Capstone Two - Final Project Report

Diabetes Readmission Prediction

1. Introduction

Problem statement:

- What are the factors leading to high readmission of diabetes patients' to hospital within 30 days of discharge?
- Can we predict if a patient will get readmitted or not based on their medical record?

Background:

Diabetes is a chronic health disease caused due to excess glucose in the blood and failure of our pancreas to produce insulin properly. This leads to several health complications to patients such as cardiovascular risk, frequent infections, blindness (retinopathy) etc. The data from 130 US hospitals from 1999-2008 shows there is a high percentage of diabetes patients getting readmitted within 30 days of their last discharge. This poses the question what factors are responsible that makes patients' getting readmitted within a short time. Finding these factors will help in improving the quality of care in diabetes patients', lowering the medical cost involved and identifying medicines effective in treating diabetes.

Project goals:

- Identifying one or more attributes that caused diabetes patients' getting readmitted
- Understanding the relationship between different attributes
- Key medicines used in treatment that helped patients' not getting readmitted

2. Dataset

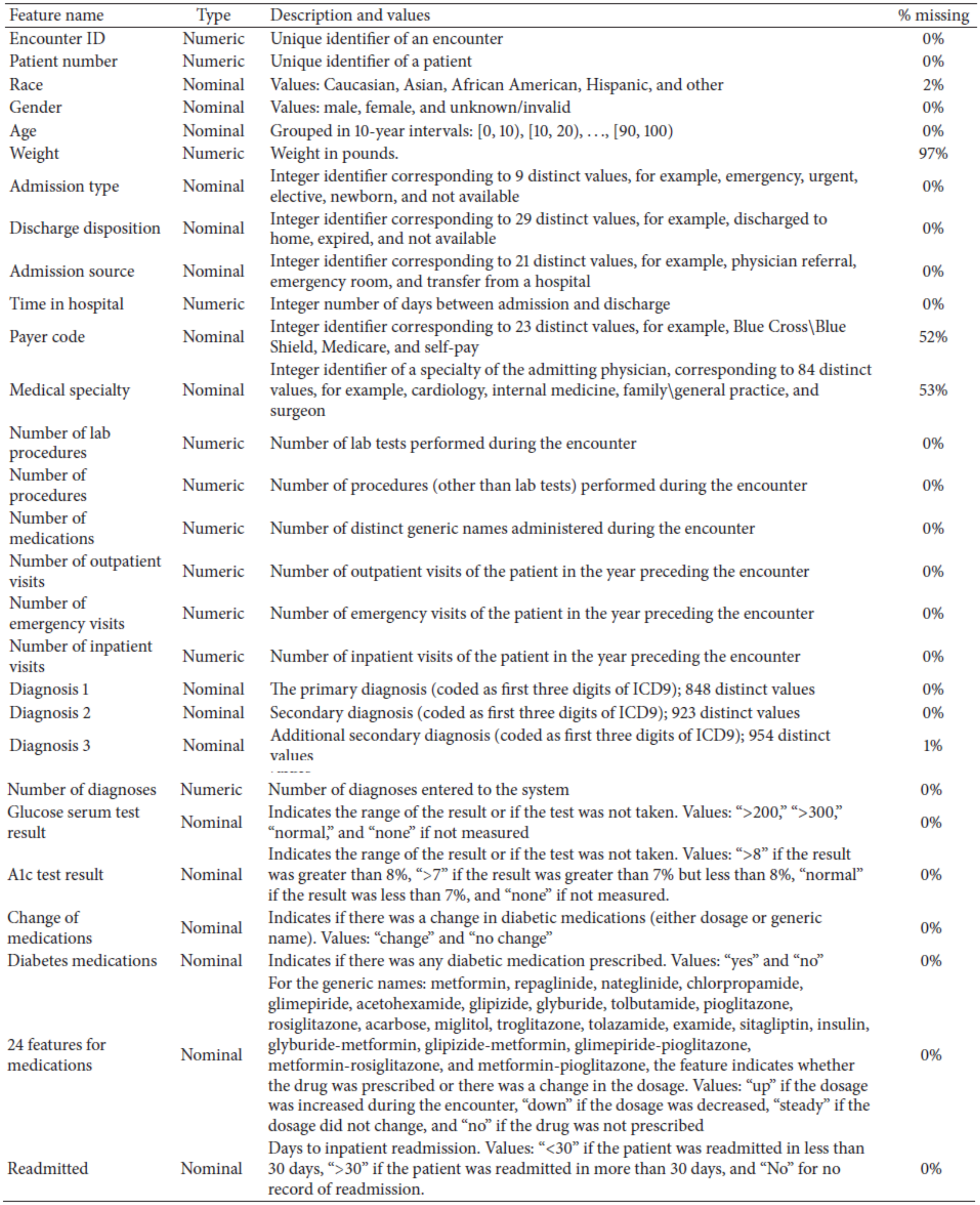
Data originally deposited in UCI Machine Learning repository. This data derived from the published article named 'The relationship between diabetes mellitus and 30-day readmission rates' from Clinical Diabetes and Endocrinology journal.

