CLASSIFICATION OF LARGE-SCALE LYME DISEASE DATA

ABSTRACT. Lyme disease is greatly under-studied; however, patients have banded together to pool their data via the MyLymeData survey through LymeDisease.org to facilitate research. In our project, we apply a variety of classification methods on the MyLymeData dataset to classify a patient's Wellness and generate treatment recommendations for Unwell patients. The suggestions include antibiotics to take, alternative treatments to try, and potential side effects of treatments. With these models, we hope to educate clinicians and patients more on antibiotics, with the goal of improving patient outcomes.

1. INTRODUCTION

In the United States alone, 300,000 new cases of Lyme disease are diagnosed every year [3]. If recognized and treated promptly, there are little to no lasting side effects. However, some patients develop chronic Lyme disease (CLD) despite treatment while others are not diagnosed or misdiagnosed. Despite the significantly lower quality of life, higher activity limitations, and greater medical costs for CLD patients compared to other chronic disease patients, little has been done to understand the disease further [3]. Our goals in this paper are to utilize the MyLymeData to understand structural differences between Well and Unwell patients and identify the best set of antibiotic or alternative treatments for each patient with minimal side effects. We identify these antibiotics and alternative treatments using a recommender framework, which we also use to predict side effects. In addition, we investigate which questions from the MyLymeData survey are most important for identifying Well and Unwell patients to both better treat patients and simplify the survey.

1.1. **MyLyme Dataset.** The dataset consists of survey answers from 440 Well and 3686 Unwell patients describing those who have and have not recovered from Lyme disease. These are distinguished by answers to a self-identification question in the survey. We assume self-identification is accurate. These Well and Unwell sets are further divided into Baseline and Phase 1 questions. Baseline questions are identical for both Well and Unwell, while Phase 1 is tailored to either Well or Unwell. In each of these four subgroups, the data is finally divided by data type: binary, categorical, scalar, and string. We study only binary, categorical, and scalar data. Rows correspond to patients, while columns correspond to questions.

Previously, the raw survey data was encoded using one-hot vectors for categorical and binary type questions. As a result, each column of scalar data corresponds to a single question, but each column of categorical and binary data corresponds to an answer choice within a question. In order to contrast Well and Unwell patients, we need a combined matrix that includes both sets of patients. Since this requires each set to have the same questions, we only extract the baseline data, filter out questions that are only answered by one group and re-index questions for the Unwell group so that the sequence of questions for the two groups is the same. We refer to this set as the matched Well and Unwell baseline data, which is further subdivided into binary, categorical, and scalar data. Some experiments combine the binary, categorical, and scalar matrices and refer to this as the combined data.

1.2. **Contributions.** Our contribution is a recommender framework. It is trained on data from a group of patients with known responses and predicts for patients who either have unknown or suboptimal responses. We implement this framework using neural networks, but we can apply the same idea using other classifiers such as SVM and simple binary classification. This framework is also adaptable for other

areas. We can implement this framework in a variety of contexts, such as recommending antibiotics for Unwell patients based on Well patients' antibiotic responses, predicting the level of side effects patients will experience, suggesting exercise protocols, etc. In addition, we identify questions within the MyLymeData that are related to the classification of Well and Unwell groups and the effectiveness of antibiotic treatments.

1.3. **Organization of Paper.** We begin by introducing the framework, techniques, and classification methods used. Next, we study the Lyme disease dataset and provide insights into four problems. In Section 3 we use various classic machine learning methods to classify Well and Unwell patients. Then in Section 4 we use the recommender framework for antibiotics. In Section 5 we use the framework to predict the most effective alternative treatment for a new patient, and in Section 6 we predict the level of side effects a new patient may experience due to an alternative treatment. Lastly are our conclusions and references, with additional experiments and notes on the dataset in the appendices.

2. Techniques and Classification Methods

The MyLymeData we use is highly unbalanced. As a result, we use a variety of over- and under-sampling methods to reduce bias in classification. We use a variety of classic classification methods as Well as a newly proposed method, simple binary classification (SBC) [6], to analyze the MyLyme data. We will describe support vector machines (SVM), non-negative matrix factorization (NMF), logistic regression, neural networks, SBC, random forests, and ensemble learning. To evaluate the accuracy of these classifiers, we use 10-fold cross-validation. We also describe our recommender framework in more detail.

2.1. **Recommender Framework.** Many datasets, including the MyLymeData dataset, contain missing data due to underlying properties of the data points. For example, Well patient do not answer certain Unwell patient related questions regarding the level of side effects experienced for given treatment, because it is assumed these patient do not experience certain side effects.

However, predicting with such discrepancy is highly desirable. For example, doctors would like to know how Well patients would react to these treatments if their treatment regime were changed. For this reason we propose the framework in Figure 1.

Our framework works in the following way:

(1) train a classifier/predictor based on known group features and response variables
(2) evaluate accuracy of classifier based on known group to make sure the predictor is effective
(3) classify/predict on unknown group
(4) If original response variables are known, analyze by comparing the results with the original labels

2.2. **Unbalanced data.** The dataset has a 10:1 ratio of Unwell to Well patients. Because of this imbalance, many of the classifiers can return high accuracy rates by choosing the majority class every time. To compensate, we use four different options to form a balanced training set.

- Option 1: Under-sample by truncating the 3686 Unwell patients to the first 440 patients, which creates two equal class sizes of 440 rows each.
- Option 2: Under-sample by selecting a random 440 Unwell patients, again for two equal classes of 440 rows each.
- Option 3: Over-sample by duplicating the Well dataset eight times for a total of 3520 rows while keeping the Unwell dataset with 3686 rows.
- Option 4: Over-sample by selecting two rows of the Well baseline dataset, averaging them element-wise, and appending this new row to the Well dataset and repeating until we have two classes each with 3686 rows.

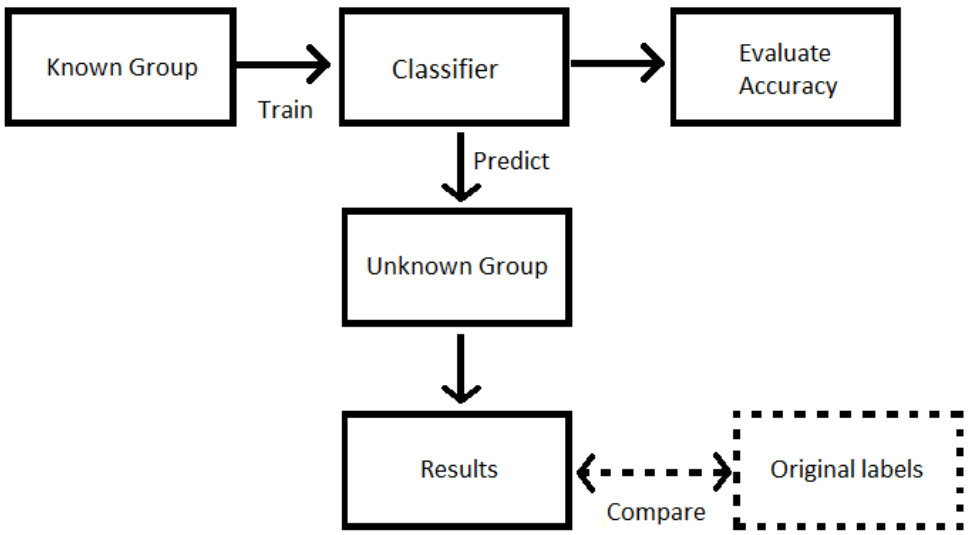


Figure 1 Recommender Framework: Train classifier on known group and evaluate the accuracy. Then predict on the unknown group. If possible, compare to base truth to evaluate accuracy of predictions.

2.3. **Support Vector Machines.** Support Vector Machines seek the boundary that linearly separates two classes of data with the largest margin, either with (soft-SVM) or without exceptions (hard-SVM) [7]. SVM takes in a set of vectors and their signs, indicating class labels, and then finds the hyperplane that optimally separates the data points with different labels. In the case of hard-SVM, this hyperplane is found by maximizing the distance from the hyperplane to the point nearest the hyperplane while keeping a positive distance between the hyperplane and all datapoints. For soft-SVM, the optimization problem is

$$\min_{\mathbf{w},b,\boldsymbol{\xi}} \left(\lambda ||\mathbf{w}||^2 + \frac{1}{m} \sum_{i=1}^{m} \xi_i \right) \text{ s.t. } \forall i,\ y_i(\langle \mathbf{w}, \mathbf{x}_i \rangle + b) \geq 1 - \xi_i \text{ and } \xi_i \geq 0.$$

This introduces the slack variable ξ, which allows for misclassified data points when placing the hyperplane. This is helpful for non-linearly seperable cases. We can control the trade-off between large margin and correct placement of $\mathbf{x}_i$ on either side of the hyperplane using λ. With small λ, indicating few exceptions, soft-SVM behaves like hard-margin SVM. We apply hard-margin SVM to the matched baseline data for Unwell and Well patients in order to predict their Unwell and Well labels. For multi-class SVM, we use an error-correcting-output-code (ECOC) classifier using binary SVM learners [1]. The Matlab function for ECOC classifier takes the same vector inputs as SVM, but can predict amongst many classes instead of only a positive and a negative class. To do this, it sets the first class as positive and the next as negative, classifies using SVM, then repeats with the second class as positive and the next as negative, and so on. This method is particularly useful when classifying according to scalar data because each entry has a range of possible values rather than a binary entry. With ECOC, we can classify according to scalar data such as a patient's reported level of side effects and maintain the range of responses as multiple classes.

2.4. **Non-negative Matrix Factorization.** Non-negative matrix factorization is a tool commonly used in topic-modeling and data reduction. The method approximately factors a non-negative data matrix X into two non-negative matrices A and S by solving the optimization problem

$$\min_{A,S} \|X - AS\|_F \text{ s.t. } A_{ij} \geq 0, S_{if} \geq 0$$

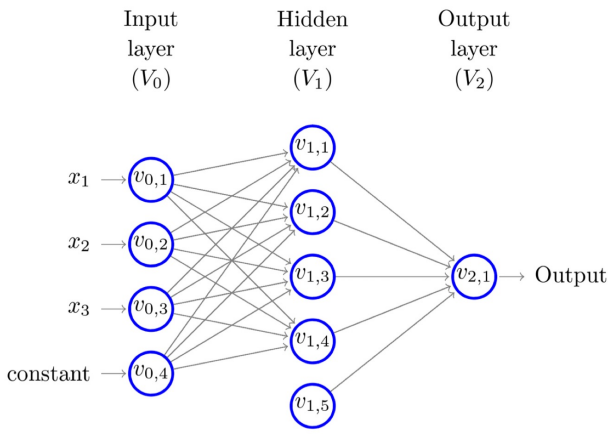


Figure 2 Feed Forward Neural Net [7]

[5]. The basis matrix is $A \in \mathbb{R}_+^{m \times r}$, and the encoding matrix, or feature matrix, is $S \in \mathbb{R}_+^{r \times n}$. In the context of MyLyme data, the rows of X represent patients and columns correspond to questions from the surveys. We apply NMF as a data compression method by using A as input in place of X for a variety of classification methods. In this way, we input fewer features and evaluate whether accuracy is preserved. It also provides insight into the rank of the baseline matrices. It should be noted that the NMF objective function is non-convex and therefore, many algorithms can get trapped in local minima. To mitigate this problem, we run n trials of NMF initialized with rng(n) and average the results. The NMF function in Matlab uses either alternating-least squares or multiplicative update algorithm implementations to approximate the solution to the minimization problem.

2.5. **Logistic Regression.** Logistic regression is a classification method with two types. Binary logistic regression classifies according to binary output variables, and multinomial logistic regression classifies according to output variables with more than two classes [2]. Given a training set $S = (x_1, y_1), ..., (x_m, y_m)$, where x_i is a data point and y_i is its corresponding true label, we substitute them into the logistic loss function (2.1), to compute the value of the weights. Note the logistic loss function is a convex function with respect to x, so we compute the weights, w, according to the convex program (2.1).

$$\underset{w \in \mathbb{R}^d}{\operatorname{argmin}} \frac{1}{m} \sum_{i=1}^{m} \log(1 + e^{-y_i \langle w, x_i \rangle}) \tag{2.1}$$

To predict the class of an unlabeled data point x, we simply compute $\frac{1}{1+e^{-\langle w,x \rangle}}$ and label based on the separating threshold, 0.5.

2.6. **Neural Networks.** A feedforward neural network can be described as a directed acyclic graph in which each node of the graph corresponds to a neuron and edges are the connections between them. Because there are no cycles, the vertex set V can be partitioned into several layers, where vertices in the same layer will not influence each other. Each neuron is modeled by a real scalar function called the activation function. Common activation functions are the sign function, the sigmoid function, and the threshold function. The input for each neuron is the weighted sum of the outputs of the neurons to which it is connected. In this way, each neuron outputs a nonlinear combination of inputs from the previous layer. Figure 2 shows a simple neural network from [7].

2.7. **Simple Binary Classification.** Simple binary classification (SBC) from is an new efficient method to classify binary data [6]. The algorithm has been tested on facial recognition and handwritten digit recognition in the YaleB and MNIST datasets [6]. When given non-binary input, the data set must first be transformed by a random matrix into binary data. If a matrix X is the original data, we preprocesses X by $Q = sign(AX)$. The algorithm uses the sign pattern of Q to classify. The membership weight coefficients $r(l,i,t,g) = \frac{P_{g|t}}{\Sigma_{j=1}^{G} P_{j|t}} \frac{\Sigma_{j=1}^{G} |P_{g|t}=P_{j|t}}{\Sigma_{j=1}^{G} P_{j|t}}$ describe the probability a data point should be classified in a certain group by sign pattern of Q. The algorithm has two hyperparameters: the number of layers in the algorithm and the number of measurements, or hyperplanes which separate the data into cells in each layer. Increasing the number of measurements creates smaller, more precise cells to classify the data. Increasing the number of layers increases the complexity of the model. Increasing either of these allows the algorithm to classify the data more accurately, although the algorithm is random so there is inherent variability even when the number of layers and measurements are high.

2.8. **Random Forest and Ensemble Learning.** Ensembled learning is a method that trains a bag of weak classifiers that are trained on part of the dataset or features and predicts based on decisions of all classifiers [4]. Random Forest classification is one of the most popular ensemble classifiers. A set of ensemble classifiers consists of a collection of independently trained weaker classifiers, such as decision trees or neural networks; the overall prediction is dependent on the combined decisions of all classifiers.

The algorithm is as follows. For each classifier, draw n samples without replacement, then grow an unpruned classification tree: at each node of decision tree, choose random samples for predictors and choose the best split from these variables. To predict on new data, we simply apply the decision trees and a decision strategy (e.g. majority votes).

2.9. **Cross-Validation.** Cross-validation is a statistical method used to estimate the applicability of machine learning models. We use 10-fold cross-validation to compare and select a model for a given predictive modeling problem because it is easy to understand, easy to implement, and the results in general have a lower bias than other methods. An advantage of cross-validation (CV) is that there is no need to hold one part of the known data to be validation set, so we have more samples for training dataset. This technique shuffles the dataset randomly, then splits the dataset into k groups, leaving one group as the test data (for validation) and the remaining k-1 groups as training data. We fit our model on the training set and evaluate it on the testing set iteratively until each of the k groups has been treated as the testing set.

