Final Capstone Project Report

Boyu Liu (URMC-oncology)

Introduction

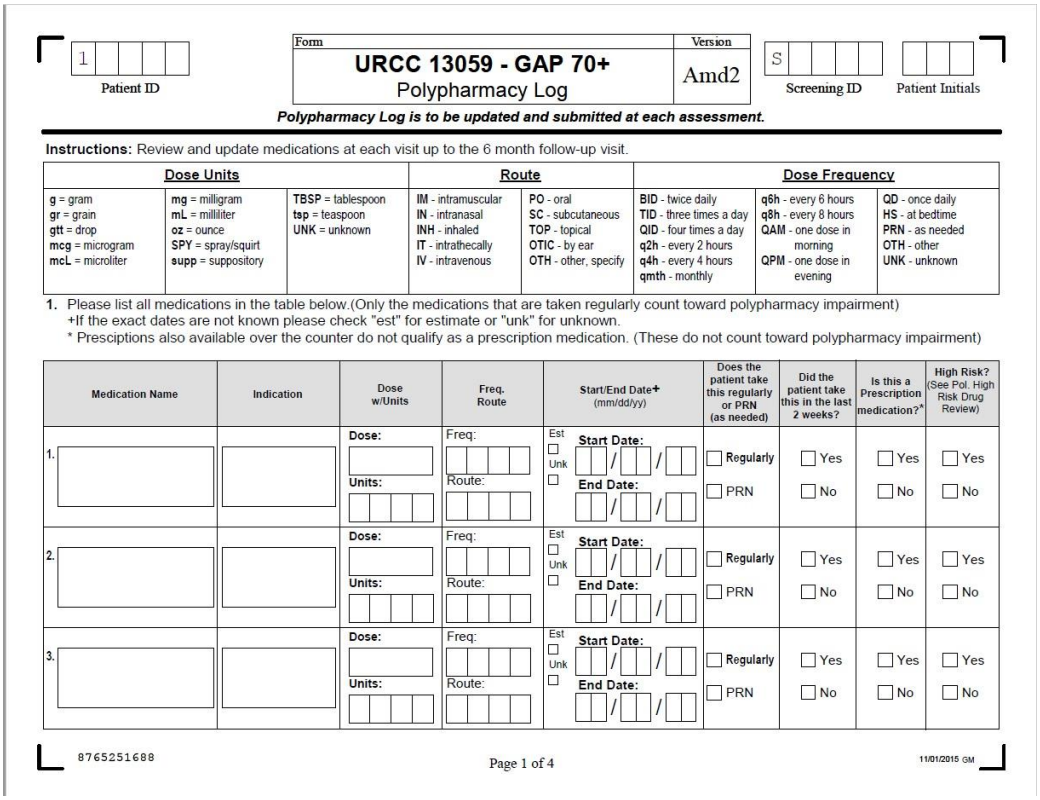
In this project, we are focusing on solving the polypharmacy problem of cancer patients. Polypharmacy is the simultaneous use of multiple medications. [1] However, the risks of polypharmacy overshadow the benefits of it. It may increase the risk of death of patients. Also, polypharmacy has other side effects, such as causing patients to fall.

For this purpose, the URMC team gives us two stages in this capstone project. Objective One is to do data cleaning. We need to design an algorithm which takes trade names as input and return the correct ingredients as generic names. Also, we need to correct the misspelled trade names in the dataset based on the medical dictionary that URMC team provides. After changing the trade names to corresponding generic names, we need to group the result by the patient IDs to further studies.

Objective Two is to explore the dataset, form and resolve a hypothesis which might be interesting for ourselves. The hypothesis that we formed should also be useful for clinical use. After working on the dataset for almost one month, we completed the Objective One and had already formed several possible hypotheses.

Data Collection

For data collection part, URMC team collected the data using some specific types of forms, which will be filled for each cancer patient involved in this cancer study. After filling the forms, they will scan these forms into computer. That is the reason why there might be some misspelled names in the dataset. Human handwritings sometimes are unrecognizable for computer. I will provide an example of those forms in Figure 1 as follow.



1 Patient ID

Form: URCC 13059 - GAP 70+ Polypharmacy Log | Version: Amd2

S Screening ID | Patient Initials

Polypharmacy Log is to be updated and submitted at each assessment.

Instructions: Review and update medications at each visit up to the 6 month follow-up visit.

Dose Units			Route		Dose Frequency		
g = gram **gr** = grain **gtt** = drop **mcg** = microgram **mcL** = microliter	**mg** = milligram **mL** = milliliter **oz** = ounce **SPY** = spray/squirt **supp** = suppository	**TBSP** = tablespoon **tsp** = teaspoon **UNK** = unknown	**IM** - intramuscular **IN** - intranasal **INH** - inhaled **IT** - intrathecally **IV** - intravenous	**PO** - oral **SC** - subcutaneous **TOP** - topical **OTIC** - by ear **OTH** - other, specify	**BID** - twice daily **TID** - three times a day **QID** - four times a day **q2h** - every 2 hours **q4h** - every 4 hours **qmth** - monthly	**q6h** - every 6 hours **q8h** - every 8 hours **QAM** - one dose in morning **QPM** - one dose in evening	**QD** - once daily **HS** - at bedtime **PRN** - as needed **OTH** - other **UNK** - unknown