- **Dataset Name:** Diabetes 130-US hospitals for years 1999-2008 Data Set
- **Source:** https://www.kaggle.com/saurabhtayal/diabetic-patients-readmission-prediction

This dataset consists of three different classes of patients and their records have been stored.

- Class 1: No – Patients not readmitted within 30 days
- Class 2: >30 – Patients readmitted within 30 days
- Class 3: <30 – Patients readmitted within 30 days **(our target variable class)**

Feature Information:



Feature name	Type	Description and values	% missing
Encounter ID	Numeric	Unique identifier of an encounter	0%
Patient number	Numeric	Unique identifier of a patient	0%
Race	Nominal	Values: Caucasian, Asian, African American, Hispanic, and other	2%
Gender	Nominal	Values: male, female, and unknown/invalid	0%
Age	Nominal	Grouped in 10-year intervals: [0, 10), [10, 20), . . ., [90, 100)	0%
Weight	Numeric	Weight in pounds.	97%
Admission type	Nominal	Integer identifier corresponding to 9 distinct values, for example, emergency, urgent, elective, newborn, and not available	0%
Discharge disposition	Nominal	Integer identifier corresponding to 29 distinct values, for example, discharged to home, expired, and not available	0%
Admission source	Nominal	Integer identifier corresponding to 21 distinct values, for example, physician referral, emergency room, and transfer from a hospital	0%
Time in hospital	Numeric	Integer number of days between admission and discharge	0%
Payer code	Nominal	Integer identifier corresponding to 23 distinct values, for example, Blue Cross\Blue Shield, Medicare, and self-pay	52%
Medical specialty	Nominal	Integer identifier of a specialty of the admitting physician, corresponding to 84 distinct values, for example, cardiology, internal medicine, family\general practice, and surgeon	53%
Number of lab procedures	Numeric	Number of lab tests performed during the encounter	0%
Number of procedures	Numeric	Number of procedures (other than lab tests) performed during the encounter	0%
Number of medications	Numeric	Number of distinct generic names administered during the encounter	0%
Number of outpatient visits	Numeric	Number of outpatient visits of the patient in the year preceding the encounter	0%
Number of emergency visits	Numeric	Number of emergency visits of the patient in the year preceding the encounter	0%
Number of inpatient visits	Numeric	Number of inpatient visits of the patient in the year preceding the encounter	0%
Diagnosis 1	Nominal	The primary diagnosis (coded as first three digits of ICD9); 848 distinct values	0%
Diagnosis 2	Nominal	Secondary diagnosis (coded as first three digits of ICD9); 923 distinct values	0%
Diagnosis 3	Nominal	Additional secondary diagnosis (coded as first three digits of ICD9); 954 distinct values	1%
Number of diagnoses	Numeric	Number of diagnoses entered to the system	0%
Glucose serum test result	Nominal	Indicates the range of the result or if the test was not taken. Values: ">200," ">300," "normal," and "none" if not measured	0%
A1c test result	Nominal	Indicates the range of the result or if the test was not taken. Values: ">8" if the result was greater than 8%, ">7" if the result was greater than 7% but less than 8%, "normal" if the result was less than 7%, and "none" if not measured.	0%
Change of medications	Nominal	Indicates if there was a change in diabetic medications (either dosage or generic name). Values: "change" and "no change"	0%
Diabetes medications	Nominal	Indicates if there was any diabetic medication prescribed. Values: "yes" and "no"	0%
24 features for medications	Nominal	For the generic names: metformin, repaglinide, nateglinide, chlorpropamide, glimepiride, acetohexamide, glipizide, glyburide, tolbutamide, pioglitazone, rosiglitazone, acarbose, miglitol, troglitazone, tolazamide, examide, sitagliptin, insulin, glyburide-metformin, glipizide-metformin, glimepiride-pioglitazone, metformin-rosiglitazone, and metformin-pioglitazone, the feature indicates whether the drug was prescribed or there was a change in the dosage. Values: "up" if the dosage was increased during the encounter, "down" if the dosage was decreased, "steady" if the dosage did not change, and "no" if the drug was not prescribed	0%
Readmitted	Nominal	Days to inpatient readmission. Values: "<30" if the patient was readmitted in less than 30 days, ">30" if the patient was readmitted in more than 30 days, and "No" for no record of readmission.	0%

Table 1: Feature information about all variables listed in the dataset

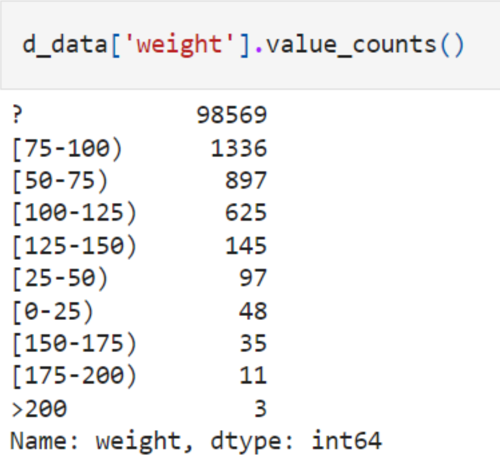
Number of instances	101767
Number of total variables	50
Number of independent variables	49
Number of dependent variables	1
Number of numerical variables	17
Number of categorical variables	33

