Research Internship (PRe)

Discovering novel drug-drug interactions using machine learning and clinical data

Lucien Perdrix

June 2023 - August 2023

University's internship adviser :
Andrea Simonetto
andrea.simonetto@ensta-paris.fr
Department of Applied Mathematics
Optimization and control team

Companie's internship adviser :
Nicholas Tatonetti
nicholas.tatonetti@cshs.org
Department of Computational Biomedicine

Disclosure agreement

This document is non-confidential. It may be communicated outside of school in hard copy but also broadcast in electronic format.

Acknowledgements

I express my gratitude to Mr. Nicholas P. Tatonetti for approving my application to join his research lab, granting me the opportunity to explore his research and interact with his team. Through my engagement with his medical research team, I gained a profound understanding of the significance of Biomedical Informatics and its cutting-edge techniques. Additionally, I acquired valuable insights into leadership and team management, as Mr. Tatonetti consistently exhibits a supportive and motivating approach toward every member of his laboratory.

I also want to express my appreciation to the fellow members of the lab for their warm reception and unwavering support throughout the entire internship period. Their welcoming attitude contributed significantly to my sense of belonging and integration within the team.

Abstract

Discovering novel drug-drug interactions using machine learning and clinical data

Drug interactions are a major concern with rising polypharmacy, and patients/providers must be vigilant about preventing hazardous drug combinations to avoid unexpected adverse outcomes. In this study we are aiming to use machine learning to discover novel drug-drug interactions and to validate our results using clinical data.

The state of the art machine learning methods on this field of research are using different kind of medical data to find potential drug-drug interactions. Every results of these models have to be validate using clinical data. The state of the art method to validate drug safety hypothesis is the retrospective cohort study. In most of the previous paper and study about drug-drug interaction, the focus is on a specific adverse event and pair of drugs.

During this research project, my contribution was :

- Reimplementing a signal detection machine learning model and adapt it to different training and application data
- Designing a validation protocol using a new database of clinical data
- Adapting actual method in order to generalize it and to be able to process our study to a large number of adverse events

Keywords : Drug-drug interaction, adverse event, machine learning, clinical data, drug safety

Table of contents

List of figures

List of Tables

Introduction

Drug-drug interactions is one of the major risks of unexpected adverse events[1]. When multiple drugs are taken concurrently, they can interact with each other in various ways, leading to changes in their absorption, distribution, metabolism, or elimination within the body. The problem of drug-drug interaction becomes even a more important concern when 72% of US citizens are taking two or more drugs (The National Health and Nutrition Examination Survey) and elderly patients with 65 years of age and older use 2 to 6 prescribed drugs and one to four non-prescribes[2]. The high prevalence of individuals simultaneously taking multiple medications underscores the importance of understanding and addressing drug-drug interactions. However, the identification of novel drug-drug interactions remains a challenging task, as it is an area of study that receives less attention in clinical trials compared to the focus on adverse reactions caused by single drugs [3]. The complex nature of drug interactions, combined with the vast number of potential drug combinations, makes it difficult to thoroughly investigate and predict all possible interactions. Nonetheless, comprehensive knowledge and research in this area are crucial to ensure patient safety and minimize the occurrence of adverse events.

To address this challenge, one potential approach is the utilization of signal detection methods to identify hidden drug-drug interactions using data from adverse drug event reporting systems. This methodology involves selecting a specific adverse event and constructing a profile based on the known side effects of individual drugs that can potentially lead to this particular adverse event. Subsequently, the method involves examining pairs of drugs that correspond to this profile, enabling the prediction of potential interactions between them.

In our study, we will begin by describing the signal detection methods we implemented. We will use and evaluate the method on a large amount of adverse events. We will then need to find and use a method to corroborate the results of our model. Specifically, we will focus on selecting particular adverse events for which our signal detection methods have demonstrated high performance. This particular adverse event should also have an effective and feasible way to validate the results. The validation method will be attempted on clinical data.

1 The signal detection model

1.1 Introduction to the model

In 2011 Nicholas Tatonetti and the TLab develop a new machine learning algorithm for discovering hidden DDIs [4]. This model is a supervised machine learning model which at this time was trained using a large amount of adverse event reports from the FDA's publicly available AERS. The study focused on eight Severe Adverse Effect (SAE) and aimed to create profiles for these SAE using logistical regression and adverse effects of the drugs that cause these SAEs. Then, it involves applying these trained models on pairs of medications and find pairs of drug that match these profiles. The first task in this project was to adapt and reimplement this project using a new database and more modern methods in order to identify Drug Drug Interactions candidates that are putatively responsible for the adverse event we study on. My approach is to build a binary classification model to class a drug pairs as responsible of a specific adverse event or not.

1.2 Useful TLab Database

1.2.1 OnSIDES

I will use the OnSIDES (ON label SIDE effectS resource) database, which was developed by TLab in 2022 (OnSIDES V2). This database incorporates the use of the PubMedBERT language model [5] and includes 200 manually curated labels [6]. By employing this model, over 2.7 million adverse reactions were extracted from approximately 42,000 drug labels sourced from the FDA drug structured labels. The performance of the model surpasses previous benchmarks, as evidenced by its improved F1 Score and Precision score compared to the previous best model, TAC [7]. I intend to utilize this comprehensive database for training machine learning models.

1.2.2 OFFSIDES and TwoSIDES

OffSIDES is a database created by TLab that provides a list of drug side-effects that are not yet listed in official FDA labels. It was created mining adverse event reports. I will use this data base to train the machine learning models.

TwoSIDES is quiet the same but for pair of drugs. Also it is a data base of drug side-effects related to pair of drugs.

I will use this data as application data in order to discover potential drug pairs candidates.

1.3 The supervised machine learning model

In the following section, I will describe the machine learning model we used. The model is related to one specific adverse event. In order to describe the model and evaluate its performance, we focused on few adverse events that represent certain specificity of the model. We will later expand our approach to a broader sample of adverse events.

1.3.1 Description of the model

The machine learning model used in this study is a logistic regression binary classifier. To train a supervised machine learning classification model, we require two variables for each example:

- **Features**: These are the independent variables that the algorithm learns from and should be measurable. They encompass the characteristics or attributes of the examples that contribute to predicting the outcome.
- **Response**: This refers to the dependent variable that the model seeks to predict. In the case of a classification problem like this, the response variable represents the class or label assigned to each example, indicating whether it experiences the adverse event in question (in this case, "death") or not.

By incorporating these two kind of variables into the training process, we can develop a predictive model that learns from the features to classify example drugs and predict wether or not this drug is responsible for the adverse event we study on.

1.3.2 Construction of training data

1.3.2.1 Features

The features in my model are built using the frequency of reports linking an adverse event to a drug. I build these features using OffSIDES. OffSIDES database was constructed by extracting from adverse event reports drugs' names and adverse event's names. In OffSIDES, each row represents a drug and an adverse event. To train the model, we need to construct the training data, it means acquire for a large amount of examples drugs both the features variable and the associated response variable. One approach is to create a matrix of frequencies using OffSIDES. This matrix represents the association between drugs and adverse events and has the following structure:

	AE 1	AE 2	...	AE NAE
Drug 1	f_{11}	f_{12}	...	f_{1NAE}
Drug 2	f_{21}	f_{22}	...	f_{2NAE}
...	...	...	...	...
Drug ND	f_{ND1}	f_{ND2}	...	f_{NDNAE}

Table 1.1: Training Frequency matrix of drug-adverse event associations.

In this table :

- ND : number of unique drugs in OffSIDES database
- NAE : number of unique adverse event in OffSIDES database
- $f_{ij} = \frac{Number\ of\ reports\ with\ D_i\ and\ AE_j}{Number\ of\ reports\ with\ D_i}$

The frequencies are also available in OffSIDES. Also by browsing OffSIDES we can for each drugs and each AE find the corresponding frequency. (N.B.: if there is no row of OffSIDES with a specific drug and a specific AE, the corresponding frequency in the matrix will be 0)

1.3.2.2 Response variable

In our model, the response variable is a binary variable (0 or 1) indicating whether the sample drug is associated with the adverse event we stuty on or not. It represents whether the adverse event is considered responsible for the occurrence of the adverse event.

In order to find the response data associated with the features, I used OnSIDES. For each training example, which corresponds to each drug in the OffSIDES dataset and each row of the frequency matrix, I checked whether or not the drug is associated with the adverse event using the OnSIDES database.

By doing so, I computed a binary vector consisting of 0s and 1s, where each element indicates whether the drug is associated with the adverse event in OnSIDES or not. This binary vector serves as the response variable for the training set, denoted as "y".

This approach allows us to pair the frequency matrix (containing the features) with the corresponding response variable, enabling the training of the predictive model.

