01_eda_proposal

December 5, 2025

1 Predicting Hospital Readmission Risk

1.1 Student A – Data Engineer / Preprocessing Lead (Binger)

1.2 1. Setup & Load

[2]:
```
# 1.1 Imports & Display Settings
# ------------------------------
import os
import pandas as pd
import numpy as np

# Visualization
import missingno as msno
import matplotlib.pyplot as plt
import seaborn as sns
import re

# Plot style
plt.style.use("default")
sns.set()

# Pandas display options for easier inspection
pd.set_option("display.max_columns", 50)
pd.set_option("display.width", 120)

print("Libraries imported.")
```

```
Libraries imported.
```

[3]:
```
!pip install missingno
```

```
/etc/zshenv:4: unmatched "
zsh:1: command not found: pip
```

[4]:
```
from pathlib import Path

# Path to ../results/figures relative to the notebook
FIG_DIR = Path("..") / "results" / "figures"
```

```
FIG_DIR.mkdir(parents=True, exist_ok=True)  # create if it doesn't exist
```

[5]:
```
print("Dataset: Diabetes 130-US Hospitals for Years 1999-2008 (UCI Repository)")
```

```
Dataset: Diabetes 130-US Hospitals for Years 1999-2008 (UCI Repository)
```

[6]:
```
# 1.2 Define Paths & Load the CSV
# -------------------------------
RAW_DATA_PATH = "../data/raw/diabetic_data.csv"

df = pd.read_csv(RAW_DATA_PATH)
print("Data loaded.")
```

```
Data loaded.
```

1.3 2. Review the Dataset

[7]:
```
# 2.1 Quick Peek
# --------------
# First 5 rows
df.head()
```

[7]:
```
   encounter_id  patient_nbr             race  gender      age weight
admission_type_id  discharge_disposition_id  \
0       2278392      8222157        Caucasian  Female   [0-10)      ?
6                         25
1        149190     55629189        Caucasian  Female  [10-20)      ?
1                          1
2         64410     86047875  AfricanAmerican  Female  [20-30)      ?
1                          1
3        500364     82442376        Caucasian    Male  [30-40)      ?
1                          1
4         16680     42519267        Caucasian    Male  [40-50)      ?
1                          1

   admission_source_id  time_in_hospital payer_code         medical_specialty
num_lab_procedures  num_procedures  \
0                    1                 1          ?  Pediatrics-Endocrinology
41                0
1                    7                 3          ?                         ?
59                0
2                    7                 2          ?                         ?
11                5
3                    7                 2          ?                         ?
44                1
4                    7                 1          ?                         ?
51                0
```

```
   num_medications  number_outpatient  number_emergency  number_inpatient
diag_1  diag_2 diag_3  number_diagnoses  \
0                1                  0                 0                 0
250.83       ?      ?                 1
1               18                  0                 0                 0
276  250.01    255                 9
2               13                  2                 0                 1
648     250    V27                 6
3               16                  0                 0                 0
8  250.43    403                 7
4                8                  0                 0                 0
197     157    250                 5

  max_glu_serum A1Cresult metformin repaglinide nateglinide chlorpropamide
glimepiride acetohexamide glipizide  \
0           NaN       NaN        No          No          No             No
No            No        No
1           NaN       NaN        No          No          No             No
No            No        No
2           NaN       NaN        No          No          No             No
No            No    Steady
3           NaN       NaN        No          No          No             No
No            No        No
4           NaN       NaN        No          No          No             No
No            No    Steady

  glyburide tolbutamide pioglitazone rosiglitazone acarbose miglitol
troglitazone tolazamide examide citoglipton  \
0        No          No           No            No       No       No
No         No      No          No
1        No          No           No            No       No       No
No         No      No          No
2        No          No           No            No       No       No
No         No      No          No
3        No          No           No            No       No       No
No         No      No          No
4        No          No           No            No       No       No
No         No      No          No

  insulin glyburide-metformin glipizide-metformin glimepiride-pioglitazone
metformin-rosiglitazone  \
0      No                  No                  No                        No
No
1      Up                  No                  No                        No
No
2      No                  No                  No                        No
No
```

```
3      Up                  No                  No                        No
No
4  Steady                  No                  No                        No
No

  metformin-pioglitazone change diabetesMed readmitted
0                     No     No          No         NO
1                     No     Ch         Yes        >30
2                     No     No         Yes         NO
3                     No     Ch         Yes         NO
4                     No     Ch         Yes         NO
```

[8]:
```
df.shape
df.dtypes
```

[8]:
```
encounter_id                int64
patient_nbr                 int64
race                       object
gender                     object
age                        object
weight                     object
admission_type_id           int64
discharge_disposition_id    int64
admission_source_id         int64
time_in_hospital            int64
payer_code                 object
medical_specialty          object
num_lab_procedures          int64
num_procedures              int64
num_medications             int64
number_outpatient           int64
number_emergency            int64
number_inpatient            int64
diag_1                     object
diag_2                     object
diag_3                     object
number_diagnoses            int64
max_glu_serum              object
A1Cresult                  object
metformin                  object
repaglinide                object
nateglinide                object
chlorpropamide             object
glimepiride                object
acetohexamide              object
glipizide                  object
glyburide                  object
```

```
tolbutamide                object
pioglitazone               object
rosiglitazone              object
acarbose                   object
miglitol                   object
troglitazone               object
tolazamide                 object
examide                    object
citoglipton                object
insulin                    object
glyburide-metformin        object
glipizide-metformin        object
glimepiride-pioglitazone   object
metformin-rosiglitazone    object
metformin-pioglitazone     object
change                     object
diabetesMed                object
readmitted                 object
dtype: object
```

[9]:
```
# 2.2 Shape and Column Listprint("Number of rows:", df.shape[0])
# ----------------------------------------------------------------
print("Number of columns:", df.shape[1])
print("\nColumns:")
print(df.columns.tolist())
```

```
Number of columns: 50

Columns:
['encounter_id', 'patient_nbr', 'race', 'gender', 'age', 'weight',
'admission_type_id', 'discharge_disposition_id', 'admission_source_id',
'time_in_hospital', 'payer_code', 'medical_specialty', 'num_lab_procedures',
'num_procedures', 'num_medications', 'number_outpatient', 'number_emergency',
'number_inpatient', 'diag_1', 'diag_2', 'diag_3', 'number_diagnoses',
'max_glu_serum', 'A1Cresult', 'metformin', 'repaglinide', 'nateglinide',
'chlorpropamide', 'glimepiride', 'acetohexamide', 'glipizide', 'glyburide',
'tolbutamide', 'pioglitazone', 'rosiglitazone', 'acarbose', 'miglitol',
'troglitazone', 'tolazamide', 'examide', 'citoglipton', 'insulin', 'glyburide-
metformin', 'glipizide-metformin', 'glimepiride-pioglitazone', 'metformin-
rosiglitazone', 'metformin-pioglitazone', 'change', 'diabetesMed', 'readmitted']
```

