**Project Report: Predictive Model for Readmission Risk in Diabetic Patients**

**Introduction:**

Mount Sinai is seeking an automated classification system to predict the likelihood of readmission for diabetic patients based on their medical history. This report outlines the development and evaluation of predictive models to address this need.

**Stakeholder and Problem Statement:**

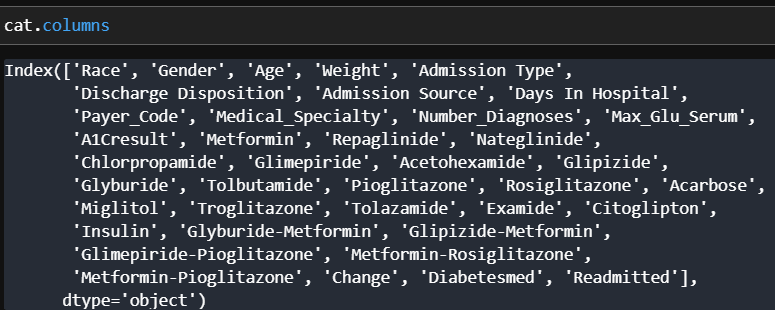
**Stakeholder:** Mount Sinai

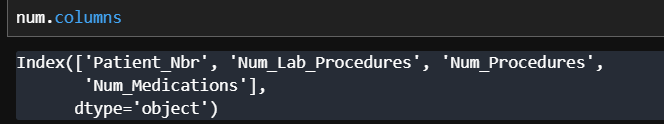
**Problem Statement:** Develop a predictive model to identify diabetic patients at risk of readmission based on patient data, clinical indicators, and contextual factors.

**Dataset:**

The dataset sourced from Kaggle contains various patient features, including demographic information, medical history, and treatment details. The target variable is "Readmitted," with two classes: "Yes" and "No." The dataset initially had 43 features, including 4 numeric and 39 categorical features. The target variable “Readmitted” has 2 classes “Yes” representing 70% of the dataset and “No” representing 30% of the dataset indicating imbalance in the data set. The dataset had around 36538 data of unique patients. Out of these features “Patient\_Nbr” is set as index as it’s the number the patient has been assigned.

