Data Science Capstone (DSC383W) Final Report
URMC Project Team
Junchao Shen

1. Introduction/Motivation

For this capstone project, our team is collaborating with polypharmacy team from the Wilmot Cancer Center and the University of Rochester, Medical Center to solve the real-life medical problems clinicians are faced with in the daily practice. Polypharmacy team at URMC is currently running two large prospective national trials visa the National Cancer Institute Community Oncology Research Program. In the past years, they recruited cancer patients from 7 national research bases, 44 community sites and over 900 community-based hospitals into a large comprehensive clinical trial. Up to date, their database has the records of 691 patient's information, including information ranging from basic demographic data to more complicated clinical lab test, insurance plan and etc. Based on their records, the 691 patients recruited in the study are taking over 7426 different medication. On average, each patient is taking 11 medications, which is the core issue Polypharmacy team hope to study. Polypharmacy is characterized by the simultaneous use of multiple medications and is extremely common in older patients.

However, one challenge polypharmacy team is facing during their research and need the help from our team as data scientist is that they need process with tons of medication data every day, and a great amount of those data are messy and inaccurate. Upon recruitment to the study, each matriculant is asked to manually write the list of generic names of medication he/she is taking on a questionnaire, which is then collected and recorded into the computer by a clinical coordinator. However, patients are not clinicians, and it can impose quite a challenge for them to correctly spell and write down the correct medication name. Aside from the spelling issue, most people are more familiar with the trade name of the medication but have a lack of knowledge about the generic name corresponding to the trade name. For instance, Tylenol is controversially the most frequently used medication, but very few people among the aging population can correctly associate it with acetaminophen, which is the generic name of Tylenol.

Therefore, polypharmacy team requested us to make an algorithm that automatically identify the spelling errors in their medication database, correct the wrongly spelled ones and finally convert all medication list into their generic name. If more than one generic drug included in the same medication, for example, Xanax—a commonly prescribed psychiatry drugs for treating general anxiety disorder (GAD), contain both acetaminophen and oxycodone, our algorithm should list both ingredients separately.

With this algorithm, clinicians no longer need to manually check the spelling and form of drugs in the medication list line by line. This can potentially save physicians a lot of time from doing paper work and information checking and enable them to spend more time with their patients for better diagnosis and prognosis.

For the next stage, we met with clinicians and statisticians from URMC team to formulate a clinical hypothesis and finally decided to develop a predictive model to help clinicians identify which patients were at a higher risk of falling based on clinical observations, demographic information, financial and social support they received and etc.

Our motivation for developing such a model is attributed to our talk with first-line oncologists who talked about how dangerous falling could be for aging cancer patients during one routine meetings. Both Dr. Ramsdale and Dr.Mohamed agreed that physicians working with aging patients didn't have very effect ways to tell how likely a patient was likely to fall until the fall had caused severe injuries to the patients already. Previous literature studied several clinical features that potentially linked to falling prediction, but a linear model normally failed to include a very high dimensionality of features in predictive model and therefore, the performance of those models, more often than not, could be quite disappointing. Therefore, we decided to apply a bunch of machine learning algorithm to help solve that problem. A couple of machine learning algorithm, like Support Vector Machine, Neural Network, worked extremely well with high dimensional dataset that linear models normally failed to work well on.

The goal of such a model is to help clinicians with identifying those patients who are under high risk of falling, so that more preventative interventions can be initiated at an early stage to further increase the prognosis of aging cancer patients.

2. Data Set Description/Provenance

a). Medication List Dataset

The medication list we are supposed to clean for the first phase of this project is called GAP dataset. GAP is the abbreviation of geriatric assessment intervention for patients aged 70 and over receiving chemotherapy for advanced cancer: reducing chemotherapy toxicity in older adults. There are two columns in the simplified GAP dataset, patient ID and the medication the patient is taking, as is shown in the following:

GAP_Polypharmacy_Data.Students

Dummy id	drug_name
96	METAPROLOL
96	OXYCODONE ACETAMINOPHEN
96	FINASTERIDE
96	AMLODIPINE
96	SPIRONOLACTONE
96	MEGACE
96	LISINOPRIL
96	GLIMEPIRIDE
96	RANITIDINE
96	NAPROXEN
96	ASPIRIN
96	GABAPENTIN

Figure 1. A brief overview of the GAP dataset

As can be inferred from the Figure 1, different rows can have the same patient id, indicating that the same patient can be taking a variety of different medications. The drug name columns are the target for cleaning, and it includes both generic name and trade name for different drugs.

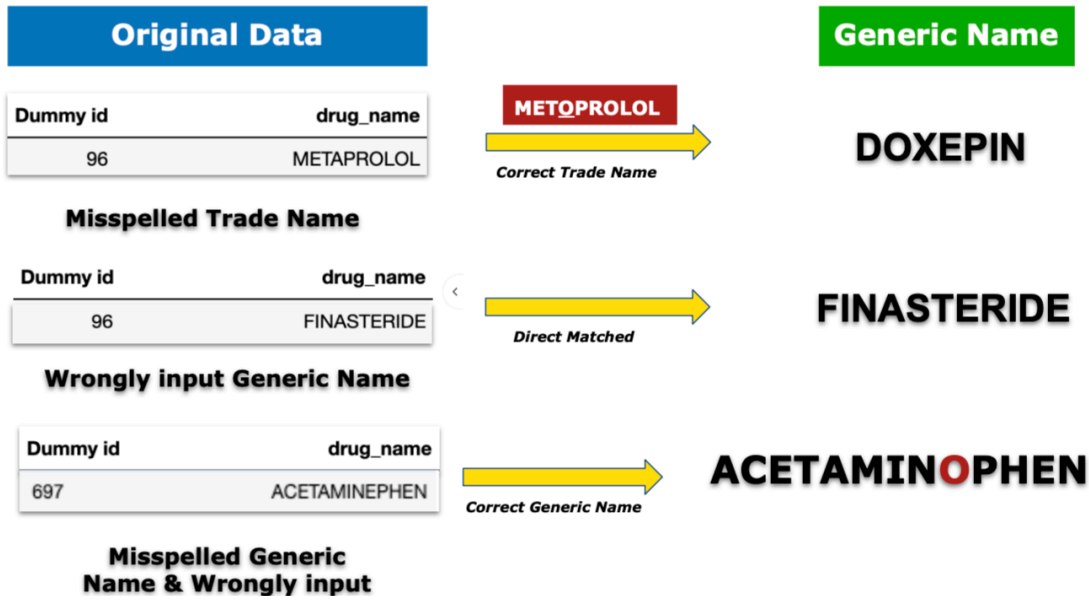

Figure 2. Three types of errors that should be captured and corrected by our algorithm