1. Please list all medications in the table below.(Only the medications that are taken regularly count toward polypharmacy impairment)
   +If the exact dates are not known please check "est" for estimate or "unk" for unknown.
   * Presciptions also available over the counter do not qualify as a prescription medication. (These do not count toward polypharmacy impairment)

	Medication Name	Indication	Dose w/Units	Freq. Route	Start/End Date+ (mm/dd/yy)	Does the patient take this regularly or PRN (as needed)	Did the patient take this in the last 2 weeks?	Is this a Prescription medication?*	High Risk? (See Pol. High Risk Drug Review)
1.			Dose: Units:	Freq: Route:	Est ☐ Unk ☐ Start Date: / / End Date: / /	☐ Regularly ☐ PRN	☐ Yes ☐ No	☐ Yes ☐ No	☐ Yes ☐ No
2.			Dose: Units:	Freq: Route:	Est ☐ Unk ☐ Start Date: / / End Date: / /	☐ Regularly ☐ PRN	☐ Yes ☐ No	☐ Yes ☐ No	☐ Yes ☐ No
3.			Dose: Units:	Freq: Route:	Est ☐ Unk ☐ Start Date: / / End Date: / /	☐ Regularly ☐ PRN	☐ Yes ☐ No	☐ Yes ☐ No	☐ Yes ☐ No

8765251688 Page 1 of 4 11/01/2015 GM

Figure 1. Form for collecting polypharmacy log of patients [2]

Therefore, from the form provided above, because of that disadvantage of manual input by human, we can see the importance of correcting the misspelled names for this project.

Also, other data related to biomarkers, such as BMI, MNA, weight, are collected in labs of URMC team.

Data Preparation

Data visualization and Exploratory analysis

For Objective One, currently, we are basically provided several datasets by URMC team for the use of project. Three of them are dictionaries for identifying the misspelled names and for corresponding relation between trade names and generic names. Also, we have a dataset of 60 cancer patients, which are clean as a test dataset for our data cleaning algorithm. There are 28 columns and 480 rows in this dataset. Meanwhile, we also have a bigger dataset with only two columns, which are Dummy ID of the patients and trade names of medications. URMC team wants us to form a reasonable data cleaning algorithm to apply on this dataset after testing on the 60-patients dataset.

Because the 60-patients dataset is clean, we group the dataset by patient ID and draw a graph to provide an illustration for the dataset as follow.

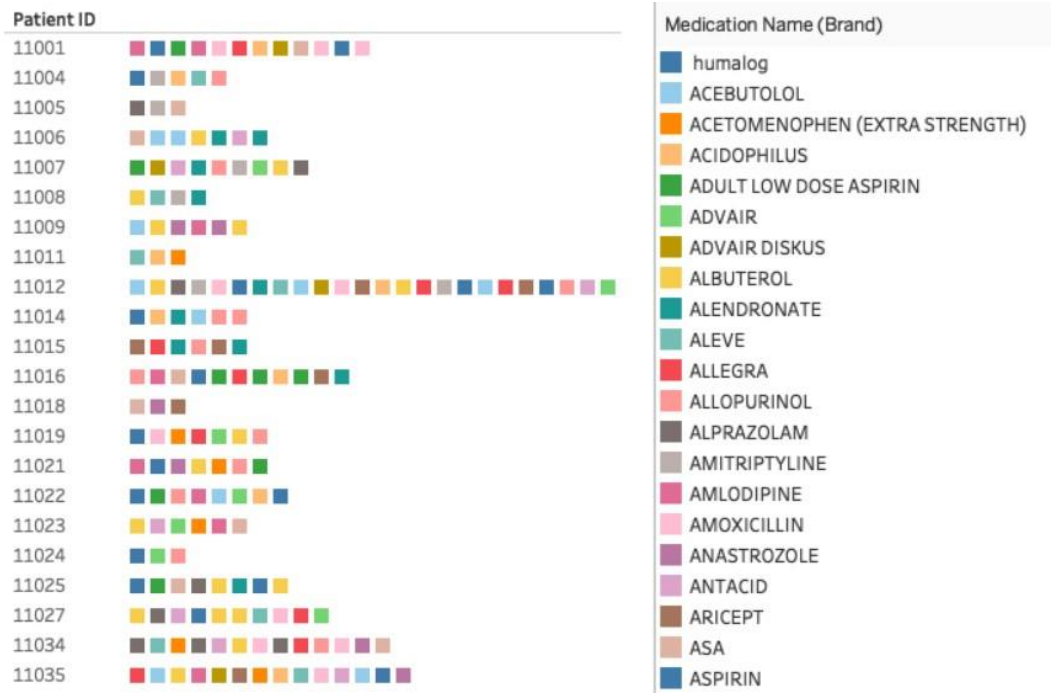


Figure 2. Part of illustration of 60 patients dataset grouped by ID

Meanwhile, the dictionary (master list) that the team gives us are very important, which shows the corresponding relationships between trade names and generic names. One trade name might be corresponding to two or more generic names. Also, there might be some generic names which are similar to trade names in the column of trade names.

