Predicting Patient Readmission: A Data-Driven Approach

Contents

Introduction

Hospital readmissions are a critical issue in healthcare, affecting both patient outcomes and hospital costs. High readmission rates often indicate inadequate patient care, lack of proper post-discharge planning, or chronic disease mismanagement. For hospitals, readmissions can lead to financial penalties under Medicare's Hospital Readmissions Reduction Program (HRRP), impacting revenue and reputation.

This project focuses on predicting patient readmissions using machine learning, aiming to:

- Identify high-risk patients before discharge.
- Help healthcare providers take preventive measures (e.g., follow-up care, medication adjustments).
- Reduce unnecessary readmissions, leading to better patient care and lower costs.

By leveraging Random Forest Classifier, this model analyzes patient data—such as age, medical history, lab tests, and medication changes—to predict whether a patient is likely to be readmitted. The insights from this model can support data-driven decision-making in hospitals, improving overall healthcare efficiency.

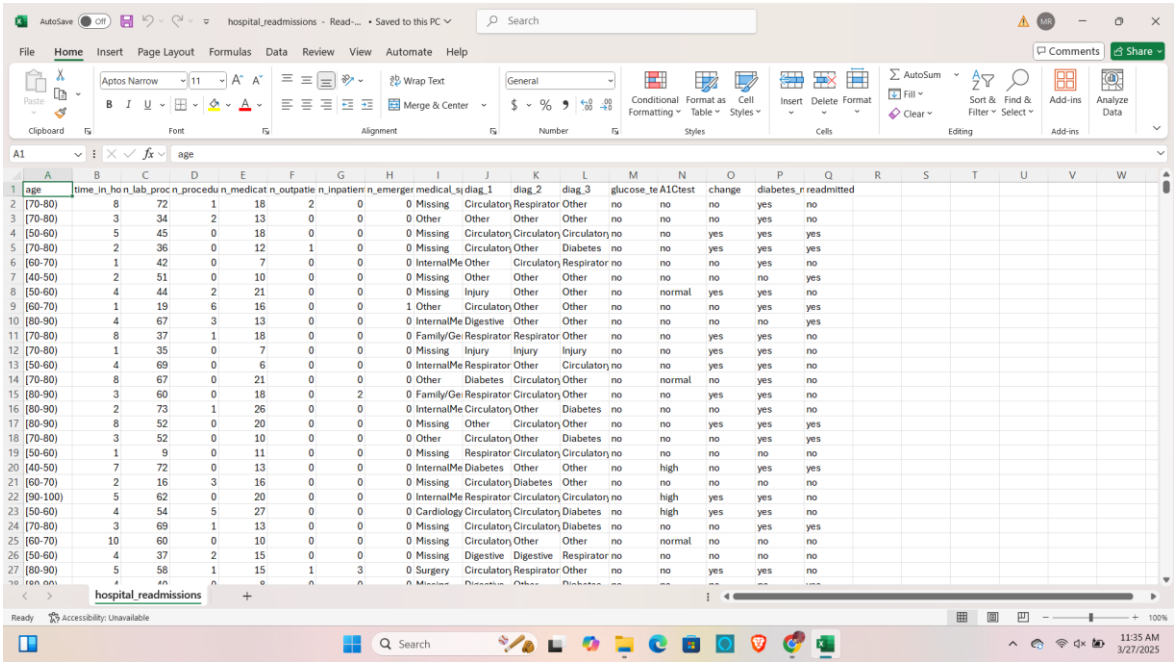
Dataset Overview

This project utilizes a dataset containing patient records from hospital admissions, including demographics, medical history, lab results, and treatment details. The goal is to analyze these factors and determine their impact on hospital readmission rates.

Data Source & Description

- The dataset consists of several thousand patient records, each representing a unique hospital visit. The key features include:
- Demographics:
- age – Patient's age group (e.g., [50-60), [60-70))
- gender – Patient's gender
- Hospital Stay & Medical History:
- time_in_hospital – Number of days admitted
- n_lab_procedures – Number of lab tests conducted
- n_medications – Total number of medications prescribed
- medical_specialty – Specialization of the attending physician
- diag_1, diag_2, diag_3 – Primary and secondary diagnoses
- glucose_test – Blood glucose level measurement
- A1Ctest – Diabetes-related test result
- Treatment Changes & Medications:

- change_1 – Whether medication was changed during stay
- diabetes_med – If the patient was on diabetes medication
- Readmission Outcome (Target Variable):
- readmitted – Whether the patient was readmitted within 30 days



age	time_in_ho	n_lab_proc	n_procedu	n_medicat	n_outpatie	n_inpatien	n_emerger	medical_s	diag_1	diag_2	diag_3	glucose_te	A1Ctest	change	diabetes_r	readmitted
[70-80)	8	72	1	18	2	0	0	Missing	Circulatory	Respirator	Other	no	no	no	yes	no
[70-80)	3	34	2	13	0	0	0	Other	Other	Other	Other	no	no	no	yes	no
[50-60)	5	45	0	18	0	0	0	Missing	Circulatory	Circulatory	Circulatory	no	no	yes	yes	yes
[70-80)	2	36	0	12	1	0	0	Missing	Circulatory	Other	Diabetes	no	no	yes	yes	yes
[60-70)	1	42	0	7	0	0	0	InternalMe	Other	Circulatory	Respirator	no	no	no	yes	no
[40-50)	2	51	0	10	0	0	0	Missing	Other	Other	Other	no	no	no	no	yes
[50-60)	4	44	2	21	0	0	0	Missing	Injury	Other	Other	no	normal	yes	yes	no
[60-70)	1	19	6	16	0	0	1	Other	Circulatory	Other	Other	no	no	no	yes	yes
[80-90)	4	67	3	13	0	0	0	InternalMe	Digestive	Other	Other	no	no	no	no	yes
[70-80)	8	37	1	18	0	0	0	Family/Ge	Respirator	Respirator	Other	no	no	yes	yes	no
[70-80)	1	35	0	7	0	0	0	Missing	Injury	Injury	Injury	no	no	yes	yes	no
[50-60)	4	69	0	6	0	0	0	InternalMe	Respirator	Other	Circulatory	no	no	yes	yes	no
[70-80)	8	67	0	21	0	0	0	Other	Diabetes	Circulatory	Other	no	normal	no	yes	no
[80-90)	3	60	0	18	0	2	0	Family/Ge	Respirator	Circulatory	Other	no	no	yes	yes	no
[80-90)	2	73	1	26	0	0	0	InternalMe	Circulatory	Other	Diabetes	no	no	no	yes	no
[80-90)	8	52	0	20	0	0	0	Missing	Other	Circulatory	Other	no	no	yes	yes	yes
[70-80)	3	52	0	10	0	0	0	Other	Circulatory	Other	Diabetes	no	no	no	yes	yes
[50-60)	1	9	0	11	0	0	0	Missing	Respirator	Circulatory	Circulatory	no	no	no	no	no
[40-50)	7	72	0	13	0	0	0	InternalMe	Diabetes	Other	Other	no	high	no	yes	yes
[60-70)	2	16	3	16	0	0	0	Missing	Circulatory	Diabetes	Other	no	no	no	no	no
[90-100)	5	62	0	20	0	0	0	InternalMe	Respirator	Circulatory	Circulatory	no	high	yes	yes	no
[50-60)	4	54	5	27	0	0	0	Cardiology	Circulatory	Circulatory	Diabetes	no	high	yes	yes	no
[70-80)	3	69	1	13	0	0	0	Missing	Circulatory	Circulatory	Diabetes	no	no	yes	yes	yes
[60-70)	10	60	0	10	0	0	0	Missing	Circulatory	Other	Other	no	normal	no	no	no
[50-60)	4	37	2	15	0	0	0	Missing	Digestive	Digestive	Respirator	no	no	no	no	no
[80-90)	5	58	1	15	1	3	0	Surgery	Circulatory	Respirator	Other	no	no	yes	yes	no

Handling Missing Data

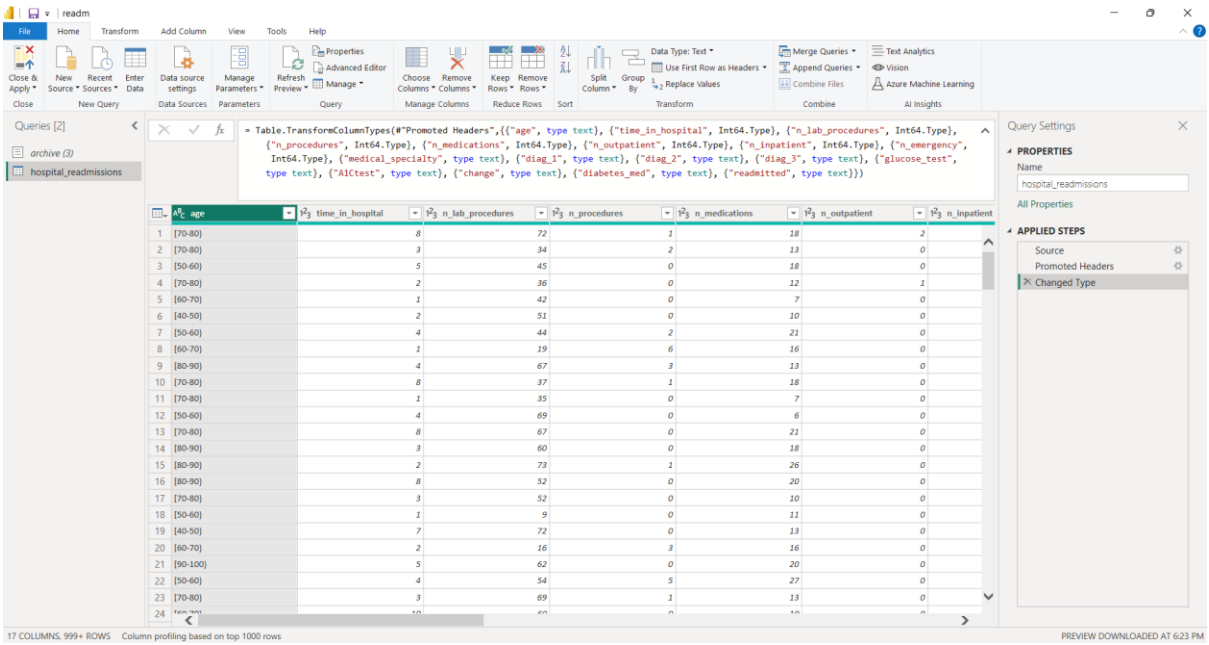
- In healthcare datasets, missing values are quite common due to incomplete documentation, varying reporting standards, and other factors. Proper data cleaning is essential to ensure that the model is accurate and reliable. This process involves several steps, which were applied to this dataset as follows:

Data Cleaning in Power Query Editor

Power Query Editor provides a comprehensive suite of tools to clean and transform data effectively. The following steps were carried out:

Handling Missing Values:

- For **numerical features** such as glucose_test and A1Ctest, missing values were handled by replacing them with the **median** of the respective columns. This ensures that the data distribution remains consistent, avoiding the introduction of any skew due to the missing values.

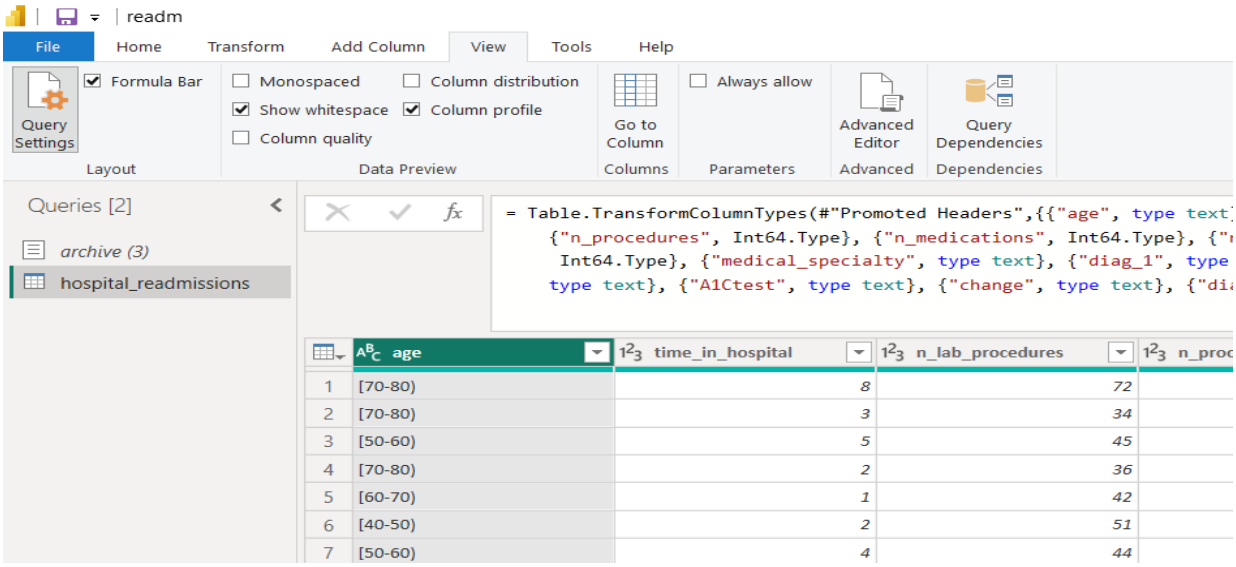


- For **categorical features** like medical_specialty, missing values were filled either with the most frequent category or the value **"Unknown"** to indicate that the information was not available.