At the first sight, we identified a couple of problems existed, as is showed in Figure 2. In the first row, for example, metoprolol is known as a trade name instead of the generic name clinicians want. Besides, in the third row, despite the fact that acetaminophen is a generic name, it is misspelled by putting an "e" instead of "o" after the first letter "n". These are the problems we want our algorithm to identify and correct.

Some weird symbols were also found to exists in the original GAP dataset, as is showed in Figure 3. Our goal was to totally remove those irrelevant symbols and ensure that the dataset was clean before running our algorithms on the dataset.

	PPMedName1	generic_N_1
0	VALIUM DIAZEPAM\r\n	DIAZEPAM
1	TYLENOL	ACETAMINOPHEN
5	TYLENOL CODEINE NUMBER 3	ACETAMINOPHEN
19	ALLOPURINAL	ACETAMINOPHEN / OXYCODONE
21	XANAX	ACETAMINOPHEN / OXYCODONE

Figure 3. Messy symbols like \n \r was identified in the original dataset

To check the spelling of medication, a correct version of medication master list was provided by the URMC team for the purpose of cross reference (Figure 4). As is showed in the following master list, there are three columns listed in the dataset that correspond to the class of drugs, trade name of drugs and its corresponding generic name respectively. However, a few trade names to generic name mapping was found to be problematic when we were using this master list as a reference for our auto cleaning algorithm. The wrong mapping was then manually corrected during our weekly meeting.

PHARMACOLOGICAL_CLASS	DRUG_NAME	GENERIC_NAME
ANTIBIOTICS, NITROIMIDAZOLE	FLAGYL ER	METRONIDAZOLE
ANTI-INFLAMMATORIES,TOPICAL OPHTHALMIC	FLAREX	FLUOROMETHOLONE ACETATE
ANTISPASMODICS,URINARY	FLAVOXATE HCL	FLAVOXATE HCL
IMMUNOGLOBULINS	FLEBOGAMMA DIF	IMMUNE GLOBULIN,GAMMA(IGG)
ANTIARRHYTHMICS	FLECAINIDE ACETATE	FLECAINIDE ACETATE
NONSALICYLATE NSAIS,ANTIRHEUMATIC	FLECTOR	DICLOFENAC EPOLAMINE
PROSTAGLANDINS	FLOLAN	EPOPROSTENOL SODIUM (GLYCINE)
ALPHA BLOCKERS/RELATED	FLOMAX	TAMSULOSIN HCL
ANTI-INFLAMMATORIES,NASAL	FLONASE	FLUTICASONE PROPIONATE
ANTI-INFLAMMATORIES,INHALATION	FLOVENT DISKUS	FLUTICASONE PROPIONATE
ANTI-INFLAMMATORIES,INHALATION	FLOVENT HFA	FLUTICASONE PROPIONATE
ANTIFUNGALS	FLUCONAZOLE	FLUCONAZOLE
ANTIARRHYTHMICS	FLUCONAZOLE IN DEXTROSE	FLUCONAZOLE IN DEXTROSE,ISO-OS
ANTIFUNGALS	FLUCONAZOLE IN SALINE	FLUCONAZOLE IN NACL,ISO-OSM
ANTIFUNGALS	FLUCONAZOLE-NACL	FLUCONAZOLE IN NACL,ISO-OSM
ANTIFUNGALS	FLUCYTOSINE	FLUCYTOSINE
ANTINEOPLASTICS,ANTIMETABOLITES	FLUDARABINE PHOSPHATE	FLUDARABINE PHOSPHATE
GLUCOCORTICOIDS	FLUDROCORTISONE ACETATE	FLUDROCORTISONE ACETATE
ANTIDOTES/DETERRENTS,OTHER	FLUMAZENIL	FLUMAZENIL
ANTI-INFLAMMATORIES,NASAL	FLUNISOLIDE	FLUNISOLIDE
GLUCOCORTICOIDS	FLUOCINOLONE ACETONIDE	FLUOCINOLONE ACETONIDE
GLUCOCORTICOIDS	FLUOCINOLONE ACETONIDE	FLUOCINOLONE/SHOWER CAP
ANTI-INFLAMMATORIES,TOPICAL OTIC	FLUOCINOLONE ACETONIDE OIL	FLUOCINOLONE ACETONIDE OIL
ANTI-INFLAMMATORY,TOPICAL	FLUOCINONIDE	FLUOCINONIDE
ANTI-INFLAMMATORY,TOPICAL	FLUOCINONIDE EMOLLIENT	FLUOCINONIDE/EMOLLIENT BASE
ANTI-INFLAMMATORY,TOPICAL	FLUOCINONIDE-E	FLUOCINONIDE/EMOLLIENT BASE
FLUORIDE	FLUOR-A-DAY	SODIUM FLUORIDE/XYLITOL
FLUORIDE	FLUORIDE	SODIUM FLUORIDE
FLUORIDE	FLUORITAB	SODIUM FLUORIDE
ANTI-INFLAMMATORIES,TOPICAL OPHTHALMIC	FLUOROMETHOLONE	FLUOROMETHOLONE
ANTINEOPLASTICS,ANTIMETABOLITES	FLUOROPLEX	FLUOROURACIL
ANTINEOPLASTICS,ANTIMETABOLITES	FLUOROURACIL	FLUOROURACIL
ANTIDEPRESSANTS/SNRI	DULOXETINE HCL	DULOXETINE HCL
ANTIDEPRESSANTS/SSRI	ESCITALOPRAM OXALATE	ESCITALOPRAM OXALATE
ANTIPSYCHOTICS, 1ST GENERATION	FLUPHENAZINE DECANOATE	FLUPHENAZINE DECANOATE
ANTIPSYCHOTICS, 1ST GENERATION	FLUPHENAZINE HCL	FLUPHENAZINE HCL
BENZODIAZEPINE DERIVATIVE SEDATIVES/HYPNOTICS	FLURAZEPAM HCL	FLURAZEPAM HCL
NONSALICYLATE NSAIS,ANTIRHEUMATIC	FLURBIPROFEN	FLURBIPROFEN
OPHTHALMICS,OTHER	FLURBIPROFEN SODIUM	FLURBIPROFEN SODIUM
ANTINEOPLASTIC,OTHER	FLUTAMIDE	FLUTAMIDE
GLUCOCORTICOIDS	FLUTICASONE PROPIONATE	FLUTICASONE PROPIONATE
ANTILIPEMIC AGENTS/STATIN	FLUVASTATIN SODIUM	FLUVASTATIN SODIUM

