

## Research and implementation of the medical text analysis algorithm for predicting mortality

Zhenisgul Rakhmetullina<sup>1</sup>, Saule Belginova<sup>2</sup>, Alibekkyzy Karlygash<sup>1</sup>,  
Aigerim Ismukhamedova<sup>3</sup>, Shynar Tezekpaeva<sup>1</sup>

<sup>1</sup>Department of Engineering Mathematics, D. Serikbayev East Kazakhstan Technical University, Ust-Kamenogorsk, Kazakhstan

<sup>2</sup>Department of Information Technology, University Turan, Almaty, Kazakhstan

<sup>3</sup>Digitalization Center, Kazakh-American Free University, Oskemen, Kazakhstan

### Article Info

#### Article history:

Received Dec 4, 2023

Revised Feb 22, 2024

Accepted Mar 10, 2024

#### Keywords:

Analysis algorithm

Data mining

Diagnosis

Disease prediction

Logistic regression

### ABSTRACT

Mortality prediction has a role to play in the development of a descriptive measure of the quality of care that provides a fair and equitable means of comparing and evaluating hospitals. This article describes a study of a medical text analysis algorithm for mortality prediction that used big data in the form of unstructured medical notes. The article describes the concept of using text mining technology for medical systems, a method for preprocessing medical data to predict patient mortality, an algorithm for predicting patient deaths based on the logistic regression classifier and presents a software module for implementing the proposed algorithm.

This is an open access article under the [CC BY-SA](https://creativecommons.org/licenses/by-sa/4.0/) license.



### Corresponding Author:

Saule Belginova

Department of Information Technology, University Turan

Almaty, Kazakhstan

Email: sbelginova@gmail.com

## 1. INTRODUCTION

In recent years, the healthcare sector has seen exponential growth in digital data generation, especially in the form of electronic medical records (EHRs) and clinical notes. The vast amount of unstructured textual information contained in these records contains invaluable information that, when used effectively, can significantly improve patient care and clinical decision-making. One of the most important applications of this data is to predict patient treatment outcomes, especially mortality. The ability to accurately predict mortality is of paramount importance in the field of medicine, as it allows health professionals to identify high-risk patients in advance and carry out timely interventions. While traditional methods rely on structured clinical data, the rich and detailed information contained in the unstructured text provides an attractive opportunity for more detailed and accurate predictions.

This article is devoted to the research and implementation of a reliable algorithm for analyzing medical texts designed specifically for predicting mortality. Using the logistic regression classifier, our algorithm aims to extract significant patterns and insights from the extensive corpus of texts in EHRs. By analyzing the descriptions presented in clinical notes, the algorithm seeks to identify subtle indicators and risk factors associated with mortality that may be overlooked by conventional methods. The goals of this study include the development of a complex algorithm capable of coping with the complexities of a medical text, optimizing its performance by applying a logistic regression classifier on medical datasets MIMIC-III, which is a freely available database with unidentified data on the health status of about 50,000 patients in

Boston hospitals. Through this integrated approach, we aim to contribute to ongoing efforts to improve the predictive capabilities of healthcare systems, which will ultimately lead to improved patient outcomes and more efficient allocation of resources.

This work examines the current situation with mortality forecasting in healthcare, highlights the limitations of existing methods and the potential benefits offered by advanced text analysis methods. Next, we delve into the methodology underlying our algorithm, describing in detail the key components and the methods used. Then we present the results of our algorithm on the corresponding datasets, comparing its results with existing models. Finally, we discuss the implications of our findings, potential areas for further research, and the practical application of our algorithm in real clinical settings.

## 2. THE PROPOSED METHODS FOR MEDICAL SYSTEMS

The relevance of mortality forecasting can be considered at two main levels. At the individual level, mortality prognosis seeks to provide an objective risk assessment for clinical decision-making. Doctors, armed with quantitative estimates of the probability of death, can prioritize patient care, prepare targeted action plans, and offer informed prognoses. The probability of mortality can also serve as an approximate measure of the effectiveness of treatment. The prognosis can further be utilized to assess the suitability of patients for new treatments. Moreover, the probability of a fatal outcome provides doctors, patients, and/or their families with reasons to decide on the aggressiveness of treatment or the refusal of life support.

At the organizational level, the probability of mortality is used to quantify the severity of diseases when allocating resources. Given that intensive care units are significant cost centers, considering the probability of mortality becomes essential to ensure efficient resource allocation. Mendez-Tellez and Dorman [1], identifying patients who may not survive their stay in the department can lead to significant cost savings.

The risk-adjusted death rate (RAMR) is the death rate adjusted for the predicted risk of death. It is commonly used to observe and/or compare the performance of certain institutions or individuals, such as hospitals or surgeons. Thus, mortality prediction plays a role in developing a descriptive measure of the quality of care, providing a fair and equitable means of comparing and evaluating hospitals.

Recently, text analysis tools have been used in healthcare research; for example, Cerrito and Cerrito [2] analyzed electronic medical records from a hospital emergency department over a six-month period using text analysis. They found that such complaints were dealt with differently depending on the on-call doctor. Such differences can impact the quality and cost of medical care. Thus, text analysis of prior expert treatment can provide on-call physicians with an optimized treatment plan. It may also lead to the development of protocols to address treatment disparities.

The use of advanced analytics for quantitative and qualitative research of clinical and diagnostic data has the potential to uncover hidden medical knowledge by identifying correlations, causal relationships, and connections between seemingly independent variables. Consequently, the introduction and application of data mining methods in the modern healthcare system is gradually expanding. In this context, Jadhao [3] the authors discuss various disciplines, methods, models, algorithms and results, emphasizing how these methods can contribute to a variety of studies, including, but not limited to, long-term prospective and retrospective studies, population studies, correlation studies, multicenter and multiracial studies, step-by-step studies, meta-analysis, pharmacovigilance and much more. The classification of models for predicting mortality in the medical field can be broadly categorized into several types, depending on the methodology and data used see in Table 1. Here are some common classifications.

It is important to note that the choice of model depends on factors such as the nature of the data, interpretability requirements, computational resources, and the specific goals of the mortality prediction task. Additionally, model performance should be evaluated using appropriate metrics and validated on independent datasets to ensure generalizability. The use of data mining in medicine is of great interest for the future of the medical industry. Despite several obstacles, such as the complex nature of medical data, there are many successful examples of its application to date. The potential inherent in deep data analysis can address various problems not only in the field of medicine but also in all spheres of life.

