

## Prediction of the healthcare resource utilization using multi-output regression models

Liwen Cui, Xiaolei Xie, Zuojun Shen, Rui Lu &amp; Haibo Wang

**To cite this article:** Liwen Cui, Xiaolei Xie, Zuojun Shen, Rui Lu & Haibo Wang (2018) Prediction of the healthcare resource utilization using multi-output regression models, IISE Transactions on Healthcare Systems Engineering, 8:4, 291-302, DOI: 10.1080/24725579.2018.1512537

To link to this article: <https://doi.org/10.1080/24725579.2018.1512537>



Published online: 01 Mar 2019.



Submit your article to this journal 



Article views: 654




[View related articles](#) 

View Crossmark data 

Citing articles: 9 View citing articles 



# Prediction of the healthcare resource utilization using multi-output regression models

Liwen Cui<sup>a</sup> , Xiaolei Xie<sup>b</sup>, Zuojun Shen<sup>c</sup>, Rui Lu<sup>d</sup>, and Haibo Wang<sup>e</sup>

<sup>a</sup>Department of Industrial Engineering, Tsinghua University, Beijing, China; <sup>b</sup>Department of Industrial Engineering, Tsinghua University, Beijing, China; <sup>c</sup>Department of Industrial Engineering, Tsinghua University, Beijing, China; <sup>d</sup>Department of Automation, Tsinghua University, Beijing, China; <sup>e</sup>China Standard Medical Information Research Center, Shenzhen, Guangdong, China

## ABSTRACT

With the rapidly increasing healthcare cost and the scarcity of inpatient resources, it is of paramount importance to accurately predict the healthcare resource utilization. Previous research mainly focuses on predicting the healthcare cost using single-output models. However, the intensity of the healthcare resource utilization is reflected by multiple measures. For example, the Diagnosis Related Group (DRG) system adopted in China measures the healthcare resource utilization using both cost and length of stay (LoS), which motivates us to jointly predict these two measures. Compared to constructing several independent single-output models for each task, using multi-output models can provide unified prediction rules, reduce the training time, and improve the generalization by leveraging the correlations across tasks. We utilize four multi-output machine learning models, including the multi-task Lasso, the decision tree, the random forest, and the neural network. We evaluate their performance based on the Electronic Health Record (EHR) dataset with approximately 750,000 records. Based on extensive numerical experiments, we provide a guideline for model selection and construction. This research has the potential to improve the management of healthcare resources and provide decision support for healthcare payment system.

## KEYWORDS

Multi-output regression;  
HER; healthcare  
management;  
machine learning

## 1. Introduction

Higher efficiency in the healthcare system and improved payment policy are strongly desired worldwide. Predictive analytics of the healthcare resource utilization is instrumental for better allocation and management of medical resources. Furthermore, prediction models with high accuracy can facilitate decision making about pricing and reimbursement policy. Recently, there has been growing interest in healthcare cost prediction using machine learning techniques. However, it is insufficient to use cost as the only measure for healthcare resource utilization. For instance, the Diagnosis Related Group (DRG) system, a widely used management tool to group patients with similar consumption of medical resources, uses both cost and length of stay (LoS) to measure the resource utilization by patients. Moreover, the clinical pathway, created to reduce variations in care delivery to improve care quality and efficiency, also focuses on cost and LoS. These motivate us to jointly consider multiple measures in prediction models, depending on the implementation background and practical purposes (Zhou *et al.*, 2011; Lee *et al.*, 2010).

In this article, we use multi-output models to jointly predict cost and LoS. These methods can be easily extended to solve other problems that require learning of several tasks simultaneously. Compared to building two independent single-output models for the prediction of cost and LoS,

respectively, the advantages of using the multi-output model are multifold. First, the multi-output model provides unified and simpler decision rules. For example, in the DRG system, the adoption of two different single-output models provides two sets of grouping rules, which is hard, if not impossible, to implement in practice. Second, the training process of the multi-output model requires fewer computing resources than that of single-output models, since only one model needs to be trained instead of several independent models. Finally, the multi-output model improves predictive generalization. We calculate the Pearson correlation (Benesty *et al.*, 2009) of cost and LoS from the Electronic Health Record (EHR) dataset. The value (0.5) indicates a moderate correlation between these two quantities. With correlated outputs, the multi-output model can often increase the generalization ability by sharing parameters across prediction tasks (Ben-David and Schuller, 2003; Bakker and Heskes, 2003). In the case study, we demonstrate that the prediction accuracy can be improved by upward of 20%.

We use hospital inpatient records to conduct joint prediction. Each record in the EHR dataset contains medical codes, including diagnosis codes and surgical operation codes. The presented challenges are high dimensionality and event sparsity of medical codes, which makes the commonly used, one-hot vector representation inappropriate.

We use five dimensionality reduction methods to generate feature vectors, all of which outperform the one-hot vector representation. With feature vectors generated by dimensionality reduction methods, we train four multi-output models and compare their performance under different conditions. Based on the results of numerical experiments, we provide a guideline for model selection and construction.

Our main contributions are summarized as follows:

1. To the best of our knowledge, we are the first to use multi-output models to jointly predict cost and LoS with practical implication, which can save training time and improve prediction accuracy.
2. We utilize five dimensionality reduction methods to transform medical codes to feature vectors. We provide recommendations on their usage, considering the interpretability, ease of use, and the prediction performance.
3. We utilize four multi-output models for our multi-output regression problems. A comprehensive evaluation of their performance is provided, with varying dataset sizes, feature vectors, and accuracy metrics.
4. Based on numerical experiments, we provide a guideline for prediction model selection and construction, considering the model interpretability, accuracy metrics, dataset size, etc. We believe that the result will benefit researchers and practitioners who work on similar problems.

The rest of this article is structured as follows: [Section 2](#) reviews the related work. [Section 3](#) briefly introduces multi-output models we used in this article. The EHR dataset and the process of data preprocessing are described in [Section 4](#). [Section 5](#) presents the case study. We conclude our work in [Section 6](#).

## 2. Literature review

Health information technology (HIT) has the potential to improve the efficiency and effectiveness of hospital service, which can lead to significant cost savings. Hillestad *et al.* (2005) estimate that the widespread adoption of the EHR system could eventually save more than \$81 billion annually in the U.S. Many countries have focused on accelerating the adoption of HIT. Jha *et al.* (2009) provide a reliable estimate of the adoption of EHR databases in U.S. hospitals. Rea *et al.* (2012) introduce the Strategic Health IT Advanced Research Projects (SHARP) Program, which develops open-source services and components to support the sharing and reuse of operational clinical data stored in EHR from the Mayo Clinic and Intermountain Healthcare. The increasing availability of healthcare records will significantly improve accountability in healthcare at both the population level and individual patient level (Mkanta *et al.*, 2016; Takahashi *et al.*, 2013; Ryu *et al.*, 2015).

In recent years, many researchers have utilized various data-mining techniques to discover information hidden in complex data structures. Murdoch and Detsky (2013) and Nambiar *et al.* (2013) discuss the application of big data to

healthcare, highlighting both opportunities and roadblocks. Wu *et al.* (2010), Jensen *et al.* (2012) and Ross *et al.* (2014) review research topics and data-mining techniques using EHR databases. Bjarndóttir *et al.* (2016) summarize modern applications of insurance claims data in healthcare research. Aylin *et al.* (2007) compare risk prediction models for death in hospitals based on administrative data. Yoo *et al.* (2012) and Tomar and Agarwal (2013) provide a thorough review of the utilization of popular data-mining algorithms in healthcare.

Previous research on the prediction of healthcare resource utilization mainly focuses on single variable prediction. Based on medical and cost data of patients over the first two years, Bertsimas *et al.* (2008) use classification trees and clustering algorithms to predict healthcare costs in the third year. Sushmita *et al.* (2015) investigate the use of regression trees, M5 model trees and random forests to predict the costs of individual patients, given their medical and cost history. Lee *et al.* (2004) predict payments for colorectal cancer patients in a Korean hospital using regression trees and artificial neural networks. As another important component of healthcare resources, LoS has also attracted research attention. Xie *et al.* (2015) and Xie *et al.* (2016) utilize bagged regression trees, along with insurance claim data, to predict LoS in an upcoming year for patients.

