

Phase 3 : Project Development Part 1

Project Title:

Machine learning model deployment with IBM cloud Watson Studio.

Problem Statement:

Become a wizard of predictive analytics with IBM Cloud Watson Studio. Train machine learning models to predict the outcomes in real time. Deploy the models as web services and integrate them into your applications. Unlock the magic of data driven insights and make informed decisions like never before.

House Price Prediction Analysis - Part 1

Project overview

Data Understanding

Data Visualization

Data Preparation

Modeling

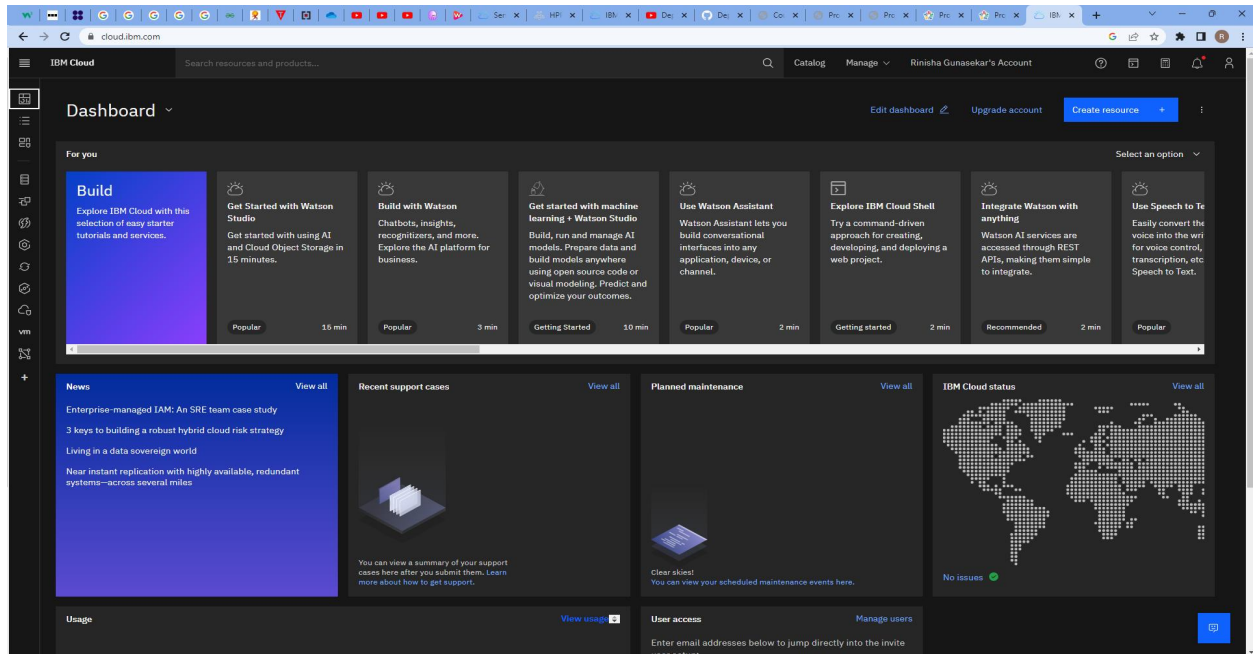
Evaluation

Project Overview:

