Data mining techniques applied to medicine

Miguel Alcón Doganoc
Universitat Politècnica de Catalunya
Barcelona, Spain
miguel.alcon@est.fib.upc.edu

ABSTRACT

The increasing amount of data collected in whatever field leads to Data Mining and Machine Learning to increase its importance in our lives. Because of that, we want to use this data to impact directly in the most important aspect of people's lives: health. In this work, we focus on the detection of the degree of presence of heart disease in a patient. We try different classifier methods to predict it using the well-known Hearth Disease data set [?] of the UCI Machine Learning Repository.

1 INTRODUCTION

2 DATA SET

Our work is based on the processed Cleveland data offered by the Hearth Disease data set. The Cleveland data is composed of 14 features of the patient and its heart, which are:

(1) **Age.** In years.
(2) **Sex.** Binary representation of the patient's gender.
(3) **Chest pain type.** Integer value between 0 and 3, representing if the pain is a typical angina, atypical angina, non-anginal pain or asymptomatic.
(4) **Resting blood pressure**. In mm Hg on admission to the hospital.
(5) **Serum cholesterol.** In mg/dl.
(6) **Fasting blood sugar.** Binary representation of: the fasting blood sugar of the patient > 120 mg/dl.
(7) **Resting electrocardiographic results.** Integer values between 0 and 2, representing the severity of the results.
(8) **Maximum heart rate achieved.** Integer value.
(9) **Exercise induced angina.** Binary representation of 'yes' or 'no'.
(10) **Old peak.** Decimal value that represents the ST depression induced by exercise relative to rest.
(11) **Slope.** Three integer values representing the slope of the peak exercise ST segment.
(12) **Number of major vessels colored by fluoroscopy** Integer value between 0 and 3.
(13) **Thalassemia.** Three integer values representing whether it is a fixed or reversible defect, or the observations are normal.
(14) **Target**. Integer value between 0 and 4 that represents the presence of hearth disease in the patient.

Related with the target value, it has 164, 55, 36, 35 and 13 instances with values 0, 1, 2, 3 and 4, respectively. This implies that we have an unbalanced data set, which can lead to prediction difficulties. For more information about the features take a look at [?].

3 OBJECTIVES

The main objective of our work is to achieve the best possible accuracy with a restricted number of classifier methods, which are Naive Bayes (NB), Nearest Neighbors (NN), Decision Trees (DT) and Super Vector Machines (SVM). We select these classifiers due to its relevance and renown.

4 EXPERIMENTATION

Our experiment suffered several modifications along its lifetime. We started with a straightforward experiment that divides data into train and test (with 80% of train data), trains different models using each of the classifiers and predicts the target value. We also performed a 10-fold cross validation with each of them. With this, we wanted to know if one of the classifiers works better by default, and get a first impression about how difficult will be predicting the target. As you can see in 1, we obtained very low accuracy in our first predictions. We had to improve that.

Method	NB	NN	DT	SVM
Train & Test	0.583	0.600	0.500	0.617
Cross-validation	0.534	0.536	0.510	0.603

Table 1: Accuracy predicting the target variable in the first experiment.

In order to improve the accuracy of our predictions, we decided to perform feature elimination. We implemented our version of this method, which is described in section 5 and depends on the classifier that we are training. With it, we select the less relevant features in the data set, i.e., the ones that make us fail more often, and we remove them. We repeated the first experiment with the new data. For each classification method, we obtained the following non-relevant features:

- **NB:** age, maximum heart rate achieved and number of major vessels colored by fluoroscopy.
- **NN:** resting blood pressure, serum cholesterol and maximum heart rate achieved.
- **DT:** age, sex, chest pain, exercise induced angina, and thalassemia.
- **SVM:**. any feature.

In table 2, you can see that removing all the selected features did not go well. With SVM, our feature selector did not select any feature, so we could not remove anything. Even though these bad results, we know that the elimination of the selected features can lead to better accuracy. We use this information in later experiments.

At this point, we included to our experiment the resultant confusion matrices of our predictions. They helped us to notice that the

Method	NB	NN	DT	SVM
Train & Test	0.55000	0.58333	0.43333	-
Cross-validation	0.51795	0.55649	0.43059	-

Table 2: Accuracy predicting the target variable after feature elimination.

predictions of the target for value 0 were a lot much more accurate than the other ones. You can see this in table 3, with the confusion matrix of the best classifier until now.

