UNIVERSITY OF LEICESTER

Module: CO7093 - Big Data & Predictive Analytics
Group: 105
Assignment: Classification & Clustering
Date of Submission: 27/03/2024

Group Members:

Student Id (User Name)	First Name	Last Name
239011272 (br149)	Bhuvan	Rajpoot
239063608(ar668)	Amisha	Rastogi
239055059 (rgs29)	Rahul	Sawant
239024455 (tsd16)	Tejas	Dige
239061748 (dmd23)	Devershree	Deshmukh

Table of Contents

1. Case Study

In this case study, we examine the increasing issues of Diabetes in the US brought on due to multiple factors like junk food and high sugar consumptions . This study makes use of a dataset from the 130 United States hospitals of 10 years(1999-2008) that includes 101,766 cases with 47 features associated with the clinical treatment of the patients. A detailed description of the hospital stay, lab result, medication history and the readmitted cases of the patients is offered by this dataset.

2. Objective

Predicting the probability of the patient readmission within 30 days of discharge is the main goal . We are aiming to predict the number of readmissions associated with diabetes by recognizing and reducing possible risk through the use of clustering techniques and predictive analytics.

3. Data Understanding:

“Data is the new oil”. Understanding data could be one of the most crucial parts of predictive analysis. It is important to understand the behaviour of data and relevance of the columns with respect to the problem statement.

- **Describing the data :**

As mentioned earlier, the data consist of a dataset from 130 U.S. hospitals from the year 1999-2008. Well, the dataset consist of 47 features and 101,766 instances which include several important features like :

- **num_lab_procedures:** A higher number of lab procedures may highlight complex conditions requiring increased monitoring.
- **num_procedures:** May indicate the severity of a patient's condition and complexity of treatments.
- **num_medications:** Patients taking multiple medications may have more complex conditions, increasing risk.
- **time_in_hospital:** Longer hospital stays can signal more severe illnesses.
- **medical_specialty:** Certain specialties (cardiology, oncology, etc.) are associated with high readmission rates.
- **max_glu_serum**, **A1Cresult**, **diabetesMed:** Specifically crucial for predicting readmissions in diabetic patients.
- **Readmitted** - is used as the Target Variable.

- **Data Cleaning and Transformation :**
- Cleaning the data includes various techniques, Some of them which we employed including the one mentioned in the courseworks are described below :
- Very first step which we did was to use the shape function to know the number of rows and columns, followed by looking for null values in the same.
- The null values seemed significantly lesser, upon printing the dataset we learned that the data had ? (Question Marks) instead of Null Values which was incorrect.
- Therefore we replaced ? with Nan, and inorder to understand the datatypes of each and every column, we used the **dtypes** function.Followed by converting the data type of Target Variable to Binary Feature (0 and 1).

```
# Convert 'readmitted' to binary classification
def convert_readmission(value):
  if value == '<30':
    return 1
  else:
    return 0

data['readmitted'] = data['readmitted'].apply(convert_readmission)
```

- Dropping columns with more than 90% missing values was the next step we performed, which was used to drop columns having more than 90% of missing values.
- Near zero Variance columns were also dropped, as they had values which were nearly zero.
- Well, apart from those cleaning steps mentioned in the coursework we also did cleaning which seemed relevant to model building.
- We've clustered admission_type_id, discharge_disposition_id, and admission_source_id into similar groups, which aids in dimension reduction.

```
#We've clustered admission_type_id, discharge_disposition_id, and admission_source_id into similar groups, which aids in
data['admission_type_id']=data['admission_type_id'].apply(lambda x : 5 if x in (6,8) else x)
data['admission_type_id']=data['admission_type_id'].apply(lambda x : 1 if x == 4 else 2 if x==7 else x )
```

- We have dropped data of patients using discharge_disposition_id column for ID = [11,13,14,19,20,21], where these patients are either expired or hospice.

```
#This is related to expired or Hospice, so no impact on readmission
data = data.loc[~data.discharge_disposition_id.isin([11,13,14,19,20,21])]
```

- The age columns was of type object and had values like [10-20,20-30,...] where we converted to int, For example Patient between age 10-20 are directly allocated to 2.

```
def replace_age_ranges(feature):
  age_ranges = ['[0-10)', '[10-20)', '[20-30)', '[30-40)', '[40-50)', '[50-60)','[60-70)', '[70-80)', '[80-90)', '[90-100
  values = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]

  # The function iterates through the age_ranges and replaces each occurrence in the 'feature' with the corresponding num
  for i, age_range in enumerate(age_ranges):
    feature = feature.replace(age_range, values[i])

  return feature
data['age']=replace_age_ranges(data['age'])
```

