Daily Student Stress Level Prediction

Project Category: Life Science

Name: Megan Worrel
Stanford University
mworrel@stanford.edu

Name: Luis Aragon
Stanford University
laragon2@stanford.edu

Project Github: https://github.com/mworrel44/StudentLife

1 Introduction

Stress is a common factor in the college experience with roughly half of college students reporting moderate to severe psychological stress [1]. Given that chronic stress has been identified as a precursor to serious mental health issues like depression, it's important to understand what causes stress and how it can be alleviated [2]. Advancement of mobile technology has made it possible to collect large amounts of multimodal data. This data can be used to construct a data-centric representation of a person's behaviors/environment and ultimately measure or predict stress in a more objective manner. Classification models that have been proposed for this task of predicting states of wellbeing are often either 1) applied generally over a group of people, failing to capture the individualized nature of stress, or 2) applied to a specific individual, failing to generalize to new individuals. Taylor et al, propose a Multi-Task Multi-Kernel Learning (MTMKL) model for predicting stress on the SNAPSHOT dataset that is both individualized and generalizable [3]. We intend to test the robustness of this model by developing a MTMKL model for the comparable StudentLife dataset.
The StudentLife dataset is further explained in section 3. The dataset includes various measurements about a student's behavior and responses about their wellbeing throughout their 10 week quarter[4]. From the data, features are extracted that include the following modalities: phone usage, deadlines, survey responses, and phone sensor information, from a particular student on a particular day. The input to our model is thus a student's responses and data that correspond to our 4 main modalities for a given day. The output from our models is a binary classification - unstressed (0) or stressed (1) - predicting whether the student was stressed or unstressed on the given day. This is a supervised learning task since the dataset also contains responses to a survey where students indicated their stress level on each day. For the MTMKL model, we further cast this as a multi-task problem where each task corresponds to the binary classification of stressed/unstressed for a cluster of students. This produces more personalized predictions for each cluster rather than predicting for the entire group of students. Students are clustered based off of K-means clustering that is further discussed in section 4 and 5. The results of the MTMKL model are compared to logistic regression and an XGBoost model which perform a single-task binary classification for the entire group of students.

2 Related Work

On the StudentLife dataset in particular, two key models have been published which perform the multi-class classification task of predicting the stress levels 1-5. First, Mikelsons et all. utilize a Multilayer Perceptron (MLP) with 4 fully connected layers which focuses solely on the GPS data for its stress prediction[5]. From the GPS data, Mikelsons et all. extracted 8 location features recording measurements such as maximum displacement per day and total distance covered per day. This model is not personalized to individual students and achieves an F1 score of 0.42 predicting the multi-class stress level. Second, Shaw et all. create an LSTM Multitask Auto-encoder Network, "CALM-Net[6]. Cleverly utilizing an LSTM which contains both shared fully connected layers and individualized MLP layers for each student, this model is personalized to each student while also learning about patterns shared among all students. Further, the use of LSTM allows for encoding temporal information from each day to the next, which is useful as there is likely a dependency between tomorrow's stress level and today's stress level. Shaw et all. extract many features from the StudentLife dataset for their prediction including a variety of phone sensor data such as activity inferences, phone charging, and phone lock and other features such as day of the week, sleep rating from survey responses, and time to next deadline. Their CALM-Net model achieves a best F1 score of 0.594 on the multi-class stress level prediction. Overall, both of these models utilize a form of neural network, but it is clear that CALM-Net is state-of-the-art on the StudentLife dataset for multi-class stress level prediction.
The SNAPSHOT study collected a similar dataset to the StudentLife dataset and contains physiological, survey, and phone data from 68 students over 30 days[7]. We studied Jaques et all.'s MTMKL model (also described by Taylor et all.[3]) extensively and drew much inspiration from their results in developing our own MTMKL model for the StudentLife dataset. The MTMKL model is explained in more detail in Section 4.2. Utilizing 200 features across 4 categories - mobility, survey, physiology, and smartphone - their MTMKL model was able to achieve an accuracy of 70.91% on a binary stress classification task. The MTMKL model performs quite well even without the use of neural networks, although it should be noted that the binary classification task is easier than the multi-class stress level classification task.
Bogomolov et all., also perform binary stress level classification on a similar dataset containing phone data, weather, and individual characteristics[8]. They compare a Random Forest algorithm, SVM, neural network, and Generalized Boosted Model and find that the Random Forest performs the best with an accuracy of 72.28%. We use these results as inspiration for comparing the XGBoost Random Forest model to our MTMKL model. Finally, Jaques et all., also perform the comparable task of happy/sad binary prediction on a given day using physiology, survey, and phone data. They achieve a best accuracy score of 68.48% using an SVM[9]. Both of these models are

not personalized to individual students.
Overall, it is clear that stress prediction is a challenging task with accuracies typically around 60-70%, especially since ground truth labels are noisy self-reported stress levels. Further, it's clear that models incorporating some level of personalization tend to out-perform other models on the same dataset.

