INDEX

Table of Contents

List of figures and tables

CHAPTER 1

Introduction.

The main purpose of this Project is to facilitate healthcare institutions in predicting readmission of a diabetic patient by allowing the model to learn the relation among features and their importance in determining whether the patient will be readmitted or not. This helps the hospitals in providing the best inpatient treatment and improve the cost efficiency of healthcare centers. At the same time, it is important to identify the key factors responsible for the readmission of a diabetic patient. Hospital readmission is a crucial healthcare quality measure that helps in determining the level of quality of care that a hospital offers to a patient and has proven to be immensely expensive. The results are encouraging with Patients having changes in medication while admitted having a high chance of getting readmitted. Identifying Prospective patients for readmission could help the hospital systems in improving their inpatient care, thereby saving them from unnecessary expenditures. Diabetes is one of the chronic non-communicable diseases that are on the rise with massive urbanization and a drastic change of lifestyle in many countries. It is expected to turn into the seventh most prevalent mortality factor by 2030 and millions of deaths could be prevented each year through better analytics. When assessing the quality of care delivered by a health center, readmission is the metric of choice. It measures the number of patients that need to come back to the hospital after their initial discharge. The readmission can be classified into three broad categories such as unavoidable, planned, and unplanned. The unavoidable readmission that is highly predictable mostly due to the nature of the pathology or patient's condition (i.e. cancer phase IV, metastasis). Secondly in the planned readmission which is directly prescribed by the healthcare professional to the patient (i.e. check-up, transfusion). Lastly, the unplanned is defined as readmission that shouldn't have happened given the practitioner's diagnosis and could have been avoided if proper care was given to the patient post-discharge. Unavoidable and planned readmissions already are highly anticipated. However, predicting unplanned readmission is of prime interest due to its inherent uncertainty.

In this project we will demonstrate how to build a model predicting readmission in Python using the following Steps:

- ❖ Data exploration

- Data Preprocessing
- Feature engineering
- Building training/validation/test samples
- Model selection
- Model evaluation

1.1 Purpose:

In our research, 10 years (1999-2008) of clinical care data from 130 hospitals and 70,000 diabetic inpatients in the US was used for data mining and analytics for two objectives:

(1) This project aims to predict whether a diabetes patient will be readmitted within 30 days of discharge. Different machine learning models and ensemble methods are applied to train this data.

(2) To extract critical risk factors that correlates with readmission of diabetic patients.

1.2 Scope of the project:

The beneficiaries of this project are twofold, the patient themselves who will benefit in terms of disease management, overall health and early detection. The health service providers will gain, they will have a better understanding of the data where action can be taken to reduce early readmissions associated with the patient diagnosis. Early detection and treatment are essential in order to provide Better treatment to patients and potentially saving lives and reducing readmitted patients treatment healthcare costs.

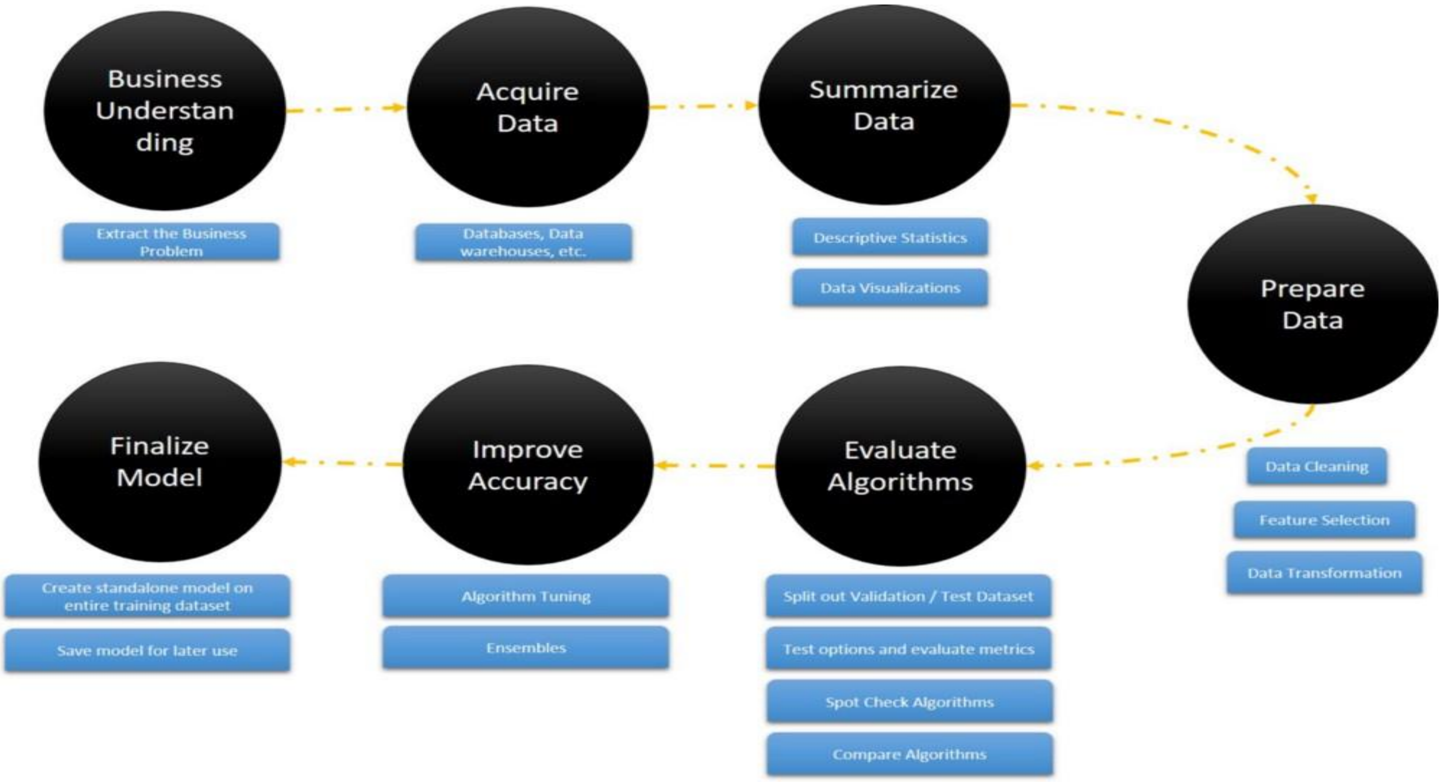


Figure 1 Data Science Project Life Cycle

CHAPTER 2
Data Exploration.

2.1 Dataset:

For our analysis, we have picked up a publicly available dataset from UCI Machine Learning repository2. It covers data on diabetes patients across U.S. hospitals during a 10-year period from 1999 to 2008. There are 101,766 unique hospital admissions in the dataset from approximately 70,000 unique patients. The dataset is spread over 50 features including patient characteristics, conditions, tests and 23 medications.

2.2 Characteristics of the Dataset:

1) All encounters are hospital admissions.

2) Only diabetic encounters are included (at least one of the three primary diagnosis was diabetes)

3) The patient stayed in the hospital for between 1 and 14 days

4) Laboratory tests were performed on the patient

5) Some form of medication was given to the patient during the stay at the hospital

2.3 Dataset Description:

Range Index: **101766 entries**, Data columns (**total 50 columns**):

- **Encounter ID** Unique identifier of an encounter

- **Patient number** Unique identifier of a patient

- **Race** Values: Caucasian, Asian, African American, Hispanic, and other

- **Gender** Values: male, female, and 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, newborn, 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
- **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** Numeric 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 first three digits of ICD9); 848 distinct values
- **Diagnosis 2** Secondary diagnosis (coded as 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 to the system 0%
- **Glucose serum test result:** 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 different kind of medical drugs.
- **Readmitted** Days to inpatient readmission. Values: “❤0” 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

2.4 Performance Metrics:

Our task is a classification problem so we can use performance metrics like *precision, recall, Accuracy* and *F1-score*.

