Group Project
Comp9417 Machine Learning and Data Mining
T3, 2019

Group Name:

Kill Four Vegetable Chickens (KFVC)

Group Members:

Given Name	Surname	ZID
Xiangyu	Tian	z5223517
Yuao	Jiang	z5217274
Anqige	Wu	z5199351
Rong	Zhen	z5225226

Contents

1. Introduction

It is reasonable for people to believe that there is a significant correlation between students' mental health and behavior trends. Researchers at Dartmouth College explored a sensing app named StudentLife integrated with MobieEMA, a flexible ecological momentary assessment, helping to collect data relating to many aspects of students' lives including mental well-being, education outcomes, behavioral trends. (Wang & etc., 2014). Based on the original public dataset from StudentLife provided on the official website, this report is aimed at introducing data processing, modeling establishment, and accuracy comparison.

2. Dataset

2.1 Overview of Dataset

The Student Life dataset used in this project is a collection of sensing data from the phones of 48 Dartmouth students over the 10-week term. This dataset is split into two parts: Input which represents the X group and Output which represents the Y group. The input includes 10 features named as：Activity, Audio, Bluetooth, Conversation, Dark, GPS Location, Phone-charge, Phone-lock, Wi-Fi and Wi-Fi-location. The output includes a flourishing scale aiming to measure self-perceived success and PANAS scores which is a measure of positive and negative affect.

It should be noticed that timestamps were generally used in this dataset. A timestamp in this dataset is a sequence of numbers that can represent when a certain event occurred (En.wikipedia.org, 2019). Datetime function in python can convert the timestamp into a certain time in the form of Year-Date-Time, offering readings exactly on seconds. Further processing with timestamp will be discussed in Section 2.2 and Section 3.2.1.

2. 2 Brief Introduction of each feature

Some data can be similarly processed where the description will be stated together. Most of them are used or partially used for model construction while some features may not have strong correlations with the target which will be discarded. Information is provided in the table below. Processing details will be provided in Section 3.

<table>
<tr><th>Dataset</th><th>Description</th></tr>
<tr><td>Audio</td><td rowspan="2">The datasets for audio and activity are similar can be processed in a similar way as two data sets have the same form of data, which is a certain inference that was detected at certain timestamps.

According to specifications (Studentlife.cs.dartmouth.edu, 2019), there are four inferences that can be found: stationary, walking, running and Unknow. There are three inferences for audio that can be found: silence, voice, noise and unknown. However, the inference “unknown” mentioned in the dataset was clarified not occurring in the dataset.

It should be noticed that the sensing app will generate activity inference every 2- 3 seconds during the continuous 1 minute and then pause for 3 minutes, which indicates any interval larger than 1 minute could not be taken into consideration. The audio and activity data can normally reflect the impact on mental health and physical health.</td></tr>
<tr><td>Activity</td></tr>
<tr><td>Conversation</td><td rowspan="3">There are two fields in conversation data files: conversation start timestamp and conversation end timestamp. The duration of students’ conversation time generally reflects social performance.

Similarly, there are two fields in Phone-lock, Phone-charge as well as light data files: start time and end time. The light data files record when the phone was in a dark environment for a significant long time (>=1 hour) which is connected to one’s</td></tr>
<tr><td>Phone Lock</td></tr>
<tr><td>Phone Charge</td></tr>
</table>

Light	sleeping time and it is considered to be associated with mental health. The Phone-lock data was recorded when the phone was locked for a significant long time (>=1 hour) which can help to clearly describe students' daily performance in study and leisure time. The Phone-charge data was recorded when the phone was plugged in and charging for a significant long time (>=1 hour) which contributes to one's schedule arrangement.
GPS Location	GPS coordinates were collected every 10 minutes. Data fields include time, provider, network type, latitude, longitude, altitude, bearing, speed, travel-state, accuracy. In fact, all these fields are hardly associated with mental health and some fields are used before. For example, travel-state includes two inference "stationary" and "running" which are related to activity & audio dataset. Therefore, GPS Location is not taken into consideration.
Bluetooth	Its' important fields are time, MAC, class_id and level. It is considered that the number of Bluetooth that one student connects to other students reflects one's social performance. Bluetooth scans every 10 minutes.
Wi-Fi	The Wi-Fi fields are time, BSSID, frequency, level. There are two fields in the Wi-Fi location: time and location.
Wi-Fi Location	Wi-Fi scans frequently and Dartmouth Network Services provides Dartmouth College's Wi-Fi AP deployment information which helps to calculate a participant's on-campus rough location. It is considered that the number of active-area of Wi-Fi locations per day will be calculated. The number is associated with mental health and study time.

3. Methods

3. 1 Detailed extraction of each feature

There are 23 original features and 6 original sets of target values extracted from the whole datasets which can be grouped in a CSV file as shown in Appendix 3 where the first 23 columns are features and last 6 columns are targets, corresponding to each student with different Uid in the leftmost column. Details of the method for extraction will be stated separately in the same sequence as the table in Section 2.2 from Section 3.1.1 to Section 3.1.5.

3.1.1 Audio and Activity:

In consideration of the duration of Audio and activity, it can be calculated by extracting separate intervals and sum up by classification using the default dictionary of python. Each inference was regarded as a key in the dictionary where the values are the summation of every interval of different inferences. It should be noticed that there are intervals of 3 minutes that are not included since every interval longer than 3 minutes is pause therefore ignored during collecting data.

In terms of the occurrences of each inference, it is considered that there is a connection between the counting of different inferences and the students' mental health. It can be calculated by using the default dictionary of python and each inference is regarded as a key in the dictionary where the values represent the occurrences of each inference.

3.1.2 Conversation & Phone Lock & Phone Charge & light:

It is discussed in group that these four features have a similar data structure, therefore there is a simple way to extract the needed data: calculate all the end-time minus start-time timestamp and add them together which represents one student's total time spending on these four features during the whole experiment time. It is discussed later that this data can't reflect the daily performance of each student in terms of these features, which is more reasonable to contribute to the final result, that's why the average time of each student spending on these features per day is used.

3.1.3 Bluetooth

Firstly, a timestamp is transferred to date and time and is grouped by day. Secondly, the count of class_id field represents the number of Bluetooth connecting to other portable devices at the same timestamp which is used to calculate the number of Bluetooth that every student connects to other devices per day. It is calculated by using a list and set in python to delete the repeated IDs.

3.1.4 Wi-Fi Location

Dartmouth Campus Map shown in Appendix 1 is used to find every mentioned location in Wi-Fi-Location features. As for the location details, it is divided into two classes: 'in' location and 'near' location. It is assumed that the 'near' location implies students are on their way to the final destination. Thus, only 'in' location data is extracted as features. After programming, there is a list of 100 names of locations mentioned and these locations are divided into different classifications as study, recreation, daily activity and others depending on their general functions. The next step is to calculate every student average occurrence in these four types of locations per day which can be helpful to measure their study period and mental health.

3.1.5 Min-Max pre-processing

Also known as min-max scaling or min-max normalization, is the simplest method and consists of re-scaling the range of features to scale the range in [0, 1] or [−1, 1]. Selecting the target range depends on the nature of the data. The general formula for a min-max of [0, 1] is given in Appendix 2. This pre-processing method is used to deal with all the X features which are already extracted from the Student-Life dataset.

3.2 Primary pre-processing and analyzing of features

In order to primarily check the distribution of the data, the scatter plot is used for each feature data after normalization. As shown in Appendix 2, it can be witnessed that nodes in some features are not evenly diverging e.g. “Feature Others” extracted from Wi-Fi locations, which reflects that this king of a feature may influence the correct model construction. Using the same way, 7 features including activity_time0, activity_time1, activity_time3, activity_count0, activity_count1, activity_count3 and others from Wi-Fi locations were discarded.

