PHASE 3 PROJECT SUBMISSION P Hariharan

Given Statement:

Machine Learning Model Deployment with IBM Cloud Watson Studio

Phase 3: Development Part 1 In this part you will begin building your project. Start building the machine learning model using IBM Cloud Watson Studio.

Define the predictive use case (e.g., customer churn prediction) and select a relevant dataset. Use IBM Cloud Watson Studio's tools to import the dataset, preprocess the data, select features, and train the machine learning model.

Define the Predictive Use Case: Start by defining your predictive use case. For example, if you're building a customer churn prediction model, you need to specify the goal of your model, which is to predict whether a customer is likely to churn or not.

Select a Relevant Dataset: Choose a dataset that is relevant to your use case. In this case, you might need historical customer data that includes information about customers who have churned and those who haven't. Ensure that your dataset is in a suitable format for analysis.

Import the Dataset: In IBM Cloud Watson Studio, you can import your dataset. Watson Studio provides tools to upload and manage data. You can typically upload datasets in various formats like CSV, Excel, or connect to databases.

Preprocess the Data: Data preprocessing is crucial. Use Watson Studio's data preparation tools to clean and transform your dataset. This may involve handling missing values, encoding categorical variables, and scaling or normalizing features.

Preprocess the Data: Data preprocessing is crucial. Use Watson Studio's data preparation tools to clean and transform your dataset. This may involve handling missing values, encoding categorical variables, and scaling or normalizing features.

Select Features: Feature selection is important for model performance. Use techniques like feature engineering and feature selection to choose the most relevant attributes for your model.

Train the Machine Learning Model: Once your data is prepared and features are selected, you can use Watson Studio to train your machine learning model. You can choose from various algorithms and techniques depending on your use case. It's important to split your data into training and testing sets to evaluate the model's performance.

Evaluate and Tune the Model: After training, assess the model's performance using evaluation metrics. Watson Studio provides tools for this purpose. You might need to fine-tune the model by adjusting hyperparameters to improve its

Save the Model: Once satisfied with the model's performance, save it in Watson Studio. This will allow you to deploy it later.

This is a high-level overview of the initial steps for building a machine learning model in IBM Cloud Watson Studio. Remember that the specific tools and steps may vary based on your chosen use case and dataset, so consult the Watson Studio documentation and resources for detailed guidance.

Objectives

Import data to a project.

Build a machine learning model.

Deploy the model and try out the API.

Test a machine learning model.

Monitor the deployed model

Retrain your model.

Import data to a project

A project is how you organize your resources to achieve a particular goal. Your project resources can include data, collaborators, and analytic tools like Jupyter notebooks and machine learning models.

You can create a project to add data and open a data asset in the data refiner for cleansing and shaping your data.

Create a project

If you do not have an existing Object Storage service, go to the IBM Cloud® catalog and create an instance of Object Storage.

Step 1: From the catalog, create Watson Studio

Select a region

Select a Lite pricing plan

Change the Service name to watson-studio-tutorial

Select a resource group and click Create

Click on the Launch in twisty and select IBM watsonx.

Create a project by clicking on the upper left hamburger menu and selecting Projects > Vew all projects then New project.

In the subsequent page click Create an empty project.

Provide iris\_project as the project name.

Under Storage, choose an existing Object Storage service verified to exist a few steps earlier.

Click Create. Your new project opens and you can start adding resources to it.

Step 2: Associate the Machine Learning service

In the top navigation menu, of the iris-project click on Manage then select the Services & integrations section on left.

Click Associate Service.

If you have an existing Watson Machine Learning service instance, skip to the next step. Otherwise continue with the following steps to create a new instance.

Click New service and then click on the Watson Machine Learning tile.

Select a region same as the Watson Studio service and choose a Lite plan.

Enter machine-learning-tutorial as the Service name and select a resource group.

Click Create to provision a Machine Learning service.

