**House Price Prediction Data Pre-processing**

Creating a document for pre-processing data for a house price prediction project involves several essential steps. In this document, I'll outline the key steps, provide explanations, and even include some sample code using Python and popular libraries like Pandas and Scikit-Learn. Before starting, ensure you have the necessary data and tools installed, including Python, Pandas, and Scikit-Learn.

**Table of Contents**

**Introduction**

**Data Exploration**

**Data Cleaning**

**Handling Missing Data**

**Data Transformation**

**Feature Engineering**

**1. Introduction**

The goal of this document is to guide you through the process of preparing your data for a house price prediction project. Pre-processing is a crucial step as it ensures the data is in the right format and quality for training machine learning models.

**2. Data Exploration**

Before diving into pre-processing, it's essential to understand your dataset. Here are some common tasks during data exploration:

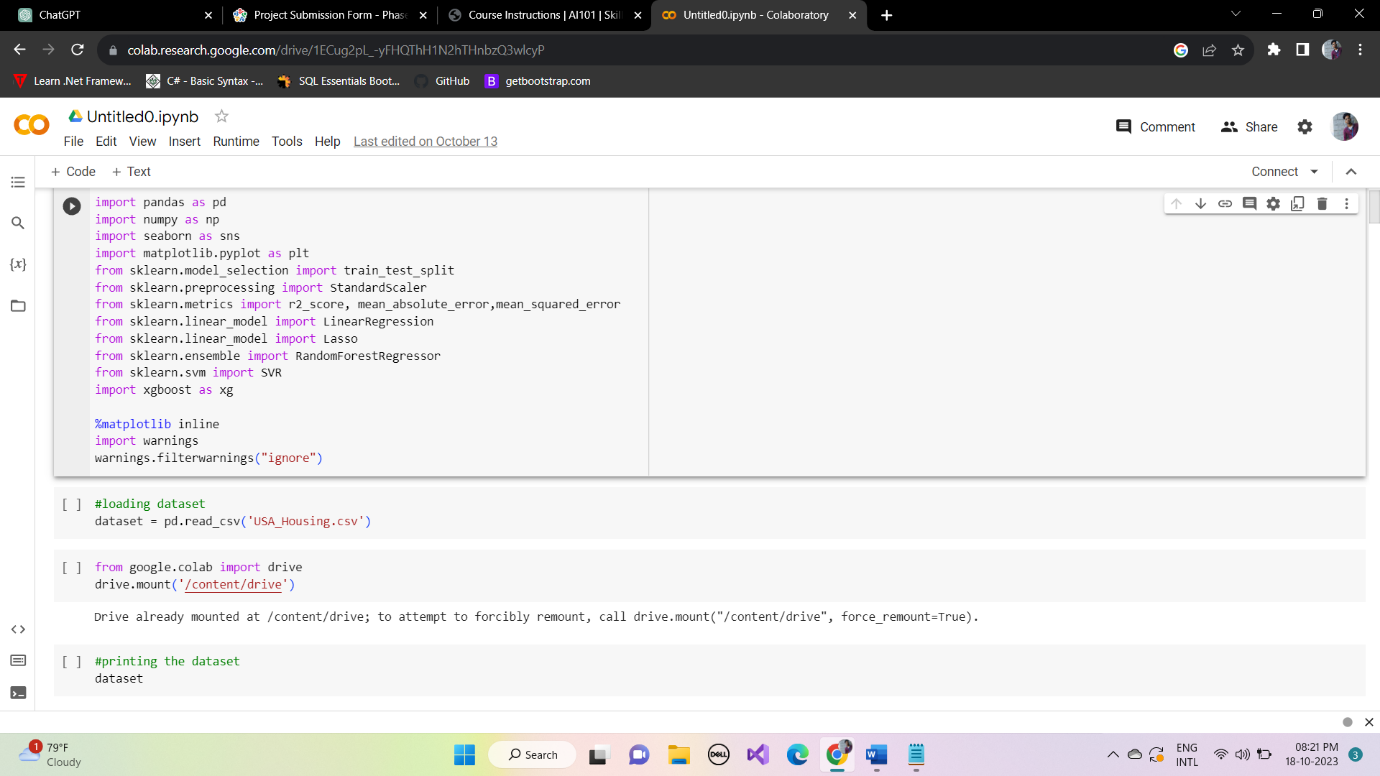
Load the Data: Read your dataset into a Pandas DataFrame.

**import pandas as pd**

**data = pd.read\_csv('house\_price\_data.csv')**

**Data Summary: Get an overview of the data with functions like data.head(), data.info(), and data.describe().**

Visualizations: Create plots and graphs to understand the data's distribution, correlations, and outliers. Libraries like Matplotlib and Seaborn are helpful.