3. 3 Pre-processing of target values from survey data

As for the survey data, it has 3 scores for each measure. The flourishing score gives one measure and panas (Positive and Negative Affect Schedule) score includes positive and negative scores. The pre-processing steps of survey data are as follows:

- Step 1: It is obvious that there are several missing values in the output files, which means a method should be used to complete the empty data. In order to get the final score, he average score was calculated for each question and multiplies the total question numbers.

- Step 2: As shown in Appendix 3, there are two students with Uid as u25 and u41 who did not answer the questionnaire at all, which is going to be deleted in the survey data. For the panas score, positive answers and negative answers should be distinguished and the sum of each score should be calculated separately.

- Step 3: Most students have two types of scores (pre and post) from questionnaires, while others only finished one questionnaire. In order to get an accurate evaluation, missing values were expected to be the average score of pre and post scores which will not impose much influence. If there is only one score for a student, the only score will be the average score.

- Step 4: After calculating the average score of each student(panas_positive, panas_negative and flourishing score), the score will be sorted in ascending order to get the median value and scores will be divided into two groups("high" vs "low"), which represented by 1 and 0. If the score equals the median value, it was regarded as being in the "high" group.

3. 4 Presentation of implemented models & Evaluate Metrics

3.4.1 Implemented Models

Method	Logic & Design Choices	Time Complexity
Random Forest	1. Random Forest is an ensemble model and is believed to have better performance over lots of other models. 2. Random Forest uses the general technique of bootstrap aggregating as its training algorithm, which can prevent overfitting compared to Decision Tree. 3. Random Forest does not need the feature-selection process and it can provide the importance of each feature with feature_importances_ function. This list can be helpful to get a better understanding of the processed dataset when choosing other training-models. 4. This dataset has less than 30 features and may have imbalanced features, therefore it is considered that Random Forest can relatively	O(M(m* n *log n)) Assuming there are n instances, m attributes and M trees growing in a Random Forest model

	bring out a good result.	
KNN	1. It is easy to achieve the goal of classification and the algorithms of KNN is simple. 2. KNN can have great tolerance on the noisy data since the only provided data will be trained and no extra data needed to define the model. 3. Only a few nearest data will be considered during making classification. KNN is a non-probabilistic supervised learning algorithm that can reduce the impact of inequality on data collection. 4. The classification is directly achieved by using the relations from samples which can significantly reduce the error of inappropriate feature selection, providing more independence to the classification. The Hassanat distance function can be used as it has the tested to have best performance over all common distance functions (Arxiv.org, 2019).	O(n * k) Assuming there are n instances, k represents the number of feature dimensions. Overall, the complexity of this model is relatively low as the features are limited.
Logistic Regression	1. Logistic Regression is designed to easily deal with binary-classification problems which are suitable for the processed dataset. 2. It has a small computational demand.	O(n * k) Assuming there are n instances, k represents the number of feature dimensions. Overall, the complexity of this

		model is relatively low as the number of features are limited.
SVM	1. It is easy to solve the machine learning problem which can avoid overfitting and can deal with a small amount of sample data. The input data of this project is around 45 samples, which means it is suitable for the SVM model. 2. It can be applied to high-dimensional. The dataset has less than 30 features, which means SVM is a good model to fit. 3. The more features the dataset has, the margin will be larger, and the generalization will be better.	O(n * k) Assuming there are n instances, k represents the number of feature dimensions. Overall, the complexity of this model is relatively low as the number of features are limited

3.4.2 Evaluate Metrics

The function named sklearn.model_selection.cross_val_score is used in all four models as required which is an essential way to prevent overfitting and get as much information as possible from the raw data. The parameter cv for all models was set as 10 therefore 10 folds of cross-validation are used.

The function named sklearn.metrics.classification_report is used to evaluate all four models which include three important indices: recall, precision and F-Score. Formulae of these evaluate measures are provided in Appendix 4. It is discussed that using the same evaluation methods to assess all models is helpful when comparing the performance among different models.

4. Results

4.1 Original results and evaluation

Table 4.1 shows all original results of accuracy, which was selected as the main tool to select model, based on cross validation before hyper-parameter tuning.

Table 4.1: Original results of accuracy

	Panas_Positive	Panas_Negative	Flourishing
Random Forest	0.665	0.593	0.520
KNN	0.683	0.622	0.652
Logistic Regression	0.667	0.685	0.643
SVM	0.725	0.613	0.520

According to Table 4.1, SVM has is the outstanding model on panans_Positive but performs not as well as Logistic Regression on other scores. The model of KNN has an average performance on all scores. Random Forest is a good model on Panas_Positve score in but has unsatisfactory results on others. Generally, among these models, Logistic Regression is the best model since it can remain at a good performance on all scores and using feature selection method can strengthen its high-level performance.

4.2 Feature Selection

4.2.1 Feature Importance Criteria

For the best model logistic regression, reclusive feature eliminator cross-validation was used to process feature selection. The change of number selected along with the cross-validation accuracy score for each target values can be plotted as Figure 4.2.1. By using this result as well as specific features sort using functions of rfecv.support_ and rfecv.sranking_, and according to the rules of selecting fewer features but achieving the optimal effect, Table 4.2.1 can be achieved to show specific features to achieve the optimal accuracy. It should be mentioned that for Panas_N (the red line in Figure 4.2.1), the optical number is 1 but in practice, one feature is less reasonable to define a model, therefore, the second optimal number 3 was used for the future model optimization.

As mentioned in section 3.4.1.1, it is not necessary to do the feature-selection in Random Forest as using more features will bring out a better result. As for logistic regression and support vector machine, recursive feature elimination and cross-validated (RFECV) was used. KNN model was tested has an unreasonable score and feature selection would increase this irrationality, under testing of many methods of feature selection, no obvious results can be achieved.