Cost and LoS are both important aspects of healthcare resources. In this article, we investigate the use of multi-output regression models for joint prediction. Borchani *et al.* (2015) provide a review of multi-output regression methods. Caruana (1998) describes opportunities for applying multi-task learning in real problems.

## 3. Overview of multi-output regression models

In this article, we utilize four multi-output regression models in machine learning, including the multi-task Lasso, the decision tree, the random forest, and the neural network.

### 3.1. Multi-task Lasso

The multi-task Lasso (Obozinski and Taskar, 2006; Zhou *et al.*, 2011) is a variant of Lasso (Tibshirani, 1996), which is also a linear model that estimates sparse coefficients as Lasso, but requires that the selected features are the same for all the regression tasks. Mathematically, its objective function minimizes the summation of the least squared penalty with a mixed  $l_1/l_2$  penalty added:

$$\min_w \frac{1}{2N} \|WX - Y\|_{Fro}^2 + \alpha \|W\|_{21},$$

where  $N$  is the number of training records, and  $\alpha$  is a constant that multiplies the penalty term. The larger the value of  $\alpha$  is, the sparser the coefficients  $W$  are.  $Fro$  indicates the Frobenius norm:

$$\|A\|_{Fro} = \sqrt{\sum_{ij} a_{ij}^2},$$

and the  $l_1/l_2$  penalty reads:

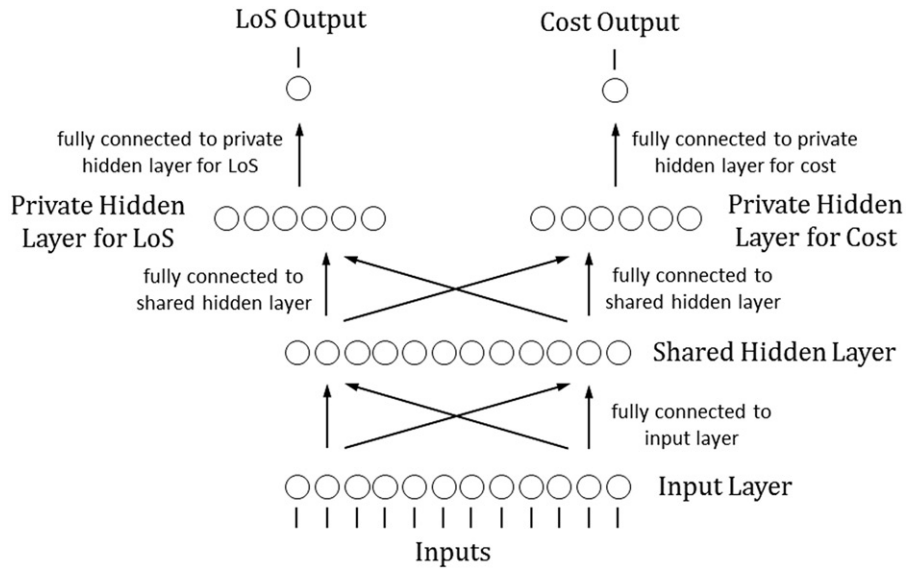


Figure 1. The architecture of the neural network for a multi-output regression problem.

$$\|A\|_{21} = \sum_i \sqrt{\sum_j a_{ij}^2}.$$

We can interpret the results and learn the importance of each feature by observing the coefficients  $W$  of the multi-task Lasso.

### 3.2. Decision tree

The decision tree is a tree-like model that predicts the value of a target variable by learning simple decision rules inferred from the features. This model is widely applied in healthcare data analytics (Bertsimas *et al.*, 2008; Sushmita *et al.*, 2015) due to its simplicity and strong interpretability. In the decision tree, each internal node represents a test on an attribute (e.g., whether or not a patient has a certain disease), each branch represents the outcome of the test, and each leaf node represents a prediction result. The paths from the root node to leaf nodes construct regression rules.

There are various decision tree algorithms (Quinlan, 1986, 1993). We choose Classification and Regression Trees (CART) for our regression tasks because this algorithm supports numerical target variables. CART (Breiman *et al.*, 1984) constructs binary trees using the feature and threshold that yield the largest impurity reduction at each node. The regression criterion of impurity used to determine locations for future splits is Mean Squared Error (MSE). To extend original CART to support multi-output problems, the impurity function of a node should be redefined as the summation of MSE over all responses (De'Ath, 2002). For node  $m$  with  $N_m$  observations, the impurity function is redefined as:

$$H_m = \frac{1}{N_m} \sum_{i \in N_m} \left( y_i^{(1)} - \bar{y}_m^{(1)} \right)^2 + \left( y_i^{(2)} - \bar{y}_m^{(2)} \right)^2,$$

where  $y_i^{(1)}(y_i^{(2)})$  is the value of LoS (cost) for record  $i$  and  $\bar{y}_m^{(1)}(\bar{y}_m^{(2)})$  is the mean of LoS (cost) in node  $m$ .

### 3.3. Random forest

Although the decision tree can provide simple decision rules, its prediction performance can be unstable, because small variations in the data might result in a completely different tree being generated. In addition, practical decision tree learning algorithms are mostly based on the greedy algorithm where locally optimal decisions are made at each node. Such algorithms cannot guarantee to return the globally optimal decision tree. These disadvantages of the decision tree can be mitigated by training multiple trees in an ensemble learner; i.e. the random forest (Liaw and Wiener, 2002), with the cost of increasing the complexity of decision paths generated by aggregating the regression rules of each tree.

We build the random forest based on the multi-output decision tree. In the ensemble, each tree is built using a bootstrap sample of the training set. When splitting a node during the construction of a tree, we pick the split based on a random subset of features. We obtain the final results by taking the average of the responses given by all trees.

### 3.4. Neural network

The neural network (Hagan *et al.*, 1996) can learn complex nonlinear relationships between the feature and the response. This model consists of three types of layers: the input layer, the hidden layer, and the output layer. To solve the multi-output regression problem, we adopt the architecture of neural network shown in Fig. 1 (Caruana, 1998), which has a shared hidden layer for the two tasks, and two private hidden layers for cost prediction and LoS prediction, respectively.

The underlying assumption is that there is a common pool of factors shared between the two tasks, which is captured by the shared hidden layer. These soft constraints imposed on the generic parameters can yield better generalization, assuming the sharing is justified. The task-specific

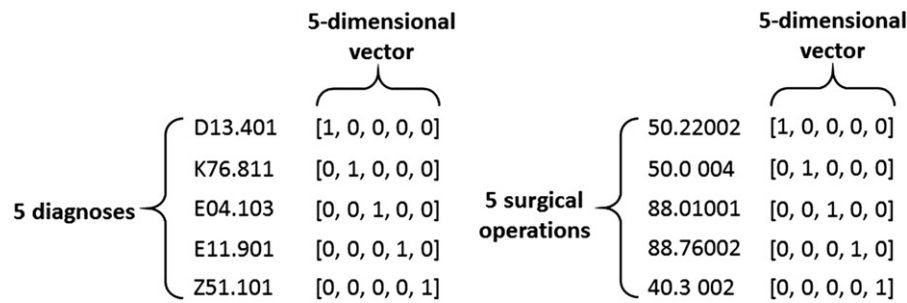


Figure 2. An illustration of one-hot vector representations for medical codes.

factors are captured by the private hidden layers. Besides, we use a “dropout” technique (Srivastava *et al.*, 2014) in input and hidden layers as the regularization method to further improve the generalization ability. However, it should be noticed that the neural network is a “black box model,” which means that the model cannot provide clear prediction rules. This makes the results difficult to interpret in practice.

