Worker Paid Leave Usage Simulation (Worker PLUS) Model

Issue Brief: Model Testing

January 2021

OVERVIEW

This brief summarizes the model testing efforts and results for the Worker Paid Leave Usage Simulation (Worker PLUS) model developed by IMPAQ International (IMPAQ) and the Institute for Women's Policy Research for the Chief Evaluation Office at the U.S. Department of Labor.[i] In the Worker PLUS model, the simulation results, such as the statuses of taking leaves and needing leaves, are produced by first training predictive models using the 2018 Family and Medical Leave Act (FMLA) Employee Survey microdata, and then applying the trained models on the state samples of the 2014–2018 American Community Survey (ACS) Public Use Microdata Sample (PUMS). Hence, we test the model by performing three steps. First, we conduct internal cross-validation using the 2018 FMLA data, where model validity is established by splitting the FMLA dataset into a training subsample for training predictive models and a testing subsample for evaluating the performance of the models in predicting population-level counts of leave takers and leave needers and number of leaves. Second, we perform a similar internal cross-validation to evaluate the performance of the models in predicting individual-level outcomes, including leave-taking and leave-needing statuses for different leave reasons. Third, we conduct an external validation of the simulated program benefit outlays, which is computed based on the simulated leave taker and leave needer populations in the ACS PUMS. The simulated outlays are compared against actual outlay data published by state-run paid leave programs in California, New Jersey, and Rhode Island, the three states where sufficient data have been reported and can be used for model testing.

To facilitate understanding of the potential impacts of different policy alternatives on workers' leave-taking behaviors and program costs, the U.S. Department of Labor's Chief Evaluation Office contracted with IMPAQ International, and its partner Institute for Women's Policy Research (IWPR), to develop the Worker Paid Leave Usage Simulation (Worker PLUS) model, an open-sourced microsimulation tool based on public microdata and predictive modeling. The model and other relevant materials are publicly available at [*hyperlink*].

In this issue brief, we report findings from testing and validating the Worker PLUS model using data from the 2018 U.S. Department of Labor Family and Medical Leave Act Employee Survey; the 2014–2018 American Community Survey Public Use Microdata Sample; and benefit outlay data published by state paid leave programs in California, New Jersey, and Rhode Island. This brief also discusses the implication of the model testing results on choice of simulation methods, assessment of program take-up rates, and estimation of program benefit outlays.

For a comprehensive assessment of model performance, each type of test is repeated for different simulation methods offered by the Worker PLUS model. For cross-validation, we also include model performance from random draw, which predicts leave-taker and leave-needer statuses purely at random, and thus can be considered a baseline method.

UNDERSTANDING THE NEED FOR DIFFERENT SIMULATION METHODS

Before discussing the model testing results, we first provide a brief introduction to the intuition behind each simulation method offered by the model, including the traditional logistic regression and other machine learning-based algorithms. During the simulation process, all simulation methods have a common goal: classification. Examples include classifying an eligible worker as either a leave taker for the reason of his or her own illness or as a worker with unmet leave needs due to financial constraint. Most classification tasks of the model are binary, i.e., they classify a worker as positive vs. negative for a given outcome of interest.[ii]

Traditionally, the binary classification tasks are handled by logistic regression, where the outcome variable is the logit of the probability of interest (e.g., probability of being a leave taker for a given reason). Unlike probability (bounded by 0 and 1), the logit term has no bounds and hence can be flexibly modeled and predicted by explanatory variables (e.g., worker demographics and job characteristics). Despite this advantage, logistic regression may not always maximize the predictive power of models because of

(i) non-linearity in explanatory variables, e.g., effect of age on probability of leave taking varies by occupation;

(ii) missing data, e.g., unreported income data commonly seen in surveys;
(iii) large number of categories in variables, e.g., the many industry and occupation codes, which lead to inconsistent estimates;
(iv) sensitivity to outliers, e.g., model estimates can be easily distorted by a few workers with extremely high wages; and
(v) overfitting, i.e., model is too complex, resulting in spuriously good cross-validation performance but poor out-of-sample predictions.

In modern data science, the above issues have been addressed by different machine learning-based classifiers.[iii] While the details of construction and implementation of these classifiers are beyond the scope of this brief, a summary of the strength of each classifier should be helpful for understanding the motivation of considering them as alternatives to traditional linear methods such as logistic regression. The summary is provided in **Exhibit 1**. The summary should be used only for the purpose of motivating the use of these methods. It does not necessarily mean that methods that address more issues perform better than others. Instead, the performance of different classifiers is highly dependent upon data input and performance metrics, and machine learning methods may also be outperformed by logistic regression. Therefore, understanding the relative performance of the candidate simulation methods offered by the Worker PLUS model remains an empirical question, and it is precisely the aim of this brief.

Exhibit 1: Issues Addressed by Machine Learning Classifiers

	Logistic Regression Regularized	*k* Nearest Neighbor (KNN)	Naïve Bayes	Random Forest	XGBoost (XGB)	Ridge Regression (Ridge)	Support Vector Classifier (SVC)
Non-linearity		✓	✓	✓	✓		✓
Missing data		✓			✓		
Many categories			✓	✓	✓		
Sensitivity to outliers	✓					✓	✓
Overfitting	✓				✓	✓	

In the simulation engine of the Worker PLUS model, the traditional and machine learning simulation methods are implemented by a suite of the Python packages (for the Python simulation engine) and R libraries (for the R simulation engine). The exact versions of these packages and libraries used for model testing are provided in Appendix A. Model users should note that both Python and R are open-source programming languages; hence, these packages and libraries can be updated by contributors over time. Therefore, for replicating the results in this brief, we recommend using the links provided in Appendix A to check for any substantial changes that have been made to the algorithms for these packages and libraries. For exact replication, we recommend using the same versions as those listed in Appendix A.

HOW ACCURATE ARE MODEL PREDICTIONS FOR LEAVE-TAKING BEHAVIORS AT THE POPULATION LEVEL?

Key Finding: Logit Generalized Linear Model (GLM), Logit Regularized, Naïve Bayes, and XGBoost are the methods that can most closely predict total number of leave takers, leave needers, and leaves. Larger deviations are found on other methods, but the over- or underestimations of these quantities are unlikely to affect program outlay estimates and program take-up estimation given the low take-up rates of paid leave programs.

Our model testing focuses on two types of workers identified from the 2018 FMLA Employee Survey, the *leave takers* and *leave needers.* Leave takers are the workers who take leaves from work for any number of days during a year, and leave needers are those who have unmet leave needs. A leave needer can be either a leave taker or a non-leave taker. The variable used to define them is *leave_cat*. Leave takers are workers with *leave_cat* value equal to 1 (leave taker only) or 4 (leave taker and leave needer), and leave needers are workers with *leave_cat* value equal to 2 (leave needer only) or 4 (leave taker and leave needer). Among the 4,470 workers in the survey data, 1,829 are leave takers, and 912 are leave needers. Using the population weight variable *combo_trimmed_weight*

from the survey data, we estimate an underlying worker population of 21.5 million leave takers and 10.0 million leave needers in the country.

