Cloud Application Development

Phase 5: Project Documentation & Submission

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 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!

Problem definition:

The project involves training a machine learning model using IBM Cloud Watson Studio and deploying it as a web service. The goal is to become proficient in predictive analytics by creating a model that can predict outcomes in real-time. The project encompasses defining the predictive use case, selecting a suitable dataset, training a machine learning model, deploying the model as a web service, and integrating it into applications.

Project Objective:

A cloud-based predictive analytics solution has been designed using IBM Cloud Watson Studio that empowers users to train, deploy, and integrate machine learning models into their applications. The goal is to harness the power of data-driven insights, enabling users to make informed decisions in real-time and unlocking the magic of predictive analytics.

Design Thinking Process:

Design thinking is a human-centered approach to solving problems and creating innovative solutions. To address the problem statement of becoming a wizard of predictive analytics with IBM Cloud Watson Studio, you can follow a design thinking process with the following steps:

***Empathize*:**Understand the target users or audience who want to become wizards of predictive analytics.

Define:Clearly define the problem you're solving, considering the users' perspective.

***Ideate*:**Brainstorm creative solutions to address the defined problem.

***Prototype*:**Create a prototype or mockup of the cloud application that will help users become predictive analytics wizards.

Test:Gather feedback from potential users by allowing them to interact with the prototype.

Implement:Develop the cloud application based on the refined prototype.

Deploy:Deploy the application on a cloud platform, such as IBM Cloud, for accessibility and scalability.

***Iterate*:**Continuously gather user feedback and make improvements to the application.

Predictive use case :

Description:

PMML is an XML based language used to represent predictive models created as the result of the predictive modeling process. It allows for predictive models to be easily shared between applications.

Key Components:

- Data Collection
- Data Preprocessing
- Model Selection
- Model Training
- Validation and Evaluation
- Fine-tuning

Datasetselection: (“house-price-prediction”)

Data collection for a dataset with labels is a critical step in building a machine learning model for tasks like classification or prediction

The "House Price Prediction" dataset contains information about residential properties and their sale prices. It's used to build predictive models for estimating house prices, with features like bedrooms, bathrooms, and square footage. Data scientists use it for regression practice, and it has applications in real estate, finance, and urban planning. Various versions of this dataset may focus on specific regions or property types.

Model Training:

The decision tree algorithm is a machine learning method used for "House Price Prediction" dataset in Scikit-Learn (sklearn). It works by recursively splitting data based on features to create a tree structure that predicts house prices. Scikit-Learn's implementation allows users to easily build, train, and evaluate decision tree models for predicting house prices from dataset features.

Model Deployment and Integration:

Deploying a trained decision tree model as a web service on IBM Cloud Watson Studio typically involves the following steps:

1. Prepare the Model
2. Create a Watson Machine Learning (WML) Instance
3. Package Dependencies
4. Deployment Configuration
5. Deploy the Model
6. Endpoint Testing

The deployed model has to be integrated into web applications or other systems that need to use the object detection service. We can make HTTP requests to the endpoint to get predictions for new images.

Development Phase:

Step 1: Set Up IBM Watson Studio

1. Create an IBM Cloud Account:

If we don't already have an IBM Cloud account, we'll need to create one. Visit the IBM Cloud website (https://cloud.ibm.com) and sign up for a free account.

2. Access IBM Watson Studio: After creating our IBM Cloud account, log in to IBM Cloud

3. Create a New Project

i. Once logged in, navigate to the IBM Watson Studio service.

ii. Create a new project within Watson Studio to organize your work. Give your project a name and, if applicable, a description.

4. Define the Project Type
5. Configure the Project:
6. Invite Collaborators
7. Data Import

Step 2:Data Collection and Preparation:
Data Collection:

• Identify Data Sources: Determine where our data will come
from. It can be internal databases, external sources, or a
combination of both. Identify the specific data needed to address
your predictive analytics problem.

• Data Retrieval: Obtain the data from the identified sources.
Depending on the source, we may need to use SQL queries, web
scraping, APIs, or other methods to retrieve the data.

• Data Import: Import the collected data into your IBM Watson
Studio project. We can typically do this via the project interface
or using programming languages like Python or R.

Data Preparation:

Data Inspection:

Explore the imported data to understand its structure. Use basic commands and tools to check the data's dimensions, data types, and an initial sample of records.

Data Cleaning:

Address missing data: Identify and handle missing values using techniques like imputation, removal, or interpolation.

Remove duplicates: Check for and eliminate duplicate records from the dataset.

Outlier detection and handling: Identify outliers and decide whether to remove, transform, or keep them based on their impact on the analysis.

Data normalization:

If your data contains features with different scales, apply scaling techniques like Min-Max scaling or standardization.

Data Splitting:

Split our data into training, validation, and test sets. The training set is used for model training, the validation set for model tuning, and the test set for final model evaluation.

Data Preprocessing:

Apply any further data preprocessing steps such as standardization, dimensionality reduction, or other techniques that can improve the model's performance.

Data Export:

Save the cleaned and pre-processed data in a format that is easy to work with for building machine learning models. Common formats include CSV, Excel, or database tables.The quality and preparation of our data play a significant role in the success of our predictive analytics project. By completing this step effectively, we ensure that your data is ready for model development and that the models can learn meaningful patterns and make accurate predictions.

Step 3: Model Building:

1)Select Appropriate Algorithms:

Choose machine learning algorithms that are suitable for our specific predictive analytics problem. Common choices include linear regression, decision trees, random forests, support vector machines, or neural networks.

2)Data Preparation:

Ensure our data is prepared and cleaned, as discussed in the earlier steps. This involves handling missing values, encoding categorical data, and scaling or normalizing features.

3)Split Data:

Divide our dataset into three sets: a training set, a validation set, and a test set. The training set is used to train the model, the validation set is for tuning hyperparameters, and the test set is for final model evaluation.

4)Model Training:

Train our chosen machine learning

model on the training data. Watson Studio provides an integrated environment to do this using popular libraries like scikit-learn or TensorFlow.

Step 4: Model Deployment :

Model deployment is a critical step in our project for becoming proficient in predictive analytics with IBM Cloud Watson Studio. This step involves making your trained machine learning models accessible to applications through web services. Here's a guide on how to deploy your models:

1. Select the Model:

Choose the best-performing machine learning model that you want to deploy. This should be the model that we have thoroughly trained and evaluated.

2. Preprocessing and Transformation:

Ensure that any preprocessing steps (e.g., feature scaling or one-hot encoding) used during model training are replicated during deployment. This ensures that incoming data to the deployed model is processed correctly.

3. Create a Deployment Space:
 - In IBM Watson Studio, navigate to the model deployment section.
 - Create a deployment space or project where you can manage your deployment assets.

4.Model Deployment:
 - In the deployment space, select the model we want to deploy.
 - Follow the prompts to deploy the model. We may be asked to specify the deployment environment and configuration settings. Watson Studio makes this process user-friendly.

5. API Endpoint:
 Once deployed, our model will have a unique API endpoint that can be used to make predictions. This endpoint is typically a URL that accepts input data and returns predictions.

6. Scalability and Load Balancing:

Depending on our use case, consider setting up load balancing and ensuring the deployed model can handle a scalable number of requests.

7. Security and Access Control:

Implement appropriate security measures to protect our API endpoint. Ensure that only authorized applications or users can access and use the model.

8. Documentation:

Create documentation for the API endpoint. Include details about the expected input format, response format, andany required authentication or API keys.

9. Testing:

Before integrating the model into applications, test the API endpoint to ensure it's functioning as expected. Use sample data to validate that it returns accurate predictions. Model deployment in IBM Cloud Watson Studio is designed to be user-friendly and efficient. By following these steps, you ensure that your trained models are accessible and capable of making realtime predictions, which is a fundamental part of unlocking datadriven insights for informed decision-making.

Step 5: Integration with Applications:

In this step, we'll seamlessly incorporate the deployed machine learning models into your existing or new software applications, such as web apps, mobile apps, or backend services. Integration enables these applications to utilize the predictive power of your models in real-time, which, in turn, empowers data-driven decision-making.

1. API Integration:

Our deployed models typically expose an API endpoint that applications can communicate with. These APIs allow applications to send input data to the model and receive predictions as responses. Integration usually involves making HTTP requests to this API.

2. Data Input and Output Formats:

Ensure that our applications understand the expected input data format for the model. This might involve converting user inputs into a format suitable for the model, and then processing the model's predictions.

