Structured Data Assignment

Documentation

Problem 2 - Drugs are generally administered/prescribed by the physicians for a certain period of time or they are administered at regular intervals, but for various reasons patients might stop taking the treatment. Consider following example for better understanding. Let's say you get a throat infection, the physician prescribes you an antibiotic for 10 days, but you stop taking the treatment after 3 days because of some adverse events. In the above example ideal treatment duration is 10 days but patients stopped taking treatment after 3 days due to adverse events. Patients stopping a treatment is called drop off. We want to study drop off for "Target Drug", the aim is to generate insights on what events lead to patients stopping on "Target Drug". Assume ideal treatment duration for "Target Drug" is 1 year, come up with analysis showing how drop-off rate is, drop off rate is defined as number of patients dropping off each month. Then come up with analysis to generate insights on what events are driving a patient to stop taking "Target Drug".

In the context of medical treatment, patients are often prescribed specific drugs for a defined period or at regular intervals. However, there are instances where patients discontinue their treatment prematurely, which is referred to as "drop off." This analysis aims to study the drop off rate for a specific drug, referred to as the "Target Drug," with an ideal treatment duration of one year. The goal is to gain insights into the factors and events leading patients to stop taking the "Target Drug."

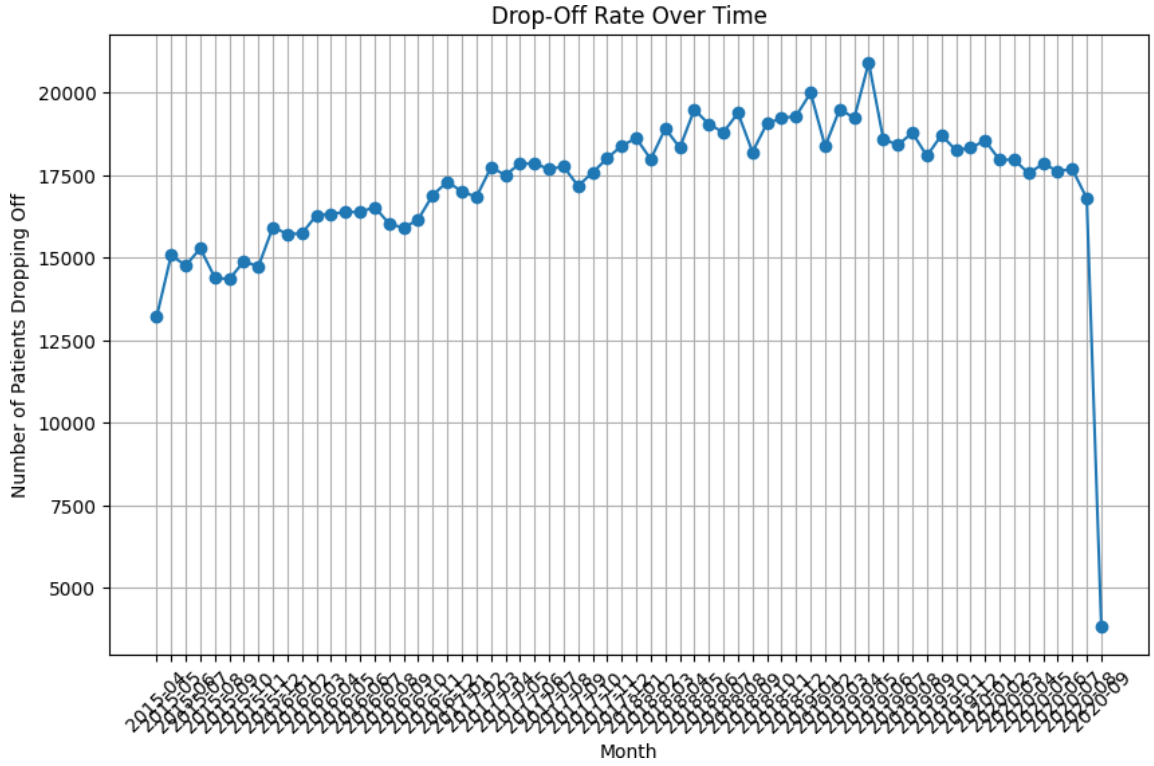
Steps:

1. **Import necessary libraries:**
- pandas for data manipulation and analysis.
- matplotlib.pyplot for data visualization.
2. **Load the dataset:**
- Reads the data from a Parquet file named "train.parquet" using pd.read_parquet.
3. **Data Preprocessing:**
- Converts the "Date" column to datetime format using pd.to_datetime.
- Defines the ideal treatment duration as one year using pd.DateOffset.
4. **Calculate Drop-off Rate Over Time:**
- Creates a new column "Month" by extracting the month and year from the "Date" column using dt.to_period('M').
- Groups the data by month and counts the number of unique patients in each group.
5. **Data Visualization - Drop-off Rate Over Time:**
- Plots the drop-off rate over time.
- Sets labels and a title for the plot.
- Displays the plot with x-axis labels rotated for readability.
6. **Analyse Common Incidents Leading to Drop-off:**
- Examines the "Incident" column to identify common incidents associated with patients stopping treatment.
- Counts the frequency of each unique incident.
7. **Data Visualization - Common Incidents Leading to Drop-off:**
- Plots a graph of common incidents and their frequency.
- Sets labels and a title for the plot.
- Displays the plot with x-axis labels rotated for readability.

This project provides a visual representation of the drop-off rate for the "Target Drug" over time and identifies common incidents leading to patients discontinuing the treatment. The documentation outlines the problem statement, the steps taken to analyse the data, and the visual representations created for better understanding.

Program & Results:

```
import pandas as pd
import matplotlib.pyplot as plt


# Load the dataset
data = pd.read_parquet("sample_data/train.parquet")


# Convert the 'Date' column to datetime
data['Date'] = pd.to_datetime(data['Date'])


# Assuming the ideal treatment
duration is 1 yearideal_duration =
pd.DateOffset(years=1)


# Calculate drop-off rate by counting the number of patients who
stopped each monthdata['Month'] = data['Date'].dt.to_period('M')
dropoff_rate = data.groupby('Month')['Patient-Uid'].nunique()


# Convert the 'Month' period to a string for plotting
dropoff_rate.index = dropoff_rate.index.strftime('%Y-%m')

# Plot the drop-off rate
over time
plt.figure(figsize=(10, 6))
plt.plot(dropoff_rate.index, dropoff_rate.values, marker='o',
linestyle='-')plt.xlabel('Month')
plt.ylabel('Number of Patients
Dropping Off')plt.title('Drop-Off
Rate Over Time')
plt.grid(True)
plt.xticks(rotation=45) # Rotate x-axis labels for better
readabilityplt.show()
```



To analyze what events are driving drop-off, examine the 'Incident' column and identify common incidents associated with patients stoppcommon_incidents = data['Incident'].value_counts()

Print the most common incidents
print("Common Incidents Leading to
Drop-Off:")
print(common_incidents.head(10))

Common Incidents Leading to Drop-Off:

DRUG_TYPE_6	561934
DRUG_TYPE_1	484666
PRIMARY_DIAGNOSIS	431902
DRUG_TYPE_0	300005
DRUG_TYPE_7	258782
DRUG_TYPE_2	256841
DRUG_TYPE_8	160066
DRUG_TYPE_3	127676
TEST_TYPE_1	96810
TARGET DRUG	67218

Name: Incident, dtype: int64

```
# Plot graph for common incidents leading
to drp-off ( Incidents Vs Frequency)
plt.figure(figsize=(10, 6))
plt.plot(common_incide
nts, marker='o',
linestyle='-')
plt.xlabel('Incidents')
plt.ylabel('Frequency')
plt.title('Common
Incidents Leading to
Drop-Off:')
plt.grid(True)
plt.xticks(rotation=45) # Rotate x-axis
labels for better readabilityplt.show()
```