PREDICTING READMISSION RATE AMONG PATIENTS WITH DIABETES

PROJECT REPORT

Submitted by

Project Group 9

NAME	CONTRIBUTION (%)
RITHIK REDDY CHALLA	25
SHRAMAN ARYA	25
SRIKANTH MUNGI	25
VENKAT NAVNEETH BURLA	25

MASTER OF SCIENCE

in

DATA ANALYTICS ENGINEERING

NORTHEASTERN UNIVERSITY

COLLEGE OF ENGINEERING

BOSTON – 02115

Northeastern University

AUGUST 2022

TABLE OF CONTENTS

ABSTRACT

According to estimates, 9.3 percent of the American population have diabetes, of which 28 percent of the cases are undiagnosed. Due to its great incidence, diabetes is a typical comorbid illness among hospitalized patients. To assess the complexity of their patient populations and enhance quality, government organizations and healthcare systems have recently placed a greater emphasis on 30-day readmission rates. This project mainly emphasizes on patient readmission, the dataset in consideration houses data collected from 130 U.S hospitals. There are certain features which are of peculiar interest like ; type of medication, type of diagnosis, and Alc test results, which play a key role in a patient getting readmitted apart from other features. The classification models used to predict the readmission rates are Logistic regression, Support Vector Machine and Neural Networks.

INTRODUCTION

The way hyperglycaemia, or in other terms High Blood Glucose, is managed in hospitalized patients has a big impact on their outcome both in terms of morbidity and mortality. There is a proper scope for analyzing and assessing diabetic care during hospitalization which could help in treating the patients better and further reduce the chances of hyperglycaemia.

Hence the goal of this project is to analyze the readmission rates of diabetic patients who have already been treated in a hospital. The patient data were divided based on their ages into intervals of 10 years i.e. e.g. 0-10; 10-20; etc. The mid value of these ranges was chosen to convert it to categorical variables.

Since this is a medical dataset, the terminologies used are complex and difficult to comprehend. It is also natural for them to have anomalies, in the form of missing data. Features like weight and payer code have missing values greater than 90%, so these features were dropped since they will affect the performance of the machine learning models.

After performing EDA, feature engineering and dimensionality reduction were done and the project proceeded to the training phase, where we compared the baseline model with the model mentioned above.

DATA DESCRIPTION

The Dataset consists of 101766 records and 55 features including the target variable "Readmission".

Features in the Dataset:
Encounter ID: Unique identifier of an encounter

Patient number: Unique identifier of the patient
Race: Consists of Values like Caucasian, Asian, African American, Hispanic, and others
Gender: Consists of values like Male, Female, or unknown/invalid
Age: Grouped in 10-year intervals: 0, 10), 10, 20), ..., 90, 100)
Weight: weight in pounds

Admission Type: Integer identifier corresponding to 9 distinct values, for example, emergency, urgent, elective, new-born, and not available.
Discharge disposition: Integer identifier corresponding to 29 distinct values, for example, discharged to home, expired, and not available.

Admission source: Integer identifier corresponding to 21 distinct values, for example, physician referral, emergency room, and transfer from a hospital.
Time in hospital: Integer number of days between admission and discharge.
Payer code: Integer identifier corresponding to 23 distinct values, for example, Blue Cross/Blue Shield, Medicare, and self-pay.

Medical specialty: Integer identifier of a specialty of the admitting physician, corresponding to 84 distinct values, for example, cardiology, internal medicine, family/general practice, and surgeon.

Number of lab procedures: Number of lab tests performed during the encounter.
Number of procedures: Number of procedures (other than lab tests) performed during the encounter.

Number of medications: Number of distinct generic names administered during the encounter.

Number of outpatient visits: Number of outpatient visits of the patient in the year preceding the encounter
Number of emergency visits: Number of emergency visits of the patient in the year preceding the encounter

Number of inpatient visits: Number of inpatient visits of the patient in the year preceding the encounter
Diagnosis 1: The primary diagnosis (coded as the first three digits of ICD9); 848 distinct values.