3. Security:

Implement security measures, including authentication and authorization, to ensure that only authorized applications can access and utilize your model's API. This helps protect sensitive data and the integrity of our predictive analytics.

4. Scalability:

Consider the scalability of your application and model integration. Ensure that our applications can handle a high volume of requests and that our deployment infrastructure can scale accordingly.

5. Error Handling:

Implement robust error-handling mechanisms in our applications to deal with scenarios where the model may fail to make predictions or encounters unexpected input data.

6. Testing:

Rigorously test the integration of the model within our applications using a variety of input data and scenarios. Verify that the predictions align with your expectations and are used appropriately.

By successfully integrating your predictive analytics models with our applications, we enable them to provide data-driven insights and real-time decision support to users. This is a pivotal step in making informed decisions, as outlined in your problem statement, and harnessing the magic of data-driven insights in a practical and actionable way.

Step 6: Monitoring and Maintenance:

To become proficient in predictive analytics with IBM Cloud Watson Studio, monitoring and maintenance are essential for ensuring that our predictive models consistently deliver accurate and valuable insights.

- Metric Tracking:

Use specific performance metrics relevant to our problem. For instance, if we're predicting customer churn, track metrics like accuracy, precision, recall, or F1-score to evaluate the model's performance.

- Continuous Performance Evaluation:

Regularly assess the performance of your deployed machine learning models. This involves measuring how well they make predictions and checking if they're still accurate.

- Alert Systems:

Set up automated alerts or notifications that signal potential issues with your models. These alerts act as early warning systems, similar to smoke detectors in a house.

- Data Updates:

If the data your models were trained on becomes outdated, consider updating it. New data ensures that your models remain relevant and effective, just like updating a map with new roads.

- Periodic Retraining:

Depending on how rapidly your data evolves, periodic retraining of your models might be necessary. This process involves using fresh data to fine-tune the models for improved accuracy, akin to studying to improve your knowledge.

- Documentation Updates:

Keep your documentation current. This documentation acts as a guide for anyone who manages or interacts with your models.

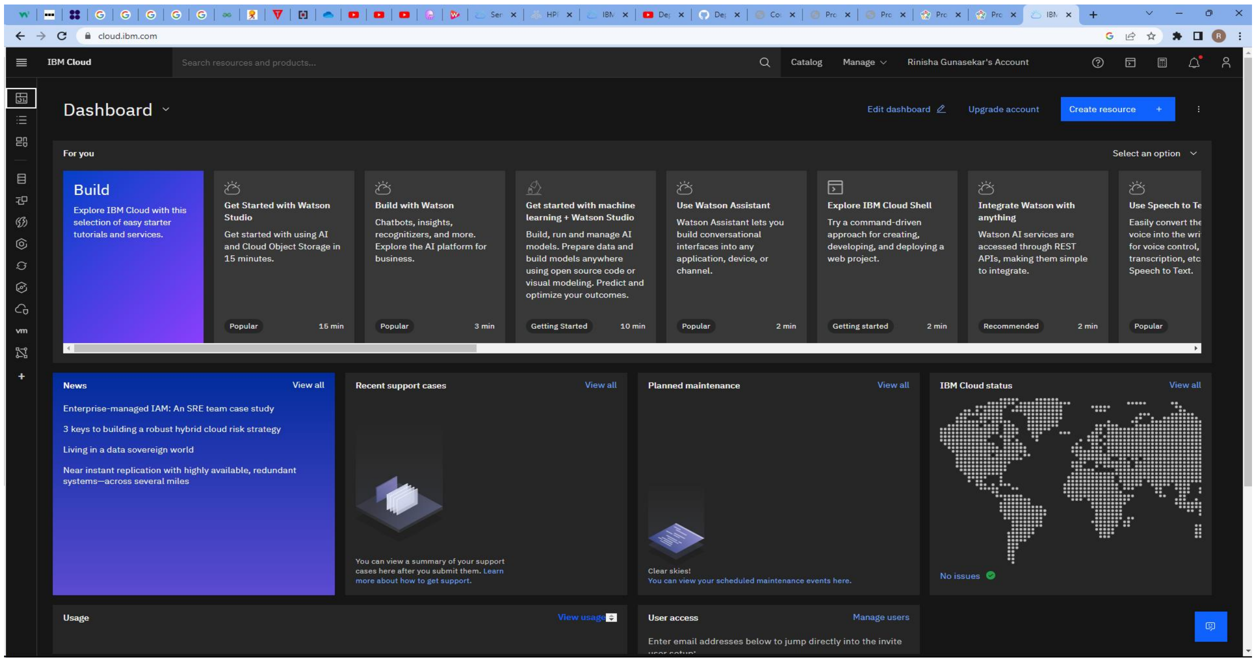
By the above steps, we can ensure that our predictive analytics capabilities remain up-to-date, accurate, and secure. Regular monitoring and maintenance are essential to unlock the full potential of data-driven insights and informed decision-making with IBM Cloud Watson Studio.

Real-Time Insights:

By deploying and integrating predictive models, we can gain access to real-time insights based on data. These insights help us to understand what's happening right now.

<u>Steps for Project Development:</u>

Step 1: Login to IBM cloud



cloud.ibm.com

IBM Cloud

Search resources and products...

Catalog Manage Rinisha Gunasekar's Account

Dashboard

Edit dashboard Upgrade account Create resource

For you Select an option

Build
Explore IBM Cloud with this selection of easy starter tutorials and services.

Get Started with Watson Studio
Get started with using AI and Cloud Object Storage in 15 minutes.
Popular 15 min

Build with Watson
Chatbots, insights, recognizers, and more. Explore the AI platform for business.
Popular 3 min

Get started with machine learning + Watson Studio
Build, run and manage AI models. Prepare data and build models anywhere using open source code or visual modeling. Predict and optimize your outcomes.
Getting Started 10 min

Use Watson Assistant
Watson Assistant lets you build conversational interfaces into any application, device, or channel.
Popular 2 min

Explore IBM Cloud Shell
Try a command-driven approach for creating, developing, and deploying a web project.
Getting started 2 min

Integrate Watson with anything
Watson AI services are accessed through REST APIs, making them simple to integrate.
Recommended 2 min

Use Speech to Te
Easily convert the voice into the wri for voice control, transcription, etc Speech to Text.
Popular

News View all
Enterprise-managed IAM: An SRE team case study
3 keys to building a robust hybrid cloud risk strategy
Living in a data sovereign world
Near instant replication with highly available, redundant systems—across several miles

Recent support cases View all
You can view a summary of your support cases here after you submit them. Learn more about how to get support.

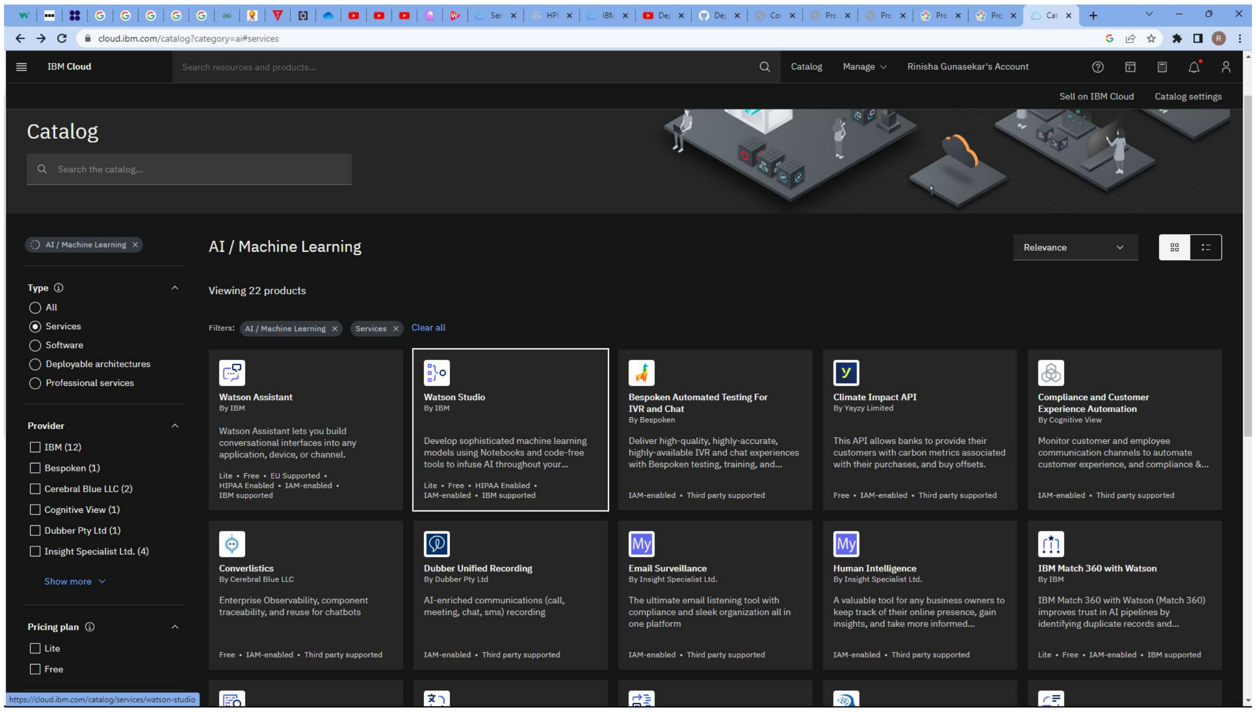
Planned maintenance View all
Clear skies!
You can view your scheduled maintenance events here.

