IE 7300
Statistical Learning for Engineering

Diabetes Readmission Analysis and Prediction

Project Group 6

Ankita Yadav
Apeksha UdayaKumar
Raj Deelip Soundar Raj
Santosh Yedla

1. Project Goal:

The primary goal of this project is to create a machine-learning model that can accurately predict whether a patient with diabetes will be readmitted to a hospital after their initial discharge. The readmission of patients with diabetes is a major concern for healthcare providers due to the associated high costs, increased mortality rates, and reduced quality of life for patients. By developing an accurate predictive model, healthcare providers can identify patients who are at higher risk of readmission.

2. Introduction:

The project aims to gain insights from a given dataset and identify the most relevant features to create a predictive model. The First step in this project is to perform Exploratory Data Analysis to gain a better understanding of the dataset and identify any missing data. We will then perform Feature Engineering to select the most relevant features for our model and transform the data into a format that can be used for Machine Learning. We will use a combination of supervised learning algorithms, including Logistic Regression, Naive bayes without using sklearn. We also performed Neural Networks. Then, we will evaluate the performance for each of the models by using the metrics such as accuracy, precision, recall, and F1 score.

3. Business Value:

- Predicting which patients will be readmitted to the hospital has a lot of significance. By predicting the readmission rate, hospital will be able to better manage the occupancy and will be able to make necessary changes to provide very good patient care.
- Knowing whether the patient will be readmitted will help the hospital staffs' plan the medical care plans for the patient and will enable them to provide preventive care for the patients.
- Also, patients will be able to save a lot of money by knowing if they must do further treatment beforehand.

4. About the Dataset:

Link: Diabetes 130 US hospitals for years 1999-2008

Diabetes 130-US hospitals for years 1999-2008 dataset is available on the UCI Machine Learning Repository. Following is the overview of the dataset.

- The dataset used in the report contains information on diabetes patients from 130 US hospitals with patient encounters from 1999-2008
- The dataset has a total of 100,000 instances and 50 attributes.
- The dataset is multivariate, with a mix of numerical and categorical data types.
- The target variable is whether a patient was readmitted to the hospital after the discharge.
- Other important features in the dataset include demographic information such as race, gender, age, and weight, as well as medical information such as diagnosis codes, medications prescribed, and number of lab procedures.
- The dataset has missing values, which will need to be handled appropriately during analysis.

5. Exploratory Analysis:

5.1. Description of the data:

Column name	Type	Description
Encounter ID	int	Unique identifier of an encounter
Patient number	int	Unique identifier of a patient
Race	category	Caucasian, Asian, African American, Hispanic, and other
Gender	category	male, female, and unknown/invalid
Age	category	(0, 10), 10, 20), …, 90, 100)
Weight	int	Weight in pounds
Admission type	category	Integer identifier corresponding to 9 distinct values
Discharge disposition	category	Integer identifier corresponding to 29 distinct values

Admission source	category	Integer identifier corresponding to 21 distinct values
Time in hospital	int	Integer number of days between admission and discharge
Payer code	category	Integer identifier corresponding to 23 distinct values
Medical specialty	category	Integer identifier of a specialty of the admitting physician
Number of lab procedures	int	Number of lab tests performed during the encounter
Number of procedures	int	Numeric Number of procedures (other than lab tests) performed during the encounter
Number of medications	int	Number of distinct generic names administered during the encounter
Number of outpatient visits	int	Number of outpatient visits of the patient in the year preceding the encounter
Number of emergency visits	int	Number of emergency visits of the patient in the year preceding the encounter
Number of inpatient visits	int	Number of inpatient visits of the patient in the year preceding the encounter
Diagnosis 1	category	The primary diagnosis (coded as first three digits of ICD9)
Diagnosis 2	category	Secondary diagnosis (coded as first three digits of ICD9)
Diagnosis 3	category	Additional secondary diagnosis (coded as first three digits of ICD9)
Number of diagnoses	int	Number of diagnoses entered to the system
Glucose serum test result	category	“>200,” “>300,” “normal,” and “none”
A1c test result Indicates the range of the result or if the test was not taken. Values	category	“>8,” “>7,” “normal,” and “none”
Change of medications	category	“change” and “no change”

Diabetes medications	category	"yes" and "no"
24 features for medications for the generic names	category	(24 Columns) "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
Readmitted	category	"<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

5.2. Checking for NaN values:

```
df = df.replace("?", np.NaN)
df.isnull().sum().sort_values(ascending = False)
```

```
weight                    98569
medical_specialty         49949
payer_code                40256
race                       2273
diag_3                     1423
diag_2                      358
```

The dataset contained "?" values instead of NaN values, which were replaced with NaN values. The count of NaN values in each column was obtained. Columns "weights," "payer code," and "medical specialty" were removed from the dataset because they had more than 50% of their data points as null. The columns "encounter ID" and "patient number" were also removed since they were only indexes and did not contribute to the prediction.

As for columns with less than 10% of Null values, in case of categorical, we imputed rows using mode.