[10]:
```
# 2.3 Data Types + Non-null Counts
# --------------------------------
# Quick infor summary: non-null counts and dtypes
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 101766 entries, 0 to 101765
```

```
Data columns (total 50 columns):
 #   Column                    Non-Null Count   Dtype
---  ------                    --------------   -----
 0   encounter_id              101766 non-null  int64
 1   patient_nbr               101766 non-null  int64
 2   race                      101766 non-null  object
 3   gender                    101766 non-null  object
 4   age                       101766 non-null  object
 5   weight                    101766 non-null  object
 6   admission_type_id         101766 non-null  int64
 7   discharge_disposition_id  101766 non-null  int64
 8   admission_source_id       101766 non-null  int64
 9   time_in_hospital          101766 non-null  int64
 10  payer_code                101766 non-null  object
 11  medical_specialty         101766 non-null  object
 12  num_lab_procedures        101766 non-null  int64
 13  num_procedures            101766 non-null  int64
 14  num_medications           101766 non-null  int64
 15  number_outpatient         101766 non-null  int64
 16  number_emergency          101766 non-null  int64
 17  number_inpatient          101766 non-null  int64
 18  diag_1                    101766 non-null  object
 19  diag_2                    101766 non-null  object
 20  diag_3                    101766 non-null  object
 21  number_diagnoses          101766 non-null  int64
 22  max_glu_serum             5346 non-null    object
 23  A1Cresult                 17018 non-null   object
 24  metformin                 101766 non-null  object
 25  repaglinide               101766 non-null  object
 26  nateglinide               101766 non-null  object
 27  chlorpropamide            101766 non-null  object
 28  glimepiride               101766 non-null  object
 29  acetohexamide             101766 non-null  object
 30  glipizide                 101766 non-null  object
 31  glyburide                 101766 non-null  object
 32  tolbutamide               101766 non-null  object
 33  pioglitazone              101766 non-null  object
 34  rosiglitazone             101766 non-null  object
 35  acarbose                  101766 non-null  object
 36  miglitol                  101766 non-null  object
 37  troglitazone              101766 non-null  object
 38  tolazamide                101766 non-null  object
 39  examide                   101766 non-null  object
 40  citoglipton               101766 non-null  object
 41  insulin                   101766 non-null  object
 42  glyburide-metformin       101766 non-null  object
 43  glipizide-metformin       101766 non-null  object
 44  glimepiride-pioglitazone  101766 non-null  object
```

```
 45  metformin-rosiglitazone   101766 non-null  object
 46  metformin-pioglitazone    101766 non-null  object
 47  change                    101766 non-null  object
 48  diabetesMed               101766 non-null  object
 49  readmitted                101766 non-null  object
dtypes: int64(13), object(37)
memory usage: 38.8+ MB
```

[11]:
```
# 2.4 Basic Numeric Summary
# -------------------------
df.describe()
```

[11]:
```
       encounter_id   patient_nbr  admission_type_id  discharge_disposition_id
admission_source_id  time_in_hospital  \
count  1.017660e+05  1.017660e+05      101766.000000             101766.000000
101766.000000     101766.000000
mean   1.652016e+08  5.433040e+07           2.024006                  3.715642
5.754437          4.395987
std    1.026403e+08  3.869636e+07           1.445403                  5.280166
4.064081          2.985108
min    1.252200e+04  1.350000e+02           1.000000                  1.000000
1.000000          1.000000
25%    8.496119e+07  2.341322e+07           1.000000                  1.000000
1.000000          2.000000
50%    1.523890e+08  4.550514e+07           1.000000                  1.000000
7.000000          4.000000
75%    2.302709e+08  8.754595e+07           3.000000                  4.000000
7.000000          6.000000
max    4.438672e+08  1.895026e+08           8.000000                 28.000000
25.000000         14.000000

       num_lab_procedures  num_procedures  num_medications  number_outpatient
number_emergency  number_inpatient  \
count       101766.000000   101766.000000    101766.000000      101766.000000
101766.000000     101766.000000
mean            43.095641        1.339730        16.021844           0.369357
0.197836          0.635566
std             19.674362        1.705807         8.127566           1.267265
0.930472          1.262863
min              1.000000        0.000000         1.000000           0.000000
0.000000          0.000000
25%             31.000000        0.000000        10.000000           0.000000
0.000000          0.000000
50%             44.000000        1.000000        15.000000           0.000000
0.000000          0.000000
75%             57.000000        2.000000        20.000000           0.000000
0.000000          1.000000
```

```
max            132.000000        6.000000        81.000000          42.000000
76.000000         21.000000

       number_diagnoses
count     101766.000000
mean           7.422607
std            1.933600
min            1.000000
25%            6.000000
50%            8.000000
75%            9.000000
max           16.000000
```

[12]:
```
# Basic stats including categorical columns
df.describe(include="all").T.head(15) # first 15 rows of summary
```

[12]:
```
                           count unique        top   freq                mean
std      min         25%  \
encounter_id              101766.0    NaN        NaN    NaN  165201645.622978
102640295.983457  12522.0  84961194.0
patient_nbr               101766.0    NaN        NaN    NaN   54330400.694947
38696359.346534    135.0  23413221.0
race                        101766      6  Caucasian  76099                NaN
NaN      NaN         NaN
gender                      101766      3     Female  54708                NaN
NaN      NaN         NaN
age                         101766     10    [70-80)  26068                NaN
NaN      NaN         NaN
weight                      101766     10          ?  98569                NaN
NaN      NaN         NaN
admission_type_id         101766.0    NaN        NaN    NaN           2.024006
1.445403      1.0         1.0
discharge_disposition_id  101766.0    NaN        NaN    NaN           3.715642
5.280166      1.0         1.0
admission_source_id       101766.0    NaN        NaN    NaN           5.754437
4.064081      1.0         1.0
time_in_hospital          101766.0    NaN        NaN    NaN           4.395987
2.985108      1.0         2.0
payer_code                  101766     18          ?  40256                NaN
NaN      NaN         NaN
medical_specialty           101766     73          ?  49949                NaN
NaN      NaN         NaN
num_lab_procedures        101766.0    NaN        NaN    NaN          43.095641
19.674362      1.0        31.0
num_procedures            101766.0    NaN        NaN    NaN            1.33973
1.705807      0.0         0.0
num_medications           101766.0    NaN        NaN    NaN          16.021844
```

```
8.127566      1.0         10.0
```

	50%	75%	max
encounter_id	152388987.0	230270887.5	443867222.0
patient_nbr	45505143.0	87545949.75	189502619.0
race	NaN	NaN	NaN
gender	NaN	NaN	NaN
age	NaN	NaN	NaN
weight	NaN	NaN	NaN
admission_type_id	1.0	3.0	8.0
discharge_disposition_id	1.0	4.0	28.0
admission_source_id	7.0	7.0	25.0
time_in_hospital	4.0	6.0	14.0
payer_code	NaN	NaN	NaN
medical_specialty	NaN	NaN	NaN
num_lab_procedures	44.0	57.0	132.0
num_procedures	1.0	2.0	6.0
num_medications	15.0	20.0	81.0

[13]:
```
# 2.5 How Many Unique Values per Column?
# ---------------------------------------
# Number of unique values per column
nunique = df.nunique().sort_values(ascending=False)
nunique.head(20)
```

[13]:
```
encounter_id                101766
patient_nbr                  71518
diag_3                         790
diag_2                         749
diag_1                         717
num_lab_procedures             118
num_medications                 75
medical_specialty               73
number_outpatient               39
number_emergency                33
discharge_disposition_id        26
number_inpatient                21
payer_code                      18
admission_source_id             17
number_diagnoses                16
time_in_hospital                14
age                             10
weight                          10
admission_type_id                8
num_procedures                   7
dtype: int64
```

[14]:
```
# 2.6 Check the Target Variable (readmitted)
# ------------------------------------------
df["readmitted"].value_counts()
```

[14]:
```
readmitted
NO     54864
>30    35545
<30    11357
Name: count, dtype: int64
```

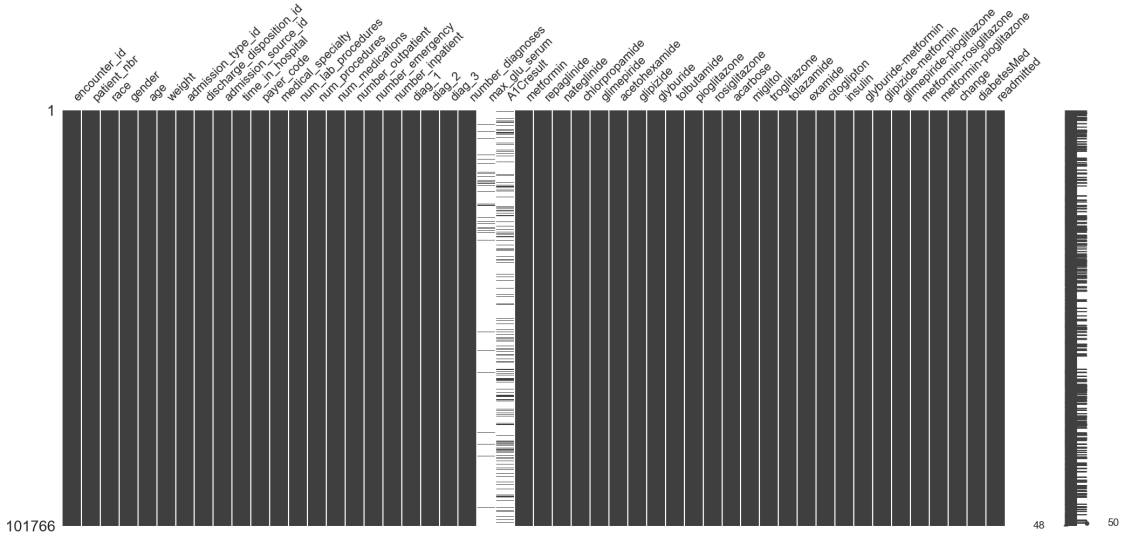
[15]:
```
# And with proportions
df["readmitted"].value_counts(normalize=True)
```

