In any Machine Learning model, the first crucial step is to truly understand the data we're working with. In this project, we'll be using the california housing dataset and we'll be working on Google Colab, a fantastic platform for data analysis and machine learning in a collaborative environment.

Setting Up Google Colab

Access Google Colab:

- Open your web browser and navigate to Google Colab.
- Sign in with your Google account.

Create a New Notebook:

- Click on "New" to create a new notebook.
- Rename your notebook to something descriptive like "Predicting House Prices."

1. Importing Libraries and Loading Data

Since Google Colab comes pre-installed with essential libraries, we can skip the installation step. However, if you are using a different platform, make sure to install the required libraries using commands like **pip install numpy**, **pip install pandas** and so on.

Now, let's import the necessary libraries and load our dataset, using the following code:

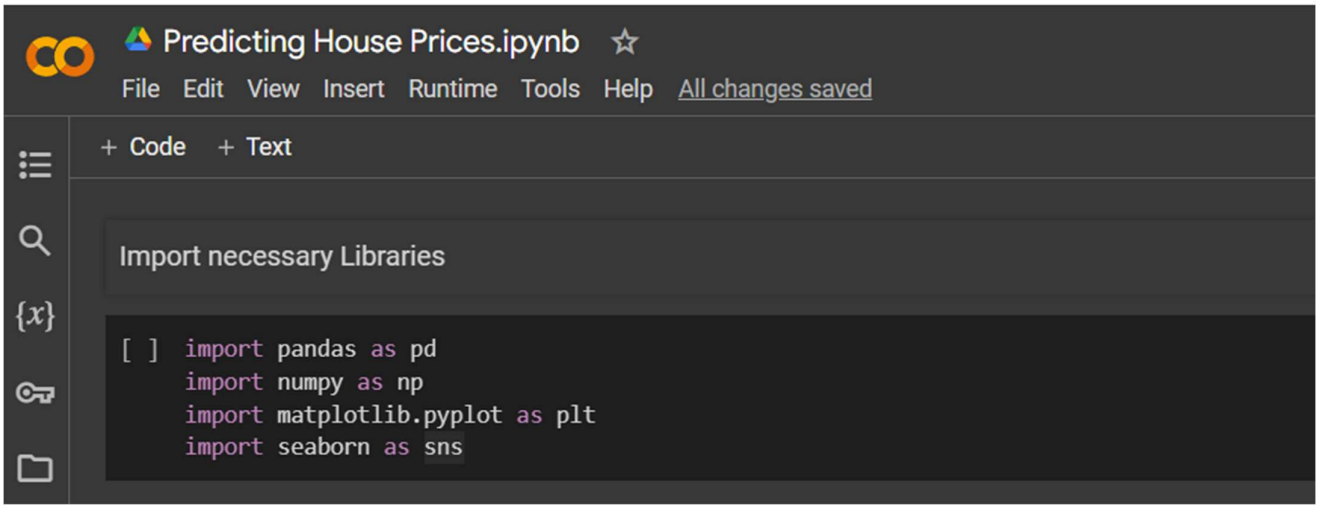
```
import pandas as pd

import numpy as np

import matplotlib.pyplot as plt

import seaborn as sns
```

It will look like this in Google Colab:



Importing Libraries

2. Load the Dataset

Now the step is to load the data, in our project we'll be using the california housing dataset so we'll load the data in a variable 'housing' using scikit-learn library and check the type using 'type()' function as follows:

```
Load the Dataset California housing

[2] from sklearn.datasets import fetch_california_housing
    housing = fetch_california_housing()

[3] type(housing)

    sklearn.utils._bunch.Bunch
```

Loading the Dataset

3. Checking type of dataset

The **sklearn.utils._bunch.Bunch** is a data structure commonly used in scikit-learn to represent datasets. It is essentially a dictionary-like object that contains key-value pairs. Each key corresponds to a dataset attribute and the values associated with each key represent the data for that attribute. The dataset will appear like this:

```
[4] housing

{'data': array([[   8.3252    ,   41.        ,    6.98412698, ...,    2.5
          37.88      , -122.23      ],
        [   8.3014    ,   21.        ,    6.23813708, ...,    2.10984183,
          37.86      , -122.22      ],
        [   7.2574    ,   52.        ,    8.28813559, ...,    2.80225989,
          37.85      , -122.24      ],
        ...,
        [   1.7       ,   17.        ,    5.20554273, ...,    2.3256351 ,
          39.43      , -121.22      ],
        [   1.8672    ,   18.        ,    5.32951289, ...,    2.12320917,
          39.43      , -121.32      ],
        [   2.3886    ,   16.        ,    5.25471698, ...,    2.61698113,
          39.37      , -121.24      ]]),
 'target': array([4.526, 3.585, 3.521, ..., 0.923, 0.847, 0.894]),
 'frame': None,
 'target_names': ['MedHouseVal'],
 'feature_names': ['MedInc',
  'HouseAge',
  'AveRooms',
  'AveBedrms',
  'Population',
```

Checking the type of Dataset

4. Getting information about a value of the key

```
[5]  print(housing.DESCR)
```

```
.. _california_housing_dataset:

California Housing dataset
--------------------------

**Data Set Characteristics:**

    :Number of Instances: 20640

    :Number of Attributes: 8 numeric, predictive attributes and the target

    :Attribute Information:
        - MedInc        median income in block group
        - HouseAge      median house age in block group
        - AveRooms      average number of rooms per household
        - AveBedrms     average number of bedrooms per household
        - Population    block group population
        - AveOccup      average number of household members
        - Latitude      block group latitude
        - Longitude     block group longitude
```

The 'print(housing.DESCR)' statement outputs a detailed description of the California Housing dataset. It provides essential information about the dataset's characteristics, including the number of instances, attributes and a list of the numeric, predictive attributes along with their meanings.

5. Getting predictive attributes of the dataset

```
[6]  print(housing.feature_names)
```

```
['MedInc', 'HouseAge', 'AveRooms', 'AveBedrms', 'Population', 'AveOccup',
```

This statement would print the names of the features (predictive attributes) in the California Housing dataset, providing a quick reference to the variables included in the dataset.

6. Getting target values from the dataset

The print(housing.target) statement would print the target values of the California Housing dataset. 'housing.target' refers to the target variable, which is typically the variable we want to predict. It contains the actual values that we are trying to model or predict.

```
[7]  print(housing.target)
```

```
[4.526 3.585 3.521 ... 0.923 0.847 0.894]
```