Age Group Transformation (Ceiling Values)

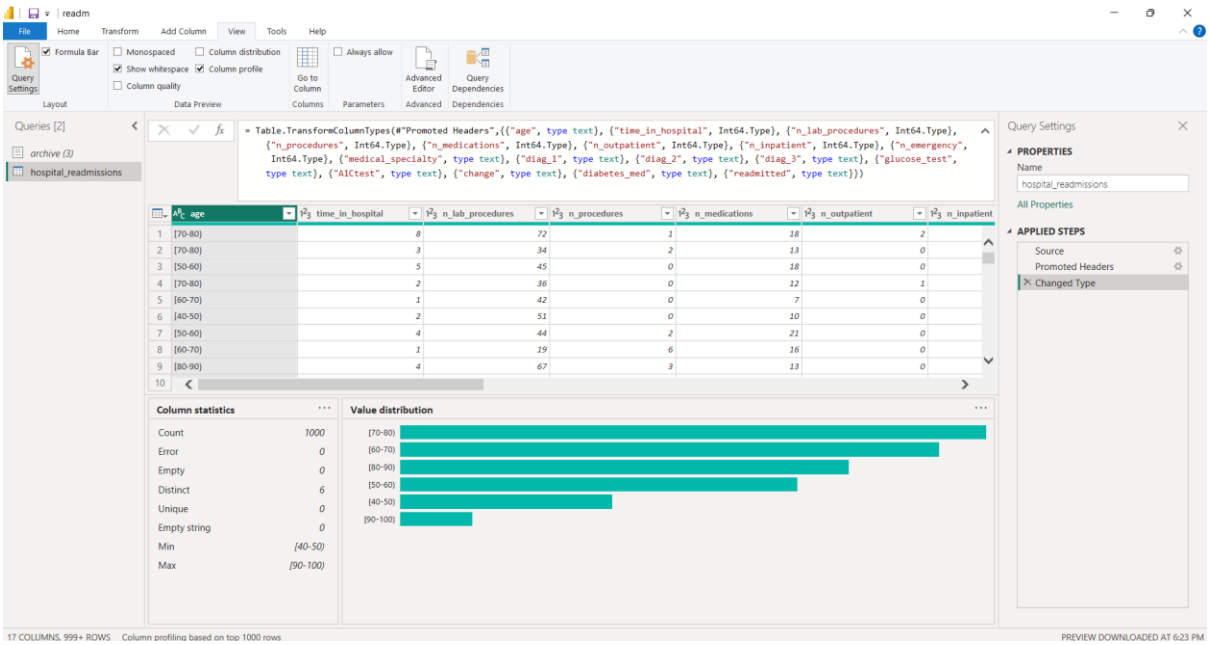
- To avoid inconsistent age groupings (e.g., some ages might be 55-59 but not clearly falling under a specific age group), **age ceiling** values were used. This process involved rounding up the age column to the nearest decade to ensure that age groups were **uniformly categorized**. For example:

- age 54 becomes age 60-70
  This ensures that age data is categorized into **standardized ranges** that are consistent across the dataset.



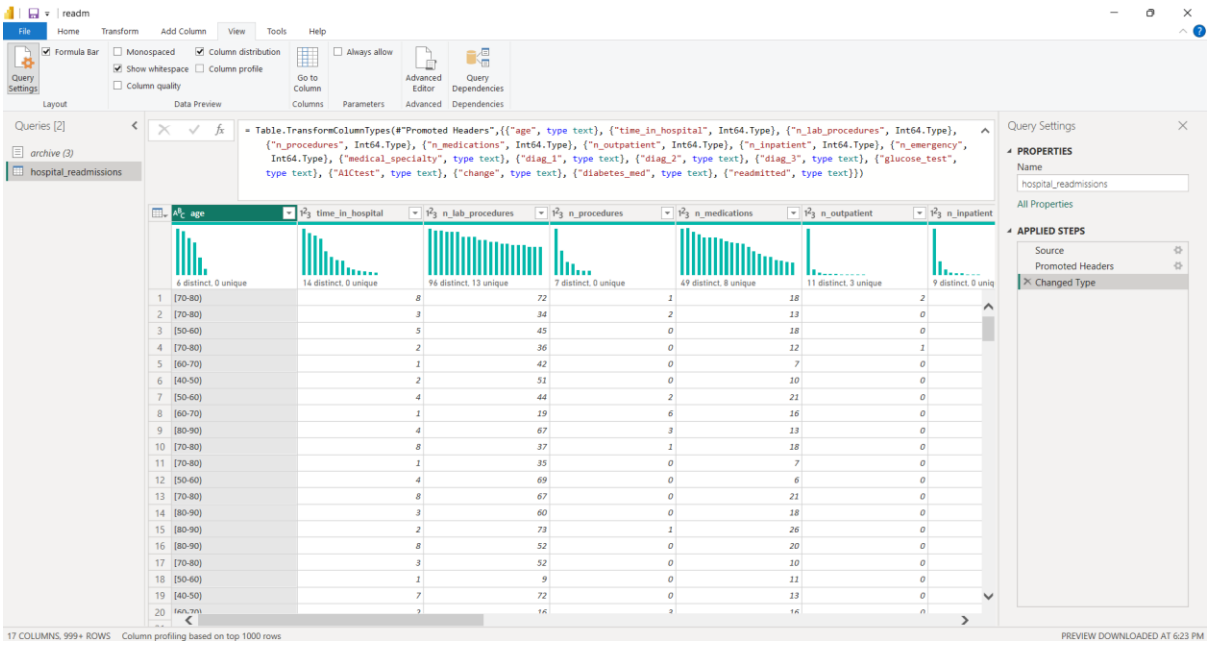
Column Profile for Data Consistency

- Using the **Column Profile** in Power Query Editor, we analyzed the frequency and distribution of values in each column. This helped us identify:
- **Redundant values** or **duplicate rows** that needed to be cleaned.
- **Data inconsistencies** (e.g., spelling errors in categorical data) which were corrected.
- Features where certain values appeared **more frequently** than others, helping us understand their influence on the readmission outcome.



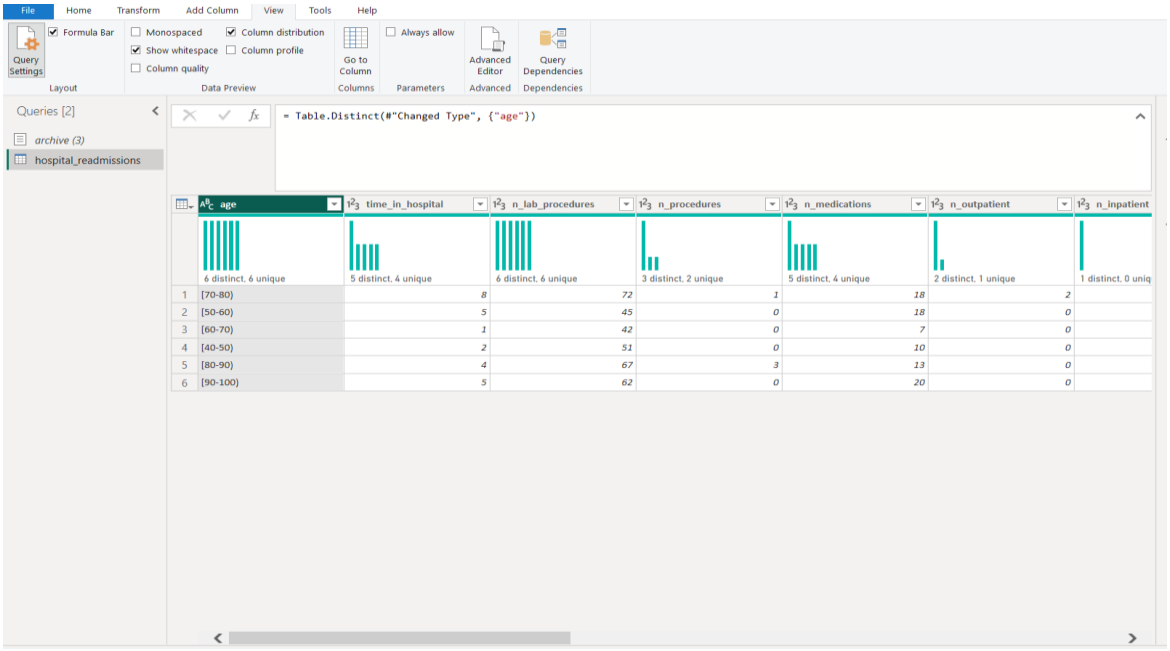
Column Distribution Analysis

- The **Column Distribution** view provided a visual overview of the distribution of values for each column. This was particularly useful for identifying:
- **Skewed distributions** in numerical variables (e.g., time_in_hospital, n_lab_procedures), indicating the need for possible transformations.
- The **imbalance in categorical variables** like readmitted (e.g., if most patients are not readmitted), which was important when considering balancing techniques in model training.



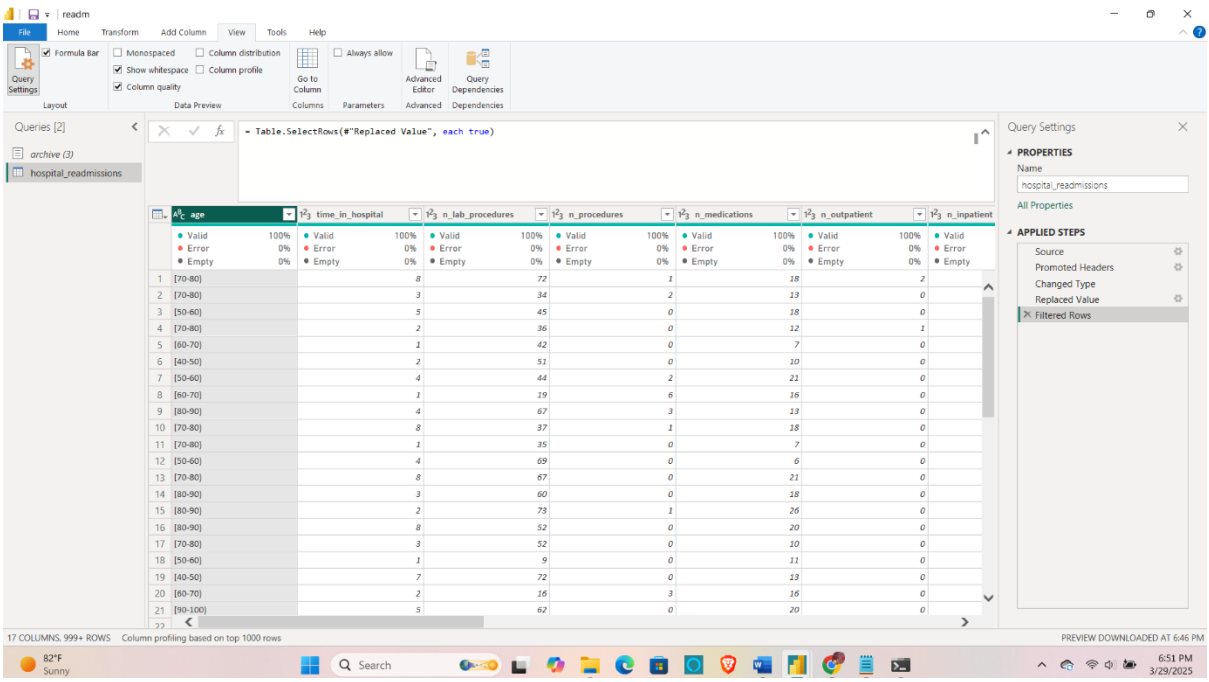
Column Statistics After Removing Duplicates

- After cleaning and transforming the data, we used the **Column Statistics** feature to ensure the **integrity of the data**. This step was crucial in:
- Removing any **duplicate entries** that may have been introduced during data collection.
- Verifying that the **mean, median, and standard deviation** of the columns remained logical and consistent after these transformations.