[15]:
```
readmitted
NO     0.539119
>30    0.349282
<30    0.111599
Name: proportion, dtype: float64
```

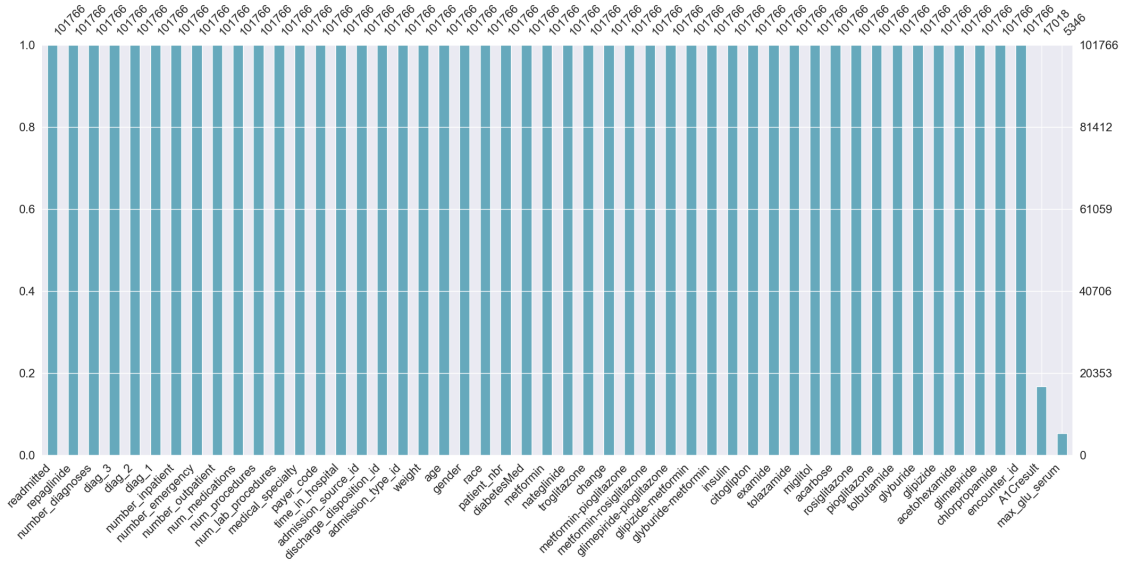
[16]:
```
# 2.7 Missing Values Overview
# ---------------------------
# How many "?" entries overall?
(df == "?").sum().sort_values(ascending=False).head(20)
```

[16]:
```
weight               98569
medical_specialty    49949
payer_code           40256
race                  2273
diag_3                1423
diag_2                 358
diag_1                  21
encounter_id             0
tolazamide               0
glyburide                0
tolbutamide              0
pioglitazone             0
rosiglitazone            0
acarbose                 0
miglitol                 0
troglitazone             0
citoglipton              0
examide                  0
acetohexamide            0
insulin                  0
dtype: int64
```

[17]:
```
msno.matrix(df)
plt.show()
```



[18]:
```
msno.bar(df,sort='descending',color='#66a9bc')
plt.show()
```



Observations:

- The dataset contains 101,766 encounters and 50 attributes.
- The target variable `readmitted` is imbalanced, with the majority of encounters not readmitted within 30 days.
- Several columns (`weight`, `max_glu_serum`, `A1Cresult`, `medical_specialty`, `payer_code`) have a high fraction of missing values and will be removed in preprocessing.

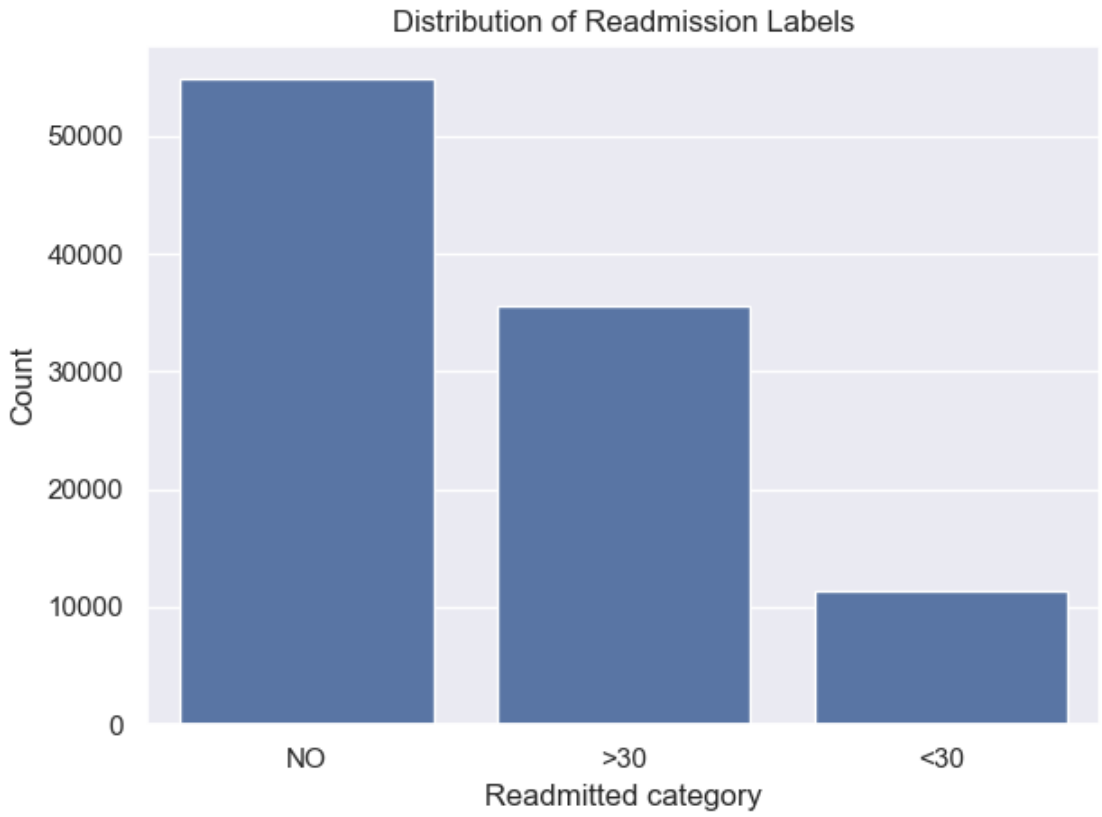
[19]:
```
# gender was coded differently so we use a custom count for this one
print('gender', df['gender'][df['gender'] == 'Unknown/Invalid'].count())
```

```
gender 3
```

[20]:
```
# Drop invalid gender rows
df = df[df['gender'] != 'Unknown/Invalid']
print("Remaining rows:", len(df))
```

```
Remaining rows: 101763
```

[21]:
```
# Bar plot of readmitted distribution
sns.countplot(x="readmitted", data=df)
plt.title("Distribution of Readmission Labels")
plt.xlabel("Readmitted category")
plt.ylabel("Count")
plt.tight_layout()
```



1.4 3. Data Cleaning

[22]:
```
missing_ratio = df.isna().sum() / len(df)
print("Top columns with missing values:")
print(missing_ratio.sort_values(ascending=False).head(10))
```

```
Top columns with missing values:
max_glu_serum    0.947466
A1Cresult        0.832768
encounter_id     0.000000
nateglinide      0.000000
glimepiride      0.000000
acetohexamide    0.000000
glipizide        0.000000
glyburide        0.000000
tolbutamide      0.000000
pioglitazone     0.000000
dtype: float64
```

[23]:
```
# Replace '?' with NaN for consistency
df = df.replace('?', np.nan)

# Drop columns with too many missing values or IDs
cols_to_drop = ['weight', 'payer_code', 'medical_specialty',
                'encounter_id', 'patient_nbr']
df.drop(columns=cols_to_drop, inplace=True)

# Check duplicates
print("Duplicates:", df.duplicated().sum())

# Re-check missing data
df.isnull().sum().sort_values(ascending=False).head(10)
```

```
Duplicates: 0
```

[23]:
```
max_glu_serum    96417
A1Cresult        84745
race              2271
diag_3            1423
diag_2             358
diag_1              21
tolbutamide          0
pioglitazone         0
rosiglitazone        0
acarbose             0
dtype: int64
```

Columns with excessive missing values (weight, payer_code, medical_specialty) or identifiers (encounter_id, patient_nbr) were removed to improve model stability.