5.3. Statistics of Numerical Columns:

```
#Statistics of Numerical columns
df.describe().T
```

	count	mean	std	min	25%	50%	75%	max
encounter_id	101766.0	1.652016e+08	1.026403e+08	12522.0	84961194.0	152388987.0	2.302709e+08	443867222.0
patient_nbr	101766.0	5.433040e+07	3.869636e+07	135.0	23413221.0	45505143.0	8.754595e+07	189502619.0
admission_type_id	101766.0	2.024006e+00	1.445403e+00	1.0	1.0	1.0	3.000000e+00	8.0
discharge_disposition_id	101766.0	3.715642e+00	5.280166e+00	1.0	1.0	1.0	4.000000e+00	28.0
admission_source_id	101766.0	5.754437e+00	4.064081e+00	1.0	1.0	7.0	7.000000e+00	25.0
time_in_hospital	101766.0	4.395987e+00	2.985108e+00	1.0	2.0	4.0	6.000000e+00	14.0
num_lab_procedures	101766.0	4.309564e+01	1.967436e+01	1.0	31.0	44.0	5.700000e+01	132.0
num_procedures	101766.0	1.339730e+00	1.705807e+00	0.0	0.0	1.0	2.000000e+00	6.0
num_medications	101766.0	1.602184e+01	8.127566e+00	1.0	10.0	15.0	2.000000e+01	81.0
number_outpatient	101766.0	3.693572e-01	1.267265e+00	0.0	0.0	0.0	0.000000e+00	42.0
number_emergency	101766.0	1.978362e-01	9.304723e-01	0.0	0.0	0.0	0.000000e+00	76.0
number_inpatient	101766.0	6.355659e-01	1.262863e+00	0.0	0.0	0.0	1.000000e+00	21.0
number_diagnoses	101766.0	7.422607e+00	1.933600e+00	1.0	6.0	8.0	9.000000e+00	16.0

5.4. EDA for each column:

For all the columns where we had NaN values, we imputed values into those rows using mode. And then, transformed the columns into binary columns. Some of the finding and changes while we did exploratory analysis includes,

Readmitted:

The dataset's target feature is the "Readmitted" column, which represents the number of days to inpatient readmission. If a patient was readmitted within 30 days, it is marked as "30," if readmitted after more than 30 days, it is marked as ">30," and if no record of readmission exists, it is marked as "NO." About 60% of data points are about the patients who were never re admitted. We decided to reduce the values to two for analysis purposes and mapped them according to the following rule: "NO" was converted to 0 and "30" and ">30" were converted to 1.

Race:

Caucasian patients account for 76% of all records in the dataset, while African Americans, Hispanics, Asians, and Other races account for the

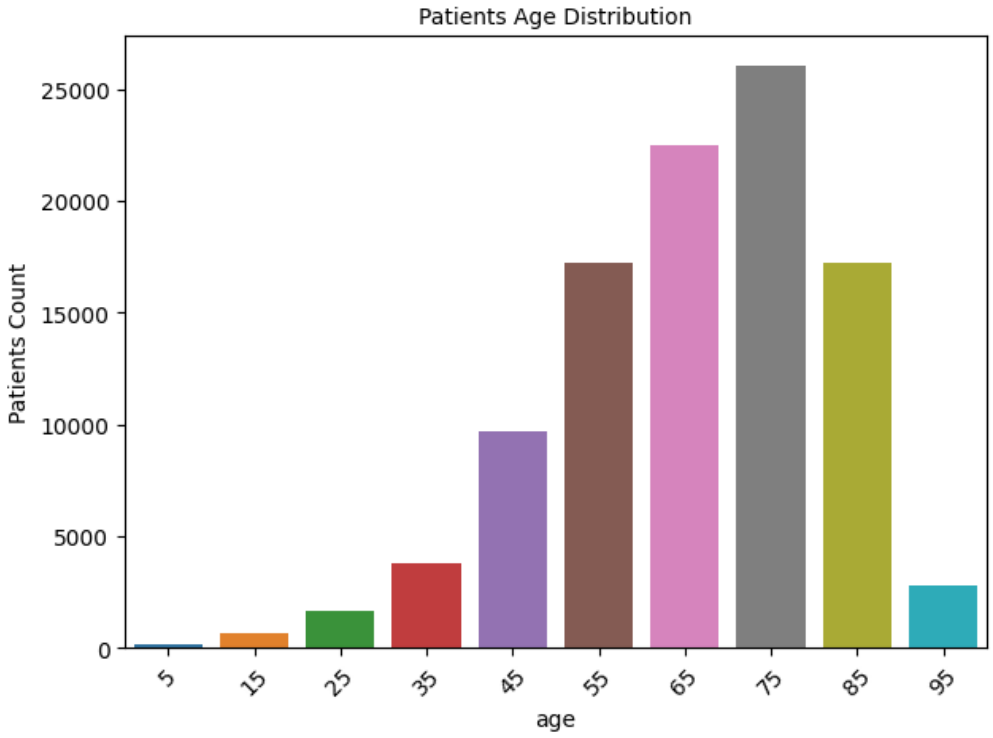
remaining 24%. However, as the number of records for Hispanic and Asian races were almost negligible, we combined them with the "Other" category.

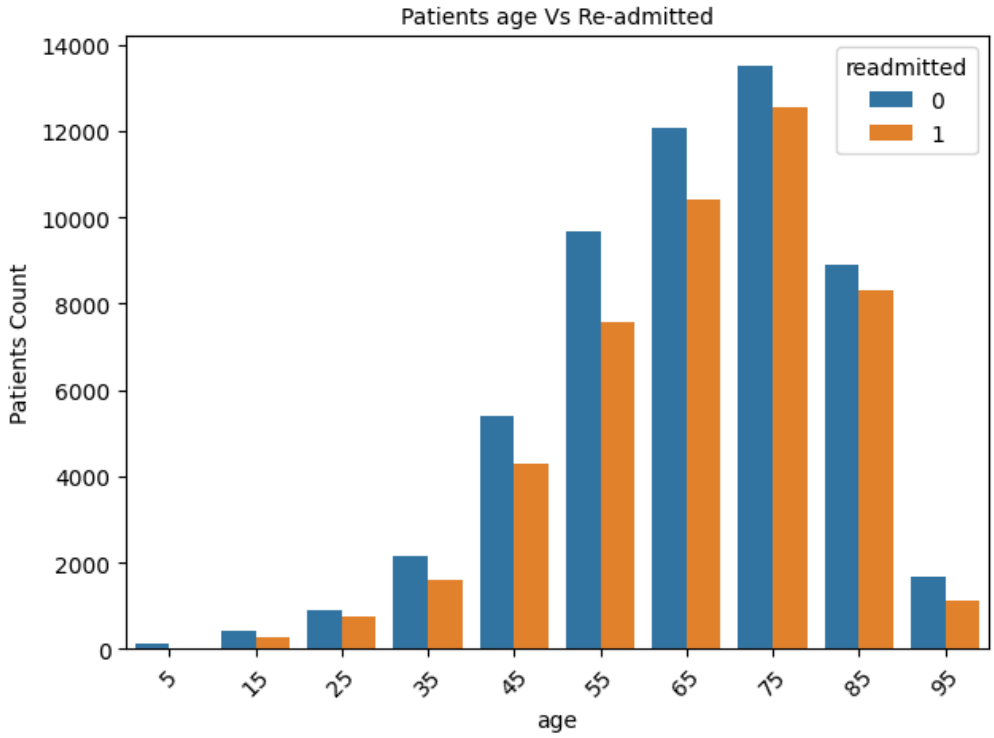
Gender:

While male and female population consist of about 40-60% of the data, the 'Unknown/Invalid' just had 3 patients, which is negligible as compared to the other categories, hence, we will drop those records before converting the column to binary with 'Female'=0 and 'Male'=1.

Age:

Since, age is an ordered category (i.e., data is grouped in ordered categories) we took the mean of the lower value and the upper value in the bin. More than 70% of the patients who came to the hospital for diabetes related issue are above the age range of 45.





Admission_type_id:

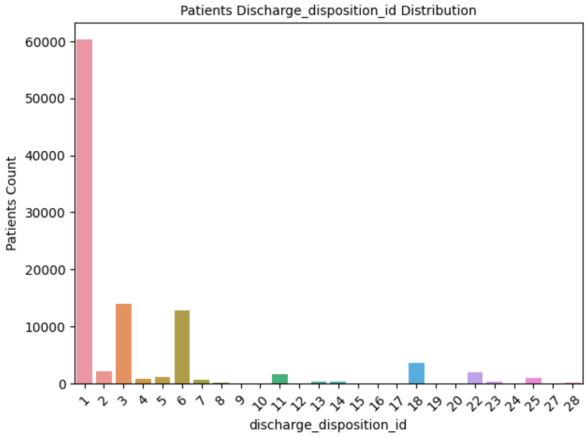
There are several admission types ID, including Emergency, Urgent, Elective, Newborn, Not Available, NULL, Trauma Center, and Not Mapped. We used the following rule to map them:

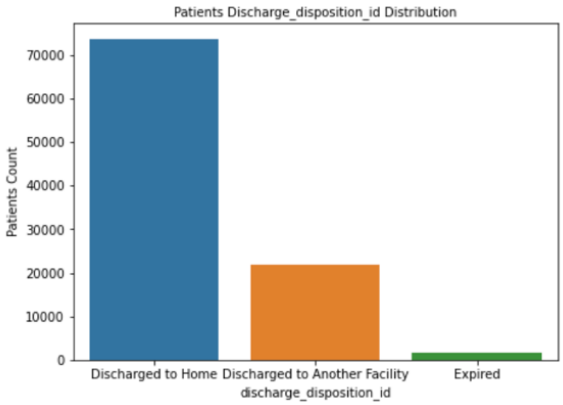
- Urgent and Emergency were combined into a single category called 'Emergency'.
- Not Available, Not Mapped, or Null were mapped as 'Null'.
- Trauma Center and Newborn categories accounted for less than 0.05% of the data, they were eliminated.

About 70% of the patients were admitted in Emergency

discharge_disposition_id:

The discharge_disposition_id column is an integer identifier that represents 29 different outcomes, including being discharged to home, expiration, or the outcome being unavailable as shown below.



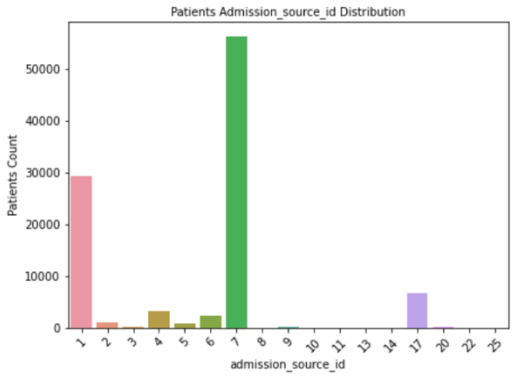


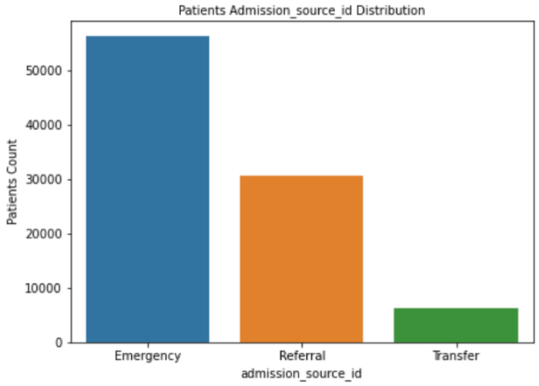
We had too many data points for the column with 29 different values. We grouped some values together to make things easier. For example, if a patient was discharged, we classified it as "Home." We classified it as "Another Facility" if they were sent somewhere else. We labeled a patient as "Expired" if he or she died or was in hospice. We also removed some specific values with 'Null' had little meaning. After that, we excluded data for those who passed away because they will never be readmitted. As a result, we can delete the data of patients who were marked as "Expired" from our dataset. The column was then converted to binary, by replacing 'Discharged to Another Facility' with 1 and 'Discharged to Home' with 0.

admission_source_id:

Since, there are many values which can be mapped into one category. We have combined them as following:

- Referral: Physician Referral, Clinic Referral, HMO Referral
- Transfer: Transfer from a hospital, Transfer from a Skilled Nursing Facility (SNF), Transfer from another health care facility, Court/Law Enforcement, Transfer from critical access hospital, Normal Delivery, Premature Delivery, Sick Baby, Extramural Birth, Transfer from hospital input/same fac result in a seperate column, Born inside this hospital, Born outside this hospital, Transfer from Ambulatory Surgery Center, Transfer from Hospice, Transfer From Another Home Health Agency, Readmission to Same Home Health Agency
- Emergency: Emergency Room
- Null: NULL, Not Available, Not Mapped, Unknown/Invalid