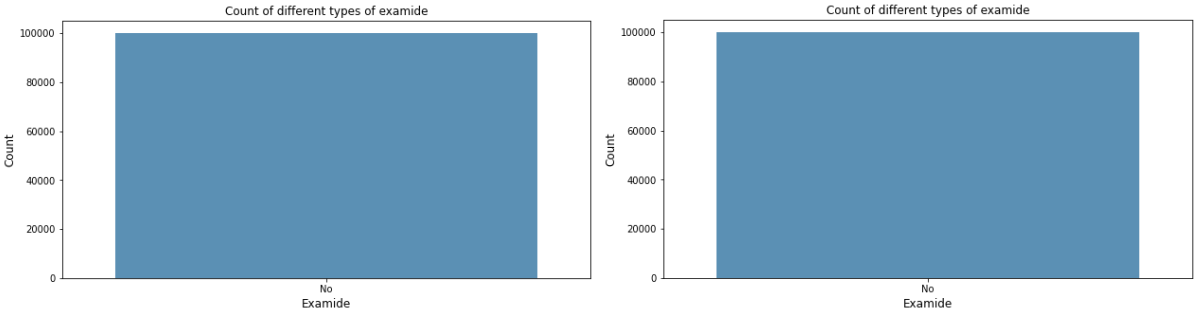
Diagnosis 2: Secondary diagnosis (coded as the first three digits of ICD9); 923 distinct values.

Diagnosis 3: Additional secondary diagnosis (coded as first three digits of ICD9); 954 distinct values

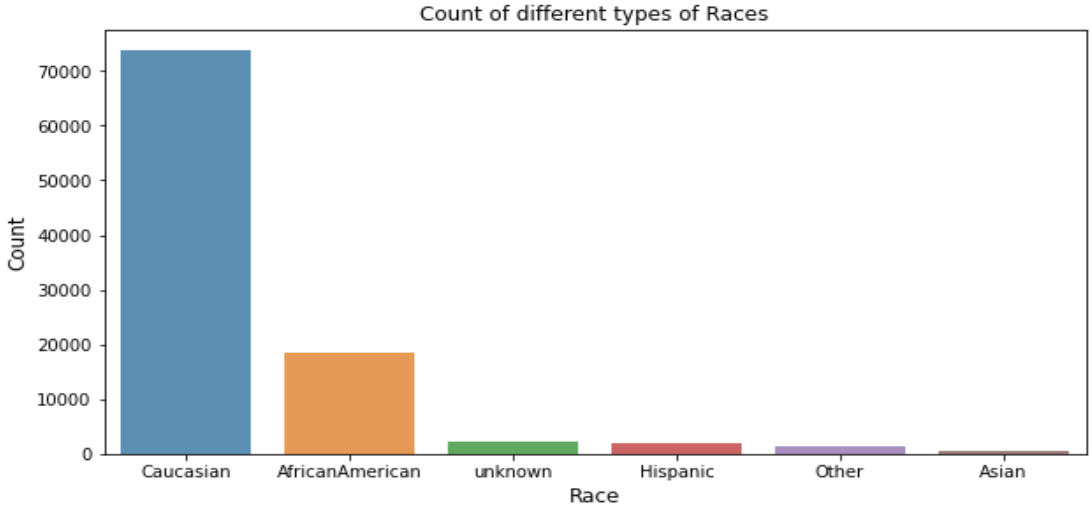
Number of diagnoses: Number of diagnoses entered in the system
Glucose serum test results: Indicates the range of the result or if the test was not taken. Values: “>200,” “>300,” “normal,” and “none” if not measured
A1c test result: Indicates the range of the result or if the test was not taken. Values: “>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.
Change of medications: Indicates if there was a change in diabetic medications (either dosage or generic name). Values: “change” and “no change”.
Diabetes medications: Indicates if there was any diabetic medication prescribed. Values: “yes” and “no”.
24 features for medications: 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.
Readmitted: 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. [1]

EXPLORATORY DATA ANALYSIS

1. The Dataset consists of 13 Numerical features and 37 categorical features and the target attribute is ‘readmitted’.
2. Few columns contain no values so replaced the columns with NaN values.
3. Calculated the percentage of missing values for each column and found out that columns such as ‘weight’, ‘payer_code’, and ‘medical_speciality’ have more than 30% of missing values, so dropped these three columns for data.
4. The Target variable has three values i.e.., ‘No’, ‘>30’, ‘<30’ as we are focusing on people admitting so we labeled as yes for <30 and >30, and No for value for NO.
5. The variables like ‘examide’ and ‘citoglipton’ have only one value which doesn’t add any value to the model training and hence is removed.