Figure 4.2.1 Cross validation score for different number of selected features

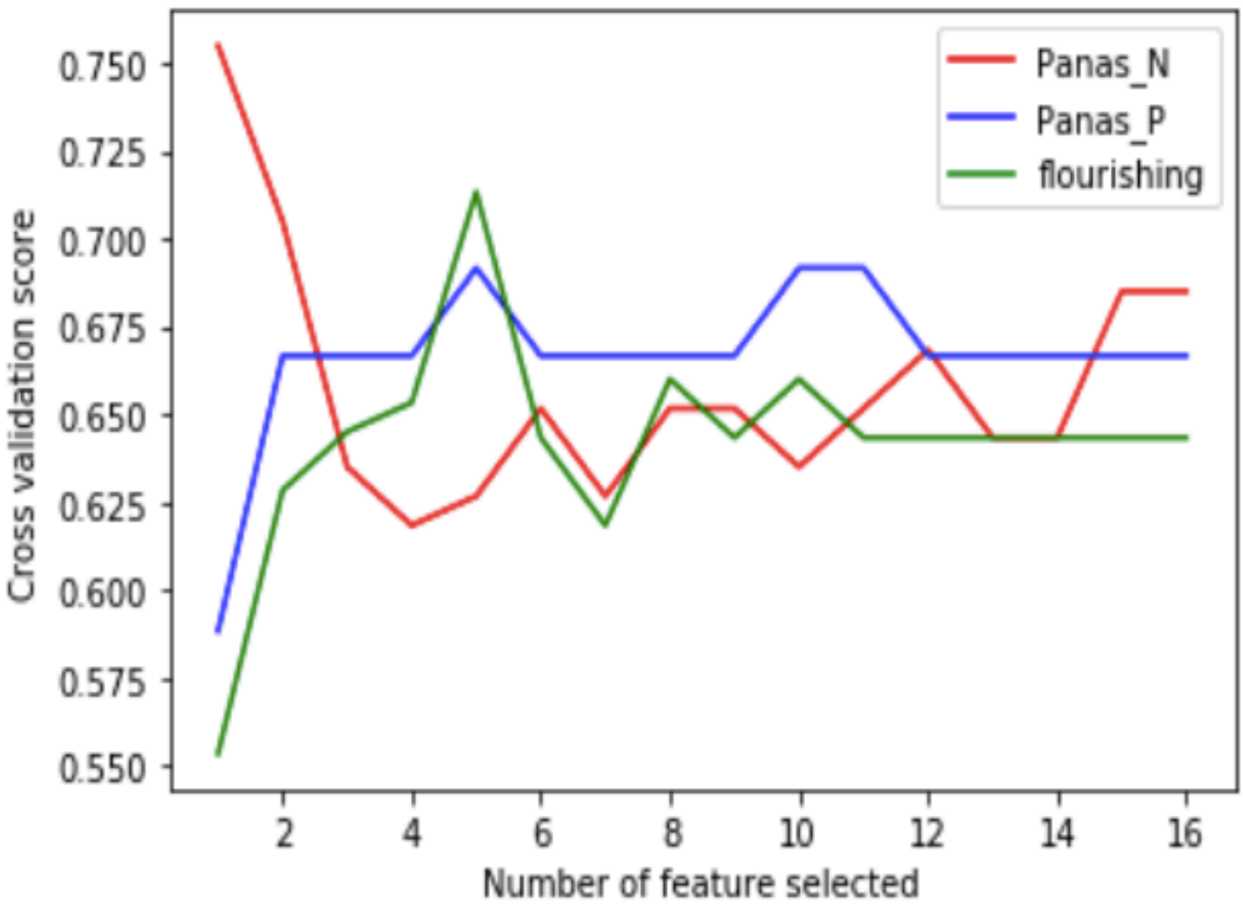


Table 4.2.1 The optical number and specific feature selected for different target values

	Pans_N	Pans_P	Flourishing
Number selected	3	5	5
Features selected	bluetooth, audio_count_2, audio_count_2	bluetooth, activity_time2 , audio, daily_activity, recreation	audio_count_2, audio_time_1, audio_time_2, daily_activity, recreation

As for other models, results in figures of SVM is provided in Appendix 7 buy using the same function of RFECV, feature selection of KNN model was analyzed by principal component analysis (PCA) pipelined with grid search, selected features were formed into a new group of features as dimensionality reduction. In terms of random forest, due to the algorism of random forest, the evaluation metrics are not stable and by testing lots of time, the range of the results is large which means that it is not necessary to use evaluate metrics to assess the performance of Random Forest Model.

4.2.2 Results After Feature Selection

By using the specific features selected as mentioned in Section 4.2.1, the logistic regression model was constructed again. Table 4.2.2 shows the results of accuracy after feature selection, compared with Table 4.1, noticeable increases can be found in for all target values which indicates the feature selection is reasonable and perform well on this model.

Table 4.2.2 Results after Feature Selection

	Panas_Positive	Panas_Negative	Flourishing
Random Forest	0.665	0.593	0.520
KNN	0.683	0.622	0.652
Logistic Regression	**0.692**	**0.738**	**0.713**
SVM	0.725	0.613	0.520

4.3 Hyperparameter Tuning and Evaluation Metrics

The hyperparameter tuning for each model after feature selection was performed by using Bayesian optimization and processes with the best parameters are provided in Appendix 7. Table 4.3 is a summary of accuracy for all models related to different targets.

Table 4.3: Results after Feature Selection and Hyperparameter Tuning

	Panas_Positive	Panas_Negative	Flourishing
Random Forest	0.733	0.710	0.600
KNN	0.700	0.655	0.730
Logistic Regression	0.692	0.730	0.713
SVM	0.725	0.680	0.673

5. Discussion

5.1 Comparison of all models

Among Recall, Precision, Accuracy, it is discussed that Accuracy is the most important index. For this binary classification model, the reason to choose accuracy from all evaluation metrics as the main standard mainly due to the fact that compared with precision and recall, the accuracy is a more general evaluation score which can take all samples into consideration, therefore, no biased focus will be introduced.

Specific ranking for all models in terms of different survey results concluded from Table 4.3 is as followed (">" represents better than):

- Panas_negative survey: Logistic Regression > Random Forest(ranges from 50% to 70%)> SVM > KNN(could be overfitting)
- Panas_positive survey: Random Forest (ranges from 50% to 70%)> SVM > KNN(could be overfitting) > Logistic Regression
- Flourishing: KNN (could be overfitting) > Logistic Regression > SVM > Random Forest (ranges from 50% to 70%).

5.2 Advantages & Disadvantages

The advantages of models are mentioned in Section 3.4 and after analyzing, different models have their own disadvantages in certain fields. The range of the results of Random Forest is too large due to the small size of the processed features which means that the results are relatively unreliable. The SVM model takes up too much calculate space which is not profitable. The accuracy of the KNN model reaches even up to 80% precision as stated in Appendix 7 before feature selection, which means it is likely for it to become overfitting in some cases. In some cases, the logistic regression model could be underfitting.

5.3 Findings based on results

The average performance of the logistic regression model is better than other models on three surveys. As mentioned in section 4, it is cleared that features including Bluetooth, Daily-Activity, Audio, Recreation are the most important features contribute to the results. This is reasonable since features are important and more representative to quantify the social activities a student participates in. It indicates that a student who frequently talks with others and socializes in different places with different people is more likely to own a positive attitude towards life.

6. Conclusion

The datasets from the sensing app can be processed and selected to build models for predicting mental health. The model of logistic regression is reasonable and performs well on original results and can achieve higher accuracy by feature selection and hyperparameter tuning, although it could not be ignored that the better model can be built if more users could be involved in data collection and higher competency can be achieved. According to the results, it can be predicted that social ability and social performance significantly affect the student’s mental health. Students should be encouraged to spend more time on social activities.

7. References

En.wikipedia.org. (2019). Unix time. [online] Available at: https://en.wikipedia.org/wiki/Unix_time [Accessed 25 Nov. 2019].

Home.dartmouth.edu. (2019). [online] Available at: https://home.dartmouth.edu/sites/home/files/campus_map_11x17.pdf [Accessed 25 Nov. 2019].

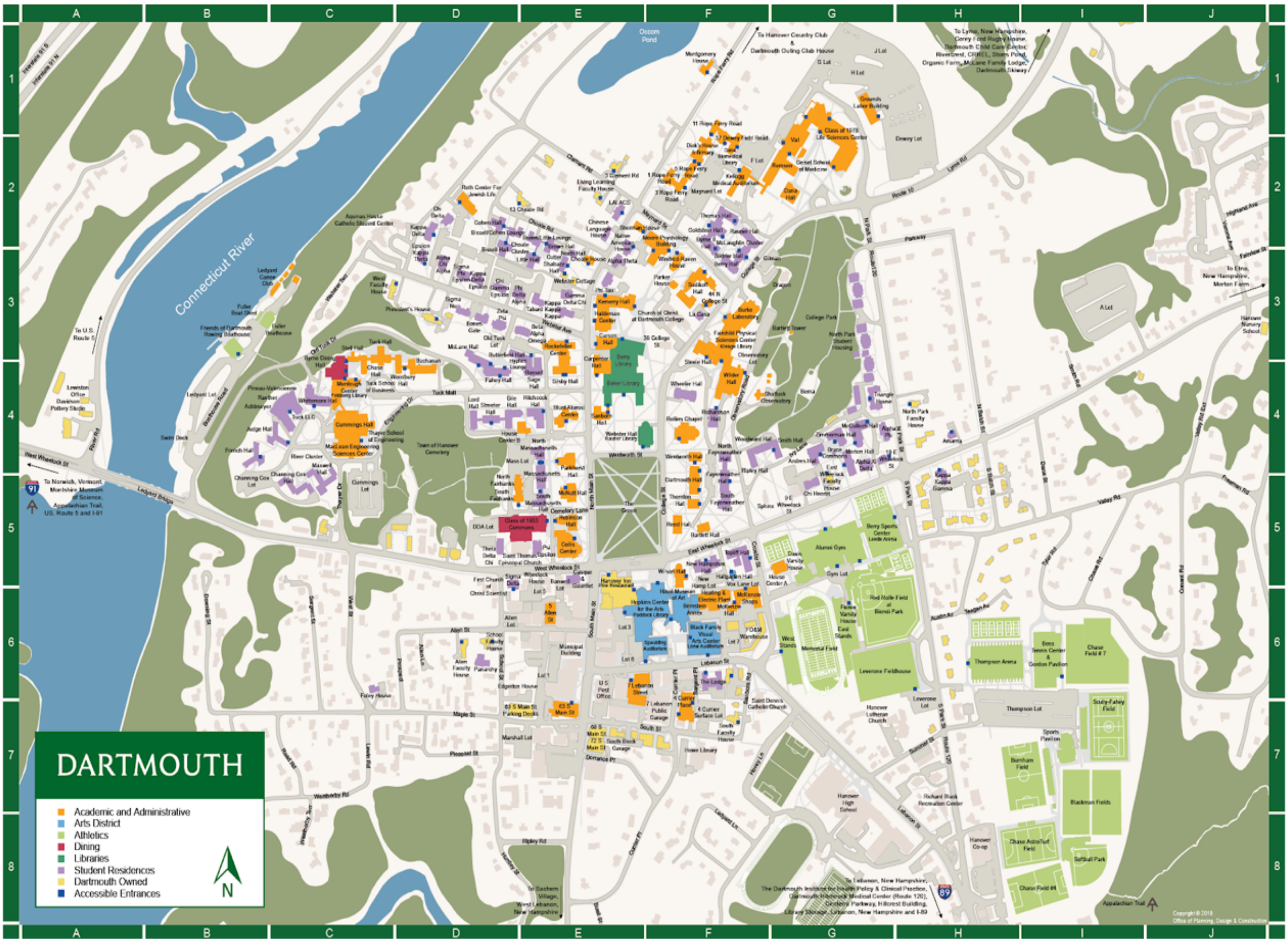
Mary Ann Liebert, Inc., publishers. (2019). Effects of Distance Measure Choice on K-Nearest Neighbor Classifier Performance: A Review | Big Data. [online] Available at: https://www.liebertpub.com/doi/abs/10.1089/big.2018.0175 [Accessed 25 Nov. 2019].