Table 2: Distribution of different type of variables and their counts

3. Data Wrangling

This dataset has total 101767 instances with 50 different variables consisting of both numerical and categorical variables. Broadly dataset explains each patient and their hospital encounter with ID's, number of diagnoses, number of lab tests performed, time spent at hospital, glucose and A1C results. List of medications taken by each patient and dose prescribed were given. Few demographic information also included such as race, gender, age and weight of each patient. The target variable is denoted as 'readmitted' with three classes.

Initial exploration found no missing values in the dataset but in actual some columns had missing entries. Specifically, the 'weight' column contained 97% of values missing and all the entries were mentioned as '?' values. As more than 90% of the values were missing, it is of no use, so we dropped this column from the dataset.



```
d_data['weight'].value_counts()
```

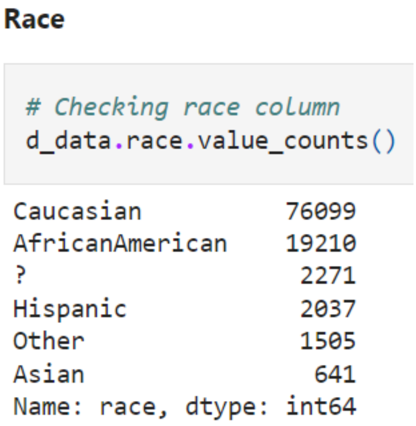
```
?            98569
[75-100)      1336
[50-75)        897
[100-125)      625
[125-150)      145
[25-50)         97
[0-25)          48
[150-175)       35
[175-200)       11
>200             3
Name: weight, dtype: int64
```

Followed by, we investigated other columns to find the counts of different entries. 'Age' column provided 10 different age groups; most population were above 40 years old. Later we checked on the 'gender' column distribution. It showed 54% of female entries, 46% of male entries and 3 entries as unknown/invalid. We removed these three entries from dataset as it is not clear what gender these entries.

The variable names ‘payer code’, ‘medical specialty’ had almost 40% instances were missing and these two variables does not add reliable information for the prediction, so we dropped these two columns as well. The diagnostics information of patients had three different types of columns – ‘diag1’, ‘diag2’, ‘diag3’.

Each denoted primary, secondary and additional diagnostic tests performed on the patients coded with distinct values (Refer table 1 for more information). Each variable had 800-900 distinct codes which makes the info very sparse and difficult to collate these columns for analysis. Hence, we did not include these columns as well.

We looked at the demographics by ‘race’ column and found 2271 entries were denoted as ‘?’. These were either missing or not classified. But the race column denoted some entries as ‘Other’, so converted these entries into other category. See below,



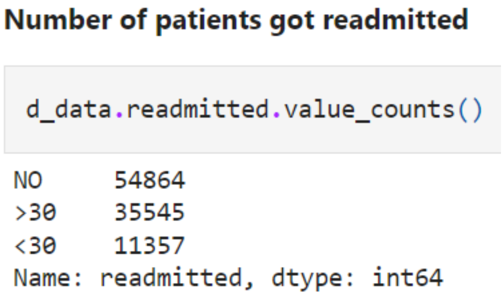
Race

```
# Checking race column
d_data.race.value_counts()
```

```
Caucasian          76099
AfricanAmerican    19210
?                   2271
Hispanic            2037
Other               1505
Asian                641
Name: race, dtype: int64
```

Target variable

The target variable column – readmitted had three different classes and did not contain any missing values. But we found class imbalance between three classes, where the <30 class had least rows compared to other two classes. See below,



Number of patients got readmitted

```
d_data.readmitted.value_counts()
```

```
NO     54864
>30    35545
<30    11357
Name: readmitted, dtype: int64
```

We also found some columns needed datatype conversion from numeric to object and vice versa. Remaining columns were included for the study and taken further for exploratory data analysis.

3. Exploratory Data Analysis

Time in hospital

We initially explored the ‘time in hospital’ column which gives information about how many days patients stayed at hospital. Below bar plot displays the number of patients and days they spent at hospital.

