

# Multitask Probabilistic Modelling in Healthcare

Andrei Perov, 3041544

Ilkin Bakhtiarov, 3041388

Moritz Kath, 3041550



**LEUPHANA**  
UNIVERSITÄT LÜNEBURG

Faculty of Business and Economics

M.Sc. Management and Data Science

Probabilistic Modelling, Summer semester 2021

<b>Introduction</b>	<b>3</b>
<b>Base Paper Summary</b>	<b>5</b>
Multitask Learning	5
Bayesian Multi-Task Learning	6
Study Design of the Base Paper	9
<b>Our Study Design and Modelling Approach</b>	<b>11</b>
Dataset Description	11
Variable Selection	12
Simulation of Data	13
BMTL in pyMC3	14
<b>Results of Our Implementation</b>	<b>17</b>
BMTL results	17
Comparison of the models	22
Description of the Single- & Multi-task Logistic Regressions	22
Description of Neural Network Model	23
Comparison	23
<b>Conclusion &amp; Outlook</b>	<b>24</b>
Discussion of the Results	24
Limitations of our implementation	25
Outlook	26
<b>References</b>	<b>27</b>
<b>Appendices</b>	<b>29</b>
Correlation Structure across Tasks in BMTL	29
Plate Diagram of BMTL for Logistic Regression Model	30
Experiment Design	31
Selecting Predictors with Logistic Regression	32
Declaration of Authorship and Team Contribution Chart	33

# 1. Introduction

With a growing and aging global population, chronic diseases will become more prevalent in all countries in the future. In the United States alone, almost half the population lives with a chronic condition, a share projected to grow (Lin et al., 2017). In addition to the high prevalence, the cost associated with the care for these chronic conditions is estimated to be the significant part of all healthcare expenses, alone in the United States exceeding 80% of all costs (Gerteis & Izrael, 2014, as cited in Lin et al., 2017). The rising adoption of electronic patient records and data availability, in general, enables the application of various statistical and machine learning techniques, which are expected to result in preventive and personalized actions to improve individual patients' well-being and reduce healthcare costs (Agarwal & Gao, 2010, as cited in Lin et al., 2017).

Many of these chronic health conditions like diabetes or chronic kidney diseases can result in myocardial infarction, which is linked to high mortality. The conditions that lead to myocardial infarctions will be examined in more detail in the predictive model of this paper. Specifically, the consequences of myocardial infarctions will be explained in a Bayesian Multitask Learning (BMTL) modeling approach. This approach could improve healthcare providers' ability to identify at-risk patients and implement preventive measures before adverse health effects occur.

Especially the dynamic and multifaceted nature of the medical field makes the choice of multitask learning appropriate for our modeling decision. Multitask learning

leverages the idea that predictors can be correlated with each other when predicting interrelated learning outcomes. Instead of focussing on modeling one specific event, we try to model different outcomes in one model. For example, patients with chronic diseases are often at risk for many adverse health effects such as stroke, renal failure, or myocardial infarction. Patients with myocardial infarctions, on the other hand, might face different effects following the infarction event. The information to predict one of these outcomes might be related to the other outcomes, thus resulting in an improved predictive performance for each predicted event when using a unified multitask framework.

The remainder of this paper is structured into three parts. In the first part, we will summarize Lin et al.'s 2017 paper *Healthcare Predictive Analytics for Risk Profiling in Chronic Care: A Bayesian Multitask Learning Approach* (hereinafter referred to as the *base paper*) and describe the BMTL model used in their approach in detail.

In the second part, we will apply the BMTL framework to our own dataset, describing the data and our variable selection and data simulation procedure. In this chapter, we will explain the adaptation of the model of the base paper to our dataset. In the third chapter, we will compare and discuss the results of our implementation. Different approaches to multitask learning (MTL) will be compared and related to the findings of the base paper. In the final chapter, we summarize the results and discuss the limitations of our approach and give an outlook on possible improvements and further research in the field.

## 2. Base Paper Summary

### a. Multitask Learning

Multitask Learning (MTL) is the main component of our base paper and our work. With this approach, related (classification) tasks are trained jointly to improve each task's predictive outcome. Instead of building separate models for related outcomes, we expect some synergy effects between the predictors for these related tasks, which can be taken advantage of through a unified modeling framework (Lin et al., 2017).

The application MTL is particularly meaningful in fields where relationships between features and outcomes are not immediately apparent but more subtle and where hidden connections might exist (Bakker and Heskes, 2005, as cited in Lin et al., 2017). This might especially be true in sciences such as biology, medicine or genetics, and in various other fields. In our literature research, we found three general approaches to MTL:

1. Neural networks: sharing of common hidden layers (Bakker & Heskes, 2003, as cited in Lin et al., 2017)
2. Optimizing common regularizer function (Huang et al., 2012, as cited in Lin et al., 2017)
3. Bayesian approach: Common Prior Distribution (Archambeau et al., 2011, as cited in Lin et al., 2017)

Each one of these approaches has their own characteristics and can prove to be relevant in different scenarios. The (1) neural network approach requires an Artificial Neural

Network as a baseline model, and the (2) common regularizer function only works for positively correlated learning tasks to prevent negative transfer (Lin et al., 2017, p. 477).

The (3) Bayesian approach is more flexible because any model or algorithm can also be represented in a Bayesian formulation but requires more computational power for sampling the data or simulation to fit the model (Chipman et al., 2002, as cited in Lin et al., 2017).

For our problem, we will mainly focus on implementing a Bayesian Multi-Task Learning (BMTL) approach. Furthermore, we also implement the other MTL types to obtain comparative results.

## b. Bayesian Multi-Task Learning

Bayesian Multi-Task Learning (BMTL) model used by Lin et al. (2017) is a stack of logistic regression models with different adverse health events set as target variables with the same descriptive variables. The authors' main contribution to the literature is that they "enabled multitask learning by imposing a hierarchical correlation structure as a channel to transfer information over tasks" in BMTL (Lin et al., 2017, p 482.)

The model is built based on simple logistic regression models commonly used for classification tasks, where there are only two possible outcomes - the event has or has not occurred. This can be denoted as  $y_i^{(k)} \in \{0, 1\}$  and  $y_i^{(k)} \sim \text{Bernoulli}(\theta_i^{(k)})$ . Compact mathematical representation of the logistic regressions for  $N$  observations,  $J$  variables, and  $K$  tasks (events) can be written as:

$$\text{logit}(\theta_i^{(k)}) = \alpha^k + \sum_{j=1}^J \beta_j^{(k)} x_i, \quad k = 1 \dots K, j = 1 \dots J, i = 1 \dots N,$$

Where  $\theta_i^{(k)}$  is the probability of event  $k$  given  $x_i$  observation. and  $\alpha^{(k)}$  and  $\beta_j^{(k)}$  are event-specific intercepts and coefficients. The occurrence of events can be extracted with the Logit function, i.e.  $\text{logit}(\theta) = \log(\theta/(1 - \theta))$ .