Figure 4. Master Dataset for Medication References

b). Falling Prediction Dataset

For falling prediction, two datasets were used for building up the predicative model. The first dataset used was the GAP dataset. This is the same dataset used for the first phase of this project.

However, unlike the simplified GAP dataset that included the medication and patient ID only, as was shown before, the complete GAP dataset has 301 rows with 87 columns (Figure 5).

	id	cancertype	stage	stage_other	chemo	monoclonal_ab	hormonal_tx	oral_tx	radiation_tx	treatment_type	...	assessment_fall	FH1	FH1aMonths	FH
0	96.0	GI	2	NaN	1	0.0	0.0	0.0	NaN	1	...	1.0	0.0	NaN	
1	97.0	Lung	2	NaN	1	1.0	0.0	0.0	NaN	2	...	1.0	0.0	NaN	
2	98.0	Lung	1	NaN	1	0.0	0.0	0.0	NaN	1	...	1.0	1.0	NaN	
3	99.0	GI	2	NaN	1	1.0	0.0	1.0	NaN	2	...	1.0	0.0	NaN	
4	100.0	Other	2	NaN	1	0.0	0.0	1.0	NaN	2	...	1.0	1.0	1.0	

5 rows × 87 columns

Figure 5. Schema of the complete GAP dataset

Despite the number of attributes in the dataset, we found that a variety of variables in the dataset were the categorical variables coded by 0 and 1 instead of numerical variables. Besides, we counted the number of missing variables included in the dataset and found that over half of the variables had significant amount of missing values, shown in figure 6.

id	2
Assessment	0
ImpairedPolypharmacy	0
ImpairedBOMC	0
ImpairedMiniCog	0
ImpairedWeight	0
ImpairedBMI	0
ImpairedMNA	0
ImpairedTUG	0
ImpairedSPPB	0
ImpairedADL	0
ImpairedIADL	0
ImpairedPH	0
ImpairedFalls	0
ImpairedCom	0
ImpairedGAD7	0
ImpairedGDS	0
ImpairedMS	0
KPS	1
Grade	2
Marital	2
Live	2
LiveOther1	289
LiveOther2	293
Health	4
Employment	2
EmploymentOther	292
Driving	2
Age	2
Feel	6
ZipCode	286
Gender	2
Ethnicity	2
Race	3

Insurance	2
InsuranceOther	214
Income	3
Living	2
Services	6
Hospital	137
HospitalCost	280
NursingHome	136
NursingHomeCost	300
Visits	139
VisitsCost	238
Sx	139
SxCost	284
Dental	137
DentalCost	262
Meds	136
MedsCost	187
HomeCare	137
HomeCareCost	298
Rehab	136
RehabCost	296
Meals	136
MealsCost	295
Transport	136
TransportCost	292
OtherExpDesc	289
OtherExp	173
OtherExpCost	289
Help	131
HelpWho	294
HelpAmount	294
DelayMed	130
IncomeAvail	129
Enough	129

Figure 6. Number of Missing Values by Attribute

Due to the large number of missing value and categorical variables, we decided to develop a data cleaning methodology before starting any analysis. We decided to fill up the missing value by taking the average of the column where missing value is located for numerical variables. For categorical variables encoded by 1 and 0, the missing values were replaced by the column mode instead. Aside from addressing the missing values, other problems we identified was that certain categorical variables had very messy level factors, which could bring more noise into the prediction. For example, for the insurance plan feature could have more than ten different levels. This occurred because insurance plan feature was collected from a clinical questionnaire for each patient to fill out. Some patients had very complicated insurance situation or simply mistakenly chose more than one options, which made this feature very messy, which was shown in Figure 7. In order to simplify the problem, we decided to decrease the number of levels for categorial variables like insurance plan. The most frequent four response remained the same while the rest of insurance plan options were combined to form a new level called "other."

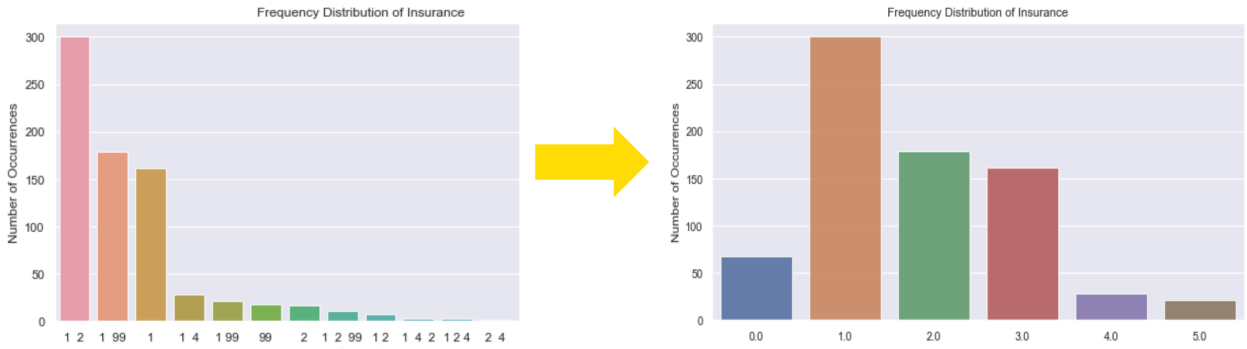

Figure 7: Categorical variables like insurance plan have messy factor levels

3. Exploratory Analysis

After some initial processing, we now moved on to do some explanatory analysis on the GAP dataset we used for this project.

a). Demographic Data Analysis

The first question to ask when analyzing a clinical medicine dataset is how well the patients in the clinical trial represent the demographics of the population. Therefore, we used the bar plot to visualize the gender and sex of the patients involved in this study (Figure 8). The figure below showed that our dataset fairly and equally represented both male and female in a 50:50 ratio. Both gender group had similar means and standard deviation for age.