Check the checkbox next to the Machine Learning service and click Associate service.

Step 3: Build a machine learning model

In the top navigation menu, click on iris-project, click on Assets in the top bar.

Click on New task + and search for auto.

Click on the Build machine models automatically tile.

Set the name to iris\_auto.

Under Watson Machine Learning service instance, notice the service previously associated.

Click Create.

Once the model is created,

Add training data by clicking Select data from project.

Choose the iris\_initial.csv file under Data asset.

Click Select asset.

If prompted, answer No to Create a time series forecast?.

Select Species as your What do you want to predict?.

Click Experiment settings.

Select Data source.

Under Training and holdout method, set Holdout data split to 14% by moving the slider.

On the left menu, Click on Prediction:

Set Prediction type to Multiclass classification.

Set Optimized metric as Accuracy.

Click on Save settings.

Click on Run experiment.

The AutoAI experiment may take up to 5 minutes to select the right Algorithm for your model.

Step 4: Deploy and test your model

In this section, you will deploy the saved model and test the deployed model,

Under the created model, click on Promote to deployment space.

Under Target Space, select Create a new deployment space. You use deployment spaces to deploy models and manage your deployments.

Set the Name to iris\_deployment\_space.

Select the Object Storage storage service used in previous steps in the corresponding drop down.

Select the machine-learning-tutorial service in the corresponding drop down.

Click Create.

Click on Promote.

From the received notification, navigate to the deployment space.

In the Deployments > iris\_deployment\_space:

Click on the name of the model you just created.

Click the New deployment button.

Select Online as the Deployment type, provide iris\_deployment as the name and then click Create.

Under Deployments tab, once the status changes to Deployed, Click on the Name in the table. The properties of the deployed web service for the model will be displayed.

TEST THE DEPLOYED MODEL

{

"input\_data": [{

"fields": ["sepal\_length", "sepal\_width", "petal\_length", "petal\_width"],

"values": [

[5.1,3.5,1.4,0.2], [3.2,1.2,5.2,1.7]

]

}]

}

Step 5: Try out the API

Along with the UI, you can also do predictions using the API scoring endpoint by exposing the deployed model as an API to be accessed from your applications.

Under API reference tab of the deployment, you can see the Endpoint under Direct link and code snippets in various programming languages.

Copy the Endpoint in a notepad for future reference.

In a browser, launch the IBM Cloud Shell and export the scoring End-point to be used in subsequent requests. Make sure you don't close this window/tab..

To use the Watson Machine Learning REST API, you need to obtain an IBM Cloud Identity and Access Management (IAM) token. Run the below command

Copy the complete IAM token along with Bearer from the above response and export it as an IAM\_TOKEN to be used in the subsequent API requests

Run the below cURL code in the cloud shell to see the prediction results.

export SCORING\_ENDPOINT='<SCORING\_ENDPOINT\_FROM\_ABOVE\_STEP>'

ibmcloud iam oauth-tokens --output JSON | jq -r .iam\_token

export IAM\_TOKEN=$(ibmcloud iam oauth-tokens --output JSON | jq -r .iam\_token)

echo $IAM\_TOKEN

curl -X POST --header 'Content-Type: application/json' --header 'Accept: application/json' --header "Authorization: $IAM\_TOKEN" -d '{"input\_data": [{"fields": ["sepal\_length", "sepal\_width", "petal\_length","petal\_width"],"values": [[5.1,3.5,1.4,0.2], [3.2,1.2,5.2,1.7]]}]}' $SCORING\_ENDPOINT

Step 6: Monitor your deployed model with IBM Watson OpenScale

IBM® Watson OpenScale tracks and measures outcomes from your AI models, and helps ensure they remain fair, explainable, and compliant wherever your models were built or are running. Watson OpenScale also detects and helps correct the drift in accuracy when an AI model is in production.

For ease of understanding, the tutorial concentrates only on improving the quality (accuracy) of the AI model through Watson OpenScale service.