- We applied Z-Score on a single feature named num_lab_procedure, as it is a normal distribution. Filtering out data points that are more than 3 standard deviations away is applied.
- Also we applied IQR, on features like 'num_medications','time_in_hospital', 'num_procedures', 'number_outpatient','number_diagnoses', 'number_emergency', 'number_inpatient' as they had **skewed distribution.**

```
# List of features to apply Z-score method
z_score_features = ['num_lab_procedures']  # Assuming normal distribution

# List of features to apply IQR method- Skewed distribution
iqr_features = ['num_medications','time_in_hospital', 'num_procedures', 'number_outpatient','number_diagnoses', 'number_e

# Apply Z-score method for assumed normal distribution features
for feature in z_score_features:
    z_scores = np.abs(stats.zscore(data[feature]))
    filtered_entries = (z_scores < 3)  # Filtering out data points that are more than 3 standard deviations away
    data = data[filtered_entries]

# Apply IQR method for skewed distribution features
for feature in iqr_features:
    Q1 = data[feature].quantile(0.25)
    Q3 = data[feature].quantile(0.75)
    IQR = Q3 - Q1
    filtered_entries = ((data[feature] >= (Q1 - 1.5 * IQR)) &
                        (data[feature] <= (Q3 + 2.5 * IQR)))
    data = data[filtered_entries]

# Calculate the final size of the dataset after outlier handling
df_cleaned1 = len(data) / len(data) * 100

df_cleaned1
```

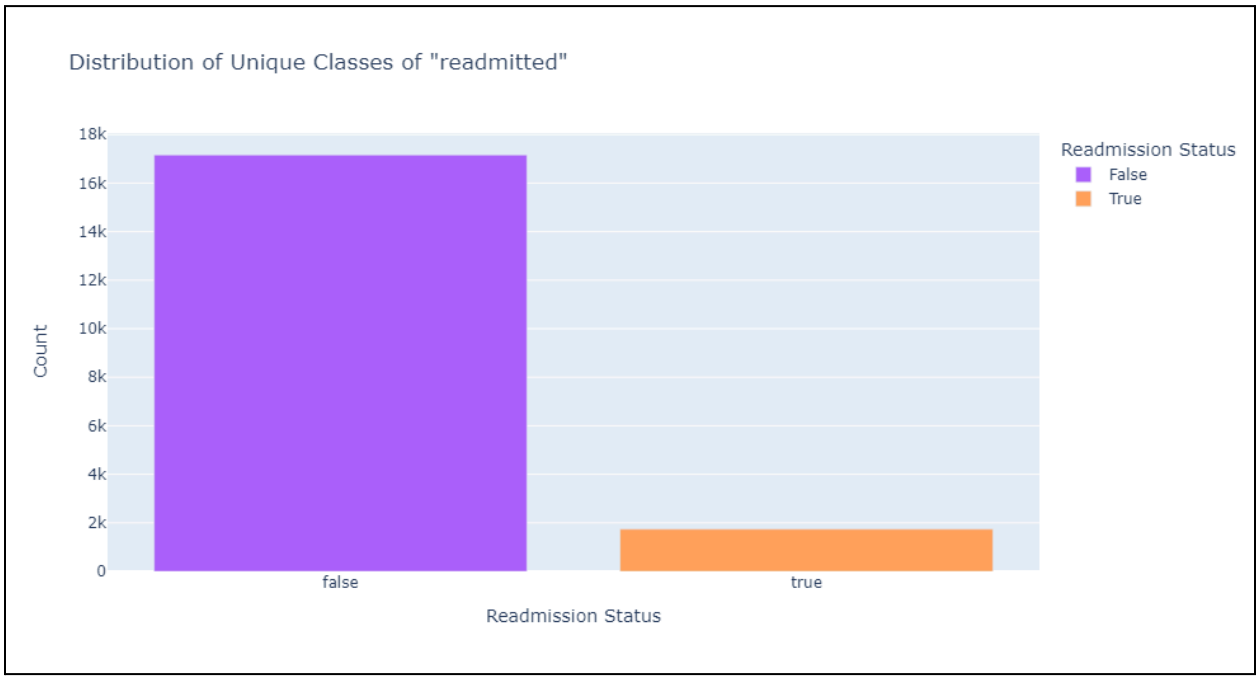
- We did employ the Feature Normalisation technique called MinMaxScaler **on columns** like 'number_emergency', 'number_inpatient', 'number_diagnoses', 'num_medications', 'time_in_hospital'. We used MinMax as it can be used for data visualisation and also provides better performance and faster convergence. It is bounded to a range 0 to 1.

