```
# 🚨 Deletes everything from previous attempts
!rm -rf spark-*
!rm -rf /content/spark-*
!rm -rf /usr/local/lib/python*/dist-packages/pyspark
!rm -rf /usr/local/lib/python*/dist-packages/findspark
```

```
# ✅ Install Java (Spark needs it)
!apt-get install openjdk-11-jdk -y

# ✅ Download Apache Spark 3.5.1 (Hadoop 3)
!wget -q https://archive.apache.org/dist/spark/spark-3.5.1/spark-3.5.1-bin-hadoop3.tgz

# ✅ Extract Spark
!tar -xzf spark-3.5.1-bin-hadoop3.tgz

# ✅ Install findspark
!pip install -q findspark
```

```
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  fonts-dejavu-core fonts-dejavu-extra libatk-wrapper-java
  libatk-wrapper-java-jni libxt-dev libxtst6 libxxf86dga1 openjdk-11-jre
  x11-utils
Suggested packages:
  libxt-doc openjdk-11-demo openjdk-11-source visualvm mesa-utils
The following NEW packages will be installed:
  fonts-dejavu-core fonts-dejavu-extra libatk-wrapper-java
  libatk-wrapper-java-jni libxt-dev libxtst6 libxxf86dga1 openjdk-11-jdk
  openjdk-11-jre x11-utils
0 upgraded, 10 newly installed, 0 to remove and 38 not upgraded.
Need to get 5,367 kB of archives.
After this operation, 15.2 MB of additional disk space will be used.
Get:1 http://archive.ubuntu.com/ubuntu jammy/main amd64 fonts-dejavu-core all 2.37-2build1 [1,041 kB]
Get:2 http://archive.ubuntu.com/ubuntu jammy/main amd64 fonts-dejavu-extra all 2.37-2build1 [2,041 kB]
Get:3 http://archive.ubuntu.com/ubuntu jammy/main amd64 libxtst6 amd64 2:1.2.3-1build4 [13.4 kB]
Get:4 http://archive.ubuntu.com/ubuntu jammy/main amd64 libxxf86dga1 amd64 2:1.1.5-0ubuntu3 [12.6 kB]
Get:5 http://archive.ubuntu.com/ubuntu jammy/main amd64 x11-utils amd64 7.7+5build2 [206 kB]
Get:6 http://archive.ubuntu.com/ubuntu jammy/main amd64 libatk-wrapper-java all 0.38.0-5build1 [53.1 kB]
Get:7 http://archive.ubuntu.com/ubuntu jammy/main amd64 libatk-wrapper-java-jni amd64 0.38.0-5build1 [49.0 kB]
```

```
Get:8 http://archive.ubuntu.com/ubuntu jammy/main amd64 libxt-dev amd64 1:1.2.1-1 [396 kB]
Get:9 http://archive.ubuntu.com/ubuntu jammy-updates/main amd64 openjdk-11-jre amd64 11.0.28+6-1ubuntu1~22.04.1 [214 kB]
Get:10 http://archive.ubuntu.com/ubuntu jammy-updates/main amd64 openjdk-11-jdk amd64 11.0.28+6-1ubuntu1~22.04.1 [1,342 kB]
Fetched 5,367 kB in 1s (3,580 kB/s)
Selecting previously unselected package fonts-dejavu-core.
(Reading database ... 126675 files and directories currently installed.)
Preparing to unpack .../0-fonts-dejavu-core_2.37-2build1_all.deb ...
Unpacking fonts-dejavu-core (2.37-2build1) ...
Selecting previously unselected package fonts-dejavu-extra.
Preparing to unpack .../1-fonts-dejavu-extra_2.37-2build1_all.deb ...
Unpacking fonts-dejavu-extra (2.37-2build1) ...
Selecting previously unselected package libxtst6:amd64.
Preparing to unpack .../2-libxtst6_2%3a1.2.3-1build4_amd64.deb ...
Unpacking libxtst6:amd64 (2:1.2.3-1build4) ...
Selecting previously unselected package libxxf86dga1:amd64.
