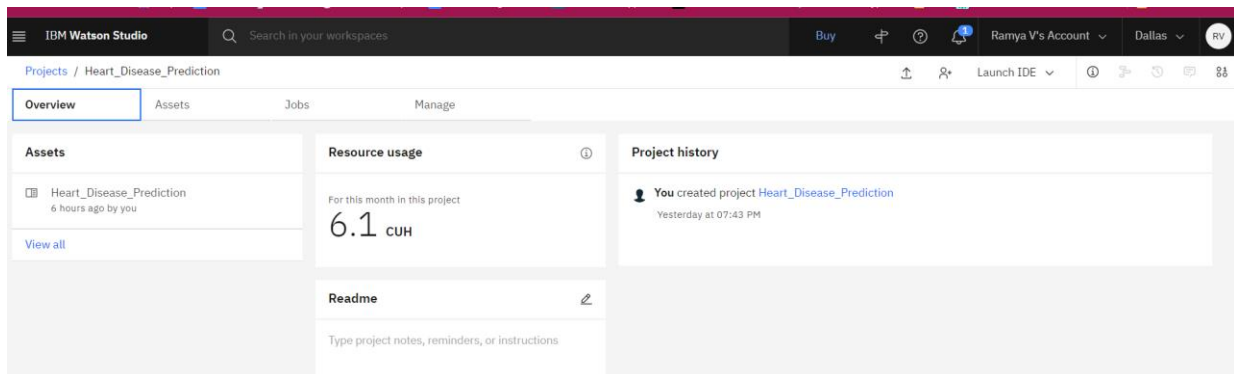


## Sprint 4

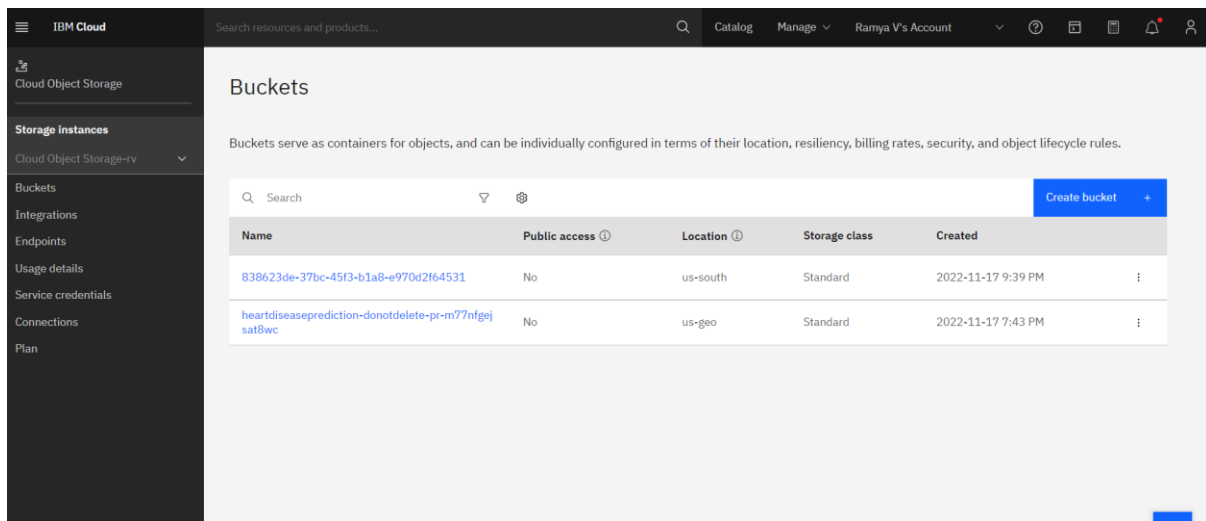
### Heart Disease prediction Using ML Model in IBM Cloud

Date	18 November 2022
Team ID	PNT2022TMID52957
Project Name	Visualizing and Predicting Heart Diseases with an Interactive Dashboard

#### Step 1: Create Watson Studio project



#### Step 2: Create Cloud Object Storage



### Step 3: From project Upload Dataset and Machine Learning Model

Projects / Heart\_Disease\_Prediction

Overview Assets Jobs Manage

Find assets Import assets New asset +

2 assets

All assets

Name	Last modified
Heart_Disease_Prediction Notebook	6 hours ago Modified by you
Heart_Disease_Prediction.csv CSV	21 hours ago Modified by you

Asset types

- Data (1)
- Data assets (1)
- Notebooks (1)

Data in this project

Drop data files here or browse for files to upload

### Step 4: Create Watson Machine Learning Account

Resource list / Watson Machine Learning-bw Active cpdaas

Details Actions...

