_iter = 10000. Then choose the best combination of these two hyperparameters: the number of neurons in hidden layer (h), alpha (L2 penalty parameter). Based on "Bias-Variance Tradeoff", we will pick h and alpha from the following choices: h = [10, 50, 100, 200, 300], alpha = [0.01, 0.1, 1, 10, 100].

For example, when predicting the Flourishing Scale, we set alpha = 0.01 previously, and then to choose the best h. Here is the result:



From this figure we can see that the best h is 200, thus we get the combination 1 (alpha = 0.01, h = 200), for convenience, we use (0.01, 200) for short. Then we will repeat the above process to obtain totally 5 combinations.

Next, we will use these 5 combinations to train the model and compare the MSE value on test set from each model. Finally, choose the combination with the smallest MSE value for Flourishing Scale dataset. Same for other two datasets.

The following tables show the MSE of train set and test set:

Table 5-1: Result of Flourishing Scale(post)

(alpha, h)	(0.01, 200)	(0.1, 100)	(1, 200)	(10, 50)	(100, 200)
Train MSE	4.757	2.093	20.490	49.222	56.238
Test MSE	1028.013	1658.035	259.405	102.687	**62.758**

For the Flourishing Scale(post) dataset, we can find that the optimal alpha = 100 and h = 200. In this case, the MSE is 62.758.

Table 5-2: Result of PANAS Negative Score(post)

(alpha, h)	(0.01, 10)	(0.1, 10)	(1, 10)	(10, 10)	(100, 10)
Train MSE	22.600	23.026	23.747	27.578	39.486
Test MSE	71.843	**68.493**	106.416	122.053	102.953

For the PANAS Negative Score(post) dataset, we can find that the optimal alpha = 0.1 and h = 10. In this case, the MSE is 68.493.

For PANAS Positive Scores, we find that the 'logistic' activation is better than 'relu'. Therefore, the result below is under the 'logistic' activation.

Table 5-3: Result of PANAS Positive Score(post)

(alpha, h)	(0.01, 300)	(0.1, 50)	(1, 100)	(10, 50)	(100, 300)
Train MSE	7.780	3.990	16.342	20.482	42.423
Test MSE	42.078	39.322	41.602	**38.166**	45.067

For the PANAS Positive Score(post) dataset, we can find that the optimal alpha = 10 and h = 50. In this case, the MSE is 38.166.

Discussion

Now, we have already implemented three classification methods and two regression approaches. Obviously, there are some differences among them when using them on different datasets.

For classification, we present a table below:

Table 6-1: The maximal mean accuracy of three models on three different datasets

	Logistic Regression	**Perceptron**	**SVM**
Flourishing	0.811	0.750	**0.839**
PANAS Negative	**0.636**	0.575	**0.636**
PANAS Positive	0.714	**0.793**	0.686

From the results above, we can see that SVM performs well on Flourishing dataset, maybe in this dataset, most of the features are useful, and SVM is good at addressing high dimensional space problem, so it gets the highest mean accuracy. For PANAS Negative dataset, Logistic Regression and SVM are both better than Perceptron. However, Perceptron gets the best result on PANAS Positive datast, probably because this dataset is easy to be linearly separated and Perceptron is easy to use compared with the other two methods. Besides, Logistic Regression is easy to occur underfitting problem and only used for binary classification.

For regression, there also shows a result below:

Table 6-2: The minimal mean MSE of two models on three different datasets

	Linear Regression	**Neural Network**
Flourishing	65.170	**62.758**
PANAS Negative	**63.046**	68.493
PANAS Positive	**36.747**	38.166

From the table above, we can find that Linear Regression, except Flourishing Scale dataset, performs better than Neural Network on the other two datasets. Maybe for these two datasets, Neural Network is relatively too complex to fit and much easier to cause overfitting. Neural Network is more suitable for the nonlinear dataset but it is time-consuming. In contrast, Linear Regression is much easier to implement but performs poorly on nonlinear dataset.

Conclusion

For this project, we want to use the Inputs to predict three scores which can be calculated from the Outputs. In order to achieve it, we consider this task as two different problems: classification and regression. Before training, it is important to do feature selection to pick up useful features, as well as do preprocessing on the features, which can speed up the training rate, avoid overfitting and improve model performance.

For classification, we apply Logistic Regression, Perceptron and SVM. And we find that no one method always performs well, it depends on the form of the dataset. SVM is more suitable for high dimensional features, however, Logistic Regression and Perceptron are useful for linear dataset and both of them are easy to implement.

For regression, we use Linear Regression and Neural Network. Neural Network is a relatively complex model for nonlinear dataset and Linear Regression is much simpler and only performs well on linear dataset.

In conclusion, from this project, we clearly know that how to do pre-processing before training and testing, which means that how to choose the meaningful features from datasets based on the specific questions, how to further filter out features that are not useful and how to scale the features to get what we want. Besides, we know exactly how to choose more suitable models of the specific datasets to make sure higher performance and how to adjust the multiple parameters of each model during training and test, especially, moderating the penalty to avoid overfitting and get higher accuracy. Moreover, we distinctly know that how to choose a more appropriate method to evaluate the models and find that the advantages and disadvantages of every supervised learning method we used under the specific datasets. More importantly, we found the necessity and importance of teamwork.