[24]:
```
# Check duplicates
print("Duplicates:", df.duplicated().sum())

# Re-check missing data
df.isnull().sum().sort_values(ascending=False).head(10)
```

```
Duplicates: 0
```

[24]:
```
max_glu_serum    96417
A1Cresult        84745
race              2271
diag_3            1423
diag_2             358
diag_1              21
tolbutamide          0
pioglitazone         0
rosiglitazone        0
acarbose             0
dtype: int64
```

[25]:
```
# Drop heavy missing-value columns
cols_to_drop = ['max_glu_serum', 'A1Cresult']
df.drop(columns=cols_to_drop, inplace=True)
print(f"Dropped columns: {cols_to_drop}")
```

```
Dropped columns: ['max_glu_serum', 'A1Cresult']
```

[26]:
```
# Re-check missing data
df.isnull().sum().sort_values(ascending=False).head(10)
```

[26]:
```
race             2271
diag_3           1423
diag_2            358
diag_1             21
tolazamide          0
tolbutamide         0
pioglitazone        0
rosiglitazone       0
acarbose            0
miglitol            0
dtype: int64
```

[27]:
```
# Handle small missing subsets
# Fill missing 'race' with mode (most common value)
df['race'] = df['race'].fillna(df['race'].mode()[0])

# Fill diagnosis codes with a placeholder
for col in ['diag_1', 'diag_2', 'diag_3']:
```

```
    df[col] = df[col].fillna('Unknown')
```

[28]:
```
# Re-check missing data
df.isnull().sum().sort_values(ascending=False).head(10)
```

[28]:
```
race             0
examide          0
glyburide        0
tolbutamide      0
pioglitazone     0
rosiglitazone    0
acarbose         0
miglitol         0
troglitazone     0
tolazamide       0
dtype: int64
```

Features `max_glu_serum` and `A1Cresult` were dropped due to excessive missingness (>80%). Minor missing values in `race` were imputed with the most frequent category, while missing diagnosis codes were replaced with "Unknown" to retain encounters.

1.5 4. Feature Encoding & Target Transformation

In this section, we perform the final preprocessing and feature transformation steps to prepare the dataset for modeling. This includes encoding the target variable, transforming categorical features into numeric form, and creating new derived features.

[29]:
```
# 4.1 Encode Target Variable (readmitted_binary)
# ------------------------------------------------
# Convert readmission categories ("NO", ">30", "<30") into a binary target␣
 ↪variable.

df["readmitted"].value_counts()

readmit_map = {'<30': 1, '>30': 0, 'NO': 0}
df['readmitted_binary'] = df['readmitted'].map(readmit_map).astype('int8')

# Check distribution
print(df['readmitted_binary'].value_counts())
print(df['readmitted_binary'].value_counts(normalize=True))

# Note: The target is highly imbalanced (~89% no readmission vs. 11% readmitted␣
 ↪within 30 days)
```

```
readmitted_binary
0    90406
1    11357
Name: count, dtype: int64
readmitted_binary
```

```
0    0.888398
1    0.111602
Name: proportion, dtype: float64
```

The target variable `readmitted_binary` represents whether a patient was readmitted within 30 days (`1`) or not (`0`). The dataset exhibits a significant class imbalance, which will be addressed in the modeling phase.

[30]:
```
# 4.2 Convert Age Ranges to Midpoints (age_mid)
# ---------------------------------------------
# Convert categorical age intervals (e.g., "[40-50)") into numeric midpoints (e.
 ↪g., 45).

age_map = {
    '[0-10)': 5, '[10-20)': 15, '[20-30)': 25, '[30-40)': 35,
    '[40-50)': 45, '[50-60)': 55, '[60-70)': 65, '[70-80)': 75,
    '[80-90)': 85, '[90-100)': 95
}
df['age_mid'] = df['age'].map(age_map)
```

The new numeric feature `age_mid` preserves ordering information from the categorical `age` variable while simplifying numerical analysis and visualization.

[31]:
```
# 4.3 Create Service Utilization Feature
# --------------------------------------
# Combine outpatient, emergency, and inpatient visits into a single aggregated␣
 ↪feature.

df['service_utilization'] = (
    df['number_outpatient'] +
    df['number_emergency'] +
    df['number_inpatient']
)
```

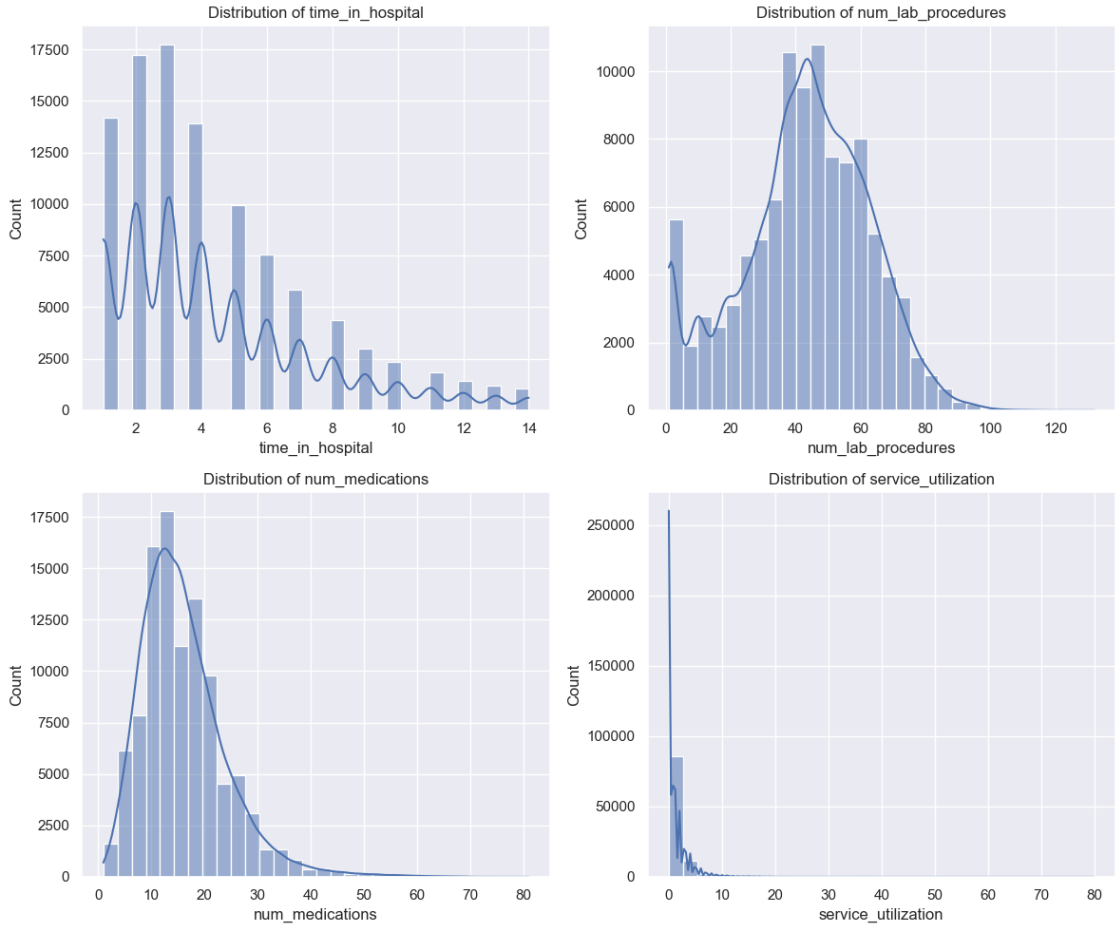
We created age_mid (age interval midpoints) and service_utilization (total prior visits) as simple feature transformations for modeling.