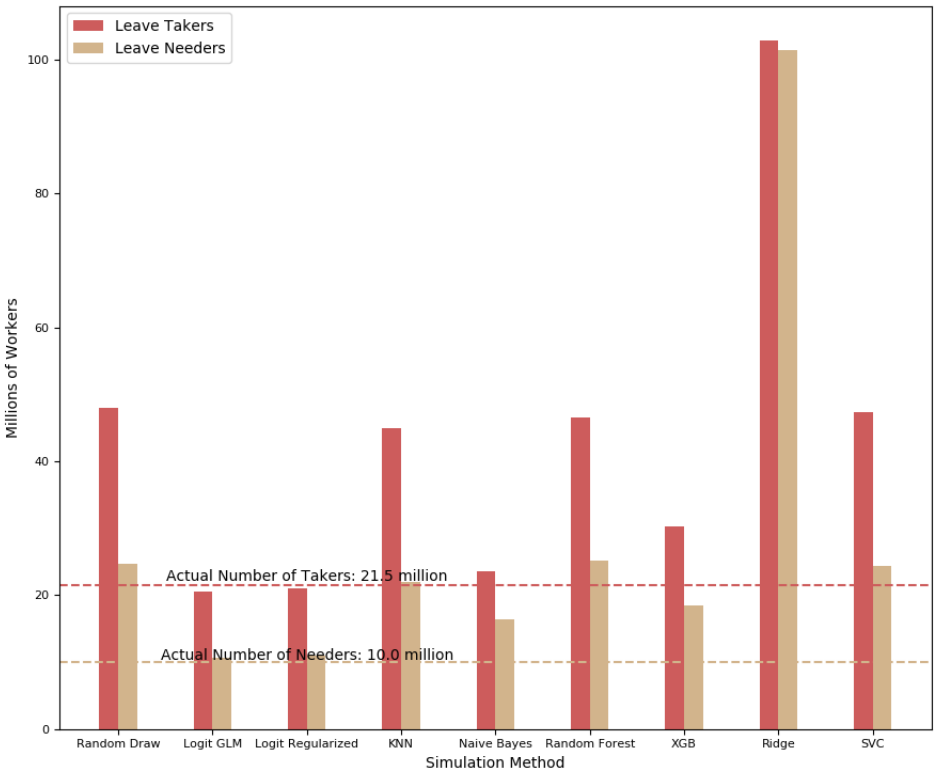
With leave takers and leave needers defined, we then use weighted *k*-fold cross-validation to evaluate the predictive performance of the simulation methods. For evaluating performance at the population level, this validation strategy is implemented in the following steps:

1. Randomly splitting the entire sample (i.e., the 2018 FMLA Employee Survey data) into *k* subsamples that have (roughly) the same sample size. These subsamples are called *folds*.
2. Setting one of the *k* folds aside as the testing sample, and using the rest of the folds (*k–1*) together as the training sample to train a predictive model, for a given outcome of interest (e.g., leave-taker status) and a given simulation method (e.g., *Logistic Regression Regularized*).
3. Using the trained model to make predictions on the testing sample and derive the population-level prediction (e.g., predicted total number of leave takers) for the testing sample.
4. Repeating Steps 2 and 3 and aggregating the predictions across *k* testing samples to obtain the population-level prediction for the entire sample.
5. Repeating Steps 1 through 4 multiple times and obtaining the average population-level prediction over these iterations. This step aims to mitigate the effect of random noise on predictions.
6. Comparing the population-level predictions to actual numbers.

In practice, we set $k = 10$ to maintain a sufficient training sample size (i.e., about 90% of entire 2018 FMLA data), and to manage total runtime (i.e., a total of 10 models need to be trained during Steps 1 through 4 for each simulation method). To ensure representativeness, we also adopt the sample weights in model training and computing the population-level predictions.

Exhibit 2 shows the predicted total number of leave takers and leave needers using the 2018 FMLA data under weighted 10-fold cross-validation for different simulation methods. Model tuning is performed for all simulation methods to ensure maximized performance while maintaining the needed logic (e.g., males cannot take maternity leave).[iv] We also include a baseline method *Random Draw*, which represents a random prediction by preserving the mean value of the training sample in the testing sample. The two horizontal lines in **Exhibit 2** show actual numbers of takers (21.5 million) and needers (10.0 million) estimated from the entire FMLA sample. The comparison shows that *Logit GLM*, *Logit Regularized,* and *Naïve Bayes* are the methods that produce the closest estimates to the actual numbers, while the other classifiers all result in substantial overestimation.

Exhibit 2: Population-Level Validation Results, Worker Counts



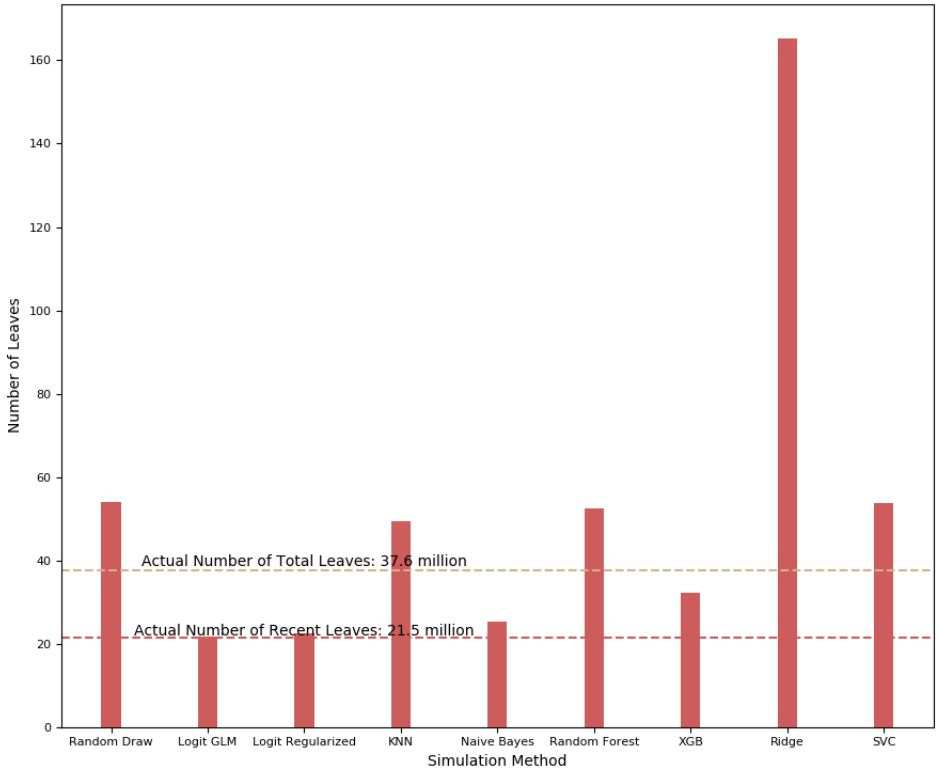
Note: Worker counts are the sum of predicted worker counts over 10 folds of testing samples from the 10-fold cross-validation using 2018 FMLA Employee Survey data. *Random Draw* represents a random prediction by preserving the mean value of the training sample in the testing sample. *Logit GLM* represents traditional logistic regression, which is implemented by the *statsmodels* package in Python. *XGB* represents XGBoost classifier, which is implemented by the *xgboost* package in Python. All other simulation methods are implemented by the *scikit-learn* package in Python. All actual numbers are estimated from the 2018 FMLA data.

Exhibit 3 shows the predictions of total number of leaves taken.[v] There are two benchmarks for leave counts, (i) the *Actual Number of Recent Leaves* (21.5 million), and (ii) the *Actual Number of Total Leaves* (37.6 million). The first measure is the one used for our modeling, as the most recent leave is the only reliable recall of leave taking in the FMLA survey data. It serves as a lower-bound benchmark for our prediction, since each worker can have at most one most recent leave in the sample, while this constraint is not imposed for leave prediction in our model to account for multi-leave takers. The second measure is derived directly from a question in the FMLA survey that asks for total number of leaves taken, including the six leave types considered in the model, plus other less common leave types.[vi] Therefore, the second measure should be considered as an upper-bound benchmark for our model. The comparisons show that *Logit Regularized*, *Naïve Bayes*, and *XGBoost* are the simulation methods that lead to predictions falling between the two bounds. *Logit GLM* (the traditional logistic regression) underestimates the leave count, while the other classifiers overestimate.

