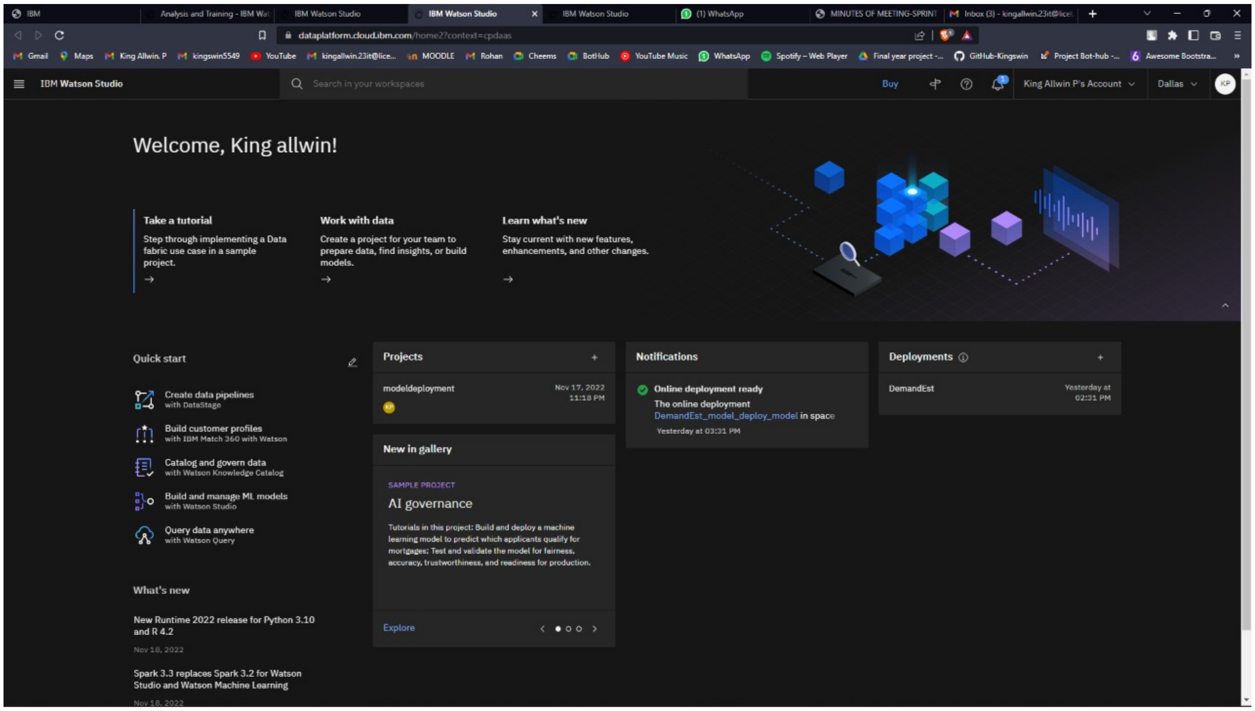
Sprint IV
Train the Model in IBM

Date	18 November 2022
Team ID	PNT2022TMID27576
Project Name	DemandEst - AI powered Food Demand Forecaster

Screenshots:

1. **Register for IBM Cloud**
2. **Train the ML model on IBM**
3. **Integrate Flask with Scoring Endpoint**

Register for IBM Cloud



Welcome, King allwin!

Take a tutorial
Step through implementing a Data fabric use case in a sample project.
→

Work with data
Create a project for your team to prepare data, find insights, or build models.
→

Learn what's new
Stay current with new features, enhancements, and other changes.
→

Quick start

Create data pipelines
with DataStage

Build customer profiles
with IBM Match 360 with Watson

Catalog and govern data
with Watson Knowledge Catalog

Build and manage ML models
with Watson Studio

Query data anywhere
with Watson Query

What's new

New Runtime 2022 release for Python 3.10 and R 4.2
Nov 18, 2022

Spark 3.3 replaces Spark 3.2 for Watson Studio and Watson Machine Learning
Nov 18, 2022

