Data Glacier

Your Deep Learning Partner

Healthcare - Persistency of a Drug

Groupe name: Ehiafarin
Country: Iran
Groupe member: Alireza Ehiaei
Email: arh.ehiaei@yahoo.com
Specialization: Data Science

Exploratory Data Analysis

September-2021

Table Of Contents

Introduction

Machine learning methods would be the best approach to analyze the efficiency of drugs due to the high number of factors and their complex relationship that affect on the results of drug therapy.

Machine learning methods have higher accuracy in the prediction of drug administration results such as the drug persistency of patients that is the aim of this research.

Before starting to use methods, we will have a look at the data we are going to use in this model and the problem we want to solve.

Data understanding

There are 3424 rows (patients) and 69 columns (features). The target variable is the Persistency_Flag variable that includes 1289 drug persistent patients and 2135 non persistent patients.

In previous research, I have discussed three ways to handle missing values. So, in this research we focus on development the predicting model, testing and evaluating its accuracy.

Problem Statement

After reading the data, the data set is separated into a y vector that is the Persistency_Flag variable and an X matrix of explanatory variables including other variables. The problem is predicting y given a set of variables of the X matrix.

In this research different predicting models have been developed to see which one has higher accuracy.

Random Forests

The first method we used is random forest. Random forest is a Supervised Machine Learning Algorithm that is used widely in Classification and Regression problems. It builds decision trees on different samples and takes their majority vote for classification and average in case of regression.

```
1  import sklearn as sk
2  from sklearn.ensemble import RandomForestClassifier
3
4  y = data_without_null_1.iloc[:,1]
5  X = data_without_null_1.iloc[:,1:]
6
7  RF = RandomForestClassifier(n_estimators=100, max_depth=2, random_state=0)
8
```

Method 1, Random Forests

We can again fit them using sklearn, and use them to predict outcomes, as well as get mean prediction accuracy. The accuracy of this method on our data is 0.89.

```
RF.fit(X, y)

# define the model evaluation procedure
cv = KFold(n_splits=3, shuffle=True, random_state=1)
# evaluate the model
result = cross_val_score(RF, X, y, cv=cv, scoring='accuracy')
# report the mean performance
print('Accuracy of the model is: %.3f' % result.mean())
```

```
Accuracy of the model is: 0.897
```

k-Fold Cross-Validation

In the above code k-Fold Cross-Validation is used to determine the accuracy of the model (0.89). It is common to evaluate machine learning models on a dataset using k-fold cross-validation.

The k-fold cross-validation procedure divides a limited dataset into k non-overlapping folds. Each of the k folds is given an opportunity to be used as a held-back test set, whilst all other folds collectively are used as a training dataset. A total of k models are fit and evaluated on the k hold-out test sets and the mean performance is reported.

Prediction

Now, we can use the developed model to predict the drug persistency of a patient. For example, assume we want to use it for a patient with information mentioned in the row 3000:

```
yhat = RF.predict([list(data_without_null_1.iloc[3000,1:])])
if yhat == 1:
    print('The patient is predicted to be persistent' )
else:
    print('The patient is predicted to be non-persistent')
```

```
The patient is predicted to be persistent
```

ROC Curves

In this part, we discover Receiver Operating Characteristic curve, or ROC Curves that is used to interpret the prediction of probabilities for binary classification problems.

When making a prediction for a binary or two-class classification problem, there are two types of errors that we could make.

False Positive. Predict an event when there was no event.

False Negative. Predict no event when in fact there was an event.

ROC Curves

ROC is a plot of the false positive rate (x-axis) versus the true positive rate (y-axis) for a number of different candidate threshold values between 0.0 and 1.0. Put another way, it plots the false alarm rate versus the hit rate.

The true positive rate is calculated as the number of true positives divided by the sum of the number of true positives and the number of false negatives. It describes how good the model is at predicting the positive class when the actual outcome is positive.

ROC Curves summarize the trade-off between the true positive rate and false positive rate for a predictive model using different probability thresholds.