As stated earlier, Lin et al. were able to implement the baseline logistic regression in BMTL by building a hierarchical structure across different tasks. They achieved this by deriving the correlations of the regression coefficients across all learning tasks (see *Appendix A* for the visual representation). The paper includes the representation of the correlation structure as matrices, where each predictor requires its own correlation matrix. In doing so, the authors could achieve regression coefficients ( $\beta_j^{(k)}$ ) each with a corresponding matrix that involves information across all the tasks and can unanimously contribute to all classification tasks.

Building on top of the base logistic regression model, Lin et al. structured their BMTL model with a set of prior distributions. The summary of their model and model parameters can be given as:

The probability of the adverse health event occurrence ( $\theta$ ), as mathematically represented above, is determined by the intercept ( $\alpha$ ) and coefficients ( $\beta$ ) in a set of  $K$  logistic regression models. A plate diagram of the BMTL logistic regression model to be described below is provided in *Appendix B*. Given the model, the following assumptions are made about the parameters:

(i) The regression coefficients for the  $j^{th}$  predictor across tasks (logistic regressions) follow a multivariate normal (MVN) distribution with zero means and scaled covariance matrix:

$$\beta_j = [\beta_j^{(1)}, \beta_j^{(2)}, \dots, \beta_j^{(K)}]^T,$$

$$\beta_j \sim \text{MVN}(0, r_j^2 \Sigma_j)$$

Where  $r_j^2 \Sigma_j$  is the scaled covariance matrix;

(ii)  $r_j$  is the shrinkage scalar (one for each predictor), that is also known as Horseshoe prior and is used as a shrinkage coefficient for the covariance matrix ( $\Sigma_j$ ) in the MVN distribution of the  $j^{th}$  covariate:

$$r_j = \tau_j \psi,$$

$$\tau_j, \psi \sim \text{Half-Cauchy}(0, 1)$$

Where both multiplicative components of the scalar ( $\tau$  and  $\psi$ ) follow Half-Cauchy distribution with mean zero and scale parameter 1;

(iii)  $\Sigma_j$  is the covariance matrix in the MVN distribution of the  $j^{th}$  covariate, and it is parameterized as:

$$\Sigma_j = \text{diag}(\sigma_j) * \Omega_j * \text{diag}(\sigma_j),$$

$$\sigma_j = [\sigma_j^{(1)}, \sigma_j^{(2)}, \dots, \sigma_j^{(K)}]^T$$



Where  $\sigma_j$  is a vector of standard deviations in which  $\sigma_j^{(K)}$  is the standard deviation of  $\beta_j^{(K)}$  (the coefficient for predictor  $j$  in task  $k$ ); and, the  $diag(\sigma_j)$  is a diagonal matrix with elements of  $\sigma_j$  on the diagonal.

(iv)  $\sigma_j^{(k)}$  is the standard deviation of the  $j^{th}$  coefficient in the  $k^{th}$  logistic regression, and it follows Half-Cauchy distribution with mean zero and the scale parameter is 2.5:

$$\sigma_j^{(k)} \sim \text{Half-Cauchy}(0, 2.5)$$

The coefficients of the given feature for all logistic regressions belong to the same multivariate distribution and share information through covariance matrix.

(v)  $\Omega_j$  is the correlation matrix for each predictor of the  $j^{th}$  coefficient across tasks, and it follows the Lewandowski, Kurowicka, and Joe (LKJ) distribution:

$$\Omega_j = \text{LKJ}(K, 1)$$

Where the first parameter of the distribution specifies the dimension of the desired correlation and, therefore, equals to  $K$  – the number of tasks in Lin et al.'s study. The second parameter controls the degree to which the correlation matrix shrinks toward the identity matrix and is set to be 1 to reflect no prior information on  $\Omega_j$ .

Given the hierarchical structure of the model with complex prior distributions for a large set of parameters of the model to estimate and high correlation between the features, the authors chose No-U-Turn Sampler (NUTS) to fit the model to data. NUTS is a variant of the Hamiltonian MCMC sampler. This type of MCMC samplers suppresses local

random walk and explores the marginal variances of the probability space (Lin et al., 2017).

### c. Study Design of the Base Paper

Lin et al. have designed their study based on the Electronic Health Record (EHR) dataset on patients diagnosed with type 2 diabetes collected from a Taiwanese hospital between 2003 and 2012. The EHR data included over fourteen thousand (14,000) observations per hundred and seventy-nine (179) explanatory variables, with patient-specific information ranging from demographics to labs and exams, diagnoses, and treatments. The target variables of interest were three adverse effects – stroke, myocardial infarction, and acute renal failure.

The authors describe their study design in three steps. In the first step, they have divided each patient's medical history in half and randomly sampled a visit ( $v_{0i}$ ) from the first half of the patient's records. The visual representation of the steps is presented in *Appendix C*. In the second step, they used the information available prior to the visit for training the BMTL. In the third step, they divided the information available after the visit into five windows of time (from a year to five years) to see whether the patient had experienced any of the three reverse health effects in one of the windows. The authors could evaluate the performance of their model by comparing the predictions made by the BMTL model to the actual occurrence of the events in the right window of time and calculating the area under the receiver operating characteristics curve (also known as C-statistic). Furthermore, Lin et al. compared the performance of the BMTL model to the

performance of several other machine learning models commonly used for classification tasks – Bayesian Logit, Logit, Logit with Lasso regularizer, MTL-Logit, MTL-Tree, and MTL-ANN – and concluded that their model outperformed all the other models tested on the same data.

### 3. Our Study Design and Modelling Approach

#### a. Dataset Description

We applied the BMTL modeling technique from the base paper of Lin et al. to another dataset (described below) with a medical topic. The original dataset is not available because of privacy reasons. The chosen dataset is called Trajectories, bifurcations, and pseudo-time in large clinical datasets: applications to myocardial infarction and diabetes data (Golovenkin, S.E. et al., 2020) and contains information on complications of myocardial infarctions in admitted hospital patients. Myocardial infarction is a common symptom of various underlying medical conditions and can lead to various adverse health effects. Attributed to modern lifestyles, the prevalence of this condition is predicted to rise in the future. Therefore, the accurate prediction of its occurrence is necessary to take preventive steps promptly.

