UNSW SYDNEY

COMP 9417 Group Project

Group name: my cat is a shoegazer

Group member:

Yichen Liu (z5233576)

Shaohong Xu (z5193639)

Jiaqi Sun (z5233100)

Dan Su (z5226694)

1. Introduction

The features we've been using are based on Student Life dataset[1]. The original dataset doesn't contain any classified label, so in order to retrieve the labels, we've been given two more datasets which are Flourishing Scale and PANAS score. We did some calculations on those two datasets to get the labels. Feature preprocessing was implemented to obtain 78 potential features from the sensing data for prediction. By using several feature extraction like univariate feature selection, PCA etc., the top 10 features with the highest score were selected and applied to other measures. More details are in section 2.

Our goal in this project is to make reasonable predictions towards students' mental health conditions. We will regard it as a binary classification problem in the following sections.

2. Dataset

2.1 Dataset Info

In order to find answers related to the college student life, Student Life dataset was collected by researchers from Dartmouth College, The University of Texas and Northeastern University. The whole dataset based on smartphone sensing was collected through a class of 48 students across a 10-week term, tracking their life habits and daily workload.

The research group recruited volunteers from a computer science programming class at Dartmouth College in the 2013 spring term and offered them the same Android type smartphone to implement the data collection followed by the orientation session. Each student was given a one-on-one tutorial of the Student Life system and trained to properly use the app. In addition, students were required to fill the questionnaire before and after the data collection for analysis. The surveys included different measures (eg. Panas and Flourishing) to evaluate the behavioral and psychological states[2].

The dataset covers activity, conversation, GPS location, mood state, sociability, etc. During the 10 week period, the smartphone continuously collected the user information by sensing the different information (accelerometer, proximity, audio, light sensor readings, location).

With the expectation of data quality, the students were notified to get back on track once they deviated, which promoted a promising set of statistics.

2.2 Data Grouping

The Flourishing scale contains 8 statements which are separated into 7 levels. The total value of the 8 items can indicate psychological health to some extent. In this project, we set a threshold to divide the values into two classes, indicating the higher level of subjective well-being and lower level of subjective well-being. Considering the missing values in the flourishing result, the students with a missing value were dropped in order to increase the accuracy of classification. The remaining data of each student were added up, and then divided into 0 or 1 by using the median value as the threshold.

The PANAS score shows the level of positive affect and negative affect. The positive and negative items were computed separately, and the total scores are classified into two classes (0 and 1) respectively. For each measure, the group method is the same as mentioned in the Flourishing scale classification.

In addition, vertical comparison, which combines the pre and post data and computes their mean value, has been applied to enrich the data analysis level. When computing the mean value of pre- and post- scores, we found the students who are both in pre- and post- dataset by intersecting two lists. We hope it could indicate the relation between the student's variation across the research period and their mental/physical states.

3. Methods

3.1 Brief abstract

During the process of finding out the most applicable regression and classification models, decision tree and k nearest neighbours were initially applied which didn't perform well, then we moved on principal component analysis, neural networks, support vector machine, and finally random forest classification. Considering the limitation of the amount of examples, we applied cross-validation to prevent stochastic prediction results. In data processing, various feature extraction methods were attempted to gain higher performance scores. We have also

done many comparisons, such as between data which splits the day/night and not, between different classifiers, between different classes, etc. Many possible approaches to process the data have been applied and will be specified in this report.

3.2 Feature selection

Pre-selection stage:

By looking through all the sensing dataset in the input, we decided to select some dataset which might have the high positive effect we were interested in for further feature extraction, including conversation level, phone usage level, audio and activity level, and Bluetooth connection. There are three methods to increase the dimension of a feature.

- 10 weeks division: As the experiment was conducted in 10 weeks, we first split the data of each student by using week timestamp. By finding the first timestamp of all data which represents the start time of the whole experiment, the data was separated into 10 parts according to the time slot. Every data in one time slot was considered as a feature.
- Night-time extraction: Some features might have a higher influence at night, such as the dark dataset for predicting sleep quality. For obtaining the local time the event occurred, we used the datetime module to transform the timestamp into Coordinated Universal Time (UTC), and then find the corresponding hour in the timestamp. We defined the nighttime is between 18pm and 8am, and extracted the time duration in this interval for further use.
- Daytime and nighttime separation: This method is similar to the night-time extraction, but the time is separated into two parts: daytime (8am - 18pm) and nighttime (18pm - 8am).