Studentlife.cs.dartmouth.edu. (2019). StudentLife Dataset. [online] Available at: https://studentlife.cs.dartmouth.edu/dataset.html [Accessed 25 Nov. 2019].

Wang, Rui, Chen, Fanglin, Chen, Zhenyu, Li, Tianxing, Harari, Gabriella, Tignor, Stefanie, Zhou, Xia, Ben-Zeev, Dror & Campbell, Andrew 2014, 'StudentLife: assessing mental health, academic performance and behavioral trends of college students using smartphones', in Proceedings of the 2014 ACM International Joint Conference on pervasive and ubiquitous computing, ACM, pp. 3–14.

8. Appendices

Appendix 1: Figure of Dartmouth College Map



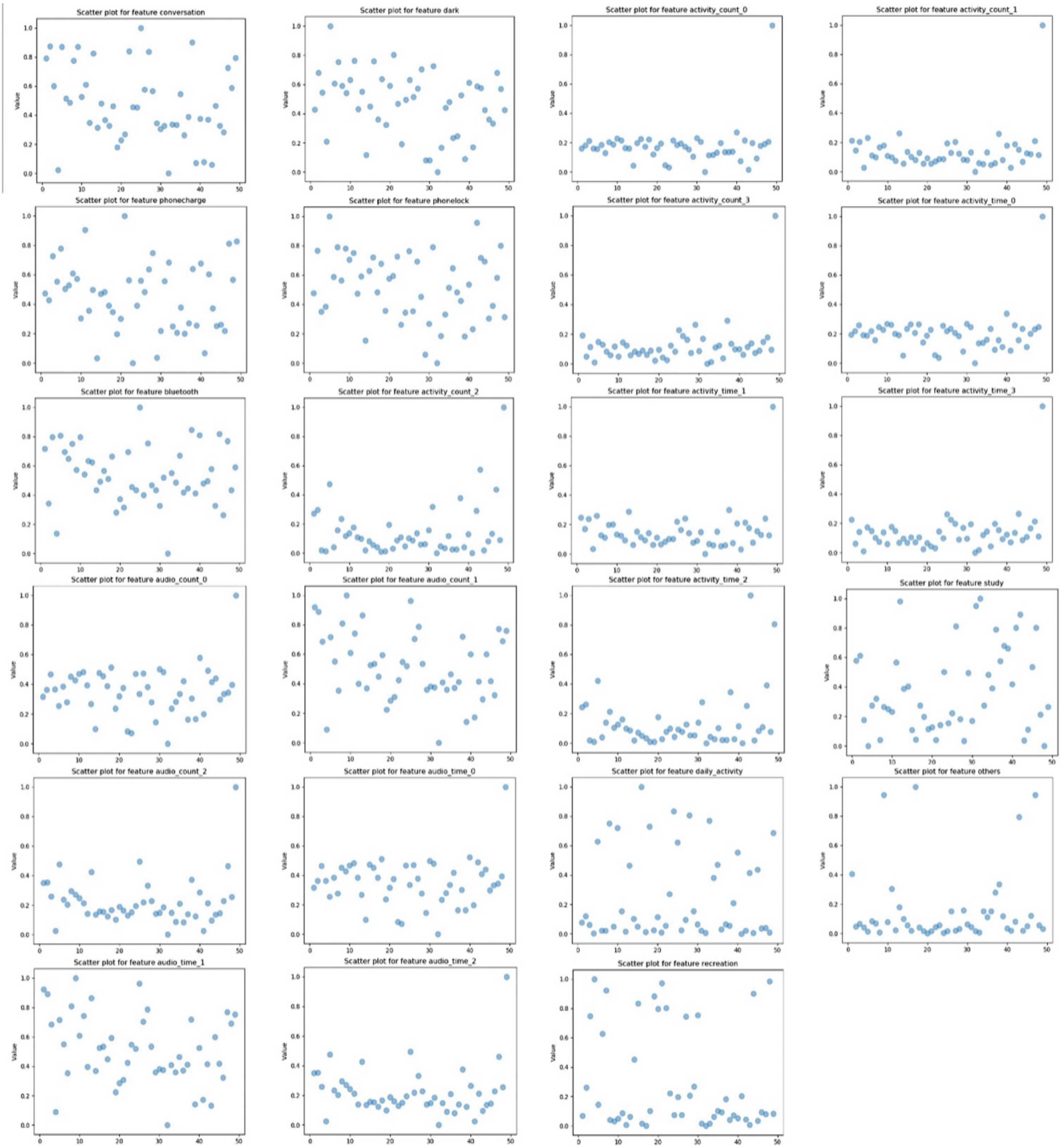
Appendix 2: Minmax function used for normalisztion