3 Dataset and Features

The StudentLife dataset contains data on the behaviors and wellbeing of 49 college students taking CS65 (Smartphone Programming) across 10 weeks at Dartmouth College [4]. We use the 1589 student responses to daily stress surveys reporting stress as a level from 1 to 5 as our ground truth labels [4]. We map this to a binary classification problem where levels of 1 and 2 correspond to 0 and levels 3 through 5 correspond to 1. Given the clear class imbalance, we used Synthetic Minority Oversampling Technique (SMOTE), which generates artificial samples for the minority classes, to supplement the minority classes and alleviate the class imbalance problem [10]. Table 1 shows the percentage of data points that correspond to each classification before and after using SMOTE. Further, we extracted a total of 181 features from the dataset for each student daily according to 4 modalities - phone, deadlines, survey responses, and sensor. The 2 phone features are number of sms and number of calls obtained from a table of all messages and phone calls for the students throughout the quarter. The 146 deadline features are binary features indicating whether the student had a deadline on a given day. The 28 survey features are extracted from daily surveys measuring sleep, exercise, activity, behavior, study space, and mood (for example, "Do you feel AT ALL sad right now?" Yes (1) or No (2)). The 5 sensor features include inferences from phone sensors about conversations, dark surroundings, phone lock, activity, and GPS. These inferences were performed by the dataset creators. In order to include sensor GPS data, Gaussian Mixture Model (GMM) was used to cluster particular longitude and latitude to certain clusters following the precedent of Jaques et all.[9]. The GPS feature thus encodes how the student's GPS data on that day compares to other students' GPS data on other days, clustered into 6 different patterns. In order to determine the best setting for the GMM on the GPS data, multiple combinations of number of components and covariance types was ran and a BIC comparison on these runs chose the GMM settings. Figure 1 below depicts the BIC values of the multiple GMM setting combinations that was run.

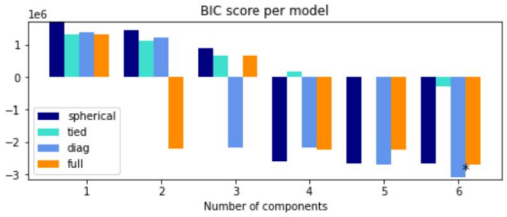

Figure 1: GPS GMM Parameter Comparison

Before running the MTMKL algorithm, all of the features were normalized. We split our data into 80% train (1414 samples), 10% validation (177 samples), and 10% test data (177 samples).
For K-means, students were clustered using pre/post-study survey responses to 8 different surveys on personality, loneliness, sleep, perceived stress, depression, positive and negative affect, and health, GPA, and overall piazza usage for their shared class CS 65. 147 total features were used to cluster the students. Examples of these features include total GPA, GPA for spring quarter 2013, number of questions asked on piazza for CS 65, "In the last month, how often have you been upset because of something that happened unexpectedly?", and "I see myself as someone who tends to be lazy."

Table 1: Count percentage of binary stress level labels

Level	Pre SMOTE Count%	Post SMOTE Count%
0	58	51
1	42	49

4 Methods

After the data was parsed into its features k-means was used to cluster similar students. Using the results of these clustering, we then run logistic regression and XGBoost as our baseline models where the student's predicted cluster is included as a feature in the model. Finally, using the clusters as tasks and the 4 modalities of our features as kernels we run the Multi-task Multi-kernel learning algorithm on the same training, validation, and test sets. K-means clustering and the MTMKL model are further expanded upon in the following subsections.

