**DATA WAREHOUSING WITH IBM CLOUD**

**DB2 WAREHOUSE**

**Phase 3 submission Document**

**Development Phase 1**

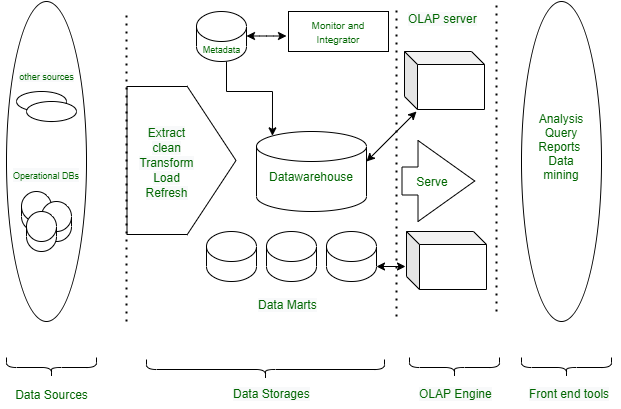
****

**Introduction:**

Data warehousing with IBM Cloud DB2 Warehouse represents a powerful and scalable solution for organizations looking to harness the full potential of their data. IBM Cloud DB2 Warehouse is a cloud-based data warehousing platform that combines the reliability and performance of IBM's DB2 database with the agility and flexibility of cloud computing. This cutting-edge technology allows businesses to efficiently store, manage, and analyze vast amounts of data, enabling data-driven decision-making and insights.

With IBM Cloud DB2 Warehouse, you can seamlessly integrate and consolidate data from various sources, transforming raw information into valuable business intelligence. This data warehousing solution offers features like high availability, data compression, and advanced analytics capabilities, making it an ideal choice for organizations seeking to improve their data storage and analytical capabilities.

In this era of data-driven decision-making, leveraging IBM Cloud DB2 Warehouse can empower businesses to unlock the true potential of their data, drive innovation, and gain a competitive edge in an increasingly digital world. Whether you are looking to optimize operational efficiency, enhance customer experiences, or uncover new revenue opportunities, data warehousing with IBM Cloud DB2 Warehouse can be a key enabler for your organization's success.

****

The **loading** process is the physical movement of the data from the computer systems storing the source database(s) to that which will store the data warehouse database. The entire process of transferring data to a data warehouse repository is referred to in the following ways:

**Initial Load:**

For the very first time loading all the data warehouse tables.

**Incremental Load:**

Periodically applying ongoing changes as per the requirement. After the data is loaded into the data warehouse database, verify the referential integrity between the dimensions and the fact tables to ensure that all records belong to the appropriate records in the other tables. The DBA must verify that each record in the fact table is related to one record in each dimension table that will be used in combination with that fact table.

**Full Refresh:**

Deleting the contents of a table and reloading it with fresh data.

**Steps**

1. **Clone the repo:**

Clone the icp4d-customer-churn-classifier repo locally. In a terminal, run the following command:

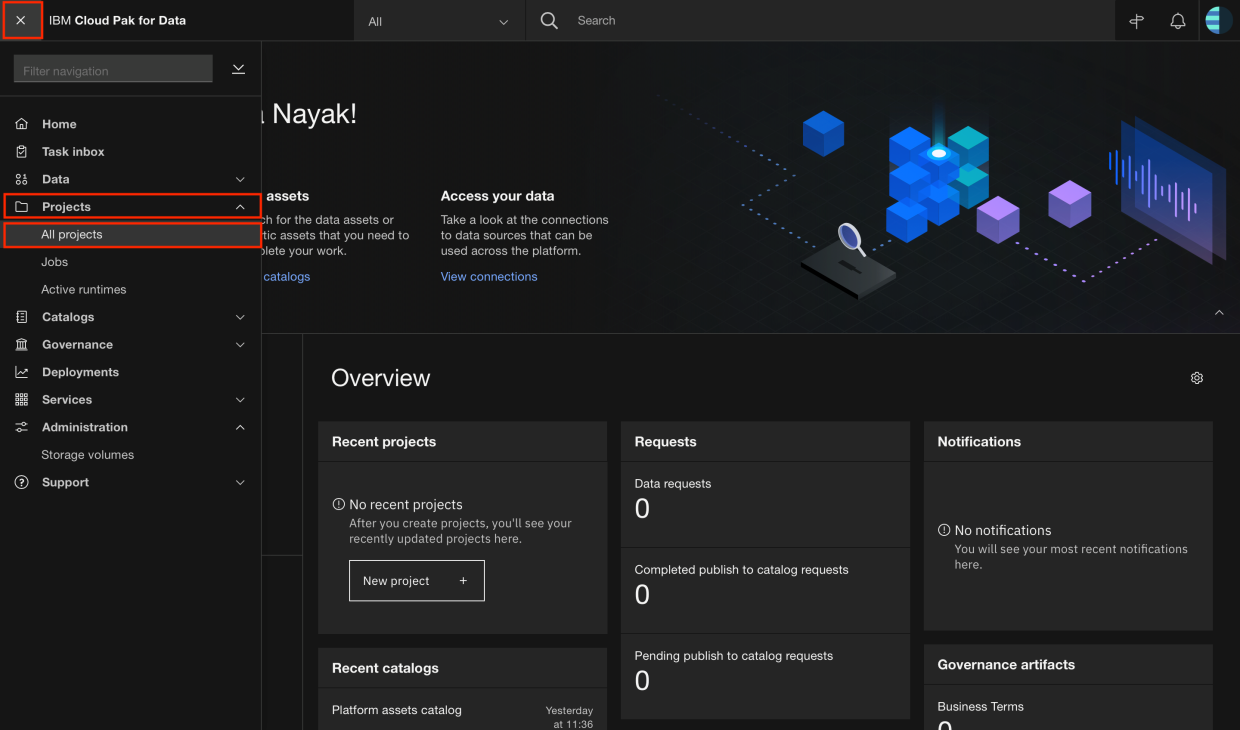
**git clone** [**https://github.com/IBM/icp4d-customer-churn-classifier**](https://github.com/IBM/icp4d-customer-churn-classifier)