3. Classifying Well and Unwell

The first classification problem we examined was predicting patients' Well or Unwell status. We use the matched baseline questions from the Well and Unwell datasets, remove the labels of Well and Unwell, and then apply a variety of classification methods to the dataset. We also investigate which columns in the baseline matrices impact classification of Well and Unwell most. Accuracy is defined the classification rate from 10-fold cross-validation.

3.1. **Support Vector Machines.** We used SVM on the set of matched Well and Unwell baseline question in order to classify patients into Well and Unwell groups. Accuracy rates are listed in Table 1 and are generally 89-99%. We also split this combined binary, scalar, and categorical dataset into its three components and ran SVM on those datasets. We chose to use a linear kernel rather than Gaussian kernel for classification because it improved accuracy. Initially, SVM returned nearly perfect results (99.08% accurate) because of the difference in class size between Well (440 patients) and Unwell (3686 patients). There was not a significant difference between random selection (Option 2) and truncation (Option 1)

Classifier, kernel, input	Binary	Categorical	Scalar	Combined
(a) *fitcsvm*, linear kernel, all	**0.9435**	**0.8921**	**0.8929**	**0.9380**
(b) *fitcsvm*, Gaussian (rbf) kernel, all	0.9321	0.8885	0.8875	0.9200
(c) *fitclinear*, linear kernel, all	**0.9438**	**0.8934**	**0.8934**	0.9115
(d) *fitcsvm*, linear kernel, under-sample	0.8955	0.6670	0.5557	0.8830
(e) *fitcsvm*, linear kernel, under-sample	0.8795	0.6523	0.5920	0.8750
(f) *fitcsvm*, linear kernel, over-sample	0.8913	0.6593	0.5859	0.8884
(g) *fitcsvm*, linear kernel, over-sample	0.9369	0.7183	0.6286	**0.9381**
(h) *fitclinear*, linear kernel, over-sample	0.9348	0.7139	0.6244	0.8988

Table 1 Classification Accuracy Input data for (a), (b), and (c) are the matched set of Well and Unwell baseline questions. The next rows create equal class sizes for Well and Unwell. Input data for (d) are under-sampled by Option 1 (all options detailed in Section 2.2). Input data for (e) are under-sampled by Option 2. Input data for (f) are over-sampled by Option 3. Input data for (g) and (h) are over-sampled by Option 4. The *fitcsvm* is the standard SVM in Matlab while *fitclinear* is the SVM designed for high-dimensional data. The two highest accuracies per data type are bolded.

for under-sampling (Section 2.2). However, over-sampling by appending averaged columns (Option 4) performed better than over-sampling by duplicating data (Option 3). In addition to the standard Matlab SVM package *fitcsvm*, we tried a classifier intended for larger dimensions (*fitclinear*), but it did not perform as Well as *fitcsvm* when classifying binary, scalar, and categorical data together. Table 2 displays the confusion matrices for the SVM models described in Table 1. These matrices are for the combined datasets of binary, categorical, and scalar together only. These figures allow us to see the type of classification errors made. A false negative, when a patient is predicted to be Well when actually Unwell, is more significant than a false positive, when a Well patient who is predicted to be Unwell. The former patient would never be healed while the latter recovered patient would continue to receive treatment longer than necessary. A reevaluation at a later time may correct the misclassification for this Well patient, while the misclassified Unwell patient will not be reevaluated after being predicted to be Well. In every case but Table 2f, the false positives outweigh the false negatives as hoped. We also used the outputs of NMF on matched baseline binary, scalar, and categorical matrices as inputs for SVM (Figure 3). For an original matrix $X \approx AS$, the input for SVM is the matrix A that has the same number of rows (patients) as the original matrix X. We use the original non-factored matrix results as a baseline for comparison. Maximum accuracies with alternating least squares updated NMF are within 0.005 of the original accuracies, but fewer columns are needed. Maximum accuracy for binary data is achieved at 26 columns in the factorized matrix of 65 total, categorical at 56 of 61 columns, scalar at 26 of 27 columns, and combined at 33 of 153 columns. However, each data type has a sharp increase in accuracy when the number of columns is greater than 10, regardless of the size of the original matrix (see Figure 3). This is significant for circumstances when data compression is necessary before running a classifier, because the original accuracy is Well preserved with a smaller NMF factor matrix. We ran the experiment again with the multiplicative updates for the NMF algorithm, still using matched baseline data in binary, categorical, scalar, and combined forms, and then using the A factor matrix in SVM (Figure 4). With multiplicative updates, the increase in accuracy beginning at k=10 columns holds only for binary and categorical data. The maximum accuracy is at the full column size (Table 3.) If we exclude the full column size, the maximum accuracy is still found near the full column size. For the full number of topics and alternating least squares updated NMF, the scalar and categorical accuracies are within 0.005 of

	Predicted Unwell	Predicted Well	
Actual Unwell	3599 87.23%	87 02.11%	97.64% 02.36%
Actual Well	159 03.85%	281 06.81%	63.86% 36.14%
	95.77% 04.23%	76.36% 23.64%	89.40% 10.60%

(a) *fitcsvm*, linear kernel, input data are the matched set of Well and Unwell baseline questions

	Predicted Unwell	Predicted Well	
Actual Unwell	3683 89.26%	3 0.07%	99.92% 0.08%
Actual Well	174 04.23%	266 06.45%	60.45% 39.55%
	95.49% 04.51%	98.88% 01.12%	89.34% 10.66%

(b) *fitcsvm*, Gaussian (rbf) kernel, input data are the matched set of Well and Unwell baseline questions

	Predicted Unwell	Predicted Well	
Actual Unwell	3495 84.71%	191 04.63%	94.82% 05.18%
Actual Well	174 04.22%	266 06.45%	60.45% 39.55%
	95.26% 04.74%	58.21% 41.79%	89.34% 10.66%

(c) *fitclinear*, linear kernel, input data are the matched set of Well and Unwell baseline questions

	Predicted Unwell	Predicted Well	
Actual Unwell	413 46.93%	27 03.07%	93.86% 06.14%
Actual Well	63 07.16%	377 42.84%	85.68% 14.32%
	86.76% 13.24%	93.32% 06.68%	50.00% 50.00%

(d) *fitcsvm*, linear kernel, input data are matched baseline data under-sampled by Option 1

	Predicted Unwell	Predicted Well	
Actual Unwell	411 46.71%	29 03.30%	93.41% 06.59%
Actual Well	57 06.48%	383 43.52%	87.05% 12.95%
	87.82% 12.18%	92.96% 07.04%	50.00% 50.00%

(e) *fitcsvm*, linear kernel, input data are matched baseline data under-sampled by Option 2

	Predicted Unwell	Predicted Well	
Actual Unwell	3414 47.38%	272 03.78%	92.62% 07.38%
Actual Well	520 07.22%	3000 41.63%	85.23% 14.77%
	86.78% 13.22%	91.69% 08.31%	51.15% 48.85%

(f) *fitcsvm*, linear kernel, input data are matched baseline data over-sampled by Option 3

	Predicted Unwell	Predicted Well	
Actual Unwell	3441 46.68%	245 03.32%	93.35% 06.65%
Actual Well	187 02.54%	3499 47.46%	94.93% 05.07%
	94.85% 05.15%	93.46% 06.54%	50.00% 50.00%

(g) *fitcsvm*, linear kernel, input data are matched baseline data over-sampled by Option 4

	Predicted Unwell	Predicted Well	
Actual Unwell	3320 45.04%	366 04.97%	90.07% 09.93%
Actual Well	481 06.53%	3205 43.48%	86.95% 13.05%
	87.35% 12.65%	89.75% 10.25%	50.00% 50.00%

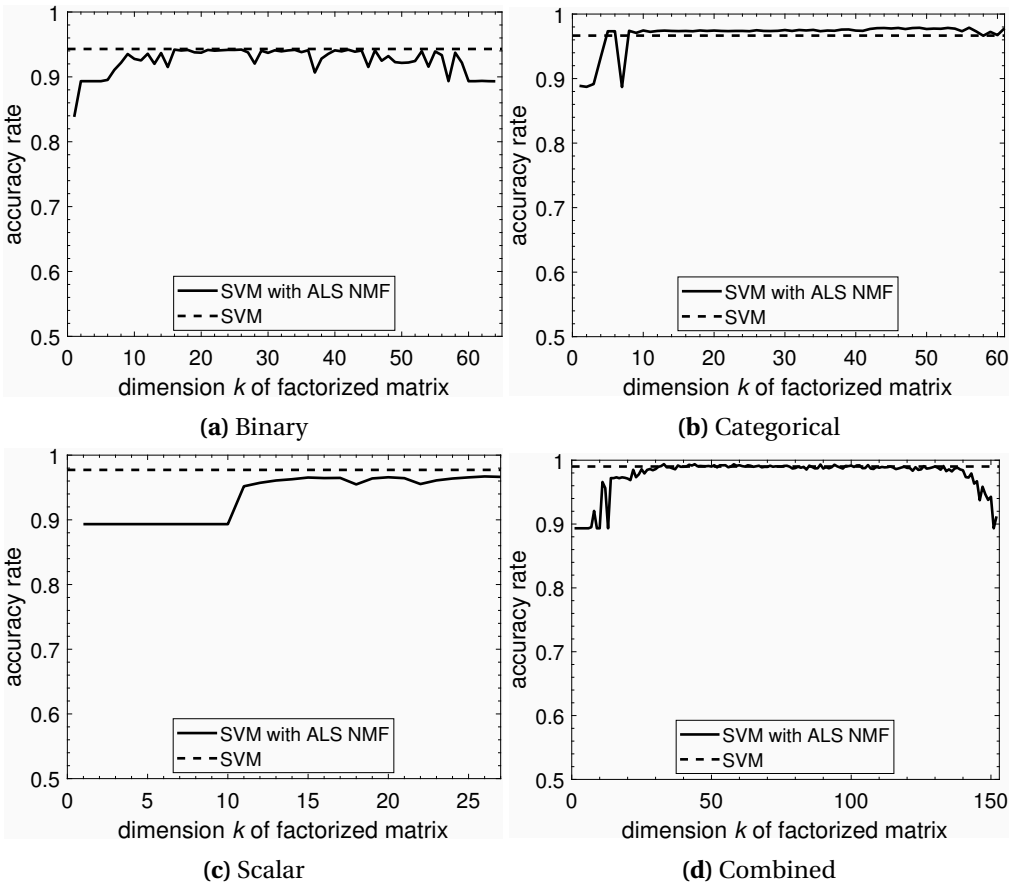
(h) *fitclinear*, linear kernel, input data are matched baseline data over-sampled by Option 4

Table 2 Confusion Matrices These matrices correspond to the final column of Table 1, the combination of binary, scalar, and categorical data together with a variety of classifiers, kernels, and input data sets.

	Binary	Categorical	Scalar	Combined
Original Accuracy	0.9435	0.8921	0.8929	0.9380
Number of Columns	65	61	27	153
Accuracy	0.9440	0.9440	0.9664	0.9909
Number of Columns	59	57	25	147
Accuracy	0.9406	0.9745	0.9007	0.9729

Table 3 SVM with multiplicative updated NMF: Maximum accuracy with NMF factor matrix as input for SVM. Input data are the matched baseline questions. The first row lists the accuracies from SVM without NMF as a comparison (Table 1 row 1). The middle set of rows indicates the maximum accuracy with the full set of columns. The last set of rows indicates the next highest accuracy with fewer columns in the factor matrix.

each other, but binary and combined do not yield number results. Using one less than the full number of columns for binary and combined yields a difference 0.05 and 0.08 between the original accuracy and the accuracy with NMF. Upon further investigation, the norm difference $\|X - AS\|$ as the number of columns in the factor matrix A approaches the total number of columns for that data type is very small (less than 10^{-8}) but non-zero. The factors themselves are not the original multiplied by the identity matrix as expected and do not reconstruct X perfectly. This is why the maximum number of columns as input to SVM does not yield the same result as SVM alone. When creating the NMF factor matrices, we received warnings that the factors converged to less than full rank. For example, the combined matrix



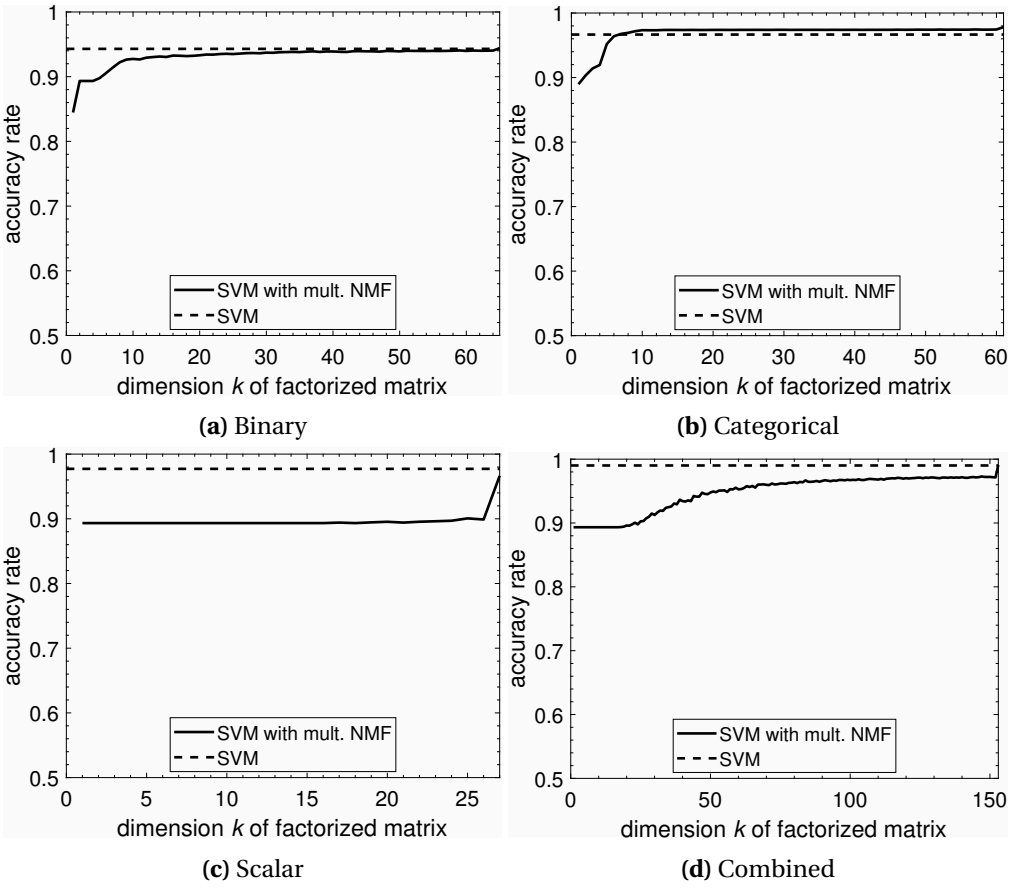
(a) Binary **(b)** Categorical

(c) Scalar **(d)** Combined

Figure 3 SVM with ALS NMF input: The input dataset is the combined Well and Unwell baseline questions. Each point plotted corresponds to the average accuracy of SVM trained on the corresponding NMF factor matrix with k columns and alternating least-squares updates. Each trial n from 1 to 50 of NMF was initialized with rng(n).

converged to rank 151 when the number of columns requested was the full 153. In this case, the rank of the original combined matrix is 151, which means the NMF factor matrix cannot have rank larger than 151. To reduce the number of warnings, we switched from the default NMF algorithm of alternating least squares to multiplicative updates. In a brief test, this yielded A and S matrices with more zero and integer entries in the factor matrices with multiplicative updates instead of many very small non-zero entries with alternating least squares.