Therefore, after considering the clean techniques mentioned in the coursework and some of the additional cleaning approaches employed we decided to go with the shape of the dataset : (18898, 32).

4. Data Visualisation

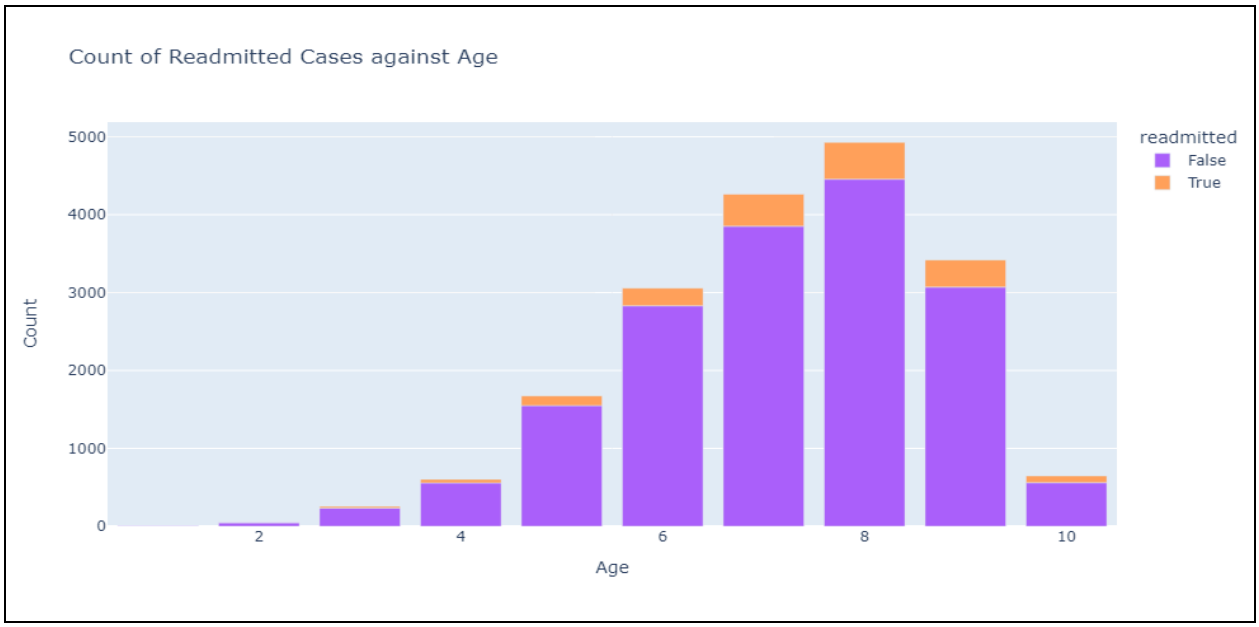
i. Plot the distribution of unique classes of the target variable, i.e., readmitted

This bar chart highlights the significant imbalance between readmitted ('True') and non-readmitted ('False') individuals. The high difference in bar heights visually emphasises the far fewer readmitted cases



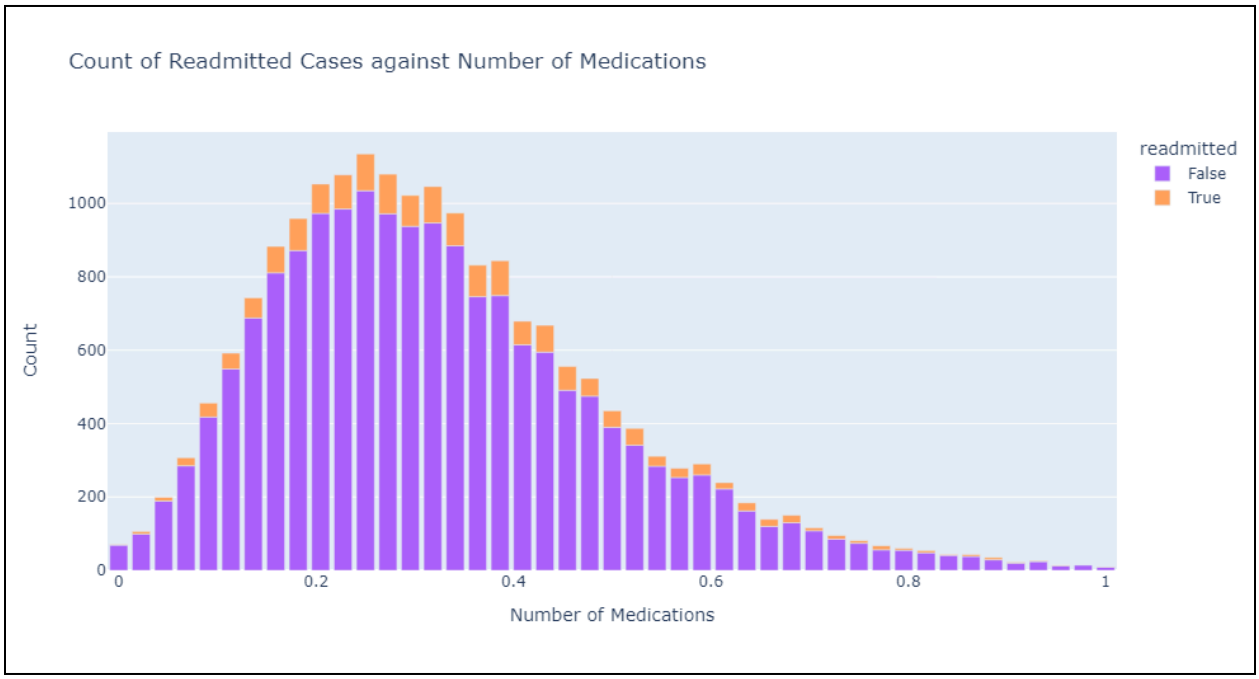
ii. Plot the count of the number of readmitted cases against age.

