

Lab Center – Hands-on Lab

Session 136

**Session Title Hands-On Lab: Make Your Model
More Precise by Virtualizing Data Sources and
Transparentize It by Watson OpenScale**

(Latest PDF will be available in the public github repository :

https://github.com/vishkamat/DAF_136)

Qi Jun Wang, IBM, wangqij@us.ibm.com

Vishwanath Kamat, IBM, vkamat@us.ibm.com

Disclaimer

IBM's statements regarding its plans, directions, and intent are subject to change or withdrawal without notice at IBM's sole discretion. Information regarding potential future products is intended to outline our general product direction and it should not be relied on in making a purchasing decision.

The information mentioned regarding potential future products is not a commitment, promise, or legal obligation to deliver any material, code or functionality. Information about potential future products may not be incorporated into any contract.

The development, release, and timing of any future features or functionality described for our products remains at our sole discretion I/O configuration, the storage configuration, and the workload processed. Therefore, no assurance can be given that an individual user will achieve results like those stated here.

Information in these presentations (including information relating to products that have not yet been announced by IBM) has been reviewed for accuracy as of the date of initial publication and could include unintentional technical or typographical errors. IBM shall have no responsibility to update this information. **This document is distributed "as is" without any warranty, either express or implied. In no event, shall IBM be liable for any damage arising from the use of this information, including but not limited to, loss of data, business interruption, loss of profit or loss of opportunity.** IBM products and services are warranted per the terms and conditions of the agreements under which they are provided.

IBM products are manufactured from new parts or new and used parts.

In some cases, a product may not be new and may have been previously installed. Regardless, our warranty terms apply."

Any statements regarding IBM's future direction, intent or product plans are subject to change or withdrawal without notice.

Performance data contained herein was generally obtained in controlled, isolated environments. Customer examples are presented as illustrations of how those customers have used IBM products and the results they may have achieved. Actual performance, cost, savings or other results in other operating environments may vary.

References in this document to IBM products, programs, or services does not imply that IBM intends to make such products, programs or services available in all countries in which IBM operates or does business.

Workshops, sessions and associated materials may have been prepared by independent session speakers, and do not necessarily reflect the views of IBM. All materials and discussions are provided for informational purposes only, and are neither intended to, nor shall constitute legal or other guidance or advice to any individual participant or their specific situation.

It is the customer's responsibility to insure its own compliance with legal requirements and to obtain advice of competent legal counsel as to the identification and interpretation of any relevant laws and regulatory requirements that may affect the customer's business and any actions the customer may need to take to comply with such laws. IBM does not provide legal advice or represent or warrant that its services or products will ensure that the customer follows any law.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products about this publication and cannot confirm the accuracy of performance,

compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products. IBM does not warrant the quality of any third-party products, or the ability of any such third-party products to interoperate with IBM's products. **IBM expressly disclaims all warranties, expressed or implied, including but not limited to, the implied warranties of merchantability and fitness for a purpose.**

The provision of the information contained herein is not intended to, and does not, grant any right or license under any IBM patents, copyrights, trademarks or other intellectual property right.

IBM, the IBM logo, ibm.com and [names of other referenced IBM products and services used in the presentation] are trademarks of International Business Machines Corporation, registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of IBM trademarks is available on the Web at "Copyright and trademark information" at: www.ibm.com/legal/copytrade.shtml.

© 2019 International Business Machines Corporation. No part of this document may be reproduced or transmitted in any form without written permission from IBM.

U.S. Government Users Restricted Rights — use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM.

Table of Contents

Disclaimer	2
Introduction.....	6
Section I	8
Build virtualized data assets.....	8
Login to Cloud Pak for Data.....	8
Section II.....	12
Build a machine learning model using jupyter notebook	12
Pre-requisites	12
Working with Analytics Project	12
Section III.....	16
Setup Watson OpenScale to monitor and manage ML model	16
Section IV	24
Viewing Monitoring Model Results in Dashboard	24

Introduction

In this lab you will learn how build machine learning models using data sets that are stored in different repositories through Data Virtualization features available in IBM Cloud Pak for Data platform build on containers. The lab will also showcase how to make a machine learning model lifecycle management using Watson OpenScale feature also part of Cloud Pak for Data platform.

The lab is divided in 5 sections:

Section I – Build virtualized data assets that will be used in ML Model development process

- In this section you will learn a self-service data access process where data scientists can request for data sets that are not available readily for a given data science project.
- Data Engineer can act on a data request and fulfill using a Data Virtualization feature and assign to a analytics project

Section II – Build a machine learning model using jupyter notebook

- Work with data sets made available in previous section (or use one that is already exists within your project)
- Train and test the model
- Finally deploy the model to be used as web service

Section III– Deploy ML model in Watson Machine Learning environment and create REST Endpoint for scoring

- Setup WML deployment using jupyter notebook
- Test run model to verify REST endpoint

Section IV– Setup Watson OpenScale to monitor and manage ML model

- Setup OpenScale configuration for deployed ML model
- Run test scoring requests so that OpenScale can analyze model and payload
- View various reports and management matrix about deployed ML model

Section V-Setup Watson OpenScale to monitor and manage ML model

- Setup OpenScale configuration for deployed ML model
- Setup model monitoring details

Section VI-Run test scoring requests so that OpenScale can analyze model and payload

- Run test scoring requests so that OpenScale can analyze model and payload
- Save scoring test result to db2

Section VII-Viewing Monitoring Results in Dashboard

- Check how to monitor results

Section I

Build virtualized data assets

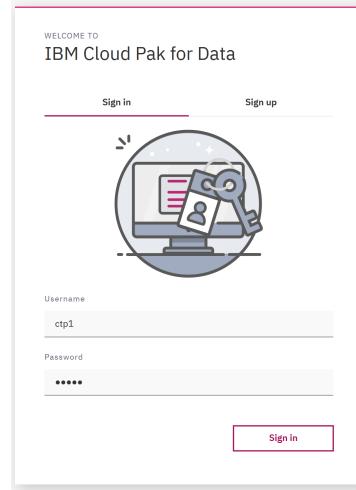
In this section we will :

- In this section you will learn a self-service data access process where data scientists can request for data sets that are not available readily for a given data science project.
- Data Engineer can act on a data request and fulfill using a Data Virtualization feature and assign to a analytics project

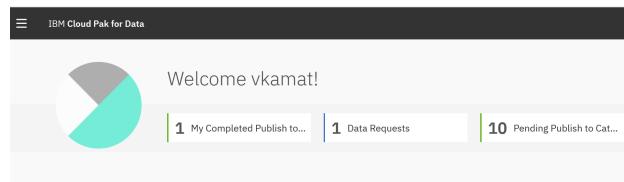
NOTE : Instructor will login as data scientist user and request for new data request. You will be acting on the request in the queue.