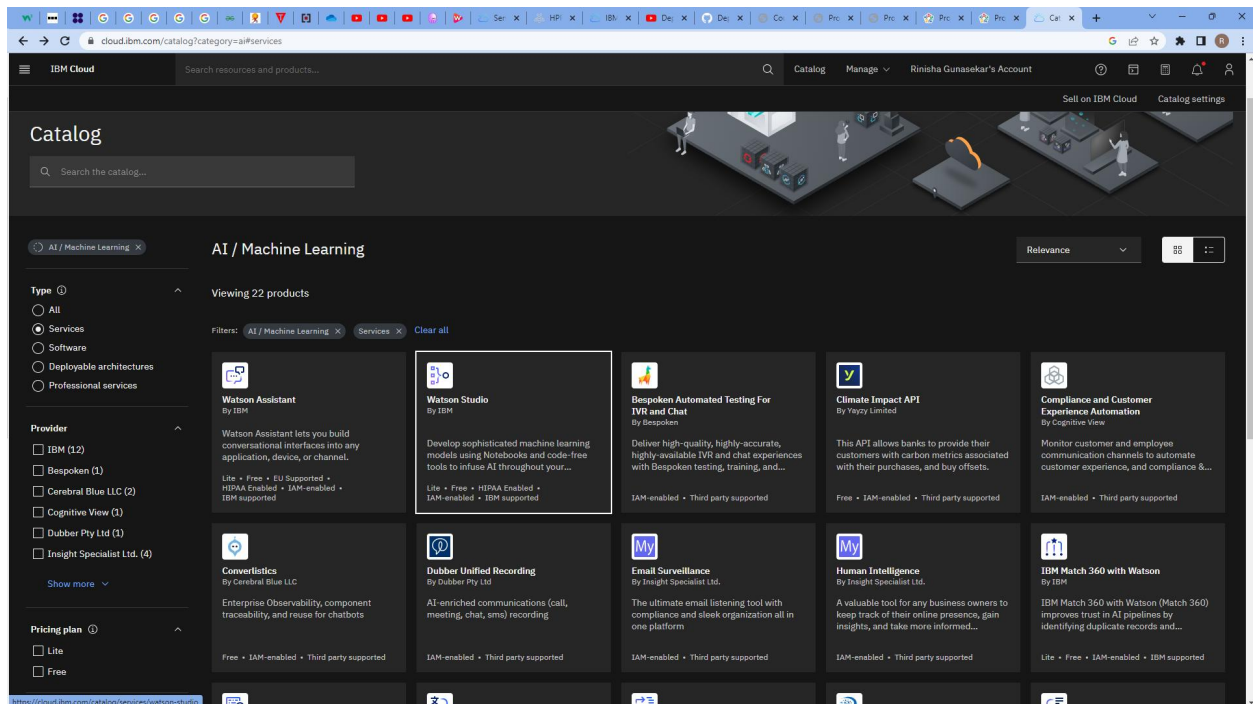
House Price Prediction Analysis aims to use Machine learning analysis algorithms to predict the price of houses based on their features like number of rooms, number of bedrooms, age of the house, population of the respective area where the house is located, location of the house and the area income with other relevant factors if available. By this Machine Learning model user can predict the price of the house that can be sold.

Step wise process for the House Price Prediction Analysis Machine learning model deployment.