3.2. **Simple Binary Classification.** In this section, we use simple binary classification (SBC) to classify Well patients and Unwell patients. We use option 2.2 to undersample the matched baseline data because the algorithm performs best on balanced data. The average accuracy is denoted by Average Correct Classification Rate Total (ACCRT). For every numerical experiment, when we increase the number of measurements, the number of layers, and hyperparameters in the SBC algorithm, the accuracy will increase until coming to a limit. The trial number is for a given number of measurement and number of layer, the times we run the algorithm to approximate accurate average accuracy as there is inherent variability even when we use high hyperparameter. In our experiments, if the curve is quite smooth, then we don't need to increase trial number.



(a) Binary **(b)** Categorical

(c) Scalar **(d)** Combined

Figure 4 SVM with multiplicative updated NMF: The input dataset is the combined Well and Unwell baseline questions. Each point plotted corresponds to the average accuracy of SVM trained on the corresponding NMF factor matrix with k columns and multiplicative updates. Each trial n from 1 to 50 of NMF was initialized with rng(n).

Figure 5 demonstrates the accuracy limit. It can be seen that when we increase number of layers from 5 to 20 and not further increase trial number, there is no obvious improvement. In the following three subsections, we will test the performance of SBC on different sections of the patient data. Our goal is to correctly classify new patients as Well or Unwell. The accuracy of our algorithm is the number of correctly classified patients out of the total number of patients. We will then discuss several ideas about improving classification accuracy.

3.2.1. *Using different types of data as input.* We first input the matched baseline binary data, undersampled by option 2 from Section 2.2, for a total of 880 patients. As shown in Figure 6 the average correct classification rates increase as we increase the number of measurements from 20 to 60 or increase the number of layers from 5 to 15. The trend is obvious from the three graphs mentioned above. See Table 4 for accuracies for the maximum measurement number(60) of Figure 6.

After using each part of our baseline data to classify Well patients and Unwell patients, we combine the data together and test whether the average correct classification rate improves. Figure 6d shows the average correct classification rate when we use the combined data as input. As mentioned previously, when we use a large number of layer and number of measurements, accuracy comes to a limit. However, we may be able to increase accuracy by modifying the number of trials or other hyperparameters. Table

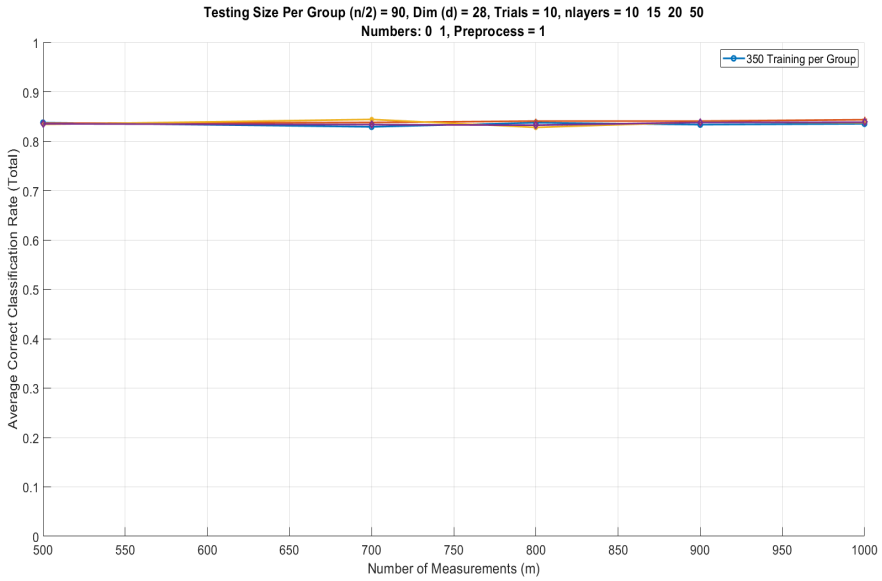


Figure 5 SBC on matched scalar questions. The input is under-sampled baseline scalar data: 440 Well patients data and 440 Unwell patients data with option 1 from 2.2. m= 500, 700, 800, 900, 1000; L= 5, 10, 15, 20. When layer number and measurement number are increasing, the accuracy rate comes to a limit.

Layers	Categorical	Binary	Scalar	Combined
5	0.6322	0.7172	0.6291	0.6649
10	0.6327	0.7456	0.6456	0.6842
15	0.6261	0.7481	0.6532	0.6922

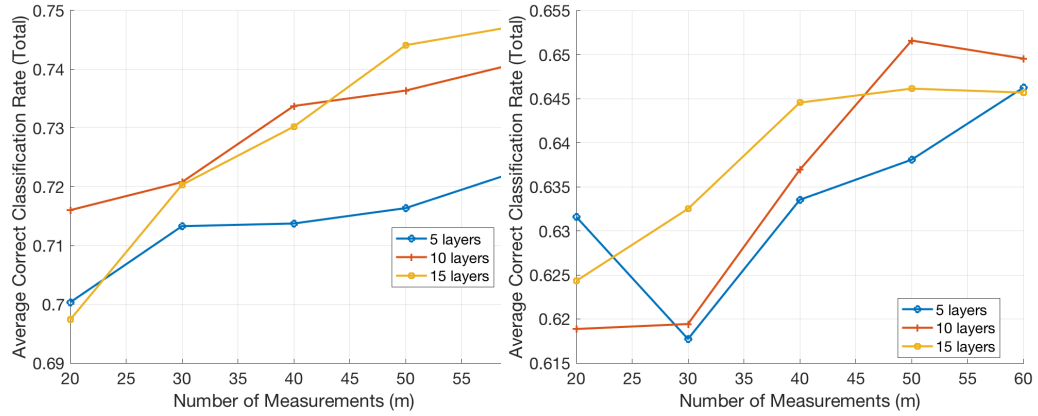
Table 4 Average Correct Classification Rate SBC with the number of measurements fixed as 60. This table shows the average correct classification rate of different data types and different layer numbers when measurement number is fixed at 60. The input is under-sampled baseline categorical data: 440 Well patients data and 440 Unwell patients data. We use 10-fold cross validation. The number of training data is 396 per group. For each number of measurements and layers, we run the algorithm 10 times.We use the data without sensitive

Type	Category	Binary	Scalar
Average Accuracy Limit	0.6250	0.7384	0.7136

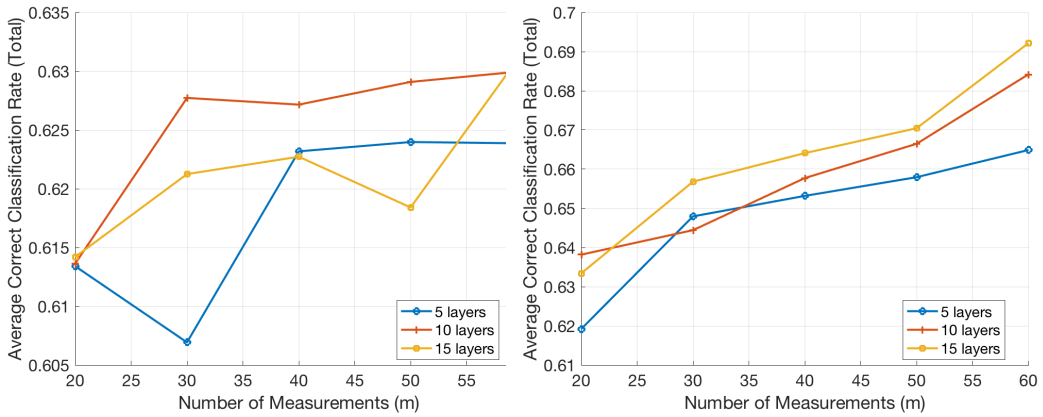
Table 5 Average Correct Classification Rate This table shows average correct classification rate limit of SBC for each data type when the number of limit is fixed as 5. We select accuracy when m is 2000 a limit accuracy. For each number of measurements and layers, we run the algorithm 5 times to calculate average accuracy. We use option 2 to undersample and 396 training data for each group

5 shows the accuracy limits for each data type and training set size when the number of layer of number is fixed as 5.

3.2.2. *SBC with NMF.* The data for each patient consists of 153 matched baseline questions. In this section, we use NMF to decrease the dimension of our data and test the classification performance of



(a) Binary **(b)** Scalar



(c) Categorical **(d)** Combined

Figure 6 SBC on under-sampled baseline data by option 2: 440 Well patients data and 440 Unwell patients data. We use 10-fold cross validation and number of training data is 396 per group. m= 20, 30, 40, 50, 60; L= 5, 10, 15. For each number of measurements and layers, we run the algorithm 20 times to calculate average accuracy. The data we use there excluded sensitive questions.

kind	Category	Binary	Scalar
Without NMF	0.8931	0.6596	0.6733
With NMF	0.9244	0.7952	0.7970

Table 6 Average Correct Classification Rate This table shows limit of the average correct classification rate of SBC for each data type with NMF or without NMF. L= 5, n= 40. We use 350 data for training each group and 90 data for testing each group. For each number of measurements and layers, We average result over 20 trials.The data we use there included some sensitive questions which leads to relative higher accuracy.

SBC. Figure 7 shows performance of SBC algorithm without NMF and average correct classification rate increase from around 59 percent to 67 percent when the number of layer is fixed as 5. Figure 8 shows the experiments result of using scalar data preprocessed by NMF to classify Well patients and Unwell patients. Original dimension of scalar data is 27 and even if we use NMF to decrease the dimension of our data from 27 to 9, 10, 11, 12, we get better results for every measurement and layer number.

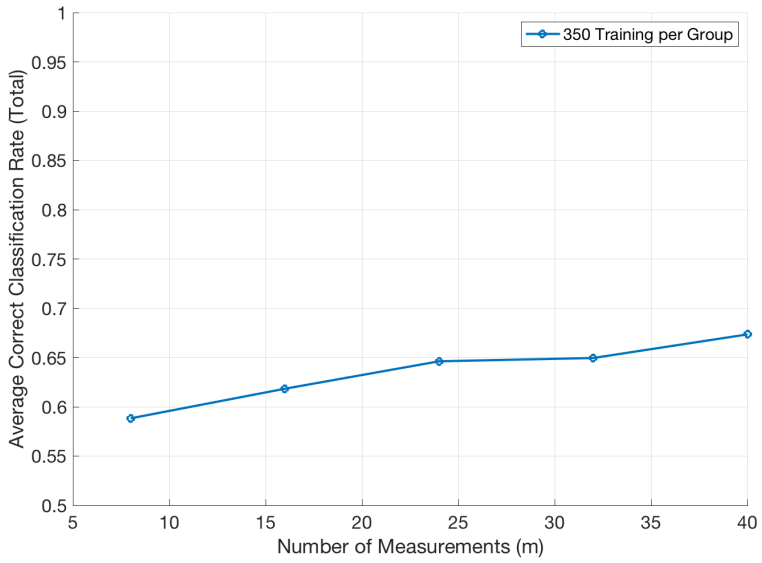


Figure 7 SBC on matched baseline scalar questions. The input is under-sample baseline scalar data by option 2: 440 Well patients data and 440 Unwell patients data. We use 350 data for training each group and 90 data for testing each group. m= 8, 16, 24, 32, 40. L= 5. For every measurement number and layer number, we run the algorithm 300 time to approximate precise accuracy.

The NMF works not only for scalar data but each data type. Table 6 shows the improvement of average accuracy after preprocessing our data with NMF for SBC algorithm when the number of measurement and the number of layer is fixed as 40 and 5.Although the result is improved after NMF in some situation, the good performance is not always existing.For example, we can see from Figure 9, the accuracy improves when NMF topic number ranges from 14 to 42. But for categorical data, no matter what NMF topic number we use, the accuracy decreases .9 and Figure 10 show the average accuracy when using SBC to classify Well patient and Unwell patients with different NMF algorithm.

3.2.3. *Directly using Binary data.* When dealing with data, we first transform the data into binary data by applying random matrix and a sign function.To be more clear, that step can be formulated as $Q = sign(AX)$,and then we use sign pattern of Q to do classification. For binary data, this transformation is unnecessary. Directly using binary data means that we get rid of matrix A and use $Q = sign(X)$to do classification in our algorithm. However, without the random matrix, we can not further increase the number of layers as the number of measurements will be fixed by the dimension of data points. For example, the dimension of binary data of patients are 65 which is exactly the number of binary type question in survey. When we directly use the algorithm to classify Well and Unwell patients, with 65 measurement(dimension of our data point) and 5 layer, the accuracy is around 67%. Table 7 shows that when we directly use the binary data, the accuracy increases when the number of measurements is fixed as 65 whereas the number of layer increase from 10 to 40.