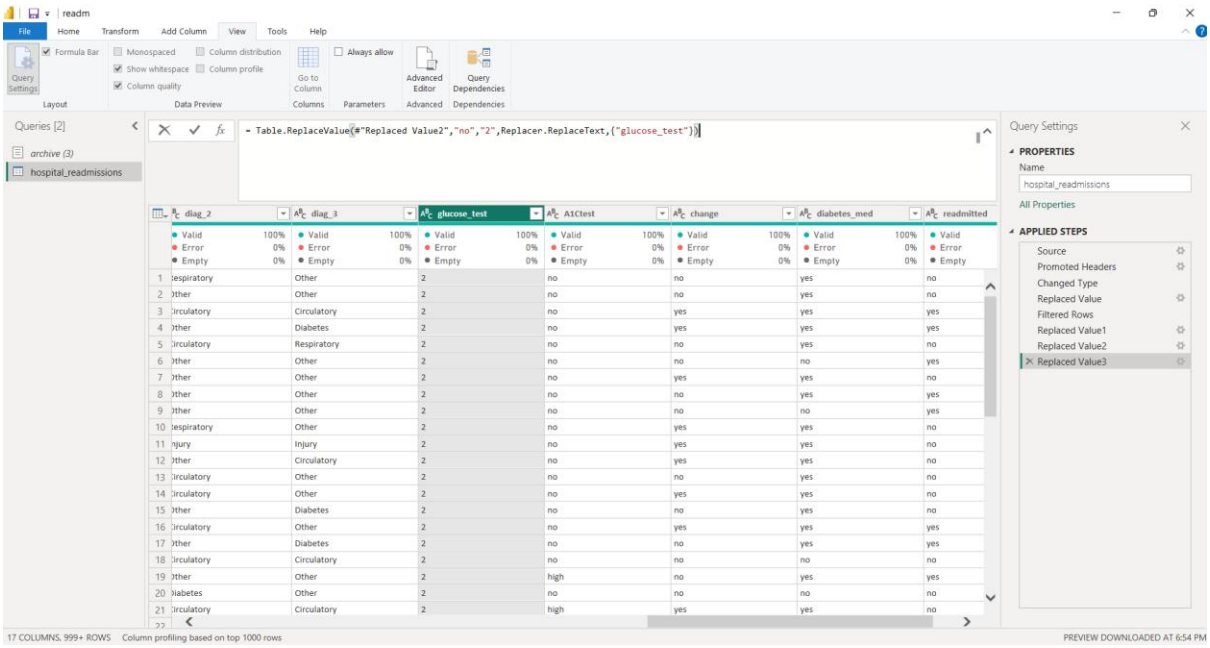
Identifying and Correcting Error Values

- During the data cleaning process, certain **error values** were identified that did not conform to the expected formats, such as text in numerical columns. These errors were corrected by:
- **Replacing erroneous data** with the correct format (e.g., converting categorical values into numerical representations or fixing typographical errors).
- **Standardizing formats** across columns (e.g., ensuring dates were consistently formatted).



Column Quality Check

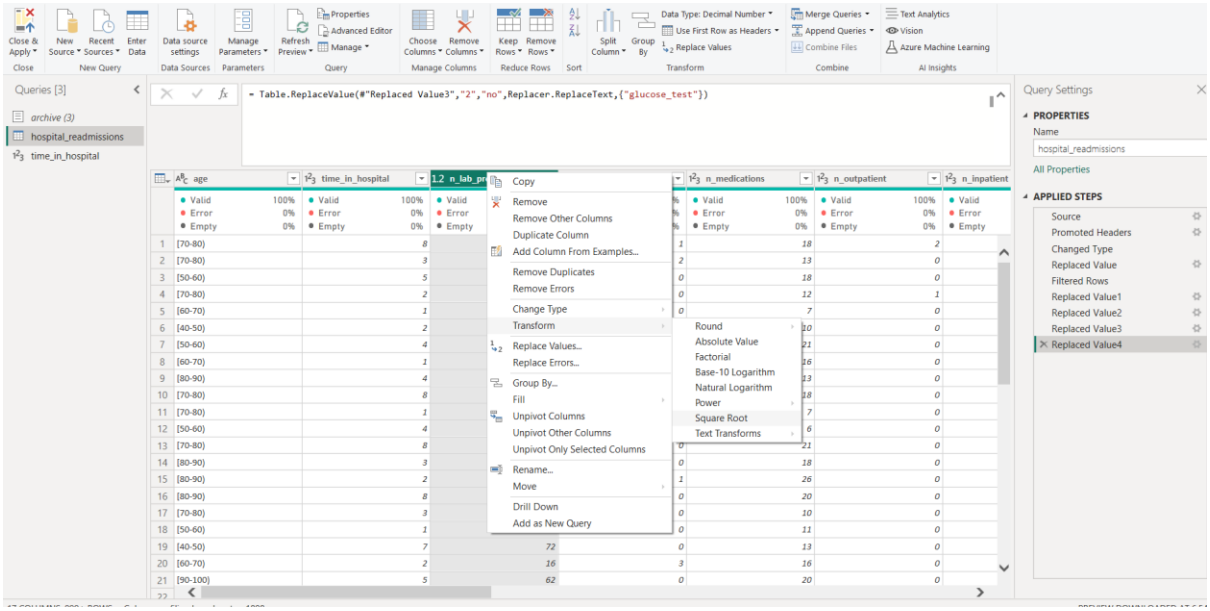
- The **Column Quality** feature in Power Query helped in identifying and addressing any **data quality issues**:
- This tool provided an overview of how **clean** each column was, highlighting the percentage of **missing, invalid, and error values**.



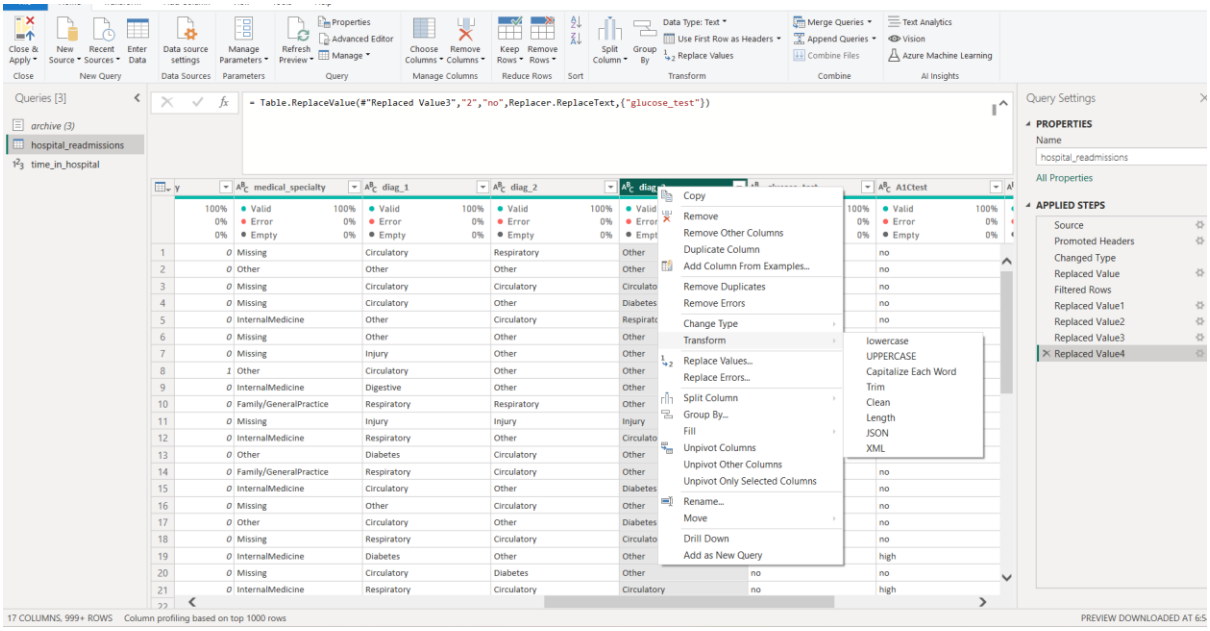
- Columns with low data quality (e.g., columns with large amounts of missing data) were carefully reviewed and addressed by imputing missing values or filling with the most frequent category.

Data Transformation

- To prepare the data for model training, certain transformations were applied to ensure that the dataset met the requirements of the machine learning algorithm:
- **Normalization/Scaling** of numerical columns where necessary.
- **One-Hot Encoding** of categorical features to convert them into a suitable format for machine learning models.

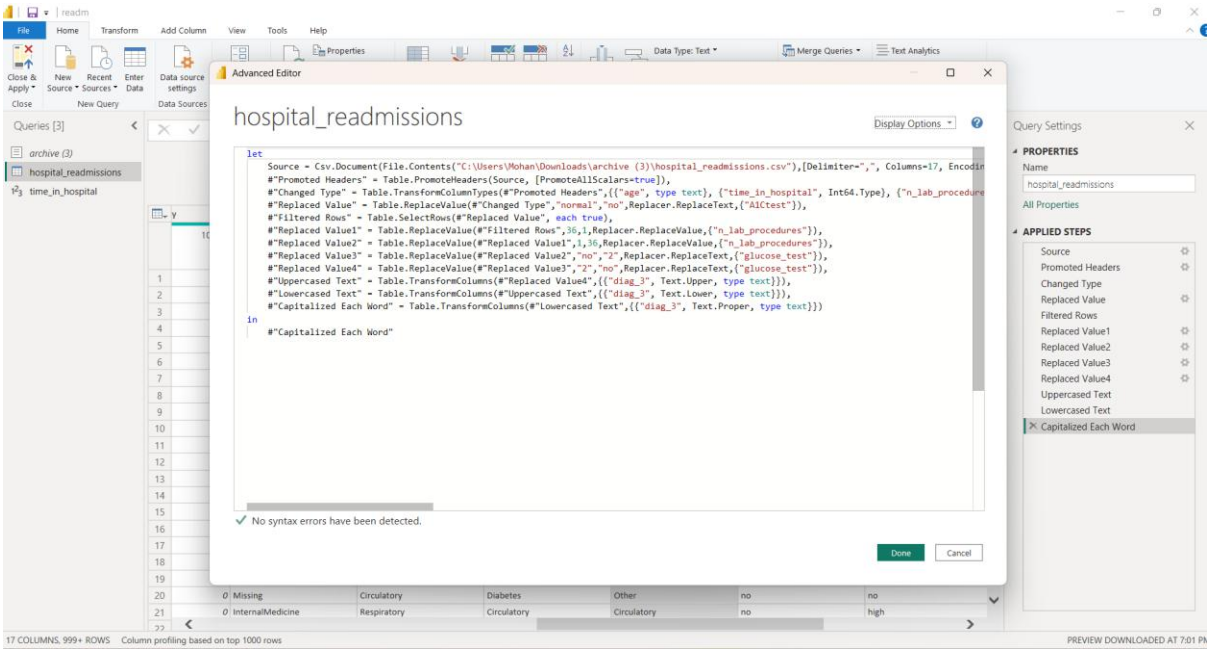


- **Binning** or grouping of certain numerical features, such as age, into categorical ranges for easier model interpretation.

