Professional Program on Data Analyst

Exercise: Handling Missing Data and Analysing the Data with Python

Objective:

● Handle missing data and analyse a dataset using Python and Pandas on Kaggle.

Tasks:

1. Load the dataset from the CSV file.
2. Check for missing values in the dataset.
3. Handle missing data by filling or dropping rows/columns.
4. Verify that the missing values are handled properly.
5. Analyse the data to find specific insights.

How to Use Kaggle

Kaggle is an online community for data scientists and machine learning practitioners. It provides a platform for users to find and publish datasets, explore and build models in a web-based data science environment, and participate in competitions. Here’s how to use Kaggle for this exercise:

1. **Create a Kaggle Account:**
   - Go to Kaggle and create an account if you don't have one.
2. **Upload Your Dataset:**
   - Navigate to the "Datasets" section and upload your dataset (`housing_data.csv`).
3. **Create a New Notebook:**
   - Go to the "Code" section, create a new notebook, and choose Python as the language.
4. **Write and Execute Your Code:**
   - Use the provided code template to handle missing data and analyze the dataset.

Task Details

Part 1: Handling Missing Data

1. Load the dataset from the CSV file you created (`housing_data.csv`).

```
import pandas as pd

# Step 1: Load the dataset
df = pd.read_csv('/kaggle/input/housing_data.csv')
```

2. Check for missing values in the dataset.

```
# Step 2: Check for missing values
print("Missing values in the dataset:")
print(df.isnull().sum())
```

3. Handle missing data by filling missing values or dropping rows/columns:

- For `Price` and `Bedrooms`, fill missing values with the **mean** of the respective columns.

```
# Step 3: Handle missing data
# Fill missing 'Price' with mean of the column
df['Price'] = df['Price'].fillna(df['Price'].mean())

# Fill missing 'Bedrooms' with mean of the column
df['Bedrooms'] = df['Bedrooms'].fillna(df['Bedrooms'].mean())
```

4. Verify that the missing values are handled properly.

```
# Verify if missing values are handled
print("\\\\nDataset after handling missing values:")
print(df.isnull().sum())
```

Part 2: Analyse the Data

1. Find the average price of all houses.

```
# Part 2: Data Analysis

# 1. Find the average price
average_price = df['Price'].mean()
print(f"\\\\nAverage price of houses: ${average_price:,.2f}")
```

2. Find the house with the highest price and the house with the lowest price.

```
# 2. Find the house with the highest price
max_price_house = df.loc[df['Price'].idxmax()]
print(f"\\\\nHouse with the highest price:\\\\n{max_price_house}")
```

```
# 3. Find the house with the lowest price
min_price_house = df.loc[df['Price'].idxmin()]
print(f"\\\\nHouse with the lowest price:\\\\n{min_price_house}")
```

3. Filter houses with a price greater than 600,000 and display their details.

```
# 4. Filter houses with price greater than 600,000
filtered_houses = df[df['Price'] > 600000]
print("\\\\nHouses with price greater than 600,000:")
print(filtered_houses)
```

Summary

In this exercise, you learned how to:

1. Load and inspect a dataset using Pandas.
2. Identify and handle missing data by filling or dropping rows/columns.
3. Verify that missing values are properly handled.
4. Perform basic data analysis to find average prices and filter data based on specific criteria.

Dataset for Practice: Download the sample dataset to practise these tasks and replicate the exercise.