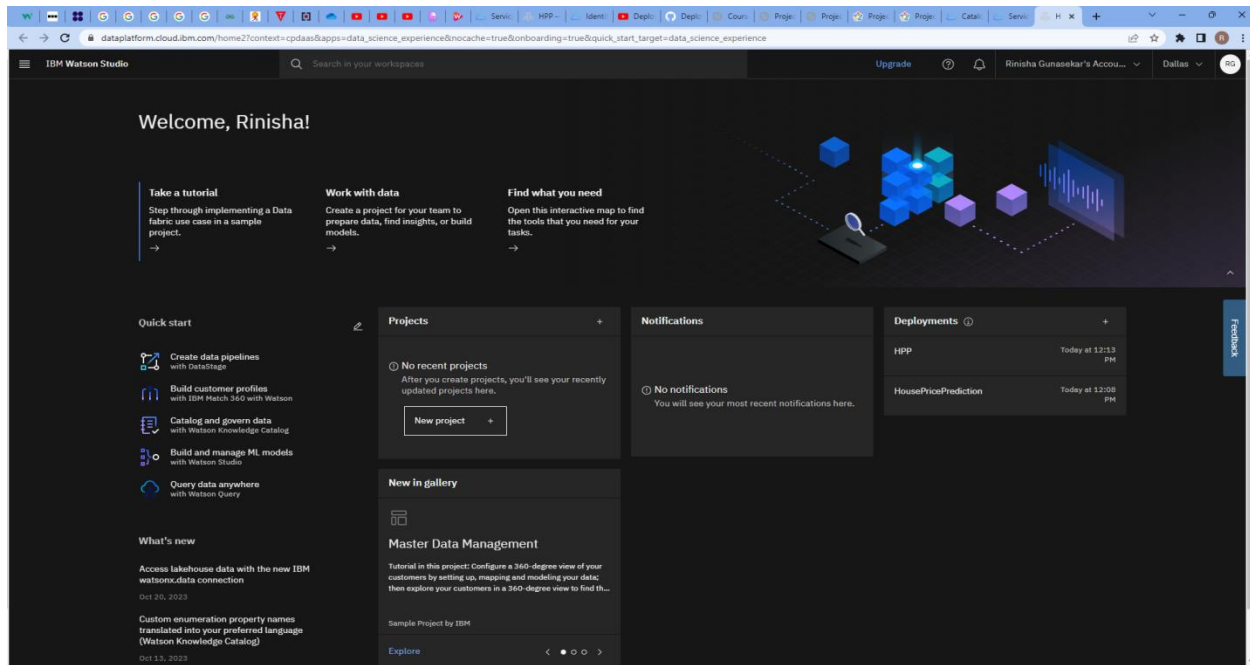
Step 1: Login to IBM cloud



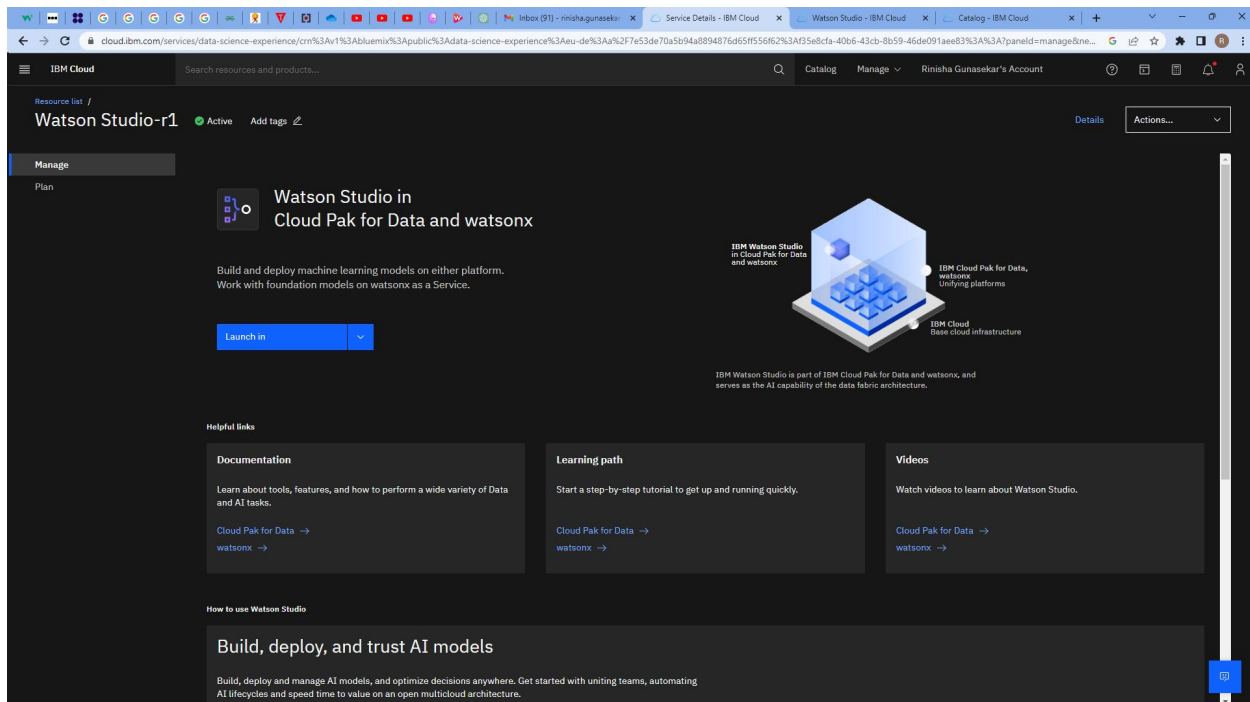
Step 2: Go to catalog and create a Watson Studio service in AI category.