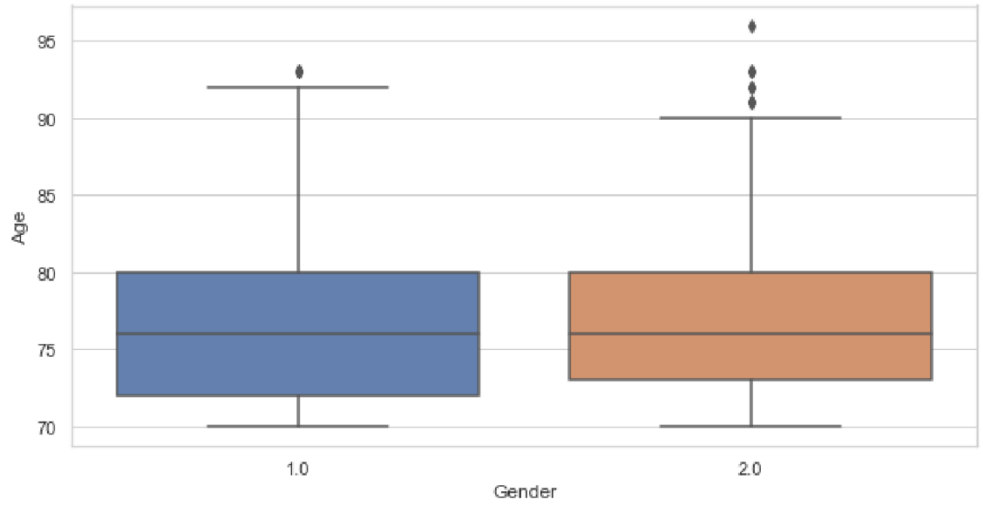

Figure 8. Demographic Analysis for the GAP dataset (Mean +/- SEM)
*1 = male; 2 = female

b). Feature Selection

Among a large number of variables and features existed in our dataset, we wanted to understand if there is any covariances between different variables. If features were related to each other in some way, such correlation could have a significant effect on our model performance. Besides, reducing the dimensionality of dataset could solve the "curse of dimensionality" problem. Therefore, we decided to apply a couple of feature selection algorithm to reduce the number of attributes and find the variables most relevant to our research.

The first feature reduction algorithm we applied was called RFECV (Recursive Feature Elimination Using Cross Validation). RFECV returned a rank of feature importance. RFECV worked by iteratively eliminated the variables and calculated the performance of tree-based model tested on the same training set. After a bunch of iterations, it could find out which variables contributed most to models with the best performances. The result of RFECV could be found in Figure 9.

Interesting, among the 20 most important features selected by the RFECV algorithm, 14 of them were clinical observations found to be closely correlated with falls in the previous literatures. This was a good sign because our result validated that clinical features were indeed reliable indicator of falling to a great extent.

An alternative way to find the covariance between variables was to plot a correlation matrix and sorted out the pair of variables with a high correlation coefficient (Figure 10). Based on the features selected from the RFECV algorithm, we plotted a correlation map. A high correlation between FH2, FH3 and fall were discovered. FH2 and FH3 were two indexes based on a question from the falling questionnaire. Basically, FH2 and FH3 measured patients' fear level of falling. A high correlation indicated that patients who were worried about falling more were more likely to fall compared to patients who didn't fear that much. Other than that, low correla-