Link: <https://www.kaggle.com/datasets/brandao/diabetes?rvi=1>





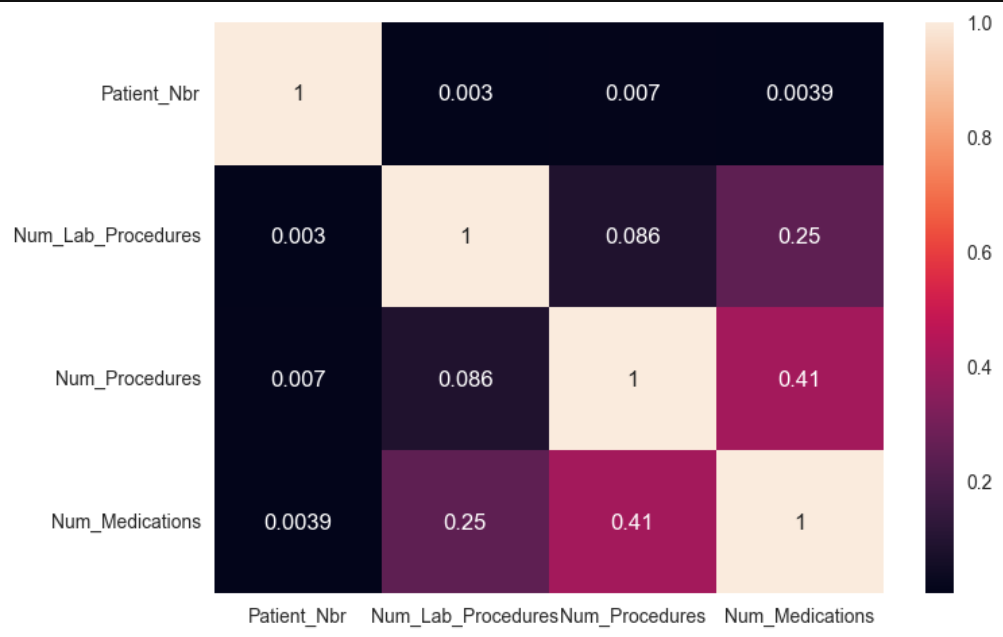
**Data Preprocessing:**

* **Handling Missing Values:**

Identified missing values in "Max\_Glu\_Serum" and "A1Cresult" columns, Maximum Glucose Serum Level and A1C (glycated hemoglobin) respectively had around 92% and 84%. Features including “Gender”, “Weight”, “Admission Type”, “Admission Source”, “Discharge Disposition”, “Payer\_Code”, “Race”, had Unknown/Invalid values. These values were converted to value in that feature with the highest frequency that is mode of the feature as they are categorical values.

