Big Data Coursework

Introduction:

The main goal of the problem statement is to develop a classification model that uses a dataset of clinical care data from 130 US hospitals spanning 10 years to predict whether a patient will be readmitted to the hospital within 30 days. Because it can help identify patients who are at high risk of readmission and enable focused treatments to improve patient outcomes and save healthcare costs, this prediction is critical for healthcare professionals.

In our analysis, we employed logistic regression as a foundational modeling technique to investigate the relationship between predictor variables and a binary outcome. Logistic regression is particularly well-suited for scenarios where the dependent variable is categorical and binary, making it ideal for tasks such as classification. By fitting a logistic regression model to the data, I aimed to discern the probability of occurrence of a particular event or outcome based on a set of predictor variables. This approach allowed for the exploration of the impact of various features on the likelihood of the target outcome, providing valuable insights into the underlying patterns and associations within the dataset.

Dataset

The dataset provided is a comprehensive collection of patient records from a healthcare facility. Each row represents an individual patient encounter, capturing a wide array of relevant information. The dataset encompasses demographic details such as race, gender, and age group, as well as clinical attributes like admission type, discharge disposition, and source of admission. Furthermore, it includes quantitative measures pertaining to the patient's hospital stay, including the duration of hospitalization, the number of laboratory procedures, medical procedures, and prescribed medications.

For instance, it captures information on various diabetic medications prescribed, such as metformin, insulin, and other oral hypoglycaemic agents, suggesting a significant proportion of patients with diabetes. Additionally, the presence of variables like serum glucose levels and haemoglobin A1C results further corroborates the dataset's focus on diabetes and its management. Furthermore, it captures information on the patient's payer code, which may be relevant for studying healthcare financing and insurance-related aspects.

Data Cleaning and Transformation

The dataset given to us contained the data of 130 US hospitals diabetic patients. As the data wasn’t clean, we started with cleaning the data using various methods mentioned below in this section.

We used pandas to clean the data because it has powerful functions to handle missing values, perform data transformation and manipulation.

Step 1: Data Loading and Initial Inspection:
The first step involved loading the dataset using the Pandas library in Python. We inspected the shape of the dataset and examined its columns to gain a preliminary understanding of the data structure.

```
import pandas as pd

# Loading the dataset
df = pd.read_csv('diabetic_data.csv', keep_default_na=False)

# Displaying the shape of the data
print("Shape of the data:", df.shape)
print("Columns of the DataFrame:", df.columns)
```

Step 2: Column Deletion:
We identified the 'encounter_id' column as unnecessary for our analysis and thus decided to remove it from the dataset.

```
# Deleting the column 'encounter_id'
df.drop(columns=['encounter_id'], inplace=True)
print("Columns of the DataFrame after deletion:", df.columns)
```

Step 3: Handling Missing Values: We identified columns containing '?' as missing values and replaced them with NaN. Then, we calculated the summary of missing values after the replacement.

```
# Identifying columns containing '?'
columns_with_question_mark = df.columns[df.isin(['?']).any()]

# Replacing '?' with NaN only in columns containing '?'
df[columns_with_question_mark] = df[columns_with_question_mark].replace('?', np.nan)

# Summary of missing values after replacing '?'
missing_after = df.isnull().sum()

print("\nMissing Values After Replacement:")
print(missing_after)
```

Step 4: Data Transformation: We replaced categorical values '<30' with 1 and '>30' and 'NO' with 0 in the 'readmitted' column.

```
# Replacing '<30' with 1 and '>30' and 'NO' with 0
df['readmitted'].replace({'<30': 1, '>30': 0, 'NO': 0}, inplace=True)

# Displaying the unique values in the 'readmitted' column to verify the replacement
print("Unique values in 'readmitted' column after replacement:", df['readmitted'].unique())
```

Step 5: Dropping Columns with High Missing Values: Columns with more than 90% missing values were dropped to maintain data integrity.

```
# Calculating percentage of missing values in each column
missing_percentage = (df.isna().sum() / len(df)) * 100

# Dropping columns with more than 90% missing values
columns_to_drop = missing_percentage[missing_percentage > 90]
df.drop(columns=columns_to_drop.index, inplace=True)
```

Step 6: Dropping Near-Zero Variance Columns: Columns with no variations or near-zero variance were dropped to avoid redundancy.

```
# Dropping columns with no variations or near-zero variance
columns_to_drop = ['examide', 'citoglipton', ...]  # List of columns to drop
df.drop(columns=columns_to_drop, inplace=True)
```

Step 7: Handling Outliers:

Outliers were removed using the Interquartile Range (IQR) method to ensure the robustness of the data. Using the Interquartile Range (IQR) is advantageous when dealing with imbalanced data because it provides a robust measure of variability that is less influenced by outliers compared to traditional scoring methods. In imbalanced datasets, where outliers and extreme values can distort summary statistics, the IQR offers a more reliable assessment of data spread by focusing on the middle 50% of observations. This makes it a valuable tool for understanding the central tendency and variability within each class, ultimately leading to more accurate analyses and model evaluations.

```
  cleaned_df = df.copy()
  for col in columns:
    q1 = df[col].quantile(0.25)
    q3 = df[col].quantile(0.75)
    iqr = q3 - q1
    lower_bound = q1 - threshold * iqr
    upper_bound = q3 + threshold * iqr
    cleaned_df = cleaned_df[(cleaned_df[col] >= lower_bound) & (cleaned_df[col] <= upper bound)]
  return cleaned_df
```

Note: We found 16K repeated patient values, we didn't drop them because, when 2nd time, the patient visited, the readmitted column had different values.

