FDA Risk of Side Effects from Medical Drugs

IST718 Final Project

- Assessed the OPEN FDA Adverse Drug Reactions data set for risk predictions
- Random Forest Model shows 93% accuracy at prediction of risk classes and optimal performance with predicting High Risk irreversible scenarios
    - Recommend implementation of the model to assist medical professionals when prescribing medication
    - Recommend looking into cloud providers who can allow the model to be tuned and scaled to all medications available in the data set

DECEMBER 19, 2021

ASSADOUR DERDERIAN, JOHN LEO BARTLETT, KYLE WALTER

1 Introduction

Healthcare is one of the most noted fields that use massive data to analyze future results that detect the probability of disease and dictate the actual outcome of an emergency. The result may have a significant effect on the success of rehabilitation. Without a doubt, it is a critical factor between life and death - where every second saved quadruples the chance for a patient to live.

One of the significant problems involved in the side effect a drug causes when you take another medicine. Prescribed medication must not provide side effects that correlate with other medications that a patient takes or nullify the actual treatment of that medicine.

The FDA provides the dataset we used. It shows the side effects of various medicines. It is helpful for doctors who seek the best treatment for their patients. The Open FDA mission describes it purpose as "The goal of the project is to create easy access to public data, to create a new level of openness and accountability, to ensure the privacy and security of public FDA data, and ultimately to educate the public and save lives."

As there are literally thousands of medications on the market, the hope is to provide medical staff tools that can help them make informed decisions about the risk of various medicines.

The Data

Data from the FDA's open-source site called OPEN FDA was collected to complete this project. The FDA currently has requirements for Medical Professionals, Drug Manufacturers, and Lawyers to report known drug events by the end of each quarter to the FDA. This has been an evolving process and FDA has made all data collected since 2004 going forward accessible to the public.

After the data files were collected there were some challenges getting it into a form that could be easily worked with for the project. First, during the 2014 data redesign the structure of the data was slightly changed, and earlier years do not easily map field by field to the data collected from 2014 going further along.

Additionally, the volume of requirements for reporting has been growing so in more recent years to where the volume can be quite daunting with which to work.

Open FDA

Like other US government agencies, the FDA is required under the freedom of information act to release data to the public for review and analysis purposes. To meet this requirement the FDA provides access to the data through 2 sources.

1. Via an API
2. Via downloadable quarterly reports

Both sources of data return the same JSON structure, however the API has some significant limitations to large data analysis. First each call of the API returns of JSON of only 1000 reports. There is a daily limit of 26 calls from single IP address. This may work well for smaller samples or reviewing controlled scenarios like newly released medications, the use of the API for collection was vetoed as it wouldn't provide an appropriate amount.

The API also provides a feed that returns the web address where each of the quarterly files are saved. This was used along with Python's wget package to download all the files.

THE JSON

Whether utilizing the API or the downloaded the file the information is presented in the form of JSON file. This was the second major hurtle to overcome in working with the data. The JSON itself had multiples nested levels, as if an entire relational database had been dumped into the respective file.

Through some help with the documentation and trial and error, it was found that three levels would be the most useful for the analysis. First level was the "drugs" level which it was the main case report. It contained information about the patient such as age, weight (in kg), and sex. It also provided information about the provider of the report which could come from various types of medical professionals, lawyers, customers, and drug manufacturers. In addition, there were several Boolean columns indicating how serious the outcome of the case was. Outcomes range from not serious to death.

The next level that was of interest was the reactions. The nested table was quite small, only three columns, but contained the reported reactions that were observed.

The last level of interest in the JSON file was the medicinal products. This reported the medications that each of the patients were taking. This table was quite large with not only the medication but active drug, FDA approval, and many other features from the trials reported to the FDA.

FLATTENING

Once the JSON file layout was understood. The next step became how to flatten the data in a table that could be used for the analysis and subsequent machine learning model. Using Python as the flattening method, a large loop was written. The loop started by reading in JSON File and creating a table of the drug events. The data was filtered by primary source for the reports that were submitted by medical professionals.

Next the columns with the columns representing the additional tables were flattened using the JSON Normalize command. Those columns with the safety report id were sent to dictionaries and flattened into separate data frames and then only medicinal product and reaction were merged back with the main data frame. The JSON columns were dropped to keep the tables from growing too large.