Manage Plan Connections

Watson Machine Learning in Cloud Pak for Data

Use Watson Machine Learning on Cloud Pak for Data to put AI models to work. Deploy, monitor, and update models to get the insights you need from your data modeling.

Launch in IBM Cloud Pak for Data

IBM Watson Machine Learning in Cloud Pak for Data

IBM Cloud Pak for Data Unifying platform

IBM Cloud Base cloud infrastructure

IBM Watson Machine Learning is part of IBM Cloud Pak for Data and serves as the data science capability of the data fabric architecture.

Helpful links

- Documentation: Learn about the tools and capabilities you need to run, monitor, and update your AI assets.
- Learning path: Check out sample projects, notebooks, and data sets to help you be productive.
- Videos: Watch videos to learn about Watson Machine Learning and Cloud Pak for Data as a Service.

### Step 5: Train ML Model in IBM Cloud

#### Step 1: Import Libraries and Upload data in IBM Watson Studio And Connect it to ML Model

IBM Watson Studio

Search in your workspaces

Buy

Ramya V's Account Dallas RV

Projects / Heart\_Disease\_Prediction / Heart\_Disease\_Prediction

File Edit View Insert Cell Kernel Help

Not Trusted | Python 3.9

```
In [2]: import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression

In [3]: 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='im-E4U4KHeH1ssj0KvUEI7DAjMqTGL65VC6ntak9x9',
    ibm_auth_endpoint='https://iam.cloud.ibm.com/oidc/token',
    config=Config(signature_version='oauth'),
    endpoint_url='https://s3.private.us.cloud-object-storage.appdomain.cloud')

bucket = 'heartdiseaseprediction-donotdelete-pr-m77nfgesat8wc'
object_key = 'Heart_Disease_Prediction.csv'

body = cos_client.get_object(Bucket=bucket,Key=object_key)['Body']
# add missing __iter__ method, so pandas accepts body as file-like object
```

Projects / Heart\_Disease\_Prediction / Heart\_Disease\_Prediction

File Edit View Insert Cell Kernel Help Not Trusted | Python 3.8

```
# @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='1M-E4U4KHeH11ssj0KvuE17DAjMgTGL65VC6ntak9x9',
                              ibm_auth_endpoint='https://iam.cloud.ibm.com/oidc/token',
                              config=Config(signature_version='oauth'),
                              endpoint_url='https://s3.private.us.cloud-object-storage.appdomain.cloud')

bucket = 'heartdiseaseprediction-donotdelete-pr-m77nfgejsat8wc'
object_key = 'Heart_Disease_Prediction.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(lambda __iter__: body)

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

Out[3]:

	Age	Sex	Chest pain type	BP	Cholesterol	FBS over 120	EKG results	Max HR	Exercise angina	ST depression	Slope of ST	Number of vessels fluro	Thallium	Heart Disease
0	70	1	4	130	322	0	2	109	0	2.4	2	3	3	Presence
1	67	0	3	115	564	0	2	160	0	1.6	2	0	7	Absence
2	57	1	2	124	261	0	0	141	0	0.3	1	0	7	Presence
3	64	1	4	128	263	0	0	105	1	0.2	2	1	7	Absence

In [4]: data

Out[4]:

	Age	Sex	Chest pain type	BP	Cholesterol	FBS over 120	EKG results	Max HR	Exercise angina	ST depression	Slope of ST	Number of vessels fluro	Thallium	Heart Disease
0	70	1	4	130	322	0	2	109	0	2.4	2	3	3	Presence
1	67	0	3	115	564	0	2	160	0	1.6	2	0	7	Absence
2	57	1	2	124	261	0	0	141	0	0.3	1	0	7	Presence
3	64	1	4	128	263	0	0	105	1	0.2	2	1	7	Absence
4	74	0	2	120	269	0	2	121	1	0.2	1	1	3	Absence
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
265	52	1	3	172	199	1	0	162	0	0.5	1	0	7	Absence
266	44	1	2	120	263	0	0	173	0	0.0	1	0	7	Absence
267	56	0	2	140	294	0	2	153	0	1.3	2	0	3	Absence
268	57	1	4	140	192	0	0	148	0	0.4	2	0	6	Absence
269	67	1	4	160	286	0	2	108	1	1.5	2	3	3	Presence

270 rows x 14 columns

In [29]: X= data[['Age','Sex','Chest pain type','BP','Cholesterol','FBS over 120','EKG results','Max HR','Exercise angina','ST depression','Slope of ST','Number of vessels fluro','Thallium']].to\_numpy()  
Y= data['Heart Disease'].to\_numpy()

## Step 2: Train and Test ML Model and Check For Accuracy

IBM Watson Studio Search in your workspaces Buy Remya V's Account Dallas RV

Projects / Heart\_Disease\_Prediction / Heart\_Disease\_Prediction

File Edit View Insert Cell Kernel Help Not Trusted | Python 3.9

