BIG DATA ANALYTICS PROJECT

Gian Marco Baroncini

Dataset Selection

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:

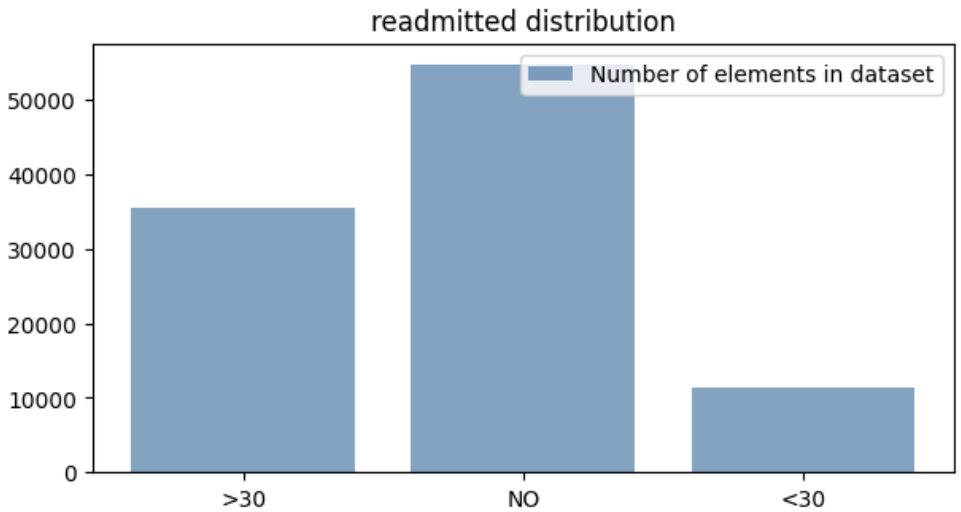
- Instances: 101766
- Features: 47
- Feature Type: Categorical, Integer

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



Structure Of The Notebook

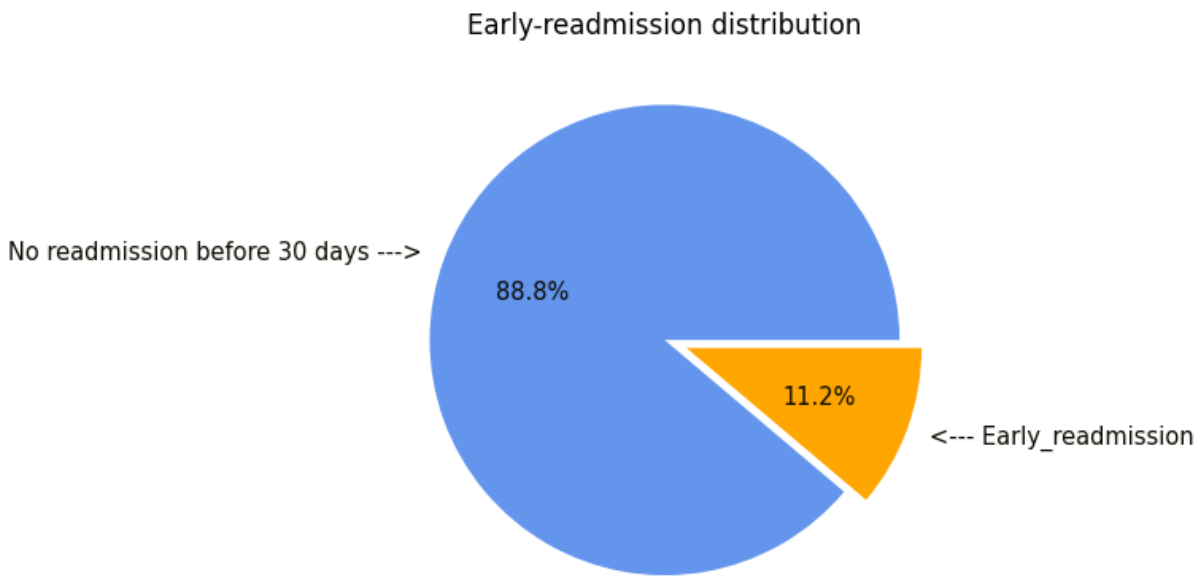
- Spark session initialisation and dataframe creation
- Data Exploration:
  - Inspection of some important features and dimensions
  - Display of some explanatory graphs
- Data pre-processing
- Classification tasks:
  - predicting all 3 readmission classes —> 'NO', '<30', '>30'
  - binary classification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

Data Exploration

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:



Data Pre-processing

Adjustments were necessary for some key features for the followng **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

Outline Of Executed Tasks

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.