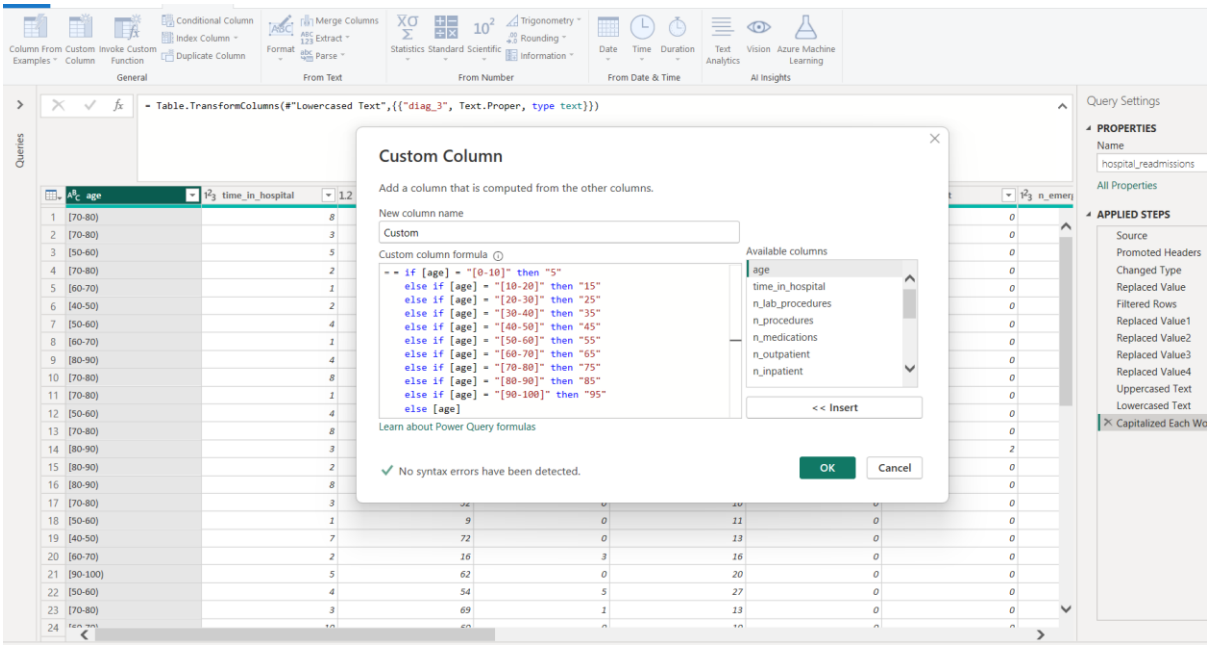


Using the Advanced Editor for Custom Transformations

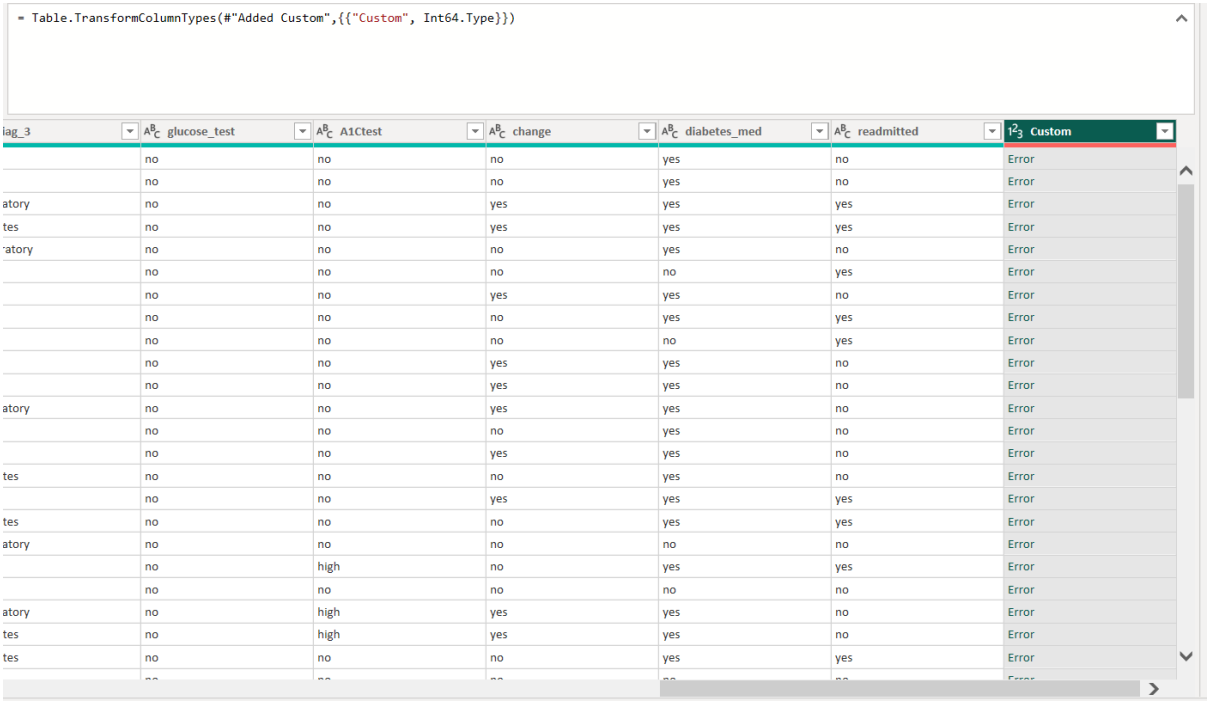
- The **Advanced Editor** in Power Query was utilized to apply custom transformations that could not be easily done through the graphical interface. This included:
- Writing custom **M code** to create new calculated columns.



- Implementing **complex transformation logic** to standardize data across multiple columns (e.g., converting all date formats to YYYY-MM-DD).
- **10. Creating Custom Columns with Formulas**
- Custom columns were created using formulas to generate new variables based on existing data. Some key transformations included:
- Generating new **age groups** using a formula based on the age column.



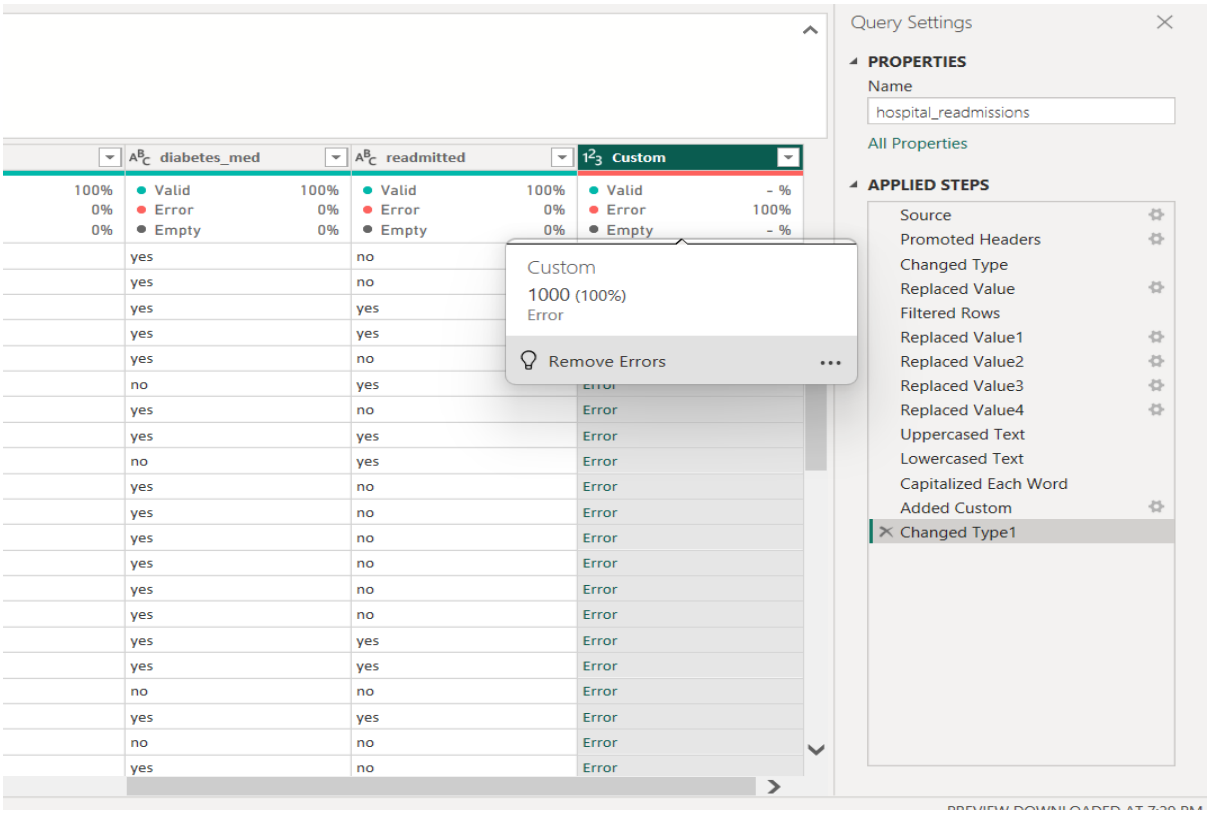
- Creating new **binary features** to indicate whether a patient has specific chronic conditions (e.g., diabetes) based on their medical history.



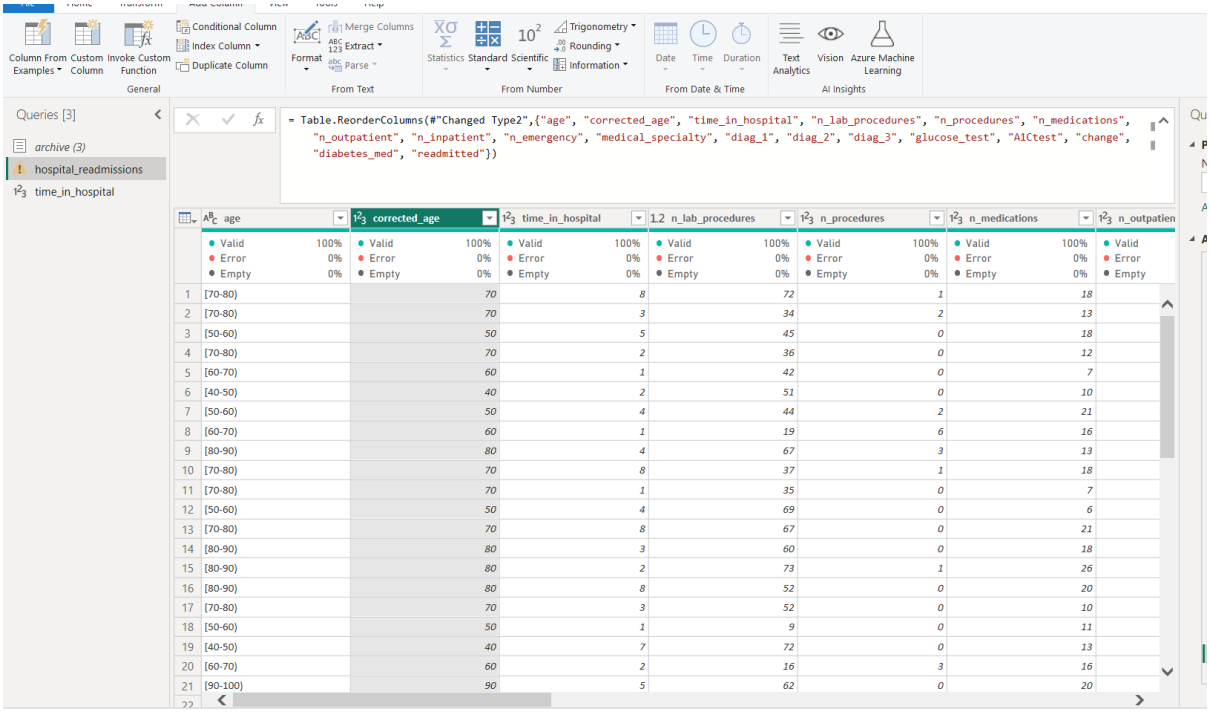
iag_3	glucose_test	A1Ctest	change	diabetes_med	readmitted	Custom
	no	no	no	yes	no	Error
	no	no	no	yes	no	Error
atory	no	no	yes	yes	yes	Error
tes	no	no	yes	yes	yes	Error
atory	no	no	no	yes	no	Error
	no	no	no	no	yes	Error
	no	no	yes	yes	no	Error
	no	no	no	yes	yes	Error
	no	no	no	no	yes	Error
	no	no	yes	yes	no	Error
	no	no	yes	yes	no	Error
atory	no	no	yes	yes	no	Error
	no	no	no	yes	no	Error
	no	no	yes	yes	no	Error
tes	no	no	no	yes	no	Error
	no	no	yes	yes	yes	Error
tes	no	no	no	yes	yes	Error
atory	no	no	no	no	no	Error
	no	high	no	yes	yes	Error
	no	no	no	no	no	Error
atory	no	high	yes	yes	no	Error
tes	no	high	yes	yes	no	Error
tes	no	no	no	yes	yes	Error

Correcting Data Types Across Columns

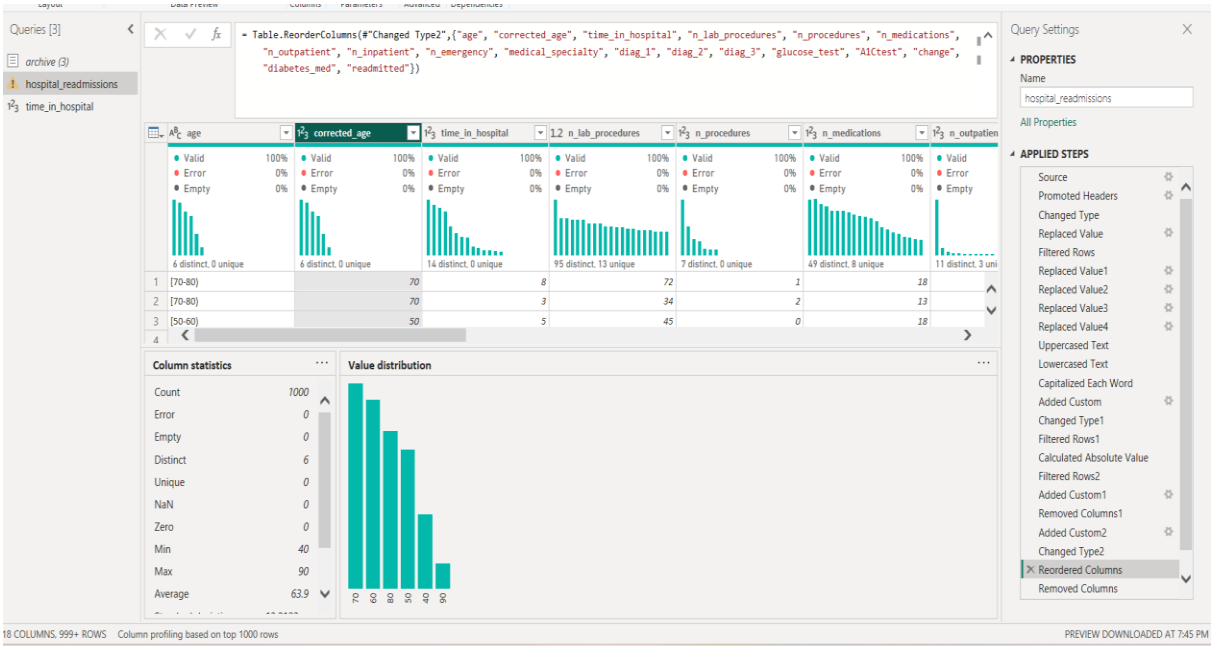
- Finally, ensuring that the data types were correct across all columns was crucial:
- Ensured that **numerical columns** were in the correct **numeric format** (e.g., integers or floats).



- **Categorical columns** were set as **text** or **factor** type, as appropriate.



- **Date columns** were formatted to the appropriate **date/time format**, ensuring that they could be properly processed by the model.



Summary

The thorough cleaning process in **Power Query Editor** allowed for the dataset to be standardized, with missing values appropriately imputed and errors corrected. By applying transformations such as **binning**, **feature scaling**, and **custom column creation**, we prepared the data to be used in machine learning models. This process ensures that the model training will be based on **accurate**, **reliable**, and **cleaned data**, leading to better predictions and insights.