The conversation dataset records the timestamp of the start time and end time. Since the conversation duration and frequency are always considered as a kind of key factor influencing people's sociability and psychological health[3,4], many operations have been done to obtain meaningful conversation features. First, considering the time, we tried to divide the total time into 10 weeks and capture the difference between the two classes. Besides, we also took the night time the conversation activities happened into account, so the

Unix timestamp was first transformed into UTC and then into the Eastern Time to obtain the conversation feature at night (18pm - 8am).

By looking up some literature, we found that healthy sleep also plays an important role in mental health, so we decided to separate the related data described in reports, including phone usage condition (phone lock, phone charge), dark duration, silence time and stationary time, and select the most important features[3,5]. The phone usage datasets, including 'phone lock' and 'phone charge', and the dark dataset have the same two fields: start time and end time, which can be used to compute the duration. For other features like activity feature and audio feature, we focused on the occurrence of 0 value, which means stationary and silence respectively. In this part, we first split all the data into 10 weeks to measure the mental health, and then we also extracted the timestamp at night for precisely predicting sleep quality.

We paid attention to the audio dataset and activity dataset as they can represent the social activity the students attend[3,6]. The total frequency recorded in the dataset are computed with respect to different status and split into 10 weeks for further use. For the audio dataset, there are 3 main inferences: silence (0), voice (1) and noise (2). Stationary (0), walking (1) and running (2) are inferences in the activity dataset. In addition, in order to improve the precision of predicting, the separation of the daytime (8am - 18pm) and the nighttime (18pm - 8am) was also considered and implemented on these two datasets.

The Bluetooth dataset was divided into two parts according to the timestamp, some research showed that the Bluetooth connection times can indicate the location that people prefer to go. Depressed people tend to stay at home or visit the uncrowded area where few devices can be detected, while positive people are likely to attend more activities so more connections are detected. The time attending activity also represents the personality and mental condition of a person, so the time division was also used in this dataset.

post-extraction stage:

We first checked all the features preprocessed if they have nan value and try to remove features with zero variance or all features with the same sample value. Highly correlated or collinear features may result in overfitting. When a pair of variables are highly correlated, we should delete a variable to reduce the dimension without losing too much information.

According to the sklearn module, we adopted a lot of feature extraction methods as the following:

- Univariate feature selection: Univariate feature selection is based on the univariate statistical test to select the optimal feature.
- Recursive feature elimination: Given an external estimator that assigns weights to features (such as coefficients of a linear model), Recursive Feature Elimination (RFE) will select features by recursively considering fewer and fewer feature sets.
- SelectFromModel: removing the least import features from the previous estimator after fitting.
- Principal component analysis (PCA): Linear dimensionality reduction using Singular Value Decomposition of the data to project it to a lower dimensional space.

3.3 Methods applied

With the help of pandas and numpy module, the preprocessed data could be encapsulated in matrices with vectorized form, and then it is effective to apply them in different methods we use as the following lists:

- K Nearest Neighbors
- Random Forest Classification
- Support Vector Machine
- Logistic Regression
- Neural Network

3.4 Normalization

When dealing with attributes of different sizes, we generally need to normalize the data, otherwise, it can lead to the dilution of equally important attributes (on a lower scale) because the values of other attributes are on a larger scale. In summary, if there are multiple attributes, but the values of the attributes are in different ranges, this can lead to a bad data model when performing data mining operations.

Here we use the min-max normalization, which rearranges the data into relative value from 0 to 1 as the following formula:

$$Yi = [Xi - min(X)]/[max(X) - min(X)]$$

3.5 Data augmentation

Data augmentation is another way to avoid overfitting. In this project, we had two times of features than training examples so it has to be suffering from overfitting. After extracting features and normalization, for example, we may have a matrix of size (40,10). We first repeat the matrix to let it become (4000,10). Then for each row, we add Gaussian noises of (mu = 0, sigma = 0.33). At last, we shuffle the whole matrix. By this means, we could get a (4000,10) dataset instead of (40,10) one. We compared neural networks' performance with and without augmentation in the following sections.