6. Columns 'encounter_id', and 'patient_nbr' have unique values in each row which is not useful in model training so dropped from the data.
7. The variable ‘Race ’ has 95% values related to only two values and is not uniformly distributed so we have dropped this variable from the data set.



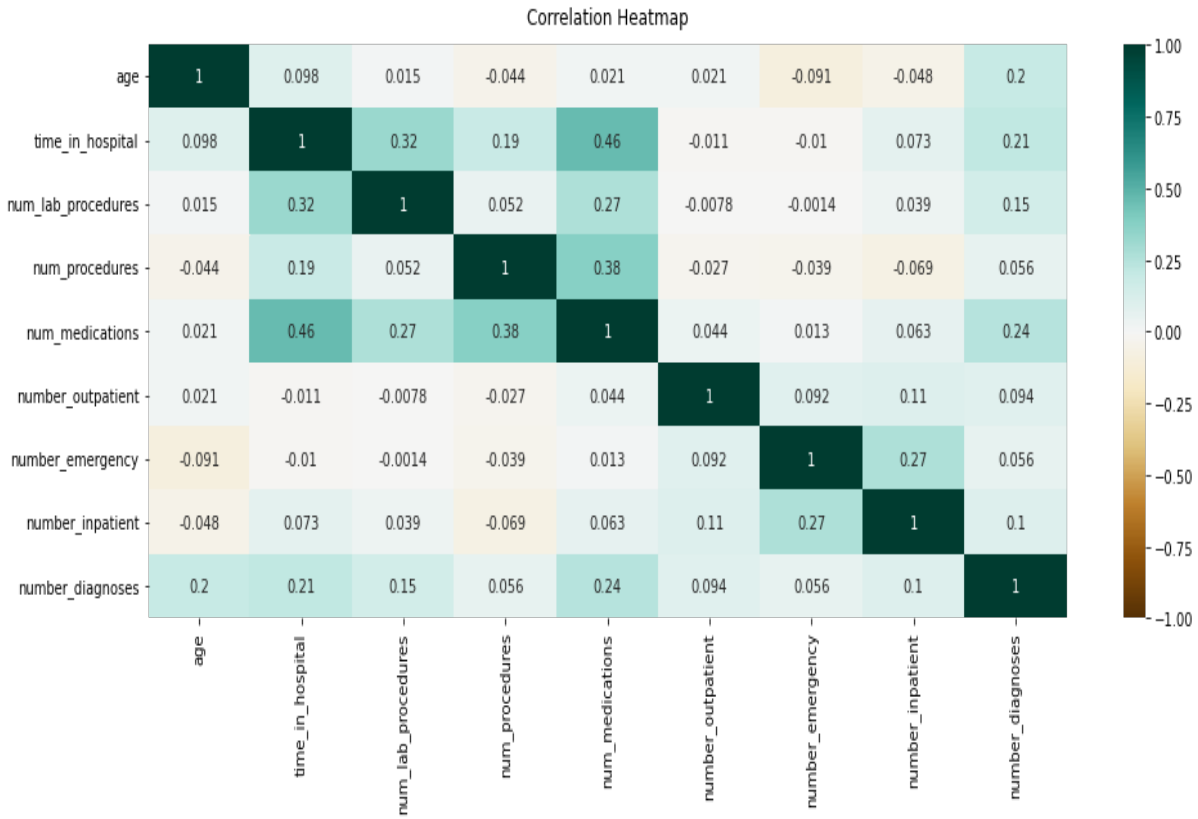
Feature Selection:

Univariate Analysis:

F-scores have been calculated for the numerical attributes and observed that columns 'admission_type_id', 'discharge_disposition_id' have fewer F-scores so we dropped them from the data.