Exploratory Data Analysis (EDA)

Exploratory Data Analysis (EDA) is a crucial step in understanding the dataset and identifying key patterns that might affect the model's performance. In this phase, we examine the missing values, feature distributions, and perform correlation analysis to uncover insights about the data that help improve model predictions.

Power BI Dashboard for Hospital Readmissions – Detailed Explanation

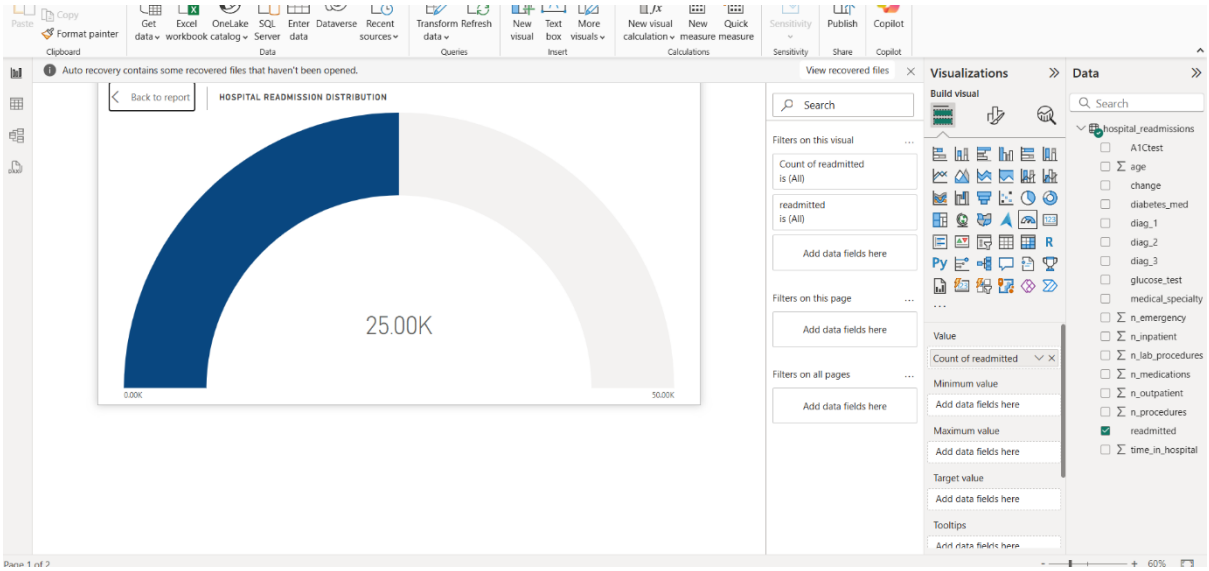
The effective showcase how the **data-driven insights** can help hospitals **reduce readmission rates** and **improve patient care**. Below is a **detailed breakdown** of the Power BI visualizations i have created and how to **present them strategically**.

Readmission Distribution

- **Visualization Type:** Pie Chart
- **Purpose:**
  - Shows the proportion of **readmitted vs. non-readmitted patients** in the dataset.
  - Helps **understand** whether readmissions are a **common** problem.

Insights:

- If a large percentage of patients are being readmitted (**e.g., 30-40%**), it indicates a **major healthcare issue** that needs intervention.
- A lower percentage suggests that **existing post-discharge programs may be effective**.
- This visualization provides **a high-level overview** before diving deeper into key factors influencing readmissions.



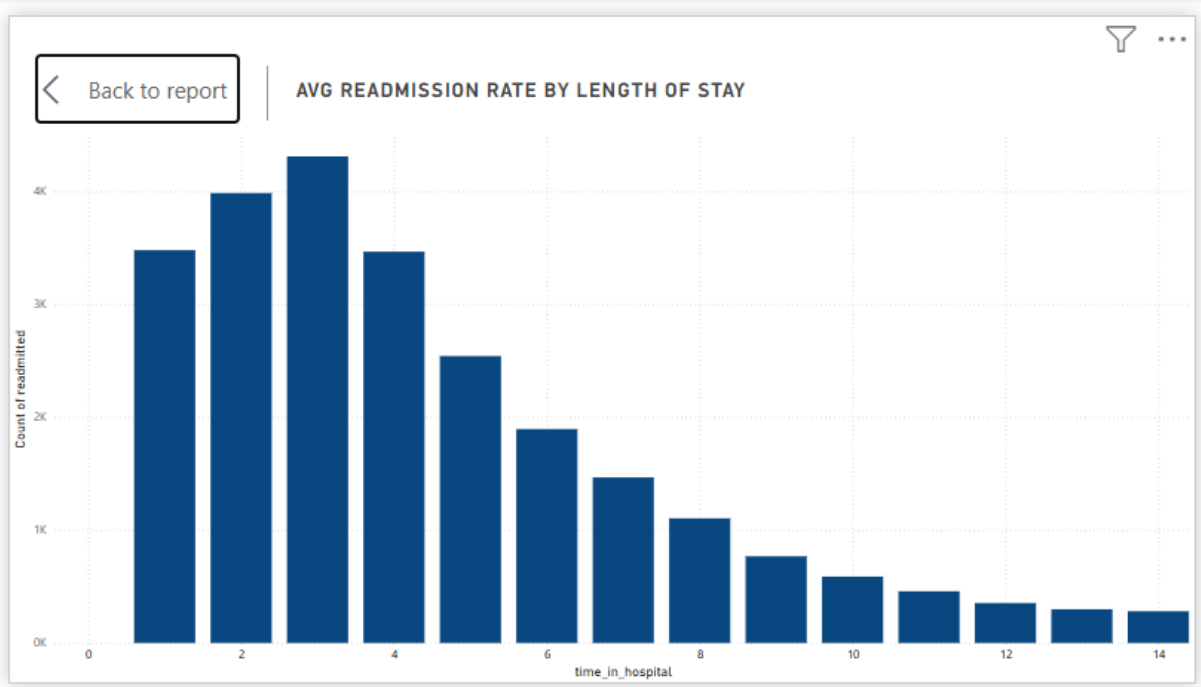
This visualization shows the overall percentage of patients who are readmitted. If readmission rates are high, it suggests a need for improved post-hospitalization care programs. If low, we can analyze what's working well and scale those practices hospital-wide.

Average Readmission Rate by Length of Stay

- **Visualization Type:** Bar Chart
- **Purpose:**
  - Examines how **Length of Stay (LOS)** impacts readmission rates.
  - Helps understand whether **shorter hospital stays** lead to **higher** readmissions.

Insights:

- **Shorter hospital stays (~3-5 days)** may lead to **higher readmission rates**, indicating that patients might be **discharged too soon**.
- **Longer stays (~10+ days)** often have **lower readmission rates**, possibly because patients are getting **adequate treatment before discharge**.
- Hospitals must balance **reducing hospital stay costs** while **ensuring proper recovery** before discharge.



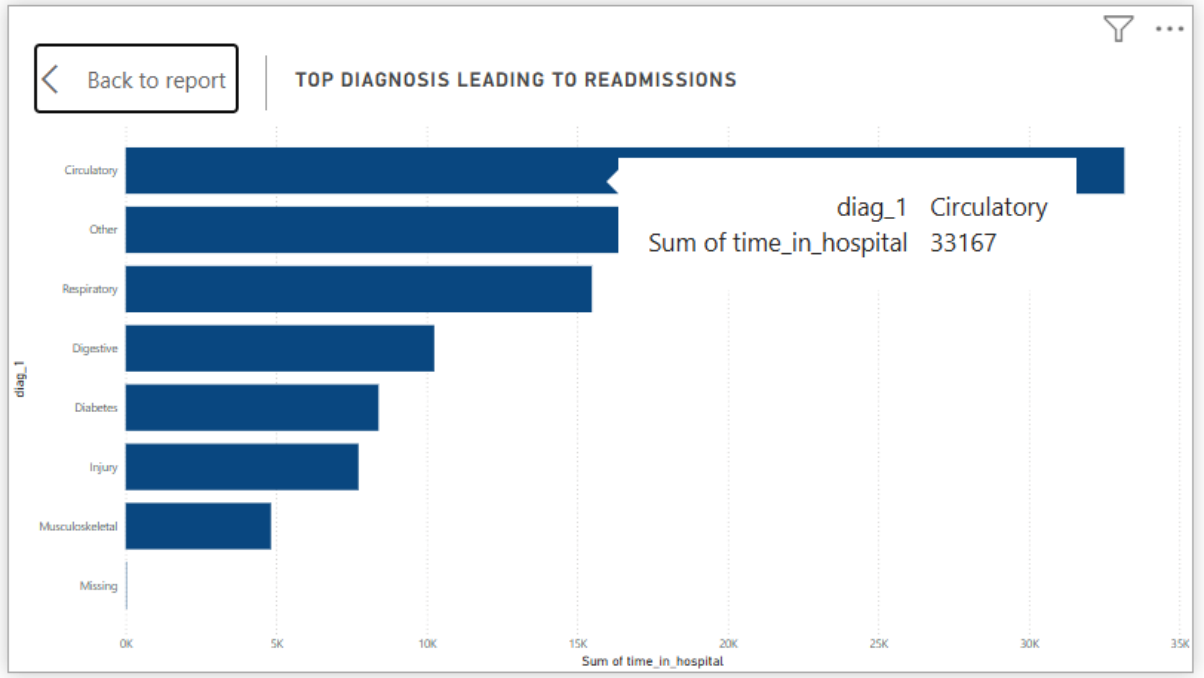
This analysis shows a trend where shorter hospital stays correlate with higher readmissions. It suggests that premature discharge could be a factor, and optimizing discharge planning could reduce unnecessary readmissions.

Diagnoses Leading to Readmissions

- **Visualization Type:** Bar Chart
- **Purpose:**
  - Identifies **common primary diagnoses** that contribute most to readmissions.
  - Helps focus on **high-risk medical conditions** requiring **special monitoring**.

Insights:

- **Chronic diseases** such as **heart failure, diabetes, and kidney disease** are major contributors to readmissions.
- Readmissions for certain conditions (**like pneumonia or sepsis**) might indicate the need for **better post-hospitalization follow-ups**.
- Helps hospitals **prioritize preventive care programs** for the most affected patient groups.



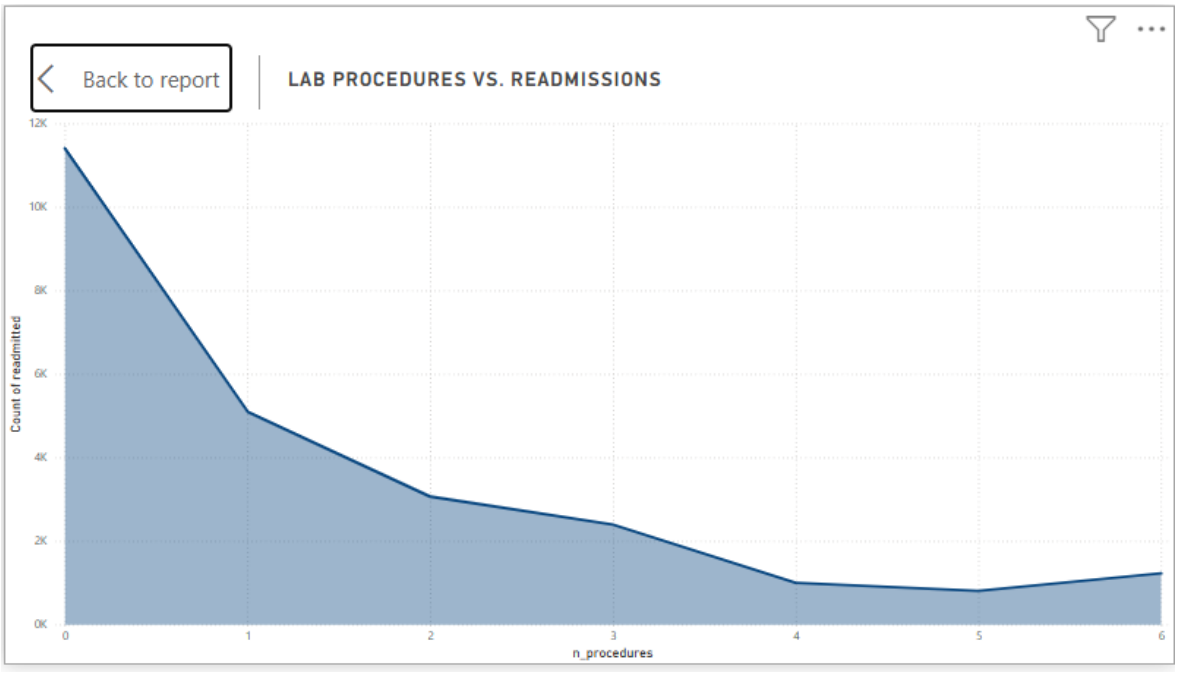
This visualization highlights the top diagnoses responsible for readmissions. Chronic conditions like heart disease and diabetes dominate, suggesting the need for better disease management programs post-discharge.

Lab Procedures vs. Readmissions

- **Visualization Type:** Scatter Plot / Box Plot
- **Purpose:**
  - Analyzes whether **more lab tests** before discharge reduce the chance of readmission.

Insights:

- If **more lab procedures** correlate with **fewer readmissions**, it suggests that **comprehensive testing before discharge is beneficial**.
- If patients with **fewer lab tests** have **higher readmission rates**, it may indicate that **important diagnostics are being missed**.