Login to Cloud Pak for Data

- a. Login using your given userid/password in Cloud Pak for Data portal for e.g. :
<https://services-uscentral.skytap.com:10068>



- b. The request will be available for all Data Engineer users within Cloud Pak for Data. Click on “Go to your home page” option on top right corner of the screen. This will show all items that need your attention. As shown below, you will see a “1 Data Request” :



- c. Click on Data Requests. This will show details of the request as sent by data scientist user. You can click as assign the request to yourself, so that other “data engineers” can focus on other requests that may be in the queue. Click on top

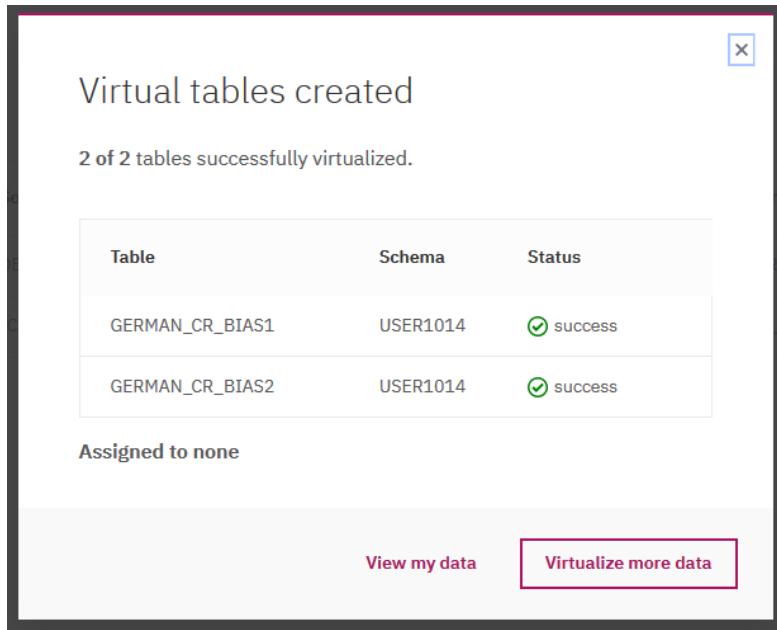
right “Action” drop-down menu. (if request is already assigned you will see “Unassign” option in the menu.)

The screenshot shows the 'Data request' screen in the IBM Cloud Pak for Data interface. A data request named 'ctp1_customer' is selected. In the top right corner, there is a vertical action dropdown menu with options: 'Virtualize data' (highlighted), 'Transform data', 'Virtualize data', and 'Unassign'. The main panel displays details about the request, including the requester ('opt1'), project ('vlpri'), status ('No issue link'), deadline ('High'), and a note: 'Need customer_usage_history and customer_demographics info. Also need these 2 tables joined for easy processing'.

- d. This will bring you back to all requests. Click on the request one more time and choose option “Virtualize” to fulfill the request. Note that data requests can be fulfilled by either creating a Transform Job or Virtualize option. Select GERMAN_CR_BIAS1 and GERMAN_CR_BIAS2 tables. Click on “Add to cart”
- e. This will enable “View cart” button on top right. Click on it. This will bring following screen with options assign this to a “Data request” or a project etc. Click on “neither” and also uncheck “Submit to catalog”. The data engineer may want to review all the data before making it available to catalog and/or to a project. Click on “Virtualize”

The screenshot shows the 'Review cart and virtualize tables' dialog box. It lists two tables: 'GERMAN_CR_BIAS1' and 'GERMAN_CR_BIAS2'. Both tables are associated with 'USER1014' schema and 'DB2INST1' source schema. The 'Hosts' column shows '10.1.1.1' and '10.1.1.1' respectively. The 'Source' column shows 'MORTGAGE' and 'icpd_mongo'. The 'Grouped tables' column shows '1' for both. At the bottom, there are radio buttons for 'Data request' (selected), 'Project' (unchecked), and 'Neither' (selected). There is also a checkbox for 'Submit to catalog' (unchecked) and a prominent 'Virtualize' button.

- f. This bring up a confirmation window showing that 2 tables have been made available as virtualized assets. Data Engineer can provide these assets as is to a “request” or assign to a project. Data Virtualization provides ways to make these assets available to users. We will further refine this dataset with a pre-defined join condition so that end-users need not worry about finding joining keys. Click on “View my data” button.



- g. This will bring back to following screen with option to define join. Click on “Join view” option.

The screenshot shows the "Data virtualization" section of the IBM Cloud Pak for Data interface. The search bar at the top has "german" typed into it. The "Available tables" section lists two tables: "GERMAN_CR_BIAS1" and "GERMAN_CR_BIAS2". To the right of the table names are their respective schemas: "DB2INST1" and "ICPD_MONGO", and databases: "MORTGAGE" and "icpd_mongo". There is also an "Add to cart" button. On the left, there are filters for "Asset type" (Schema name, Table name, Column name) and "Databases" (IBM DB2 Family (1)).

- h. This will bring the 2 tables with column definition side-by-side. Here you can select joining condition by selecting a column and dragging to corresponding column in second table. Use ID column in this example as shown below. Unselect ID and _ID (underscore) columns from the list because we are not going to use those for any further processing and click on “Join” :

The screenshot shows the 'Join virtual objects' screen in IBM Cloud Pak for Data. It displays two tables: Table 1: GERMAN_CR_BIAS1 and Table 2: GERMAN_CR_BIAS2. The left panel lists columns for Table 1, and the right panel lists columns for Table 2. A central area shows the selection of join keys: 'ID' from Table 1 is selected and mapped to 'ID' in Table 2. The interface also includes sections for 'Join Keys' and 'Filters'.

- This will bring the screen to enter view name and schema name. Create your own name (your intial_joinv) and choose a default schema name provided. For subsequent steps, there is a usable virtual asset has been already created for you, so you can use name you want. Click on “Create view” option on top .

The screenshot shows the 'Join virtual objects: Review' screen. It displays a preview of the joined data from 'View Name: ctp1_join_req' and 'Schema Name: CHURNSCH'. The preview table includes columns: AGE, CARDOWNER, CHILDREN, ESTINCOME, GENDER, ID, STATUS, CHURN, and DROPPED. The data shows several rows of customer information.

- In the next screen you can assign the view to a data request or project. There is an option “neither” that can be used if Data Engineers decide to create objects that can be later on assigned to different projects as needed. Click on “Project” and select “vkprj”. Also select “Submit to catalog” so that it is available for others to see from catalog. Click “Create view”. This will show confirmation screen as follows :

The screenshot shows the 'Join view created' confirmation screen. It displays the following details:

- View:** ctp1_join_req
- Schema:** CHURNSCH
- Assigned to project:** vkprj
- Published to catalog:** Succeeded

At the bottom, there are buttons for 'Virtualize more data', 'View my data', and 'Go to project'.

- At this point you have successfully created a virtualized asset based on request that came from a data scientist user. Selecting “Data request” before creation would also show that request was completed when requesting user logs in to Cloud Pak for Data.

