Diabetic Readmission Prediction

Problem Definition – We have Data from 200 hospitals in the US from the year 1998 to 2008. Our problem is we have to identify on the basis of data, whether after taking treatment in the hospital after how many days the patient will again get readmitted.

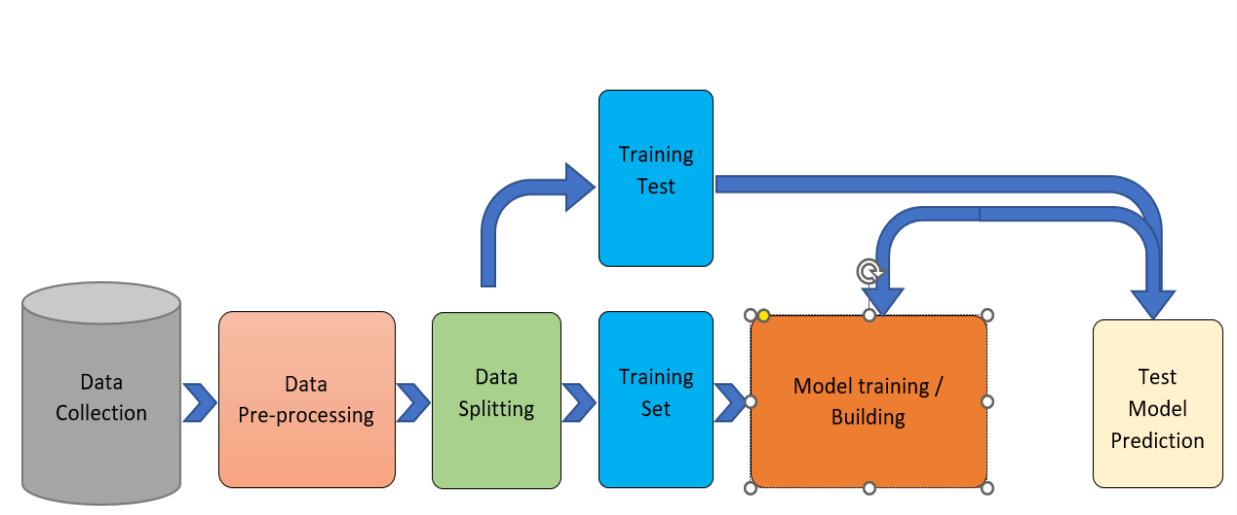
With this problem solution – we can get to know the severity of the disease; I mean by seeing his medical report and diagnosis.

We can help the healthcare industry by suggesting that the patient who follows has the disease, and how early the can patient recover, based on historical data throughout the world.

Dataset - The dataset represents ten years (1999-2008) of clinical care at 130 US hospitals and integrated delivery networks. Each row concerns hospital records of patients diagnosed with diabetes, who underwent laboratory, medications, and stayed up to 14 days. The goal is to determine the early readmission of the patient within 30 days of discharge.

Dataset Link - https://archive.ics.uci.edu/dataset/296/diabetes+130-us+hospitals+for+years+1999-2008