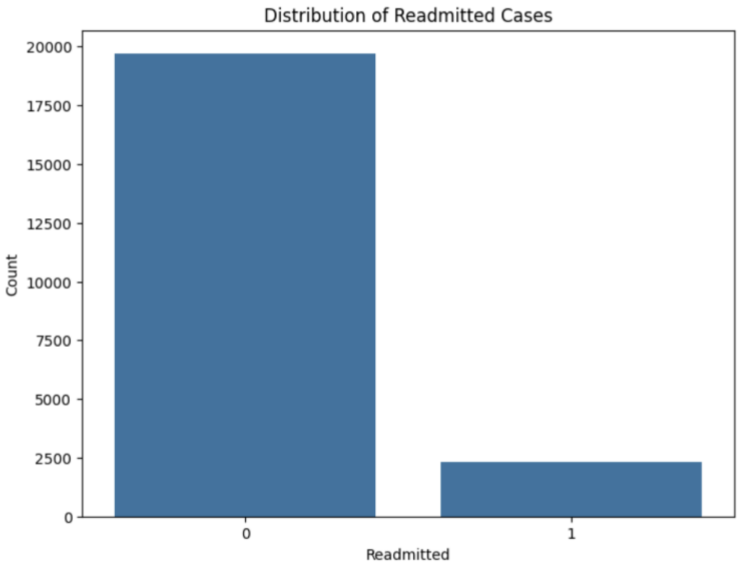
Data Visualisation

In this section, we present visualizations derived from the pre-processed dataset titled "cleaned_df", focusing on key variables such as readmission status, age, medications, and demographic factors.

Distribution of Readmitted Cases:

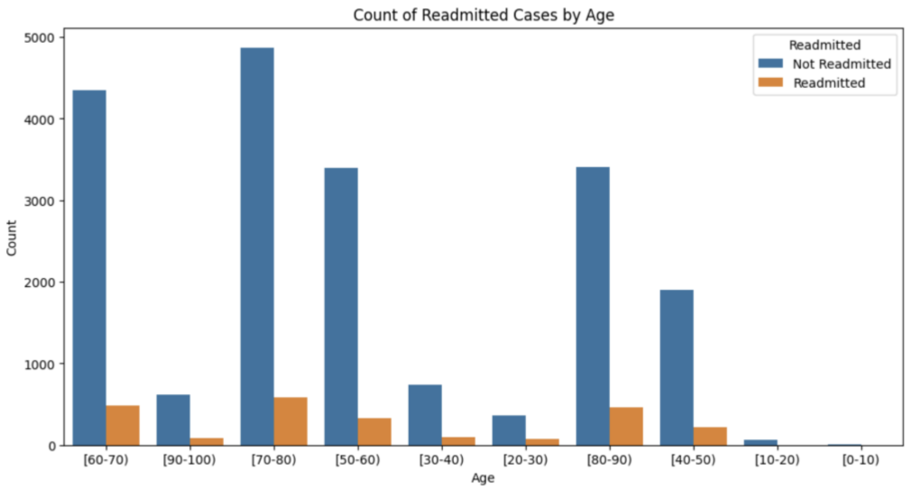
We understand that the distribution of readmitted cases is crucial for assessing the prevalence and impact of readmissions within the dataset.

The graph suggests that readmission is a relatively uncommon event in this dataset.



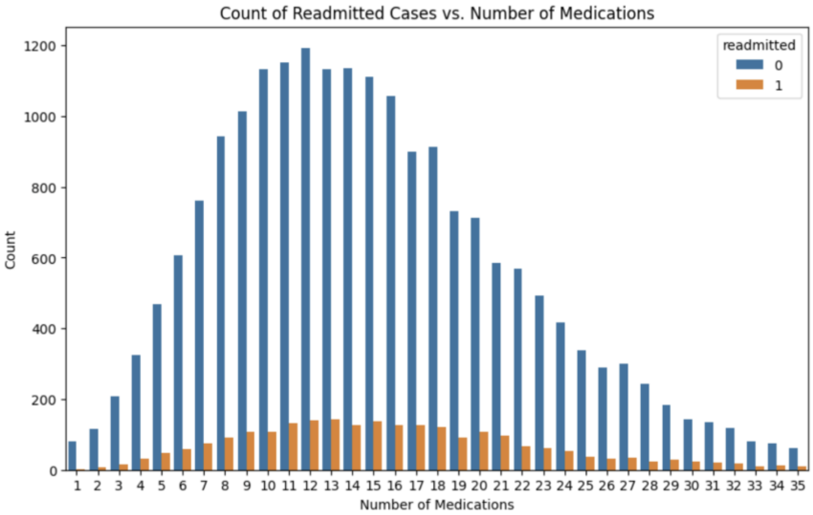
Readmitted Cases by Age: Examining readmitted cases across different age groups enables us to distinguish potential age-related patterns in readmissions. By grouping readmissions by age and visualizing them through a countplot, we can identify age cohorts that may be more susceptible to readmissions.