The data was collected in the early 90s in the Krasnoyarsk Interdistrict Clinical Hospital in Russia. With 1700 observations, the dataset is smaller than that of the base paper. Due to computational constraints, we selected a subset from the total of a hundred and twelve (112) predictor variables and twelve (12) target variables for our analysis. We

chose three target variables and a hundred and four (104) predictors for this comparative approach.

The target variables from this dataset are binary variables, thus resembling the structure binary classification targets from the base paper. Unfortunately, the targets are imbalanced, which can affect model performance negatively. Generally, the dataset is sparse, with most predictors being of boolean type and value of zero. Specifically, the small number of observations proved to be detrimental to predictive performance. This is why we chose to use a simulation technique described in Chapter 3. c. to overcome some of the aforementioned downsides of our base dataset.

## b. Variable Selection

Due to computational limitations, we had to select a subset of the provided variables and targets of the chosen dataset to render our BMTL model in due time. We analyzed all 12 provided target variables to select appropriate target variables and selected the top 3 least unbalanced targets. The three selected target variables are the following:

1. LET\_IS: Lethal Outcome of Myocardial Infarction
2. ZSN: Chronic Heart Failure
3. FIBR\_PREDS: Atrial Fibrillation

For the predictors, we first excluded the observations collected only over the span of three days after a patient's admission to the hospital. These observations only included the patients' reported pain complaints and painkiller intake, and they were adding

complexity into the model without a potential contribution to performance as they were collected for three days in a row and were small in size. Furthermore, we excluded variables with more than 500 missing values among all observations (1700 observations).

Then we used the multilabel logistic regression model from the Scikit-Learn (Cournapeau et al., 2011) package (a technique described in Chapter 4.b.i) with the remaining 104 predictors to further select the variables with higher weights in the learning task. In addition, we applied L2 regularization (L1 is not supported for ChainClassifier in Scikit-Learn) to this model, and we chose the predictors with the highest corresponding coefficients and with the better predictive power for the three before selected target variables (Fan, J., & Li, R., 2020). In Appendix D, we provide the line plots of the regression coefficients per each target variable. It can be seen from the figures that the values of some of the coefficients peak for different target variables. We have captured the values with the highest weight in predicting each of the dependent variables and, eventually, were able to extract 19 of the variables.

Furthermore, in order to decrease computation time, we applied grid search to extract seven final variables. We have used these variables in the final BMTL model and compared the results to check their performances.

### c. Simulation of Data

Using the nineteen selected variables, we have simulated four thousand (4,000) observations per predictor variable that follow the same distribution as the original

dataset. The primary purpose of this step was to create a large enough dataset so our BMTL model could fit better and produce interpretable results, but also not too big to have manageable computation time.

In order to simulate the dependent variables, however, we have used the regression intercept and coefficients generated by the single-task logistic regression models (will be discussed in Chapter 4.b.i) applied per each of the three target variables in the original dataset. Then we used the logistic regression model to generate four thousand (4,000) outcomes for each of the three dependent variables and recorded them as simulated target variables. The generated values were saved as the simulated dataset to be applied in our BMTL model and compare its performance against the other models as described in detail in Chapter 4.

#### d. BMTL in pyMC3

Choosing the framework, we had several criteria: simple and efficient implementation in python, significantly big and helpful community, support of the NUTS (no U-turn sampler) MCMC sampling. Given our requirements, PyMC3 (Salvatier et al., 2016) was chosen as the best compromise.

During the implementation of our model, we mainly followed the original description from the base paper (Lin et al., 2017), but we faced some problems with pyMC3 that forced us to make corrections in our modeling.

For example, implementation of covariance matrix in the base paper (Lin et al., 2017) is described by the equation  $\Sigma_j = \text{diag}(\sigma_j) \Omega_j \text{diag}(\sigma_j)$ , but in PyMC3 such calculation of covariance matrix leads to a lot of divergences through MCMC sampling (3500 out of 8000 samples). We used a better alternative - Cholesky decomposition of the covariance matrix that performs without additional divergences (Rochford, A., nd).

The second challenge was connected with multiple covariance matrices, one for each regression coefficient. In pyMC multivariate distributions of such regression coefficients can not be represented by one function (Lao, H., 2020). Finally, to create a model that takes into account the correlations among coefficients of different tasks, we used a computationally inefficient way. We loop over each predictor to get a regularizer and a covariance matrix for multivariate normal distribution of corresponding regression coefficient. Then, we aggregate all derived distributions into a mixture distribution. From this mixture distribution through MCMC sampling, we get optimal regression coefficients.

After getting 19 meaningful features out of 104, the first step of modeling was implementing the single task Bayesian logistic regression to get convergence on a simple well-explored dataset. The model performed well on the simple dataset (AUC 0.759), but AUC was between 0.5 and 0.6 for each of the target variables with our chosen dataset.

The next step was to implement a simple version of the BMTL model without a horseshoe regularizer and with only one covariance matrix for multivariate normal distribution of the regression coefficients. We chose to implement this model first to try to create a multitask setting with manageable sampling time. Performance still was low on

the chosen dataset. At the same time, the model performed well (AUC 0.9) on an artificial multilabel dataset from the Scikit-Learn package.

Next, we implemented multi-task Bayesian logistic regression, where each regression coefficient was sampled from a multivariate normal distribution with an individual covariance matrix scaled by a unique regularization parameter. This model was very close to the model presented in the paper (Lin et al., 2017).

For the evaluation of the model, we drew 1500 samples of the targets' predictions (each time for the whole test set) and took a mode for each observation. The problem with such a setting is that MCMC sampling takes too much time going through the 'for' loop. In each iteration, the NUTS tries to fit many parameters into the dataset, and the deep hierarchy of the BMTL model makes it computationally inefficient. It takes up to 15 hours to run the model with 19 variables, 2000 tuning, and 4000 targeted MCMC samples. Because of this reason, we decided to change the set of features from 19 variables to 7 variables with a 6.5 percent loss of AUC for Multilabel Lasso regression from the Scikit-Learn package. This transition allowed us to lower the run time below 4 hours. Google Collab showed the lowest run time compared to a powerful computer (i9, 3.2 GHz). The number of MCMC samples was estimated empirically, given that further increasing of these numbers (2000 and 4000) does not increase the classification power of the model.

We discovered that the increased complexity of the model with multiple covariance matrices for our chosen dataset does not lead to better classification (higher AUC) compared to the simplified BMTL model version with only one covariance matrix.



The final hypothesis was that the BMTL model needs more data points to learn given imbalanced target variables, and, therefore, we simulated the chosen dataset with 4000 observations. In this setting, the computation time for the final BMTL model increased two times. Overall, the classification quality had improved, but the BMTL model still performed on the same level as other multitask learning models.