Section II

Build a machine learning model using jupyter notebook

In this section we will :

Build machine learning model using Spark and Jupyter notebook

- Create a analytics project using a project export
- Review and execute notebook to build model
- Save model for further testing and evaluation

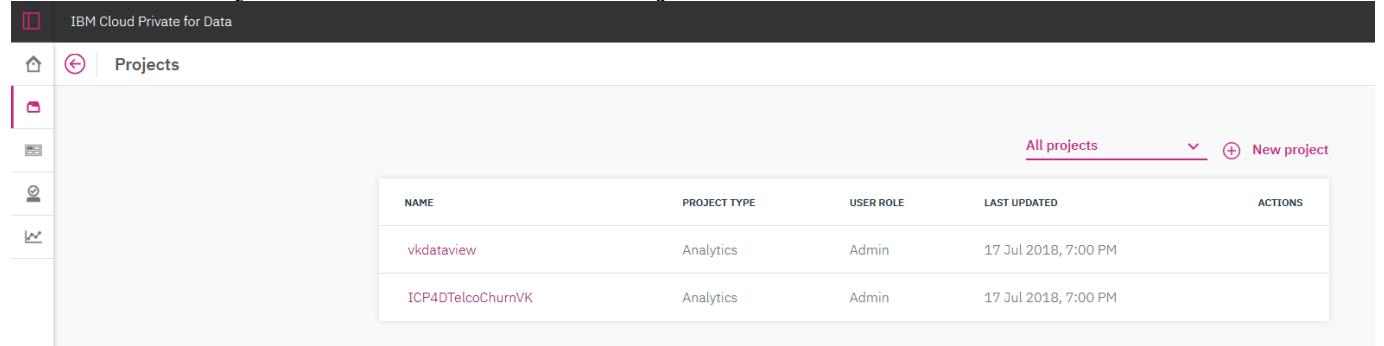
Pre-requisites

- Previously created analytics project. Download this from box folder
<https://ibm.box.com/s/09onnb3vhuglx6c3v7d93sm3n7fm7tbc>

Rename file after download to your laptop for e.g.: Lab136_VK.zip

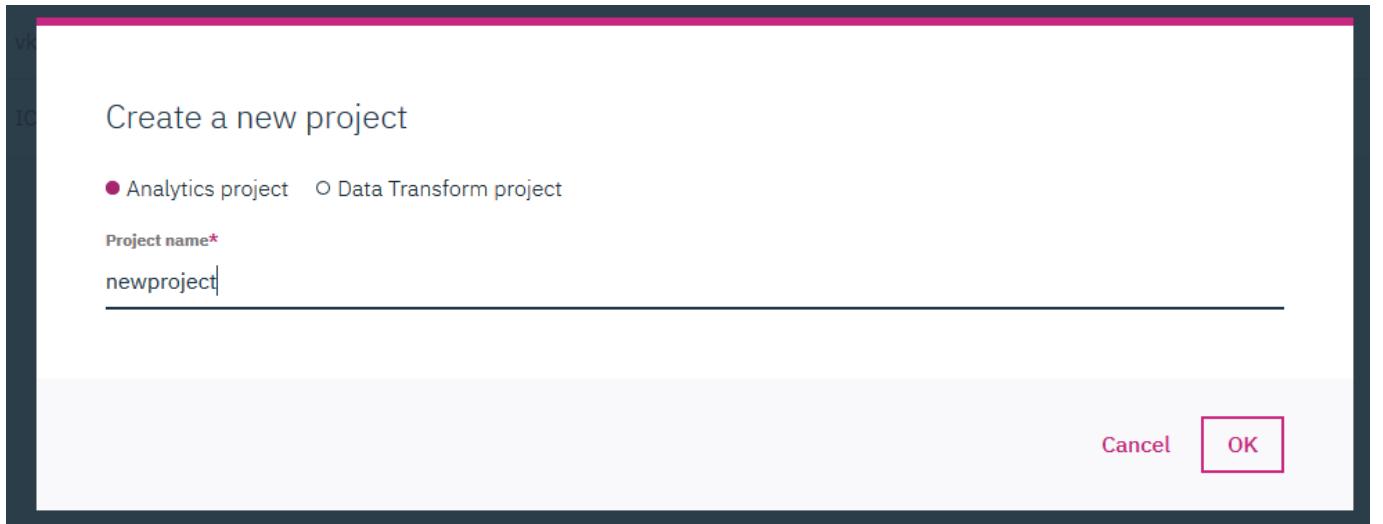
Working with Analytics Project

1.1. Click Projects and then Click on New Project



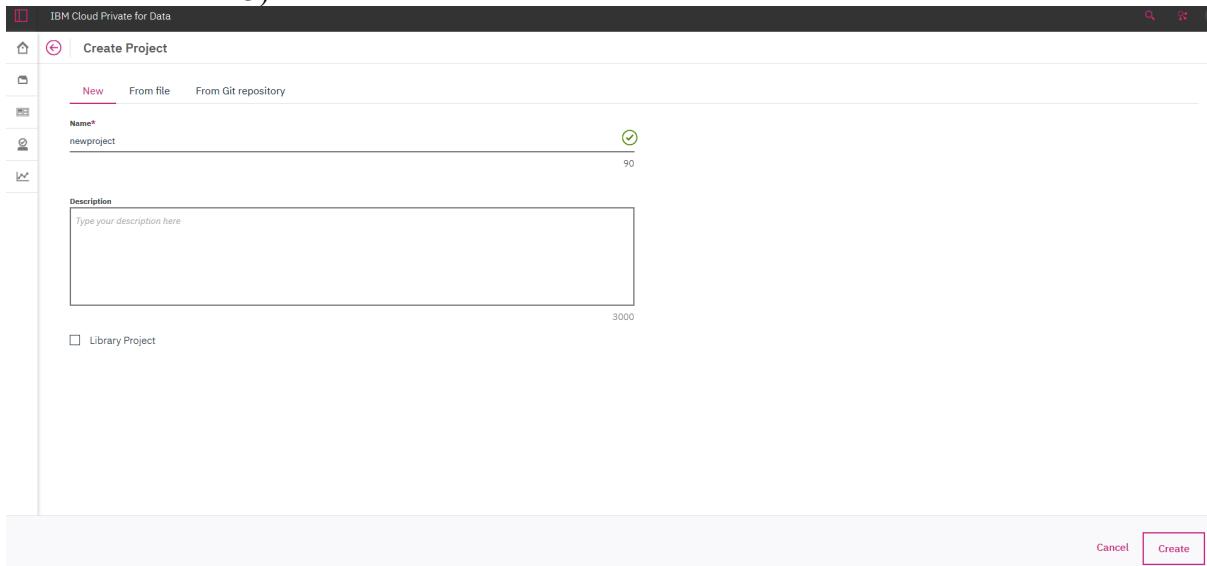
NAME	PROJECT TYPE	USER ROLE	LAST UPDATED	ACTIONS
vkdataview	Analytics	Admin	17 Jul 2018, 7:00 PM	
ICP4DTelcoChurnVK	Analytics	Admin	17 Jul 2018, 7:00 PM	

