Predicting US Hospital Readmissions for Diabetic Patients

GROUP NO.-9
Team Members-
1.Abhishrut Khanna(2017006)
2.Bhupesh Singh Kainth(2017040)
3.Pankaj Yadav(2017074)

INDRAPRASTHA INSTITUTE *of*
INFORMATION TECHNOLOGY
DELHI

Introduction

- In our project, our training model is predicting whether a diabetic patient will have to be readmitted to the hospital or not based on the treatment received by him/her.

- We are using Diabetes 130 US hospitals for years 1999-2008 dataset. The dataset has clinical data of 130 US hospitals across 10 years(1999-2008). It contains **50 features** and **1,01,766 instances** representing the patient and hospital outcomes.

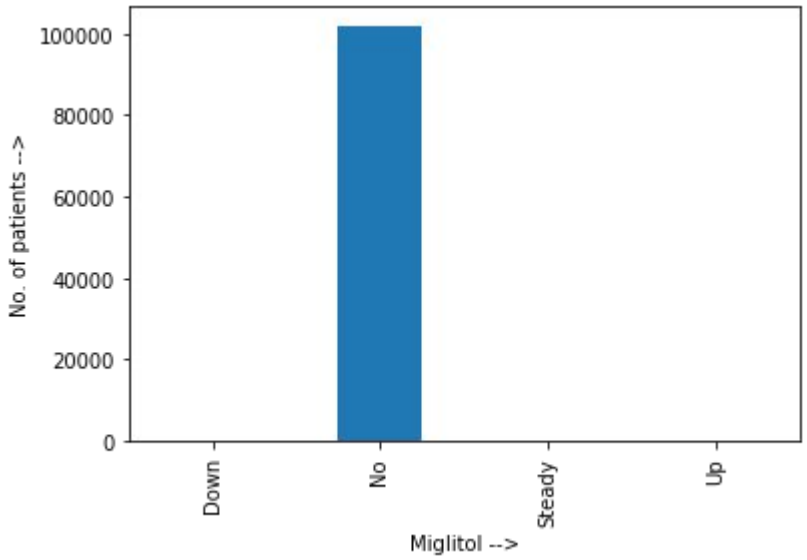
Related Work

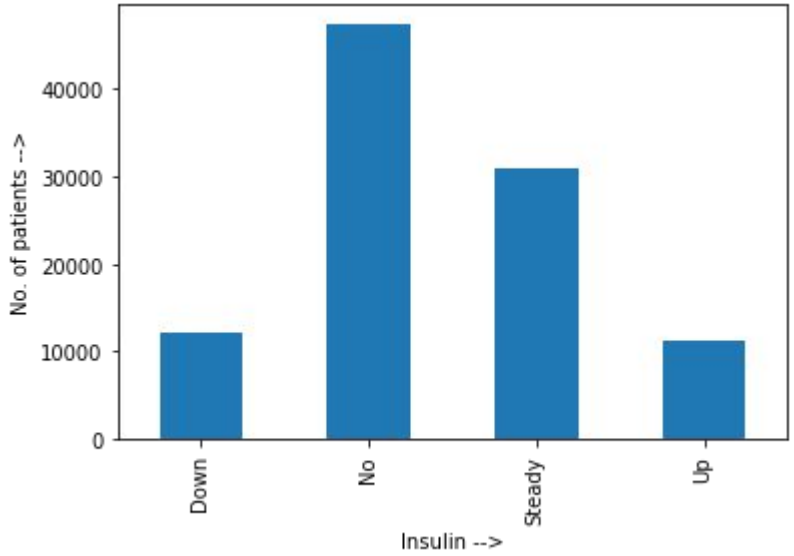
- The concept of predicting early readmission is of wide importance in USA. There have been few researches that used machine learning algorithms to predict readmission of a patient.
- Beata Strack(2014) used logistic regression models to assess the impact of features on readmission.
- C. Chopra, S. Sinha, S. Jaroli, A. Shukla, and S. Maheshwari(2017) used Recurrent Neural Network and produced an accuracy of 81.12%.
- A work by C.-Y.Lin, H.S.Singh, R.Kar and U. Raza(2018) conducted at Berkeley produced a better performance of 94%.
- A.Hammoudeh, G. Al-Naymat, I. Ghannam and N. Obied(2018) used Convolutional Neural Network to obtainthe state of art performance i.e. performance of 95 %.