IBM Cloud status View all
No issues

Usage View usage

User access Manage users
Enter email addresses below to jump directly into the invite

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



cloud.ibm.com/catalog?category=ai#services

IBM Cloud | Search resources and products... | Catalog | Manage | Rinisha Gunasekar's Account

Sell on IBM Cloud | Catalog settings

Catalog

Search the catalog...

AI / Machine Learning ×

Type
- All
- Services (selected)
- Software
- Deployable architectures
- Professional services

Provider
- IBM (12)
- Bespoken (1)
- Cerebral Blue LLC (2)
- Cognitive View (1)
- Dubber Pty Ltd (1)
- Insight Specialist Ltd. (4)

Show more

Pricing plan
- Lite
- Free

AI / Machine Learning

Relevance

Viewing 22 products

Filters: AI / Machine Learning × Services × Clear all

Watson Assistant
By IBM
Watson Assistant lets you build conversational interfaces into any application, device, or channel.
Lite • Free • EU Supported • HIPAA Enabled • IAM-enabled • IBM supported

Watson Studio
By IBM
Develop sophisticated machine learning models using Notebooks and code-free tools to infuse AI throughout your...
Lite • Free • HIPAA Enabled • IAM-enabled • IBM supported

Bespoken Automated Testing For IVR and Chat
By Bespoken
Deliver high-quality, highly-accurate, highly-available IVR and chat experiences with Bespoken testing, training, and...
IAM-enabled • Third party supported

Climate Impact API
By Yayzy Limited
This API allows banks to provide their customers with carbon metrics associated with their purchases, and buy offsets.
Free • IAM-enabled • Third party supported

Compliance and Customer Experience Automation
By Cognitive View
Monitor customer and employee communication channels to automate customer experience, and compliance &...
IAM-enabled • Third party supported

Converlistics
By Cerebral Blue LLC
Enterprise Observability, component traceability, and reuse for chatbots
Free • IAM-enabled • Third party supported

Dubber Unified Recording
By Dubber Pty Ltd
AI-enriched communications (call, meeting, chat, sms) recording
IAM-enabled • Third party supported

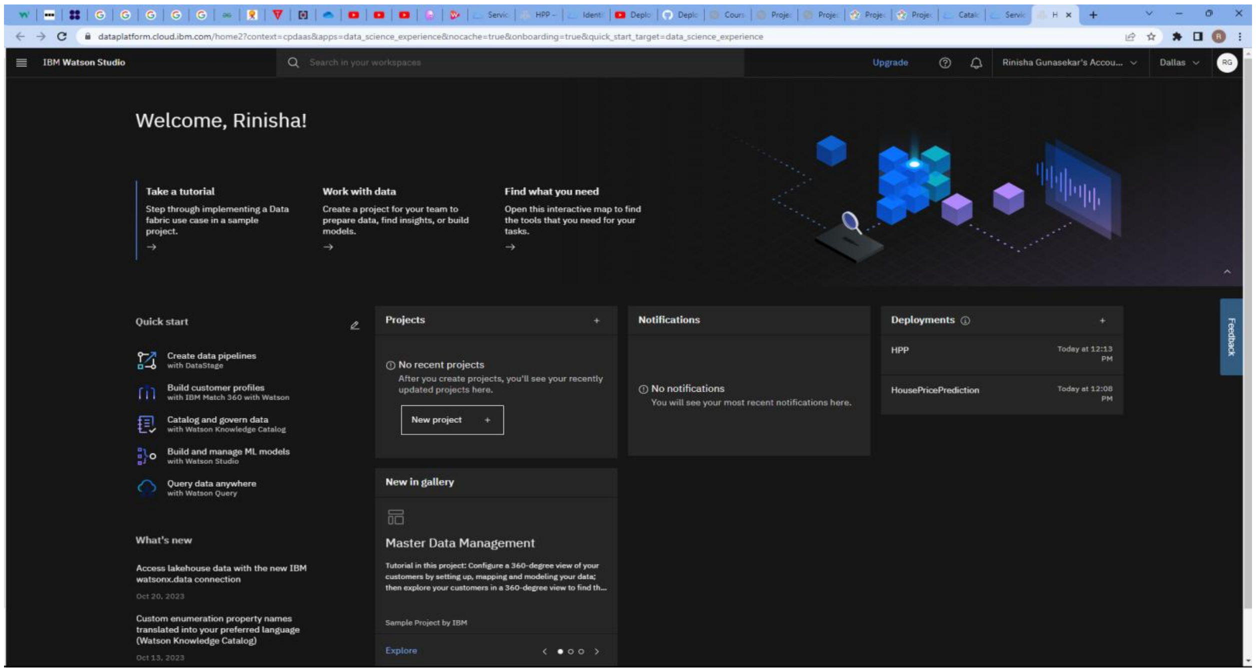
Email Surveillance
By Insight Specialist Ltd.
The ultimate email listening tool with compliance and sleek organization all in one platform
IAM-enabled • Third party supported

Human Intelligence
By Insight Specialist Ltd.
A valuable tool for any business owners to keep track of their online presence, gain insights, and take more informed...
IAM-enabled • Third party supported

IBM Match 360 with Watson
By IBM
IBM Match 360 with Watson (Match 360) improves trust in AI pipelines by identifying duplicate records and...
Lite • Free • IAM-enabled • IBM supported

https://cloud.ibm.com/catalog/services/watson-studio

Step 3: Get started to launch Watson Studio Dashboard.



dataplatform.cloud.ibm.com/home2?context=cpdaas&apps=data_science_experience&nocache=true&onboarding=true&quick_start_target=data_science_experience

IBM Watson Studio | Search in your workspaces | Upgrade | Rinisha Gunasekar's Accou... | Dallas | RG

Welcome, Rinisha!

Take a tutorial
Step through implementing a Data fabric use case in a sample project.
→

Work with data
Create a project for your team to prepare data, find insights, or build models.
→

Find what you need
Open this interactive map to find the tools that you need for your tasks.
→

Quick start
- Create data pipelines with DataStage
- Build customer profiles with IBM Match 360 with Watson
- Catalog and govern data with Watson Knowledge Catalog
- Build and manage ML models with Watson Studio
- Query data anywhere with Watson Query

Projects +
No recent projects
After you create projects, you'll see your recently updated projects here.
New project +

Notifications
No notifications
You will see your most recent notifications here.