[32]:
```
# 4.4 Save Processed Dataset
# --------------------------
# Save the cleaned and feature-engineered dataset for use in modeling.

output_path = "../data/processed/diabetic_data_clean.csv"
df.to_csv(output_path, index=False)
print(f"Saved processed dataset to {output_path}")
```

```
Saved processed dataset to ../data/processed/diabetic_data_clean.csv
```

The processed dataset is saved for subsequent modeling and explainability analysis.

[33]:
```
# 4.5 Basic Visualizations for Dataset Description
import matplotlib.pyplot as plt
import seaborn as sns

numeric_vars = ['time_in_hospital',
                'num_lab_procedures',
                'num_medications',
                'service_utilization']

plt.figure(figsize=(12, 10))

for i, col in enumerate(numeric_vars, 1):
    plt.subplot(2, 2, i)
    sns.histplot(df[col], kde=True, bins=30)
    plt.title(f"Distribution of {col}")
    plt.xlabel(col)

plt.tight_layout()
plt.savefig(FIG_DIR / "fig2_distribution.png", dpi=300, bbox_inches="tight")
plt.show()
```



[34]:
```
# ----------------------------------------------------------------------
# SETUP: Import Libraries and Load Data
# ----------------------------------------------------------------------
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns

# Set a professional plot style
sns.set_style("whitegrid")
plt.rcParams['figure.dpi'] = 150 # High resolution for export

# NOTE: Adjust this path if your diabetes_cleaned.csv is in a different location
try:
    df = pd.read_csv("diabetes_cleaned.csv")
    print("Cleaned data loaded successfully.")
except FileNotFoundError:
    print("Error: 'diabetes_cleaned.csv' not found. Please check the file path.
 ↪")
    exit()

# ----------------------------------------------------------------------
# 1. DISTRIBUTION OF KEY NUMERIC VARIABLES
# Supporting Section 3.4: Shows skewness and feature concentration
# ----------------------------------------------------------------------
print("\nGenerating Figure 1: Distribution of Key Numeric Variables...")

key_numerics = ['time_in_hospital', 'num_lab_procedures', 'num_medications',␣
 ↪'service_utilization']
titles = [
    'Time in Hospital (Days)',
    'Number of Lab Procedures',
    'Number of Medications',
    'Composite Service Utilization'
]

fig, axes = plt.subplots(2, 2, figsize=(12, 8))
axes = axes.flatten()

for i, col in enumerate(key_numerics):
    sns.histplot(df[col], kde=True, ax=axes[i], bins=30, color='#1f77b4')
    axes[i].set_title(f'Distribution of {titles[i]}', fontsize=12)
    axes[i].set_xlabel(titles[i])
    axes[i].set_ylabel('Count (Log Scale)')
```

```
    axes[i].set_yscale('log') # Use log scale for clearer visualization of␣
 ↪skewed data

plt.tight_layout()
plt.suptitle("Figure 3.1: Distribution of Key Numeric Variables in Cleaned␣
 ↪Dataset", y=1.02, fontsize=16)
plt.savefig('figure_3_1_distributions.png')
plt.show()


# ----------------------------------------------------------------------
# 2. READMISSION RATE BY HOSPITAL STAY DURATION
# Supporting Section 3.4: Confirms longer stays = higher risk (critical finding)
# ----------------------------------------------------------------------
print("\nGenerating Figure 2: Readmission Rate by Hospital Stay Duration...")

# Calculate the mean readmission rate (1=readmitted) for each time_in_hospital␣
 ↪group
readmission_rate_by_stay = df.groupby('time_in_hospital')['readmitted_binary'].
 ↪mean().reset_index()

plt.figure(figsize=(8, 5))
sns.lineplot(
    data=readmission_rate_by_stay,
    x='time_in_hospital',
    y='readmitted_binary',
    marker='o',
    color='#d62728',
    linewidth=2
)

# Add a horizontal line to indicate the overall mean rate (11.16%)
overall_rate = df['readmitted_binary'].mean()
plt.axhline(overall_rate, color='grey', linestyle='--', label=f'Overall Rate␣
 ↪({overall_rate*100:.2f}%)')

plt.title("Figure 3.2: Readmission Rate (%) by Hospital Stay Duration (Days)",␣
 ↪fontsize=14)
plt.xlabel("Time in Hospital (Days)", fontsize=10)
plt.ylabel("30-Day Readmission Rate", fontsize=10)
plt.yticks([0.05, 0.10, 0.15, 0.20]) # Set fixed y-ticks for clarity
plt.ylim(0.05, 0.20)
plt.legend()
plt.savefig('figure_3_2_readmission_by_stay.png')
plt.show()
```