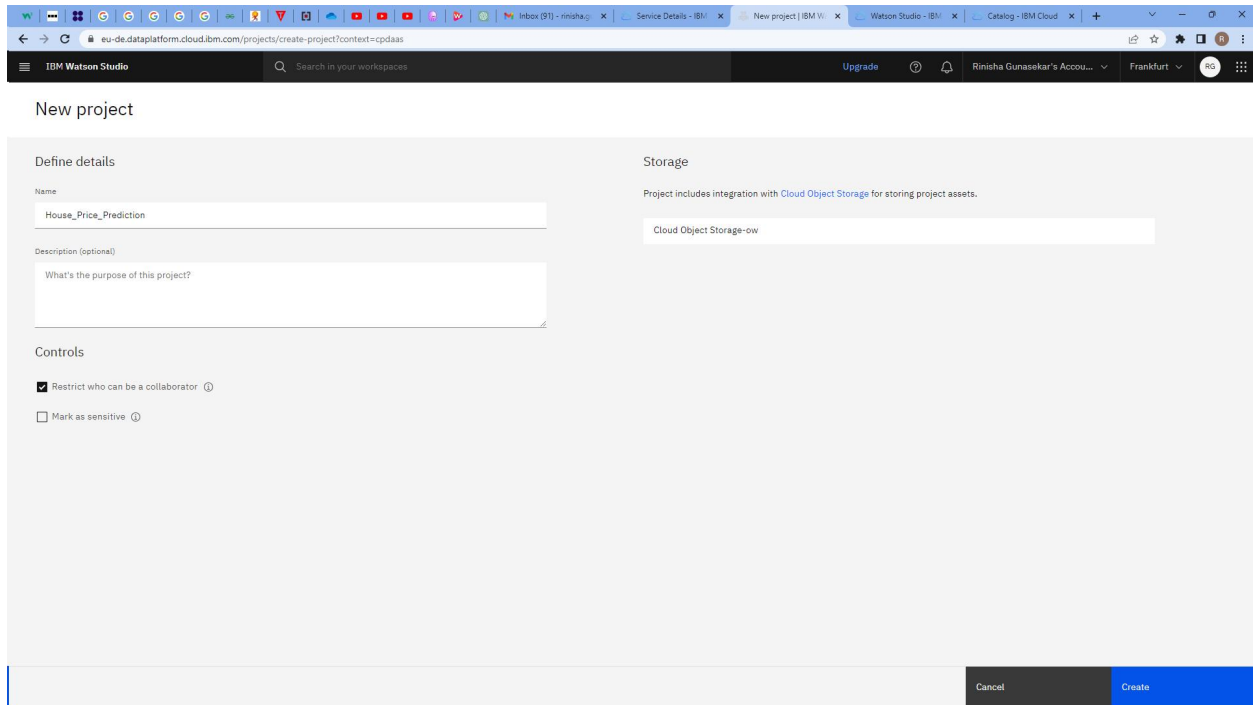
Step 3: Get started to launch Watson Studio Dashboard.



Step 4: Launch in the Watson Studio in Cloud Park for Data and watsonx



Step 5: Start a New project



The screenshot shows the 'New project' form in IBM Watson Studio. The form is divided into two main sections: 'Define details' and 'Storage'. In the 'Define details' section, the 'Name' field is filled with 'House_Price_Prediction'. The 'Description (optional)' field is empty. In the 'Storage' section, the 'Cloud Object Storage-ow' field is filled. Below these sections, there are 'Controls' with two checkboxes: 'Restrict who can be a collaborator' (checked) and 'Mark as sensitive' (unchecked). At the bottom right, there are 'Cancel' and 'Create' buttons.

Define details

Name
House_Price_Prediction

Description (optional)
What's the purpose of this project?

Storage

Project includes integration with [Cloud Object Storage](#) for storing project assets.