For Objective Two, URMC team gave us two datasets. First one is called COACH, which contains 457-patients data. Second one is called GAP, which contains 302-patients data. Because features in these two datasets are almost the same, we decided to combine these two datasets.

Then, we began to draw a confusion matrix to illustrate the correlation among these features.

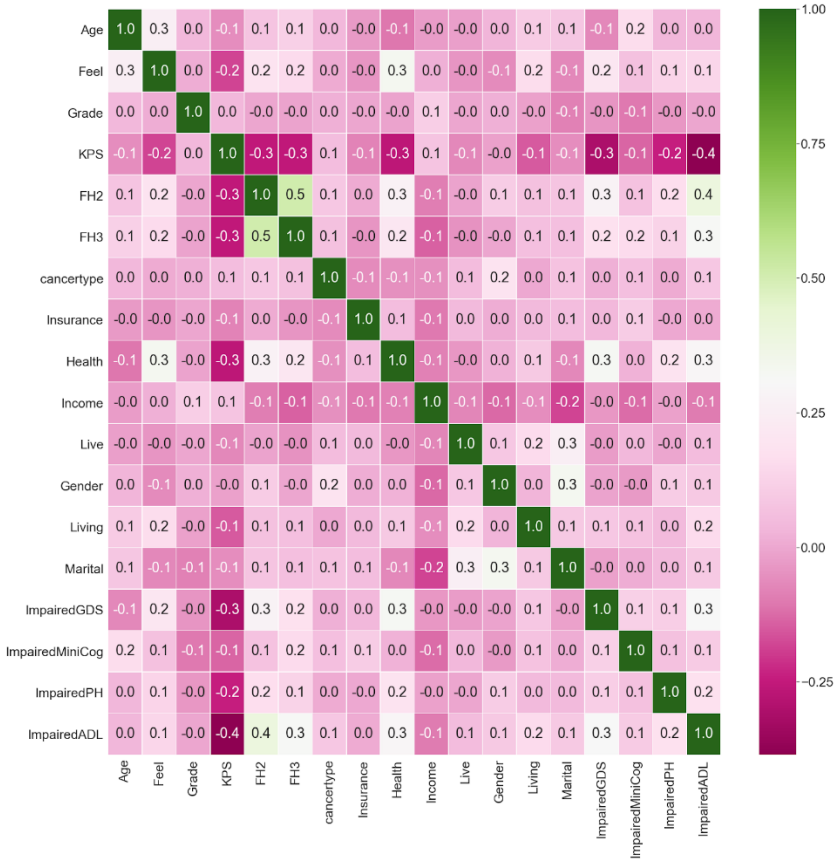


Figure 3. Part of the confusion matrix

Because of there are more than 70 features in the dataset, we cannot show such a great amount of feature's correlation in a single confusion matrix. In the concept of confusion matrix, we usually consider 0.5 as a hint which shows that there might be some correlation between two features. As a result, in Figure 4, we can see that FH2 and FH3 have a correlation of 0.5. Although it is not very high, in clinical area, we should consider there might be some correlation. FH2 is a question in survey, which asks people, in past 12 months, "are you afraid that you might fall". FH3 is also a question in survey, which asks people, "do you limit your activities because of being afraid of falling". We found out that these two are highly correlated with each other in clinical area. Then, we decide to use only one of them in our further modeling.

Data Reformat and Data cleaning

For Objective One, after basically understanding the datasets, we begin to create an algorithm to reformat the datasets for future use. Our algorithm is separated into two parts. The first part is to correct the misspelled names in the column of trade names using the given dictionary. The second part is to change the trade name to correct

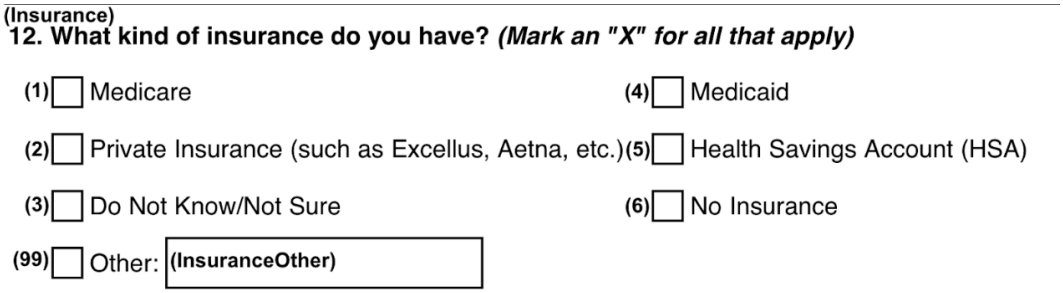
generic names based on the dictionary.

To be specific, we firstly read the dictionary from the csv file and use it as our dictionary in Difflib package[3] for correct the misspelled names. Difflib is a python package, which is designed to match misspelled words based on given dictionary and difference level(cutoff). Also, we match the trade names with not only the correct trade names but also the correct generic names in the dictionary. That is because we found out there might also be some generic names in the drug name column in the dataset.