Preparing to unpack .../3-libxxf86dga1_2%3a1.1.5-0ubuntu3_amd64.deb ...
Unpacking libxxf86dga1:amd64 (2:1.1.5-0ubuntu3) ...
Selecting previously unselected package x11-utils.
Preparing to unpack .../4-x11-utils_7.7+5build2_amd64.deb ...
Unpacking x11-utils (7.7+5build2) ...
Selecting previously unselected package libatk-wrapper-java.
Preparing to unpack .../5-libatk-wrapper-java_0.38.0-5build1_all.deb ...
Unpacking libatk-wrapper-java (0.38.0-5build1) ...
Selecting previously unselected package libatk-wrapper-java-jni:amd64.
Preparing to unpack .../6-libatk-wrapper-java-jni_0.38.0-5build1_amd64.deb ...
Unpacking libatk-wrapper-java-jni:amd64 (0.38.0-5build1) ...
Selecting previously unselected package libxt-dev:amd64.
Preparing to unpack .../7-libxt-dev_1%3a1.2.1-1_amd64.deb ...
Unpacking libxt-dev:amd64 (1:1.2.1-1) ...
Selecting previously unselected package openjdk-11-jre:amd64.
Preparing to unpack .../8-openjdk-11-jre_11.0.28+6-1ubuntu1~22.04.1_amd64.deb ...
Unpacking openjdk-11-jre:amd64 (11.0.28+6-1ubuntu1~22.04.1) ...
Selecting previously unselected package openjdk-11-jdk:amd64.
Preparing to unpack .../9-openjdk-11-jdk_11.0.28+6-1ubuntu1~22.04.1_amd64.deb ...
```

```
import os

os.environ["JAVA_HOME"] = "/usr/lib/jvm/java-11-openjdk-amd64"
os.environ["SPARK_HOME"] = "/content/spark-3.5.1-bin-hadoop3"

# Safely append Spark bin to PATH
os.environ["PATH"] = os.environ.get("PATH", "") + os.pathsep + os.path.join(os.environ["SPARK_HOME"], "bin")
```

```
import os, findspark

# Set environment variables
os.environ["JAVA_HOME"] = "/usr/lib/jvm/java-11-openjdk-amd64"
os.environ["SPARK_HOME"] = "/content/spark-3.5.1-bin-hadoop3"
os.environ["PATH"] += os.pathsep + os.path.join(os.environ["SPARK_HOME"], "bin")

# Initialize Spark
findspark.init("/content/spark-3.5.1-bin-hadoop3")

from pyspark.sql import SparkSession
spark = SparkSession.builder.appName("MySparkApp").getOrCreate()

print("✅ Spark setup complete!")
print("Spark Version:", spark.version)
```

```
✅ Spark setup complete!
Spark Version: 3.5.1
```

```
data = [("Alice", 25), ("Bob", 30), ("Cathy", 27)]
df = spark.createDataFrame(data, ["Name", "Age"])
df.show()
```

```
+-----+---+
| Name|Age|
+-----+---+
|Alice| 25|
|  Bob| 30|
|Cathy| 27|
+-----+---+
```

```
# ------------------------------------------------------------
# 🏥 PATIENT READMISSION PREDICTION USING PYSPARK + RANDOM FOREST
# ------------------------------------------------------------

# Import required libraries
from pyspark.sql import SparkSession
from pyspark.sql.functions import when, col, isnan, count, desc
from pyspark.ml import Pipeline
from pyspark.ml.feature import StringIndexer, VectorAssembler, StandardScaler
```

```
from pyspark.ml.classification import RandomForestClassifier
from pyspark.ml.evaluation import BinaryClassificationEvaluator
```

```
from google.colab import files

# Upload the CSV
uploaded = files.upload()  # A file picker will appear

# Get the filename (adjust if your file has a different name)
filename = list(uploaded.keys())[0]

# Read CSV with Spark
df = spark.read.csv(filename, header=True, inferSchema=True)
print("✅ Data Loaded Successfully!")
print(f"Total Rows: {df.count()} | Total Columns: {len(df.columns)}")
```

```
Choose Files diabetic_data.csv
diabetic_data.csv(text/csv) - 19159383 bytes, last modified: 21/10/2025 - 100% done
Saving diabetic_data.csv to diabetic_data.csv
✅ Data Loaded Successfully!