Cloud Object Storage-ow

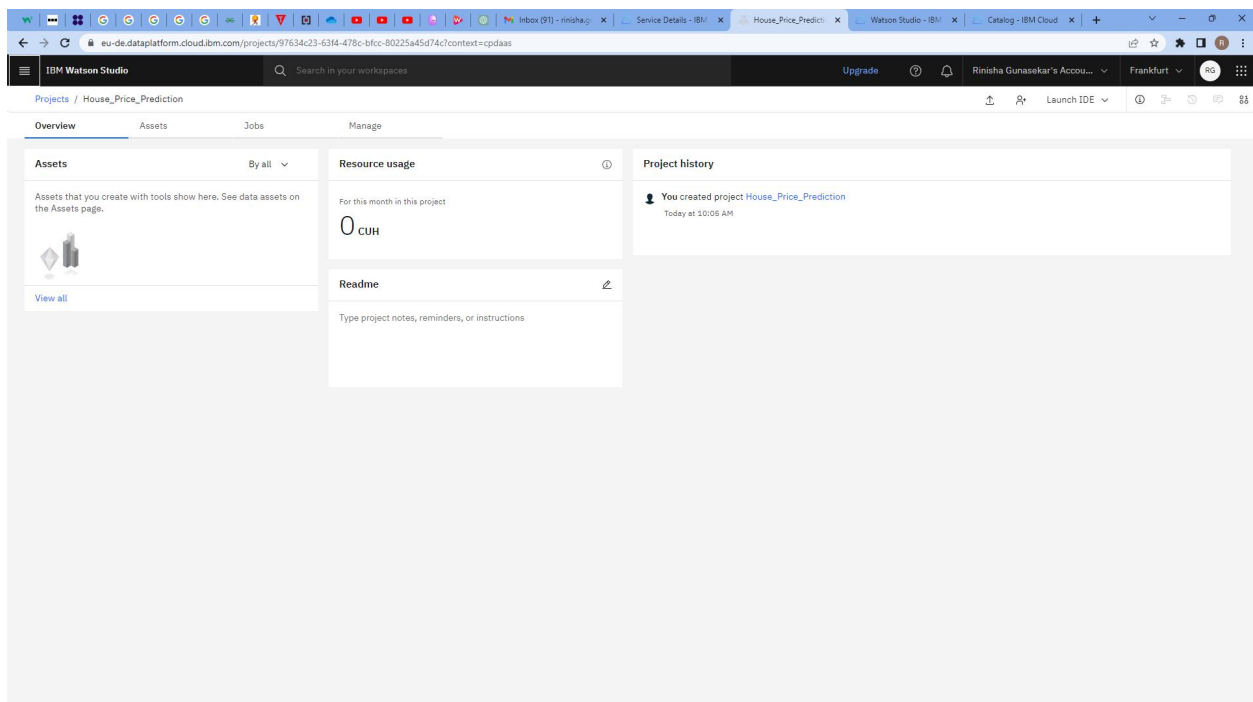
Controls

☒ Restrict who can be a collaborator ⓘ

☐ Mark as sensitive ⓘ

Cancel Create

Step 6: Create a project named *House_Price_Prediction*



The screenshot shows the project overview page for 'House_Price_Prediction' in IBM Watson Studio. The page has a navigation bar with 'Overview', 'Assets', 'Jobs', and 'Manage'. The 'Overview' tab is selected. The main content area is divided into four sections: 'Assets', 'Resource usage', 'Project history', and 'Readme'. The 'Assets' section shows a bar chart and a 'View all' link. The 'Resource usage' section shows a gauge for 'CUH' and a 'Readme' link. The 'Project history' section shows a message: 'You created project House_Price_Prediction Today at 10:05 AM'. The 'Readme' section is empty.

Projects / House_Price_Prediction

Overview Assets Jobs Manage

Assets By all

Assets that you create with tools show here. See data assets on the Assets page.

Resource usage ⓘ

For this month in this project

0 CUH

Project history

You created project House_Price_Prediction Today at 10:05 AM

Readme ⓘ

Type project notes, reminders, or instructions

Step 7: Import all assets

The screenshot displays the IBM Watson Studio web interface. The browser address bar shows the URL: eu-de.dataplatform.cloud.ibm.com/projects/97634c23-63f4-478c-bfcc-80225a45d74c/assets?context=cpdaas#. The interface has a dark header with the IBM Watson Studio logo and a search bar. Below the header, there are tabs for 'Overview', 'Assets', 'Jobs', and 'Manage'. The 'Assets' tab is active, showing a list of assets. On the left, there is a sidebar with '1 assets' and 'Asset types' (Data). The main area shows a table with columns 'Name' and 'Last modified'. One asset is listed: 'USA_Housing.csv' with a 'Now Modified by you' timestamp. On the right, there is a 'Data in this project' panel with a drop zone for uploading files.