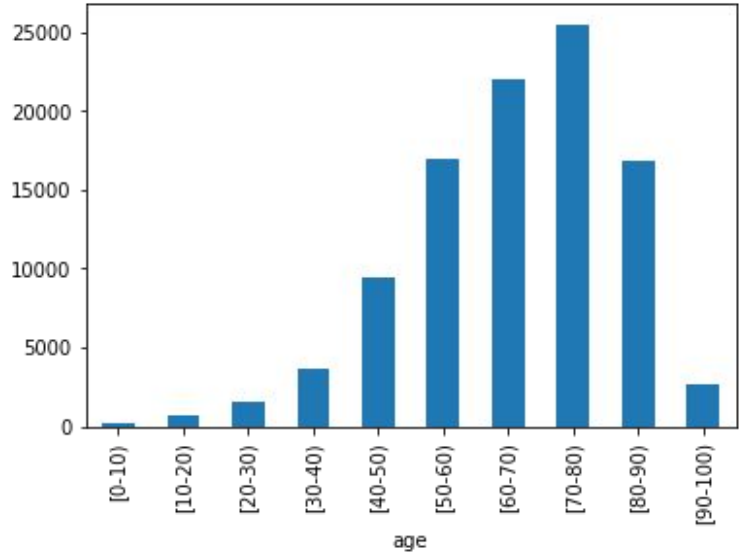
Preprocessing

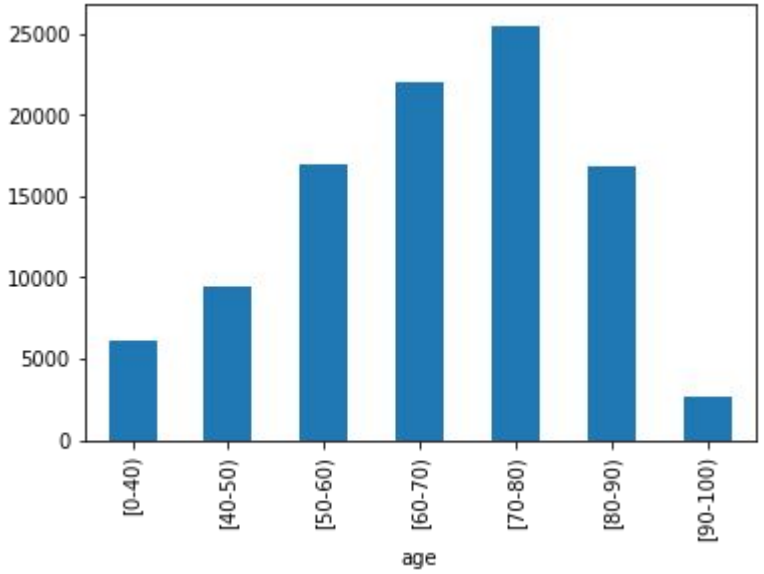
- When we analysed the data, the first challenge we faced is that the data was big and contained a lot of "**?**", **Unknown/Invalid**, and **uneven** values.
- We observed their value distribution and bar graphs and removed the redundant features to reduce the dataset to **99492 rows** and **22 features**.
- There were several columns with string values in the reduced dataset like race,age,gender,etc. So we replaced all such values to numeric values by appropriate conventions like 1 for Yes and 0 for No.
- After that we observed(from plot B) that the resulting dataset was non-linear and noisy and thus can't be separated linearly.
- We split the dataset into two parts-**training(80%)** and **testing(20%)** and applied **Logistic regression** and **SVM** models with different **kernels(rbf and poly)** and parameters(decision function shape, degree) on it.
- We observed the performance of the models using different metrics like **accuracy score, confusion matrix and classification report** .

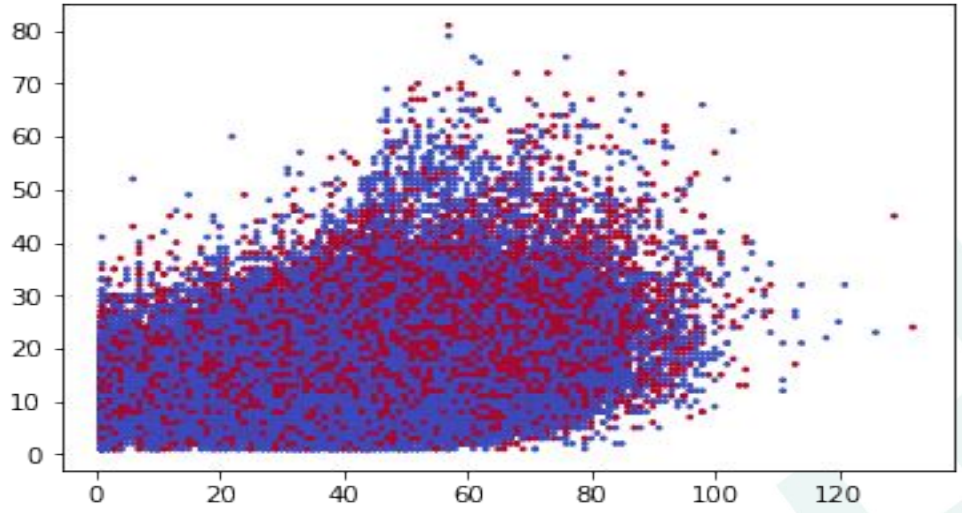
Plots


No. of patients -->
100000
80000
60000
40000
20000
0
Down
No
Steady
Up
Miglitol -->