1.2. This will bring option to create a Analytics project or Data Transform project



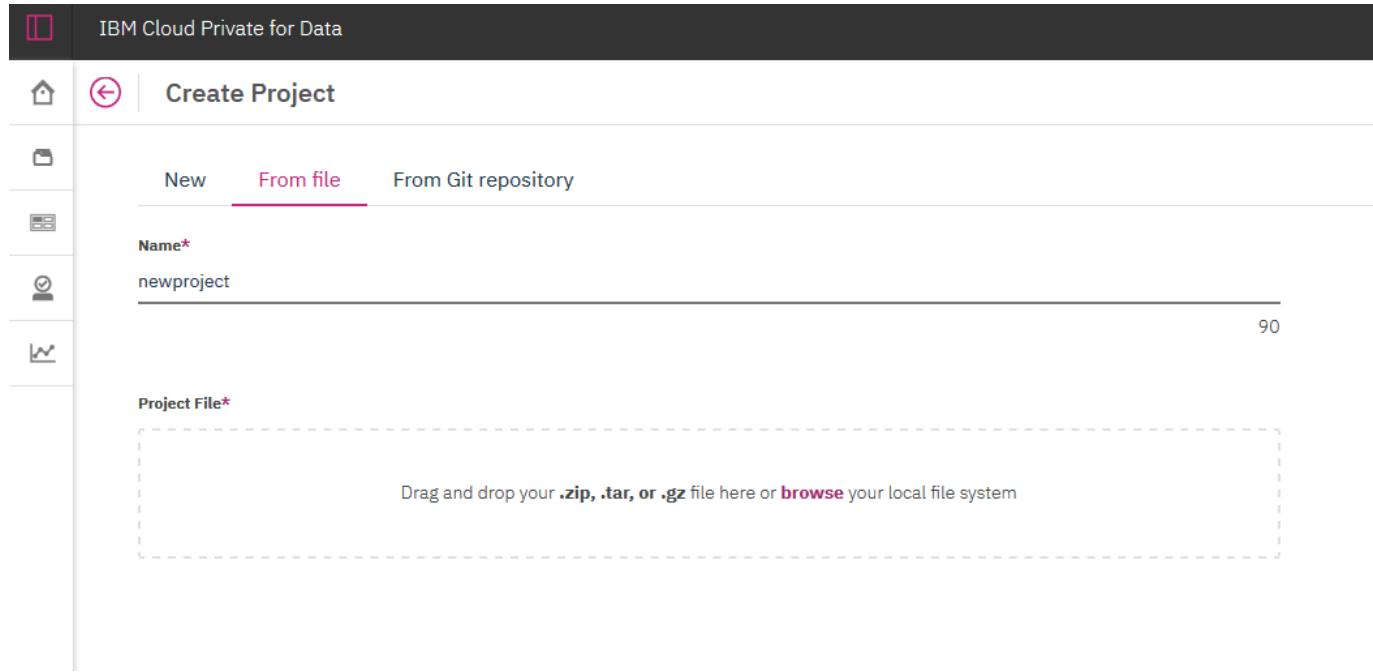
1.3. Type newproject and click OK

1.4. This will bring “Create Project” screen (do not create the project as yet, we will do this in section 3) :



1.5. Select Analytics project

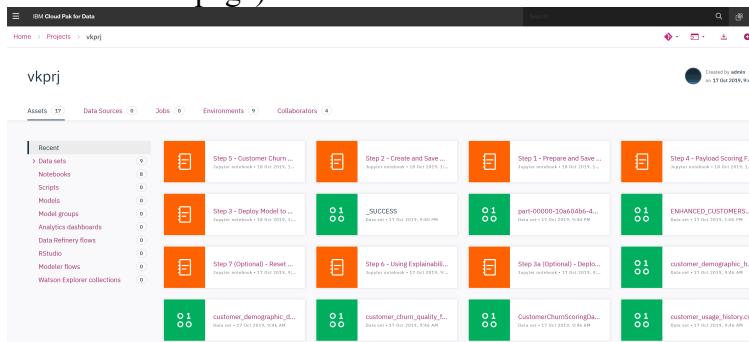
1.6. To import project select ‘**from file**’. Under ‘Notebook File’ drag and drop or browse for the Zip file you downloaded earlier (**YOURINITIALS_lab136.zip**).



1.7. Name the Project to whatever you want. For example enter “newproject”. (note by default zip/tar name will used, make sure to remove any such extension from Name field)

1.8. Click ‘**Create**’, then the project is created. This will take 2-4 minutes based on concurrent activity on the system.

1.9. Click ‘**Assets**’ to view all available assets that are in this project. (may have to switch to different tab to refresh Assets page)



We will be using a data set that has been enriched and made available in a Virtualized table for this lab.

Now we will be loading the existing notebook “Step 2 – Create and Save Model” . The notebook contains comments which will guide you through its use. As you review each cell click in it with

your mouse and use the run button on the toolbar to execute it (short-cut is to use **Shift-Enter** key to execute current selected cell

1. Navigate to the ‘Notebooks’ tab and Click on 3 dots ⋮ right of notebook named “german-credit-risk-mode”. Select “Open in Jupyter with Python 3.5, Scala 2.11, R 3.4.3, Spark 2.2.1”

2. This will open notebook with proper kernel required to run the steps in the notebook.
3. Run all the cells in the first section “1 Import data and prepare model training data”.
4. Run all the cells in the second section “2 Train the model”.
5. Run all the cells in the third section “3 Save the model in Analytics Project” and **remember to change the name of the model**.
6. Run all the cells in the fourth section “4 Save the model to WML” and **remember to change the name of the model**.
7. Run all the cells in the fifth section “5 Deploy the model” and remember to change the name of the model. In addition, in following cells, **you need to change the model deployment id "aeb581bb-f4b2-4405-a33e-855440908444" to your new id**:

Remember to change the following model deployment id "aeb581bb-f4b2-4405-a33e-855440908444" to your new id

```
In [21]: deployment_details = wml_client.deployments.get_details('aeb581bb-f4b2-4405-a33e-855440908444')

In [22]: icp4d_churn_scoring_url = deployment_details['entity']['scoring_url']
print(icp4d_churn_scoring_url)

https://10.1.1.1:31843/v3/scoring/online/aeb581bb-f4b2-4405-a33e-855440908444
```

8. Run all the cells in the sixth section “6 Save Training Data to db2” and **remember to change “TABLE_NAME”**.

6 Save Training Data to db2 table (remember to change TABLE_NAME)

```
: jdbcuri = "jdbc:db2://10.1.1.4:30285/BLUDB"

properties = {
    "user": "user999",
    "password": "y39cFCX_x4i%G6*",
    "driver": "com.ibm.db2.jcc.DB2Driver",
    "sslConnection": "false"
}

TABLE_NAME = "modeltrn_german_risk"

: df_data.write.jdbc(url=jdbcuri, table=TABLE_NAME, mode="append", properties=properties)
```

Section III

Setup Watson OpenScale to monitor and manage ML model

In this section we will :

- Setup OpenScale configuration for deployed ML model
- Setup model monitoring details

