

DEPLOY ON IBM CLOUD OUTPUT

Date	16 November 2022
Team ID	PNT2022TMID10674
Project Name	Project – Machine Learning based Vehicle Performance Analyzer

The screenshot shows the IBM Watson Studio interface. The top navigation bar includes 'Service Details - IBM Cloud', 'IBM Watson Studio', and 'VehiclePerformanceAnalysisModel'. The main content area displays the deployment details for 'analysis_deploy_model', which is marked as 'Deployed' and 'Online'. The 'API reference' tab is active, showing the endpoint: `https://eu-gb.ml.cloud.ibm.com/ml/v4/deployments/f4a6cc62-cd58-47a3-af62-6a940301a611?space_id=42b68706-c255-41ca-87bf-bb...`. The 'Code snippets' tab is also visible, showing a Python script for making API calls. The right sidebar provides additional details about the deployment, including its creation and update times, deployment ID, software specification (runtime-22.1-py3.9), and associated asset.

The screenshot shows the IBM Watson Studio interface with the 'Projects / Machine Learning Based Vehicle ... / VehiclePerformanceAnalysisModel' path. The main content area displays two code snippets. The first snippet, titled 'Importing Libraries', shows the following code:

```
In [1]: import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import statsmodels.formula.api as smf
```

The second snippet, titled 'Importing Dataset', shows the following code:

```
In [4]: import os, types
import pandas as pd
from botocore.client import Config
import ibm_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 = ibm_boto3.client(service_name='s3',
    ibm_api_key_id='tbfWf2aSCCB8th6613YrkIVL15GkP2b7U9HQ177ZkE36',
    ibm_auth_endpoint='https://iam.cloud.ibm.com/oidc/token',
    config=Config(signature_version='oauth'),
    endpoint_url='https://s3.private.eu.cloud-object-storage.appdomain.cloud')

bucket = 'machinelearningbasedvehicleperfor-donotdelete-pr-u0yovtj1ysrhov'
object_key = 'car performance.csv'
```

```
1 import requests
2
3 # NOTE: you must manually set API_KEY below using information retrieved from your IBM Cloud account.
4 API_KEY = "VIJAXb1Vp23FVn6FxaWNFEECIbJRWptpHaal7jNGzuTE"
5 token_response = requests.post("https://iam.cloud.ibm.com/identity/token", data={"apikey":
6     API_KEY, "grant_type": "urn:ibm:params:oauth:grant-type:apikey"})
7 mltoken = token_response.json()["access_token"]
8
9 header = {'Content-Type': 'application/json', 'Authorization': 'Bearer ' + mltoken}
10
11 # NOTE: manually define and pass the array(s) of values to be scored in the next line
12 payload_scoring = {"input_data": [{"fields": [['f0', 'f1', 'f2', 'f3', 'f4', 'f5']], "values": [[8,160,380,3504,82,1]]}]}
13
14 response_scoring = requests.post('https://eu-gb.ml.cloud.ibm.com/ml/v4/deployments/f4aecc62-cd58-47a3-af62-6a940301a611/pre
15     headers={'Authorization': 'Bearer ' + mltoken})
16 print("Scoring response")
17 print(response_scoring.json())
18 pred=response_scoring.json()
19 output=pred['predictions'][0]['values'][0][0]
20 print(output)
```

PS D:\ibm\spring\ibm cloud\Performance Prediction> python scoringendpoint.py

Scoring response

```
{'predictions': [{'fields': ['prediction'], 'values': [[17.7]]}]}
```

17.7

PS D:\ibm\spring\ibm cloud\Performance Prediction>

```
17 @app.route('/')
18 def home():
19     return render_template('index.html')
20
21 @app.route('/y_predict', methods=['POST'])
22 def y_predict():
23     """
24     For rendering results on HTML GUI
25     """
26     x_test = [[int(x) for x in request.form.values()]]
27     print(x_test)
28     #sc = load('scalar.save')
29     payload_scoring = {"input_data": [{"fields": [['f0', 'f1', 'f2', 'f3', 'f4', 'f5']], "values": x_test }]}
30
31     response_scoring = requests.post('https://eu-gb.ml.cloud.ibm.com/ml/v4/deployments/f4aecc62-cd58-47a3-af62-6a940301a611
32     headers={'Authorization': 'Bearer ' + mltoken})
33     print("Scoring response")
34     print(response_scoring.json())
35     pred=response_scoring.json()
36     output=pred['predictions'][0]['values'][0][0]
37     print(output)
38     if(output<9):
39         ped="Worst performance with mileage " + str(output) + ". Carry extra fuel"
40     if(output>9 and output<=17.5):
41         ped="Low performance with mileage " + str(output) + ". Don't go to long distance"
42     if(output>17.5 and output<=29):
43         ped="Medium performance with mileage " + str(output) + ". Go for a ride nearby."
```

WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.

Running on <http://127.0.0.1:5000>

Press CTRL+C to quit

- * Restarting with stat
- * Debugger is active!
- * Debugger PIN: 282-314-124