Total Rows: 101766 | Total Columns: 50
```

```
# ------------------------------------------------------------
# 4️⃣ Clean Dataset
# ------------------------------------------------------------
df = df.drop("encounter_id", "patient_nbr").replace('?', None)
df = df.withColumn("readmitted_flag", when(col("readmitted") == "<30", 1).otherwise(0))
df = df.na.fill({"race": "Unknown", "gender": "Unknown", "age": "Unknown"})

# ------------------------------------------------------------
# 5️⃣ Missing Values Overview
# ------------------------------------------------------------
missing = df.select([count(when(col(c).isNull() | isnan(c), c)).alias(c) for c in df.columns])
missing.show(vertical=True)

# ------------------------------------------------------------
# 6️⃣ Summary Stats for Numeric Columns
# ------------------------------------------------------------
numeric_cols = [
    "time_in_hospital","num_lab_procedures","num_medications",
```

```
    "number_outpatient","number_emergency","number_inpatient","number_diagnoses"
]
df.describe(numeric_cols).show()
```

```
 weight                    | 98569
 admission_type_id         | 0
 discharge_disposition_id  | 0
 admission_source_id       | 0
 time_in_hospital          | 0
 payer_code                | 40256
 medical_specialty         | 49949
 num_lab_procedures        | 0
 num_procedures            | 0
 num_medications           | 0
 number_outpatient         | 0
 number_emergency          | 0
 number_inpatient          | 0
 diag_1                    | 21
 diag_2                    | 358
 diag_3                    | 1423
 number_diagnoses          | 0
 max_glu_serum             | 0
 A1Cresult                 | 0
 metformin                 | 0
 repaglinide               | 0
 nateglinide               | 0
