Reproducible Workflows for Exploring and Modeling EMA Data

Ching-Yun Yu and Yi Shang
Dept. of EECS
University of Missouri
Columbia, Missouri, USA
cytbm@missouri.edu
shangy@missouri.edu

Timothy Trull
Dept. of Psychological Sciences
University of Missouri
Columbia, Missouri, USA
trullt@missouri.edu

***Abstract*—Improper use of substances like cannabis may lead to physical, emotional, economic, and social problems. Therefore, it is significant to elucidate the inter-individual and intra-individual influences along with contextual influences that predict the use of cannabis. TigerAware is a mobile survey data collection platform that holds unique promise to advance research in addiction and substance use. This paper presents a novel method to support Ecological Momentary Assessment (EMA) studies. We propose to extract useful information from TigerAware survey data using data mining and machine learning methods, and structure customizable survey analyses into reproducible workflows. Through our analysis pipeline for EMA, researchers are able to discover meaningful information from survey data with minimal duplication of effort and improve the efficiency and rigor of the process.**

Keywords—mobile survey, data analytics, machine learning, digital health, data mining, ecological momentary assessment, cannabis

I. INTRODUCTION

Substance abuse could adversely affect societal health and is also an economic burden. We set out to address this issue by performing exploratory analyses of survey data from TigerAware, an advanced data collection platform [1], and building predictive models. While clinical psychological assessment typically relies on global retrospective self-reports collected at clinic visits, Ecological Momentary Assessment (EMA) [2] involves real-time repeated sampling of subjects' current psychological states and behaviors in their natural environments. EMA maximizes ecological validity, minimizes recall bias, and allows for the study of micro-processes that influence behavior in real-world contexts. However, EMA data analysis is challenging because real-life EMA data are often incomplete, noisy, and multi-modal.

We propose to develop reproducible analysis pipelines to support EMA in addiction research, focusing on two areas. First, using data mining methods to discover significant relationships and patterns in EMA data, compute clusters of participants, and identify outlier and strongly correlated participants. Second, using various machine learning algorithms to build models to predict items of interests, such as the amount of cannabis smoked since the last report from factors such as positive affect, negative affect, pain rating, impulsivity, motivation, and to interpret the predictions in comparison with the ground truth. Such efficient, scalable, modular, and tested workflows enable researchers to manage and understand survey data in a rigorous and reproducible way.

II. RELATED WORK

A. TigerAware

TigerAware is an innovative mobile survey and sensor data collection and analytics system developed at the University of Missouri that supports advanced EMA studies. TigerAware provides researchers with the technical tools they need to facilitate data collection, including external device integration, compliance monitoring, advanced notification configurations, and a broad array of question types. Researchers without programming knowledge can easily create customizable surveys on smartphones used by participants and collect real-life data from built-in wearable sensors and smartphone sensors. In this paper, we set out to process TigerAware survey data to extract useful information.

B. Reproducible Analysis Pipeline for Data Streams

Reproducible Analysis Pipeline for Data Streams (RAPIDS) [3] aims to standardize the preprocessing, feature extraction, analysis, visualization, and reporting of data streams from mobile sensors to support a wider range of wearable devices and smartphone sensing applications, and to encourage transparency and open science in mobile sensing research. RAPIDS consists of a set of Python and R scripts that are executed on top of reproducible virtual environments, orchestrated by a workflow management system, and organized according to a consistent file structure for data science projects. RAPIDS addresses the significant variation in mobile sensor data across teams, individuals, and time, relying on open algorithms and software packages to standardize data processing and analysis as well as on open discussions, documentation, and software distribution tools to support open science, reproducibility, and code sharing. Based on the ideas of RAPIDS, we extend a novel pattern to discover useful information in EMA data. In this paper, we design reproducible workflows to explore and model the EMA survey data collected by TigerAware and demonstrate the robustness of these pipelines.

C. Sample Dataset

We collected a moderate set of survey data from a real-life EMA study on cannabis use using TigerAware, including three surveys administered to fifty-three participants [4]. Surveys included Morning Report, Random Prompt, and Cannabis Use. Morning Report was completed daily by participants and asked about sleep quality, mood, and substance use. Random Prompt was completed four times a day by participants and asked about a broader range of content, including impulsivity, pain rating, substance use motivation, and the participants' setting. Cannabis Use was completed only when cannabis was used and logged by participants. Each participant took this set of surveys repeatedly for up to 14 days. We divided this dataset into three tables, each of which contained responses to one type of survey. Each row in each table represents a participant's response to a survey, and each column represents a question from the survey.

III. Methods and Implementation

We propose to use a workflow manager [5] to organize the analysis pipelines into contained and scalable steps that support different behavioral features and visualization. Researchers do not need to write any computer code, but only use plain text files for configuration. Each analysis step is executed only when its input or parameters change, and when this happens, any dependent steps are automatically recalculated. This means that the workflows are efficient because every survey for each participant goes through exactly the same processing in isolated steps, and its input and output files could be examined at any time. Such capabilities enable researchers to divide an analysis workflow into small parts that could be audited, shared in an online repository, reproduced on other computers, and understood by other people as they follow a familiar and consistent structure.

A. Compute Clusters of Participants

For clustering the participants, we combined the rows for each participant in the input data (from the TigerAware study) by calculating the average of the survey responses that were non-missing values. In this way, the number of rows equals the number of participants. The first step in this workflow (Fig. 1) is to retrieve these data and then pre-process them. We replaced the missing values in each column with the average of the non-missing values in each column and removed the constant columns, such as those with all zeros. Subsequently, we performed clustering using the X-means algorithm [6], which is based on a heuristic approach to determine the correct number of centroids. It starts with the smallest set of centroids and then iteratively explores whether it makes sense to use more centroids based on the data. If a cluster is split into two, this is determined by the Bayesian Information Criterion [7] to balance the trade-off between accuracy and model complexity. To visualize the results, we used visualization tools for centroid-based clustering models such as heat maps and centroid charts to display the size of all clusters found and the basic features of each cluster. In addition, we used Principal Component Analysis (PCA) [8] and t-distributed Stochastic Neighborhood Embedding (t-SNE) [9] methods to reduce the clustered data to two dimensions and visualize them as scatter plots on a two-dimensional map and compare them. With the rich functionality included in this workflow, it is easy to find important relationships in EMA data.

