Machine Learning Engineer Nanodegree
Capstone Project
Jaewoo Park

September 29th 2017

I. Definition

Project Overview

There has been serious advancements in the usage of artificial intelligence in healthcare. Some go as far as to say that we are in midst of the Fourth Industrial Revolution, and a branch of industry that is heavily impacted is healthcare. With the help of artificial intelligence, healthcare and medicine could improve and optimize patient routes and treatment plans, keep better record of a patient's medical history, actively screen using preventative methodologies and detect anomalies and diseases more accurately than any human could.

To demonstrate this merging of artificial intelligence and healthcare, IBM Watson for oncology has been used in various hospitals around the world to suggest cancer treatment plans. A lot of research is currently ongoing to apply machine learning to cancer detection, and so far there have been some success in these studies. Deep learning has been used to classify skin cancer [1], and various contests and research has been performed on developing kernels for classifying Lung Cancer [2].

Another area that has been a heavy focus for research and development is preventive medicine. There have recently been countless efforts to make diagnosis, prognosis and preventive early detection easier and readily available with the massive patient data that is already sitting in a medical institution's database. This project aims to leverage machine learning to improve both ends of the spectrum (the tertiary care facilities and patients) by using data and machine learning.

The dataset that is investigated consists of medical data from patients with a preexisting Diabetes condition collected over 10 years from 130 hospitals [7], and the machine learning algorithms explored include: random forest ensemble [8], support vector machines [9], and logistic regression.

Problem Statement

Diabetes is a serious problem. Currently, approximately 422 million people in the world have diabetes (type I and type II), and approximately 1.5 million people die due to a form of diabetes every year (an additional 2.2 million deaths were cause by higher-than-optimal levels of blood glucose). Patients, however, with Diabetes can live long and healthy lives if Diabetes is detected and well-managed. However, early diagnosis and intervention is key in living a healthy life with a diabetic condition. Diabetes, if left mismanaged can lead to blindness, amputation and kidney failure [3]. So it is imperative to really keep a good track of diabetes patients from the hospitals' standpoints.

Also with the recent legislature, hospitals are subject to a fine (to be more specific they face a penalty from Medicare) if they discharge a patient and they are readmitted within a 30 day window [4]. So, focusing on retaining the patient if they are likely to be readmitted within the next month is beneficial not just for the patient but also the healthcare institutions because they are able to avoid such hefty fines. This is an area where machine learning can help. If a kernel can be developed to predict the whether or not a patient will be readmitted in the near future with attributes including medical records, this would save patient lives, as well as levy a lesser financial burden on hospitals.

This project looks at the medical records, and demographic patient data collect over several years in several US based hospitals on diabetic patients coming in for treatment. It aims to create a simple machine learning kernel to be able to more accurately predict patient readmission as opposed to a simple guess, and to reproduce or improve upon the result of other previous studies on diabetic readmissions.

The following are the tasks involved:

1. Download the dataset from the UCI Machine Learning repository, and explore the dataset.
2. Interpret the different features, and preprocess the dataset so that they are easier to manipulate and analyze.
3. View the correlation between features, and sort the importance of the features for a feature selection process
4. Performance test on 3 different machine learning algorithms (Logistic regression, Random Forest, Support Vector Machines)
5. Further conduct fine feature selection process
6. Cross validate different machine learning algorithms
7. Fine tune the parameters on the algorithms for the best performance.
8. Compare and reflect to benchmark.

Metrics

The evaluation metric will primarily be the classification accuracy of the test data, which is a common metric for a binary classification problem. Accuracy is calculated by taking both the true negatives and true positives, and dividing by the size of the dataset that is being analyzed.

Accuracy takes into account false negative error, and false positive error. Overall accuracy is a good metric because it addresses an ill optimized status quo.

- False positive (patient is predicted to be readmitted → patient will be kept for further analysis and treatment) means more hospitalization for the patient and use of space and resources that could be used to accommodate other patients.

- For false negatives (patient is falsely predicted to be not readmitted, is released, and is readmitted), it is in both the hospital and patients interest for the patient to not be readmitted. Hence, it is ideal if less patients are predicted not to be readmitted but actually are.

Along the lines of Occam's Razor, and my computational limitations, simplicity of the model – computation time could also be a metric. This will mostly be qualitative, and the main metric will be classification accuracy.

<u>II. Analysis</u> (2-4)