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

Appendix 3: All extracted data collected in a CSV file

	conversation	dark	phonecharge	phonelock	bluetooth	activity_count_0	activity_count_1	activity_count_2	activity_count_3	activity_time_0	activity_time_1	activity_time_2	activity_time_3	audio_count_0	audio_count_1	audio_count_2	audio_time_0	audio_time_1	audio_time_2	f_preP	f_preN	f_postP	f_postN	p_pre	p_post
u00	6.753	7.431	4.756	8.981	9.417	403461	30659	8990	18881	1012199	84128	23316	48491	1016780	938101	594677	971340	903564	570633	32	30	31	24	48.047	45
u01	7.34	10.968	4.393	13.6	6.172	438583	20954	9778	5547	1096955	57979	25038	14707	1100677	909899	597906	1048520	876171	574502	30	19	27	17	45	46
u02	5.385	9.086	6.649	6.985	10.104	485697	29510	678	11633	1215516	80742	1886	31008	1286966	715488	441940	1221711	687630	423999	23	15	28	13	46	44
u03	1.255	4.381	5.359	7.513	4.39	404853	4623	423	1503	1002408	12597	1099	3993	1106665	145978	58403	1047290	139998	55588	30	27	25	20	34	31
u04	7.322	15.459	7.039	17.312	10.172	398779	33114	15733	14890	994693	88322	40234	37622	909387	745735	801609	864695	715645	766664	27	14	34	19	27	33
u05	4.775	9.934	4.974	10.772	9.219	441160	16144	1400	12943	1094029	43775	3857	32657	1140455	587289	406018	1084717	563716	386786	32	27	33	17	48	50
u07	4.584	11.989	5.173	13.973	8.837	359551	14550	5226	8689	896360	39285	13417	22826	951672	398870	353014	902722	382833	338311	30	21	16	17	49	47
u08	6.644	9.704	5.761	10.377	9.71	470389	24232	7794	6308	1177780	68125	20493	16855	1259405	832152	501055	1199320	800311	481251	30	26			38.756	
u09	7.327	9.044	5.49	13.839	8.148	447910	25739	3910	11879	1118022	68647	10179	30456	1217147	1015423	460328	1158728	975249	440891	28	11	32	12	46	47
u10	4.858	10.287	3.477	12.596	10.091	505222	15897	4553	5487	1253576	45916	12055	14186	1291459	642022	420314	1227217	615801	400975	33	20	43	28	39	39
u12	5.446	12.138	7.99	13.359	7.896	490507	14629	5869	14372	1226087	42390	15393	38116	1316049	768945	365760	1251490	738807	350295	42	18			49	
u13	3.594	7.454	3.872	8.931	8.689	411622	11039	3612	12437	1032276	32612	9486	32135	1154904	443415	249232	1087097	421611	235925	13	21			44	
u14	6.997	9.149	4.94	10.791	8.607	404004	37582	3317	6212	1005840	97191	8435	15966	926932	885091	715100	882559	851000	686546	40	43	36	12	52	53
u15	3.339	3.097	1.435	3.847	6.957	228410	8207	681	8598	568226	21830	1772	21685	628352	412700	243288	598258	396513	233028	30	25	33	25	43	48
u16	4.535	7.715	4.734	11.37	7.475	461341	19653	2668	7087	1143032	52324	6780	15763	1301039	562036	276382	1235653	540181	264950	18	20	17	33	42	41
u17	3.719	12.097	4.814	12.83	8.106	500153	14980	1897	9218	1239388	39447	4855	23253	1266863	571437	270499	1201940	548705	258111	31	17	27	37	37	38
u18	3.426	6.498	4.124	9.078	7.638	424135	11704	1352	6585	1052530	32155	3462	16738	1141423	488628	219406	1084088	469139	209957	16	28	17	35	37	
u19	4.39	10.361	3.795	12.155	8.955	497301	18658	361	9299	1237667	49108	899	23298	1367062	626762	292077	1299060	602784	279950	31	22	34	15	42	42
u20	2.383	5.945	2.656	7.084	5.633	342723	8284	422	2769	849200	21824	1104	7233	874622	274848	183884	830422	263858	175007	29	13	25	26	45	45
u22	2.731	9.694	3.453	10.566	6.439	409639	13632	6462	9765	999317	39164	16913	15390	1022557	334951	328110	970195	321683	313636	26	15			46	
u23	3.013	12.674	8.723	10.839	5.931	453871	8613	1015	4831	1121458	23743	2647	10707	1121267	356095	285100	1068143	341341	269867	26	25	28	24	35	42
u24	7.098	7.97	5.421	12.95	9.219	233511	10438	2886	3006	582894	29348	7477	8102	600507	466501	230689	570202	448071	221258	26	15	22	18	41	43
u25	4.349	4.11	1.18	5.595	7.159	209138	12750	3583	12287	521607	35233	9366	31568	578282	584153	265330	550807	561035	254308						
u27	4.34	8.396	4.13	6.918	6.97	486447	12892	1620	8711	1209994	35928	4206	22276	1291467	556219	335280	1228359	534933	321718	23	18	19	23	31	31
u30	8.247	10.258	5.409	13.522	11.869	441052	28153	3354	22286	1096479	74951	8845	56079	1051896	980034	830169	1001444	941521	795481	34	33	34	21	52	56
u31	5.22	8.644	4.828	7.045	6.667	456379	18930	2875	18513	1146292	55934	7564	48454	1296519	732750	376038	1234778	703995	360625	31	17	20	37	16	36
u32	7.082	9.439	5.969	12.434	9.741	423133	29127	4483	16204	1056779	81699	12184	42542	1131508	811151	561486	1076278	779702	539329	29	14	41	15	54	56
u33	5.144	11.29	6.819	8.607	7.261	395212	18132	1999	7797	986481	49551	5175	20179	949326	570334	392091	903375	549950	376816	19	34	26	28	31	28
u34	3.567	2.561	1.457	2.375	6.975	322269	12531	2054	25727	652811	27804	5168	36680	709031	404955	248599	674004	388695	237816	28	10	24	14	49	23
u35	3.276	2.566	2.837	5.678	6.03	510084	11688	5271	9034	1244730	31397	13459	21413	1350267	424582	259764	1279675	407446	248440	27	12	20	15	48	44
u36	3.424	11.614	5.386	13.982	7.719	474728	19311	10478	16639	1180238	51029	26506	42066	1312791	417470	323688	1246613	400643	310080	32	11	32	19	46	46
u39	1.103	1.435	6.33	1.423	3.208	162565	517	32	721	401519	1431	79	1820	449477	60265	18488	426497	57732	17650	20	10			15	
u41	3.501	3.767	3.054	4.341	7.981	336339	8889	1609	2066	835191	24110	4271	5456	872656	449938	260032	828597	432003	248596						
u42	3.484	7.598	2.735	6.698	7.404	341580	8003	1126	11120	840080	20759	2903	26569	956564	404623	162545	908563	389288	156961	24	29	36	21	45	16
u43	4.98	8.169	4.029	9.57	9.018	364531	19422	3870	12475	908861	52054	9880	32043	1050924	503667	362523	1001754	484565	347732	31	18	32	21	47	42
u44	2.964	4.704	2.685	11.689	6.839	459497	7283	903	4181	1138072	19606	2256	10535	1206021	414365	153521	1144717	398347	146145	32	11	28	22	48	41
u45	3.863	4.89	3.221	9.08	7.067	368962	8661	901	28352	697193	20838	2220	42820	743268	453971	243372	707466	436522	233413	27	23	27	18	48	49
u46	7.537	8.808	6	8.176	10.527	367770	36947	12561	13649	894383	101406	32828	33791	992983	747337	628819	943266	717619	605551	34	19	31	34	42	44
u47	1.618	2.683	3.097	4.31	6.766	374324	12077	1324	10056	755879	26547	2737	20965	745869	196514	221741	706293	188208	211091	34	31	35	11	50.425	52
u49	3.791	9.995	6.281	9.933	10.214	572515	25929	4266	10031	1476187	70375	10973	27045	1483761	633415	489563	1323879	538841	432341	35	18	34	16	51	55
u50	1.659	3.829	1.693	5.111	7.368	274127	4536	84	6639	677993	11967	228	16500	806139	224082	61855	764622	214759	59073	25	11			48	
u51	3.747	9.691	5.717	16.593	7.484	489705	27029	9591	11589	1215909	72583	24028	29891	1333036	456827	365261	1265832	438481	349052	24	14	26	12	40.047	48
u52	1.532	9.502	3.997	12.788	8.212	186605	21951	18918	13987	896856	60895	95539	56230	1193196	342474	176358	1127272	180117	167189	40	35	26	39	34	37
u53	4.41	7.38	3.095	12.427	6.03	458617	9991	678	7766	1138049	27915	1793	19961	1240274	633120	241413	1178318	607937	230810	28	17	33	17	50	49
u54	3.436	6.483	3.143	6.246	10.286	301196	18317	2733	9112	756679	53698	8072	24316	984389	459977	254131	937356	442585	243873						
u56	3.124	6.092	2.823	7.65	5.475	432927	17574	4375	14822	1038974	44845	10347	35981	1051230	369141	392504	997140	354059	375001	22	9	29	15	46	48
u57	6.28	10.982	7.31	10.676	9.864	449404	30212	14446	17584	1128789	82333	37416	45769	1069543	795674	777514	1015031	763030	742900	35	12			50	
u58	5.291	9.392	5.457	14.116	6.968	476456	16549	2957	9793	1182837	44321	7604	24853	1159024	719280	436902	1100072	691671	419655	28	15			51	
u59	6.766	7.408	7.426	6.388	8.317	1674611	142134	33104	95518	3566306	337600	76939	208583	2241609	785090	1657431	2139078	748072	1588113	34	21	37	29	43	50