Text mining refers to the discovery of knowledge from text data. Text contains abundant qualitative information that is challenging to use in statistical modeling. In the field of healthcare, doctors express their opinions in words containing useful information not captured elsewhere. This information can be further used to develop intelligent models to improve the healthcare process. However, traditional model building requires quantitative, tangible information. Text mining converts text into numeric form, enabling its use for analysis. Several text mining algorithms are available, suitable for various problem areas. This method is widely used in the fields of sociology and communication to extract intangible information hidden in words. The question is whether text mining can enhance the quality of healthcare.

Table 1. Classification of models for predicting mortality

Classification	Model	Description
Traditional statistical models	Logistic regression	Widely used for binary classification tasks, including mortality prediction.
	Cox proportional hazards model	Suitable for survival analysis, considering the time until an event (e.g., death) occurs.
Machine learning models	Decision trees	Constructed based on a series of decisions, which can be interpretable.
	Random forest	Ensemble of decision trees for improved accuracy and robustness.
	Support vector machines (SVM)	Useful for binary classification, separating classes with a hyperplane.
Deep learning models	Naive bayes	Assumes independence among features and is efficient for large datasets.
	Neural networks	Multi-layered models capable of learning complex patterns
	Recurrent neural networks (RNN)	Suitable for sequential data, capturing temporal dependencies.
	Long short-term memory (LSTM) networks	A type of RNN designed to address the vanishing gradient problem.
Ensemble models	Transformer models	Effective for processing sequential data and capturing long-range dependencies.
	Boosting algorithms (e.g., AdaBoost, gradient boosting)	Combine weak learners to form a strong classifier.
Rule-based models	Ensemble of neural networks	Combine multiple neural networks to enhance predictive performance.
	Expert systems	Incorporate domain knowledge and rules defined by experts.
Hybrid models	Fuzzy logic systems	Handle uncertainty by allowing partial membership to different classes.
	Combination of statistical and machine learning models	Integrating the strengths of both approaches for improved performance.
Survival analysis models	Kaplan-meier estimator	Non-parametric estimator for survival functions.
Time series models	Accelerated failure time (AFT) models	Parametric models for survival analysis.
	Autoregressive integrated moving average (ARIMA)	Suitable for time-series mortality data.
Text-based models	Exponential smoothing methods	Capture trends and seasonality in time-series data.
	Natural language processing (NLP) models	Analyze medical text data for mortality prediction.
Clinical scoring systems	Topic modeling	Identify key topics in medical records related to mortality.
	Sequential organ failure assessment (SOFA)	Assesses the performance of several organ systems.
	Acute physiology and chronic health evaluation (APACHE)	Predicts the risk of mortality based on severity of illness.

The general strategy for building predictive models supplemented by text analysis is as follows:

- Collection of documents;
- Initial data analysis using standard “stop lists”;
- Creation and improvement of “start lists”;
- Restart and clustering;
- Interpretation of clusters;
- Association with numerical values;
- Building predictive models using data mining.

The process of text mining begins with the collection of documents to be analyzed. Domain knowledge plays a vital role in extracting knowledge from text. The field expert decides on the criticality of word occurrence. Extracting and cleaning data is a time-consuming process that requires the close attention of subject matter experts to ensure the validity of the data and completeness of information.

Most text mining tools provide two analysis options: ignore frequently used terms in your analysis or analyze against an exclusive list of terms. The tool we used allows you to create start and stop lists. The stop list contains terms that should be ignored in the document when performing analysis, for example, frequently occurring terms such as “of,” “on,” “the,” which are not of great value, are ignored. This provides a more reliable analysis of term occurrence. After the initial run is completed, the results show all the terms that occur in the document set, along with their frequency and relationship. This association means that terms appear at the same time; for example, if the term “dyspnea” always appears in a document with the term “heart attack,” it indicates that patients have these common symptoms.

An effective way to conduct analysis is to use a start list. The initial list restricts the analysis to a specific list of terms. For example, if we are only studying the relationship between smoking and cancer,

we can create a list of smoking-related terms and cancer-related terms and perform the analysis only under these conditions. After the analysis is completed, it is possible to cluster documents based on the similarity (or difference) of the terms they contain. The cluster identification number and distance between clusters provide useful numerical data that can be used to represent textual information in the traditional model-building process. A simulation approach was chosen to simulate the process of risk formation, risk management and decision-making. Simulation models allow you to take into account many variables, such as age, gender, medical history, lifestyle, and other factors that affect the likelihood of mortality. A graphical model explaining the process of formation of management risks is considered in work [4].

### 3. METHOD

#### 3.1. Literature review

In today's scenarios many healthcare decisions are being taken by predictive modeling and machine learning techniques [5]. Medical information systems, in fact, represent a complex monitoring system that takes into account various aspects, criteria and management goals, which contributes to improving the efficiency of management of distributed facilities. The authors of the paper consider software that is used to track distributed objects based on a multi-level and multidimensional model [6]. For a better data analysis in healthcare, we need to understand the concept of logistic regression as well as others terms, which are linked with it. So that we can clearly understand the concept behind it and implement in medical research [7], [8].

The use of logistic regression has only recently been made possible with the widespread availability of microcomputers and statistical computing packages, and as a result, logistic regression models are now commonly used in biomedical research for modeling a dichotomous response variable as a function of a set of explanatory variables [9]. The goal of logistic regression is to create an equation that can be used to estimate the probability of an event of interest for the dependent outcome based on one or more independent variables [10]. Detailed descriptions of the application of logistic regression for processing medical data and its theoretical foundations are considered in the works [11]–[18]. Goodman [19] an epidemiologic framework using odds ratios is applied to the interpretation of logistic regression model estimates. Hasija and Chakraborty [20] used the logistic regression to classify heart diseases and it is implemented in Python using the Scikit-learn library. Rowe [21] describes how logistic regression can be used to distinguish factors that genuinely affect an outcome from those that are merely confounded. A key part of the output from logistic regression is the odds ratio associated with a particular factor. By examining and comparing patients with positive and negative test results, Vittinghoff *et al.* [22] concluded that, in addition to evaluating a predictor of primary interest, it is important to investigate the importance of additional variables that may influence the observed association and therefore alter our inferences about the nature of the relationship.