Deployments +

HPP	Today at 12:13 PM
HousePricePrediction	Today at 12:08 PM

What's new

Access lakehouse data with the new IBM watsonx.data connection
Oct 20, 2023

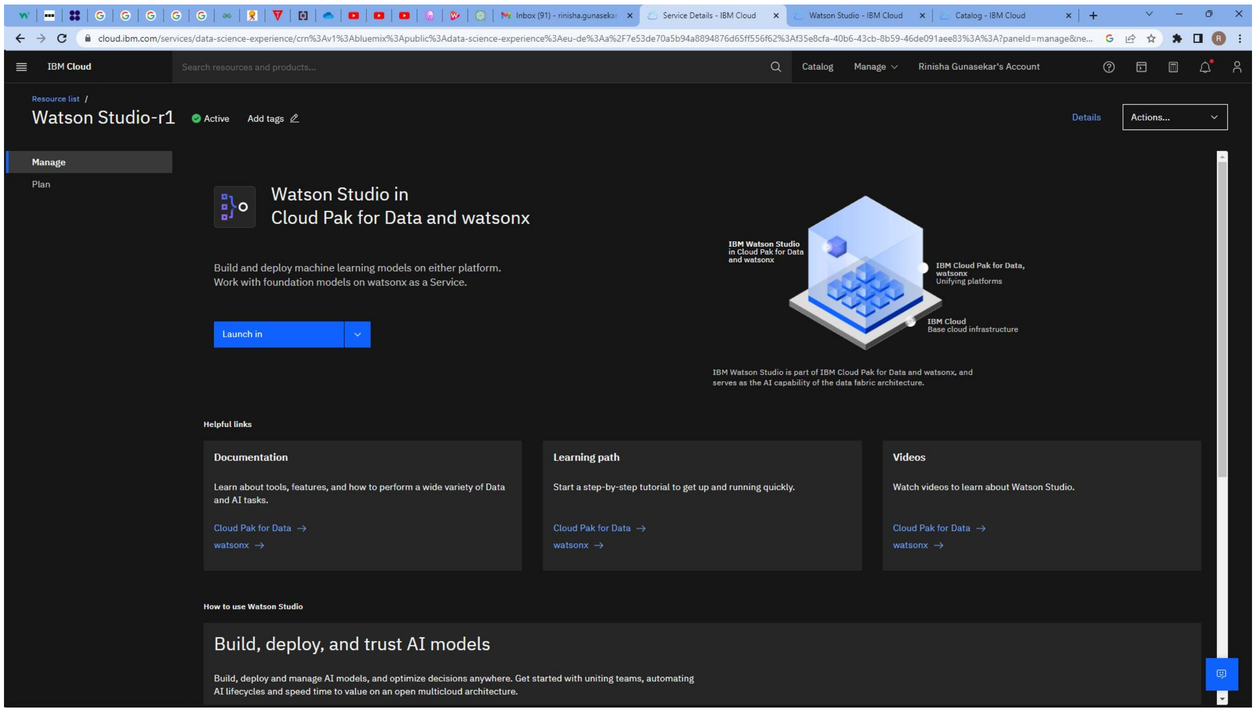
Custom enumeration property names translated into your preferred language (Watson Knowledge Catalog)
Oct 13, 2023

New in gallery

Master Data Management
Tutorial in this project: Configure a 360-degree view of your customers by setting up, mapping and modeling your data; then explore your customers in a 360-degree view to find th...
Sample Project by IBM
Explore

Feedback

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



cloud.ibm.com/services/data-science-experience/crn%3Av1%3Abluemix%3Apublic%3Adata-science-experience%3Aeu-de%3Aa%2F7e53de70a5b94a8894876d65ff556f62%3Af35e8cfa-40b6-43cb-8b59-46de091aee83%3A%3A?paneld=manage&ne...
IBM Cloud
Search resources and products...
Catalog
Manage
Rinisha Gunasekar's Account
Resource list /
Details
Actions...
Manage
Plan
Launch in
IBM Watson Studio in Cloud Pak for Data and watsonx
IBM Cloud Pak for Data, watsonx Unifying platforms
IBM Cloud Base cloud infrastructure

Watson Studio-r1 Active Add tags

Watson Studio in Cloud Pak for Data and watsonx

Build and deploy machine learning models on either platform. Work with foundation models on watsonx as a Service.

IBM Watson Studio is part of IBM Cloud Pak for Data and watsonx, and serves as the AI capability of the data fabric architecture.

Helpful links

Documentation
Learn about tools, features, and how to perform a wide variety of Data and AI tasks.
Cloud Pak for Data →
watsonx →

Learning path
Start a step-by-step tutorial to get up and running quickly.
Cloud Pak for Data →
watsonx →

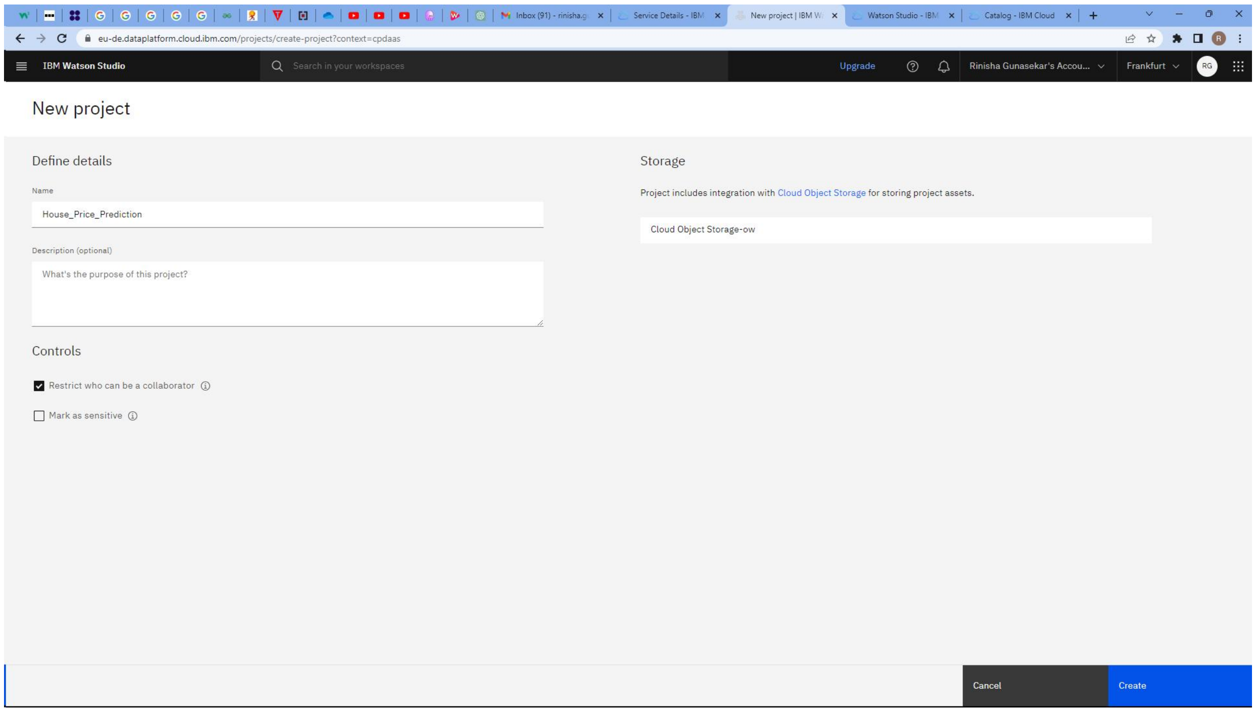
Videos
Watch videos to learn about Watson Studio.
Cloud Pak for Data →
watsonx →

How to use Watson Studio

Build, deploy, and trust AI models

Build, deploy and manage AI models, and optimize decisions anywhere. Get started with uniting teams, automating AI lifecycles and speed time to value on an open multicloud architecture.

Step 5: Start a New project



eu-de.dataplatform.cloud.ibm.com/projects/create-project?context=cpdaas
IBM Watson Studio
Search in your workspaces
Upgrade
Rinisha Gunasekar's Accou...
Frankfurt
Cancel
Create