Models used in the PROJECT (5)

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

Horovod

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.

Results and error analysis

	DT	RF	MLPC	keras NN	PCA+NN
All 3 classes	F1=0.52	F1=0.52	/	/	/
General readmission	F1=0.62	/	/	/	/
Early readmission	F1=0.83, N1=41	/	N1=0	F1=0.84, N1=397	N1=215

Table 1: Results as F1-score and/or N1; N1 stands for "numbers of 1 predicted

N.B. In the case of "early readmission" there are 2261 ones in the test set. For "keras NN" inside the 397 ones predicted there are 138 correct ones, for a for a hit rate of 35%. For "PCA+NN" there are 75 correct ones over 215; hit rate equal to 35% again.

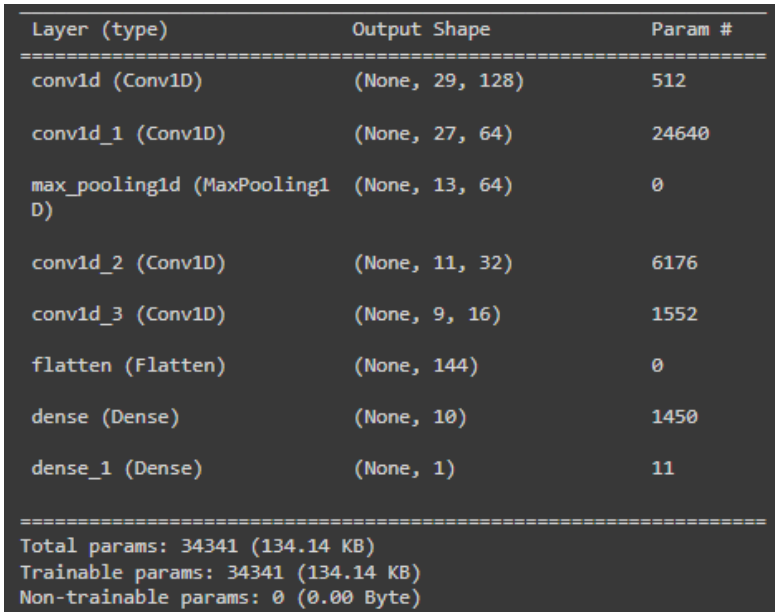
In conclusion: the last two models work proportionally in the same way, you just have to decide whether it is better to have false positives or false negatives.

Best Model for early readmission

The **best** model is the **keras network** which exploits 1D convolution layers.

It is significantly better at predicting more early readmissions.

The hitting rate is not perfect, but by playing a little with the **threshold** on the output of the sigmoid, a satisfactory tradeoff between 1 predicted and missed 1 can be found.



Layer (type)	Output Shape	Param #
conv1d (Conv1D)	(None, 29, 128)	512
conv1d_1 (Conv1D)	(None, 27, 64)	24640
max_pooling1d (MaxPooling1D)	(None, 13, 64)	0
conv1d_2 (Conv1D)	(None, 11, 32)	6176
conv1d_3 (Conv1D)	(None, 9, 16)	1552
flatten (Flatten)	(None, 144)	0
dense (Dense)	(None, 10)	1450
dense_1 (Dense)	(None, 1)	11

Total params: 34341 (134.14 KB)
Trainable params: 34341 (134.14 KB)
Non-trainable params: 0 (0.00 Byte)

Figure 1: Details of custom network

Models used in the PROJECT WORK (4)

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

Results

The **results** are very poor, the main reasons are the limited number of instances and the simplicity of the models used.

It was a result I expected, but I wanted to try to implement different models from the previous ones, putting the score aside for a moment and despite knowing that they were not the most suitable.

As it is shown in the notebook the models often confuse the classification of 0 and 1 or even never recognise early readmission by consistently predicting 0.

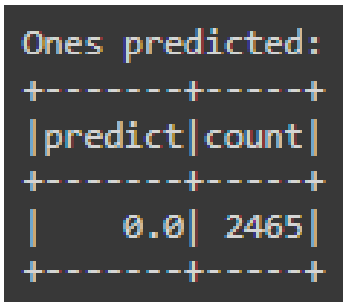


Figure 2: Prediction of keras network

Future Work

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

Inspection of the notebook

The work was too extensive to include the details in the presentation and to fully understand the technical choices.

So now **let's see the code** to discuss the technical details.