4.1 K-means Clustering

K-means clustering is a data clustering algorithm that aims to choose centroids that minimizes the distance to points within that centroid's corresponding cluster. The centroids are initialized randomly and then updated corresponding to the updates in Figure 2

1. Initialize **cluster centroids** $\mu_1, \mu_2, \ldots, \mu_k \in \mathbb{R}^d$ randomly.

2. Repeat until convergence: {

For every i, set

$$c^{(i)} := \arg\min_j ||x^{(i)} - \mu_j||^2.$$

For each j, set

$$\mu_j := \frac{\sum_{i=1}^{n} 1\{c^{(i)} = j\} x^{(i)}}{\sum_{i=1}^{n} 1\{c^{(i)} = j\}}.$$

}

Figure 2: K-Means Algorithm

4.2 Primary Model: Multi Task Multi Kernel Learning (MTMKL)

The MTMKL model solves the multi-task problem of predicting stress for our clusters of students from K-means and uses multiple kernels, each corresponding to one modality of our data. The overall objective function for MTMKL is given

$$min_\eta \left[max_\beta \left(\Omega(\{\eta^t\}_{t=1}^T) + \sum_{t=1}^{T} J_t(\beta^t, \eta^t) \right) \right]$$

$J_t(\beta^t, \eta^t)$ corresponds to solving an SVM for a single task t with the least squares loss function[7]. $\Omega(\{\eta^t\}_{t=1}^T)$ is the L1 regularization term with gradient

$$\frac{\partial \Omega(\eta^t)}{\partial \eta_m^t} = -v \sum_{t=1}^{T} \eta_m^t$$

T corresponds to the number of tasks (clusters) and m corresponds to the number of modalities/kernels in our features. The partial derivative of the objective function for each task t and modality m is given by

$$\frac{\partial O_\eta}{\partial \eta_m^t} = -2\frac{\partial \Omega(\eta^t)}{\partial \eta_m^t} - \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N} \beta_i^t \beta_j^t y_i^t y_j^t (k_m^t(x_i^{(t,m)}, x_j^{(t,m)}) + \frac{\mathbb{1}[i=j]}{2C})$$

where N is the number of samples[7]. Overall, the general MTMKL algorithm can be written
1. Initialize η^t for each task t
2. While maximum change in η^t for each task t $< \epsilon$ = 0.001:
    3. Solve the multi-kernel SVM model for each task with objective function

$$\max_{\beta^t} J(\beta^t, \eta^t) = \max_{\beta^t} \left[\sum_{i=1}^{N} \beta_i^t - \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N} \beta_i^t \beta_j^t y_i^t y_j^t (k_\eta^t(x_i^{(t,m)}, x_j^{(t,m)}) + \frac{\mathbb{1}[i=j]}{2C}) \right]$$

where

$$k_\eta^t = \sum_{m=1}^{M} \eta_m^t k_m^t(x_i^{(t,m)}, x_j^{(t,m)})$$

for each modality m and $\eta_m^t > 0$ and $\sum_{m=1}^{M} \eta_m^t = 1$. This calculates the value of each β.
    4. Using the β's from the previous step, update η^t for each task with update rule

$$\eta^t := \eta^t + \alpha \left[2v \sum_{t'=1}^{T} \eta_m^{t'} + \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N} \beta_i^t \beta_j^t y_i^t y_j^t (k_m^t(x_i^{(t,m)}, x_j^{(t,m)}) + \frac{\mathbb{1}[i=j]}{2C}) \right]$$

The algorithm alternates between solving the SVM model for each task to compute all of the β's and updating all of the η's. Solving each SVM is personalized to each cluster while the L1 regularization term combines the η's across all tasks and encourages them to be similar in value, thus sharing information between the different tasks/clusters.

5 Experiments/Results/Discussion

5.1 Logistic Regression

As a baseline model logistic regression was run on our training set and then run with our test set. On this run we evaluated the model's training accuracy, test accuracy and test F1 score. Figures 3 and 4 show the logistic regression models's classification report and heatmap. Table 3 lays out the aforementioned metrics that were evaluated along with XGBoost and MTMKL metrics.

	precision	recall	f1-score	support
0	0.64	0.52	0.58	107
1	0.43	0.56	0.49	70
accuracy			0.54	177
macro avg	0.54	0.54	0.53	177
weighted avg	0.56	0.54	0.54	177

Figure 3: Logistic Regression Report

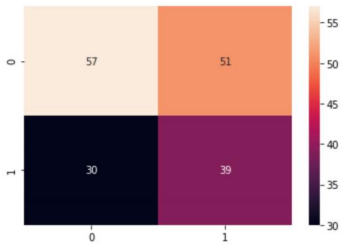

Figure 4: Logistic Regression Heatmap

5.2 XGBoost

Similar to logistic regression XGBoost is utilized as a baseline to compare to. The XGBoost model is trained on the same dataset as logistic regression and tested on the same set. The performance, however, is considerably better than that of logistic regression. Figures 5 and 6 represent the classification report and heatmap with table 3 outlining the model's training and test accuracies as well as the test F1 score with the logistic regression and MTMKL values. Given the vast difference in training and test accuracies it can be seen that the XGBoost model was overfit. Despite overfitting the model's performance on the test set outperforms the logistic regression model.