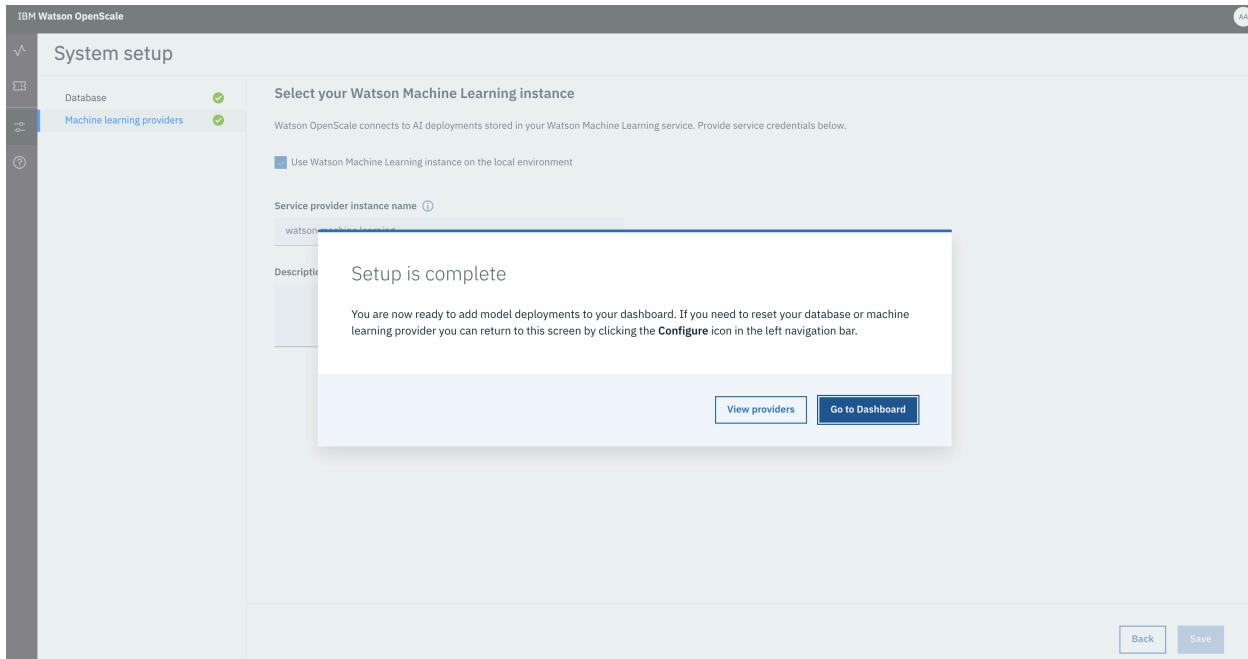
1. Navigate to AI OpenScale web console:

<https://services-uscentral.skytap.com:10068/aiopenscale/>

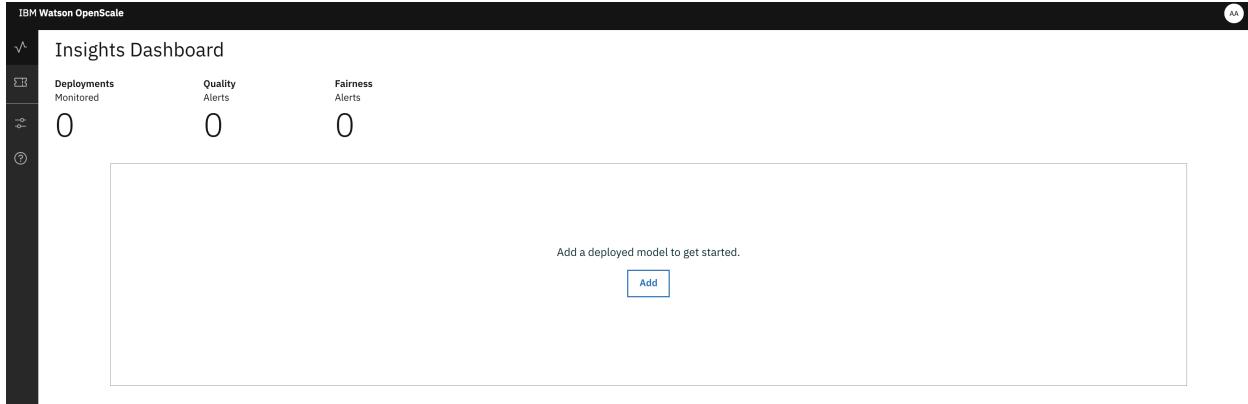
2. Initial Configuration of Watson Open Scale

Configure Db2 warehouse and Watson Machine Learning for OpenScale.

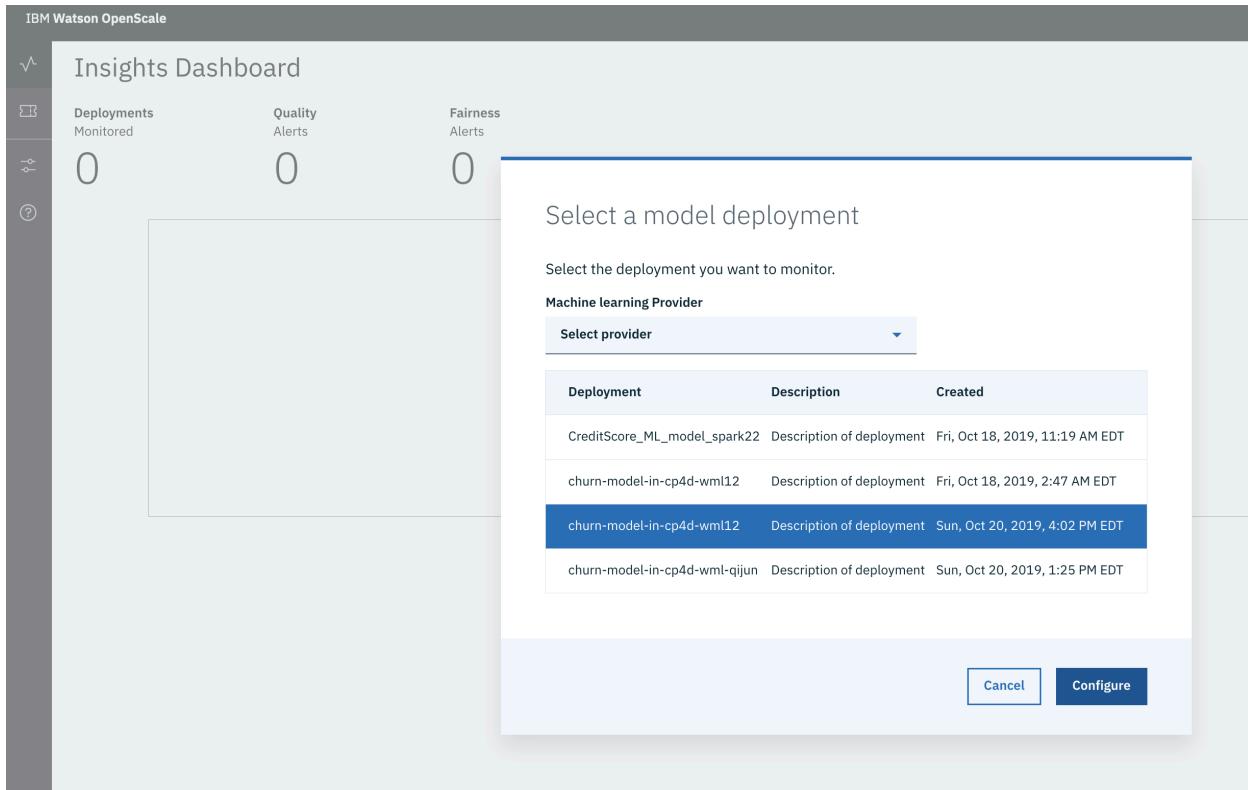
The screenshot shows the 'System setup' page of the IBM Watson OpenScale web interface. On the left, there's a sidebar with a tree view showing 'Database' (selected) and 'Machine learning providers'. The main area is titled 'Connect to your database instance' with a sub-instruction: 'Enter the connection information for your database, then click Connect to verify the connection and fetch schemas.' It includes fields for 'Database type' (set to 'Db2'), 'Hostname or IP address' (10.1.1.4), 'Port' (30285), 'Use SSL' (unchecked), 'Database' (BLUDB), 'Username' (user999), 'Password' (redacted), and 'Schema' (USER999). A 'Connect' button is visible, and a green success message 'Success!' is displayed next to it. A note at the bottom says 'Note: Only compatible schemas are shown.' At the bottom right are 'Back' and 'Save' buttons.