It appears that there are more readmissions among older age groups. Specifically, the age group (70-80) has the highest number of readmissions, followed by (60-70) and (80-90). This suggests that older adults may be more susceptible to readmission than younger adults.

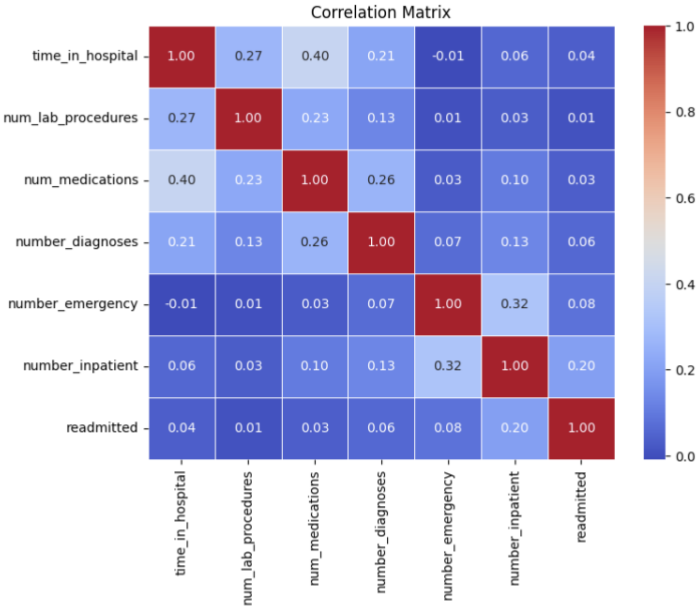


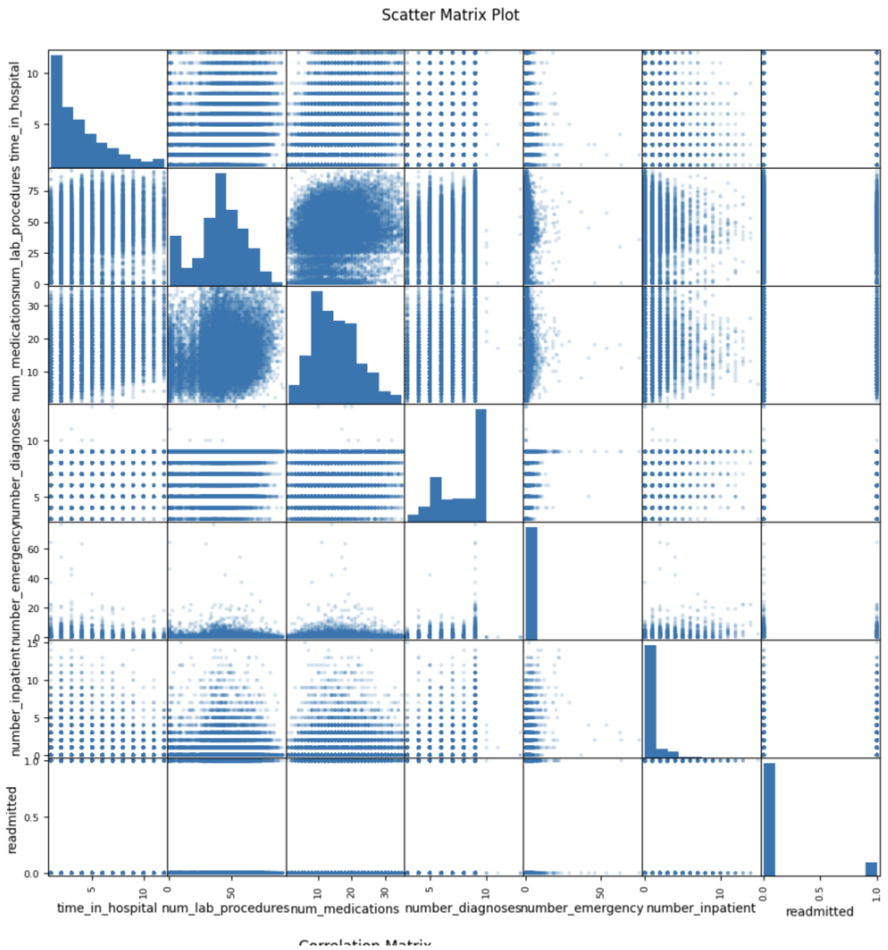
Readmitted Cases vs. Number of Medications: Visualizing the count of readmitted cases against the number of medications provides insights into medication-related factors influencing readmissions. This visualization allows healthcare practitioners to identify potential correlations between medication regimen complexity and readmission risk, guiding medication management strategies to optimize patient care and mitigate readmission risks.

It appears that the number of readmitted cases tends to increase as the number of medications prescribed increases. As the number of medications increase the readmission increases initially, but the graph reduces gradually after a certain point. This suggests a varying correlation between the complexity of a patient's medication regimen and their risk of being readmitted to the hospital.