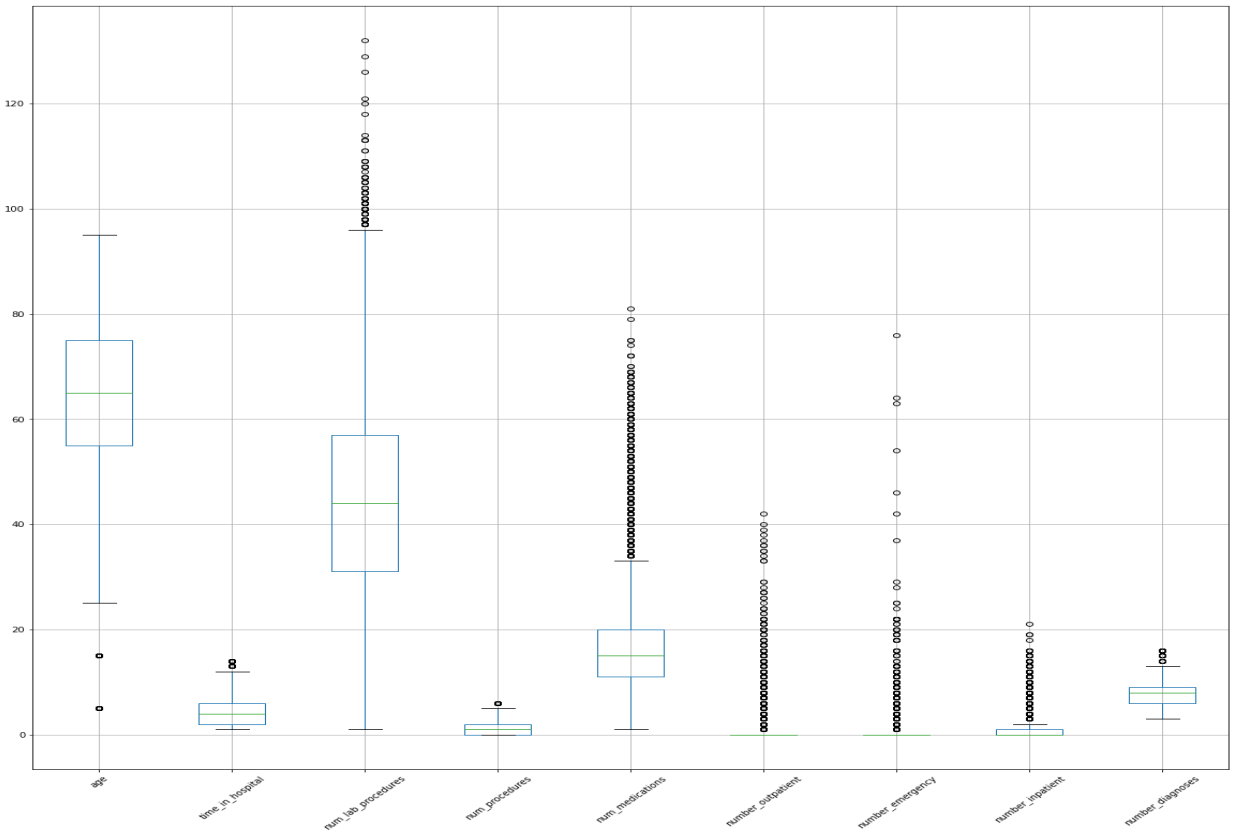
Multivariate Analysis:

Plotted correlation heat map to find the correlation among the Numerical variables.



Outlier Handling:

To detect the outliers in the values of the numerical attributes the box plots have been plotted which can be seen in the fig below. We have observed that number_emergency, number_impatient, and number_diagnosis have outliers.