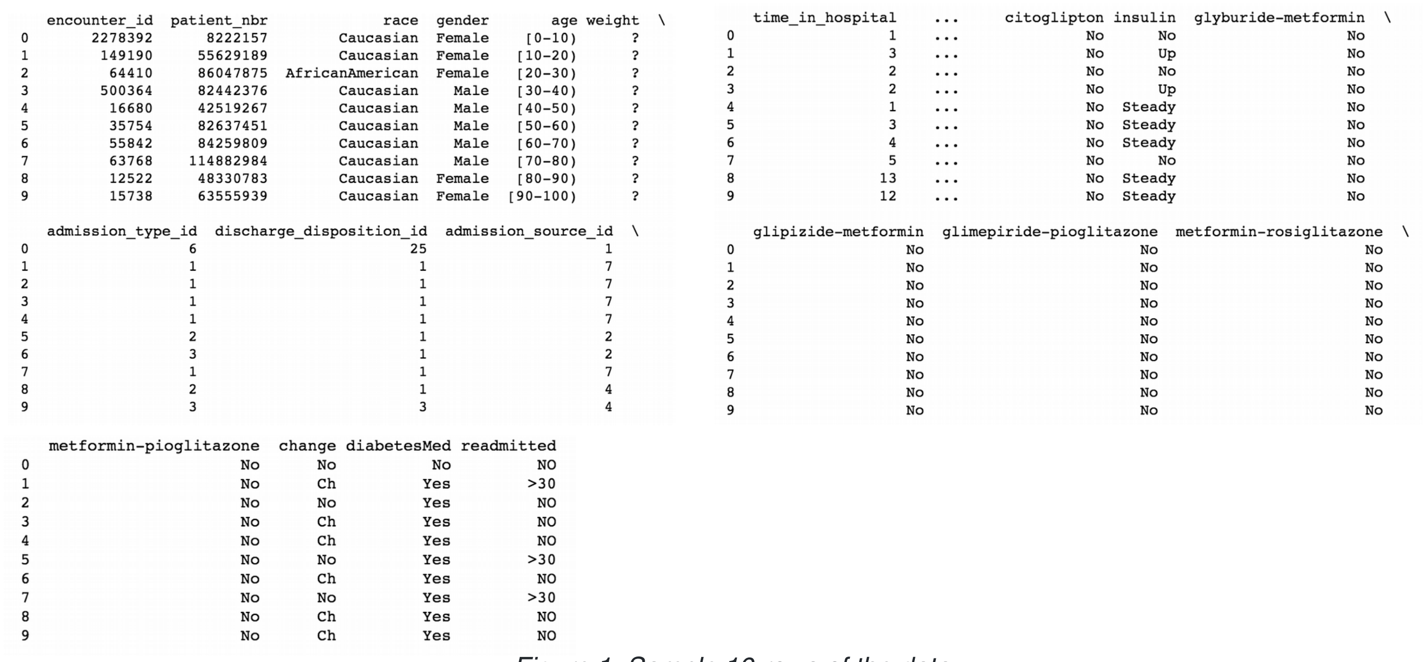
Data Exploration

The dataset I am investigating was collected over the course of 10 years of clinical care at 130 hospitals in the US, consists of patient chart data which include demographic data such as race, gender, and age, medical record data such as number of lab tests performed on the patient, number of procedures, number of medications administered during the visit, diagnosis values including glucose serum test result, A1c test result, and if there had been a change in the prescription of medication during the encounter. Finally, there is data on whether or not the patient was readmitted or not (categorical in ranges <30 days, > 30 days, and no readmission). A full list of features is provided below.

Column interpretation

- UUIDs: (encounter_id, patient_nbr)
- Patient Demographics: (gender, age, weight)
- Electronic Medical Record data - admits, discharges etc : (admission_type_id, discharge_disposition_id, admission_source_id, payer_code, medical_specialty)
- Time spent in hospital between admit and discharge : time_in_hospital
- Encounter related Statistics : (num_lab_procedures, num_procedures, num_medications)
- Patient visits that year Statistics: (number_outpatient, number_emergency, number_inpatient)
- Diagnosis values : (diag_1, diag_2, diag_3, number_diagnoses)
- Test results : (max_glu_serum, A1Cresult)
- Medication delivery : (metformin, repaglinide, nateglinide, chlorpropamide, glimepiride, acetohexamide, glipizide, glyburide, tolbutamide, pioglitazone, rosiglitazone, acarbose, miglitol, troglitazone, tolazamide, examide, citoglipton, insulin, glyburide-metformin, glipizide-metformin, glimepiride pioglitazone, metformin-rosiglitazone, metformin-pioglitazone)
- Medications changed? : (change)
- Diabetes medications prescribed? : (diabetesMed)
- Was patient readmitted? : (readmitted) - Label

And the data (taken the head of the data for the first 10 rows) looks as the following. Along with the descriptive statistics of non-object values in the dataframe.



```
   encounter_id  patient_nbr             race  gender       age weight  \
0       2278392      8222157        Caucasian  Female    [0-10)      ?
1        149190     55629189        Caucasian  Female   [10-20)      ?
2         64410     86047875  AfricanAmerican  Female   [20-30)      ?
3        500364     82442376        Caucasian    Male   [30-40)      ?
4         16680     42519267        Caucasian    Male   [40-50)      ?
5         35754     82637451        Caucasian    Male   [50-60)      ?
6         55842     84259809        Caucasian    Male   [60-70)      ?
7         63768    114882984        Caucasian    Male   [70-80)      ?
8         12522     48330783        Caucasian  Female   [80-90)      ?
9         15738     63555939        Caucasian  Female  [90-100)      ?

   admission_type_id  discharge_disposition_id  admission_source_id  \
0                  6                        25                    1
1                  1                         1                    7
2                  1                         1                    7
3                  1                         1                    7
4                  1                         1                    7
5                  2                         1                    2
6                  3                         1                    2
7                  1                         1                    7
8                  2                         1                    4
9                  3                         3                    4

   time_in_hospital    ...     citoglipton insulin  glyburide-metformin  \
0                 1    ...              No      No                   No
1                 3    ...              No      Up                   No
2                 2    ...              No      No                   No
3                 2    ...              No      Up                   No
4                 1    ...              No  Steady                   No
5                 3    ...              No  Steady                   No
6                 4    ...              No  Steady                   No
7                 5    ...              No      No                   No
8                13    ...              No  Steady                   No
9                12    ...              No  Steady                   No

   glipizide-metformin  glimepiride-pioglitazone  metformin-rosiglitazone  \
0                   No                        No                       No
1                   No                        No                       No
2                   No                        No                       No
3                   No                        No                       No
4                   No                        No                       No
5                   No                        No                       No
6                   No                        No                       No
7                   No                        No                       No
8                   No                        No                       No
9                   No                        No                       No

   metformin-pioglitazone  change diabetesMed readmitted
0                      No      No          No         NO
1                      No      Ch         Yes        >30
2                      No      No         Yes         NO
3                      No      Ch         Yes         NO
4                      No      Ch         Yes         NO
5                      No      No         Yes        >30
6                      No      Ch         Yes         NO
7                      No      No         Yes        >30
8                      No      Ch         Yes         NO
9                      No      Ch         Yes         NO
```

Figure 1: Sample 10 rows of the data

	time_in_hospital	num_lab_procedures	num_procedures	num_medications	number_outpatient	number_emergency	number_inpatient	number_diagnoses
count	101766	101766	101766	101766	101766	101766	101766	101766
mean	4.395987	43.095641	1.33973	16.021844	0.369357	0.197836	0.635566	7.422607
std	2.985108	19.674362	1.705807	8.127566	1.267265	0.930472	1.262863	1.9336
min	1	1	0	1	0	0	0	1
25%	2	31	0	10	0	0	0	6
50%	4	44	1	15	0	0	0	8
75%	6	57	2	20	0	0	1	9
max	14	132	6	81	42	76	21	16

Table 1: Descriptive statistics for non-object type data

These non-object features will be analyzed further by viewing the scatter distribution, but as you can see from these percentiles, most of the data sits below the range (maximum – minimum) of the respective features.

This data is very in line with the problem statement in that it provides plenty of data on the hospital’s interaction with a patient with diabetes and data on if the patient was readmitted or not. So it is quite fitting to answer the question of readmission post treatment for a serious diabetic condition.

