APA- Practical Work 2017–2018

Albert Ribes Kerstin Winter

January 17, 2018

Contents

1 Introduction

1.1 Desciption of the work and its goals

The goal of this project is to build a classification model to predict whether a lung cancer patient will die within one year after surgery or not. To do so we will study a dataset with real lung cancer patients.

As this is very sensitive information, our priority will be to minimize the amount of false negatives, i.e., avoid predicting a patient will not die within one year when it certainly does.

The data is taken from `https://archive.ics.uci.edu/ml/datasets/Thoracic+Surgery+Data#` [**zieba2013boosted**]

1.2 Desciption of available data

The data we are working with is about patients who underwent major lung resections for primary lung cancer in the years from 2007 to 2011. For each patient we are given information about his diagnosis and effects produced by the cancer.

The dataset is very limited in the number of instances available: it only has 470. In addition, the distribution of the predicted class isn't quite balanced, since only 70 of the patients died in one year period. This may become a problem in some of the prediction models due to the fact that the results will be biased towards the biggest class. However, we can suppose that the data has been collected uniformly and that this proportion is similar to the real one.

For each patient we have 16 different atributes. 3 of them are numerical, and the rest are categorical. From those, 10 are binary. The response atribute is also binary.

1.3 Instructions for running the code

- Cual debe ser el working directory
- La variable que contiene el path que tiene que modificar
- Los resultados ya están calculados en un working space, para ahorrar tiempo.

1.3.1 Needed packages

2 Related Previous Work

Nuestros datos son complicados, puesto que están muy balanceados y son muy pocos. El paper que referenciamos enfrente este tipo de problemas y propone usar SVM. Comentar que también usan este dataset.

Boosted SVM for extracting rules from imbalanced data in application to prediction of the post-operative life expectancy in the lung cancer patients

3 Data exploration process

3.1 Pre-processing

3.1.1 Treatment of missing values

Our dataset do not have missing values, so there is no need to treat them.

3.1.2 Treatment of anomalous values

The age of the patients is not well distributed. Most of them are over 60 years old, and only 4 of the patients are under 40. Due to this, it is very likely that the conclusions of our study will only be applicable to the elder people. However, we are reluctant to remove the younger patients.

3.1.3 Treatment of incoherent values

The variable FEV1 which is the Forced Expiratory Volume in 1 second, shows a few anomalously high values. Depending on factors like age and sex of a patient the average value of the FEV1 should be around 3-6 litres, whereas the dataset shows values up to 86. As most of the values are within 0 and 10 we assume that the dataset contains the FEV1 in litres. To decide which values are to be determined as outliers we calculate the FEV1/FVC ratio which gives the percentage of the lung volume exhaled in the first second over the whole exhaled volume. All patients having a unrealistic ratio higher than 100%, which are 22 patients, are determined to be outliers and eliminated. We chose not to apply any stricter constraints because the dataset does not include the sex of the patients which influences the normal values of the FEV1 much.

Source of knowledge about FEV1 and FVC: https://www.nuvoair.com/blogs/blog/do-you-know-how-to-interpret-the-results-of-your-spirometry-test

El enlace antiguo ha caido, ahora es este:

https://www.nuvoair.com/do-you-know-how-to-interpret-the-results-of-your-spirometry-test.html

Poner bien la referencia a la página

3.1.4 Coding of non-continuous or non-ordered variables

As most of the dataset variables are logical, we have coded them as logical in R. We have converted the variable "DGN" to many binary variables, each one saying wether the patient showed that diagnosis or not.

Originally, the variables "PERFORMANCE" and "SIZE" were categorical, but as they seem to have some kind of order, we have coded them as numeric. The "PERFORMANCE" can have values $1, 2, 3$ and the variable "SIZE" can have values $1, 2, 3, 4$. Both of them are then normalized.

"AGE" is also normalized in the range $[0, 100]$

3.1.5 Possible elimination of irrelevant variables

Some of the variables of our dataset are not well represented. In particular:

DGN	There is just one patient with DGN = 1 and just 8 have DGN = 8
PAD	Just 8 patients have PAD = True
ASHTMA	Just 2 patients have ASHTMA = True
MI	Just 2 patients have MI = True

As we have very few instances we will train and run the models on two different datasets. One of them, thoraric.original, will contain all of the original attributes, and the other, thoraric.removed, will contain all attributes but "DGN.1", "DGN.8","PAD","ASHTMA" and "MI".

3.1.6 Creation of new useful variables (Feature extraction)

We have not created any new variable as most of the ones we have are logical and it doesn't seem to be any relation among them.

3.1.7 Normalization of the variables

We need to normalize only our numeric variables, which are the AGE, FVC and FEV1. As we have converted the variables "PERFORMANCE" and "SIZE" to numeric (as we will see in section 3.1.8) we also need to normalize them. To normalize the age we will only consider cases between 0 and 100 years old. For FVC and FEV1 the range will correspond to the maximum and minimum observed values with a margin of 10%.