After reviewing my results, I realized that a lot of negative labels in the response variable was "false negative". Indeed, when I built the response variable, I looked for each drugs in OffSIDES and labelled it positive or negative whether or not this drugs was related in OnSIDES to the studied adverse event. Some drugs were labelled negative not because they were not realted to the adverse event but ratheur because we were not able to find the drug in OnSIDES.

The process of finding the drugs of OffSIDES in OnSIDES use a mapping method. Indeed, I need to map element of OffSIDES to OnSIDES.

Here is a brief description of my basic process to build the response variable :

- Build a list of all OnSIDES element related to the studied adverse event
- For each drugs in OffSIDES, if the drugs is in the previous list, label it to 1 (0 otherwise)

My first approach to compare drugs of OffSIDES and drugs in OnSIDES list was based on drugs's name comparison. I evaluated the method by trying to map both database (find the number of OnSIDES drugs which are also in OffSIDES). With this drug's name comparison approach I found a score of 32%(of OnSIDES elements in OffSIDES). Then, I tried to improve this mapping using RxNorm[8]. RxNorm is a normalized system for drugs. RxNorm provides an index for concepts. This index is used in OnSIDES and OffSIDES. By using these indexes to map the two dataset

I improved the score to 79%. This could be explained by the fact that it is more efficient to compare numbers than strings. I finally tried a last approach using an RxNorm API furnished by the National Library of Medicine[9] to map the ids in OffSIDES to the SPL ids in OnSIDES which are unique identifier for drugs' labels. This method was approximately 10% better than the previous one using RxNorm but it was also much more time computationnaly expensive due to the API use (that is why I was not able to perform the entire mapping and have a precise evaluation of the performance). In a compromise time/performance I choose the second method using RxNorm.

After choosing the mapping method, I was finally able to build the response variable and I have now built the full training dataset :

	AE 1	AE 2	...	AE NAE	Label
Drug 1	f_{11}	f_{12}	...	f_{1NAE}	1
Drug 2	f_{21}	f_{22}	...	f_{2NAE}	0
...	...	...	...	...	...
Drug ND	f_{ND1}	f_{ND2}	...	f_{NDNAE}	0

Table 1.2: Training Frequency matrix of drug-adverse event associations

1.3.3 The model

In this study I trained a Logistic Regression model. In logistic regression, the response variable y_i follows a Bernoulli distribution with a parameter π that represents the probability of the positive class given certain features X_i. $\pi(X_i) = \mathbb{P}(y_i = 1|x_i)$

Considering the logit function :

$$logit(\pi(X_i)) = log\left(\frac{\pi(X_i)}{1-\pi(X_i)}\right)$$

The logistic regression assumes that the *logit* function is linear in X_i, such that :

$$logit(\pi(X_i)) = \beta_0 + \beta_1 X_{i1} + ... + \beta_{iN}$$

The study consist now at estimate the β parameters that fit best to the data. We use for this a Maximum Likelihood Estimation method. Thus we need to maximize the following expression :

$$\ell(\beta; y) = \sum_{k=1}^{n} log(\mathbb{P}(y_k) \tag{1.1}$$

$$\tag{1.2}$$

with $y = (Y_1, ..., Y_N)$

Given that N independent Bernoulli distributions follow a Binomial distribution with parameter N, we can express this relationship as follows :

$$\forall i, Y_i \sim \mathcal{B}(N, \pi(x_i))$$

We can now find the maximum argument of this function using for example the Newton method.

For my project I use Python environment of programming and everything is already programmed in the $sklearn.linear_model$ library. Such that we can easily fit a logistic regression given X and y as training data with the following code :

```
from sklearn.linear_model import LogisticRegression

model=LogisticRegression()
model.fit(X,y)
```

1.3.4 Optimization of model's hyperparameters

After choosing the model and build the training data, the next step was to change the model hyperparameters to improve its performance.

1.3.4.1 Imbalanced class The first issue I met when I tried to train the model is the imbalanced class. Indeed after the previous step consisting of building the training dataset, I found a small amount of positive labels . If we trained the model naively it will always return 0 as output. Indeed, this will have a high accuracy. However, in my study the accuracy is not a good indicator. Indeed, I am aiming to discover new drug-drug interactions (positive labels) and in term of health safety it is better to have False Positive case than False Negative. Thus we will look at another classic indicator so called the *recall* score which is defined by :

$$Recall = \frac{TP}{TP + FN}$$

where TP is the number of true positive predictions and FN is the number of false negative predictions

The expression shows that a good recall score demonstrate the ability of the model to find positive case.

To solve the problem of imbalanced class I change a hyperparameter in the $LogisticRegression$ function. By setting the $class_weight$ parameter to $'balanced'$, the weight of both class is normalized by its frequency in the training set :

```
model=LogisticRegression(class_weight='balanced')
model.fit(X,y)
```

The following figure illustrates the impact of $class\_weight$ parameter on model's performance. It improves the recall score :

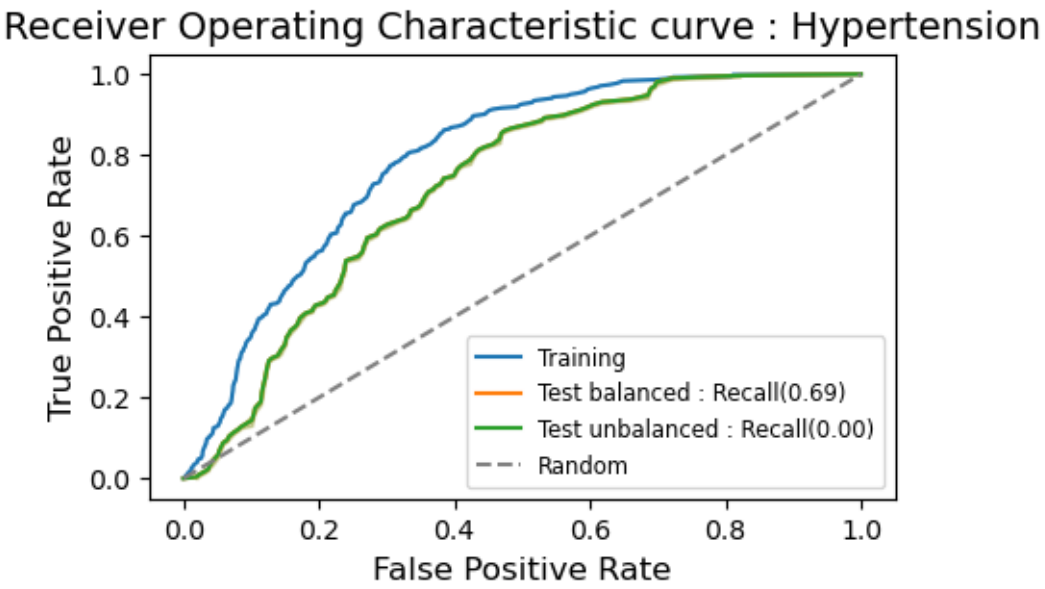

Figure 1.1: Model's performance with and without balanced class

After different tries, I also process to a resampling method to train the model : cross-validation. It takes few steps :

- Split the training data into n folds ($n = 5$)
- For each fold, the model is trained on the remaining k-1 folds and evaluated on the current fold. This process is repeated k times, with each fold acting as the evaluation set once.
- We do not retrain the model from zero at every step so that we use all the training set to fit the final model
- Compute the mean score of each steps

Training the model using cross-validation is good to prevent our model to overfit. Indeed with the small number of positive samples, our model could overfit to the negative samples and never predict positive samples. Cross-validation allows us to train on more data and to prevent the model from this. Python furnish an adapted function to train a model using cross-validation which is : $LogisticRegressionCV$ from the library $sklearn.linear_model$. At last but not least, the mapping method chosen earlier was also a big improvement because it allows me to get more positive samples. The performance on recall score was approximately 30% better with the mapping using RxNorm (Recall : 0.81) than for the mapping using drug's names(Recall : 0.63).

1.3.4.2 Overfitting problem In the case of our study there is a massive risk of overfitting when we train the model.

Overfitting is when the model is perfeclty (even too) fit to the training data which makes it less performant with the testing data.

I have already mentionned this problem above with imbalanced class but there is another kind of overfitting risk.

After the previous step we recovered 2726 example drugs in the training set and we recovered 14543 features from OffSIDES. The large number of features compared to training samples could cause overfitting.

One solution is the feature selection, in other words to use only a smaller part of the features to train the model. In the case of the original article, they used the forward feature selection. This selection uses Fisher's exact test method to perform an analyse of enrichment. However, this is a old-fashioned way to process it.