Miettinen *et al.* [23] examines applicability of logistic regression as the statistical framework for prognostic clinical research, not merely as an option but as the only appropriate statistical-model framework for this research. A reliable approach to the classification of big medical data is proposed by the Awad *et al.* [24], which can improve the performance of algorithms. The work uses advanced parallel k-means preprocessing, a clustering technique that identifies patterns and structures in data. Also, the authors exploited the central processing unit (CPU) acceleration capabilities of the neural engine to further improve the speed and efficiency of our approach. The effectiveness of using logistic regression for analyzing medical data in combination with other methods has been shown in many works [25]–[31]. Most of the research on medical data analysis is related to the spread of Covid, where logistic regression is used [30] with methods such as: geographically weighted logistic regression (GWLR) [32]; single factor logistic regression analysis and multiple factor logistic regression analysis [33]; Oonishi *et al.* [34] the logistic regression analysis was conducted to examine the correlation between language mismatch between patients and healthcare providers and the requirement for professional medical interpretation.

#### 3.2. Mathematical description of the logistic regression classifier

The logistic regression classifier is a model that predicts the probability that an instance belongs to a particular category. It is widely used for binary classification problems, where there are two possible outcomes, often denoted as 0 and 1. Here's the mathematical description of logistic regression.

The hypothesis function  $h_{\theta}(x)$  for logistic regression is defined as (1):

$$h_{\theta}(x) = \frac{1}{1+e^{-\theta^T x}} \quad (1)$$

here:  $h_{\theta}(x)$  is the predicted probability that  $y = 1$  given the input features  $x$ .

$\theta$  represents the model parameters (weights).

$x$  is the input feature vector.

$\theta^T$  denotes the transpose of  $\theta$ .

$e$  is the base of the natural logarithm (approximately 2.71828).

The decision boundary is the line that separates the two classes (0 and 1). It is determined by in (2):

$$\theta^T x = 0 \quad (2)$$

this equation represents the points where the predicted probability  $h_\theta(x)$  is equal to 0.5, and it is used to classify instances into different classes. The cost function for logistic regression is given by the log-likelihood of the observed data under the logistic regression model. For a single training example  $(x, y)$ , the cost function is:

$$J(\theta) = -y \log(h_\theta(x)) - (1 - y) \log(1 - h_\theta(x)) \quad (3)$$

the overall cost function for the entire training set is the average of the individual costs over all training examples:

$$J(\theta) = -\frac{1}{m} \sum_{i=1}^m \left[ y^{(i)} \log(h_\theta(x^{(i)})) + (1 - y^{(i)}) \log(1 - h_\theta(x^{(i)})) \right] \quad (4)$$

here,  $m$  is the number of training examples.

The goal of training is to find the values of  $\theta$  that minimize the cost function. This is often done using optimization algorithms like gradient descent. Once the model is trained, for a new input  $x$ , the predicted class is determined by comparing  $h_\theta(x)$  to a threshold (commonly 0.5). If  $h_\theta(x) \geq 0.5$ , the predicted class is 1; otherwise, it is 0. Logistic regression can be extended to handle multiclass classification using techniques like one-vs-all or one-vs-one.

### 3.3. Medical text analysis algorithm for predicting mortality based on the logistic regression classifier

Developing a medical text analysis algorithm for predicting mortality based on a logistic regression classifier involves several steps. Here's a general outline of the process.

- i) Data collection: collecting a dataset that includes medical text data along with corresponding labels indicating whether the patient survived or not. This dataset might include electronic health records, clinical notes, or other relevant medical text sources.
- ii) Data preprocessing.
  - Text cleaning: removing the irrelevant characters, numbers, and symbols.
  - Tokenization: splitting the text into individual words or tokens.
  - Stopword removal: removing common words that do not carry much meaning.
  - Stemming/lemmatization: reducing the words to their root form.
- iii) Feature extraction: converting the preprocessed text data into numerical features that can be used by the logistic regression model. Common techniques include:
  - Bag of words (BoW): representing each document as a vector of word frequencies.
  - Term frequency-inverse document frequency (TF-IDF): reflects the importance of a word in a document relative to its frequency across all documents.
- iv) Data splitting: splitting the dataset into training and testing sets to evaluate the model's performance on unseen data.
- v) Logistic regression model: training a logistic regression model using the training data. Use the mortality status as the binary outcome variable (0 for survival, 1 for mortality).
- vi) Model evaluation: evaluating the model's performance on the testing set using metrics such as accuracy, precision, and recall. Adjusting the model if needed.
- vii) Interpretability: one advantage of logistic regression is its interpretability. Analyze the coefficients assigned to each feature to understand which words or phrases contribute most to the prediction of mortality.
- viii) Fine-tuning: experiments with hyperparameter tuning to optimize the model's performance. This might involve adjusting regularization parameters or trying different feature extraction techniques.
- ix) Deployment: after testing the model's performance, using the model to make predictions on new, previously unknown data. It is necessary ensure that it integrates well with the medical systems in use.
- x) Continuous improvement: continuously monitor and update the model as more data becomes available or as medical practices evolve.

Predicting mortality based on medical text is a complex task, and the success of the model depends on the quality and representativeness of the data, as well as the collaboration with domain experts in the medical

field. It is also necessary to ensure compliance with confidentiality rules and ethical standards when processing medical data.

## 4. RESULTS AND DISCUSSION

### 4.1. Description and collection of initial research data

One of the most difficult tasks is to collect reliable and complete data for research. For the study, MIMIC-III medical data were taken from Boston Hospital. MIMIC-III is a large, freely accessible database containing unidentified health data related to more than 45,000 patients who were in intensive care units at Beth Israel Deaconess Medical Center from 2001 to 2012. The data schema is shown in Figure 1.

Patients over 89 years of age have had their date of birth changed at any time in the database to hide their age and comply with health insurance portability and accountability Act (HIPAA) or the HIPAA. The change process was as follows: the age of the patient was determined at his first admission. Then the date of birth was set exactly 300 years before their first admission.