1) Precision :

Precision is (TP/TP+FP) where TP True Positive and FP is False Positive We can think of precision as out of all points that are predicted as positive points by model how many of them are indeed positive points. Precision is a good measure when false positive cost is high which is the case here in our case study. For a patient which doesn't need readmission if our model predicts that the patient needs readmission that is false positive then the hospital will keep that patient in the hospital and that increases hospitalization cost.

2) Recall :

Recall is (TP/TP + FN) where TP is True Positive and FN is False Negative We can think of Recall as out of all points that are actually positive how many of them are predicted to be positive by model. Recall is used when False Negative cost is high that is indeed the case here. For the patient which needs readmission if the model predicts that it doesn't then the hospital will discharge him but the patient will eventually readmit again and that increases the cost.

3) Harmonic F1-Score :

As from above we know that False Negative cost and False Positive cost both are important for us so it would be good if we have a measure which combines both. F1-score does the same for us; it combines Recall and Precision into single equation.

*F1_score = (2 * Precision * Recall) / (Precision + Recall)*

4) AUC : Normal precision and recall are calculated using single threshold. This single threshold might not classify all points correctly. AUC is nothing but area under the ROC curve, ROC curve is drawn by calculating FPR and TPR and putting them on x and y axis respectively. So AUC takes all possible thresholds while calculating FPR and TPR lists and hence it reveals us the real power of the model.

5) Confusion Matrix: Confusion Matrix as the name suggests gives us a matrix as output and describes the complete performance of the model.

CHAPTER 3
Data Preparation.

3.1 Data Pre-processing:

Data Pre-processing is done in followings steps as follows:

(1) Filling missing values: The missing values in the categorical columns are replaced by "nan" which is treated as another category. On the other hand, for filling the missing data in continuous-valued columns, we experimented with three commonly used techniques. Replacing missing data with: • Average of the column • Median of the column • Constant value of 0. Out of these three, the best results were obtained by filling with a median of the column since it is the best representative of the distribution of data.

(2) Removing Outliers and Inconsistencies: It is important to maintain a single record for every patient id. So continuing along the path of this research, only the first encounter with the patient is kept, and the rest of the patient's records are dropped.

(3) Feature Encoding: The dataset contains three classes, with 11.2% of the 70,000 patients readmitted within 30 days, 34.9% readmitted after 30 days, and 53.9% are not readmitted at all. Since we have to predict whether a patient will be readmitted within 30 days or not, dropping patients readmitted after 30 days would result in the loss of one-third of data. Therefore, we define two classes, Yes and No, Yes suggesting that the patient will be readmitted within 30 days else No.

(4) Normalizing continuous variables: Many columns such as "number inpatient" ,"number outpatient", "number emergency" were highly skewed with high kurtosis. To reduce the skewness, log(x+1) transformation was used. The features in the dataset vary in scale, unit, and range. Due to this, the features with a broad range in the dataset can have a disproportionate impact on the prediction. To overcome this problem, the data is normalized so that each feature has a mean of 0 and a standard deviation of 1.

3.2 Feature Engineering:

In this section, we will create features for our predictive model. For each section, we will add new variables to the Data-frame and then keep track of which columns of the Data-frame we want to use as part of the predictive model features. We will break down this section into numerical features, categorical features and extra features.

Through this process we created 143 features for the machine learning model. The break-down of the features is:

- 8 numerical features
- 133 categorical features
- 2 extra features

Here is a final features list that we are going to keep to build the models and to perform Exploratory data analysis :

'race', 'gender', 'ages', 'admission', 'discharge', 'admsource', 'time in hospital', 'payer code', 'num lab procedures', 'num procedures', 'num medications', 'number outpatient', 'number emergency', 'number inpatient', 'diag1', 'diag2' 'number diagnoses', 'max glu serum', 'A1Cresult', 'insulin', 'change', 'diabetesMed'

3.3 Data Modelling:

After feature engineering came the process of developing a predictive model. A preferred metric to optimize had to be selected for the model. The objective was to predict whether or not a given patient would be readmitted to the hospital within 30 days. In the data, this was true of approximately 10% of patients. Because of the considerable imbalance in outcomes, accuracy is not the ideal metric to use. For highly imbalanced classification outcomes, the highest accuracy is often achieved by the model simply guessing the more likely outcome for every observation, which in this case would yield an accuracy of 90%, but would have no predictive value. Instead, for imbalanced classification problems, it is often preferred to judge model performance by scoring the predicted probabilities of the model. A number of metrics can be used to measure predicted probabilities, including F1 Score, Briar Score, and Area Under the Curve (AUC). We settled on AUC as our metric of choice as it is commonly employed in the healthcare industry for classification scoring.

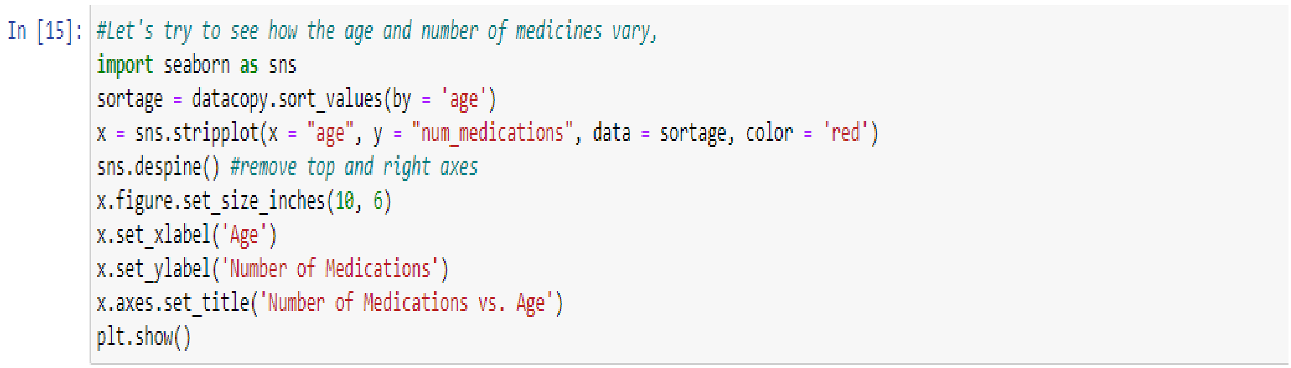
After the above processes are completed the data is further randomly split into 70% training data and 30% testing data for implementing some machine-learning models on it to achieve the objective.

Our objective of modeling is further developed to improve the prediction of readmission (True Positive) while balancing the number of False Positive and False Negative.

CHAPTER 4
Exploratory Data Analysis.

Critical Factors were taken into account and following EDA was carried out.

(1) Between Age and Number of medicines.



```
In [15]: #Let's try to see how the age and number of medicines vary,
         import seaborn as sns
         sortage = datacopy.sort_values(by = 'age')
         x = sns.stripplot(x = "age", y = "num_medications", data = sortage, color = 'red')
         sns.despine() #remove top and right axes
         x.figure.set_size_inches(10, 6)
         x.set_xlabel('Age')
         x.set_ylabel('Number of Medications')
         x.axes.set_title('Number of Medications vs. Age')
         plt.show()
```