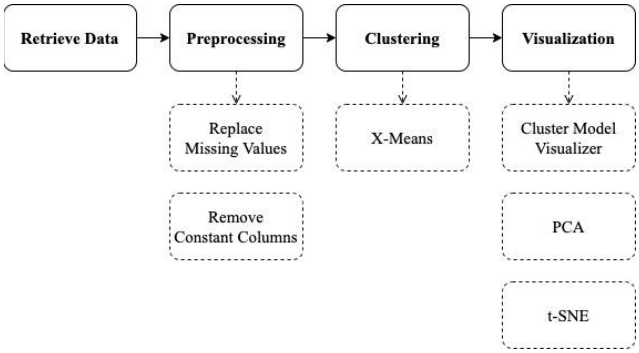

Fig. 1. The clustering workflow.

B. Identify Outliers among Participants

To identify outlier participants, we preprocessed the input data (Fig. 2) in the same way as the clustering workflow. Then, we generated ID columns to associate outlier information with the original data. These ID columns were removed after joined. Regarding outlier search, there are two methods to choose from. First is the outlier detection method recommended by Ramaswamy, Rastogi, and Shim in "Efficient Algorithms for Mining Outliers from Large Data Sets" [10]. They suggested ranking each point according to its distance from the k-th nearest neighbor and declaring the first n points in that ranking as outliers. We could specify the values of k and n by setting the number of neighbors and the number of outliers parameters, respectively. Another approach is based on the local outlier factors [11], where locality is given by a few nearest neighbors whose distance is used to estimate the density. By comparing the local densities of an object and its neighbors, we could identify regions with similar densities and points with significantly lower densities than their neighbors. Similarly, we used PCA and t-SNE methods to reduce the results to two dimensions and visualize them as scatter plots and compare them.

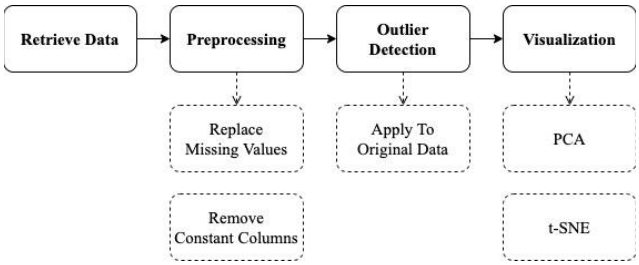

Fig. 2. The outlier detection workflow.

C. Identify Strongly Correlated Participants

We selected a few survey questions, such as 5, with the largest variations in response values from participants and used the values from these five columns to calculate the correlation coefficients between participants so that we could focus on the more varied and interesting responses. The workflow (Fig. 3) for identifying correlated participants is relatively simple. After replacing the missing column values with the average of the non-missing column values and removing the constant columns, we used these processed input data to calculate the correlation matrix. Strongly correlated participants will be identified.

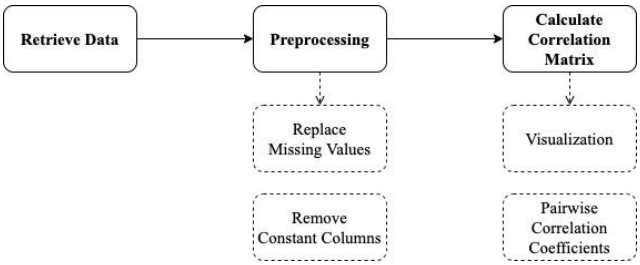

Fig. 3. The correlation workflow.

D. Build Predictive Models

The workflow of building predictive models consists of 6 steps. To illustrate the workflow, we used the example of predicting the amount of cannabis use since the last report by participants. For the input data, we selected forty-three attributes from the Cannabis Use and Random Prompt survey data to focus on mood, impulsivity, pain, alcohol use, cigarette use, craving, cannabis potency, substance use motivation, effects of cannabis, and who participants were with and where they were. We combined all positive emotion attributes, negative emotion attributes, and impulsivity attributes into three attributes by calculating the means of those column values that were not missing. After defining predictors, we defined two classes (less than or greater than 0.2 grams) for the amount of cannabis used by participants for prediction.

The first step of this workflow (Fig. 4) is to retrieve the input data and define some columns as special roles, such as label and ID columns, to be excluded from the predictors. After replacing the missing column values with the average of the non-missing column values, the unlabeled data were noted as the scoring data, and the labeled data were divided into the training and validation sets in the ratio of 6:4. A random seed is set here to ensure that the data split according to this ratio are always the same in each run.

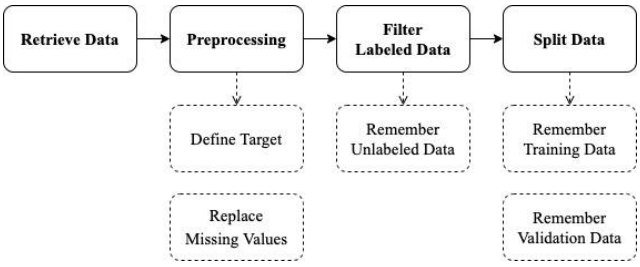

Fig. 4. Preprocessing in the prediction workflow.