The initial trial was to convert to one large data frame, this failed as the memory resources needed to perform this were too large for the computers available. Instead, each of the cleaned tables were output with 25 columns into separate CSV files. Due to the time that this took, it was decided to utilize the last 5 full years of data from 2016 through the end of 2020. The initial columns can be seen in the table below.

1	V1
2	index
3	safetyreportid
4	serious
5	receiptdate
6	companynumb
7	occurcountry
8	seriousnesshospitalization
9	primarysourcecountry
10	primarysource.

11	patient.patientonsetage
12	patient.patientonsetageunit
13	patient.patientagegroup
14	patient.patientweight
15	patient.patientsex
16	reportduplicate.duplicatesource
17	seriousnessother
18	seriousnesslifethreatening
19	authoritynumb
20	seriousnessdeath
21	seriousnesscongenitalanomali
22	seriousnessdisabling
23	reactionmedrapt
24	medicinalproduct
25	receiver

Consolidation

The results of the previous step were hundreds of CSV files. The next step was needing to convert these into one file for the analysis. For this step a small script was written into R using data.table and R’s native lapply function to combine all of the files.

The resulting data frame had the 25 columns and around 56 million records. The file exported as a CSV from R and the rest of the project was completed using Python.

Preprocessing

Now that a single data frame had been created in a csv file, the data was called back into Python utilizing Dask data frame to deal with the large size (about 7GiB). The data was reviewed and about 9 columns dropped having provided little useful information or left-over indices from the previous flattening steps.

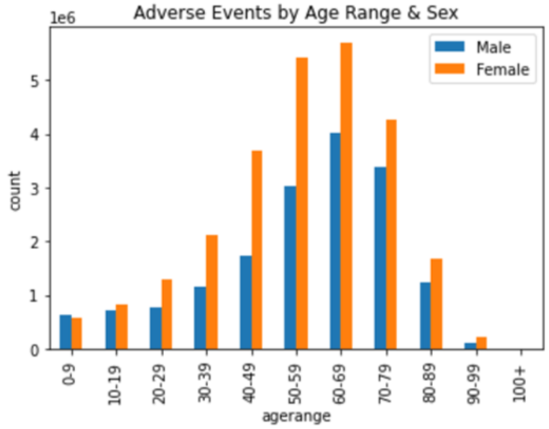
The resulting Data Frame

1	safetyreportid
2	serious
3	receiptdate
4	seriousnesshospitalization
5	patient.patientonsetage
6	patient.patientagegroup
7	patient.patientweight
8	patient.patientsex
9	seriousnessother
10	seriousnesslifethreatening
11	seriousnessdeath
12	seriousnesscongenitalanomali
13	seriousnessdisabling
14	reactionmedrapt
15	medicinalproduct
16	receiver

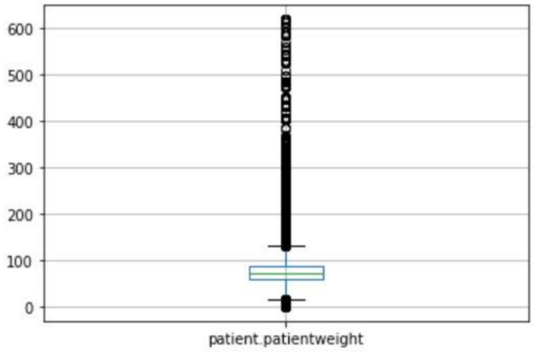
2 EXPLORATORY ANALYSIS

Now that the data was in a form that could be analyzed, several reviews were completed. First while trying to understand the patient profiles, it was found that several records either did not have an age or that age appeared to be fat fingered, and the patient had been living since the Middle Ages. It was later learned that this was a common practice for John and Jane Doe patients where the age is unknown.

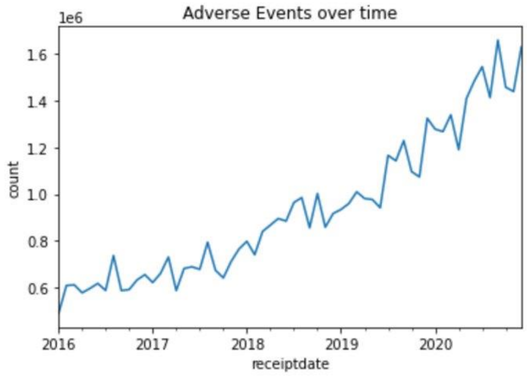
The stratification of ages when reviewing the data appears to be in the normal ranges for life expectancy. Moreover, it follows the normal understanding when paired with biological sex that women are more likely to seek medical care



