Team member's details:
Group Name: **Data Dominators**

chau.devin031602@gmail.com	Devin Chau	United States	San Jose State University	Data Science

rohankhatri0507@gmail.com	Rohan Khatri	United States	San Jose State University	Data Science

ethan05dy@gmail.com	Ethan Dy	United States	San Jose State University	Data Science

Project: Data Science:: Healthcare - Persistency of a drug

Problem description

We are building a predictive model that classifies patients into “persistent” or “non-persistent” categories based on factors like their demographics, medical history, physician characteristics, and treatment details. Factors like the patient level such as their age, risk factors, previous test results, or provider type allows for insights into why some patients continue therapy while others drop off. Thus understanding “persistence” levels. By analyzing these data points and finding patterns, the predictive model helps explain patient behavior and supports the creation of targeted interventions to improve adherence.

Data Cleansing and Transformation

Data Normalization and Duplicate Checks

- **MinMaxScaler**: The numerical columns were normalized to a common scale using MinMaxScaler to ensure that all variables contributed equally to the model.
- **Duplicate Checks**: Detected duplicate rows based on the 'PatientID' column, which could skew analysis. The duplicates were logged for further review.
- **Standardization of Data Types**: Columns were converted to appropriate data types to ensure consistency across the dataset, minimizing errors during analysis.

Consolidated Results

My Contribution

Standardizing data types:

```
categorical_cols = df.select_dtypes(include=['object']).columns.tolist()
numerical_cols = df.select_dtypes(include=['int64', 'float64']).columns.tolist()

df[categorical_cols] = df[categorical_cols].astype('category')

encoded_data = df.copy()
for col in categorical_cols:
    encoded_data[col] = encoded_data[col].cat.codes


```

Use MinMaxScalar to normalize and check for duplicate rows in the PatientID column:

```
risk_bins = [-1, 0, 2, 5, float('inf')]
risk_labels = ['None', 'Low', 'Moderate', 'High']
encoded_data['Risk_Level'] = pd.cut(encoded_data['Count_Of_Risks'].astype(int),
                                    bins=risk_bins,
                                    labels=risk_labels)

```

```
[ ]  scaler = MinMaxScaler()
     encoded_data[numerical_cols] = scaler.fit_transform(encoded_data[numerical_cols])

```

```
[ ]  duplicates = df.duplicated(subset='PatientID').sum()

     summary_stats = encoded_data.describe(include='all')

     {
         "categorical_cols": categorical_cols,
         "numerical_cols": numerical_cols,
         "duplicates_found": duplicates,
         "summary_stats": summary_stats
     }

```

- **Summary of Steps**:
    - Encoded risk levels into categorized bins.
    - Scaled numerical columns using MinMaxScaler.
    - Detected duplicates and provided a summary of the dataset.
- **Observations**:

- Duplicate entries detected: duplicates.
- Statistical summary of all variables provided insights into data distribution.

Github Repo link
https://github.com/ethan05d/DataGlacier-Internship/tree/main/Week%209