The above results show that not all simulation methods result in population-level predictions that closely track population estimates from the FMLA survey data. However, we note that the under- or overestimations would generally have limited effect on program outlay estimates, because the population-level totals of leave takers and leave needers only provide a pool of *potential* participants in a paid leave program. This worker pool is used for drawing program participants in simulation, and the pool size does not affect the pre-determined program take-up rates. Given the low take-up rates seen in existing programs, the pool size would remain a secondary factor for determining program outlays as long as the target number of program participants can be drawn. The primary factors affecting program outlays would be (i) program eligibility rules (which are common across simulation methods, given the same program), (ii) number of program participants (already calibrated to actual program data in model testing), (iii) worker wages, (iv) tendency of

workers to extend their leave after program implementation, and (v) leave lengths under the program. Later in this brief when we perform model testing for the program benefit outlay predictions, we will show how factors (iii) through (v) lead to the variations in program outlay estimates across simulation methods.

Exhibit 3: Population-Level Validation Results, Leave Counts



Note: Leave counts are the sum of predicted leaves over 10 folds of testing samples from the 10-fold cross-validation using 2018 FMLA Employee Survey data. Random Draw represents a random prediction by preserving the mean value of the training sample in the testing sample. Logit GLM represents traditional logistic regression, which is implemented by the *statsmodels* package in Python. XGB represents XGBoost classifier, which is implemented by the *xgboost* package in Python. All other simulation methods are implemented by the *scikit-learn* package in Python. All actual numbers are estimated from the 2018 FMLA data.

HOW ACCURATE ARE MODEL PREDICTIONS FOR LEAVE-TAKING BEHAVIORS AT THE INDIVIDUAL LEVEL?

Key Finding: Predictive performance varies substantially across outcomes to be predicted. In general, predictive performance is stronger for maternity disability than other leave reasons and for leave needs than leave taking. Overall predictive power is not strong at the individual level due to limitations in data available from both the FMLA Employee Survey and the ACS.

To assess the individual-level predictive performance, we consider three model performance measures:

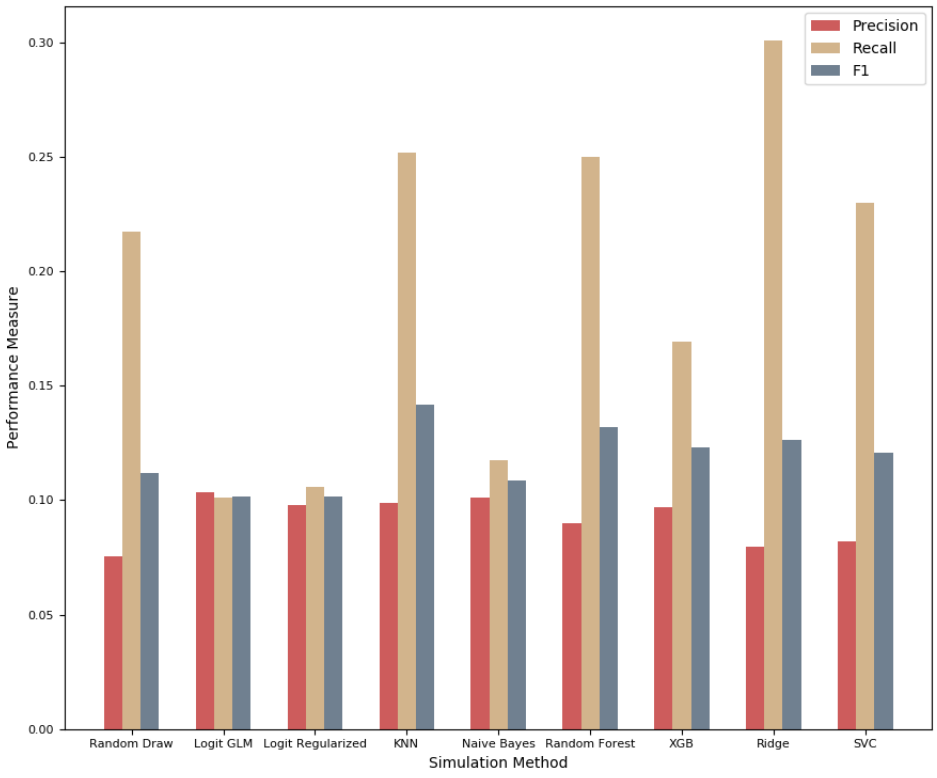
- *Precision*, defined as total number of true positives predicted (e.g., number of predicted leave takers that are indeed leave takers) divided by total number of positive predictions (e.g., total number of leave takers predicted) in data. The total number of positive predictions is the sum of the number of true positives and the number of false positives.
- *Recall*, defined as total number of true positives predicted divided by total number of positive cases (e.g., total number of actual leave takers).
- The *F1*-score, defined as $F1 = \frac{1}{\frac{1}{2}\times\left(\frac{1}{Precision}+\frac{1}{Recall}\right)}$, namely, the harmonic mean of *Precision* and *Recall*.

We are interested in *Precision* and *Recall* because they both focus on true positive cases predicted, e.g., correctly identifying individual leave takers and needers from the worker sample. This is particularly relevant for events that do not commonly occur, such as workers taking medical leaves for themselves or for a family member, or workers having unmet leave needs. In fact, the 2018 FMLA survey estimate suggests that about 80% of the worker population did not take any leave or have any leave needs over the past year. Therefore, predicting true negatives (e.g., non-leave takers) does not indicate good predictive performance of a model (e.g., even a dummy classifier that naively predicts no leave taking for *all workers* would achieve 80% accuracy). Instead, model performance would largely rely upon the capability of predicting true positives. We introduce the *F1*-score measure due to the trade-off between achieving high *Precision* and *Recall*. Namely, a classifier could make one lucky prediction of true positive while predicting negative for all the rest of the cases, achieving 100% *Precision* but extremely low *Recall*, or alternatively, predict all cases as positive and achieve 100% *Recall* but low *Precision*. The formulation of *F1* ensures a balanced metric, which would be greater not only for greater *Precision* and *Recall* values, but also for *smaller* differences between the two values.

