BioMed Team Proposal

E-mail: biomed@chalearn.org)
GitHub: https://github.com/BioMed-AIC/Opioids

Mariem BOUHAHA (bouhahamariem7@gmail.com)
Thomas FOLTETE (thomas.foltete@gmail.com)
Guillaume WELSCH (guillaume.welsch@gmail.com)
Guillaume COLLIN (guillaume.collin@etu.emse.fr)
Hoël PLANTEC (hoelplantec@hotmail.fr)

Saving Lives by Detecting Opioid Prescribers

Context

Drug overdose and opioid-involved deaths continue to increase in the United States. Opioids are powerful pain relievers that disrupt transmission of pain signals through the nervous system. According to the *Centers for Disease Control*, more than six out of ten overdose deaths involve an opioid. Unfortunately, the number has been in perpetual increase during the past 15 years (1), highlighting a growing addiction problem. As to avoid systematic opioid prescriptions as pain relievers, it would be interesting to identify unnecessary prescriptions, limiting therefore opioid addiction rates and overdose deaths.

Our aim behind this challenge is to use predictive modelling to identify significant opioid prescriptions. For that we will be using data (2) with prescription records for 250 common opioid and non-opioid drugs written by 25.000 professionals in 2014.

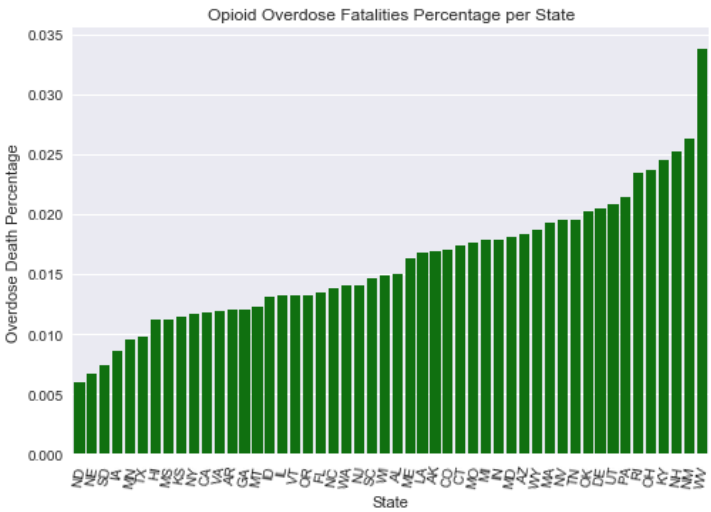


Figure 1: Percentage of opioid overdose deaths per state in 2014

Material and Method

Presenting the data

The dataset we used is a subset of a larger one, published by (2), containing *25.000 prescription* instances. For each instance, we have prescriber-related characteristics like his *gender*, his *specialty* and his *state* of origin. The data also contains *250 numerical features* related to drug names, where for each prescriber, we indicate the number of times his prescriptions included these drugs. Instances are labeled using a binary variable that takes **1** whenever the individual prescribed opiate drugs more than 10 times in the year and **0** otherwise. The main data is in *prescriber-info.csv*. There is also *opioids.csv* that contains the names of all opioid drugs included in the data and *overdoses.csv* that contains information on opioid-related drug overdose fatalities.

In the following we visualize the composition of the categorical features we have like gender, state and specialty.