New project

Define details

Name: House_Price_Prediction

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

Controls

[✓] Restrict who can be a collaborator

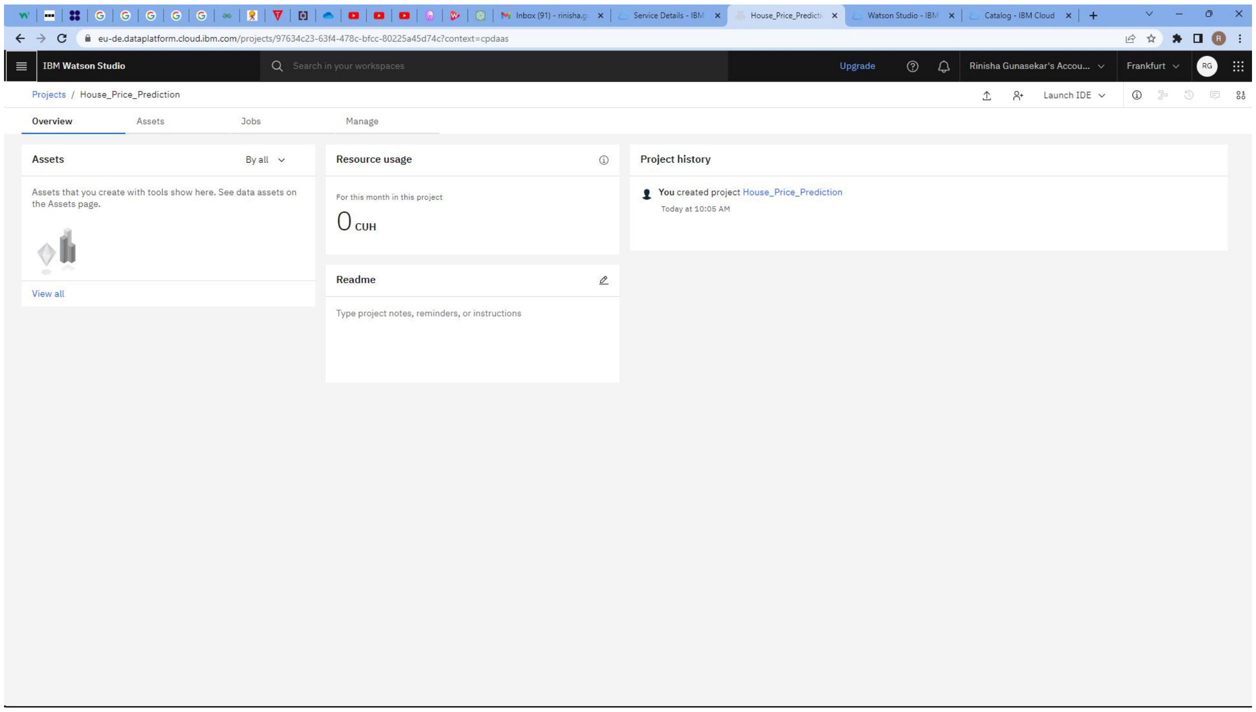
[ ] Mark as sensitive

Storage

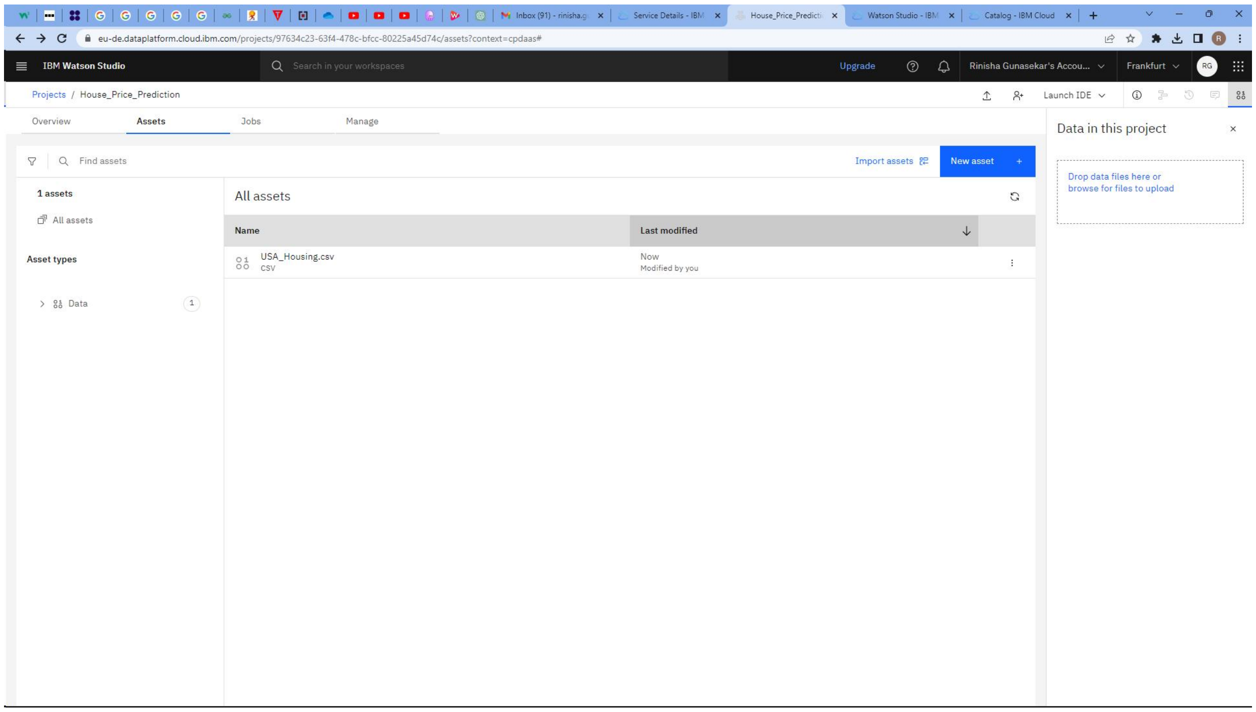
Project includes integration with Cloud Object Storage for storing project assets.

Cloud Object Storage-ow

Step 6: Create a project named *House_Price_Prediction*

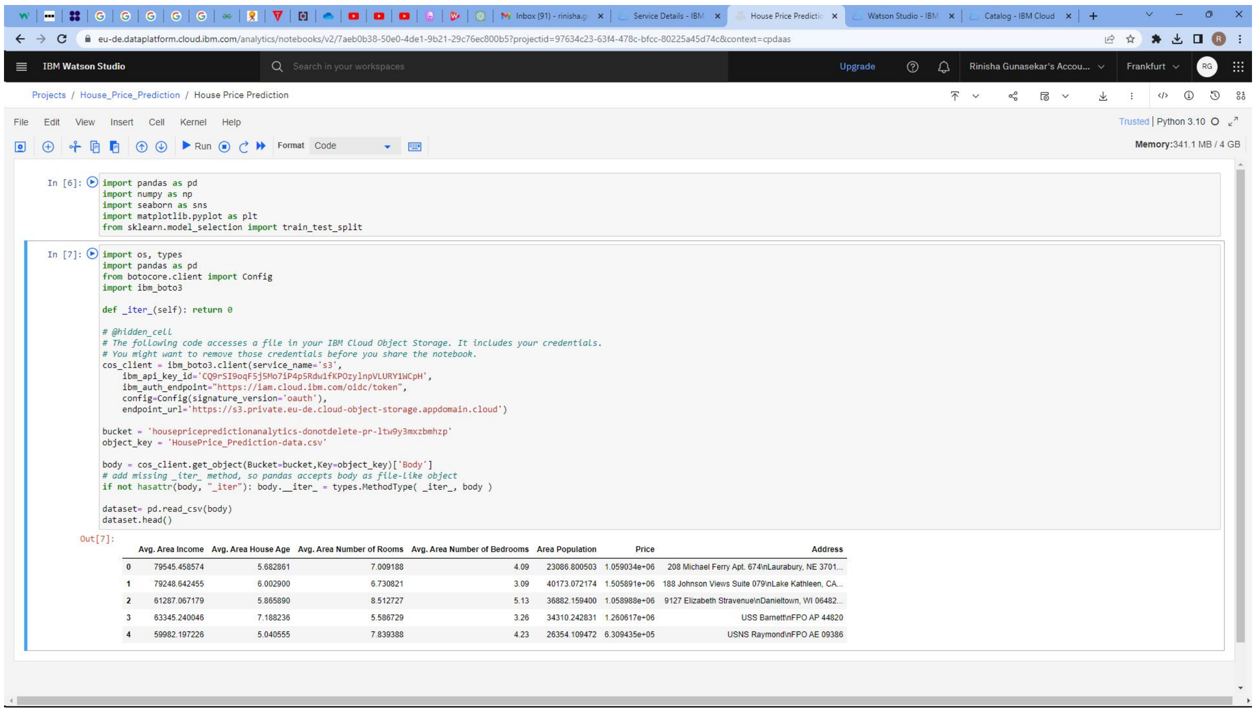

eu-de.dataplatform.cloud.ibm.com/projects/97634c23-63f4-478c-bfcc-80225a45d74c?context=cpdaas
IBM Watson Studio
Search in your workspaces
Upgrade
Rinisha Gunasekar's Accou...
Frankfurt
Projects / House_Price_Prediction
Launch IDE
Overview
Assets
Jobs
Manage
Assets
By all
Assets that you create with tools show here. See data assets on the Assets page.
View all
Resource usage
For this month in this project
0 CUH
Readme
Type project notes, reminders, or instructions
Project history
You created project House_Price_Prediction
Today at 10:06 AM

Step 7: Import all assets


eu-de.dataplatform.cloud.ibm.com/projects/97634c23-63f4-478c-bfcc-80225a45d74c/assets?context=cpdaas#
IBM Watson Studio
Search in your workspaces
Upgrade
Rinisha Gunasekar's Accou...
Frankfurt
Projects / House_Price_Prediction
Launch IDE
Overview
Assets
Jobs
Manage
Find assets
Import assets
New asset
1 assets
All assets
Asset types
Data 1
All assets
Name
Last modified
USA_Housing.csv
CSV
Now
Modified by you
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.