The second step is automatic feature engineering [12] (Fig. 5), which is optional. It aims to select a subset of features when there are too many features, i.e., survey questions, to build robust models. Machine learning models tend to overfit when the number of training examples is relatively small compared to the number of independent features. After finding the optimal feature set, this feature set would be applied to the training data. Since real-life EMA survey data are often incomplete and noisy, automatic feature engineering in this workflow would be very helpful in this case. Also, we provide several hyperparameters for researchers to tune and optimize.

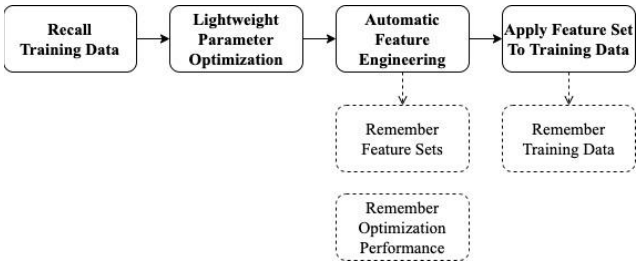

Fig. 5. Automatic feature engineering in the prediction workflow.

The cannabis dataset we used has 43 features selected by the researchers from over 100 features. We performed this process for comparison. Before starting feature engineering, lightweight parameter optimization was performed to find a function that would reduce the generalization error on the dataset. For both the optimal parameters finding process and automatic feature engineering, they were cross validated to estimate the statistical performance of the learned model. All parameters in these processes are customizable.

For machine learning models, there are many options, such as Naive Bayes, Generalized Linear Model, Logistic Regression, Fast Large Margin, Deep Learning, Decision Tree, Random Forest, Gradient Boosted Trees, and Support Vector Machine. For model learning (Fig. 6), most confidence values are not resembling probabilities and sometimes could be severely skewed, so an improved Platt scaling method [13] was used to rescale the confidences produced by the model. This requires a labeled calibration dataset that the model has not been built on, so the training set was divided in a ratio of 9:1 before training. The trained model was then fed into the testing process to evaluate its performance.

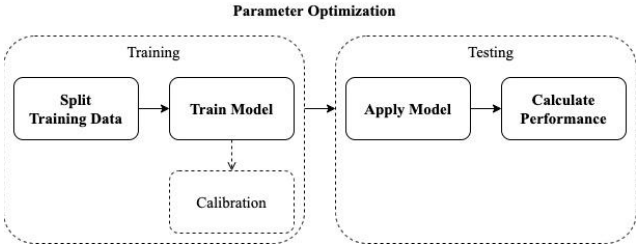

Fig. 6. The model learning process for finding optimal parameters in the prediction workflow.

The model learning process of feature engineering is similar to parameter optimization, but it incorporates a cost-sensitive learning algorithm. The algorithm first implements the original Meta Cost [14] using the multiple bagged models and generates several artificial data points that are close to the one to be predicted. Confidences for those similar but generated data points are then used as distribution for the confidences for the point to be predicted. These confidences are averaged and used as input to compute the expected cost. Finally, the prediction with the lowest expected cost, that is, confidences times cost/benefit for this prediction and the possible errors, is selected as the final prediction. We tested the robustness of this model learning process on our survey dataset and obtained good performance.

The third step is to perform the actual model training. Similarly, we first conducted cross-validation [15] to find the optimal hyperparameters for the model. Once the parameters were found, the same training process as in the second step (Fig. 6) was performed. Subsequently, confidence scaling was applied to optimize accuracy for a given cost matrix using the cost-sensitive learning algorithm we mentioned in the feature engineering section.

The fourth step is to interpret the prediction results (Fig. 7). After inputting the trained model, training data, validation data, and scoring data, the explain operator would output the explained prediction results and the weights of the attributes. The explained prediction results would be displayed as a table highlighting the attributes that most strongly contradict or support each prediction result and their numerical details. On the other hand, the local attribute weights would be identified by the correlation of a neighboring set of data points generated by the operator. Although the relationship between attributes and predictions may be highly nonlinear globally, the local linear relationship is sufficient to explain the prediction results. In addition, this operator could compute model-specific but model agnostic global attribute weights. If the true labels of the test data are known, then all supporting and contradictory local explanations positively affect the weights of correct and incorrect predictions, respectively. In contrast, if the true labels are unknown, the global weights use only the supporting local weights.

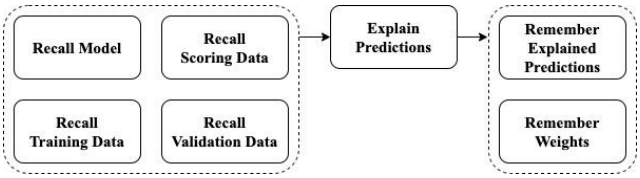

Fig. 7. The explanation process in the prediction workflow.

The fifth step is to validate the model. We implemented two methods. First, we directly calculated the classification error of the model according to the labeled data. Second, we split the data into a specified number of batches by using the mod function on the row number. We then performed a multiple hold-out set validation with Robust Estimation [16] (Fig. 8), which has a similar quality of performance estimation as the cross-validation method, but with a shorter runtime. To avoid the excessive influence of outliers on the average, we removed the highest and lowest values of the main criteria performance and then calculated the average of the input performance vectors.

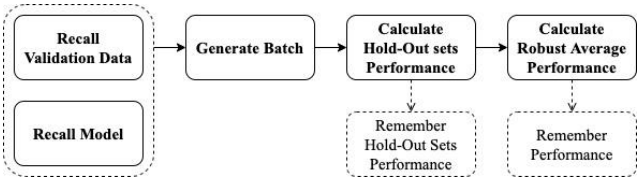

Fig. 8. The validation process in the prediction workflow.

