HEDG

HEALTH, ECONOMETRICS AND DATA GROUP

THE UNIVERSITY of York

WP 24/09

Entity embedding of high-dimensional claims data for hospitalized exacerbation prediction

Johannes Cordier; Alexander Geissler and Justus Vogel

August 2024

http://www.york.ac.uk/economics/postgrad/herc/hedg/wps/

Entity embedding of high-dimensional claims data for hospitalized exacerbation prediction

Johannes Cordier*, Alexander Geissler, and Justus Vogel

Chair of Health Economics, Policy and Management, School of Medicine, University of St. Gallen

31.05.2024

Abstract

This study addresses the challenges of using high-dimensional claims data, typically represented by categorical features, for prediction tasks. Traditional one-hot encoding methods lead to computational inefficiencies and sparse data issues. To overcome these challenges, we propose using entity embedding, a technique that has shown promise in natural language processing, to transform categorical claims data into dense, low-dimensional vectors as input for downstream prediction tasks. Our study focuses on predicting hospitalizations for patients with Chronic Obstructive Pulmonary Disease using the Word2Vec Continuous Bag-of-Words model. Our findings indicate that entity embedding enhances model performance, achieving an AUC of 0.92 compared to 0.91 with one-hot encoding, and improves specificity from 0.55 to 0.60 for a recall of 0.95. Additionally, entity embedding significantly reduces required computation power. These results suggest that entity embedding not only captures the dynamics of medical events more effectively but also enhances the efficiency of training predictive models, making it a valuable tool for healthcare and insurance analytics.

Funding: We gratefully thank the Groupe Mutuel for granting us access to their insurance claims data.

*Corresponding Author, johannes.cordier@unisg.ch

1 Introduction

Health insurance claims data are commonly constructed as categorical features with a vast amount of categories, if each tariff item receives its own category. In Switzerland, for instance, the outpatient fee-for-service catalogue "tarif médical" (TARMED) has more than $4,600$ items, the inpatient DRG-based payment catalogue for acute somatic care (SwissDRG) has more than $1,000$ items. Moreover there are several tens of thousands of different medications and plenty of other services (e.g., ergotherapy, or physiotherapy) covered by the Swiss health insurance, all with specific item codes (De Pietro et al., 2015). These high-dimensional data are further complicated by two factors: (1) the same service can be provided and coded multiple times in a certain time period (e.g., an insuree sees the general practitioner twice in a given week), and (2) there are codes that do not provide any additional information about the medical treatment (e.g., in the TARMED, the first and last five minutes of any physician visit are both always coded separately and must be recorded on each associated claim).

Still, these data are highly valuable for a variety of research questions, as the data is standardized and widely available. First and foremost predicting events based on insuree characteristics, or confounding for average treatment effects in non-randomized settings. Accordingly, methods exist to either encode high-dimensional features or to reduce dimensionality by feature selection. Both approaches come with limitations, however.

The standard method for encoding high-dimensional features is one-hot encoding, also called dummification (Zheng & Casari, 2018; Kuhn & Johnson, 2019), where the categorical feature is transformed into $N_{categories} - 1$ binary columns. One-hot encoding is straightforward and widely used, but it can be computationally expensive and often leads to sparse data representations Murphy (2012). Most insurees only receive a (relatively small) subset of available services, which means the majority of observations in a column vector is equal to zero. If all or most health services provided to an insuree were to be considered in a model, this model would need to be able to deal with sparse high-dimensional insuree vectors. Most out-of-the-box methods like linear regressions, random forests, and gradient-boosted trees are less efficient when dealing with sparse data compared to dense data, however, or cannot deal with them at all, as in the case of a linear regression due to failing matrix inversion. Moreover, we expect that one-hot encoding might have trouble dealing with same services rendered multiple times and including codes without additional information in their model matrix without increasing the dimensionality even more.

Alternatively, studies aim to reduce the number of considered features and deal with sparse data by feature selection, reducing dimensionality in turn. feature selection can be carried out either through statistical methods (for example shrinkage methods like LASSO (Tibshirani, 1996), or forward selection (Efroymson, 1960; Wilkinson & Dallal, 1981)), or through manual selection by the practitioner. Both approaches come with limitations, however. Forward selection leads to significant optimism due to inflated Type I error and an overestimation of the explained variance (Rencher & Pun, 1980; Blanchet et al., 2008). Shrinkage methods require significant sample size and computation power and potentially valuable information is lost in the process due to its linearity if interaction terms are not included. Manual selection requires decisions on feature inclusion and exclusion. Deliberate, well-reasoned decisions for all available features are unrealistic in light of the vast amount of categories, however. Moreover, such decisions can be arbitrary and are prone to biases for example when combining multiple medications.

An alternative to one-hot encoding is entity embedding. Entity embedding has emerged as a powerful technique for representing categorical features (Guo & Berkhahn, 2016). By mapping categories

to continuous, dense, low-dimensional vectors, entity embedding captures intricate relationships and contextual information inherent in the data. This seems especially promising in our research setting in which a model must learn what items are relevant predictors and weight them accordingly. Entity embedding originates from natural language processing (NLP) as words in a sentence can be treated as a high-dimensional categorical feature. One-hot encoding of words would lead to sparse vectors, similar to what we expect might happen when applied to claims data. NLP researchers developed various methods for entity embedding. Initially, bag-of-word methods where the frequency of occurrence of each word is counted, forming a numerical representation of the text, were developed Harris (1954). Newer standard methods such as Word2Vec improve upon the bag-of-word model by capturing semantic relationships between words through distributed representations in a continuous vector space Mikolov et al. (2013a). Unlike bag-of-word, Word2Vec can preserve word context, allowing for a more nuanced understanding of word meanings and similarity between words Goldberg & Levy (2014). Entity embedding has also been used in the healthcare context, e.g., for learning medical concepts such as diagnoses and treatments from electronic health records using categorical (Wu et al., 2021) or text data (Choi et al., 2016; Wang et al., 2019; Chowdhury et al., 2019).

Building on the promising results advanced entity embedding methods yield for NLP and the applications mentioned above, we investigate two research question in the health economic context:
1: *Is entity embedding of claims data a feasible and more efficient method than one-hot encoding?*
2: *Can entity embedding enhance the performance of downstream forecasting tasks?*

We will exploit data from a large Swiss health insurance to train our entity embedding model. Specifically, we will use continuous bag-of-words (CBOW) from the Word2Vec methodology to encode the medical history of insurees with Chronic Obstructive Pulmonary Disease (COPD). COPD is a progressive lung disorder characterized by persistent respiratory symptoms and airflow limitation, commonly caused by significant exposure to toxic gases or by smoking for Chronic Obstructive Lung Disease (GOLD). COPD is a major cause of morbidity and mortality worldwide, necessitating comprehensive management strategies to alleviate symptoms Vogelmeier et al. (2017); Lozano et al. (2012); of Disease Study 2020 (2018). Exacerbations, defined as acute worsening of respiratory symptoms, significantly contribute to disease morbidity and mortality. Exacerbations can be treated in outpatient settings for mild cases, but for severe cases a hospital stay is required. COPD patients will never fully recover from a hospitalization, and deterioration of their health status by an exacerbation is persistent Wedzicha & Seemungal (2007). Preventing hospitalized exacerbations will thus preserve patients' quality of life, life expectancy, and additionally will decrease costs for the healthcare system.

We aim at predicting hospitalizations using claims data. The results from our experiment showcase that entity embedding can be used to increase efficiency for training prediction models in a setting of claims data. With our approach, the outcome-relevant parts of an insuree's medical history are identified and used to predict individual hospitalization risks at two months, one month and two weeks prior hospitalization. Claims data have the advantage that they are always collected for all rendered health services and are usually available in a standardized form, underscoring the scalability and potential impact of our model.