```
In [31]: X_train,X_test,Y_train,Y_test=train_test_split(X,Y,test_size=0.05)

In [32]: X_train.shape
Out[32]: (256, 13)

In [33]: log_reg= LogisticRegression()
log_reg.fit(X_train,Y_train)
Y_pred=log_reg.predict(X_test)

In [34]: from sklearn.metrics import accuracy_score
print ("Accuracy : ", accuracy_score(Y_test,Y_pred))

Accuracy : 0.9285714285714286

In [35]: print (X_test)
print (Y_pred)

[[ 43.  1.  4. 110. 211.  0.  0. 161.  0.  0.  1.  0.
  65.  1.  4. 110. 248.  0.  2. 158.  0.  0.6  1.  2.
  6.   ]
 [ 64.  1.  4. 145. 212.  0.  2. 132.  0.  2.  2.  2.
  6.   ]
 [ 70.  1.  2. 156. 245.  0.  2. 143.  0.  0.  1.  0.
  3.   ]
 [ 54.  1.  4. 124. 266.  0.  2. 109.  1.  2.2  2.  1.
  7.   ]]
```

## Step 3: Check with varies sample data

```
IBM Watson Studio  Search in your workspaces Buy Ramya V's Account Dallas RV

Projects / Heart_Disease_Prediction / Heart_Disease_Prediction

File Edit View Insert Cell Kernel Help Not Trusted Python 3.9

[ 6. ]
[ 58.  0.  3. 120. 340.  0.  0. 172.  0.  0.  1.  0.
  3. ]
[ 54.  0.  3. 135. 304.  1.  0. 170.  0.  0.  1.  0.
  3. ]]
['Absence' 'Presence' 'Presence' 'Absence' 'Presence' 'Absence' 'Absence'
 'Absence' 'Absence' 'Absence' 'Absence' 'Presence' 'Absence' 'Absence']

In [38]: heart_disease={'Age':[50], 'Sex':[1], 'Chest pain type':[3], 'BP':[120], 'Cholesterol':[170], 'FBS over 120':[0], 'EKG results':[0], 'Max HR':[130], 'Exercise angina':[0], 'ST depression':
        'thallium':[7]}
df2 = pd.DataFrame(heart_disease, columns= ['Age', 'Sex', 'Chest pain type', 'BP', 'Cholesterol', 'FBS over 120', 'EKG results', 'Max HR', 'Exercise angina', 'ST depression', 'Slope of ST
        'thallium']).to_numpy()
y_pred=log_reg.predict(df2)
print(y_pred)

['Absence']

In [37]: !pip install ibm_watson_machine_learning

Requirement already satisfied: ibm_watson_machine_learning in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (1.0.257)
Requirement already satisfied: importlib-metadata in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from ibm_watson_machine_learning) (4.8.2)
Requirement already satisfied: requests in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from ibm_watson_machine_learning) (2.26.0)
Requirement already satisfied: certifi in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from ibm_watson_machine_learning) (2022.9.24)
Requirement already satisfied: packaging in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from ibm_watson_machine_learning) (21.3)
Requirement already satisfied: tabulate in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from ibm_watson_machine_learning) (0.8.9)
Requirement already satisfied: pandas<1.5.0,>=0.24.2 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from ibm_watson_machine_learning) (1.3.4)
Requirement already satisfied: urllib3 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from ibm_watson_machine_learning) (1.26.7)
Requirement already satisfied: lxml in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from ibm_watson_machine_learning) (0.3.3)
Requirement already satisfied: ibm-cos-sdk==2.11.* in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from ibm_watson_machine_learning) (2.11.0)
Requirement already satisfied: ibm-cos-sdk-core==3.11.0 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from ibm_watson_machine_learning) (3.11.0)
```

## Step 6: Connecting a Machine Learning Service as an API Client

### Step 1: Create API Key in IBM Cloud

