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

The data cleansing and transformation process included a combination of techniques to enhance data quality and ensure compatibility with modeling tools:

- **Data Normalization and Duplicate Checks:**
  - MinMaxScaler was used to normalize numerical columns, bringing all variables to a common scale and ensuring equal contribution to the model.

```
[ ]  1 scaler = MinMaxScaler()
     2 encoded_data[numerical_cols] = scaler.fit_transform(encoded_data[numerical_cols])
     3
```

- Duplicate rows were detected based on the 'PatientID' column to prevent skewed analyses, and they were logged for further review.

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

- Columns were standardized to appropriate data types, reducing potential errors during analysis.

- **Outlier Handling:**
  - Devin Chau removed outliers from the Dexa_Freq_During_Rx column by excluding entries where its value was zero.

```
df = df[df['Dexa_Freq_During_Rx'] != 0]
```

- Rohan Khatri applied the IQR method to cap extreme values within acceptable ranges, addressing potential data skew.

```
# Using IQR
Q1 = df['Dexa_Freq_During_Rx'].quantile(0.25)
Q3 = df['Dexa_Freq_During_Rx'].quantile(0.75)
IQR = Q3 - Q1
lower_bound = Q1 - 1.5 * IQR
upper_bound = Q3 + 1.5 * IQR

df['Dexa_Freq_During_Rx_No_Outliers'] = np.where(
    df['Dexa_Freq_During_Rx'] > upper_bound, upper_bound,
    np.where(df['Dexa_Freq_During_Rx'] < lower_bound, lower_bound, df['Dexa_Freq_During_Rx'])
)
df['Dexa_Freq_During_Rx_No_Outliers'].describe()

```

- **Binary Encoding:**
  - Devin Chau converted categorical binary values ('N' and 'Y') into numerical representations (0 and 1), ensuring compatibility with classifier models.

```
binary_columns = [col for col in df.columns if set(df[col].dropna().unique()) == {'N', 'Y'}]

for col in binary_columns:
    df[col] = df[col].replace({'N': 0, 'Y': 1}).astype(int)
```

- **Risk Level Encoding:**
    - Ethan Dy categorized patient risk levels into bins such as 'None', 'Low', 'Moderate', and 'High' based on Count_Of_Risks values, enabling targeted risk analysis.

```
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)

```

- **Feature Scaling:**
    - Ethan Dy scaled numerical features using MinMaxScaler to normalize their range, ensuring model fairness.

Consolidated Results

Observations

- **Outlier Removal**: Improved data consistency by removing erroneous values from Dexa_Freq_During_Rx.
- **Binary Encoding**: Facilitated the use of binary data in models by converting values into integers.
- **Risk Level Encoding**: Categorized patient risk levels for analysis.
- **Normalization**: Ensured all variables contributed equally to the model.
- **Duplicate Checks**: Identified potential data redundancies.
- **Logistic Regression**: Provided risk predictions with an evaluated accuracy.
- **IQR Outlier Handling**: Capped extreme values to enhance data quality.