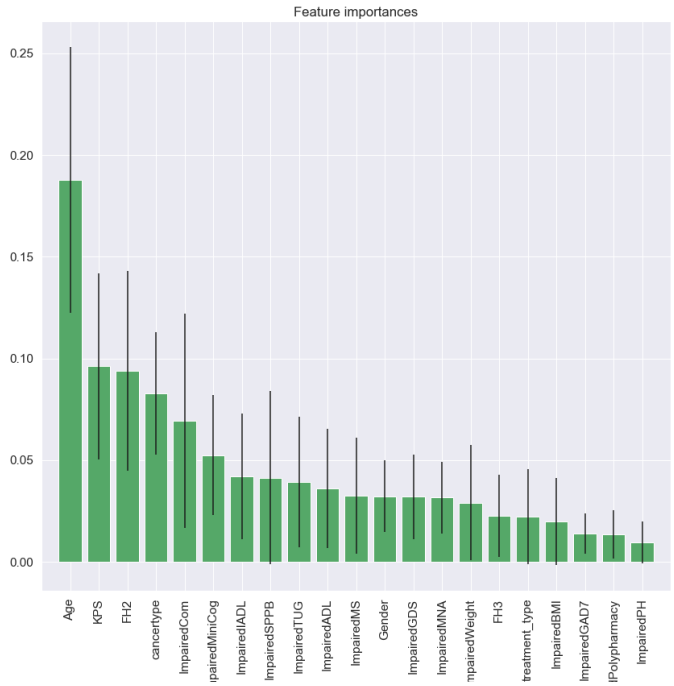

Figure 9 RFECV feature selection algorithm ranked the 20 most important features

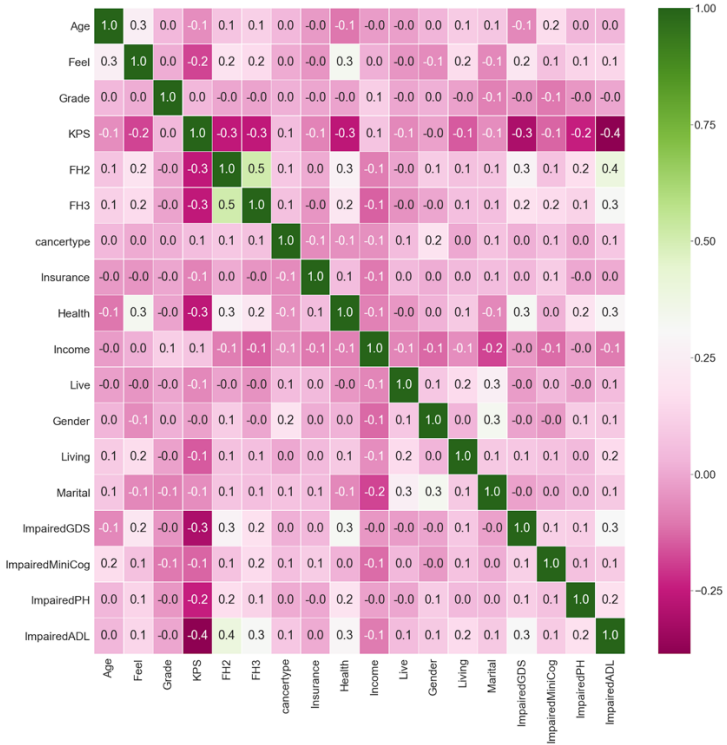

Figure 10: The Correlation Map between each pair of variables

-tion was observed in other pairs. A low correlation could be a good thing because that indicated that we successfully removed features that had linear relationship through feature selection process. In other words, the dimensionality reduction step was pretty successful.

4. Model Development

a). Auto Cleaning Algorithm

The first part of this auto cleaning algorithm basically iteratively compared the medication to both trade name and generic name in the master list. If a match was found in the trade name, the algorithm would automatically find the generic name corresponding to the trade name in the master list to convert it to the generic name. If a match was found in the generic name directly, nothing needed to be done. However, in the case that no matches were found in the master list at all, our algorithm would automatically connect to a Python library known as difflib. The difflib was a text alignment algorithm used to evaluate the difference between two words. Basically, we wanted to use difflib to calculate the similarity score between the unmatched words in the uncleaned dataset to every medication in the master drug list. After that, the algorithm would return three words with the highest similarity score to the unmatched (misspelled) medication. Based on the threshold we set for the algorithm, it can either automatically decide to give one clear specific suggestion for revision or give multiple suggestions of possible correction to let the users to determine which correction to take in the extremely complex situation. (An image representation of how our algorithm worked was found in Figure 11)

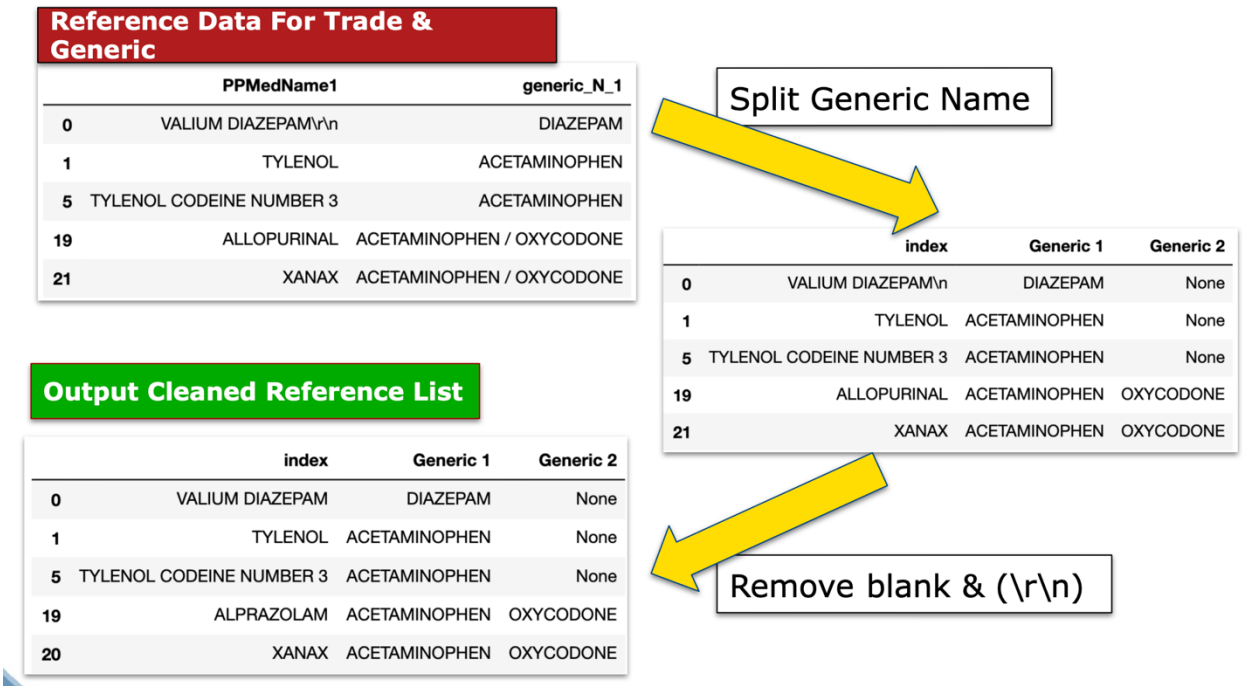

Figure 11. The flowchart illustrated how our algorithm worked

Aside from the autocorrection part, our algorithm could also return the medication list grouped by patients, so that clinician could better visualize what are the generic names of a list of medications a patient is taking (Figure 12).