```
Requirement already satisfied: pyparsing<=3.0.5,>=2.0.2 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from packaging->ibm_watson_machine_learning) (3.0.4)

In [39]: from ibm_watson_machine_learning import APIClient
wml_credentials={
    "url":"https://us-south.ml.cloud.ibm.com",
    "apikey":"YgTKlJaxJfw0Buhk6_FCGdCAATzUwBHBznlerI9fJ-KB"
}
client=APIClient(wml_credentials)

In [40]: 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'])

In [41]: space_uid=guid_from_space_name(client,'Heart Diseases Prediction Model')
print("Space UID="+space_uid)

Space UID=d18c6095-1931-4722-9c35-d10688009ffa

In [42]: client.set.default_space(space_uid)

Out[42]: 'SUCCESS'

In [43]: client.software_specifications.list()

-----
```

### Step 2: Deploy Model

```
Requirement already satisfied: pyparsing<=3.0.5,>=2.0.2 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from packaging->ibm_watson_machine_learning) (3.0.4)

In [39]: from ibm_watson_machine_learning import APIClient
wml_credentials={
    "url":"https://us-south.ml.cloud.ibm.com",
    "apikey":"YgTKlJaxJfw0Buhk6_FCGdCAATzUwBHBznlerI9fJ-KB"
}
client=APIClient(wml_credentials)

In [40]: 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'])

In [41]: space_uid=guid_from_space_name(client,'Heart Diseases Prediction Model')
print("Space UID="+space_uid)

Space UID=d18c6095-1931-4722-9c35-d10688009ffa

In [42]: client.set.default_space(space_uid)

Out[42]: 'SUCCESS'

In [43]: client.software_specifications.list()

-----
```

## Step 7: Create Deployment Space Where Model will be deployed

```
IBM Watson Studio Search in your workspaces Buy Ramya V's Account Dallas RV

Projects / Heart_Disease_Prediction / Heart_Disease_Prediction

File Edit View Insert Cell Kernel Help Not Trusted | Python 3.9

Requirement already satisfied: py parsing>=3.0.5,>=2.0.2 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from packaging->ibm_watson_machine_learning) (3.0.4)

In [39]: from ibm_watson_machine_learning import APIClient
wml_credentials={
    "url": "https://us-south.ml.cloud.ibm.com",
    "apikey": "YgTKlJaxJfW0Buhk6_FC6dcAATzuWBNBzn1eRI9fJ-KB"
}
client=APIClient(wml_credentials)

In [40]: 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'])

In [41]: space_uid=guid_from_space_name(client,'Heart Diseases Prediction Model')
print("Space UID="+space_uid)
Space UID=d18c6095-1931-4722-9c35-d10688009ffa

In [42]: client.set.default_space(space_uid)
Out[42]: 'SUCCESS'

In [43]: client.software_specifications.list()

-----
NAME ASSET_ID TYPE
default_py3.6 0062b8c9-8b7d-44a0-a9b9-46c416adcbd9 base
kernel_spark3.2-scala2.12 020d69ce-7ac1-5e68-ac1a-31189867356a base
nvidia-nvml 1.3-nv3.7-rtf 069ea13d-33d6-57d8-h513-49120e15d788 base
```

## Step 8: Create Deployment Space as Heart Diseases Prediction Model

```
IBM Watson Studio Search in your workspaces Buy Ramya V's Account Dallas RV

Projects / Heart_Disease_Prediction / Heart_Disease_Prediction

File Edit View Insert Cell Kernel Help Not Trusted | Python 3.9

Requirement already satisfied: py parsing>=3.0.5,>=2.0.2 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from packaging->ibm_watson_machine_learning) (3.0.4)