1. **Set up an analytic project:**

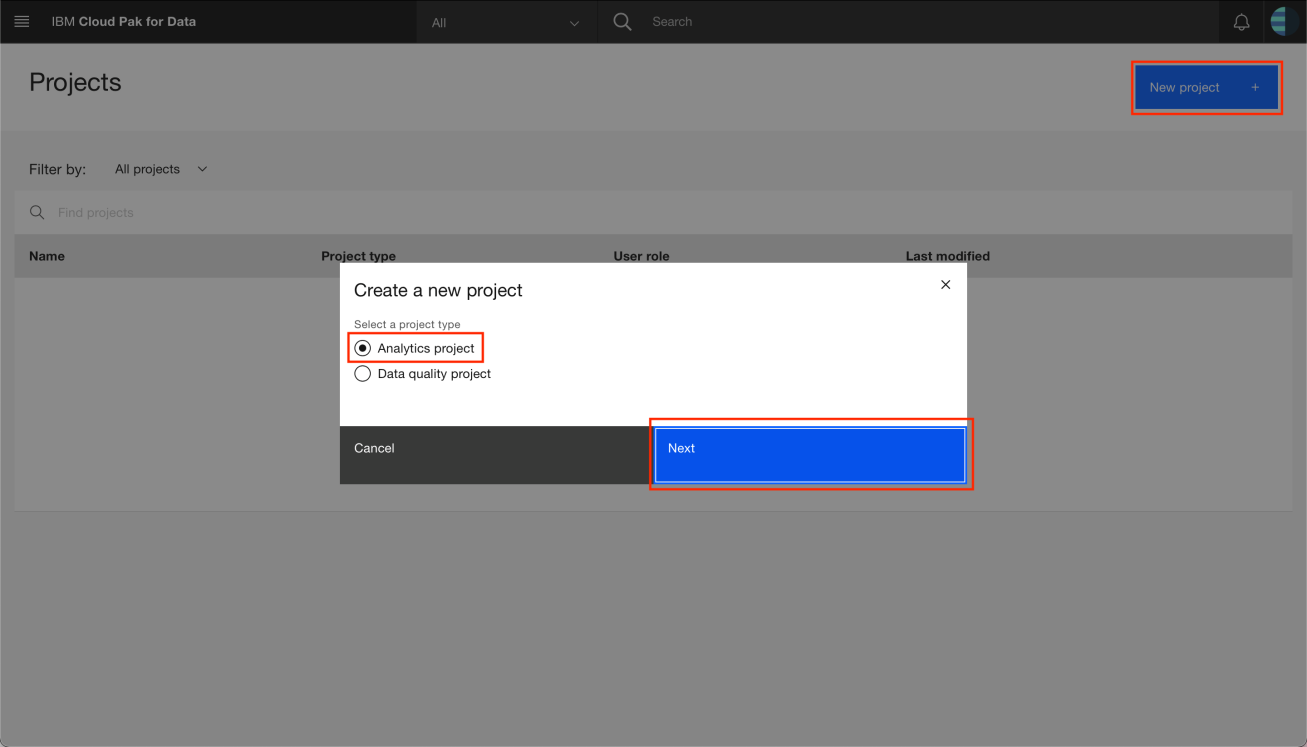
To get started, open the Projects page and set up an analytics project to hold the assets that you want to work with, and then get data for your project.

#### [Create a project](https://github.com/IBM/icp4d-customer-churn-classifier#create-a-project)

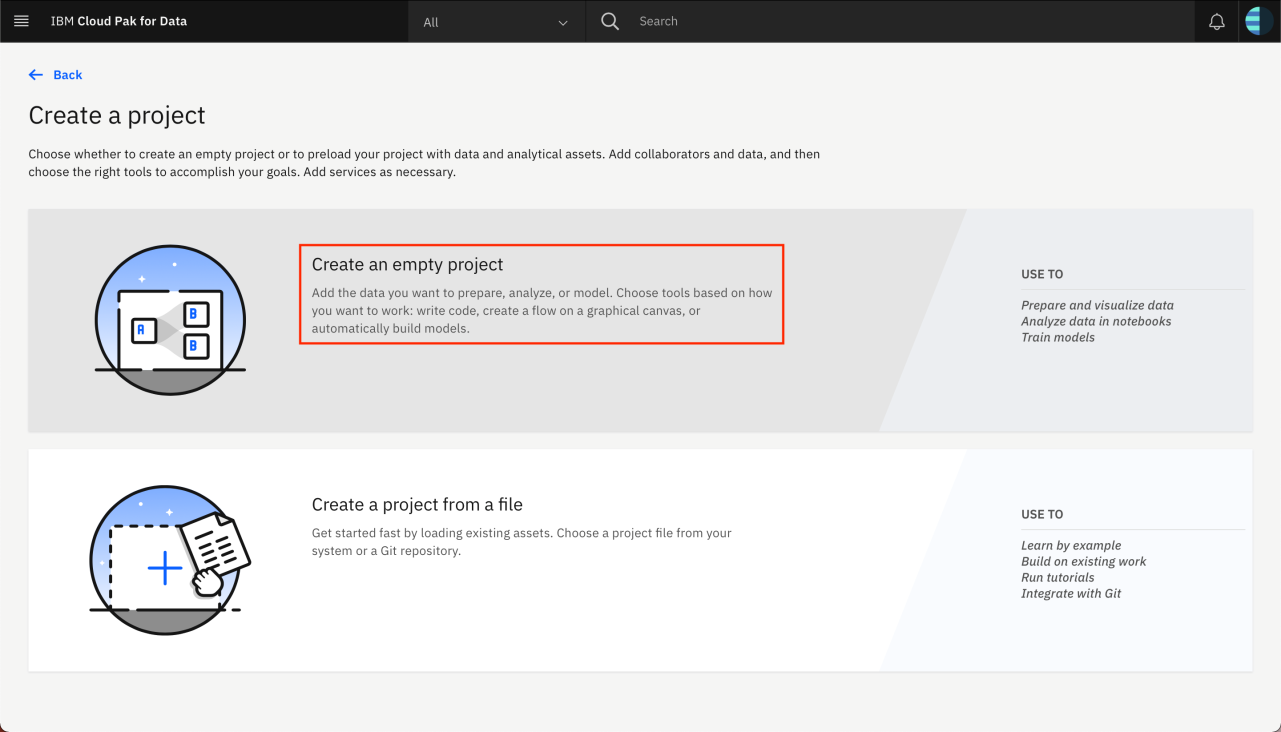
* Launch a browser and navigate to your Cloud Pak for Data deployment.
* Go to the (☰) menu, expand Projects and click All projects:

****

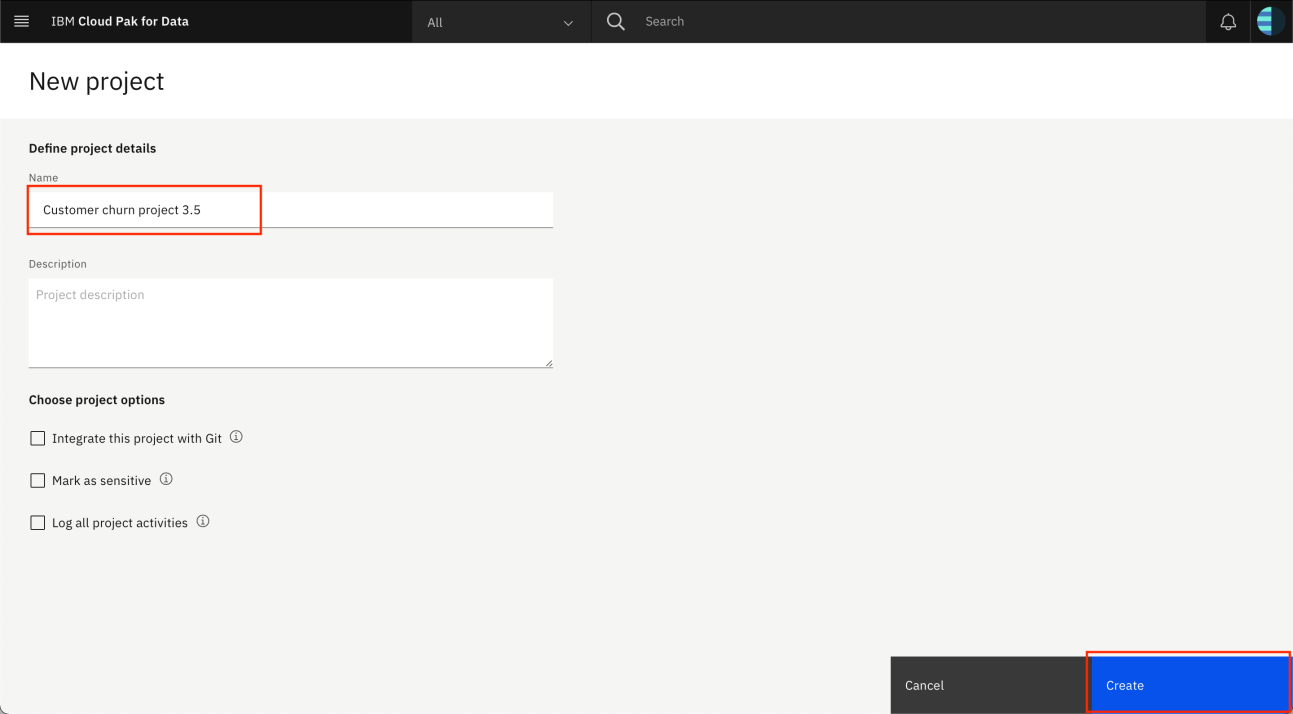
* Click on *New project +*. In the dialog that pops up, select the project type as Analytics project:



