**Project Description**

The "Healthcare Appointments Analytics" project is a comprehensive data analysis initiative that delves into the intricacies of medical appointments and explores factors influencing patient attendance and no-show rates. By leveraging advanced analytical techniques, the project aims to uncover valuable insights that can positively impact patient engagement and resource allocation in healthcare settings.

**Overview:**

Read the raw data using pandas and create a dataframe.

Check for duplicates, null values and take necessary actions.

Appropriate data type conversion for the columns.

Creating new feature(s) as required from the given dataset.

Drop unnecessary columns from the dataframe and export the dataset as a .csv file.

Use the exported dataset to create a database and use SQL to extract insights using any SQL client (MySQL Workbench).

Used jupyter notebook to show the results as it is easy to share and visualize the code and output.

Read the patients data using 'Hospital\_patients\_datasets.csv' through pandas library and return the dataset for further analysis.

Find the counts of duplicated values are returned as a integer. This information can be useful in identifying duplicate data and deciding on appropriate strategies to deal with them, such as imputation or deletion.

Then find the counts of null values for each column in the DataFrame. This provides insights into the presence and extent of missing data in the dataset after duplicates have been dropped.

The converting\_dtype() function reads a dataset from a CSV file, converts the 'ScheduledDay' and 'AppointmentDay' columns to pandas datetime objects with date-only information, and returns the modified dataset.

The converting\_dtype() function reads a dataset from a CSV file, converts the 'ScheduledDay' and 'AppointmentDay' columns to pandas datetime objects with date-only information, and returns the modified dataset.

drops the columns 'PatientId', 'AppointmentID', and 'Neighbourhood' from the DataFrame, and returns the modified DataFrame with these columns removed.

The create\_bin() function generates age group labels and categorizes the 'Age' values into bins with 20-year intervals. It creates a new column 'Age\_group' containing the corresponding age group label for each individual and returns the modified DataFrame ds with this additional column for further analysis.

Removing the 'Age' column from the DataFrame containing age group information, and it returns the DataFrame ds without the 'Age' column, making it more focused on the categorized age groups for subsequent analysis.

the convert() function streamlines the process of converting the 'NoShow' column into a binary format, representing appointment attendance, and returns the DataFrame ds with this modified column for further analysis and modeling.

The function export\_the\_dataset() exports the cleaned DataFrame to a new CSV file named 'patients.csv'. It uses the pandas library to write the data to the CSV file.

The DataFrame obtained from the previous task is used as input and return the cleaned dataframe 'df'.

import the cleaned dataset and create the table name 'patients' that should contains the below columns.

Gender

ScheduledDay

AppointmentDay

Scholarship

Hypertension

Diabetes

Alcoholism

Handicap

SMSRecevied

NoShow

Age\_group

1. How many values are there in the given dataset?
2. Count the number of appointments for each day in the given dataset:
3. Calculate the average number of appointments (Set to nearest whole number) per day in the given dataset.
4. Find the day with the highest number of appointments in the given dataset.
5. Calculate the monthly average number of appointments in the given dataset.
6. Find the month with the highest number of appointments in the given dataset.
7. Calculate the weekly average number of appointments in the given dataset.
8. Find the week with the highest number of appointments in the given dataset.
9. What is the distribution of appointments based on gender in the dataset?
10. Calculate the number of appointments per weekday in the given dataset.
11. Calculate the average time between scheduling and the appointment day in the given dataset.