Exploratory Visualization

The dataset has 101765 rows of data with 50 columns as features. The data has a roughly even distribution of the label (re-admittance) which sits between 40-50% for 1 (readmitted), and 50-60% for 0 (not readmitted). The figure below illustrates this. This is a good distribution to work with since the model won't be heavily biased for one label.

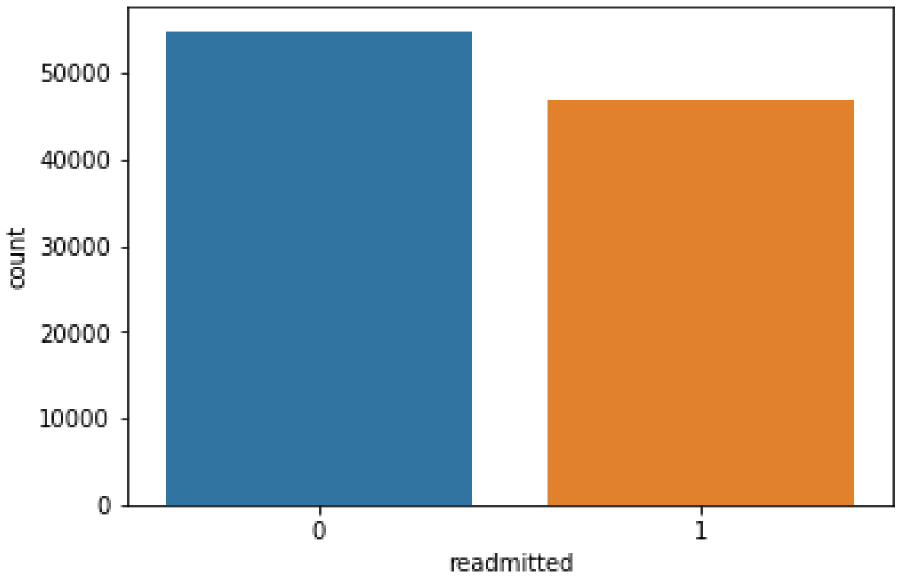


Figure 2: Distribution of the label (patient readmitted)

There also are quite a bit of missing data. The most prominent being weight, which is categorical, however 97% are missing and are coded as '?'. The next being Payer Code, and Medical Specialty which are indicative of who is responsible for paying for the hospital encounter, and the specialty of the admitting physician. (An interesting note is that a Dentist admitted a patient for diabetes in certain cases). Refer below for a list of features with missing data. This is an interesting characteristic of the dataset. Since weight is heavily skewed towards missing, it is likely beneficial to just not include the feature in training the model. However, for payer code and medical specialty, it is likely that it would be better if we kept the feature and labelled the missing as a category of the feature, it may suffice, since the fact that they had not documented this piece of data (although may come from human error) may indicate other features that may indicate that the patient was rushed in, or there was mismanagement etc.

Feature Name	**% Missing**	**Missing Entities**
Race	2%	2,035
Weight	97%	98,712
Payer Code	52%	52,917
Medical Specialty	53%	53,935
Diagnosis 3	1%	1,017

Table 2: Features with missing data and its respective percentage and numbers

Algorithms and Techniques

This is a binary classification problem with ~50 features. Hence it is predicted that logistic regression would be very robust in handling a problem such as this. However, it is worthwhile to find out feature importance, and to compare to different algorithms to visualize whatever difference in the metric the outcome will have. Hence this project will also use random forest and support vector machine algorithms to compare the robustness as compared to logistic regression.

Also, cross validation will be performed to make the modeling process more stable, and to prevent under-fitting or over-fitting, and parameter tuning will be done to best optimize the model.

The algorithms and techniques that will be used to perform this analysis are as follows:

- Logistic Regression
  - Logistic regression is a linear model for classification which uses a linear function to predict a binary solution {0, 1}. Since the same methods for linear regression is not effective at classifying a binary solution, a sigmoid function is used to “squash” the values of weight*value to either 0 or 1. The weights are calculated by training examples under a cost function. Argmax of the cost function becomes the weight [11].
  - Chosen because of simplicity, quick run time, and the binary classification nature of this dataset. It also handles noise pretty well.
- Random Forest [8]
  - Random Forest is an ensemble learning method using multiple decision trees. It randomly selects n features from the total features, and calculates the node using the best split point. It splits the node into daughter nodes using the best split, and this is done until the designated number of nodes is reached. Then this is repeated for the number of trees designated [13]. How it “combines” these trees is, it takes the test input and uses the rules of each randomly created tree to predict the outcome. It then calculates the votes of the each predicted target, and considers the highest voted the final prediction [12].
  - Chosen because of its ability to handle high dimensionality really well, and its relative simplicity, and the fact that it runs really well out of the box.
- Support Vector Machines [9]
  - Support vector machine is a discriminative classifier that employs separation by hyperplanes to classify labels. The hyper plane (conceptualized in the diagram below) is drawn by finding a plane that gives the largest minimum distance to the training examples.
  - Chosen since this is a classification problem and also to account for the case of potential non-linear separability, and usability for high dimensions

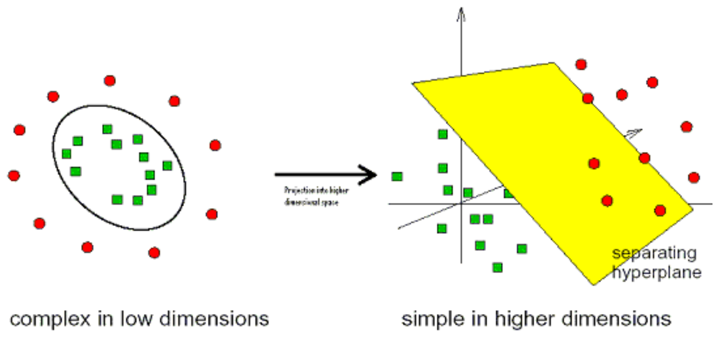

Projection into higher dimensional space
separating hyperplane
complex in low dimensions
simple in higher dimensions

Figure 3: Hyperplane divides the labels for support vector machine [10]

- Feature Selection
    - Feature importance
    - Feature correlation analysis
    - Feature classification distribution
- Cross validation
- GridSearch for parameter Tuning
    - Random Forest (parameters that will be tuned)
        - max_features: maximum number of features Random Forest is allowed to try in each individual tree
        - n_estimators: number of trees you want to build initially
        - min_sample_leaf: minimum size for the end node of the tree
        - max_depth: maximum depth of the tree (optimize)
    - Logistic Regression (parameters that will be tuned)
        - C: Inverse of regularization strength
        - penalty: the norm used for penalty

Benchmark

The benchmark for this project will be 2 things. The first is an absolute benchmark which is a simple coin toss guess, whereby 50% of the guesses will be correct (Accuracy of 50%). The predictions from the project must be over this benchmark. The other will be a comparative bench mark which compares to similar studies. The range is approximately 63% – 76% [5], and hence the result will be compared to this range to validate a successful reproduction or improvement.