2 Data and Methods

In the following sections we introduce our data and outline our methodology. For an overview see Figure 6 in the appendix.

2.1 Data source and observation period

We use anonymised claims data at insuree level of a Swiss health insurance with a 10% market share of the Swiss mandatory health insurance market. The observation period is from 2015 to 2020.
Our analyses specifically use the billing data part of the claims data. The billing data contain all interactions of the insurees with all service providers in the mandatory health insurance, including hospitals, outpatient physicians, other healthcare professionals (e.g., physiotherapists) and prescriptions. For each interaction, the treatment and billing date, the exact items as described in the respective tariff, and the amount in Swiss franks are recorded.

Inclusion and exclusion criteria, sample sizes

In Switzerland, health insurances are not allowed keep diagnose codes of their insurees in their records. Thus, we use prescribed specific COPD medications[1] as main inclusion criteria for identifying COPD patients in our dataset (see Figure 1) Bischof et al. (2024). As long- and short-acting COPD medications are also prescribed to asthma patients (Agustí et al., 2023), we exclude insurees below the age of 40 as asthma is more prevalent than COPD in this age group. This identification approach was also followed in other studies. This yields a sample of 173,914 insurees with 197,698,144 interactions (Sample A). We use Sample A to train our entity embedding model.

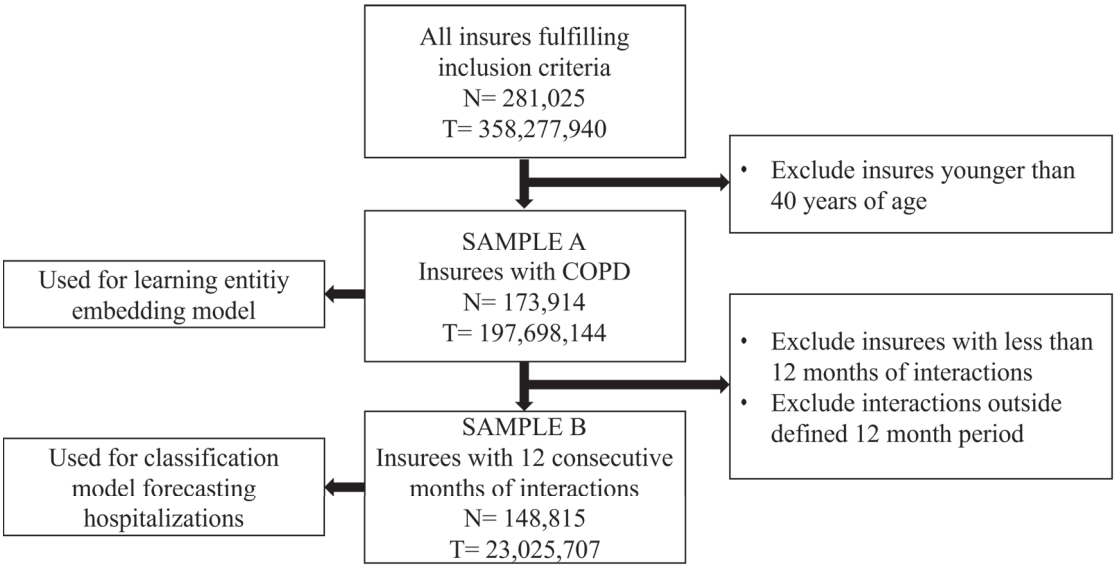

Figure 1: Construction of samples

Note: N stands for the number of insurees and T for the number of services provided to these insurees.

[1]The medications long-acting and short-acting beta2 agonists, long-acting and short-acting muscarinic antogonists and corticosteroids.

We predict hospitalized exacerbation which can be identified via DRG base code E65. We identify hospitalized exacerbations by the base Diagnosis-related Group (DRG) code E65. We train our prediction model with interactions from a period of 12 consecutive months per insuree. For insurees without hospitalization, we choose a random 12 month period from the available data to ensure that data are independent and identically distributed (iid). For insurees with hospitalization, we consider the 12 months prior hospitalization. Thus, for learning our prediction model, we need to exclude insurees with less than 12 months of interactions yielding a sample of 148'815 insurees with a total of 23'025'707 interactions (Sample B)[2].

2.2 Methods

Encoding Claims Data for Classification of Hospitalized Exacerbation

We aim to utilize all items within the available data set as features for the classification task of hospitalizations. We compare two different methodologies for encoding, namely one-hot encoding and entity embedding.

One-Hot Encoding

Dummification of categories in our dataset results in $63,998 - 1 = 63,997$ binary columns, in which the presence of a category is indicated by 1 and its absence by 0.

Entity Embedding via Word2Vec

As second encoding approach, we adjust the Word2Vec method to our research context. Word2Vec is a neural network model designed to learn distributed representations of words in a continuous vector space from large text corpora Mikolov et al. (2013b). There are several contributions that apply entity embedding and word2vec on categorical features (Waldemar et al., 2022; Russac et al., 2018; Wu et al., 2021). The core idea of Word2Vec is to capture sequential relationships between words, in our case items from the reimbursement catalogue, by predicting context items given a target item using the Continuous Bag-of-Words (CBOW) model. Training the model involves backpropagation and stochastic gradient descent (SGD) to update the input weight matrix W_{input} and output weight matrix W_{output} to minimize the negative log-likelihood loss function of an item based on the items in the neighbourhood Bengio et al. (2003); Mikolov et al. (2013a). Accordingly, the training process involves the following five steps:

1. **Input Layer**: One-hot encoded vectors represent each context item (the $63,998$ items in the reimbursement catalogue in our sample). These vectors are averaged to form the input vector x:

$$x = \frac{1}{2m} \sum_{i=1}^{2m} [w_{c_i}] \tag{1}$$

   w_{ci} is the i-th context item. These are the items surrounding the target word within the

[2]Note that the number of interactions per insuree is considerably lower for Sample B compared to Sample A as the latter includes all interactions of included insurees and not only interactions of 12 months.

window size m[3] (in our case we chose 25), and W_{input} is the input weight matrix. The input weight matrix W_{input} is a matrix of $V \times N$, where V is the item size (number of items in the reimbursement catalogue, $63,998$) and N is the embedding size. A heuristic rule for the embedding size[4] is the $\sqrt{V}$, in our case 252. It maps one-hot encoded input vectors of items to dense vector representations in a lower-dimensional space. Each row of W_{input} corresponds to an item in the data, and the row's entries represent the item's embedding. The transformation converts the items into continuous vector representations that capture relationships between the items. In our case, this procedure might identify that insurees that receive a specific service, for example a spirometry testing their lung capacity, have a higher likelihood that there is a change in COPD medication than that they are for example hospitalized for a hip replacement.

2. **Hidden Layer**: The input vector x is transformed to the hidden layer h using a linear transformation and an activation function f[5]:

$$h = f(W_{\text{input}}^T x) \tag{2}$$

The N-dimensional hidden layer h represents the dense vector obtained after transforming the averaged input vector x using the weight matrix W_{input}, where f is the activation function Bengio et al. (2003); Mikolov et al. (2013a). The activation function f determines how the input data is transformed at each layer of the neural network Le & Mikolov (2014).

3. **Output Layer**: The hidden layer h is projected to the output layer to obtain the unnormalized log probabilities for each item in the reimbursement catalogue.:

$$\hat{y} = W_{\text{output}} h \tag{3}$$

The unnormalized log probabilities capture the likelihood of observing each item given the item context, allowing for the calculation of item embeddings. This projection involves a linear transformation of the hidden layer weights by W_{output}, mapping the hidden layer representation to the output layer. These log probabilities provide insights into the contextual relationships between items and are crucial for learning meaningful item embeddings Mikolov et al. (2013a).