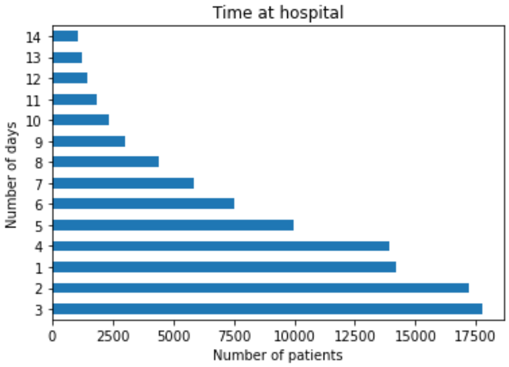
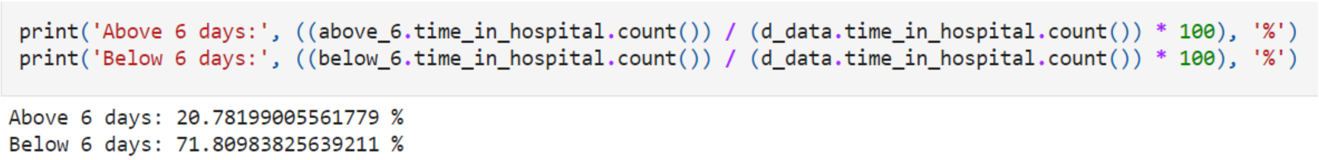


Figure 1: Number of days patients spent at hospital

As we can see from above plot, large group of patients were spent less than 6 days at hospital. The average time each patient spent at hospital is 4.5 days. Some group of patients stayed very longer up to 14 days. We tried to identify the percentage of patients stayed below 6 days and above 6 days and displayed below.



```
print('Above 6 days:', ((above_6.time_in_hospital.count()) / (d_data.time_in_hospital.count()) * 100), '%')
print('Below 6 days:', ((below_6.time_in_hospital.count()) / (d_data.time_in_hospital.count()) * 100), '%')
```

```
Above 6 days: 20.78199005561779 %
Below 6 days: 71.80983825639211 %
```

More than 70% of the patients stayed below 6 days in hospital and 20% of the patients stayed above 6 days. Clearly, this split rises some important questions to be asked.

- Why some group of patients staying longer days in hospital?
- How large number of patients avoided staying in hospital for long time?

Change of medications

We investigated whether specific group of patients were given any change in their medications that may have helped them to avoid hospital readmission within 30 days. EDA found 60% of the patients were given change in medication, whereas 40% of patients were not given. Intuitively, we can understand medicines can have positive impact on the patients, some medicines can improve health significantly that might help patients avoid hospital readmission. So, we next explored the medicines columns.

Distribution of medicines

Our dataset had 24 different medicine combinations given to each patient and prescribed into four categories.

- No – not prescribed
- Steady – prescribed, but no change in dose
- Up – prescribed, increased dose
- Down – prescribed, decreased dose

We visualized these distributions of all medicines using count plots and found important medicines as features for our prediction.