Appendix 4: Scatter plot s for all features after normalization



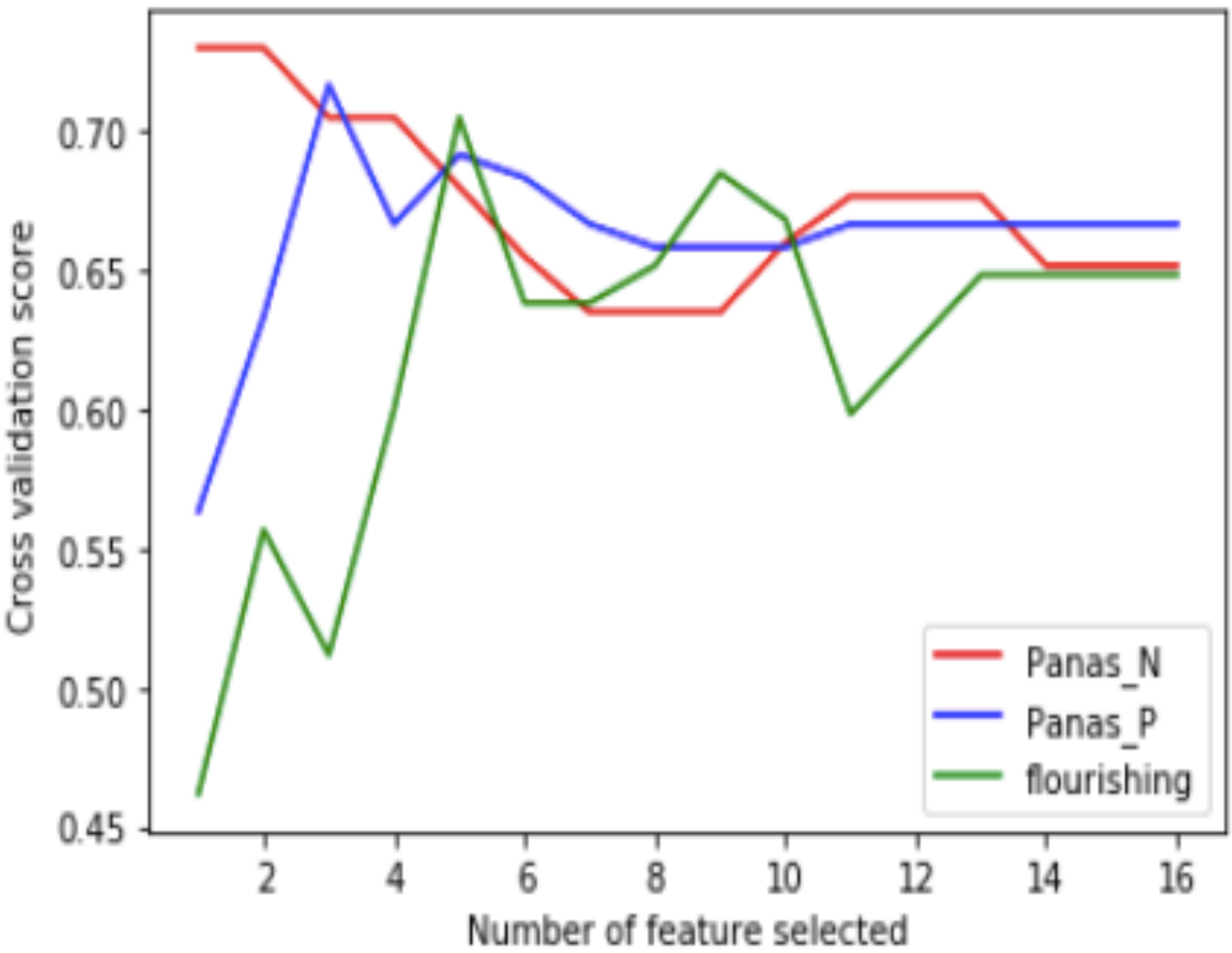
Appendix 5: Formulae of metrics

$$\text{recall} = \frac{|\{\text{relevant documents}\} \cap \{\text{retrieved documents}\}|}{|\{\text{relevant documents}\}|}$$

$$\text{precision} = \frac{|\{\text{relevant documents}\} \cap \{\text{retrieved documents}\}|}{|\{\text{retrieved documents}\}|}$$

$$F = \frac{2 \cdot \text{precision} \cdot \text{recall}}{(\text{precision} + \text{recall})}$$

Appendix 6: Figure of Cross validation score for different number of selected features