A demo version is freely available, which contains information about 100 patients and 123 cases of hospitalization. In our case, we will use the full version for the dissertation. The database includes information such as demographic data, measurements of vital signs performed at the patient's bedside, laboratory test results, procedures, medications, notes of caregivers, reports on imaging and mortality (both in and out of the hospital).

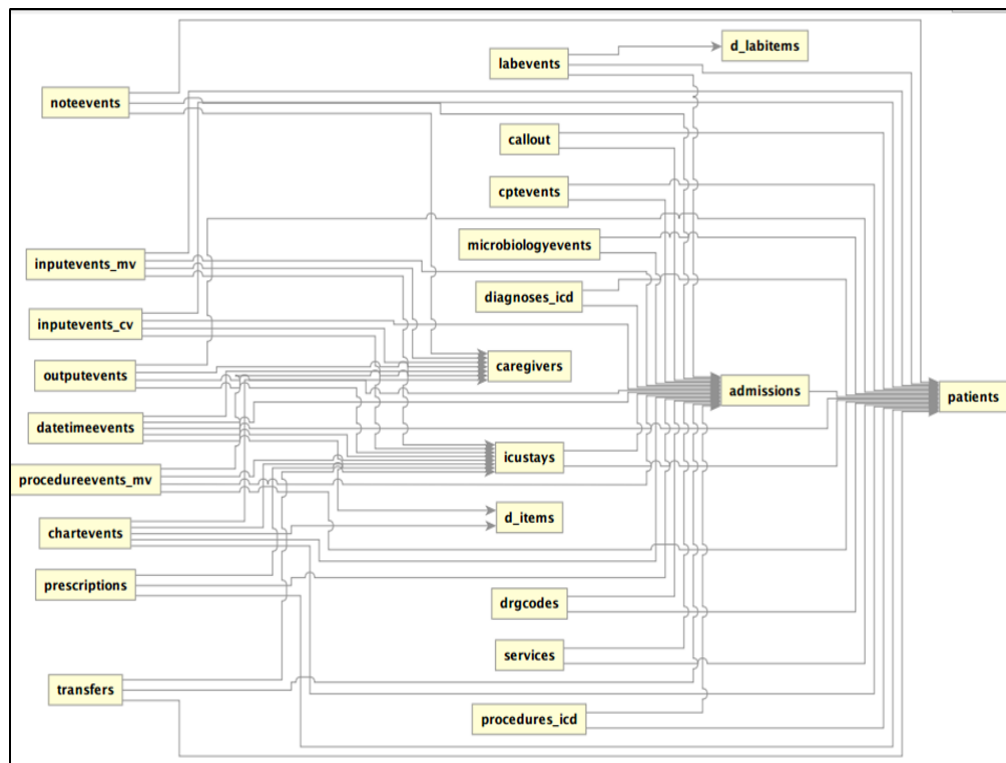


Figure 1. Schema of MIMIC-III tables

The tables are linked by identifiers, which usually have the suffix “ID”. For example, HADM\_ID refers to a unique hospitalization, and SUBJECT\_ID refers to a unique patient. One exception is ROW\_ID, which is just a row identifier unique to this table.

The following tables are used to determine and track the stay of patients:

- ADMISSIONS: each unique hospitalization for each patient in the database (identifies the HADM\_ID).
- CALLOUT: information about when the patient was released for discharge from the intensive care unit and when he was discharged.
- ICUSTAYS: each unique stay in intensive care in the database (defines ICUSTAY\_ID).
- PATIENTS: each unique patient in the database (defines SUBJECT\_ID).

- SERVICES: clinical service where the patient is registered.
- TRANSFERS: transfer of a patient from bed to bed within the hospital, including admission and discharge from the intensive care unit.

Each ICUSTAY\_ID (unique ICU stay) corresponds to one HADM\_ID (unique hospitalization) and one SUBJECT\_ID (unique patient). Each HADM\_ID corresponds to one SUBJECT\_ID. One SUBJECT\_ID can correspond to multiple HADM\_IDs (multiple hospitalizations of the same patient) and multiple ICUSTAY\_IDs (multiple ICUs remain either within the same hospitalization, or across multiple hospitalizations, or both).

The following tables contain data collected in the intensive care unit:

- CAREGIVERS: each caregiver who has written data to the database (defines CGID);
- CHARTEVENTS: all graphic observations for patients;
- DATETIMEEVENTS: all recorded observations that are dates, such as dialysis times;
- INPUTEVENTS\_CV: appointment for patients monitored with the Philips CareVue system during their stay in the intensive care unit;
- INPUTEVENTS\_MV: admission for patients monitored with the iMDSoft Meta vision system during their stay in the intensive care unit;
- NOTEEVENTS: unidentified notes, including nurses' and doctors' notes, ECG reports, imaging reports, and discharge summary;
- OUTPUTEVENTS: output information about patients in the intensive care unit;
- PROCEDUREEVENTS\_MV: procedures for a subset of patients who were observed in the intensive care unit using the iMDSoft meta vision system;

The following tables contain the data collected in the hospital record system:

- CPTEVENTS: procedures written as codes of current procedural terminology (current procedural terminology codes)
- DIAGNOSES\_ICD: inpatient diagnoses coded using the international statistical classification of Diseases and related health problems (ICD) system;
- DRGCODES: diagnostic groups used by the hospital for billing.
- LABEVENTS: laboratory measurements for both inpatient and outpatient settings;
- MICROBIOLOGYEVENTS: microbiological measurements and sensitivity from the hospital database;
- PRESCRIPTIONS: medicines ordered and not necessarily prescribed for a given patient;
- PROCEDURES\_ICD: procedures for patients coded using the international statistical classification of diseases and related health problems (ICD) system.

The following tables are dictionaries:

- D\_CPT: high-level codebook for current procedural terminology (current procedural terminology);
- D\_ICD\_DIAGNOSES: dictionary of international statistical classification of diseases and related health problems (ICD) codes relating to diagnoses;
- D\_ICD\_PROCEDURES: code dictionary of the international statistical classification of diseases and related health problems relating to procedures;
- D\_ITEMS: a dictionary of element identifiers appearing in the MIMIC database, excluding those related to laboratory tests;
- D\_LABITEMS: dictionary of element identifiers in the laboratory database.