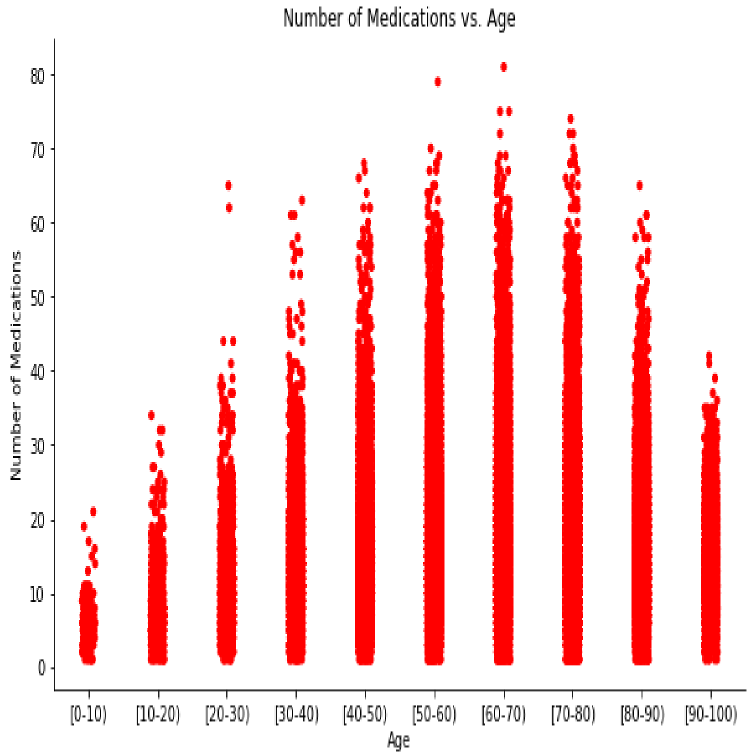
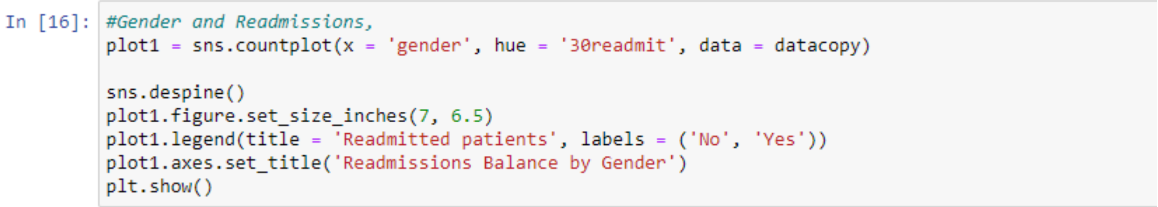


Figure 4.1 EDA on Age & Number of medicines.

(2) Between gender & Readmissions.



```
In [16]: #Gender and Readmissions,
         plot1 = sns.countplot(x = 'gender', hue = '30readmit', data = datacopy)

         sns.despine()
         plot1.figure.set_size_inches(7, 6.5)
         plot1.legend(title = 'Readmitted patients', labels = ('No', 'Yes'))
         plot1.axes.set_title('Readmissions Balance by Gender')
         plt.show()
```

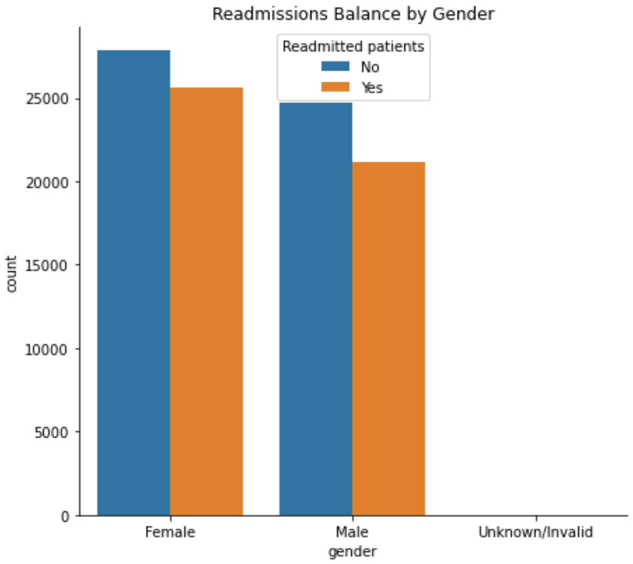
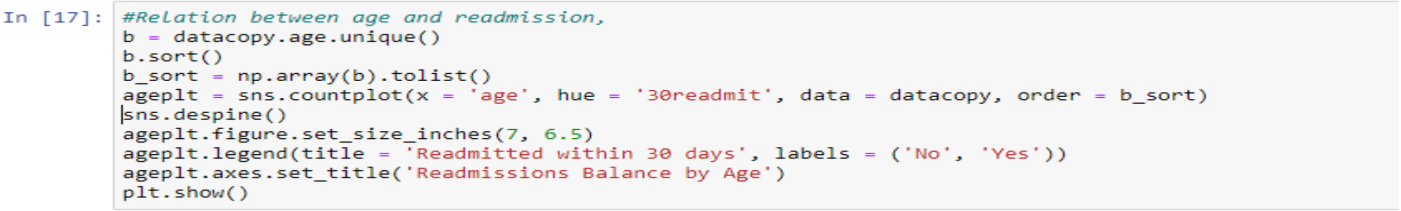


Figure 4.2 EDA on gender & Readmissions.

(3) Between Age & Readmission.



```
In [17]: #Relation between age and readmission,
         b = datacopy.age.unique()
         b.sort()
         b_sort = np.array(b).tolist()
         ageplt = sns.countplot(x = 'age', hue = '30readmit', data = datacopy, order = b_sort)
         sns.despine()
         ageplt.figure.set_size_inches(7, 6.5)
         ageplt.legend(title = 'Readmitted within 30 days', labels = ('No', 'Yes'))
         ageplt.axes.set_title('Readmissions Balance by Age')
         plt.show()
```

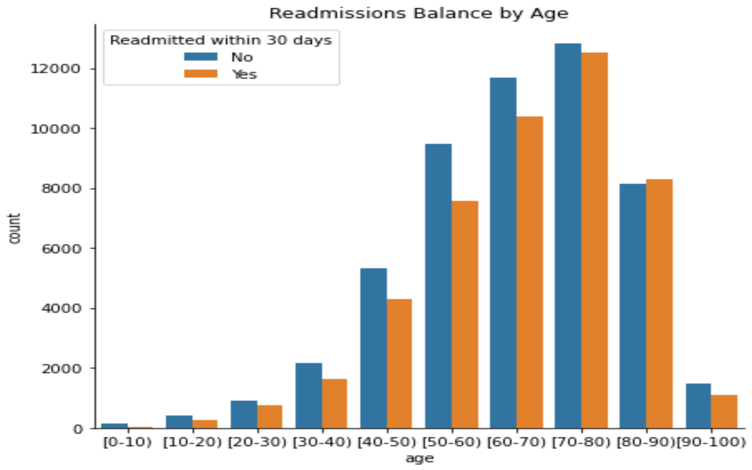


Figure 4.3 EDA on Age & Readmission

(4) Plotting the Numerical Values in our Dataset :