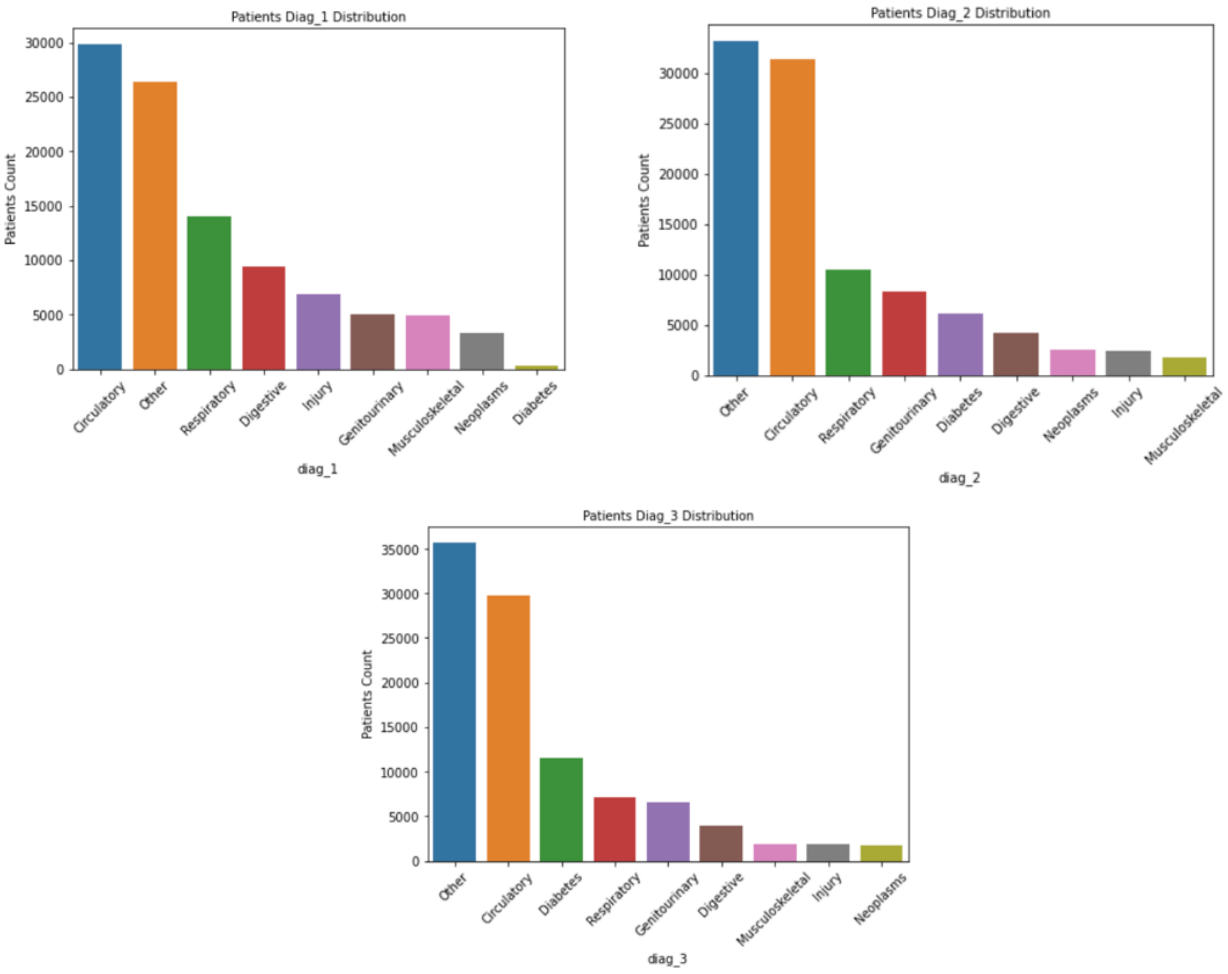
Diagnosis: diag_1, diag_2, diag_3:

CODE RANGE	ICD-9-CM SECTIONS
001-139	INFECTIOUS AND PARASITIC DISEASES (001-139)
140-239	NEOPLASMS (140-239)
240-279	ENDOCRINE, NUTRITIONAL AND METABOLIC DISEASES, AND IMMUNITY DISORDERS (240-279)
280-289	DISEASES OF THE BLOOD AND BLOOD-FORMING ORGANS (280-289)
290-319	MENTAL, BEHAVIORAL AND NEURODEVELOPMENTAL DISORDERS (290-319)
320-389	DISEASES OF THE NERVOUS SYSTEM AND SENSE ORGANS (320-389)
390-459	DISEASES OF THE CIRCULATORY SYSTEM (390-459)
460-519	DISEASES OF THE RESPIRATORY SYSTEM (460-519)
520-579	DISEASES OF THE DIGESTIVE SYSTEM (520-579)
580-629	DISEASES OF THE GENITOURINARY SYSTEM (580-629)
630-679	COMPLICATIONS OF PREGNANCY, CHILDBIRTH, AND THE PUERPERIUM (630-679)
680-709	DISEASES OF THE SKIN AND SUBCUTANEOUS TISSUE (680-709)
710-739	DISEASES OF THE MUSCULOSKELETAL SYSTEM AND CONNECTIVE TISSUE (710-739)
740-759	CONGENITAL ANOMALIES (740-759)
760-779	CERTAIN CONDITIONS ORIGINATING IN THE PERINATAL PERIOD (760-779)
780-799	SYMPTOMS, SIGNS, AND ILL-DEFINED CONDITIONS (780-799)
800-999	INJURY AND POISONING (800-999)
E000-E999	SUPPLEMENTARYCLASSIFICATION OF EXTERNAL CAUSES OF INJURY AND POISONING (E000-E999)
V01-V91	SUPPLEMENTARY CLASSIFICATION OF FACTORS INFLUENCING HEALTH STATUS AND CONTACT WITH HEALTH SERVICES (V01-V91)

Reference: ICD – 9 CM Sections

There are around 750+ unique ICD 9 codes in each of the diagnosis colu mns. Based on the ICD-9 groups we will group the codes and group them

into following category - Circulatory, Respiratory, Digestive, Injury, Geni tourinary, Musculoskeletal, Neoplasms, Diabetes, Other.