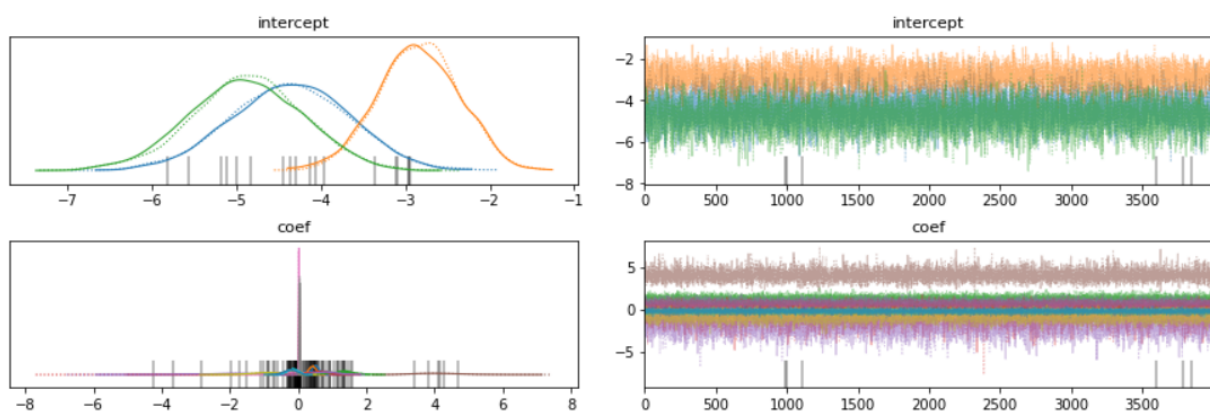
## 4. Results of Our Implementation

### a. BMTL results

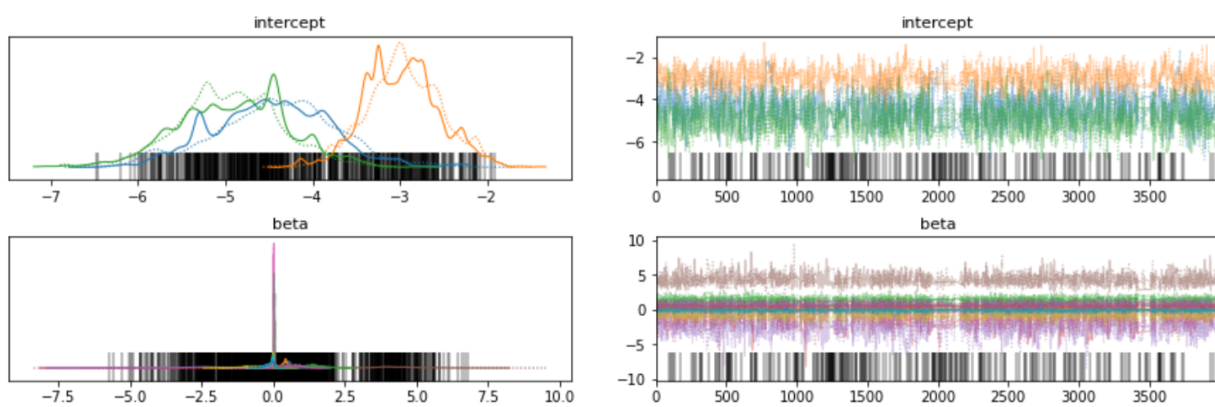
To evaluate the performance of our models, we used AUC scores, the same as the authors of the base paper. However, we have also used *recall* (the ratio of true positives to all positives in the test set) and *precision* (the ratio of true positives to all predicted positives). It is crucial to get high recall because the aim of healthcare predictions is to detect the development of adverse or lethal health conditions and take early prophylactic measures. Therefore the model should be trained to catch positive cases (recall is close to 1) and simultaneously control the number of false positives (precision as close to 1 as possible) to avoid unnecessary interventions for patients who do not need it.

Below we present two figures from PyMC3 with the performance of BMTL models with one covariance matrix for all predictors (BMTL-simple) and the BMTL model with individual regularizer and covariance matrix for each predictor (BMTL-full). MCMC sampling includes four thousand (4,000) targeted samples, with two thousand (2,000) tuning (burn-in) samples.

**Figure 1. Performance of the BMTL-simple model (a single covariance matrix), real data**



**Figure 2. Performance of the BMTL-full model (7 covariance matrices), real data**

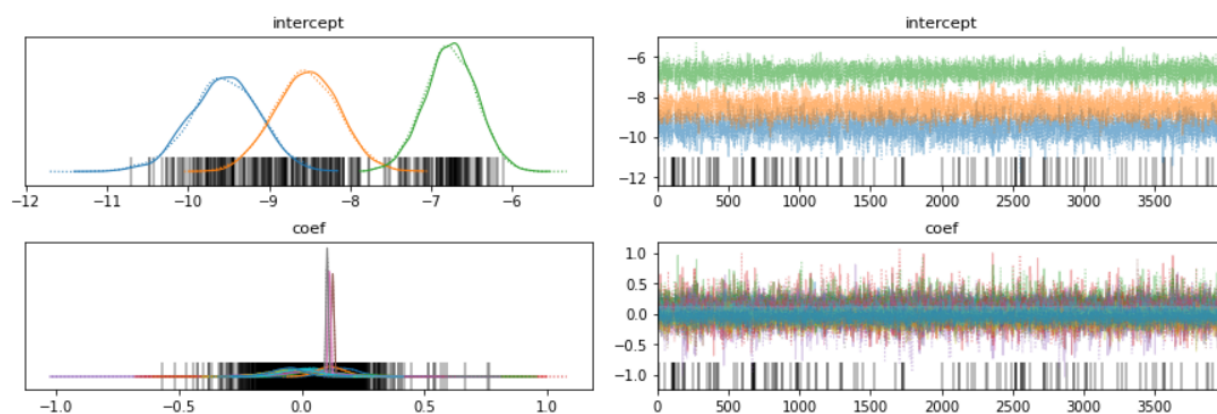


The first two figures above reflect the fit of the models to the real data (1663 observations) with seven ( 7 ) predictive variables. We can see that the shapes of the

