Diabetes 130-US hospitals for years 1999-2008

Final Report

Group 4

Phanindra Reddy Myakal

Sai Aakash Reddy Reddivari

Siddanth Emani

Tejashwini Vemavarapu

Abstract

Understanding the significance and characteristics of each element in the dataset and using the appropriate data to create better models are the goals of using healthcare data. Diabetes is a chronic disease that affects 7% of the world's population and is defined by elevated blood glucose levels, which raise the risk of stroke and mortality rates. The determination of patient readmission to hospitals following first diabetic treatment is our aim. We looked at numerous patient indicators when analyzing diabetic data from 130 US hospitals and categorized patients as being readmitted before or after 30 days. We used a variety of machine learning models that we trained to find trends, and we found a link between the quantity of inpatient admissions and the frequency of patients readmitted before 30 days. Comparatively, neural networks displayed the highest accuracy.

Introduction

By managing readmissions of diabetic patients, the aim is to conserve resources and money. To guarantee the accuracy and applicability of the collected data to diabetes, several criteria were put in place. To investigate historical patterns in the management of diabetes among patients admitted to US hospitals, an analysis of a sizable clinical database was carried out. The goal of this research is to steer future efforts that can improve patient safety.

https://www.hindawi.com/journals/bmri/2014/781670/#B8

Clinical data databases include crucial but complicated data, such as missing values, inconsistent or partial records, and high dimensionality. Both the quantity and their characteristics contribute to this complexity.

The dataset includes both category and numeric columns, but some of the columns are inconsistent or have missing values in them. Given that the data was collected from 130 US institutions, certain anomalies were anticipated. Additionally, a few columns that track patients' prescription drugs were added to see if differences in those factors may affect readmission rates.

https://www.hindawi.com/journals/bmri/2014/781670/#B8

Goal

To predict the risk of readmission for diabetic patients in US hospitals based on various patient and hospital characteristics.

Dataset

https://archive.ics.uci.edu/ml/datasets/diabetes+130-us+hospitals+for+years+1999-2008

Dataset Description

The diabetes dataset from the UCI Machine Learning Repository contains data on over 100,000 hospital admissions of patients with diabetes to 130 hospitals in the United States from 1999 to 2008. The goal of this analysis is to build a logistic regression model to predict whether a patient will be readmitted within 30 days of their initial hospitalization based on their demographic information, medical history, and medications. There are 37 categorical and 13 numerical variables.

1. Age: age of patient in years
2. Gender: gender of patient (male or female)
3. Admission_type_id: identifier corresponding to the type of admission (emergency,
   urgent, elective, etc.)
4. Time_in_hospital: number of days between admission and discharge
5. Medical_specialty: specialty of the admitting physician
6. Num_lab_procedures: number of lab procedures performed during the stay
7. Num_procedures: number of non-lab procedures performed during the stay
8. Num_medications: number of distinct medications administered during the stay
9. Change: whether there was a change in medication during the stay (yes or no)
10. DiabetesMed: whether any diabetes medication was prescribed (yes or no)
11. Readmitted: whether the patient was readmitted within 30 days (yes, no, or >30)

The dependent variable for the logistic regression model is the binary variable

"Readmitted", which can be coded as 1 for patients who were readmitted within 30 days and 0 for those who were not.