#### 4. Data description and preprocessing

The Hospital Quality Monitoring System (HQMS) is a national database, established by the Ministry of Health of China, starting in 2011, as a nationwide initiative to ensure care delivery quality and efficiency. Our EHR data are extracted from HQMS, which includes information of all inpatients in five major hospitals in Beijing, China, from January 2013 to December 2015, containing around 750,000 records. Each record consists of patient demographic information (including age and gender), diagnosis codes (including 1 “primary diagnosis” and at most 10 “secondary diagnoses,” which are coded following the ICD-10 rule), surgical operation codes (including 1 “primary surgical operation” and at most 9 “secondary surgical operations,” which are coded following the ICD-9 rule), healthcare resource utilization (including LoS and cost), and other information (including number of hospitalizations, insurance type, inpatient department, etc.). We aim to jointly predict LoS and cost of the individual patient, given his/her demographic information, number of hospitalizations, and medical codes (diagnosis codes and surgical operation codes).

In this section, we show the process of data cleaning, and then illustrate how to construct feature vectors out of the EHR dataset.

##### 4.1. Data cleaning

Our data cleaning process consists of two steps. First, we filter out records with incomplete demographic information or containing “NA” term in medical codes, which account for 7% of all records. Second, we exclude records with “rare codes,” defined as medical codes with total occurrences fewer than 50 out of 750,000 cases. If a medical code has limited appearance in a dataset, we are unable to provide a statistically significant prediction of this code based on the dataset. In practice, the current DRG system does not consider the healthcare resource utilization for patients with

rare diseases. The statistical result shows that a small subset of codes accounts for a large fraction of total occurrences in the EHR dataset. Specifically, the number of distinct medical codes is around 16,000 in the original EHR dataset, while this number decreases to around 3,000 after the data cleaning.

After the previous two steps, we obtain a dataset containing around 600,000 records.

##### 4.2. Feature construction and dimensionality reduction for medical codes

As we have mentioned, we use demographic information, number of hospitalizations, and medical codes (diagnosis codes and surgical operation codes) to construct feature vectors. Age and number of hospitalizations are numerical features, while gender and medical codes are categorical features which need an extra encoding procedure before being used by prediction models. The common and fundamental way to represent categorical variables is using one-hot vector. If we have  $N$  categories, this method takes a  $1 \times N$  vector to represent a certain category, which consists of “0”s in all cells with the exception of a single “1” in a cell used uniquely to identify that category. For example, we can use a two-dimensional vector to express the gender of patients, with  $[1, 0]$  indicating male and  $[0, 1]$  indicating female. The medical codes can be expressed in a similar way. We show a simple example of one-hot vector representations for medical codes in Fig. 2, where there are only five distinct diagnosis codes and five distinct surgical operation codes.

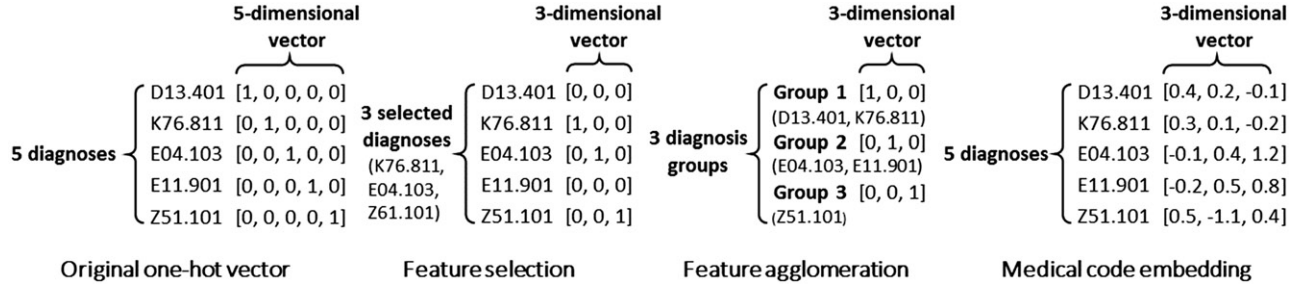
The feature vector components of a patient in this simple case are shown in Table 1. We find that the dimension of the final feature vector after component connection is 24, and 20 of them are responsible for the medical code representation.

However, in practice, the number of distinct medical codes is often in the scale of hundreds to thousands. In our case, after the data cleaning, we still have more than 3,000 distinct medical codes, which means the dimension of feature vectors generated by the one-hot method will be more than 6,000. In addition, most medical codes only appear in a few records, and each record only contains a few medical codes. The high dimensionality and event sparsity of medical codes can easily lead to over-fitting and poor generalization. We use five dimensionality reduction methods in three types to solve these challenges. We show, in the case study,



**Table 1.** Transformation of an electronic health record to the feature vector.

Type	Original value	Feature vector component
age	58	[58]
gender	female	[0,1]
number of hospitalization	1	[1]
primary diagnosis	D13.401	[1,0,0,0,0]
secondary diagnoses	K76.811, E04.103	[0,1,1,0,0]
primary surgical operation	50.22002	[1,0,0,0,0]
secondary surgical operations	50.0 004, 88.01001, 88.76002	[0,1,1,1,0]

**Figure 3.** Examples of different dimensionality reduction methods.

that these methods all outperform the original one-hot vector representation.

Simple examples of these three types of dimensionality reduction methods are shown in Fig. 3. With feature selection, only the selected medical codes are considered in one-hot vector construction. With feature agglomeration, medical codes are divided into groups. We construct a group-based one-hot vector, instead of a code-based one-hot vector. With medical code embedding, each code is mapped to a low-dimensional continuous vector. The first two types of dimensionality reduction methods maintain strong interpretability, while the embedding methods cannot provide clear correspondences between medical codes and feature dimensions, making the resulting feature vectors hard to interpret in practical application.

#### 4.2.1. Feature selection

The goal of feature selection is to select a limited number of “important” medical codes for one-hot vector construction.

**Removing features with low variance (RFLV).** This is a simple approach of feature selection, which removes all features whose variance is lower than a certain threshold. Since the original one-hot features are boolean features, with values being either one or zero, these features are Bernoulli random variables. The variance of such variables is given by:

$$\text{Var}[X] = p(1-p),$$

where  $p$  is the percentage of one or zero of the samples. If we want to remove all features that are either one or zero in more than 80% of the samples, we should select, using variance threshold,  $0.8 \star (1-0.8) = 0.16$ . The nearer  $p$  approaches 0.5, the fewer medical codes will be selected.

**Feature selection using Lasso.** Lasso regression can also be used for feature selection. Since many of its estimated coefficients are zero, we can implement Lasso regression on the

original one-hot feature vector first, and then select medical codes with non-zero coefficients. Recall that for Lasso, with larger  $\alpha$  comes fewer selected medical codes.

#### 4.2.2. Feature agglomeration

**Feature agglomeration using hierarchical clustering.** This approach aims to group medical codes that appear with high frequency within a specific group of patients into clusters (groups), and then use the cluster-based one-hot vector instead of code-based one-hot vector to achieve dimensionality reduction. Specifically, for a code-based one-hot vector, each dimension corresponds to a medical code. On the other hand, for a cluster-based one-hot vector, each dimension corresponds to a medical code group. For example, in Fig. 3, the code-based one-hot vector of code D13.401 is [1,0,0,0,0], which indicates that there are five distinct medical codes, and the first dimension corresponds to D13.401. The cluster-based one-hot vector of this code is [1,0,0], which indicates that there are three medical code groups after clustering, and D13.401 belongs to the group that corresponds to the first dimension. We perform a hierarchical clustering using a bottom-up approach to group codes: each medical code starts in its own cluster, and then clusters are successively merged together with criterion that minimizes the variance of the clusters being merged. The technical details can be found in Rokach and Maimon (2005) and Ward (1963). For feature agglomeration, the number of clusters is the dimension of the cluster-based one-hot vector.