I will use a regularization term to prevent from over fitting [10]. The idea behind is quiet simple, fitting our logistic regression is similar to solve the optimization problem :

$$\min_{\beta} y - \beta x$$

There is a risk of overfitting when the dimension of β is higher than x dimension. Indeed, in this case we have more than one β coefficient to fit for each sample. Also we can perfectly fit β to the training data. However, the model will perform very poorly in generalization. Doing a feature selection is equivalent to set some coefficients of β to 0, to reduce β dimension thus. We can do this by penalizing the objective function :

$$\min_{\beta} y - \beta x + \lambda f(\beta)$$

λ is called the regularization strength.
f could be 3 different penalization functions :

- ℓ_1 regularization : $f(\beta) = |\beta|$
- ℓ_2 regularization : $f(\beta) = |\beta|^2$
- $elastic - net$ regularization : $f(\beta) = |\beta| + |\beta|^2$

The *LogisticRegression* function in Python does not allow me to do elastic-net regularization. Therefore I computed the cross-validation recall score and AUROC(Area Under the ROC curve) : it measures the model's ability to discriminate between positive and negative classes across different classification thresholds. The AUROC score summarizes the trade-off between the true positive rate (sensitivity) and the false positive rate (1 - specificity) across all possible classification thresholds.).

The following findings are presented :

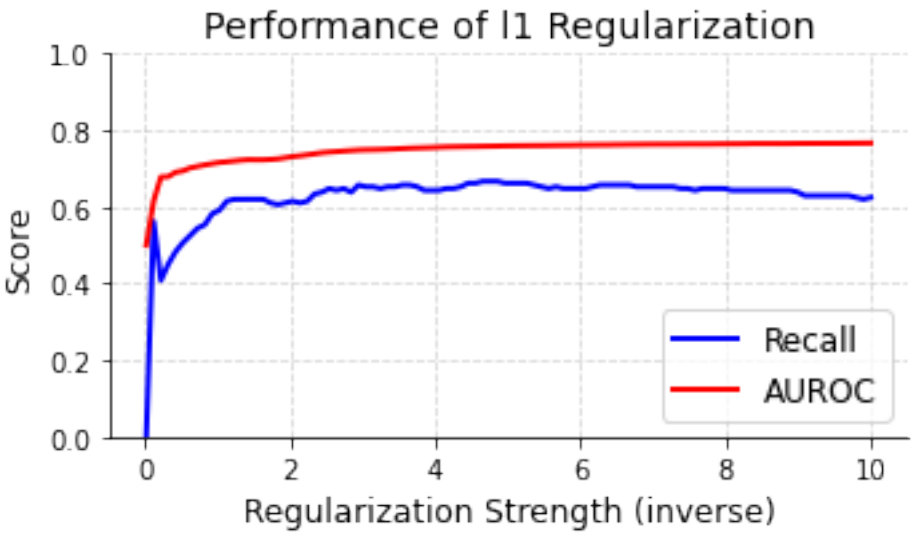

Figure 1.2: Performance of the model with ℓ_1 regularization

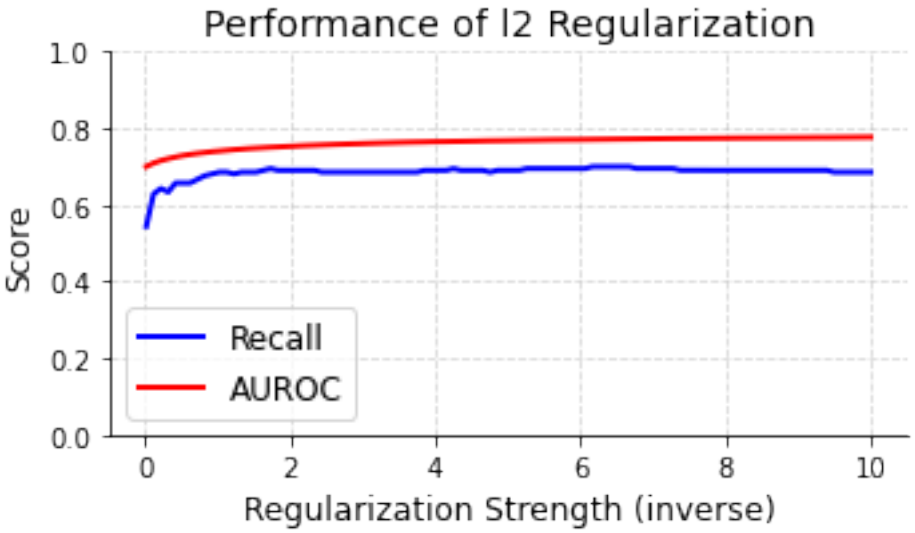

Figure 1.3: Performance of the model with ℓ_2 regularization

After analyzing the previous results, I find out that the best regularization was the ℓ_2 with a slightly better recall score. Then I choose the regularization strength parameter by maximizing the recall score.

For instance in the case of hyper tension we choose : $C = 6.2$ so a parameter $\lambda = 0.16$.

After that I compare the model trained with and without these methods to avoid overfitting to see if our method was really efficient. I took the example of hypertension detection with my model and I computed the ROC curves with (Figure 1.3) and without(Figure 1.4) regularization. We observe that without regularization the performance is really high on training set but there is a huge gap with testing performance especially with the recall score. That confirm what I was expecting from the theory above. When I added regularization(Figure 1.4), the performance on training set is of course not as good but the performance on testing set is much higher (almost two time better recall score).

Also, to conclude, the regularization applied to the model has decreased overfitting and increased the performance.

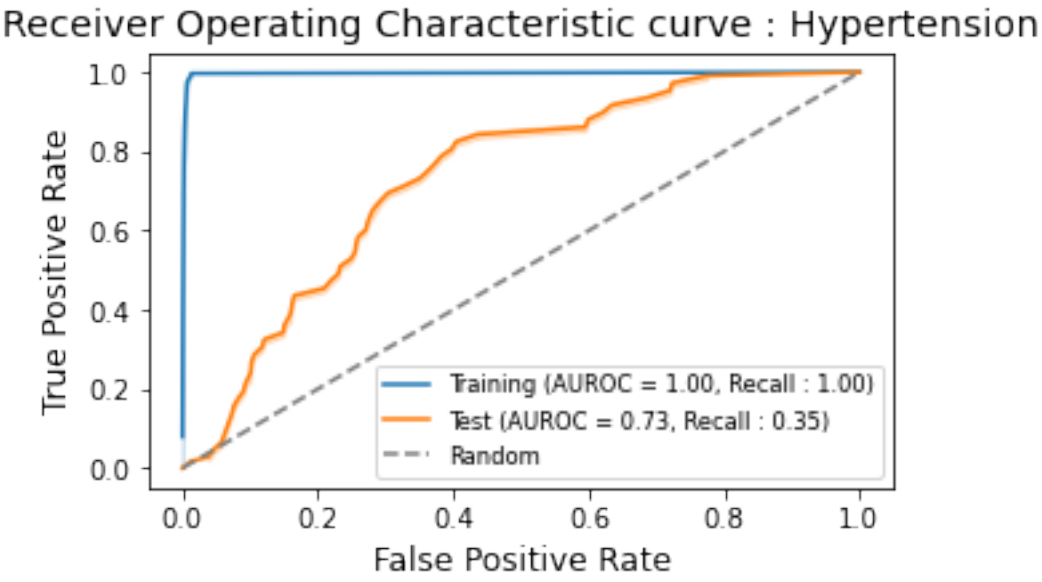

Figure 1.4: ROC curve without regularization

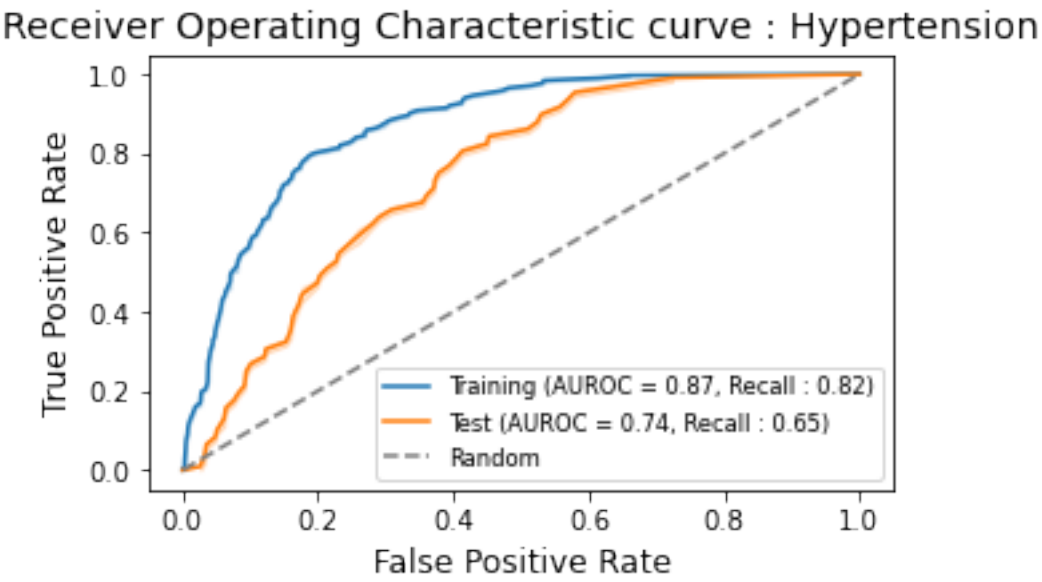

Figure 1.5: ROC curve with regularization

1.4 Conclusion

After dealing with all the issues we met and optimizing the hyperparameters of the model in order to improve its performance we are now able to use the model for a larger amount of adverse events. Indeed, we can build and train one different model for each adverse event we want just by changing the response variable of the training set using the method we described previously.