#### 4.2. Software implementation of the medical text analysis algorithm

The essence of the developed program is to predict the probability of mortality based on the input data. The forecast will be carried out by training data from MIMIC-III. The input data is the patient's medical records, which is like the records from the NOTEEVENTS table, the output is a percentage value that reveals the probability of mortality of this patient. The higher the output value, the greater the probability of patient mortality (i.e., 0% is a low probability, 100% is a high probability).

The program is developed in the high-level Python programming language. It consists of two parts:

- Creation of vector representation of words from the text and learning model;
- Creation of the program interface.

The description of the program operation in the IDEF0 notation is shown in Figure 2. According to the diagram, the program consists of 4 main phases:

- Filtering and sampling data,
- Creation of vector meanings of words,
- Create a trained model, and
- Forecast data.

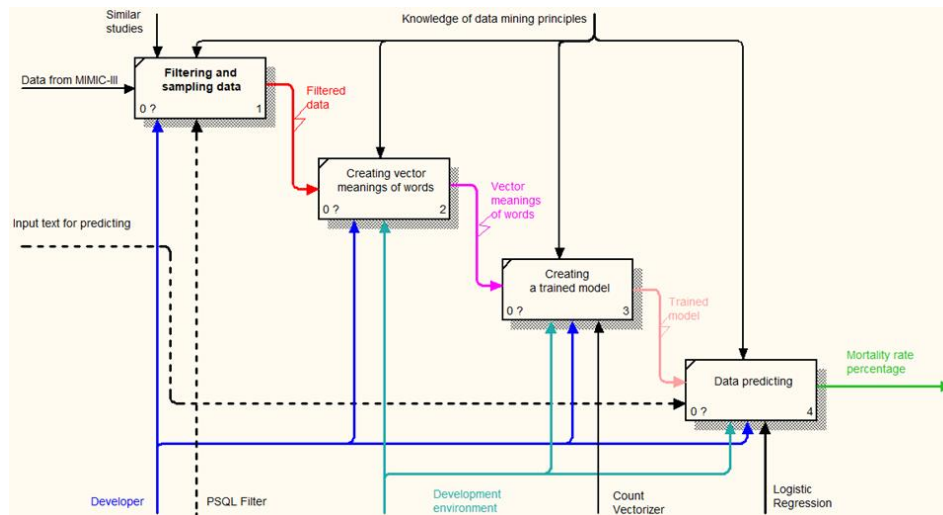


Figure 2. Expanded diagram IDEF0 program for predicting mortality

Filtering and sampling data are the first step in the process of the program. As noted in that subsection, the following two columns play a major role in predicting and classifying text: TEXT and EXPIRE\_FLAG. The program will be trained on texts and the EXPIRE\_FLAG flag. But one nuance is important here, which relates to the quality of the records. The fact is that the TEXT columns with notes are unstructured, and if you do not use filtering at this stage, the quality and correctness of the results that our future model will show may suffer significantly. The lack of structuring of the data, especially in the case of notes from MIMIC-III data, consists in the presence of symbols and empty lines that are unnecessary for training. In addition to the above two columns, we also have an ISERROR column, which is responsible for the correctness of records. A value of 1 or true in this column will mean that this entry is marked as erroneous, respectively, to train our future model, we should include only those rows in which the value of ISERROR is Null. To eliminate the above problems with note data, an SQL query was presented that links two tables and filters the data by according to the required criteria. The SQL query code is shown in Figure 3.

```
SELECT TRIM(LOWER(regexp_replace(noteevents.text, '\s+', ' ', 'g')))
AS text, patients.expire_flag, noteevents.iserror
FROM noteevents
FULL JOIN patients ON patients.subject_id = noteevents.subject_id
WHERE noteevents.iserror IS NULL
```

Figure 3. SQL query code for filtering

This SQL query, in addition to filtering notes from the TEXT column, performs the function of combining the NOTE EVENTS and PATIENTS tables with a common SUBJECT\_ID column. Subsequently, the result of the query is saved to a new file, and a new table is obtained with the cleared text of the notes and the EXPIRE\_FLAG flag responsible for the death of the patient. Data sampling was carried out on the same principle as filtering. The selection was carried out similarly to the work [35]; it includes filtering patients who had any of the following signs:

- Stay in the intensive care unit for less than 4 hours;
- Age less than 16 years at the time of admission;
- Organ donation as the purpose of the visit.

In the IDEF0 diagram shown in Figure 2, the output of the filtering phase is filtered data, which in turn is the input to the next phase of generating vector word values. To implement the creation of word-trained models, first, you need to create translations of notes from human language into a language that will be understood by the algorithm. It is necessary to develop a way of representing words in mathematical form that captures the meaning of the word and its relationship to other words. To obtain vector representations of words from the data filtered in the previous step, the count vectorizer algorithm provided by the Scikit-learn external library for Python was used. The part of the code responsible for the representation of words is shown in Figure 4.



```
#BOW

vectorizer1 = CountVectorizer()
vectorizer1.fit(text_train)
joblib.dump(vectorizer1, "vectorizer.pkl")
vectorizer = joblib.load("vectorizer.pkl")

X_train = vectorizer.transform(text_train)
X_test = vectorizer.transform(text_test)

#print(get_top_n_words(text_train))
```

Figure 4. A fragment of the program for obtaining vector representations of words from text data

Scikit-learn provides a few supervised and unsupervised learning algorithms using a consistent Python interface. It is licensed under the permissive simplified BSD license and distributed under many Linux distributions, encouraging academic and commercial use. The library is built on scientific Python (SciPy). This stack, which includes:

- NumPy: basic package of n-dimensional arrays;
- SciPy: a fundamental library for scientific computing;
- Matplotlib: complex 2D/3D plotting;
- IPython: advanced interactive console;
- Sympy: symbolic mathematics;
- Pandas: data structures and analysis.