Flowchart –



Some Visualization representation of dataset-

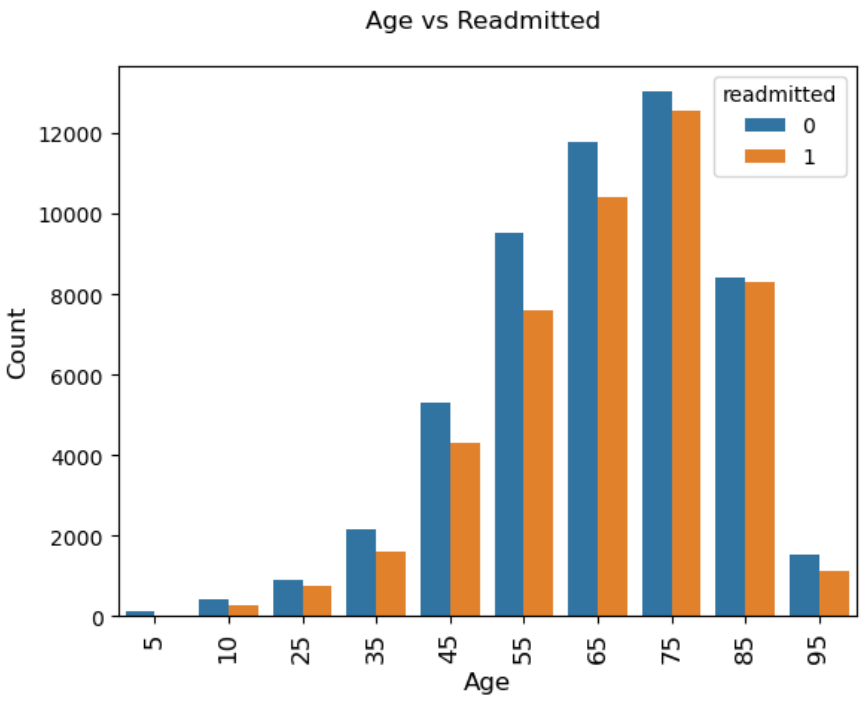


Fig -1 Distribution of Age with Readmission

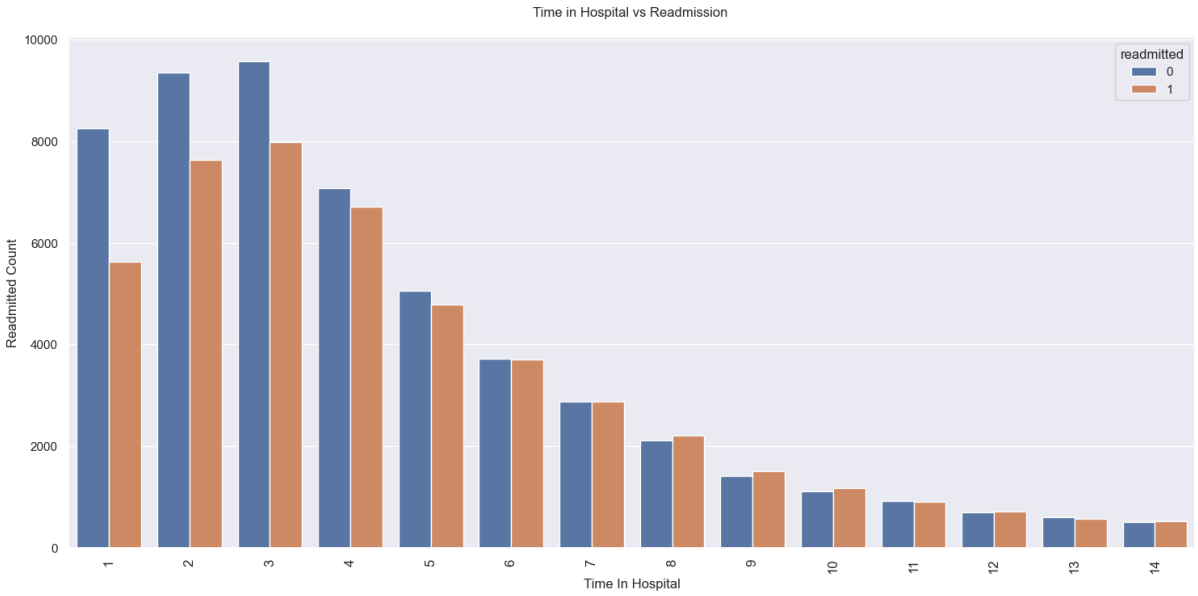


Fig -2 Time in Hospital vs Readmission

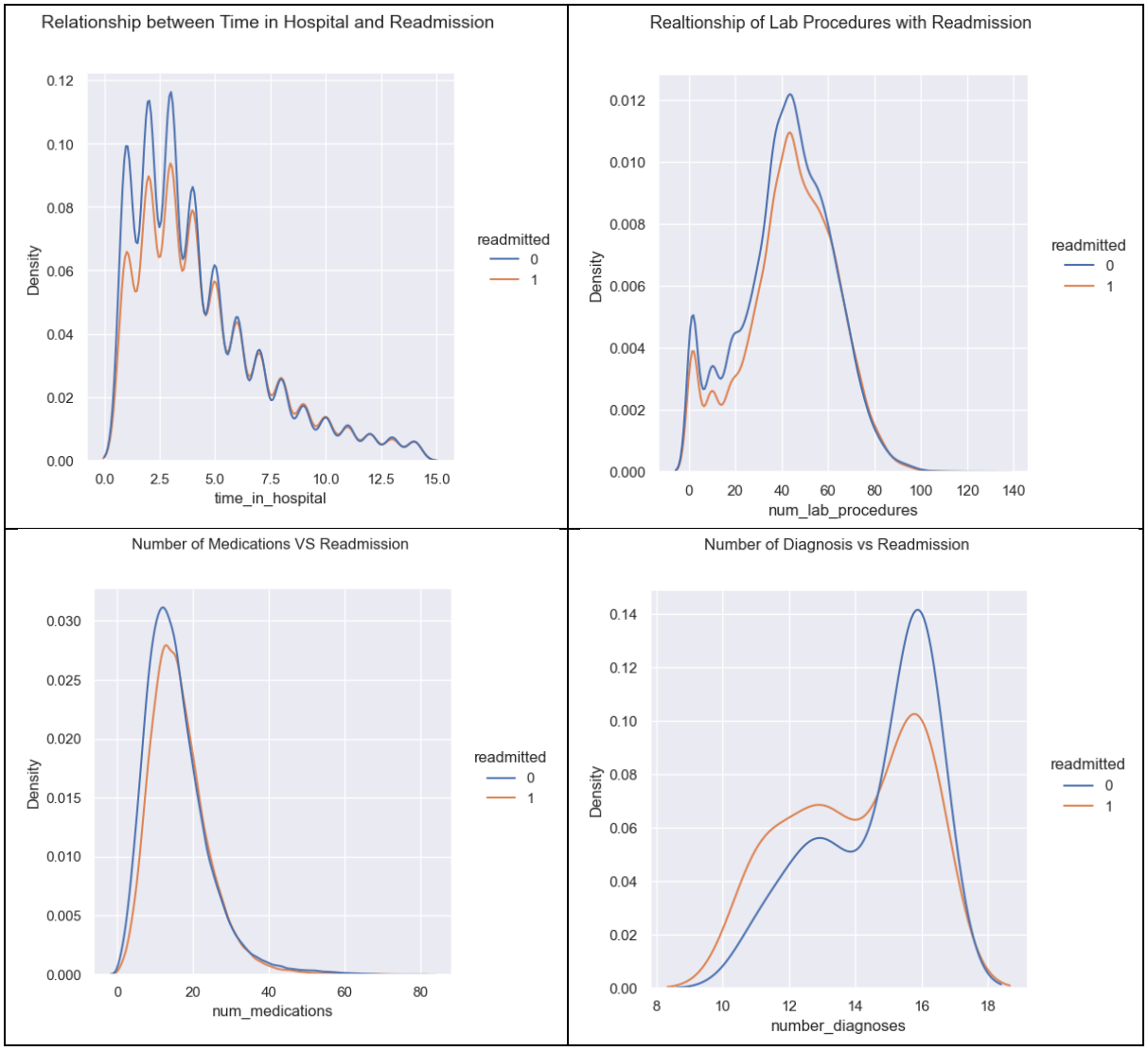


Fig -3 Comparing different aspect Vs Readmission

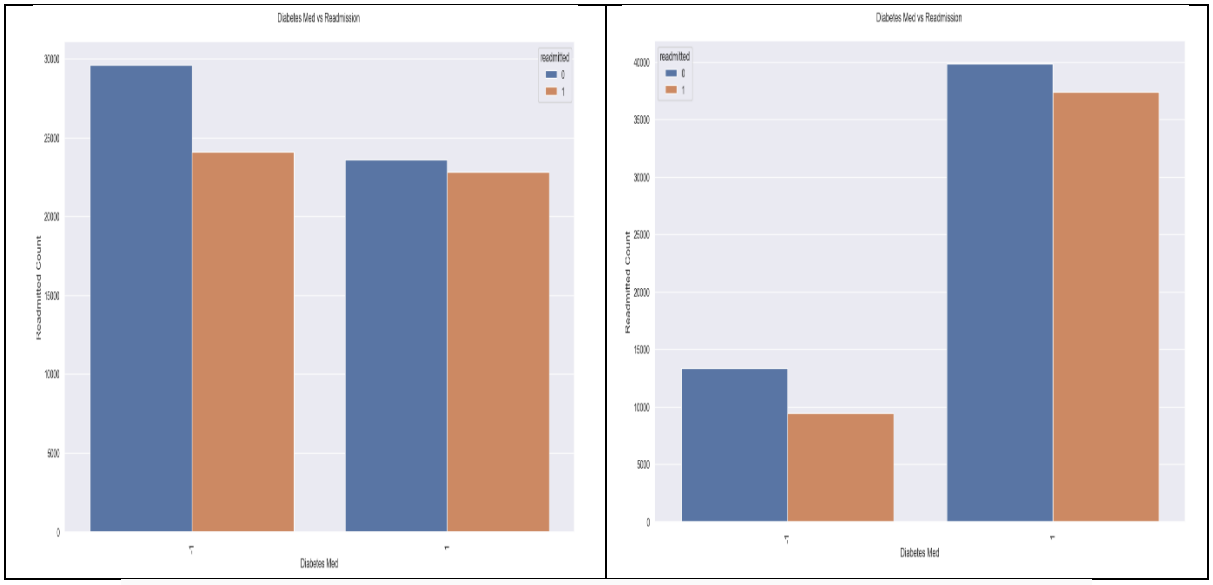


Fig -4 Change in Medicine and Diabetic Med vs Readmission

We have 20 columns about the drugs prescribed, this requires domain knowledge that's which are only essential medicine for the diabetic patient. So, we have performed logistic regression on those columns with our target variable. But, In the end, domain knowledge is most essential for these columns.

Data Preprocessing – The Dataset Contains 50 columns, out of 50 columns 37 columns were categorical data and 13 were numerical data.

·Data cleaning – some of the columns like weight, payer_code, medical_speciality contains more than 45% missing values. So, we have dropped these columns and there were some missing values in the race column.