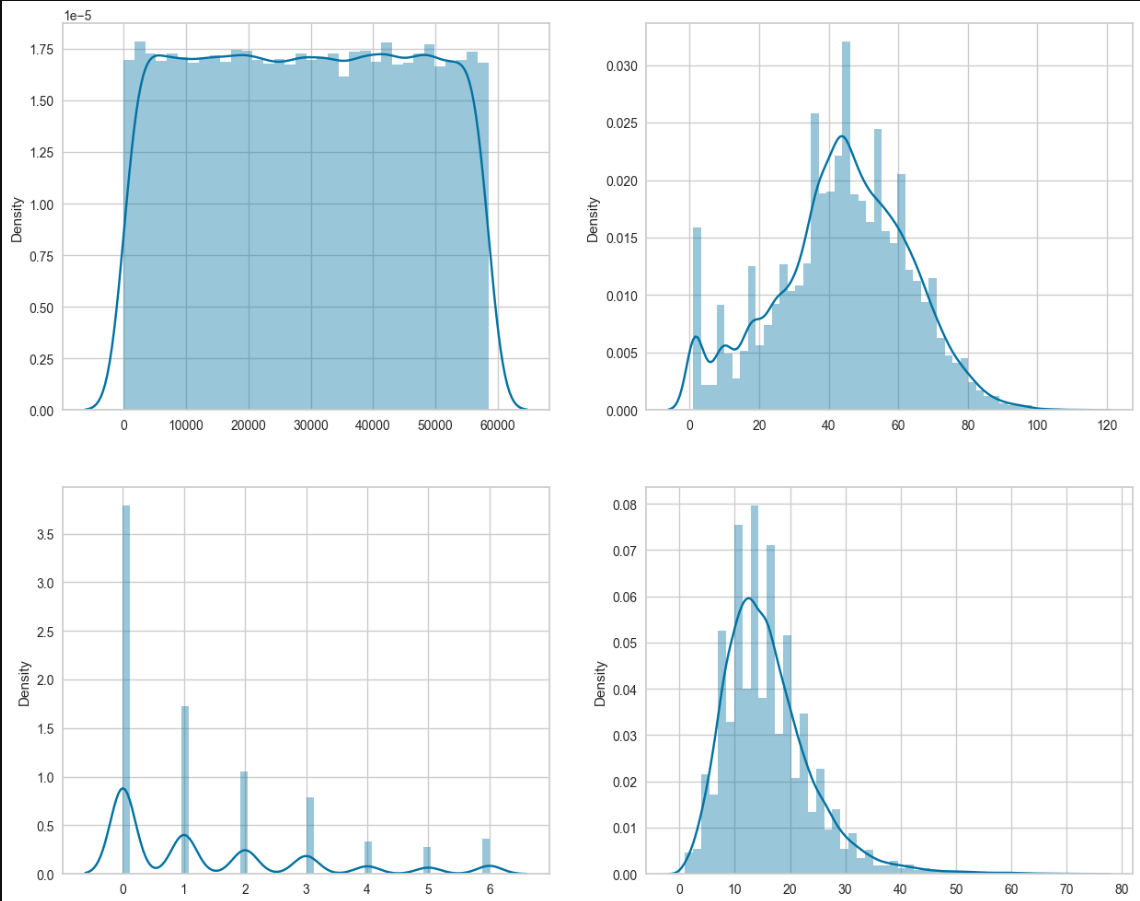
* **Correlation Test:**

When exploring heatmap I found out that “Num\_Medications” that is Number of Medications was causing multicollinearity in the dataset as it had mild influence in “Num\_Procedures” and “Num\_Lab\_Procedures” that is Number of procedures and number of laboratory procedure. This requires the removal multicollinearity as the model should be fit with independent variables.

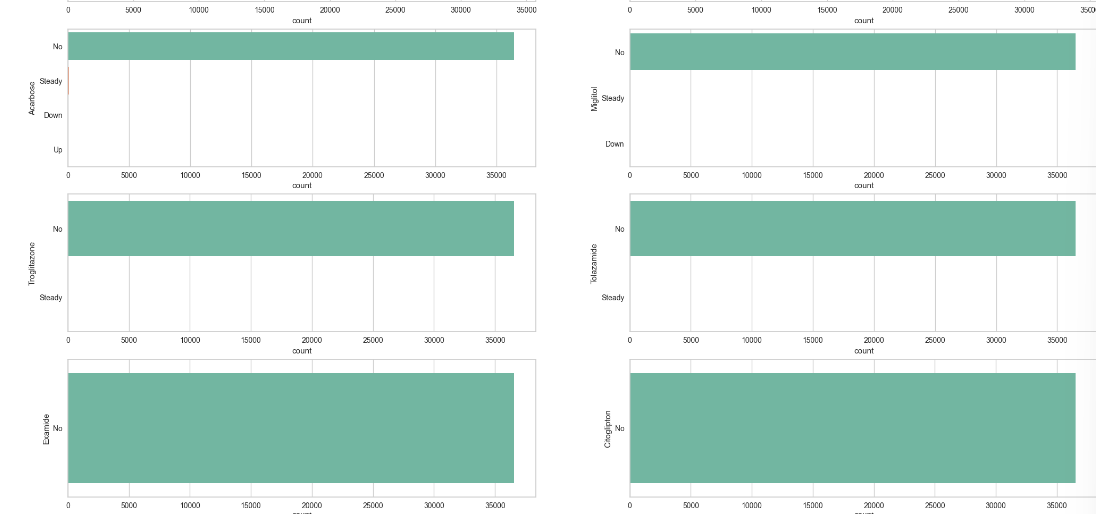


* **Histogram and Countplot:**

On Exploring the density/histogram of the variables we found that “Num\_Procedures” behaves like an ordinal variable and “Num\_Lab\_Procedures“ didn’t follow accurate normal distribution and “Num\_Medications” was right skewed.