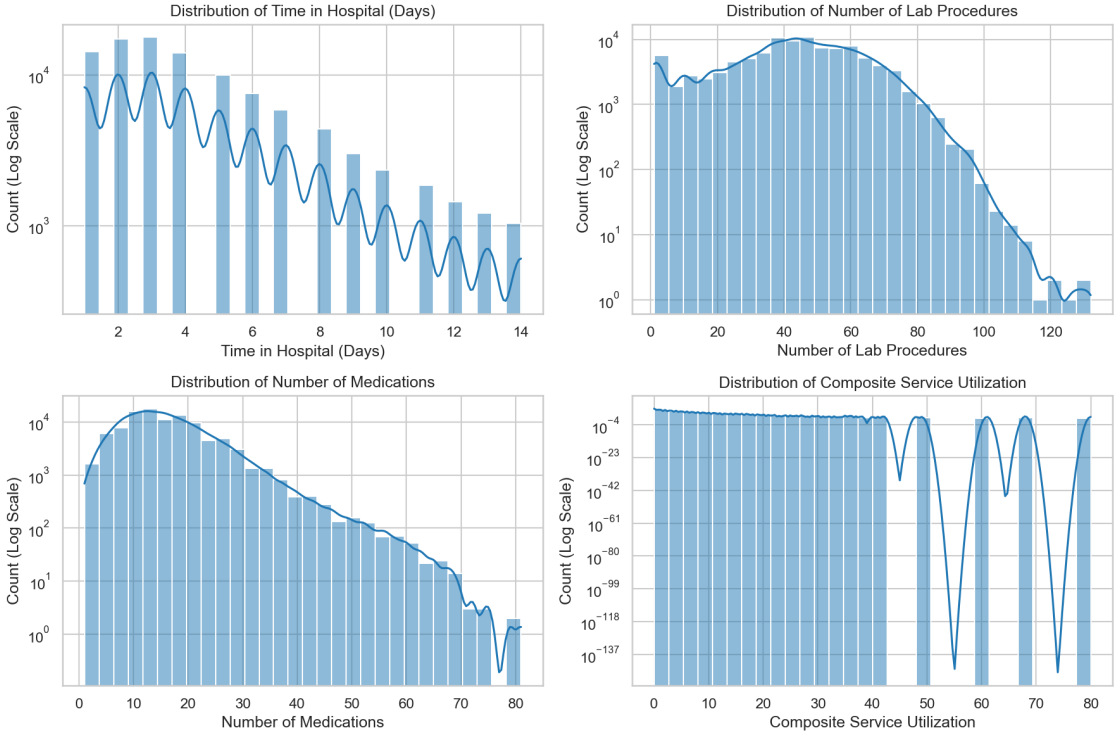
```
# ----------------------------------------------------------------------
# 3. AVERAGE MEDICATIONS & TIME IN HOSPITAL BY AGE GROUP
# Supporting Section 3.4: Shows complexity/severity increases with age
# ----------------------------------------------------------------------
print("\nGenerating Figure 3: Average Metrics by Age Group...")

# The 'age' column is already categorical (e.g., [50-60)), suitable for grouping
age_trends = df.groupby('age').agg(
    avg_medications=('num_medications', 'mean'),
    avg_time_in_hospital=('time_in_hospital', 'mean')
).reset_index()

# Explicitly define the correct chronological order for the age categories
age_order = ['[0-10)', '[10-20)', '[20-30)', '[30-40)', '[40-50)',
             '[50-60)', '[60-70)', '[70-80)', '[80-90)', '[90-100)']

# Ensure age groups are plotted in the correct order by setting the category␣
 ↪order
age_trends['age'] = pd.Categorical(age_trends['age'], categories=age_order,␣
 ↪ordered=True)
age_trends = age_trends.sort_values('age')


fig, ax1 = plt.subplots(figsize=(10, 6))

color1 = '#2ca02c' # Green for medications
sns.lineplot(data=age_trends, x='age', y='avg_medications', ax=ax1, marker='s',␣
 ↪color=color1)
ax1.set_ylabel('Average Medications', color=color1)
ax1.tick_params(axis='y', labelcolor=color1)

# Create a second axis for Time in Hospital
ax2 = ax1.twinx()
color2 = '#ff7f0e' # Orange for time in hospital
sns.lineplot(data=age_trends, x='age', y='avg_time_in_hospital', ax=ax2,␣
 ↪marker='o', color=color2)
ax2.set_ylabel('Average Time in Hospital (Days)', color=color2)
ax2.tick_params(axis='y', labelcolor=color2)

ax1.set_title("Figure 3.3: Average Clinical Metrics by Patient Age Group",␣
 ↪fontsize=14)
ax1.set_xlabel("Age Group", fontsize=10)
plt.tight_layout()
plt.savefig('figure_3_3_age_trends.png')
plt.show()
```

```
# ----------------------------------------------------------------------
# 4. CORRELATION HEATMAP (REFINED)
# Supporting Section 3.4 & 4.3: Justifies baseline model choice
# ----------------------------------------------------------------------
print("\nGenerating Figure 4: Correlation Heatmap...")

# Select only numeric columns for correlation matrix
numeric_cols = df.select_dtypes(include=[np.number]).columns.tolist()
corr_df = df[numeric_cols].corr()

# 1. Focus only on correlations with the target variable, 'readmitted_binary'
target_corr = corr_df['readmitted_binary'].sort_values(ascending=False).
 ↪drop('readmitted_binary')

# 2. Select the top 10 absolute correlations for better visualization in the␣
 ↪report
# We include 'service_utilization' even if it's not in the top 10 absolute (it␣
 ↪is in the top 5 here)
top_features = target_corr.abs().sort_values(ascending=False).index[:10].
 ↪tolist()
target_corr_top = target_corr.loc[top_features]


plt.figure(figsize=(5, 8))
sns.heatmap(
    target_corr_top.to_frame(),
    annot=True,
    fmt=".3f",
    cmap='vlag',
    cbar=False,
    linewidths=0.5,
    linecolor='black'
)
plt.title("Figure 3.4: Top 10 Feature Correlations with Readmission Target",␣
 ↪fontsize=14)
plt.yticks(rotation=0)
plt.tight_layout()
plt.savefig('figure_3_4_correlation_heatmap.png')
plt.show()

print("\nAll four required EDA figures generated and saved as PNG files.")
print("The files are: figure_3_1_distributions.png,␣
 ↪figure_3_2_readmission_by_stay.png, figure_3_3_age_trends.png,␣
 ↪figure_3_4_correlation_heatmap.png.")
```

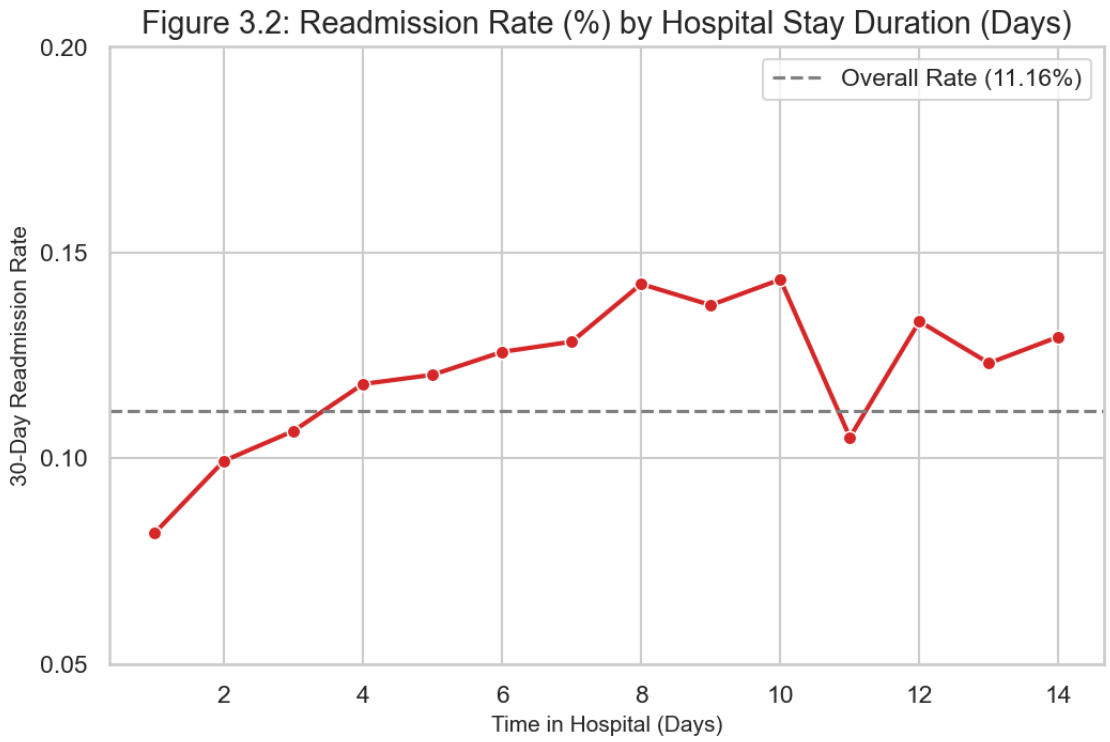
```
Error: 'diabetes_cleaned.csv' not found. Please check the file path.
```

Generating Figure 1: Distribution of Key Numeric Variables…

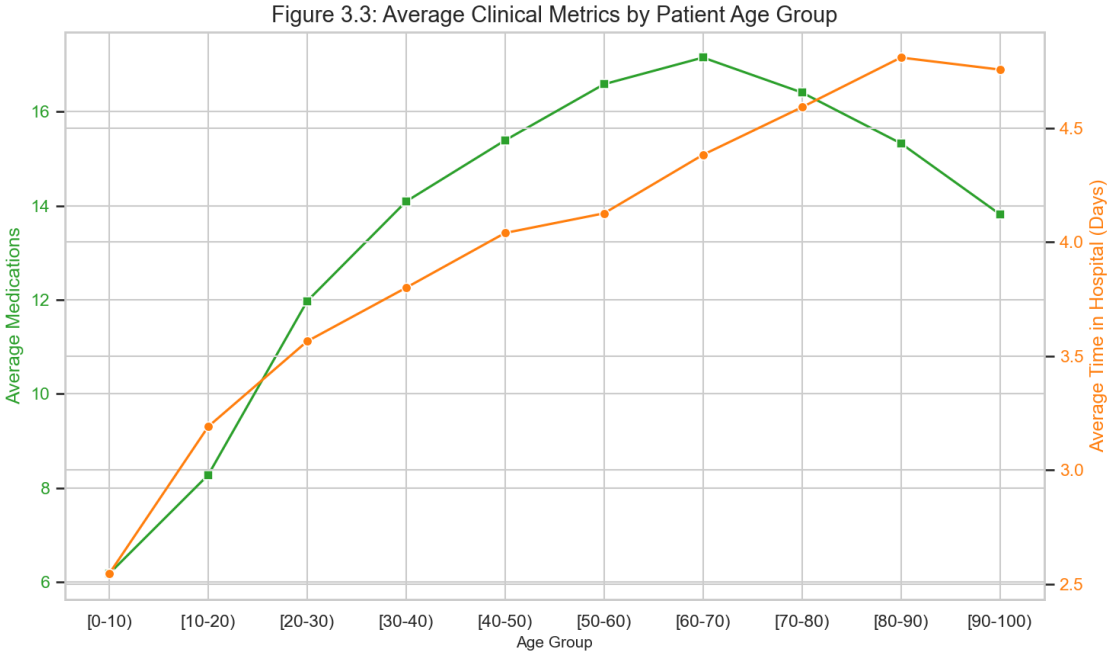
Figure 3.1: Distribution of Key Numeric Variables in Cleaned Dataset