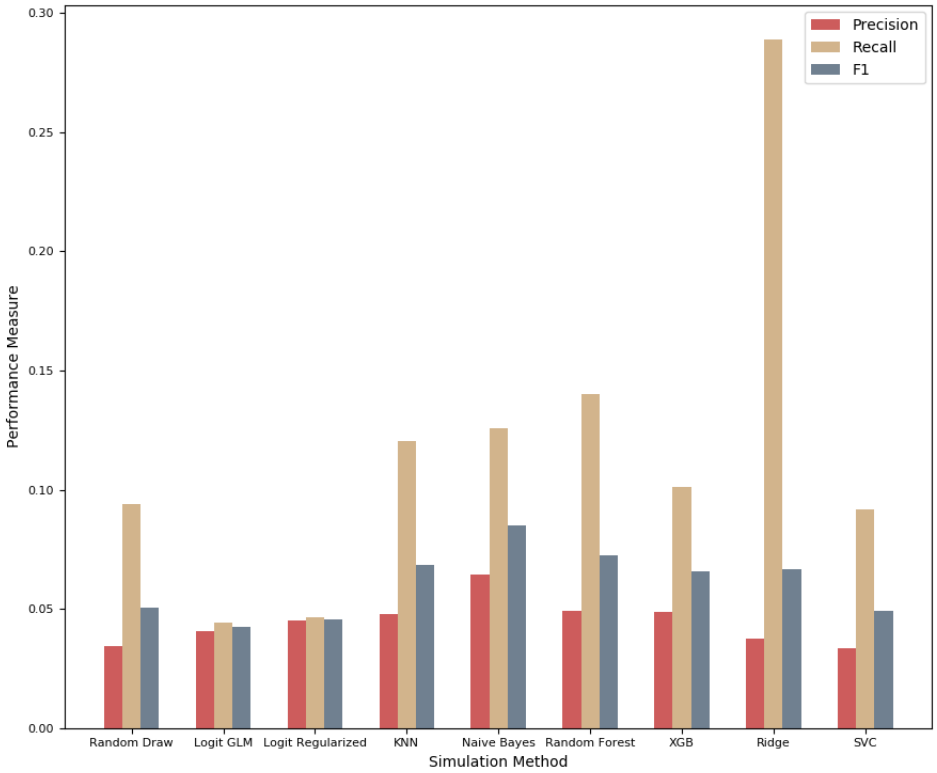
Exhibit 4 shows the model's predictive performance metrics for leave takers and leave needers for the two main leave types, own illness and maternity disability, accounting for about 60% of leaves taken and unmet leave needs among workers and over 80% of program outlays.[vii] The three performance metrics are plotted for each simulation method as well as the *Random Draw* baseline. Overall, the predictive performance at the individual level is not strong given the limited number of explanatory variables available from *both* the FMLA survey and the ACS. The performance metrics also suggest that information on demographics and jobs in current FMLA and ACS data has limited explanatory power in relation to leave taking and leave needs. Therefore, the model results (e.g., the simulated group of leave takers and leave needers, the simulated leave lengths, and the simulated program benefit outlays) are primarily driven by factors other than these labor force characteristics. These factors may include program eligibility, program take-up behavior, and the length of additional leaves taken by the eligible workers.

Exhibit 4: Individual-Level Validation Results, Leave Takers and Leave Needers Due to Own Illness or Maternity Disability

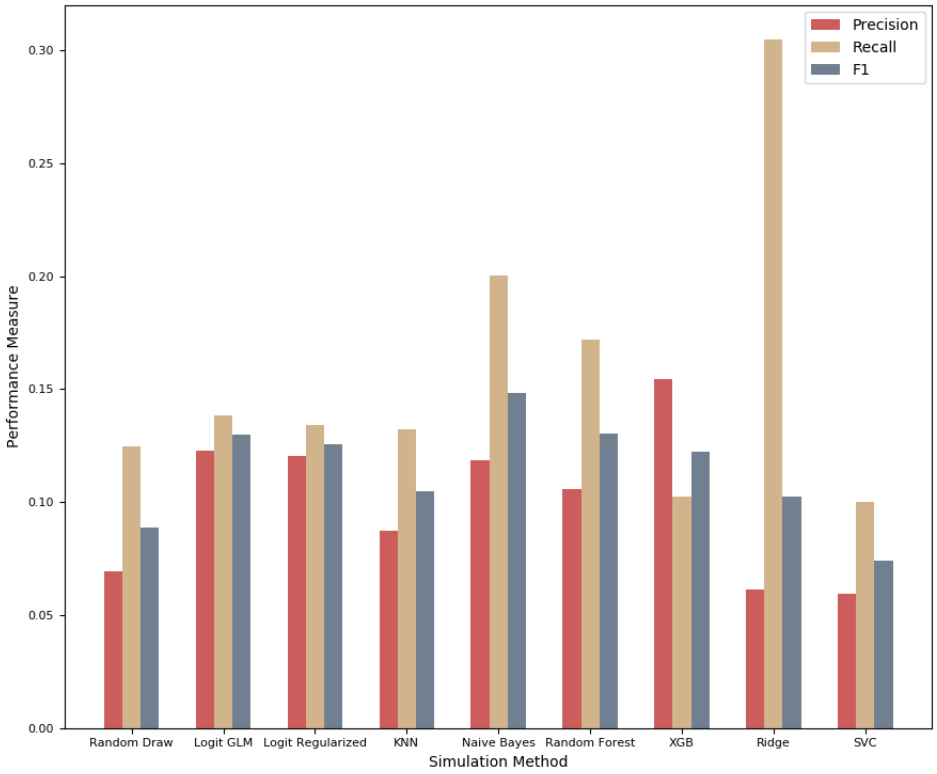
(a) Leave Takers Due to Own Illness



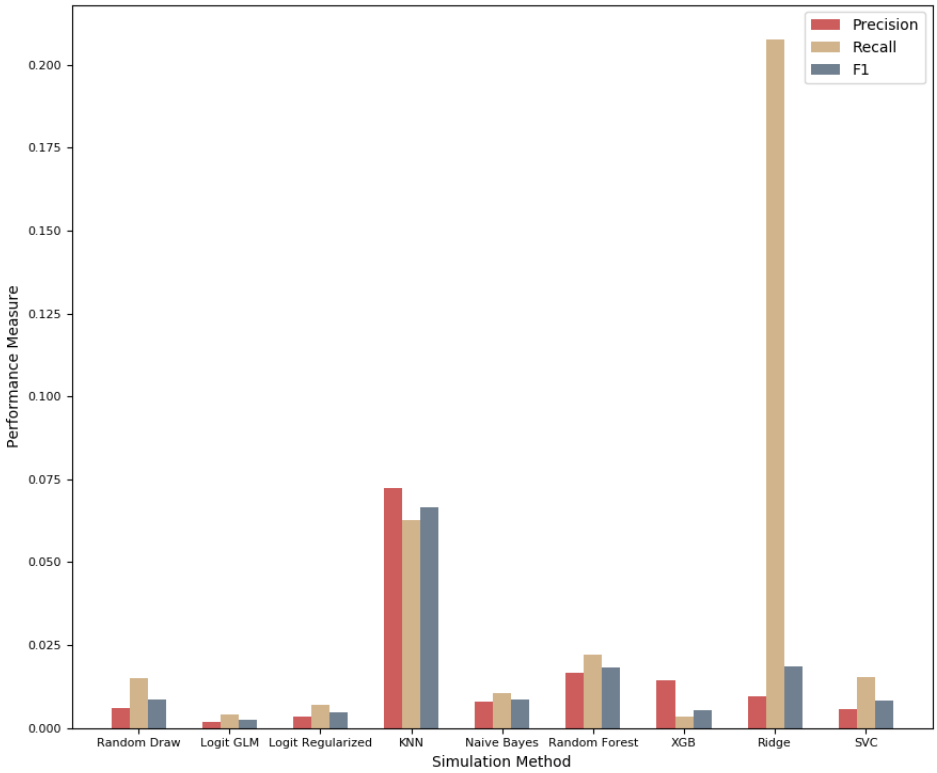
(b) Leave Needers Due to Own Illness



(c) Leave Takers Due to Maternity Disability



(d) Leave Needers Due to Maternity Disability



Note: Predictive performance metrics are obtained from repeating the 10-fold cross-validation 10 times using 2018 FMLA Employee Survey data. Random Draw represents a random prediction by preserving the mean value of the training sample in the testing sample. Logit GLM represents traditional logistic regression, which is implemented by the *statsmodels* package in Python. XGB represents XGBoost classifier, which is implemented by the *xgboost* package in Python. All other simulation methods are implemented by the *scikit-learn* package in Python. *Precision* is defined as total number of true positives predicted divided by total number of positive predictions. *Recall* is defined as total number of true positives predicted divided by total number of positive cases. *F1*-score is defined as the harmonic mean of *Precision* and *Recall*.