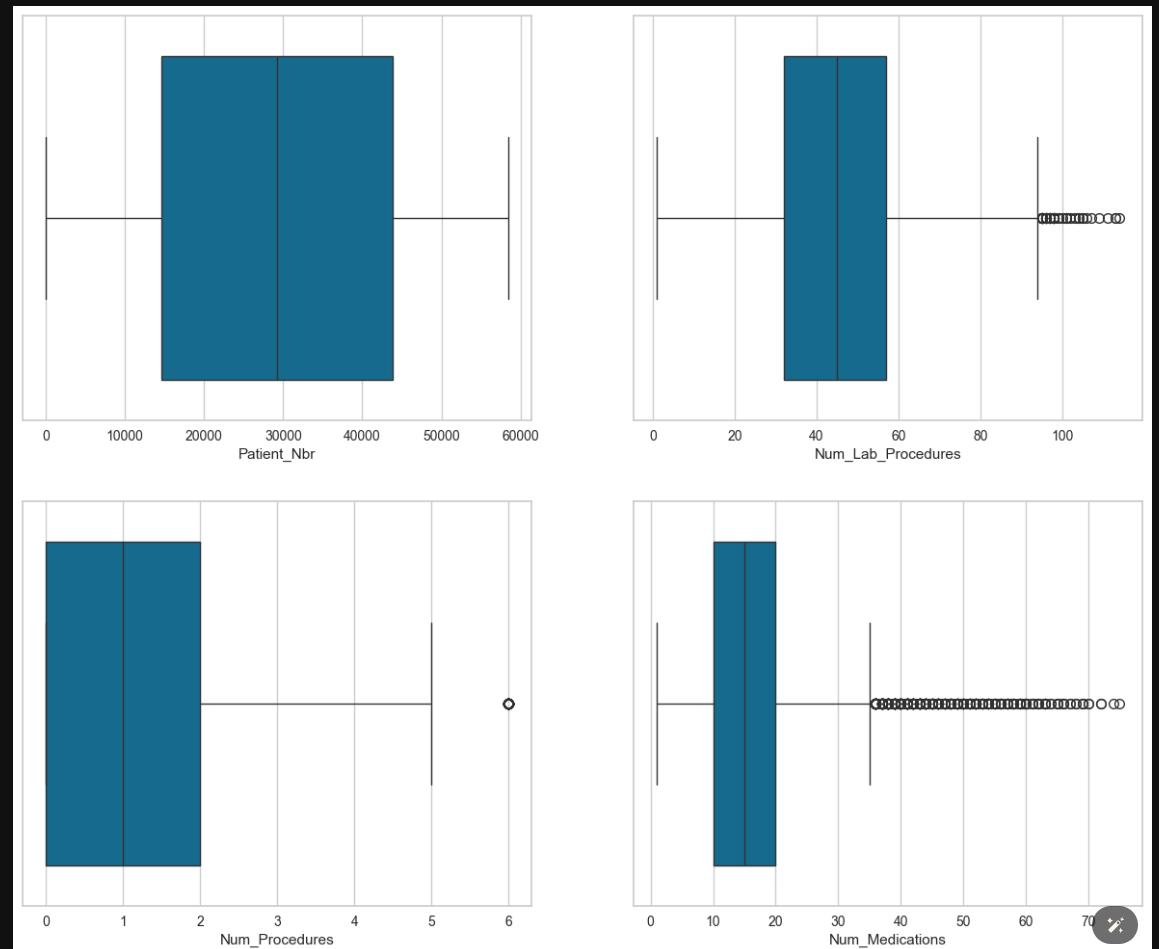


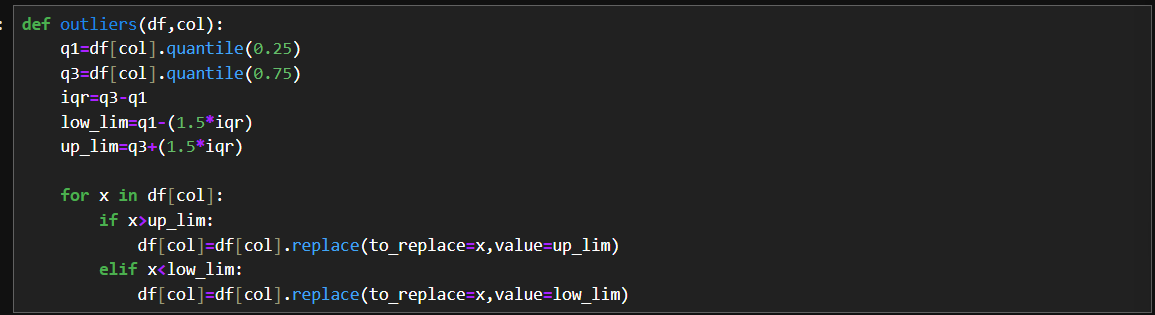
In the categorical variables, “Chlorpropamide”, “Acetohexamide”, “Tolbutamide”, “Troglitazone”, ”Acarbose”, “Miglitol”, “Tolazamide”, “Examide”, “Citoglipton”, “Glipizide-Metformin”, “Glimepiride-Pioglitazone”, “Metformin-Rosiglitazone”, “Metformin-Pioglitazone” all consisted majorly of one category that is “No”.