Extensions or modules for SciPy care are conventionally called SciKits. Thus, the module provides learning algorithms and is called scikit-learn. With the help of the tool described above, a dictionary was created, which is based on data from NOTEEVENTS. With the help of a dictionary, it will subsequently be possible to create semantic vectors for sentences from the test and training sets. The part of the program code responsible for separating the data into training and test samples is shown in Figure 5.

```
#train_test_split

text_train, text_test, expire_flag_train, expire_flag_test = train_test_split(text,
                                                                              expire_flag,
                                                                              test_size=0.25,
                                                                              random_state=100)
```

Figure 5. A fragment of the program code for separating the data into training and test samples

Therefore, the third step or third phase of the development process will be the creation of a model trained on the vector representation of words. But before training data, it is necessary to split the data into training and test sets. In the dataset, the training set is implemented to build the model, while the test (or validation) set is designed to test the built model. The data points in the training set are excluded from the test set (validation). Typically, the dataset is divided into a training set, a validation set in each iteration, or is divided into a training set, a validation set, and a test set in each iteration. For the task of predicting a lethal outcome, a proportion of 75% and 25% was chosen for the training and test samples, respectively. The part of the code that is responsible for this division of data is shown in Figure 6.

Based on the received training sample, a model with the logistic regression classification will be created, the mathematical representation of which is presented below. Part of the training code is shown in Figure 7. An important criterion for assessing the quality of various models is the value of their accuracy. For the created model, its accuracy was also calculated using the score () function. The code responsible for calculating the accuracy of the created model is shown in Figure 8. A screenshot of the model accuracy value is shown in Figure 8.

The final, fourth phase of the process of predicting mortality from medical data is the prediction of the data itself. The forecast is carried out on the model trained in the third phase, using the predict\_proba method. The corresponding code is shown in Figure 9.

```
#LogisticRegression
classifier = LogisticRegression(solver='lbfgs', max_iter=1000) #default 100
classifier.fit(X_train, expire_flag_train)
filename = '1k_model.sav'
pickle.dump(classifier, open(filename, 'wb'))

# load the model from disk

loaded_model = pickle.load(open(filename, 'rb'))
score = loaded_model.score(X_test, expire_flag_test)
print("accuracy: ", score)
```

Figure 6. A fragment of the program code for training the prediction model

```
# load the model from disk

loaded_model = pickle.load(open(filename, 'rb'))
score = loaded_model.score(X_test, expire_flag_test)
print("accuracy: ", score)
```

Figure 7. A fragment of the program code for calculating the accuracy of the created model

```
accuracy: 0.816
```

Figure 8. Accuracy of the created model

The code fragment shown in Figure 9 was taken from the first part of the program, which is responsible for creating and saving the model. Here, data input for the forecast works via the input () method, i.e., via the command line. To save dictionaries and models, pickle and joblib tools were used for their subsequent use. The pickle module implements a powerful Python object serialization and deserialization algorithm. “Pickling” is the process of converting a Python object into a byte stream, and “unpickling” is the reverse operation, because of which the byte stream is converted back into a Python object. Since the byte stream can be easily written to a file, the pickle module is widely used to save and load complex objects in Python. Joblib is part of the SciPy ecosystem and provides utilities for Python pipelining. It provides utilities for saving and loading Python objects that make efficient use of NumPy data structures.

```
x_new = vectorizer.transform(inp)
y_new = classifier.predict_proba(x_new)[:, 1]
print('Mortality probability:', float(y_new)*100, '%')

clear_session()
```

Figure 9. A fragment of the program code responsible for predicting mortality

#### 4.3. Development of the graphical interface of the mortality predicting program

In the second part of the program, the user interface of the program is created. The export of the program code written in Python was carried out using the tkinter tool. Tkinter is a Python package designed to work with the Tk library. The Tk library contains graphical user interface (GUI) components written in the Tcl programming language.

The GUI refers to all those windows, buttons, text input fields, scrollers, lists, and radio buttons, checkboxes. that you see on the screen when you open an application. Through them you interact with the program and control it. All these interface elements will be called widgets.

The window has an input field where the user enters text with medical data, and when clicking on the “Forecast” button, the red text will change to the value of the probability of mortality calculated from the entered text. The interface of the death prediction program is an input box where you can insert text, a percentage representation of the mortality forecast, as well as buttons “About the program” and “Predicting”.

The text entered in the input box is transformed into a vector representation of words based on the previously created dictionary using the `transform()` method. The interface was made using the `tkinter` library. Figures 10 and 11 show the views of the program window.

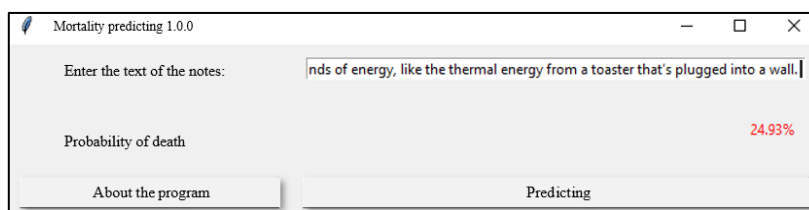


Figure 10. The program window with the entered text about the patient's condition

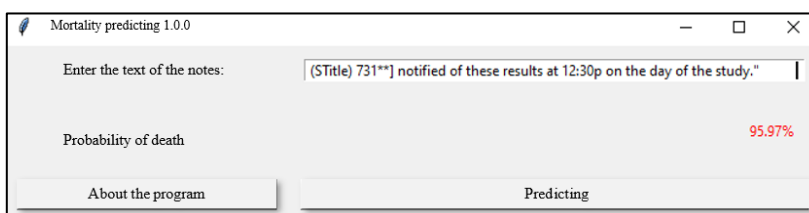


Figure 11. The program window with the entered text about the patient's condition

The program predicts a fatal outcome with a probability of 95.97%. The predicting is carried out based on a previously trained model using the `predict_proba()` method. When you click on the "Predicting" button, the `clicked()` method is executed. The code of the method executed when the predicting button is clicked is shown in Figure 12.

```
def clicked():
    inp = [txt.get()]
    x_new = vectorizer.transform(inp)
    y_new = loaded_model.predict_proba(x_new)[:, 1]
    rslt = round(float(y_new) * 100, 2)
    res.configure(text=str(rslt)+'%')
```

Figure 12. A fragment of the program code of the method executed when the "Predicting" button is clicked

## 5. CONCLUSION

In conclusion, the use of logistic regression in mortality forecasting proves to be a powerful and efficient tool, capable of considering numerous factors influencing this process. Our research findings underscore the accuracy and predictive ability of this method, allowing for a more detailed and systematic assessment of risks. Logistic regression not only provides a model with high adaptability to diverse data but also offers interpretable coefficients, crucial for understanding the impact of various factors on the probability of death. These conclusions support the prospect of further applying logistic regression in medical statistics and justify its significance for precise and reliable mortality forecasting across different populations. Now, due to the lack of medical data of sufficient volume and quality of the MIMIC-III level in the territory of the Republic of Kazakhstan, the use of this technique in the healthcare of the Republic of Kazakhstan is difficult. However, in the future, with the appearance of a similar database, mortality forecasting is feasible using the proposed method and tools.