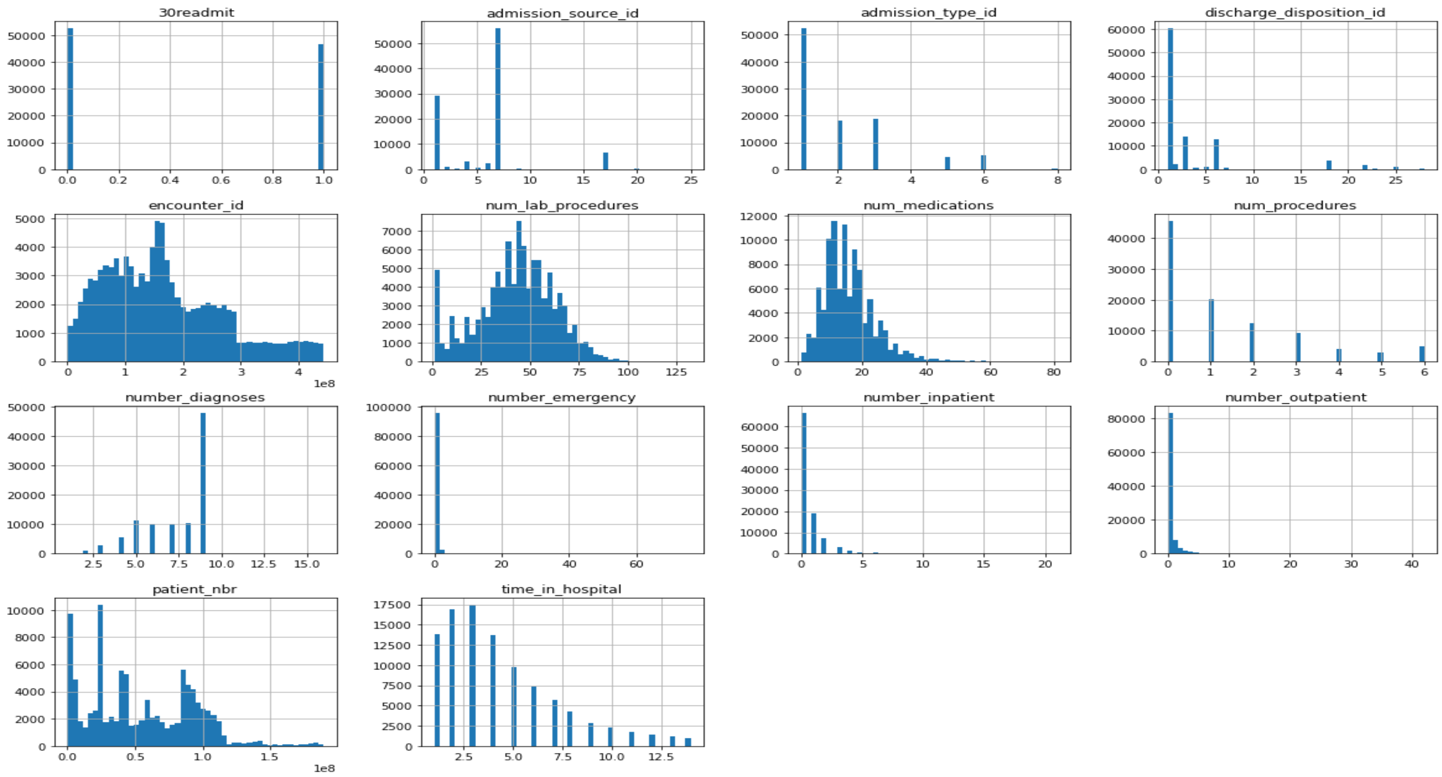


Figure 4.4 Plotting the Numerical values in the Dataset.

(5) Exploring the categorical values in the Dataset.

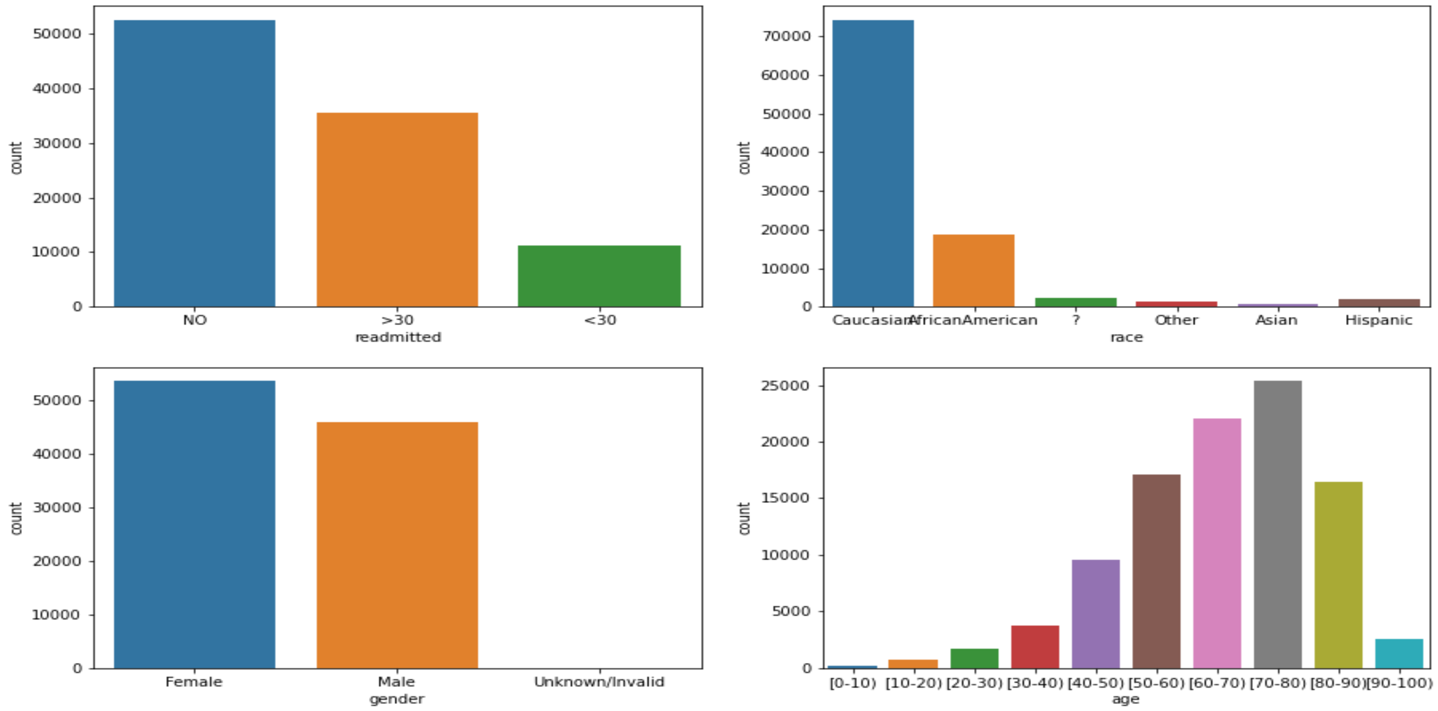


Figure 4.5 Plotting the categorical Values in the dataset.

CHAPTER 5

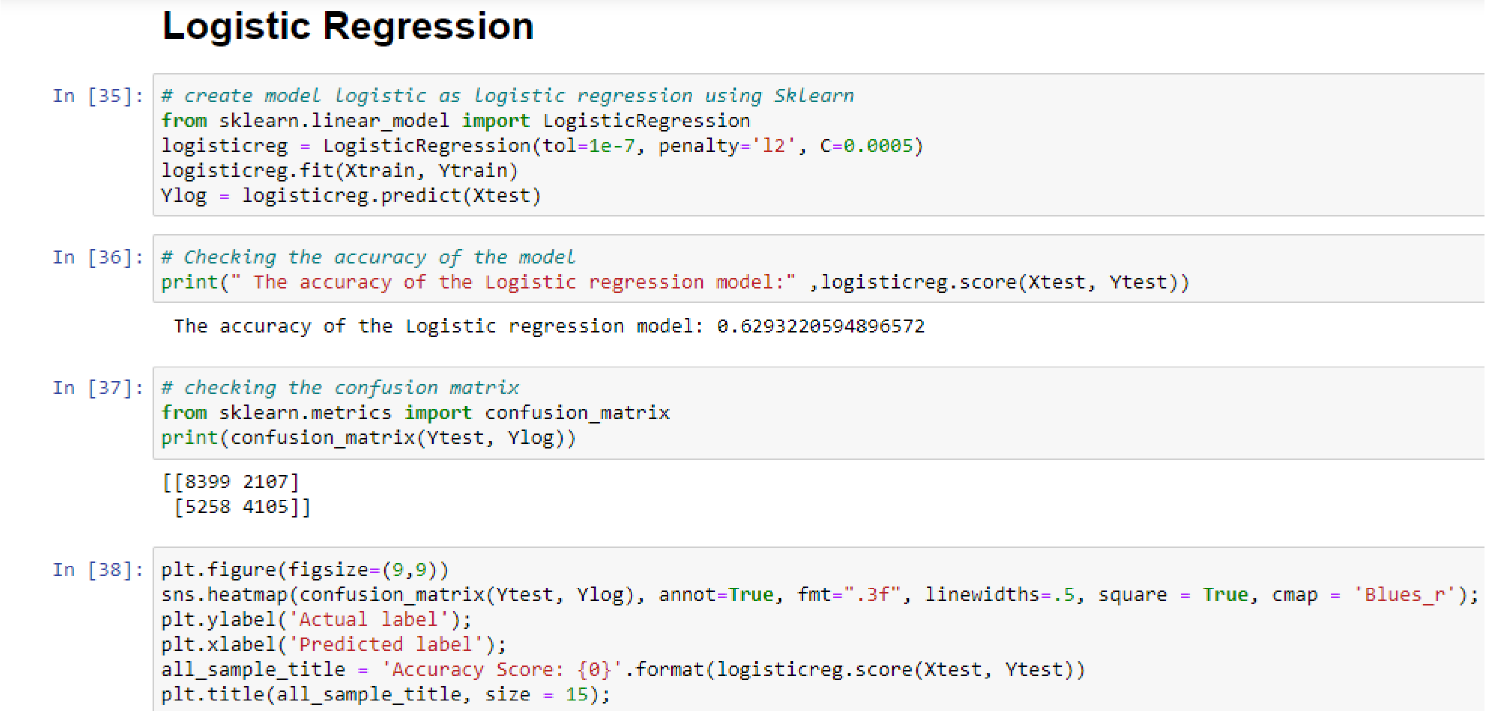
Model Selection.