2 Application

2.1 Selection of the adverse events with the most positive responses

After having find the model and its hyperparameters, we are finally able to train it for different adverse events.

However, I found out that the model was not performing well in the case where positive labels in the response variable are in a very small number :

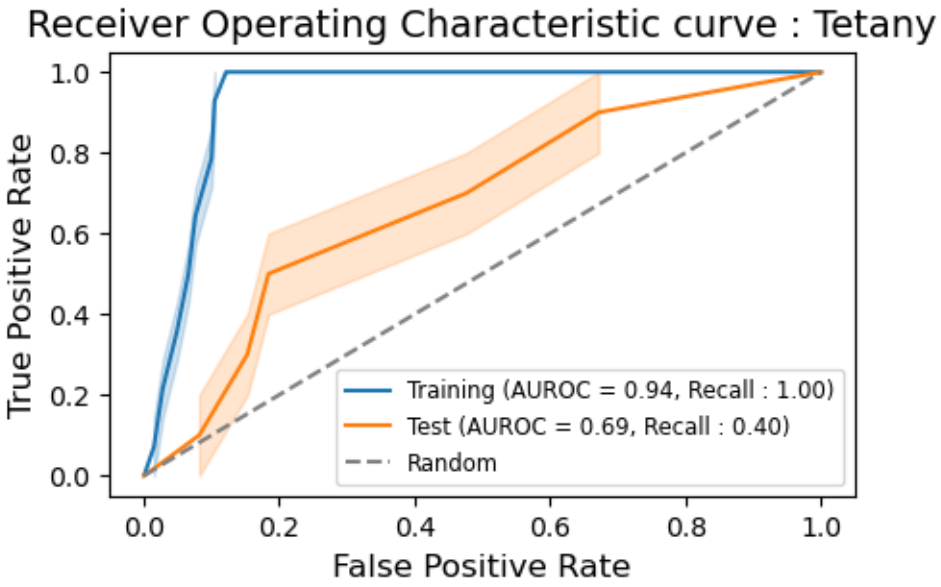

Figure 2.1: ROC curve for tetany (23 positive labels)

The form of the ROC curve indicates that the model tends to give always the same predicition : the majority class (0 in our case). In order to improve my results, I focused on two things :

- Improve the mapping between OnSIDES and OffSIDES to have more positive labels (described earlier)
- Select the adverse events I study on their number of positive labels

2.2 Train the model on the most dangerous adverse events

Moreover, in order to choose the adverse events I study on, I used a dataframe built by Undina Gisladottir, a member of the TLab. This dataframe identifies the most dangerous adverse events regarding of different criteria.

2.3 Adapt the model for each adverse event

For each model, so force for each adverse event we study on, the X training variable is always the same : it is the frequency matrix we described at the beginning of this project.

However, the y response variable is different. We build the specific response variable using the mapping method described above and the OnSIDES database checking for each drugs whether or not it is associated with the adverse event we study on.

2.4 Computational challenge

The next step in the project for me was to build a model for the larger amount of dangerous adverse events possible. Indeed, the idea of the project is to make a general approach and to be able to discover novel drug-drug interactions for the most adverse events possible. The computation time to build one model was of the order of few minutes. It is time consuming because for one model I need to optimize the regularization weight and process the cross-validation. However, for a large amount of models, I was confronting to days or even weeks of computation if I used my own device. Thus, I try to used the High Performance Computing device of the Department of Computational Biomedicine at Cedars-Sinai which provides to Cedars-Sinai's researcher a large amount of CPU and GPU to process complicated codes. However, I was not able to access to this ressource because of the hard protocol to get it and the fact that I could not have the access to the VPN.

During all the time I have I processed ten by ten the adverse events and at the end of my internship I have results for nearly 200 adverse events.

2.5 Results

Finally, I trained a Logistic regression for each of the most dangerous adverse events.

Afterwards, I used the models on data from TwoSIDES.

TwoSIDES has exactly the same structure than OffSIDES but for pair of drugs.Thus, I was able to build a frequency matrix for each adverse events and apply the models on these matrix. With data from TwoSIDES instead of OffSIDES like in the training, the model labels pair of drugs instead of single drugs. We then obtained for each adverse event a list of potential drug-drug interaction related to this adverse event.

The following figure represents a sample of my results :

	Recall	Auroc	Prediction
General physica	0,93	0,96	247,00
Febrile neutrope	0,89	0,95	536,00
Septic shock	0,85	0,92	691,00
Hip fracture	0,30	0,88	772,00
Drug abuse	0,68	0,88	1 045,00
Staphylococcal i	0,70	0,88	284,00
Type 2 diabetes	0,40	0,87	222,00
Drug dependenc	0,87	0,87	430,00
Psychotic disord	0,67	0,86	457,00
Suicide attempt	0,74	0,86	959,00

Figure 2.2: Sample of the results on each models trained

This tab represent for each adverse event, the performance of the model trained and the number of drug-drug interaction candidates(positive labels) extract of TwoSIDES for each adverse event.

3 Validation of the model on clinical data

The validation on clinical data is an important step in our work. Indeed our model was built to predict potential drug-drug interactions candidates that may increase the risk of several adverse events. We need to corroborate or refute the results of our model using clinical data.

This step could be really difficult depending on the adverse event studied or on the data we use. Some studies about the validation of drug-drug interaction hypotheses have already been made [11] and I was able to adapt it for my own case of study.

In the following part, I described the method used and the process of finding informations about each adverse event in order to figure out how to validate the potential drug-drug interactions for these adverse events.

3.1 Clinical database

For the validation step we need a database of clinical data. The database I used for the corroboration study is called $CS\_Analyse\_No\_Phi$. It is the PHI_free sibling database of $CS\_Analyse$ which means that there is no Protected Health Information in this database.

This database contains near real-time data of Cedars-Sinai hospital from all major care related data domains such as :

- Patient
- Bed
- Diagnosis
- Procedures
- Results
- Medications

For my study, I use the database with no Protected Health Information such as :

- patient names
- relative names
- all dates
- all phone numbers
- full and partial Social Security Numbers

Indeed, I did not need these informations and using PHI_data require special access and training. For my case of use the only problem was about the date indeed I needed to compute delta of dates and dates are shifted in no-phi data. However, the shift is always the same for each patient so it is not a concern for delta of dates.

After investigating this database, I have build the following database's schema of all the data I might need to validate my results :

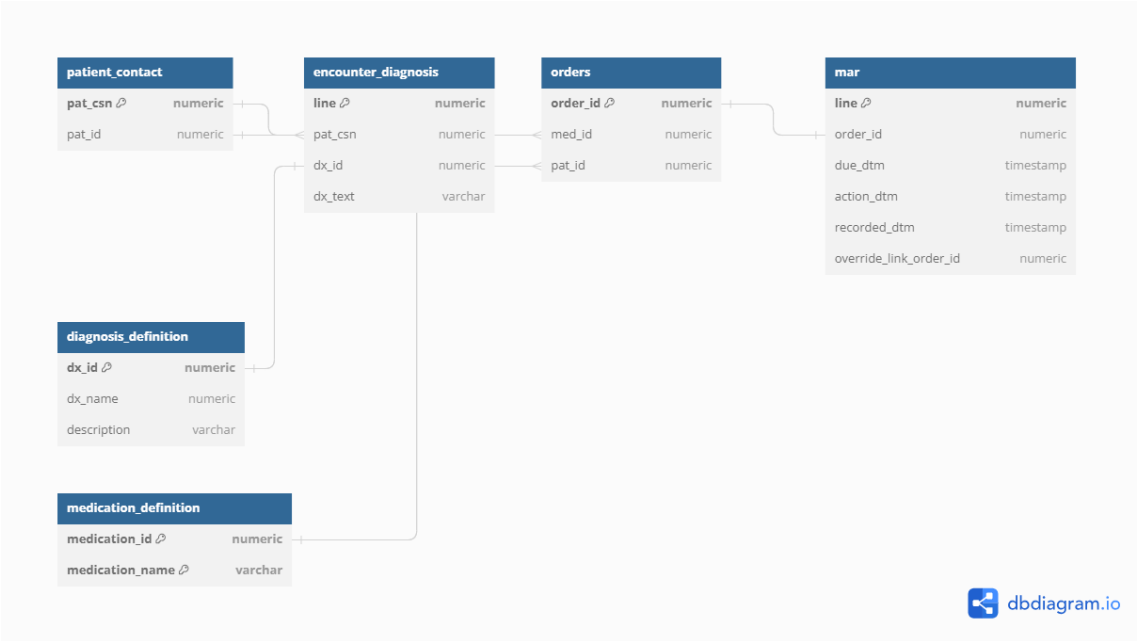

Figure 3.1: cs_analyse_no_phi schema

The following is a brief description of the tables I used :