	precision	recall	f1-score	support
0	0.61	0.63	0.62	84
1	0.66	0.63	0.64	93
accuracy			0.63	177
macro avg	0.63	0.63	0.63	177
weighted avg	0.63	0.63	0.63	177

Figure 5: XGBoost Report

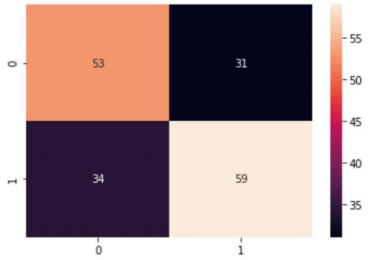

Figure 6: XGBoost Heatmap

5.3 K-means Clustering

In order to find the optimal amount of clusters that would fit the data, the elbow method was used on the sum of distances was plotted against the number of clusters of k-means run on the dataset. Figure 7 is the generated elbow curve that has a kink in the curve at # of clusters = 3. Using PCA on the data to reduce the dimensionality to 2 for visualization purposes shows how the 3 clusters are represented in the dataset in Figure 8. The important features that the XGBoost algorithm during classification given a student's assigned cluster is

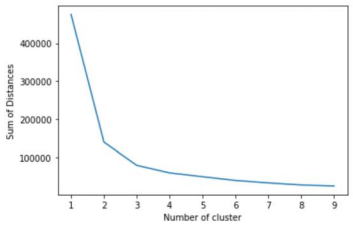

Figure 7: K-means Elbow Method

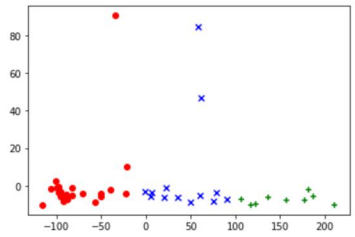

Figure 8: K-means Clusters

shown in Figure 9.

The differences between the clusters can be examined by looking at the clusters gpa and piazza usage in Figure 10 as well as survey responses pertaining to positive and negative characteristics visualized in the heatmaps in Figures 11 and 12 respectively.

5.4 Multi-Task Multi-Kernel Learning (MTMKL)

Once the datasets are clustered MTMKL was run sweeping over various combinations of hyperparameters of step size = [0.001,0.01], C = [1.0, 10.0, 100.0], V = [1.0, .1, .01], kernel = ['linear', 'rbf'], and regularizer = ['L1', 'L2]. The best settings was determined by the combination of hyperparameters that maxamized the AUC of the validation set. The best setting that was determined is expressed in Table 2.

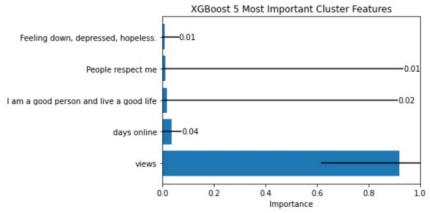

Figure 9: XGBoost Cluster Important Features

	gpa all	days online	views	contributions	questions
cluster 0	3.570450	55.280000	284.080000	17.086957	5.760000
cluster 1	3.134222	34.538462	164.538462	4.833333	0.615385
cluster 2	3.029000	14.666667	54.333333	1.111111	0.666667

