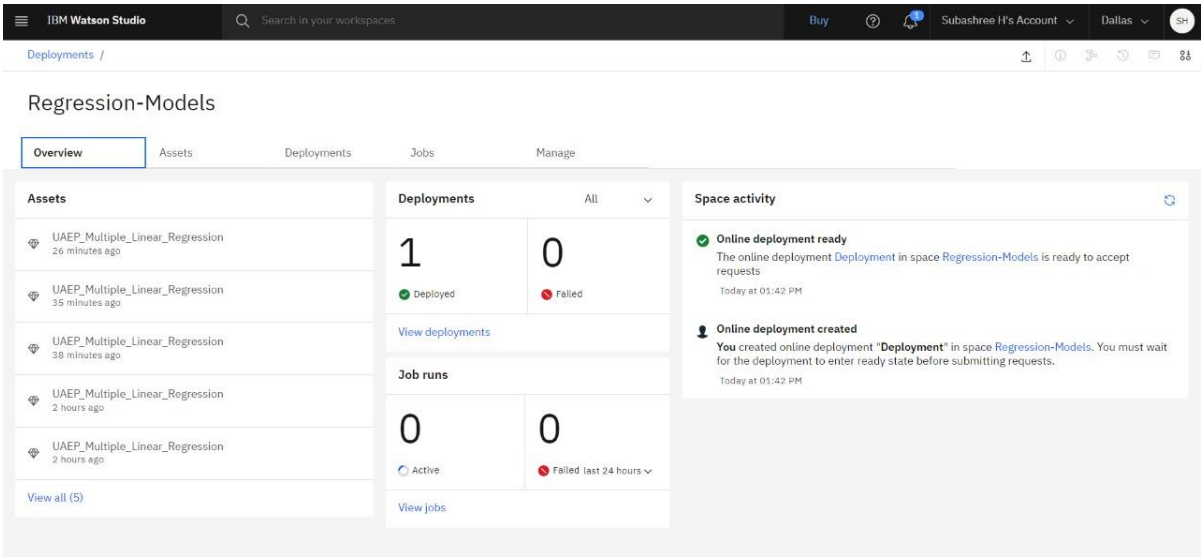
Training ML Model on IBM Watson

TEAM ID: PNT2022TMID32680

PROJECT: University Admit Eligibility Predictor

i) Setting up Watson Studio for running Jupyter notebooks



ii) Training and saving the model in IBM Watson Machine Learning Service

```
from ibm_watson_machine_learning import APIClient
wml_credentials = {
    "url": "https://us-south.ml.cloud.ibm.com",
    "apikey": "CB8dTF8Y0_o4amdiEL2FfNBLZ-EdGUGdLqrDk7Z2gzDP"
}

client = APIClient(wml_credentials)
```

Creating Deployment Space

```
def guid_from_space_name(client, space_name):
    space = client.spaces.get_details()
    return (next(item for item in space['resources'] if item['entity']['name'] == space_name)['metadata']['id'])
```

```
space_uid = guid_from_space_name(client, 'Regression-Models')
print("Space UID = " + space_uid)
```

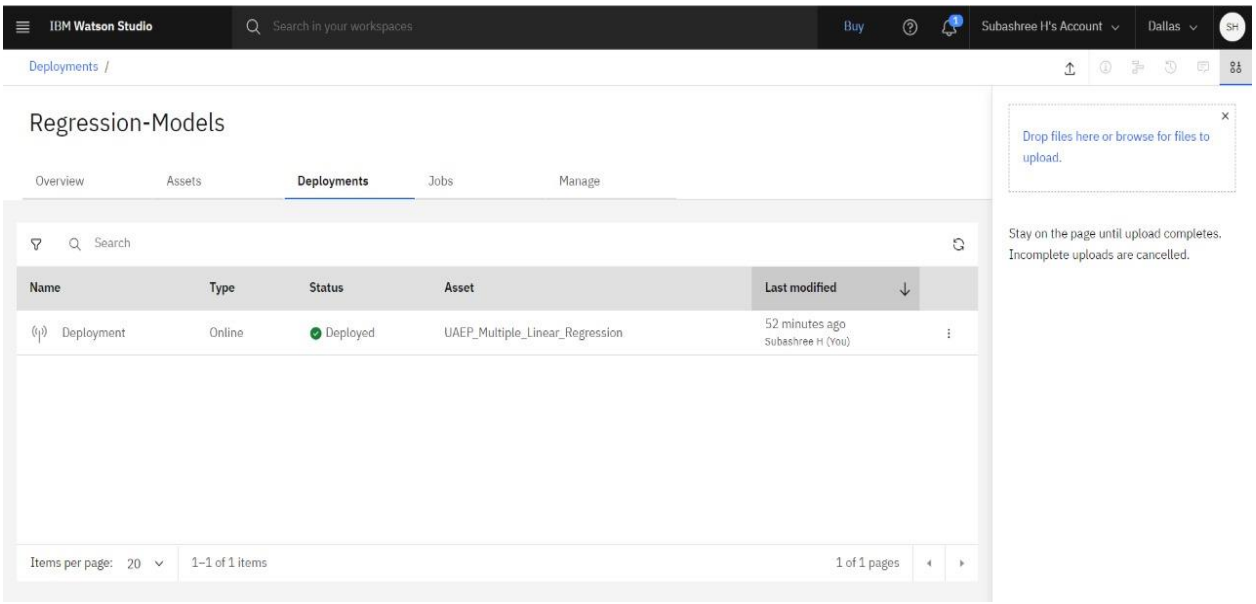
```
Space UID = fccf9bb0-2d96-46c9-a5f0-c7026edf7ad3
```

```
client.set.default_space(space_uid)
```

```
]: 'SUCCESS'
```

Assets:

Deployments:



iii) 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 = "CB8dTF8Y0_o4amdiEL2FfNBLZ-EdGUGdLqrDk7Z2gzDP"
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"]], "values": [[326, 110, 2, 3.5, 4, 9.23]]}]}

response_scoring = requests.post('https://us-south.ml.cloud.ibm.com/ml/v4/deployments/f68fb233-7af6-473e-8e77-0b1ca625695f/predictions?version=2022-11-18', json=payload_scoring
 headers={'Authorization': 'Bearer ' + mltoken})
print("Scoring response")
print(response_scoring.json())
```

```
Scoring response
{'predictions': [{'fields': ['prediction'], 'values': [[[0.8274874358563875]]]}]}
```