Provision IBM Watson OpenScale service

In this section, you will create a Watson OpenScale service to monitor the health, performance, accuracy and quality metrics of your deployed machine learning model.

Create a IBM Watson OpenScale service

Select a region preferably Dallas. Create the service in the same region where you created the Machine Learning service.

Choose Lite plan.

Set the service name to watson-openscale-tutorial.

Select a resource group.

Click Create.

Once the service is provisioned, Click Manage on the left pane and click Launch Application.

Click on Manual setup to manually setup the monitors.

Selecting a deployment

In this section, as part of preparing your model for monitoring you will set up and enable monitors for each deployment that you are tracking with IBM Watson OpenScale.

By clicking on the Edit icon on the Database tile, choose Free lite plan database as your Database type and click Save. This is to store your model transactions and model evaluation results.

Click on Machine learning providers

Click on Add machine learning provider and click the edit icon on the Connection tile.

Select Watson Machine Learning(V2) as your service provider type.

In the Deployment space dropdown, select the deployment space iris\_deployment\_space you created above.

Leave the Environment type to Pre-production.

Click Save.

On the far left pane:

Click the icon for Insights dashboard(first icon) to add a deployment

Click on Add to dashboard to start the wizard on the Select model location page.

On the Deployment spaces tab click on the iris\_deployment\_space radio button

Click Next

On the Select deployed model page:

Click iris\_deployment

Click Next

On the Provide model information page:

Data type: Numerical/categorical

Algorithm type: Multi-class classification

Click View summary

Click Finish

The iris\_deployment pre production dashboard is now displayed.

Click Actions > Configure monitors

Click the pencil icon on the Training data tile to start the wizard.

In the Select configuration method page

Click Use manual setup

Click Next

In the Specify training data method page

For Training data option choose Database or cloud storage

For Location choose Cloud Object Storage

For Resource instance ID and API key, run the below command in the Cloud Shell. Make sure to change the value after --instance-name to match the name of the Object Storage instance you have been using for this tutorial.

Copy and paste the Credentials resource\_instance\_id. It will begin with crn and end with two colons ::.

Copy and paste the Credentials api key without any trailing spaces.

Click Connect.

Select the Bucket that starts with irisproject-donotdelete-.

Select iris\_initial.csv from the Data set dropdown.

Click Next

In the Select the feature columns and label column method page

The defaults should be correct. Species as the Label/Target and the rest as Features.

Click Next

In the Select model output method page

The defaults should be correct, prediction for Prediction and probability for Probability.

Click View summary

Click Finish

Click the pencil icon on the Model output details tile to start the wizard.

In the Specify model output details method page

The defaults should be correct.

Click Save

On the left pane, click on Quality under Evaluations and click the edit icon on the Quality thresholds tile

In the Quality thresholds page set the following values:

Accuracy 0.98

Click Next

In the Sample size page

Set Minimum sample size to 10

Click Save

On the left pane, Click on Go to model summary

The quality monitor (previously known as the accuracy monitor) reveals how well your model predicts outcomes.

As the tutorial uses a small dataset, configuring Fairness and Drift won't have any impact.

Evaluate the deployed model

In this section, you will evaluate the model by uploading a iris\_retrain.csv file which contains 10 instances of each species. Download iris\_retrain.csv.

Click on Actions and then Evaluate now.

Choose from CSV file as your import option and click on browse, upload the iris\_retrain.csv file.

Click and click on Upload and evaluate.

After the evaluation is completed, you should see the dashboard with different metrics.

To understand the quality metrics, refer to Quality metric overview

ibmcloud resource service-key $(ibmcloud resource service-keys --instance-name "cloud-object-storage-tutorial" | awk '/WDP-Project-Management/ {print $1}')

Step 7: Remove resources

Navigate to IBM Cloud® Resource List.

Under Name, enter tutorial in the search box.

Delete the services which you created for this tutorial.