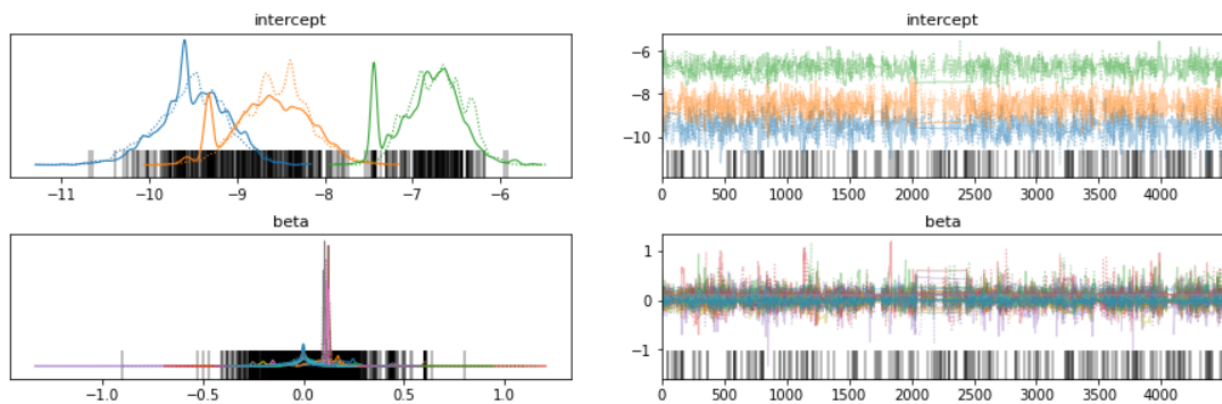
parameter distributions for BMTL-full and BMTL-simple are different, and mean values are close to each other. Even if the posteriors of the complex BMTL models resemble their priors, they look different: skewed and not smooth. That could be explained by a lower acceptance rate – the BMTL-full model has more significant gaps between the observations (right graph). This could be because of the complexity of the model. For each iteration of MCMC sampling, BMTL-full needs to take care of additional  $\sigma$ ,  $\tau$ , and  $\beta$  (see Chapter 2.B for parameter specifications) with one less than the equal number of variables.

The following two graphical results (Figure 3 and Figure 4) reflect the fit of the models to the simulated data (4000 observations) based on the real data with nineteen predictive variables. For the analysis, we used seven previously selected variables.

**Figure 3. Performance of the BMTL-simple model (7 covariance matrices), simulated data**



**Figure 4. Performance of the BMTL-full model (7 covariance matrices), simulated data**



For the simulated data, distributions of the parameters of the BMTL-full model have spikes and higher variances corresponding to the distribution of the BMTL-simple model. Overall, patterns resemble the ones in real data: smooth and symmetric distributions for the BMTL-simple model and disrupted for the BMTL-full model. It illustrates that even the size of the dataset does not influence the acceptance rate of MCMC sampling.

Sampling chains (dashed and continuous lines of the same color on graphs) for the individual parameters seem well converged and stationary for all mentioned cases, besides different peaks for BMTL-full models.

Next, we will look at the performance of the four models: BMTL-full, BMTL-simple, multi-task logistic regression (MTLR), and neural network. As provided in Table 1, the performance of the models on the chosen dataset is relatively poor to make a meaningful comparison of the models. That is why we present only the AUC scores to give an understanding of the scales. The number of variables for the chosen dataset can be

significantly reduced. Still, even for a set of 104 variables, the classification results are not good – the average AUC score is lower than 0.6.

<b>Table 1. Evaluation of models on real data (1663 observations, 7 predictors)</b>					
Metric	Target variable	Models			
		BMTL-full	BMTL-simple	MTLR	NN
AUC	FIBR_PRED	0.526	0.524	0.518	0.532
	ZSN	0.541	0.537	0.528	0.615
	LET_IS	0.612	0.605	0.580	0.590

To evaluate the BMTL models, we used the simulated data with variables distributed as in the chosen dataset (as described in Chapter 3.c). As provided in Table 2, with the simulated dataset, all models perform better. We sacrificed the accuracy for a smaller set of variables to decrease computational time. However, such results still allow us to compare the performances of different multitask learning algorithms.

<b>Table 2. Evaluation of models on simulated data (4000 observations, 7 predictors)</b>					
Metric	Target variable	Models			
		BMTL-full	BMTL-simple	MTLR	NN
AUC	FIBR_PRED	0.648	0.662	0.665	0.654
	ZSN	0.620	0.616	0.615	0.603
	LET_IS	0.700	0.705	0.699	0.694
Recall	FIBR_PRED	0.355	0.384	0.401	0.379
	ZSN	0.306	0.298	0.298	0.256
	LET_IS	0.623	0.630	0.617	0.635
Precision	FIBR_PRED	0.647	0.665	0.633	0.619

	ZSN	0.566	0.564	0.556	0.591
	LET_IS	0.702	0.708	0.705	0.684

Furthermore, we did not find evidence that the more complex BMTL model with multiple regularization scalers and covariance matrices outperforms the simpler version with one covariance matrix used. In the next part of the chapter, we will introduce other models and compare them with the BMTL approach.

## b. Comparison of the models

Mirroring Lin et al.'s 2017 paper, in order to test our BMTL model, we have built several other machine learning models to compare performances. In particular, we have built three Single-task Logistic Regression models, one for each predictor, as well as the two multi-task models to predict all three dependent variables at once: one Multi-task Logistic Regression model and a Neural Network.

### i. Description of the Single- and Multi-task Logistic Regressions

When building the Single-task Logistic Regression models, we used Scikit-Learn's Logistic Regression classifier from the Linear Models module (Cournapeau et al., 2011). Logistic regression, also known as logit regression, maximum-entropy classification, or the log-linear classifier, is a linear model used for classification tasks. The mechanism behind it follows the Logit function described under the simple Logistic Regression classifier in Chapter 2.b. We have used this model for two purposes: first, to extract the intercept and

coefficients for each of the three target variables (LET\_IS, ZSN, FIBR\_PREDS) from the original dataset so we could simulate the target variables as described in Chapter 3.c; second, we have used it on the simulated data to compare its predictive power against the BMTL models and evaluate their performances.

Additionally, we have built a Multi-task Logistic regression to compare its performance against the BMTL model. We used Scikit-Learn's method called ChainClassifier that combines the three single-task logistic classifiers into a single multi-label model capable of exploiting correlations among targets (Cournapeau et al., 2011.)

## ii. Description of Neural Network Model

The modeling approach with a multi-layer Neural Network was realized in the PyTorch library. We used five fully connected layers with BatchNorm regularization between layers and ReLU activations. The hidden layers range between 32 and 1024 neurons and terminate in a final layer with three output neurons. We used 100 training epochs and a train-test split of 70/30. The multitask learning in this neural network is achieved by sharing the hidden layers that supply the final head of the neural network.

Although the neural network has a considerable size and depth, it could not substantially outperform either the logistic regression classifier or the BMTL approach. This could be attributed partly to the small size of our dataset. Neural networks typically need a lot of data for training, and we did not meet this requirement in our approach

(Walczak, 2001). Retraining the model with the simulated dataset with a larger number of observations improved the models' accuracy.