Then, we use the dictionary of corresponding relations between trade names and generic names to change the trade names to one or more generic names.

For Objective Two, because of there are so many missing values in the whole dataset, data cleaning is very crucial for this dataset. There are three types of features in the dataset: numeric, categorical and nonnumeric. For numeric features, the strategy is to replace the missing values with mean of elements in that feature. For categorical features, the strategy is to replace missing values with mode of elements in that feature. For nonnumeric features, we decide to transfer them to categorical features.

The data processing of nonnumeric feature is the hardest part. Here is an example of a feature called Insurance, which is also called through a survey in Figure 5.


(Insurance)
12. What kind of insurance do you have? *(Mark an "X" for all that apply)*

(1) ☐ Medicare
(4) ☐ Medicaid
(2) ☐ Private Insurance (such as Excellus, Aetna, etc.)
(5) ☐ Health Savings Account (HSA)
(3) ☐ Do Not Know/Not Sure
(6) ☐ No Insurance
(99) ☐ Other: (InsuranceOther)

Figure 4. Illustration of Insurance feature

As we can see from figure above, it is a very complicate feature. It allows patients to choose more than one choice. As a result, we need to count the occurrence of each unique choice in our dataset. And then, we will choose top five most frequent choice and transform them into categorical feature, which is corresponding to one to five. For the remaining choices, we add the occurrence of each together and transform them into zero. As a result, we have a categorical feature ranged from 0 to 5.

Objective One – Data preprocessing

As we said before, after creating a data cleaning algorithm, we first need to test it on the 60-patients dataset to see whether the algorithm works well on real-life dataset.

Firstly, we use the difflib package with unique generic names dictionary to match the column of drug names. If exactly same names are found, they will be labeled as GEN_Found; if similar names are found based on 90% threshold that we set, they will be changed to matching names in the dictionary, and be labelled as GEN_Changed.

Their generic names will be directly changed to that matching names, which can be shown in the graph as follow.

	testing_misspelled_names	checked_name_list	generic_list	mark
0	AMLODIPINE	AMLODIPINE	[AMLODIPINE]	Gen_Found
1	ASPIRIN	ASPIRIN	[ASPIRIN]	Gen_Found
2	FINASTERIDE	FINASTERIDE	[FINASTERIDE]	Gen_Found
3	GABAPENTIN	GABAPENTIN	[GABAPENTIN]	Gen_Found
4	GLIMEPIRIDE	GLIMEPIRIDE	[GLIMEPIRIDE]	Gen_Found
5	LISINOPRIL	LISINOPRIL	[LISINOPRIL]	Gen_Found
6	METAPROLOL	METOPROLOL	[METOPROLOL]	Gen_Changed
7	OXYCODONE ACETAMINOPHEN	OXYCODONE ACETAMINOPHEN	[]	Unknown

Figure 5. Part of the result of first step

After that, we use the difflib package to match the words with trade names dictionary. At the same time, we will change the matching trade names directly to corresponding generic names based on the dictionary, which are stored in the generic list.

	testing_misspelled_names	Correction	new_generic_list	ans	new_mark
0	AMLODIPINE	AMLODIPINE	[AMLODIPINE]	[AMLODIPINE]	Gen_Found
1	ASPIRIN	ASPIRIN	[ASPIRIN]	[ASPIRIN]	Gen_Found
2	FINASTERIDE	FINASTERIDE	[FINASTERIDE]	[FINASTERIDE]	Gen_Found
3	GABAPENTIN	GABAPENTIN	[GABAPENTIN]	[GABAPENTIN]	Gen_Found
4	GLIMEPIRIDE	GLIMEPIRIDE	[GLIMEPIRIDE]	[GLIMEPIRIDE]	Gen_Found
5	LISINOPRIL	LISINOPRIL	[LISINOPRIL]	[LISINOPRIL]	Gen_Found
6	METAPROLOL	METOPROLOL	[METOPROLOL]	[METOPROLOL]	Gen_Changed
7	OXYCODONE ACETAMINOPHEN	OXYCODONE ACETAMINOPHEN	[FLUOXETINE]	[OXYCODONE, ACETAMINOPHEN]	Found
8	RANITIDINE	RANITIDINE	[RANITIDINE]	[RANITIDINE]	Gen_Found
9	SPIRONOLACTONE	SPIRONOLACTONE	[SPIRONOLACTONE]	[SPIRONOLACTONE]	Gen_Found

Figure 6. Part of the result of second step

However, after these two steps, we find out that our accuracy is not as high as we expected. After carefully browsing the result we have, we find out that there are some results, which are caused by different correct answers in these three dictionaries

17	COZAAR	COZAAR	[LOSARTAN POTASSIUM]	[LOSARTAN]	Found

Figure 7. Example of so-called incorrect answers

provided by URMC team, which can be shown as follow.

As we can see, both two answers are correct, which have been verified by ourselves in the dictionaries. Therefore, we can consider these cases as correct answers rather than wrong. Meanwhile, because the correct answers in 60-patients dataset are entered by researchers in URMC team by hand, there are some cases that we are unable to find corresponding generic names in the dictionaries and we leave the generic list column in blank. They just directly entered the brand name into correct generic names column.