- patient_contact : Patient contacts which include hospital encounters, clinic appointments, office visits, phone calls, etc. The pat_csn is a single id for patient.
- encounter_diagnosis : The doctor-entered diagnoses for each patient encounter.
- diagnosis_definition : Diagnosis name. We will use it to identify the adverse event in the diagnosis.
- mar : Medication Administration Record. A record for each med that was given to a patient. It even includes missed meds. We will use it to get the time a medication was taken.
- orders : All the orders for patients. We will use it for lab tests and medications
- medication_definition : Medication name
- order_results : the results of each orders. We will use it to get the results of lab tests.
- lab_component_definition : we will use it to identify the lab tests.

3.2 The retrospective cohort study

The general approach use to validate results of drug safety hypothesis is the retrospective cohort study[11][12][13].

A retrospective cohort study looks back in time to analyze the medical records of a group of people who share a common exposure or condition. The goal is to uncover links between that exposure or condition and subsequent health outcomes. Researchers follow the health histories of the cohort over a period of time, comparing the frequency of outcomes in the exposed group to the frequency in an unexposed group. This allows them to estimate the risk associated with the exposure and draw conclusions about potential cause-and-effect relationships. The key advantage of a retrospective design is the ability to efficiently examine exposures and outcomes that have already occurred.

In our case of use, the exposure is the ordering of the candidate pair of drugs and the outcome is the adverse event associated. Also we need to perform the study for each adverse event and each drug-drug interaction candidate.

Presented below are the few steps to conduct a retrospective cohort studies :

- Build at least one exposure group and one control group (for instance : if we want to prove that taking Ibuprofen increase the risk of cardiac arrest, we need a group of patient that took ibuprofen (exposure group) and a group of patient that did not take it in this period of time)
- Find the measurement for the outcome of interest (for instance : if the outcome is the cardiac arrest, we need to find a way to measure a cardiac arrest, to identify it for a given patient)
- Select a statistical method to measure the pair of drugs-outcome association

3.3 Outcome measurement

As I previously spoke about, we need now to find a way to measure the outcome. It means that we have to find relevant measurements that reveal proof of a certain adverse event.

Since the beginning of the project, the study is at a high level and does not get specific to one or few adverse events. In order to pursue in that way, we try to find measurements for a large amount of adverse events.

To do so, I used GPT4 which is a free AI tool available in Microsoft Bing. The idea was to find a prompting method in order to get, for each adverse event, the lab tests that are associated with the detection of this pathology. The used of AI for this part was a huge help in order to perform a massive search in the web. The AI was also able to provide me the LOINC codes(an international standard for health measurement) for these lab tests. The LOINC codes will be the identifier we will use in the clinical database.

Of course, I perform a double check on the informations provides by AI in order to have complete and accurate results.

I was able to look at these for the 20 most relevant adverse events in my results. I defined relevancy on the highest AUROC scores of the model trained. In order to have better results we choose to process first the adverse events that have an associated model with the best performances.

The following figure is a sample of the results :

AE	Lab tests	LOINC codes
Febrile neutrope	Absolute Neutro	751-8
	White Blood Cel	6690-2
Sepsis	White Blood Cel	6690-2
	Lactate	32693-4(Blood)
	C-reactive protei	71426-1(Blood)/
	Prothrombin time	5902-2 / 34529-
Hip fracture	DEXA scan	83311-1
Staphylococcal i	Wound culture te	6462-6

Figure 3.2: Lab tests and LOINC code extraction results

The next step was then to find the number of lab results for each LOINC codes we found in the previous step. Indeed we need an large enough amount of results to build the cohort groups for the studies.

Base on the database schema I described previously, I was able to access this number by a simple SQL request which is :

```
1  select count(*) from cs_analyse_no_phi.order_result or2
2  where or2.lab_component_id in
3  (
4  select lab_component_id from cs_analyse_no_phi.lab_component_definition
   lcd2
5  where loinc_code = '6690-2'
6  )
```

After processing this request for every LOINC codes we extracted from previous step we obtained the following tab :

AE	Lab tests	LOINC codes	Number
Febrile neutrope	Absolute Neutro	751-8	
	White Blood Cel	6690-2	5 711 229
Sepsis	White Blood Cel	6690-2	5 711 229
	Lactate	32693-4(Blood)	0/382 507
	C-reactive protei	71426-1(Blood)/	0/124 394
	Prothrombin time	5902-2 / 34529-8	989 658/0
Hip fracture	DEXA scan	83311-1	No LOINC
Staphylococcal i	Wound culture te	6462-6	0

Figure 3.3: Number of results for each lab tests

As we can see in the tab, for some LOINC codes we have zero results which may be due to the specificity of the lab test. Nevertheless, some lab tests do not have a LOINC code associated in the database which cause a problem of accuracy because we just have string describing this lab test. We can see this case with DEXA scan which is a very common test. I was able to find out that DEXA scan exist in the table $lab\_component\_definition$ but there were no LOINC code associated. It is a problem for the generalization of the process because when there is no LOINC code in the database we need to manually find out the test in the $lab\_component\_definition$ table which is time consuming.

3.4 Patient's groups

As I mentionned earlier, in order to corroborate or refute a drug-drug interaction for a specific adverse event, we need to conduct one retrospective cohort study.

Given one drug-drug interaction candidate for one adverse event, the exposure of the associated cohort study will be the pair of drugs and the outcome will be the adverse event.

The specificity of our cohort study is that the patients in both the exposure and the control group will be the same. However, the values of the lab results will be different. I have build the following schema describing the validation experience :

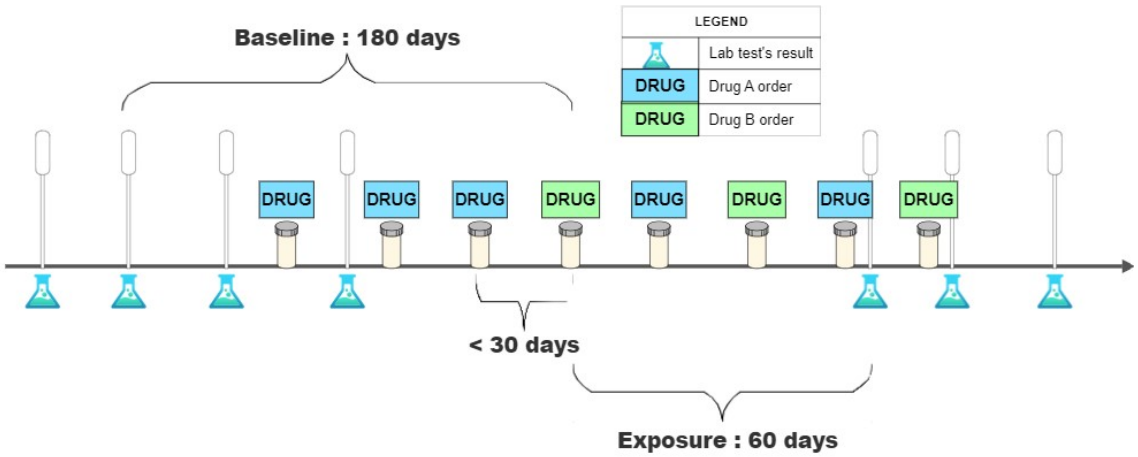

Figure 3.4: Schema of the cohort study experience

Here is a description of the experience for a given drug pair (drug A, drug B) and a given lab test (find in the way we described earlier) :

- We first identify in the database all patients who had orders for both drug A and drug B within a 30 day period. This allows us to find the subset of patients who were prescribed both medications together within a 30 days timeframe. We choose the first instance when both drugs were orders in a 30 days timeframe.
- We define two time period :
  - The baseline : 180 days period before the order of drug B. It is the control period when the patient is not under the putative effect of the drug-drug interaction.
  - The exposure : 60 days period after the order of drug B. It is the period when the patient is under the putative effect of the drug-drug interaction
- Among the patients identify in the first step, we find those who have at least one lab result in the baseline and at least one in the exposure
- We finally process to the statistical analysis of the lab result in order to find whether or not there is an association between the drug A and drug B interaction and the results of the lab test

3.5 Statistical analysis

In this part, I will describe the general statistical analysis of lab results.