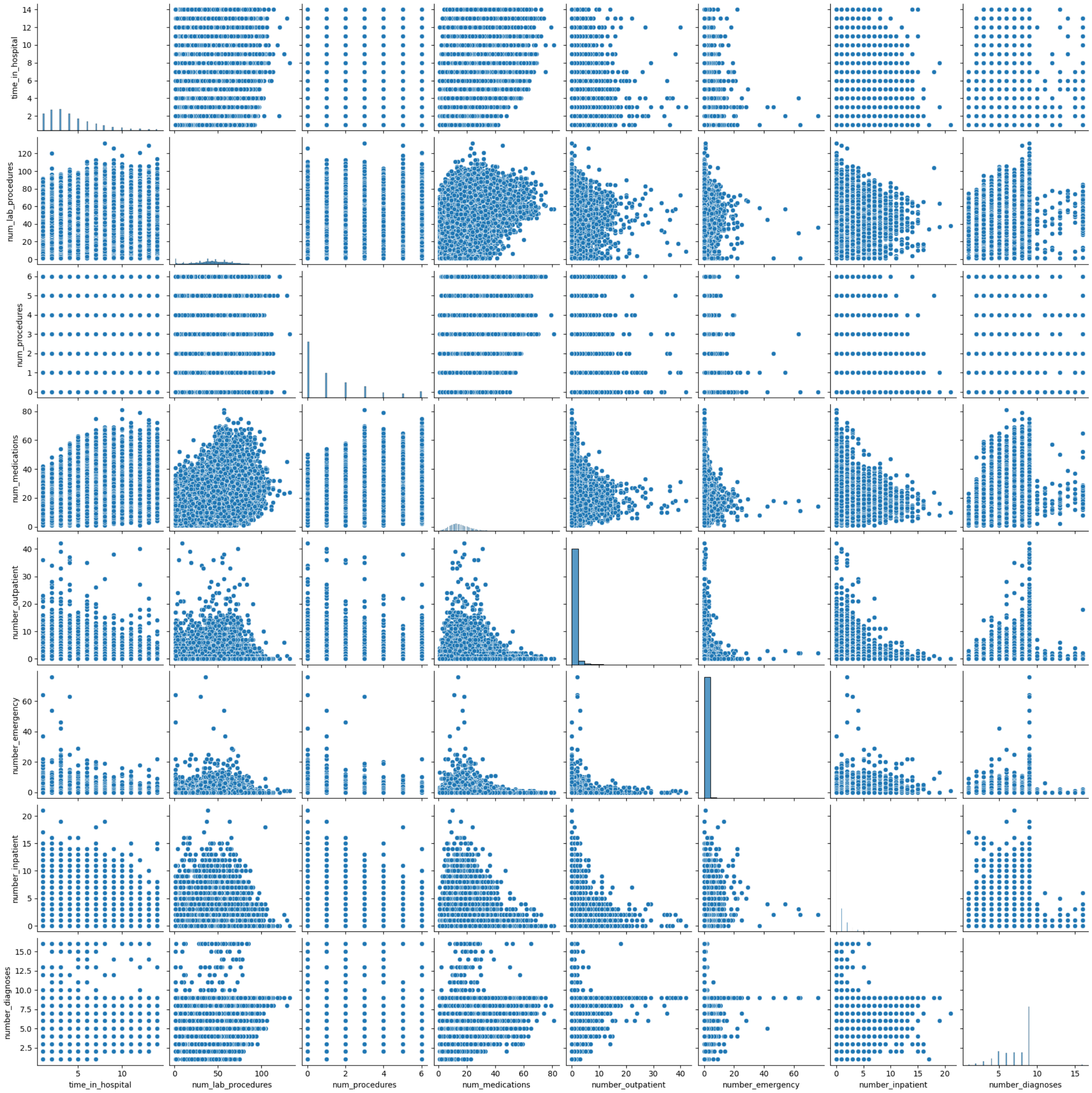
Data Preprocessing:

1. There are 13 numerical columsn and 37 categorical columns
2. Remove instances with missing values or incomplete data
3. 'Encounter_id', 'patient_nbr', 'weight', 'examide', 'citoglipton', 'diag_1', 'diag_2', 'diag_3' are removed.
4. Missing values are termed as ‘?’ which are replaced with NA
5. Encode categorical variables using one-hot encoding or label encoding
6. Standardize numerical variables using Z-score normalization

Statistical Analysis:

Multivariate Analysis Using pairplot()

1. Pairplot of numerical columns

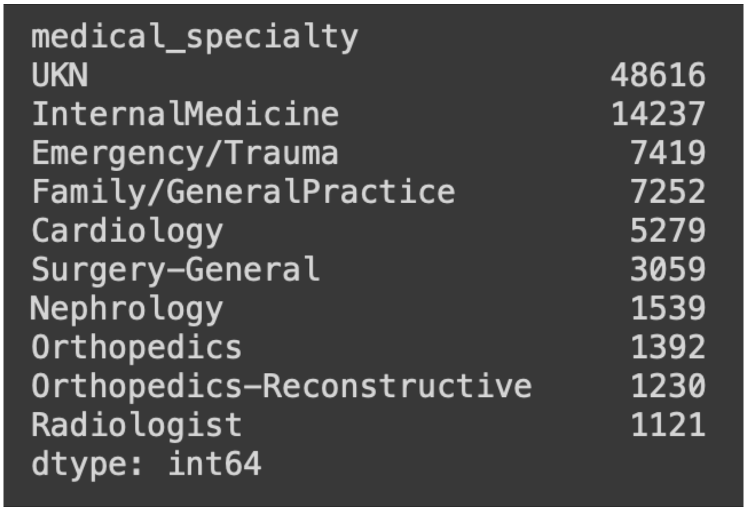


Feature Selection:

1. Use domain knowledge and exploratory data analysis to identify relevant features
2. Apply statistical techniques like correlation analysis, chi-square test, or feature importance score to select important features.

Feature Engineering:

1. If one-hot encoding were employed, the column "medical_speciality" would have a huge number of columns due to its 73 distinct values/categories. Only the top 10 categories, which make up around 93% of the data, have been chosen in order to make the data easier to understand.



```
medical_specialty
UKN                              48616
InternalMedicine                 14237
Emergency/Trauma                  7419
Family/GeneralPractice            7252
Cardiology                        5279
Surgery-General                   3059
Nephrology                        1539
Orthopedics                       1392
Orthopedics-Reconstructive        1230
Radiologist                       1121
dtype: int64
```

2. The dataset's 'Age' column is broken up into ranges or bins which are replaced by the average to make the model training easier. For example, the range [70-80] has been transformed into the number 75.
3. Three initial values are included in the column "readmitted": "30," ">30," and "NO." The class of interest is "<30," which is denoted by the number “1” and the rest as “0”.

Model Training:

1. Split the dataset into training (70%), validation (15%) and test (15%) sets
2. Fit a logistic regression model to the training data
3. Evaluate the model's performance on the validation set using metrics such as accuracy, precision, recall, and F1 score.
4. Tune the model hyperparameters using techniques to improve the model's performance.

Model Performance

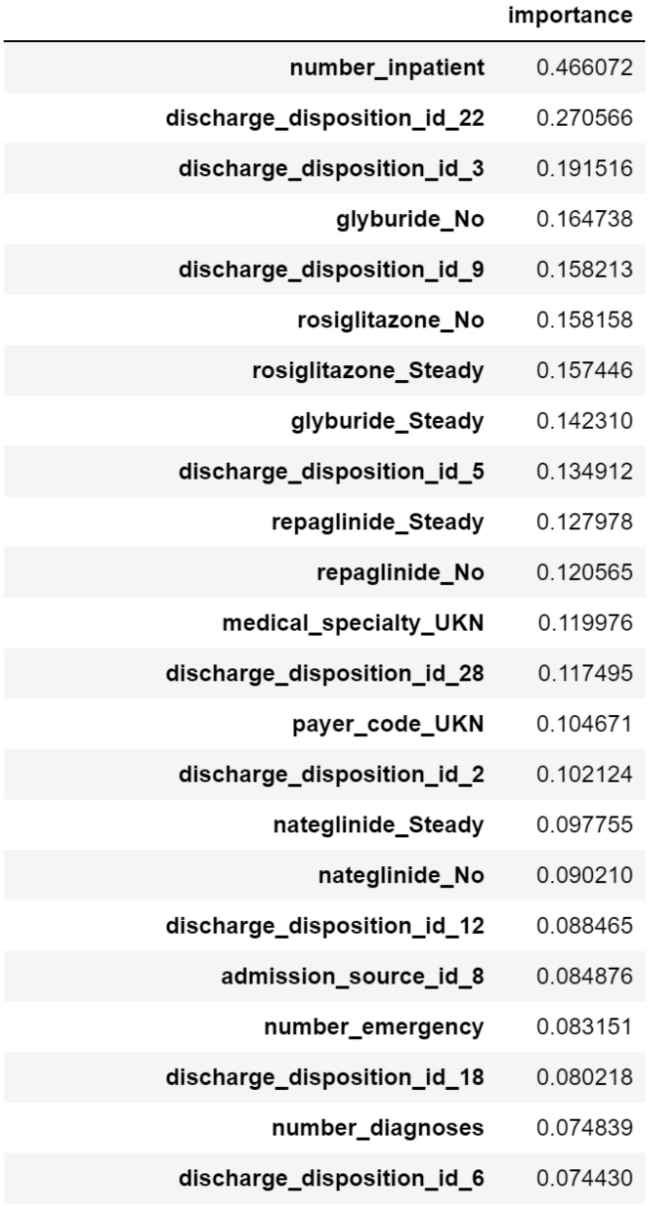
1. Logistic Regression

The first model we used for our investigation is a form of algorithm called logistic regression, which may predict the likelihood of a particular category. This model was chosen because it works well with binary data that is linearly separable, independent, and devoid of outliers. A binary output is created using the sigmoid function.