4. **Softmax Function**: The softmax function converts these log probabilities into a probability distribution over the entire vocabulary:

$$P(w_t | w_{c_1}, w_{c_2}, \ldots, w_{c_{2m}}) = \frac{e^{\hat{y}_{w_t}}}{\sum_{i=1}^{V} e^{\hat{y}_i}} \tag{4}$$

where V is the vocabulary size.

The softmax function converts unnormalized log probabilities into a probability distribution over the entire vocabulary. It takes the unnormalized log probabilities generated by the output

[3]The window size is the predefined neighbourhood considered on either side of the target item in the insuree time series. The window size parameter can be tuned, however this would be computationally too expansive.

[4]Increasing the embedding size will increase the performance of entity embedding for downstream tasks, but it will also increase runtime. $V = N$ would result in one-hot encoded vectors

[5]Gensim uses a sigmoid function.

layer and normalizes them to produce probabilities. This normalization ensures that the probabilities sum up to 1, representing a valid probability distribution. The softmax function is defined as an exponential function applied to the unnormalized log probabilities divided by the sum of exponentiated log probabilities. This transformation enables the model to output probabilities for each item in the vocabulary, facilitating tasks such as item prediction and classification Goldberg & Levy (2014).

The objective of the Word2Vec model is to maximize the log probability of observing the target item given its context items via SGD to minimize the negative log-likelihood loss. The model aims at predicting the target item w_t based on the surrounding context items w_{c_i}. Maximizing the log probability involves adjusting the model parameters $\mathbf{W}_{\text{input}}$ and $\mathbf{W}_{\text{output}}$ to improve the model's predictive performance Goldberg & Levy (2014).

5. **Objective Function**:

$$\max \log P(w_t | w_{c_1}, w_{c_2}, \ldots, w_{c_{2m}}) \tag{5}$$

The embedding vectors learned by the CBOW, which we use for our downstream tasks, are denoted by $\mathbf{v}_w$ and are the columns of the input weight matrix $\mathbf{W}_{\text{input}}$. CBOW is particularly efficient for large datasets and excels at learning representations for frequent items Mikolov et al. (2013a). For our downstream task of predicting hospitalized exacerbation, medication and specialist visits are more relevant and frequent in the given data. [6].
For the estimation of word2vec and the subsequent embedding vectors we use GENSIM 4.3.2 in Python 3.10.10.

Term Frequency-Inverse Document Frequency

Using the embedding vectors from section 2.2 results in an insuree matrix where each row indicates one received service. An approach to aggregate the insuree matrix to one row is a weighted mean of the embedding, where the weights are the so called Term Frequency-Inverse Document Frequency (TF-IDF). TF-IDF is based on the intuition that terms that appear frequently in a document but rarely across the entire document collection are more discriminatory and representative of the document's content (Sparck Jones, 1972; Christopher et al., 2008).
The Term Frequency (TF) of item w in an insuree's medical history d, denoted as $t(w, d)$, measures the frequency of occurrence of w within d. It is calculated as the ratio of the number of occurrences of w in medical history d, denoted by $n_{w,d}$, to the total number of items in d:

$$t(w, d) = \frac{n_{w,d}}{\sum_{w' \in d} n_{w',d}}$$

The Inverse Document Frequency (IDF) of w, denoted as $d^{-1}(w)$, quantifies the rarity of w across the entire collection of medical histories. It is calculated as the logarithm of the ratio of the number of insurees to the number of medical histories containing the term w:

$$d^{-1}(w) = \log\left(\frac{N}{\text{df}(w)}\right)$$

[6] An alternative could be the Skip-Gram model that is more effective for learning representations of infrequent items Goldberg & Levy (2014)

The TF-IDF score of a term w in a medical history d, denoted as $tfidf(t, d)$, combines the TF and IDF:

$$td^{-1}(w, d) = t(w, d) \times d^{-1}(w)$$

TF-IDF assigns higher weights to items that are frequent within the medical history of a certain insuree but rare across the collection of medical histories of all insurees. TF-IDF thus captures the discriminatory power of items in representing the content of medical histories.
Combining the entity embeddings ($\mathbf{v}_w$) and the TF-IDF via a weighted mean leads to insuree specific vectors, where $\mathbf{v}_w$ are the embedding vectors. The insuree vectors F serve as our input features for the gradient boosting trees.

$$F = \frac{\sum \mathbf{v}_w * td_w^{-1}}{\sum td_w^{-1}} \tag{6}$$

2.3 Evaluation

Evaluating entity embeddings and word2vec is challenging because there is no straightforward metric to measure their quality directly. One method is to use cosine distance, which assesses how similar two embeddings are based on their orientation in the vector space, providing insight into item similarities (Mikolov et al., 2013a). Alternatively, embeddings can be evaluated through downstream classification tasks by using them as features for a classifier. The performance of the classifier indicates the quality of the embeddings in capturing relevant information compared to alternative approaches, in our case one-hot encoding. To guarantee out-of-sample testing and generalizable performance, we apply 5-fold cross-validation before the entity embedding so that no data from the validation set can leak into the model through the entity embedding. Additionally, all observations with less than 12 months of observation periods were used in the entity embedding as long as they were not part of the validation set.

Cosine Distance

Cosine distance is a similarity measure used to quantify the similarity between two vectors by measuring the cosine of the angle between them, where $\mathbf{a} \cdot \mathbf{b}$ is the dot product of the vectors and $\|\mathbf{V_1}\|$ and $\|\mathbf{V_2}\|$ represent their Euclidean norms. Cosine distance ranges from 0 to 2, with 0 indicating identical direction, 1 indicating orthogonality, and 2 indicating opposite directions Mikolov et al. (2013a).

$$\text{CosineDistance}(\mathbf{v}_1, \mathbf{v}_2) = 1 - \frac{\mathbf{v}_1 \cdot \mathbf{v}_2}{\|\mathbf{v}_1\|\|\mathbf{v}_2\|} \tag{7}$$

In our context, two embedding vectors with a cosine distance of zero implies that these items appear within the same context items and can be interpreted as similar items.

Classification Algorithm

Gradient boosting is a powerful ensemble learning technique widely used for classification tasks. It operates by iteratively training a series of weak learners, in our case decision trees, with each subsequent learner focusing on the errors of its predecessors. This iterative process minimizes the overall prediction error (Chen & Guestrin, 2016; Hastie et al., 2009). In this study, we implement

the gradient boosting algorithm using the XGBoost (Extreme Gradient Boosting) framework due to its efficiency and effectiveness in handling large-scale datasets with high-dimensional features. XGBoost enhances traditional gradient boosting by incorporating regularization techniques to prevent overfitting and parallel processing capabilities for faster model training (Chen & Guestrin, 2016). We tuned for the models using one-hot encoded features and the entity embedded features learning rate (0.3, 0.15, 0.05), and tree depth (6, 4) and set the number of rounds to 2,000. The learning rate controls the contribution of each tree to the overall model, aiming at balancing between underfitting and overfitting. The tree depth determines the maximum depth of an individual tree, affecting the model's complexity and ability to capture intricate patterns. The number of rounds (or number of boosting iterations) specifies how many trees are sequentially added to the model. It directly relates to the learning rate because a lower learning rate typically requires more rounds to converge to an optimal solution, while a higher learning rate needs fewer rounds but risks overfitting. Adjusting both parameters helps balance model performance and computational efficiency (Chen & Guestrin, 2016).
The gradient boosting model was trained on a subset of Sample B using a 5-fold cross-validation strategy to ensure generalizability. During each fold, the model was trained on a portion of the data and evaluated on the remaining unseen data.[7]
For the estimation of gradient boosting trees we use XGBoost 2.0.3 on Python 3.10.10.