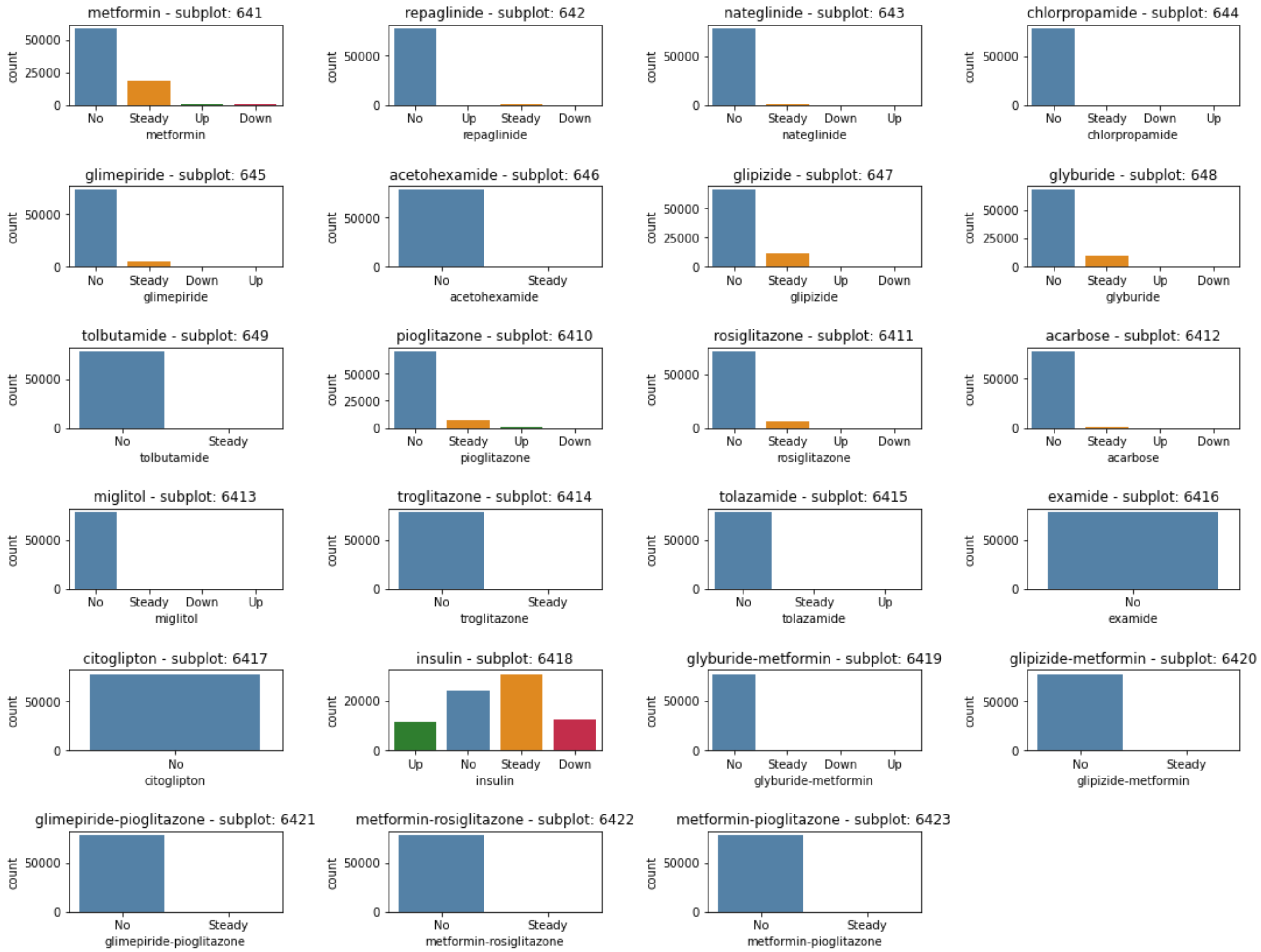


Figure 2: Distribution of all medicines and their prescription of dosage levels

From above plots, we found below medicines have visible change in their prescription level, so these may be helpful for further investigation.

- Insulin
- Metformin
- Glimepiride
- Glipizide
- Glyburide
- Pioglitazone
- Rosiglitazone

4. Feature Engineering

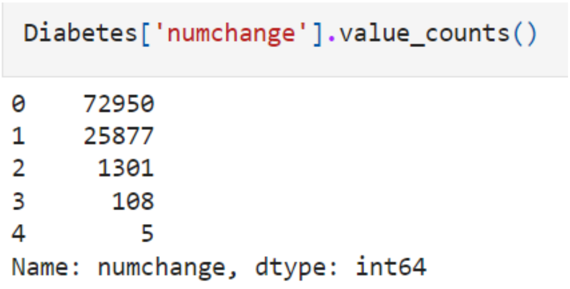
We generated new features out of existing columns to gain more insights about the data that might help in building our prediction models.

Service utilization

We combined information from number of outpatients, number of inpatients and number emergency columns to measure how much each patient utilized the hospital over the years and stored this information in a new column named ‘service utilization’.

Number of medication changes

All the medicines columns were prescribed into four categories that may be spare for our analysis. So, to simplify our analysis we created a new column named ‘number of changes’ to capture how many patients got their medicine prescription changed over time. We encoded these change in range from 0 to 4 as below.



```
Diabetes['numchange'].value_counts()
```

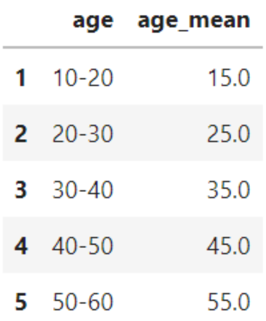
```
0    72950
1    25877
2     1301
3      108
4        5
Name: numchange, dtype: int64
```

Number of medications used

Another possibly related factor could be the total number of medications used by the patient, which may indicate severity of their condition and or the intensity of care. So, we created another feature by counting the medications used during the encounter from above column.

Mean age

The age column characterized the patients into 10 different age categories which may be difficult for analysis. So, we converted each of the age category to their average age, hence keeping all the entries as numeric. For example, age [10-20] category will have mean age of 15 as their numeric entry.



	age	age_mean
1	10-20	15.0
2	20-30	25.0
3	30-40	35.0
4	40-50	45.0
5	50-60	55.0

Encoding target variable

The target variable contained three different classes. Since our aim is to predict the possibility of readmission within 30 days. We encoded the three classes into binary classification.

Class	Binary class	Description
No	0	Not readmitted
>30	0	readmitted after 30 days
<30	1	readmitted within 30 days

4. Preprocessing the data

Our dataset had 10 numeric feature columns and remaining columns as categorical features after cleaning and EDA. We processed both type of features separately and merged into final DataFrame.