learningRate	Tolerance	Accuracy_training_data	Recall_training_data	Accuracy_test_data	Recall_test_data
0.001	0.0000001	0.5412442980233148	0.5404206791687786	0.5312709703395517	0.5320886814469078
0.01	0.001	0.5430816016218956	0.5399138367967562	0.5338880687156087	0.5344224037339557
0.001	0.00000001	0.5412442980233148	0.5404206791687786	0.5312709703395517	0.5320886814469078
0.000000001	0.001	0.6106183476938672	0.588317283324886	0.6121997047376191	0.5641773628938156
0.00000000001	0.0000001	0.6094145970603142	0.5881905727318804	0.6105891826600456	0.5653442240373395

To determine which hyperparameter combinations would work best with our model, we experimented with a variety, including learning rate and tolerance. As we sought to decrease False Negatives, the Recall measure was our primary area of attention. The model can predict class label '0' as '1' with more accuracy than the other way around. Therefore, lowering False Negatives raises Recall. Since our class label data is balanced, we also gave accuracy some thought as a performance parameter. We discovered that with learning rate = '0.00000000001' and tolerance = '0.0000001', we were able to improve accuracy and recall. As a consequence, we included these hyperparameters in our final model, which had an accuracy and recall of 62.2% and 57.2%, respectively.

Feature Importance



	importance
number_inpatient	0.466072
discharge_disposition_id_22	0.270566
discharge_disposition_id_3	0.191516
glyburide_No	0.164738
discharge_disposition_id_9	0.158213
rosiglitazone_No	0.158158
rosiglitazone_Steady	0.157446
glyburide_Steady	0.142310
discharge_disposition_id_5	0.134912
repaglinide_Steady	0.127978
repaglinide_No	0.120565
medical_specialty_UKN	0.119976
discharge_disposition_id_28	0.117495
payer_code_UKN	0.104671
discharge_disposition_id_2	0.102124
nateglinide_Steady	0.097755
nateglinide_No	0.090210
discharge_disposition_id_12	0.088465
admission_source_id_8	0.084876
number_emergency	0.083151
discharge_disposition_id_18	0.080218
number_diagnoses	0.074839
discharge_disposition_id_6	0.074430

Model Interpretation:

1. Interpret the model coefficients to understand the impact of each feature on the target variable.
2. Analyze the model's predictions using tools like confusion matrix, ROC curve, and precision-recall curve to gain insights into the model's behavior.
3. Use the model to generate predictions on new data and evaluate its performance on an independent test set.

2. Neural Networks:

A neural network is a collection of mathematical algorithms that imitates how the human brain functions in order to find patterns and correlations within a dataset. Organic or synthetic neurons can be found in neural networks.

https://www.investopedia.com/terms/n/neuralnetwork.asp

Our neural network's hidden layers were chosen to use the RELU activation function. This feature was chosen in part because of its quickness and ease of use. A deep network trained using ReLu tends to converge quicker and more consistently than one trained with sigmoid activation, according to prior studies.

https://stats.stackexchange.com/questions/126238/what-are-the-advantages-of-relu-over-sigmoid-function-in-deep-neural-networks