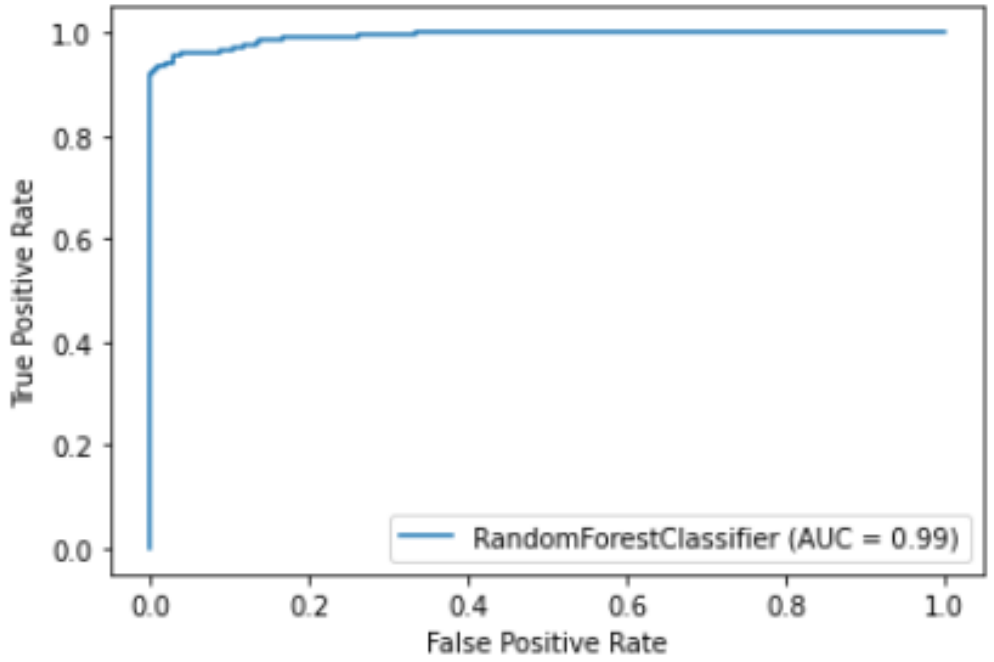
ROC Curves

The area under the curve (AUC) can be used as a summary of the model skill. A skilful model will assign a higher probability to a randomly chosen real positive occurrence than a negative occurrence on average. This is what we mean when we say that the model has skill. Generally, skilful models are represented by curves that bow up to the top left of the plot.

A no-skill classifier is one that cannot discriminate between the classes and would predict a random class or a constant class in all cases.

ROC Curves

As it is shown the are under curve for the random forest we developed is big enough to trust to the predictions of the model.



Method 2, Logistic Regression

Logistic Regression is a type of Generalized Linear Model (GLM) that uses a logistic function to model a binary variable based on any kind of independent variables.

Logistic regression is used to obtain odds ratio in the presence of more than one explanatory variable. The procedure is quite similar to multiple linear regression, with the exception that the response variable is binomial.

Method 2, Logistic Regression

Logistic regression is applied to predict the categorical dependent variable, and our problem is predicting a binary target variable. So, logistic regression can be a good choice to solve our classification.

```
import sklearn as sk
from sklearn.linear_model import LogisticRegression
import pandas as pd
import os

y = data_without_null_1.iloc[:,1]
X = data_without_null_1.iloc[:,1:]

LR = LogisticRegression(random_state=0).fit(X, y)
# define the model evaluation procedure
cv = KFold(n_splits=3, shuffle=True, random_state=1)
# evaluate the model
result = cross_val_score(model, X, y, cv=cv, scoring='accuracy')
# report the mean performance
print('Accuracy: %.3f' % result.mean())
```

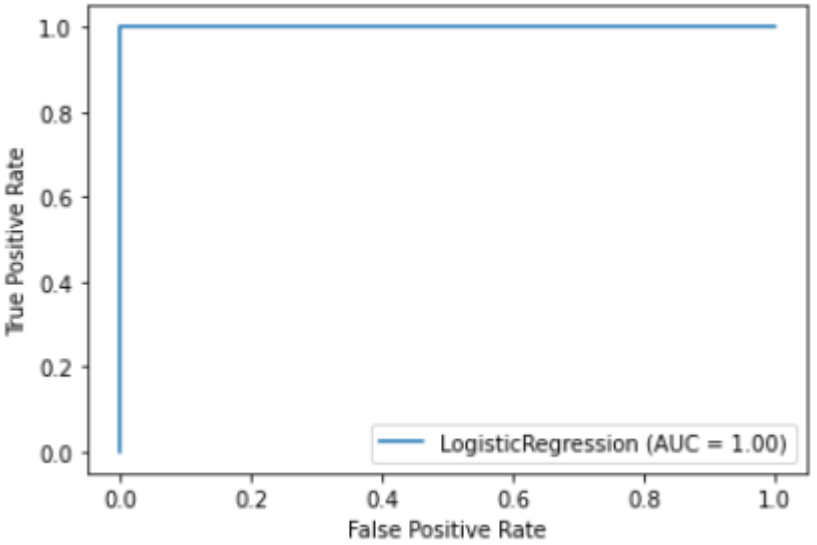
```
Accuracy: 0.811
```

Prediction

The accuracy of LR developed model is high and similar to RF model in previous part it can used for predicting drug persistency of patients.

```
yhat = LR.predict([list(data_without_null_1.iloc[3000,1:])])
if yhat == 1:
    print('The patient is predicted to have drug persistence.' )
else:
    print('The patient is predicted to not have drug persistence.')
```

```
The patient is predicted to have drug persistence.
```



Final Recommendation

All machine learning models are some kind of mathematical model that need numbers to work with. The target variable in this research was a categorical type that is a type of data that is used to group information with similar characteristics, while numerical data is a type of data that expresses information in the form of numbers.

Therefore we encoded the categorical variable into numbers to be able to use in different machine learning methods.

Final Recommendation

If we had just few explanatory variables, we could use frequency tables, pie charts, or bar charts to analyse their relationship on the target variable. But due to the high number of variables we used two machine learning methods to predict the drug persistency of patients. Both have more than 0.8% accuracy in prediction but the random forest method was more accurate and has a higher true and false positive rate as the number of explanatory variables increases in a dataset in compare to logistic regression.

Data and code link

Data and code are uploaded at:

https://github.com/Alireza-Ehiaei/Data_Sciences/tree/main/Drug_Persistency1/EDA

Thank You