Notably, since we attempted to classify the input data into two classes, we also calculated the performance of the binomial classification to create a Receiver Operating Characteristic (ROC) [17] graph for comparing models. Since the prediction for each example may or may not be correct, this resulted in a 2x2 confusion matrix with four entries, true positive (TP), false positive (FP), false negative (FN), and true negative (TN). The threshold was varied, and a point (x, y) was plotted for each threshold value:

$$x = FP / (FP + TN) \quad (1)$$

$$y = TP / (TP + FN) \quad (2)$$

The final step is to create a production model (Fig. 9) by training on the combined training and validation datasets with the same optimal parameters from the previously trained model. After that, we could reuse this production model for future predictions.

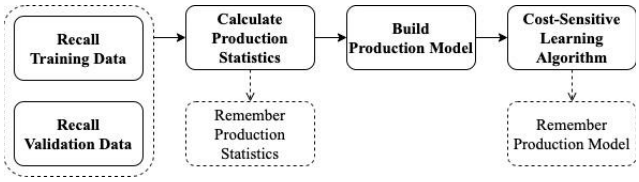

Fig. 9. The production process in the prediction workflow.

The training process for the production model is similar to the second step, but it uses an exception handling operator to capture error messages (Fig. 10). This operator has two subprocesses, Try and Catch, just like the exception handling constructs used in many programming languages. It first tries to execute the Try subprocess. If there are no errors, this operator would deliver the result of the Try subprocess. If there are any errors that the process does not stop, it would continue to execute the Catch subprocess and returns the result of the Catch subprocess.

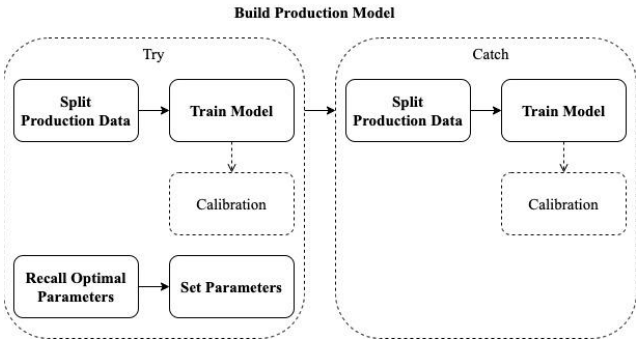

Fig. 10. The training process for the production model in the prediction workflow.

E. Customizable Data Analysis

Since we have built several reproducible data analysis workflows, we could combine them for more advanced analysis. For instance, we would like to further analyze outlier participants and compare it to using only the outlier detection workflow. First, we fed the TigerAware survey data into the prediction workflow and used several desired models to make predictions. Then, we used the performance calculation process in the fourth step to obtain the prediction accuracy for each

participant by setting the filter operator to select only the participant IDs we were interested in. After obtaining the prediction results, we used the visualization tool in the clustering workflow to display the accuracy distribution of each participant as a histogram in increasing order and selected the five participants with the lowest accuracy in each model.

To compare with the outlier detection workflow, we fed the input data into this workflow and obtained five participants by setting the number of outliers to five. Finally, we fed all the selected participants into the correlation workflow to see the correlation matrix visualization and the pairwise correlation coefficients between them to find what they have in common. This is a good example of the practical applicability of combining the reproducible data analysis workflows we built, and shows that they are customizable and easy to use.

IV. Preliminary Results

We used RapidMiner (a data science platform that allows users to design data analysis processes in a plug-and-play fashion) [18] to demonstrate the expected results of our proposed method. The analysis was performed based on the cannabis data, i.e. four TigerAware survey datasets:

1) Cannabis Use survey data, which contains fifty-one participants with one hundred and twenty-two questions each.

2) Random Prompt survey data, which contains fifty-three participants with one hundred and twenty-one questions each.

3) Morning Report survey data, which contains fifty-three participants with fifty-four questions each.

4) Cannabis Use and Random Prompt survey data, which contains fifty-three participants with forty-three specific questions each.

A. Compute Clusters of Participants

After executing the clustering workflow for the TigerAware survey data, we obtained the cluster sizes for all participants as well as the top three attributes that have the greatest impact on each cluster (Fig. 11). Since we used the X-means method for clustering, each of our datasets was clustered into two. The ratios of the number of clusters for these four datasets are 42:9, 16: 37, 44:9, and 30:23, all within reasonable limits. These surveys are analyzed individually and do not influence each other.

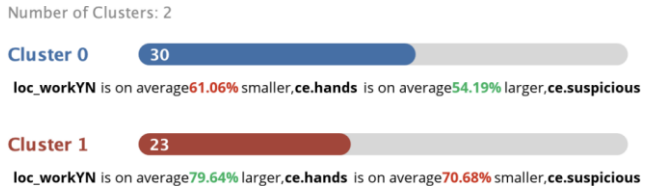

Fig. 11. The clustering overview for the dataset consisting of Cannabis Use and Random Prompt survey data.

Also, we used the visualization tool in the workflow to plot the centroid charts (Fig. 12) for our datasets to show the values for the cluster centroids. We could clearly see the distribution of the attributes of the clusters and the differences between them to determine whether they are clustered well. These resources could be very helpful for analyzing and exploring EMA survey data.

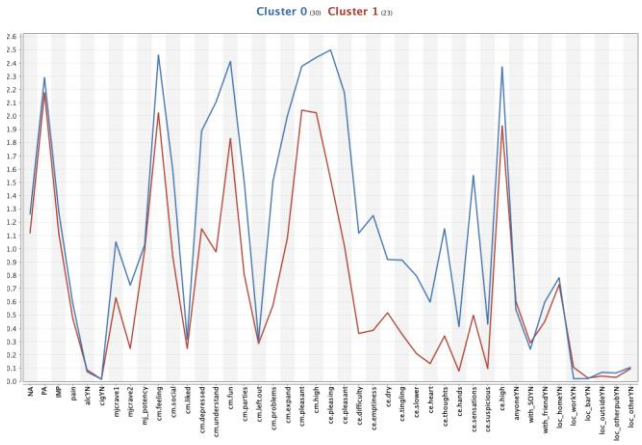

Fig. 12. The centroid chart for the dataset consisting of Cannabis Use and Random Prompt survey data.