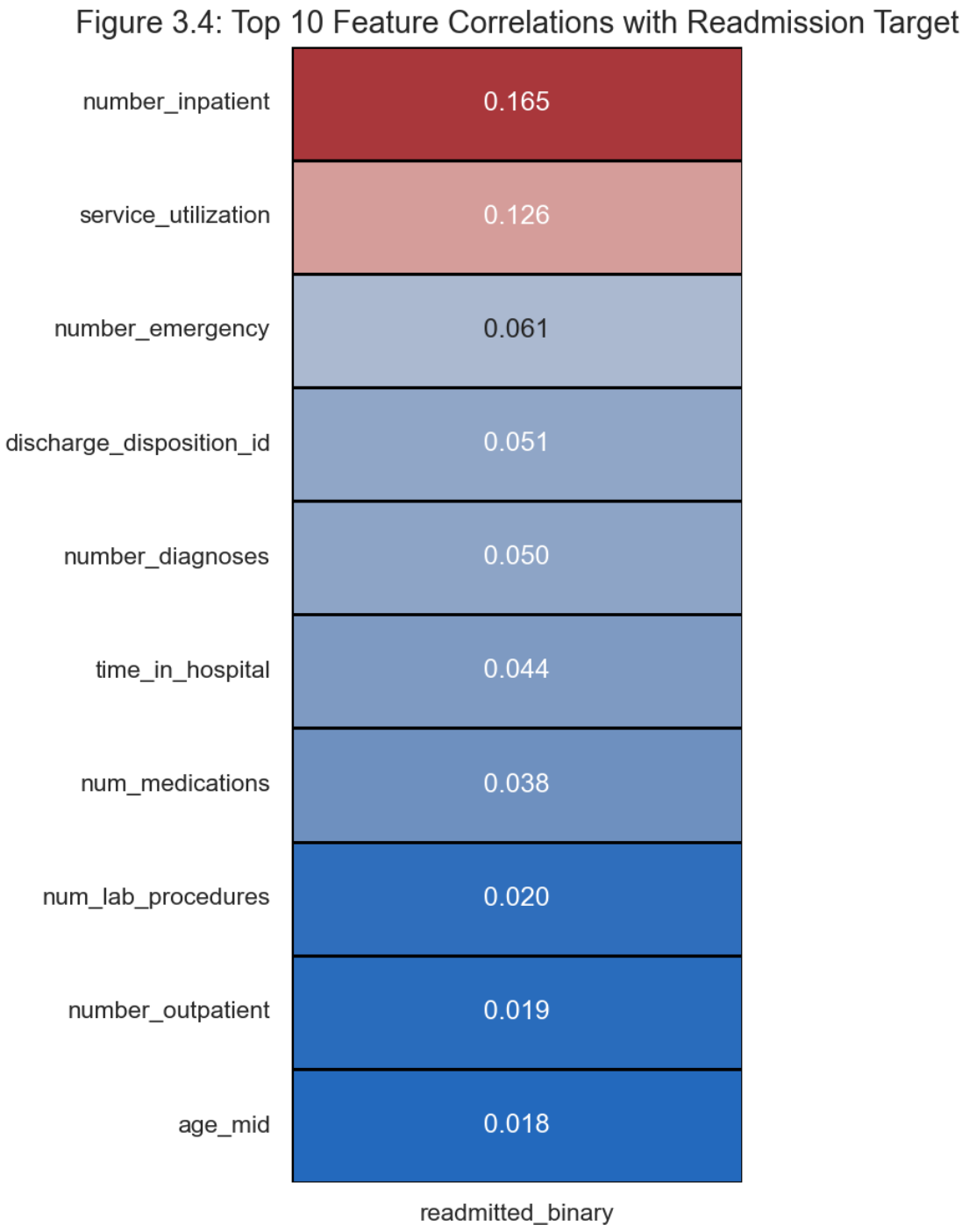
Generating Figure 2: Readmission Rate by Hospital Stay Duration…




```
Generating Figure 3: Average Metrics by Age Group…
```



```
Generating Figure 4: Correlation Heatmap…
```




```
All four required EDA figures generated and saved as PNG files.
```

The files are: figure_3_1_distributions.png, figure_3_2_readmission_by_stay.png, figure_3_3_age_trends.png, figure_3_4_correlation_heatmap.png.

1.6 5. Exploratory Data Analysis (EDA)

In this section, we visualize distributions, relationships, and trends in the cleaned dataset to gain insights that may inform feature importance and model design.

[35]:
```
# 5.1 Dataset Overview
# ---------------------
print(df.shape)
df.info()
df.describe(include='all')
```

```
(101763, 46)
<class 'pandas.core.frame.DataFrame'>
Index: 101763 entries, 0 to 101765
Data columns (total 46 columns):
 #   Column                    Non-Null Count   Dtype
---  ------                    --------------   -----
 0   race                      101763 non-null  object
 1   gender                    101763 non-null  object
 2   age                       101763 non-null  object
 3   admission_type_id         101763 non-null  int64
 4   discharge_disposition_id  101763 non-null  int64
 5   admission_source_id       101763 non-null  int64
 6   time_in_hospital          101763 non-null  int64
 7   num_lab_procedures        101763 non-null  int64
 8   num_procedures            101763 non-null  int64
 9   num_medications           101763 non-null  int64
 10  number_outpatient         101763 non-null  int64
 11  number_emergency          101763 non-null  int64
 12  number_inpatient          101763 non-null  int64
 13  diag_1                    101763 non-null  object
 14  diag_2                    101763 non-null  object
 15  diag_3                    101763 non-null  object
 16  number_diagnoses          101763 non-null  int64
 17  metformin                 101763 non-null  object
 18  repaglinide               101763 non-null  object
 19  nateglinide               101763 non-null  object
 20  chlorpropamide            101763 non-null  object
 21  glimepiride               101763 non-null  object
 22  acetohexamide             101763 non-null  object
 23  glipizide                 101763 non-null  object
 24  glyburide                 101763 non-null  object
 25  tolbutamide               101763 non-null  object
 26  pioglitazone              101763 non-null  object
 27  rosiglitazone             101763 non-null  object
```

```
 28  acarbose                  101763 non-null  object
 29  miglitol                  101763 non-null  object
 30  troglitazone              101763 non-null  object
 31  tolazamide                101763 non-null  object
 32  examide                   101763 non-null  object
 33  citoglipton               101763 non-null  object
 34  insulin                   101763 non-null  object
 35  glyburide-metformin       101763 non-null  object
 36  glipizide-metformin       101763 non-null  object
 37  glimepiride-pioglitazone  101763 non-null  object
 38  metformin-rosiglitazone   101763 non-null  object
 39  metformin-pioglitazone    101763 non-null  object
 40  change                    101763 non-null  object
 41  diabetesMed               101763 non-null  object
 42  readmitted                101763 non-null  object
 43  readmitted_binary         101763 non-null  int8
 44  age_mid                   101763 non-null  int64
 45  service_utilization       101763 non-null  int64
dtypes: int64(13), int8(1), object(32)
memory usage: 35.8+ MB
```