3. Add a deployment and configure model details:



4. Choose the new deployment name (remember to change to your model name):



5. Payload logging setting for the model: choose following options.

[Dashboard](#) / [Configure](#)

[◀](#) **churn-model-in-cp4d-wml12**

Payload logging

- Model details
- Quality
- Fairness
- Explainability
- Drift *beta*

Specify type of input

Select the type of data the deployment analyzes and the algorithm type.

Data type ⓘ

Numeric/categorical

Algorithm type

Binary classification

Save

6. Go back to notebook “german-credit-risk-model” and run all the cells in the seventh section “7 Payload Scoring for OpenScale configuration”.

7 Payload Scoring for OpenScale configuration

```

1: fields = ["CheckingStatus", "LoanDuration", "CreditHistory", "LoanPurpose", "LoanAmount", "ExistingSavings", "EmploymentDuration", "InstallmentPercent", "Sex", "OthersOnLoan", "CurrentResidenceDuration", "OwnsProperty", "Age", "InstalmentPlans", "Housing", "ExistingCreditsCount", "Job", "Dependents", "Telephone", "ForeignWorker", "CheckingStatus_IX", "CreditHistory_IX", "EmploymentDuration_IX", "ExistingSavings_IX", "ForeignWorker_IX", "Housing_IX", "InstalmentPlans_IX", "Job_IX", "LoanPurpose_IX", "OthersOnLoan_IX", "OwnsProperty_IX", "Sex_IX", "Telephone_IX", 'features', 'rawPrediction', 'probability', 'prediction', 'predictedLabel']
values = [
    ["no_checking", 13, "credits paid to date", "car_new", 1343, "100_to_500", "1_to_4", 2, "female", "none", 3, "savings_insurance", 46, "none", "own", 2, "skilled", 1, "none", "yes"],
    ["no_checking", 24, "prior payments delayed", "furniture", 4857, "500_to_1000", "1_to_4", 4, "male", "none", 4, "savings_insurance", 46, "none", "free", 2, "management_self-employed", 1, "none", "yes"],
    ["0_to_200", 26, "all credits paid back", "car_new", 863, "less_100", "less_1", 2, "female", "co-applicant", 2, "real_estate", 29, "none", "own", 1, "skilled", 1, "none", "yes"],
    ["0_to_200", 14, "no credits", "car_new", 2368, "less_100", "1_to_4", 3, "female", "none", 3, "real_estate", 23, "none", "rent", 1, "management_self-employed", 1, "none", "yes"],
    ["0_to_200", 17, "credits paid to date", "car_new", 832, "100_to_500", "1_to_4", 2, "male", "none", 2, "real_estate", 42, "none", "own", 1, "skilled", 1, "none", "yes"],
    ["no_checking", 33, "outstanding_credit", "appliances", 5696, "unknown", "greater_7", 4, "male", "co-applicant", 4, "unknown", 54, "none", "free", 2, "skilled", 1, "yes", "yes"],
    ["0_to_200", 13, "prior_payments_delayed", "retraining", 1375, "100_to_500", "4_to_7", 3, "male", "none", 3, "real_estate", 37, "none", "own", 2, "management_self-employed", 1, "none", "yes"]
]
payload_scoring = {"fields": fields, "values": values}
scoring_response = wml_client.deployments.score(icp4d_churn_scoring_url, payload_scoring)
print('Single record scoring result:', '\n fields:', scoring_response['fields'], '\n values:', scoring_response['values'][0])
print(scoring_response['values'][0])

```

Single record scoring result:
fields: ['CheckingStatus', 'LoanDuration', 'CreditHistory', 'LoanPurpose', 'LoanAmount', 'ExistingSavings', 'EmploymentDuration', 'InstallmentPercent', 'Sex', 'OthersOnLoan', 'CurrentResidenceDuration', 'OwnsProperty', 'Age', 'InstalmentPlans', 'Housing', 'ExistingCreditsCount', 'Job', 'Dependents', 'Telephone', 'ForeignWorker', 'CheckingStatus_IX', 'CreditHistory_IX', 'EmploymentDuration_IX', 'ExistingSavings_IX', 'ForeignWorker_IX', 'Housing_IX', 'InstalmentPlans_IX', 'Job_IX', 'LoanPurpose_IX', 'OthersOnLoan_IX', 'OwnsProperty_IX', 'Sex_IX', 'Telephone_IX', 'features', 'rawPrediction', 'probability', 'prediction', 'predictedLabel']
values: ['no_checking', 13, 'credits_paid_to_date', 'car_new', 1343, '100_to_500', '1_to_4', 2, 'female', 'none', 3, 'savings_insurance', 46, 'none', 'own', 2, 'skilled', 1, 'none', 'yes', 0.0, 0.0, 1.0, 0.0, 0.0, 0.0, 0.0, 1.0, 0.0, [2], [1, 3, 5, 13, 14, 15, 16, 17, 18, 19, 20], [1.0, 1.0, 1.0, 13.0, 2.0, 3.0, 13.0, 46.0, 2.0, 1.0]], [12.571932961952987, 7.428067038047013], {0.6285966480976494, 0.3714033510235063}], 0.0, 'No Risk']

7. Go to OpenScale again and click on “I’m finished” button.

Dashboard / Configure

churn-model-in-cp4d-wml12

Payload logging																	
Model details																	
Quality																	
Fairness																	
Explainability																	
Drift <small>beta</small>																	
<div style="border: 1px solid #ccc; padding: 5px; display: inline-block;"> I'm finished How do I send a scoring request? </div>																	
Details <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 30%;">Deployment name</td> <td>churn-model-in-cp4d-wml12</td> </tr> <tr> <td>Data type</td> <td>Numeric/categorical</td> </tr> <tr> <td>Algorithm type</td> <td>Binary classification</td> </tr> <tr> <td>Datamart ID</td> <td>00000000-0000-0000-000000000000</td> </tr> <tr> <td>Feedback table name</td> <td>Feedback_223751ec-fa68-47d6-992d-3037d7ab89be</td> </tr> <tr> <td>Deployment ID</td> <td>bec4ff-c979-42de-95b4-bdc5dd61e64</td> </tr> <tr> <td>Subscription ID</td> <td>223751ec-fa68-47d6-992d-3037d7ab89be</td> </tr> <tr> <td>Binding ID</td> <td>999</td> </tr> </table>		Deployment name	churn-model-in-cp4d-wml12	Data type	Numeric/categorical	Algorithm type	Binary classification	Datamart ID	00000000-0000-0000-000000000000	Feedback table name	Feedback_223751ec-fa68-47d6-992d-3037d7ab89be	Deployment ID	bec4ff-c979-42de-95b4-bdc5dd61e64	Subscription ID	223751ec-fa68-47d6-992d-3037d7ab89be	Binding ID	999
Deployment name	churn-model-in-cp4d-wml12																
Data type	Numeric/categorical																
Algorithm type	Binary classification																
Datamart ID	00000000-0000-0000-000000000000																
Feedback table name	Feedback_223751ec-fa68-47d6-992d-3037d7ab89be																
Deployment ID	bec4ff-c979-42de-95b4-bdc5dd61e64																
Subscription ID	223751ec-fa68-47d6-992d-3037d7ab89be																
Binding ID	999																

8. Configure model details.

The screenshot shows the 'Model details' configuration page for a model named 'churn-model-in-cp4d-wml12'. The left sidebar lists several tabs: 'Payload logging' (selected), 'Model details' (highlighted with a green checkmark), 'Quality', 'Fairness', 'Explainability', and 'Drift beta'. The main content area is titled 'Model details' and contains a sub-section titled 'Provide information about the model deployment and training data to prepare Watson OpenScale for monitoring and providing explanations for model transactions.' A small icon of a triangle and an 'X' is displayed above the text. In the bottom right corner of the main area, there is a blue 'Begin' button.