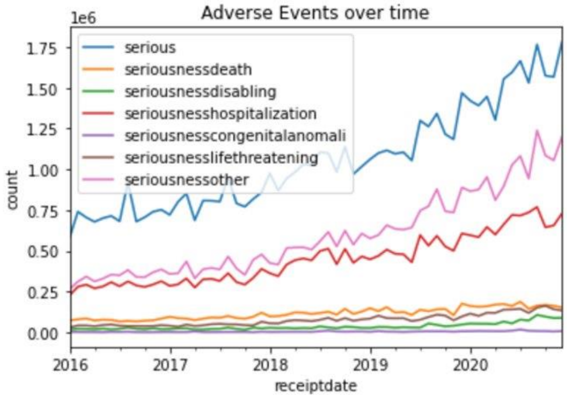
In addition to age, weight was another group where many missing values was found. These were replaced temporarily by -1 but later removed in the models. Most of the input weights fall in a normal range when looking at a box and whisker plot. There were a few extreme outliers such as a patient who is 635kg.



Lastly reviewing drug events over time was another slice of data. While there is sharp increase over the entire time period reviewed, it is worth noting that federal agencies are enforcing compliance of various reporting, not just in health but in other fields such as finance, the expectation is that as more hospitals become more compliant the trend will only continue to rise. In addition, 2020 has seen the onslaught of the coronavirus and both trials of medicine for helping with the symptoms as well as people self-medicating were top of the new charts towards the end of the of the below charts:



What makes reporting more evident as the result for the rise is across the various seriousness categories most serious outcomes are very flat. Hospitalization is up in late 2019 and 2020 likely in line with Coronavirus Pandemic and congenital anomalies as well. While the serious tag reflects overall cases reported.



This should feel good for the most part. As more cases are reported their outcome is showing to be more in favor of small to no effect on the public. However, can this data be used to predict the risk of serious outcome?

3 THE MODELS

The first step in creating the models was to come up with methodology. The seriousness, as evidenced in the prior chart, is spread across columns. Instead, a risk categorization was devised by the outcome of the patient taking the medication. This is important when talking about medium risk. 5 categories were defined:

1. Low Risk – Seriousness overall column indicates the reaction was not serious
2. Medium Low Risk – If the seriousness was labeled seriousness other. It was felt that if the patient had more serious event that required hospitalization, death, or disability that those seriousness tags would be the ones with a "true" value

3. Medium Risk – This represented Congenital Anomaly, more commonly known as birth defects. This is the category with the fewest events. It is also one that the patient does not directly get lasting effects from, and the child born would not know life without the defect.
4. Medium High Risk – Hospitalization or Life-threatening seriousness. While quite serious, these are still within the medical communities' guidelines to correct.
5. High Risk – Death and Disability seriousness. These two where the medical community cannot perform some action to restore the patient's quality of life.

The risk was converted from lowest to highest so that if a higher risk assignment was found the rating would land the highest event. In addition, due to memory limitation, the medical drugs were subset to the 500 most reported in the data set.

Reactions were additionally dropped after much review. Since the goal was to predict the risk level of assigning the medication to a patient, the reaction, if any, would be unknown to the medical professional assigning the medication.

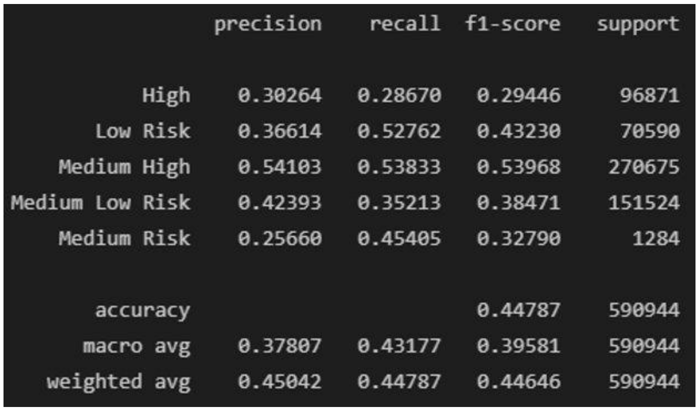
Using pandas Pivot table function the medications were pivoted to columns with a count of if the medication occurred in each case or not as Boolean value assigned. This created a sparse matrix that could be fed to the models for predicting the risk classification. This also had the effect of removing cases that showed up multiple times in records because the patients had been taking more than 1 medication and/or had more than 1 reaction reported by their medication professionals. This reduced the record set to 1.8 million records, a nice reduction from the 56 million for the time period that had be gathered through the JSON conversion.