#### 4.2.3. Medical code embedding

Word embedding is an important research topic in Natural Language Processing (NLP), which aims to find continuous vector representations for words. To implement word embedding, we need a corpus containing various sentences, where each sentence is composed of words. In our problem, the EHR dataset consists of records of patients, while each

record is composed of medical codes. Thus, word embedding in NLP can be generalized to medical code embedding.

We apply two word embedding techniques with deep learning, which provide state-of-the-art performance on several datasets for measuring syntactic and semantic word similarities (Mikolov *et al.*, 2013a).

**Continuous bag-of-words (CBOW) model.** The first model is CBOW. The training criterion is to correctly classify the current (middle) word based on the context (words around it). Formally, given a sequence of training words  $w_1, w_2, w_3, \dots, w_T$ , the objective of CBOW is to maximize the average log probability:

$$\frac{1}{T} \sum_{t=1}^T \sum_{-c \leq j \leq c, j \neq 0} \log p(w_t | w_{t+j}),$$

where  $c$  is the size of the training context. The conditional probability is defined using the softmax function:

$$p(w_j | w_i) = \frac{\exp(v_{w_j}^T v_{w_i})}{\sum_{k=1}^W \exp(v_{w_k}^T v_{w_i})},$$

where  $v_{w_j}$  is the vector representation of word  $w_j$ , and  $W$  is the number of distinct words.

**Continuous Skip-gram model.** Instead of predicting the current word based on the words before and after it, the Skip-gram model aims to predict surrounding words given the current word. Thus, the objective of Skip-gram is:

$$\frac{1}{T} \sum_{t=1}^T \sum_{-c \leq j \leq c, j \neq 0} \log p(w_{t+j} | w_t).$$

The definition of the conditional probability is the same as CBOW. This method has also been adopted by Min *et al.* (2016) for the prediction of healthcare cost.

In order to implement these two methods, we construct a medical sentence by combining all medical codes of a patient and randomly shuffling them. Next, we use the medical corpus that consists of all these medical sentences to train the models and obtain the vector representation of each medical code. The detailed training process can be found in Mikolov *et al.* (2013b). Except for the dimension of the feature vector, we need to decide the size of the training context  $c$ . We set it to be 2, since the length of more than 80% medical sentences is shorter than 5.

As we have mentioned earlier, all five dimensionality reduction methods contain a hyper parameter that determines the dimension of medical code vector. We decide the value of these hyper parameters by cross-validation. Specifically, we observe the performance of prediction models on validation sets based on medical code vectors with varying dimension sizes, and choose the one that makes the prediction model achieve the highest accuracy.

## 5. Case study

We use the python package scikit-learn to build the multi-task Lasso, the decision tree, and the random forest. The neural network is implemented by python package lasagne.

### 5.1. Accuracy metrics

We employ four metrics (Sheiner and Beal, 1981; Cameron and Windmeijer, 1997) to evaluate the prediction accuracy of different prediction models.

The  $R^2$  coefficient

For each output, the  $R^2$  coefficient is defined as:

$$R^2 = 1 - \frac{\sum_i (y_i - \hat{y}_i)^2}{\sum_i (y_i - \bar{y})^2},$$

where  $y_i$  is the true response of sample  $i$ ,  $\hat{y}_i$  is the predicted response of sample  $i$ , and  $\bar{y}$  is the average response. The best possible value of  $R^2$  is 1.0, indicating a perfect fit. A constant model that always predicts the average response  $\bar{y}$ , disregarding the input features, would result in a  $R^2$  score of 0.0. It can be negative because the model can be arbitrarily worse. In addition, we use the average  $R^2$  of cost and LoS as a unified metric to evaluate the prediction performance of the multi-output regression models, which is defined as:

$$\bar{R}^2 = \frac{1}{2} (R_{\text{cost}}^2 + R_{\text{LoS}}^2).$$

*The root mean squared error (RMSE)* RMSE is a common-used metric for the evaluation of prediction accuracy. It is defined as the root of the expected value of the squared error:

$$\text{RMSE} = \sqrt{\frac{\sum_i (y_i - \hat{y}_i)^2}{N}},$$

where  $N$  is the number of samples.

*The mean absolute error (Mean-AE)* Mean-AE is defined as the expected value of the absolute error:

$$\text{Mean-AE} = \frac{1}{N} \sum_i |y_i - \hat{y}_i|.$$

*The mean absolute error (Mean-AE)* The mean absolute error (Mean-AE) This metric does not square the prediction errors and thus is less sensitive to outliers compared to  $R^2$  and RMSE. This makes it a widely adopted metric for healthcare cost prediction (Bertsimas *et al.*, 2008), since there are a few patients with very unpredictable high cost.

*The median absolute error (Median-AE)* Median-AE is defined as the median of all absolute errors between the real response and the predicted response:

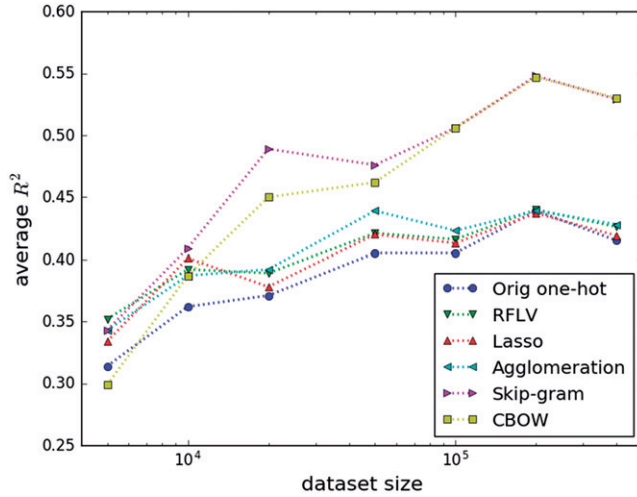
$$\text{Median-AE} = \text{median}(|y_1 - \hat{y}_1|, \dots, |y_N - \hat{y}_N|).$$

This metric is even more robust to outliers compared to Mean-AE.

Since LoS and cost have very different magnitudes, we standardize these two responses of the training set by removing the mean and scaling to unit variance before training. We transform the responses of the test set based

**Table 2.** The best dimension of feature vectors under varying dataset sizes.

Dataset size	RFLV	Lasso	Agglomeration	Skip-gram	CBOW
5,000	1478	324	1000	80	80
10,000	1362	166	1000	80	200
20,000	1390	914	1000	200	800
50,000	1439	510	1000	80	200
100,000	2559	472	1000	80	40
200,000	2509	1012	2000	80	80
400,000	3726	731	1000	200	400

**Figure 4.** Comparison of predictive ability with feature vectors generated by different dimensionality reduction methods.**Table 3.** The best dimension of feature vectors for different prediction models.

	Lasso	Decision Tree (DT)	Random Forest (RF)	Neural Network (NN)
Agglomeration	2000	1000	1000	2000
Skip-gram	80	80	80	2000

on the mean and variance computed based on the training set before prediction. We next scale back the responses to the original representations before computing the accuracy metrics. It should be noted that  $R^2$  is a dimensionless quantity; i.e., there is no need to specify the unit. The unit of the three other metrics is day for LoS prediction, and thousand-yuan for cost prediction.

## 5.2. Dimensionality reduction

In order to choose the suitable feature vectors for prediction tasks, we compare the prediction performance of the random forest with the feature vectors constructed by the five dimensionality reduction methods. The one-hot vector is used as the benchmark. We set the number of trees in the random forest to be 10, the maximum depth of each tree to be 50, and the minimum number of samples required for each node to be 50.

We find the best dimension of feature vectors obtained by each method through cross-validation under varying dataset sizes. The result is shown in Table 2. We can see that the trends of the best dimension of agglomeration and Skip-gram are much more stable than that of the three other

methods, which makes these two methods much easier to use in practice.

We set the size of the test set to be 20% of the corresponding training set size. The  $\bar{R}^2$  score curves on test sets of benchmark and the five dimensionality reduction methods with their best dimension are shown in Fig. 4. In general, the five sets of feature vectors generated by dimensionality reduction methods significantly outperform the one-hot feature vectors (benchmark). Among them, the prediction model with the feature vectors generated by Skip-gram achieves the highest accuracy.