For this we Split the data into training and validation data sets. The training data will contain 80 % of the data and validation will contain remaining 20%. After developing and validating prediction model with training and validation data sets, the prediction algorithm was applied to testing sets.
We will then select the best model based on performance on the validation set.

For this, we will first compare the performance of the following 4 machine learning models using default Hyper-parameters :

(1) Logistic regression :

Logistic regression is a traditional machine learning model that fits a linear decision boundary between the positive and negative samples. This linear function is then passed through a sigmoid function to calculate the probability of the positive class. Logistic regression is an excellent model to use when the features are linearly separable. One advantage of logistic regression is the model is interpretable — i.e. we know which features are important for predicting positive or negative.



Logistic Regression

```
In [35]: # create model logistic as logistic regression using Sklearn
         from sklearn.linear_model import LogisticRegression
         logisticreg = LogisticRegression(tol=1e-7, penalty='l2', C=0.0005)
         logisticreg.fit(Xtrain, Ytrain)
         Ylog = logisticreg.predict(Xtest)
```

```
In [36]: # Checking the accuracy of the model
         print(" The accuracy of the Logistic regression model:" ,logisticreg.score(Xtest, Ytest))

          The accuracy of the Logistic regression model: 0.6293220594896572
```

```
In [37]: # checking the confusion matrix
         from sklearn.metrics import confusion_matrix
         print(confusion_matrix(Ytest, Ylog))

         [[8399 2107]
          [5258 4105]]
```

```
In [38]: plt.figure(figsize=(9,9))
         sns.heatmap(confusion_matrix(Ytest, Ylog), annot=True, fmt=".3f", linewidths=.5, square = True, cmap = 'Blues_r');
         plt.ylabel('Actual label');
         plt.xlabel('Predicted label');
         all_sample_title = 'Accuracy Score: {0}'.format(logisticreg.score(Xtest, Ytest))
         plt.title(all_sample_title, size = 15);
```

Figure 5.1 Applying Logistic Regression on the Dataset.