This combined bar chart shows the count of individuals who were and were not readmitted across different age groups. This visualisation helps in identifying that older group people are readmitted more rather than young people.



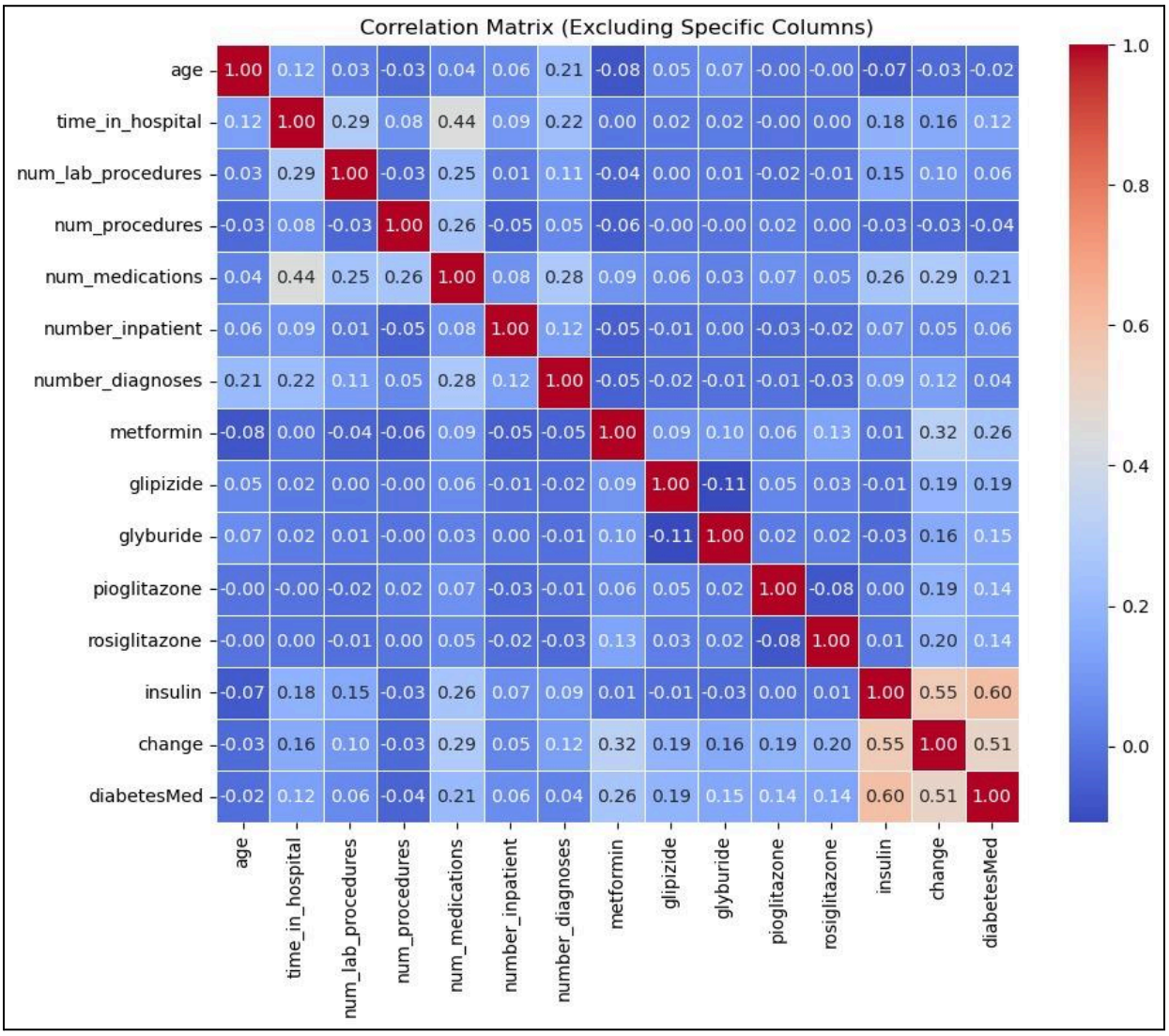
iii. Plot a graph that displays the count of target variables against the number of medications.