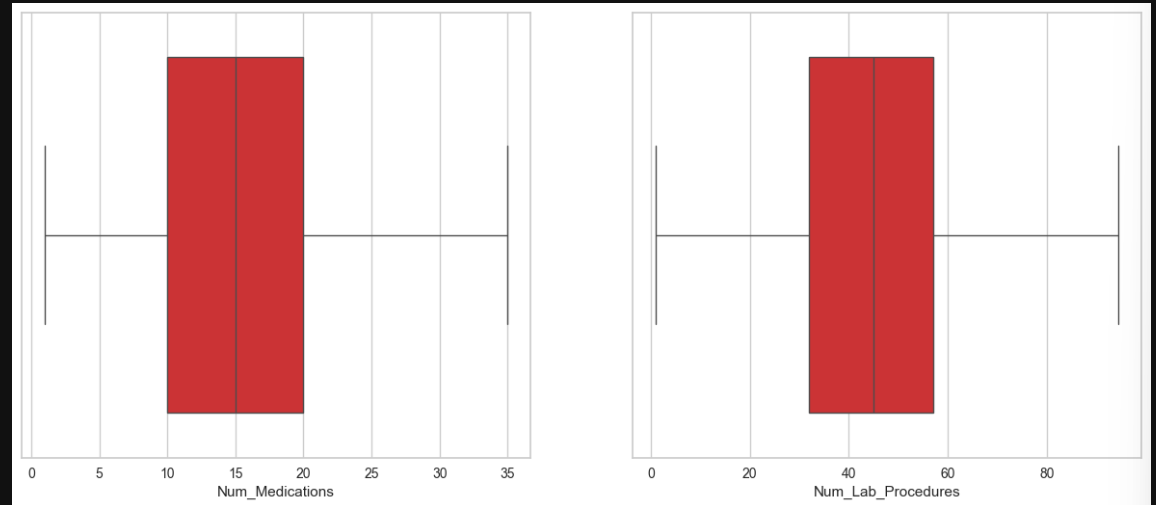
* **Outlier Treatment:**

Extreme values in numerical variables were clipped using the interquartile range.



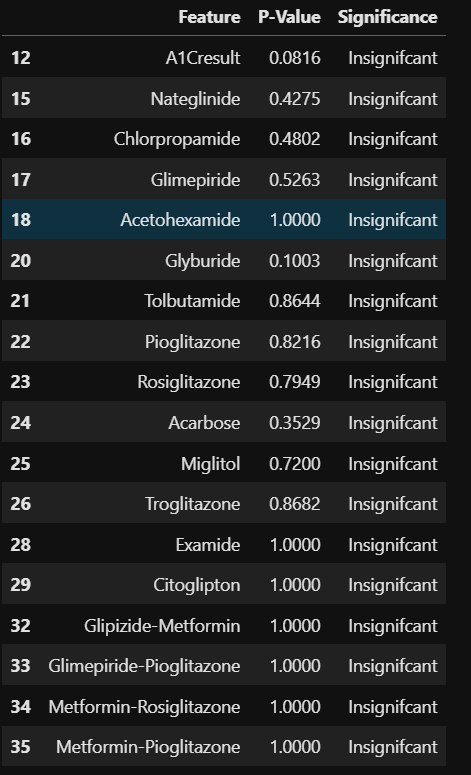


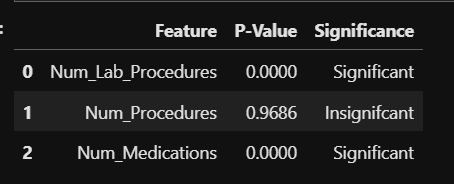
The Outliers were decided to be clipped so as to not lose the information from that specific datapoints. Using a function we clipped of the extreme values to upper and lower limit that is 75th Quantile plus 1.5 times the Inter-Quantile Range and 25th quantile minus 1.5 times the Inter-Quantile range respectively.