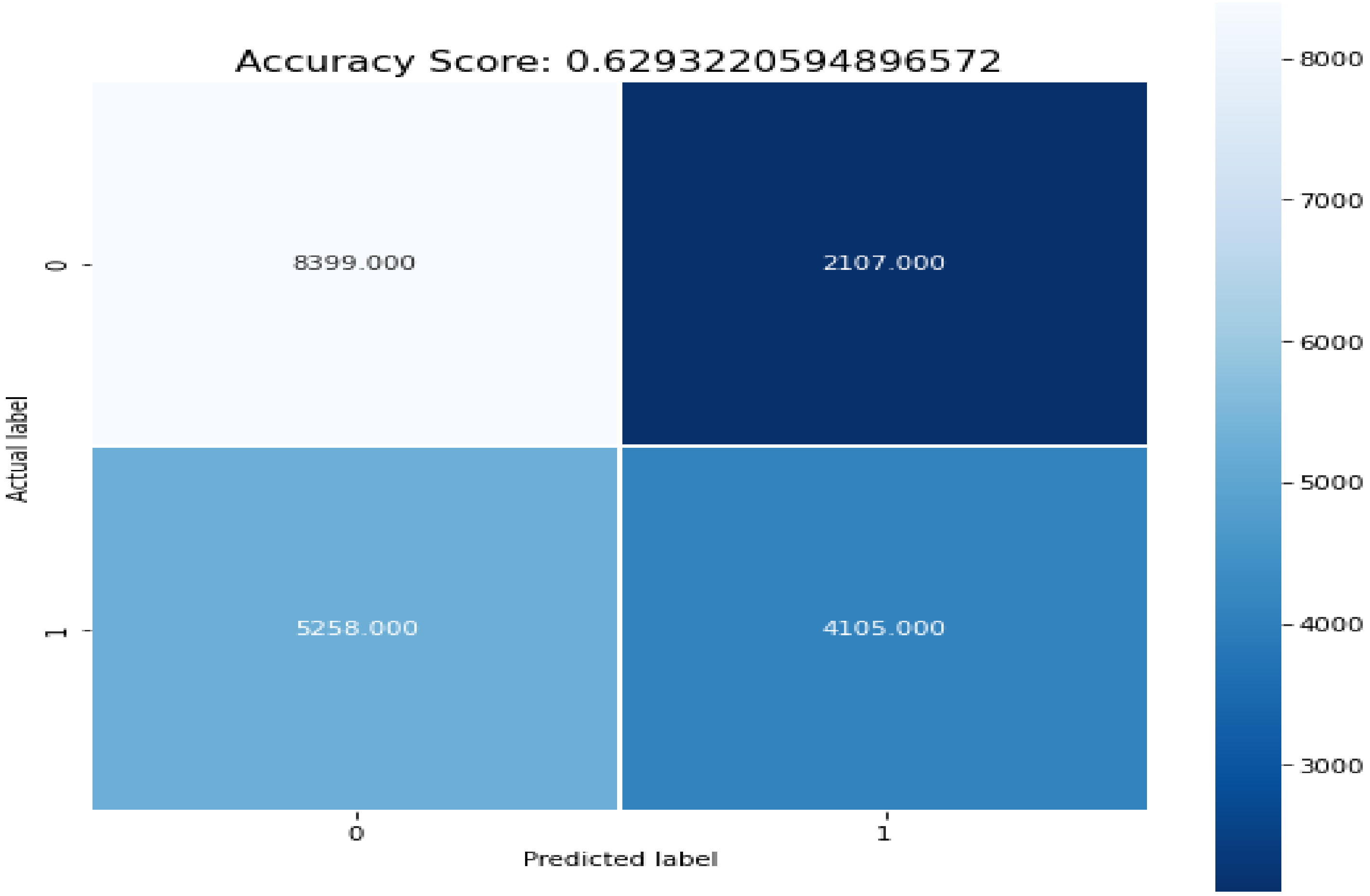


Figure 5.2 Confusion Matrix for logistic Regression

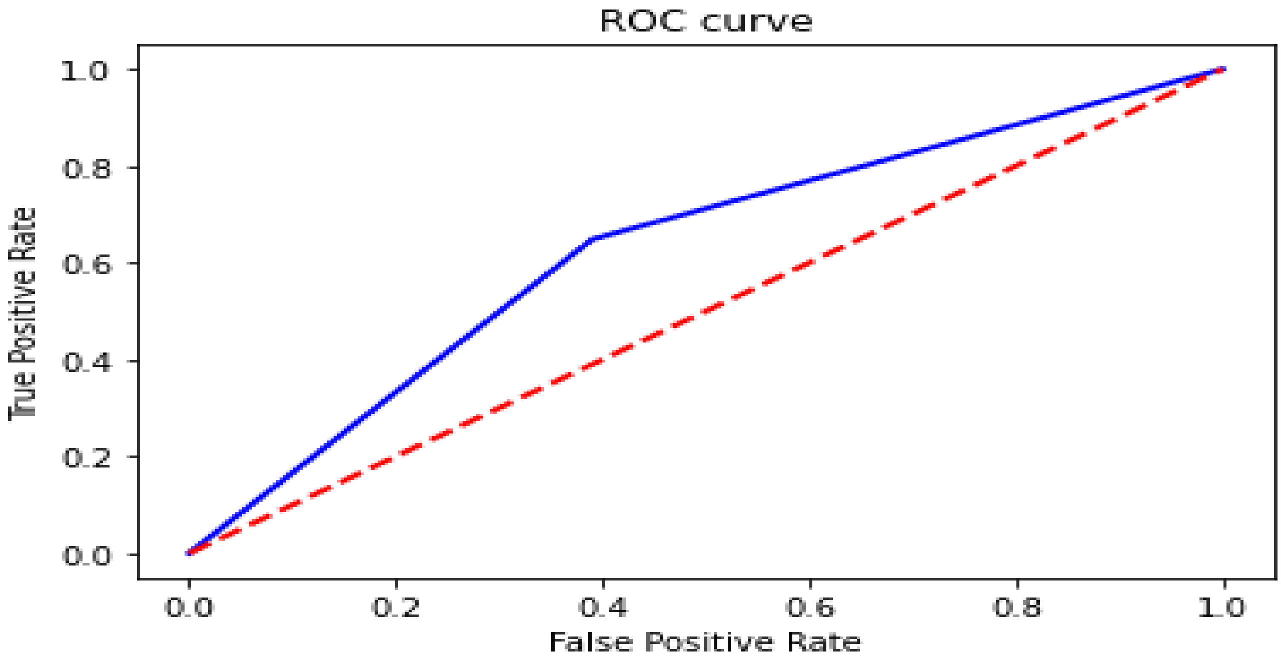


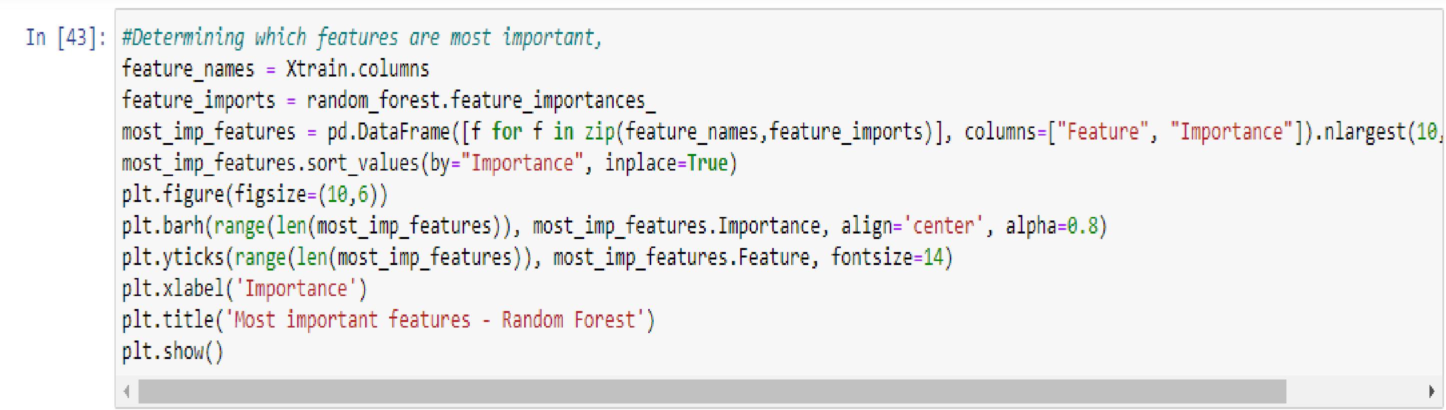
Figure 5.3 ROC Curve for Logistic Regression.

(2) Random Forest Classifier

In random forest models, multiple trees are created and the results are aggregated. The trees in a forest are de-correlated by using a random set of samples and random number of features in each tree.

It is based on the concept of **ensemble learning,** which is a process of combining multiple classifiers to solve a complex problem and to improve the performance of the model.

Another great quality of the random forest algorithm is that it is very easy to measure the relative importance of each feature on the prediction.



```
In [43]: #Determining which features are most important,
         feature_names = Xtrain.columns
         feature_imports = random_forest.feature_importances_
         most_imp_features = pd.DataFrame([f for f in zip(feature_names,feature_imports)], columns=["Feature", "Importance"]).nlargest(10,
         most_imp_features.sort_values(by="Importance", inplace=True)
         plt.figure(figsize=(10,6))
         plt.barh(range(len(most_imp_features)), most_imp_features.Importance, align='center', alpha=0.8)
         plt.yticks(range(len(most_imp_features)), most_imp_features.Feature, fontsize=14)
         plt.xlabel('Importance')
         plt.title('Most important features - Random Forest')
         plt.show()
```

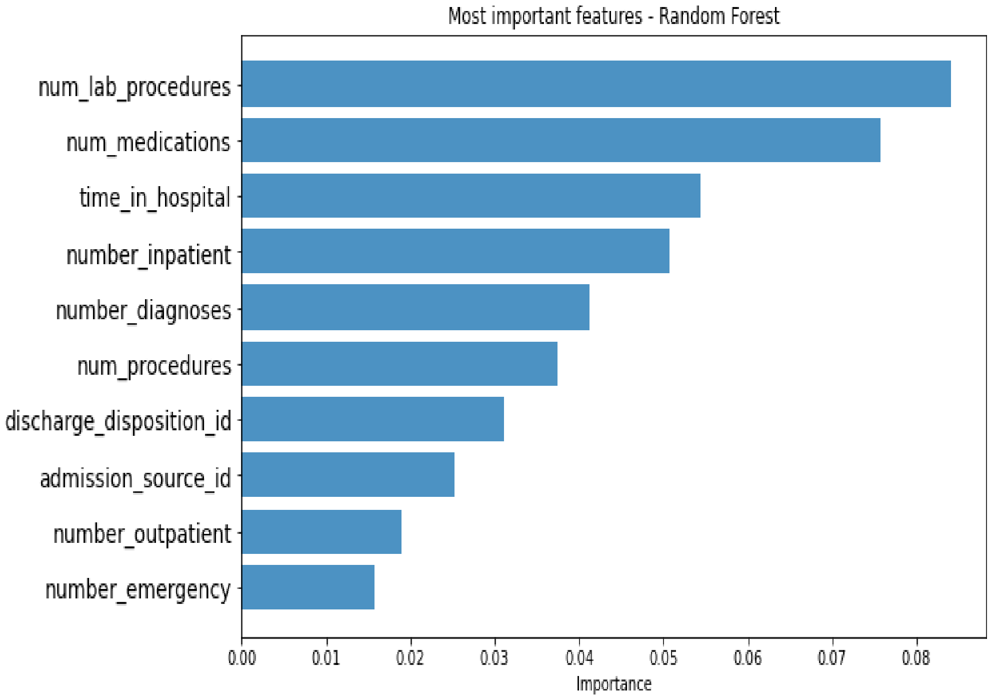


Figure 5.4 Feature Selection with Random Forest.