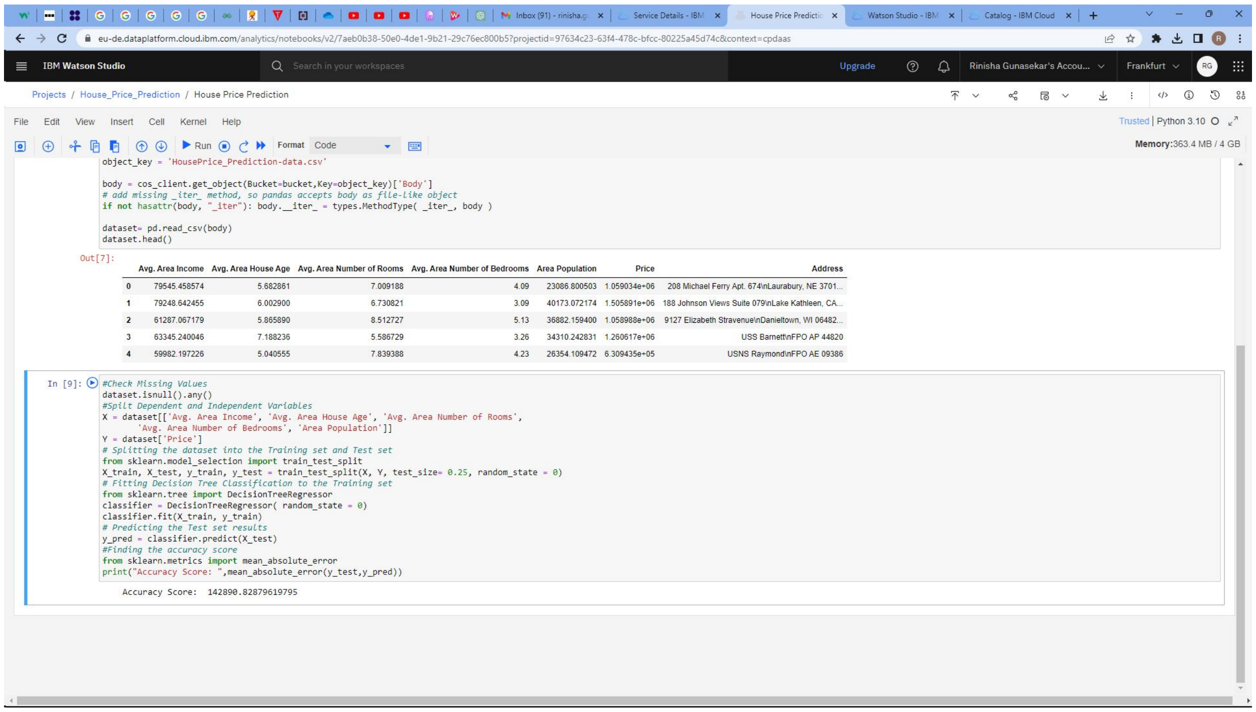
```
In [6]: import pandas as pd
        import numpy as np
        import seaborn as sns
        import matplotlib.pyplot as plt
        from sklearn.model_selection import train_test_split
```

```
In [7]: import os, types
        import pandas as pd
        from botocore.client import Config
        import ibm_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 = ibm_boto3.client(service_name='s3',
            ibm_api_key_id='CQ9rSI9oqF5j5Mo7iP4p5Rdw1fKPOzylnpVLURY1WCpH',
            ibm_auth_endpoint="https://iam.cloud.ibm.com/oidc/token",
            config=Config(signature_version='oauth'),
            endpoint_url='https://s3.private.eu-de.cloud-object-storage.appdomain.cloud')

        bucket = 'housepricepredictionanalytics-donotdelete-pr-ltw9y3mxzbmhzp'
        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. 674\nLaurabury, NE 3701...
1	79248.642455	6.002900	6.730821	3.09	40173.072174	1.505891e+06	188 Johnson Views Suite 079\nLake Kathleen, CA...
2	61287.067179	5.865890	8.512727	5.13	36882.159400	1.058988e+06	9127 Elizabeth Stravenue\nDanieltown, WI 06482...
3	63345.240046	7.188236	5.586729	3.26	34310.242831	1.260617e+06	USS Barnett\nFPO AP 44820
4	59982.197226	5.040555	7.839388	4.23	26354.109472	6.309435e+05	USNS Raymond\nFPO AE 09386

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



```
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. 674\nLaurabury, NE 3701...
1	79248.642455	6.002900	6.730821	3.09	40173.072174	1.505891e+06	188 Johnson Views Suite 079\nLake Kathleen, CA...
2	61287.067179	5.865890	8.512727	5.13	36882.159400	1.058988e+06	9127 Elizabeth Stravenue\nDanieltown, WI 06482...
3	63345.240046	7.188236	5.586729	3.26	34310.242831	1.260617e+06	USS Barnett\nFPO AP 44820
4	59982.197226	5.040555	7.839388	4.23	26354.109472	6.309435e+05	USNS Raymond\nFPO AE 09386

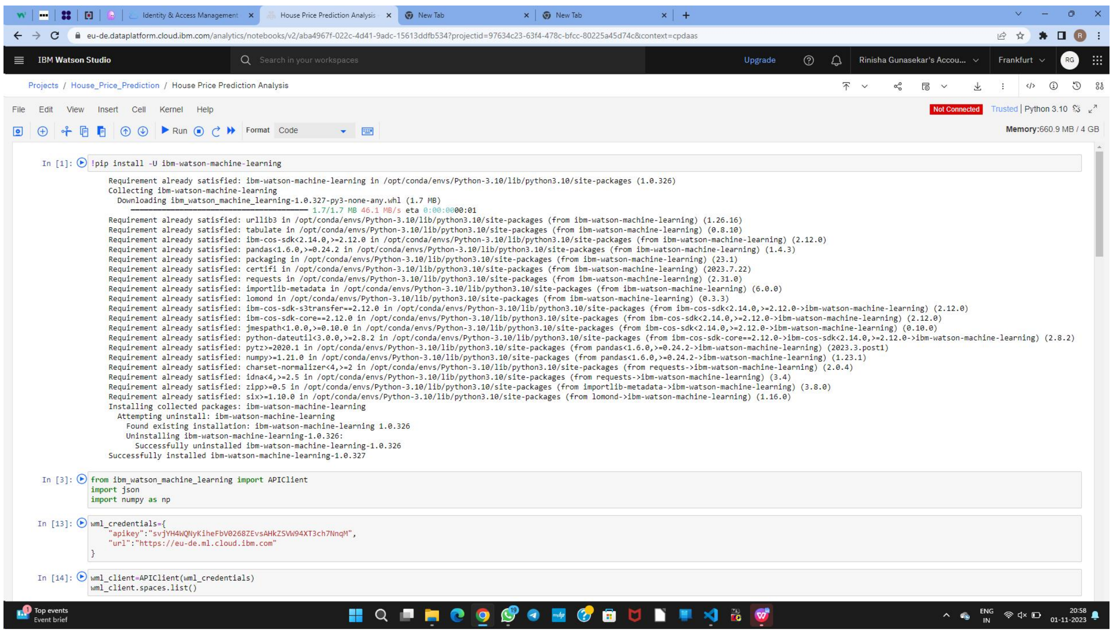
In [9]:

```
#Check Missing Values
dataset.isnull().any()
#Spilt 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:  142890.82879619795
```