III. Methodology (3-5)

Data Preprocessing

The preprocessing step is the lengthiest step that required most manipulation. As you can see below, the features consist of integer values, and object values. The object values contain string data which are in actuality floats (only a few decimal points), missing (string value of '?'), or a code used by that specific healthcare facility. Hence a lot of the data had to be brushed in order for it to be of use.

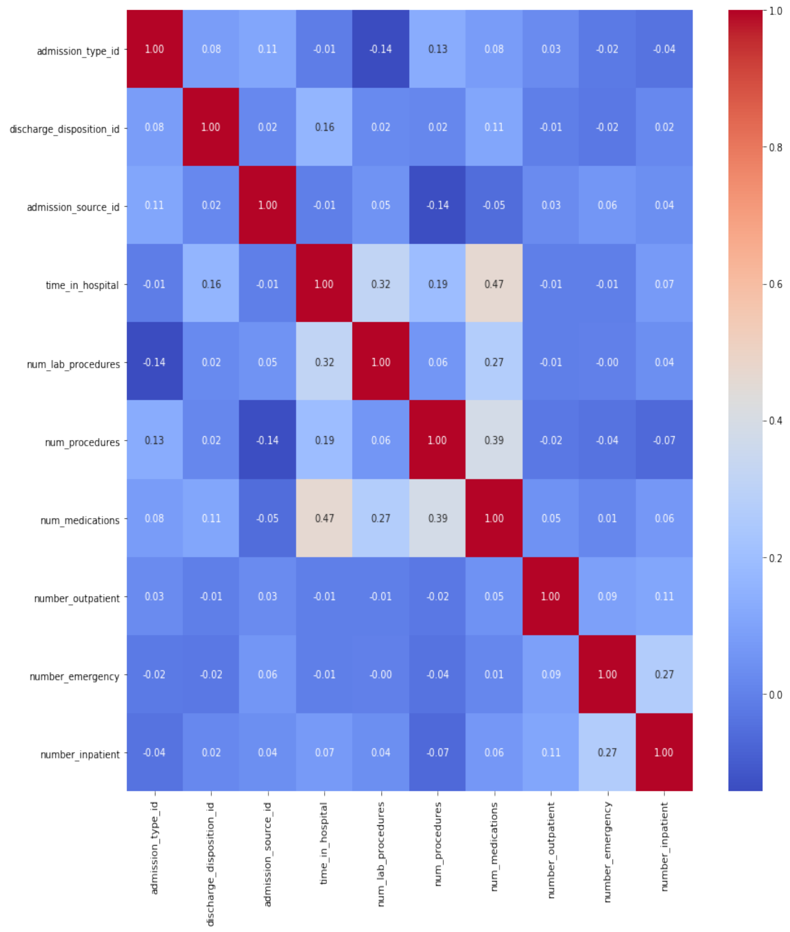
```
encounter_id                101766 non-null int64
patient_nbr                 101766 non-null int64
race                        101766 non-null object
gender                      101766 non-null object
age                         101766 non-null object
weight                      101766 non-null object
admission_type_id           101766 non-null int64
discharge_disposition_id    101766 non-null int64
admission_source_id         101766 non-null int64
time_in_hospital            101766 non-null int64
payer_code                  101766 non-null object
medical_specialty           101766 non-null object
num_lab_procedures          101766 non-null int64
num_procedures              101766 non-null int64
num_medications             101766 non-null int64
number_outpatient           101766 non-null int64
number_emergency            101766 non-null int64
number_inpatient            101766 non-null int64
diag_1                      101766 non-null object
diag_2                      101766 non-null object
diag_3                      101766 non-null object
number_diagnoses            101766 non-null int64
max_glu_serum               101766 non-null object
A1Cresult                   101766 non-null object
metformin                   101766 non-null object
repaglinide                 101766 non-null object
nateglinide                 101766 non-null object
chlorpropamide              101766 non-null object
glimepiride                 101766 non-null object
acetohexamide               101766 non-null object
glipizide                   101766 non-null object
glyburide                   101766 non-null object
tolbutamide                 101766 non-null object
pioglitazone                101766 non-null object
rosiglitazone               101766 non-null object
acarbose                    101766 non-null object
miglitol                    101766 non-null object
troglitazone                101766 non-null object
tolazamide                  101766 non-null object
examide                     101766 non-null object
citoglipton                 101766 non-null object
insulin                     101766 non-null object
glyburide-metformin         101766 non-null object
glipizide-metformin         101766 non-null object
glimepiride-pioglitazone    101766 non-null object
metformin-rosiglitazone     101766 non-null object
metformin-pioglitazone      101766 non-null object
change                      101766 non-null object
diabetesMed                 101766 non-null object
readmitted                  101766 non-null object
```