3 Results

3.1 Descriptives

Table 1 shows the descriptive statistics and data properties. In our dataset (Sample B), there are 4,029 insurees with and 145,714 insurees without a hospitalization, indicating a considerable class imbalance. Insurees with hospitalizations generally have more interactions[8] with the healthcare system (0.53 versus 0.15). This means that insurees who are later hospitalized have, on average, an interaction with the healthcare system every second day compared to every sixth or seventh day for those not hospitalized.

[7]The training data, and only the training data was used for the entity embedding and the training of the gradient boosting model.
[8]Interactions are individual receipts. A receipt can include multiple treatments

	Hospitalized Exacerbations	No Hospitalisation
Sample Size	4'029	145'714
Average number of interactions per day for 12 months	0.53 (0.63)	0.15 (0.16)
2 months	0.64 (0.72)	
1 month	0.70 (0.75)	
2 weeks	0.80 (0.80)	
1 week	0.95 (0.86)	
1 day	2.27 (1.41)	
Average number of days with interactions for 12 months	0.26 (0.24)	0.09 (0.09)
2 months	0.31 (0.26)	
1 month	0.33 (0.26)	
2 weeks	0.36 (0.27)	
1 week	0.42 (0.27)	
1 day	1.00 (0.00)	
Share of only LA	20.45%	15.43%
Share of only SA	10.72%	9.27%
Share of SA and LA	50%	5.48%
Share neither	18.71%	69.45%
Average (standard deviation) number of unique services	113.17 (74.25)	62.02 (41.82)
Number of zeros in one-hot encoded data	99.83%	99.91%

Note: In the column Hospitalized Exacerbations the periods are prior to the hospitalization. We only provide the average number of interactions and the days with interactions for insurees without hospitalisations only for the 12 month period, because it will not differ for the other period lengths, as the observation period is chosen randomly for the insurees without hospitalisations. LA stands for long-acting COPD medication and SA stands for short-acting COPD medication.

Table 1: Descriptive statistics and data properties

For hospitalized insurees, the number of interactions increases as the time of hospitalization approaches. On the day before hospitalization, hospitalized insurees have, on average, more than two interactions[9].

When a patient interacts with the healthcare system, they often have more than one interaction on the same day. This means that if a patient has one interaction, they are likely to have another interaction soon after. This pattern is likely because, after visiting a doctor, a patient might also buy medication. We also notice that patients in the hospitalized exacerbation group have at least one healthcare interaction every four days, compared to every ten days for those who are not hospitalized.

Moreover, the patients in the two classes differ in the type of medication they are taking. The rates of patients receiving only long-acting or only short-acting medication are relatively similar. However, half of insurees in the group of hospitalized exacerbations take both long- and short-acting medication compared to only 5.48% in the group without hospitalization. Additionally, 18.71% of insurees later hospitalized for exacerbations bought neither short- nor long-acting medication,

[9]As the Swiss outpatient tariff "TARMED" is a fee-for-service tariff one physician visit can include multiple interactions.

compared to 69.45% of those not hospitalized, indicating that insurees in the hospitalized group are already in a worse stage of COPD.
Furthermore, insurees in the exacerbation group receive almost twice the number of unique services, with 113.17 unique services compared to 62.02 for those not hospitalized.
Lastly, the share of zeros in the one-hot encoded data is similar between the two groups at a very high share of 99.83% and 99.91%.

3.2 Feature Encoding

One-hot encoding

Encoding the data as binary data results in a table with dimensions 139,743 x 63,998, where 99.91% of the values are zeros, as shown in Table 1. This leads to a CSV file size of 19.04 GB when using one-hot encoding compared to a 0.54 GB CSV file in the long format with dimensions 139,743 x 3 (the columns are patient identifier, date of treatment and item). Working with the one-hot encoded data requires significantly memory, and the encoding process takes in our case a bit more than one hour to transform the data from the long format to one-hot encoded data.

Entity Embedding

The entity embedding of the 225-dimensional vectors from the long data took roughly three hours (see Table 2). We visualize the trained embedding vectors of all items in the reimbursement catalogue in Figure 2. The vectors with a length of 225 are projected to a 2-dimensional space using principal component analysis (PCA). PCA transforms the entity embeddings into a set of orthogonal components that maximize the variance, effectively reducing the dimensionality while preserving as much information as possible (Jolliffe, 2002).

Step Description	OH	EE
Preprocessing	1.13	2.94
Gradient Boosting	14.27	3.30
Total	15.40	6.24

Table 2: Runtime in hours by step for one cross-validation fold

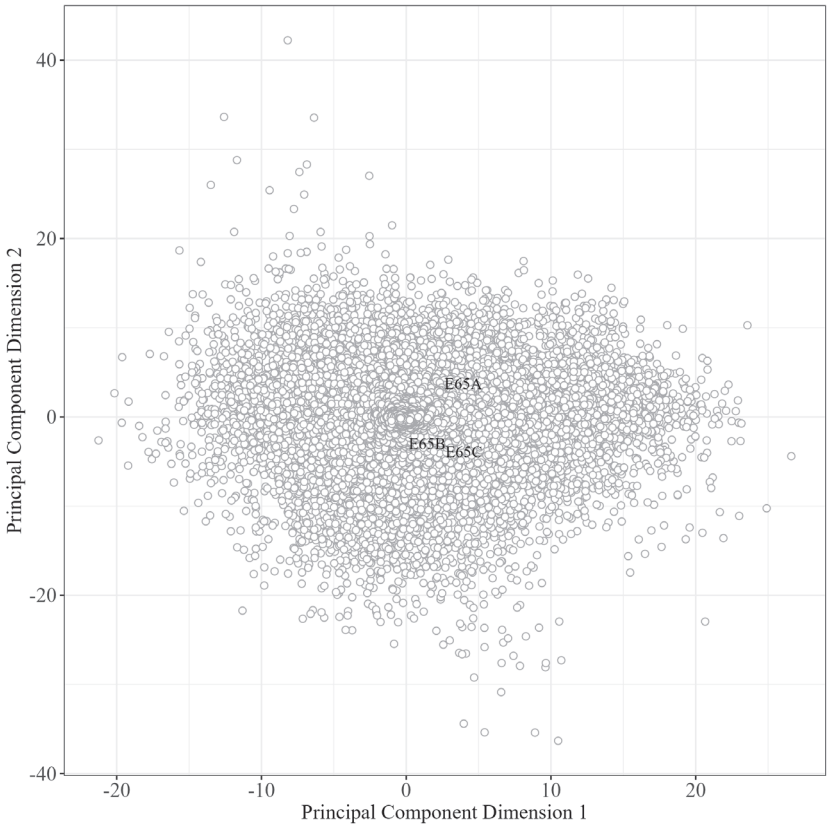

Figure 2: Two-dimensional projection of entity embeddings

We denote the positions of the DRG codes for hospitalized exacerbations (E65A, E65B, and E65C) in Figure 2 to show how close these items are to each other. In the first two principal components, there appear to be many other items closer than the hospitalized exacerbations. However, analyzing the 225-dimensional vectors using cosine distance reveals that for each of them, the two closest items are the other items for hospitalized exacerbations. Table 3 shows the cosine distance of the five closest items. The embedding model can learn that these three items and their respective treatments are similar without knowing they belong together.