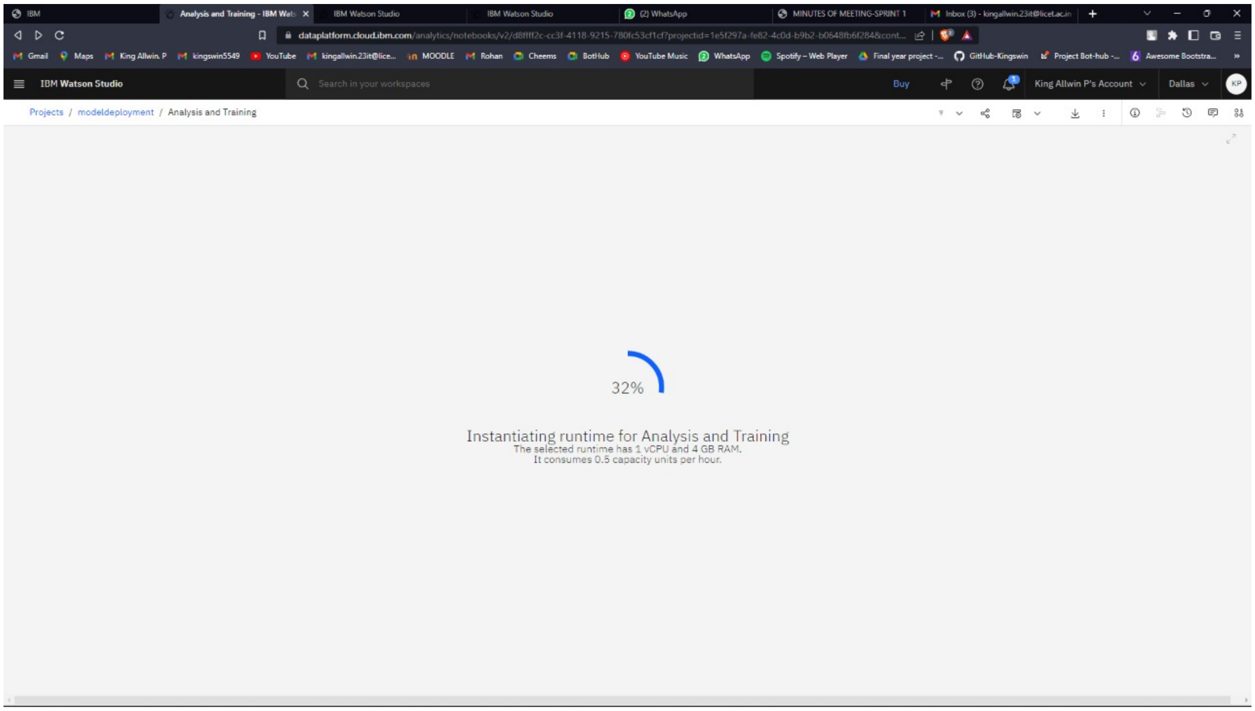
Projects +
modeldeployment — Nov 17, 2022 11:18 PM

New in gallery
SAMPLE PROJECT
AI governance
Tutorials in this project: Build and deploy a machine learning model to predict which applicants qualify for mortgages; Test and validate the model for fairness, accuracy, trustworthiness, and readiness for production.
Explore

Notifications
Online deployment ready
The online deployment DemandEst_model_deploy_model in space
Yesterday at 03:31 PM