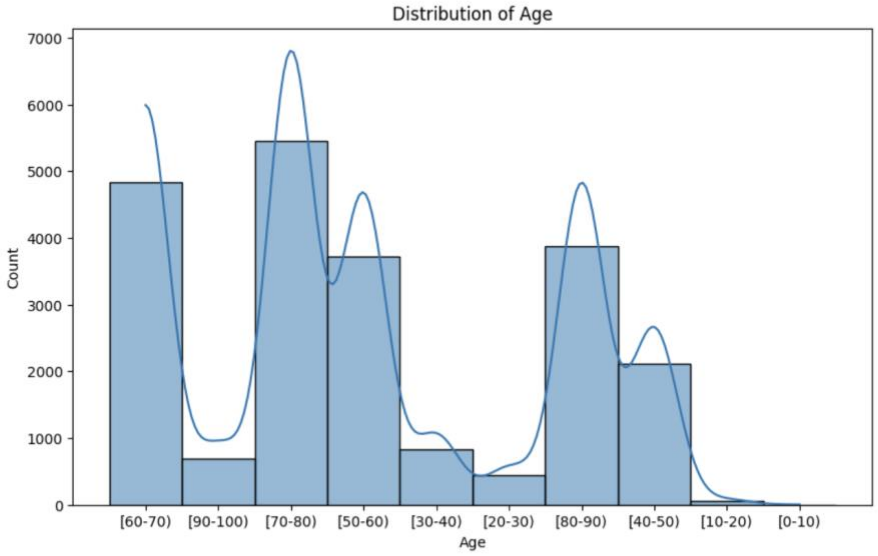


Correlation Matrix and Pairplot: The correlation matrix visualizes the strength and direction of linear relationships between variables, offering insights into potential associations that may influence readmission outcomes. Similarly, the pairplot provides a comprehensive overview of pairwise relationships, allowing for the identification of non-linear correlations and patterns.



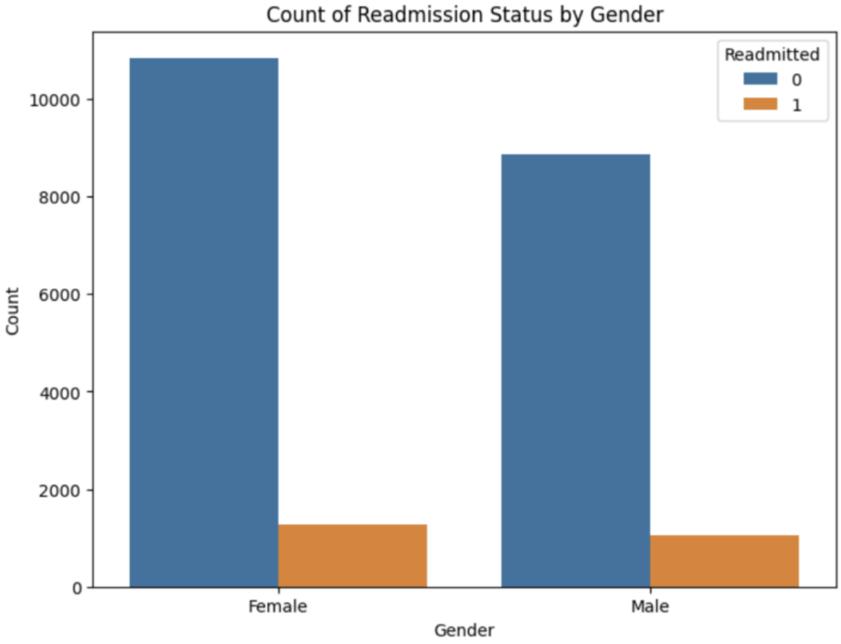


Distribution of Age: The histogram plot visualizes the distribution of patient ages, providing a comprehensive overview of the age demographics within the dataset. By segmenting age into bins and plotting the count of patients within each bin, the histogram illustrates the frequency distribution of ages across the dataset. Additionally, the inclusion of a kernel density estimate (KDE) enhances the visualization by providing a smoothed representation of the age distribution curve.