[35]:

	race	gender	age	admission_type_id	discharge_disposition_id	admission_source_id	\
count	101763	101763	101763	101763.000000	101763.000000	101763.000000	
unique	5	2	10	NaN	NaN	NaN	
top	Caucasian	Female	[70-80)	NaN	NaN	NaN	
freq	78370	54708	26066	NaN	NaN	NaN	
mean	NaN	NaN	NaN	2.024017	3.715515	5.754459	
std	NaN	NaN	NaN	1.445414	5.279919	4.064110	
min	NaN	NaN	NaN	1.000000	1.000000	1.000000	
25%	NaN	NaN	NaN	1.000000	1.000000	1.000000	
50%	NaN	NaN	NaN	1.000000	1.000000	7.000000	
75%	NaN	NaN	NaN	3.000000	4.000000	7.000000	
max	NaN	NaN	NaN	8.000000	28.000000	25.000000	

	time_in_hospital	num_lab_procedures	num_procedures	num_medications

```
number_outpatient  number_emergency  \
count      101763.000000       101763.000000   101763.000000    101763.000000
101763.000000     101763.000000
unique               NaN                 NaN             NaN              NaN
NaN               NaN
top                  NaN                 NaN             NaN              NaN
NaN               NaN
freq                 NaN                 NaN             NaN              NaN
NaN               NaN
mean            4.396018           43.095909        1.339691        16.021835
0.369368          0.197842
std             2.985092           19.674220        1.705792         8.127589
1.267282          0.930485
min             1.000000            1.000000        0.000000         1.000000
0.000000          0.000000
25%             2.000000           31.000000        0.000000        10.000000
0.000000          0.000000
50%             4.000000           44.000000        1.000000        15.000000
0.000000          0.000000
75%             6.000000           57.000000        2.000000        20.000000
0.000000          0.000000
max            14.000000          132.000000        6.000000        81.000000
42.000000         76.000000

        number_inpatient  diag_1  diag_2  diag_3  number_diagnoses metformin
repaglinide nateglinide chlorpropamide  \
count      101763.000000  101763  101763  101763     101763.000000    101763
101763      101763         101763
unique               NaN     717     749     790               NaN         4
4           4              4
top                  NaN     428     276     250               NaN        No
No          No             No
freq                 NaN    6862    6752   11555               NaN     81776
100224      101060         101677
mean            0.635585     NaN     NaN     NaN          7.422649       NaN
NaN         NaN            NaN
std             1.262877     NaN     NaN     NaN          1.933578       NaN
NaN         NaN            NaN
min             0.000000     NaN     NaN     NaN          1.000000       NaN
NaN         NaN            NaN
25%             0.000000     NaN     NaN     NaN          6.000000       NaN
NaN         NaN            NaN
50%             0.000000     NaN     NaN     NaN          8.000000       NaN
NaN         NaN            NaN
75%             1.000000     NaN     NaN     NaN          9.000000       NaN
NaN         NaN            NaN
max            21.000000     NaN     NaN     NaN         16.000000       NaN
```

```
NaN         NaN            NaN

       glimepiride acetohexamide glipizide glyburide tolbutamide pioglitazone
rosiglitazone acarbose miglitol  \
count       101763        101763    101763    101763      101763       101763
101763   101763   101763
unique           4             2         4         4           2            4
4        4        4
top             No            No        No        No          No           No
No       No       No
freq         96572        101762     89078     91113      101740        94436
95399   101455   101725
mean           NaN           NaN       NaN       NaN         NaN          NaN
NaN      NaN      NaN
std            NaN           NaN       NaN       NaN         NaN          NaN
NaN      NaN      NaN
min            NaN           NaN       NaN       NaN         NaN          NaN
NaN      NaN      NaN
25%            NaN           NaN       NaN       NaN         NaN          NaN
NaN      NaN      NaN
50%            NaN           NaN       NaN       NaN         NaN          NaN
NaN      NaN      NaN
75%            NaN           NaN       NaN       NaN         NaN          NaN
NaN      NaN      NaN
max            NaN           NaN       NaN       NaN         NaN          NaN
NaN      NaN      NaN

       troglitazone tolazamide examide citoglipton insulin glyburide-metformin
glipizide-metformin  \
count        101763     101763  101763      101763  101763              101763
101763
unique            2          3       1           1       4                   4
2
top              No         No      No          No      No                  No
No
freq         101760     101724  101763      101763   47380              101057
101750
mean            NaN        NaN     NaN         NaN     NaN                 NaN
NaN
std             NaN        NaN     NaN         NaN     NaN                 NaN
NaN
min             NaN        NaN     NaN         NaN     NaN                 NaN
NaN
25%             NaN        NaN     NaN         NaN     NaN                 NaN
NaN
50%             NaN        NaN     NaN         NaN     NaN                 NaN
NaN
```

```
75%             NaN        NaN     NaN          NaN     NaN                  NaN
NaN
max             NaN        NaN     NaN          NaN     NaN                  NaN
NaN

       glimepiride-pioglitazone metformin-rosiglitazone metformin-pioglitazone
change diabetesMed readmitted  \
count                    101763                  101763                  101763
101763      101763     101763
unique                        2                       2                       2
2           2          3
top                          No                      No                      No
No         Yes         NO
freq                     101762                  101761                  101762
54754       78361      54861
mean                        NaN                     NaN                     NaN
NaN         NaN        NaN
std                         NaN                     NaN                     NaN
NaN         NaN        NaN
min                         NaN                     NaN                     NaN
NaN         NaN        NaN
25%                         NaN                     NaN                     NaN
NaN         NaN        NaN
50%                         NaN                     NaN                     NaN
NaN         NaN        NaN
75%                         NaN                     NaN                     NaN
NaN         NaN        NaN
max                         NaN                     NaN                     NaN
NaN         NaN        NaN

        readmitted_binary        age_mid  service_utilization
count       101763.000000  101763.000000        101763.000000
unique                NaN            NaN                  NaN
top                   NaN            NaN                  NaN
freq                  NaN            NaN                  NaN
mean             0.111602      65.966854             1.202795
std              0.314878      15.941022             2.291805
min              0.000000       5.000000             0.000000
25%              0.000000      55.000000             0.000000
50%              0.000000      65.000000             0.000000
75%              0.000000      75.000000             2.000000
max              1.000000      95.000000            80.000000
```

[ ]:
```
# 5.2 Class Imbalance Visualization
# --------------------------------
plt.figure(figsize=(5, 4))
sns.countplot(x='readmitted_binary', data=df)
```

```
plt.title("Distribution of 30-day Readmission Labels")
plt.xlabel("Readmitted within 30 days (1 = Yes, 0 = No)")
plt.ylabel("Count")
plt.tight_layout()
plt.savefig(FIG_DIR / "fig_readmission_distribution.png", dpi=300,␣
 ↪bbox_inches="tight")
plt.show()
```

The dataset is imbalanced, with only ~11% of patients readmitted within 30 days. This imbalance highlights the need for resampling or class-weighting techniques in later modeling stages.

[ ]:
```
# 5.3 Continuous Feature Distributions
# ------------------------------------
plt.figure(figsize=(6,4))
sns.boxplot(x='readmitted_binary', y='time_in_hospital', data=df)
plt.title("Hospital Stay Duration vs Readmission")
plt.xlabel("Readmitted within 30 days (1 = Yes, 0 = No)")
plt.ylabel("Time in Hospital (days)")
plt.show()
```

[ ]:
```
# 5.4 Feature Relationships (categorical features)
# ------------------------------------------------
fig, axes = plt.subplots(1, 2, figsize=(12, 5))

sns.countplot(x='race', hue='readmitted_binary', data=df, ax=axes[0])
axes[0].set_title("Race vs Readmission")
axes[0].set_xlabel("Race")
axes[0].tick_params(axis='x', rotation=45)

sns.countplot(x='admission_type_id', hue='readmitted_binary', data=df,␣
 ↪ax=axes[1])
axes[1].set_title("Admission Type vs Readmission")
axes[1].set_xlabel("Admission Type ID")
axes[1].tick_params(axis='x', rotation=45)

plt.tight_layout()
plt.show()

plt.figure(figsize=(6, 4))
sns.boxplot(x='readmitted_binary', y='service_utilization', data=df)
plt.title("Service Utilization vs Readmission")
plt.xlabel("Readmitted within 30 days (1 = Yes, 0 = No)")
plt.ylabel("Total Prior Visits")
plt.show()
```

Features `A1Cresult` and `max_glu_serum` were dropped earlier due to excessive missingness (>80%). We instead examine relationships between readmission and remaining features such as race, admission type, and service utilization.