* Click on the top tile for Create an empty project:

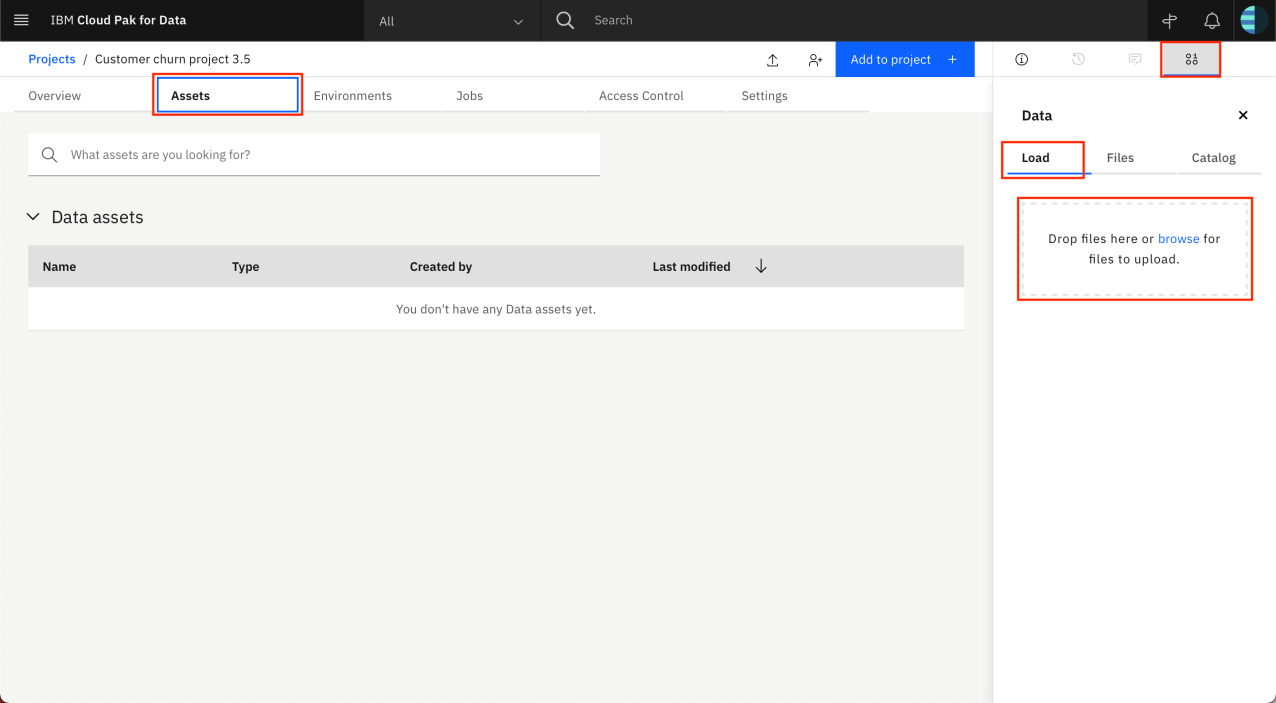
****

* Give the project a unique name, an optional description and click Create:

****

**Add the data asset:**

* In your project, on the Assets tab, click the 01/00 icon and the Load tab, then either drag the [data/mergedcustomers.csv](https://github.com/IBM/icp4d-customer-churn-classifier/blob/master/data/mergedcustomers.csv) file from the cloned repository to the window or navigate to it using browse for files to upload:

****

1. **Create a Space for Machine Learning Deployments:**

Before we create a machine learning model, we will have to set up a deployment space where we can save and deploy the model.