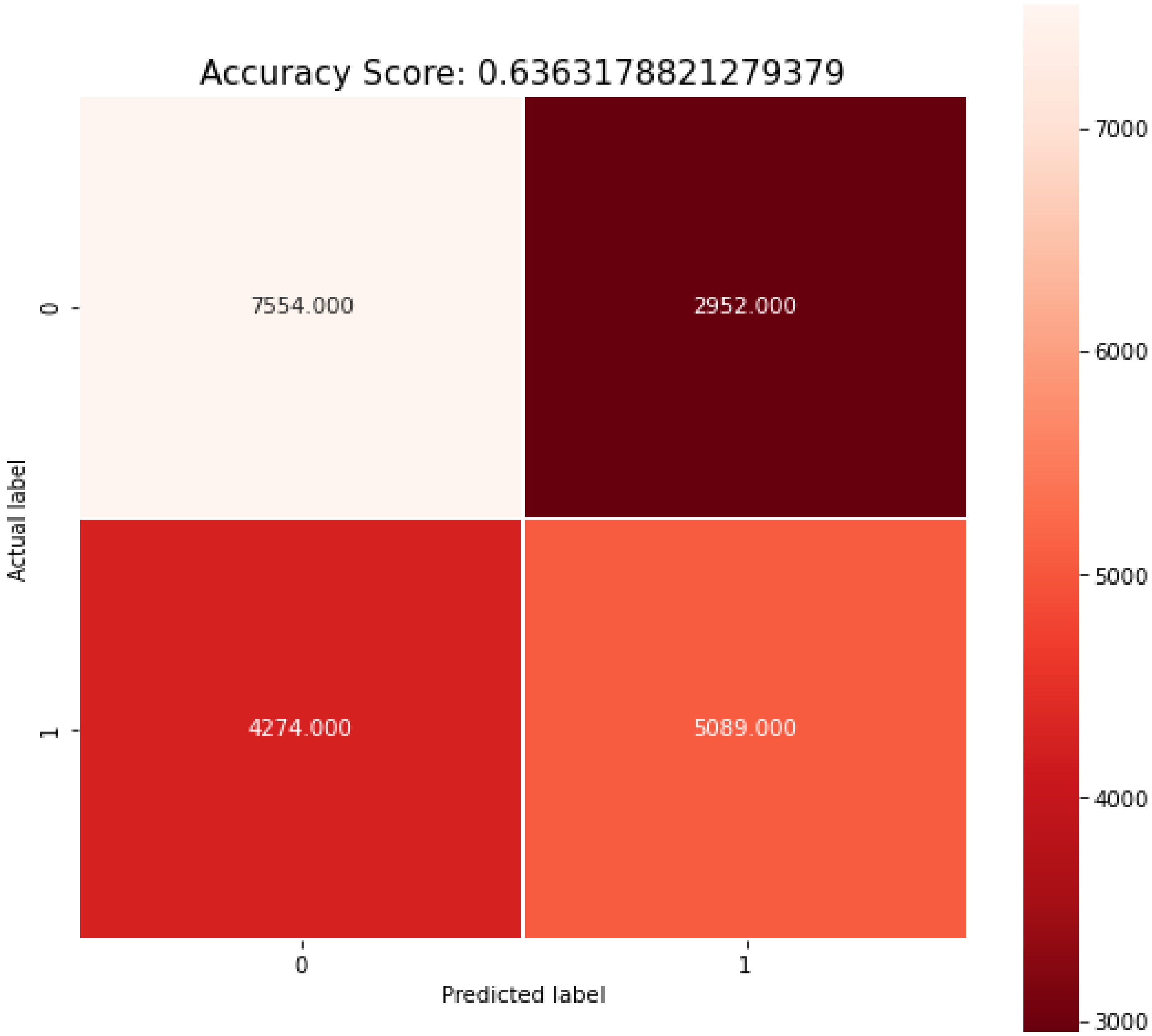


Figure 5.5 Confusion Matrix For Random Forest Classifier.

(3) AdaBoosted Classification model :

AdaBoost algorithm, short for Adaptive Boosting, is a Boosting technique that is used as an Ensemble Method in Machine Learning. It is called Adaptive Boosting as the weights are re-assigned to each instance, with higher weights to incorrectly classified instances. Boosting is used to reduce bias as well as the variance for supervised learning. It works on the principle where learners are grown sequentially.

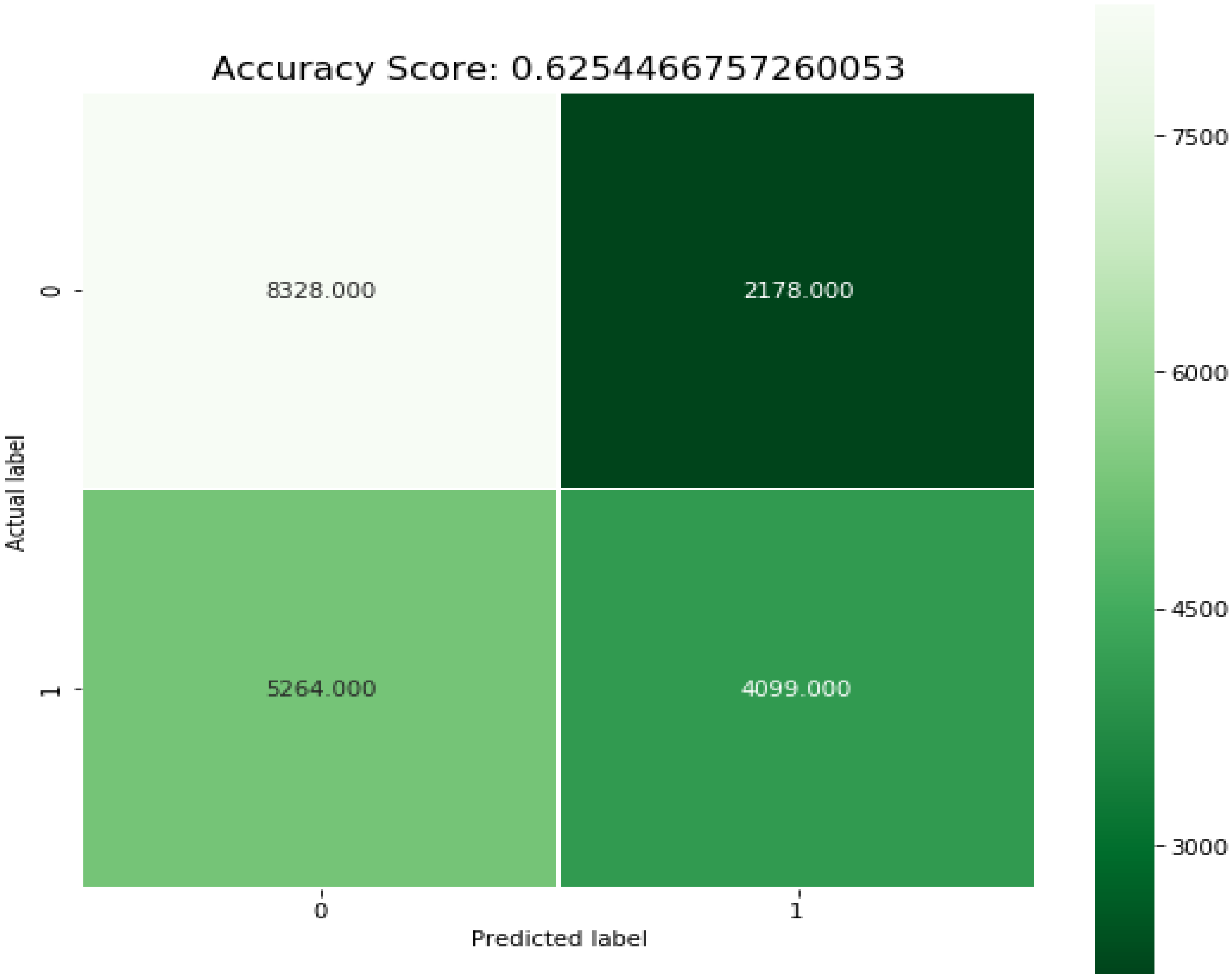


Figure 5.6 Confusion Matrix For Adaboosted Classification Model.

(4) Hyperparameters Tunning for AdaBoosted :

The experiment result shows the method that adopt Bayesian optimization algorithm for hyperparameter optimization and apply the optimized hyperparameter value to the AdaBoost algorithm does not only improves the classification accuracy of the AdaBoost algorithm, but also avoids overfitting and underfitting of the model.

CHAPTER 6

Model Evaluation.

From the above we can see that the accuracy levels of AdaBoost after tuning and Random forest is among the best, about 64%. The accuracy of all the models are similar and ranges between 62-64%. Further, applying more pre-processing techniques might help. The dataset needs more data cleaning and data fitting to achieve a higher degree of accuracy.

Looking at the false positives and the recall value which is approx 60% in Random forest, it gives us better results than the rest.