 chlorpropamide            | 0
 glimepiride               | 0
 acetohexamide             | 0
 glipizide                 | 0
 glyburide                 | 0
 tolbutamide               | 0
 pioglitazone              | 0
 rosiglitazone             | 0
 acarbose                  | 0
 miglitol                  | 0
 troglitazone              | 0
 tolazamide                | 0
 examide                   | 0
 citoglipton               | 0
 insulin                   | 0
```

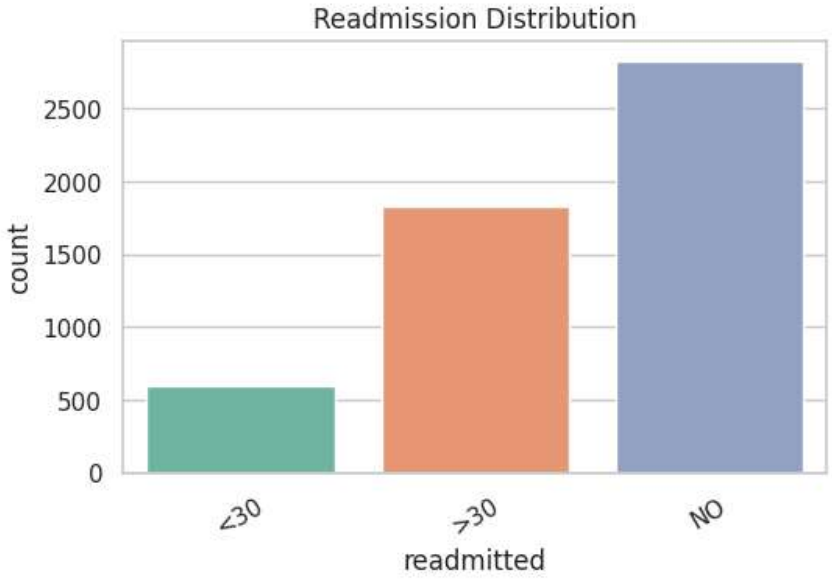
```
 metformin-pioglitazone   | 0
 change                   | 0
 diabetesMed              | 0
 readmitted               | 0
 readmitted_flag          | 0
```

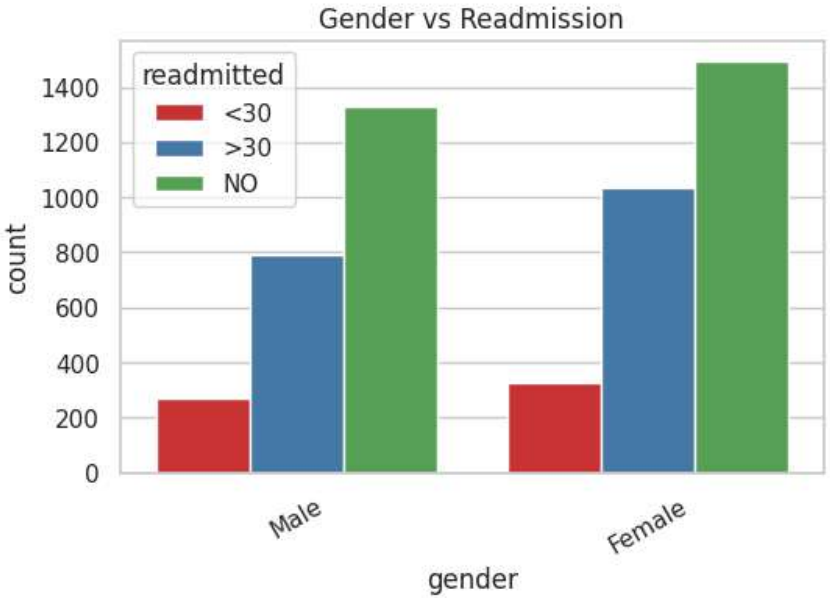
```
+-------+------------------+------------------+------------------+------------------+-------------------+------------------+-----
|summary|  time_in_hospital|num_lab_procedures|   num_medications|  number_outpatient|   number_emergency|  number_inpatient| numb
+-------+------------------+------------------+------------------+------------------+-------------------+------------------+-----
|  count|            101766|            101766|            101766|             101766|             101766|            101766|
|   mean| 4.395986871843248| 43.09564098028811|16.021844230882614|0.36935715268360747|0.19783621248747127| 0.635565906098304|7.422
| stddev|2.9851077674712636|19.674362249142053| 8.127566209167295| 1.2672650965326762|  0.930472268422466|1.2628632900973216| 1.93
|    min|                 1|                 1|                 1|                  0|                  0|                 0|
|    max|                14|               132|                81|                 42|                 76|                21|