3.6 Tuning method and cross validation

We designed similar functions for each method separately. The function includes a classifier /regression model object. After fitting and transforming the training data into the object, we use the GridSearchSV module from sklearn to tune the highest parameter with K-fold cross validation strategy for the consideration of inadequate samples and print current metrics scores. Since this is a kind of operator, we can apply them every time we modify the feature and check the improvements.

3.7 Level of method complexity

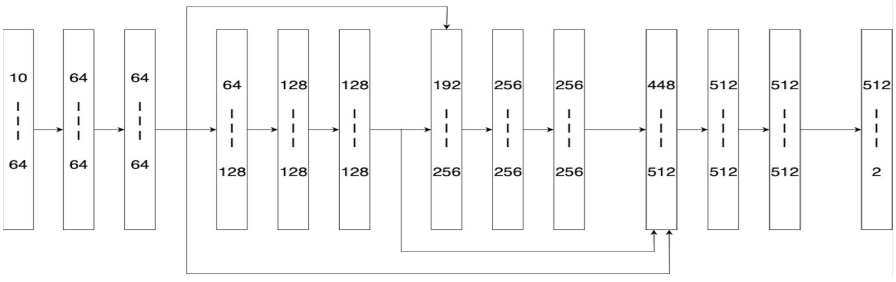
K-nearest neighbour(KNN) algorithm can classify the data by measuring the distance between different features, the classification result largely depends on the value of k. When we using KNN algorithm to classify new data, we need to calculate the distances between the new data and each old data in the dataset, so the computational complexity is proportional to the number of data in the dataset($O(n)$). And the time complexity is also $O(n)$, where n is the dataset size. Therefore, the KNN algorithm is suitable for the dataset which has fewer samples.

Random Forest(RF) algorithm problem deal with the high variance problem of Decision Tree, it is an ensemble tree with a random subset of features. Under the default parameters, the complexity of the Random Forest algorithm is $O(m \cdot n \cdot log_2 n)$, where m is the number of trees in the forest and m is the dataset size. But we can decrease the complexity by setting these parameters: min_sample_split, min_sample_leaf, max_leaf_nodes and max_depth. This algorithm is suitable for the data with a low dimension but needs high accuracy.

Support Vector Machine(SVM) aims to find a decision boundary between different categories, this interface separate two types of samples and maximize the margin as much as possible. The complexity of the SVM algorithm depends on the number of support vectors $O(n\_samples \cdot n\_features)$ and that's why it is not easy to generate overfitting. The SVM algorithm has a wide range of uses and it performs well on many kinds of datasets.

Logistic Regression(LR) algorithm can use new data to update the weights, do not need to reuse the old training data. We can assume the LR as a single layer Neural Network. By introducing the Sigmoid function into Linear Regression, LR can map the output into the range of (0, 1), then predict the probability to classify the data. If we swap the Sigmoid function into Softmax function, we can deal with the multi-class classification problem.

The time complexity of the Neural Network is hard to compute. The model we've been using has about 987k parameters in the networks. All layers are linear layers. Batch size is 32.



3.8 Method choices and design choices

KNN algorithm is very easy and the time complexity is very low, besides the math behind KNN is easy to explain. This algorithm does not have an assumption on data, it makes sure that the result has high accuracy. But this algorithm is lazy learning, this issue the speed of prediction is very low when we have a large number of features[7].

RF algorithm select data and feature randomly. Because of this random choice, this algorithm has good anti-noise property and can avoid overfitting to some extent. But training the model and predicting the result is slow by using Random Forest algorithm.

SVM algorithm is friendly to the small training dataset, the final function is determined by a few support vectors, according to this way, SVM can find the key samples and discard the tolerant samples. But SVM does not have good performance in dealing with the multi-class classification. This object is a binary classification problem.

LR algorithm can be implemented easily, during the classification process, just need a small amount of calculation, the processing speed is fast but just need small memory space. But LR is easy to get an under-fitting problem.

