Silver Layer Transformation: Diabetes 30-Day Readmission Project

Automated Pipeline for Cleaning, Feature Engineering, and Validation.

Source: Databricks Notebook Execution Log

Pipeline Objectives & Success Criteria

The goal of this process is to transform raw Bronze data into a high-quality Silver dataset suitable for training Machine Learning models to predict patient readmission.

Standardization
Convert non-standard missing value indicators (`?`) into valid null types for Spark processing.

Noise Reduction
Remove high-cardinality identifiers (`encounter_id`, `patient_nbr`) to prevent model overfitting.

Feature Engineering
Transform the multi-class readmitted field into a binary target variable `readmitted_30` (1 if <30 days, else 0).

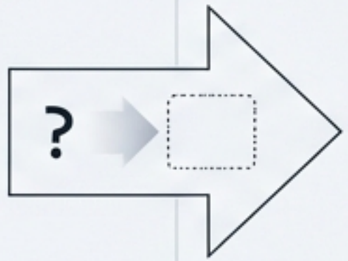
Validation
Verify schema integrity and assess class balance prior to persistence.

Outcome: A versioned Delta table '**diabetes_readmissions.silver_diabetes_readmission**' ready for downstream analytics.

Phase I: Ingestion & Standardization

```
# STEP 1: Read Data
silver_df =
    spark.table("diabetes_readmissions.b
    ronze_diabetes_readmission")

# STEP 2: Replace '?' with null values
silver_df = silver_df.replace("?", None)
```



- **Ingest:** Loads raw diabetes readmission data from the Bronze Delta table.
- **The Issue:** The source system encodes missing values as the string character '?'. Spark interprets this as a valid string, skewing statistical profiles.
- **The Fix**: We explicitly replace '?' with None (SQL NULL). This allows downstream aggregators to correctly identify and handle missing data.

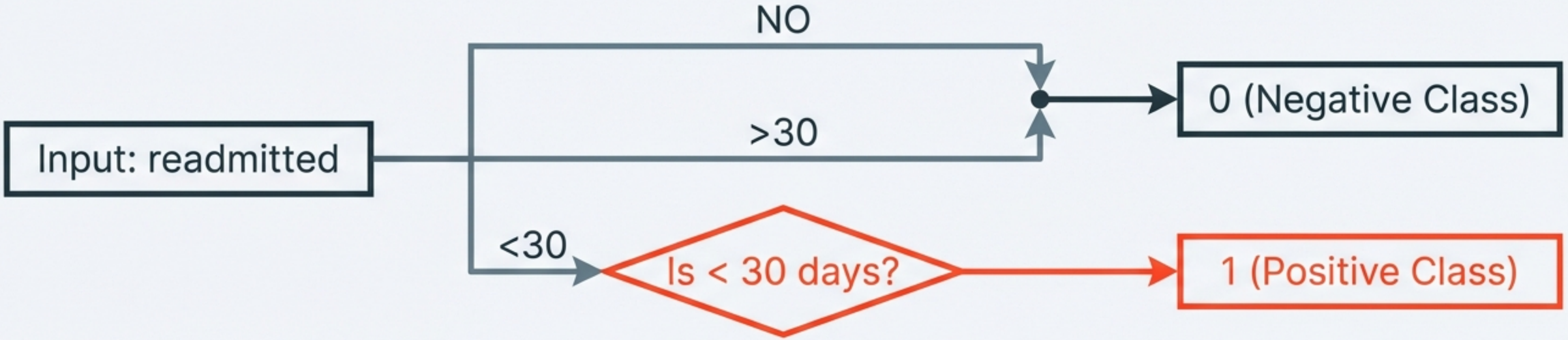
Phase II: Noise Reduction & Privacy


```
# Step 3: Drop identifier columns
# Remove unique identifiers that are not useful for modeling.
silver_df = silver_df.drop("encounter_id", "patient_nbr")
```

Data Science Context

- **Action:** Dropping `encounter_id` and `patient_nbr`.
- **Rationale:** These are unique identifiers with high cardinality. If left in the dataset, ML models might 'memorize' these IDs rather than learning patterns (**overfitting**). They offer no predictive signal for readmission risk.
- **Result:** The dataset is reduced to feature columns (race, gender, age, etc.) relevant to clinical analysis.