Feature Engineering:

1. In the 'admission_source_id' we have 21 different categorical values, so in order to convert the categorical values into Numerical using one-hat encoding results in adding different columns which increases the dimension of data resulting Curse of dimensionality. So we have used the hash encoding instead of one-hat encoding.
2. Hashing Encoding used to transform the one feature into another using Hashing function. In our code we used the hashing function to transform the categorical data into Numerical.
3. One Hot Encoding was implemented to convert categorical values to numeric values. Our training data is more useful and expressive thanks to one hot encoding, and it is also simple to scale. Using numeric values also makes it simpler to calculate a probability for our values.
4. We have performed Chi Squared Test for the Categorical Columns to understand the feature importance and obtained the below result

	columns	chi-Scores
0	gender	15.153822
1	diag_1	1650.705501
2	diag_2	106.430192
3	diag_3	4102.178271
4	max_glu_serum	1.252089
5	A1Cresult	0.000128
6	metformin	17.139786
7	repaglinide	0.655010
8	nateglinide	0.010572
9	chlorpropamide	0.000375
10	glimepiride	0.003631
11	acetohexamide	1.123818
12	glipizide	1.620800
13	glyburide	0.791366
14	tolbutamide	0.681170
15	pioglitazone	0.486458
16	rosiglitazone	0.543684
17	acarbose	0.058533
18	miglitol	0.000080
19	troglitazone	0.461699
20	tolazamide	1.600180
21	insulin	0.976170
22	glyburide-metformin	0.003579
23	glipizide-metformin	1.089996
24	glimepiride-pioglitazone	1.123818
25	metformin-rosiglitazone	1.779648
26	metformin-pioglitazone	0.889824

27	change	80.651257
28	diabetesMed	76.275063

5. We have also used F Score for numerical column values.

	columns	Anova Scores
0	age	104.162908
1	time_in_hospital	265.565798
2	num_lab_procedures	207.860898
3	num_procedures	180.723028
4	num_medications	233.600679
5	number_outpatient	677.806577
6	number_emergency	1050.774113
7	number_inpatient	5073.463092

Standardization:

To make sure that the data is consistent, data standardization is done using the min-max scalar and standard scalar. It is observed that the results were better for standard scalar when compared to min-max scalar standardization.

Principal Component Analysis:

1. Since there is large amounts of data, there are a lot of features also. In order to tackle this, it is necessary to dimensionally reduce the data. To achieve this, Principal component Analysis is done, PCA breaks down a set of features in a dataset into a smaller number of characteristics called principle components while attempting to preserve as much information as possible from the original dataset.
2. By implementing PCA after standardization it was observed that there was 90 percent of the data in 60 principal components, whereas in min-max standard scalar there were 30 principal components with 90 percent of the data.

MODELS USED

Logistic Regression:

1. Based on a given dataset of independent variables, logistic regression calculates the likelihood that a particular event will occur, such as if the patient will be readmitted or not. The dependent variable's range is 0 to 1, since the outcome is a probability.
2. Logistic regression is chosen as the baseline model for this project. It is employed when the data is linearly separated, independent, contain no outliers, and has an outcome that is binary which is the case with our dataset. We have run the logistic regression model in two ways i.e. by reducing the dimensions using PCA and other way is without dimensionality reduction. The results are shown in the below tables.

Logistic Regression without PCA

Learning Rate	Precision	Recall	Accuracy
0.001	0.478025341	0.560745315	0.504868483
0.00001	0.61981282	0.504056868	0.620900669
0.0000001	0.616338751	0.517340418	0.621103523

Logistic Regression with PCA

Learning Rate	Precision	Recall	Accuracy
0.001	0.489516230	0.556544840	0.517918723375481
0.00001	0.61300134	0.506462267	0.617063357901142
0.0000001	0.612513098	0.503661951	0.616268848

3. We have tried varying the learning rate for both the models and have observed that both of them didn't vary much but the run time for Model with PCA (3.29 minutes) was less when compared to Model without PCA (4.179 minutes) .

4. We have also checked for Threshold values between 0.4, 0.5 and 0.6 i.e. for which threshold value the model was efficient.