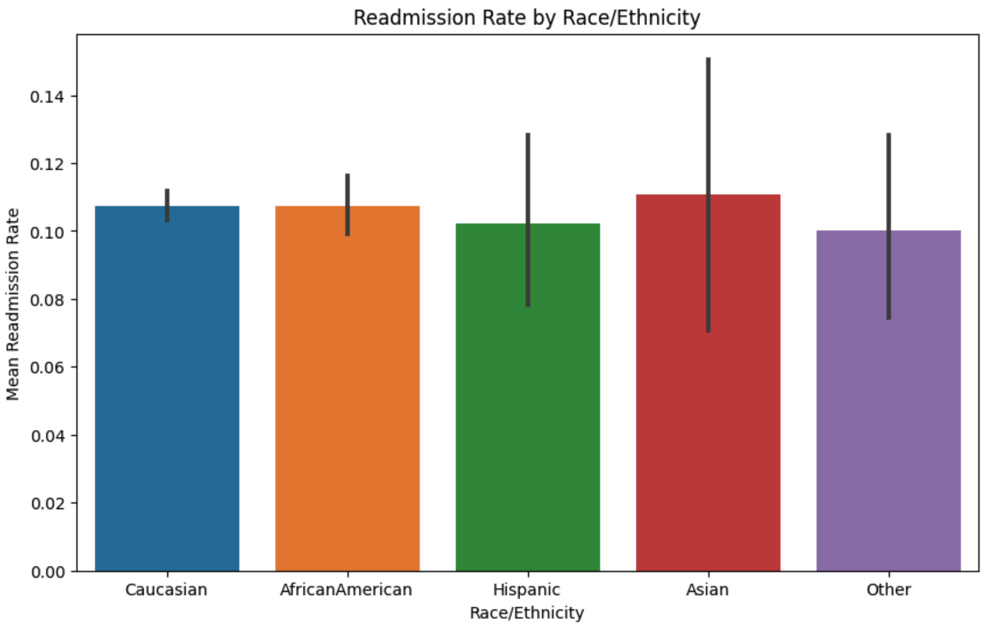
Distribution of Readmission Status by Gender: Understanding how readmission status varies across different genders is essential for identifying potential gender-based disparities . The countplot visualizing the distribution of readmission status by gender allows us to discern any discrepancies in readmission rates between male and female patients.

This data suggests that female patients in this dataset are more likely to be readmitted than male patients. However, it is important to note that without knowing the total number of patients for each gender, it is difficult to say definitively whether there is a gender-based disparity in readmission rates. For instance, if there were many more female patients overall than male patients, then this could account for the higher number of female readmissions simply because there were more female patients to begin with.



Readmission Rate by Race/Ethnicity: Examining readmission rates across different race/ethnicity groups provides insights into healthcare disparities and inequalities. The bar plot depicting the mean readmission rate by race/ethnicity enables us to compare readmission rates between different racial and ethnic groups.

- Asian patients appear to have the highest mean readmission rate.
- Caucasian patients have a similar mean readmission rate than African American patients but higher than Hispanic patients.
- Hispanic patients have the lowest mean readmission rate among the racial/ethnic groups shown.



Model Building

- **Data Preparation:**
  We begin by identifying the most pertinent features that could potentially influence the target variable, 'readmitted.'

```
numerical_predictors = ['time_in_hospital', 'num_lab_procedures',
'num_medications', 'number_diagnoses', 'number_emergency',
'number_inpatient']
categorical_predictors = ['discharge_disposition_id', 'diag_1', 'change', 'diabetesMed']
```

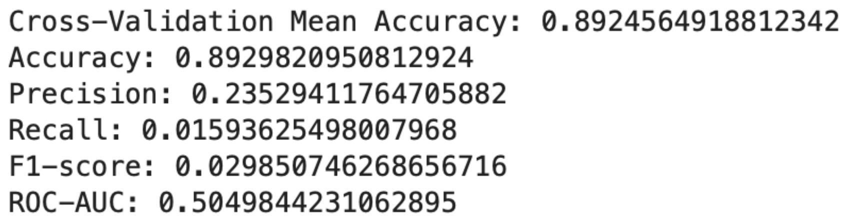
  Subsequently, we employ a stratified splitting strategy to divide the dataset into training and test sets of 80:20.

- **Model Creation:**
  With the dataset prepared, we proceed to construct our logistic regression model using a pipeline-based approach. This involves defining distinct preprocessing steps tailored to handle numerical and categorical data.

  Numerical features undergo standardization, a process that ensures all features are on the same scale, preventing certain features from dominating due to larger magnitudes.
  Categorical features, on the other hand, undergo one-hot encoding, transforming categorical variables into a binary format suitable for machine learning algorithms.

Finally, we combine these preprocessing steps into a cohesive pipeline and integrate a logistic regression classifier, which serves as the core predictive engine.

- **Model Evaluation:**
  With the model trained on the training dataset, we evaluate its performance using a comprehensive array of metrics.



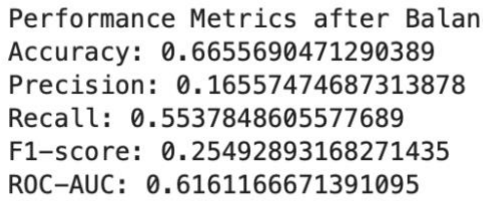
```
Cross-Validation Mean Accuracy: 0.8924564918812342
Accuracy: 0.8929820950812924
Precision: 0.23529411764705882
Recall: 0.01593625498007968
F1-score: 0.029850746268656716
ROC-AUC: 0.5049844231062895
```

- **Data Balancing:**
  Recognizing the presence of class imbalance, wherein one class (e.g., readmitted cases) significantly outnumbers the other, we employ the RandomOverSampler technique to rebalance the dataset.

  This technique involves oversampling the minority class (i.e., readmitted cases) to ensure that both classes are adequately represented in the dataset. By doing so, we mitigate the inherent bias towards the majority class, thereby enhancing the model's ability to learn from both classes equally.

- **Model Training with Balanced Data:**
  With the dataset rebalanced, we proceed to train a new logistic regression model using the balanced data. This model benefits from a more equitable representation of both classes, thereby minimizing the risk of bias towards the majority class By training on balanced data, the model can effectively learn the underlying patterns and relationships within the data, leading to improved predictive performance.