	0	1	2	3	4
0	35	0	0	1	0
1	6	0	1	1	1
2	1	1	2	1	0
3	1	2	2	0	2
4	2	0	0	1	0

Table 3: Confusion matrix resultant of the predictions with SVM.

The difference in accuracy between target 0 and the other values may be preceded by the unbalance of the data set. We tried to fix it with a new design of the experiment. We can divide this second experiment into three stages:

(1) **Predict whether the patient has presence of heart disease or not.** We start the first stage (binary prediction) with the change of the target value to 1 of all instances that have a target value greater than 0, we train the model with the data, and then we perform the predictions over it. These predictions are made with data from training and testing. With the train predictions, we want to obtain all the conflictive instances for taking them into account when training the model of the second stage.

(2) **Predict the degree of the presence of heart disease of patients who suffer from it.** We start the second stage (degree prediction) returning their real values to the target variable of all instances. Then, we take as the new training data set all instances of the first training data that have a target value greater than 0, together with the conflictive training data of the first stage. With it, we train the model and perform the final prediction of the instances that we predicted to 1 previously.

(3) **Fusion the results of previous stages.** In the third stage (fusion), we obtain the final accuracy and confusion matrix of the whole experiment.

In the first stage, before training, we performed the feature elimination with the binary targets. We only deleted the features when the predictions were better than without it. Only DT obtained better results, with an increment from 0.583 to 0.750. Selected features were age and serum cholesterol. The final accuracy of each stage is shown in table 4.

As we can see, for NB and DT we obtained better predictions, for NN worst and for SVM the same (compared with *Train & Test* field in table 1). We still had the same maximum accuracy, which is 0.617, so we had to keep experimenting if we wanted to get a better prediction process.

Stage	NB	NN	DT	SVM
Binary pred.	0.917	0.750	0.800	0.900
Degree pred.	0.095	0.095	0.269	0.167
Fusion	0.617	0.533	0.600	0.617

Table 4: Accuracy predicting the target variable in the second experiment.

Looking at table 4, we realized that each classifier obtains different accuracies for the degree predictions, whatever was the accuracy for the binary prediction. Hence, we thought that we could use different classifiers in stages 1 and 2 of the second experiment, in order to improve the whole prediction process. The third experiment just implements that. It performs the second experiment with different classifiers for both predicting stages. The final predictions of this experiments are shown in table 5. Each row represents the predictions calculated in the fusion process, using the classifiers on the left for the binary prediction, and the classifiers on the top for the degree prediction.

	NB	NN	DT	SVM
NB	0.616	0.683	0.650	0.650
NN	0.600	0.533	0.617	0.633
DT	0.483	0.566	0.600	0.533
SVM	0.583	0.650	0.650	0.617

Table 5: Accuracy predicting the target variable in the third experiment.

We finally achieved to increase the global prediction, from 0.617 to 0.683. The prediction process using NB in the first stage and NN in the second stage gave us this maximum accuracy. This is the process that we are going to select to perform the fourth and last experiment.

The fourth experiment consists on performing feature elimination to obtain the optimum prediction process combining NB and NN. As you can see in table 6, sex was the only feature that was really unnecessary for the combination of NB and NN. Deleting it, we obtained an accuracy of 0.70, almost 0.1 more than at the start. In table 7 we show the confusion matrix of this, the best classification process that we achieved.

5 IMPLEMENTATION

All the experiments were programmed with Python 3. We used the well-know *scikit-kearn* Python package to apply the classifier methods, cross-validation, train and test split, etc.

Feature elimination was implemented by ourselves. For the selection of the features, the methodology that we used is the following: for one classifier method and for each possible feature, we delete the feature from both training and testing data, we train the model,

Erased feature	Accuracy
(1)	0.667
(2)	0.700
(3)	0.667
(4)	0.650
(5)	0.633
(6)	0.683
(7)	0.650
(8)	0.633
(9)	0.667
(10)	0.650
(11)	0.667
(12)	0.683
(13)	0.650

Table 6: Accuracy of the classifier process deleting one feature.

	0	**1**	**2**	**3**	**4**
0	35	0	0	1	0
1	3	2	2	2	0
2	0	1	3	1	0
3	1	3	1	2	0
4	0	3	0	0	0

Table 7: Confusion matrix of the final classification process.

we predict the testing instances, and finally we compute the accuracy of the model. If this accuracy is greater than the one obtained without deleting the feature, we consider that it is not relevant, so it can be permanently removed.

For more details of the implementation you can always take a look to the Python script that we developed.

6 CONCLUSION