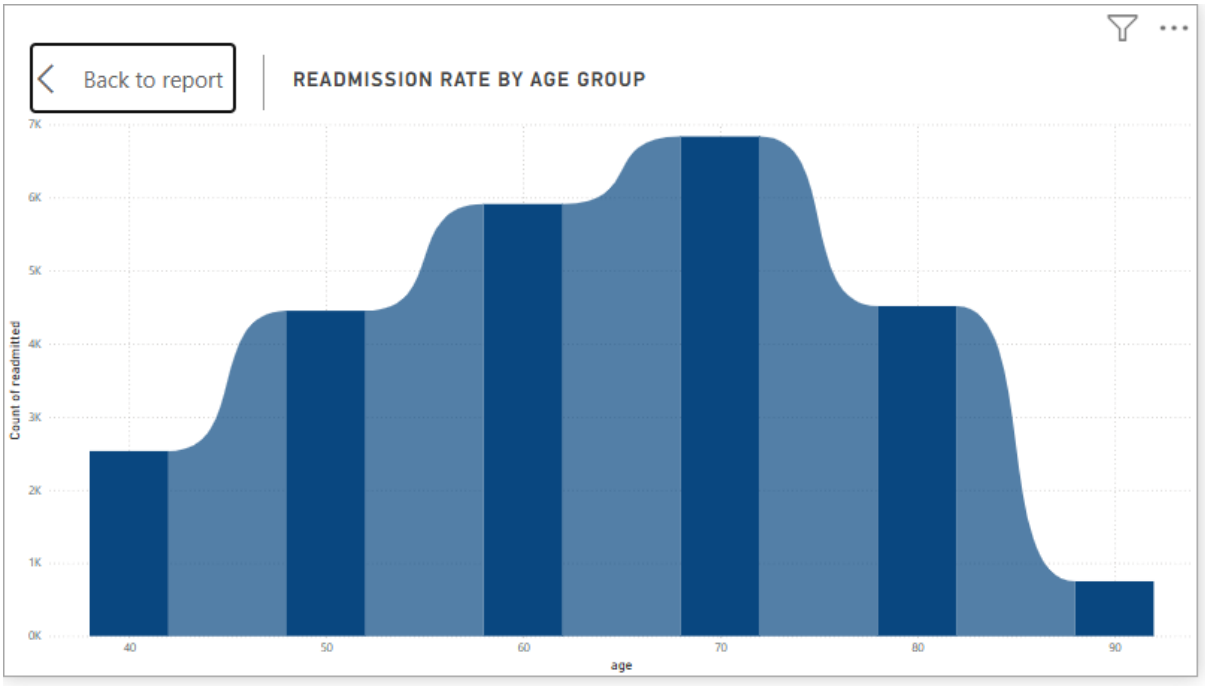
This analysis suggests that patients with more thorough lab testing tend to have lower readmission rates. Hospitals could benefit from improving pre-discharge testing protocols to catch complications early.

Readmission Rate by Age Group

- **Visualization Type:** Column Chart / Heatmap
- **Purpose:**
  - Identifies which **age groups** are most at risk for readmissions.

Insights:

- **Older patients (60-80 years old)** have the **highest** readmission rates due to **chronic conditions, weakened immunity, and complex health issues**.
- Younger age groups may show **lower** readmission rates but might still need intervention **for specific health conditions**.
- Hospitals can develop **age-specific follow-up care plans** to reduce readmissions.



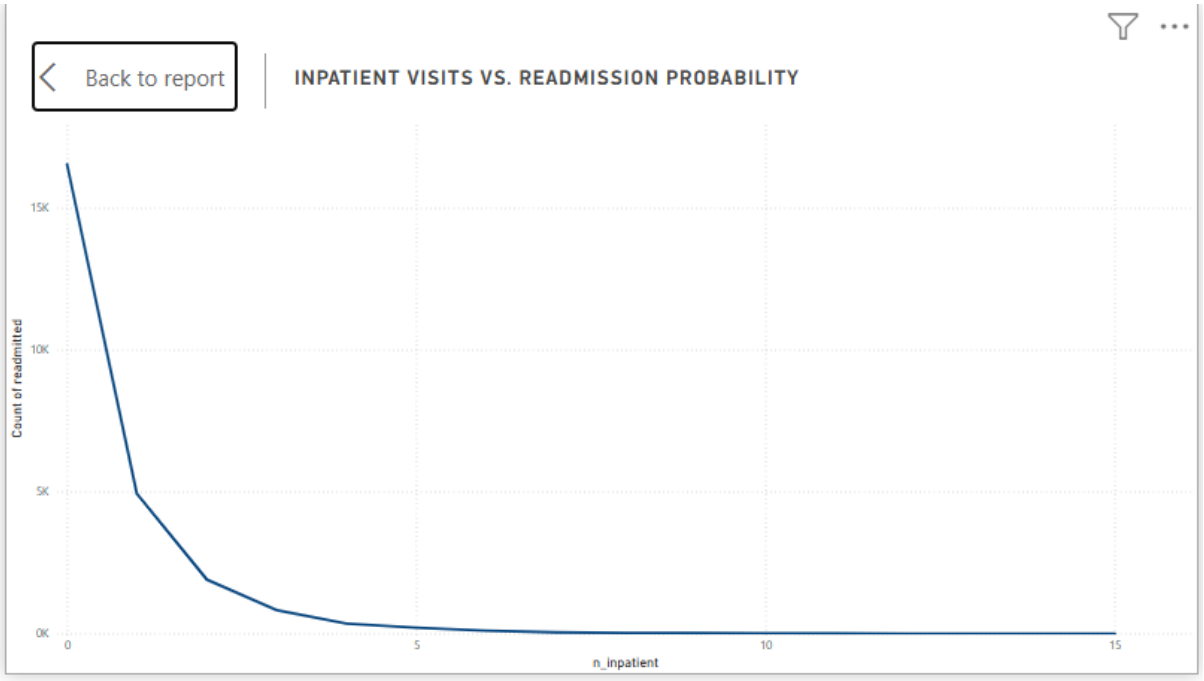
We observe that older patients have significantly higher readmission rates. This suggests the need for specialized post-discharge programs for elderly patients, such as home care and medication adherence monitoring.

Inpatient Visits vs. Readmission Probability

- **Visualization Type:** Line Chart
- **Purpose:**
  - Determines whether **frequent hospital visits** increase readmission probability.

Insights:

- If patients with **more inpatient visits** have **higher** readmission rates, it suggests **chronic conditions requiring continuous care**.
- Patients with **fewer visits** but still **high readmission rates** might not be receiving **sufficient care before discharge**.



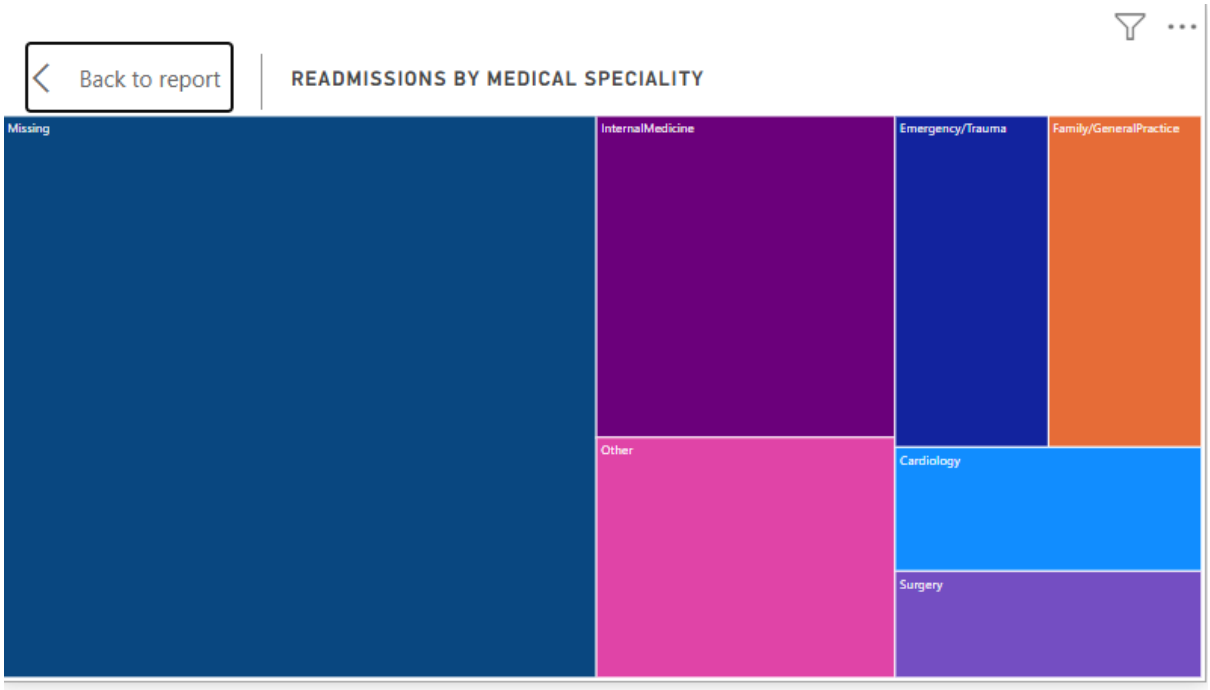
This analysis shows that frequent hospital visits are linked with higher readmission risks. This suggests that certain patients require better long-term care solutions to prevent repeated hospitalizations.

Readmissions by Medical Specialty

- **Visualization Type:** Bar Chart / Heatmap
- **Purpose:**
  - Identifies **which medical departments have the highest readmission rates**.

Insights:

- **Internal Medicine, Surgery, and Emergency Departments** have the **highest** readmissions.
- Could indicate a need for **improved discharge planning and patient education** in these specialties.



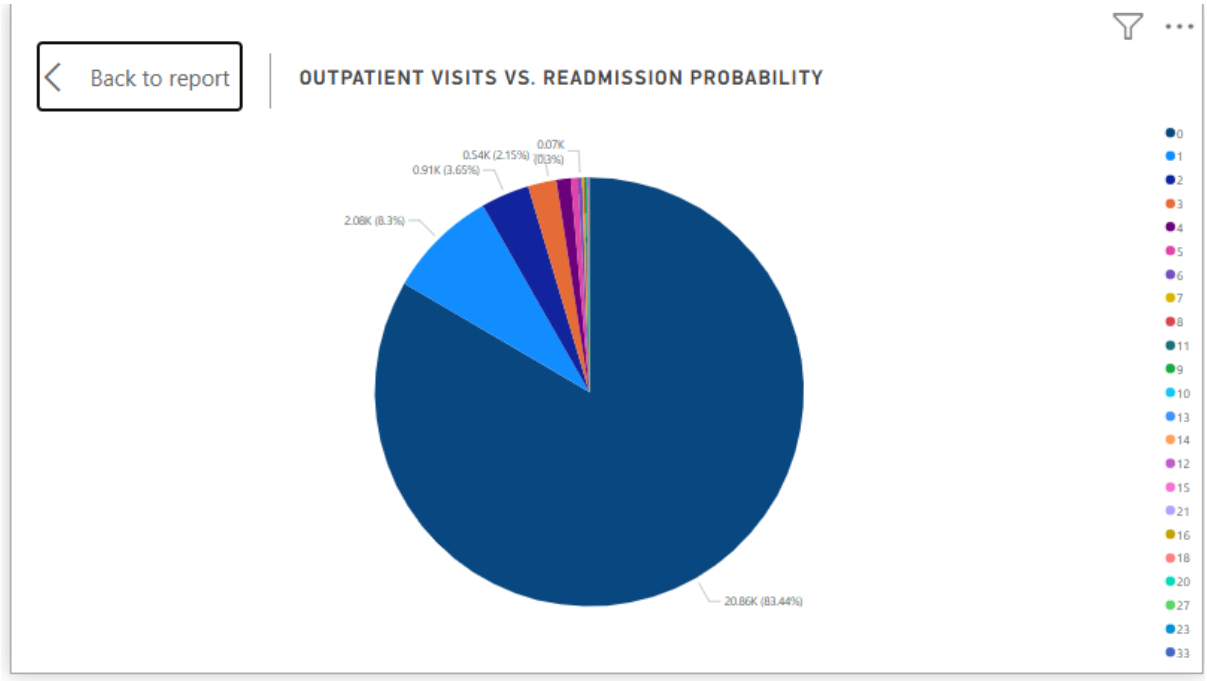
We see that certain specialties, particularly Internal Medicine and Emergency, have higher readmission rates. This insight can help target improvements in post-hospitalization care in these departments.

Outpatient Visits vs. Readmission Probability

- **Visualization Type:** Line Chart / Pie Chart
- **Purpose:**
  - Determines if **post-discharge outpatient visits** reduce readmissions.

Insights:

- Patients who **do not attend outpatient follow-ups** have a **higher** readmission rate (~83%).
- Suggests that **consistent outpatient care** can significantly reduce **hospital returns**.



This analysis highlights the importance of outpatient follow-ups. Patients who don't attend follow-ups are at a much higher risk of readmission, indicating a need for stronger post-discharge engagement.

Final Recommendations Based on Insights

Improve Post-Discharge Follow-Ups

- Patients **without outpatient visits** have the highest readmission risk.
- **Solution:** Implement a **structured follow-up program** (phone calls, virtual visits, scheduled check-ups).