This histogram overlaid with another histogram (or a stacked bar chart) shows the count of readmitted cases against the number of medications, distinguishing between readmitted (True) and those who were not (False). This indicates number of medications affecting readmission rates.



iV. Correlation matrix

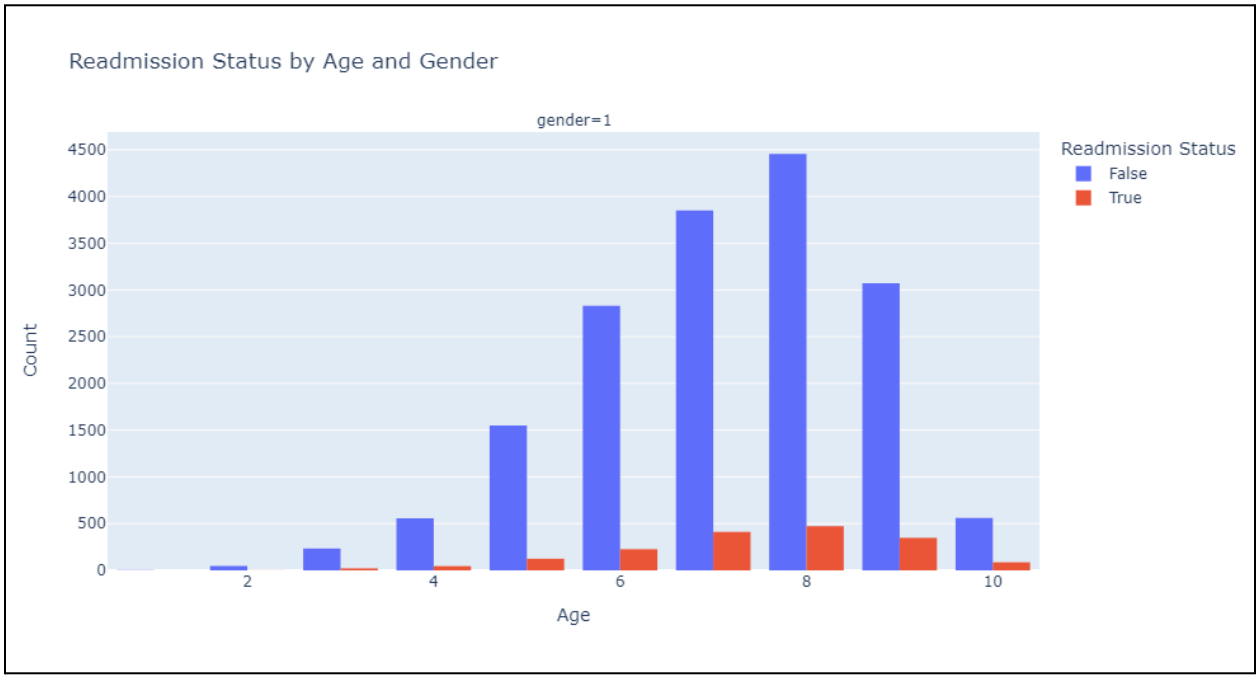
This heat map shows the correlation coefficients between different features in the dataset. It helps identify which variables have a strong relationship with each other. We can say that these variables are highly correlated with each other ‘num_medications’, 'time_in_hospital', 'insulin','change', 'diabetesMed','metformin'



V. Additional Plots

a. Readmission status by Age and Gender:

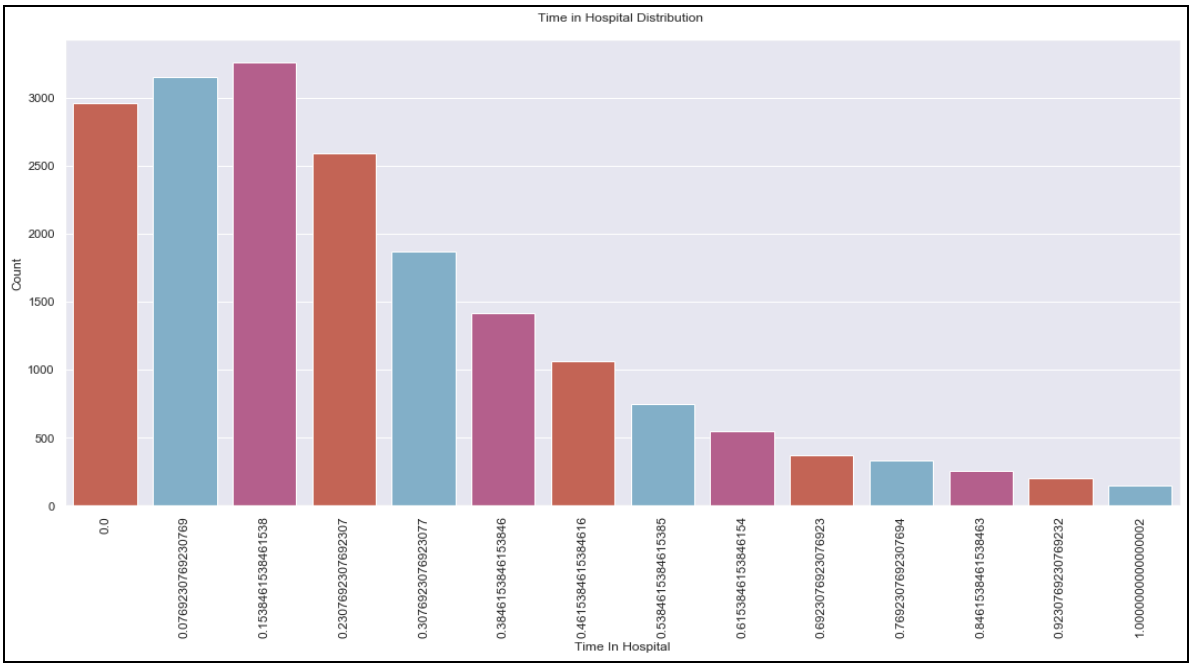
This bar chart breaks down readmission counts by age and gender. Each age group has a pair of bars representing males and females, allowing analysis of how readmission rates are influenced by both factors.



b. Time in Hospital:

This bar chart exhibits a right-skewed distribution of hospital stay lengths. Most stays are concentrated towards shorter durations, with a decreasing frequency of longer stays.

From the Graph and Mean of the Time in Hospital, We found that the majority of the people stay in hospital 2-3 Days.



5. Model Building

- **Initial Data Cleaning and Preprocessing:**
- After completing the initial steps of data cleaning and preprocessing, we were left with a dataset comprising 18898 rows and 32 columns.
- We employed Logistic Regression, a statistical method tailored for binary outcome prediction, in our initial predictive modelling endeavour. This technique is adept at forecasting outcomes when only two distinct possibilities exist.
- In our case, we utilised Logistic Regression to anticipate whether a patient would be readmitted within 30 days or not.
- To prepare our dataset for analysis, we performed label encoding. This transformation was necessary because our dataset comprised various data types, necessitating conversion into numerical format.

- **Feature Selection:**

  Selected features based on correlation with the target variable.

  - Chosen features: 'num_medications','number_diagnoses','number_inpatient', 'time_in_hospital','insulin','change','diabetesMed','metformin'

- Features identified through correlation matrix analysis to focus on influential predictors for readmission prediction.
- Using two less correlated variables can help in mitigating logistic regression models which can be prone to overfitting, especially with many features.

- **Data Splitting:**
- Divided dataset into training and testing sets.
- Adopted an 80:20 ratio for splitting, allocating 80% to training and 20% to testing.

Ensured model learned from training data while evaluating performance on unseen testing data.

- **Model Evaluation:**
- For evaluation of the model we have employed cross validation using k-fold technique, with the parameter as n-splits=10.Cross validation is a very robust method of accessing the model performance by dividing the dataset into multiple folds . The k-fold technique specifically divides the data into k equal-sized folds.
- Model exhibited low recall and F1 score for positive readmissions.
- Precision score is less, indicating fraction of predicted readmissions were true positives.
- Overall accuracy was 0.91, but performance in detecting positive readmissions fell short.

```
Classification Report on Test Set:
              precision    recall  f1-score   support

       False       0.91      1.00      0.95      3447
        True       0.00      0.00      0.00       333

    accuracy                           0.91      3780
   macro avg       0.46      0.50      0.48      3780
weighted avg       0.83      0.91      0.87      3780
```