Figure 10: Cluster GPA and Piazza Usage

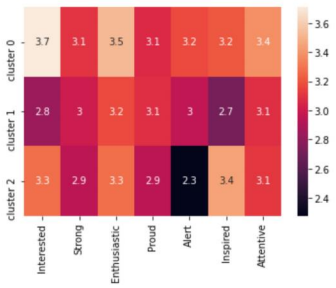

Figure 11: Positive Characteristic Cluster Response

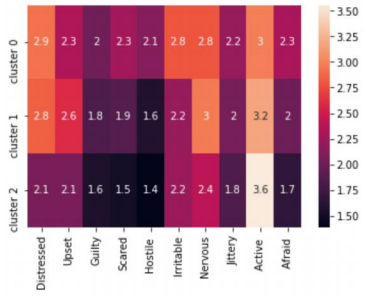

Figure 12: Negative Characteristic Cluster Response

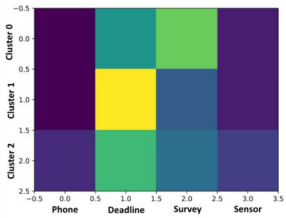

Figure 13: MTMKL Weight Heatmap

Table 2: MTMKL Best Hyperparameters

Step Size	C	V	Kernel	Regularizer
0.01	1.0	1.0	Linear	L1

Similar to the logistic regression and XGBoost the evaluation of the model is on the training and test accuracies as well as the test F1 score. Table X has the MTMKL performance values. The weights for each task and kernel is visualized in Figure 13 as a heatmap. Which expresses that the tasks are most responsive to the deadline and survey kernels as opposed to the phone and sensor kernels. This may be due to the significant percentage of the data features coming from the deadline and survey kernels and a quite small percentage from the other two kernels.

The evaluation metrics used on all 3 models was training and test accuracies along with the test F1 score. These metrics are in Table X and it can be seen that the MTMKL model outperforms XGBoost and logistic regression on the test data and does not appear to suffer from overfitting the training data.

Table 3: Model Performances

Model	Training Accuracy	Test Accuracy	Test F1 Score
Logistic Regression	0.5962	0.5367	0.525
XGBoost	0.9009	0.5762	0.58
MTMKL	0.6669	0.6441	0.64

6 Conclusion/Future Work

Multi task multi kernel learning has been shown to be generalizable in that a new student can just be processed into a cluster and then use the MTMKL model to predict their stress level. The algorithm also proves to be individualized since each student belongs to a cluster that has inherent properties tied to final gpa, piazza usage and survey responses, compared to other models that does prediction independent of the student. It was shown that the MTMKL model has slightly better performance than the XGBoost model and they both perform significantly better than the logistic regression model.
Future work could include testing the results of a neural network on the dataset, particularly utilizing a RNN to capture the temporal relationship between stress levels recorded on each day. Additionally, this problem can be recast as a multiclass prediction task with the original 5 stress labels rather than mapping to a binary stress/no stress classification.

7 Contributions

Megan extracted the stress labels and a variety of features from the StudentLife dataset including the phone usage, deadlines, and survey responses. She also extracted the necessary features and ran the k-means algorithm to cluster the students. She ran the XGBoost algorithm and contributed her results to the report. Finally, she helped develop the images used to display the results of the MTMKL model.

Luis extracted features from the sensor data which include activity, phonelock, conversations, dark, and gps. Ran Gaussian Mixture Model on the GPS data to have clusters of gps locations as a feature. MTMKL best hyperparameter tuning on complete feature set.

References

[1] John Yang and Claire Mufsen. College students' stress levels are 'bubbling over.' here's why, and how schools can help. PBS NewsHour, Nov 2021.

[2] Jon Cooper. The link between stress and depression. WebMD, Apr 2022.

[3] Sara Taylor and Natasha Jaques et. al. Personalized multitask learning for predicting tomorrow's mood, stress, and health. *IEEE Transactions on Affective Computing*, 11(2):200–13, Apr 2020.

[4] Rui Wang and et. al. Fanglin Chen. Studentlife: Assessing mental health, academic performance and behavioral trends of college students using smartphones. *Association for Computing Machinery*, pages 3–14, 2014.

[5] Gatis Mikelsons and et. al. Matthew Smith. Towards deep learning models for psychological state prediction using smartphone data: Challenges and opportunities. *arXiv*, Nov 2017.

[6] Abhinav Shaw and et. al. Natcha Simsiri. Personalized student stress prediction with deep multitask network.

[7] Natasha Jaques and Sara Taylor et. al. Multi-task, multi-kernel learning for estimating individual wellbeing. *MIT Media Lab*, page 7, 2015.

[8] Michela Ferron Fabio Pianesi Alex Pentland Andrey Bogomolov, Bruno Lepri. Daily stress recognition from mobile phone data, weather conditions and individual traits. *ACM Multimedia*, page 10, 2014.

[9] Natasha Jaques, Sara Taylor, and Asaph Azaria. Predicting students' happiness from physiology, phone, mobility, and behavioral data. *IEEE Xplore*, pages 222–28, 2015.

[10] Smote. `https://docs.microsoft.com/en-us/azure/machine-learning/component-reference/smote`, Nov 2021. Accessed: 2022-04-30.

[11] Iman Deznaby Madalina Fiterau Tauhidur Rahaman Abhinav Shaw, Natcha Simsiri. Personalized student stress prediction with deep multitask network. *Proceedings of the 1st Adaptive Multitask Learning Workshop*, page 4, 2019.