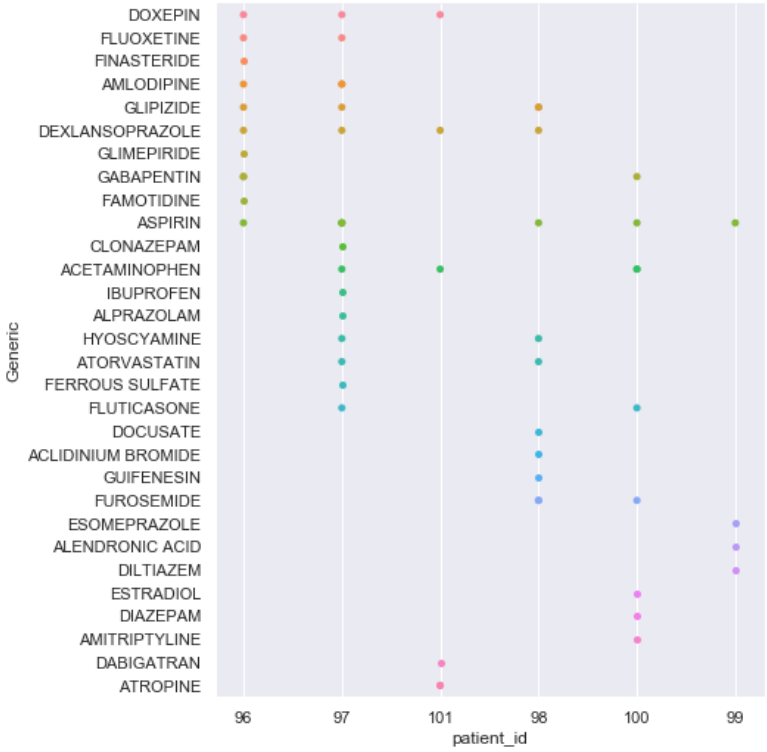

Figure 12 Visualize the number of medication patients took

b) Falling Rate Prediction Model

After feature selection was done, we started to train the machine learning models. However, one problem we quickly discovered after the initial attempt of training was that the dataset was highly imbalanced in terms of the number of falling patients and not falling patients. As a result, the model trained based on highly skewed dataset would give a biased prediction. Figure 13 showed the distribution of falling patients and control group in both GAP and COACH dataset.

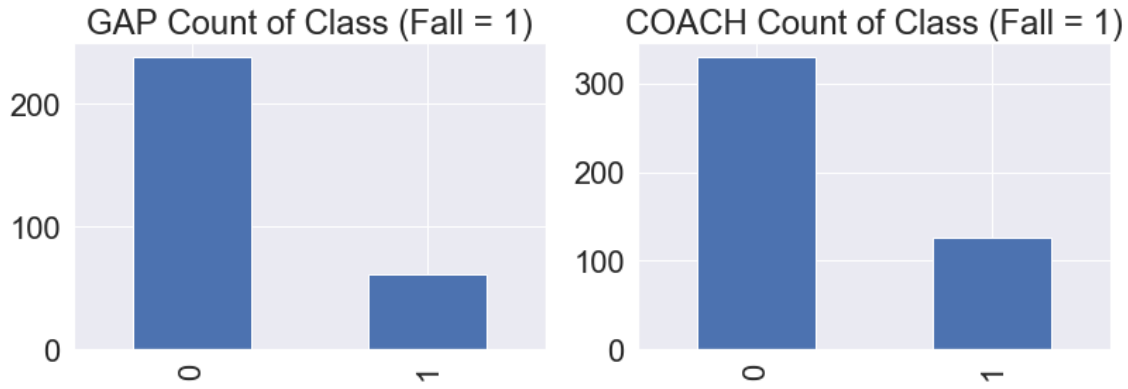

Figure 13 Both GAP and COACH dataset are highly imbalanced

To solve the imbalanced dataset problem for model training, we applied an under-sampling technique known as *instance hardness threshold* (See Figure 14.) The under-sampling techniques effectively removed the points near the decision line boundary until a relatively balanced training dataset was formed. However, one problem about under-sampling would be that it significantly reduced the number of data points used to train the model and could oversimplify the real-life situation.