- **Class Imbalance and Oversampling:**
- Calculated class imbalance ratio for target variable which was less
- Using RandomOverSampler technique for oversampling due to high class imbalance.
- Even after oversampling, recall improved to 0.59, precision to 0.12, and F1 score to 0.20, with overall accuracy of 0.58.

```
Classification Report on Test Set:
              precision    recall  f1-score   support

       False       0.94      0.58      0.72      3447
        True       0.12      0.59      0.20       333

    accuracy                           0.58      3780
   macro avg       0.53      0.58      0.46      3780
weighted avg       0.86      0.58      0.67      3780
```

- **Decision for an Improved Model:**
- Aggressive data cleaning resulted in significant data loss.
- Decided to work on an improved model while retaining more data.
- Recognized the need to address the problem of low prediction by optimising data retention strategies.

5. Improved Model

- **Initial data preprocessing steps:**

To ensure enhancement of our baseline model, we prioritised maximal retention of available data. Upon initial dataset importation, our foremost step entailed examining the prevalence of null or missing values across columns.Furthermore it became apparent that numerous entries were designated as '?', thus replacing these '?' entries with 'nan' enabled a more accurate assessment of null value counts.Utilising a dictionary lookup, age ranges were mapped to specific numerical values within the 'age' column.

For streamlined analysis, 'readmitted' column values denoted as '<30' were assigned a value of 1, while those labelled as '>30' and 'NO' were replaced with 0. Evaluation of unique values within the 'readmitted' column highlighted a significant imbalance, necessitating careful consideration during model construction to ensure accurate prediction of patient readmittance.

Additionally coming to model development, columns exhibiting no variation and those harbouring excessive null data were prudently dropped. Subsequent removal of rows containing null values yielded a dataset comprising 98,000 rows and 26 columns.

Subsequent segregation of numerical and categorical columns enabled calculation of z-scores for each numerical attribute, facilitating identification of outliers. The percentage of outliers in each numerical column was then determined and reported.

Histogram plots were generated for numerical columns, with their skewness assessed to guide outlier removal using both z-score and interquartile range (IQR) methods. Following this data refinement process, the final dataset shape stood at (86,359, 26), marking it as primed for the commencement of enhanced model construction.

After label encoding and segregating the data into feature and target variables, we applied the recursive feature selection method to identify the top 15 features from the dataset. The resulting features were stored in a list termed as 'selected_features' and were subsequently printed.

```
Selected Features:
gender
admission_type_id
discharge_disposition_id
time_in_hospital
num_procedures
num_medications
number_emergency
number_inpatient
number_diagnoses
max_glu_serum
A1Cresult
metformin
glipizide
glyburide
pioglitazone
```

In alignment with the approach adopted in the baseline model, logistic regression was chosen for the classification task. However, unlike the previous iteration where feature selection relied on correlation matrices, recursive feature selection was employed here.

Despite maintaining a substantial amount of data, the initial logistic regression model yielded unsatisfactory results, particularly in terms of recall, precision, and F1 scores for patients requiring readmittance.

```
Accuracy Report:
Accuracy: 0.8963254593175853
Classification Report:
              precision    recall  f1-score   support

           0       0.90      1.00      0.95     23222
           1       0.00      0.00      0.00      2686

    accuracy                           0.90     25908
   macro avg       0.45      0.50      0.47     25908
weighted avg       0.80      0.90      0.85     25908
```