About more than 60% of the patients were diagnosed with Circulatory.

Max_glu_serum:

The column "Glucose serum test result" displays the test result range or whether the test was not performed. The values are classified as ">200," ">300," "normal," or "none." We grouped ">200" and ">300" as 1, "Nor m" as 0, and "None" as -99 to simplify the data.

A1Cresult:

We have different values for the A1c test result column that indicate the range of the result or if the test was not taken. If the result was greater than 8%, the value is ">8," and if it was less than 7%, the value is "norma l." We replaced these values based on the A1c test result range. We grou ped the values ">7" and ">8" to one, "Norm" to zero, and "None" to -99.

Time_in_hospital:

It represents the total number of days a patient spends in the hospital, starting with admission and ending with discharge. Majority of the patie nts spent around 5 days in the hospital.

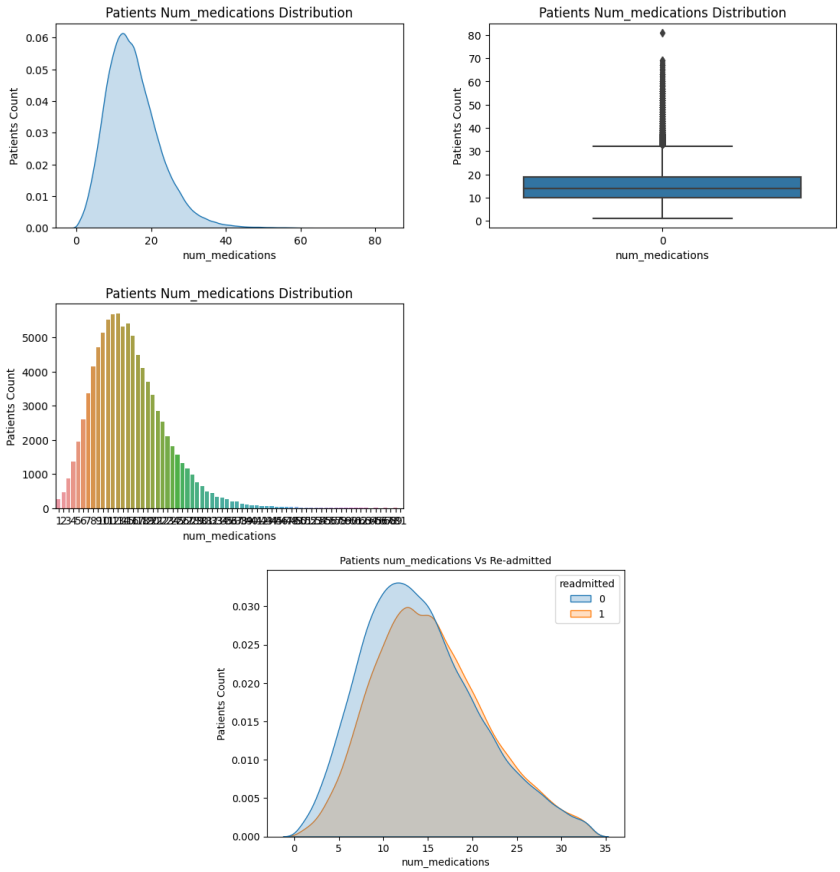
Num__lab_procedures:
About 80% of the patients had lab tests performed after visiting the hos pital.

Num_procedures:

Approximately 65% of the patients had tests other than lab tests performed, after visiting the hospital

num_medications:

About 75% of the patients were under less than 20 medications. However, we do not see much difference in distribution between readmitted and other patients



Num_outpatient:

The number of times the patient received medical treatment as an outpatient in the preceding year.

The distribution of outpatient visits is highly skewed, making any clear trend difficult to identify. We observed that patients who had less than five outpatient visits were less likely to be readmitted. For the outpatient visits greater than or equal to 5, we discovered that there were more cases of readmission than non-readmission.

Num_emergency:

The number of times the patient sought medical attention in an emergency department in the preceding year.

After slicing the distribution for a detailed insight, we found out that the patients who did not have any emergency visits (0 visits) are not likely to

be readmitted to the hospital as those who did. Patients are more likely to be readmitted to the hospital as the number of emergency visits approaches 10.

Num_inpatient:

The number of times the patient was admitted to a hospital as an inpatient in the previous year.

We looked at the Number of Inpatient Visits column to gain a better understanding. According to the data, patients with fewer than or equal to 5 in-patient visits are not likely to be readmitted. However, patients with more than 5 in-patient visits have a higher likelihood to be readmitted.

Medicinal Columns:

We have 24 columns on different medications given to the patients. The columns indicate whether the dosage of the medicine has been increased, maintained, decreased or stopped. We notice that only few medicines like insulin, metformin, glipizide, glyburide which are predominantly used in diabetes treatment are being prescribed to the patients. Other medicines have been sparsely prescribed. Hence, we decided to remove those columns.

6. Data Engineering and Pre-Processing:

6.1. Processing outliers:

For numerical features, we set the upper bound and lower bound (3 std from mean). Thus, removing any outliers.