The objective of the analysis is to compare the lab results during the baseline and the lab results during the exposure.

The first measurement we will compute is the ratio between the average lab results value of all the patients during baseline and average lab results value of all the patients during exposure.

Given N the number of patients find in the previous part, B_p the number of lab results for patient p during baseline, E_p the number of lab results for patient p

during exposure and v_{ip} the result value for lab test i and patient p, the ratio R is given by :

$$R = \frac{\frac{1}{N}\sum_{j=1}^{N}\frac{1}{B_p}\sum_{i=1}^{B_p} v_{ip}}{\frac{1}{N}\sum_{j=1}^{N}\frac{1}{E_p}\sum_{i=1}^{E_p} v_{ip}}$$

This ratio is a good indicator of the direction of the influence of both drugs on the lab results. Indeed the ratio indicates whether the order of both drugs tends to increase ($R < 1$) or decrease ($R > 1$) the lab result.

Moreover, we will process to a statistical test which is a classic approach in drug safety study but some challenges appear when it comes to this stage[14].

We are trying to compare the average of two different samples : one sample is made of the values of the lab tests during the baseline for each patient, the other sample is made of the values of the lab tests during the exposure for each patient. However, these samples are related because it is the same patients in both sample.

To compare both samples, we choose the paired t-test[15].

The Paired t-Test is used to compare the means between two related groups on a continuous, normally distributed outcome variable. This statistical test can only be used when there are two matched or paired groups with a quantitative measurement taken from each unit in both groups. The t-Test assesses whether the average difference between the paired observations is significantly different from zero.

This test is adapted to our case of study because we are trying to compare the results of a lab test for the same patient. We compare for one patient the average value of all lab results during the baseline for this patient and all lab results during the exposure for this patient.

To simplify the following description of the test, we consider that there is only one lab result for each patient in each period (one result during baseline and one result during exposure). This is not a loss of generality because indeed we can consider that this unique value is the average value of all the result values.

Here is the description of the statistical test :

3.5.0.1 Hypotheses

With μ_B the average lab result of the baseline sample :

$$\mu_B = \frac{1}{N}\sum_{j=1}^{N} v_j^B$$

where v_j^B is the value of the lab result for patient j during baseline.

With μ_E the average lab result of the exposure sample :

$$\mu_E = \frac{1}{N}\sum_{j=1}^{N} v_j^E$$

where v_j^E is the value of the lab result for patient j during exposure.

We have, for the paired t-test, the following hypotheses :

$$(H_0) : \mu_B = \mu_E \tag{3.1}$$
$$(H_1) : \mu_B \neq \mu_E \tag{3.2}$$

In order to simplify the notations, we note :

$$\mu_{diff} = \mu_B - \mu_E$$

Under this notation, the test hypotheses become :

$$(H_0) : \mu_{diff} = 0 \tag{3.3}$$
$$(H_1) : \mu_{diff} \neq 0 \tag{3.4}$$

3.5.0.2 Test statistic

Given n the sample size, $\bar{x}_{diff}$ the sample mean differences (difference between baseline sample and exposure sample), s_{diff} the sample standard deviation of the differences, the test statistic is a Student's statistic :

$$t = \frac{\bar{x}_{diff} - 0}{\frac{s_{diff}}{\sqrt{n}}}$$

The assumptions Underlying a t-test are :

- Normal Distribution : The data within each group (sample) follows a normal distribution with a mean μ (population mean) and a common variance σ^2.
- Chi-Square Distribution (for $s^2(n-1)/\sigma^2$): The ratio of the sample variance s^2_{diff} multiplied by $(n-1)$ to the population variance σ^2 follows a chi-square distribution with $(n-1)$ degrees of freedom.

3.5.0.3 Result of the test

The next step is to compute the p-value of the test. If the p-value is small enough (we choose as treshold p<0.05), we reject the null hypothesis and so there is statistical significance of the difference between the mean of the two samples. It means that statistically, we have proof that the pair of drugs, when the drugs are taken simultaneously, has changed the result of the lab test.

4 Results

4.1 Lab test and adverse event

In the previous parts, I described a general approach to get drug-drug interaction candidates for any kind of adverse event and then a general method to corroborate or refute each candidate regardless of the adverse event study.

However, in order to maximize my odds to have results and to succeed in the process of validation, I choose to first investigate the adverse events related to the lab tests with the largest number of results in the clinical database.

That is why I focused on the lab test : "White Blood Cell count" and so to the adverse event : "Febrile neutropenia".

Febrile neutropenia is a pathology that append when there is a case of fever during a period of neutropenia which is a disease caracterized by a low number of neutrophils in the blood [16].

White blood cells count (LOINC 6690-2) is a relevant lab test for the measurement of this pathology because neutrophils are a type of white blood cells. A decrease of the number of neutrophils in the blood may revealed a case of neutropenia.[17]

4.2 Data engineering

The realisation of the validation process required a data engineering process. Indeed, the result of the lab tests in the database are not always numerical values and are sometimes not relevant.

Moreover, I needed to build a SQL script in order to get the data I needed. I used DBeaver as main data management tool in addition to Gsheet and Python.

Given one pair of drugs candidate the general process was the following :

- Identify the number of patients that have taken both drugs in the 30 days period and have at least one white blood cells count during the baseline and one during the exposure
- If the previous number is high enough, get for each of these patients every lab test result during the baseline and during the exposure and compute the average for both period. All of these average values form the two test samples.
- Perform the t-test using the two samples previously build and compute the p-value
- If $p < 0.05$, the result is corroborated.

Because the query of the database process was very time consuming I choose to extract from the database every numerical lab results during the baseline and during the exposure for each patient. The SQL query I used for instance for the case of the drug pair (sodium bicarbonate, papaverine) is :

```
SELECT DISTINCT o1.pat_id, r1.result_value, r2.result_value,o1.order_dtm

FROM cs_analyse_no_phi.orders o1
JOIN cs_analyse_no_phi.orders o2 ON o1.pat_id = o2.pat_id
JOIN cs_analyse_no_phi.medication_definition m1 ON o1.med_id = m1.
    medication_id
JOIN cs_analyse_no_phi.medication_definition m2 ON o2.med_id = m2.
    medication_id
JOIN cs_analyse_no_phi.order_result r1 ON o1.pat_id = r1.pat_id

JOIN cs_analyse_no_phi.order_result r2 ON o2.pat_id = r2.pat_id

JOIN cs_analyse_no_phi.lab_component_definition l1 ON r1.lab_component_id =
     l1.lab_component_id
JOIN cs_analyse_no_phi.lab_component_definition l2 ON r2.lab_component_id =
     l2.lab_component_id
WHERE lower(m1.medication_name) like '%papaverine%'
AND lower(m2.medication_name) like '%sodium bicarbonate%'
AND r1.result_value ~ '^[0-9]+(.)([0-9]+)$'
AND r2.result_value ~ '^[0-9]+(.)([0-9]+)$'
AND DATEDIFF(DAY, CAST(o2.order_dtm AS DATE), CAST(o1.order_dtm AS DATE))
    <= 30
AND DATEDIFF(SECOND, CAST(o2.order_dtm AS TIME), CAST(o1.order_dtm AS TIME)
    ) <= 30
AND l1.loinc_code = '6690-2'
AND l2.loinc_code = '6690-2'
AND CAST(r1.result_dtm AS DATE) BETWEEN DATEADD(DAY, -180, CAST(o2.
    order_dtm AS DATE)) AND CAST(o2.order_dtm AS DATE)
AND CAST(r2.result_dtm AS DATE) BETWEEN CAST(o2.order_dtm AS DATE) AND
    DATEADD(DAY, 60, CAST(o2.order_dtm AS DATE))
```

After mining these data, I used Gsheet to compute the average result during exposure and baseline for each patient. I then used Python to perform the paired t-test and other computations on data.

4.3 Results

I was able to perform the clinical validation for the first drug-drug interaction candidates for febrile neutropenia and the white blood cells count as lab test. A sample of the results is presented in the following tab(Figure 4.1). For each candidate pair of drugs, we have the number of patients in each sample, the ratio I detailled earlier and the p-value of the paired t-test. In the case of neutropenia the ratio is a good indicator because if the ratio is superior to 1, it means that the number of white blood cells has decreased after taking both drugs and if the ratio is inferior to 1, it means that the number of white blood cells has increased. For neutropenia, we focused on ratio superior to one because neutropenia is characterized by a decrease of a certain type of white blood cell.

We can see that some drug-drug interactions emerged with p-value inferior to 0.05.

I double checked on drug labels if they were already known as responsible for decreasing the number of white blood cells and I found no evidence.

To conclude, we found that papaverine and levofloxacin is likely be a novel drug-drug interaction responsible dor a decrease of the number of white blood cells and so neutropenia.