### iii. Comparison

According to Table 2, there is no significant advantage of the BMTL approach. At least we could not identify it through our studies. For our settings, the amount of data does not increase the difference in performance between BMTL models and the other two models. The base paper (Lin et al., 2017) states that BMTL performs better than other multi-task models given imbalanced targets. However, we could not draw the same conclusion based on our findings.

Based on recall and precision metrics to evaluate healthcare-specific performance, we found that the BMTL approach works as well with multilabel neural networks and multi-task logistic regression. For the seven variables and four thousand observations settings, the computational time for sampling the BMTL-full model is about four hours, for BMTL-simple - about fifteen minutes. The evaluation, however, takes two minutes. In contrast, it takes less than three minutes to train for multi-task logistic regression and neural networks and several seconds to evaluate.

As an advantage, Bayesian models bring confidence in the results and provide information about training processes that can be used for fine-tuning the model and smart reparametrization.



## 5. Conclusion & Outlook

### a. Discussion of the Results

Overall, we could not entirely reproduce the findings of the base paper. In the base paper, the BMTL approach could significantly outperform single task learning models other MTL models such as logistic regression or multi-head neural networks. In our case, the performance of the BMTL model is comparable. Even if some targets are predicted with a slightly better fit, we cannot conclude that BMTL models have significant advantages over neural networks or multitask logistic regression.

Looking at the computational complexity of BMTL models, we find that the long computing time of MCMC sampling significantly increases the overall training time of probabilistic modeling. For our chosen dataset, the BMTL model does not reward the corresponding increase in predictive performance considering these computational expenses. With different datasets, the approach could prove to be beneficial. Reasons for this contrary outcome will be discussed in the following chapter.

### b. Limitations of our implementation

The main limitation in our implementation was the small dataset with many missing values and hidden flaws. It has too few observations for such a sparse structure, so it is hard to train models in a meaningful way. Simulation helped increase the amount of data, but a larger base dataset would have been an advantage.

Additionally, the data set proved to be quite imbalanced between the predicted classes, and also, the predictors were relatively sparsely populated, which resulted in even more unsatisfactory performance. It could also be that the target variables that we chose (the only ones with a manageable balance of classes) were not the best candidates to apply BMTL models.

Using the PyMC3 (Salvatier et al., 2016) as the framework for Bayesian sampling might not have been the best choice for the specific model introduced in the paper Lin et al., 2017. The package does not natively support some of the necessary features, such as covariance matrix for each predictor represented in one function. We suspect that the other frameworks primarily aimed at Bayesian Frameworks, such as STAN (Stan Development Team, 2020) , could be more appropriate to deal with these modeling approaches.

### c. Outlook

While the motivation for multitask learning and predictive modeling for healthcare analytics remains a valid and promising endeavor, procurement of a suitable dataset for academic research turned out to be difficult. The procurement of large patient health records data should be pursued to facilitate research in this domain. Nevertheless, privacy concerns limit the possibilities in this regard.

Different modeling approaches should be considered and compared to BMTL and other approaches. Due to the sequential nature of patients' medical history, multiheaded recurrent neural networks should be considered.

In either way, the immense rewards for predicting and preventing adverse health outcomes for individuals warrants further research in this field.

## References

Agarwal, R., & Gao, G. (2010). The Digital Transformation of Healthcare: Current Status and the Road Ahead. *Information Systems Research*, 21(4), 796-809.

Archambeau, C., Guo, S., & Zoeter, O. (2011). Sparse Bayesian Multi-Task Learning. *Advances in Neural Information Processing Systems*, 45(1), pp. 1755-1763.

Bakker, B., & Heskes, T. (2005). Task Clustering and Gating for Bayesian Multitask Learning. *Journal of Machine Learning Research*, 4(1), 83-99.

Chipman, H. A., George, E. I., & McCulloch, R. E. (2002). Bayesian Treed Models. *Machine Learning*, 48(1-3), 299-320.

Fan, J., & Li, R. (2020). Comment: Feature Screening and Variable Selection via Iterative Ridge Regression. *Technometrics: a journal of statistics for the physical, chemical, and engineering sciences*, 62(4), 434–437. DOI: 10.1080/00401706.2020.1801256

Gerteis, J., & Izrael, D. (2014). Multiple Chronic Conditions Chartbook: 2010 Medical Expenditure Panel Survey Data (14th ed., Vol. 38). AHRQ Publications.

Golovenkin, S.E., Bac, J., Chervov, A., Mirkes, E.M., Orlova, Y.V., Barillot, E., Gorban, A.N. and Zinovyev, A. (2020). Trajectories, bifurcations, and pseudo-time in

large clinical datasets: applications to myocardial infarction and diabetes data.

GigaScience, 9(11), p.giaa128., DOI: 10.1093/gigascience/giaa128

Huang, J., Breheny, P., & Ma, S. (2012). A Selective Review of Group Selection in High-Dimensional Models,. Statistical Science, 27(4), 481-499.

Lao H. (Oct 2020), a PyMC3 developer, replied to Raul's blog question How to build a multi-covariance model in pymc3? Retrieved from: <https://discourse.pymc.io/t/how-to-build-a-multi-covariance-model-in-pymc3/4622>

Lin, Y., Chen, H., Brown, R.A., Li, S., Yang, H. (2017). Healthcare Predictive Analytics for Risk Profiling in Chronic Care: A Bayesian Multitask Learning Approach. MIS Quarterly. Vol. 41 Issue 2, p473-A3. 27p. DOI: 10.25300/MISQ/2017/41.2.07

Pedregosa, F., Varoquaux, G., Gramfort A., et al. (2011), Scikit-learn: Machine Learning in Python, JMLR 12, pp. 2825-2830, retrieved from <http://jmlr.csail.mit.edu/papers/v12/pedregosa11a.html>

Rochford, A. (ND). LKJ Prior for fitting a Multivariate Normal Model. PyMC3 rtd-docs. Retrieved from: [pymc3-testing.readthedocs.io/en/rtd-docs/notebooks/LKJ.html](https://pymc3-testing.readthedocs.io/en/rtd-docs/notebooks/LKJ.html)

Salvatier J, Wiecki TV, Fonnesbeck C. (2016) Probabilistic programming in Python using PyMC3. PeerJ Computer Science 2:e55. DOI: 10.7717/peerj-cs.55