The Neural Network has several advantages, it has a high accuracy of classification, strong robustness and fault-tolerant to noise. But the Neural Network need lots of parameters and cannot show the learning process to the users.

3.9 Evaluation metrics

In this project, we use the following evaluation metrics to measure the performance of the algorithm.

Accuracy: the ratio of the number of samples correctly classified by the classifier to the total number of samples. $A = \frac{TP + TN}{TP + TN + FP + FN}$

Precision: $P = \frac{TP}{TP + FP}$, where the TP is correctly classified result and the denominator is actually classified as True. The precision score shows the ability of the classifier to distinguish the negative samples.

Recall: $R = \frac{TP}{TP + FN}$, where the denominator is the samples which should be classified as True. The Recall score shows the ability of the classifier to identify the positive samples.

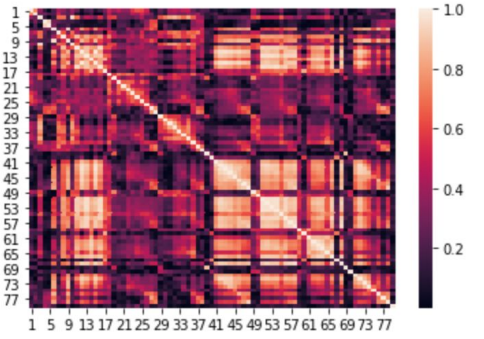
F1-score: $F1 = \frac{2 \cdot Precision \cdot Recall}{Precision + Recall} = \frac{2TP}{2TP + FN + FP}$, F1-score is a combination of Precision and Recall. The higher the F1-score, the more stable the classification model.

ROC-AUC score: Use true positive rate as the vertical axis and false positive rate as the horizontal axis. Take a different threshold to make a graph. The larger the area under the curve(AUC), the more ideal the model. In other words, the closer the curve is to the upper left corner (true_positive_rate = 1, false_positive_rate = 0), the better the performance of this model.

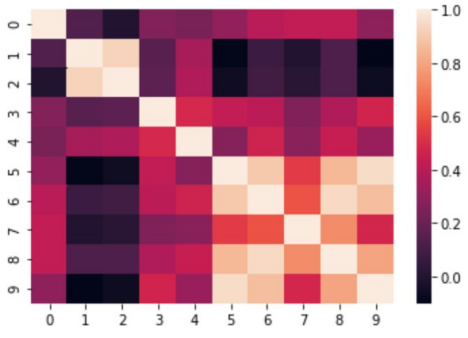
4. Results

After applying feature extraction and comparing their correlations with each other. High correlations mean they are linearly dependent and there is no need to keep both of them. The following features are our final choices:

Conversation_week9, act_sit_week6, act_walk_week4, act_walk_week9, act_run_week1, act_run_week9, act_run_week10, audio_silent_week2, audio_silent_week4, audio_silent_week6.



Heatmap ooverallfeatures



Heatmap over 10 selected features

The graphs above are correlation heatmaps. Before applying feature extraction, there are a lot of features having high correlations with each other (light colors). After extraction, most of the highly correlated features got eliminated. And this would be a huge improvement towards overfitting.

flourishing post	auc_roc	accuracy	precision	recall	f1_score
rfc	0.944	0.756	0.593	0.611	0.585
lr	0.778	0.567	0.481	0.889	0.622
knn	0.819	0.700	0.574	0.556	0.552
svm	0.806	0.728	0.685	0.611	0.626
neural networks	0.897	0.833	0.943	0.771	0.840
neural networks (augmentation)	0.880	0.811	1.000	0.611	0.759
panas positive post	auc_roc	accuracy	precision	recall	f1_score
rfc	0.685	0.683	0.685	0.667	0.626
lr	0.843	0.583	0.581	0.944	0.686

knn	0.532	0.394	0.296	0.389	0.330
svm	0.815	0.622	0.519	0.333	0.385
neural networks	0.675	0.569	0.676	0.550	0.591
neural networks (augmentation)	0.800	0.711	0.640	0.889	0.744
panas negative post	auc_roc	accuracy	precision	recall	f1_score
rfc	0.185	0.450	0.093	0.111	0.100
lr	0.204	0.359	0.365	0.778	0.487
knn	0.444	0.448	0.222	0.278	0.222
svm	0.231	0.581	0.000	0.000	0.000
neural networks	0.208	0.233	0.206	0.333	0.255
neural networks (augmentation)	0.662	0.632	1.000	0.125	0.222

The table above lists all the performances achieved by various classifiers on different labels. To unify the conditions, we choose post as our test examples.