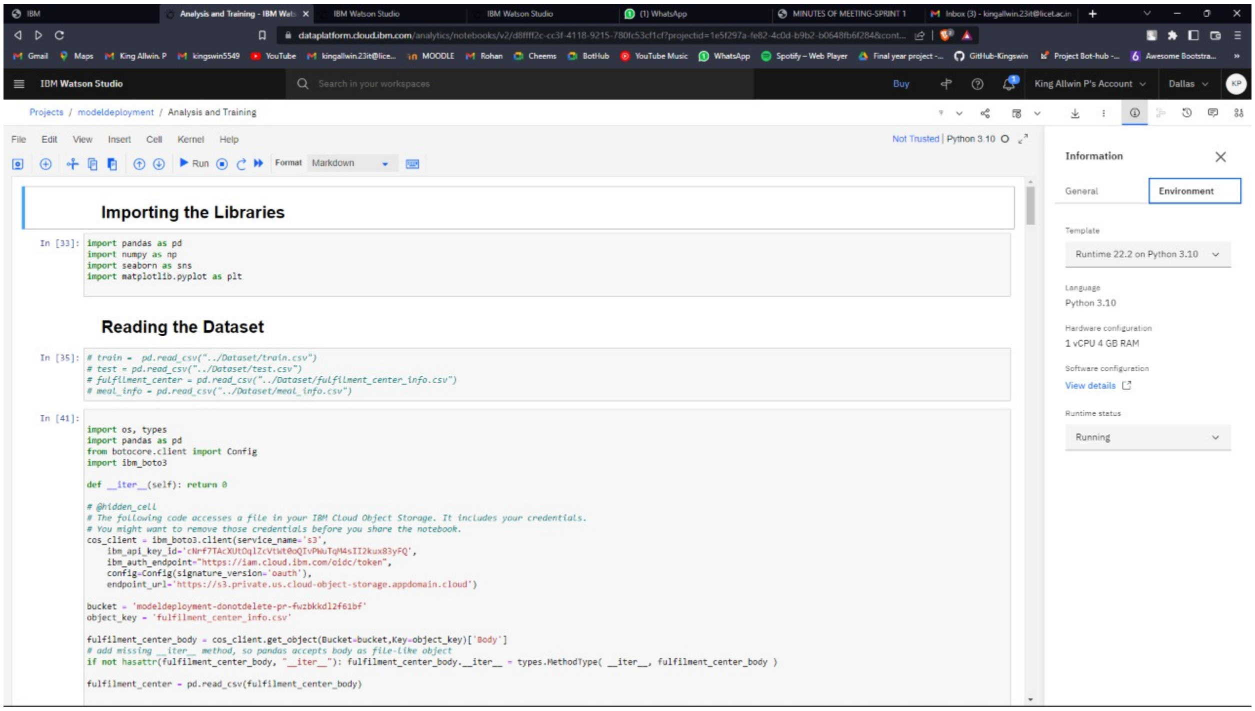
Deployments +
DemandEst — Yesterday at 02:31 PM

Train the ML model on IBM



32%

Instantiating runtime for Analysis and Training
The selected runtime has 1 vCPU and 4 GB RAM.
It consumes 0.5 capacity units per hour.



Importing the Libraries

```
In [33]: import pandas as pd
         import numpy as np
         import seaborn as sns
         import matplotlib.pyplot as plt
```

Reading the Dataset

```
In [35]: # train =  pd.read_csv("../Dataset/train.csv")
         # test = pd.read_csv("../Dataset/test.csv")
         # fulfilment_center = pd.read_csv("../Dataset/fulfilment_center_info.csv")
         # meal_info = pd.read_csv("../Dataset/meal_info.csv")
```

```
In [41]:
         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='cNrf7TAcXUtOqlZcVtWt0oQIvPWuTqM4sII2kux83yFQ',
             ibm_auth_endpoint="https://iam.cloud.ibm.com/oidc/token",
             config=Config(signature_version='oauth'),
             endpoint_url='https://s3.private.us.cloud-object-storage.appdomain.cloud')

         bucket = 'modeldeployment-donotdelete-pr-fwzbkkdl2f61bf'
         object_key = 'fulfilment_center_info.csv'

         fulfilment_center_body = cos_client.get_object(Bucket=bucket,Key=object_key)['Body']
         # add missing __iter__ method, so pandas accepts body as file-like object
         if not hasattr(fulfilment_center_body, "__iter__"): fulfilment_center_body.__iter__ = types.MethodType( __iter__, fulfilment_center_body )

         fulfilment_center = pd.read_csv(fulfilment_center_body)
```