Logistic Regression with PCA:

Threshold 0.4:

```
predicted value  readmitted
1                1             7811
                 0             7119
0                0             3316
                 1             1473
dtype: int64
```

Threshold 0.5:

```
predicted value  readmitted
0                0             7496
                 1             4663
1                1             4621
                 0             2939
dtype: int64
```

Threshold 0.6:

```
predicted value  readmitted
0                0             9436
                 1             6978
1                1             2306
                 0              999
dtype: int64
```

Logistic Regression without PCA:

Threshold 0.4:

```
predicted value  readmitted
1                1             7741
                 0             6848
0                0             3586
                 1             1544
dtype: int64
```

Threshold 0.5:

```
predicted value  readmitted
0                0             7473
1                1             4783
0                1             4502
1                0             2961
dtype: int64
```

Threshold 0.6:

```
predicted value  readmitted
0                0             9337
                 1             6689
1                1             2596
                 0             1097
dtype: int64
```

5. From the above results we could see that True Positives and True Negatives correctly being classified for the threshold value of 0.5, We have considered this value as the threshold. Hence the ideal parameters that can be used to get ideal accuracy and recall are learning rate 0.0001 and threshold 0.5 where the accuracy was acquired as 62% and also a better recall value of 61% among the other results.

Neural Networks:

1. The adjective "neural" refers to a neuron, while "network" refers to a graph-like structure. Neural nets are another name for neural network. These are programs that attempt to act like a typical human nervous system to solve any problem. In general, neural networks are made up of three layers. The layers are the input layer, output layer and one hidden layer.[2]

2. *Activation Function:* An activation function aids in determining a neural network's output. This sort of function is tied to each neuron in the network and assesses whether the network should be engaged or not based on the relevance of each neuron's input to the model's prediction. In this project, ReLU and Softmax Activation functions are used.

3. In ReLu, when the input is positive there is no gradient saturation problem. The calculation speed is much faster because it has only a linear relationship. It outperforms activation functions like sigmoid and tanh in both forward and backward propagation (exponent needs to be calculated by sigmoid and tanh, which will take longer). ReLu was used as an activation function for the input layers.

4. A vector of numbers is transformed into a vector of probabilities via the mathematical operation known as SoftMax, where the probability of each value are inversely proportional to the relative scale of each value in the vector. SoftMax has been used for output layers as an activation function.

5. When we have standardized the data using MinMax Scaler and applied the neural networks, there was an accuracy of 55% and when standardized using StandardScaler and then the neural networks was applied to yield an accuracy of 60%.

6. We have run the neural networks model on both (with and without PCA) and the results have been obtained as follows:

Neural Networks without PCA

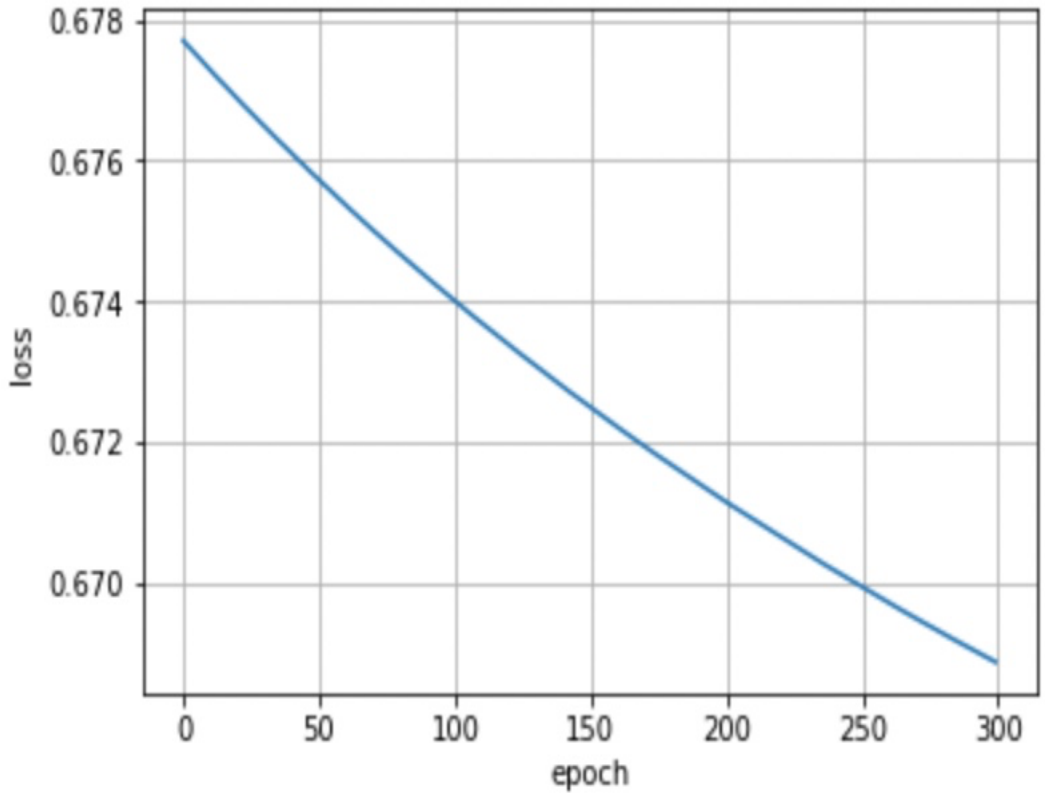
	precision	recall	f1-score	support
0	0.61	0.65	0.63	10434
1	0.58	0.53	0.55	9285
accuracy			0.60	19719
macro avg	0.59	0.59	0.59	19719
weighted avg	0.59	0.60	0.59	19719