VICTOZA	VICTOZA	[]	[LIRAGLUTIDE]	Unknown

Figure 8. Example of blank generic list

Similarly, we consider these as correct as well. After all the modification to the algorithm, we calculate our accuracy rate when applying this algorithm to 60-patients dataset. We achieve an accuracy rate of 0.887097 for the 60-patients dataset. Because of the relative high accuracy rate, we decide to apply this algorithm to that bigger dataset with more patients that URMC provides us.

First step of matching generic names:s

	Algorithm correction	Directed matching	Unknow value
0	390	3597	3680

Figure 9. Result of matching generic names

At the first step, we directly match 3597 words and correct 390 misspelled names.

Second step of matching trade names:

	Total Correction Number	Directed matching	Algorithm correction	Unknow value	Warnings
Times	5980	1785	208	1687	10

Figure 10. Result of matching trade names

After the second step, the total number of names that we match and correct is 5980. Judging from that, our algorithm performs well on the bigger dataset, even if further mediation may be possible.

Data preprocessing - Resampling

Because the dataset that we are given by URMC team are highly imbalanced, we need to preform some resampling techniques to balance the dataset for a reasonable result. In our project, we perform three types of resampling techniques.

SMOTE (Synthetic Minority Over-sampling Technique) [4]

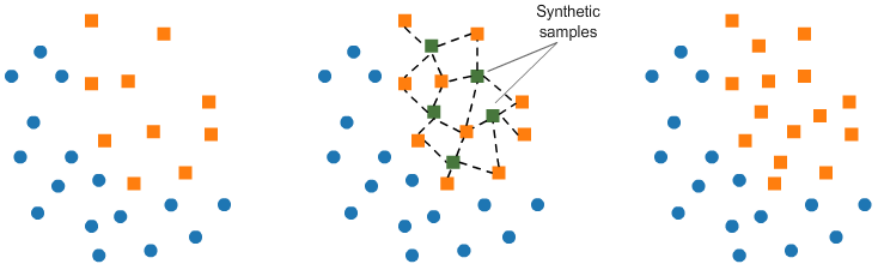


Figure 11. Illustration of SMOTE

SMOTE is an over-sampling technique, which will generate artificial data point in minority class based on KNN classifier. After generating synthetic samples, majority class and minority class become balanced.

Tomek Link Under-sampling Technique [5]

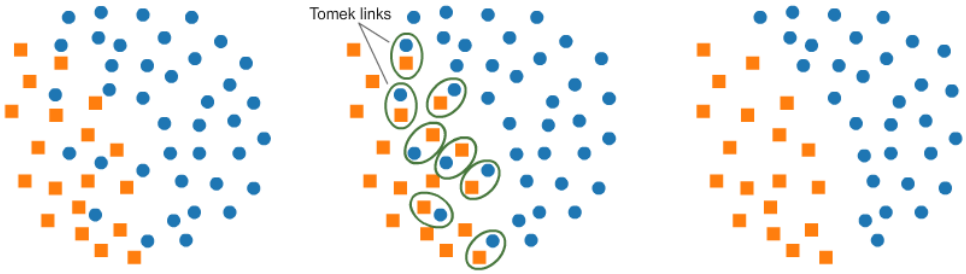


Figure 12. Illustration of Tomek Link

Tomek's Link is formed by two data points from different classes which are nearest neighbors of each other. This technique will eliminate the such link by eliminating data points from majority class. As a result, it will generate a cleaner class boundary for classification.

Instance Hardness Threshold [6]

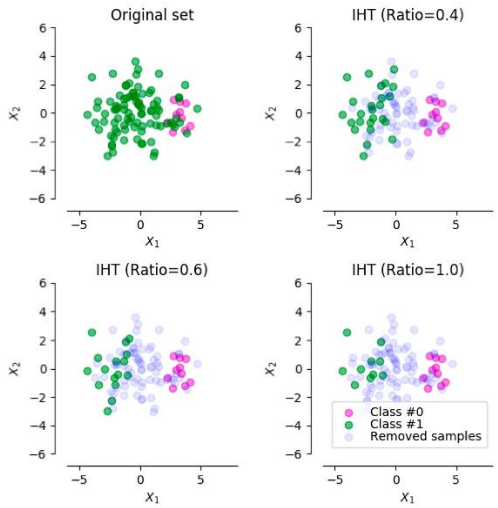


Figure 13. Illustration of Instance Hardness Threshold

Instance hardness threshold is an under-sampling technique which uses an estimator to estimate the instance hardness of the samples. It reduces less important data points in the dataset to perform under-sampling.

Data preprocessing - Feature Selection

Because of there are more than 70 features in the dataset, the dimensionality of this dataset is very high. Feature selection is important step to get a better prediction result. We used two feature selection techniques in our modeling.

RFECV (Recursively feature elimination using cross validation) [7]

It is a method that is widely used in data analysis. This method uses random forest classifier to do classification on the train dataset using cross validation. It can not only rank the feature importance but also return how many features are needed for best accuracy or largest AUC area.