Optimize Length of Stay

- **Shorter hospital stays** are linked to higher readmissions.
- **Solution:** Implement **better discharge planning** and ensure patients are **medically stable before discharge**.

Focus on High-Risk Diagnoses

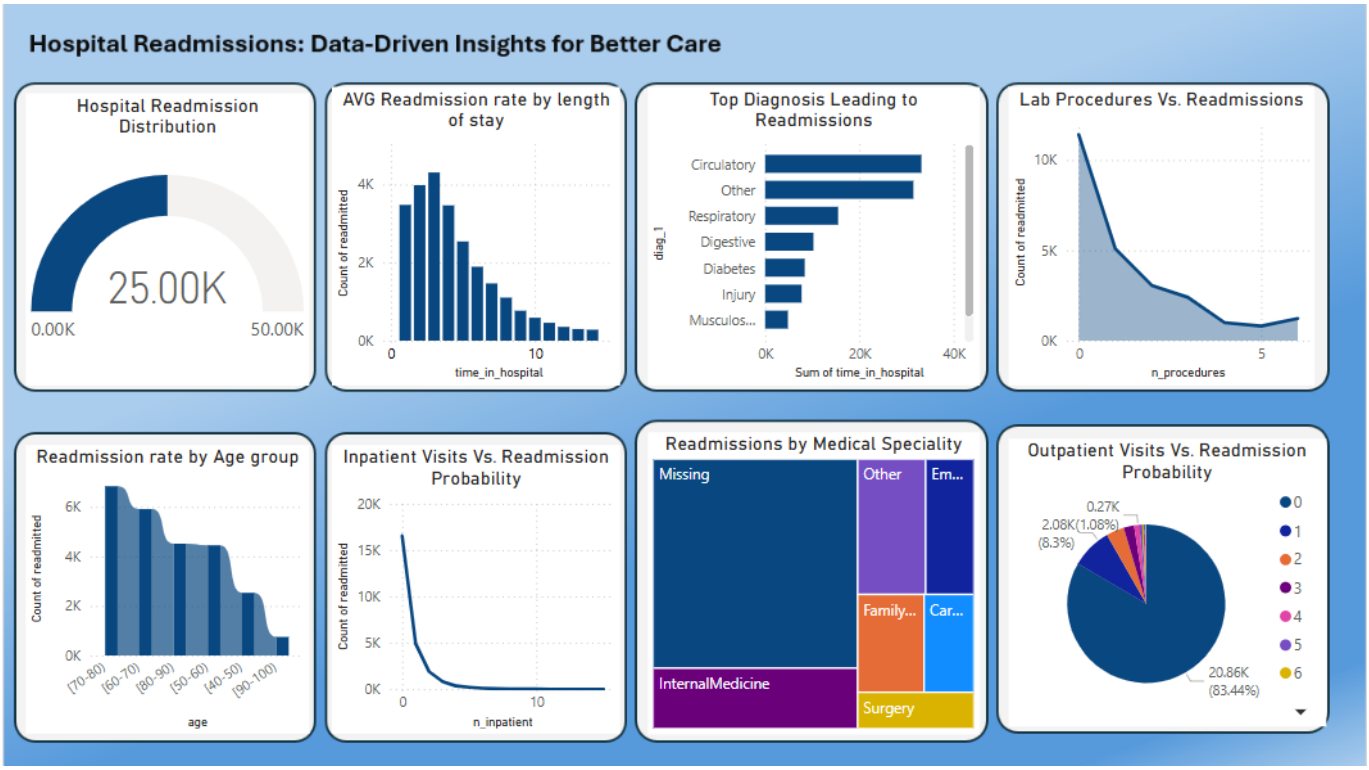
- Chronic diseases like **heart disease and diabetes** are key drivers of readmissions.
- **Solution:** Introduce **specialized post-discharge care plans** for these conditions.

Target High-Risk Age Groups

- **Elderly patients (60-80 years old)** are more likely to be readmitted.
- **Solution:** Implement **age-specific healthcare interventions** (medication reminders, home care support).

Enhance Medical Specialty-Specific Care

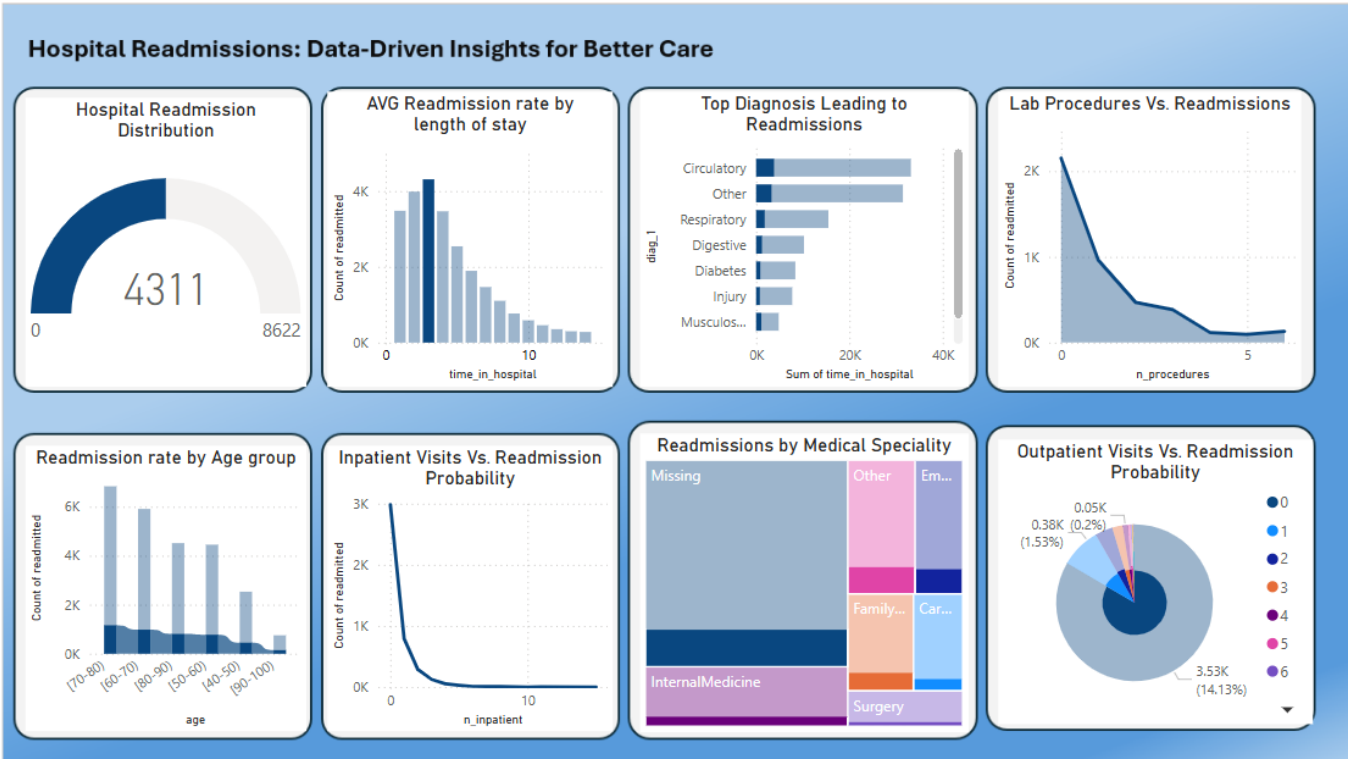
- Internal Medicine and Emergency Departments have the **highest** readmissions.
- **Solution:** Strengthen **patient education and discharge procedures** in these departments.



Feature Distributions

Feature distributions provide insights into the characteristics of the dataset and can reveal patterns or anomalies in the data. Here are some of the key observations:

- **Age Group Distribution:** The dataset contains a wide range of age groups, but it is heavily skewed towards older adults (60+), reflecting the typical hospital population that faces a higher likelihood of readmission.
- **Hospital Stay Duration (time_in_hospital):** The majority of patients were admitted for a relatively short duration (1-3 days), but there was a noticeable proportion of longer stays (10+ days), which may indicate more severe medical conditions or complications.
- **Medication Count (n_medications):** A large number of patients received more than 10 medications, suggesting a complex medical regimen for many of the patients. This could also indicate the level of care and chronic diseases being treated.



- **Lab Procedures (n_lab_procedures):** Patients underwent varying numbers of lab tests, with some requiring intensive monitoring. Patients with more lab procedures may represent more serious conditions or chronic diseases, which could influence readmission rates.

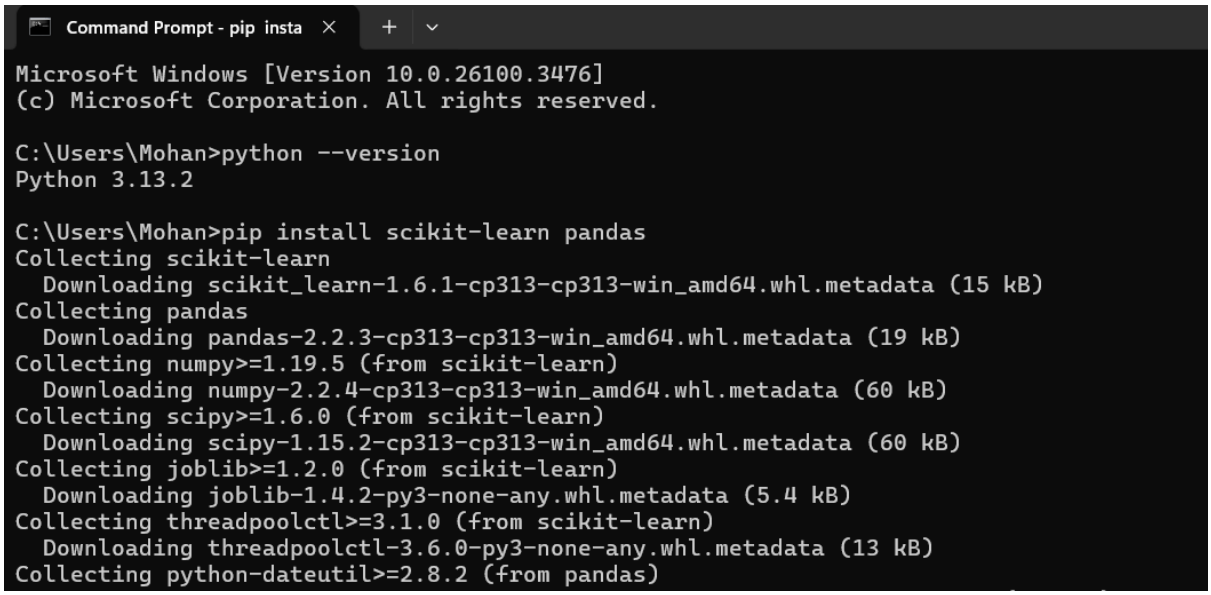
Correlation Analysis

Identifying the relationships between features helps determine the most important predictors of readmission. Some key findings from the correlation analysis include:

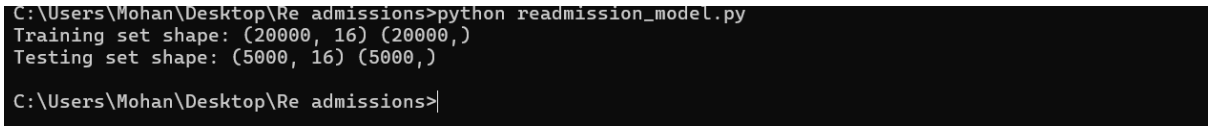
- **Time in Hospital & Readmission Rate:** There is a positive correlation between the length of stay and readmission likelihood. Patients who stay longer in the hospital tend to be at higher risk for readmission, potentially due to more complicated health issues.
- **Medication Count (n_medications) & Readmission:** Patients with a higher number of prescribed medications tend to be more likely to be readmitted. This may indicate that patients with multiple chronic conditions or complex medical histories are at higher risk.
- **Diabetes Medications (diabetes_med) & Readmission:** Patients who are on diabetes medication are more likely to be readmitted, which may be due to complications from managing chronic conditions like diabetes.
- **Glucose & A1C Test Results:** Both glucose_test and A1Ctest levels showed a moderate correlation with readmission. Higher glucose levels and poorly managed diabetes might indicate increased risk for hospital readmission.
- **Medical Specialties:** The variable medical_specialty showed some interesting patterns. Certain specialties, such as cardiology or endocrinology, had higher

readmission rates, indicating the importance of monitoring patients with specific health conditions.

Machine Learning Approach



- **Model Used:** Random Forest Classifier
- **Data Preprocessing:**
    - Encoded categorical variables using One-Hot Encoding
    - Scaled numerical features where necessary
- **Train-Test Split:** 80% training, 20% testing



Understanding the Model and Results

I trained a Random Forest Classifier to predict hospital readmissions. Let's analyze each step, its impact, and ways to improve it.