The performance metrics of the balanced model is



```
Performance Metrics after Balancing Data:
Accuracy: 0.6655690471290389
Precision: 0.16557474687313878
Recall: 0.5537848605577689
F1-score: 0.25492893168271435
ROC-AUC: 0.6161166671391095
```

Part 2: Improved Model

Data Preprocessing:

- **Dropping Irrelevant Columns:**
  Columns such as patient encounter IDs and weight measurements were dropped as they offered little relevance to predicting hospital readmissions. Including these columns could introduce noise and complexity without improving predictive accuracy.

- **Handling Missing Values:**
  '?' values were replaced with NaN to ensure consistent treatment of missing data across the dataset. Addressing missing values is crucial for maintaining data integrity and preventing biases in subsequent analyses.

- **Converting Categorical Labels:**
  Categorical labels in the 'readmitted' column were converted to binary format ('1' for readmitted and '0' for not readmitted) to simplify the classification task. This transformation aligned with the binary nature of readmission outcomes and facilitated clearer interpretation.

- **One-Hot Encoding:**
  Categorical columns were subjected to one-hot encoding to convert non-numeric labels into a numerical format suitable for model training. One-hot encoding preserves categorical information without imposing ordinality assumptions, ensuring unbiased representation of categorical variables.

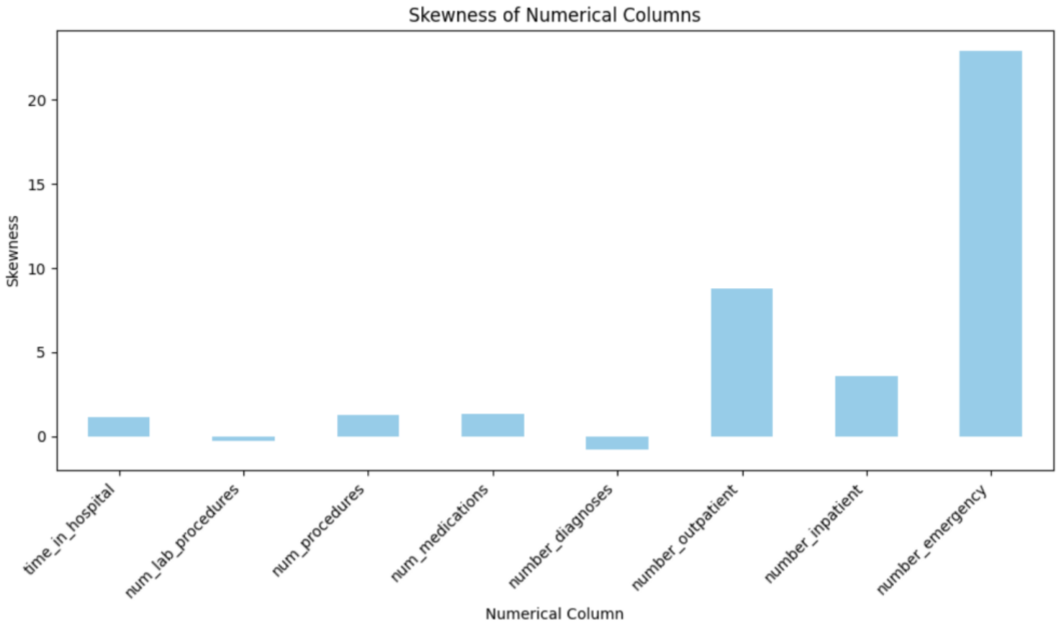
  Note : We didn’t use one hot encoding to diag_1, diag_2, diag_3 One-hot encoding creates a binary column for each unique value in the feature, resulting in a significant increase in the number of dimensions. This can lead to a high-dimensional dataset, making computations slower and more memory-intensive, especially for machine learning algorithms that are sensitive to the curse of dimensionality.

- **Handling Missing Values in Diagnosis Columns:**
  Rows with missing values in crucial diagnosis columns were dropped to maintain the integrity of diagnosis data used for predictive modelling. Diagnosis information is pivotal for predicting readmissions, and ensuring its completeness enhances the reliability of predictive models.

  Note: We dropped those missing values because there were only we rows with the missing values and as there were many distinct values in diag_1, diag_2, diag_3 we didn’t want to replace them with the mode as well because it may lead to inaccurate data.

- **Outlier Detection and Removal:**
We checked skewness to identify potential outliers in numerical columns. Outliers identified through skewness analysis were removed using the Interquartile Range (IQR) method. This step aimed to improve model robustness and reliability by mitigating the impact of outliers on predictive accuracy and statistical analyses.

We have removed the outliers based on the below skewness graph.