6.2. Encoding:

We performed one hot encoding for race, and admission source columns. As for other categorical columns we did binary encoding (as they only had two categories)

(E.G: race)

	race_caucasian	race_africanAmerican
0	1	0
1	0	1
2	1	0
3	1	0
4	1	0

6.3. Feature Engineering:

The Features diag_1, diag_2, diag_3 falls under the same categories, so dividing them into 9 different columns for each diagnosis. As diagnosis 1 has higher priority while prescribing medication/procedure for patients we decided to give 3 as the value for the diagnosis if it falls in diag_1, if the diagnosis falls in diag_2 we gave 2 and 1 for diag_3.

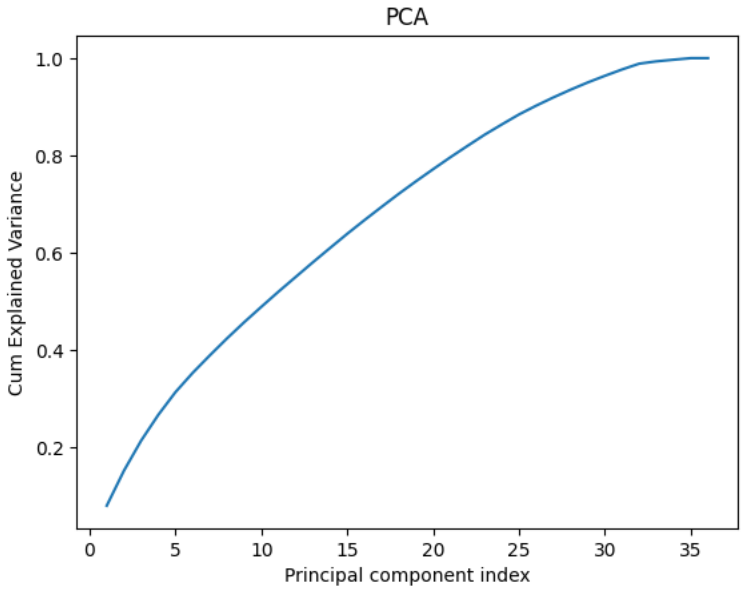
```
df[['Diabetes', 'Circulatory', 'Respiratory', 'Digestive', 'Injury', 'Musculoskeletal', 'Genitourinary',
       'Genitourinary', 'Neoplasms', 'Other']].head(5)
```

	Diabetes	Circulatory	Respiratory	Digestive	Injury	Musculoskeletal	Genitourinary	Genitourinary	Neoplasms	Other
0	0	0	0	0	0	0	0	0	0	6
1	2	0	0	0	0	0	0	0	0	4
2	0	1	0	0	0	0	0	0	0	5
3	1	0	0	0	0	0	0	0	5	0
4	0	5	0	0	0	0	0	0	0	1

6.4. Dimension Reduction:

We performed dimensional reduction, as some models like hard margin SVM, KNN are resource intensive. In order to do that we performed PCA.

First, we scaled the data with standard scalar and then performed the principal component analysis. In order to decide on the number of components to keep, we plotted the cumulative explained variance as a line chart. Analyzing the plot we decided to take the component up until which the majority of the variance is explained.



In this operation it's important to avoid losing too much variance. We chose to select the first 30 principal components, which captured 97% of the total explained variance, after considering the tradeoff variance between these two factors.

6.5. Final Data Set:

After performing feature engineering, we ended up with about 80k rows and 30 features to predict whether a diabetic patient will be readmitted or not.

7. Algorithms:

As mentioned earlier, we had performed one dimensional reduction to reduce the number of dimensions. With the optimized data set, we performed 3 types of classification. The classification models were created without use of external packages like sklearn.

7.1. Logistic Regression:

Logistic regression is a parametric classification technique that is simple to develop and commonly utilized in industrial problems. It is a quick and effective solution with little need for storage because of how little

computation is required. The objective of logistic regression is to describe the likelihood of belonging to each class as a categorical output, and to model the relationship between predictor factors and a target variable. This is accomplished by modeling the target variable as a linear function of the predictors and using a threshold to categorize a given record based on the probabilities that result. The result is a dependable and efficient method to classification difficulties that may be employed in a wide range of applications.

7.2. Naïve Bayes:

Naive Bayes is a useful tool for machine learning practitioners because of its scalability, capacity for dealing with nonlinear issues, and lack of data-related assumptions. Due to its effectiveness and simplicity, naive bayes is a common algorithm for classification applications. The algorithm bases its calculations on the assumption that the attribute values are conditionally independent given the target value, and then computes probabilities for each hypothesis. This eliminates the need to compute each attribute value separately and enables tractable computations. A popular implementation of the algorithm (for continuous data), the Gaussian Naive Bayes algorithm, assumes that the probabilities follow a Gaussian distribution.

7.3. Neural Network:

A neural network is a computer software that simulates how the human brain processes information to solve issues. The network is made up of multiple layers that function like neurons, including an input layer, an output layer, and hidden layers in between. The layers are linked by nodes, resulting in a network that processes data. Deep learning has a wide range of real-world applications, including speech recognition, natural language processing etc. They are especially beneficial for difficult tasks where regular algorithms may not be effective.

8. Model:

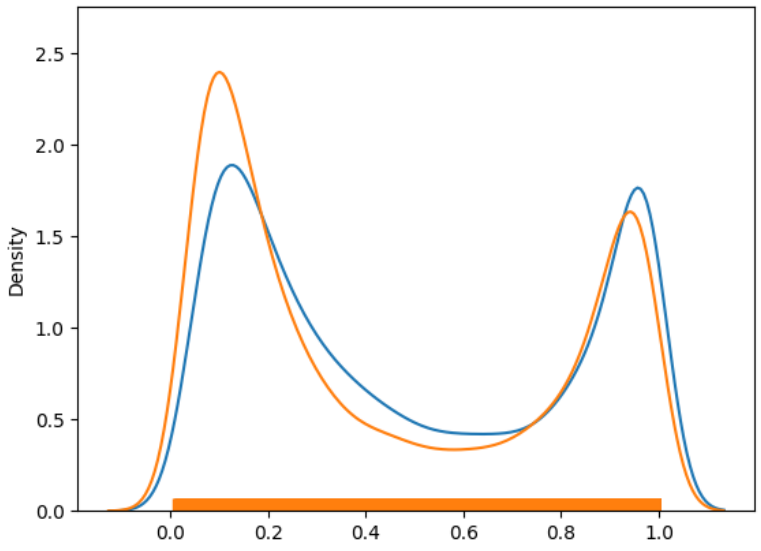
For this classification challenge, we used three models: logistic regression, naive bayes, and neural networks. The dataset we're working with contains a about 50-50 mix between our target class and the other

class. Because the data is relevant to patient care, appropriately categorizing the target class is critical. As a result, we have prioritized recall and accuracy in our evaluation metrics. We have separated the dataset into train and test sets, and we will train the models while optimizing their hyper-parameters. Finally, we shall present the test set's outcomes.

