

BIG DATA ANALYTICS PROJECT

Gian Marco Baroncini

The chosen dataset is **Diabetes 130-US hospitals for years 1999-2008**:

<https://archive.ics.uci.edu/dataset/296/diabetes+130-us+hospitals+for+years+1999-2008>.

The dataset represents ten years (1999-2008) of clinical care at 130 US hospitals. Each row concerns hospital records of patients diagnosed with diabetes.

The **goal** is to determine the early readmission of the patient within 30 days of discharge.

Characteristics:

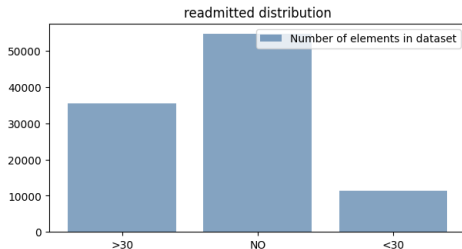
- Instances: 101766
- Features: 47
- Feature Type: Categorical, Integer

Labels description

The feature I tried to make predictions about is **'readmitted'**. It has 3 unique values:

- 'NO' which means no readmission
- '<30' indicates early readmission within 30 days of release
- '>30' which says that the patient returned to hospital but after 30 days

And this is its distribution

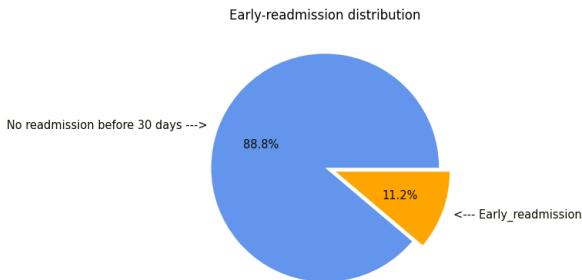


- Spark session initialisation and dataframe creation
- Data Exploration:
 - Inspection of some important features and dimensions
 - Display of some explanatory graphs
- Data pre-processing
- Classification tasks:
 - predicting all 3 readmission classes —> 'NO', '<30', '>30'
 - binary classification of 'NO' and 'Readmitted'
 - classification of early readmission
- Project work: Classification of early readmission for only instances whit A1Cresult

This section is divided into two parts:

- visualisation of structure and important properties of the dataset
- plots of some distributions and graphs

For example the percentage of early readmission rows in the dataset:



Adjustments were necessary for some key features for the following **problems**:

1. too many unique values (e.g. diag1)
2. useless information and noise
3. useless columns that do not carry information
4. categorical data (e.g. age)

The **solution** adopted:

1. reduction of unique values via domain knowledge
2. creation of new features (e.g. health index)
3. deletion of some columns
4. conversion to numerical values and normalisation

I divided the work into main project and project work. In the first case I tried to predict different things, so as to **experiment with as many models as possible** and become familiar with spark.

The results are not so good, few examples in the dataset of early readmission, so predicting that feature is too hard and **complex task**.

The fact remains that **this was not the main purpose**, but rather to learn how to handle a large amount of data and process it more efficiently with spark.

Despite this, I worked as if it were a research task analysing the properties of the data, modifying them **to achieve better performance** and not just creating the required models but testing them in different forms and combining them.

In **conclusion**, the results mirror those of other work carried out on this dataset.

Following the scema introduced before:

1. predicting all 3 classes
 - DT classifier
 - random forest
2. predicting general readmission
 - DT classifier (to have a comparison)
3. predicting early readmission
 - DT classifier
 - MLPC
 - custom network with keras over horovod frameworks
 - PCA and keras network

What is Horovod?



It's a free and open-source software framework for **distributed deep learning** training using TensorFlow, Keras, PyTorch, and Apache MXNet. Horovod is hosted under the Linux Foundation AI. Horovod has the goal of improving the speed, scale, and resource allocation when training a machine learning model.

The **horovod.spark package** provides a convenient wrapper around Horovod that makes running distributed training jobs in Spark clusters easy.

In situations where training data originates from Spark, this enables a tight model design loop in which data processing, model training, and model evaluation are all done in Spark.

Unfortunately, the dataset with only those rows that have the test performed is extremely limited. This leads to results that are not too satisfactory, but in line with the previous ones.

Methods used:

- K-means clustering
- DT classifier (always to have a comparison and also because it is one of the best models for tabular data)
- linear SVM
- keras neural network

Try to make some prediction on A1Cresult —> was it done or not?