However, it should be noted that the embedding methods (CBOW and Skip-gram) cannot provide clear correspondence between medical codes and feature dimensions, and thus greatly weaken the interpretability of the prediction model from clinical perspective. On the other hand, the three other dimensionality reduction methods can provide clear explanation of their feature vectors, where each cell corresponds to a medical code (RFLV/Lasso) or a medical code cluster (agglomeration). Among these three sets of feature vectors with strong interpretability, the prediction model with the feature vectors generated by agglomeration achieves the highest accuracy.

Similarly, we compare dimensionality reduction methods by the other three multi-output regression models. The results show that for Lasso, feature vectors generated by agglomeration have the best predictive performance, while for decision tree and neural network, feature vectors generated by Skip-gram are the best.

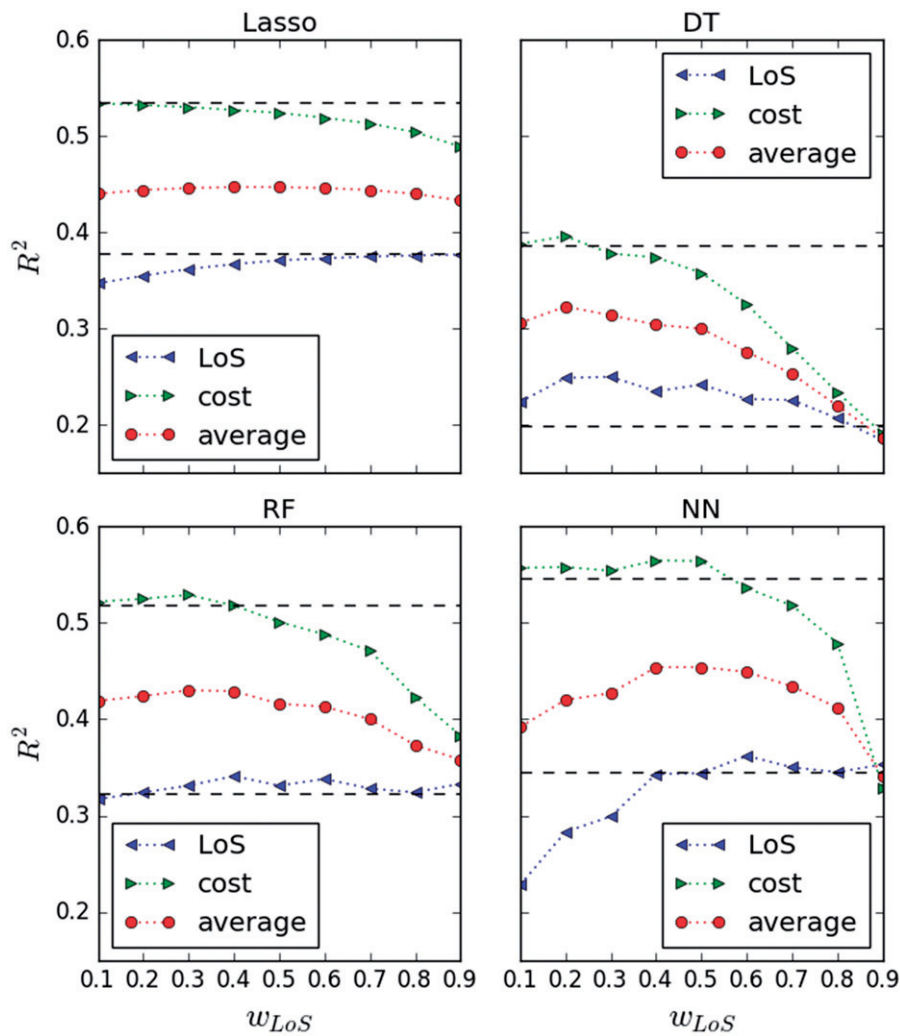
In summary, considering the ease of implementation, the prediction performance, and the model interpretability, we adopt two sets of feature vectors generated by Skip-gram and agglomeration for further analyses. The best dimension is determined by cross-validation for the four prediction models. The result is shown in Table 3.

For parameters in each machine learning model, some can be automatically optimized by training; the others (denoted as hyper parameters) should be set before training. Given these two sets of feature vectors, we set the hyper parameters of regression models under varying dataset sizes through cross-validation.

## 5.3. The influence of the weights of prediction tasks

As we have mentioned, we standardize LoS and cost before training. In order to compare the performance of multi-output models and single-output models, we add weights  $w_{LoS}$  and  $w_{cost}$  to these two responses that satisfy  $w_{LoS} + w_{cost} = 1$ , and  $w_{LoS}, w_{cost} \in [0, 1]$ . When  $w_{LoS} = 1(0)$ , the multi-output regression problem degrades to the single-output regression problem for LoS (cost) prediction. We show  $R^2$  on test sets under varying weights  $w_{LoS}$  in Fig. 5. The range of  $w_{LoS}$  (x-axis) in each subgraph is from 0.1 to 0.9, which shows the influence of weights on the performance of multi-output models. For the ease of comparison, the results corresponding to  $w_{LoS} = 0$  or 1 are illustrated by horizontal dotted lines, which show the performance of single-output models.





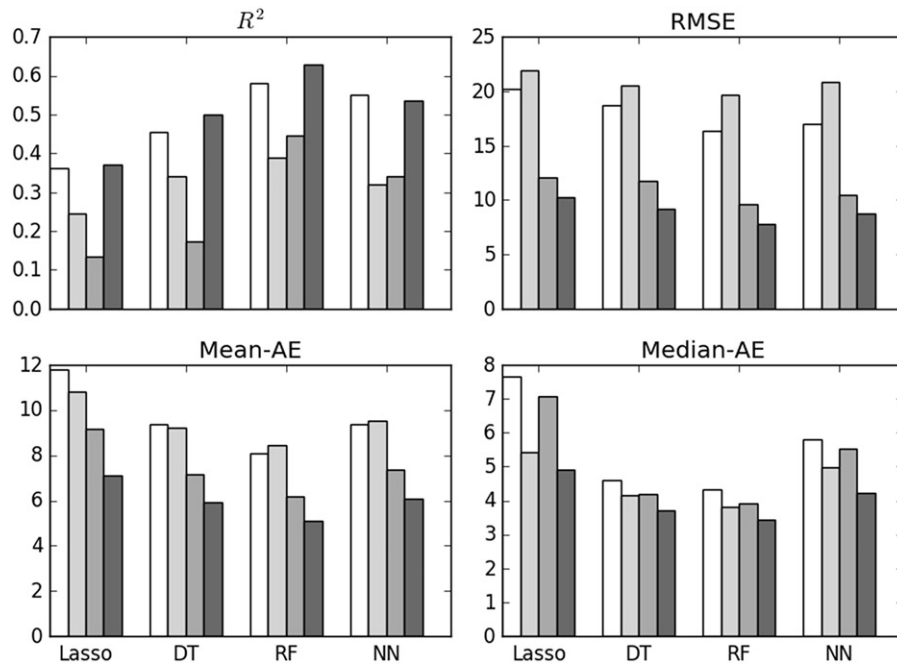
**Figure 5.**  $R^2$  on test sets of four multi-output regression models under varying weights  $w_{LoS}$ . Training set size is set to be 20,000. Test set size is set to be 4,000. Feature vectors are generated by agglomeration. In each graph, the upper dotted line corresponds to  $w_{LoS} = 0.0$  (single-output regression problem for cost), and the lower dotted line corresponds to  $w_{LoS} = 1.0$  (single-output regression problem for LoS).