8.1. Logistic Regression:

With Logistic regression, the input is multiplied by the optimal parameters learned through iterations and the output we get will be a sigmoid function which will give the probability between 0 and 1. Then with the threshold value we decided, we will group them into 1 and 0. In order to achieve a higher recall and accuracy we will try to optimize the parameters such as learning rate, epsilon, regularization term and threshold.

In our case, initial we initially used the following parameters to train our model: MaxIteration = 1,00,000; learningRate = 0.0001, lambda = 0.0001and threshold =0.4. However, the accuracy and recall we achieved for this model was .52 and .49.



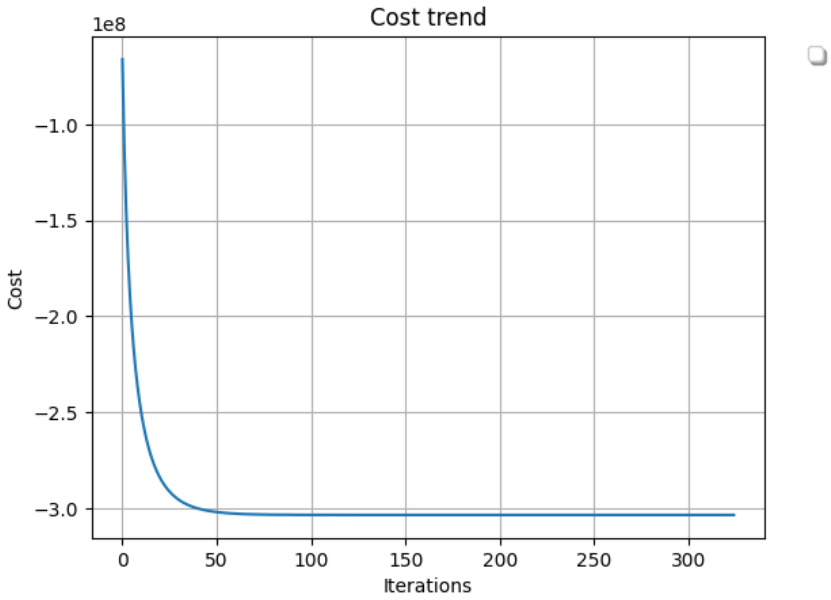
Density plot for base logistic regression

As the recall was very late for this model (presumably by higher learning rate) we decided and ran the model for different combinations of values for hyperparameters like learning rate, threshold and lambda.

Learning rate	Accuracy	Recall	Precision	F1- score	Threshold
0.1	0.5319	0.464	0.4956	0.4791	0.4
0.01	0.532	0.464	0.4957	0.4793	0.5
0.001	0.5319	0.468	0.4956	0.4816	0.4
0.0001	0.5291	0.494	0.4929	0.4936	0.4
0.00001	**0.6048**	**0.692**	**0.5603**	**0.6192**	**0.45**
0.000001	0.5638	0.841	0.5186	0.6461	0.4
0.0000001	0.5638	0.841	0.5186	0.6461	0.4

From the models we trained we chose the one with following parameters: learning rate: 0.00001, threshold = 0.45, lambda = 0.003. For this model we got accuracy of 0.605 and recall of 69.2.

Please find below the cost function for the selected logistic regression model:



Considering that false negatives are more of an issue than false positives, we decided to use this model. This model acted as a perfect tradeoff between accuracy, recall and precision.

Learning rate	Accuracy	Recall	Precision	F1-score	Threshold
0.00001	0.605	0.692	0.56	0.619	0.45
0.00001	0.564	0.841	0.519	0.646	0.4

8.2. Naïve Bayes:

Next we performed naïve Bayes. since this classification model is non parametric we were not able to perform any hyper parameter tuning. So, Assuming each column is independent we modeled gaussian naïve bayes classification for the dataset. On fitting the test data to the model, we obtained an accuracy of 55.78% and an error rate of 44.22%. The recall value of 49.97% indicates that the model was only able to rightly predict ~50% of the true positives. Considering the dataset is related to patient healthcare, we should give more importance to recall. So this model is not suitable for our business requirement.

```
Confusion Matrix is as follows (class 1 is target class)
           Predicted 0  Predicted 1
Actual 0       8356.0       5399.0
Actual 1       5882.0       5875.0

Accuracy is  55.78 %
Error is  44.22 %
Recall is  49.97 %
Precision is  52.11 %
```