Appendix 7: Hyperparameter Tuning and Evaluation Metrics

<table>
<tr><th>Model</th><th>Hyperparameter Tuning</th><th>Evaluation Metrics</th></tr>
<tr><td rowspan="5">Random Forest</td><td>Flourishing:
<table>
<tr><th>iter</th><th>target</th><th>max_depth</th><th>max_fe...</th><th>min_sa...</th><th>n_esti...</th></tr>
<tr><td>1</td><td>0.56</td><td>13.55</td><td>0.465</td><td>17.15</td><td>11.59</td></tr>
<tr><td>2</td><td>0.5967</td><td>11.82</td><td>0.6768</td><td>11.48</td><td>233.0</td></tr>
<tr><td>3</td><td>0.5433</td><td>11.35</td><td>0.3318</td><td>22.02</td><td>155.9</td></tr>
<tr><td>4</td><td>0.5517</td><td>6.045</td><td>0.5026</td><td>9.975</td><td>31.84</td></tr>
<tr><td>5</td><td>0.5717</td><td>12.42</td><td>0.5315</td><td>15.65</td><td>220.5</td></tr>
<tr><td>6</td><td>0.585</td><td>5.328</td><td>0.3074</td><td>24.85</td><td>249.3</td></tr>
<tr><td>7</td><td>0.5917</td><td>5.149</td><td>0.1322</td><td>2.029</td><td>248.1</td></tr>
<tr><td>8</td><td>0.5817</td><td>15.0</td><td>0.1</td><td>2.0</td><td>80.77</td></tr>
<tr><td>9</td><td>0.5233</td><td>14.83</td><td>0.3986</td><td>24.65</td><td>68.52</td></tr>
<tr><td>10</td><td>0.555</td><td>14.9</td><td>0.2854</td><td>2.105</td><td>249.5</td></tr>
<tr><td>11</td><td>0.5633</td><td>5.0</td><td>0.999</td><td>2.0</td><td>128.7</td></tr>
<tr><td>12</td><td>0.4533</td><td>15.0</td><td>0.999</td><td>2.0</td><td>10.0</td></tr>
<tr><td>13</td><td>0.58</td><td>15.0</td><td>0.999</td><td>2.0</td><td>170.3</td></tr>
<tr><td>14</td><td>0.5283</td><td>5.0</td><td>0.1</td><td>25.0</td><td>10.0</td></tr>
<tr><td>15</td><td>0.5883</td><td>5.0</td><td>0.999</td><td>2.0</td><td>192.1</td></tr>
<tr><td>16</td><td>0.5633</td><td>15.0</td><td>0.999</td><td>25.0</td><td>31.53</td></tr>
<tr><td>17</td><td>0.585</td><td>5.0</td><td>0.1</td><td>2.0</td><td>73.31</td></tr>
<tr><td>18</td><td>0.6</td><td>5.032</td><td>0.3127</td><td>2.217</td><td>225.3</td></tr>
<tr><td>19</td><td>0.555</td><td>5.0</td><td>0.1</td><td>2.0</td><td>163.7</td></tr>
<tr><td>20</td><td>0.5633</td><td>5.125</td><td>0.1882</td><td>24.93</td><td>110.2</td></tr>
</table></td><td>None</td></tr>
<tr><td colspan="2">{'target': 0.6, 'params': {'max_depth': 5, 'max_features': 0.313, 'min_samples_split': 2, 'n_estimators': 225}}</td></tr>
<tr><td><table>
<tr><th>iter</th><th>target</th><th>max_depth</th><th>max_fe...</th><th>min_sa...</th><th>n_esti...</th></tr>
<tr><td>1</td><td>0.6733</td><td>14.34</td><td>0.2717</td><td>19.51</td><td>236.9</td></tr>
<tr><td>2</td><td>0.6817</td><td>9.222</td><td>0.827</td><td>22.21</td><td>64.3</td></tr>
<tr><td>3</td><td>0.5767</td><td>11.95</td><td>0.7624</td><td>5.312</td><td>111.3</td></tr>
<tr><td>4</td><td>0.6483</td><td>5.461</td><td>0.2547</td><td>21.24</td><td>31.62</td></tr>
<tr><td>5</td><td>0.6433</td><td>9.445</td><td>0.5561</td><td>13.56</td><td>30.84</td></tr>
<tr><td>6</td><td>0.6017</td><td>5.0</td><td>0.999</td><td>2.0</td><td>250.0</td></tr>
<tr><td>7</td><td>0.7017</td><td>5.0</td><td>0.999</td><td>25.0</td><td>250.0</td></tr>
<tr><td>8</td><td>0.71</td><td>15.0</td><td>0.999</td><td>25.0</td><td>10.0</td></tr>
<tr><td>9</td><td>0.64</td><td>15.0</td><td>0.1</td><td>25.0</td><td>250.0</td></tr>
<tr><td>10</td><td>0.7017</td><td>5.0</td><td>0.999</td><td>25.0</td><td>177.4</td></tr>
<tr><td>11</td><td>0.6567</td><td>5.0</td><td>0.1</td><td>25.0</td><td>212.3</td></tr>
<tr><td>12</td><td>0.7017</td><td>15.0</td><td>0.999</td><td>25.0</td><td>132.3</td></tr>
<tr><td>13</td><td>0.685</td><td>14.89</td><td>0.9661</td><td>19.45</td><td>11.42</td></tr>
<tr><td>14</td><td>0.6017</td><td>15.0</td><td>0.999</td><td>2.0</td><td>173.1</td></tr>
<tr><td>15</td><td>0.685</td><td>15.0</td><td>0.999</td><td>25.0</td><td>42.32</td></tr>
<tr><td>16</td><td>0.7017</td><td>5.0</td><td>0.999</td><td>25.0</td><td>140.2</td></tr>
<tr><td>17</td><td>0.7017</td><td>15.0</td><td>0.999</td><td>25.0</td><td>161.8</td></tr>
<tr><td>18</td><td>0.6567</td><td>15.0</td><td>0.2431</td><td>25.0</td><td>93.76</td></tr>
<tr><td>19</td><td>0.6017</td><td>15.0</td><td>0.109</td><td>2.0</td><td>61.3</td></tr>
<tr><td>20</td><td>0.7017</td><td>5.73</td><td>0.8601</td><td>24.61</td><td>235.5</td></tr>
</table>
Panas_Negative:</td><td>None</td></tr>
<tr><td colspan="2">{'target': 0.71, 'params': {'max_depth': 15.0, 'max_features': 0.999, 'min_samples_split': 25.0, 'n_estimators': 250}}</td></tr>
<tr><td>Panas_Positive:</td><td>None</td></tr>
</table>

```
|   iter    |  target   | max_depth | max_fe... | min_sa... | n_esti... |
-------------------------------------------------------------------------
|  1        |  0.6917   |  13.8     |  0.6799   |  6.731    |  149.6    |
|  2        |  0.6717   |  13.6     |  0.4726   |  5.122    |  220.6    |
|  3        |  0.6967   |  9.529    |  0.8391   |  16.27    |  238.3    |
|  4        |  0.6717   |  7.325    |  0.6863   |  19.83    |  66.56    |
|  5        |  0.6717   |  6.523    |  0.4605   |  24.49    |  45.57    |
|  6        |  0.6917   |  14.91    |  0.5239   |  24.99    |  176.7    |
|  7        |  0.6267   |  15.0     |  0.999    |  2.0      |  10.0     |
|  8        |  0.6833   |  5.032    |  0.2144   |  24.36    |  174.8    |
|  9        |  0.6717   |  13.75    |  0.7927   |  24.47    |  249.8    |
|  10       |  0.68     |  15.0     |  0.999    |  25.0     |  113.7    |
|  11       |  0.6583   |  5.269    |  0.9746   |  2.964    |  249.9    |
|  12       |  0.6467   |  5.0      |  0.7978   |  24.79    |  225.1    |
|  13       |  0.6217   |  5.0      |  0.999    |  25.0     |  10.0     |
|  14       |  0.6833   |  5.0      |  0.1      |  2.0      |  43.15    |
|  15       |  0.7083   |  5.0      |  0.1      |  2.0      |  109.1    |
|  16       |  0.6967   |  5.0      |  0.9989   |  24.87    |  136.2    |
|  17       |  0.6717   |  15.0     |  0.1      |  2.0      |  69.43    |
|  18       |  0.7133   |  14.98    |  0.21     |  6.545    |  246.9    |
|  19       |  0.7333   |  14.9     |  0.1247   |  24.13    |  146.7    |
|  20       |  0.6383   |  14.94    |  0.1029   |  2.676    |  117.3    |
=========================================================================
```

{'target': 0.733, 'params': {'max_depth': 14, 'max_features': 0.124, 'min_samples_split': 24., 'n_estimators': 146}}

KNN

```
|   iter    |  target   | n_neig... |
-------------------------------------
3
|  1        |  0.5967   |  3.617    |
9
|  2        |  0.68     |  9.271    |
10
|  3        |  0.705    |  10.45    |
5
|  4        |  0.6517   |  5.88     |
11
|  5        |  0.73     |  11.87    |
15
|  6        |  0.71     |  15.0     |
13
|  7        |  0.73     |  13.3     |
12
|  8        |  0.73     |  12.57    |
12
|  9        |  0.73     |  12.57    |
12
|  10       |  0.73     |  12.54    |
12
|  11       |  0.73     |  12.9     |
12
|  12       |  0.73     |  12.3     |
12
|  13       |  0.73     |  12.4     |
13
|  14       |  0.73     |  13.24    |
12
|  15       |  0.73     |  12.82    |
=====================================
```

Flourishing:

	precision	recall	f1-score
0	0.817	0.7	0.714
1	0.727	0.767	0.724
accuracy			0.73
macro_avg	0.772	0.73	0.717
weighted_avg	0.769	0.73	0.719

{'target': 0.73, 'params': {'n_neighbors': 11}}

```
|   iter    |  target   | n_neig... |
-------------------------------------
6
|  1        |  0.6217   |  6.021    |
6
|  2        |  0.6217   |  6.125    |
12
|  3        |  0.6633   |  12.28    |
13
|  4        |  0.6433   |  13.16    |
3
|  5        |  0.66     |  3.766    |
1
|  6        |  0.63     |  1.0      |
9
|  7        |  0.6467   |  9.949    |
2
|  8        |  0.6183   |  2.591    |
15
|  9        |  0.6017   |  15.0     |
11
|  10       |  0.6467   |  11.18    |
8
|  11       |  0.655    |  8.307    |
4
|  12       |  0.6433   |  4.472    |
9
|  13       |  0.6467   |  9.014    |
7
|  14       |  0.6383   |  7.541    |
11
|  15       |  0.6467   |  11.98    |
=====================================
```