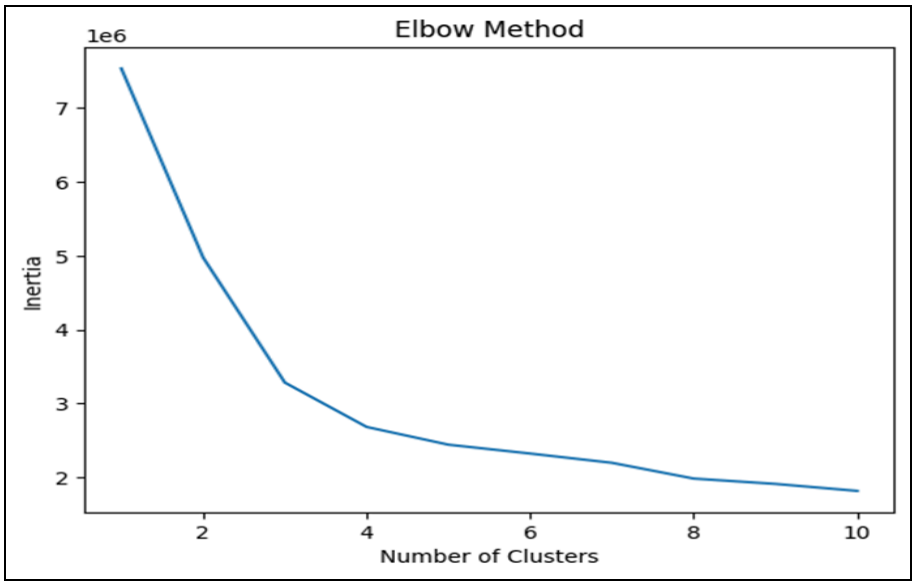
- **K – means clustering:**

To enhance the model's accuracy, we incorporated k-means clustering technique. This approach aimed to group similar data points, thereby refining the model's predictive capabilities. Following the application of k-means clustering, an elbow graph was utilised to determine the optimal number of clusters, which was identified as 3.

```
# Elbow Method for Choosing Number of Clusters
inertia = []
for i in range(1, 11):
    kmeans = KMeans(n_clusters=i, random_state=42)
    kmeans.fit(X_rfe)
    inertia.append(kmeans.inertia_)

# Plot Elbow Method
plt.plot(range(1, 11), inertia)
plt.title('Elbow Method')
plt.xlabel('Number of Clusters')
plt.ylabel('Inertia')
plt.show()
```

Subsequently, local classifiers were constructed based on the clusters generated by k-means. Despite marginal improvements observed after training logistic regression models on these classifiers, it became evident that further enhancements were necessary to fully leverage the model's potential. Consequently, data imbalance was addressed through oversampling using the Synthetic Minority Over-sampling Technique (SMOTE), coupled with hyperparameter tuning.

```
Classification Report for Cluster 2:
              precision    recall  f1-score   support

           0       0.87      1.00      0.93      1218
           1       0.25      0.01      0.01       190

    accuracy                           0.86      1408
   macro avg       0.56      0.50      0.47      1408
weighted avg       0.78      0.86      0.80      1408
```



Hyperparameter tuning was conducted using the GridSearchCV method to optimise model performance. The combination of SMOTE and hyperparameter tuning significantly ameliorated the model's efficacy, leading to a noteworthy enhancement in accuracy.

```
Best Parameters: {'C': 100, 'penalty': 'l2'}

Accuracy Report after Hyperparameter Tuning and Oversampling with SMOTE:
Accuracy: 0.644985402713378
Classification Report:
              precision    recall  f1-score   support

           0       0.67      0.58      0.62     23315
           1       0.63      0.71      0.67     23269

    accuracy                           0.64     46584
   macro avg       0.65      0.65      0.64     46584
weighted avg       0.65      0.64      0.64     46584
```

Conclusion:

We used logistic regression with an extensively processed dataset in our first model. Nevertheless, the categorization report fell short of expectations because of the imbalance in the target variable's classes. We oversampled the minority class using the RandomOverSampler technique in order to overcome this problem, and the outcome was an accuracy score of 0.58. The precision stayed poor at 0.12 even though the recall increased to 0.59.

After improvements, the Improved model's total accuracy was 0.64. Crucially, we made great improvement by concentrating on patient readmission prediction. Significant improvements were observed in readmitted patients' memory, precision, and F1 score, which increased to 0.71, 0.63, and 0.67, respectively. These results indicate significant improvements over the baseline model and show the effectiveness of our improved strategy.

References:

- **Gupta, P., Bagchi, A. (2024). Introduction to Pandas. In: Essentials of Python for Artificial Intelligence and Machine Learning. Synthesis Lectures on Engineering, Science, and Technology. Springer, Cham. https://doi.org/10.1007/978-3-031-43725-0_5**

- **Bai, S. *et al.* (2019) 'Data Visualization Model Methods and Techniques', *IOP Conference Series: Earth and Environmental Science*. IOP Publishing, 252(5), p. 052063. doi: 10.1088/1755-1315/252/5/052063.**

- **Nick, T.G., Campbell, K.M. (2007). Logistic Regression. In: Ambrosius, W.T. (eds) Topics in Biostatistics. Methods in Molecular Biology™, vol 404. Humana Press. https://doi.org/10.1007/978-1-59745-530-5_14**

- **Douzas, G., Bacao, F. and Last, F. (2018) 'Improving imbalanced learning through a heuristic oversampling method based on k-means and SMOTE', *Information Sciences*, 465, pp. 1–20. doi: 10.1016/j.ins.2018.06.056.**

- **Na, S., Xumin, L. and Yong, G. (2010) 'Research on k-means Clustering Algorithm: An Improved k-means Clustering Algorithm', in *2010 Third International Symposium on Intelligent Information Technology and Security Informatics*, pp. 63–67. doi: 10.1109/IITSI.2010.74.**

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