Removing skewness and kurtosis in numeric columns

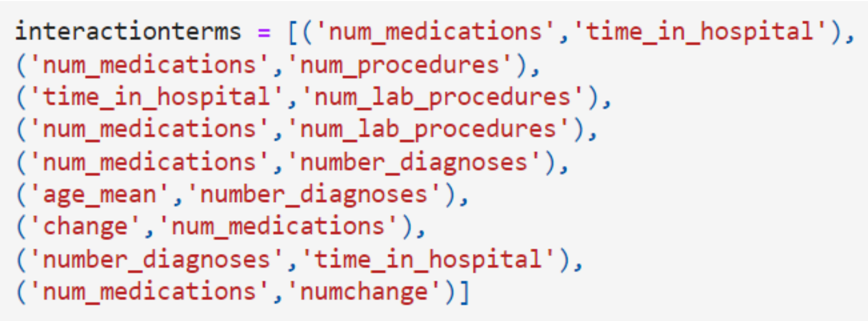
We found skewness and kurtosis in certain numeric columns and used log transformation to remove them. Also, we performed datatype conversion in the columns required.

Encoding categorical features

We encoded all the categorical columns such as gender, glucose level, A1C result and diabetes change level columns into numeric levels before analysis.

Interaction terms

To find the relationship between different variables, we created new interaction terms between two columns and stored them as separate columns.



```
interactionterms = [('num_medications','time_in_hospital'),
('num_medications','num_procedures'),
('time_in_hospital','num_lab_procedures'),
('num_medications','num_lab_procedures'),
('num_medications','number_diagnoses'),
('age_mean','number_diagnoses'),
('change','num_medications'),
('number_diagnoses','time_in_hospital'),
('num_medications','numchange')]
```

Correlation heatmap

We generated correlation map for all the features after encoding all the variables into numeric types. This helped to visualize the important relationships between several variables.

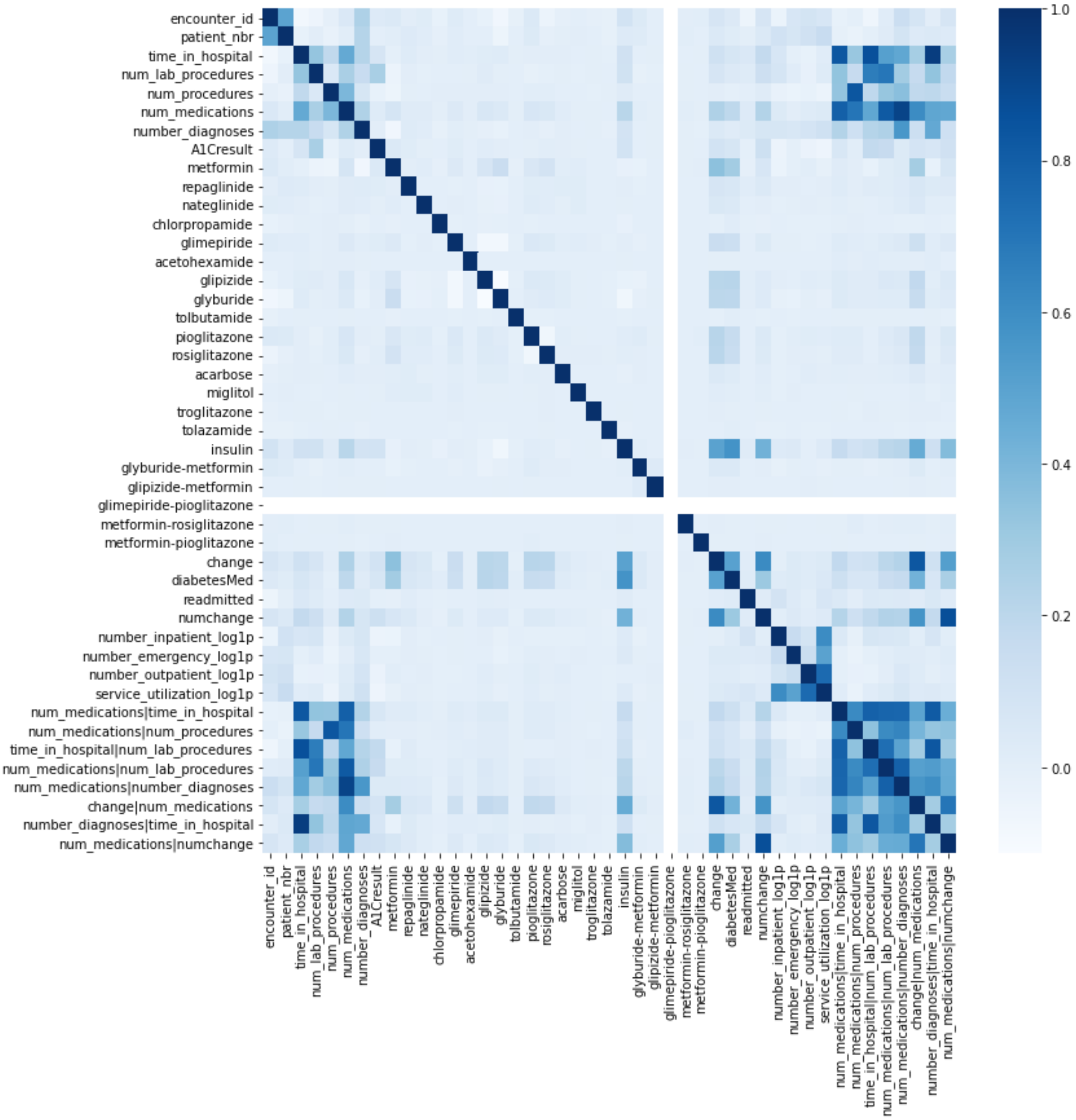


Figure 3: Correlation heatmap of all features

Observations from heatmap:

- Time in hospital and number of medications are positively correlated
- Num of procedures and number of medications are positively correlated
- Age and number of emergencies are negatively correlated

We scaled the data using standard scaler method and set their mean level as zero and standard deviation as one to normalize the values of all the features.