3.1.8 Transformation of the variables

Acording to the paper we found the accetable range for skewness in a numeric variable is $(-2, +2)$. In the original dataset, the one which contained all the patients, the skewness of variables AGE, FVC and FEV were out of this range. But after eliminating 22 of the patients (in 3.1.3) all of them are inside the acceptable range, so there's no need to transform them.

3.2 Renaming of the atritutes

In order to improve legibility we have renamed some of the atributes in the original dataset for us to better understand the meaning of each one. The following table maps the name in the original dataset to our pre-processed dataset.

Original	Our
DGN	DGN
PRE4	FVC
PRE5	FEV1
PRE6	PERFORMANCE
PRE7	PAIN
PRE8	HAEMOPTYSIS
PRE9	DISPNOEA
PRE10	COUGH
PRE11	WEAKNESS
PRE14	SIZE
PRE17	DIABETES
PRE19	MI
PRE25	PAD
PRE30	SMOKE
PRE32	ASTHMA
AGE	AGE
Risk1Y	DIED

4 Resampling protocol

After the pre-processing we have 2 datasets with 448 instances. *thoraric.original* has 23 attributes and *thoraric.removed* has 19. 0.85% of the patients have the target variable (DIED) equals FALSE.

The following lines explain the resampling protocol that we have used. All the numbers used can easilly be tuned in the scripts provided.

To test our models we split our data into two different datasets, one for training and one for testing. The testing dataset will contain $\frac{1}{3}$ of all the patients. As they are chosen randomly, it is expected that the proportion of patients in each of the classes is kept. We will only use the testing dataset in the end, to test our models.

In the training dataset we have 299 patients, and it is expected that over 254 of them will be DIED = FALSE and 45 will be DIED = TRUE. As most of the models are sensitive to non-balanced datasets, we need to do something to balance our training dataset. We use the *bagging* algorithm to generate 51 balanced datasets, and each one of them will be used to train 51 models of the same type. That number could be different, but it is good for it to be odd, to avoid ties. Then, to generate a prediction, each of the models will be used to make a *hard* vote, and the class predicted by the majority will be the answer. This way we have constructed what we have called a *super-model* built with simpler models.

Each of the *bags* will contain the same amount of instances. As we want them to be balanced, all of them will contain every TRUE instance in the training dataset, and a random sample of the same size of the FALSE instances. Thus, each bag will have over $2 \times 45 = 90$ instances.

5 Metric used to evaluate the models

Explicar porque usamos esta métrica, qué queremos incentivar, etc

6 Results obtained using linear/quadratic methods

6.1 Naive Bayes

Naive Bayes Algorithm has the advantage that it doesn't distinguish between types of data, and it doesn't perform any implicit transformation. The typical disadvantadge of this method is that it assumes independence on the attributes of the data.

As we have mixed data, the advantage is very appropriate. Looking at our data, it seems like most of the attributes are independent, so we can assume the results will be good. In fact, as we are working with a reduced dataset (in which we have removed some of the attributes), we could expect that the results will be even better in that one.

As this is method is so simple, is doesn't need parameters, so we don't event need the crossvalidation process and we don't have to chose any hyperparameter.

The results obtained using this model are:

Table 1: NB Original

Prediction	Reference	
	False	True
False	32	3
True	93	21

Table 2: NB Removed

Prediction	Reference	
	False	True
False	122	18
True	5	4

Table 3: F2 on NB

	Original	Removed
Naive Bayes	0.2990654	0.9413580

At the first sight, it stands out the fact that the results are very different from the "original" dataset and the "removed" one. While "original" shows results over 0.3, the other one shows much better results. This leads us to think that the attributes we removed weren't indeed independent from the others, and keeping theme out has helped the model.

6.2 KNN

KNN will look for similar patients and will predict the majority among them. This method is very sensitive to very imbalanced data, since with a lot of patients in one class it is very probably that many of them will be very close to the positive cases. Hence, we hope that our resampling method will be suitable for this model, as it only trains it with a balanced dataset of patients.

Nearest Neighbours is also sensitive to non-normalized dataset. Although the method used in R states that is does it, we have also normalized the data, just in case.

As most of our data is boolean, we have chosen to measure similarity with the Manhattan distance, instead of the Euclidean one. We expect results wil be beter with Manhattan distance since it reflects beter de diferences among patients.

Looking at the models created with crossvalidation we see that the average number of neighbours considered is 6.45 for "Original" and 5.91 for "Removed", which seems like normal values for k. The results obtained using this model are:

Table 4: KNN Original

Prediction	Reference	
	False	True
False	58	8
True	67	16

Table 5: KNN Removed

Prediction	Reference	
	False	True
False	73	8
True	54	14

Table 6: F2 on KNN

	Original	Removed
K-neares neighbours	0.5123675	0.6196944

The $F2$ score is not very good for non of the datasets, but looking at the confussion matrix is seems that the model achieves very well the task of avoiding real positives predicted as negatives.