+-------+------------------+------------------+------------------+------------------+-------------------+------------------+-----
```

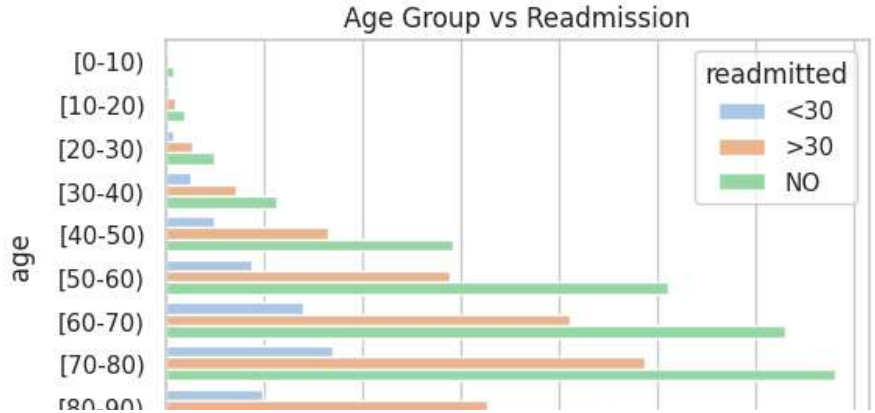
```
# ------------------------------------------------------------
# 7️⃣ Convert Small Sample to Pandas for Visualization
# ------------------------------------------------------------

import matplotlib.pyplot as plt
import seaborn as sns

# Sample 5% for visualization
pdf = df.sample(False, 0.05, seed=42).toPandas()

# Essential numeric columns
numeric_cols = ["time_in_hospital", "num_lab_procedures", "num_medications",
                "number_outpatient", "number_emergency", "number_inpatient",
                "number_diagnoses"]

sns.set(style="whitegrid")

# Create figure with compact size
fig, axes = plt.subplots(3, 2, figsize=(12, 12))
fig.tight_layout(pad=4.0)

# 1️⃣ Readmission Distribution
sns.countplot(x="readmitted", data=pdf, ax=axes[0,0], palette="Set2")
axes[0,0].set_title("Readmission Distribution")

# 2️⃣ Gender vs Readmission
sns.countplot(x="gender", hue="readmitted", data=pdf, ax=axes[0,1], palette="Set1")
```

```
axes[0,1].set_title("Gender vs Readmission")

# 3️⃣ Age vs Readmission
sns.countplot(y="age", hue="readmitted", data=pdf, order=sorted(pdf["age"].unique()), ax=axes[1,0], palette="pastel")
axes[1,0].set_title("Age Group vs Readmission")

# 5️⃣ Number of Medications vs Readmission
sns.boxplot(x="readmitted", y="num_medications", data=pdf, ax=axes[2,0], palette="Set3")
axes[2,0].set_title("Num Medications vs Readmission")

# 6️⃣ Time in Hospital vs Readmission
sns.boxplot(x="readmitted", y="time_in_hospital", data=pdf, ax=axes[2,1], palette="Set2")
axes[2,1].set_title("Time in Hospital vs Readmission")

# Improve layout for visibility
for ax in axes.flatten():