We can see that the multi-task Lasso can hardly improve prediction accuracy by learning the two responses jointly. However, its  $R^2$  curves are very flat; thus, instead of learning two single-output Lasso models, we can set  $w_{LoS} = 0.5$  and use multi-output models to halve the training time. The prediction accuracy of the three other models can all benefit from learning two tasks jointly. An interesting phenomenon in Fig. 5 is that the optimal weights are not 0.5 for the two prediction tasks, which is the default choice for the establishment of multi-output models. Generally speaking, better prediction performance can result from giving the cost a higher weight. The values of  $w_{LoS}$  with which the two responses achieve the highest accuracy are very close. We set  $w_{LoS} = 0.2$  for the decision tree,  $w_{LoS} = 0.3$  for the random forest, and  $w_{LoS} = 0.4$  for the neural network, to achieve an improvement of prediction accuracy by 3% ~20%. Among these three models, the decision tree obtains the most significant prediction accuracy improvement through multi-output models. The decision tree is widely used in practice for its simplicity and strong interpretability. Thus, the integration of multi-output models can greatly benefit the practical applications. As for the random forest

and neural network, except for the prediction accuracy improvement, the integration of multi-output models can lead to significant training time savings, from minutes to hours.

#### 5.4. Discussion of training dataset preprocessing

Since the accuracy metrics may be unduly influenced by very high cost or very long LoS, the quantile-truncated metrics are widely adopted to evaluate the performance of regression models (Duncan *et al.*, 2016; Bertsimas *et al.*, 2008); i.e., ranking the health records according to the utilization of healthcare resources, and then computing the metrics after the  $1-\alpha$  fraction of the largest residuals, which corresponds to very high cost or very long LoS, is removed from the test set. In practice, this corresponds to the case where the high-risk detection is integrated into the prediction system. Thus, incoming high-risk patients with a tendency to incur extremely high healthcare resource utilization will be filtered out, and wait for physicians to judge their situations individually.



**Figure 6.** The values of four accuracy metrics on test sets under different quantile-truncated schemes. Training set size is set to be 20,000. Test set size is set to be 4,000. Feature vectors are generated by Skip-gram. For clarity, we only show the values of metrics for cost prediction; the results for LoS prediction are similar. In each graph, the four bars of each model from left to right correspond to: 100%train + 100%test, 95%train + 100%test, 100%train + 95%test, 95%train + 95%test.

**Table 4.** Performance of multi-output regression models on a medium-sized dataset.

Metric	Objective	Agglomeration				Skip-gram			
		Lasso*	DT*	RF*	NN**	Lasso**	DT**	RF**	NN**
$R^2$	LoS	<b>0.388</b>	0.195	0.375	0.363	0.331	0.311	0.403 <sup>†</sup>	0.257
	cost	<b>0.541</b>	0.329	0.508	0.520	0.349	0.449	0.578 <sup>†</sup>	0.294
RMSE	LoS	<b>3.76</b>	4.31	3.79	3.83	3.93	3.99	3.71 <sup>†</sup>	4.14
	cost	<b>9.45</b>	11.40	9.78	9.65	11.30	10.40	9.05 <sup>†</sup>	11.70
Mean-AE	LoS	2.79	3.25	<b>2.76</b>	2.72	2.93	2.91	2.69 <sup>†</sup>	3.38
	cost	6.43	7.59	<b>6.34</b>	6.30	7.77	6.80	5.93 <sup>†</sup>	9.16
Median-AE	LoS	2.12	2.55	<b>1.97</b>	1.89 <sup>†</sup>	2.30	2.16	1.98	3.05
	cost	4.77	5.60	<b>4.34</b>	3.97	5.47	4.33	3.92 <sup>†</sup>	7.86

Notes. \* interpretable model;

\*\* uninterpretable model;

<sup>†</sup> best score.

**Table 5.** Performance of multi-output regression models on a large-scale dataset.

Metric	Objective	Agglomeration				Skip-gram			
		Lasso*	DT*	RF*	NN**	Lasso**	DT**	RF**	NN**
$R^2$	LoS	<b>0.495</b>	0.394	0.454	0.514	0.379	0.473	0.554 <sup>†</sup>	0.538
	cost	<b>0.663</b>	0.570	0.619	0.670	0.476	0.673	0.738 <sup>†</sup>	0.709
RMSE	LoS	<b>3.30</b>	3.62	3.44	3.25	3.66	3.38	3.10 <sup>†</sup>	3.17
	cost	<b>7.68</b>	8.67	8.17	7.61	9.58	7.56	6.78 <sup>†</sup>	7.15
Mean-AE	LoS	<b>2.38</b>	2.61	2.46	2.27	2.75	2.39	2.19 <sup>†</sup>	2.26
	cost	<b>4.87</b>	5.46	5.12	4.69	6.73	4.68	4.18 <sup>†</sup>	4.45
Median-AE	LoS	<b>1.74</b>	1.86	1.76	1.53 <sup>†</sup>	2.10	1.68	1.54	1.63
	cost	<b>3.08</b>	3.47	3.24	2.73	4.77	2.74	2.48 <sup>†</sup>	2.67

Notes. \* interpretable model;

\*\* uninterpretable model;

<sup>†</sup> best score.

On the other hand, whether to filter out records with high resource utilization in the training set has been ignored in previous research. We discuss this issue in two cases: with or without 5% fraction of the largest residuals in the test set being removed before prediction. We show the values of four accuracy metrics on test sets under different quantile-truncated schemes in Fig. 6.

Recall that, for  $R^2$ , higher value is better, while for the three other accuracy metrics, lower value is better. We can see that if the test set is truncated (corresponding to the two rightmost bars of each model in each subgraph of Fig. 6), it is always better to truncate the training set as well. On the other hand, if we need to predict the healthcare resource utilization for all incoming patients, things become more

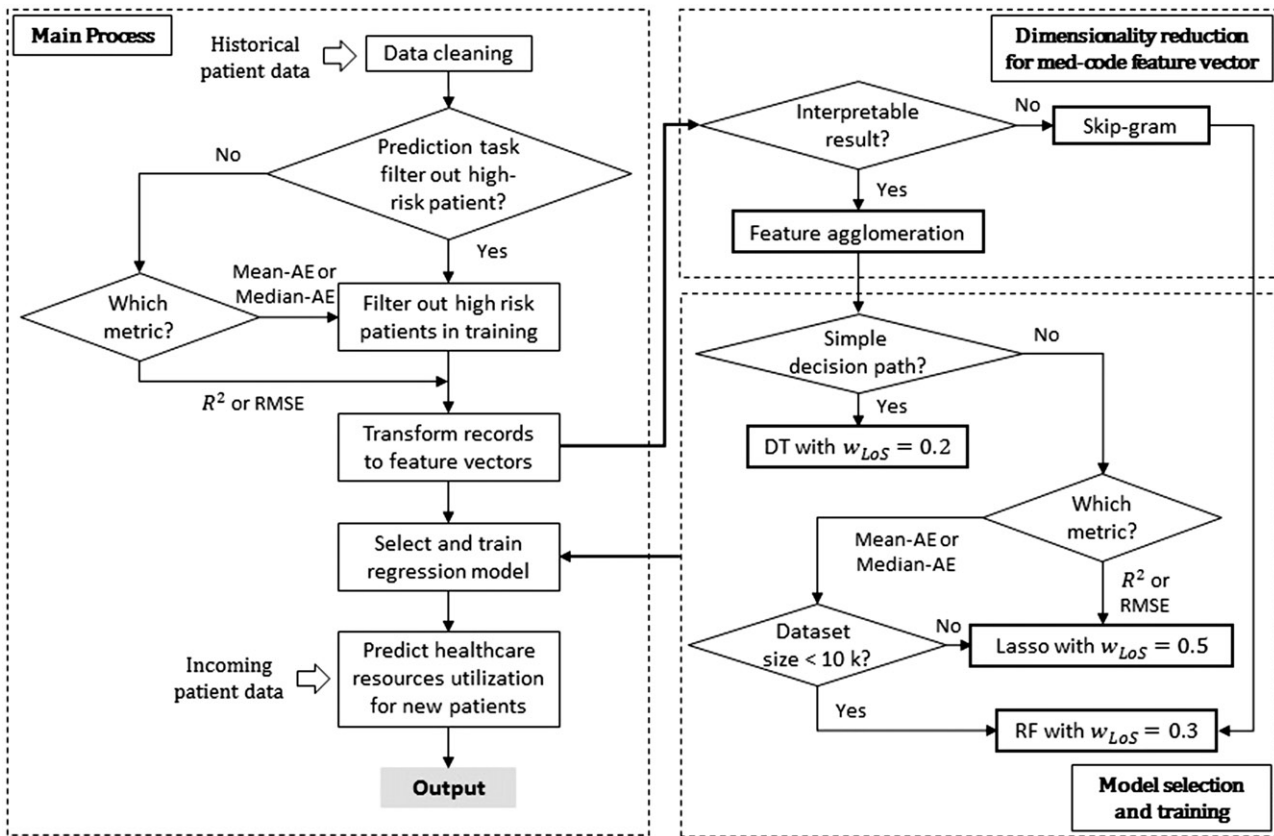


Figure 7. A guideline for the selection and construction of multi-output models.