Hence by comparing all the above models and plotting a graph showing comparison of their ROC values of all the models we get:

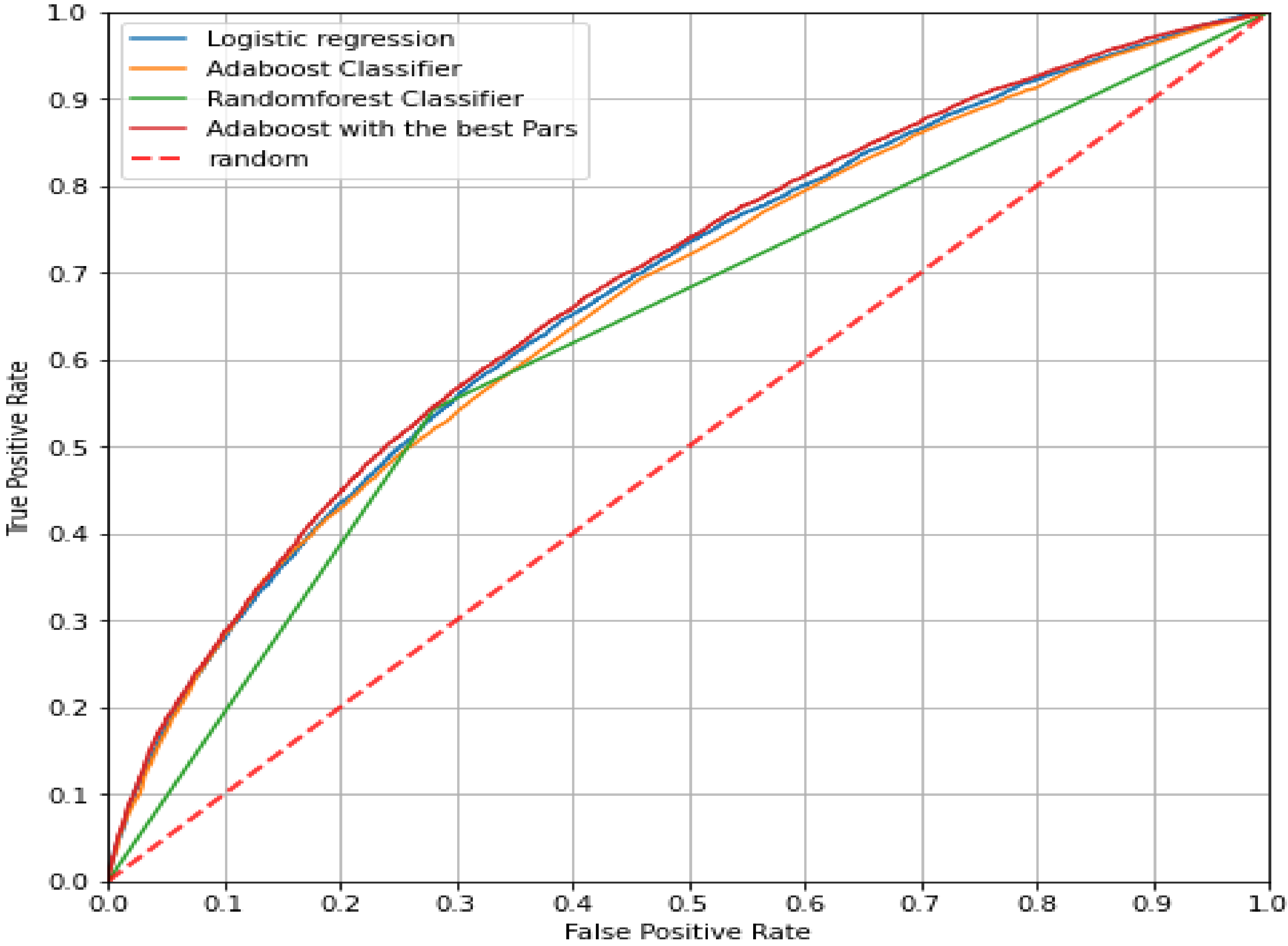


Figure 6 Comparison of the ROC curve between different models.

CHAPTER 7

Conclusion.

30-day hospital readmission of diabetes patients is of prime importance for health centers and is found very stressful due to the current models limit in term of performance and generalizability. To cope with this challenge, this study implemented a comprehensive pre-processing framework in order to improve the initial data quality, hence empowering the model's efficiency. The suggested pre-processing framework included comprehensive data cleaning, data reduction and transformation aiming at better optimizing and selecting prominent features for 30-day unplanned readmission among diabetes patients

We Performed Explanatory Data Analysis and applied extensive feature engineering, feature selection and on Diabetes patient's hospital readmission data. We used Logistic Regression, Decision Tree, Random Forest, and Adaboosted classifiers to predict the readmission rate. All of our algorithms are evaluated using the area-under-the-curve (AUC), which is equivalent to the c-statistic in the binary classification scenario. Our results showed that number of inpatient and number of diagnoses are the two most critical factors in readmission prediction. These results provided valuable suggestions to inpatients monitoring policy that may reduce short-term readmission and public healthcare cost in the future. The overall comparison of methods on the basis of their accuracy sensitivity and specificity was generated as the output with Random forest Classifier as the best model with 64% accuracy.

Of the 100,000 cases, 78,363 were diabetic and over 47% were readmitted.Based on the classes that models produced, diabetic patients who are more likely to be readmitted are either women, or Caucasians, or outpatients, or those who undergo less rigorous lab procedures, treatment procedures, or those who receive less medication, and are thus discharged without proper improvements or administration of insulin despite having been tested positive for HbA1c.

Diabetic patients who do not undergo vigorous lab assessments, diagnosis, medications are more likely to be readmitted when discharged without improvements and without receiving insulin administration, especially if they are women, Caucasians, or both.

CHAPTER 8

Future Scope.

The studied dataset provides an array of information both in term of administrative data, demographics and medical data about hospital readmissions of diabetes patients. However, various limitations should be acknowledged. The data at hand has a limited time range (1999-2008), the availability of information spanning across a wider period could improve significantly the performance of the models. Furthermore, a newer set of data would be preferable to have more realistic information about hospital readmission for diabetes patients in recent years.

While our model preforms better than our initial model using all the features available, there is room for improvement as we did not do an exhaustive tuning of the parameters. After testing several other models, we came to the conclusion that more than testing new models, we need to focus on gaining more domain expertise. We need to better understand the features and how to interpret their different values as new feature engineering will yield the best results.

The original dataset was severely imbalanced; the availability of data with more samples of the underrepresented class could help mitigate this problem. Also, there can be experimentation with other machine algorithms to gauge their performance. Furthermore, there can be more analysis of the embedding matrices to help interpret and visualize the distinguishing features. The effect of varying the dimensions of each embedding is another potential area of study. In order to provide a valid assessment and to find future directions which might lead to improvements in patient safety, we must determine all the contributing factors for predicting readmission of diabetes patients. Indeed, we must closely study classical and machine learning approaches for predictive models and investigate the relationship of readmission rates with the predictors.

Other technologies like AI (Artificial intelligence) can be applied to get some better results. Also the Above dataset can be utilized to its fullest if some columns were added onto it like medicines prescribed by the doctors then this project can serve as Drug recommendation system besides predicting Readmission of patients. If the AI is used as a guidance parallel with traditional methods to identify symptoms it could be a helpful tool to reduce work from the doctors and speed up the process to give the patient a reliable answer.

CHAPTER 9

References.

1. Strack, Beata, et al. "Impact of HbA1c measurement on hospital readmission rates: analysis of 70,000 clinical database patient records." BioMed research international 2014 (2014).
2. Bhuvan, Malladihalli S., et al. "Identifying Diabetic Patients with High Risk of Readmission." arXiv preprint arXiv:1602.04257 (2016).
3. Sushmita, Shanu, et al. "Predicting 30-Day Risk and Cost of" All-Cause" Hospital Readmissions." Workshops at the Thirtieth AAAI Conference on Artificial Intelligence. 2016.
4. Readmissions reduction program. centers for medicare and Medicaid services. https://www.cms.gov/medicare/medicare-fee-for-servicepayment/acuteinpatientpps/readmissions-reduction-program.html.
5. Diabetes 130-us hospitals for years 1999-2008 data set. https://archive.ics.uci.edu/ml/datasets/diabetes+130-us+hospitals+for+years+1999-2008.
6. Goudjerkan, T., and Jayabalan, M. Predicting 30-day hospital readmission for diabetes patients using multilayer perceptron. International Journal of Advanced Computer Science and Applications 10 (03 2019), 268–275.
7. Hammoudeh, A., AlNaymat, G., Ghannam, I., and Obeid, N. Predicting hospital readmission among diabetics using deep learning
8. Ostling, Wyckoff, Ciarkowski, Pai, Choe, Bahl, Gianchandani (2017). “The relationship between diabetes mellitus and 30-day readmission rates” in Clinical Diabetes and Endocrinology. 3:1.