In addition, we used PCA and t-SNE methods to reduce the clustered data to two dimensions and visualize them as scatter plots (Fig. 13), and we found that they clustered well without any overlap. It is worth noting that attributes of the Morning Report survey are not all included in the other surveys, and it uses a different time frame.

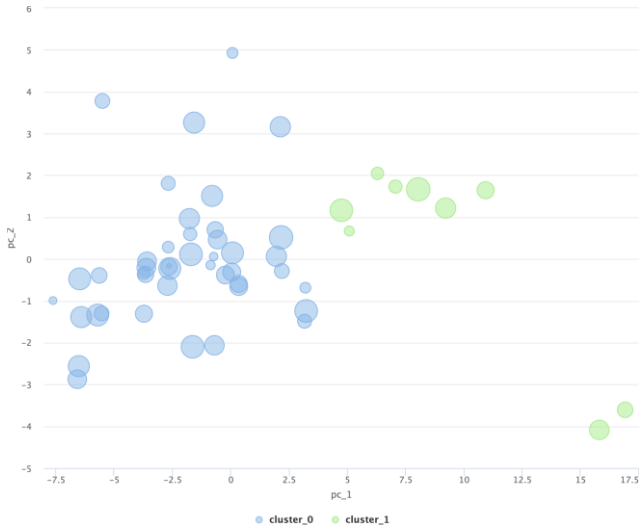

Fig. 13. The PCA result of the clustering for the Morning Report survey.

B. Identify Outliers among Participants

We used the local outlier factors approach to identify regions with similar densities and points with significantly lower densities than their neighbors in the four surveys. The original results are a set of numbers, and to see the distribution more clearly, we reduced the results to two dimensions using PCA and t-SNE methods and visualized them as colored scatter plots (Fig. 14). The redder points indicate they are more likely to be outliers and vice versa. The denser points are shown in blue, while the points further away from the cluster gradually change color. From these plots, we could easily identify the relationship between the outlier points. This approach primarily shows outlier relationships, but researchers can also choose to use the outlier detection method to quickly find a specific number of outliers.

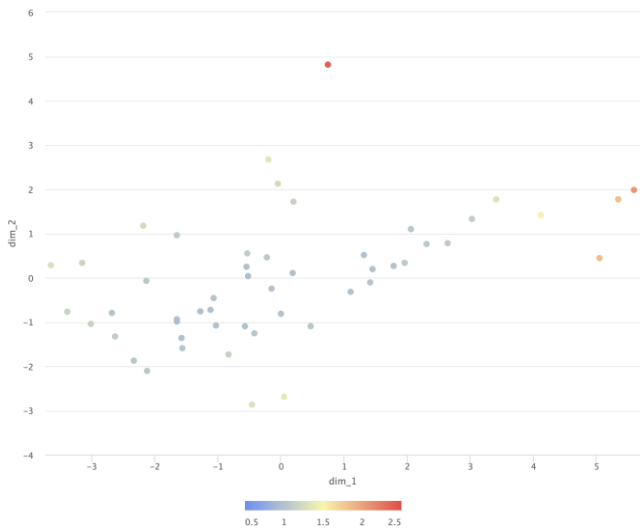

Fig. 14. The t-SNE result of the outlier detection for the Cannabis Use survey.

C. *Identify Strongly Correlated Participants*

We selected the five attributes with the greatest variations in values from each dataset and used only the values from these five columns to calculate the correlation coefficients between participants. Then, we visualized the numbers in this correlation matrix as different colors (Fig. 15), with the redder regions meaning that participants on the X-axis are more correlated with those on the Y-axis, and the bluer regions meaning that participants on the X-axis are less correlated with those on the Y-axis. We could see that the situation is almost the same for most participants, but we still found some regions that are completely blue between participants.

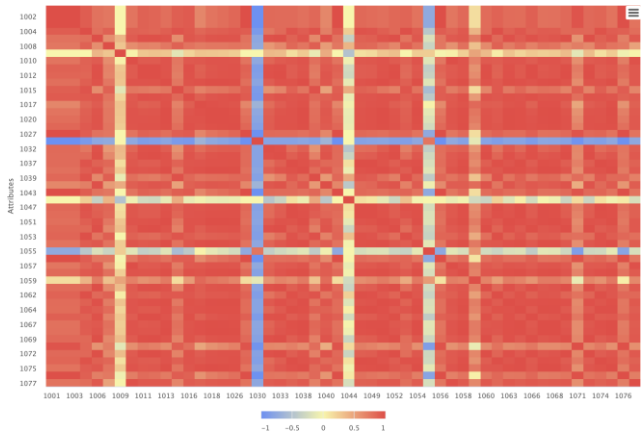

Fig. 15. The correlation matrix for the Random Prompt survey.

In Fig. 15, we can find that the regions between participants 1009, 1044, 1055, and other participants are mainly yellow, which means that they are not correlated with other participants. Furthermore, participant 1030 is strongly negatively correlated with almost all other participants (the regions corresponding to it are mostly blue). This is consistent with the results we obtained from the outlier detection workflow. In that workflow, the outliers we obtained using the efficient outlier detection algorithm included participant 1030.