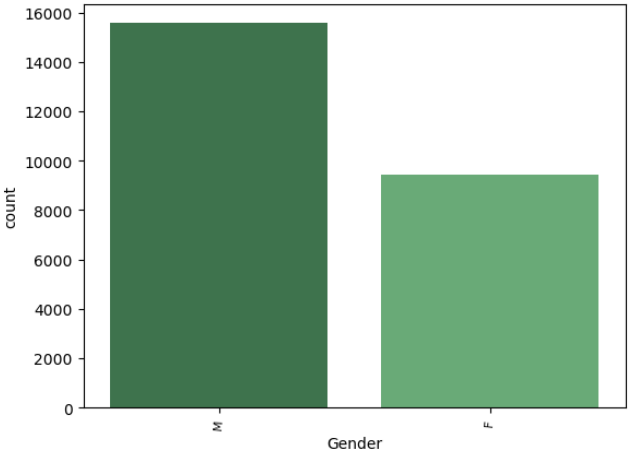


Figure 2: Distribution of male and female prescribers

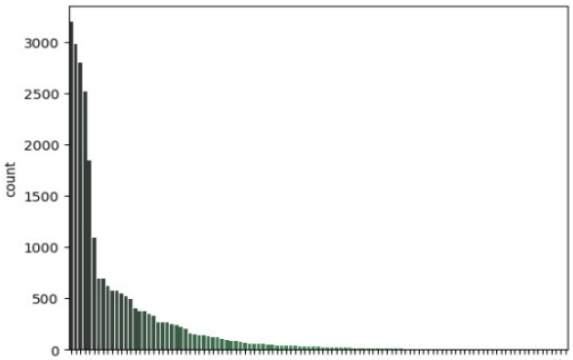


Figure 3: Specialty CountPlot

In our data, we have 109 different prescriber specialties where most of them are not very frequent, as shown in the above figure. Top 6 most frequent ones are shown in the table below:

Internal Medicine	3194
Family Practice	2975
Dentist	2800
Nurse Practitioner	2512
Physician Assistant	1839
Emergency Medicine	1087

Figure 4: Most frequent specialties

The same behavior was observed for the state feature. We have 57 unique state attributes, distributed among prescribes as follows:

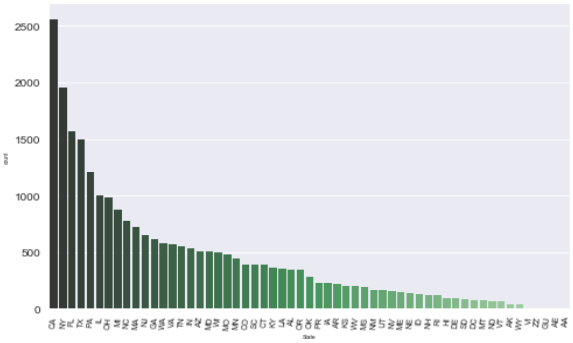


Figure 5: State CountPlot

As for the target variable, we have the following class (0/1) distribution:

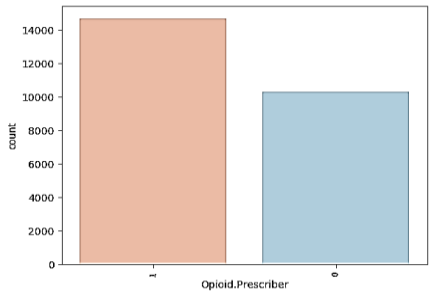


Figure 6: Prescriber Class distribution

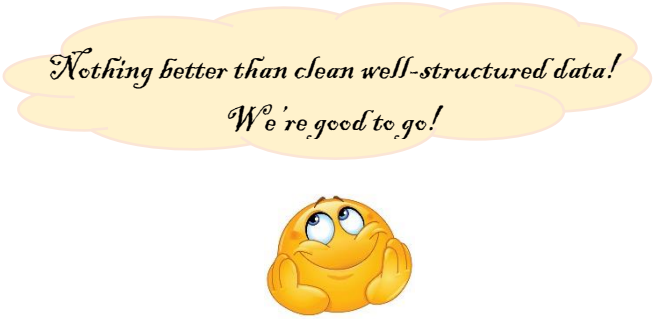
These observations should therefore be taken into consideration while preprocessing the data.

Preprocessing the data

As we all know, machine learning is only as effective as the data that drives it. In other words, if you want to implement effective machine learning, you need to pay attention to data quality. For that, we had to preprocess and clean our data before starting modeling.