Drug A	Drug B	n(Size sample fo	Ratio	p-value
Furosemide	digoxin anti	227	0,99	0,71
Papaverine	Sodium Bicarbor	337	0,98	0,23
Simvastatin	Mannitol	239	0,91	9,05E-06
Cefazolin	Mannitol	434	0,99	0,64
Cephalexin	Omega 3	762	1,02	0,09
Papaverine	Levofloxacin	203	1,18	1,90E-07
Labetalol	Cefuroxim	522	1,05	0,024
Zaleplon	Insuline	77	1,07	0,88
Sitagliptine	cefuroxime	103	1	0,87
methylprednisolc	esmolol	372	1.12	3,60E-06

Figure 4.1: Extract of the results for neutropenia and white blood cells count

4.4 Discussion

Even if the previous results are encouraging and interesting, there are few points to discuss.

The first thing is about the normal distribution hypothesis during the test. We make this hypothesis regarding our samples and it is usually verified for large samples. On the following graph that represents the distribution for one sample we were able to verify this hypotheses :

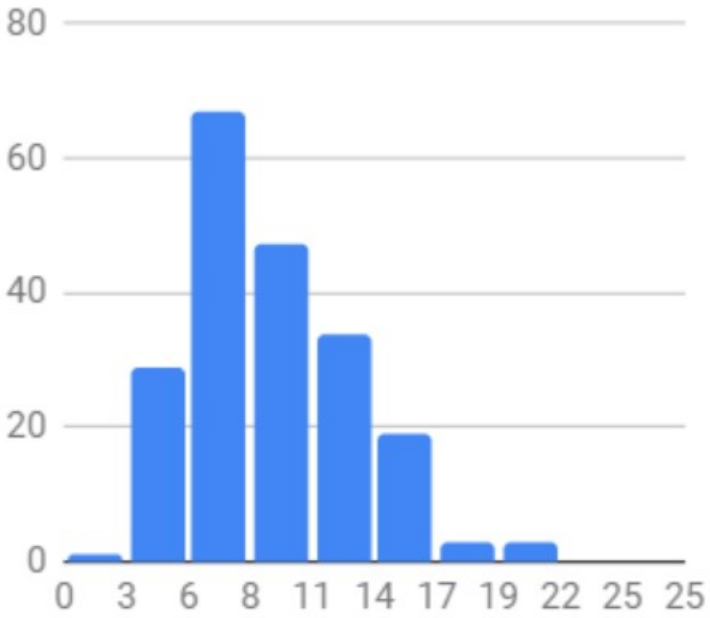

Figure 4.2: One distribution of lab results during baseline

We can see on this figure that the normal distribution hypotheses seems to be respected even if it is one example.

The second thing is about the cohort study. Indeed, in this study we only take into account the prescription of drugs as parameter for the samples. However other factors as the sex, the age or comorbidity may influence the result of the test. Taking into account these factors may have improve the accuracy of our study.

The last point is that some correlated result may be due to only one both drugs. Indeed, when a drug pair was labelled as positive by the model I used an additional algorithm to delete the pair of drugs with one drug already known as responsible for the adverse event (using OnSIDES). However, I was not able to delete the pair of drugs with one drug known as responsible for changing the result of the associated lab test. For instance, in our results, Simvastalin and Mannitol may be a drug-drug interaction responsible for increasing the number of white blood cells. However, I manually double check and Mannitol was already known as responsible for an increase of white blood cells.

Conclusion

Discovering novel drug-drug interactions has remained a major challenge in pharmacovigilance and medical research, especially as multiple medications rises.

Recent machine learning developments for large-scale drug interaction signal detection hold promise for elucidating hidden interactions. I was able to extend a previous method by implementing it on new data sources from the TLab, a leading medical informatics group. The enhanced machine learning pipeline demonstrated strong performance for extracting candidate drug-drug interactions.

Moreover, access to Cedars-Sinai's extensive clinical databases that are database for one of the top hospital in the United States enabled validation of the predicted interactions. The analysis provided statistical confirmation of several novel drug-drug interactions, with implications for improving medication safety. While promising, the study had limitations including sample size and demographic factors. Future work should incorporate additional parameters like age, sex, and ethnicity across larger adverse event and drug cohorts.

Overall, this project advances the automated discovery of drug-drug interactions through integration of robust machine learning with real-world clinical data. The validated novel findings represent an important contribution towards preventing adverse polypharmacy outcomes.

Personal review

This research internship has been a truly rewarding journey on both professional and personal fronts. I was fortunate to explore advanced bioinformatics research within the Department of Computational Biomedicine at Cedars-Sinai Medical Center, renowned as one of the top hospitals in the US. Moreover, I had the valuable chance to work closely with my mentor, Mr. Nicholas Tatonetti, and the other members of his TLab. My mentor has played a pivotal role in significant advancements in his field and generously shared his expertise with me, even on topics I wasn't initially familiar with.

I am highly satisfied with my internship experience with Mr. Nicholas Tatonetti. He allowed me to work independently, enabling me to make discoveries on my own, which I believe is crucial in the field of research. Additionally, he provided valuable guidance and aided my progress, particularly through our weekly meetings.

I'm also grateful to my internship mentor for including me in the bi-weekly project management meetings of the Cedars-Sinai lab. This gave me a deeper insight into how a research team operates and how scientific publications are developed. The weekly meetings with teams from both New York and Los Angeles teams were equally great, allowing me to gain a better understanding of everyone's roles within the laboratory.

From a personal perspective, this internship has enabled me to gain a deeper understanding of the various challenges and facets inherent in the role of a data scientist, as well as the complete process involved in executing a data science project (including data analysis, data engineering, and more).

Glossary

Signification	Name
AE	Adverse Event
DDI	Drug-Drug Interaction
AI	Artificial Intelligence
OnSIDES	ON label SIDE effectS resource
PHI	Protected Health Information
FDA	U.S. Food and Drug Administration
AERS	Adverse Event Reporting System
SAE	Severe Adverse Event
LOINC	Logical Observation Identifiers Names & Codes

References

[1] Pirmohamed M . Ml O : *Drug Interactions of Clinical Importance.*
London: Chapman & Hall, 1998.

[2] Theodore J Gaeta, Melissa Fiorini, Kimberly Ender, Joseph Bove, Jose Diaz : *Potential drug-drug interactions in elderly patients presenting with syncope.*
The Journal of Emergency Medicine, 2002.

[3] Van der Heijden PG Van Puijenbroek EP Van Buuren S et al. : *On the assessment of adverse drug reactions from spontaneous reporting systems: the influence of under-reporting on odds ratios.*
Stat Med2002;21:2027–44.

[4] Nicholas P Tatonetti, Guy Haskin Fernald, Russ B Altman : *A novel signal detection algorithm for identifying hidden drug-drug interactions in adverse event reports*
Journal of the American Medical Informatics Association, Volume 19, Issue 1, January 2012, Pages 79–85

[5] Yu Gu and Robert Tinn and Hao Cheng and Michael Lucas and Naoto Usuyama and Xiaodong Liu and Tristan Naumann and Jianfeng Gao and Hoifung Poon : *Domain-Specific Language Model Pretraining for Biomedical Natural Language Processing*
2020

[6] Dina Demner-Fushman, Sonya E Shooshan, Laritza Rodriguez, Alan R Aronson, Francois Lang, Willie Rogers, Kirk Roberts, Joseph Tonning : *A dataset of 200 structured product labels annotated for adverse drug reactions*

[7] Roberts, Demner-Fushman, Tonning : *Overview of the TAC 2017*

[8] Stuart J Nelson, Kelly Zeng, John Kilbourne, Tammy Powell, Robin Moore : *Normalized names for clinical drugs: RxNorm at 6 years*
2011

[9] Oliver Bodenreider , Ronald Cornet , Daniel J. Vreeman : *Recent Developments in Clinical Terminologies — SNOMED CT, LOINC, and RxNorm*
Yearb Med Inform 2018

[10] Xue Ying : *An Overview of Overfitting and its Solutions*
2019

[11] Lorberbaum T, Sampson KJ, Chang JB, Iyer V, Woosley RL, Kass RS, Tatonetti NP. : *Coupling Data Mining and Laboratory Experiments to Discover Drug Interactions Causing QT Prolongation*
2016

[12] John-Michael Gamble: *An Introduction to the Fundamentals of Cohort and Case–Control Studies*
2014

[13] Cristiano Moura, Nília Prado, Francisco Acurcio: *Potential drug-drug interactions associated with prolonged stays in the intensive care unit: a retrospective cohort studys*
2011

[14] Amy Xia and Qi Jiang: *Statistical Evaluation of Drug Safety Data*
2013

[15] Wikipedia : *Student's t-test*

[16] Sheena Punnapuzha; Paul K. Edemobi; Amr Elmoheen.: *Febrile Neutropenia*
2023