Neural Networks with PCA

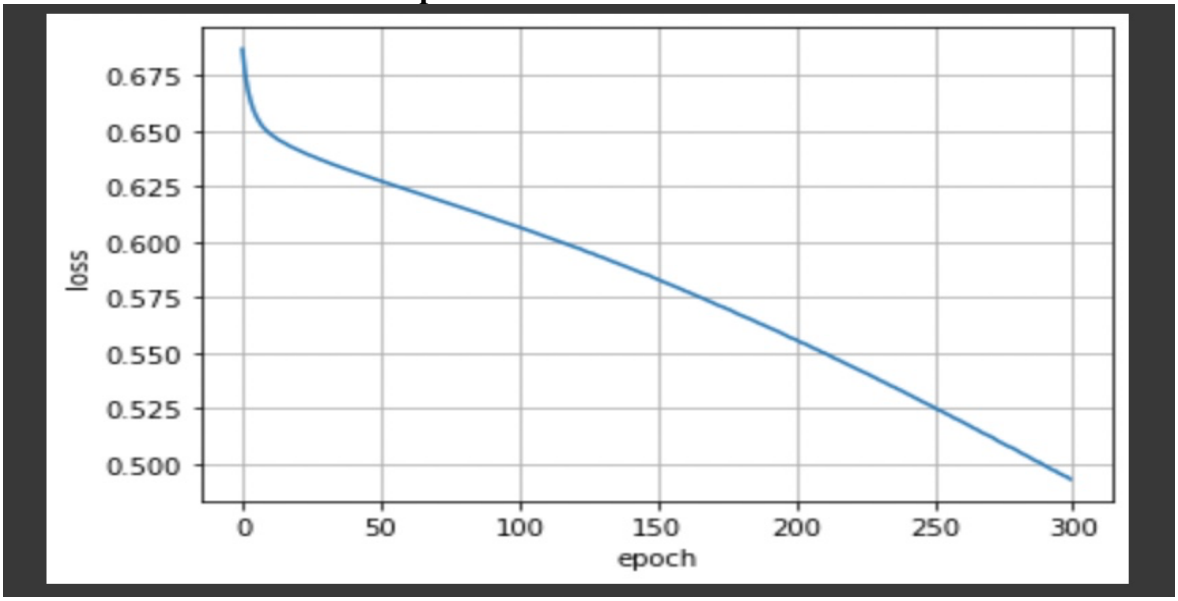
	precision	recall	f1-score	support
0	0.59	0.72	0.65	10435
1	0.59	0.44	0.50	9284
accuracy			0.59	19719
macro avg	0.59	0.58	0.58	19719
weighted avg	0.59	0.59	0.58	19719

7. Loss vs Epoch graphs for Neural Networks models with PCA and without PCA as shown below determines how much error has been obtained at that particular epoch.