	E65A	E65B	E65C
1	E65C (0.127)	E65C (0.125)	E65B (0.125)
2	E65B (0.131)	E65A (0.131)	E65C (0.127)
3	42.10.30.002 (0.252)	42.10.30.002 (0.261)	42.10.30.002 (0.253)
4	7680538440221 (0.264)	E40C (0.295)	E77F (0.264)
5	E40C (0.280)	E64C (0.305)	7680538440221 (0.311)

Note: E40C is the DRG code for "Diseases and disorders of the respiratory organs with ventilation > 24 hours." E64C is the DRG code for "Respiratory insufficiency, more than one day of hospitalization, without extremely severe CC, without pulmonary embolism, age > 9 years." E77F stands for "Other infections and inflammations of the respiratory organs without complex diagnosis, without extremely severe CC, without complicating procedure, age > 0 years, with severe CC or with para/tetraplegia." 7680538440221 is the GTIN code for a short-acting beta 2-agonist (SABA). 42.10.30.002 stands for "Mobile pressurized oxygen gas supply" provided by the Lungenliga, a Swiss non-profit health organization.

Table 3: Five closest items in cosine distance to base DRG E65 of hospitalized exacerbations

Using the weighted mean of the entity-embedded items from Section 2.2, we show the first two PCA dimensions of the entity-embedded insurees (see Fig. 3). There is some separability of insurees in the first two dimensions between insurees with a hospitalisation and without an hospitalisation. However a more sophisticated algorithm is required to classify insurees.

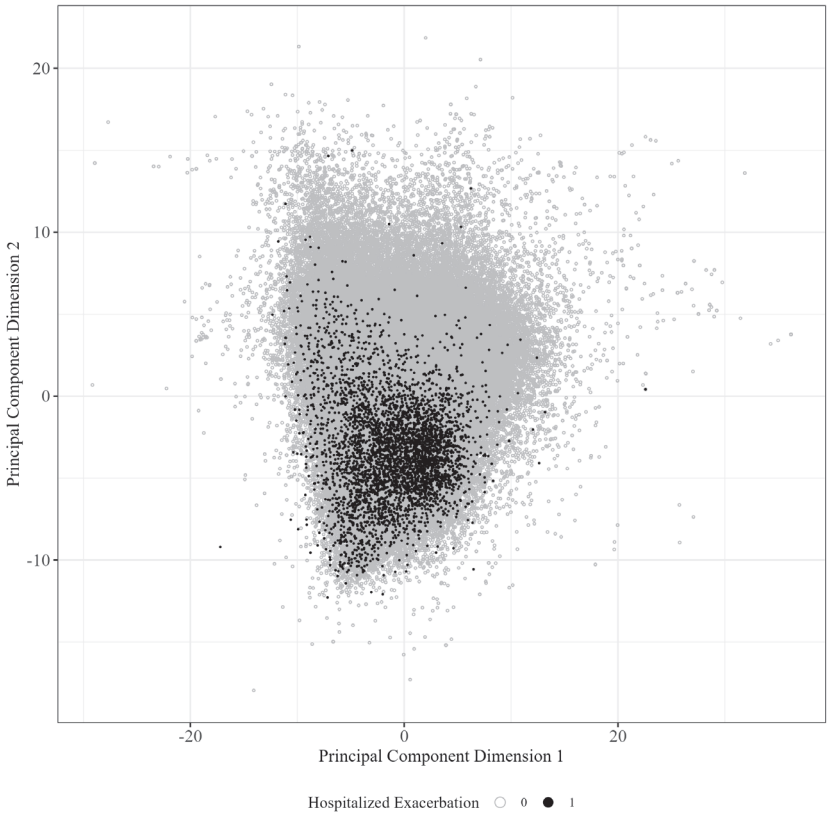

Figure 3: Two-dimensional projection of entity-embedded insurees

3.3 Classification Task

In this section, we examine the results of predicting hospitalized exacerbations at one month prior to a hospitalization, comparing one-hot encoded features and entity-embedded features via continuous bag-of-words. Table 7 in the appendix shows the 20 most important one-hot encoded features used in the gradient-boosted classification trees. The most used feature is an outpatient service determining the minimum erythema dose, meaning the tolerance of human skin to solar radiation. This service appears five times in the data, i.e., 0.003% of the insuree sample receive this service. Thus, this feature might in fact only be a statistical artifact which could lead to overfitting if used often by the gradient-boosted classification trees.
Table 4 summarizes the average performance metrics out of sample from 5-fold cross-validation, showing that entity embedding slightly outperforms one-hot encoding without considering runtime. Adjusting for runtime by reducing the number of rounds of the gradient-boosting classification tree would result in unusable results, because the model model is not able to generalize any predictions. Figure 4 shows the receiver-operating characteristic curve.

	EE	OH
Precision	0.14	0.13
Accuracy	0.86	0.85
Balanced Accuracy	0.85	0.84
Recall	0.84	0.83
FPR	0.14	0.15
FNR	0.16	0.17
F1	0.24	0.22
AUC	0.92	0.91

Table 4: Performance metrics

Note: EE stands for entity-embedding model and OH stands for one-hot encoded features.

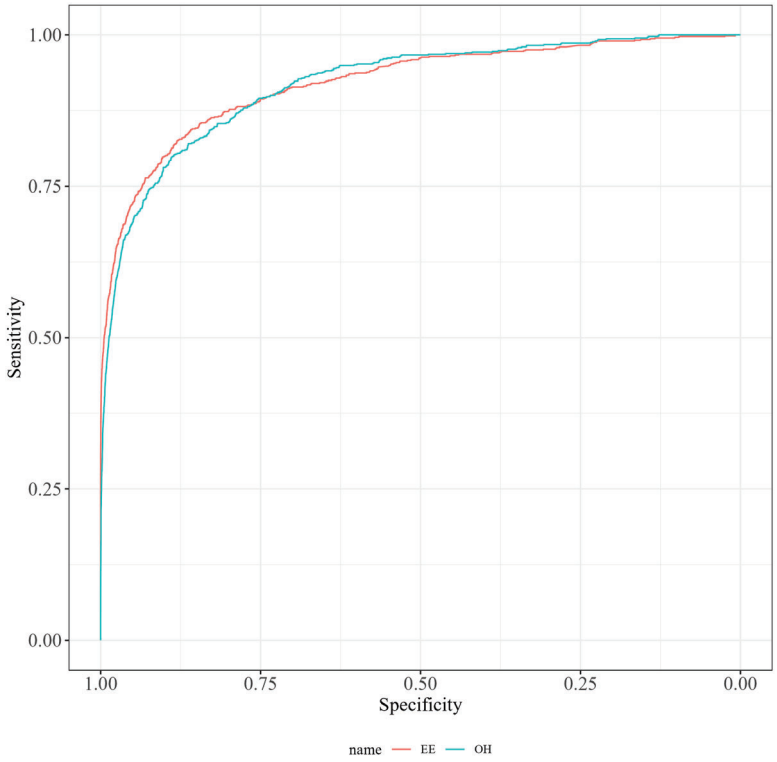

Figure 4: Receiver-operating characteristic curve

In a practical setting, the practitioner needs to specify a cut-off value for the classification to become an alert. This threshold can be set by aiming for a predefined recall. We set the recall at 0.95, 0.90, 0.85. Table 5 shows that entity embedding loses less in specificity than one-hot encoding when targeting high recall values.

recall	EE specificity	OH specificity
0.95	0.60	0.55
0.90	0.73	0.73
0.85	0.82	0.85

Table 5: Specificity with predefined recall

Note: We set the recall at 0.95, 0.90 and 0.85 and report the resulted specificity for both entity embeddded (EE) and one-hot encoded (OH) features

Furthermore, we trained the model using enity embedded features at three additional points in time in addition to one month prior to a potential hospitalisation: one week, two weeks, and two months prior to the hospitalisations. The specificity of predictions improves as the days decrease for a predefined recall of 0.9 and 0.95 (see Figure 5), due to the availability of more recent and relevant data. These results align with descriptive statistics as insurees utilise more services before an exacerbation (see Table 1).