Building the Improved Model

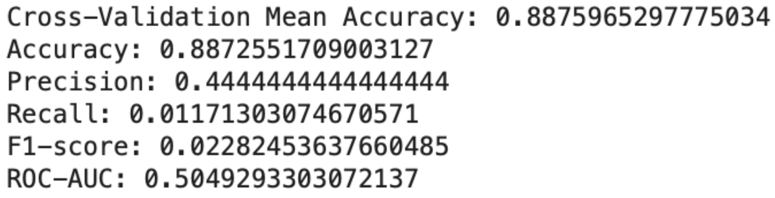
- **Data Preprocessing:**
Non-numeric columns were excluded from consideration, focusing solely on numerical attributes pertinent to readmission risk assessment in diabetic patients. Addressing missing data is imperative to maintain data integrity; therefore, appropriate imputation techniques were employed to handle missing values. Furthermore, outliers, which could potentially skew the analysis, were identified and eliminated using the Interquartile Range (IQR) method, thereby enhancing the dataset's quality and reliability.

- **Feature Selection and Engineering:**
Initially we performed correlation analysis to gauge the strength of association between numerical features and the target variable, 'readmitted.' Features exhibiting a correlation coefficient exceeding a predefined threshold were retained for further analysis. Subsequently, Random Forest Classifier was employed to assess the importance of individual features, aiding in the identification of influential predictors. To refine the feature set, Recursive Feature Elimination (RFE) was implemented, iteratively selecting the top features based on their importance, thereby enhancing the model's interpretability and performance.

- **Model Training and Evaluation:** Following feature selection, a Logistic Regression model was chosen for its suitability in binary classification tasks and interpretability. The dataset was partitioned into training and test sets to facilitate model training and evaluation. The Logistic Regression model was trained on the training data and subsequently evaluated using cross-validation to estimate its performance on unseen data. Performance metrics such as accuracy, precision, recall, F1-score, and Receiver Operating Characteristic Area Under the Curve (ROC-AUC) were computed to comprehensively assess the model's predictive capability and robustness.

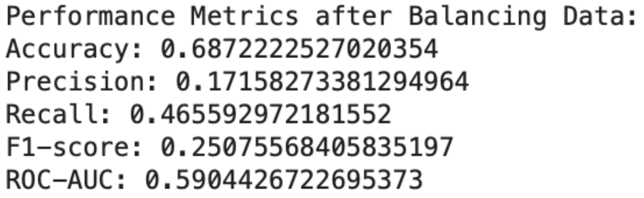
- **Handling Class Imbalance:** RandomOverSampler was employed to balance the dataset by increasing the representation of minority class instances. The balanced dataset was then used to train a Logistic Regression model, allowing for a fair comparison of model performance against the unbalanced dataset.

The performance metrics are as follows:



```
Cross-Validation Mean Accuracy: 0.8875965297775034
Accuracy: 0.8872551709003127
Precision: 0.4444444444444444
Recall: 0.01171303074670571
F1-score: 0.02282453637660485
ROC-AUC: 0.5049293303072137
```

After balancing the data:



```
Performance Metrics after Balancing Data:
Accuracy: 0.6872222527020354
Precision: 0.17158273381294964
Recall: 0.465592972181552
F1-score: 0.25075568405835197
ROC-AUC: 0.5904426722695373
```

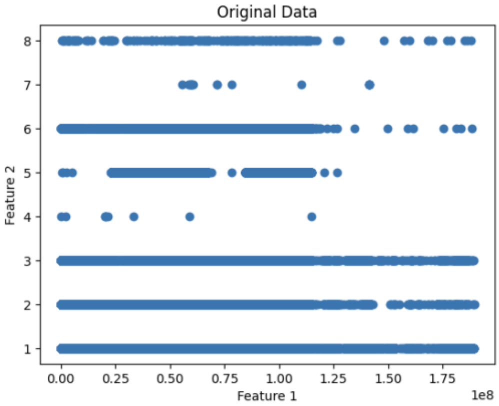
K Means Clustering

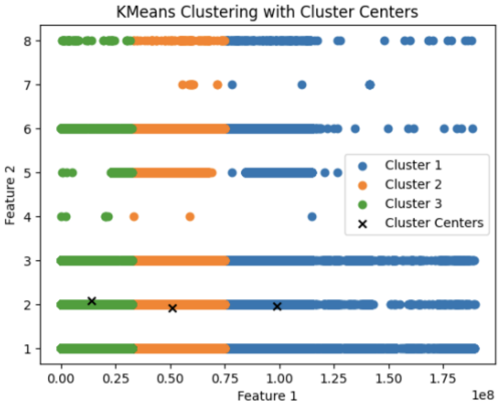
- **Standardization:**
  We began by standardizing the dataset to address any variations in scale among features. Through this process, we transformed numerical attributes to have a mean of zero and a standard deviation of one. This standardization ensured that the clustering process was not influenced by differences in measurement units across the dataset.

- **Determination of Cluster Count:**
  Next, we deliberated on the optimal number of clusters by combining our domain expertise with an exploratory analysis of the dataset. Through this collaborative effort, we decided to partition the dataset into three clusters. This decision aimed to capture distinct groupings of diabetic patients based on their numerical attributes, facilitating a comprehensive understanding of patient profiles.

