Introduction to Machine Learning in Healthcare

Project assignment

Welcome to Introduction to Machine Learning in Healthcare. During the next four weeks you will be introduced to the main concepts, algorithms and training strategies for machine learning models. You are expected to apply this knowledge in a project due on the **11th of July**, in which the final deliverables are a **report**, a **presentation** as well as the **code** produced. This document introduces the dataset to explore and sets the outline of both the report and the presentation.

The problem

Hyperglycemia management in a hospital setting has a strong impact in patients' clinical outcomes, namely in both morbidity and mortality. This has set a formal protocol of glucose targets within Intensive Care Units (ICU). However, this is still not the case for non-ICU patient admissions. The database here presented aims to examine historical patterns of diabetes care in patients with diabetes admitted to several US hospitals. Specifically, it was built to discover if there are any markers of attention to diabetes care in a large number of individuals identified as having a diagnosis of diabetes mellitus. [1]

In this project, you are expected to train a machine learning model that can predict a clinical outcome measure, in this case **re-admission within 30 days** after a diabetic encounter discharge.

The data set

"The data set represents 10 years (1999-2008) of clinical care at 130 US hospitals and integrated delivery networks. It includes over 20 features representing patient and hospital outcomes. Information was extracted from the database for encounters that satisfied the following criteria:

1. It is an inpatient encounter (a hospital admission).
2. It is a diabetic encounter, that is, one during which any kind of diabetes was entered to the system as a diagnosis.
3. The length of stay was at least 1 day and at most 14 days.
4. Laboratory tests were performed during the encounter.
5. Medications were administered during the encounter." [2]

[1] Beata Strack, Jonathan P. DeShazo, Chris Gennings, Juan L. Olmo, Sebastian Ventura, Krzysztof J. Cios, and John N. Clore, "Impact of HbA1c Measurement on Hospital Readmission Rates: Analysis of 70,000 Clinical Database Patient Records," BioMed Research International, vol. 2014, Article ID 781670, 11 pages, 2014.
[2] This dataset was adapted from Clore,John, Cios,Krzysztof, DeShazo,Jon, and Strack,Beata. (2014). Diabetes 130-US Hospitals for Years 1999-2008. UCI Machine Learning Repository. https://doi.org/10.24432/C5230J

This dataset is composed of 23 variables:

Feature name	**Type*** **	**Description and values**
encounter_id	num	Unique identifier of an encounter
patient_nbr	num	Unique identifier of a patient
race	cat	Values: Caucasian, Asian, African American, Hispanic, and other
gender	cat	Values: male, female, and unknown/invalid
age	cat	Grouped in 10-year intervals: [0, 10), [10, 20), ..., [90, 100)
admission_type_id	cat	Integer identifier corresponding to 9 distinct values, for example, emergency, urgent, elective, newborn, and not available
discharge_disposition_id	cat	Integer identifier corresponding to 29 distinct values, for example, discharged to home, expired, and not available
admission_source_id	cat	Integer identifier corresponding to 21 distinct values, for example, physician referral, emergency room, and transfer from a hospital
time_in_hospital	num	Integer number of days between admission and discharge
num_lab_procedures	num	Number of lab tests performed during the encounte
num_procedures	num	Number of procedures (other than lab tests) performed during the encounter
num_medications	num	Number of distinct generic names administered during the encounter
number_outpatient	num	Number of outpatient visits of the patient in the year preceding the encounter
number_emergency	num	Number of emergency visits of the patient in the year preceding the encounter
number_inpatient	num	Number of inpatient visits of the patient in the year preceding the encounter
diagnosis	cat	Primary, secondary and additional secondary codes (coded as CCS*)
number_diagnoses	num	Number of diagnoses entered to the system
A1Cresult	cat	Indicates the range of the result or if the test was not taken. Values: “>8” if the result was greater than 8%, “>7” if the result was greater than 7% but less than 8%, “normal” if the result was less than 7%, and “none” if not measured.
insulin	cat	Indicates whether insulin was prescribed “yes” or not “no”.
other_meds**	cat	Indicates whether any other drugs were prescribed “yes” or not “no”.
insulin_change,	cat	Indicates whether insulin had a change in dosage. Values: “up” if the dosage was increased during the encounter, “down” if the dosage was decreased, “steady” if the dosage did not change.
other_meds_change	cat	Indicates whether other drugs had a change in dosage. Values: “up” if the dosage was increased during the encounter, “down” if the dosage was decreased, “steady” if the dosage did not change.
diabetesMed	cat	Indicates if there was any diabetic medication prescribed. Values: “yes” and “no”

* Clinical Classifications Software

** metformin, repaglinide, nateglinide, chlorpropamide, glimepiride, acetohexamide, glipizide, glyburide, tolbutamide, pioglitazone, rosiglitazone, acarbose, miglitol, troglitazone, tolazamide, examide, citoglipton, glyburide-metformin, glipizide-metformin, glimepiride-pioglitazone, metformin-rosiglitazone, metformin-pioglitazone.

*** Feature type: num - numerical; cat - categorical

The goal

Optimize a machine learning model to predict re-admission in patients that had a previous diabetic encounter. The target feature is identified as “readmitted”.

Machine Learning Prediction Project Outline

1. Introduction

- Objective: Define the prediction task and the goal of the project.
- Dataset Description: Provide a brief description of the dataset, including the source and the type of data.

2. Data Exploration and Preprocessing

- Data Loading: Show how to load the dataset.
- Initial Exploration: Perform basic statistics and visualizations to understand the data.
- Data Cleaning: Handle missing values, outliers, and duplicates.
- Feature Engineering: Create new features if necessary.
- Data Transformation: Scale or normalize the data if required.

3. Exploratory Data Analysis (EDA)

- Visualizations: Use plots to explore the relationships between features and the target variable.
- Correlations: Calculate correlations between features and the target variable.
- Insights: Summarize the findings from the EDA.

4. Model Selection

- Model Types: Discuss different types of models (e.g., linear regression, decision trees, random forests, etc.) and their suitability for the task.
- Baseline Model: Start with a simple model to establish a baseline performance.

5. Model Training

- Data Splitting: Split the dataset into training and testing sets.
- Training: Train multiple models on the training data.
- Hyperparameter Tuning: Use techniques like grid search or random search to optimize model parameters.

6. Model Evaluation

- Metrics: Choose appropriate evaluation metrics (e.g., accuracy, precision, recall, F1-score).
- Validation: Use cross-validation to assess model performance.
- Model Comparison: Compare the performance of different models.

7. Model Interpretation

- Feature Importance: Identify the most important features in the best-performing model.
- Model Insights: Provide insights based on the model's predictions.

8. Conclusion

- Summary: Summarize the key findings and results of the project.
- Limitations: Discuss any limitations encountered during the project.
- Future Work: Suggest possible improvements or next steps.

9. Documentation and Presentation

- Report: Prepare a comprehensive report detailing each step of the project.
- Presentation: Create a presentation to showcase the project findings.

Tips for Students

- Reproducibility: Ensure that all code and analyses can be reproduced by others.
- Clarity: Write clear and concise code with comments.
- Visualization: Use visualizations effectively to communicate insights.
- Collaboration: Work in teams and collaborate using version control systems like Git.