Loss vs Epoch for NN model with PCA



Loss vs Epoch for NN model without PCA



8. Neural networks model with PCA ran for a less amount of time when compared to NN without PCA which had taken almost double time to run as shown below.

1	NN with pca	6.948959306875865 mins
2	NN without pca	12.6601042350133327 mins

9. We have varied the epoch models and tested the model for all the values and came to a point where epoch values greater than 300 are yielding almost the same accuracy values as the epoch value equal to 300. Hence epoch value 300 was considered ideal for the model.

10. Neural networks model with PCA has yield a better recall rate when compared to without PCA. Hence Neural networks model with PCA can be considered ideal for this dataset.

SVM:

1. In a high- or infinite-dimensional space, a support vector machine creates a hyper-plane or set of hyper-planes that can be used for classification, regression, or other tasks. Inferentially, the hyper-plane with the greatest distance from the nearest training data points of any class (referred to as the functional margin) achieves a decent separation since, generally speaking, the higher the margin, the smaller the generalization error of the classifier.

2. We have used two different types of Kernels for our dataset.

   a. Linear Kernel: When the data can be split using a single line, or when it is linearly separable, a linear kernel is utilized. When there are a large number of features, Linear Kernel in SVM is helpful
   The performance of the SVM model using Linear Kernel is as shown below:

Linear Kernel Model without PCA

	precision	recall	f1-score	support
-1	0.60	0.58	0.59	5218
1	0.54	0.56	0.55	4642
accuracy			0.57	9860
macro avg	0.57	0.57	0.57	9860
weighted avg	0.57	0.57	0.57	9860

Linear Kernel Model without PCA

	precision	recall	f1-score	support
-1.0	0.61	0.74	0.67	2609
1.0	0.61	0.47	0.53	2321
accuracy			0.61	4930
macro avg	0.61	0.60	0.60	4930
weighted avg	0.61	0.61	0.60	4930

   b. Quadratic Kernel: This is used for non linear models.
   c. The best model can be determined as the SVM model with a Linear Kernel as it has a got a better accuracy value.

Bias-Variance Trade-off:

We have achieved Bias Variance Trade-off in two ways:

1. We have used hyper parameter tuning by changing parameters like learning rate, tolerance etc. and used different values to run the model several times to obtain optimal parameter values for the respective models to yield better accuracy and recall values.
2. Splitting the data has also helped us achieve this as we have split the data into three parts i.e. Training, Validaton and testing data. We could build the model on the training set , test different values/parameters for the validation set and we could test the best parameters on testing set and generate best accuracy, precision and recall values.

DISCUSSION AND CONCLUSION:

1 Among all the models, We could observe that our baseline model i.e. Logistic Regression with model with PCA has fared better when compared to others. The training run time as well for Logistic Regression with model with PCA model was less than other models.

2 We could have achieved a better accuracy by eliminating few other features but lack of domain knowledge was the reason we did not eliminate them.

	model_name	train_time	accuracy
0	logistic without pca	4.297275014718374 mins	0.611745
1	logistic with pca	2.8174980362256368 mins	0.617678
2	NN with pca	6.945407950878144 mins	0.568183
3	NN without pca	4.3954021692276 mins	0.616512
4	SVM without pca	2.9907443563143414 mins	0.500203
5	SVM with pca	3.3405279755592345 mins	0.519473

3 A few other features which had a high feature importance score when the F score and Chi Squared tests were performed, like num_inpatient, num_emergency etc. can be used for readmissions in the hospitals.

REFERENCES:

1) **Dataset:** https://archive.ics.uci.edu/ml/datasets/diabetes+130-us+hospitals+for+years+1999-2008
2) https://www.hindawi.com/journals/bmri/2014/781670/
3) https://github.com/andrewwlong/diabetes_readmission/
4) https://www.irjet.net/archives/V8/i11/IRJET-V8I1111.pdf