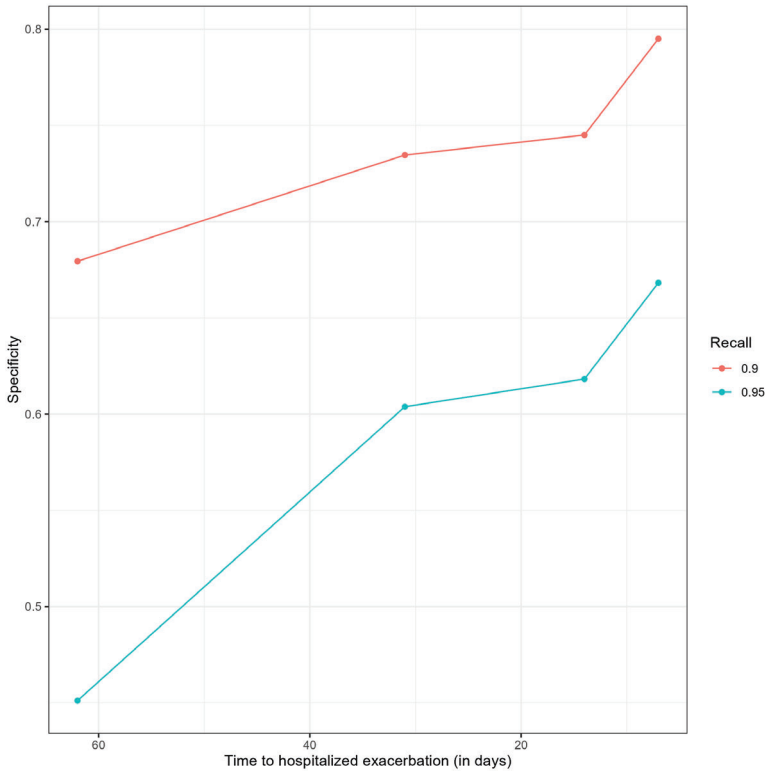

Figure 5: Specificity at 2 months, 1 month, 2 weeks, and 1 week prior to the hospitalized exacerbation

Table 6 shows the results for four subgroups based on whether the insurees had long-acting and/or

short-acting medication prescribed. The best performance is achieved by the subgroup receiving both short-acting and long-acting medication. The lowest performance is achieved by the subgroup not receiving either. This means that it is easier for the algorithm to predict a hospitalized exacerbation for patients in worse COPD stages.

	Only LA	Only SA	SA and LA	Neither
Precision	0.13	0.10	0.54	0.02
Accuracy	0.81	0.78	0.81	0.73
Balanced Accuracy	0.79	0.76	0.82	0.76
Recall	0.77	0.74	0.82	0.79
FPR	0.19	0.22	0.19	0.27
FNR	0.23	0.26	0.18	0.21
F1	0.22	0.17	0.65	0.03
AUC	0.87	0.84	0.90	0.82

Note: LA stands for long-acting COPD medication and SA stands for short-acting COPD medication.

Table 6: Performance Metrics for medication subgroups

4 Discussion

We propose an entity embedding approach based on the Word2Vec methodology for high-dimensional claims data that can significantly reduce dimensionality while retaining relevant insuree information and generalizing sparse categorical features. We apply this approach to a classification task that predicts hospitalised exacerbations in patients with COPD and compare it with standard one-hot encoding.

Our results demonstrate that entity embedding effectively captures the complexity of the interactions of received health services, which is reflected in the slightly better performance metrics compared to one-hot encoding. Specifically, entity embedding achieved an AUC of 0.92 and an F1 score of 0.24, while the one-hot encoded model achieved an AUC of 0.91 and an F1 score of 0.22. Additionally, entity embedding leads to a model that is more robust to class imbalance and captures the nuances of the patient's healthcare journey better. Furthermore, it significantly decreases the overall runtime from 15.40 hours to 6.24 hours. An additional advantage would be that if a foundation model based on all insurees is trained and then only applied to a specific task the preprocessing step would be even shorter.

The entity embedding approach allowed us to visualize and analyze the proximity of different healthcare services and medications. The embeddings for DRG codes related to hospitalized exacerbations (E65A, E65B, E65C) were found to be close to each other, indicating that the model effectively clusters similar medical events. This clustering is confirmed by the cosine distance analysis, which shows that for each exacerbation-related DRG code, the closest items are other exacerbation-related codes and health services.

When comparing one-hot encoded features to entity-embedded features in gradient-boosted classification trees, we found that entity embedding slightly outperforms one-hot encoding in terms of predictive accuracy. Moreover, the one-hot encoded model showed a propensity for overfitting, evidenced by the importance of rarely occurring features, such as the outpatient service for determining the minimum erythema dose.

In practical applications, setting a high recall (e.g., 0.95) for the classification task, entity embedding demonstrated better specificity compared to one-hot encoding, suggesting it is more effective in reducing false positives and according false positive alarms. The performance of the model also improved as the prediction horizon shortened, from two months to one week before hospitalization, due to the availability of more recent and relevant data.
Subgroup analysis based on medication patterns revealed that the model performs best for insurees taking both short-acting and long-acting medications, reflecting the greater predictability in patients with more complex treatment regimens. On the other hand, the lowest performance was observed in the subgroup not receiving either type of medication, indicating the challenge of predicting exacerbations in patients with less severe or less actively managed COPD.
We observed that hospitalized insurees have significantly more interactions with the healthcare system compared to those not hospitalized. This increase in healthcare interactions as the hospitalization date approaches highlights the potential for early intervention which could be enabled by continuous risk prediction using our approach.

4.1 Findings from the literature

Word2Vec has been extensively validated in the context of NLP. By extending this concept to categorical data, researchers have successfully applied entity embedding to various domains, demonstrating substantial improvements in model performance. Yi et al. (2022) provides an overview of current open-source methodologies and applications focusing on statistics and drug discovery. Dahouda & Joe (2021) applies entity embedding, specifically TF-IDF, with fewer features than in our study, which also results in a higher F1 score and requires less memory. Wu et al. (2021) showcases ME2Vec, which uses hierarchical entity embedding of medical professionals, services, and patients with 394 and 3,157 unique services in their two applications.

4.2 Limitations

Our study has limitations that stem from the data, the chosen methodology for entity embedding, the classification learner, and the feasibility of implementation. Regarding the data, we might not have labeled all hospitalized exacerbation cases, as we find that there are DRG codes with very similar embeddings to the base DRG E65, all related to lung diseases (e.g., E40C, E64C, E77F).
Furthermore, the entity embedding is limited to item-level representations, struggling with items that are very rare in the sample or cannot generalize for items that do not appear in the training sample. The model is insensitive to syntax, treating items as individual tokens without fully capturing the order of services but only their neighborhoods. Static embeddings can also suffer from drift, not reflecting changes in item meanings over time, for example, if the exact definition of an item changes through revision of the reimbursement system.
Additionally, the embeddings are trained on insurees with COPD and might not be generalizable for insurees without COPD. Lastly, advancements such as bi-directional and transformer methodologies could improve entity embedding and address the mentioned shortcomings. However, this is outside the scope of our study and would require more data.
Even though hyperparameter tuning is not a significant component for gradient boosting trees, we could only tune a limited number of hyperparameters. When hyperparameters like learning rate, number of trees, and tree depth are not optimized, the model can easily overfit or underfit. A too-high learning rate can cause the model to miss the optimal solution, while a too-low rate

can make training excessively slow. Insufficiently deep trees may fail to capture complex patterns, whereas overly deep trees can overfit the training data. Too few boosting iterations can result in underfitting, and too many can exacerbate overfitting. Additionally, inadequate hyperparameter tuning can lead to increased training times and computational costs without corresponding gains in model performance. The sensitivity to hyperparameters requires cross-validation and extensive computational resources to identify the optimal settings. We trained our model on a limited set of hyperparameters, which could be extended.
Finally, there are limitations related to the implementability of the prediction model: Predicting hospitalized exacerbations identifies patients at risk but does not inherently provide means to prevent it, as the prediction itself does not suggest specific interventions and also does not provide a likelihood of preventing a hospitalized exacerbation. Additionally, due to the time lags in reporting receipts to the insurances, they can only update the predictions once they receive the new receipts. This would lead to delayed alerts that might not prevent any hospitalizations, as they might have already happened. A sensitivity analysis using available receipts at the respective points in time to account for the operational inefficiencies could tackle this issue.