IBM Watson Studio

Search in your workspace

Upgrade

Rinisha Gunasekar's Account

Frankfurt

Projects / House_Price_Prediction

Overview Assets Jobs Manage

Find assets

Import assets

New asset

1 assets

All assets

Name	Last modified
USA_Housing.csv	Now Modified by you

Asset types

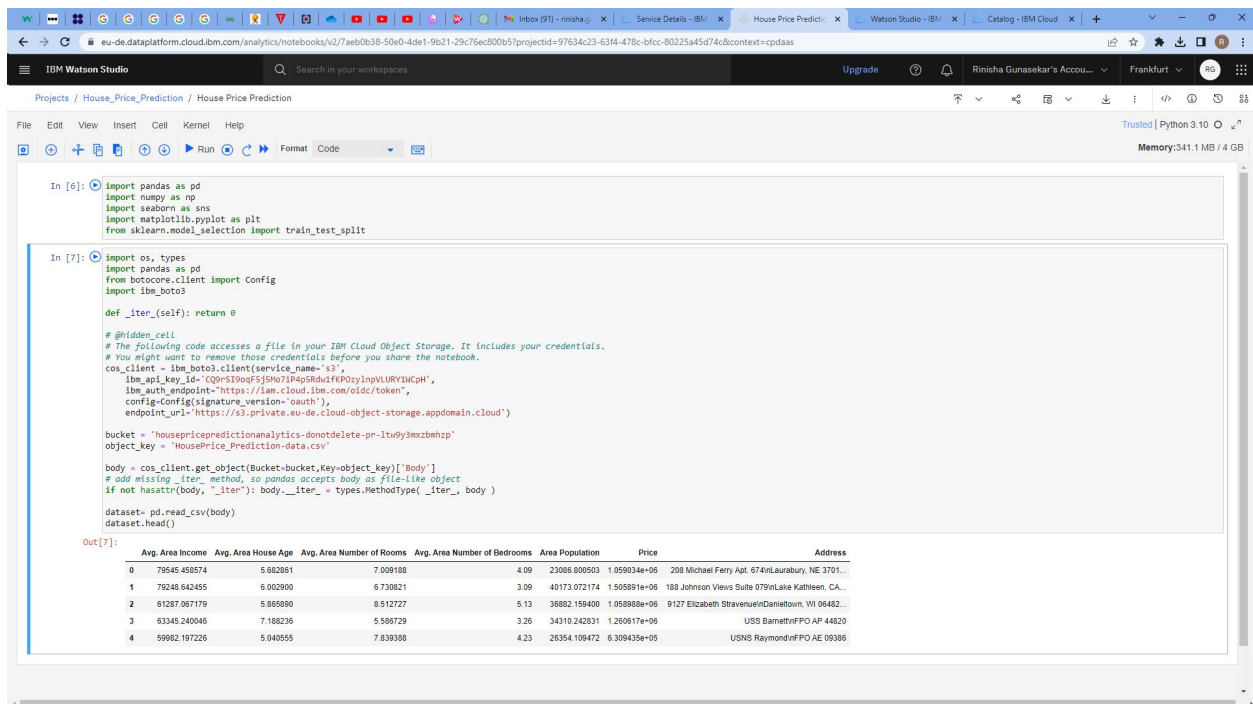
Data

Data in this project

Drop data files here or browse for files to upload

Step 8: Add a jupyter notebook instance in your project to Develop and Deploy Machine Learning Model.

Import necessary library packages.