- After getting some medical domain knowledge we have decided to combine some of the columns into one because it shows that the higher the value, the higher the risk is. Then the columns like time_in_hospital, num_procedures, num_medications, num_lab_procedures. We have combined these columns into the column severity_of_disease.

·After this in the age column, the age was in this format 0-10, 10-20, and so on. So, we have decided to replace the age column with label encoding.

·Then we have passed all the medicine-prescribed columns for data transformation using the lambda function.

Model Training and Selection – We have used ML algorithms like Random Forest Classifier, CatBoost, and Deep Learning Algorithm like RNN FNN, and LSTM.

We have used GridSearchCV to achieve better results.

Why are these algorithms for classification Problems?

Random Forest Classifier –

- Combines multiple decision trees, reducing overfitting and often achieving higher accuracy than single decision trees.
- Can effectively handle various data types (numerical, categorical) and complex relationships between features.
- Less susceptible to outliers and noise in the data, leading to more reliable predictions.
- Provides insights into feature importance through variable ranking, aiding in understanding the model's decision-making process.

CatBoost –

- Builds a sequence of decision trees iteratively, focusing on improving performance on previously misclassified examples, leading to high accuracy.
- Efficiently handles categorical features without the need for one-hot encoding, simplifying data preparation.
- Includes L1 and L2 regularization to prevent overfitting, improving model generalizability.
- Often achieves top performance in various classification tasks, making it a strong contender.

RNN -

- Specifically designed for handling sequential data like text, time series, or audio, capturing the order and relationships within the data.
- Can learn long-term dependencies within sequences, crucial for tasks like sentiment analysis in text or predicting future values in time series data.
- Adaptable to various architectures like LSTMs or GRUs, allowing for tailoring the model to specific needs.

FNN –

- Models' non-linear relationships between features, make them powerful for complex classification problems where linear methods might not suffice.
- Can automatically learn complex features from the data without explicit feature engineering, reducing manual effort.
- A fundamental architecture in deep learning with numerous applications across various domains.