## ACKNOWLEDGEMENTS

The article was carried out within the framework of the project of the Ministry of Education and Science of the Republic of Kazakhstan for grant financing of fundamental and applied scientific research of young postdoctoral scientists under the Zhas Galym project for 2022-2024. IRN: AP14972524 – "Development of VLC technologies in the management of unmanned vehicles."




## REFERENCES

- [1] P. A. Mendez-Tellez and T. Dorman, "Predicting patient outcomes, futility, and resource utilization in the intensive care unit: the role of severity scoring systems and general outcome prediction models," *Mayo Clinic Proceedings*, vol. 80, no. 2, pp. 161–163, Feb. 2005, doi: 10.4065/80.2.161.
- [2] P. Cerrito and J. C. Cerrito, "Data and text mining the electronic medical record to improve care and to lower costs," *Data Mining and Predictive Modeling*, pp. 1–20, 2011.
- [3] S. Jadhao, "Applications of data mining in healthcare and current issues," *Journal of Medical pharmaceutical and allied sciences*, vol. 10, no. 4, pp. 3384–3387, Oct. 2021, doi: 10.22270/jmpas.V10I4.1205.
- [4] Y. Marzhan, K. Talshyn, K. Kairat, B. Saule, A. Karlygash, and O. Yerbol, "Smart technologies of the risk-management and decision-making systems in a fuzzy data environment," *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 28, no. 3, p. 1463, Dec. 2022, doi: 10.11591/ijeecs.v28.i3.pp1463-1474.
- [5] U. Indira, S. Belginova, and A. Ismukhamedova, "Informational and analytical system to diagnose anemia," in *Proceedings of the Fourth International Conference on Engineering & MIS 2018*, New York, NY, USA: ACM, Jun. 2018, pp. 1–8. doi: 10.1145/3234698.3234716.
- [6] I. Uvalieva, S. Smailova, and F. Tarifa, "Development of information analysis software for the monitoring of distributed objects within socioeconomic systems," *Actual Problems of Economics*, vol. 165, no. 3, pp. 482–491, 2015.
- [7] N. R. Panda, J. K. Pati, J. N. Mohanty, and R. Bhuyan, "A review on logistic regression in medical research," *National Journal of Community Medicine*, vol. 13, no. 4, pp. 265–270, 2022, doi: 10.55489/njcm.134202222.
- [8] S. Belginova, I. Uvaliyeva, and S. Rustamov, "The application of data mining methods for the process of diagnosing diseases," *Journal of Theoretical and Applied Information Technology*, vol. 97, no. 7, pp. 1980–1998, 2019.
- [9] R. J. Rossi, "Logistic regression" in *Applied Biostatistics for the Health Sciences*, John Wiley & Sons, Inc., 2022, pp. 462–507. DOI: 10.1002/9781119722717.ch10
- [10] J. Jiang, "Survival analysis" in *Applied Medical Statistics*, John Wiley & Sons, Inc., 2021, pp. 369–398. DOI: 10.1002/9781119716822.ch17
- [11] E. Harrison and P. Riinu, *R for Health Data Science*, Chapman and Hall/CRC, 2020.
- [12] I.E. Hoffman, "Logistic regression" in *Basic Biostatistics for Medical and Biomedical Practitioners*, Academic Press, 2019, pp. 581–589
- [13] C. Lalanne and M. Mesbah, "Logistic regression" in *Biostatistics and Computer-based Analysis of Health Data using SAS*, 2017, pp. 97–113.
- [14] B. Ch. Tai and D. Machin, "Logistic regression" in *Regression Methods for Medical Research*, 2013, pp. 64–97.
- [15] W. H. Holmes and W. C. Rinaman, "Logistic regression" in *Statistical Literacy for Clinical Practitioners*, 2015, pp. 397–422.
- [16] N. Rezaei and P. Jabbari, "Linear and logistic regressions in R" in *Immunoinformatics of Cancers*, 2022, pp. 87–125.
- [17] M. N. Brunden, Th. J. Vidmar, and J. W. McKean, "Robust logistic regression" in *Drug Interaction and Lethality Analysis*, CRC Press, Taylor and Francis Group, 2019, pp. 113–118.
- [18] J. Jiang, "Logistic regression" in *Applied Medical Statistics*, John Wiley and Sons, Inc., 2021, pp. 369–398.
- [19] M. S. Goodman, "Logistic regression" in *Biostatistics for Clinical and Public Health Research*, Routledge, 2017, pp. 441–468.
- [20] Y. Hasija and R. Chakraborty, "Logistic regression" in *Hands-On Data Science for Biologists Using Python*, CRC Press, 2021, pp. 182–196.
- [21] P. Rowe, "Logistic regression" in *Essential Statistics for the Pharmaceutical Sciences*, Second Edition, John Wiley & Sons, Ltd., 2015, pp. 295–310.
- [22] E. Vittinghoff, D. V. Glidden, S. C. Shiboski, and C. McCulloch, "Logistic regression" in *Regression Methods in Biostatistics*, 2012, pp. 139–202.
- [23] O. S. Miettinen, J. Steurer, and A. Hofman, "The logistic regression model" in *Clinical Research Transformed*, 2019, pp. 61–70.
- [24] F. H. Awad, M. M. Hamad, and L. Alzubaidi, "Robust classification and detection of big medical data using advanced parallel k-means clustering, yolov4, and logistic regression," *Life*, vol. 13, no. 3, p. 691, Mar. 2023, doi: 10.3390/life13030691.
- [25] X. Wang and D. Zhang, "Choices of medical institutions and associated factors in older patients with multimorbidity in stabilization period in China: a study based on logistic regression and decision tree model," *Health Care Science*, vol. 2, no. 6, pp. 359–369, Dec. 2023, doi: 10.1002/hcs2.73.
- [26] H. Byun, S. Jeon, and E. S. Yi, "Analysis and prediction of older adult sports participation in South Korea using artificial neural networks and logistic regression models," *BMC Geriatrics*, vol. 23, no. 1, p. 676, Oct. 2023, doi: 10.1186/s12877-023-04375-2.
- [27] P. Chutel *et al.*, "Implementation of heart diseases prediction system using combination of XGBoost, logistic regression and random forest," *International Journal of Advanced Research in Science, Communication and Technology*, pp. 210–213, Nov. 2023, doi: 10.48175/IJARSCT-13633.
- [28] X. Kavelaars, J. Mulder, and M. Kaptein, "Bayesian multilevel multivariate logistic regression for superiority decision-making under observable treatment heterogeneity," *BMC Medical Research Methodology*, vol. 23, no. 1, p. 220, Oct. 2023, doi: 10.1186/s12874-023-02034-z.
- [29] P. S. Krishna, M. S. R. Sri, R. S. L. Triveni, T. Sivathmika, and R. Kanishka, "Dynamic weighted feature subset logistic regression model for heart disease prediction," in *Lecture Notes in Networks and Systems*, vol. 665 LNNS, 2023, pp. 455–464. doi: 10.1007/978-981-99-1726-6\_35.
- [30] S. H. Ko, M. C. Hsieh, and R. F. Huang, "Human error analysis and modeling of medication-related adverse events in Taiwan using the human factors analysis and classification system and logistic regression," *Healthcare*, vol. 11, no. 14, p. 2063, Jul. 2023, doi: 10.3390/healthcare11142063.
- [31] S. Wang *et al.*, "Prevalence and influencing factors of sleep disturbance among medical students under the COVID-19 pandemic," *European Archives of Psychiatry and Clinical Neuroscience*, Nov. 2023, doi: 10.1007/s00406-023-01707-6.
- [32] I. Haq, M. N. Aidi, A. Kurnia, and E. Efrwati, "A comparison of logistic regression and geographically weighted logistic regression (GWLR) on covid-19 data in west Sumatra," *BAREKENG: Jurnal Ilmu Matematika dan Terapan*, vol. 17, no. 3, pp. 1749–1760, Sep. 2023, doi: 10.30598/barekengvol17iss3pp1749-1760.
- [33] L. Zhou *et al.*, "Mental health survey of medical personnel during pre-job training in a closed-loop management system during the COVID-19 pandemic," *Frontiers in Public Health*, vol. 11, Nov. 2023, doi: 10.3389/fpubh.2023.1279153.
- [34] A. Oonishi, A. Ikeda, N. Francois, and N. Ono, "Relationship between patient-provider language discordance and the need for professional medical interpretation for international patients in Japan," *Cureus*, Oct. 2023, doi: 10.7759/cureus.47001.
- [35] A. E. W. Johnson and R. G. Mark, "Real-time mortality prediction in the intensive care unit," *AMIA ... Annual Symposium proceedings. AMIA Symposium*, vol. 2017, pp. 994–1003, 2017.