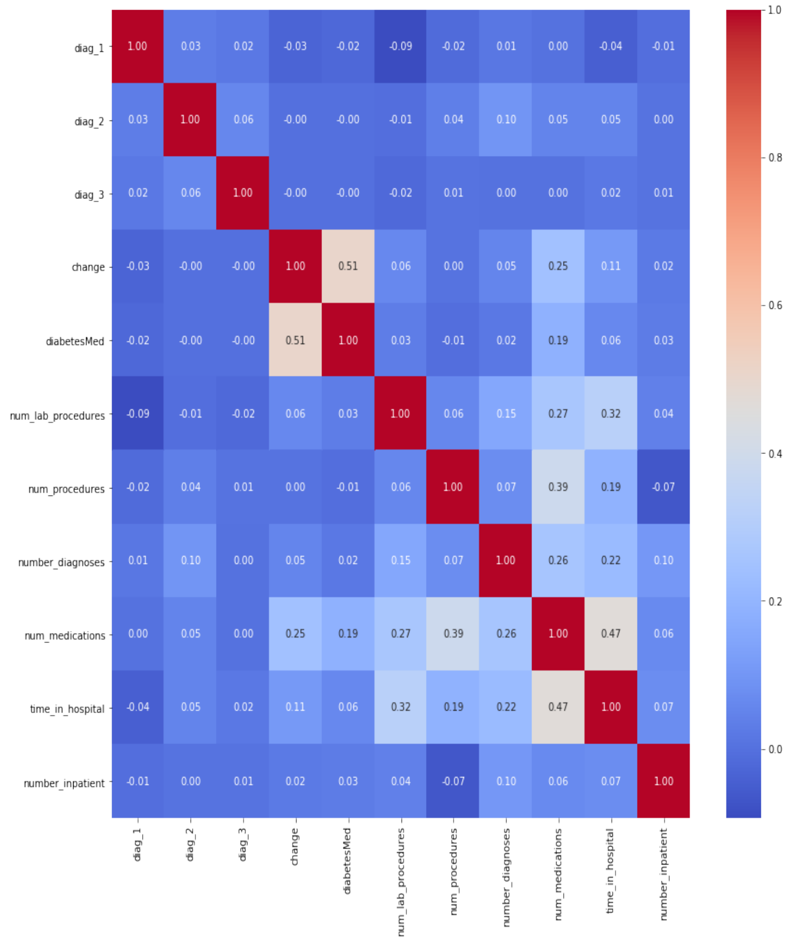
Figure 4: Dataframe information

To further breakdown the steps taken:

1. Delete features (in this case only weight) with mostly missing database
2. Remove irrelevant ids from the dataframe
3. Find the unique values of object based series in the dataframe
4. Map the data with a categorical value
5. Separate out the label, and map the results to 0 – no readmission and 1 – readmission.
6. Visualize correlation between different features of interest
7. Split train test data
8. Figure out the order of importance for the features
9. Initial trimming of features
10. Visualize segmentation and trends between certain features of interesting
11. Finalize feature selection

Figure 5: Heatmap of different features





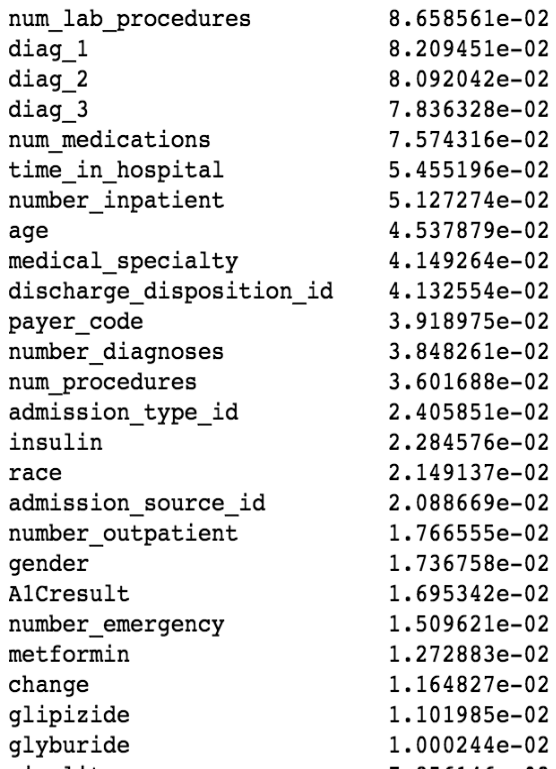
A heatmap was created to see if certain features were related to one another. Since if there are features that are heavily correlated to one another, one of them can be used rather than all the ones strongly correlated without impacting information loss a lot.

However, it was shown that this was not the case for the features we were dealing with. Although the features num_medication and time_in_hospital showed a certain degree of correlation at around 0.47, this was not as strong for them to be squashed together.

You could deduce that if a patient stays in the hospital for a longer period of time, he or she will be administered more medication – which seems logical. Another rather correlated set of features was diabetes_med and change. And this indicated that it was likely that if there was a change in a patient's medication schedule for a diabetes patient, it was likely due to a case where the patient had to be administered diabetes medication.

Since that was the case, none of the features were eliminated, and by this point, the only eliminated features were irrelevant id, and weight. Next step of figuring out which feature was more important aided much better in selecting features.

Next step was feature selection, and the result of ascending sort followed by a preliminary train test split (70% training, 30% test split) and random forest was as follows.



```
num_lab_procedures          8.658561e-02
diag_1                      8.209451e-02
diag_2                      8.092042e-02
diag_3                      7.836328e-02
num_medications             7.574316e-02
time_in_hospital            5.455196e-02
number_inpatient            5.127274e-02
age                         4.537879e-02
medical_specialty           4.149264e-02
discharge_disposition_id    4.132554e-02
payer_code                  3.918975e-02
number_diagnoses            3.848261e-02
num_procedures              3.601688e-02
admission_type_id           2.405851e-02
insulin                     2.284576e-02
race                        2.149137e-02
admission_source_id         2.088669e-02
number_outpatient           1.766555e-02
gender                      1.736758e-02
A1Cresult                   1.695342e-02
number_emergency            1.509621e-02
metformin                   1.272883e-02
change                      1.164827e-02
glipizide                   1.101985e-02
glyburide                   1.000244e-02
```

Figure 6: Sorted features in ascending order

Only features above a 1E-02 importance was retained, and the remainder was dropped. This had minimal impact on the performance of the different models and its accuracy.

Implementation

The most difficult and time consuming part of the entire implementation process was the data processing step. It required digging for every single label within every feature that had object values, and mapping them to a categorical fashion. After this was done, the most difficult process was speeding up the analysis runtime.

After the features were chunked down from the importance sort, it was decided to drop SVM as one of the algorithms to use due to the complexity and high run time to train the model based on Occam's Razor. It also was showing a low accuracy near the floor benchmark (53%) due to the complexity of the number of features, hence ultimately it was decided that it would be dropped.

At this point, logistic regression and random forests were showing similar accuracy at around 62 – 63% accuracy. This was done without cross validation or parameter tuning. C of 1 was used for logistic regression, and the default values for random forest set by sk-learn was used. Before moving on to cross validation, and parameter tuning, a scatter distribution plot was created to see if there were any potential features that had less meaning to the over all model.