The gain in predictive performance from simulation modeling is measured by the maximum increase in *F1*-score across different classifiers compared to the *F1*-score of *Random Draw.* Across the predicted outcomes, the gain is the largest for predicting leave takers due to maternity disability (**Exhibit 4** [c]), where the *F1*-score is 0.148 under the *Naïve Bayes* classifier, a 51% improvement compared to the *F1*-score of 0.098 under *Random Draw*. The gain is similar for predicting leave needers due to own illness (47%, with 0.085 under *Naïve Bayes* versus 0.058 under *Random Draw,* **Exhibit 4** [b]) but is lower for predicting leave takers due to own illness (15%, with 0.142 under *KNN* versus 0.123 under *Random Draw*, **Exhibit 4** [a]). This is not surprising, given that certain demographic variables (e.g., age, education, family income) can be more predictive of childbearing decisions but less predictive of short-term health status. Likewise, leave taking generally can occur for any workers, while having unmet leave needs is often associated with disadvantaged status such as lower wages, part-time jobs, and lower educational attainment.[viii]

The above results, however, do not directly translate to a larger gain from simulation modeling for predicting leave needers due to maternity disability. Although **Exhibit 4** (d) shows that the *KNN* classifier achieves a 230% improvement in the *F1*-score from the *Random Draw* baseline (0.066 versus 0.020), it also shows that all the other classifiers cannot outperform *Random Draw*. The lack of gain from simulation modeling in general for this outcome originates from the fact that the leave needer status due to maternity disability is extremely rare among workers (34 among 4,466 surveyed workers in the 2018 FMLA data), making it difficult to be predicted based on a limited set of worker covariates.

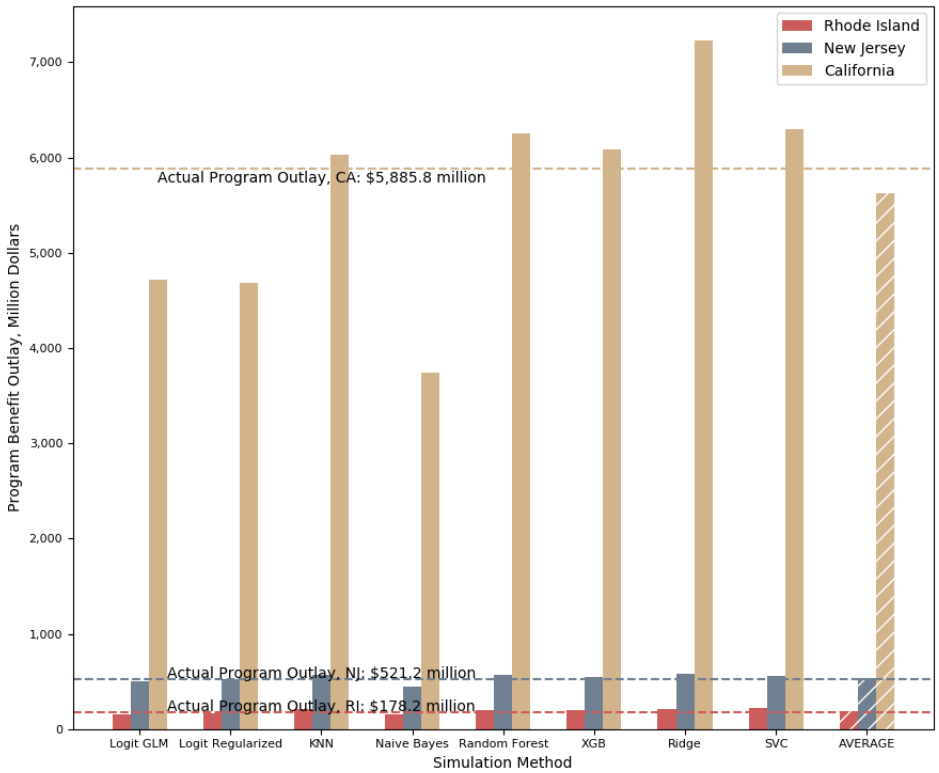
Lastly, we note that the *Recall* score is the greatest for the *Ridge* classifier for all outcomes. This is due to the many positive predictions (leave taker and leave needer statuses) made by *Ridge*, consistent with the greatest leave taker and needer counts and leave counts predicted by this method as shown in **Exhibit 2** and **Exhibit 3**. However, the high *Recall* score *alone* has limited impact on increasing the *F1*-score of *Ridge*, since the *Precision* remains low. This verifies the *F1*-score should be considered as the primary performance metric for evaluating model testing results at the individual level.

HOW ACCURATE ARE MODEL PREDICTIONS FOR BENEFIT OUTLAYS FOR PAID LEAVE PROGRAMS?

Key Finding: In the majority of cases tested, the model predicts program benefit outlays that closely track with actual outlays, with a deviation up to 15%. Larger deviations (over 20%) from actual outlays are found with Logit Regularized, Naïve Bayes, and Ridge when estimating outlays in California and with Ridge and SVC when estimating outlays in Rhode Island. The variations in predicted outlays across simulation methods can be reconciled by the difference in a set of predicted intermediate model outcomes, including workers' wages, their tendency to take longer leaves after program implementation, and lengths of leaves taken under the program.

To compare the predictions of the simulation model against actual state leave program outlays, we use benefit outlay data published in annual reports by California, New Jersey, and Rhode Island programs during 2014–2018, corresponding to the sample period of the ACS data input for model testing. **Exhibit 5** shows the comparison results between the actual and simulated outlays for each simulation method. The comparison also includes the average simulated outlays across all simulation methods. Among the 24 combinations of states and simulation methods, 12 lead to a deviation in predicted benefit outlays of up to 10% from the actual ones, 16 lead to a deviation of up to 15%, and 19 lead to a deviation of up to 20%. Larger deviations (over 20%) from actual outlays are found with *Logit Regularized*, *Naïve Bayes*, and *Ridge* when estimating outlays in California and with *Ridge* and *SVC* when estimating outlays in Rhode Island.

Exhibit 5: Simulated vs. Actual Program Benefit Outlays by Simulation Method



	California		New Jersey		Rhode Island	
	Amount	% Difference from Actual	Amount	% Difference from Actual	Amount	% Difference from Actual
Actual Program Benefit Outlays	$5,885.8	-	$521.2	-	$178.2	-
Logit GLM	$4,717.9	-19.8%	$505.4	-3.0%	$157.4	-11.6%
Logit Regularized	$4,688.2	-20.3%	$525.9	0.9%	$165.2	-7.3%
k Nearest Neighbor (KNN)	$6,025.8	2.4%	$564.7	8.4%	$205.7	15.4%
Naïve Bayes	$3,738.9	-36.5%	$443.9	-14.8%	$155.7	-12.6%
Random Forest	$6,251.6	6.2%	$564.7	8.4%	$203.5	14.2%
XGBoost (XGB)	$6,080.2	3.3%	$550.3	5.6%	$194.6	9.2%
Ridge	$7,231.9	22.9%	$578.9	11.1%	$214.9	20.6%
Support Vector Classifier (SVC)	$6,303.9	7.1%	$561.3	7.7%	$219.3	23.1%
Average of All Simulation Methods	$5,629.8	-4.3%	$536.9	3.0%	$189.5	6.4%

Note: Estimates of program outlays are produced by Worker PLUS model Python engine via the graphical user interface, using 2018 FMLA Employee Survey data and 2014–2018 ACS PUMS state samples for California, New Jersey, and Rhode Island. Actual outlay data are obtained from state program annual reports.[ix] All outlays are in 2018 million dollars. Logit GLM represents traditional logistic regression, which is implemented by the *statsmodels* package in Python. XGB represents XGBoost classifier, which is implemented by the *xgboost* package in Python. All other simulation methods are implemented by the *scikit-learn* package in Python. AVERAGE represents the mean of simulated outlays across all simulation methods.