9. Click on the Manually configure monitors:

The screenshot shows the 'Configure monitors' configuration page for the same model. The left sidebar shows the 'Model details' tab is selected. The main content area is titled 'Configure monitors' and contains a sub-section titled 'Manually configure monitors using a guided walkthrough or upload training data distribution generated by running a custom notebook.' Two options are presented: 'Manually configure monitors' (selected and highlighted with a blue border) and 'Upload training data distribution'. Below these options, a note states: 'Monitor configuration requires information about the training data. Manual configuration requires a connection to the training data for analysis. If you prefer to analyze the training data without providing a connection, choose the upload option.' At the bottom right, there are 'Back' and 'Next' buttons.

10. Set db2 as the training data location.

[churn-model-in-cp4d-wml12](#)

Payload logging

Model details

- Quality
- Fairness
- Explainability
- Drift beta

Specify the location of the training data

Training data must be stored in a Db2 database or Cloud Object Storage. Enter your connection information, then click **Test** to verify the connection.

Location

Select database type

Db2

Cloud Object Storage

Back **Next**

11. Set specify the location of the training data:

Payload logging

Model details

- Quality
- Fairness
- Explainability
- Drift beta

Specify the location of the training data

Training data must be stored in a Db2 database or Cloud Object Storage. Enter your connection information, then click **Test** to verify the connection.

Location

Db2

Hostname or IP address

10.1.1.4

Port

30285 Use SSL

Database

BLUDB

Username

user999

Password

.....

Test ✓ Success!

Back **Next**

12. Following the direction of model settings and finally get the model set as follows:

13. Configure fairness and Quality:

1) Configuring Monitor for Fairness

a) Identify Age and Gender are the attributes those you want to Monitor for Fairness. These two are sometimes automatically identified by Open Scale as attributes to be monitored for Fairness based on the training data.

- b) Use, 'F' (as False) as favorable outcome and 'T' (as True) as unfavorable outcome.
- c) For monitoring Gender, use 'F' (as Female) as Reference group and 'M' (as Male) as Monitored group. Use 80% as threshold for Bias. And use 50 as the minimum sample of records to be used.
- d) For monitoring Age, use 40 to 60 as Reference group and 22 to 35 (as Male) as Monitored group. Use 97% as threshold for Bias. And use 40 as the sample of records to be used.

2) Configuring Monitor for Quality -

- a) Use 80% as accuracy Threshold
- b) Use 50 as minimum sample and 200 as maximum sample

c) Upload the ‘german_credit_data_biased_training.csv’ provided as feedback file.

14. Go back to notebook “german-credit-risk-model” and run all the cells in the eighth section “8 More payload Scoring to get fairness records enough”

8 More payload Scoring to get fairness records enough

```
In [27]: score_data = SQLContext(sc).read.csv(os.environ['DSX_PROJECT_DIR']+ '/datasets/german_credit_data_biased_scoring.csv', header='true', inferSchema = 'true')
df_data.show(5)

+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
|Checkingstatus|LoanDuration|CreditHistory|LoanPurpose|LoanAmount|ExistingSavings|EmploymentDuration|InstallmentPercent|Sex|OthersOnLoan|CurrentResidenceDuration|OwnsProperty|Age|Inst
allmentPlans|Housing|ExistingCreditsCount|Job|Dependents|Telephone|ForeignWorker|Risk|
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| 0_to_200| 31|credits_paid_to_date| other| 1889| 100 to 500| less_1| 3|female| none| 3|savings_insurance| 32|
none| own| 1|skilled| 1| none| yes|No Risk| 1_to_4| 2|female| none| 2|savings_insurance| 37|
| less_0| 18|credits_paid_to_date| car_new| 462| less_100| 1_to_4| 2| male| none| 3| real_estate| 28|
stores| own| 2|skilled| 1| none| yes|No Risk| 1_to_4| 2| male| none| 3| savings_insurance| 32|
| less_0| 15|prior_payments_de...| furniture| 250| less_100| greater_7| 3| male| none| 3| unknown| 57|
none| own| 2|skilled| 1| yes| no|No Risk| greater_7| 3| male| none| 2|savings_insurance| 32|
| 0_to_200| 28|credits_paid_to_date| retraining| 3693| less_100| greater_7| 3| male| none| 2|savings_insurance| 32|
none| own| 1|skilled| 1| none| yes|No Risk| greater_7| 3| male| none| 2|savings_insurance| 32|
| no_checking| 28|prior_payments_de...| education| 6235| 500 to 1000| greater_7| 3| male| none| 2|savings_insurance| 32|
none| own| 2|skilled| 1| none| yes| Risk| greater_7| 3| male| none| 2|savings_insurance| 32|
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
only showing top 5 rows

In [ ]: score_data_pandas = score_data.toPandas()
r1=score_data_pandas.values[0].tolist()
payload_scoring = {"fields": fields,"values": [r1]}
scoring_response = wml_client.deployments.score(icp4d_churn_scoring_url, payload_scoring, "custchurn_rfcl")
scoring_response
```