D. *Build Predictive Models*

We trained five models using the prediction workflow on the dataset consisting of Cannabis Use and Random Prompt survey data (853 rows and 43 columns), including the decision tree, random forest, gradient boosted trees, support vector machine, and H2O's deep learning [19]. H2O's deep learning is based on multilayer feed-forward artificial neural networks trained by stochastic gradient descent using backpropagation. Table 1 shows the test accuracy and total execution time for all models. Our training was run on a MacBook Pro with sixteen gigabytes of memory. In terms of performance, the highest test accuracy we obtained came from the support vector machine at 68.7%. This is followed by the gradient boosted trees at 66.8%. Noteworthy, we found that the test accuracy of the prediction results using automatic feature engineering on our dataset was lower than the case without the method.

To ensure good predictive performance, we must have a way to objectively evaluate the performance of any model trained on our data. There are four types of attribute weights that allow us to understand why models make the predictions they do. First, the general global weights indicate how important the features are in general, but do not take any model into account such as the correlation of the attributes with the label. Second, the model-specific global weights are calculated by the averages of the local importance for the model on the validation dataset according to the modified Locally Interpretable Model Explanations (LIME) [20] method. Although they indicate what is important in general for the model, the dependency on the validation dataset sometimes makes them somewhat biased. Third, the model-based global weights, for instance, some models could produce global weights themselves, such as Linear Regression or Gradient Boosted Trees. These weights are usually good indicators, but not all models can produce them. Fourth, the model-specific local weights do not show the importance in general, but for a particular example. These weights work for all models but make an assumption of local linearity which be incorrect. In addition, they are used as input for the model-specific global weights to derive the global weights of a model. Fig. 16 shows the model-specific global weights for the support vector machine model, which has the highest accuracy on our datasets.

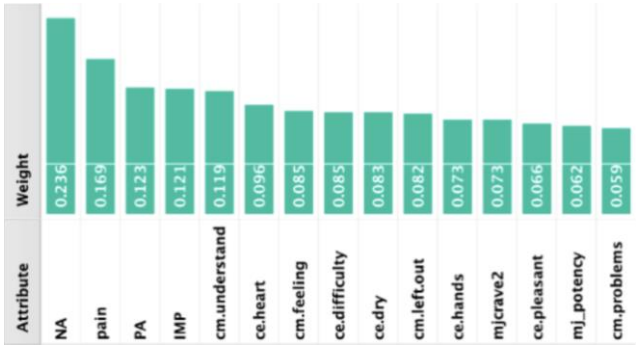

Fig. 16. The attribute weights for the support vector machine model.

In Fig. 16, the top five attributes that have the greatest impact on model predictions are negative affect (NA), pain, positive affect (PA), impulsivity (IMP), and an attribute representing understanding things with the help of cannabis (cm.understand).

The NA attribute is based on the participants' responses to whether they felt afraid, hostile, jittery, nervous, shaky, angry, scornful, loathing, sad, blue, downhearted, or alone in the past fifteen minutes. The pain attribute is based on the participants' level of pain felt since the last report. The PA attribute is based on the participants' responses to whether they felt active, alert, enthusiastic, excited, or happy in the past fifteen minutes. The IMP attribute is based on the participants' responses to whether they felt and acted on a strong impulse, did something without really thinking it through, gave up easily, or did something for the thrill of it in the last fifteen minutes. The cm.understand attribute is based on how much the participants agree that they used cannabis to understand things differently. We can find that these features are consistent with existing understandings of the factors that influence cannabis use [21].

For the execution time of machine learning models, the training time is usually the longest. However, in Table 1, the total execution times of most models are many times longer than the training times. The reason for such long execution times is the process of explaining the predictions. This process calculates the confidence value of each attribute for each row of data with respect to its prediction result and represents it with different colors according to the confidence level. If automatic feature engineering is used, the execution time would be much longer.

TABLE I. PERFORMANCE COMPARISION OF MACHINE LEARNING MODELS IN PREDICTING THE AMOUNT OF CANNABIS SMOKED.

Model	*Test Accuracy*	*Total Time*	*Training Time*	*Scoring Time*
Deep Learning (H2O)	62.4%	15 s	530 ms	107 ms
Decision Tree	50.2%	9 s	870 ms	70 ms
Random Forest	56.2%	1 min 17 s	6 s	753 s
Gradient Boosted Trees	66.8%	1 min 3 s	47 s	163 ms
Support Vector Machine	68.7%	1 min 16 s	6 s	837 ms

We obtained a ROC graph (Fig. 17) by calculating the performance of the binominal classification to help us evaluate the five machine learning models we selected. We could see that the decision tree represents the baseline, which means that it predicts the results always in one of the classes.

E. Customizable Data Analysis

We applied all workflows together to further analyze several participants with lower accuracy in the prediction workflow. We fed the dataset consisting of Cannabis Use and Random Prompt survey data into the prediction workflow and used machine learning models to make predictions for each participant individually. Then, we used the visualization tool in the clustering workflow to display the predictions as a histogram in increasing order of accuracy (Fig. 18).

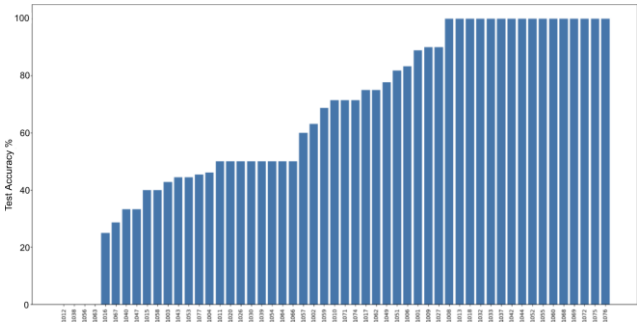

Fig. 18. The predictions of the support vector machine for each participants.