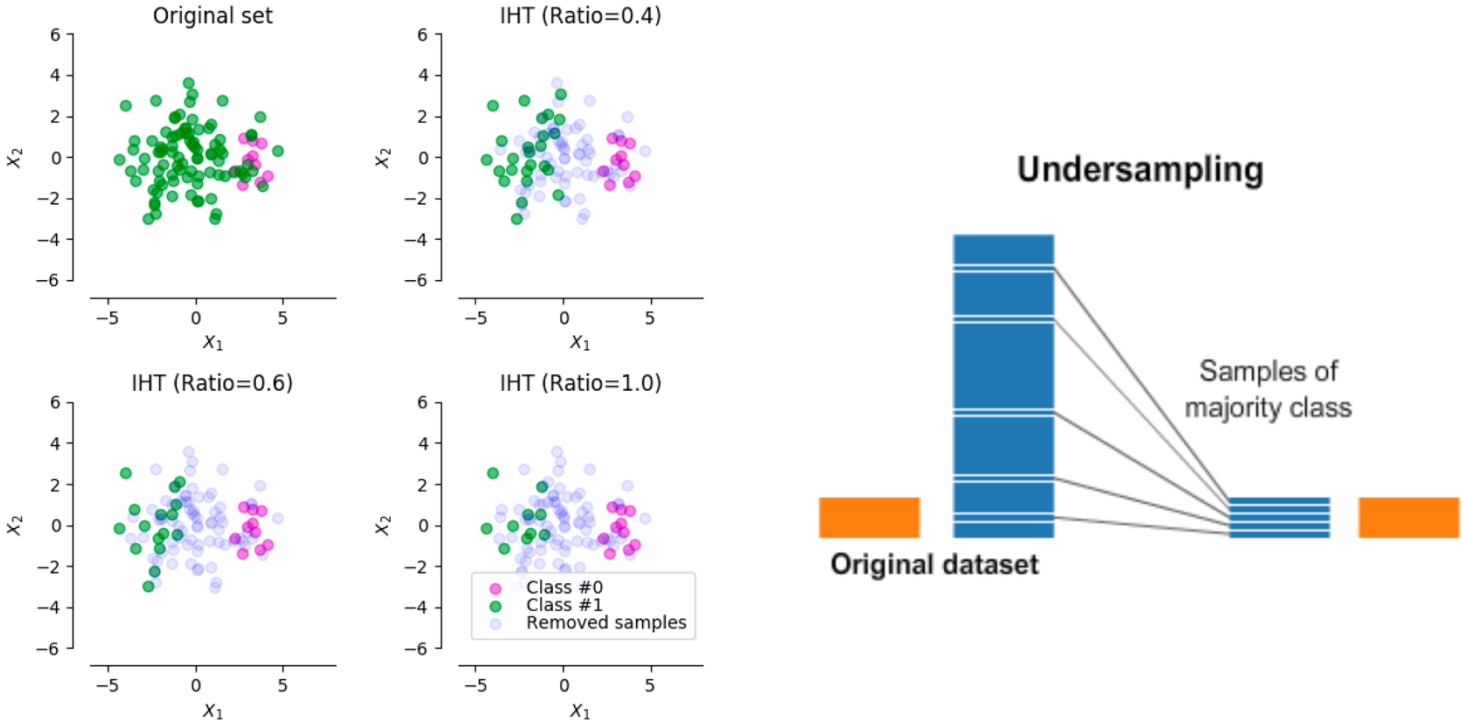

Figure 14: An illustration of how Instance Hardness Threshold worked in removing data points near decision boundary line

5.Performance and Results

Now that dataset was clean, important features were selected and class imbalance problems had been solved, the performance of machine learning model we trained to predict the falling rate in patients had significantly increased. Attached were the confusion matrix and model performance generated based on the logistic regression model (Figure 15& Figure 16).

	precision	recall	f1-score	support
0	0.87	0.43	0.57	112
1	0.33	0.82	0.47	39
micro avg	0.53	0.53	0.53	151
macro avg	0.60	0.62	0.52	151
weighted avg	0.73	0.53	0.55	151

Figure 15: Model Performance Returned by Logistic Regression Model

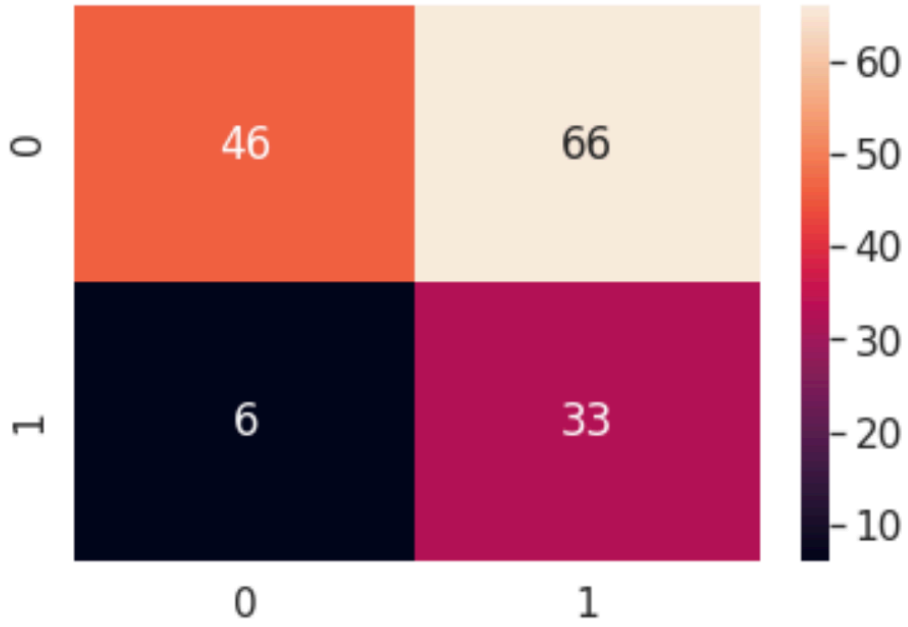

Figure 16: Confusion Matrix Generated Based on the Logistic Regression Model

In the clinical data context, the ideal situation would be that the model could have both high sensitivity and high specificity. However, for many models in disease prediction, there was always a trade-off between sensitivity and specificity. For example, in the initial model we trained without applying the under-sampling techniques, our model gave a very high specificity rate (precision rate), but super low sensitivity rate (recall rate). This can be problematic because the model basically labeled every patient as control (not falling) group and because the number of patients falling was pretty low in the population, the model had a pretty good overall accuracy rate. However, a model that only predicted one outcome, was in no sense, a practically useful model. That's why in our model, we wished to maximize the sensitivity rate (sensitivity) in the hope that specificity rate wouldn't be ruined. It turned out our model had a pretty high recall rate for falling patient and high precision rate for patients who were not falling, which was a pretty ideal outcome for clinical prediction problem like this.

AUC score was another indicator of how powerful a predictive model was in the clinical medicine. Our model reported an AUC score of 0.72, which was actually better than similar researches reported in the previous literature (Around 60%). (Marchollek., et al 2012; Mateen., et al 2016)

Aside from logistic regression model, we also trained the training dataset using MLP and random forest algorithms. It turned that random forest gave the highest performance (recall rate) compared to the other two model (Figure 18). However, one advantage of logistic regression over random forest was that logistic regression's results could be easily interpreted with odds ratio and with an output probability.