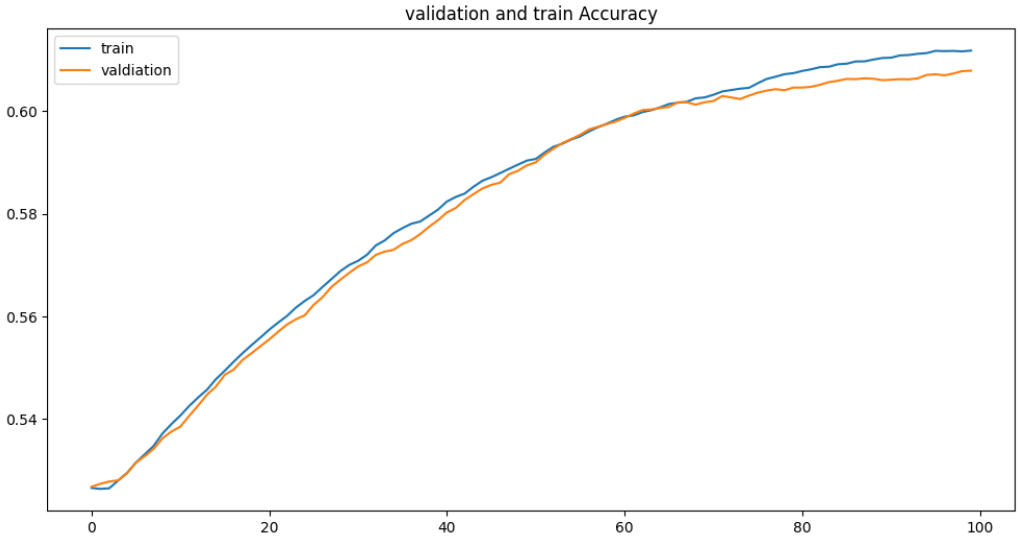
8.2. Neural Networks:

Next we performed Neural Networks. We started with a simple network model with 3 layers which are hidden, and 1 output layer. Since we know that our problem is a classifier problem, we use the sigmoid function as the activation for the output layer and the combinations of sigmoid and Relu for the hidden layer activation functions. We used 16, 32, 8 nodes in each layer going through 200 epochs with batch size of 64.

The model has a training accuracy as 64% and validation accuracy as 61%. This can be a sign that the model is **overfitting.** So we will be performing **Bias variance trade-off.**

We try to change the number of epochs, batch size, learning rate to avoid overfitting. While doing so, we achieved a model with following

parameters. Epochs: 100, batch : 128, inner layers: 2. We see this model has a better bias variance trade off



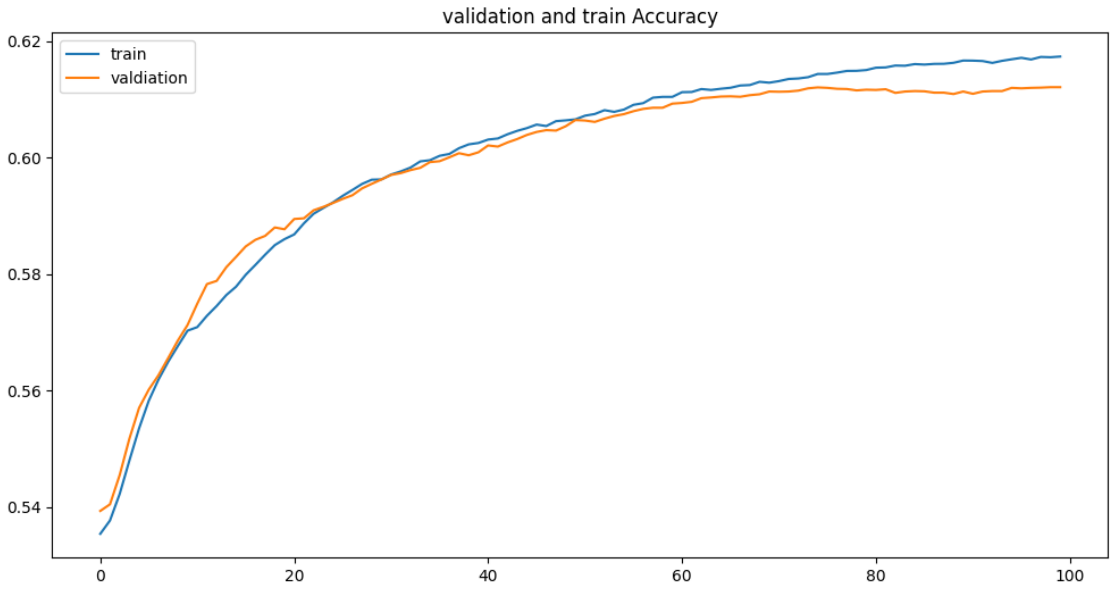
Also keeping bias-variance tradeoff in mind, we continued to build models with different combinations of hyperparameters.

Model	Epochs	Hidden Layer	Batchsize	learning rate	Threshold
1	200	3	128	0.0001	0.5
2	200	4	64	0.0001	0.5
3	200	4	128	0.0001	0.5
4	100	2	64	0.00001	0.45
5	100	2	128	0.00001	0.45

Model	F1 Score	Accuracy	Recall	precision
1	0.5329	0.6171	0.5122	0.5989
2	0.5447	0.6153	0.5389	0.5905
3	0.5316	0.6085	0.4822	0.5925

4	0.593333	0.602	0.63	0.5607
5	0.586766	0.599	0.6178	0.5587

From the Accuracy and recall scores, we decided to use the model 4 with 100 epochs, 2 hidden layers, 128 batch size and 0.00001 learning late. This model has given average accuracy while giving a average recall of 63%. Below is the validation accuracy curve for that model:



9. Discussion:

Summarizing the project, we developed three classification models with the objective of classifying and predicting whether a diabetic patient will be readmitted in the future or not. We performed data engineering steps like, dimensional reduction, encoding and other techniques to enhance the dataset. Following the accuracy and recall of the selected models.

Model	F1 Score	Accuracy	Recall
1	Logistic Regression	0.6048	0.6902
2	Naïve Bayes	0.5579	0.49
3	Neural Network	0.602	0.63

Since the distribution of the features were not that much different between the target class and the other, we did not see much accuracy in the models. However, since our dataset is based on patient welfare, we should give more importance to recall while maintaining a decent average. It would be better if we predict a non-returning patient as re-admitted than predicting a patient who might get re-admitted as won't be re-admitted. Keeping this in mind, comparing the models, I believe logistic regression is the best fit for the business need.

However, with more details on the patient history like number of previous re-admissions, medicine dosage etc. we should be able to do better predictions.