In [39]: from ibm_watson_machine_learning import APIClient
wml_credentials={
    "url": "https://us-south.ml.cloud.ibm.com",
    "apikey": "YgTKlJaxJfW0Buhk6_FC6dcAATzuWBNBzn1eRI9fJ-KB"
}
client=APIClient(wml_credentials)

In [40]: 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'])

In [41]: space_uid=guid_from_space_name(client,'Heart Diseases Prediction Model')
print("Space UID="+space_uid)
Space UID=d18c6095-1931-4722-9c35-d10688009ffa

In [42]: client.set.default_space(space_uid)
Out[42]: 'SUCCESS'

In [43]: client.software_specifications.list()

-----
NAME ASSET_ID TYPE
default_py3.6 0062b8c9-8b7d-44a0-a9b9-46c416adcbd9 base
kernel_spark3.2-scala2.12 020d69ce-7ac1-5e68-ac1a-31189867356a base
nvidia-nvml 1.3-nv3.7-rtf 069ea13d-33d6-57d8-h513-49120e15d788 base
```

Deployments /

Heart Diseases Prediction Model

Overview Assets Deployments Jobs Manage

Assets

heart\_problem\_prediction  
6 hours ago

heart\_problem\_prediction  
6 hours ago

View all (2)

Deployments

All

1 Deployed 0 Failed

View deployments

Job runs

0 Active 0 Failed last 24 hours

View jobs

Space activity

Online deployment ready  
The online deployment newdeployment in space Heart Diseases Prediction Model is ready to accept requests  
Today at 11:32 AM

Online deployment created  
You created online deployment "newdeployment" in space Heart Diseases Prediction Model. You must wait for the deployment to  
Today at 11:32 AM

Drop files here or browse for files to upload.

Stay on the page until upload completes. Incomplete uploads are cancelled.

## Step 9: Make the Space id as default one

```
IBM Watson Studio Search in your workspaces Buy Ramya V's Account Dallas RV

Projects / Heart_Disease_Prediction / Heart_Disease_Prediction

File Edit View Insert Cell Kernel Help Not Trusted Python 3.9

In [13]: from ibm_watson_machine_learning import APIClient
        wml_credentials={
            "url": "https://us-south.ml.cloud.ibm.com",
            "apikey": "ygtK1JaX3fw0Buhk6_FCGdcAATzuwBNBzn1eR19f3-KB"
        }
        client=APIClient(wml_credentials)

In [14]: 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'])

In [15]: space_uid=guid_from_space_name(client,'Heart Diseases Prediction Model')
        print("Space UID="+space_uid)
        Space UID=d18c6095-1931-4722-9c35-d10688009ffa

In [16]: client.set.default_space(space_uid)
Out[16]: 'SUCCESS'
```

## Step 10: Using Space id Deploy an scikit-learn

```
IBM Watson Studio Search in your workspaces Buy Ramya V's Account Dallas RV

Projects / Heart_Disease_Prediction / Heart_Disease_Prediction

File Edit View Insert Cell Kernel Help Not Trusted Python 3.9

In [18]: software_spec_uid=client.software_specifications.get_uid_by_name("runtime-22.1-py3.9")
        software_spec_uid
Out[18]: '12b83a17-24d8-5082-900f-0ab31fbfd3cb'

In [19]: model_details=client.repository.store_model(model=log_reg,meta_props={
        client.repository.ModelMetaNames.NAME: "heart_problem_prediction",
        client.repository.ModelMetaNames.SOFTWARE_SPEC_UID: software_spec_uid,
        client.repository.ModelMetaNames.TYPE: "scikit-learn_1.0"
    })

In [20]: model_id = client.repository.get_model_id(model_details)

In [21]: model_id
Out[21]: 'a11a0c20-18cd-4e44-ab75-610d19c2626b'

In [22]: X_train[0]
Out[22]: array([ 53. ,  0. ,  4. , 130. , 264. ,  0. ,  2. , 143. ,  0. ,
                0.4,  2. ,  0. ,  3. ])

In [23]: log_reg.predict([[ 60. ,  1. ,  4. , 117. , 230. ,  1. ,  0. , 160. ,  1. ,
                1.4,  1. ,  2. ,  7. ]])
```

## Step 11: Test And predict the Model