complicated. In general, if we use  $R^2$  or RMSE as the accuracy metric, the training set should not be truncated, while if we use Mean-AE or Median-AE as the accuracy metric, the training set should be truncated. The underlying reason is that truncating the training set will improve the model performance on low-risk patients, but affect the model performance on high-risk patients, reflected by the metrics that are sensitive to outliers ( $R^2$  and RMSE).

### 5.5. Models with varying dataset sizes

We observe the performance of four prediction models under different training dataset sizes, varying from 5,000 to 400,000. The size of the test set is set to be 20% of the corresponding training set size. Both the training records and the test records are randomly selected from the EHR dataset. The 5% fraction of the largest residuals in training sets and test sets are removed. We show the performance of multi-output models under 10,000 training records (as representative of medium-sized dataset) in Table 4. The results under 400,000 training records (as representative of large-scale dataset) are shown in Table 5.

We divide these models into two groups by interpretability. The best score in each group is shown in bold (the overall best score is indicated by a cross superscript).

If the model interpretability is required, we should use feature vectors generated by agglomeration, and choose the regression model from the multi-task Lasso, decision trees, or random forests. For the medium-sized dataset, when we use  $R^2$  or RMSE as the accuracy metric, we should use the

multi-task Lasso as the regression model. On the other hand, if we use Mean-AE or Median-AE as the accuracy metric, the random forest becomes a suitable choice. For a large-scale dataset, the multi-task Lasso achieves the highest accuracy among the three interpretable models. It should be noted that, with feature vectors generated by agglomeration, although the neural network outperforms the multi-task Lasso, this model is a “black box model” that cannot satisfy the requirement of interpretability.

On the other hand, if the model interpretability is not required, the random forest with feature vectors generated by Skip-gram achieves the highest prediction accuracy on both medium-sized and large-scale datasets.

### 5.6. A guideline for practical use

To summarize the previous discussion, we provide a guideline in Fig. 7 for the construction of multi-output models, considering factors such as the implementation environment, the model interpretability, dataset size, and accuracy metrics.

In the main process, after the data cleaning of historical patient data, we need to decide whether to filter out patients with high healthcare resource utilization based on the situation of the prediction task and the accuracy metrics. Then, we need to transform records to feature vectors. If interpretable regression rules are required, we should use feature agglomeration to do the transformation. Otherwise, we should use Skip-gram to obtain better performance on prediction accuracy. After that, we need to select the regression

model for prediction tasks. If simple decision rules that can be explained by boolean logic are required, we should use the decision tree. Recall that, in this case, the integration of the multi-output models can significantly improve the prediction accuracy. If more complicated rules are acceptable, we should choose from the multi-task Lasso or the random forest based on the accuracy metric and dataset size. If there is no requirement for model interpretability, we should use the random forest with feature vectors generated by Skip-gram to achieve the highest prediction accuracy. After the model selection and training stage, we can predict the healthcare resource utilization for incoming patients.

## 6. Conclusion and future research

In this study, we aim to jointly predict two measures of the healthcare resource utilization through multi-output regression models. This approach provides a unified and efficient tool for practical use, and can improve prediction accuracy compared to single-output models.

We utilize five methods to transform medical codes to feature vectors, and four machine learning models for predictive analytics. We observe the model performance under varying dataset sizes, compare multi-output models with single-output models by assigning weights to two responses, and discuss whether to filter out records with high resource utilization in training sets based on the implementation environment. Based on the results of numerical experiments, we provide a guideline for prediction model selection and construction, considering the model interpretability, accuracy metrics, dataset size, etc. We believe that this result can benefit researchers and practitioners who work on similar problems.

The multi-output regression models in our study can be easily extended to solve other similar problems. We show that the performance of decision trees can be significantly improved by integrating multi-output models. Since the decision tree is widely used in practice, it has practical significance to study how to further improve its prediction accuracy without affecting its strong interpretability.

Furthermore, the prediction framework can be used to support clinical decision making and the refinement of reimbursement structures, which has the potential to control healthcare expenses while maintaining optimal outcomes and motivating effective coordinated care. Clinical decision support systems integrated with prediction models can provide direct feedback on the effect of any clinical practice to healthcare resource utilization (Zikos and Ostwal, 2016). For the prospective payment model, one of the crucial components for its successful implementation is the identification of procedural homogeneous groups, to which a flat reimbursement rate can be applied (Tong *et al.*, 2015). Prediction models can serve as an important fundamental input.

## Funding

This research is supported by the National Natural Science Foundation of China under Grants 71210002, 71332005, 71501109 and 71432004.

## ORCID

Liwen Cui  <http://orcid.org/0000-0001-7045-8234>

## References

- Aylin, P., Bottle, A., and Majeed, A. (2007) Use of administrative data or clinical databases as predictors of risk of death in hospital: Comparison of models. *BMJ* **334**(7602), 1044.
- Bakker, B. and Heskes, T. (2003) Task clustering and gating for Bayesian multitask learning. *Journal of Machine Learning Research* **4**(May), 83–99.
- Ben-David, S. and Schuller, R. (2003) Exploiting task relatedness for multiple task learning. In *Learning Theory and Kernel Machines* ed. B. Schölkopf and M. K. Warmuth, 567–580. Springer, Berlin, Heidelberg.
- Benesty, J., Chen, J., Huang, Y., and Cohen, I. (2009) Pearson correlation coefficient. In *Noise Reduction in Speech Processing*, 1–4. Springer, Heidelberg, Germany.
- Bertsimas, D., Bjarnadóttir, M. V., Kane, M. A., Kryder, J. C., Pandey, R., Vempala, S., and Wang, G. (2008) Algorithmic prediction of health-care costs. *Operations Research* **56**(6), 1382–1392.
- Bjarnadóttir, M. V., Czerwinski, D., Guan, Y., Yang, H., and Lee, E. K. (2016) *The history and modern applications of insurance claims data in healthcare research. Healthcare Analytics: From Data to Knowledge to Healthcare Improvement*, ed. W. Pedrycz, 561–591, Wiley Online Library, New York, United States.
- Borchani, H., Varando, G., Bielza, C., and Larrañaga, P. (2015) A survey on multi-output regression. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery* **5**(5), 216–233.
- Breiman, L., Friedman, J. H., Olshen, R. A., and Stone, C. J. (1984) *Classification and Regression Trees*. Wadsworth, Amsterdam, Netherlands.
- Cameron, A. C. and Windmeijer, F. A. (1997) An R-squared measure of goodness of fit for some common nonlinear regression models. *Journal of Econometrics* **77**(2), 329–342.
- Caruana, R. (1998) A dozen tricks with multitask learning. In *Neural Networks: Tricks of the Trade*, 65–191. Springer, Washington, DC, United States.
- De'Ath, G. (2002) Multivariate regression trees: A new technique for modeling species–environment relationships. *Ecology* **83**(4), 1105–1117.
- Duncan, I., Loginov, M., and Ludkovski, M. (2016) Testing alternative regression frameworks for predictive modeling of health care costs. *North American Actuarial Journal* **20**(1), 65–87.
- Hagan, M. T., Demuth, H. B., Beale, M. H., and De Jesús, O. (1996) *Neural Network Design* **20**. PWS, Maryland, United States.
- Hillestad, R., Bigelow, J., Bower, A., Girosi, F., Meili, R., Scoville, R., and Taylor, R. (2005) Can electronic medical record systems transform health care? Potential health benefits, savings, and costs. *Health Affairs* **24**(5), 1103–1117.
- Jensen, P. B., Jensen, L. J., and Brunak, S. (2012) Mining electronic health records: Towards better research applications and clinical care. *Nature Reviews Genetics* **13**(6), 395–405.
- Jha, A. K., DesRoches, C. M., Campbell, E. G., Donelan, K., Rao, S. R., Ferris, T. G., Shields, A., Rosenbaum, S., and Blumenthal, D. (2009) Use of electronic health records in US hospitals. *New England Journal of Medicine* **360**(16), 1628–1638.