Panas_Negative:

	precision	recall	f1-score
0	0.61	0.8	0.68
1	0.717	0.518	0.585
accuracy			0.655
macro_avg	0.662	0.658	0.631
weighted_avg	0.666	0.655	0.631

{'target': 0.663, 'params': {'n_neighbors': 12 }}

Panas_Positive:

	iter	target	n_neig...
10	1	0.75	10.0
12	2	0.75	12.23
7	3	0.725	7.283
1	4	0.55	1.875
7	5	0.725	7.95
15	6	0.725	15.0
11	7	0.7333	11.12
4	8	0.6467	4.548
13	9	0.7083	13.69
9	10	0.7	9.253
1	11	0.55	1.0
5	12	0.6833	5.913
12	13	0.75	12.78
10	14	0.75	10.43
12	15	0.75	12.51

	precision	recall	f1-score
0	0.626	0.717	0.663
1	0.783	0.683	0.711
accuracy			0.7
macro_avg	0.705	0.7	0.686
weighted_avg	0.705	0.7	0.686

{'target': 0.75, 'params': {'n_neighbors': 10}}

Logistic Regression

Flourishing:

iter	target	C
1	0.705	1.209
2	0.705	1.36
3	0.705	1.0
4	0.705	1.151
5	0.705	1.073
6	0.705	1.406
7	0.705	1.443
8	0.705	1.411
9	0.705	1.125
10	0.705	1.459
11	0.705	1.074
12	0.705	1.327
13	0.705	1.332
14	0.705	1.42
15	0.705	1.061

	precision	recall	f1-score
0	0.85	0.65	0.685
1	0.611	0.75	0.654
accuracy			0.705
macro_avg	0.73	0.7	0.666
weighted_avg	0.727	0.705	0.668

{'target': 0.705, 'params': {'C': 1.209}}

Panas_Negative:

iter	target	C
1	0.73	1.209
2	0.73	1.36
3	0.73	1.0
4	0.73	1.151
5	0.73	1.073
6	0.73	1.406
7	0.73	1.443
8	0.73	1.411
9	0.73	1.125
10	0.73	1.459
11	0.73	1.074
12	0.73	1.327
13	0.73	1.332
14	0.73	1.42
15	0.73	1.061

	precision	recall	f1-score
0	0.676	0.834	0.751
1	0.751	0.634	0.661
accuracy			0.73
macro avg	0.745	0.733	0.734
weighted avg	0.749	0.73	0.704

{'target': 0.730, 'params': {'C': 1.209}}

SVM

Panas_Positive:

iter	target	C
1	0.6917	1.209
2	0.6917	1.36
3	0.6917	1.0
4	0.6917	1.151
5	0.6917	1.073
6	0.6917	1.406
7	0.6917	1.443
8	0.6917	1.411
9	0.6917	1.125
10	0.6917	1.459
11	0.6917	1.074
12	0.6917	1.327
13	0.6917	1.332
14	0.6917	1.42
15	0.6917	1.061

	precision	recall	f1-score
0	0.668	0.651	0.641
1	0.742	0.734	0.714
accuracy			0.692
macro_avg	0.705	0.692	0.676
weighted_avg	0.705	0.692	0.676

{'target': 0.692, 'params': {'C': 1.209}}

Flourishing:

iter	target	C	gamma
1	0.52	-4.02	-3.19
2	0.52	-3.692	-2.012
3	0.6267	3.792	-0.8289
4	0.52	1.153	-3.062
5	0.5817	3.72	-3.441
6	0.52	-1.754	-4.461
7	0.565	3.733	-2.444
8	0.5767	1.289	-0.5621
9	0.52	-4.127	-2.275
10	0.52	-4.592	-1.562
11	0.6233	5.0	-5.0
12	0.5017	5.0	0.0
13	0.52	2.463	-5.0
14	0.52	-5.0	-5.0
15	0.5017	2.908	0.0
16	0.52	-1.47	0.0
17	0.645	5.0	-2.527
18	0.52	-0.7683	-1.808
19	0.52	-5.0	0.0
20	0.5567	5.0	-3.868
21	0.6267	5.0	-1.368
22	0.52	-3.281	-5.0
23	0.56	3.984	-5.0
24	0.52	0.03449	-5.0
25	0.52	-3.421	0.0
26	0.6183	0.2159	0.0
27	0.5933	0.1641	-0.787
28	0.6733	4.59	-1.817
29	0.61	4.379	-1.335
30	0.52	-0.4006	-3.55

	precision	recall	f1-score
0	0.8	0.516	0.588
1	0.613	0.767	0.66
accuracy			0.648
macro_avg	0.705	0.642	0.62
weighted_avg	0.704	0.649	0.624

{'target': 0.673, 'params': {'C': 4.590, 'gamma': -1.817}}

Panas_Negative:

iter	target	C	gamma
1	0.585	4.634	-3.731
2	0.53	-0.943	-0.2235
3	0.53	-2.307	-3.366
4	0.53	-1.617	-1.365
5	0.53	-2.927	-4.067
6	0.68	2.188	-3.312
7	0.5767	4.457	-1.038
8	0.53	-2.075	-1.803
9	0.5767	2.252	-0.3244
10	0.53	-2.877	-1.382
11	0.53	1.788	-5.0
12	0.53	-5.0	0.0
13	0.53	-5.0	-5.0
14	0.635	2.971	-2.428
15	0.53	-5.0	-2.692
16	0.53	1.149	-2.739
17	0.585	5.0	0.0
18	0.6217	3.176	-4.134
19	0.6267	5.0	-5.0
20	0.585	3.603	0.0
21	0.53	-1.369	-5.0
22	0.6383	2.692	-3.348
23	0.53	-2.728	0.0
24	0.6633	3.902	-5.0
25	0.61	2.433	-1.626
26	0.53	1.71	-3.812
27	0.6067	5.0	-2.266
28	0.515	0.8008	0.0
29	0.53	2.953	-5.0
30	0.53	-5.0	-1.274

	precision	recall	f1-score
0	0.635	0.717	0.641
1	0.709	0.651	0.638
accuracy			0.68
macro_avg	0.67	0.683	0.637
weighted_avg	0.674	0.68	0.637

{'target': 0.680, 'params': {'C': 2.188, 'gamma': -3.312}}

Panas_Positive:

iter	target	C	gamma
1	0. 55	0. 4939	-4. 74
2	0. 6633	1. 05	-0. 5729
3	0. 6167	1. 185	-2. 752
4	0. 55	-4. 443	-2. 108
5	0. 6917	1. 226	-2. 607
6	0. 5933	3. 617	-2. 021
7	0. 7	4. 038	-3. 275
8	0. 55	-2. 575	-3. 502
9	0. 55	-4. 942	-1. 793
10	0. 55	0. 9312	-2. 994
11	0. 55	-5. 0	-5. 0
12	0. 55	-2. 45	0. 0
13	0. 6833	5. 0	-5. 0
14	0. 5183	5. 0	0. 0
15	0. 55	-0. 8548	-1. 639
16	0. 55	-5. 0	0. 0
17	0. 55	-1. 663	-5. 0
18	0. 6133	5. 0	-2. 562
19	0. 55	3. 147	-5. 0
20	0. 5183	2. 896	0. 0
21	0. 675	1. 889	-1. 454
22	0. 55	-5. 0	-3. 498
23	0. 55	-3. 474	-5. 0
24	0. 725	0. 5637	-1. 727
25	0. 675	-0. 5102	0. 0
26	0. 55	-0. 797	-3. 525
27	0. 7	5. 0	-3. 912
28	0. 55	-2. 731	-1. 585
29	0. 55	-3. 814	0. 0
30	0. 6667	4. 222	-4. 559

	precision	recall	f1-score
0	0. 626	0. 717	0. 663
1	0. 8	0. 734	0. 738
accuracy			0. 725
macro_avg	0. 713	0. 725	0. 699
weighted_avg	0. 713	0. 725	0. 699

{'target': 0.725, 'params': {'C': 0.564, 'gamma': -1.727}}