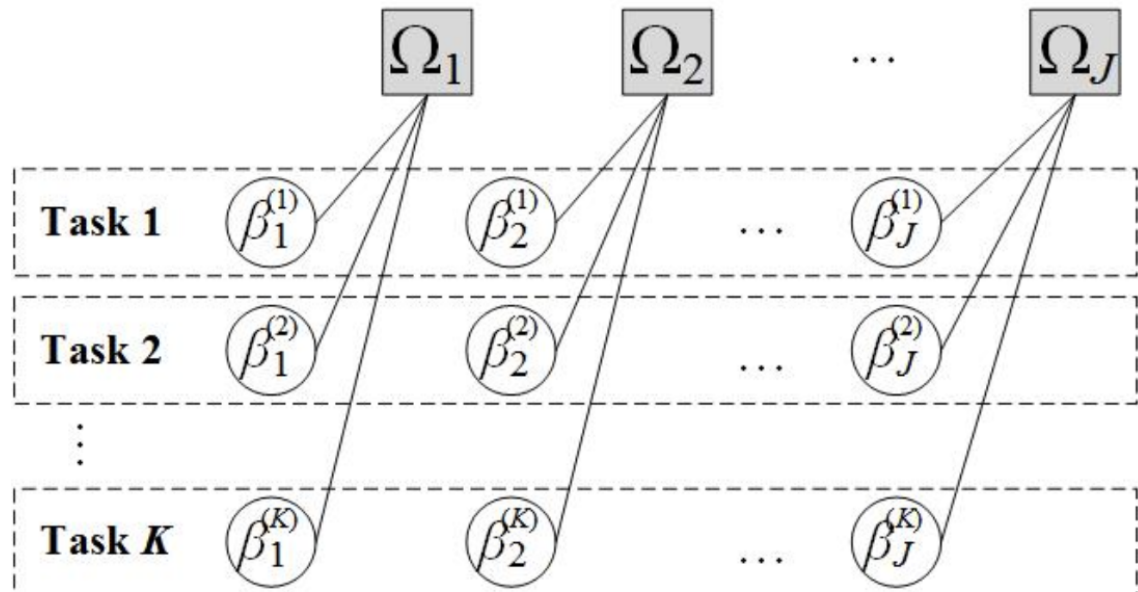
Sharpe, M. & Walczak, Steven. (2001). An Empirical Analysis of Data Requirements for Financial Forecasting with Neural Networks. Journal of Management Information Systems. 17.

Stan Development Team (2020). RStan: the R interface to Stan. R package version 2.21.2. <http://mc-stan.org/>

Walczak, S. (2001). An Empirical Analysis of Data Requirements for Financial Forecasting with Neural Networks. *Journal of Management Information Systems*, 17(4).

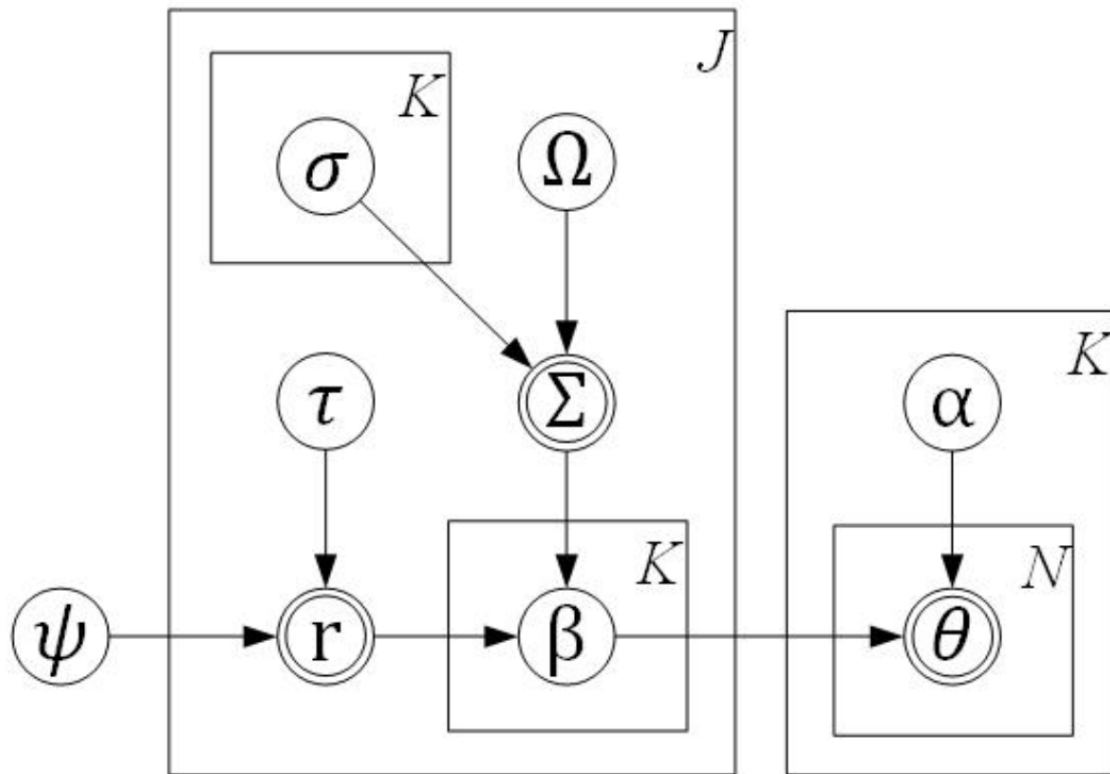
## Appendices

### A. Correlation Structure across Tasks in BMTL



Source: Lin et al., 2017, p. 479.

## B. Plate Diagram of BMTL for Logistic Regression Model



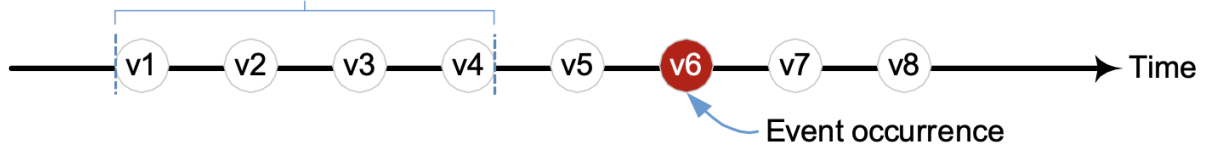
“Consistent with the conventions of a plate diagram, the symbol at the upper-right corner of each plate (rectangle) indicates the number of nodes in the respective plate, and the single- and double-bordered nodes are used to represent stochastic and deterministic (given their parent nodes) parameters, respectively.”