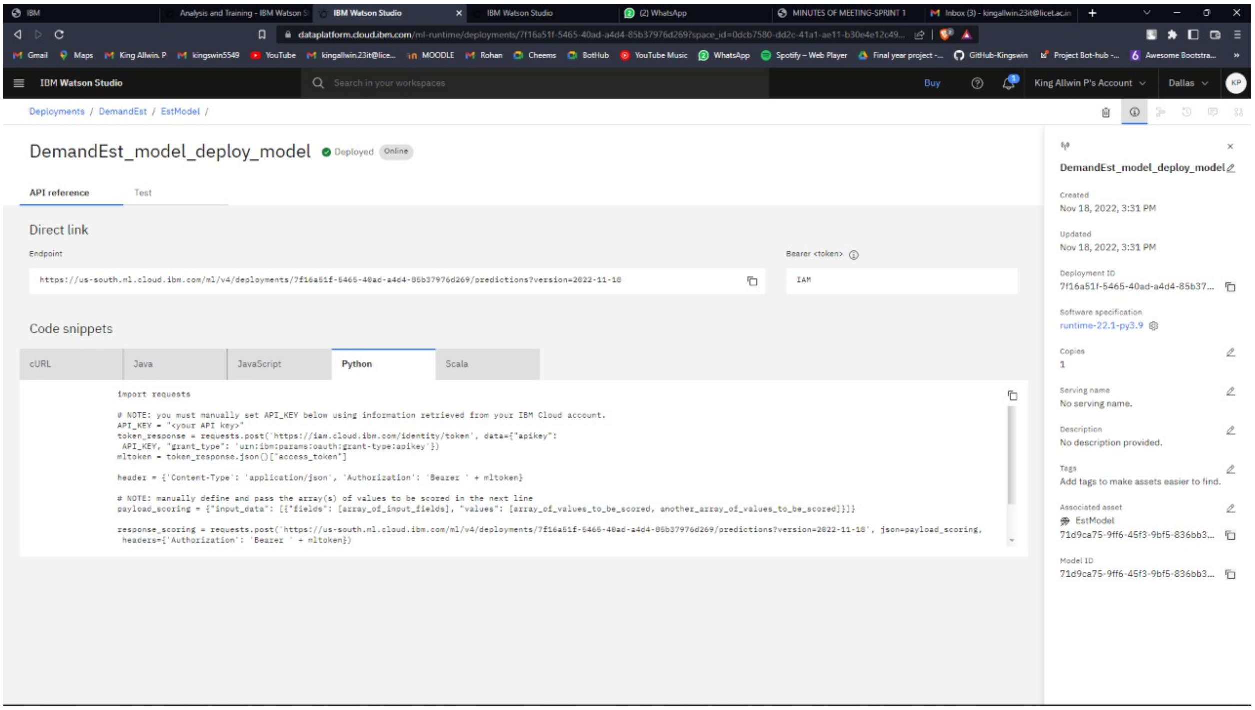
Template: Runtime 22.2 on Python 3.10

Language: Python 3.10

Hardware configuration: 1 vCPU 4 GB RAM

Software configuration: View details

Runtime status: Running


IBM Watson Studio
Search in your workspaces
Buy
King Allwin P's Account
Dallas
Deployments / DemandEst / EstModel /

DemandEst_model_deploy_model

Deployed Online

API reference | Test

Direct link

Endpoint

```
https://us-south.ml.cloud.ibm.com/ml/v4/deployments/7f16a51f-5465-40ad-a4d4-85b37976d269/predictions?version=2022-11-18
```

Bearer <token>

IAM

Code snippets

cURL | Java | JavaScript | Python | Scala

```
import requests

# NOTE: you must manually set API_KEY below using information retrieved from your IBM Cloud account.
API_KEY = "<your API key>"
token_response = requests.post('https://iam.cloud.ibm.com/identity/token', data={"apikey":
 API_KEY, "grant_type": 'urn:ibm:params:oauth:grant-type:apikey'})
mltoken = token_response.json()["access_token"]

header = {'Content-Type': 'application/json', 'Authorization': 'Bearer ' + mltoken}

# NOTE: manually define and pass the array(s) of values to be scored in the next line
payload_scoring = {"input_data": [{"fields": [array_of_input_fields], "values": [array_of_values_to_be_scored, another_array_of_values_to_be_scored]}]}

response_scoring = requests.post('https://us-south.ml.cloud.ibm.com/ml/v4/deployments/7f16a51f-5465-40ad-a4d4-85b37976d269/predictions?version=2022-11-18', json=payload_scoring,
 headers={'Authorization': 'Bearer ' + mltoken})
```

DemandEst_model_deploy_model

Created: Nov 18, 2022, 3:31 PM

Updated: Nov 18, 2022, 3:31 PM

Deployment ID: 7f16a51f-5465-40ad-a4d4-85b37...