The screenshot shows the IBM Watson Studio Jupyter Notebook interface. The browser address bar indicates the URL: `eu-de.dataplatform.cloud.ibm.com/analytics/notebooks/v2/7aeb0b38-50e0-4de1-9b21-29c76ec800b5?projectId=97634c23-6314-478c-bfcc-8022545d74c&context=cpcdaas`. The notebook title is "House Price Prediction". The interface includes a menu bar (File, Edit, View, Insert, Cell, Kernel, Help) and a toolbar with icons for file operations, running cells, and code formatting. The code editor shows two input cells. The first cell (In [6]) imports necessary libraries: `pandas`, `numpy`, `seaborn`, `matplotlib.pyplot`, and `sklearn.model_selection`. The second cell (In [7]) imports `os`, `types`, `pandas`, and `boto3`, then defines a function `_iter` to iterate over data. It also shows the configuration of `boto3` with IBM Cloud Object Storage credentials and the retrieval of a CSV file from the bucket. The output of the second cell (Out [7]) displays the first five rows of the dataset as a table.

```
In [6]: import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split

In [7]: import os, types
import pandas as pd
from boto3.client import Config
import boto3

def _iter(self): return 0

#@hidden_cell
# The following code accesses a file in your IBM Cloud Object Storage. It includes your credentials.
# You might want to remove those credentials before you share the notebook.
cos_client = boto3.client(service_name='s3',
    ibm_api_key_id='C0QnS10ogF5j9u7IP4p5Rdu1fK0zyInpVLURy1uCPH',
    ibm_auth_endpoint='https://iam.cloud.ibm.com/oidc/token',
    config=Config(signature_version='sauth'),
    endpoint_url='https://s3.private.eu-de.cloud-object-storage.appdomain.cloud')

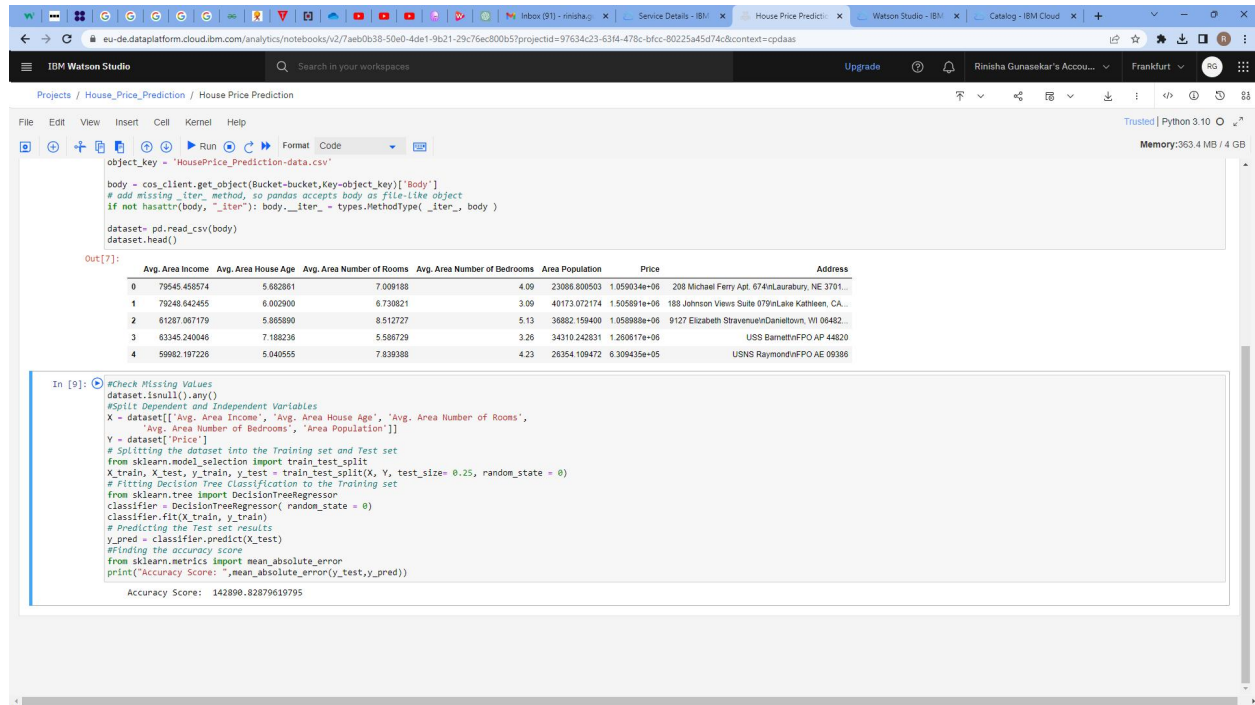
bucket = 'housepricepredictionanalytics-donotdelete-pr-1tv0y3mxzbhnp'
object_key = 'HousePrice_Prediction-data.csv'

body = cos_client.get_object(Bucket=bucket,Key=object_key)['Body']
# add missing _iter_ method, so pandas accept: body as file-like object
if not hasattr(body, "_iter"): body._iter_ = types.MethodType(_iter_, body)

dataset= pd.read_csv(body)
dataset.head()
```