Finally, we fed the five participants with the lowest accuracy in each model and the five participants detected by the outlier detection workflow into the correlation workflow and calculated the correlation matrix (Fig. 19) and the pairwise correlation coefficients between them based on the attributes to see how relevant they were. This demonstrates the practical applicability of combining our proposed data analysis workflows.

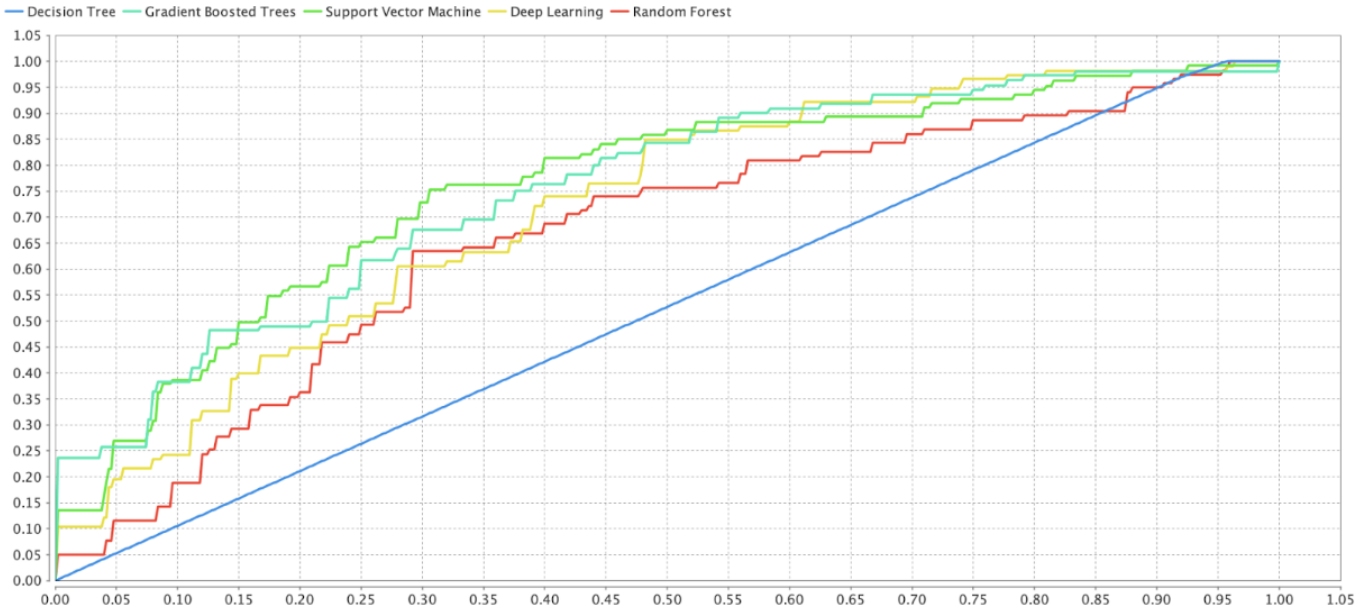

Fig. 17. The ROC graph for the five machine learning models.

Attributes	1011	1012	1015	1016	1030	1038	1043	1056	1057	1063	1072	1075
1011	1	0.786	0.520	0.689	0.451	0.805	0.071	0.519	0.649	0.688	0.807	0.505
1012	0.786	1	0.624	0.781	0.675	0.779	0.348	0.664	0.730	0.793	0.651	0.585
1015	0.520	0.624	1	0.704	0.505	0.636	0.144	0.661	0.549	0.741	0.567	0.621
1016	0.689	0.781	0.704	1	0.561	0.825	0.175	0.757	0.847	0.867	0.710	0.828
1030	0.451	0.675	0.505	0.561	1	0.522	0.664	0.727	0.605	0.566	0.364	0.399
1038	0.805	0.779	0.636	0.825	0.522	1	0.161	0.726	0.838	0.769	0.798	0.640
1043	0.071	0.348	0.144	0.175	0.664	0.161	1	0.492	0.261	0.205	0.015	0.002
1056	0.519	0.664	0.661	0.757	0.727	0.726	0.492	1	0.749	0.760	0.600	0.673
1057	0.649	0.730	0.549	0.847	0.605	0.838	0.261	0.749	1	0.839	0.614	0.668
1063	0.688	0.793	0.741	0.867	0.566	0.769	0.205	0.760	0.839	1	0.718	0.782
1072	0.807	0.651	0.567	0.710	0.364	0.798	0.015	0.600	0.614	0.718	1	0.711
1075	0.505	0.585	0.621	0.828	0.399	0.640	0.002	0.673	0.668	0.782	0.711	1

Fig. 19. A correlation matrix consisting of participants from the gradient boosted trees, support vector machine, and outlier detection workflow.

V. SUMMARY

We presented the reproducible workflows based on data mining and machine learning methods to support advanced EMA studies. Such data analysis pipelines add additional perspectives and new insights to standard methods, enrich our understanding of mobile survey data and their associated mechanisms, and offer clear opportunities for improving clinical practice. Also, we discovered important relationships and patterns in a real-life cannabis study dataset and built models to predict outcomes of interests based on relevant factors. The capabilities of our workflows support data processing, development, and reproducibility for mobile sensing projects, enabling researchers to improve the rigor and efficiency of the data analysis process. Some considerations and challenges remain, such as improving the accuracy of predictive models, and as research in this field continues to grow, methodological advances are urgently needed. In the future, we will develop R and Python scripts that can be executed in reproducible virtual environments, providing researchers with a transparent, familiar, and efficient analysis environment.

REFERENCES

[1] W. Morrison, L. Guerdan, J. Kanugo, T. Trull, and Y. Shang, "TigerAware: An Innovative Mobile Survey and Sensor Data Collection and Analytics System," *2018 IEEE Third International Conference on Data Science in Cyberspace (DSC)*, pp. 115–122, Jun. 2018, doi: 10.1109/dsc.2018.00025.