3.3. **Implementations of Logistic Regression, Random Forest, and Neural Nets in Python.** We do classification on the matched Well and Unwell baseline data. Well and Unwell patients' data are separated by binary, categorical, and scalar question types. We label 0 for Unwell patients and 1 for Well patients. We then prepare the matched baseline dataset by option 2 and option 3 in Section2.2. After that, logistic

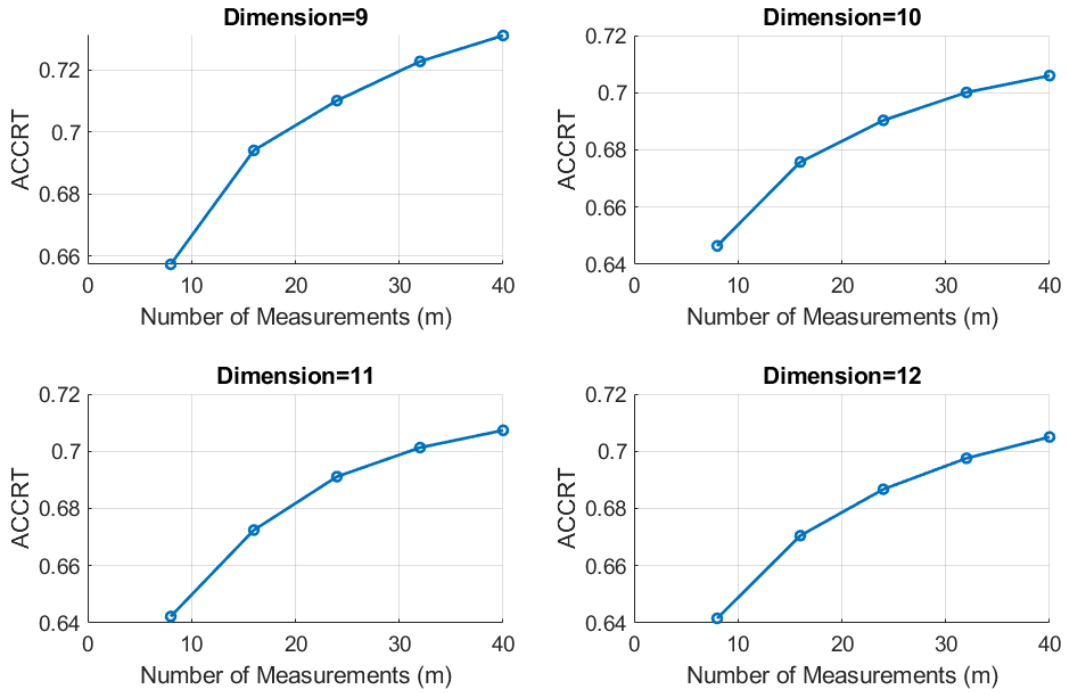


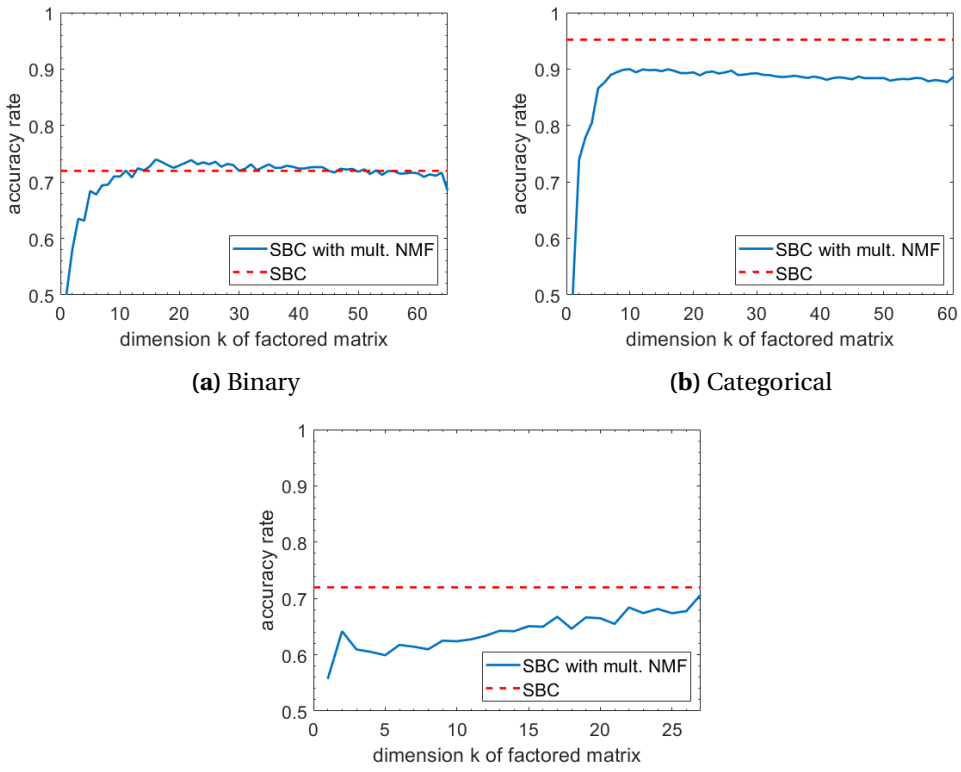
Figure 8 SBC on matched baseline scalar questions with ALS NMF. The input is under-sampled baseline scalar data: 440 Well patients data and 440 Unwell patients data. We use 350 data for training each group and 90 data for testing each group. m= 8, 16, 24, 32, 40. L= 5. For every measurement number and layer number, we run the algorithm 10 times to approximate precise accuracy.

Layers	10	20	30	40
Average Accuracy	0.6818	0.6875	0.6784	0.6909

Table 7 SBC on matched baseline binary questions: This table shows the average correct classification rate when L= 10, 20, 30, 40 for scalar data. We use 396 data from each group for training and 44 data from each group for testing. For given number of measurements and number of layer, we run the algorithm 10 time to approximate precise accuracy and we use 10 fold cross validation to measure accuracy. Notice, we used another version of simple classification when we fix random matrix A as identity matrix whose shape is 65 by 65. The data we use exclude sensitive questions.

	Binary	Categorical	Scalar	Combined
(a) 440 example cases in each of Well and Unwell groups	0.2397	0.1600	0.1749	**0.0242**
(b) 440 examples cases in Well and 3686 in Unwell group	0.1365	0.0653	0.0778	**0.0147**
(c) 2000 random example cases from two groups combined	0.1509	0.0680	0.0901	**0.0242**

Table 8 Mean-squared error of neural nets pattern recognition methods on binary, categorical, scalar and combined data. All 440 Well example cases and 440 randomly selected Unwell example cases are taken. Rng(1) is used as random number generator. An average is taken from 100 trials.



(a) Binary

(b) Categorical

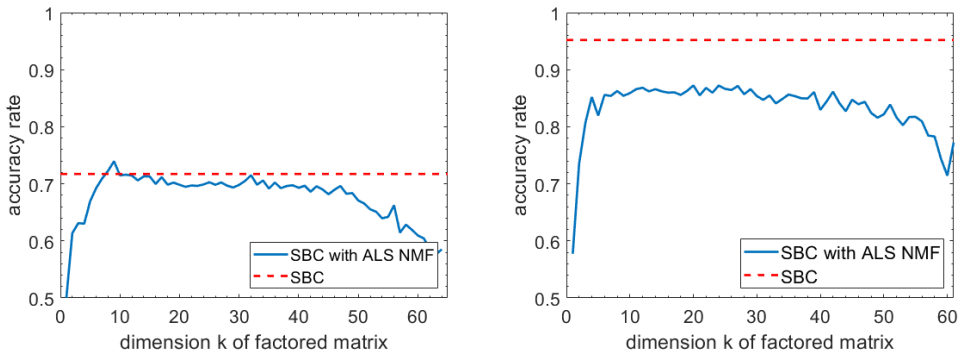
(c) Scalar

Figure 9 SBC with multiplicative updated NMF: The input set is the Well and Unwell baseline questions, undersampled by option 2. The red line is the original accuracy without NMF. We use 396 data for training each group and 44 data for testing each group. We use m=60, L=5, 20 trials per measurement and layer number combination.

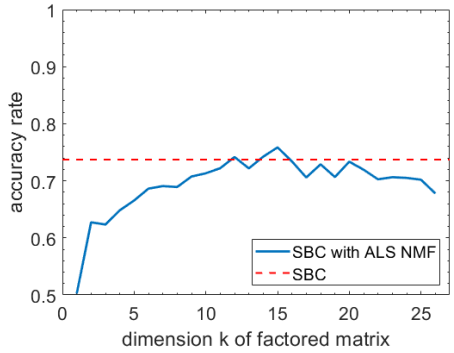
	Option 2			Option 3		
	Binary	Categorical	Scalar	Binary	Categorical	Scalar
Logistic Regression	**0.8625**	**0.9523**	0.9159	0.8987	0.9584	0.9256
Random Forest	0.8364	0.9432	**0.9455**	**0.9854**	**0.9804**	**0.9892**
Neural Nets	0.8318	0.8955	0.9250	0.9482	0.9759	0.9738

Table 9 This table shows the accuracy on classifying Well and Unwell dataset by various combinations of three classification methods and data types. The classification techniques include logistics regression, random forest, and neural nets. For data types, we have binary, categorical, and scalar data. Accuracy on the left is from data preprocessing option 2, while the one on the right is a result of option 3. The highest accuracy for each column is in bold font.

regression, random forest, and neural nets are applied on the preprocessed data to classify Well and Unwell patients and make comparisons. In all cases, option 3 yields higher accuracy on average as shown in Table9. Furthermore, keeping as much data as possible is also a good strategy in general. Therefore, we choose to do option 3 for further actions in this subsection. As shown in Table 9, Logistic Regression, Random Forest, and Neural Nets perform Well on categorical and scalar data with an accuracy level higher than 95%. Binary data, on the other hand, yield a relatively lower accuracy rate. As shown in the



(a) Binary **(b)** Categorical



(c) Scalar

Figure 10 SBC with ALS NMF: The input set is the Well and Unwell baseline questions, undersampled by option 2. The red line is the original accuracy without NMF. We use 396 data for training each group and 44 data for testing each group. We use m=60, L=5, 20 trials per measurement and layer number combination.

Table 9 option 3 section, among the three methods we implement, none of them generates a very high accuracy when working with binary data. The accuracy is about 5%, 0.4%, and 3% less respectively than the most accurate term in each classification method.

After classifying on the whole dataset, we then decompose it by NMF and apply Logistic Regression and Random Forest in Figures 12, 13, and 14. We notice that increasing the number of features does not drastically influence the accuracy level. Instead, it appears to converge to a very high performance after a small number of features. When the number of features reaches approximately ten, the accuracy converges. It means that we can reduce all baseline questions down to 10 groups of topics which still classify accurately. Grouping several questions could give us a more interpretable idea of what questions make a difference in determining patients' Wellness. Therefore, "number of features = 10" could be an ideal parameter for the Lyme Disease classification model on distinguishing Well and Unwell patients.

We also compute the confusion matrices in Table 10. These confusion matrices come from classifying Well and Unwell patients on factorized matrices from NMF. We use two classification methods: logistic regression and random forest and apply them on three data types individually. Since wrongly predicting an Unwell patient to be Well, denoted as false positive, could hinder the curing procedure, it is more concerning than predicting a Well patient as Unwell, denoted as false negative. In these matrices, we can see the number of false positives and false negatives. As shown in the confusion matrices, random forests generate fewer false positives than logistic regression does in all three data types. In this sense, random forests work better in classifying Well and Unwell groups than logistic regression.

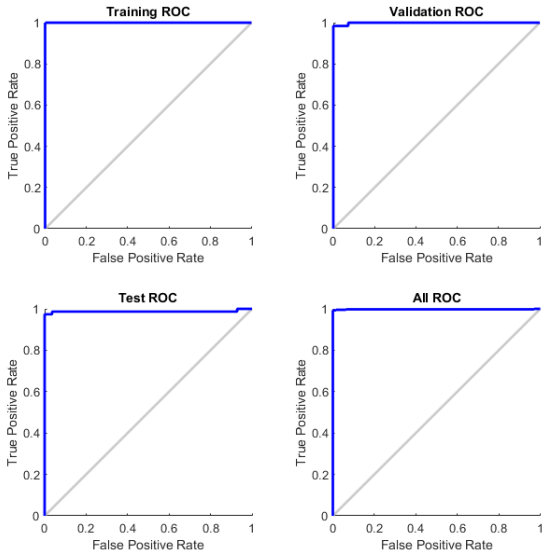
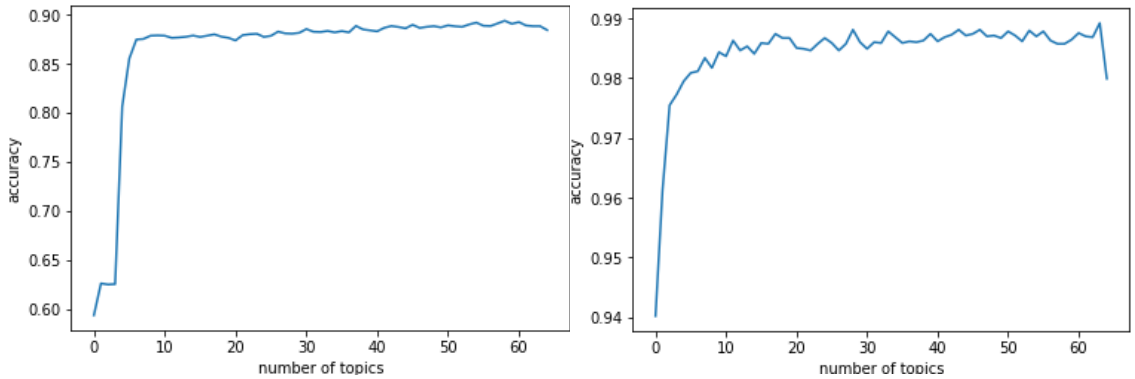


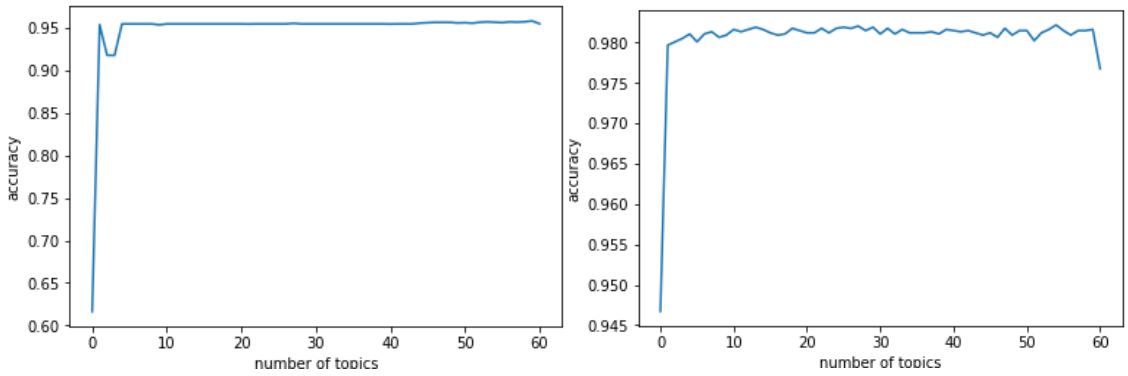
Figure 11 Receiver operating characteristic curve(ROC) curves for training, validation and testing of combined data for all 440 Well example cases and randomly selected 440 Unwell example cases. ROC is used to illustrate how discriminative the binary classifying system is. Rng(1) is used as random number generator.



(a) Logistic Regression on binary data **(b)** Random Forest on binary data

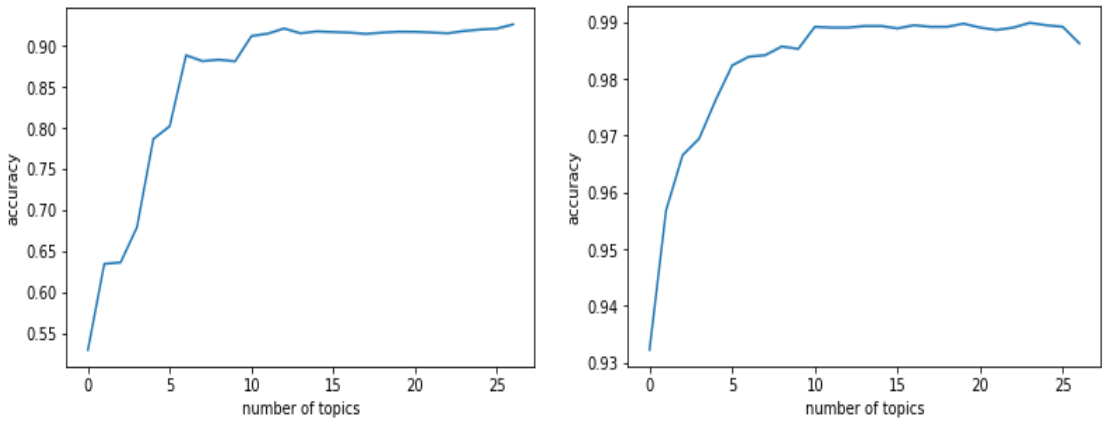
Figure 12 The input data are all the baseline questions for both Well and Unwell patients. After separating the dataset into binary, categorical, and scalar, we decompose the binary datasets by non-negative matrix factorization. We apply Logistic Regression and Random Forest to the factorized matrix. From the learning curve, which represents the accuracy as number of topics increases, we can see that as the number of question features increases, the accuracy also increases. When the number of feature reaches around ten, the accuracy converges.