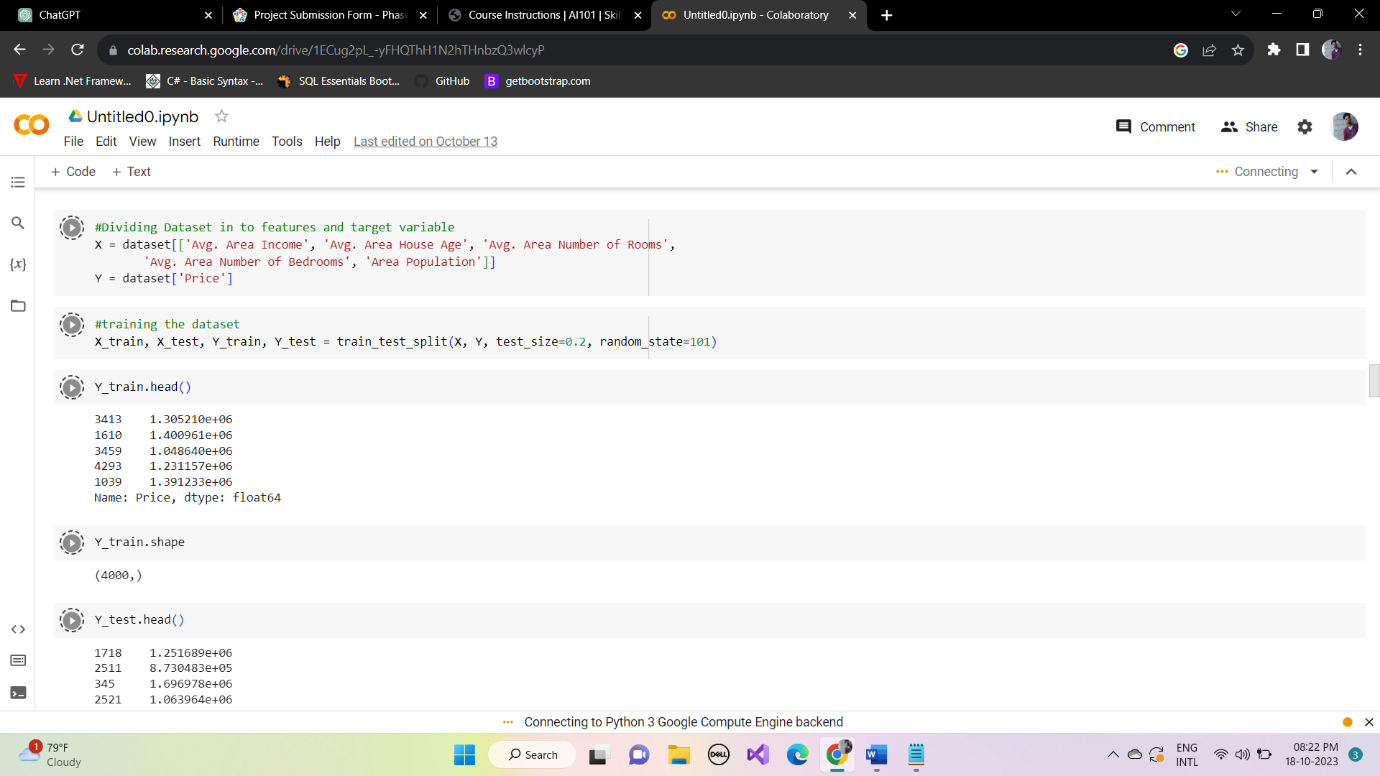
**3. Data Cleaning**

Data cleaning involves addressing issues such as duplicates and inconsistencies.

Remove Duplicates: Use data.drop\_duplicates() to remove duplicate rows.

**data = data.drop\_duplicates()**

Correct Data Types: Ensure that data types (integers, floats, strings) match the data they represent.



**4. Handling Missing Data**

Handling missing data is crucial to prevent issues during model training.

Identify Missing Data: Use data.isnull().sum() to count missing values in each column.

Impute Missing Data: You can fill missing values with a specific value or strategy (e.g., mean, median, mode).

**data['column\_name'].fillna(data['column\_name'].mean(), inplace=True)**

**5. Data Transformation**

Prepare the data for machine learning by encoding categorical variables and scaling features.

Categorical Encoding: Convert categorical variables into numerical form using techniques like one-hot encoding.

**data = pd.get\_dummies(data, columns=['categorical\_column'], drop\_first=True)**

Feature Scaling: Scale numerical features if needed. Common methods include Min-Max scaling and Standardization.

**from sklearn.preprocessing import StandardScaler**

**scaler = StandardScaler()**

**data['numerical\_column'] = scaler.fit\_transform(data['numerical\_column'].values.reshape(-1, 1))**

**6. Feature Engineering**

Feature engineering involves creating new features or transforming existing ones to improve model performance.

Create New Features: Consider creating features like age of the house, the ratio of bedrooms to bathrooms, or a total square footage feature.

Feature Selection: Use techniques like correlation analysis and feature importance to select the most relevant features.

Your data should now be cleaned and pre-processed, ready for building a house price prediction model. Remember to split your data into training and testing sets and choose an appropriate machine learning algorithm.

This document provides an overview of the pre-processing steps. The specific tasks may vary based on your dataset and project requirements. It's crucial to continually assess and refine your pre-processing steps to improve your model's performance.