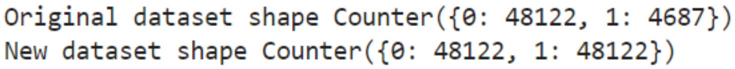
5. Training and model selection

We split our dataset into training and validation sets in the ratio of 75:25 and proceeded with training. Three different classifier models were evaluated to train our dataset and based on their performance metrics final model was selected. See below,

1. Logistic regression
2. k-Nearest Neighbors (kNN) classifier
3. Gradient boosting classifier

SMOTE for balancing data:

Prior to the training, we rectified the class imbalance problem in our dataset using Synthetic Minority Oversampling Technique (SMOTE) to balance the classes. Below the results,



```
Original dataset shape Counter({0: 48122, 1: 4687})
New dataset shape Counter({0: 48122, 1: 48122})
```

Model comparison

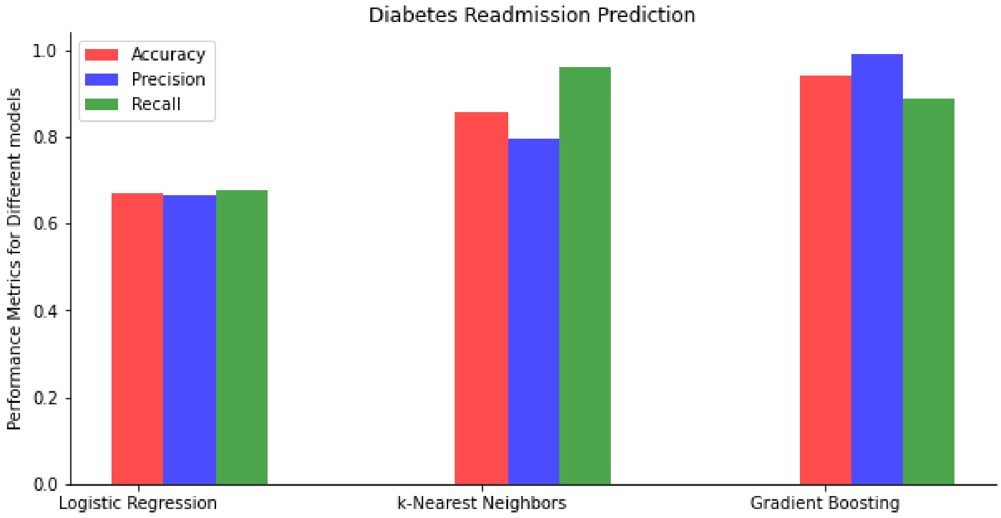


Figure 4: Comparison of performance metrics between three models

For our prediction our model, the accuracy and precision metrics are very crucial as our readmitted class size is small and we need to predict these small sizes of patients with high precision. Hence, looking at the accuracy and precision scores, the k-NN and gradient boosting models scores were good compared to logistic regression. However, considering the computational time, the gradient boosting classifier outperformed k-NN classifier in terms of their runtime. Also, the gradient boosting classifier has the top precision score among all models thus we selected gradient boosting classifier as our final model.

Final model and hyperparameter tuning:

To avoid redundancy and generalize better results on unknown data, we tuned the best parameters for the gradient boosting classifier and trained the final model. The Receiver Operating Curve (ROC) shown below produced highest AUC score of 94%. Thus, we selected the tuned gradient boosting classifier as our final model for deployment.

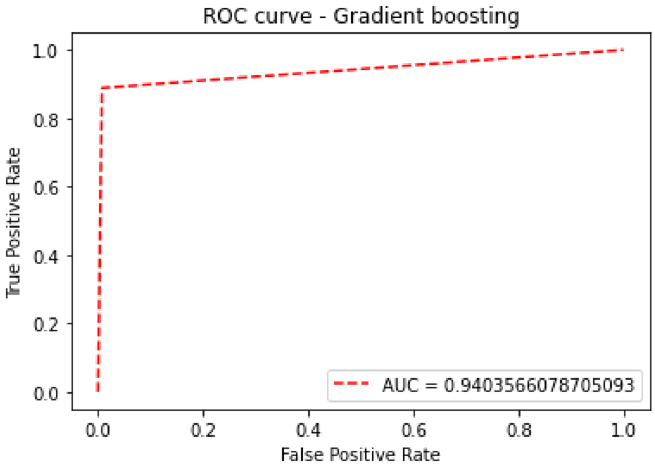


Figure 5: ROC curve of the final gradient boosting classifier

6. Summary

Out of all the models tested, the gradient boosting classifier outperformed rest of the models with highest performance metrics (Accuracy - 94%, Precision - 99%, Recall – 89%). We identified most important features of this dataset and listed below.