* **Feature Selection/Engineering:**

Chi-Square test was conducted for categorical variables where Chi-Square test is a Statistical test of independence and ANOVA was conducted for Numerical variables which tells if the Variable has an influence or not. If the p-value>0.05 the variable is insignificant and 18 categorical and 1 Numerical was considered as insignificant by the test and hence were removed.





* **Label Encoding:**

The categorical columns then were converted to numerical representation with the help of Label Encoder that encodes based on alphabetical order starting from 0. Finally, the data frame had 23 columns along with the target variable that had been completely preprocessed.

**Modelling:**

The Following Models were tried and also hyperparameter tuned to check if the the performance metrics improve or not.

1. **Logistic Regression:** Initial model due to its simplicity and interpretability. For Second Logistic Regression Model Insignificant variables including “Race”, “Gender”, “Admission\_Type”, “Days In Hospital”, “Glipzide”, “Glyburide-Metformin” were removed based on their test statistic and their p-value>0.05 and the model was built.
2. **Decision Tree:** Non-linear model to capture complex interactions. Then initially base decision tree was built that overfit and for tuning max\_features required and max\_depth the tree should grow, max\_leaf\_nodes to be present, and min\_samples\_leaf that is minimum samples required to split the node and produce leaf node were tuned so as to reduce overfitting.
3. **Random Forest:** Ensemble model for improved performance and robustness. Then a base RandomForest model was built to check its performance and it overfit the data. With same parameters of decision tree and max\_features required and number of estimators that is number of decision trees to be built as this can control overfitting of data were tuned and built to fit the dataset.
4. **K-Nearest Neighbors (KNN):** Instance-based learning for classification. Base KNN Model was built which had high degree of overfitting, hence grid search used to find the best distance measurement from Euclidean, Hamming, Manhattan and Chebyshev and number of neighbors were tuned using GridSearch.
5. **Gaussian Naïve Bayes:** Probabilistic model for classification. GridSearch was used to tune var\_smoothing that smooths the curve hence its tuned version.
6. **Support Vector Machine (SVM):** Effective for high-dimensional data and non-linear boundaries. Then base Support Vector Machine was built that had high degree of misclassification and GridSearch was carried out and ‘linear was selected as the best kernel out of linear, poly, radial bias function that can represent the hyperplane dividing the data points.
7. **AdaBoost:** Boosting ensemble method to combine weak learners. Next Base Adaboost were built which overfit the data and had high misclassifications. It was followed by their tuned versions with AdaBoost with RandomForest Forest tuned using GridSearch that has 350 estimators, max depth of the trees 8, max\_features 7 and out of box score samples as True, “Gini” as the Criterion, as the estimator to boost with 25 estimators to reduce overfitting and increase accuracy.
8. **XGBoost:** Gradient boosting algorithm known for its scalability and performance. Base XGBoost were built which overfit the data due to high estimators and had high misclassifications. Tuned XGBoost with 30 estimators, learning rate 0.1 and so that it reaches the optimum solution and no skip or oscillate without reaching the optimal solution, and max\_depth as 8 same as DecisionTree and RandomForest.
9. **Stacking Classifier:** Ensemble method combining multiple base classifiers. The best models were selected and Used to build the Stacking Classifier consisting of Adaboost Tuned with RandomForest estimator as Final estimator and XGBoost Models both Tuned and Base, Random Forest tuned with GridSearch, and Decision Tree model tuned using GridSearch as the Base estimators were used and their performance metrics were noted down.