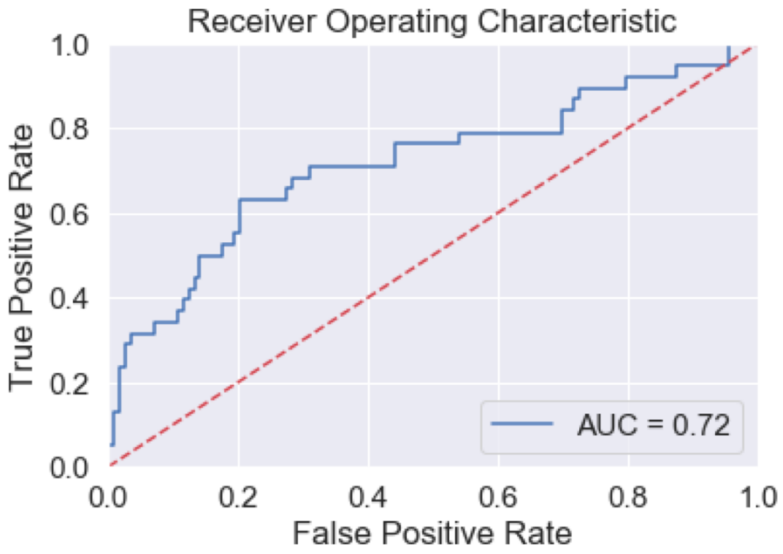

Figure 17. AUC Score for Logistic Regression Based Prediction Model

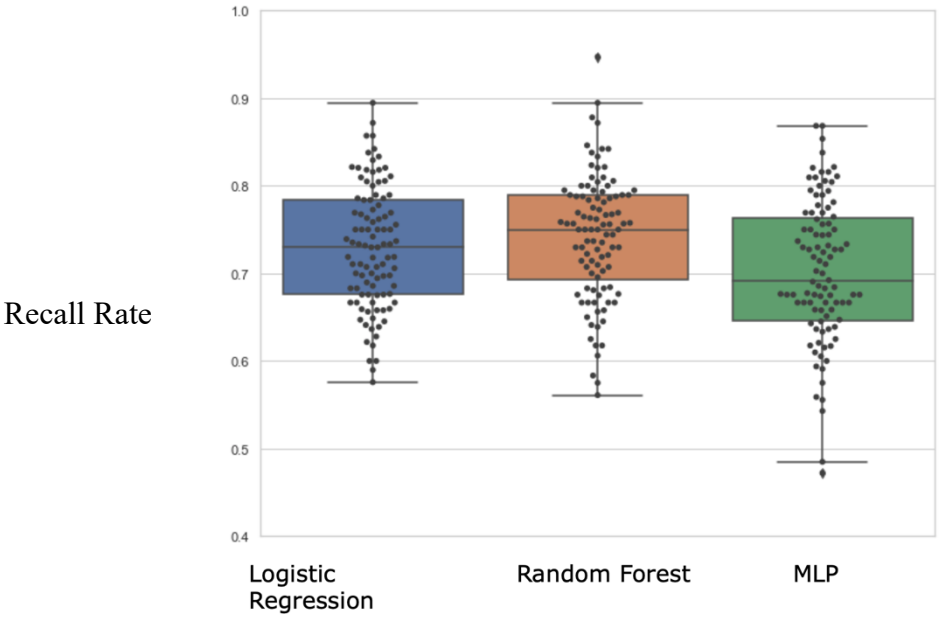

Figure 18. Recall Rate Reported by Different Machine Learning Algorithm

Finally, we used the logistic regression to calculate the probability for each patient to be classified into falling and not falling class. Due to the easy interpretability, we believed it could be a potentially helpful tool for clinicians when making diagnosis of the patients. For example, when clinicians input a bunch of clinical features and lab results into the computer, our model could automatically output a possibility of how likely this patient was going to fall. The illustration could be found in Figure 19.

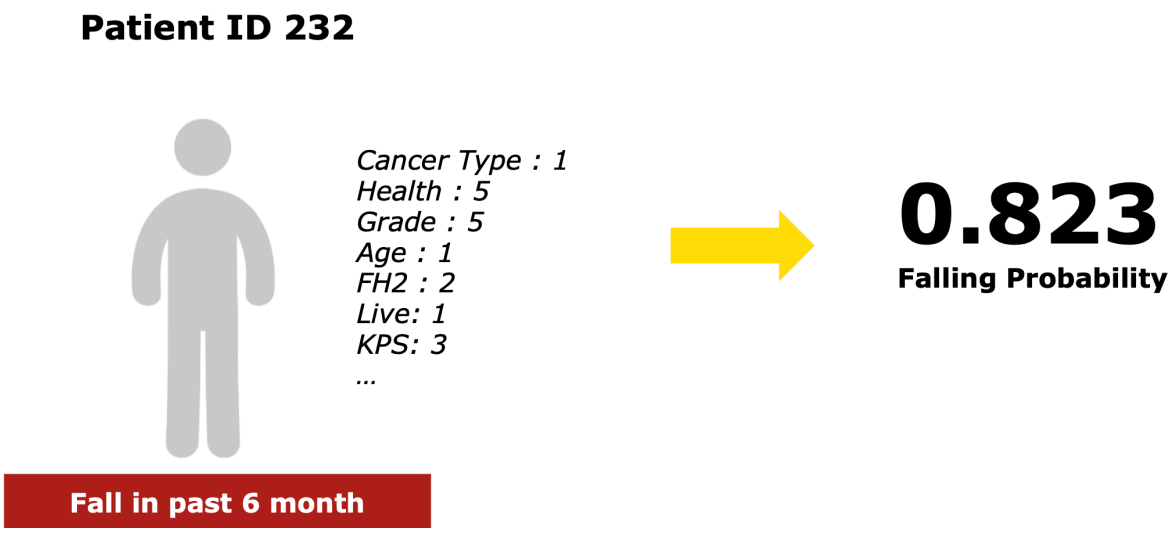

Figure 19. Illustration of our Model in Real Life Clinical Application

6.Conclusion and Next Step

To the best of our knowledge, this is the first study that used machine learning algorithms like random forest and MLP for predicting the falling rate among the aging cancer population. Meanwhile, our model performance is much better compared to most of the previous studies using other predictive models like regression model. Our study shows that machine learning has large predictive power and potential in future clinical study. Meanwhile, we would want to optimize the model by further improving its specificity in the future.

7.Contribution

We wanted to thank Sixu and Zhikang for their contribution to the first phase of the project. I was responsible for developing the hypothesis and trained and predictive model for the second phase of this project, together with Boyu. Meanwhile, I was responsible for doing literature review to compare our results and methodology to the previously published papers in the similar medical fields.

8. Reference

1. CDC. Fatalities and Injuries from Falls Among Older Adults -- United States, 1993-2003 and 2001-2005. Morbidity and Mortality Weekly Report. 2006; 55(45):1221–4. [PubMed: 17108890]

2. Ganz DA, Bao Y, Shekelle PG, Rubenstein LZ. Will my patient fall? JAMA: The Journal of the American Medical Association. 2007; 297(1):77–86.

3. Tinetti ME, Williams CS. Falls, injuries due to falls, and the risk of admission to a nursing home. N Engl J Med. 1997; 337(18):1279–84. [PubMed: 9345078]