3.4. **Feature Importance.** In this section, we investigate which aspects of the data impact classification of Well and Unwell patients. For scalar data, each question from the survey is embedded in a single column. However, the categorical and binary data are embedded as one-hot vectors. For these data types, we both investigate on an answer choice level, corresponding to a single column, and on a question level, corresponding to a cluster of columns.



(a) Logistic Regression on categorical data **(b)** Random Forest on categorical data

Figure 13 The input data are all the baseline questions for both Well and Unwell patients. After separating the dataset into three groups according to the question types, including binary, categorical, and scalar, we first decompose the categorical datasets by non-negative matrix factorization. We apply Logistic Regression and Random Forest to the factorized matrix. From the learning curve, which represents the accuracy as number of topics increases, we can see that as the number of question features increases, the accuracy also increases. When the number of feature reaches around ten, the accuracy converges.



(a) Logistic Regression on scalar data **(b)** Random Forest on scalar data

Figure 14 The input data are all the baseline questions for both Well and Unwell patients separated by data type. We first decompose the scalar datasets by non-negative matrix factorization. We apply logistic regression and random forests to the factor matrix. From the learning curve, which represents the accuracy as number of topics increases, we can see that as the number of questions, or features, increases, the accuracy also increases. When the number of features approaches ten, the accuracy converges.

3.4.1. *SVM Feature Importance.* We omitted one column of the matched baseline set at a time and compared the results of SVM with missing data to the results of SVM with the full dataset. Figure 15 depicts the differences between the original accuracy rate and the accuracy rate with a column dropped, each calculated by classification rate from 10-fold cross-validation. A spike in the figure indicates that column is significant for classifying, because without it accuracy rates decrease significantly. Note that some columns extend below the x-axis. These columns increase accuracy when omitted; however, the

	Predicted Unwell	Predicted Well	
Actual Unwell	3249 45.09%	437 6.06%	88.14% 11.86%
Actual Well	521 7.23%	2999 41.62%	85.20% 14.80%
	86.18% 13.82%	87.28% 12.72%	86.71% 13.29%

(a) Logistic regression: binary

	Predicted Unwell	Predicted Well	
Actual Unwell	3561 49.42%	125 1.74%	96.61% 3.40%
Actual Well	250 3.47%	3270 45.38%	92.90% 7.10%
	93.44% 6.56%	96.32% 3.68%	94.80% 5.20%

(b) Logistic regression: categorical

	Predicted Unwell	Predicted Well	
Actual Unwell	3157 43.81%	529 7.34%	85.65% 14.35%
Actual Well	478 6.63%	3042 42.22%	86.42% 13.58%
	86.85% 13.15%	85.19% 14.81%	86.03% 13.97%

(c) Logistic regression: scalar

	Predicted Unwell	Predicted Well	
Actual Unwell	3660 50.79%	26 0.36%	99.30% 0.71%
Actual Well	0 0.00%	3520 48.85%	100.00% 0.00%
	100.00% 0.00%	93.09% 6.92%	99.64% 0.36%

(d) Random forests: binary

	Predicted Unwell	Predicted Well	
Actual Unwell	3590 48.64%	96 1.33%	98.40% 2.60%
Actual Well	15 0.21%	3505 48.64%	99.57% 0.43%
	99.58% 0.42%	97.33% 2.67%	98.46% 1.54%

(e) Random forests: categorical

	Predicted Unwell	Predicted Well	
Actual Unwell	3629 50.36%	57 0.79%	98.45% 1.55%
Actual Well	0 0.00%	3520 48.85%	100.00% 0.00%
	100.00% 0.00%	98.41% 1.59%	99.21% 0.79%

(f) Random forests: scalar

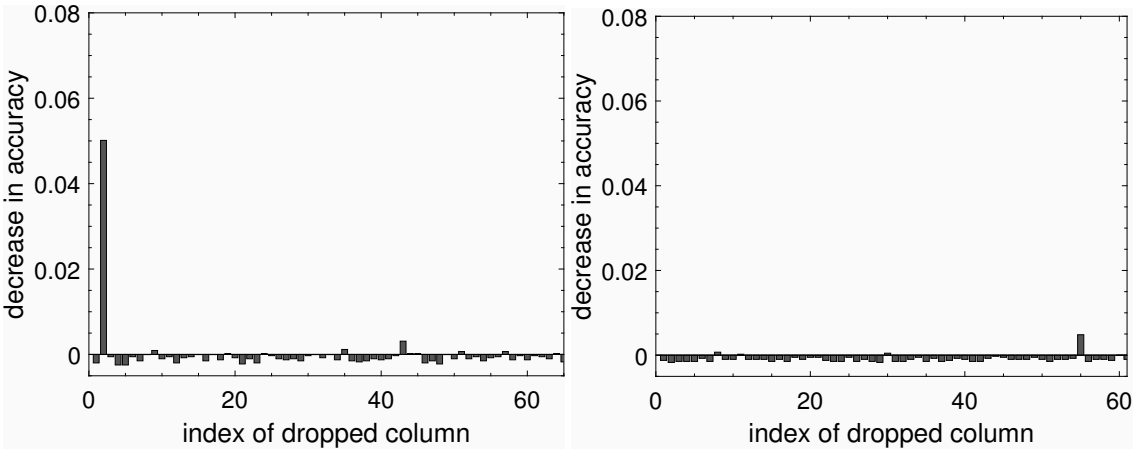
Table 10 Confusion matrices after applying NMF to matched baseline questions. We choose option 3 in Section2.2 to compute the confusion matrices.

	Binary	Categorical	Scalar	Combined
Index	2	55	23	2
Maximum	0.0501	0.0048	0.0732	0.0042

Table 11 SVM with dropped column: The input dataset is the combined Well and Unwell baseline questions. The maximums are the largest differences between the original SVM accuracy for that data type with all columns and with the selected column dropped.

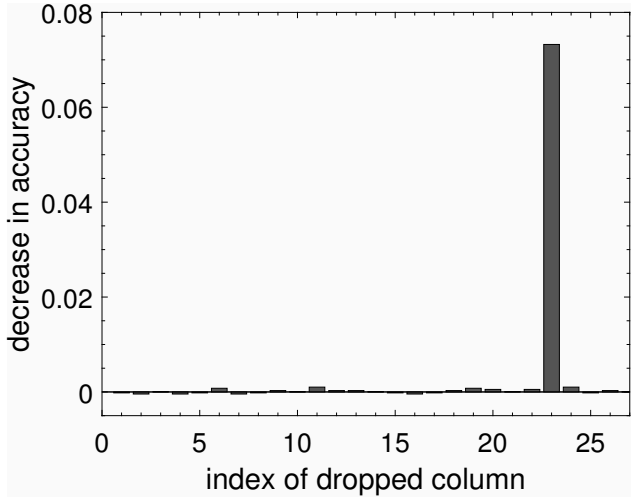
increase may be so small it is insignificant. Table 11 lists the indices and corresponding changes in accuracy for the maximum change in each data type. Column 2 in the binary data is the question TK-BITE-LOC-R1: "I don't know the location where I attained the tick bite." Column 23 in the scalar set is the question TX-IMPRV-DX-B1: "Compared to when I first began treatment, I CURRENTLY feel..." with answers scaling from not better to more than 85% better. Column 55 in the categorical data is part of a one-hot encoding for the question STAT-Well-SK-B1: "With regard to my Lyme disease status, I would say that I am CURRENTLY..." The answers range from "ill" to "Well", including options for "okay but still treating" or "unsure". Column 55 specifically corresponds to the answer choice "Well", which is how LymeDisease.org assigns Well and Unwell labels. The fact that this method finds the specific column that the organization uses to classify Well and Unwell indicates that it is a reliable method. However, the fact that binary column 2 and scalar column 23 impact accuracy so much more indicates that there are underlying differences in the two sets of patients beyond the question used to sort them. Location is an important factor, which may be tied to the discrepancies in awareness of the disease and treatments available across the country.

3.4.2. *Simple Binary Classification Feature Importance.* For categorical data, The 13th question (STAT-STAGE-B1) is the most sensitive categorical question. But that question together with 12th question is kind of indicator. Among others, the most important question is 10 (DX-STAGE-B1). 4 (D-SEX-PREF-R1), 7 (TK-TX-ME-B1), and 12 (STAT-Well-SK-B1), Questions 8 (SX-RASH-SPEC-B1) are also somewhat sensitive, but much less than the four previously mentioned. Question 13 asks the patient's current stage of Lyme disease, such as early Lyme, initial bull's eye rash, late stage untreated, and chronic Lyme disease. Question 12 asks about wellness of the patient with answers, such as "sick", "okay with antibiotics", "okay but not treating", and "well". This question is how LymeDisease.org classifies the patients into



(a) Binary

(b) Categorical

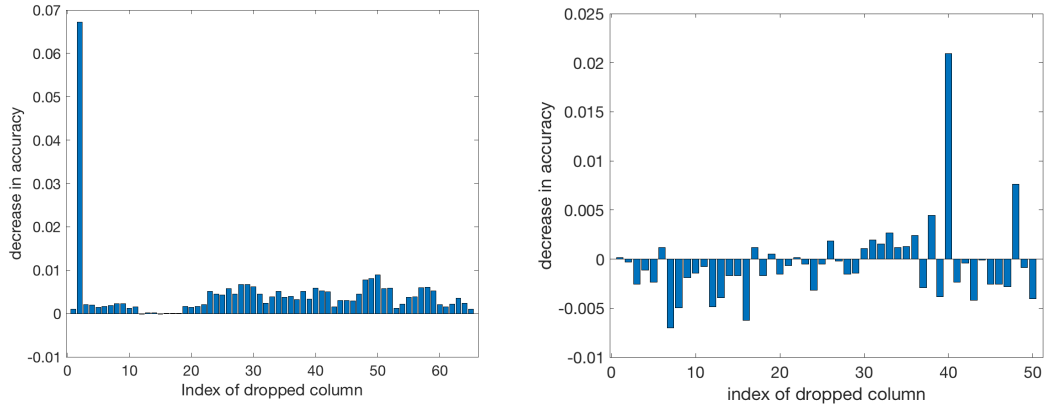


(c) Scalar

Figure 15 SVM with omitted column: The input dataset is the combined Well and Unwell baseline questions. Each bar corresponds to the difference between SVM accuracy with the full set and SVM with one column omitted. The most significant columns are binary column 2, categorical column 55, and scalar column 23.

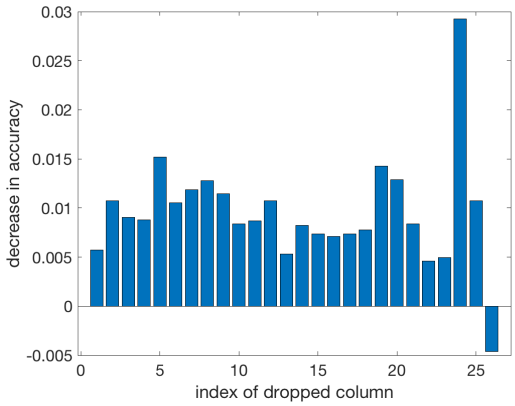
Well and Unwell. As a result, it should be expected to have a high importance in classifying Well and Unwell. The 7th categorical question is "After receiving the treatment for my tick bite to prevent Lyme disease, I..." The answers include "Developed Lyme disease anyway," "Did not develop Lyme disease," and "Don't know". This question is also similar to directly asking Well and Unwell. The 10th question is "The stage of my illness when I was first diagnosed with Lyme disease is the best described as..." which is the most sensitive questions besides indication question.

Of the 27 scalar questions, we found the 23rd question in the survey (TX-IMPRV-DX-B1) is most sensitive. This clearly relates to classifying Well and Unwell and it is kind of indication question for well and unwell classification. So when we do numerical experiments, we delete the data corresponding to this questions. Interestingly, when we omit 26th question (DX-AGE-SX-AGE-B1), our accuracy increases as opposed to staying constant or decreasing as expected. The 23rd question asks "Compared to when I first began treatment, I CURRENTLY feel..." with answers scaling from not better to more than 85%



(a) Binary

(b) Categorical

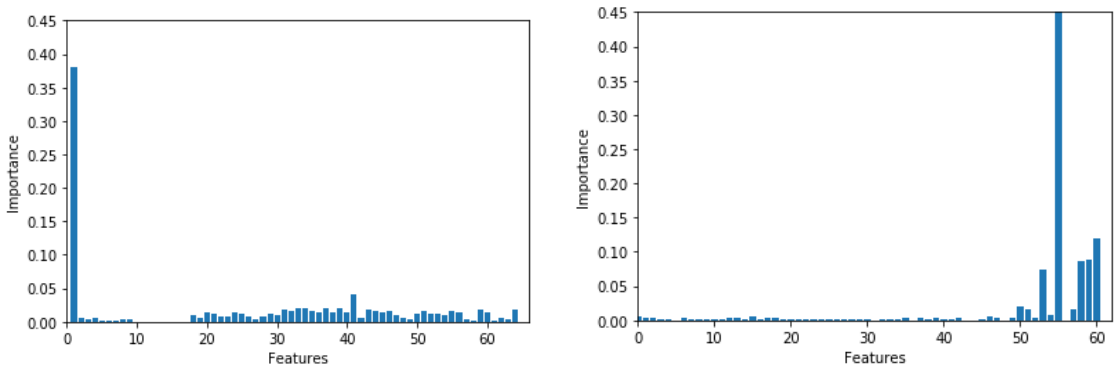


(c) Scalar

Figure 16 SBC with omitted question: The input dataset is the combined Well and Unwell baseline questions, undersampled by option 2. The x-axis indicates the index of the omitted baseline option.For categorical data, we drop options belong to a question each time. We use 396 patients from each group for training and 44 patients from each group for testing. We use 10-fold cross validation to measure accuracy. The dotted line is the baseline accuracy without any questions omitted. m= 60, L= 5. We run 10 trials per measurement number and layer number to approximate average accuracy. The data we use exclude sensitive questions.