5 Conclusion

We use a simple Word2Vec methodology on items in the Swiss reimbursement system without extensive hyperparameter tuning to forecast hospitalized exacerbations. We find that using entity embedding instead of one-hot encoding slightly improves the performance metrics of our classification model. Additionally, the one-hot encoded approach requires almost triple the runtime. These results indicate that entity embedding not only captures the dynamics of medical events but also improves the efficiency of training predictive models, highlighting its potential for broader application in healthcare analytics.
In a practical setting, using our prediction approach is easily implementable due to the standardized data structure in a reimbursement system. This could enable health insurances to become an active guide for insurees monitoring their health risks to cooperatively work on minimizing health status deterioration.

References

Agustí, A., Celli, B. R., Criner, G. J., Halpin, D., Anzueto, A., Barnes, P., Bourbeau, J., Han, M. K., Martinez, F. J., Montes de Oca, M., et al. (2023). Global initiative for chronic obstructive lung disease 2023 report: Gold executive summary. *American journal of respiratory and critical care medicine*, 207(7), 819–837.

Bengio, Y., Ducharme, R., Vincent, P., & Jauvin, C. (2003). A neural probabilistic language model. *Journal of Machine Learning Research*, 3, 1137–1155.

Bischof, A. Y., Cordier, J., Vogel, J., & Geissler, A. (2024). Medication adherence halves copd patients' hospitalization risk–evidence from swiss health insurance data. *npj Primary Care Respiratory Medicine*, 34(1), 1.

Blanchet, F. G., Legendre, P., & Borcard, D. (2008). Forward selection of explanatory variables. *Ecology*, 89(9), 2623–2632.

Chen, T. & Guestrin, C. (2016). Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining* (pp. 785–794).

Choi, E., Bahadori, M. T., Searles, E., Coffey, C., Thompson, M., Bost, J., Tejedor-Sojo, J., & Sun, J. (2016). Multi-layer representation learning for medical concepts. In *proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining* (pp. 1495–1504).

Chowdhury, S., Zhang, C., Yu, P. S., & Luo, Y. (2019). Med2meta: Learning representations of medical concepts with meta-embeddings. *arXiv preprint arXiv:1912.03366*.

Christopher, D., Raghavan, P., Schütze, H., et al. (2008). Scoring term weighting and the vector space model. *Introduction to information retrieval*, 100, 2–4.

Dahouda, M. K. & Joe, I. (2021). A deep-learned embedding technique for categorical features encoding. *IEEE Access*, 9, 114381–114391.

De Pietro, C., Camenzind, P., Sturny, I., Crivelli, L., Edwards-Garavoglia, S., Spranger, A., Wittenbecher, F., Quentin, W., Organization, W. H., et al. (2015). Switzerland: health system review.

Efroymson, M. A. (1960). Multiple regression analysis. *Mathematical methods for digital computers*, (pp. 191–203).

for Chronic Obstructive Lung Disease (GOLD), G. I. (2023). Global strategy for the diagnosis, management, and prevention of chronic obstructive pulmonary disease 2023 report.

Goldberg, Y. & Levy, O. (2014). word2vec explained: Deriving mikolov et al.'s negative-sampling word-embedding method. *arXiv preprint arXiv:1402.3722*.

Guo, C. & Berkhahn, F. (2016). Entity embeddings of categorical variables. *arXiv preprint arXiv:1604.06737*.

Harris, Z. S. (1954). Distributional structure. *Word*, 10(2-3), 146–162.

Hastie, T., Tibshirani, R., Friedman, J., Hastie, T., Tibshirani, R., & Friedman, J. (2009). Boosting and additive trees. *The elements of statistical learning: data mining, inference, and prediction*, (pp. 337–387).

Jolliffe, I. T. (2002). *Principal component analysis for special types of data*. Springer.

Kuhn, M. & Johnson, K. (2019). *Feature engineering and selection: A practical approach for predictive models*. Chapman and Hall/CRC.

Le, Q. V. & Mikolov, T. (2014). Distributed representations of sentences and documents. In *Proceedings of the 31st International Conference on Machine Learning (ICML-14)* (pp. 1188–1196). Beijing, China.

Lozano, R., Naghavi, M., Foreman, K., Lim, S., Shibuya, K., Aboyans, V., Abraham, J., Adair, T., Aggarwal, R., Ahn, S. Y., et al. (2012). Global and regional mortality from 235 causes of death for 20 age groups in 1990 and 2010: a systematic analysis for the global burden of disease study 2010. *The Lancet*, 380(9859), 2095–2128.

Mikolov, T., Chen, K., Corrado, G., & Dean, J. (2013a). Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.

Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., & Dean, J. (2013b). Distributed representations of words and phrases and their compositionality. In *Advances in Neural Information Processing Systems*, volume 26 (pp. 3111–3119). Lake Tahoe, NV.

Murphy, K. P. (2012). *Machine Learning: A Probabilistic Perspective*. Cambridge, MA: MIT Press.

of Disease Study 2020, G. B. (2018). Global, regional, and national age-sex-specific mortality for 282 causes of death in 195 countries and territories, 1980–2017: a systematic analysis for the global burden of disease study 2017. *The Lancet*, 392(10159), 1736–1788.

Rencher, A. C. & Pun, F. C. (1980). Inflation of r2 in best subset regression. *Technometrics*, 22(1), 49–53.

Russac, Y., Caelen, O., & He-Guelton, L. (2018). Embeddings of categorical variables for sequential data in fraud context. In *International conference on advanced machine learning technologies and applications* (pp. 542–552).: Springer.

Sparck Jones, K. (1972). A statistical interpretation of term specificity and its application in retrieval. *Journal of documentation*, 28(1), 11–21.

Tibshirani, R. (1996). Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 58(1), 267–288.

Vogelmeier, C. F., Criner, G. J., Martinez, F. J., Anzueto, A., Barnes, P. J., Bourbeau, J., Celli, B. R., Chen, R., Decramer, M., Fabbri, L. M., et al. (2017). Global strategy for the diagnosis, management, and prevention of chronic obstructive lung disease 2017 report. gold executive summary. *American journal of respiratory and critical care medicine*, 195(5), 557–582.

Waldemar, H., Martin, S., & Markus, W. (2022). Word2vec embeddings for categorical values in synthetic tabular generation. In *2022 International Conference on Computational Science and Computational Intelligence (CSCI)* (pp. 613–622).: IEEE.

Wang, Y., Xu, X., Jin, T., Li, X., Xie, G., & Wang, J. (2019). Inpatient2vec: Medical representation learning for inpatients. In *2019 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)* (pp. 1113–1117).: IEEE.

Wedzicha, J. A. & Seemungal, T. A. R. (2007). Copd exacerbations: defining their cause and prevention. *The Lancet*, 370(9589), 786–796.

Wilkinson, L. & Dallal, G. E. (1981). Tests of significance in forward selection regression with an f-to-enter stopping rule. *Technometrics*, 23(4), 377–380.