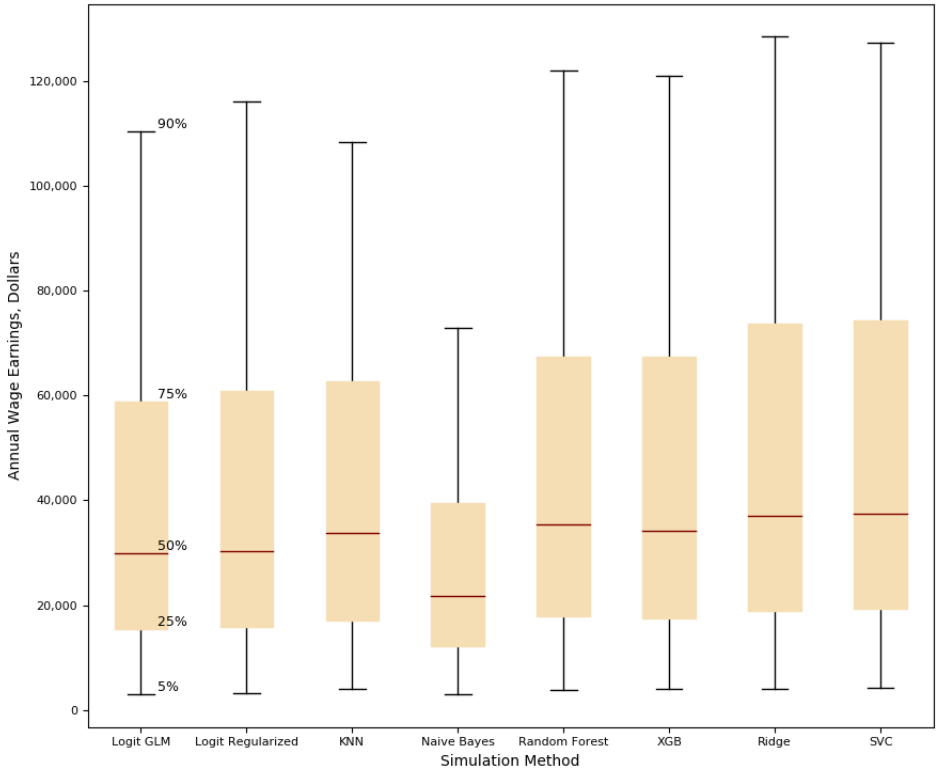
To reconcile these larger deviations in model predictions from actual outlays, we consider the intermediate model outcomes used for outlay aggregation, including (i) the set of eligible workers, (ii) the number of simulated program participants, (iii) the wages of simulated program participants, (iv) participants' tendency to taker longer leaves after program implementation, and (v) the simulated leave lengths under the program. We first note that data components (i) and (ii) do not vary across simulation methods, since the same set of eligibility rules is applied with a given program, and the model calibrates program take-up rates based on actual caseload data. What remains to be investigated are therefore components (iii) through (v).

We plot in **Exhibit 6** the results from analyzing the intermediate model outcomes across simulation methods for simulated program participants who take leave due to their own illness (the most popular leave reason, accounting for over 60% of benefit outlays) in California (the state for which the largest deviations in outlay predictions from actual outlays are seen).[ix] **Exhibit 6** (a) shows the wage percentiles among the participants. The differences in wage distributions are consistent with the differences in outlay predictions for most simulation methods—that is, methods with higher wages also lead to larger outlay estimates. For example, *Logit GLM*, *Logit Regularized*, and *Naïve Bayes* all lead to underestimation above 15% in California, and these simulation methods also simulate participants at the lowest median wage. On the other hand, the wage percentiles are the largest for *Ridge*, the simulation method that leads to the largest magnitude of outlay overestimation. However, the variations in outlay estimates do not seem to be fully accounted for by wage. For example, the wage percentiles are very similar between *Ridge* and *SVC*, but *Ridge* leads to much larger overestimation (22.9%) compared to *SVC* (7.1%), as shown in **Exhibit 5**.

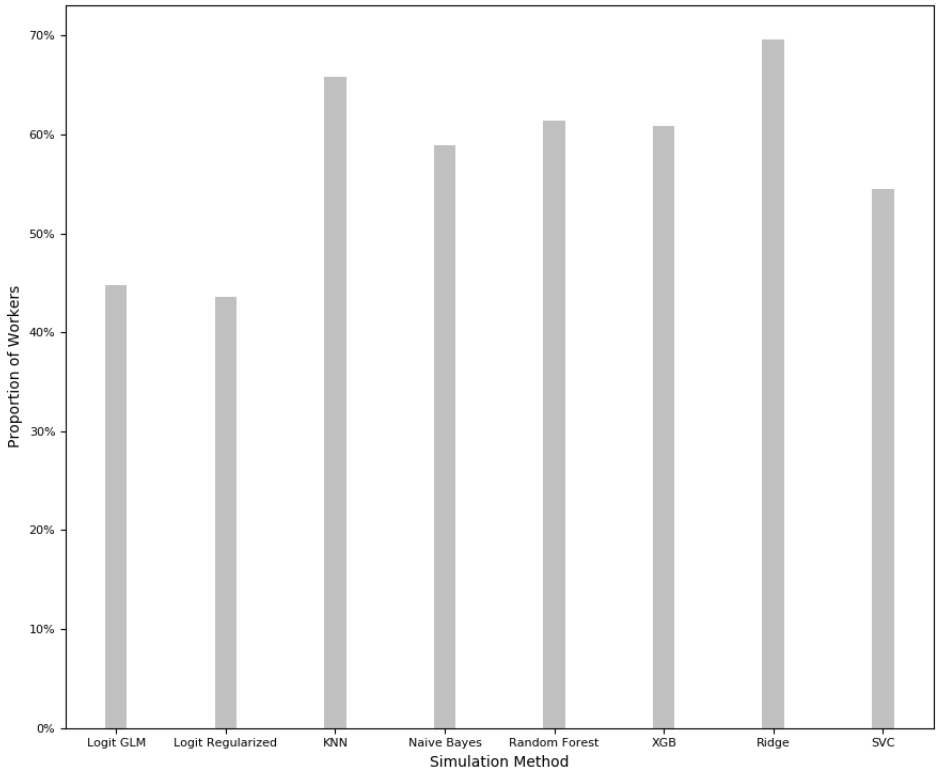
Exhibit 6 (b) and **6** (c) show how the variations in outlay estimates can be further accounted for by workers' tendency to take longer leaves after program implementation, and by the length of leave taken under the program. In particular, the simulation methods that lead to larger outlay estimates are also (i) those that predict a larger proportion of participants who are simulated to take longer leaves after program implementation (**Exhibit 6** [b]), and (ii) those that predict longer leaves under the program compared to other simulation methods (**Exhibit 6** [c]). Both relationships hold for *Ridge* and *SVC*.