- Lee, S., Zhu, J., and Xing, E. P. (2010) Adaptive multi-task lasso: With application to eQTL detection. In *Advances in Neural Information Processing Systems*, eds. J. D. Lafferty, C.K.I. Williams, J. Shawe-Taylor, R.S. Zemel, and A. Culotta, 1306–1314, Neural Information Processing Systems Foundation, Inc., Vancouver, Canada.
- Lee, S.-M., Kang, J.-O., and Suh, Y.-M. (2004) Comparison of hospital charge prediction models for colorectal cancer patients: Neural network vs. decision tree models. *Journal of Korean Medical Science* 19(5),677–681.
- Liaw, A. and Wiener, M. (2002) Classification and regression by random forest. *R New* 2(3),18–22.
- Mikolov, T., Chen, K., Corrado, G., and Dean, J. (2013a) Efficient estimation of word representations in vector space. *arXiv preprint arXiv* 3, 1301–3701.
- Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., and Dean, J. (2013b) Distributed representations of words and phrases and their compositionality. In *Advances in Neural Information Processing Systems*, eds. C.J.C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K.Q. Weinberger, 3111–3119, Neural Information Processing Systems Foundation, Inc., Lake Tahoe, United States.
- Min, X., Xie, X., Wang, H., Chen, N., and Chen, T. (2016) Medical concepts embedding and visualization. Paper presented at: *Translational Bioinformatics Conference*; October 15–16, Hyatt Regency Jeju, Jeju Island, Korea.
- Mkanta, W. N., Chumbler, N. R., Yang, K., Saigal, R., and Abdollahi, M. (2016) Cost and predictors of hospitalizations for ambulatory care-sensitive conditions among medicaid enrollees in comprehensive managed care plans. *Health Services Research and Managerial Epidemiology* 3,2333392816670301.
- Murdoch, T. B. and Detsky, A. S. (2013) The inevitable application of big data to health care. *JAMA* 309(13),1351–1352.
- Nambiar, R., Bhardwaj, R., Sethi, A., and Vargheese, R. (2013) A look at challenges and opportunities of big data analytics in healthcare. In *2013 IEEE International Conference on Big Data*, 17–22. IEEE, California, United States.
- Obozinski, G. and Taskar, B. (2006) Multi-task feature selection. Paper presented at: *International Conference on Machine Learning*; December 14–16, Florida, United States.
- Quinlan, J. R. (1986) Induction of decision trees. *Machine Learning* 1(1),81–106.
- Quinlan, J. R. (1993) *C4.5: Programs for Machine Learning*. Morgan Kaufmann, California, United States.
- Rea, S., Pathak, J., Savova, G., Oniki, T. A., Westberg, L., Beebe, C. E., Tao, C., Parker, C. G., Haug, P. J., Huff, S. M., et al. (2012) Building a robust, scalable and standards-driven infrastructure for secondary use of EHR data: The SHARP project. *Journal of Biomedical Informatics* 45(4),763–771.
- Rokach, L. and Maimon, O. (2005) Clustering methods. In *Data Mining and Knowledge Discovery Handbook*, 321–352. Springer, Massachusetts, United States.
- Ross, M., Wei, W., and Ohno-Machado, L. (2014) “Big data” and the electronic health record. *Yearbook of Medical Informatics* 9(1),97.
- Ryu, E., Takahashi, P. Y., Olson, J. E., Hathcock, M. A., Novotny, P. J., Pathak, J., Bielinski, S. J., Cerhan, J. R., and Sloan, J. A. (2015) Quantifying the importance of disease burden on perceived general health and depressive symptoms in patients within the Mayo Clinic Biobank. *Health and Quality of Life Outcomes* 13(1),1.
- Sheiner, L. B. and Beal, S. L. (1981) Some suggestions for measuring predictive performance. *Journal of Pharmacokinetics and Pharmacodynamics* 9(4),503–512.
- Srivastava, N., Hinton, G. E., Krizhevsky, A., Sutskever, I., and Salakhutdinov, R. (2014) Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research* 15(1),1929–1958.
- Sushmita, S., Newman, S., Marquardt, J., Ram, P., Prasad, V., Cock, M. D., and Teredesai, A. (2015) Population cost prediction on public healthcare datasets. In *Proceedings of the 5th International Conference on Digital Health*, 87–94. ACM, Florence, Italy.
- Takahashi, P. Y., Ryu, E., Olson, J. E., Anderson, K. S., Hathcock, M. A., Haas, L. R., Naessens, J. M., Pathak, J., Bielinski, S. J., and Cerhan, J. R. (2013) Hospitalizations and emergency department use in Mayo Clinic Biobank participants within the employee and community health medical home. In *Mayo Clinic Proceeding* 88, 963–969. Elsevier, Amsterdam, Netherlands.
- Tibshirani, R. (1996) Regression shrinkage and selection via the Lasso. *Journal of the Royal Statistical Society, Series B (Methodological)* 267–288.
- Tomar, D. and Agarwal, S. (2013) A survey on data mining approaches for healthcare. *International Journal of Bio-Science and Bio-Technology* 5(5),241–266.
- Tong, K., Berthet, J., and Shrader, G. (2015) *Predicting Patient Costs*. Ph.D. thesis, University of Chicago.
- Ward Jr., J. H. (1963) Hierarchical grouping to optimize an objective function. *Journal of the American Statistical Association* 58(301),236–244.
- Wu, J., Roy, J., and Stewart, W. F. (2010) Prediction modeling using EHR data: Challenges, strategies, and a comparison of machine learning approaches. *Medical Care* 48(6),S106–S113.
- Xie, Y., Schreier, G., Chang, D. C., Neubauer, S., Liu, Y., Redmond, S. J., and Lovell, N. H. (2015) Predicting days in hospital using health insurance claims. *IEEE Journal of Biomedical and Health Informatics* 19(4),1224–1233.
- Xie, Y., Schreier, G., Hoy, M., Liu, Y., Neubauer, S., Chang, D. C., Redmond, S. J., and Lovell, N. H. (2016) Analyzing health insurance claims on different timescales to predict days in hospital. *Journal of Biomedical Informatics* 60,187–196.
- Yoo, I., Alafaireet, P., Marinov, M., Pena-Hernandez, K., Gopidi, R., Chang, J.-F., and Hua, L. (2012) Data mining in healthcare and biomedicine: A survey of the literature. *Journal of Medical Systems* 36(4),2431–2448.
- Zhou, J., Yuan, L., Liu, J., and Ye, J. (2011) A multi-task learning formulation for predicting disease progression. In *Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 814–822. ACM, California, United States.
- Zikos, D. and Ostwal, D. (2016) A platform based on multiple regression to estimate the effect of in-hospital events on total charges. In *IEEE International Conference on Healthcare Informatics*, 403–408. Institute of Electrical and Electronics Engineers, Inc, Chicago, United States.