	Avg. Area Income	Avg. Area House Age	Avg. Area Number of Rooms	Avg. Area Number of Bedrooms	Area Population	Price	Address
0	79545.458574	5.682961	7.009188	4.09	23086.800503	1.059034e+06	208 Michael Ferry Apt. 674nLaurabury, NE 3701...
1	79248.942455	6.002900	6.730821	3.09	40173.072174	1.505891e+06	188 Johnson Views Suite 079nLake Kathleen, CA...
2	61287.967179	5.865890	8.512727	5.13	35882.158400	1.058988e+06	9127 Elizabeth StravenueinDanieletown, WI 06482...
3	63345.240046	7.188236	5.588729	3.26	34310.242831	1.260617e+06	USS BarnettinFPO AP 44820
4	59882.197226	5.040555	7.838388	4.23	26354.109472	6.309435e+05	USNS RaymondinFPO AE 09306

Step 9: Import dataset and proceed further with pre-processing steps and build the model.



The screenshot displays the IBM Watson Studio interface. The top navigation bar includes the IBM Watson Studio logo, a search bar, and user information. The main workspace shows a Jupyter notebook titled "House Price Prediction". The notebook contains two code cells. The first cell imports the dataset from a bucket and displays the first five rows. The second cell performs data preprocessing, including splitting the dataset into training and testing sets, fitting a Decision Tree Classifier, and evaluating the model's accuracy.

```
object_key = "HousePrice-Prediction-data.csv"

body = cos_client.get_object(Bucket=bucket,Key=object_key)['body']
# add missing _iter_ method, so pandas accepts body as file-like object
if not hasattr(body, "_iter_"): body.__iter__ = types.MethodType(_iter_, body)

dataset = pd.read_csv(body)
dataset.head()
```

Out[7]:

	Avg. Area Income	Avg. Area House Age	Avg. Area Number of Rooms	Avg. Area Number of Bedrooms	Area Population	Price	Address
0	79545.458574	5.682861	7.009188	4.09	23086.800503	1.059034e+06	208 Michael Ferry Apt. 674InLaurabury, NE 3701...
1	79248.642455	6.002900	6.730821	3.09	49173.072174	1.505891e+06	188 Johnson Views Suite 079nLake Kathleen, CA...
2	61287.067179	5.865890	8.512727	5.13	36882.159400	1.058988e+06	9127 Elizabeth StravenueinDanieltown, WI 06482...
3	63345.240046	7.188236	5.586729	3.26	34319.242831	1.260617e+06	USS BarnettnFPO AP 44820
4	59082.197226	5.040555	7.839388	4.23	26354.109472	6.309435e+05	USNS RaymondnFPO AE 09386

```
In [9]: #Check Missing Values
dataset.isnull().any()

#Split Dependent and Independent Variables
X = dataset[['Avg. Area Income', 'Avg. Area House Age', 'Avg. Area Number of Rooms',
             'Avg. Area Number of Bedrooms', 'Area Population']]
Y = dataset['Price']

# Splitting the dataset into the Training set and Test set
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, Y, test_size= 0.25, random_state = 0)

# Fitting Decision Tree Classification to the Training set
from sklearn.tree import DecisionTreeRegressor
classifier = DecisionTreeRegressor( random_state = 0)
classifier.fit(X_train, y_train)

# Predicting the Test set results
y_pred = classifier.predict(X_test)

#Finding the accuracy score
from sklearn.metrics import mean_absolute_error
print("Accuracy Score: ",mean_absolute_error(y_test,y_pred))

Accuracy Score: 142898.82879619795
```

Hence the model was build, trained and tested