    ax.tick_params(axis='x', rotation=30)
    ax.tick_params(axis='y', rotation=0)

plt.show()

#7  Create a separate figure for correlation
plt.figure(figsize=(7,6))
sns.heatmap(pdf[numeric_cols].corr(), annot=True, fmt=".2f", cmap="coolwarm",
            square=True, cbar=True, linewidths=0.5,
            annot_kws={"size":10})

plt.xticks(rotation=45, ha='right', fontsize=10)
plt.yticks(rotation=0, fontsize=10)
plt.title("Numeric Feature Correlation", fontsize=12)
plt.tight_layout()
plt.show()
```

```
/tmp/ipython-input-2687379296.py:23: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v0.14.0. Assign the `x` variable to `hue` and set `le

  sns.countplot(x="readmitted", data=pdf, ax=axes[0,0], palette="Set2")
/tmp/ipython-input-2687379296.py:35: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v0.14.0. Assign the `x` variable to `hue` and set `le

  sns.boxplot(x="readmitted", y="num_medications", data=pdf, ax=axes[2,0], palette="Set3")
/tmp/ipython-input-2687379296.py:39: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v0.14.0. Assign the `x` variable to `hue` and set `le

  sns.boxplot(x="readmitted", y="time_in_hospital", data=pdf, ax=axes[2,1], palette="Set2")
```

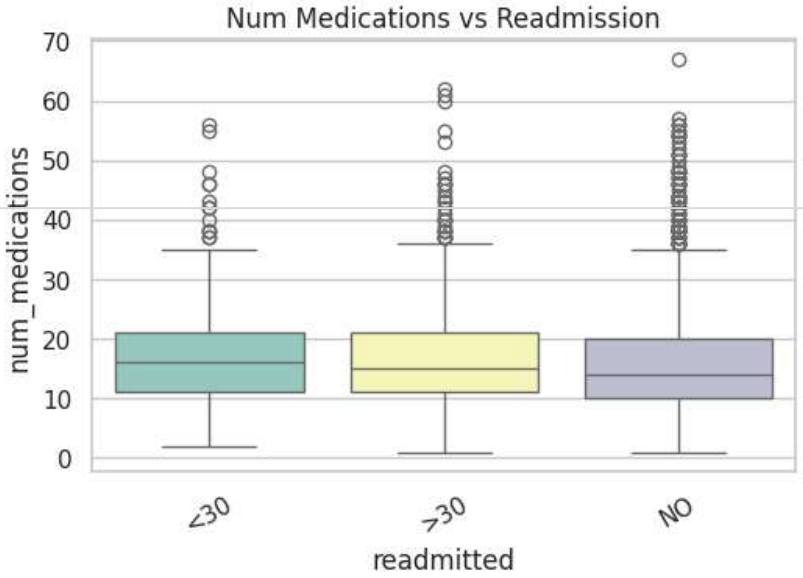


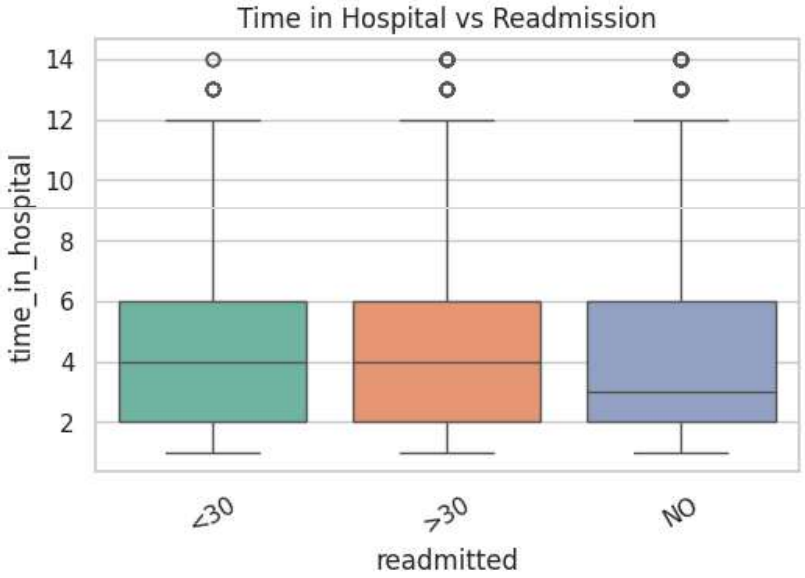


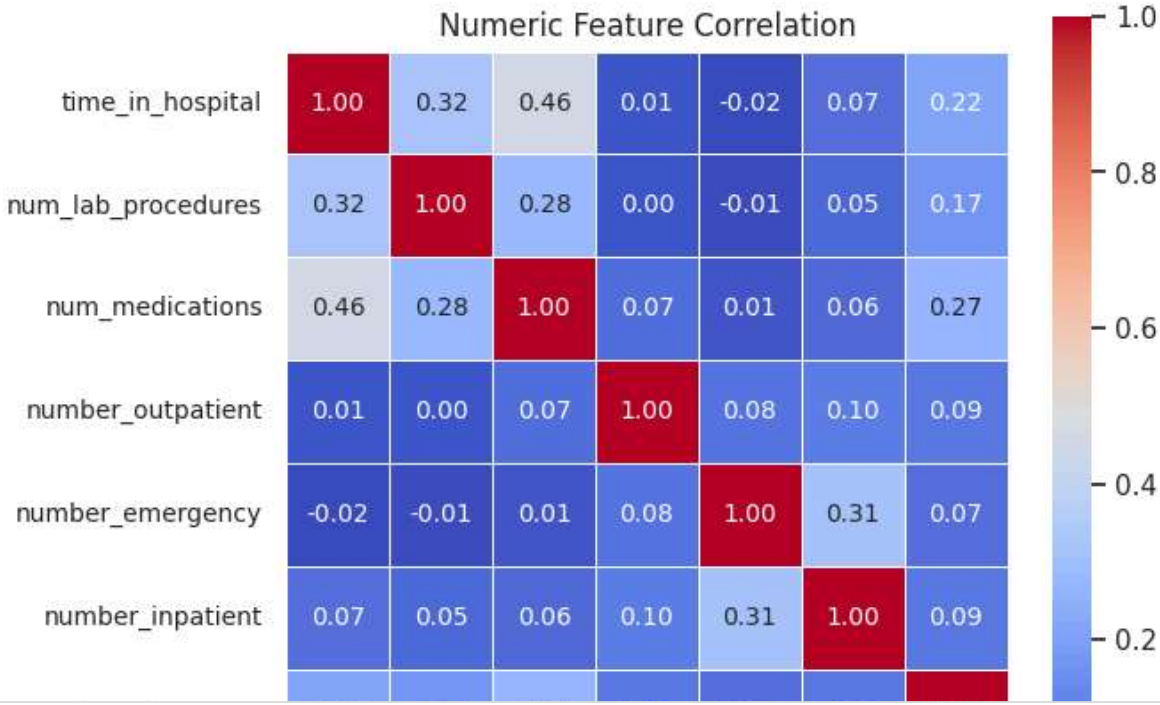



[80-90)
[90-100)
0
100
200
300
400
500
600
700
count


0.0
0.0
0.2
0.4
0.6
0.8
1.0

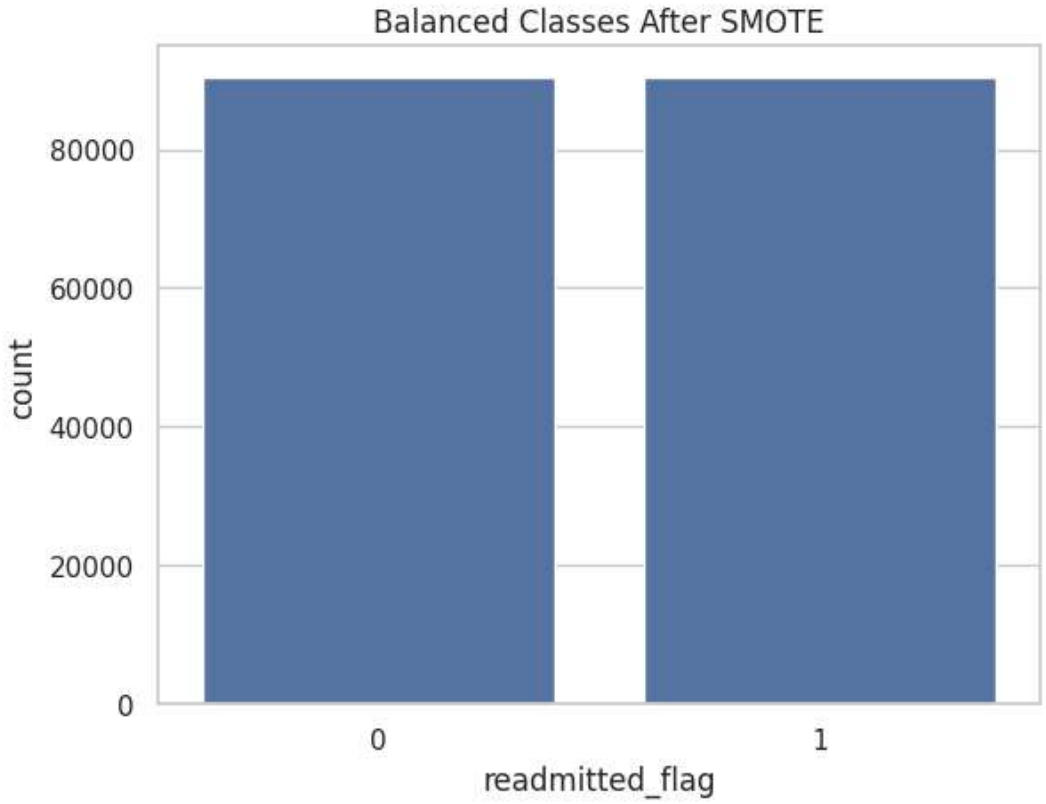

Num Medications vs Readmission
num_medications
0
10
20
30
40
50
60
70
<30
>30
NO
readmitted


Time in Hospital vs Readmission
time_in_hospital
2
4
6
8
10
12
14
<30
>30
NO
readmitted


Numeric Feature Correlation
time_in_hospital
num_lab_procedures
num_medications
number_outpatient
number_emergency
number_inpatient
1.00 0.32 0.46 0.01 -0.02 0.07 0.22