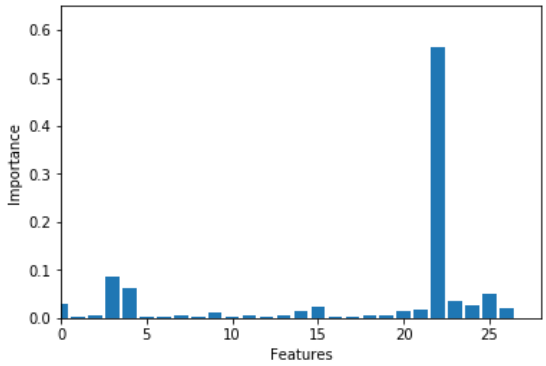
better. Beside the indication questions, the most important question is DX-NUM-HCP-B1"Before being properly diagnosed with Lyme disease, the number of physicians I saw about my symptoms was… "

For categorical and scalar data, we omitted one question at a time. However, for binary data, we omit one column at a time, which corresponds to a single answer choice from a question. Figure16a indicates several local minimum of the graph. The most significant minima at index 2 corresponds to TK-BITE-LOC-R1, stating the patient does not know the location of the most recent tick bite. The other minimum correspond to the early symptoms (SX-EARLY-SPEC-B1) answers "pain in large joints" and "fainting, shortness of breath, or chest pain"; symptoms at diagnosis (SX-DX-B1) answers "evidence of tick bite," "muscle aches," "cognitive impairment," and "psychiatric"; diagnotic tests (DX-LABS-SPEC-B1) answers "PCR" and "spinal tap"; and lastly misdiagnoses (DX-MISDX-SPEC-B1) answers "multiple sclerosis," "motor neuron disease," and "multiple systems atrophy".



(a) Binary

(b) Categorical



(c) Scalar

Figure 17 Random Forest feature importance: The input data are all the baseline questions of both Well and Unwell patients. We apply Random Forest implementation from the scikit-learn package in python and visualize its feature importance through barplot. The x-axis shows the question indices. For binary data, the question at index 1 is the highest. This refers to [TK-BITE-LOC-R1] "Where in the United States your most recent tick bite occurred" with response "I don't know the location where I attained the tick bite." For categorical data, the question at index 58 is the highest. This refers to [STAT-STAGE-B1] "The CURRENT stage of my Lyme disease would best be described as" with response "early or disseminated Lyme disease (flu-like, neurologic, cardiovascular or musculoskeletal symptoms)." For scalar data, the question at index 22 is the highest. This refers to [TK-TX-DUR-B1] "The length of time I was treated for the tick bite."

3.4.3. *Random Forest Feature Importance.* The feature importance function for random forests measures how strongly the feature and the data correlate. The most important feature questions for the three data types are:

(1) Figure 17a Binary: [TK-BITE-LOC-R1] Where in the United States your most recent tick bite occurred with response "I don't know the location where I attained the tick bite."
(2) Figure 17b Categorical: [STAT-STAGE-B1]: The CURRENT stage of my Lyme disease would best be described as "early or disseminated Lyme disease (flu-like, neurologic, cardiovascular or musculoskeletal symptoms)."
(3) Figure 17c Scalar: [TK-TX-DUR-B1]: The length of time I was treated for the tick bite was "[1]: One or two days; [2]: Two weeks or less; [3]: Three weeks or less; [4]: Four weeks or less; [5]: More than four weeks; [9]: Don't know."

Antibiotics	Original Patient Data	Model Prediction	Interpretation
Alinia	1	1	Matched
Amoxicillin	1	1	Matched
Amoxicillin Clavulanate	-1	0	Originally unanswered, predict it might not work
Biaxin	-1	1	Originally unanswered, predict it might be *beneficial*
Cedax	-1	0	Originally unanswered, predict it might not work
Cefuroxime	0	0	Matched

Table 12 Interpreting Antibiotic Recommendation (Synthetic Data). The randomly generated patient data in this example matches the recommendation for three of the six selected antibiotics. For the three antibiotics the patient did not give a response for in the survey, the model predicts which will and will not be beneficial for the patient based on similar Well patients. A patient could then take this information and ask a doctor for more information about the recommended antibiotics.

4. Recommend antibiotics for Unwell patients

4.1. **Recommender Framework with Neural Networks.** We want to use the patterns of Well patients to predict the choice of antibiotics for Unwell patients. In this way, we base our recommendations for Unwell patients on the best prescription of similar Well patients. We utilize the framework introduced in Section 1.2 and shown in Figure 1. First, we train a classifier on Well patients with baseline answers and antibiotic prescription as features and target respectively. Next, we evaluate the effectiveness of this classifier on Well patients. Then, we apply this classifier to Unwell patients. Finally, we compare our model's output with the original Unwell patients' answers. These comparisons can tell us what improvements can be made to Unwell patients' antibiotics as well as provide a sense of how well the framework works.

We make four assumptions in this framework:

- The fact that by Well patients are not ill means their antibiotics are effective for these individuals.
- We only consider the antibiotics that patients are currently taking.
- There is no routine prescription for antibiotics.
- We disregard the effect of combination therapies.

We trained a fully connected neural network (FCNN) model with the Well patients' baseline questions as input. The outputs are their responses to antibiotic related questions (embedded as one-hot vectors). An example of the output from this model is given in Table 12. The 10-fold cross-validation score on Well group is 0.96, which indicates that the model is trained successfully.

We then apply this neural net to the Unwell group, and we find that of 34% patients totally match our model's prediction. Meanwhile, 22% of patients have antibiotic regimes that differ by one out of 37 potential options from our prediction. Also, 14% of patients currently take antibiotic regimes that differ by two out of 37 from our prediction, and 30% of patients currently take antibiotics which differ by more than 2 options from our prediction. The large percentage of patients whose actual responses match our recommendations suggests that our model is reliable and matches real-world data, while the portion that differs significantly from our recommendations indicates that our model is not overfitting the data. Overall, this suggests that Unwell and Well patients have been prescribed antibiotics in inherently different ways. We can use this information to recommend improved antibiotic regimes that are more similar to Well patients and thus improve Unwell patients' health.

This framework can offer us insights on the redistribution of antibiotics. We use a state map to visualize suggested changes. We compare the original data from the survey and the predicted data from our framework by subtracting the original from the predicted value and storing it in a matrix. We then

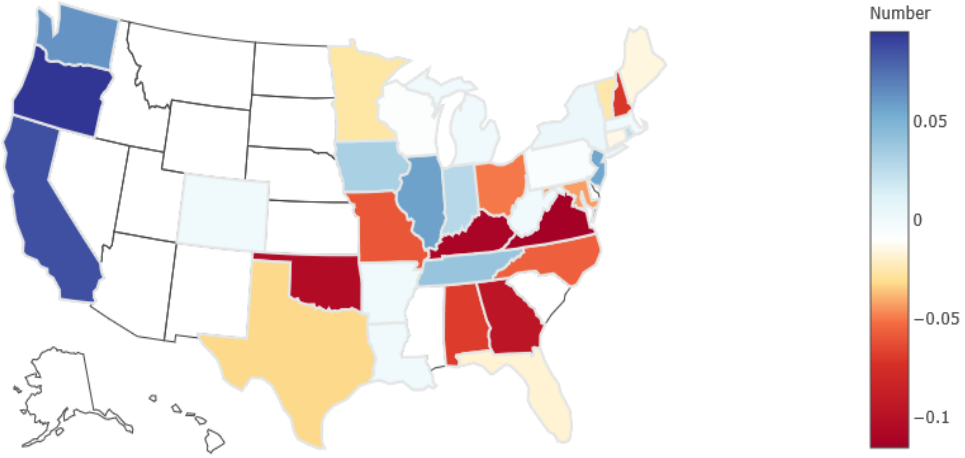


Figure 18 State map of the oral antibiotic Minocin (Minocycline) showing the percentage difference between actual data and prediction with our model. Only states with more than 13 patients in the sample are colored. This graph illustrates a potential redistribution of the antibiotic. Red color means fewer predictions, while blue suggests more. A darker color corresponds to a larger percentage difference in a state.

Antibiotics	Number of People	Antibiotics	Number of People
Alinia	49	Cleocin	63
Amoxicillin	152	Doxycycline	695
Augmentin	40	Flagyl	231
Bactrim	154	Ketek	1
Biaxin	239	Lariam	7
Cedax	6	Levaquin	34
Ceftin	235	Malarone	152
Cipro	72	Mepron	145

Table 13 Distribution of oral antibiotics usage. This table shows number of patients who are taking certain kind of oral antibiotics(Partial).

plot the difference matrix onto a state map. Each state map illustrates a potential redistribution of an antibiotic. States are colored red to indicate that our framework recommends fewer patients to take the selected antibiotic than are currently taking it in that state, while states are colored blue to indicate more patients are recommended to take that antibiotic.

4.2. **Recommender Framework with SBC.** We also test the recommender framework where the classifier trained is SBC rather than a neural network. Here, we explore whether an antibiotic treatment will

Unwell	Number of People	Well	Number of People
Bicillin	59	Bicillin	153
Claforan	2	Claforan	38
Rocephin	19	Rocephin	0

Table 14 Distribution of intramuscular antibiotics usage This table shows number of patients who are taking certain kind of intramuscular antibiotics. The question is: Over the course of my illness, at one time or another, my treatment included the following intramuscular (IM) antibiotics. 16 Well patients answer don't know; 301 patients among 440 Well patients answer that Not applicable - did not take intramuscular antibiotics

work for a patient. Instead of focusing on one antibiotic, we want to regard the antibiotics taken by patients as a general treatment. There are two kinds of patients in our initial classification situation. Case 1 includes Unwell patients who report a worse condition after a period of treatment and Unwell patients who report no change over a long period of time. People would not usually take antibiotics for a very long time due to side effects and costs. Most patients will get much better after the first three months of antibiotic treatment, and they will use more time to treat their chronic disease. In this way, we regard those Unwell patients who have taken antibiotics for more than one year and situations of whose Lyme diseases are unchanged to be those choosing ineffective treatment plan. Case 2 includes Well patients. We regard their treatments to be effective. We assume that any combination of antibiotics taken by Well patients is regarded as effective. In this section, we use the following survey questions to assign class labels:

TX-GROC-U1:In general overall, I would say that with antibiotic therapy, my Lyme symptoms are...

CTX-ABX-DUR-U1:I have been on my CURRENT antibiotic treatment protocol for...

W-200: TX-BEST-ALT-ABX-W1:Looking back, I would say the MOST EFFECTIVE treatment protocol that I have used to improve my health was...

We use the first two questions to select patients who feel antibiotics are not effective (worse and no change over long period of time); we use the last question to select patients who feel their antibiotic treatment works. Once we have selected these patients, we use their baseline data as input to the recommender framework and evaluate accuracy with 10-fold cross-validation. We undersample using option 2 from Section 2.2, but the resulting equal class sizes may be smaller than 440 if there are fewer than 440 patients who satisfy the above conditions. Figures 19 and 20 show the average accuracy when we use our algorithm to tell if the general treatments work for the patients.

4.3. **Important Features for Predicting Antibiotic Effectiveness.** Our data set consists of 440 Well patients and 3686 Unwell patients. In Section 4.1, we train our model on Well patients but predict them on Unwell patients. However, there may be a structural difference between the two groups. We hope to identify what that difference may be by looking at important features for the effectiveness of antibiotics. Moreover, we hope to find what features influence whether an antibiotic effective or not. In this section, by looking at feature importance from random forest classifier and ECOC respectively on Well group and Unwell group, we find important features of the Well and Unwell groups which influence effectiveness of antibiotics.

4.3.1. *Random Forest Feature Importance.* The input of our experiments is all matched baseline data from Unwell or Well patients. We label each antibiotic one if it is effective or two if ineffective. We assume the antibiotics a Well patient reports taking are effective while antibiotics they report not taking

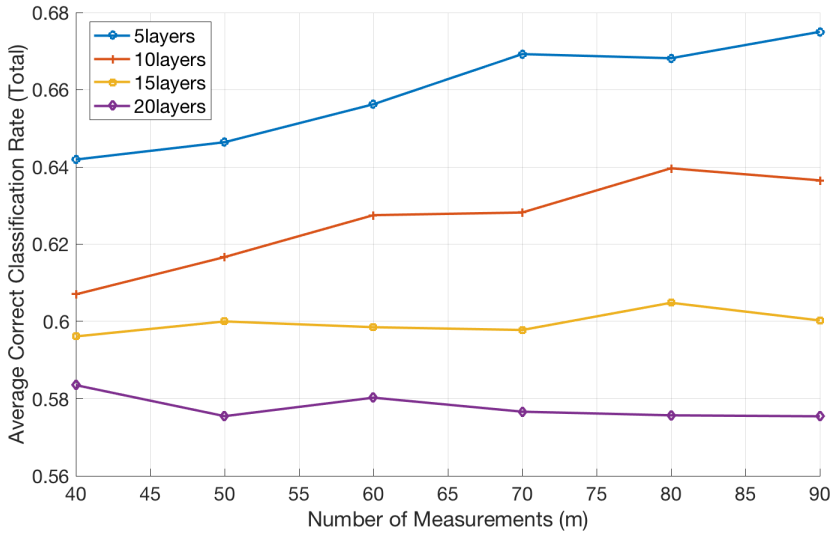


Figure 19 SBC on general antibiotics treatments. In this graph, the input are all the patients who belongs to case 1 and case 2 mentioned in 4.2. Data of each patient are the combination of scalar data, categorical data and binary data. We use 5 layers(blue), 10 layers(red), 15 layers(orange), 20 layers(purple) and measurements from 8 to 40 as hyperparameter of simple classification method. However, what surprises us is when we increase layer number the average accuracy decrease.

or do not report on are assumed to be ineffective. We label whether antibiotics are effective for Unwell patients based on a question in phase 1:[TX-GROC-U1]In general overall, I would say that with antibiotic therapy, my Lyme symptoms are better, worse, or unchanged. When we use random forest method to do classification, the algorithm returns the importance of each feature. We store the importance for each antibiotic in a matrix. For each antibiotic, we select the five most important features. Some features appear many times as one of the top five features. Finally, we select the nine features having the largest number of appearances. The second method adds up all feature importance values for each feature. We use this value to assign significance. With our experiments, we learn some questions and responses (options) in the survey that are important for classifying the effectiveness of antibiotics.