**Source:** Lin et al., 2017, pp. 478-479.

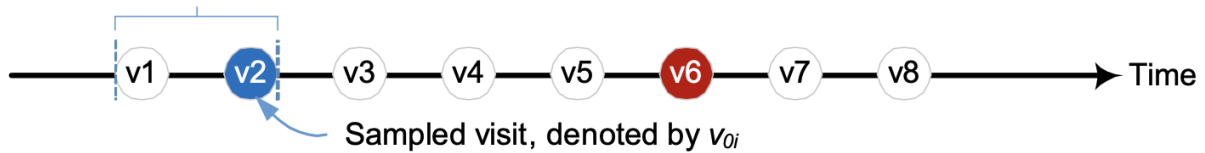


### C. Experiment Design

**Step 1:** Randomly sample a visit from the first half of the patient's medical history .



**Step 2:** Use information available at and before the sampled visit for training.

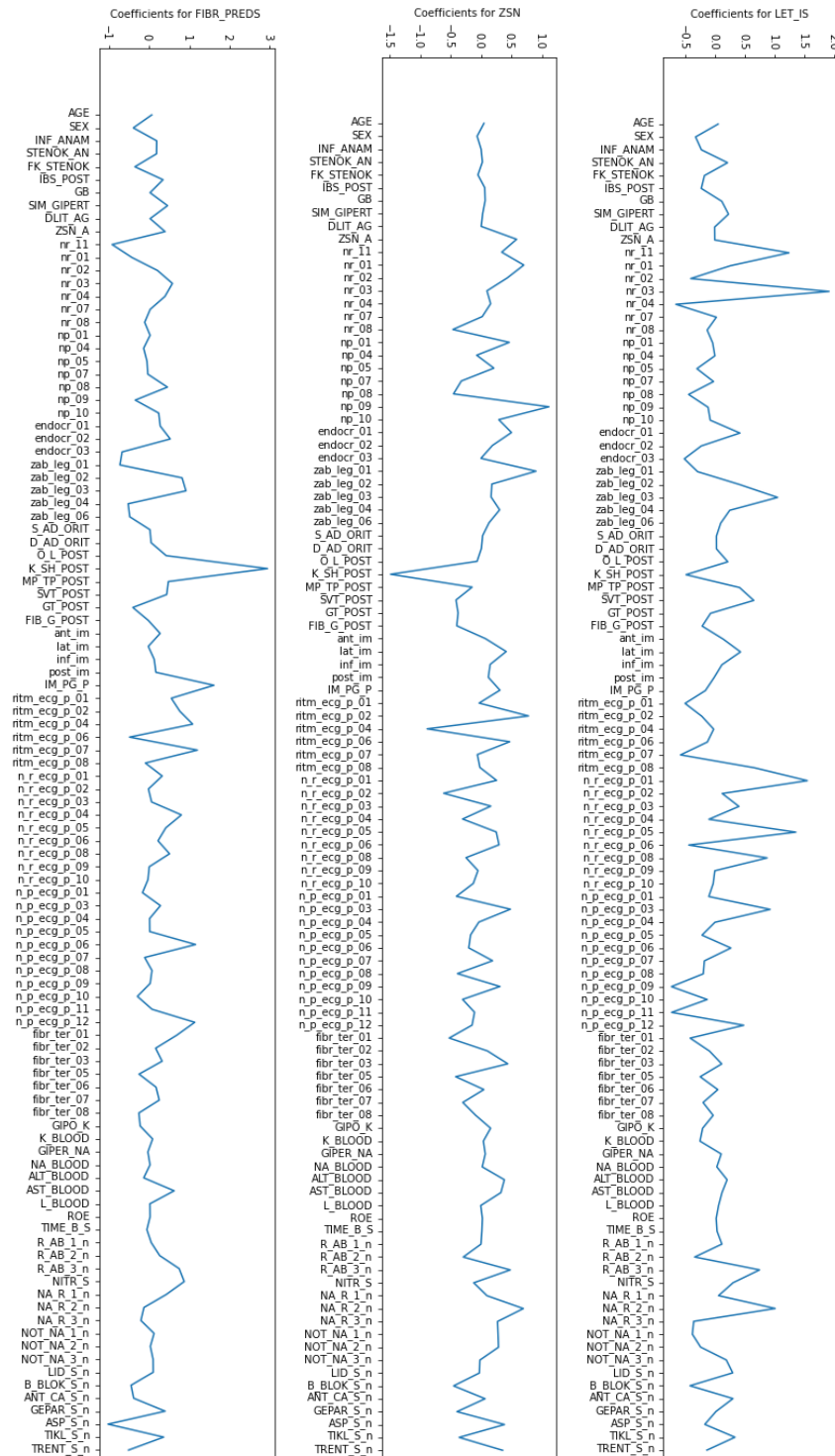


**Step 3:** Predict if an event will happen in the next  $w$  years.



Source: Lin et al., 2017, p. 484.

## D. Selecting Predictors with Logistic Regression (Ridge Regularizer)



## E. Declaration of Authorship and Team Contribution Chart

We hereby declare that,

We have authored this group project independently and that we have not used other than the declared sources/resources.

We have explicitly marked all material which has been quoted either literally or by content from the used sources.

According to our knowledge, the content or parts of this group project have not been presented to any other examination authority and have not been published. The different parts of this group project have been divided among the group members as follows.

<b>1. Project contribution (<i>general</i>)</b>	
Finding the dataset	Moritz
Studies development	Andrei, Ilkin, Moritz
Neural network model development	Moritz
Logistic regressions (single- & multi-task) in Scikit-Learn	Ilkin
BMTL model development	Andrei
Toy datasets simulation (for preliminary test of the functionality of the models)	Moritz
Final dataset simulation (from the original dataset, used for testing performance of the final models)	Ilkin & Andrei
Variable selection	Ilkin & Andrei
Results presentation	Andrei, Ilkin, Moritz (Moritz as responsible person)

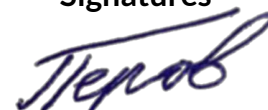
<b>2. Paper contribution (<i>per written chapter</i>)</b>
---

1. Introduction	Moritz
2.a. Multitask Learning	Moritz
2.b. Bayesian Multi-Task Learning	Ilkin & Andrei
2.c. Study Design (the Base Paper)	Ilkin
3.a. Dataset Description	Moritz
3.b. Variable Selection	Ilkin
3.c. Simulation of Data	Ilkin
3.d. BMTL in pyMC3 - Andrei	Andrei
4.a. BMTL results - Andrei	Andrei
4.b.i. Description of the Single - Multi-task Logistic regressions	Ilkin
4.b.ii. Description of Neural Network Model	Moritz
4.b.iii. Comparison	Andrei
5.a. Discussion of the results	Moritz & Andrei
5.b. Limitation of our implementation	Moritz & Andrei
5.c. Outlook	Moritz
Editing of the paper: whole team, responsible editor	Ilkin

**Authors:****Dates****Signatures**

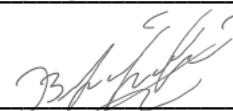
Andrei Perov:

July 30t, 2021



Ilkin Bakhtiarov :

July 30t, 2021



Moritz Kath:

July 30t, 2021