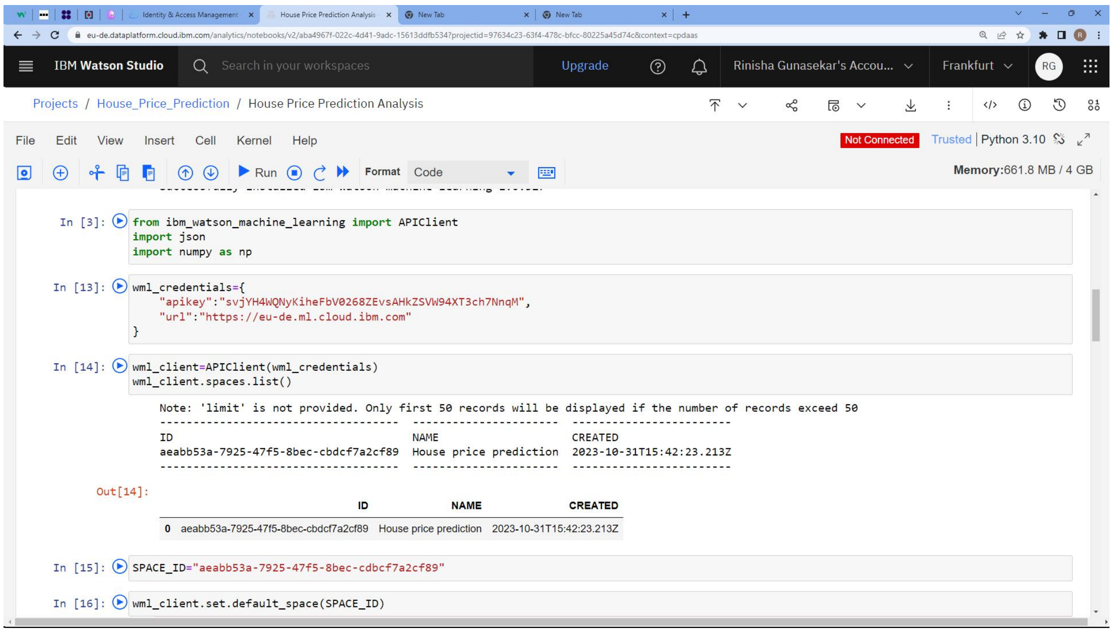
Hence the model was build, trained and tested.

Step 10: Install the ibm-watson-machine-learning



```
In [1]: !pip install -U ibm-watson-machine-learning
          Requirement already satisfied: ibm-watson-machine-learning in /opt/conda/envs/Python-3.10/lib/python3.10/site-packages (1.0.326)
          Collecting ibm-watson-machine-learning
            Downloading ibm_watson_machine_learning-1.0.327-py3-none-any.whl (1.7 MB)
               ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 1.7/1.7 MB 46.1 MB/s eta 0:00:0000:01
          Requirement already satisfied: urllib3 in /opt/conda/envs/Python-3.10/lib/python3.10/site-packages (from ibm-watson-machine-learning) (1.26.16)
          Requirement already satisfied: tabulate in /opt/conda/envs/Python-3.10/lib/python3.10/site-packages (from ibm-watson-machine-learning) (0.8.10)
          Requirement already satisfied: ibm-cos-sdk<2.14.0,>=2.12.0 in /opt/conda/envs/Python-3.10/lib/python3.10/site-packages (from ibm-watson-machine-learning) (2.12.0)
          Requirement already satisfied: pandas<1.6.0,>=0.24.2 in /opt/conda/envs/Python-3.10/lib/python3.10/site-packages (from ibm-watson-machine-learning) (1.4.3)
          Requirement already satisfied: packaging in /opt/conda/envs/Python-3.10/lib/python3.10/site-packages (from ibm-watson-machine-learning) (23.1)
          Requirement already satisfied: certifi in /opt/conda/envs/Python-3.10/lib/python3.10/site-packages (from ibm-watson-machine-learning) (2023.7.22)
          Requirement already satisfied: requests in /opt/conda/envs/Python-3.10/lib/python3.10/site-packages (from ibm-watson-machine-learning) (2.31.0)
          Requirement already satisfied: importlib-metadata in /opt/conda/envs/Python-3.10/lib/python3.10/site-packages (from ibm-watson-machine-learning) (6.0.0)
          Requirement already satisfied: lomond in /opt/conda/envs/Python-3.10/lib/python3.10/site-packages (from ibm-watson-machine-learning) (0.3.3)
          Requirement already satisfied: ibm-cos-sdk-s3transfer==2.12.0 in /opt/conda/envs/Python-3.10/lib/python3.10/site-packages (from ibm-cos-sdk<2.14.0,>=2.12.0->ibm-watson-machine-learning) (2.12.0)
          Requirement already satisfied: ibm-cos-sdk-core==2.12.0 in /opt/conda/envs/Python-3.10/lib/python3.10/site-packages (from ibm-cos-sdk<2.14.0,>=2.12.0->ibm-watson-machine-learning) (2.12.0)
          Requirement already satisfied: jmespath<1.0.0,>=0.10.0 in /opt/conda/envs/Python-3.10/lib/python3.10/site-packages (from ibm-cos-sdk<2.14.0,>=2.12.0->ibm-watson-machine-learning) (0.10.0)
          Requirement already satisfied: python-dateutil<3.0.0,>=2.8.2 in /opt/conda/envs/Python-3.10/lib/python3.10/site-packages (from ibm-cos-sdk-core==2.12.0->ibm-cos-sdk<2.14.0,>=2.12.0->ibm-watson-machine-learning) (2.8.2)
          Requirement already satisfied: pytz>=2020.1 in /opt/conda/envs/Python-3.10/lib/python3.10/site-packages (from pandas<1.6.0,>=0.24.2->ibm-watson-machine-learning) (2023.3.post1)
          Requirement already satisfied: numpy>=1.21.0 in /opt/conda/envs/Python-3.10/lib/python3.10/site-packages (from pandas<1.6.0,>=0.24.2->ibm-watson-machine-learning) (1.23.1)
          Requirement already satisfied: charset-normalizer<4,>=2 in /opt/conda/envs/Python-3.10/lib/python3.10/site-packages (from requests->ibm-watson-machine-learning) (2.0.4)
          Requirement already satisfied: idna<4,>=2.5 in /opt/conda/envs/Python-3.10/lib/python3.10/site-packages (from requests->ibm-watson-machine-learning) (3.4)
          Requirement already satisfied: zipp>=0.5 in /opt/conda/envs/Python-3.10/lib/python3.10/site-packages (from importlib-metadata->ibm-watson-machine-learning) (3.8.0)
          Requirement already satisfied: six>=1.10.0 in /opt/conda/envs/Python-3.10/lib/python3.10/site-packages (from lomond->ibm-watson-machine-learning) (1.16.0)
          Installing collected packages: ibm-watson-machine-learning
            Attempting uninstall: ibm-watson-machine-learning
              Found existing installation: ibm-watson-machine-learning 1.0.326
              Uninstalling ibm-watson-machine-learning-1.0.326:
                Successfully uninstalled ibm-watson-machine-learning-1.0.326
          Successfully installed ibm-watson-machine-learning-1.0.327

In [3]: from ibm_watson_machine_learning import APIClient
        import json
        import numpy as np

In [13]: wml_credentials={
             "apikey":"svjYH4WQNyKiheFbV0268ZEvsAHkZSVW94XT3ch7NnqM",
             "url":"https://eu-de.ml.cloud.ibm.com"
         }

In [14]: wml_client=APIClient(wml_credentials)
         wml_client.spaces.list()
```

Step 11: Import APIClient, json and numpy. Add the apikey. Create the deployment SPACE_ID

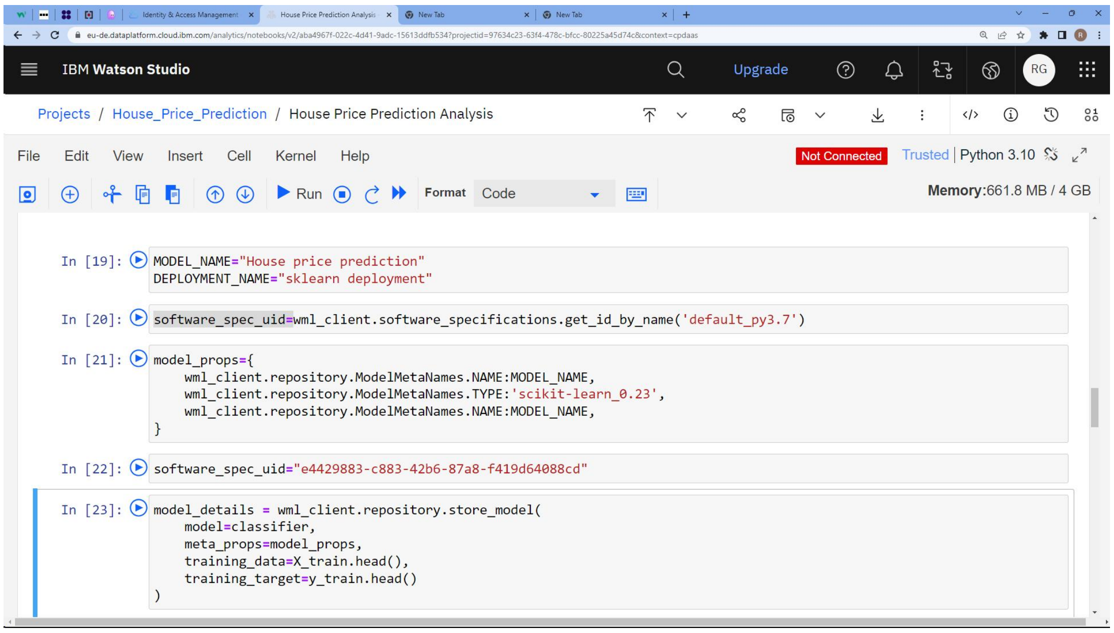


```
In [3]: from ibm_watson_machine_learning import APIClient
        import json
        import numpy as np

In [13]: wml_credentials={
             "apikey":"svjYH4WQNyKiheFbV0268ZEvsAHkZSVW94XT3ch7NnqM",
             "url":"https://eu-de.ml.cloud.ibm.com"
         }

In [14]: wml_client=APIClient(wml_credentials)
         wml_client.spaces.list()

             Note: 'limit' is not provided. Only first 50 records will be displayed if the number of records exceed 50
             ------------------------------------  --------------------  ------------------------
             ID                                    NAME                  CREATED
             aeabb53a-7925-47f5-8bec-cbdcf7a2cf89  House price prediction  2023-10-31T15:42:23.213Z
             ------------------------------------  --------------------  ------------------------

Out[14]:
```