Phase III: Feature Engineering the Target Variable

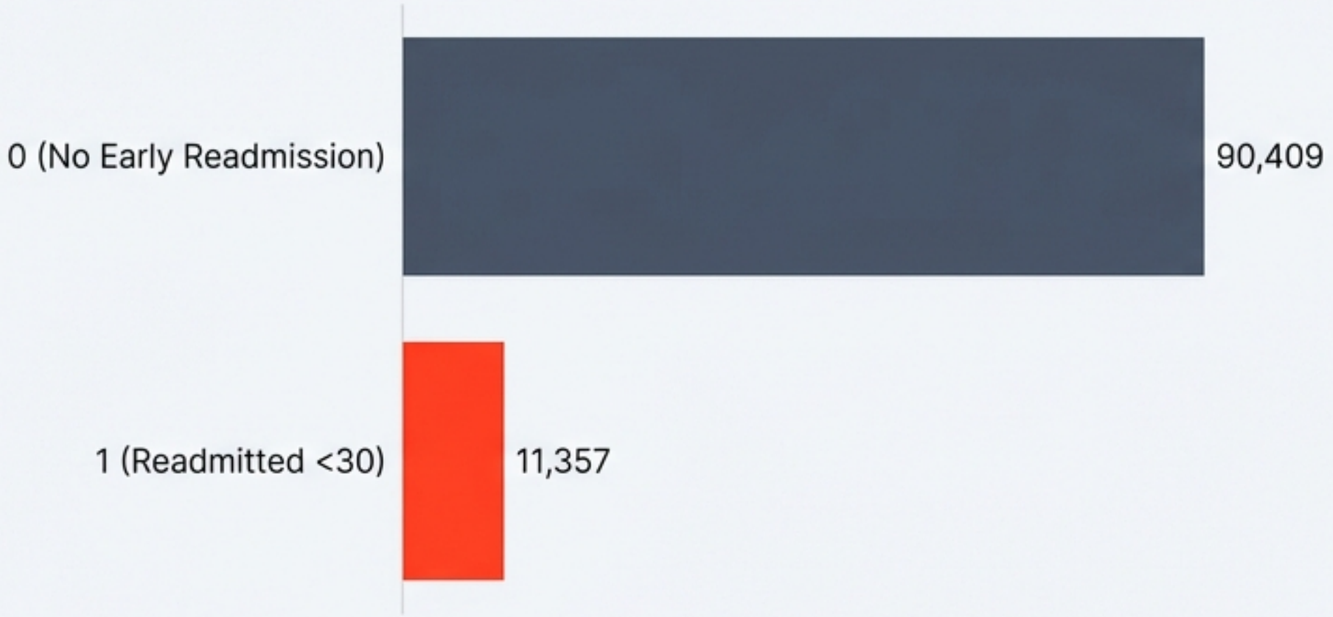



```python
# Step 4: Create 30-day readmission target variable
from pyspark.sql.functions import when, col

silver_df = silver_df.withColumn(
    "readmitted_30",
    when(col("readmitted") == "<30", 1).otherwise(0)
)
```

**Note: The original 'readmitted' column is subsequently dropped to prevent target leakage.*

Phase IV: Validation - Class Distribution Analysis




```
readmitted_30 | count
            0 | 90409
            1 | 11357
```

Severe Class Imbalance Detected

The positive class (1) represents only ~11% of the total dataset.

Implication: Downstream modeling will require resampling techniques (such as SMOTE or undersampling) or weighted loss functions to prevent the model from biasing toward the majority class.

Phase IV: Validation - Schema Integrity

```
root
 |-- race: string (nullable = true)
 |-- gender: string (nullable = true)
 |-- age: string (nullable = true)
 |-- weight: string (nullable = true)
 |-- time_in_hospital: integer (nullable = true)
 |-- num_lab_procedures: integer (nullable = true)
 |-- num_medications: integer (nullable = true)
 |-- readmitted_30: integer (nullable = true)
```

Verification Checklist

- ✓ **Categorical Fields:** 'race', 'gender' correctly typed as Strings.
- ✓ **Numerical Fields:** 'time_in_hospital', 'num_medications' correctly typed as Integers.
- ✓ **Target Variable:** 'readmitted_30' confirmed present.

Phase IV: Validation - Data Preview

Row-level inspection of the transformed dataset.

race	gender	age	time_in_hospital	num_medications	readmitted_30
Caucasian	Female	[0-10)	1	1	0
Caucasian	Female	[10-20)	3	18	0
AfricanAmerican	Female	[20-30)	2	13	0
Caucasian	Male	[30-40)	2	16	0
Caucasian	Male	[40-50)	1	8	0

Assessment: Data is tabular, structured, and features are populated correctly. The target variable is binary integer format.

Phase V: Persistence to Silver Layer

```
# STEP 9: Save Silver DataFrame as Delta Table
silver_df.write.format("delta") \
    .mode("overwrite") \
    .saveAsTable("diabetes_readmissions.silver_diabetes_readmission")
```

Format: Delta

Ensures ACID transactions, scalable metadata handling, and time travel capabilities.

Mode: Overwrite

Replaces existing table data atomically with the newly processed batch, ensuring the Silver layer reflects the most current logic.

Accessibility

Table registered in Hive Metastore, making it instantly queryable via SQL or Python by other teams.

Transformation Complete: Ready for Modeling

Pipeline Execution Summary

- ☑ **Input:** Raw Bronze Data (handled '?' artifacts).
- ☑ **Privacy:** Dropped high-cardinality identifiers.
- ☑ **Engineering:** Created `'readmitted_30'` binary target.
- ☑ **Validation:** Verified schema and distribution.
- ☑ **Output:** Validated Silver Data.

Immediate Next Steps

1. **Exploratory Data Analysis (EDA)** on the Silver Table.
2. **Gold Layer Creation:** Aggregating stats by demographic.
3. **Model Training:** Addressing the 11% class imbalance identified in Phase IV.

Silver Layer