After we run this algorithm on train dataset, we get the result as follow.

```
Optimal number of features : 16
Best features : Index(['FH2', 'Age', 'Gender', 'KPS', 'ImpairedMiniCog', 'ImpairedWeight',
       'ImpairedBMI', 'ImpairedMNA', 'ImpairedTUG', 'ImpairedSPPB',
       'ImpairedADL', 'ImpairedCom', 'ImpairedGDS', 'cancertype',
       'treatment_type', 'FH3'],
      dtype='object')
```

Figure 14. Result of RFECV

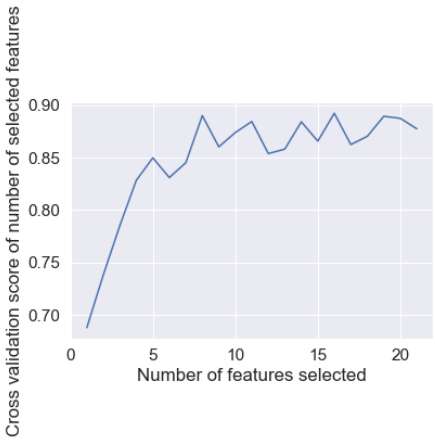


Figure 15. Graph of number of features vs. AUC

From the graph above, we can see that after number of features selected increases to 8, AUC score goes up and down between 0.85 and 0.90. That is because of high dimensionality of the dataset.

Tree-based Feature Selection

This is a feature elimination algorithm which also uses random classifier to compute the feature importance. It uses a parameter in random classifier in Sklearn package called feature_importances. The result is provided as follow:

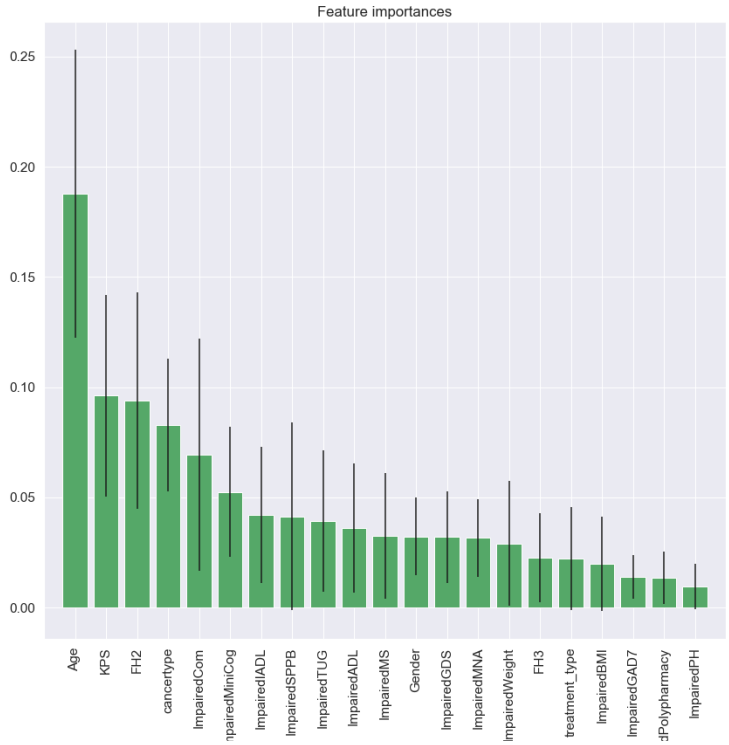


Figure 16. Graph of feature importance

Model Development

After smoothly completing the Objective One, we begin to change our focus to Objective Two. For Objective Two, we need to form a useful hypothesis after exploring the datasets. Thus, we decide to do some literature review to give us some inspirations about building the hypothesis. In several literatures, we find out that falls are very trending problems right now for cancer patients who take many kinds of medications. Our literature review summaries are provided as follow:

- Falls are one of the leading causes of injuries in the aging people; 30% aging people over 65 are victim of falls (Todd and Skelton, 2004); 10% of falls lead to serious damage. (Goldacre et al., 2002)
- Well studied predictors of falls include: age, dementia, physical performance, frailty, visual and musculoskeletal disorder, and medications such as psychotropic drugs. (Lastrucci et al., 2017)
- Measurements include: Frontal Assessment Battery (Kataoka, 2015), Short Physical Performance Battery (Pua, 2018), Fried Frailty Index Components (Sharma et al., 2019), etc.
- However, to the best of our knowledge, no classification model of falling has been published up to the date using multiple predictors.

Based on literature reviews we have done, we decide to build a predictive model which uses dataset that URMC team gave us to predict whether an elder patient is going to fall.

First Modeling

In our first try, we did not use any resampling techniques or feature selection. Basically, we perform random forest classifier on the dataset to make prediction. The result is provided as follow:

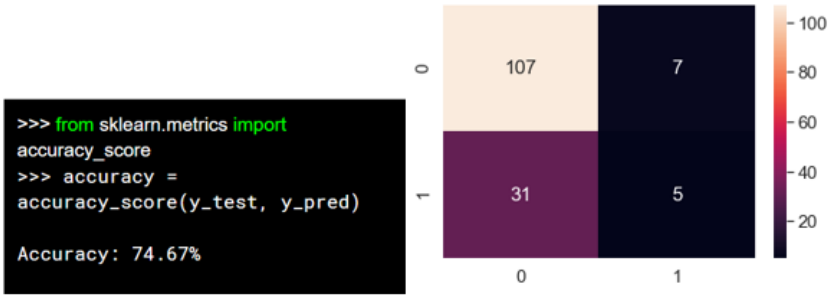


Figure 17. Accuracy and confusion matrix of random forest on raw data

After we looked at the confusion matrix, although the accuracy is relatively high, almost every data point in the test set have been classified to class 0. That is the reason why the accuracy score seems to be very high.