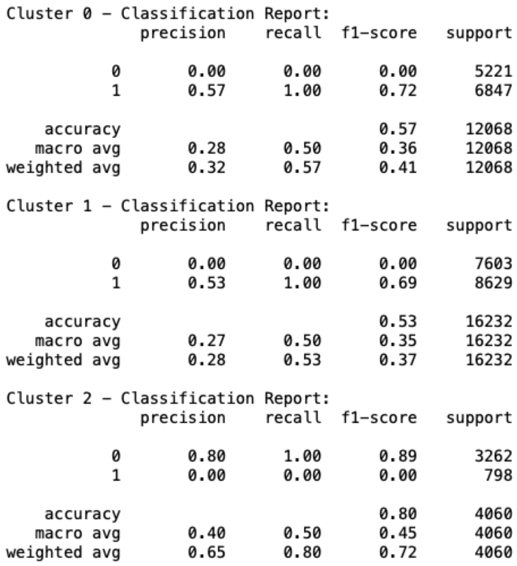
- **Application of K-Means Algorithm:**
  Leveraging the standardized dataset, we applied the K-Means algorithm with three clusters as the predetermined count. This iterative algorithm assigned each data point to the nearest cluster centroid, iteratively refining cluster centroids to minimize the within-cluster sum of squares. By executing this algorithm, we delineated clear boundaries between clusters, enabling a nuanced interpretation of patient groupings.

- **Visualization of Clusters:** To derive actionable insights from the clustering results, we employed visualization techniques. Utilizing scatter plots, we visualized the dispersion of data points within the feature space, with each cluster depicted by a unique hue. Additionally, the spatial representation of cluster centroids provided a visual understanding of their relative positions within the feature space, aiding in the interpretation of cluster characteristics and facilitating informed decision-making.





The performance metrics are follows:



```
Cluster 0 - Classification Report:
              precision    recall  f1-score   support

           0       0.00      0.00      0.00      5221
           1       0.57      1.00      0.72      6847

    accuracy                           0.57     12068
   macro avg       0.28      0.50      0.36     12068
weighted avg       0.32      0.57      0.41     12068

Cluster 1 - Classification Report:
              precision    recall  f1-score   support

           0       0.00      0.00      0.00      7603
           1       0.53      1.00      0.69      8629

    accuracy                           0.53     16232
   macro avg       0.27      0.50      0.35     16232
weighted avg       0.28      0.53      0.37     16232

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

           0       0.80      1.00      0.89      3262
           1       0.00      0.00      0.00       798

    accuracy                           0.80      4060
   macro avg       0.40      0.50      0.45      4060
weighted avg       0.65      0.80      0.72      4060
```

Analysis of all the models developed and observations of the important points:

- **Initial Model Performance**: The initial model, with aggressive data cleansing, achieved moderate accuracy but poor precision, recall, and F1-score. This indicates that while the model was able to classify the majority class accurately, it struggled with the minority class, likely due to imbalanced data.
- **Improved Model Performance**: After employing one-hot encoding and imputation techniques, there was a slight improvement in precision, indicating better performance in correctly identifying positive instances. However, recall remained low, suggesting that the model still struggled to capture all positive instances, resulting in a low F1-score.
- **Balancing Data**: Balancing the data led to improvements in recall for both the initial and improved models, indicating better performance in capturing positive instances. However, this improvement came at the cost of decreased precision, as reflected in the lower F1-score.
- **K-Means Clustering**: Implementing K-means clustering and modeling within each cluster revealed varying performance across clusters. Cluster 0 showed high precision but low recall, suggesting that while it correctly identified negative instances, it missed many positive instances. Cluster 1 had moderate precision and recall, indicating a balanced performance between identifying both positive and negative instances. Cluster 2 had high recall but very low precision, indicating a tendency to classify many instances as positive, leading to a high false positive rate.
- **Overall Assessment**: While the models showed improvements in certain performance metrics after data preprocessing and clustering, there are still areas for enhancement. Further fine-tuning of the models, exploring different algorithms, or employing advanced techniques like ensemble learning or feature engineering may help improve overall performance, especially in terms of achieving a better balance between precision and recall. Additionally, careful

consideration of the business context and the cost associated with false positives and false negatives is crucial in selecting the most suitable model for deployment.

Conclusion:

In conclusion, the analysis highlights the iterative nature of model development and the importance of thoughtful data preprocessing techniques. While aggressive data cleansing initially improved model accuracy, it resulted in poor performance metrics for minority class prediction. Employing one-hot encoding and imputation techniques led to slight improvements, particularly in precision, but overall model performance remained suboptimal, especially in terms of recall and F1-score.

Balancing the data helped enhance recall at the expense of precision, indicating a trade-off between correctly identifying positive instances and minimizing false positives. However, even with balanced data, there's room for improvement in achieving a better balance between precision and recall.

Implementing K-means clustering and modeling within each cluster revealed varying performance across clusters, with some clusters showing better precision but lower recall, and vice versa. This suggests the potential for targeted interventions based on cluster-specific characteristics.

Overall, while the models showed incremental improvements through preprocessing and clustering techniques, further refinement and exploration of alternative algorithms or advanced methodologies are warranted to achieve a more balanced and robust predictive model, especially in handling imbalanced data and optimizing the trade-off between precision and recall. Additionally, aligning model evaluation metrics with specific business objectives and considering the practical implications of false positives and false negatives are crucial steps in selecting the most suitable model for deployment.