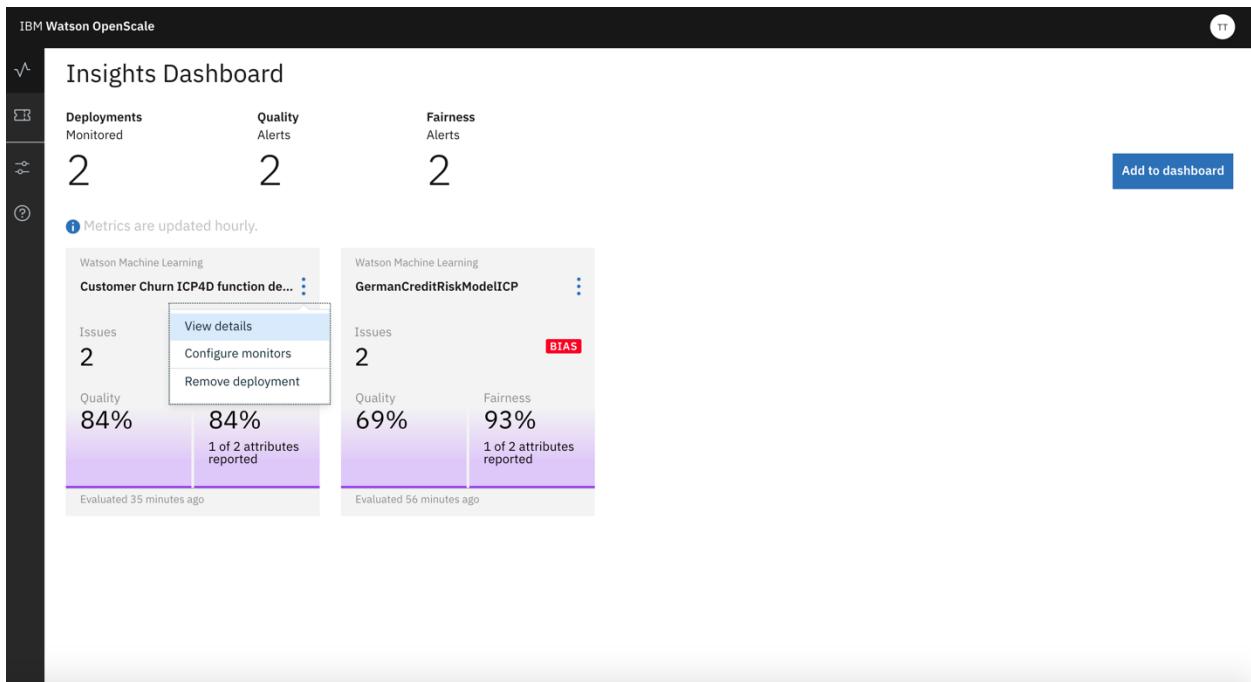
Section IV

Viewing Monitoring Model Results in Dashboard

In this section we will:

- Check how to monitor results

Now go to the Dashboard and select the tile of the Model you have already configured.



Click the same and it will take you to the detailed view of the monitored result.

Click on one of the Attribute you are monitoring for Fairness. It will take you to the Fairness Screen. There you can click on a particular point on the line.

IBM Watson OpenScale

Customer Churn ICP4D function deployment v6

Model ID: 14bb4e5d-eab2-47e0-9e27-810dcad5306e | Created date: 9/17/2019 | Configure monitors

Fairness

- GENDER (⚠)
- AGE
- Quality**

 - Area under ROC (⚠)
 - Area under PR
 - Accuracy
 - True positive rate (TPR)
 - False positive rate (FPR)
 - Recall
 - Precision
 - F1-Measure
 - Logarithmic loss

- Performance**

 - Throughput

- Analytics**

 - Predictions by Confidence

Fairness for GENDER

The models propensity to deliver favorable outcomes to one group over another. [Learn more.](#)

Time frame

- Hourly (selected)
- Daily
- Weekly
- Past 3 months
- Past week (selected)
- Yesterday
- Today
- Custom range

Date range

From: 9/13/2019 To: 9/19/2019

Fairness Score for GENDER

N/A

Tue, Sep 17, 2019, 6:00 PM CDT

Threshold: 96%

Monitored Groups

Average	N/A
M	N/A

That will take you to the details of the Fairness.

IBM Watson OpenScale

Customer Churn ICP4D function depl... : Transactions

Data Set: Payload + Perturbed

Monitored Feature: GENDER

Date and Time: 9/18/2019, 8:00 PM

Monitored groups: 62% of the group M received favorable outcomes.

Reference groups: 73% of the group F received favorable outcomes.

Favorable outcomes: F (●) T (●)

% FAVORABLE OUTCOMES

Group	Favorable Outcomes (%)
F	73%
M	62%

View Transactions

From this screen you can select View Transactions button. That will take you to the Transaction Details screen.

IBM Watson OpenScale

Customer Churn ICP4D function deployment v6: Transactions

September 17, 2019, 7:00 PM

GENDER

View
 All transactions Biased transactions

Review the transactions that were received for the selected time period. Click an Explain link to determine the features that contributed to each outcome.

Transaction ID	GENDER	Bias Detected	Outcome	Action
rfc175142711-1	F		F	Explain
rfc175143503-1	F		F	Explain
rfc175142068-1	M		T	Explain
rfc17514810-1	F		T	Explain
rfc175141118-1	F		T	Explain
rfc17514938-1	F		F	Explain
rfc175142942-1	F		T	Explain
rfc175141937-1	F		F	Explain

Payload Table [?](#) Payload_9aa024c4-1ce0-4b76-8171-b36bf90a0cf
Corrected Records [?](#) -

F : Favorable Outcome
Current Model 61.67%
De-biased Model 0.00%

T : Unfavorable Outcome
Current Model 38.33%
De-biased Model 0.00%

There you can click on Explain link for any transaction. That would generate Explanation for that transaction. It may take few minutes for WoS to generate the Explanation

IBM Watson OpenScale

Explain a transaction

rfc266642711-1 x rfc175142711-1 x

Details [?](#)
 Transaction rfc266642711-1
 Deployment churn-model-in-cp4d-wml-dep1
 Model Name churn-model-in-cp4d-wml

Minimum changes for another outcome [?](#)
Watson OpenScale has not calculated these values because they are either already at the median, or the prediction has not changed as a result of moving values away from the median.

Maximum changes allowed for the same outcome [?](#)
 GENDER F
 STATUS M
 CAROWNER N

How this prediction was determined
 The churn-model-in-cp4d-wml predicts F with 88.52% confidence. The following features were most important in determining this prediction: STATUS (29.02%), CHILDREN (24.58%), and INTERNATIONAL (14.54%).

Most important factors influencing prediction

Feature	Value	Weight
STATUS	M	29.02%
CHILDREN	2	24.58%
INTERNATIONAL	0	14.54%

CONFIDENCE

T 11.48% | 88.52% F

Factors contributing to T confidence level Factors contributing to F confidence level

Now go back to the Dashboard and clickn on Tile of your deployment again. Click on Quality/Area Uner RoC. It will show you the Quality with respect to Threshold you have set in configuration step before.

IBM Watson OpenScale

Customer Churn ICP4D function deployment v6

Model ID: 14bb4e5d-eab2-47e0-9e27-810dcad5306e | Created date: 9/17/2019 | Configure monitors

Fairness

- GENDER ⚠️
- AGE

Quality

- Area under ROC** ⚠️
- Area under PR
- Accuracy
- True positive rate (TPR)
- False positive rate (FPR)
- Recall
- Precision
- F1-Measure
- Logarithmic loss

Performance

- Throughput

Analytics

- Predictions by Confidence

Area under ROC

Area under recall and false positive rate curve. [Learn more.](#)

Time frame

Hourly Daily Weekly Past 3 months Past week Yesterday Today Custom range

From To

Area under ROC
0.84
0.12 below threshold
Wed, Sep 18, 2019, 4:00 PM CDT
Threshold 0.95

Schedule

Last Evaluation 9:46 PM CDT
Next Evaluation 10:46 PM CDT
[Check quality now](#)

If you click on the line of Quality (green line) you can see the details of the Quality.

IBM Watson OpenScale

Customer Churn ICP4D function deployment v6 : Area under ROC

Area under ROC	Area under PR	Accuracy	True positive rate (TPR)	False positive rate (FPR)	Recall	Precision	F1-Measure	Logarithmic loss
0.84	0.92	0.83	0.68	0	0.68	1	0.81	0.45

Actual	Prediction	F	T
F		26	0
T		9	19

Records Evaluated

Total	54
-------	----