Exhibit 6: Analysis of Intermediate Model Outcomes to Account for Variations in Benefit Outlay Estimates across Simulation Methods for Simulated Program Participants Due to Own Illness in California

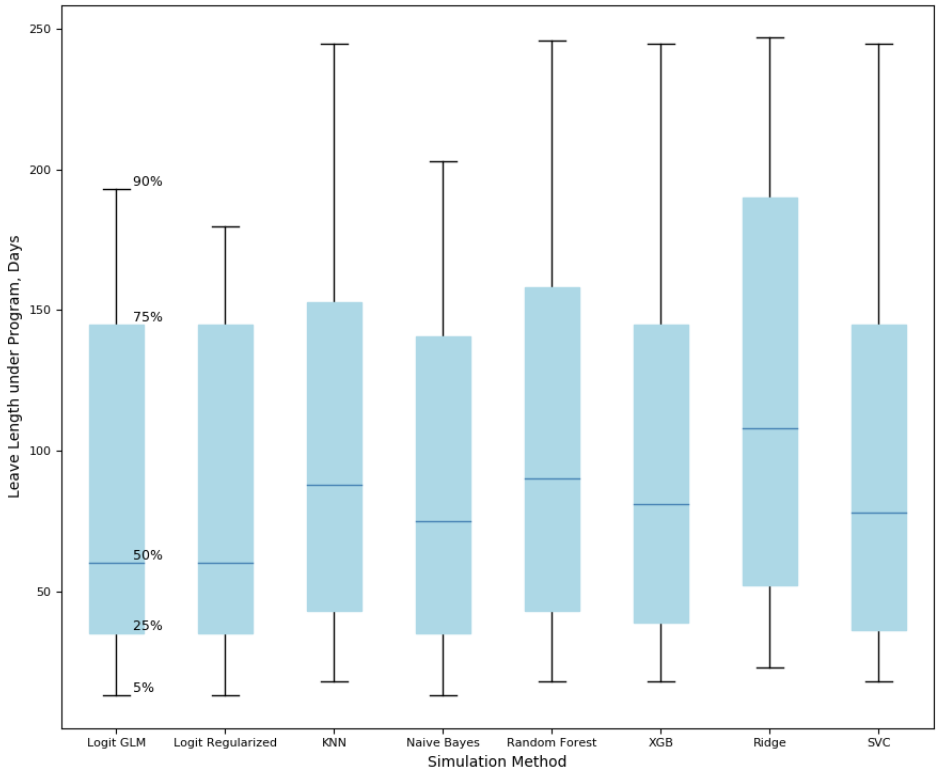
(a) Wage Percentiles



(b) Proportion of Participants with Longer Leave Length after Program Implementation vs. before Implementation



(c) Percentiles of Leave Length under Program



Note: In all exhibits, the subgroup of workers used for producing the estimates are the simulated program participants in California who take leave due to their own illness. In (a), the percentiles are derived from the distribution of annual wages (*wage12*) of the participants. In (b), the proportions are estimated using the indicator variable of whether a worker would extend leave after program implementation (*resp_len*). In (c), the percentiles are derived from the distribution of leave length in a year (*cpl_own*). Simulation results are produced by Worker PLUS model Python engine via the graphical user interface, using 2018 FMLA Employee Survey data and the 2014–2018 ACS PUMS California state sample. *Logit GLM* represents traditional logistic regression, which is implemented by the *statsmodels* package in Python. *XGB* represents XGBoost classifier, which is implemented by the *xgboost* package in Python. All other simulation methods are implemented by the *scikit-learn* package in Python.

CONCLUDING REMARKS

In this brief, we have performed model testing for the different simulation methods offered by the Worker PLUS model. Tests performed include *k*-fold cross-validations using the 2018 FMLA Employee Survey data alone, comparing the model predictions of program benefit outlays against actual program data, and reconciling the differences using intermediate outcomes from simulation. We find the following from the model testing:

- At the population level, the number of leave takers and the number of leave needers can be most accurately predicted by traditional logistic regression (*Logit GLM*) and regularized logistic regression (*Logit Regularized*), while the number of leaves taken can be most accurately predicted by regularized logistic regression (*Logit Regularized*), Naïve Bayes, and XGBoost (*XGB*).
- At the individual level, there is no strong improvement of predictive performance over the baseline method (*Random Draw*) across the simulation methods in the model. Some notable cases include Naïve Bayes for predicting leave takers due to maternity disability (a 51% improvement over baseline) and for predicting leave needers due to own illness (47%), and *k* Nearest Neighbor (*KNN*) for predicting leave takers due to own illness (15%). Overall, the individual-level testing results suggest that information on demographics and jobs in current FMLA and ACS data has limited explanatory power in relation to individual workers' leave

taking and leave needs. This limitation of the model can be addressed by future efforts in improving data collection on worker leaves.

- The comparison of simulated and actual program benefit outlays suggests that estimates from different simulation methods can be used to form lower and upper bounds for the actual outlay. The deviation of the bounds from actual outlay can be up to 20% to 30%, but the average of estimates across simulation methods is closer to the actual outlay data. The deviation of the average estimate from the actual outlay is within 6.4% across the three states tested.
- The variations in the outlay estimates across simulation methods can be reconciled by a set of different intermediate outcomes produced by the model, include workers' wages, their tendency to extend leaves after program implementation, and the leave lengths taken under the program. We find that the simulation methods that predict higher outlays (e.g., the Ridge Classifier, *Ridge*) are precisely those that predict higher wages, a larger proportion of workers extending their leaves, and longer leaves taken under the program. The variations in simulated intermediate outcomes and ultimately the simulated outlays reflect how the simulation methods make predictions differently.

The above model testing results suggest that the multiple simulation methods in the Worker PLUS model can offer a unique flexibility to users, allowing them to customize the simulation method to address their analytical needs. To facilitate choosing the most relevant methods, we provide **Exhibit 7** below as guidance. However, users should note that the suitability of simulation methods to different purposes may change over time when the model itself and the model testing results are being updated with additional data.

Exhibit 7: Guidance in Choosing Simulation Methods for Different Use Cases

	Use Cases				
	Relevant Underlying Population: All Eligible Workers Regardless of Program Participation		Relevant Underlying Population: Program Participants Only		
Simulation Method	Estimate Total Count of Leave Takers/Needers	Estimate Total Count of Leaves	Estimate a Lower Bound of Benefit Outlay	Estimate an Upper Bound of Benefit Outlay	Form an Average Estimate of Benefit Outlay (use simultaneously)
Logit GLM	✓		✓		✓
Logit Regularized	✓	✓			✓
k Nearest Neighbor (KNN)				✓	✓
Naïve Bayes		✓	✓		✓
Random Forest				✓	✓
XGBoost (XGB)		✓		✓	✓
Ridge				✓	✓
Support Vector Classifier (SVC)				✓	✓

Note: The recommendations are made based only on model testing results using the 2018 FMLA Employee Survey data and the 2014–2018 ACS PUMS data. Testing is performed by configuring the Worker PLUS model's Program and Population parameters to existing state programs in California, New Jersey, and Rhode Island.

APPENDIX A: PACKAGES AND LIBRARIES REQUIRED FOR ALL SIMULATION METHODS

Python Packages and R Libraries Needed for All Simulation Methods in the Worker PLUS Model

Simulation Method	Python Package Name	Version	R Library Name	Version
Logistic Regression GLM	statsmodels	0.12.0	survey	4.0
Logistic Regression Regularized	scikit-learn	0.23	glmnet	4.0-2
***k* Nearest Neighbor (KNN)**	scikit-learn	0.23	caret	6.0-86
Naïve Bayes	scikit-learn	0.23	bnclassify	0.4.5
Random Forest	scikit-learn	0.23	randomForest	4.6-14
XGBoost (XGB)	xgboost	1.2.0	xgboost	1.1.1.1
Ridge Regression	scikit-learn	0.23	ridge	2.5
Support Vector Classifier (SVC)	scikit-learn	0.23	e1071	1.7-3

Note: Documentation for the Python *statsmodels* package specifications and version history is available from https://www.statsmodels.org/stable/index.html. Documentation for the Python *scikit-learn* package specifications and version histories is available from https://scikit-learn.org/dev/versions.html. Documentation for the Python *xgboost* package specifications and version histories is available from https://pypi.org/project/xgboost/#history. All R library documentation and version histories are available from https://rdrr.io.