The result was as follows:

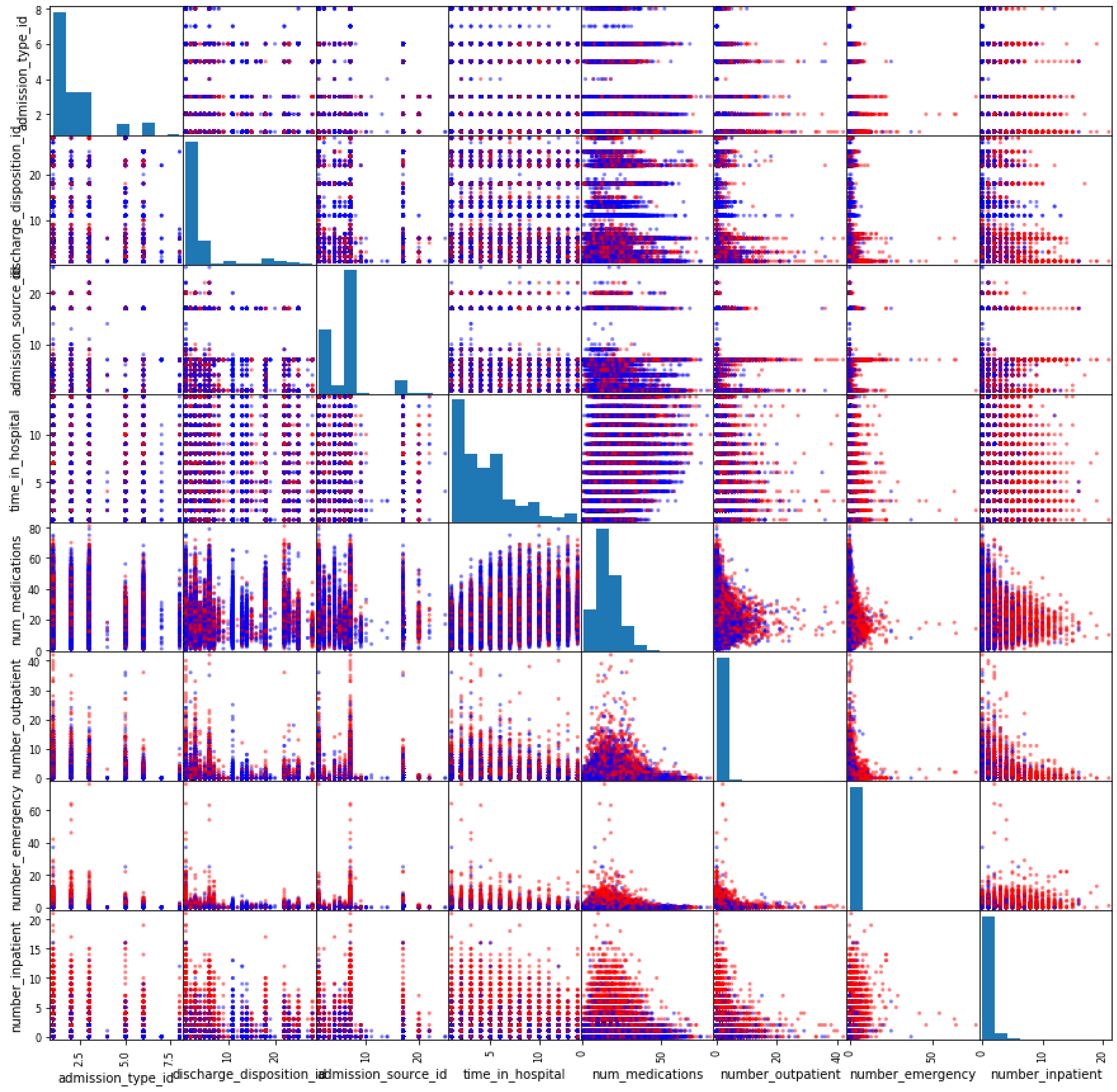


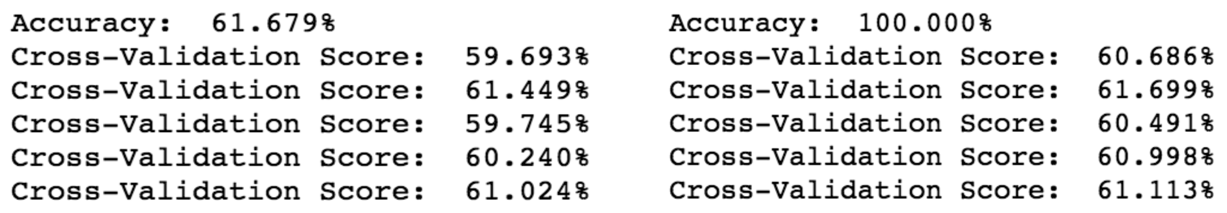
Figure 7: Scatter plot of classification distribution among different pairs of features (blue dots – not readmitted, red dots – readmitted)

This scatter plot showed a lot of interesting results, which intuitively made a lot of sense. For example, as number_inpatient (number of prior inpatient visits) increase – meaning the patient has a chronic record, regardless of num_medications (how much medication was administered during the encounter) the result was predominantly red (patient will be readmitted).

One other interesting note was admission_type_id, discharge_disposition, and admission_source_id did not contribute to classifying two categories apart since it showed no separating plane in the scatter. Hence, it was decided that these features could be dropped.

Following the scatter visualization, cross validation with 5 folds was conducted to see if the model was stable. I did an analysis where I trained the model with training data, and tried to reverse the training process by using the training data as test data. If the accuracy was a

100%, this would mean that the model may be prone to overfitting. This was done prior to conducting the cross validation to cross reference the performance of the different machine learning algorithms. The results showed a lot of consistency for logistic regression with an accuracy of 61% and the cross validation scores all being in the range with +-1, but showed clear signs of overfitting with an accuracy of 100% and cross validation scores around 60-61 for all 5 for the random forest model which is usually the case and was somewhat expected [6]. See below for the actual results for logistic regression and random forest.



```
Accuracy:  61.679%                      Accuracy:  100.000%
Cross-Validation Score:  59.693%        Cross-Validation Score:  60.686%
Cross-Validation Score:  61.449%        Cross-Validation Score:  61.699%
Cross-Validation Score:  59.745%        Cross-Validation Score:  60.491%
Cross-Validation Score:  60.240%        Cross-Validation Score:  60.998%
Cross-Validation Score:  61.024%        Cross-Validation Score:  61.113%
```

Figure 8: Results of overfitting testing and cross-validation.

Refinement

Final step of the implementation process was refinement, which involved fine tuning the parameters with GridsearchCV. The following were the grid parameters input for random forest and logistic regression.