**Model Evaluation:**

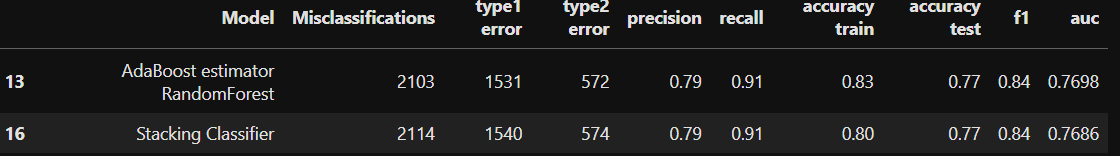
**Metrics Used:**

1. Misclassifications
2. Type 1 Error (False Positives)
3. Type 2 Error (False Negatives)
4. Precision
5. Recall
6. F1-score
7. Accuracy (train and test)
8. Area Under Curve (AUC) using ROC\_AUC score or Receiver Operating Characteristics

The models that were built were evaluated against number of Misclassifications, Type1 Error(False Positives) and Type2 Error(False Negatives), test and train accuracy, precision, recall, F1-score and Area Under Curve(AUC) calculated using ROC\_AUC score or Receiver Operating Characteristics. Accuracy for test and train sets were preliminary evaluation because of imbalance of the data. Hence Misclassifications which is the amount of false positives and false negatives in the predictions and Receiver Operating Characteristics score which tells us how well the model can distinguish between the both classes. The best model is the one that has high ROC\_AUC score and Low number of misclassifications as possible.

**Final Result:**

Once all the models were trained their evaluation metrics were recorded and compared out of all models Stacking Classifier with all the best performing model that is the AdaBoost Model, XGBoost Models both Tuned and Base, Random Forest tuned with GridSearch, and Decision Tree model tuned using GridSearch and AdaBoost Classifier with Random Forest tuned using GridSearch that has 350 estimators, max depth of the trees 8, max\_features 7 and out of box score samples as True, “Gini” as the Criterion, as the estimator to boost with 25 estimators. Both these models have Misclassifications close to 2114 for Stacking Classifier and 2103 for AdaBoost Classifier and 0.7686 and 0.7698 ROC\_AUC score respectively.



**Short-Comings and Future Implications of the Best Fit Models:**

Both models exhibit similar performance across various metrics. The AdaBoost model achieved slightly better results in terms of accuracy on the training set and AUC score. However, the Stacking Classifier demonstrated marginally lower misclassifications. Major Shortcomings of the Dataset includes Imbalanced Dataset, presence high volume of Unknown/Invalid Variables, High Degree of Dimensions that is having 23 columns.

In future, I would like balance the data as “Yes” representing 69.6% of the data and “No” representing 30.39% of the data using SMOTE by Up sampling or Down sampling the data. Removing all the Unlabeled data as Unknown/Invalid values were high. Try using Artificial Neural Networks to check if they can represent better. Use Ordinal Encoding so each representation has some meaning or importance instead of just numerical representation. Also try using feature selection methods such as forward selection, backward selection and Recursive Feature Elimination to select best combination of variables. Also try PCA to implement dimensionality reduction along with further hyperparameter tuning.

**Conclusion:**

I wouldn’t recommend the model to the stakeholder because the amount misclassification is about 23% for the best model that is 2103 out of 9135, the ROC\_AUC score is also around 0.7698 that is 77%, precision is around 0.79, recall around 0.91 with F1\_score of 0.84 and Test and Train Accuracy 0.77 and 0.83 Respectively. Overall, it seems to have relatively high recall, indicating good performance in capturing positive instances, but it also has notable misclassifications and a lower precision, suggesting room for improvement, especially in reducing false positives. Additionally, the accuracy on the testing dataset is slightly lower than on the training dataset, indicating some degree of overfitting.