LSTM -

- A specific type of RNN is designed to address the vanishing gradient problem, allowing it to effectively learn long-term dependencies in long sequences.
- Particularly effective for tasks like time series forecasting or language translation where capturing long-term relationships is crucial.
- Widely used in deep learning due to its ability to handle complex sequential data and long-term dependencies.

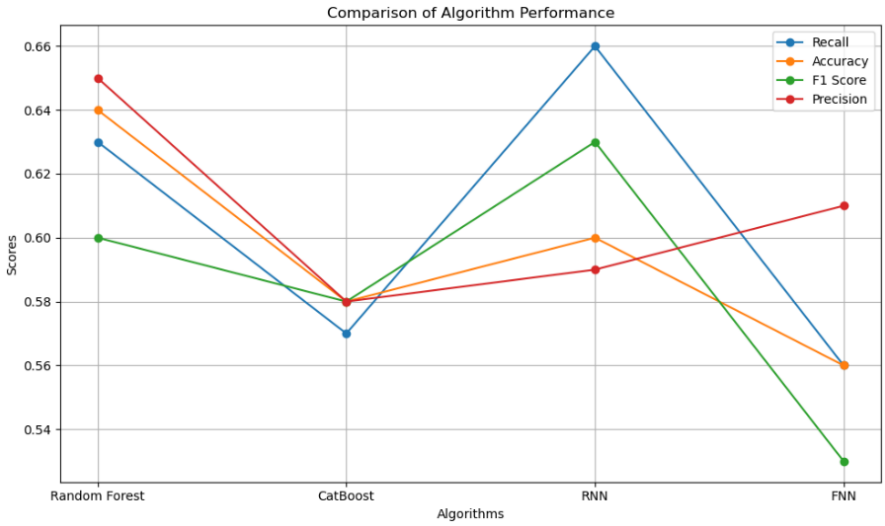
Evaluation Metrics used (Random Forest Classifier)

- Accuracy – 0.64
- Precision – 0.65
- F1- Score – 0.60
- Recall – 0.63

Accuracy suggests that the model performs moderately. Precision suggests that 0.65 positive values are guessed correctly by the model. Recall suggests that 0.63 positive values are guessed correctly among all positive values. F1-Score of 0.60 suggests that more balance is required between Precision and Recall.

For understanding of our model, we have used lime, to check how our model is performing locally.

Comparison of All Algorithms



For Better Accuracy and model performance –

We have trained 5 algorithms, now to increase the model capability we will combine the model using the Hybridisation technique and we will use stacking in it.

Challenges faced –

- One of the biggest challenges was to handle categories of categorical data. Then we have decided to is really all categories are really needed. Then We have set certain criteria to select the categories.
- Another challenge was with our target variable, the Percentage of not admitted was so much approximately 98%.
- Another, challenge we have faced is not having domain knowledge (medicine prescribed). So, based on some statistical methods we cannot guess whether this particular medicine is really useful or not.
- Finding the right parameter was a difficult task.

Conclusion –

Random Forest Classifier performs well as compared to deep learning algorithms. Random Forest's ensemble of decision trees excels in handling imbalanced datasets by naturally adapting to the distribution of the data without significant tuning. Its ability to handle high-dimensional data and mitigate overfitting, coupled with minimal hyperparameter tuning requirements, makes it an efficient and effective choice for such tasks.

Deep learning may have faced challenges in patient readmission prediction due to the dataset's size, high-class imbalance, and the complex nature of deep learning models. With over 100,000 records and imbalanced classes, deep learning models like MLPs, RNNs, and FNNs may struggle to generalize effectively, leading to overfitting and biased predictions. Additionally, the computational requirements and hyperparameter tuning complexity of deep learning pose practical limitations, especially in resource-constrained environments.

References –

- Stack machine learning models: Get better results

https://developer.ibm.com/articles/stack-machine-learning-models-get-better-results/

- What is Lime

https://www.ibm.com/topics/explainable-ai

https://towardsdatascience.com/lime-explain-machine-learning-predictions-af8f18189bfe

- Machine Learning in Health care

https://builtin.com/artificial-intelligence/machine-learning-healthcare

- Research Paper

https://www.researchgate.net/publication/332898304_Hospital_Readmission_Prediction_using_Machine_Learning_Techniques