[17] MedlinePlus: *Low white blood cells count and cancer*

Appendix

A Appendix 1

Table A.1: Table of all the results of all the models trained

AE	Recall	Auroc	Prediction
General physical health deterioration	0.93	0.96	247.00
Febrile neutropenia	0.89	0.95	536.00
Tardive dyskinesia	0.85	0.94	0.00
Septic shock	0.85	0.92	691.00
Dyskinesia	0.83	0.91	415.00
Hip fracture	0.30	0.88	772.00
Drug abuse	0.68	0.88	1045.00
Staphylococcal infection	0.70	0.88	284.00
Type 2 diabetes mellitus	0.40	0.87	222.00
Drug dependence	0.87	0.87	430.00
Dysarthria	0.75	0.87	392.00
Aggression	0.74	0.87	414.00
Psychotic disorder	0.67	0.86	457.00
Infusion related reaction	0.83	0.86	653.00
Restlessness	0.76	0.86	556.00
Suicide attempt	0.74	0.86	959.00
Agitation	0.75	0.85	488.00
Interstitial lung disease	0.71	0.85	808.00
Breast cancer	0.69	0.84	608.00
Movement disorder	0.60	0.84	424.00
Hallucination	0.72	0.84	522.00
Delirium	0.73	0.83	1128.00
Hypoxia	0.68	0.83	781.00
Depressed level of consciousness	0.78	0.83	458.00
Suicidal ideation	0.72	0.83	510.00
Oxygen saturation decreased	0.60	0.83	170.00
Osteonecrosis	0.58	0.82	762.00
Neutropenia	0.72	0.82	485.00
Urinary retention	0.68	0.82	495.00
Pulmonary embolism	0.68	0.81	976.00
Ascites	0.72	0.81	803.00
Completed suicide	0.80	0.81	1724.00
Jaundice	0.79	0.81	595.00
Fall	0.69	0.81	463.00
Overdose	0.40	0.81	420.00
Amnesia	0.69	0.81	482.00
Hyponatraemia	0.76	0.81	1418.00
Sepsis	0.68	0.81	990.00

Pancreatitis	0.77	0.80	1145.00
Seizure	0.74	0.80	473.00
Leukopenia	0.74	0.80	636.00
Alanine aminotransferase increased	0.62	0.80	297.00
Dehydration	0.69	0.80	482.00
Road traffic accident	0.70	0.80	0.00
Liver disorder	0.6	0.80	265.00
Neuropathy peripheral	0.70	0.80	561.00
Hypokalaemia	0.64	0.80	1392.00
Respiratory arrest	0.59	0.80	1151.00
Deep vein thrombosis	0.66	0.80	747.00
Bradycardia	0.69	0.79	505.00
Thrombocytopenia	0.76	0.79	605.00
Lung disorder	0.62	0.79	556.00
Aspartate aminotransferase increased	0.61	0.79	313.00
Nephrolithiasis	0.65	0.79	0.00
Platelet count decreased	0.61	0.79	389.00
Infection	0.72	0.79	605.00
Speech disorder	0.66	0.79	548.00
Epistaxis	0.72	0.79	549.00
Cardiac arrest	0.65	0.78	1115.00
Atrial fibrillation	0.67	0.78	574.00
Pneumonia	0.67	0.78	512.00
Haematoma	0.55	0.78	277.00
Tachycardia	0.73	0.78	639.00
Anaemia	0.74	0.78	602.00
Head injury	0.40	0.78	298.00
Neutrophil count decreased	0.46	0.78	0.00
Hypotension	0.74	0.78	646.00
Cataract	0.68	0.78	546.00
Pyrexia	0.75	0.78	539.00
Confusional state	0.70	0.78	679.00
Injury	0.75	0.78	601.00
Dysphagia	0.67	0.78	593.00
Fluid retention	0.59	0.78	490.00
Vomiting	0.79	0.78	628.00
Renal failure	0.64	0.78	1076.00
Cerebrovascular accident	0.65	0.78	1091.00
Respiratory failure	0.64	0.78	774.00
Transient ischaemic attack	0.59	0.77	1265.00
Thrombosis	0.59	0.77	527.00
Diabetes mellitus	0.69	0.77	607.00
Blood creatinine increased	0.60	0.77	0.00
Haemorrhage	0.65	0.77	479.00
Aphasia	0.58	0.77	397.00
Hypertension	0.79	0.77	602.00
Vertigo	0.79	0.77	674.00

Dyspnoea	0.74	0.77	598.00
Depression	0.74	0.77	574.00
Oedema peripheral	0.72	0.77	616.00
Asthenia	0.79	0.77	620.00
Anxiety	0.71	0.77	595.00
Pancytopenia	0.68	0.77	1395.00
Hepatic failure	0.67	0.77	1214.00
Gastrointestinal haemorrhage	0.67	0.76	1481.00
Pulmonary oedema	0.59	0.76	1390.00
Acute kidney injury	0.59	0.76	1025.00
Anaphylactic reaction	0.75	0.76	544.00
Haematemesis	0.60	0.76	398.00
Angioedema	0.74	0.76	545.00
Colitis	0.67	0.76	820.00
Cellulitis	0.61	0.76	511.00
Syncope	0.71	0.76	678.00
Blood creatine phosphokinase increased	0.51	0.76	440.00
Osteoarthritis	0.61	0.76	817.00
Chronic obstructive pulmonary disease	0.00	0.76	0.00
Blood alkaline phosphatase increased	0.55	0.76	0.00
Blood pressure decreased	0.40	0.76	488.00
Asthma	0.65	0.76	489.00
Epilepsy	0.45	0.76	162.00
Cerebral haemorrhage	0.56	0.76	752.00
Coronary artery disease	0.60	0.76	950.00
Rhabdomyolysis	0.58	0.75	627.00
Hyperkalaemia	0.61	0.75	715.00
Renal impairment	0.66	0.75	558.00
Arrhythmia	0.70	0.75	679.00
Chest pain	0.72	0.75	690.00
Abdominal pain	0.76	0.75	579.00
Myocardial infarction	0.74	0.75	1464.00
Cardiac failure congestive	0.68	0.74	1566.00
Pleural effusion	0.59	0.74	565.00
Cholelithiasis	0.72	0.74	724.00
Haematuria	0.65	0.74	457.00
Blood bilirubin increased	0.65	0.74	447.00
Hypoglycaemia	0.59	0.74	422.00
Oedema	0.65	0.74	518.00
Lethargy	0.65	0.74	754.00
Cognitive disorder	0.46	0.74	573.00
Cardiac failure	0.62	0.73	1296.00
Respiratory distress	0.56	0.73	1141.00
Coma	0.60	0.73	1729.00
Rectal haemorrhage	0.63	0.73	0.00
Haemoglobin decreased	0.64	0.73	507.00
Disorientation	0.60	0.73	848.00

Urinary tract infection	0.73	0.73	682.00
Blood pressure increased	0.65	0.73	661.00
Death	0.61	0.72	1126.00
Angina pectoris	0.71	0.72	929.00
Hyperglycaemia	0.70	0.72	584.00
Chronic kidney disease	0.40	0.72	479.00
Abortion spontaneous	0.33	0.71	139.00
Intestinal obstruction	0.55	0.71	1460.00
Plasma cell myeloma	0.47	0.70	305.00
Hepatic function abnormal	0.51	0.70	404.00
Lower respiratory tract infection	0.58	0.70	649.00
White blood cell count increased	0.20	0.70	101.00
Mental disorder	0.20	0.70	628.00
Loss of consciousness	0.48	0.68	286.00
Liver function test abnormal	0.54	0.68	660.00
Visual acuity reduced	0.37	0.68	201.00
Wheezing	0.52	0.68	477.00
Arthropathy	0.49	0.67	856.00
Metabolic acidosis	0.39	0.67	838.00
Neoplasm malignant	0.53	0.66	794.00
C-reactive protein increased	0.00	0.66	680.00
Cardiovascular disorder	0.42	0.66	494.00
White blood cell count decreased	0.38	0.66	334.00
Cardiac disorder	0.40	0.64	0.00
Blindness	0.45	0.63	434.00
Haematochezia	0.38	0.63	515.00
Cerebral infarction	0.37	0.63	1316.00
Mental status changes	0.30	0.62	213.00
Cardio-respiratory arrest	0.44	0.62	1415.00
Inflammation	0.67	0.59	364.00
Acute myocardial infarction	0.17	0.57	790.00
Abnormal behaviour	0.23	0.57	303.00
Toxicity to various agents	0.10	0.55	913.00
International normalised ratio increased	0.00	0.28	114.00

B Appendix 2

GitHub repository : https://github.com/Mittenover/PRe-Cedars-Sinai.git