[2] J. E. Schwartz and A. A. Stone, "Strategies for analyzing ecological momentary assessment data," *Health Psychology*, vol. 17, no. 1, pp. 6–16, Jan. 1998, doi: 10.1037/0278-6133.17.1.6.

[3] J. Vega *et al*., "Reproducible Analysis Pipeline for Data Streams: Open-Source Software to Process Data Collected With Mobile Devices," *Frontiers in Digital Health*, vol. 3, Nov. 2021, doi: 10.3389/fdgth.2021.769823.

[4] A. M. Wycoff, J. Metrik, and T. J. Trull, "Affect and cannabis use in daily life: a review and recommendations for future research," *Drug and Alcohol Dependence*, vol. 191, pp. 223–233, Oct. 2018, doi: 10.1016/j.drugalcdep.2018.07.001.

[5] J. Koster and S. Rahmann, "Snakemake--a scalable bioinformatics workflow engine," *Bioinformatics*, vol. 28, no. 19, pp. 2520–2522, Aug. 2012, doi: 10.1093/bioinformatics/bts480.

[6] J. de Andrade Silva and E. R. Hruschka, "Extending k-Means-Based Algorithms for Evolving Data Streams with Variable Number of Clusters," *2011 10th International Conference on Machine Learning and Applications and Workshops*, Dec. 2011, doi: 10.1109/icmla.2011.67.

[7] A. A. Neath and J. E. Cavanaugh, "The Bayesian information criterion: background, derivation, and applications," *Wiley Interdisciplinary Reviews: Computational Statistics*, vol. 4, no. 2, pp. 199–203, Dec. 2011, doi: 10.1002/wics.199.

[8] H. Abdi and L. J. Williams, "Principal component analysis," *Wiley Interdisciplinary Reviews: Computational Statistics*, vol. 2, no. 4, pp. 433–459, Jun. 2010, doi: 10.1002/wics.101.

[9] M. C. Cieslak, A. M. Castelfranco, V. Roncalli, P. H. Lenz, and D. K. Hartline, "t-Distributed Stochastic Neighbor Embedding (t-SNE): A tool for eco-physiological transcriptomic analysis," *Marine Genomics*, vol. 51, p. 100723, Jun. 2020, doi: 10.1016/j.margen.2019.100723.

[10] S. Ramaswamy, R. Rastogi, and K. Shim, "Efficient algorithms for mining outliers from large data sets," *ACM SIGMOD Record*, vol. 29, no. 2, pp. 427–438, Jun. 2000, doi: 10.1145/335191.335437.

[11] Jun Gao, Weiming Hu, Wei Li, Zhongfei Zhang, and Ou Wu, "Local Outlier Detection Based on Kernel Regression," *2010 20th International Conference on Pattern Recognition*, Aug. 2010, doi: 10.1109/icpr.2010.148.

[12] V. Veloso de Melo and W. Banzhaf, "Automatic feature engineering for regression models with machine learning: An evolutionary computation and statistics hybrid," *Information Sciences*, vol. 430–431, pp. 287–313, Mar. 2018, doi: 10.1016/j.ins.2017.11.041.

[13] B. Böken, "On the appropriateness of Platt scaling in classifier calibration," *Information Systems*, vol. 95, p. 101641, Jan. 2021, doi: 10.1016/j.is.2020.101641.

[14] J. Kim, K. Choi, G. Kim, and Y. Suh, "Classification cost: An empirical comparison among traditional classifier, Cost-Sensitive Classifier, and MetaCost," *Expert Systems with Applications*, vol. 39, no. 4, pp. 4013–4019, Mar. 2012, doi: 10.1016/j.eswa.2011.09.071.

[15] M. W. Browne, "Cross-Validation Methods," *Journal of Mathematical Psychology*, vol. 44, no. 1, pp. 108–132, Mar. 2000, doi: 10.1006/jmps.1999.1279.

[16] J. A. Khan, S. Van Aelst, and R. H. Zamar, "Fast robust estimation of prediction error based on resampling," *Computational Statistics & Data Analysis*, vol. 54, no. 12, pp. 3121–3130, Dec. 2010, doi: 10.1016/j.csda.2010.01.031.

[17] M. H. Zweig and G. Campbell, "Receiver-operating characteristic (ROC) plots: a fundamental evaluation tool in clinical medicine," *Clinical Chemistry*, vol. 39, no. 4, pp. 561–577, Apr. 1993, doi: 10.1093/clinchem/39.4.561.

[18] P. Sharma, D. Singh, and A. Singh, "Classification algorithms on a large continuous random dataset using rapid miner tool," *2015 2nd International Conference on Electronics and Communication Systems (ICECS)*, Feb. 2015, doi: 10.1109/ecs.2015.7125003.

[19] M. A. Ghorbani, F. Salmasi, M. K. Saggi, A. S. Bhatia, E. Kahya, and R. Norouzi, "Deep learning under H2O framework: A novel approach for quantitative analysis of discharge coefficient in sluice gates," *Journal of Hydroinformatics*, vol. 22, no. 6, pp. 1603–1619, Sep. 2020, doi: 10.2166/hydro.2020.003.

[20] M. R. Zafar and N. Khan, "Deterministic Local Interpretable Model-Agnostic Explanations for Stable Explainability," *Machine Learning and Knowledge Extraction*, vol. 3, no. 3, pp. 525–541, Jun. 2021, doi: 10.3390/make3030027.

[21] S. M. Hyman and R. Sinha, "Stress-related factors in cannabis use and misuse: Implications for prevention and treatment," *Journal of Substance Abuse Treatment*, vol. 36, no. 4, pp. 400–413, Jun. 2009, doi: 10.1016/j.jsat.2008.08.005.