	ID	NAME	CREATED
0	aeabb53a-7925-47f5-8bec-cbdcf7a2cf89	House price prediction	2023-10-31T15:42:23.213Z

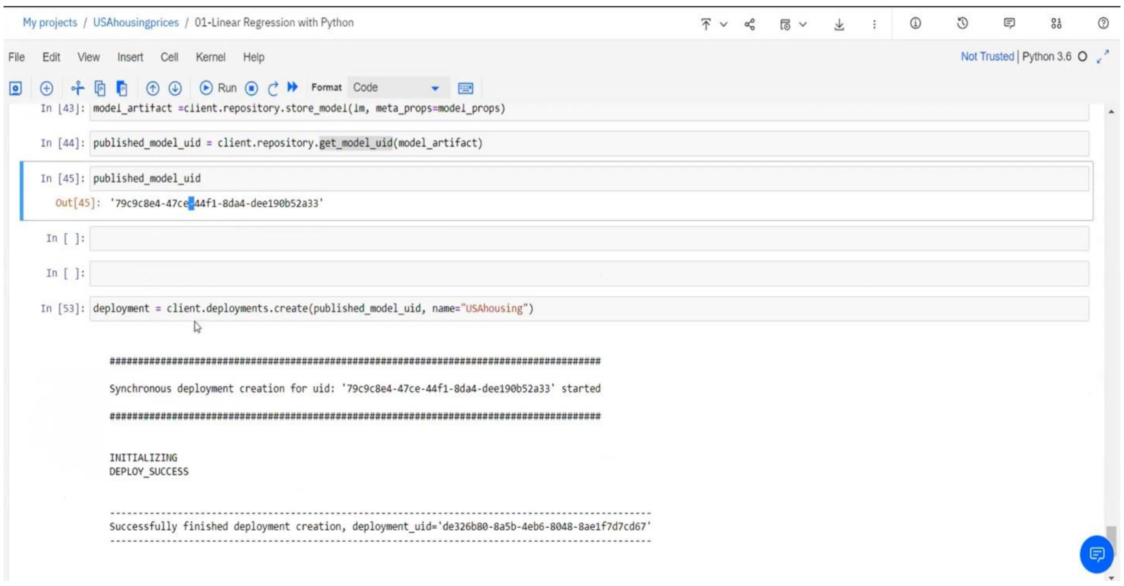
```
In [15]: SPACE_ID="aeabb53a-7925-47f5-8bec-cbdcf7a2cf89"

In [16]: wml_client.set_default_space(SPACE_ID)
```

Step 12: The props and details of the 'house price prediction' model was given.


IBM Watson Studio
Upgrade
Projects / House_Price_Prediction / House Price Prediction Analysis
File Edit View Insert Cell Kernel Help
Not Connected
Trusted | Python 3.10
Run
Format Code
Memory:661.8 MB / 4 GB

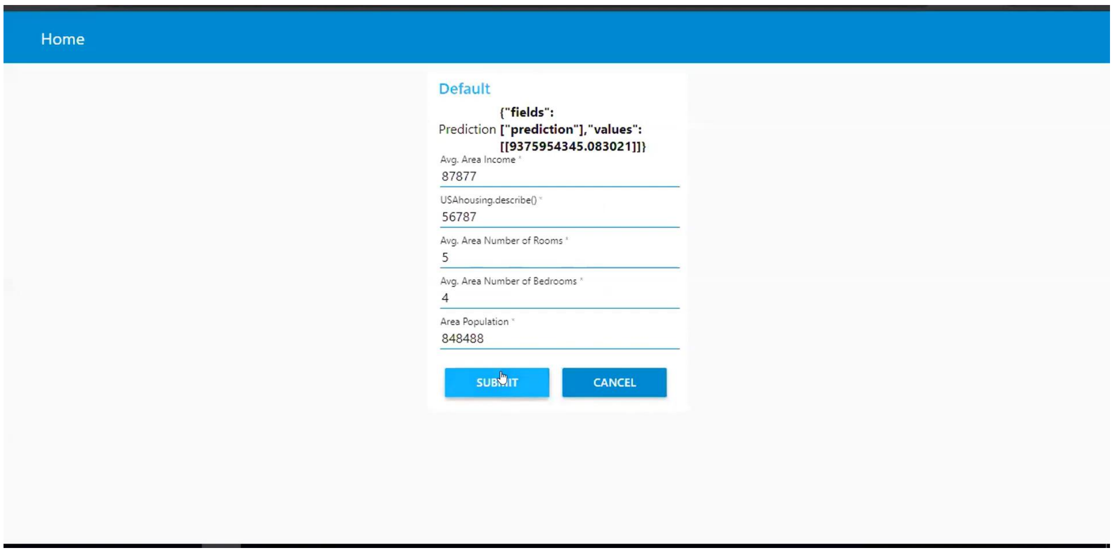
```
In [19]: MODEL_NAME="House price prediction"
         DEPLOYMENT_NAME="sklearn deployment"

In [20]: software_spec_uid=wml_client.software_specifications.get_id_by_name('default_py3.7')

In [21]: model_props={
             wml_client.repository.ModelMetaNames.NAME:MODEL_NAME,
             wml_client.repository.ModelMetaNames.TYPE:'scikit-learn_0.23',
             wml_client.repository.ModelMetaNames.NAME:MODEL_NAME,
         }

In [22]: software_spec_uid="e4429883-c883-42b6-87a8-f419d64088cd"

In [23]: model_details = wml_client.repository.store_model(
             model=classifier,
             meta_props=model_props,
             training_data=X_train.head(),
             training_target=y_train.head()
         )
```

Step 13: The deployment creation is successfully finished.


My projects / USAhousingprices / 01-Linear Regression with Python
File Edit View Insert Cell Kernel Help
Not Trusted | Python 3.6
Run
Format Code

```
In [43]: model_artifact =client.repository.store_model(lm, meta_props=model_props)

In [44]: published_model_uid = client.repository.get_model_uid(model_artifact)

In [45]: published_model_uid
Out[45]: '79c9c8e4-47ce-44f1-8da4-dee190b52a33'

In [ ]:

In [ ]:

In [53]: deployment = client.deployments.create(published_model_uid, name="USAhousing")


    #######################################################################################

    Synchronous deployment creation for uid: '79c9c8e4-47ce-44f1-8da4-dee190b52a33' started

    #######################################################################################


    INITIALIZING
    DEPLOY_SUCCESS


    ------------------------------------------------------------------------------------------------
    Successfully finished deployment creation, deployment_uid='de326b80-8a5b-4eb6-8048-8ae1f7d7cd67'
    ------------------------------------------------------------------------------------------------
```

Step 14: Now we have deployed our machine learning model as a Web service. Once the model is deployed ,it can be used to make predictions or provide other intelligent services to web users.



Default

Prediction {"fields":["prediction"],"values":[[9375954345.083021]]}

Avg. Area Income *
87877

USAhousing.describe() *
56787

Avg. Area Number of Rooms *
5

Avg. Area Number of Bedrooms *
4

Area Population *
848488

<u>Conclusion:</u>

Hence the House Price Prediction using machine learning model deployment using ibm cloud watson studio is developed successfully.