Random Forest:

'max_depth' : [None, 3],
'max_features' : ['sqrt', 'log2', None],
'n_estimators' :[10, 50, 100, 150],
'min_samples_leaf' : [1, 2, 3, 5, 10]

Logistic Regression:

'C': [0.001, 0.01, 0.1, 1, 10, 100, 1000]
'penalty': ['l1', 'l2']

The result was the following:

Random Forest	
max_depth	None
max_features	‘sqrt’ or ‘auto’
n_estimators	150
min_samples_leaf	10
Logistic Regression	
C	0.01
penalty	‘l1’

Table 3: Tuned parameters that was used in the final model

The final model was the random forest model with the parameters listed above. It had an accuracy of 62.3% with the given tuned parameters. This was an improvement over both the 60.99% accuracy that the random forest algorithm had before fine tuning, and the 60.43% accuracy logistic regression had.

IV. Results (2-3)

Model Evaluation and Validation

The final model used the random forest ensemble algorithm which outperformed SVM and logistic regression. Cross validation was done during implementation to evaluate the model. This model consisted of feature selection and algorithm selection, and is a result of the following:

- Drop mostly missing features and irrelevant ids
- Filter out the most important features above a threshold of 1E-02
- Reducing features such as admission_type_id, discharge_disposition, and admission_source_id post scatter distribution.
- Cross validation check
- Fine tuning parameters

In discussing the results, we need to discuss the robustness of the model. Robustness is defined as a property that characterizes how effective the algorithm is while being tested on the new independent but similar dataset. So if the training error is close to the test error, it can be said that the algorithm is robust.

Before fine tuning the parameters, training accuracy was preliminarily calculated, and this showed that SVM and RandomForest, without any parameters changed from default, was very high close to 0% error. While logistic regression showed a training error that was approximately the same as the testing error.

After cherry picking features, and tuning parameters, the training error was calculated again, and it was seen that the training error for random forest was approximately 0.22 that was lower than the test error for random forest, while logistic regression showed approximately 0.48 which was very close to the testing error for logistic regression. However, it was decided that using random forest would not be too big of a problem because of the nature of hospital medical data. The format rarely changes, and all that would be different from a different dataset would be missing data. Hence random forest was chosen as the final model due to its marginally better performance over logistic regression.

The final model had an accuracy of 62.3% which is marginally better than using the logistic regression algorithm.

Justification

The initial benchmarks that were originally established consisted of a floor and a comparator. For the first half, it was compared to a 50/50 guess, which the model should have absolutely outperformed.

First Benchmark	Accuracy	Result
50%	62.3%	Accuracy > First Benchmark

The second half of the benchmark was results from similar studies which roughly lay in the 60%-70% range. This model was able to perform approximately within this range, and hence this model is comparable to other previous studies.

Second Benchmark	Accuracy	Result
63%-76%	62.3%	Accuracy ~= Second Benchmark

In summary, this model produced results much better than a simple guess, was comparable to results from comprehensive studies, and was simple with a rather fast runtime.

V. Conclusion (1-2)

Free-Form Visualization

For this, I think it will be best suited to reused figure 7 and compare some of the logical hypotheses I had in my head with the actual distribution of the labels when two features were compared side by side.

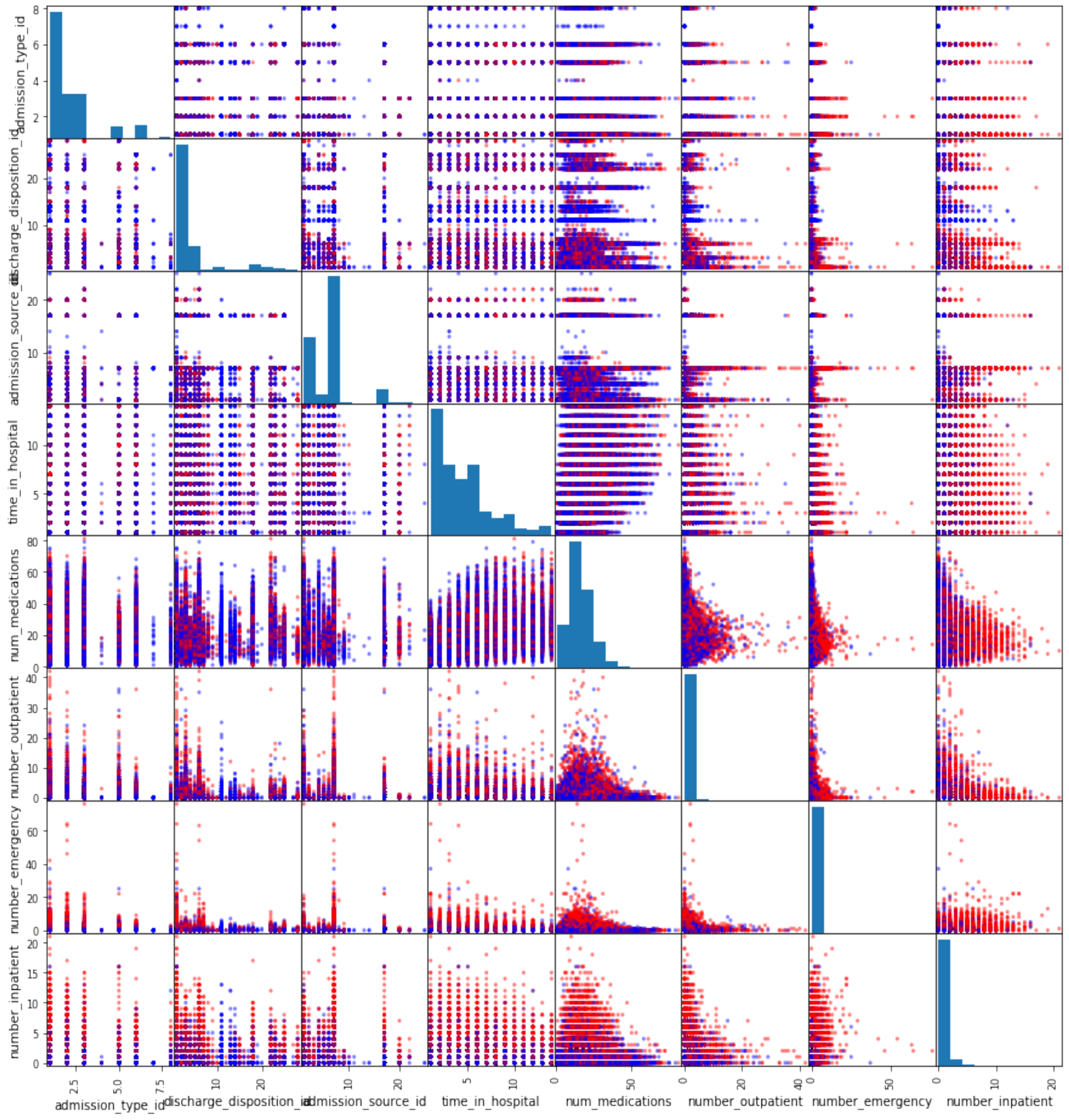


Figure 7: Scatter plot of classification distribution among different pairs of features (blue dots – not readmitted, red dots – readmitted)