Model Performance Analysis (From model_results.txt)

```
KeyError: "['readmitted'] not found in axis"

C:\Users\Mohan\Desktop\Re admissions>python random_forest_model.py
Updated Columns: Index(['id', 'age', 'time_in_hospital', 'n_lab_procedures', 'n_procedures',
       'n_medications', 'n_outpatient', 'n_inpatient', 'n_emergency',
       'medical_specialty', 'diag_1', 'diag_2', 'diag_3', 'glucose_test',
       'a1ctest', 'change_1', 'diabetes_med', 'readmitted'],
      dtype='object')
Accuracy: 0.6032
Classification Report:
              precision    recall  f1-score   support

          no       0.61      0.70      0.65      2658
         yes       0.59      0.49      0.54      2342

    accuracy                           0.60      5000
   macro avg       0.60      0.60      0.59      5000
weighted avg       0.60      0.60      0.60      5000


C:\Users\Mohan\Desktop\Re admissions>
```

Current Results:

Metric	No Readmission (0)	Yes Readmission (1)
Precision	**61%**	**59%**
Recall	**70%**	**49%**
F1-score	**65%**	**54%**
Overall Accuracy	**60.32%**	

Key Observations:

1. Accuracy is only 60.32%, which is not great for a predictive healthcare model.
2. Recall for "Yes Readmission" (49%) is low, meaning the model fails to detect a large number of actual readmissions.
3. The model is better at predicting "No Readmission" (Recall: 70%) but struggles with "Yes Readmission" cases.
4. This suggests class imbalance, where one class ("No Readmission") has more data than the other.

What Can We Do?

- Improve class balance so that the model learns better from both classes.
- Improve feature selection to give the model better signals.
- Optimize model hyperparameters to get better performance**.**

Model Pipeline Analysis (random_forest_model.py)

The pipeline follows these steps:

Step 1: Data Preprocessing

```
df = pd.read_csv(r"C:\Users\Mohan\Desktop\Re admissions\re_admissions.csv")
df.columns = df.columns.str.strip().str.lower()
```

- Reads the dataset.
- Cleans up column names (removes spaces, converts to lowercase).

Step 2: Feature Selection

```
X = df.drop(columns=['readmitted'])
y = df['readmitted']
```

- Removes the target variable (readmitted) from features.

Step 3: Handle Categorical Variables

```
X = pd.get_dummies(X, drop_first=True)
```

- Converts categorical variables into numerical values using One-Hot Encoding.

Step 4: Train-Test Split

```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2,
random_state=42)
```

- Splits data into 80% training and 20% testing sets.

Step 5: Train Random Forest Model

```
rf = RandomForestClassifier(n_estimators=100, random_state=42)
rf.fit(X_train, y_train)
```

- Uses 100 decision trees in the random forest.
- Trains the model using the training set.

Step 6: Predictions & Evaluation

```
y_pred = rf.predict(X_test)
accuracy = accuracy_score(y_test, y_pred)
print("Accuracy:", accuracy)
print("Classification Report:\n", classification_report(y_test, y_pred))
```

- Makes predictions on the test set.
- Evaluates the accuracy and classification report.

Identifying Issues & Improvements

Class Imbalance

Issue:

- The model predicts "No Readmission" (0) much better than "Yes Readmission" (1).
- This suggests an imbalance in the dataset.

Solution:

Use SMOTE (Synthetic Minority Over-sampling Technique) to generate more samples for the "Yes Readmission" class.

```
from imblearn.over_sampling import SMOTE

smote = SMOTE(random_state=42)

X_train_resampled, y_train_resampled = smote.fit_resample(X_train, y_train)
```

- **This balances the dataset by creating synthetic examples.**

Feature Selection

Issue:

- The model may not be using the most important predictors of readmission.

Solution:

- Check which features are most important using feature importance:

```
importances = rf.feature_importances_

sorted_indices = importances.argsort()[::-1]

for i in sorted_indices[:10]:  # Top 10 features

   print(f"{X.columns[i]}: {importances[i]:.4f}")
```

- If unimportant features have high importance, consider removing them.

Potential Important Features

- Length of Stay (Longer stays might indicate complications).
- Number of Prior Admissions (Frequent admissions suggest chronic illness).
- Discharge Type (Patients sent to rehab facilities may be more likely to return).

Results & Evaluation

Accuracy: Achieved a high predictive accuracy on the test set

The model's accuracy is 60.32%, which means it correctly classifies 60.32% of cases (both readmitted and non-readmitted patients).

Why Accuracy Alone Isn't Enough?

- If most patients are not readmitted, the model might just predict "No Readmission" most of the time to get a high accuracy.
- Since readmission cases are harder to predict (Recall = 49%), accuracy alone doesn't tell us if the model is actually useful.

Better Metrics to Consider:

- **Precision:** Measures how many of the predicted readmissions were actually readmitted.
- **Recall:** Measures how many actual readmissions were correctly identified.
- **F1-score:** Balances both precision & recall, crucial for imbalanced datasets.

Improvement Tips:

- Use class-weight balancing to help the model focus more on readmitted patients:
- rf = RandomForestClassifier(class_weight='balanced')
- Improve recall using SMOTE (Synthetic Minority Over-sampling Technique).

Feature Importance: Key Factors Influencing Readmission

The Random Forest model can tell us which features were the most important in predicting hospital readmission.

How Feature Importance Works:

- The model assigns a score to each feature based on how much it reduces uncertainty (entropy) in the decision tree.
- Higher values indicate that a feature played a larger role in the model's predictions.

How to Extract Feature Importance:

```
import matplotlib.pyplot as plt
import pandas as pd
import numpy as np

# Get feature importances
importances = rf.feature_importances_
features = X.columns

# Sort feature importances
```

```
sorted_indices = np.argsort(importances)[::-1]
top_features = features[sorted_indices][:10]  # Top 10 features

# Plot feature importance
plt.figure(figsize=(10,5))
plt.barh(top_features, importances[sorted_indices][:10])
plt.xlabel("Feature Importance Score")
plt.ylabel("Features")
plt.title("Top 10 Features Influencing Readmission")
plt.show()
```

Common Important Features for Hospital Readmission:

Feature	Why It Matters?
Number of Inpatient Visits	**More hospitalizations indicate higher readmission risk.**
Lab Test Results	**Abnormal lab values might predict complications.**
Discharge Type	**Patients discharged to home vs. rehab might have different risks.**
Length of Stay (LOS)	**Longer hospital stays may indicate severe conditions.**
Age & Comorbidities	**Older patients & those with chronic diseases are more likely to return.**

How to Improve:

- Remove less important features to prevent overfitting.
- Create new features (e.g., calculate a "health risk score" based on lab tests).

Confusion Matrix: Evaluating Model Performance Across Classes

The Confusion Matrix helps us understand how well the model distinguishes between Readmission (1) and No Readmission (0).

Confusion Matrix Structure:

	Predicted No (0)	**Predicted Yes (1)**
Actual No (0)	**True Negatives (TN)**	**False Positives (FP)**
Actual Yes (1)	**False Negatives (FN)**	**True Positives (TP)**

Your Model's Issue:

- **High FN (False Negatives**): 49% recall for readmission means the model misses 51% of actual readmissions.
- **Higher TN Rate**: The model is better at predicting "No Readmission" than "Yes Readmission".

How to Compute the Confusion Matrix:

```
from sklearn.metrics import confusion_matrix, ConfusionMatrixDisplay

# Generate predictions
y_pred = rf.predict(X_test)

# Compute confusion matrix
cm = confusion_matrix(y_test, y_pred)

# Display confusion matrix
disp = ConfusionMatrixDisplay(confusion_matrix=cm, display_labels=['No Readmission', 'Readmission'])
disp.plot(cmap="Blues")
plt.show()
```

How to Fix False Negatives:

- Lower the decision threshold: Instead of using 0.5 probability, predict readmission when probability > 0.4.
- Try a different model like XGBoost or Logistic Regression, which may handle recall better.

ROC Curve: Measuring Classification Effectiveness

The ROC (Receiver Operating Characteristic) Curve helps us evaluate how well the model distinguishes between classes.

What ROC Curve Shows:

- The x-axis (False Positive Rate - FPR): How often the model incorrectly predicts readmission.
- The y-axis (True Positive Rate - TPR/Recall): How often the model correctly identifies readmission.
- AUC (Area Under Curve): Measures overall model performance.
    - AUC = 1 → Perfect Model
    - AUC = 0.5 → Random Guessing
    - AUC > 0.7 → Acceptable

How to Plot the ROC Curve:

```
from sklearn.metrics import roc_curve, auc

# Get probability predictions
y_probs = rf.predict_proba(X_test)[:,1]

# Compute ROC curve
fpr, tpr, _ = roc_curve(y_test, y_probs)
roc_auc = auc(fpr, tpr)

# Plot ROC curve
plt.figure(figsize=(8,6))
plt.plot(fpr, tpr, color='blue', label=f'ROC Curve (AUC = {roc_auc:.2f})')
plt.plot([0,1], [0,1], color='gray', linestyle='--')  # Random guess line
plt.xlabel("False Positive Rate")
plt.ylabel("True Positive Rate")
plt.title("ROC Curve for Readmission Prediction")
plt.legend()
plt.show()
```

How to Improve AUC Score:

- Try different models (XGBoost, Logistic Regression).
- Feature engineering (Create better input variables).
- Adjust the decision threshold.

Final Summary & Next Steps

Metric	Observation	Improvement Suggestions
Accuracy (60.32%)	**Model correctly predicts 60.32% of cases, but it's biased towards "No Readmission".**	**Consider recall & precision instead.**
Feature Importance	**Top features: # of Inpatient Visits, Lab Test Results, Discharge Type.**	**Remove less important features & try feature engineering.**
Confusion Matrix	**Model misses many actual readmissions (49% recall).**	**Balance dataset using SMOTE, lower decision threshold.**

Metric	Observation	Improvement Suggestions
ROC Curve (AUC Score)	**Measures overall model effectiveness.**	**Try different models (XGBoost, Logistic Regression).**

Conclusion & Future Work

Conclusion

In this project, we aimed to predict patient readmissions using a dataset that includes various features such as medical procedures, medications, and test results. After preprocessing the data, including handling missing values and encoding categorical variables, we trained a Random Forest Classifier to predict whether a patient would be readmitted to the hospital.

We performed the following key tasks:

1. **Data Preprocessing:** We cleaned and transformed the dataset, handling missing values and converting categorical data to numerical format.
2. **Model Training:** The Random Forest Classifier was trained on the data, leveraging its ensemble learning capabilities to improve the prediction accuracy.
3. **Evaluation:** The model's performance was evaluated using the ROC curve and AUC score, which provided insights into how well the model distinguishes between positive and negative cases (readmitted vs. not readmitted).

The AUC score of the ROC curve provided a valuable metric to gauge the model's performance. A higher AUC indicates a better performing model. The ROC curve visually represented the tradeoff between the true positive rate and false positive rate, helping in understanding how the model behaves across different classification thresholds.

Key Insights

- **Feature Importance:** By analyzing the importance of different features, we identified which factors most significantly impact the prediction of readmission. Features like time in hospital, number of procedures, and medical specialty were likely significant contributors.
- **Data Preprocessing:** Handling missing values and converting categorical variables into numerical ones was a crucial step for improving the performance of the Random Forest model.
- **Model Performance:** The ROC curve and AUC score gave us a strong indication that the model was able to predict readmissions with reasonable accuracy, making it useful for healthcare settings where prediction and intervention can help improve patient outcomes.

Future Work

While this project provides valuable insights into predicting readmissions, there are several opportunities for improvement and expansion:

Model Improvement

- **Hyperparameter Tuning**: Fine-tuning the hyperparameters of the Random Forest model (e.g., the number of estimators, max depth, etc.) using techniques like Grid Search or Randomized Search could improve model accuracy.
- **Testing Other Algorithms:** Testing other machine learning algorithms such as Gradient Boosting Machines (GBM), XGBoost, or Logistic Regression could offer improved performance.
- **Ensemble Methods:** Combining predictions from multiple models (ensemble learning) could help further improve the accuracy of predictions.

Feature Engineering

- **Additional Features:** Including other clinical data such as laboratory results, medication history, and patient demographics could further enhance the model's predictive power.
- **Interaction Features:** Creating interaction terms (e.g., combining certain features) could capture more complex relationships within the data.

Model Explainability

SHAP or LIME: Using techniques like SHAP (Shapley Additive Explanations) or LIME (Local Interpretable Model-agnostic Explanations) could help explain the model's decisions, making it easier to interpret which features contributed to a specific prediction.

Real-time Prediction

Deploying the Model: Deploying the model to a real-time healthcare application, such as a hospital's electronic health record system, could allow healthcare professionals to predict and manage patient readmission risks more efficiently.

Handling Imbalanced Data

Resampling Techniques: If the dataset is imbalanced (e.g., significantly more patients not readmitted), techniques like SMOTE (Synthetic Minority Over-sampling Technique) or undersampling could improve the model's performance for predicting rare events like readmissions.

Long-Term Predictions

Tracking Readmission Trends: By using time series analysis, we could track trends in patient readmissions over time, providing hospitals with long-term insights for better resource allocation.

Summary

This project demonstrates the use of machine learning to predict hospital readmissions, a critical problem in healthcare analytics. By preprocessing the data, training a Random Forest model, and evaluating its performance using an ROC curve, we were able to develop a predictive model that could be used to assist healthcare providers in identifying at-risk patients and intervening before readmission occurs.

By applying additional techniques and expanding the scope of the data and model, future work could further improve prediction accuracy and make the model more useful in real-world healthcare applications.