- Gender
- Number of procedures
- Change in medication
- Metformin
- Glipizide
- Glyburide
- Insulin

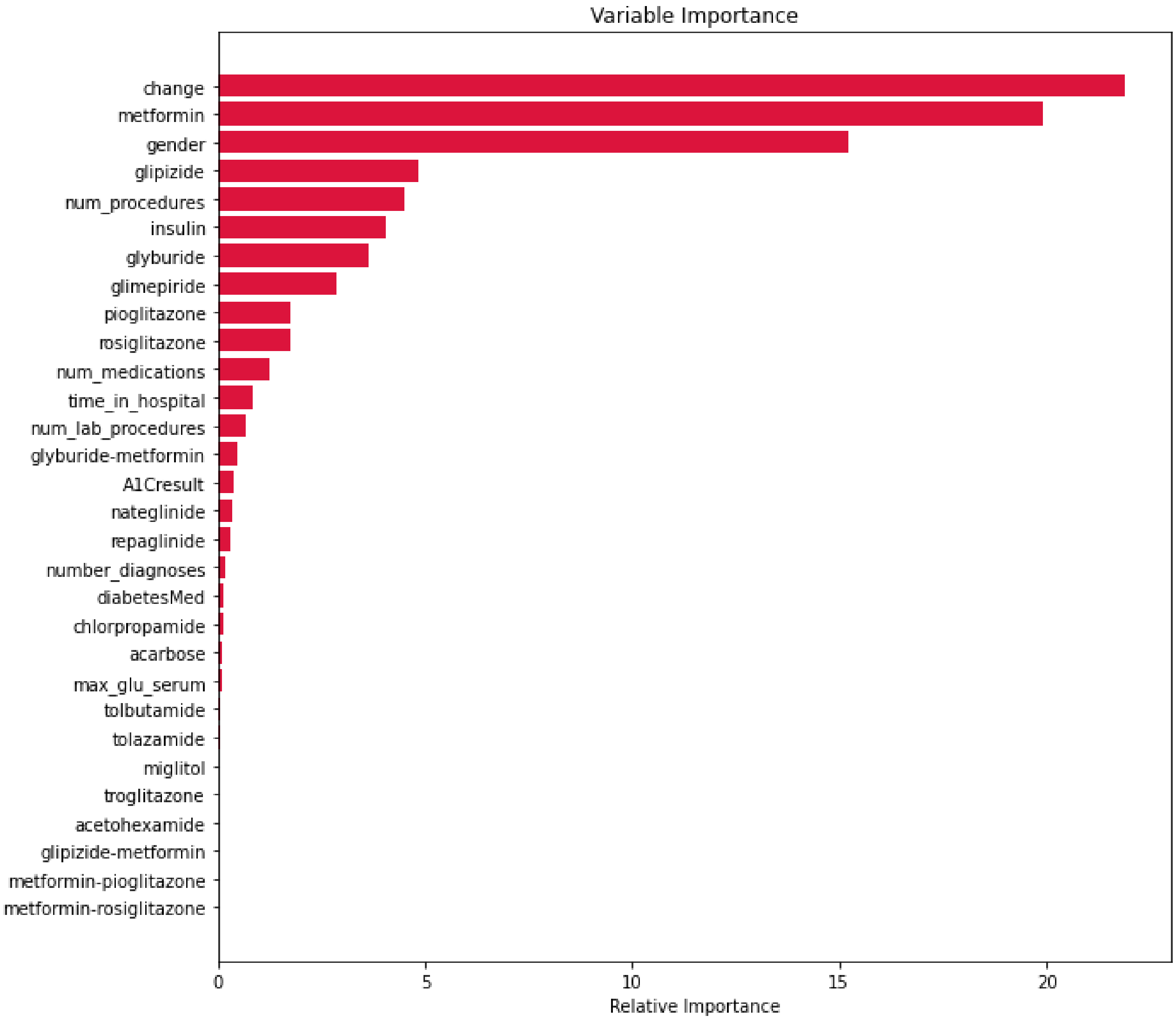


Figure 6: Features and their relative importance levels

7. Next steps

Even though the final model predicted the patient readmission with highest scores, some pitfalls in this dataset need to be addressed before applying to a larger population. First, the class imbalance problem should be resolved by including more of samples from the imbalanced class. Second, weight column contained high missing values thus we dropped from analysis. But weight information can give more insight about a patient such as their overweight or obesity status which may be linked to other diseases that may be another reason for their readmission, hence it is also an important feature needed for this hospital readmission prediction.

[Completed by: Rayees Ahamed] - https://github.com/rayees-codes