In this part, we use the feature importance assigned by the random forest method to select important options for measuring the effectiveness of antibiotics. We don't under-sample or oversample for random forest method. We select the nine most important options in the survey. Figures 21 and 22 show the relative importance of these nine features for determining the antibiotic effectiveness. The larger the feature importance value, the lighter the box will be. Figure 22shows the feature importance of several selected features on all the antibiotics. The nine most important features, listed in 22 are

(1) -Self-identified Gender: Male
(2) -Time taken to correct any misdiagnosis
(3) -False positive blood test for Lyme disease
(4) -Highest Level of Education
(5) -Feeling compared with begin of treatment
(6) -Current general health
(7) -Before being properly diagnosed with Lyme disease, the number of physicians I saw about my symptoms was

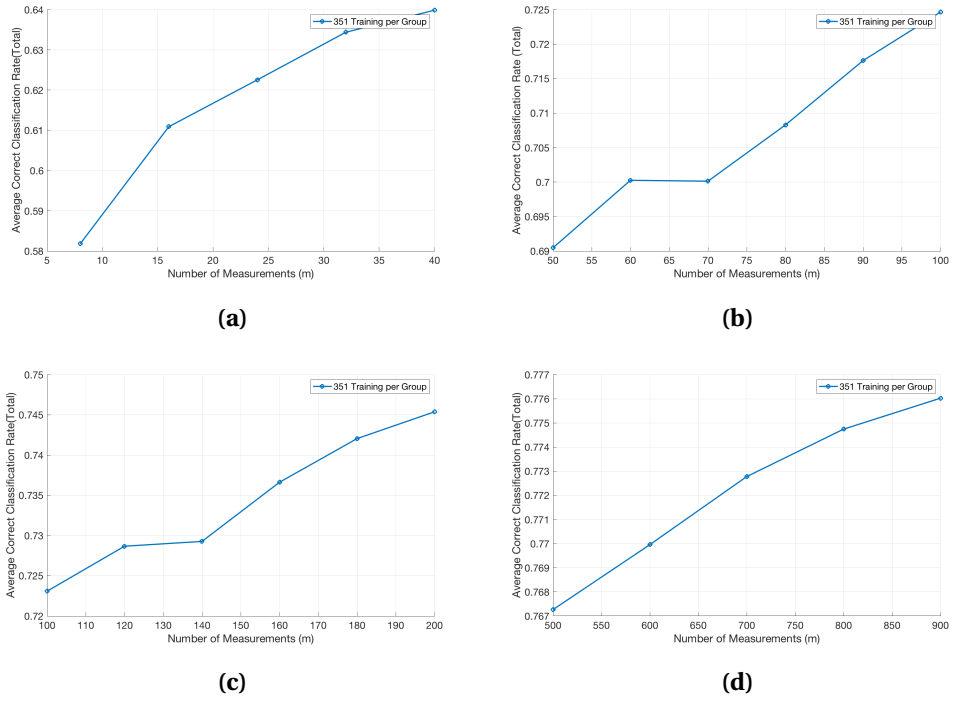


Figure 20 SBC on general antibiotics treatments :In this graph, the input are all the patients who belongs to case 1 and case 2. Data of each patient are the combination of scalar data, categorical data and binary data. We use 5 layers(blue) and measurements from 8 to 900 as hyperparameter of simple classification method and average our result by 30 trials. These graphs show the accuracy of simple classification algorithm for telling whether the general treatment for patients works or not.

(8) -Age:first experience symptoms
(9) -Age:receive a diagnosis

The nine most important features listed in Figure21 are

(1) -Ethnicity: Hispanic or Latino
(2) -The type of rash I developed after my tick bite was an irregular rash
(3) -Length of time the tick had been attached when I noticed or removed it
(4) -Time took to correct any misdiagnosis
(5) -Total Family Income
(6) -Current general health
(7) -Before being properly diagnosed with Lyme disease, the number of physicians I saw about my symptoms was
(8) -Age first experienced symptoms
(9) -Age received a diagnosis

Most important features we learn make sense. For example, in figure 22 there is one column really light which means its high importance on antibiotics effectiveness. This feature is called current health level which is directly relevant to effect of antibiotics. Another example is false positive blood test. That means healthy person is regarded as patients. In this way, it seems that no matter what antibiotics they take, symptoms will disappear soon. The fact is exactly what we guess. In our data set, 89% unwell patients with false positive blood test gets better or unchanged regardless of what antibiotics they take. However,

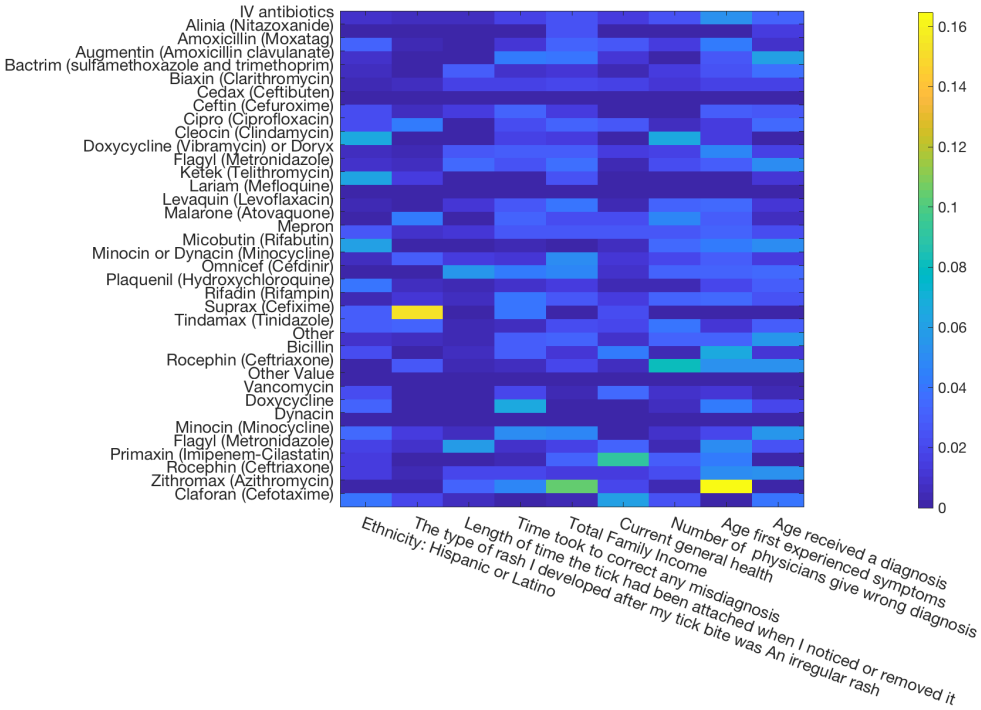


Figure 21 The graph shows the nine most important features importance from random forest during classification of effective antibiotics and ineffective antibiotics on Well patient. The lighter colors are, the more significant the features will be. For each antibiotics, we select five most important antibiotics. Some feature appear many times as top five features. Finally, we select the most important nine feature having largest number of appearance.

there are also some other important features such as highest level of education or self-identified gender whose relationship with Lyme disease is not clear.

Note that if we wish to determine importance of questions rather than importance of options, we must aggregate importance of all options corresponding to a single question.We sum importance for each question for each question; the heatmap representing these two model of importance are presented in Figure 23 and Figure 24. From the two graphs we can clearly notice that there is no much difference between importance options in Figure 23 except there are one column really light. What's more, the importance options we most belong to scalar question where each question has only one option. However, from the Figure 24, we can find that some question really important but we miss that information when we only care about option in survey.

We select the nine most important features. The important questions common between the Well and Unwell groups are listed here:

(1) -Misdiagnosed with other conditions
(2) -Initial symptoms when said patient might have Lyme disease
(3) -Positive diagnostic test standards
(4) -Experienced symptoms
(5) -Marital Status
(6) -Type of healthcare provider firstly give diagnosis of Lyme disease

For the Well group, the additional important feature are

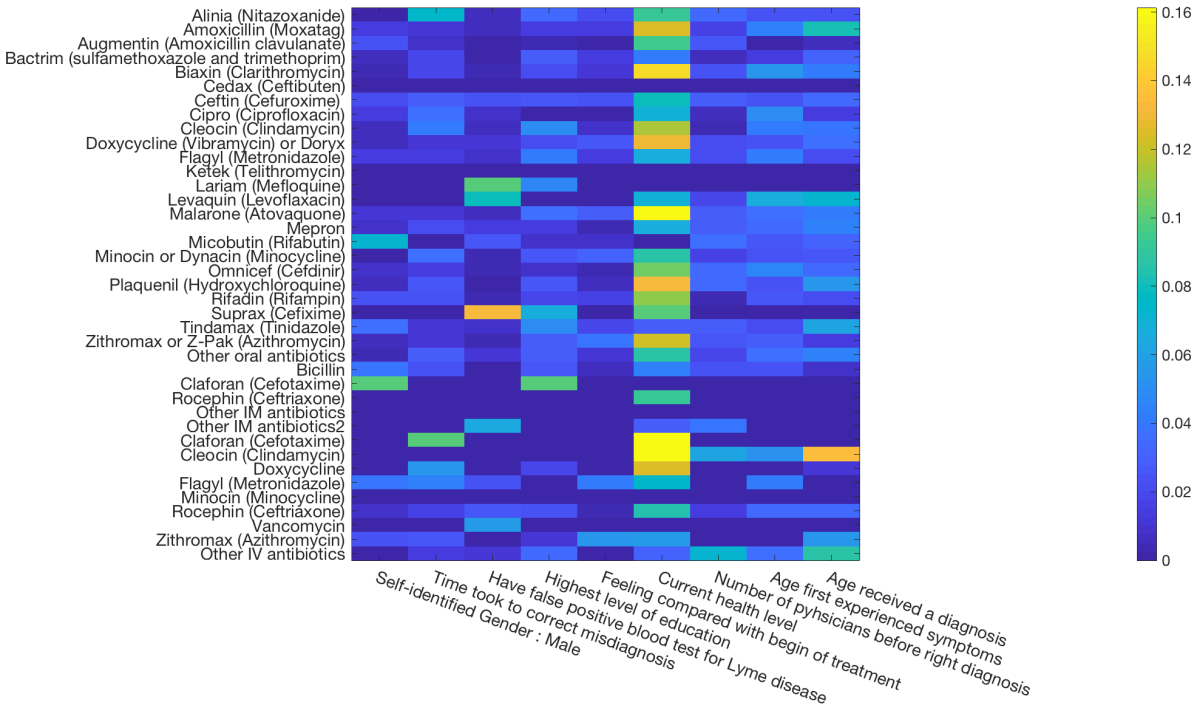


Figure 22 The graph shows the nine most important features importance from random forest during classification of effective antibiotics and ineffective antibiotics on Unwell patient. The lighter colors are, the more significant the features will be. For each antibiotic, we select the five most important antibiotics. Some feature appear many times as one of first five feature.Finally, we select the nine most important features having largest number of appearance.

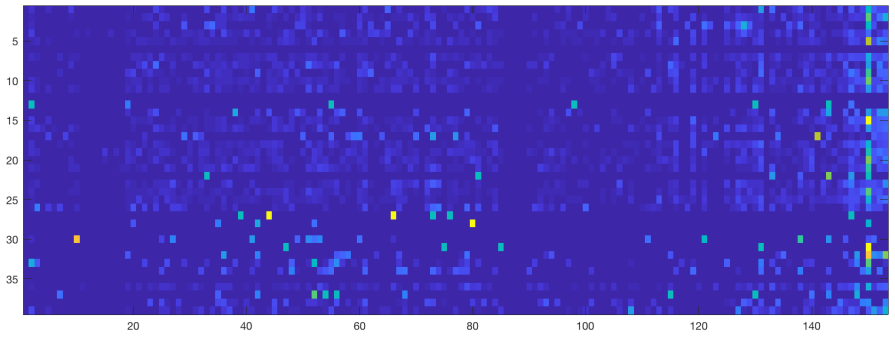


Figure 23 The graph shows the feature importance of all matched baseline options from random forest (using options in survey as feature) during classification of effective antibiotics and ineffective antibiotics on Unwell patient. The lighter colors are, the more significant the features will be.

(1) -Ethnicity
(2) -Sexual Orientation or Preference
(3) -Stage of my illness when first diagnosed with Lyme disease

For the Well group, the additional important feature are

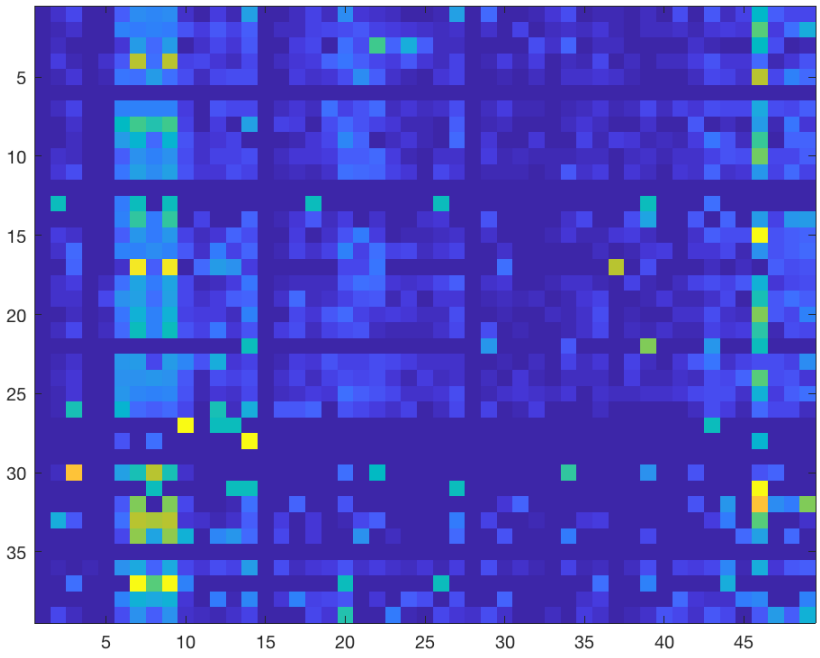
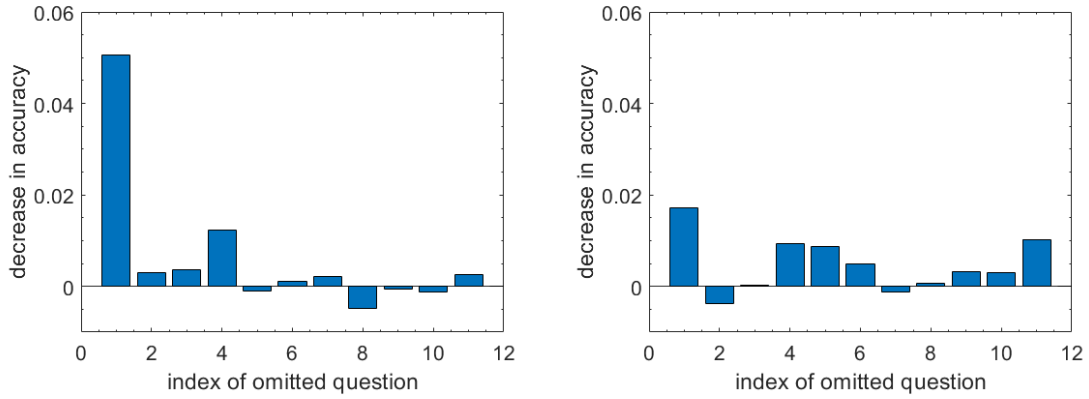


Figure 24 The graph shows the feature importance of all questions in survey from random forest during classification of effective antibiotics and ineffective antibiotics on Unwell patient. The lighter colors are, the more significant the features will be.

(1) -General health
(2) -Age first experienced symptoms
(3) -Age received a diagnosis

Several of these features that are important for the performance of antibiotics are intuitive. Age and general health impact the patient's immune system. Time between symptoms and diagnosis, time to notice and remove ticks, and misdiagnoses are highly related to the development of illness before getting proper treatment. Furthermore, symptoms, especially rash after tick, initial symptoms, experienced symptom, determines the severity and intense of Lyme disease in the initial stage. For these important features, we would like to know their exact relationship with the effectiveness of antibiotics. There are also some unintuitive features that play an important role in the performance of various antibiotics. Features, such as highest level of education, self identified gender, sexual orientation or preference, ethnicity, family income, test standard, marital status, and type of health-care provider, were also selected. This requires future analysis.