A model data frame was created using Python's iloc function to gather the risk level, patients' age, weight, sex, and the 500 most prescribed medications. At this stage the data was split in training and test set using 1/3rd for testing and 2/3rds for training model. Lastly before sending the data through the model the Risk Level was assigned to y variable and the rest of the data assigned to an x variable.

Multinomial Naïve Bayes

Multinomial Naïve Bayes is model that is trained using the bayes postulate. It has an advantage of being a fast-converging model due to its underlying independence assumption and generally preforms well with data that has a high number of dimensional features such as text. There are 503 features that the model has for training, so data set is within this area.

While the model converged quickly at just under 2 minutes. Overall, the model performance was not that great. It shows an accuracy score 44.78% which if using the random guess method 1 out of 5 choices, indicates the model has learned something, but it's ability to predict the high-risk category is only at an F1 averaged score 29.4% which is only slightly better than a random guess.

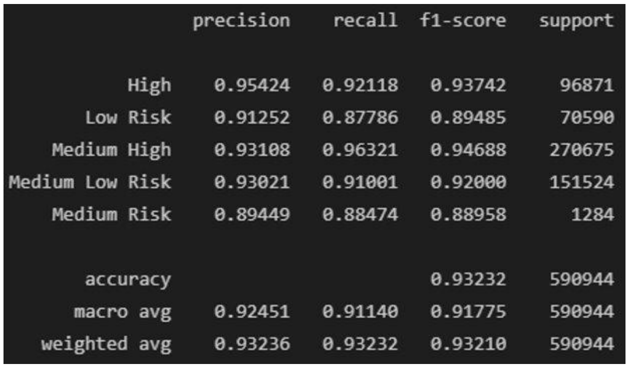


```
                  precision    recall  f1-score   support

            High    0.30264   0.28670   0.29446     96871
        Low Risk    0.36614   0.52762   0.43230     70590
     Medium High    0.54103   0.53833   0.53968    270675
 Medium Low Risk    0.42393   0.35213   0.38471    151524
     Medium Risk    0.25660   0.45405   0.32790      1284

        accuracy                        0.44787    590944
       macro avg    0.37807   0.43177   0.39581    590944
    weighted avg    0.45042   0.44787   0.44646    590944
```

Random Forest

Random Forest is a model that builds several decisions trees, another type of model named for if then tree like structure. Unlike a decision tree, random forest is controlled by the size of the forest (ie the number of trees) and when classifying takes an average of them. This helps the model preform optimally, where it’s base structure as single tree overfits to data sets easily often preform quite poorly when introduced to test data.

The same training and test sets were utilized for this model. The model convergence time was substantially longer at 45 minutes. The results however were significantly better than the Multinomial Naïve Bayes. Overall, the model preforms at 93% accuracy and all categories have F1 scores near or in the 90s as well, meaning a prediction of the model with high risk can be assured to be a good indicator of problems if the certain medicines are prescribed in combination.



```
                  precision    recall  f1-score   support

            High    0.95424   0.92118   0.93742     96871
        Low Risk    0.91252   0.87786   0.89485     70590
     Medium High    0.93108   0.96321   0.94688    270675
 Medium Low Risk    0.93021   0.91001   0.92000    151524
     Medium Risk    0.89449   0.88474   0.88958      1284

        accuracy                        0.93232    590944
       macro avg    0.92451   0.91140   0.91775    590944
    weighted avg    0.93236   0.93232   0.93210    590944
```

4 Conclusion - Recommendations

The random forest model at its current level could be utilized as a risk predicter and help medical professionals when prescribing medications avoid potentially serious situations. While this version of the model is severely limited due to computing power of graduate students, with cloud computing technology it could be expanded to a wider swath of medications, and the model can be tuned each quarter as newly released data is provided by the FDA.

It is recommended that the larger model be utilized as a requirement when prescribing medications and collection of from medical professionals of the model risk prediction in future reports. This could help understand the risk tolerance of medical professionals and keep potentially risky professionals from treating patients.

In addition, more time could potentially allow the review ingredients in the medication, thus reducing the potential feature set and making the model more universally usable. This method could also help in the prediction of potential reactions as the data will be more generalized models could learn from these potential patterns.