Follow the steps in this section to create a new deployment space.

* Navigate to the left-hand (☰) hamburger menu and choose Deployments:

1. **Insert pandas DataFrame:**

Place your cursor at the last line of the following cell:

# Use the find data 01/00 icon and under your remote data set

# use "Insert to code" and "Insert pandas DataFrame

# here.

# Add asset from file system

Click the *find data* 01/00 icon on the menu bar (last icon). On the *Files* tab, find the data set that you added to the project, click Insert to code and pandas DataFrame.

1. **Initialize Watson Machine Learning client:**

The Watson Machine Learning client is required to save and deploy our customer churn predictive model, and should be available on your IBM Cloud Pak for Data platform. Find the cell containing the code given below and insert the url, username and password for your IBM Cloud Pak for Data instance:

from ibm\_watson\_machine\_learning import APIClient

# get URL, username and password from your IBM Cloud Pak for Data administrator

wml\_credentials = {

"url": "https://X.X.X.X",

"username": "\*\*\*\*\*",

"password": "\*\*\*\*\*",

"instance\_id": "wml\_local",

"version" : "3.5"

}

client = APIClient(wml\_credentials)

print(client.version)

1. **Provide the deployment space information:**

IBM Cloud Pak for Data uses the concept of deployment spaces, which is where models can be deployed. You can list all the spaces using the .list() function.

Provide the name of the deployment space that you created in [Step 3](https://github.com/IBM/icp4d-customer-churn-classifier#3-create-a-space-for-machine-learning-deployments) above in the cell containing the following text.

#Insert the name of your deployment space here:

DEPLOYMENT\_SPACE\_NAME = 'INSERT-YOUR-DEPLOYMENT-SPACE-NAME-HERE'

Once you know the deployment space id, update the next cell with this id to set this deployment space as the default deployment space. Further down the notebook, when you deploy the model, it will be deployed to this default deployment space.

# Now set the default space to the GUID for your deployment space. If this is successful, you will see a 'SUCCESS' message.

client.set.default\_space('INSERT\_SPACE\_ID\_HERE')

1. **Run the notebook:**

Run the entire notebook using the menu Cell ▷ Run All or run the cells individually with the play button as shown here.

1. **Analyze the result:**

* A pod was instantiated – which means loading a complete compute Jupyter notebook environment (7+ GB) with all the artifacts from the ICP4D registry.
* This pod is scheduled on any VM in your cluster – wherever CPU and memory resources are available.
* IP addresses and connections are all configured automatically.
* The same working environment can be used by multiple users. If a single pod's resources are not sufficient, another environment is created automatically.
* When the number of users grow, you can add more machines to the ICP4D cluster and scheduling of resources is handled automatically.
* ICP4D's scale-out model is pretty effective.
* You no longer have to wait days or even weeks to get the compute resources.
* More users can be accommodated with same compute capacity. As one task completes, its resources are freed up to work on next one.

1. **Test the model:**

IBM Cloud Pak for Data provides various options for analytics models such as testing, scoring, evaluating, and publishing.

We can start testing using the built-in tooling.

* Navigate to the left-hand (☰) hamburger menu and choose Deployments:
* To test the model by providing data using the form, click on the Provide input using form icon and enter the following input in the form fields:
  + ID: *4*
  + GENDER: *F*
  + STATUS: *M*
  + CHILDREN: *2*
  + ESTINCOME: *52004*
  + HOMEOWNER: *N*
  + AGE: *25*
  + TOTALDOLLARVALUETRADED: *5030*
  + TOTALUNITSTRADED: *23*,
  + LARGESTSINGLETRANSACTION: *1257*
  + SMALLESTSINGLETRANSACTION: *125*
  + PERCENTCHANGECALCULATION: *3*
  + DAYSSINCELASTLOGIN: *2*
  + DAYSSINCELASTTRADE: *19*
  + NETREALIZEDGAINS\_YTD: *0*
  + NETREALIZEDLOSSES\_YTD: *251*
* Click the Predict button and the model will be called with the input data. The results will display in the *Result* window. Scroll down to the bottom (Line #110) to see either a "High", a "Low" or a "Medium" for Churn:
* To test the model by providing an input JSON, click on the Provide input data as JSON icon and paste the following data under Body:

{

"input\_data":[

{

"fields":[

"ID",

"GENDER",

"STATUS",

"CHILDREN",

"ESTINCOME",

"HOMEOWNER",

"AGE",

"TOTALDOLLARVALUETRADED",

"TOTALUNITSTRADED",

"LARGESTSINGLETRANSACTION",

"SMALLESTSINGLETRANSACTION",

"PERCENTCHANGECALCULATION",

"DAYSSINCELASTLOGIN",

"DAYSSINCELASTTRADE",

"NETREALIZEDGAINS\_YTD",

"NETREALIZEDLOSSES\_YTD"

],

"values":[

[

4,

"F",

"M",

2,

52004,

"N",

60,

5030,

23,

1257,

125,

3,

1,

1,

1000,

0

]

]

}

]

}

* In a terminal window (or command prompt in Windows), run the following command to get a token to access the API. Use your Cloud Pak for Data cluster URL, username and password:

**curl -k -X GET https://<cluster-url>/v1/preauth/validateAuth -u <username>:<password>**

* Use the export command to save the "accessToken" part of this response in the terminal window to a variable called WML\_AUTH\_TOKEN.

export WML\_AUTH\_TOKEN=<value-of-access-token>

{"predictions":[{"fields":["ID","GENDER","STATUS","CHILDREN","ESTINCOME","HOMEOWNER","AGE","TOTALDOLLARVALUETRADED","TOTALUNITSTRADED","LARGESTSINGLETRANSACTION","SMALLESTSINGLETRANSACTION","PERCENTCHANGECALCULATION","DAYSSINCELASTLOGIN","DAYSSINCELASTTRADE","NETREALIZEDGAINS\_YTD","NETREALIZEDLOSSES\_YTD","GENDERIndex","GENDERclassVec","STATUSIndex","STATUSclassVec","HOMEOWNERIndex","HOMEOWNERclassVec","features","rawPrediction","probability","prediction","predictedLabel"],"values":[[4,"F","M",2,52004,"N",60,5030,23,1257,125,3,1,1,1000,0,0.0,[1,[0],[1.0]],0.0,[2,[0],[1.0]],0.0,[1,[0],[1.0]],[1.0,1.0,0.0,1.0,4.0,2.0,52004.0,60.0,5030.0,23.0,1257.0,125.0,3.0,1.0,1.0,1000.0,0.0],[2.9466019417475726,8.67282872405483,8.380569334197599],[0.14733009708737863,0.4336414362027415,0.4190284667098799],1.0,"Low"]]}]}

A json string similar to the one below will be returned with the response, including a "High", a "Low" or a "Medium" at the end indicating the risk of churn for this customer.

**Data Warehouse Structure:**

--Examples SQL script for creating tables in Db2 Warehouse

CREATE TABLE customers (

Customer\_id INT PRIMARY KEY,

Customer\_name VARCHAR(100),

Email VARCHAR(100),\

);

CREATE TEBLE orders (

order\_id INT PRIMARY KEY,

customer\_id INT,

order\_data DATE,

total\_amuont DECIMAL(10,2)

);

**CONCLUSION:**

In this solution, we successfully designed and set up a robust data warehouse using IBM Cloud Db2 Warehouse. By following the problem definition, design thinking, development, and documentation phases, we designed a data warehouse structure, integrated data from various sources, performed ETL processes, and enabled data exploration. The solution empowers data architects to explore, analyze, and deliver actionable data for informed decision- making, contributing to unlocking valuable business insights and driving informed decisions.