We use the GridsearchCV to find out the best hyperparameter when selecting the best 10 features. The hyperparameters resulted from the computation of roc_auc score and are adopted according to the best consequence.The range of hyperparameters are as follows:

- random forest parameter: {'n_estimators':range(10,101,10), 'max_depth':[1,2,3,4]}
- lr parameter: {'penalty':['l1','l2'], 'class_weight':[None,{1:0.5, 0:0.5},{1:0.6, 0:0.4},{1:0.4, 0:0.6}]}
- knn parameter: 'n_neighbors':range(3,6), 'algorithm':['auto','ball_tree','brute']}
- svm parameter: {'kernel':['rbf','sigmoid','linear'], 'degree':[1,2,3,4]}
- GridSearchCV: model, parameter, scoring='roc_auc', cv=9

According to the metrics, the Neural Network model generally performs best, showing its powerful algorithm. The Random Forest also performs well due to its amount of multiple decision trees, whereas the rest of models have drawbacks on one or some of metrics. Through the observation of these statistics, we could declare that the negative label unfits the features, and thus it is not suitable for classification solely.

5. Discussion

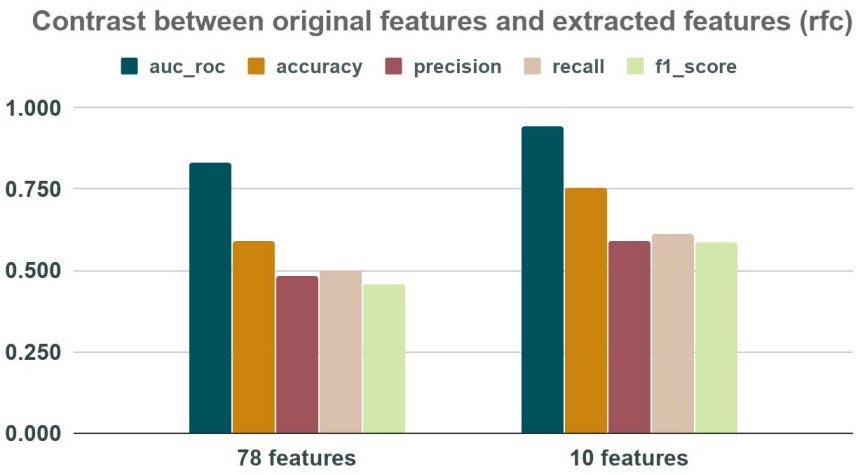


Figure 5.1

We first combined all the 78 features together and obtained the performance of the random forest model, and then extracted 10 best features from the original features. The comparison between the evaluation of the original and extracted features is shown in Figure 5.1. The result shows that after optimization by selecting the top 10 features with the highest correlation, the models can get a better performance based on the same data.