[12] Melih Kandemir and et. al. Akos Vetek. Multi-task and multi-view learning of user state. *ScienceDirect*, pages 97–106, 2014.

[13] Zakaria Jaadi. A step-by-step explanation of principal component analysis (pca). `https://builtin.com/data-science/step-step-explanation-principal-component-analysis`, Apr 2021. Accessed: 2022-04-30.

[14] Elbow method for optimal value of k in kmeans. `https://www.geeksforgeeks.org/elbow-method-for-optimal-value-of-k-in-kmeans/`, Feb 2021. Accessed: 2022-04-30.

[15] Xgboost. `https://www.geeksforgeeks.org/xgboost/`, Oct 2021. Accessed: 2022-06-06.

[16] Charles R. Harris, K. Jarrod Millman, Stéfan J van der Walt, Ralf Gommers, Pauli Virtanen, David Cournapeau, Eric Wieser, Julian Taylor, Sebastian Berg, Nathaniel J. Smith, Robert Kern, Matti Picus, Stephan Hoyer, Marten H. van Kerkwijk, Matthew Brett, Allan Haldane, Jaime Fernández del Río, Mark Wiebe, Pearu Peterson, Pierre Gérard-Marchant, Kevin Sheppard, Tyler Reddy, Warren Weckesser, Hameer Abbasi, Christoph Gohlke, and Travis E. Oliphant. Array programming with NumPy. *Nature*, 585:357–362, 2020.

[17] Wes McKinney et al. Data structures for statistical computing in python. In *Proceedings of the 9th Python in Science Conference*, volume 445, pages 51–56. Austin, TX, 2010.

[18] Tianqi Chen and Carlos Guestrin. XGBoost: A scalable tree boosting system. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '16, pages 785–794, New York, NY, USA, 2016. ACM.

[19] Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, et al. Scikit-learn: Machine learning in python. *Journal of machine learning research*, 12(Oct):2825–2830, 2011.

[20] Nitesh V Chawla, Kevin W Bowyer, Lawrence O Hall, and W Philip Kegelmeyer. Smote: synthetic minority over-sampling technique. *Journal of artificial intelligence research*, 16:321–357, 2002.

[21] Michael Waskom, Olga Botvinnik, Drew O'Kane, Paul Hobson, Saulius Lukauskas, David C Gemperline, Tom Augspurger, Yaroslav Halchenko, John B. Cole, Jordi Warmenhoven, Julian de Ruiter, Cameron Pye, Stephan Hoyer, Jake Vanderplas, Santi Villalba, Gero Kunter, Eric Quintero, Pete Bachant, Marcel Martin, Kyle Meyer, Alistair Miles, Yoav Ram, Tal Yarkoni, Mike Lee Williams, Constantine Evans, Clark Fitzgerald, Brian, Chris Fonnesbeck, Antony Lee, and Adel Qalieh. mwaskom/seaborn: v0.8.1 (september 2017), September 2017.

[22] John D Hunter. Matplotlib: A 2d graphics environment. *Computing in science & engineering*, 9(3):90–95, 2007.

[23] Martín Abadi, Paul Barham, Jianmin Chen, Zhifeng Chen, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Geoffrey Irving, Michael Isard, et al. Tensorflow: A system for large-scale machine learning. In *12th {USENIX} Symposium on Operating Systems Design and Implementation ({OSDI} 16)*, pages 265–283, 2016.

[24] Pauli Virtanen, Ralf Gommers, Travis E. Oliphant, Matt Haberland, Tyler Reddy, David Cournapeau, Evgeni Burovski, Pearu Peterson, Warren Weckesser, Jonathan Bright, Stéfan J. van der Walt, Matthew Brett, Joshua Wilson, K. Jarrod Millman, Nikolay Mayorov, Andrew R. J. Nelson, Eric Jones, Robert Kern, Eric Larson, C J Carey, İlhan Polat, Yu Feng, Eric W. Moore, Jake VanderPlas, Denis Laxalde, Josef Perktold, Robert Cimrman, Ian Henriksen, E. A. Quintero, Charles R. Harris, Anne M. Archibald, Antônio H. Ribeiro, Fabian Pedregosa, Paul van Mulbregt, and SciPy 1.0 Contributors. SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. *Nature Methods*, 17:261–272, 2020.

[25] Sara Taylor, Natasha Jaques, Ehimwenma Nosakhare, Akane Sano, and Rosalind Picard. Personalizedmultitasklearning. `https://github.com/mitmedialab/personalizedmultitasklearning`, 2019.