```
IBM Watson Studio Search in your workspaces Buy Ramya V's Account Dallas RV

Projects / Heart_Disease_Prediction / Heart_Disease_Prediction

File Edit View Insert Cell Kernel Help Trusted Python 3.9

In [19]: model_details=client.repository.store_model(model=log_reg,meta_props={
        client.repository.ModelMetaNames.NAME: "heart_problem_prediction",
        client.repository.ModelMetaNames.SOFTWARE_SPEC_UID: software_spec_uid,
        client.repository.ModelMetaNames.TYPE: "scikit-learn_1.0"
    })

In [20]: model_id = client.repository.get_model_id(model_details)

In [21]: model_id
Out[21]: 'a11a0c20-18cd-4e44-ab75-610d19c2626b'

In [22]: X_train[0]
Out[22]: array([ 53. ,  0. ,  4. , 130. , 264. ,  0. ,  2. , 143. ,  0. ,
                0.4,  2. ,  0. ,  3. ])

In [23]: log_reg.predict([[ 60. ,  1. ,  4. , 117. , 230. ,  1. ,  0. , 160. ,  1. ,
                1.4,  1. ,  2. ,  7. ]])
Out[23]: array(['Presence'], dtype=object)

In [ ]:
```

## Step 12: Create Deployment as newdeployment

The screenshot shows the IBM Watson Studio interface. At the top, there's a navigation bar with 'IBM Watson Studio', a search bar, and user account information. Below the navigation bar, the 'Deployments' tab is selected for the 'Heart Diseases Prediction Model'. The main area displays a table with deployment details:

Name	Type	Status	Asset	Last modified
newdeployment	Online	Deployed	heart_problem_prediction	7 hours ago Ramya V (you)

On the right side, there's a notification box that says 'Drop files here or browse for files to upload.' and a message: 'Stay on the page until upload completes. Incomplete uploads are cancelled.'

The screenshot shows the 'API reference' page for the 'newdeployment' asset. The 'Direct link' section displays the endpoint: `https://us-south.ml.cloud.ibm.com/ml/v4/deployments/2102715e-0bc9-42dd-9036-6a7eaf97aed5/predictions?version=2022-11-18`. The 'Code snippets' section provides a cURL example for making a prediction request. The right sidebar shows metadata for the deployment, including creation and update times, deployment ID, software specification (runtime-22.1-py3.9), and the associated asset (heart\_problem\_prediction).

## Step 13: Heart Disease prediction in python using Deployed Model in IBM cloud

### Code

```
*test_cloud_api.py - C:\Users\SSN\OneDrive - SSN Trust\Documents\SEM 7\IBM\test_cloud_api.py (3.10.3)*
File Edit Format Run Options Window Help
import requests
import json
# NOTE: you must manually set API_KEY below using information retrieved from your IBM Cloud account.
API_KEY = "YgTKLJaXJfw0Buhk6_FCGdCAAtzUwBNBznleRIfJ-KB"
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 = [{"field": ["Age", "Sex", "Chest pain type", "BP", "Cholesterol", "FBS over 120", "EKG results", "Max HR", "Exercise angina", "ST depression", "Slope of ST",
"Number of vessels fluoro", "Thallium"], "values": [[60, 1, 4, 117, 230, 1, 0, 160, 1, 1.4, 1, 2, 7]]}]

response_scoring = requests.post('https://us-south.ml.cloud.ibm.com/ml/v4/deployments/2102715e-0bc9-42dd-9036-6a7eaf97aed5/predictions?version=2022-11-18', json=payload_scoring,
headers={'Authorization': 'Bearer ' + mltoken})
print("Scoring response")
predictions=response_scoring.json()
pred=predictions['predictions'][0]['values'][0][0]
if(pred == 'Presence'):
    print("You have high probability to heart Disease Kindly approach a Doctor Take care")
else:
    print("Hey! Your Normal Take care")
```

## Output:

```
IDLE Shell 3.10.3
File Edit Shell Debug Options Window Help
Python 3.10.3 (tags/v3.10.3:a342a49, Mar 16 2022, 13:07:40) [MSC v.1929 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
= RESTART: C:\Users\SSN\OneDrive - SSN Trust\Documents\SEM 7\IBM\test_cloud_api.py
Scoring response
You have high probability to heart Disease Kindly approach a Doctor Take care
>>>
```