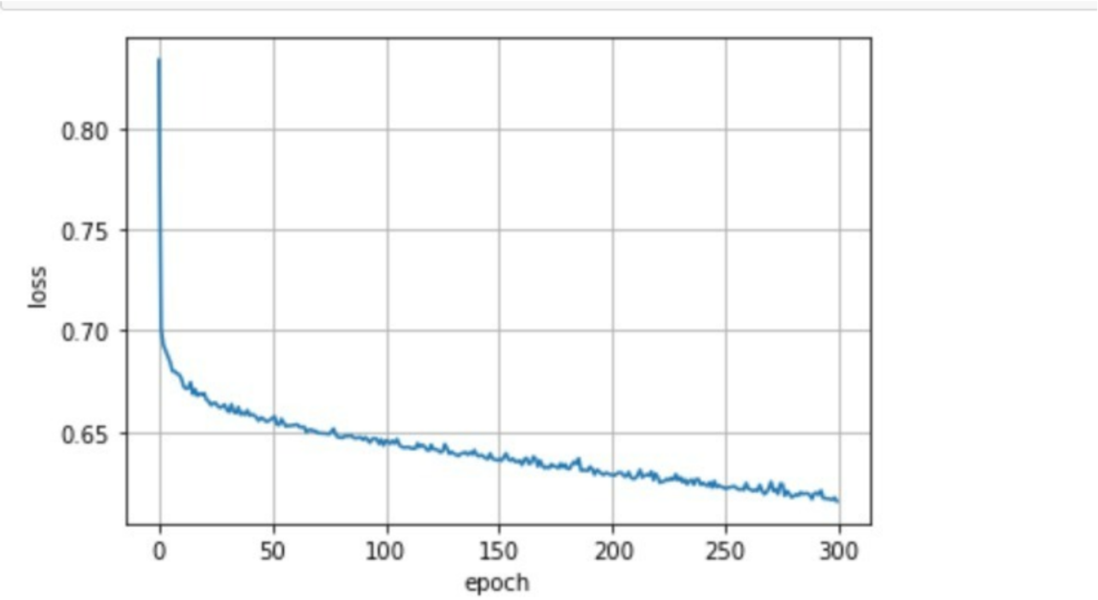
The Adam optimizer, which combines the RMSProp and AdaGrad algorithms, has been utilized. It is appropriate for optimizing models that deal with noisy and sparse gradient issues. The default parameter setting works well for the majority of issues, and it is simple to set up.

https://machinelearningmastery.com/adam-optimization-algorithm-for-deep-learning/

Input	H1	H2	H3	Output	epochs	batch	Accuracy	Recall
142	512	256	128	2	150	64	49.8	73
142	512	256	128	2	150	128	61.2	59
142	512	256	128	2	150	256	67.7	50.9
142	512	256	128	2	300	64	62.6	55.3
142	512	256	128	2	300	128	64.3	53.4
142	512	256	128	2	300	256	67	51.6

To improve the performance of our neural network, we experimented with different combinations of hyperparameters including the number of neurons in hidden layers, the quantity of epochs, and batch size. Our original combination had a poor recall of 45% but a high accuracy of roughly 72%, which was not ideal for our scenario. As a result, we adjusted the hyperparameters to give recall priority. We discovered that the final combination, which had an accuracy of 65% and a recall of 58%, fared the best among the various combinations. These outcomes outperformed our logistic regression baseline model.

Loss function on training data vs epoch



Bias and Variance Tradeoff

1. **Using Validation Set Approach**
   We evaluated our models during the modeling phase using a separate validation dataset, shielding them from the test data. By utilizing the validation data to evaluate many models, we were able to select the most accurate one to use for making predictions about the test data. We specifically divided the data into three parts: training (70%), testing (15%), and validation (15%).
2. **Random shuffling of data**
   To prevent the model from being trained on only one category of labels, we randomly sampled the data before dividing it into training and testing sets. We utilized balanced data during the training procedure to prevent the model from being biased towards any one category and to reduce the generalization error.
3. **Hyper Parameter Tuning**
   By experimenting with different combinations of hyperparameters, such as learning rate and tolerance for logistic regression and number of neurons in hidden layers, batch size, and epochs for neural network models, we have evaluated the performance of our models. The models' performance and error rates were improved and reduced by the adjustment of these hyperparameters.

Conclusion

Since the health care data provides important information, it is inappropriate to ignore any component that might first seem unnecessary. After examining the data, we found a number of connections that made sense and gave us a chance to do feature engineering. We were able to minimize the amount of features while maintaining the model's performance by combining different characteristics.

In the course of our investigation, we also discovered a few characteristics that significantly affect readmission prediction, such as "number_inpatients." We may concentrate on these crucial elements by recognizing them, which will increase the model's accuracy and help us make better choices. Feature engineering reduces noise and identifies the most important characteristics, which helps to simplify the data and enhance the performance of the model.