

# Training ML Model on IBM Watson

**TEAM ID: PNT2022TMID38951**

**PROJECT: University Admit Eligibility Predictor**

## i) Setting up Watson Studio for running Jupyter notebooks

The screenshot shows the IBM Cloud Pak for Data interface. The top navigation bar includes the IBM Cloud Pak for Data logo, a search bar, and user information (SANTHOSHKUMAR K's Acc...). The main content area is titled 'Projects / uaep'. It features a sidebar with tabs for 'Overview', 'Assets', 'Jobs', and 'Manage'. The 'Overview' tab is active, displaying a list of assets on the left, including 'graduate\_admission\_prediction\_C', 'IBM\_Watson\_ML\_Model\_Deployment', 'assignment 3', and 'UAPE'. The main area shows 'Resource usage' (10 CUH) and 'Project history' (You created project uaep on Nov 12, 2022 04:21 PM).

The screenshot shows the IBM Cloud Pak for Data interface with the 'Assets' tab selected. The left sidebar shows '5 assets' and 'Asset types' (Data: 1, Notebooks: 4). The main area displays a table of 'All assets' with columns 'Name' and 'Last modified'. The table lists five assets: 'graduate\_admission\_prediction\_C', 'IBM\_Watson\_ML\_Model\_Deployment', 'assignment 3', 'admission\_predict.csv', and 'UAPE'. A 'Data in this project' panel on the right shows a drop zone for data files.

Name	Last modified
graduate_admission_prediction_C Notebook	2 hours ago Modified by you
IBM_Watson_ML_Model_Deployment Notebook	23 hours ago Modified by you
assignment 3 Notebook	2 days ago Modified by you
admission_predict.csv CSV	5 days ago Modified by you
UAPE Notebook	5 days ago Modified by you

## Training and saving the model in IBM Watson Machine Learning Service

## PERSISTING THE MULTIPLE LINEAR REGRESSION MODEL AND DEPLOYING IT IN IBM CLOUD

```
In [60]: #Set Python Version
software_spec_uid = client.software_specifications.get_uid_by_name("runtime-22.1-py3.9")
software_spec_uid
```

```
Out[60]: '12b83a17-24d8-5082-900f-0ab31fbfd3cb'
```

```
In [61]: model_details = client.repository.store_model(model = multiple_lin_reg, meta_props={
    client.repository.ModelMetaNames.NAME: "UAEP_Multiple_Linear_Regression",
    client.repository.ModelMetaNames.TYPE: "scikit-learn_1.0",
    client.repository.ModelMetaNames.SOFTWARE_SPEC_UID: software_spec_uid
})

model_id = client.repository.get_model_id(model_details)
```

```
In [62]: model_id
```

```
Out[62]: '8083e827-e81f-40d1-84ab-20d511771869'
```

### Assets:

Deployments /

### Regression-Models

Deployment space for the University Admit Eligibility Predictor project

Overview **Assets** Deployments Jobs Manage

Find assets

1 asset

All assets 1

Asset types

Models 1

Name	Last modified
UAEP_Multiple_Linear_Regression Model	34 minutes ago Service

Import assets

### Deployments:

Deployments /

### Regression-Models

Deployment space for the University Admit Eligibility Predictor project

Overview Assets **Deployments** Jobs Manage

Search

Name	Type	Status	Asset	Last modified
UAEP_Multiple_Linear_Regression_Deployment	Online	Deployed	UAEP_Multiple_Linear_Regression	35 minutes ago Krishnan S (You)

ii) **Testing the created model using the API created for the deployed model:**

```
import requests

# NOTE: you must manually set API_KEY below using information retrieved from your IBM Cloud account.
API_KEY = "<Your-API-Key>"
token_response = requests.post('https://iam.cloud.ibm.com/identity/token', data={"apikey":
    API_KEY, "grant_type": 'urn:ibm:params:oauth:grant-type:apikey'})
mltoken = token_response.json()["access_token"]

header = {'Content-Type': 'application/json', 'Authorization': 'Bearer ' + mltoken}

# NOTE: manually define and pass the array(s) of values to be scored in the next line
payload_scoring = {"input_data": [{"field": ["GRE Score", "TOEFL Score", "University Rating", "SOP", "LOR ", "CGPA", "Research"]}, {"values": [[326, 110, 2, 3.5, 4, 9.23, 1]]}]}

response_scoring = requests.post('https://us-south.ml.cloud.ibm.com/ml/v4/deployments/uaep_deployment/predictions?version=2022-11-12', json=payload_scoring,
    headers={'Authorization': 'Bearer ' + mltoken})
print("Scoring response")
print(response_scoring.json())

Scoring response
{'predictions': [{'fields': ['prediction'], 'values': [[[0.8448151378927107]]]]}]
```