Preprocessing steps included:

- Modifying Opioid names-formation to match the corresponding attributes in the main data file.
- Cleaning up Categorical variables (identifying aberrant categories, correcting typos and cleaning the abbreviations). For that, we replaced the least frequent categories by a new one called "other". For *Specialty*, this resulted in a reduction of categories from 109 to 44 unique values. For *State*, we now have 53 unique states.
- Factorizing categorical features (like Gender, State, Specialty).
- Eliminating opioid attributes. This step is essential in the context of our objective. Since we aim at detecting opioid prescribers, opioid-related attributes shall be removed, otherwise there would be a data leakage and we would be cheating, since whenever a professional prescribes one of these drugs, he would have *class 1* in the target variable! For that, we were based on the *opioids.csv* file to identify Opioids from Non-Opioid drugs. This resulted in the elimination of 11 column variables from the initial data.



Predictive Modeling

Once our data is ready, we moved to classification, where we separate opioid (class 1) from non-opioid (class 0) prescribers. For that, we used a Random Forest Classifier with parameter tuning using *grid search with 10-fold Cross-Validation*. Random forest is an ensemble classifier that consists of many decision trees and outputs the class that is the mode of the classes output by individual trees. It is one of the most accurate learning algorithms available. What's also good about RF is that it runs efficiently on large databases and can handle multiple input variables. However, its performance is highly related to the choice of the parameters it takes as input. That's why GridSearch Cross-Validation is helpful, since it returns the "optimal" parameters to use after an exhaustive search over specified parameter values for the estimator. We chose to tune the number of trees, the minimum samples per leaf node and the splitting criterion.

```
parameters_grid = {
    'n_estimators': [50, 100, 150],
    'min_samples_leaf': [5,10,15],
    'criterion': ['entropy', 'gini']
    }
```

To evaluate the model's performance, we are going to use the *Area Under ROC Curve*, where true positive rates are plotted against false positive rates. To train the model, we used 80% of the data. The remaining 20% were used for testing.

Preliminary Results

In our experiment, we fitted a Random Forest Classifier with the following optimal hyper-parameters, resulting from the 10-fold cross validation:

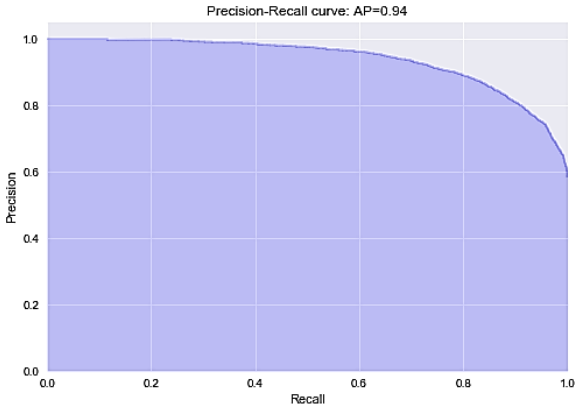
Hyper-parameter	Value
n_estimators	150
min_samples_leaf	5
criterion	'gini'

For the rest of the algorithm's hyper-parameters we used the default sklearn's values. That is mainly, max_features = sqrt(n_features), Max_depth= None and min_samples_split =2.

AUC = 0.82

It seems that we did a good job, we obtained a good accuracy just by considering a doctor's specialty and non-opioid prescription trends, allowing us to predict how likely it is that he prescribes a significant quantity of opiates.

We shall also note that, within the context of detecting significant opioid prescribers, it would be wiser to care more about precision than accuracy, since a false positive would lead to harm "innocent" professionals. For that, we may look at the precision-recall curve, which shows the tradeoff between precision and recall for different thresholds.



We obtained a high area under the curve, representing both high recall and high precision. This means that we have a low false positive rate, and a low false negative rate. High scores for both show that the classifier is returning accurate results (high precision), as well as returning a majority of all positive results (high recall).

As we previously mentioned, one of the pros to use Random Forests is the ability to assess features importance. We noted that some features had zero and close-to-zero importance, while the most important ones seemed to be "Specialty".