Initially, I conjectured that if a patient’s medical history was extensive so that they have previously had a lot of outpatient or inpatient visits, they would be more likely to be readmitted to the hospital within a short time frame. Another conjecture I made was that it would not matter how long of a treatment a patient would receive during their one encounter at the hospital.

Comparing these notions with the scatter distribution, we can look at the time_in_hospital row, and we can observe that the red dots are very scattered throughout the entire row, meaning that segmenting where the red dots would lie when compared to other features was not possible. They are spread out evenly. This is along the same line as one of my conjectures. The other point is also demonstrated if we look at the number_inpatient feature. The general trend is clearly visible in that most blue points are towards the lower end of the spectrum, while most of the red points are shown toward the higher end of the spectrum.

However, we do need to keep in mind that most of these features are heavily skewed. If we look at the distribution on the diagonal of the charts, we can see that most data points lie toward the lower end of the spectrum, and hence there may be a lot of overlap between the blue points and the red points in the lower end of the spectrum where a lot of points actually reside. We can, however, say that if a patient has a high number of previous inpatient visits, then they are likely to be readmitted because the likelihood of a red dot is higher than a blue dot.

Reflection

I wanted to tackle a problem in healthcare, and see if a data driven approach could supplement a physician or a healthcare professional’s opinions on patient diagnostics and treatment. The problem solving approach for this dataset was as follows.

1. I found a patient medical record related data that looked interesting
2. Started performing EDA on the dataset and found a column of the dataset that was 30 day re-admittance.
3. Given prior knowledge that tertiary care hospitals are fined when a patient is discharged and is readmitted back within a 30 day window, receive a financial burden from medicare.
4. Looked up prior research to find out the appropriate benchmark I could use to evaluate whether or not the conclusion I arrive at would be appropriate.
5. Features were evaluated and further trimmed down.
6. Trained classifiers based on different machine learning algorithms.
7. Evaluated the robustness and completed the model using random forest ensemble.

I found most of the problem solving approach straightforward, but found that missing data or ambiguous data was prevalent. This is likely worse in real life data, that is not closely administered or managed. The missing data was considered a category and was assumed that the hospitals had difficulty recording that data because of reasons outside of human error. However, this may not be the case, and may be due to human error or the mis-synchronization of EMR (electronic medical record) throughout the hospital network.

The conclusion I arrived at with the accuracy of roughly 63% suit the description of other previous studies that aimed to tackle this problem. However, this may be prone to more error and difficulties if there are more noise due to the lack of data or inconsistencies in the different features per patient or encounter.

Improvement

There were plenty of room for improvements. For one thing, I lacked the computational resource to run scripts that would take a prolonged amount of time, and hence could not perform a more in depth SVM, or a higher cv for gridsearchCV. This could have been resolved if I used cloud computing resources or such that I was not comfortable using at the time of this analysis.

Another point I could improve on is to research what the different avenues are for handling missing data. The only solutions I could research and think of were to fill them with nominal data, or to treat the missing data as a separate category on its own.

Also given more knowledge and resources, I would have further evaluated the different parameters for different machine learning algorithms and their applications to different datasets to start off with a more wholesome hypothesis on which algorithm would perform best for this type of dataset. I found that the most interesting aspect of this project was seeing what intuition was in line or contradictory of the actual trends, as exemplified in free-form visualization above. I would have liked to hypothesize which parameters of the random forest and logistic regression would have had the greatest impact before fine tuning.

There are other research that have concluded with a higher accuracy, but different studies use different datasets with features I did not have access to since a lot of medical data is confidential due to privacy reasons.

References

[1] https://www.nature.com/articles/nature21056.epdf?referrer_access_token=shuy6MM06sXL_Up3K_gZSNRgN0jAjWel9jnR3ZoTv0NXpMHRAJy8Qn10ys2O4tuPQzN7VitPjMSrm-_eh79EcBa7E8gqt-pkDzYFswnz9xZ5-KXIIz-q1vIeEhxcYyFWeH_8pHiygm9DWyu_08hysBauLS8xsLhzVFYyEQLpF_WX-ahz18wIB3jPkFPChRFtTi9ijo_NvI_2omWmkk3kqEQksepx_FTvN9qjvf1eJLbOCa1YFcF3VQMzgK_045RJI9DcyV26TC53xZgESNhDjDF0lYBgzJemLOn6qKw1Xpc%3D&tracking_referrer=www.technologyreview.com

[2] https://www.kaggle.com/c/data-science-bowl-2017

[3] http://www.who.int/features/factfiles/diabetes/en/

[4] https://www.cms.gov/Medicare/Medicare-Fee-for-Service-Payment/AcuteInpatientPPS/FY2016-IPPS-Final-Rule-Home-Page-Items/FY2016-IPPS-Final-Rule-Tables.html

[5] http://www.sciencedirect.com/science/article/pii/S15320464150009

[6] http://www.sciencedirect.com/science/article/pii/S2213158214001326

[7] https://archive.ics.uci.edu/ml/datasets/diabetes+130-us+hospitals+for+years+1999-2008

[8] https://link.springer.com/article/10.1023/A:1010933404324

[9] https://link.springer.com/article/10.1007%2FBF00994018

[10] http://www.improvedoutcomes.com/docs/WebSiteDocs/Introduction/Tutorials/Tutorial_9_Support_Vector_Machines/Tutorial_9__Introduction.htm

[11] Machine Learning – A Probabilistic Perspective. Kevin P. Murphy

[12] http://dataaspirant.com/2017/05/22/random-forest-algorithm-machine-learing/

[13] http://ufldl.stanford.edu/tutorial/supervised/LogisticRegression/

Refer to capstone.ipynb for the code.