4.3.2. *ECOC Feature Importance.* In addition to predicting the best antibiotic treatments, we investigate which factors related to antibiotics impact the effectiveness of the treatment most. The input data are selected columns exclusively from the Well dataset asking about antibiotics and coinfections. To classify these patients, we used responses from either the question CTX-ABX-ME-U1 or TX-GROC-U1. The first asks the patient to rate the level of effectiveness of his/her current antibiotic treatment. The second



(a) CTX-ABX-ME-U1 **(b)** TX-GROC-U1

Figure 25 ECOC Classifier Accuracy The input data consist of the columns for the question codes listed in Appendix B.1. One question (which may correspond to multiple columns) is left out each time to find which is most significant in classifying. The experiment is done twice, first classifying based on the patient's self-report of the effects of the antibiotic (CTX-ABX-ME-U1) and second based on the patient's self-report that his/her Lyme symptoms are better, unchanged, or worse (TX-GROC-U1). The classifier is an error-correcting output code classifier using binary SVM learners.

question asks whether Lyme symptoms are worse, unchanged, or better with antibiotic therapy. The predictor columns relate to current antibiotic treatment, other medications, and coinfections. The question codes and descriptions for these are listed in Appendix B.1.

We used error-correcting output codes (ECOC) classifier using SVM learners for classifying multiple classes. To classify more than two groups, the ECOC model trains the first learner on classes one and two, the second learner on classes two and three, and so on. Here, accuracy is given by $A = 1 - out\text{-}of\text{-}sample\ loss$ from 10-fold cross-validation. Out-of-sample-loss is not evaluated on the training data in order to avoid optimistic accuracy caused by overfitting to noise in the training set. Accuracy with all 11 questions classified according to CTX-ABX-ME-U1 is 0.7035 and according to TX-GROC-U1 is 0.5570. In addition to testing this set of 11 predictor questions, we tested the set less one question, rotating which was left out (Figure 25). Question code (1) made the largest impact on classification when left out, indicating that the duration of treatment for the current antibiotic is the most significant of these variables for determining the effect of the antibiotic.

5. Recommender Framework Application on the Most Effective Alternative Treatment

We can also apply the recommender framework in the context of alternative treatments. We train the model with known patients' responses regarding the performance of alternative treatments and predict on new patients.

In this section, our goal is to identify the most effective alternative treatments for new patients. We assume the effectiveness of alternative treatments is accurately represented by the Well patients' self-report, which is not standardized. We also disregard the effect of combination therapies. In order to produce recommendations on the potentially most effective alternative treatments, we trained a fully connected neural network model on the Well patients' baseline questions and used their responses to the most effective alternative treatment as the output label, question code W-260: TX-BEST-ALT-W1. The 10-fold cross-validation score on the Well group is 0.87, indicating that the model is trained relatively good. The basic procedure is:

Alternative Treatments	Effectiveness
Acupuncture	1
Collodial Silver	0
Herbal Protocols	0
Homeopathy	0
Rife Machine	0
Stem Cell Therapies	1
...	...

Table 15 Sample Interpreting Table on Application 1: Predicting Effectiveness of Alternative Treatments for a new patient (Synthetic Data). Input data is this new patient's baseline questions. Output is the predicted effectiveness of all alternative treatments on this patient from our recommender framework. 1 indicates that the alternative treatment is effective, while 0 indicates ineffective.

(1) Train the classifier with all baseline questions of Well patients as input data and the most effective alternative treatments as labels.
(2) Evaluate the effectiveness of this classifier on Well patients training data.
(3) Apply this classifier to new patients.
(4) Predict the most effective alternative treatments for the new patients.

We only train our recommender framework on the most effective alternative treatment of the Well patients as only Well patients answer these questions due to the branching of the survey. In the Table15, we see the potential output of our recommender framework on alternative treatments effectiveness. Effective alternative treatments are denoted by 1, and ineffective treatments are denoted by 0.

6. Recommender Framework Application on the Level of Side Effects of Alternative Treatment

We apply the recommender framework to predict the level of side effects of alternative treatments. In this section, we use the patterns within Unwell patient data to predict the level of side effects of alternative treatments for a new patient. We assume the level of side effects of alternative treatments are accurately represented by the Unwell patients' self-reports, which is not standardized. We also disregard the side effects of combination therapies. There is also a mismatching issue in the question of level of side effects in the Unwell scalar phase 1 data. In the *dictionary.txt* file, question code [U-2100: CTX-ALT-SPEC-U1] indicates the question: "The alternative treatment approaches I am CURRENTLY using are indicated below and rated for effectiveness... (select only those that apply)"; however, in the Unwell phase 1 *README.txt* file, question code [CTX-ALT-SPEC-U1] contains answers to level of side effects experienced with all kinds of alternative treatments. We assume that there is some typo in the file *dictionary.txt*.

We train a fully connected neural network model in the package Pytorch, with Linear as the activation function for the input (156 nodes) and output (15 nodes) layer and ReLU as the function for the hidden layers (100 and 64 nodes). We choose mean-squared-error to be the loss function. The Unwell patients' baseline questions is the input, and their answers to the level of side effects of alternative treatments as the output labels (question code U-2100: CTX-ALT-SPEC-U1). The 10-fold cross-validation score on the Unwell group is 0.96, which means that the model is trained successfully. The basic procedure is:

(1) Train the classifier with all baseline questions of Unwell patients as input data and level of side effects of alternative treatments as labels.

Alternative Treatments	Side Effect Level
Acupuncture	1
Collodial Silver	0
Herbal Protocols	3
Homeopathy	1
Rife Machine	0
Stem Cell Therapies	2
...	...

Table 16 Interpreting Table on Application 2: Predicting Level of Side Effects of Alternative Treatments for a new patient (Synthetic Data). Input data is this new patient's baseline questions. Output is the predicted level of side effects of all alternative treatments on this patient from our recommender framework. 0 means there is no effect present, 1 indicates there are mild side effects, 2 stands for moderate side effects, and 3 represents severe side effects.

(2) Evaluate effectiveness of this classifier based on Unwell patients.
(3) Apply this classifier to new patients.
(4) Predict the potential level of side effects of various alternative treatments for new patients.

We only train our recommender framework on the level of side effects of alternative treatment of the Unwell patients, since due to the branching of the survey, only Unwell patients answer these questions. In the Table 16, we see the potential output of our recommender framework on level of side effects of alternative treatments. Here, 0 means there is no effect present, 1 indicates there are mild side effects, 2 stands for moderate side effects, and 3 represents severe side effects.

7. Conclusions

In our paper, we try to better understand Lyme disease through data analysis. We apply a variety of classification methods on MyLymeData to identify what factors influence a patient's wellness. We found that knowing the geographical location of the tick bite, early symptoms such as faintness and chest pains, and psychiatric symptoms at diagnosis are critical. In order to generate medical recommendations for Unwell patients, we propose a deep recommender framework. These suggestions include antibiotics to take, alternative treatments to try, and predict side effects of treatments. With this model, we provide insights to clinicians and patients about antibiotic and alternative treatments.

8. Future Work

In the future, we would like to apply NMF with heatmaps to the SVM classification of Well and Unwell patients in order to identify important feature topics. For the feature importance investigation already done, we would like to know which specific answers to the important questions correspond to Well and Unwell. We are interested to see if it is possible to use NMF on the binary data with SBC to vary the size of input factor matrix. The goal would be to vary the number of layers by varying the size of the direct binary input. We recognize that there are questions that almost directly indicate wellness, such as "stage of illness", "improvement since treatment", and "current state of wellness". We need to remove these columns and reevaluate our experiments that use them. Another way we can balance the dataset is by redefining the Well and Unwell classes. Originally, only those patients who report full recovery are classified as Well (Question STAT-WELL-SK-B1). However, we would like to examine the results if we include those who report to be okay but not fully recovered in the Well class. We can also define the Well

and Unwell classes based on a question that asks how recovered the patient is (Question TX-IMPRV-DX-B1). In particular, we would like to evaluate the accuracy of our recommender framework in cases where we do not have a base truth to evaluate accuracy in two ways: by comparing predictions for patients who are more or less well according to TX-IMPRV-DX-B1 and by using testing validation.

ACKNOWLEDGEMENTS

We would like to thank Lorraine Johnson, JD, MBA, CEO of LymeDisease.org for her direction and providing us with MyLymeData as well as the many patients who provided their data. Many thanks to Professor Deanna Needell, Dr. Jamie Haddock, Denali Molitor, and Dr. Anna Ma for organizing and guiding the project. We are grateful to Professor Andrea Bertozzi for organizing the 2018 REU. We also thank the 2017 Lyme Disease REU team for their work embedding the MyLymeData.

REFERENCES

[1] E. Allwein, R. Schapire, and Y. Singer. Reducing multiclass to binary: A unifying approach for margin classifiers. *Journal of Machine Learning Research*, 1:113–141, 2000.

[2] Christopher M. Bishop. Pattern recognition and machine learning, 2006.

[3] Lorraine Johnson, Spencer Wilcox, Jennifer Mankoff, and Raphael B Stricker. Severity of chronic Lyme disease compared to other chronic conditions: a quality of life survey. *PeerJ*, 2:e322, 2014.

[4] Vrushali Kulkarni and Pradeep Sinha. Random forest classifiers: A survey and future research directions. 36:1144–1153, 01 2013.

[5] Hyekyoung Lee, Jiho Yoo, and Seungjin Choi. Semi-supervised nonnegative matrix factorization. *IEEE Signal Processing Letters*, 17(1):4–7, 2010.

[6] D. Needell, R. Saab, and W. Toolf. Simple classification using binary data. 2017. Submitted.

[7] Shai Shalev-Shwartz and Shai Ben-David. Understanding machine learning: From theory to algorithms, 2014.

APPENDIX A. NOTES ON MYLYMEDATA

A.1. **Extra Column in Well Phase 1 Scalar Matrix.** The first column of scalar phase 1 Well patients should be removed, because this is the index of patients. Also, this first column of indices is offset by 2.

A.2. **Effectiveness of Current Antibiotics.** Question CTX-ABX-ME-U1 asks "I consider the effectiveness of my CURRENT antibiotic treatment protocol to be:"

1 Too soon to tell
2 Mildly effective
3 Moderately effective
4 Very effective
5 Not effective
9 Don't know

However, the column in the matrix is encoded as:

-1 blank
0 Not effective
1 Don't know or Too soon to tell
2 Mildly effective
3 Moderately effective
4 Very effective

Note that the answer choices have been regrouped and renumbered from the original survey description.

A.3. **Location of Tick Bite Unknown.** The question TK-BITE-LOC-R1 asks "Where in the United States your most recent tick bite occurred." The dictionary.txt file records only one answer choice to this question, "1 I don't know the location where I attained the tick bite." The Unwell baseline dataset embeds this information using 0 and 1, while the Well baseline dataset embeds this information using 1 and 2. It is ambiguous as to which answer choice each number corresponds to. Many of our experiments found this question to be significant in classifying Well and Unwell groups, but much of this significance is likely due to the mismatched embedding rather than the question itself.

APPENDIX B. ADDITIONAL EXPERIMENTS

B.1. **Question codes used for ECOC Classifier.** Table reftab:ECOCinsampleloss records both the in-sample and out-of-sample classification rates for the ECOC experiment on antibiotic data in Section 4.3.2. In sample-error is less conservative than out-of-sample error because it is susceptible to overfitting on noise in the training set. The input questions are listed below.

(1) CTX-ABX-DUR-U1: "I have been on my current antibiotic treatment protocol for ___ months"
(2) CTX-ABX-PULSED-U1: "My current antibiotic protocol is:" (pulsed, not pulsed)
(3) CTX-ABX-SFX-U1: "The level of negative side effects I currently experience using this antibiotic treatment protocol is:" (not present, mild, moderate, severe)
(4) CTX-ABX-ROUTE-U1:"My current antibiotic protocol consists of:" (oral, intramuscular, intravenous antibiotics)
(5) CTX-ABX-ORAL-SPEC-U1: "The oral antibiotic or combination of oral antibiotics I am currently taking is/are:" (list of oral antibiotics)
(6) CTX-ABX-IM-SPEC-U1: "The intramuscular (IM) antibiotic(s) I am currently taking is/are:" (list of IM antibiotics)

Omitted question code	1-(In-Sample Loss)	1-(Out-of-Sample Loss)
Omit none	0.7444	0.7035
Omit (1)	0.6742	0.6530
Omit (2)	0.7428	0.7005
Omit (3)	0.7382	0.6999
Omit (4)	0.7412	0.6912
Omit (5)	0.7249	0.7046
Omit (6)	0.7450	0.7024
Omit (7)	0.7431	0.7013
Omit (8)	0.7363	0.7084
Omit (9)	0.7436	0.7040
Omit (10)	0.7428	0.7048
Omit (11)	0.7330	0.7010

Table 17 ECOC Classifier Accuracy The input data consist of the columns for the question codes listed in Appendix B.1. One question (which may correspond to multiple columns) is left out each time to find which is most significant in classifying based on both in-sample and out-of-sample loss. The experiment classifies based on the patient's self-report of the effects of the antibiotic (CTX-ABX-ME-U1). The classifier is an error-correcting output code classifier using binary SVM learners.

(7) CTX-ABX-IV-SPEC-U1: "The intravenous (IV) antibiotic(s) I am currently taking is/are:" (list of IV antibiotics)
(8) CTX-MEDS-O-U1: "I am currently using the medications listed below:" (list of other sleep, steroid, stomach etc. medications)
(9) DX-COIN-B1: "I have been diagnosed with a tick- borne co- infection:" (yes, no, don't know)
(10) SX-EARLY-SPEC-B1: "More specifically, I experienced the following symptoms:" (rash, flu-like, headaches, facial nerve palsy, headaches, joint pain, etc.)
(11) CSX-3-WRST-U1: "Currently, my three worst symptoms from Lyme disease are:" (fatigue, headache, heart-related, etc.)

B.2. **Feature Importance for Well and Unwell.** We used SVM with a Gaussian kernel and selected baseline questions as input in order to classify between Well and Unwell patients. We achieved 0.9920 accuracy (evaluated using 10-fold cross-validation) with the columns related to highest level of education, time the tick was attached, and the age the patient first experienced symptoms. With such a high accuracy and only a few questions selected, these questions merit further investigation as important features for classifying Well and Unwell. An additional set of columns to investigate includes whether the patient knew the location of the tick bite, the time the tick was attached, and the age the patient first experienced symptoms, with 0.9474. See the note on the question related to location of tick bite unknown in Section A.

(Eric(Kaiyuan) Chen) UNIVERSITY OF CALIFORNIA, LOS ANGELES
E-mail address: chenkaiyuan@ucla.edu

(Rong Huang) UNIVERSITY OF CALIFORNIA, LOS ANGELES
E-mail address: rosehuang1230@gmail.com

(Diyi Liu) SHANGHAI JIAO TONG UNIVERSITY
E-mail address: supper.d@sjtu.edu.cn

(Catherine Wahlenmayer) GANNON UNIVERSITY
E-mail address: wahlenma001@knights.gannon.edu

(Ada (Jiewen) Wang) UNIVERSITY OF CALIFORNIA, LOS ANGELES
E-mail address: adawang10@g.ucla.edu