0.32 1.00 0.28 0.00 -0.01 0.05 0.17
0.46 0.28 1.00 0.07 0.01 0.06 0.27
0.01 0.00 0.07 1.00 0.08 0.10 0.09
-0.02 -0.01 0.01 0.08 1.00 0.31 0.07
0.07 0.05 0.06 0.10 0.31 1.00 0.09
1.0
0.8
0.6
0.4
0.2

```
# ------------------------------------------------------------
# 8️⃣ Prepare Data for SMOTE (requires numeric input)
# ------------------------------------------------------------
categorical_cols = ["race","gender","age","A1Cresult","insulin","change"]

# Convert to Pandas for SMOTE
pandas_df = df.select(categorical_cols + numeric_cols + ["readmitted_flag"]).toPandas()

from sklearn.preprocessing import LabelEncoder
for colname in categorical_cols:
    le = LabelEncoder()
    pandas_df[colname] = le.fit_transform(pandas_df[colname].astype(str))

X = pandas_df.drop("readmitted_flag", axis=1)
y = pandas_df["readmitted_flag"]

# ------------------------------------------------------------
# 9️⃣ Apply SMOTE
# ------------------------------------------------------------
from imblearn.over_sampling import SMOTE
smote = SMOTE(random_state=42)
X_res, y_res = smote.fit_resample(X, y)

print("✅ After SMOTE Balancing:")
print(y_res.value_counts())

# Visualize class balance
sns.countplot(x=y_res)
plt.title("Balanced Classes After SMOTE")
plt.show()

# Merge back
balanced_df = X_res.copy()
balanced_df["readmitted_flag"] = y_res

# Convert back to PySpark
df_balanced = spark.createDataFrame(balanced_df)
```

```
✅ After SMOTE Balancing:
readmitted_flag
0    90409
1    90409
Name: count, dtype: int64
```




```
# ------------------------------------------------------------
# 🔟 Build Feature Pipeline (Optimized for Memory)
# ------------------------------------------------------------
from pyspark.ml.feature import StringIndexer, VectorAssembler, StandardScaler