Intentar ver por qué ha pasado esto

6.3 General Linear Model

The main problem of using this model is that is doesn't distinguish between different data types. For this model all the data are real numbers. As most of our variables are categorical, we assume it will not reflect very well our data. In addition, the power of this method relies on the basis function that the user can

define, according to the meaning he knows they may have. We haven't defined any, due to lack of medical knowledge, so this feature won't be used.

The results of running this model on the dataset are:

Table 7: GLM Original

Prediction	Reference	
	False	True
False	80	14
True	45	10

Table 8: GLM Removed

Prediction	Reference	
	False	True
False	86	13
True	41	9

Table 9: F2 on GLM

	Original	Removed
Linear Model	0.6734007	0.7084020

The $F2$ score is not as bad as expected. It fact, it performs better than KNN despite the number of rTpF being higher.

7 Results obtained using non-linear methods

7.1 Random Forest

Random forests build decision trees using bagging to get different samples and combinations of variables to train on, and classify given test data by majority vote of all decision trees. This method is said to be quite robust even to imbalanced data, but does not create bags with equal parts of both classes, so we chose to call the random forest method with our preprocessed data. As the already balanced train samples are then divided into bags the models should give a more or less good result in the testing phase. Although it is possible to define the depth of trees and number of variables to choose we did not set these parameters fixed, because after trying different combinations there was none to create results clearly superior to the others.

Table 10: RF Original

Prediction	Reference	
	False	True
False	76	11
True	49	13

Table 11: RF Removed

Prediction	Reference	
	False	True
False	78	12
True	49	10

Table 12: F2 on RF

	Original	Removed
Random Forest	0.6473595	0.6521739

The two confusion matrices show very similar results and thereby underline the characteristic of being robust, as our preprocessing does not significantly improve the results. This also show in the F2 score of the two test samples which is 0.647 for the original dataset and with 0.652 only a little better for the processed dataset. For the parameters of the models the *caret* package has chosen only one tree and one variable to classify.

7.2 Neural Network

Neural networks make use of the biological system of neurons nesting functions into each other to classify the given input data. In our case we used a single-hidden-layer neural network. We did not allow skipping becauses a test with the skipping option set did not improve the results.¿En serio? We neither set a fix regularization parameter because this also let to poorer results when testing new data on the calculated models.

Table 13: NN Original

Prediction	Reference	
	False	True
False	56	10
True	69	14

Table 14: NN Removed

Prediction	Reference	
	False	True
False	60	6
True	67	16

Table 15: F2 on NN

	Original	Removed
Neural New	0.4946996	0.5226481

The confusion matrices of the neural network also show an obvious similarity. It is remarkable that in both cases the neural network predicted about $\frac{3}{4}$ of

the true instances correctly which is a good result. But at the same time the predictions of the false instances only get few more than 50%. This leads to a lower F2 score which is 0.495 for the original and 0.523 for the processed dataset. As parameters mostly a decay of $\lambda = 0$ and a single neuron are chosen, with few outliers of up to 5 neurons and a decay of $\lambda = 0.0001$.

8 Desciption and justification of the final model chosen

8.1 Estimation of the generalization error

9 Self-assessment of successes, failures and doubts

9.1 Successes

- Learn how to train and evaluate different models in a context
- Implementation of F2 score
- Improvement of R programming skills

9.2 Failures

- Found no model that predicts better than the "always false model" (evaluated with F2 score)
- Even though the scores of the models are not that bad, the confusions matrices show that in total many instances are not predicted correctly
- Creo que nos ha faltado encontrar una buena función de evaluación, disinta a F2. Las confussion matrix que obtenemos me parecen muy bien, pero no los score

9.3 Doubts

- It seems that for KNN the Euclidean metric works better than the Manhattan metric (evaluated with F2 score)

10 Scientific and personal conclusions

11 Possible extensions and known limitations

Regarding the results of our chosen model we can see that even this model's rate of correct predictions is not very high. As we have only tried to predict with five different models, it is possible that a model we did not try to use would give better predictions for the given dataset. As the related paper suggests to use Support Vector Machines it might well be that such a model gives better predictions. Furthermore we have just begun learning how to use method of Machine Learning and therefore have few experience, so we even might not have found the best parameters to use on the models we chose.

Also we do not have much medical knowledge of the data we have been working with, so we could neither put more emphasis on possibly more significant variables nor declare variables or values as not significant. The modifications we made to the dataset were only based on removing obviously underrepresented or unrealistic data. In cooperation with someone who has medical knowledge of the dataset's variables maybe the process of training could be improved.

As a last limitation the dataset only contains few instances to work with, which are even fewer after the elimination of some instances that we defined as outliers. As the dataset was published in 2013 by now there could be more data available to train on, which we assume would also lead to better results.

Creo que se podría hacer la extensión de encontrar una métrica más adecuada que F2 para el problema que tenemos.