Wu, T., Wang, Y., Wang, Y., Zhao, E., & Yuan, Y. (2021). Leveraging graph-based hierarchical medical entity embedding for healthcare applications. *Scientific reports*, 11(1), 5858.

Yi, H.-C., You, Z.-H., Huang, D.-S., & Kwoh, C. K. (2022). Graph representation learning in bioinformatics: trends, methods and applications. *Briefings in Bioinformatics*, 23(1), bbab340.

Zheng, A. & Casari, A. (2018). *Feature engineering for machine learning: principles and techniques for data scientists*. " O'Reilly Media, Inc.".

A Appendix

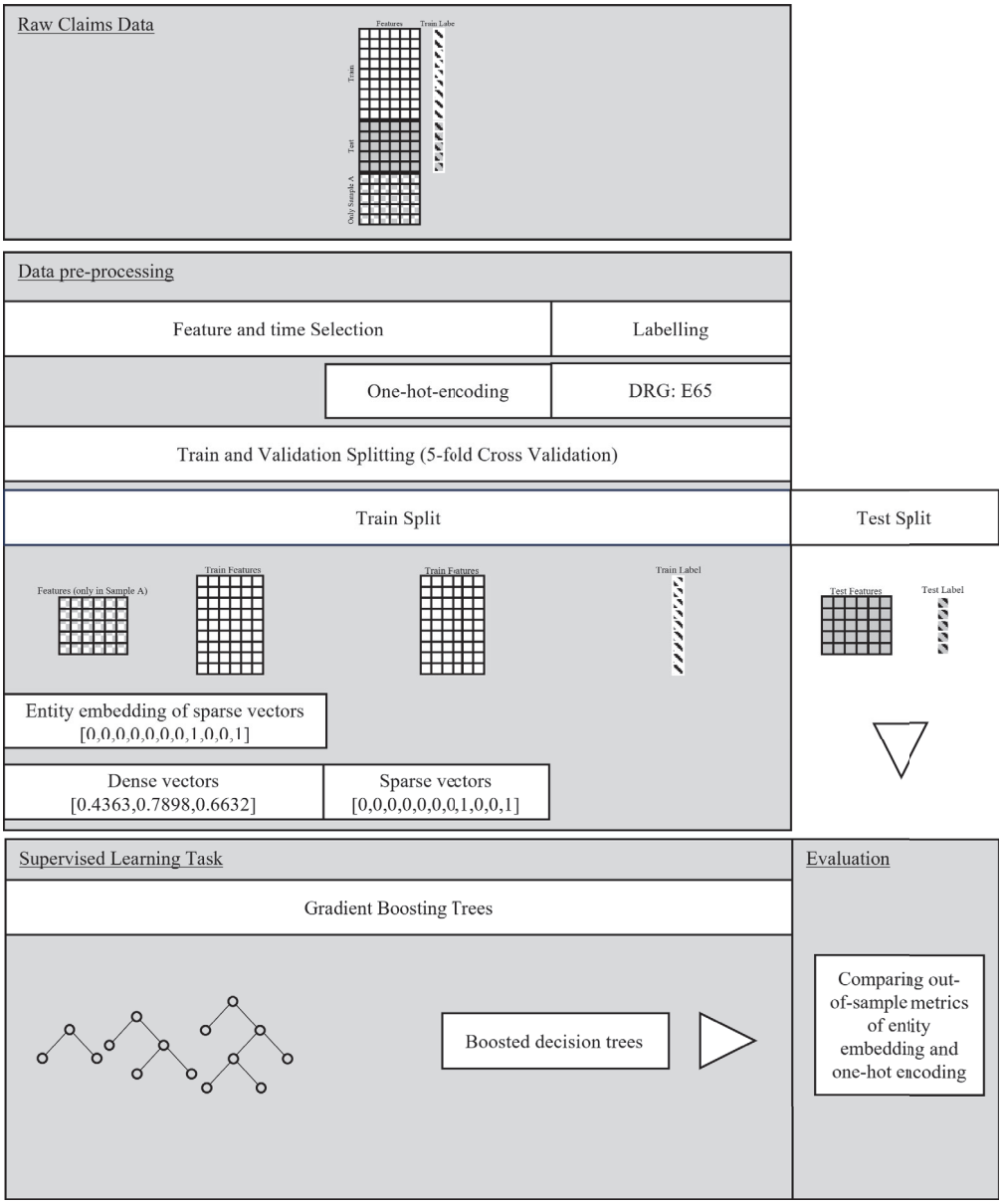

Figure 6: Model development

Note: This figure shows an overview of our model development

	Feature	Frq	Feature	Frq	Feature	Frq	Feature	Frq	Feature	Frq
1	001_04.0160	0.02	001_04.0160	0.02	001_04.0160	0.02	001_04.0160	0.02	001_04.0160	0.02
2	400_2107475	0.02	400_2107475	0.02	400_6089337	0.01	400_2107475	0.01	400_2107475	0.02
3	400_6089337	0.01	400_6089337	0.01	400_2107475	0.01	400_6089337	0.01	400_6089337	0.01
4	400_3037972	0.01	940_7680007020022	0.01	940_7680007020022	0.01	940_7680007020022	0.01	940_7680007020022	0.01
5	940_7680007020022	0.01	400_8109023	0.01	400_3037972	0.01	400_3037972	0.01	400_8109023	0.01
6	400_8109023	0.01	001_1212.00	0.01	400_3938422	0.01	010_B70J	0.01	400_3938422	0.01
7	317_4707.10	0.01	400_3037972	0.01	400_8109023	0.01	400_8109023	0.01	001_09.0120	0.01
8	400_4367447	0.01	400_4367447	0.01	010_B70J	0.01	001_09.0120	0.01	010_B70J	0.01
9	400_3938422	0.01	001_09.0120	0.01	001_09.0120	0.01	001_1212.00	0.01	400_4367447	0.01
10	001_1212.00	0.01	400_3938422	0.01	400_4367447	0.01	400_5462490	0.01	400_3037972	0.01
11	001_09.0120	0.01	532_53202	0.01	001_1212.00	0.01	400_4367447	0.01	001_1212.00	0.01
12	400_5462490	0.01	402_7680504960234	0.01	400_5853782	0.01	400_5853782	0.01	402_7680504960234	0.01
13	400_5283534	0.01	400_2062471	0.01	400_5462490	0.01	317_4707.10	0.01	400_5462490	0.01
14	400_7096724	0.01	400_7096724	0.01	940_41500	0.01	400_5127432	0.01	400_7096724	0.01
15	010_B70J	0.01	010_B70J	0.01	402_7680552740055	0.01	400_3938422	0.01	317_1731.00	0.01
16	402_7680552740055	0.01	402_7680552740055	0.01	400_7096724	0.01	402_7680552740055	0.01	400_2062471	0.01
17	400_5853782	0.01	400_1351210	0.01	402_7612449131257	0.01	400_7096724	0.01	317_4707.10	0.01
18	402_7680504960234	0.01	400_5853782	0.01	400_5283534	0.01	402_7680504960234	0.01	400_0039792	0.01
19	400_0694853	0.01	940_41500	0.01	317_4707.10	0.01	400_2062471	0.01	940_40020	0.01
20	402_7612449131257	0.01	001_15.0285	0.00	402_7680504960234	0.01	001_00.0410	0.01	400_5283534	0.01

Table 7: Top 20 one-hot encoded features by frequency (Frq) from gradient boosting

Note: The table shows how often a features is used as a splitting features in relative terms in gradient boosting for the one-hot encoded features for each of the five cross-validation splits. The most often used features are consistent over the splits, but no single item seems to be a good single predictor.