MACHINE LEARNING MODEL DEPLOYMENT WITH IBM CLOUD WATSON STUDIO

Predictive Use Case: Customer Churn Prediction

Development part 2:

Step 1: Deploy the Trained Model as a Web Service

- **Create a Deployment:**
  - Within your Watson Studio project, navigate to the model you want to deploy.
  - Click on the model and choose "Create Deployment" to deploy the model as a web service.
  - Configure deployment settings such as the number of instances and available resources.
  - Deploy the model.

- **Obtain the API Endpoint:**
  - After deployment, you will receive an API endpoint URL. This URL is essential for making predictions using the deployed model.

Step 2: Integrate the Deployed Model into Applications

Using Python :

- **Install Required Libraries:**
  - Install the `requests` library in your Python environment if you haven't already. This library will be used to make HTTP requests to the API endpoint.

bash

```
pip install requests
```

- **Make Predictions in Python:**
  - Use the obtained API endpoint to make predictions from your Python application.

python

```
import requests
import json
```

```
# Replace "YOUR_API_ENDPOINT_URL" with the actual API endpoint URL
api_endpoint = "YOUR_API_ENDPOINT_URL"

# Sample data for prediction
data = {
  "fields": ["feature1", "feature2", "feature3"],
  "values": [[value1, value2, value3]]
}

# Make a POST request to the API endpoint for predictions
response = requests.post(api_endpoint, json=data)

# Check if the request was successful
if response.status_code == 200:
  # Get the prediction results from the JSON response
  predictions = response.json().get("predictions", [])
  print("Predictions:", predictions)
else:
  print("Request failed with status code:", response.status_code)
```

output:

```
Predictions: [0.75, 0.62, 0.88]
```

- **Handle the Prediction Response:**
  - Parse the JSON response to extract the predictions in your application.
  - Integrate the predictions into your application's logic.

Step 3: Implement Error Handling and Logging (Optional but Recommended):

- Implement error handling in your application to handle cases where the API endpoint might be unreachable or if there are issues with the request.
- Implement logging to keep track of the requests and responses for debugging and monitoring purposes.

Step 4: Secure Your API Endpoint :

- Consider implementing authentication mechanisms such as API keys or tokens to secure your API endpoint.

- Configure proper access control policies to restrict access to authorized users or applications.

Step 5: Continuous Monitoring and Maintenance:

- Set up monitoring tools to keep track of the API usage, response times, and errors.
- Regularly monitor the predictions and retrain the model if its performance decreases over time or if there are significant changes in the data distribution.

Conclusion:

By following these steps, you can deploy a trained machine learning model as a web service in IBM Cloud Watson Studio and seamlessly integrate it into your applications using the provided API endpoint. Remember to replace `"YOUR_API_ENDPOINT_URL"` with the actual API endpoint URL obtained after deploying your model.