Therefore, we came back to data analysis to the whole dataset.

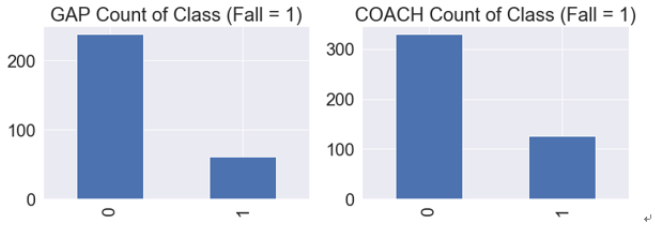


Figure 18. Class distribution of datasets we are given

Second Modeling

From the class distribution of the datasets, we can see that the datasets are highly imbalanced. Thus, we decided to perform resampling techniques that I mentioned before. We firstly perform SMOTE to over-sample the minority class and then perform Tomek Link to under-sample the majority class for both train set and test set. As a result, by combining these two techniques, the effect of resampling has been minimized. The prediction result is provided as follow:

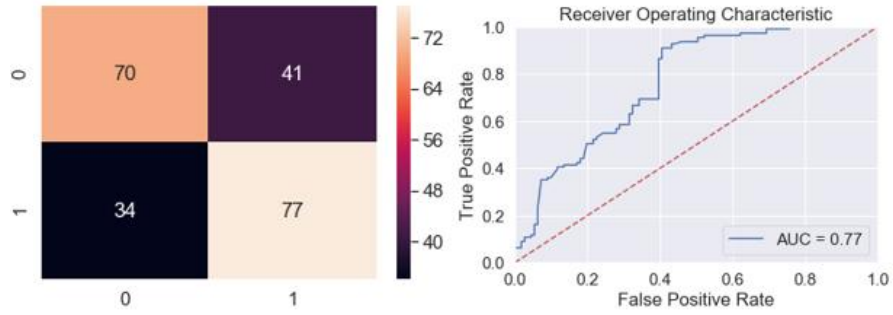


Figure 19. Confusion matrix and ROC curve of random forest classifier

The result we got is pretty decent; however, after discussing with Prof. Ajay, it is unsuitable to perform any resampling techniques to test set. Therefore, we perform

these two resampling techniques only to train set. The result is provided as follow:

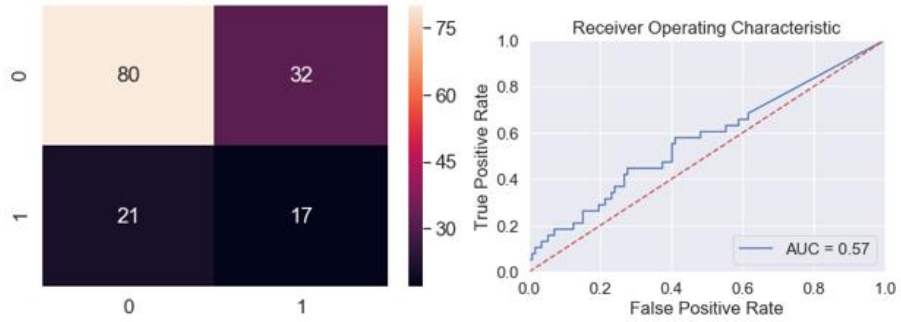


Figure 20. Confusion matrix and ROC curve of resampling on train set

When only applying resampling on train set, the result is not even better than guessing. Therefore, we began our third try.

Third Modeling

In our third try, we discussed with professor and concluded that we should only focus on the recall rate of our model, especially for medical dataset.

$$\text{Recall} = \frac{tp}{tp + fn}$$

Figure 21. Formula of recall

In medical area, people care a lot about true-positive and false-negative rather than false-positive and true-negative. That is because doctors prefer high precision when predicting that patients are going to fall rather than high precision when predicting that patients are not going to fall. Meanwhile, we also use Logistic Regression using cross-validation[13] to run the algorithm multiple times for the sake of minimizing the effect of randomness. We also use instance hardness threshold to under-sample train set. The result is provided as follow:

	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 22. Result of Logistic Regression CV