**BIOGRAPHIES OF AUTHORS**

**Zhenisgul Rakhmetullina**    is a Lecturer in Engineering mathematics at the D. Serikbayev East Kazakhstan technical university (EKTU), Ust-Kamenogorsk, Kazakhstan. She received his B.Eng., M.Eng., and Ph.D. degrees in Al-Farabi Kazakh National University, in 1992, 2000 and 2011, respectively. She has been an Associate Professor in EKTU, Ust-Kamenogorsk, Kazakhstan since 2012. She is currently the Head of the Faculty of Basic Engineering Training. Her research interests include the field of gravitating disk, differential equations, motion of a material point, perpendicular plane, potential, and mathematical modelling. She can be contacted at email: zhrakhmetullina@ektu.kz.






**Saule Belginova**    received the Master of Engineering academic degree of specialty “Mathematical and computer modeling” and the Ph.D. degree of specialty Information Systems from D. Serikbayev East Kazakhstan State Technical University (EKSTU), Oskemen, Kazakhstan, in 2014 and 2020, respectively. At the present she is an associate professor of Information Technology department in University “Turan”, Almaty, Kazakhstan. She has authored or coauthored more than 50 publications. Her research interests include soft computing, mathematical and computer modeling of processes, and intellectual analysis of medical data. She can be contacted at email: sbelginova@gmail.com.






**Alibekkyzy Karlygash**    received an academic degree of master of technical sciences in the specialty “Instrument making” and a degree of Ph.D. in the specialty “Automation and Control” from the East Kazakhstan State Technical University. D. Serikbayeva (EKTU), Ust-Kamenogorsk, Kazakhstan, in 2014 and 2022, respectively. Currently, he is an associate professor, Ph.D. of the Faculty of Information Technologies and Intelligent Systems of the East Kazakhstan Technical University. Serikbayeva D., Ust-Kamenogorsk, Kazakhstan. She is the author or co-author of more than 20 publications. Her research interests include LED systems, data communication, and self-driving cars. She can be contacted at email: karlygash.eleusizova@gmail.com.



**Aigerim Ismukhamedova**    Additionally, she holds the position of Senior Research Associate at the D. Serikbayev East Kazakhstan Technical University (EKTU). Currently serves as the Director of the Digitalization Center at the Kazakh-American Free University (KAFU). Her educational background includes pursuing a Ph.D. in Information Systems at EKTU. Since 2017, Ismukhamedova has been conducting research as part of her doctoral studies on her Ph.D. dissertation titled “Algorithmic Support for an Intelligent Clinical Decision Support System.” Under the guidance of her academic supervisor, she has published 15 scientific works related to her dissertation topic. She can be contacted at email: aigerim.ismukhamedova1@gmail.com.



**Shynar Tezekpaeva**    is a Lecturer in School of Digital Technology and artificial intelligence at the D. Serikbayev East Kazakhstan technical university (EKTU), Ust-Kamenogorsk, Kazakhstan. She received his B.Eng. degrees in Al-Farabi Kazakh National University in 1992 and M.Eng. degrees in D. Serikbayev East Kazakhstan technical university in 2012. She is Head of the educational program “Mathematical and Computer Modeling”. Her research interests include the field of computer modeling in Matlab, modern numerical and analytical packages for solving complex engineering and physical problems. She can be contacted at email: shtezekpaeva@edu.ektu.kz.