APPENDIX B: GUIDE TO REPLICATING RESULTS IN THIS ISSUE BRIEF

Model users should follow the steps below to replicate the model validation results in this brief.

1. Ensure that all model materials have been downloaded according to the Worker PLUS Model User Manual.[x]
2. Ensure that the files *validate_model.py* and *validate_model_functions.py* are placed in the same directory as other code files, such as *_5a_aux_functions.py* and *Utils.py*.
3. In *validate_model.py*, update
    a. Line 18 as local directory that contains the pre-processed 2018 FMLA employee data file *fmla_clean_2018.csv*, which is a file provided in the *./data/fmla/fmla_2018* folder in original model files;
    b. Line 20 as local directory to store output figures and CSV files, which will contain the underlying numerical results produced by cross-validation; and
    c. Line 22 as local directory to store simulation output folders (see Step 9 for requirements on renaming output folders).
4. Follow the user manual to
    a. Launch the model graphical user interface (GUI).
    b. Turn on the Advanced Parameters button.
    c. Use the 2018 FMLA employee data and the 2014–2018 ACS PUMS data as input files.
    d. Set Random Seed to 12345 and Engine Type to Python.
5. Set state to simulate and apply the corresponding parameters.
    a. In the main panel of the GUI, set State to Simulate to the desired state (CA, NJ, or RI).
    b. Under the Simulation tab of the GUI, set Existing State Program to the same state as in Step 5a. This will auto-fill all the parameters under *Program* and *Population* tabs with the pre-configured parameters for these state programs and populations.
6. In the main panel of the GUI, set Simulation Method to the desired method.
7. Click the Run button to execute the simulation.
8. After the simulation is completed, navigate to the output directory (as specified in Output Directory in the GUI), and choose the latest output folder. The latest output folder can be identified by the folder name, which contains the date stamp and time stamp when the model is executed. For example, the folder named "*output_20200924_115049_main simulation*" contains simulation output files from the simulation executed on September 24, 2020 at 11:50:49 local machine time.
9. Rename the output folder produced by Step 7 in the format "*[state]_[method]*", using the following labels for the state chosen in Step 5 and simulation method chosen in Step 6. For example, if in Step 5 the state chosen is CA (California) and in Step 6 the simulation method chosen is Logistic Regression GLM, then the output folder should be renamed as *ca_logit_glm*.

Output Folder Renaming Labels for States

State Chosen	Label
CA	ca
NJ	nj
RI	ri

Output Folder Renaming Labels for Simulation Methods

Simulation Method Chosen	Label
Logistic Regression GLM	logit_glm
Logistic Regression Regularized	logit_reg
***k* Nearest Neighbor (KNN)**	knn
Naïve Bayes	nb
Random Forest	rf
XGBoost (XGB)	xgb
Ridge Regression	ridge
Support Vector Classifier (SVC)	svc

10. Place the renamed output folder in the directory specified in Step 3c.
11. Repeat Steps 5 through 9 for all combinations of the eight simulation methods and three states. This should result in a total of 24 output folders named as *ca_logit_glm, ca_logit_reg, . . .* etc.
12. Run *validate_model.py*, and figures will be saved in the directory specified in Step 3b.
    a. (Optional) To produce plots similar to **Exhibit 6** for other states and leave reasons, update Lines 78 and 79 in *validate_model.py* using the other values for variables *st* and *t* listed in the comment.

---

[i] Details on the background of developing the Worker PLUS model and the model architecture are provided in IMPAQ (2021). Worker Paid Leave Usage Simulation (PLUS) Model User Manual.

[ii] There are also multi-label classification tasks in the model, such as classifying a worker's existing wage replacement ratio (offered by employer-paid benefit) into up to six ratio categories. Each of the simulation methods available from the model is flexible enough to handle binary and multiple-label classifications based upon the structure of the data element. The model performance, however, would mostly be dependent upon the performance of many binary classification tasks. We therefore focus on binary classification in this brief.

[iii] For details on advantages of various machine learning methods, see Varghese, D. (2018). Comparative Study on Classic Machine learning Algorithms: Quick summary on various ML algorithms. Retrieved from https://towardsdatascience.com/comparative-study-on-classic-machine-learning-algorithms-24f9ff6ab222; and Ketkar, N. (2017). Deep Learning with Python: A Hands-on Introduction. Chapter 2 Machine Learning Fundamentals. APRESS.

[iv] Model tuning steps include searching for the optimal option for missing value handling, feature variable standardization, and hyperparameter tuning. See Scikit-Learn (2020). Model Selection and Evaluation (https://scikit-learn.org/stable/model_selection.html#model-selection) for details.

[v] To be consistent with the FMLA survey data, the total number of leaves is defined as the total number of leave *reasons* for which leaves are taken over a year. For example, a worker who took two episodes of leaves due to his or her own illness and one episode of leave due to providing care to an ill child would be considered to have taken two leaves in both the FMLA survey data and our model.

[vi] The six leave types considered by the model are based on leave reasons, including one's own illness, maternity disability, bonding with a new child, caring for an ill child, caring for an ill spouse or domestic partner, and caring for a parent. The variable that reports total number of leaves in the FMLA survey is *a4_cat*, which has missing data that account for 0.13% of the total worker population, and is top-coded at six leaves.

[vii] The proportion of leave taking and leave needs for one's own illness and maternity disability is estimated using 2018 FMLA survey data. The proportion of program outlay for these leave reasons is estimated using state program data in California, New Jersey, and Rhode Island. Similar analyses can be performed for New Jersey and Rhode Island, and will generate similar results. Appendix B provides instructions on producing the corresponding exhibits for these states.

[viii] For details on leave needs among disadvantaged workers, see Gupta, P., Goldman, T., Hernandez, E., & Rose, M. (2018). Paid Family and Medical Leave is Critical for Low-wage Workers and Their Families. Center for Law and Social Policy.

[ix] Administrative program statistics including caseloads and benefit outlays are obtained from the following sources: Employment Development Department, State of California (2020). Disability Insurance Program Statistics. Retrieved from https://www.edd.ca.gov/about_edd/pdf/qsdi_DI_Program_Statistics.pdf; Employment Development Department, State of California (2020). Paid Family Leave Program Statistics. Retrieved from https://www.edd.ca.gov/about_edd/pdf/qspfl_PFL_Program_Statistics.pdf; New Jersey Department of Labor and Workforce Development (2017). Temporary Disability Insurance Workload in 2016 Summary Report. Retrieved from https://www.nj.gov/labor/forms_pdfs/tdi/TDI%20Report%20for%202016.pdf; New Jersey Department of Labor and Workforce Development (2017). Family Leave Insurance Workload in 2016 Summary Report. Retrieved from https://www.nj.gov/labor/forms_pdfs/tdi/FLI%20Summary%20Report%20for%202016.pdf; Rhode Island Department of Labor and Training (2014, 2015, 2016). TDI Annual Update.

[x] The Worker PLUS Model User Manual is provided along with model code and data files during model downloading. See IMPAQ (2021). Worker Paid Leave Usage Simulation Model User Manual.