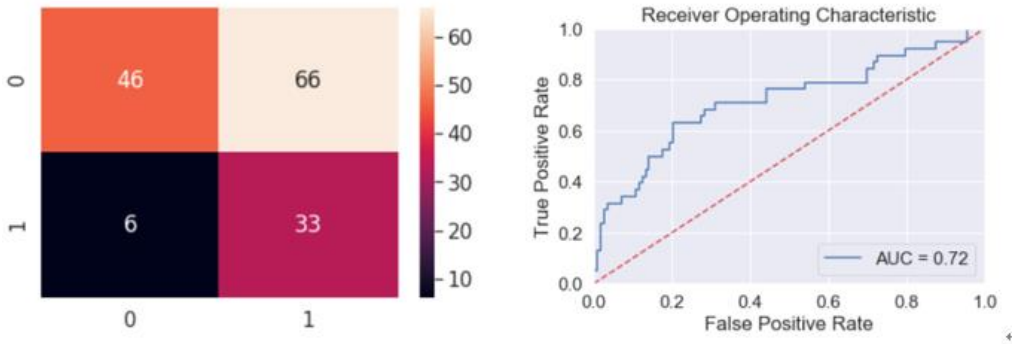


Figure 23. Confusion matrix and ROC curve of Logistic Regression CV

At this time, we got a decent recall rate when using Logistic Regression CV. Then, we began to think how we can make the prediction more understandable for clinicians. We came up with an idea of returning the probability of falling of each patient rather than directly predicting their classes. We used a parameter called predict_proba(X) in Logistic Regression package to return the probability of falling. The example is provided as follow:

cancertype	Health	Grade	Age	FH2	Live	KPS	Living	Driving	Feel	Marital	ImpairedMNA	ImpairedMiniCog	ImpairedIADL	ImpairedGDS	ImpairedCom	Probability
2	2.0	7.0	1	1.0	1.0	3	1.0	1.0	2	4.0	0	0	0	0	1	0.242482
2	5.0	1.0	0	3.0	2.0	2	2.0	0.0	1	5.0	1	0	1	1	1	0.932660
2	3.0	7.0	0	2.0	1.0	3	2.0	1.0	2	2.0	0	0	1	0	0	0.559271
6	4.0	2.0	0	1.0	1.0	2	2.0	1.0	2	6.0	1	0	1	0	0	0.448995
1	5.0	5.0	1	2.0	1.0	3	1.0	1.0	2	2.0	1	1	1	1	0	0.823329

Figure 24. Part of result of predicting probability

Conclusion

In this Capstone project, thanks to the help of URMC team and Prof. Ajay, we successfully reach the goal of this project. Sixu and Zhikang have made a lot contribution to Objective One, which is to build an algorithm to transform trade names to generic names. Junchao and I have contributed to Objective Two, which is to build a predictive model to predict falling. Junchao also did a lot literature review during initial model, which helped us a lot form a hypothesis. For all four of our team, we prepared and practiced presentation together. The high level of cooperation is shown in our project among our team members. It is a great pleasure to have such team in senior to resolve a real-life problem with URMC team together.

For future development, we are planning to build a more user-friendly interface for Objective One. Also, we want to play around with more advanced machine learning algorithm to see whether the performance will be improved.

Reference:

1. URMC-oncology presentation slides on Blackboard. URL: https://learn.rochester.edu/webapps/blackboard/execute/content/file?cmd=view&content_id=_1768437_1&course_id=_38661_1&framesetWrapped=true (Accessed 3/8/2019)
2. URCC Revised Final Appendices. Accessed through Box Drive of University of Rochester (Accessed 3/8/2019)
3. Python Package Difflib. URL: https://docs.python.org/2/library/difflib.html
4. imblearn Package SMOTE. URL: https://imbalanced-learn.readthedocs.io/en/stable/generated/imblearn.over_sampling.SMOTE.html
5. imblearn Package Tomek Link. URL: https://imbalanced-learn.readthedocs.io/en/stable/generated/imblearn.under_sampling.TomekLinks.html
6. imblearn Package Instance Hardness Threshold. URL: https://imbalanced-learn.readthedocs.io/en/stable/generated/imblearn.under_sampling.InstanceHardnessThreshold.html
7. Sklearn Package RFECV. URL: https://scikit-learn.org/stable/modules/generated/sklearn.feature_selection.RFECV.html
8. Skelton, D. A., et al. “Prevention of Falls Network Europe: a Thematic Network Aimed at Introducing Good Practice in Effective Falls Prevention across Europe.” European Journal of Ageing, vol. 1, no. 1, 2004, pp. 89–94.
9. Tinetti, Mary E. “Factors Associated with Serious Injury During Falls by Ambulatory Nursing Home Residents.” Journal of the American Geriatrics Society, vol. 35, no. 7, 1987, pp. 644–648.
10. Lastrucci, Vieri, et al. “Identification of Fall Predictors in the Active Elderly Population from the Routine Medical Records of General Practitioners.” Primary

Health Care Research & Development, vol. 19, no. 02, 2017, pp. 131–139.

11. Kataoka, Hiroshi, and Satoshi Ueno. “Low FAB Score as a Predictor of Future Falling in Patients with Parkinson’s Disease: a 2.5-Year Prospective Study.” Journal of Neurology, vol. 262, no. 9, 2015, pp. 2049–2055.
12. Sharma, Anjali, et al. “Frailty as a Predictor of Falls in HIV-Infected and Uninfected Women.” Antiviral Therapy, 2019.
13. Sklearn Package Logistic Regression CV. URL: https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LogisticRegressionCV.html