No. of patients -->
40000
30000
20000
10000
0
Down
No
Steady
Up
Insulin -->


25000
20000
15000
10000
5000
0
[0-10)
[10-20)
[20-30)
[30-40)
[40-50)
[50-60)
[60-70)
[70-80)
[80-90)
[90-100)
age


25000
20000
15000
10000
5000
0
[0-40)
[40-50)
[50-60)
[60-70)
[70-80)
[80-90)
[90-100)
age


80
70
60
50
40
30
20
10
0
0
20
40
60
80
100
120

Logistic Regression & SVM

1. We trained our dataset using Logistic Regression and Support Vector Machine(SVM).
2. Since we ran SVM models on small datasets(10K training and 5K testing) only, we obtained the best accuracy to be **0.6315**, best kernel was **'poly'** with **degree 2** and best decision function shape was **'ovr'**, on the same dataset LR's accuracy was **0.628**.
3. We observed that though the accuracy of the LR model was slightly less than that of SVM but LR was a lot faster than SVM.
4. We observed the performance of the models using different metrics like **accuracy score, confusion matrix and classification report.**

A Comparison

SVM	Logistic Regression
1. The best accuracy obtained across all 5 folds was 0.63.	1. The best accuracy obtained across all the 5 folds was 0.6283.
2. The recall in the best fold was 0.55.	2. The best recall value across all the five folds was 0.5665.
3. The precision value in the best fold was 0.62.	3. The best precision value across all the five folds was 0.628.

Random Forest

1. We trained our dataset using the Random Forest Classifier
2. We used **80%** of the dataset for training and **20%** for testing.
3. Initially, we kept the **n_estimators** as **100** with a **random state** of **50** while keeping the rest of the parameters default.
4. We kept on increasing the value of **n_estimator** till **1000**.
5. Random forest classifier gave us an accuracy of **0.6153** with the n_estimator as **1000**.
6. It was also noted that above **1000**, the model resulted in almost the same accuracy. Even after increasing the number of estimators by 10 folds.

MLP

1. We trained our dataset using Multilayer Perceptron(MLP) using two different hidden layer sizes and numbers.
2. First, the number of **hidden layers** were **3** and **number of nodes** in each layer were **(256,128,64)**. The accuracy was printed for **tanh**, **relu**, **logistic(sigmoid)** and **linear activation functions**.
3. Then we changed the number of **hidden layers** to **4** and **number of nodes** in each layer as **(512,256,128,64)**.
4. Both the above model used stochastic gradient descent **(sgd)** as a solver.
5. **Learning rate** was **0.1** and **number of iterations** were **100.**

MLP Results

3 Hidden Layers Model

1. The best accuracy obtained was **0.8317(83.17%)**
2. Obtained using **sigmoid activation function**
3. Accuracy of **0.7590** was obtained using **tanh** and **0.7070** using **relu**

4 Hidden Layers Model

1. The best accuracy obtained was **0.708(70.8%)**
2. Obtained using **sigmoid activation function**
3. Accuracy of **0.7070** was obtained using **tanh** and **0.7071** using **relu**

KNN

1. We trained KNN model for our dataset.
2. Values of **K=5,50,100,200**
3. Accuracy obtained was ranged between **0.8205** to **0.9169**
4. The **highest accuracy** was obtained for **K=5**
5. Confusion Matrix was

$$\begin{bmatrix} 13557 & 494 \\ 1159 & 4689 \end{bmatrix}$$

6. **Increasing** the number of neighbours **decreased** the accuracy.

Conclusion

1. The best results were obtained using **KNN(0.9169)** with **5 neighbours** and **MLP Classifier(0.8317)** using **sigmoid activation function.**
2. These accuracies were significant improvements over the previously trained **Random Forest classifier**, **SVM** and **Logistic regression models**.
3. The classification reports for each trained model helped us in **improving** our accuracy and model selection.
4. Our trained model accuracies were close to that specified as the **state of the art model (0.95)**.