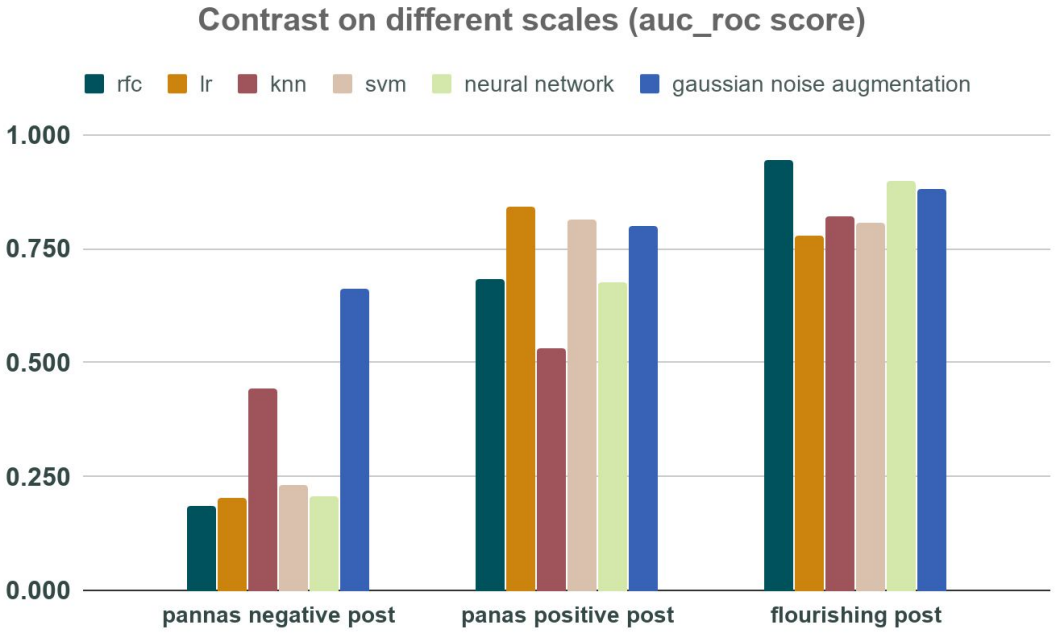


Figure 5.2

The selected 10 features were applied to other measures (positive PANAS classification and negative PANAS classification) to show the difference between measures with the same feature selection (Figure 5.2). Since the Flourishing post-experiment was used to select the best features, the models on this data have the best performance. The selected features perform the worst on negative PANAS classification. However, the auc_roc score of Gaussian noise augmentation model still maintain a relatively high level, indicating this model is robust and steady on the negative PANAS classification.