from pyspark.ml.classification import RandomForestClassifier
from pyspark.ml import Pipeline
from pyspark.ml.evaluation import BinaryClassificationEvaluator

# Index categorical features safely
indexers = [
    StringIndexer(inputCol=c, outputCol=c + "_index", handleInvalid="keep")
    for c in categorical_cols
```

```
]

# Assemble numeric + indexed categorical features
assembler = VectorAssembler(
    inputCols=numeric_cols + [c + "_index" for c in categorical_cols],
    outputCol="features_unscaled"
)

# Scale features for stability
scaler = StandardScaler(inputCol="features_unscaled", outputCol="features")

# ✅ Optimized Random Forest — smaller depth & tree count to prevent OOM
rf = RandomForestClassifier(
    labelCol="readmitted_flag",
    featuresCol="features",
    numTrees=80,        # reduced from 200 → lighter & faster
    maxDepth=10,        # reduced from 15 → avoids heap crash
    maxBins=32,         # fewer histogram bins = less memory
    minInstancesPerNode=10,
    seed=42
)

# Create full pipeline
pipeline = Pipeline(stages=indexers + [assembler, scaler, rf])
```

```
# ------------------------------------------------------------
# 1️⃣1️⃣ Train/Test Split (Stratified sampling optional)
# ------------------------------------------------------------
train, test = df_balanced.randomSplit([0.7, 0.3], seed=42)

print(f"Training rows: {train.count()} | Test rows: {test.count()}")
```

```
Training rows: 126686 | Test rows: 54132
```

```
# ------------------------------------------------------------
# 1️⃣2️⃣ Train Model
# ------------------------------------------------------------
print("🚀 Training Random Forest model... please wait ~2–4 minutes")
model = pipeline.fit(train)
print("✅ Model Training Completed Successfully!")
```