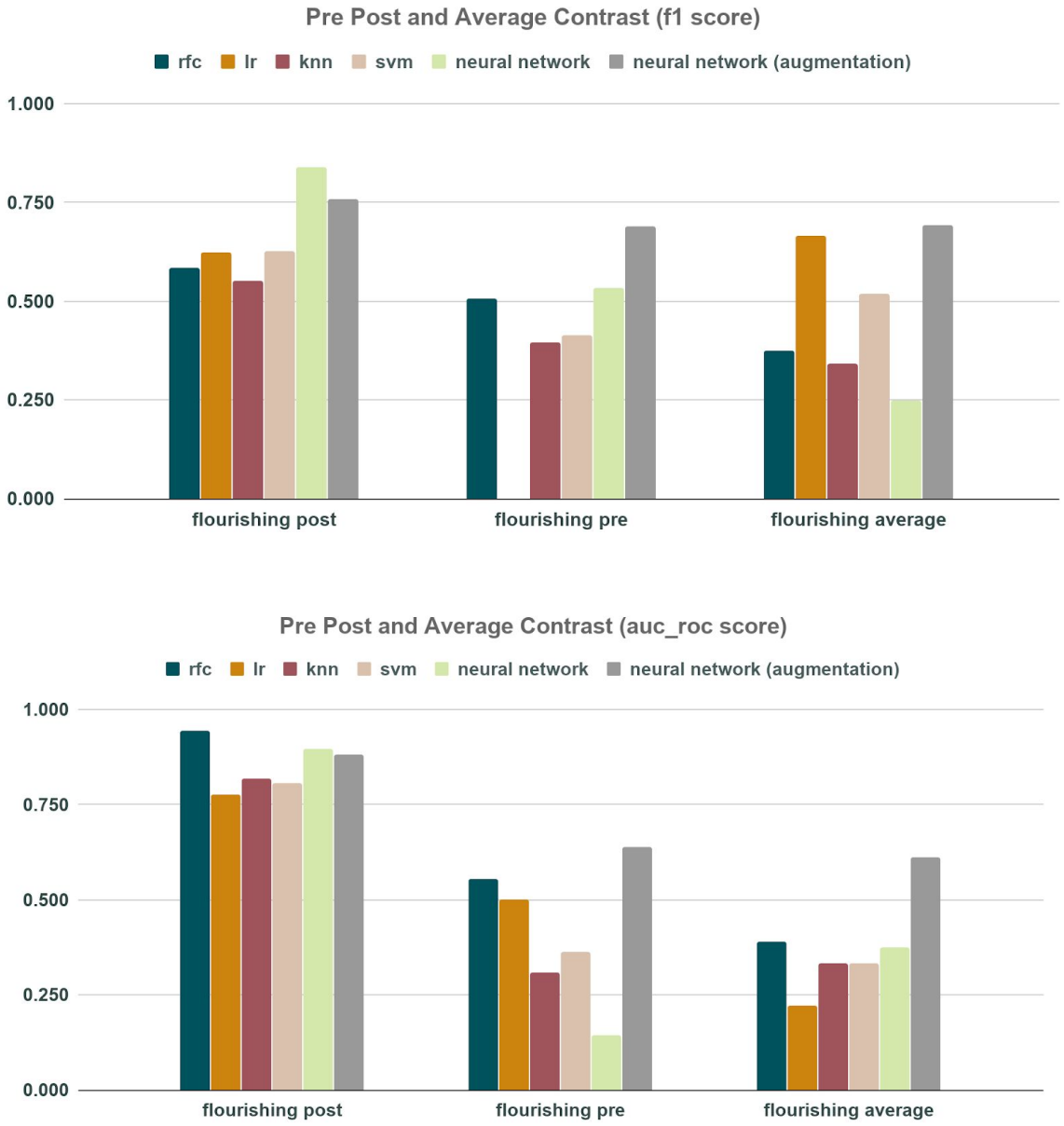


Figure 5.3

Since the self-assessment scale was implemented in two periods of time, pre-experiment and post-experiment, the pre-, post- and the average of these two scores were all applied to make

the prediction. The auc_roc score and f1 score of them applied to 6 models were shown in Figure 5.3.

The LR model and KNN model are less complex compared with other models. SVM model and Random forest model usually give better performance with more complex architecture. For different models, the neural network augmentation shows outstanding stability, but the time complexity is extremely high and we need so much time and data to train the model.

6. Conclusion

This project extracts several features from the StudentLife dataset. After preprocessing the data, we select the best 10 features to test the performance of five models (Random Forest, Logistic Regression, KNN, SVM and Neural Network). Comparing the results of five models on a fixed dataset, the Neural Network has the best performance. Data augmentation on neural network would slightly affect the performance but it still proved its validity and provided more robustness. A fine-tuned neural network with enough data (using augmentation) in flourishing post dataset could be a good start for predicting students' mental conditions.

Over all datasets, the prediction on flourishing post achieved the best results. The panas positive sentiments performs better than the negative sentiments. This suggests positive sentiments would be a more accurate measure than negative ones.

We could also see from the features that the most impacting features are around week 4, week 6 and week 9. Our interpretation is that those features focus on the time around mid-term and final exams. We consider this as a strong relation between exams and students' mood.

All of these suggest the potential relationship embedded in the dataset. Hopefully it could help the future research in the related field.

Reference:

1. StudentLife Dataset 2014. http://studentlife.cs.dartmouth.edu/.
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. Lane, Nicholas D., et al. "Bewell: A smartphone application to monitor, model and promote wellbeing." 5th international ICST conference on pervasive computing technologies for healthcare. 2011.
4. Rabbi, Mashfiqui, et al. "Passive and in-situ assessment of mental and physical well-being using mobile sensors." Proceedings of the 13th international conference on Ubiquitous computing. ACM, 2011.
5. Chen, Zhenyu, et al. "Unobtrusive sleep monitoring using smartphones." Proceedings of the 7th International Conference on Pervasive Computing Technologies for Healthcare. ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering), 2013.
6. Lu, Hong, et al. "The Jigsaw continuous sensing engine for mobile phone applications." Proceedings of the 8th ACM conference on embedded networked sensor systems. ACM, 2010.
7. Parvin, Hamid, Hoseinali Alizadeh, and Behrouz Minati. "A modification on k-nearest neighbor classifier." Global Journal of Computer Science and Technology (2010).