Software specification: runtime-22.1-py3.9

Copies: 1

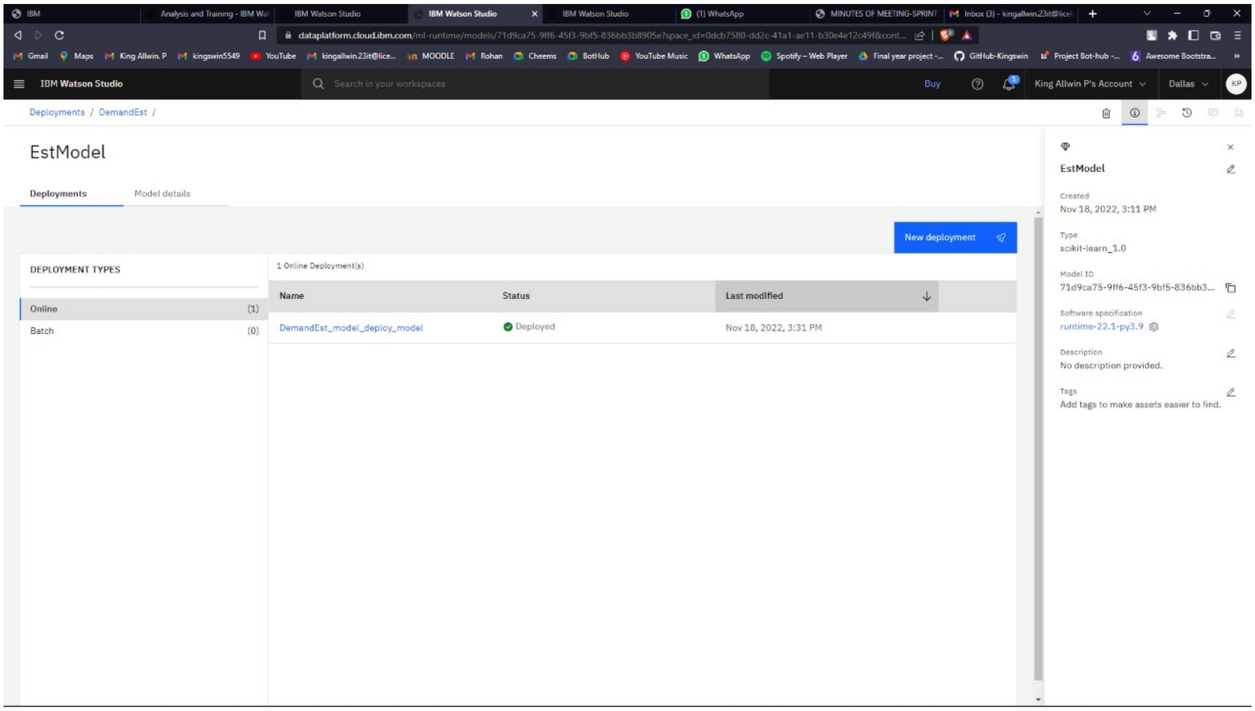
Serving name: No serving name.

Description: No description provided.

Tags: Add tags to make assets easier to find.

Associated asset: EstModel 71d9ca75-9ff6-45f3-9bf5-836bb3...

Model ID: 71d9ca75-9ff6-45f3-9bf5-836bb3...



EstModel

Deployments | Model details

DEPLOYMENT TYPES

Online	(1)
Batch	(0)

1 Online Deployment(s)

Name	Status	Last modified
DemandEst_model_deploy_model	Deployed	Nov 18, 2022, 3:31 PM

EstModel

Created: Nov 18, 2022, 3:11 PM

Type: scikit-learn_1.0

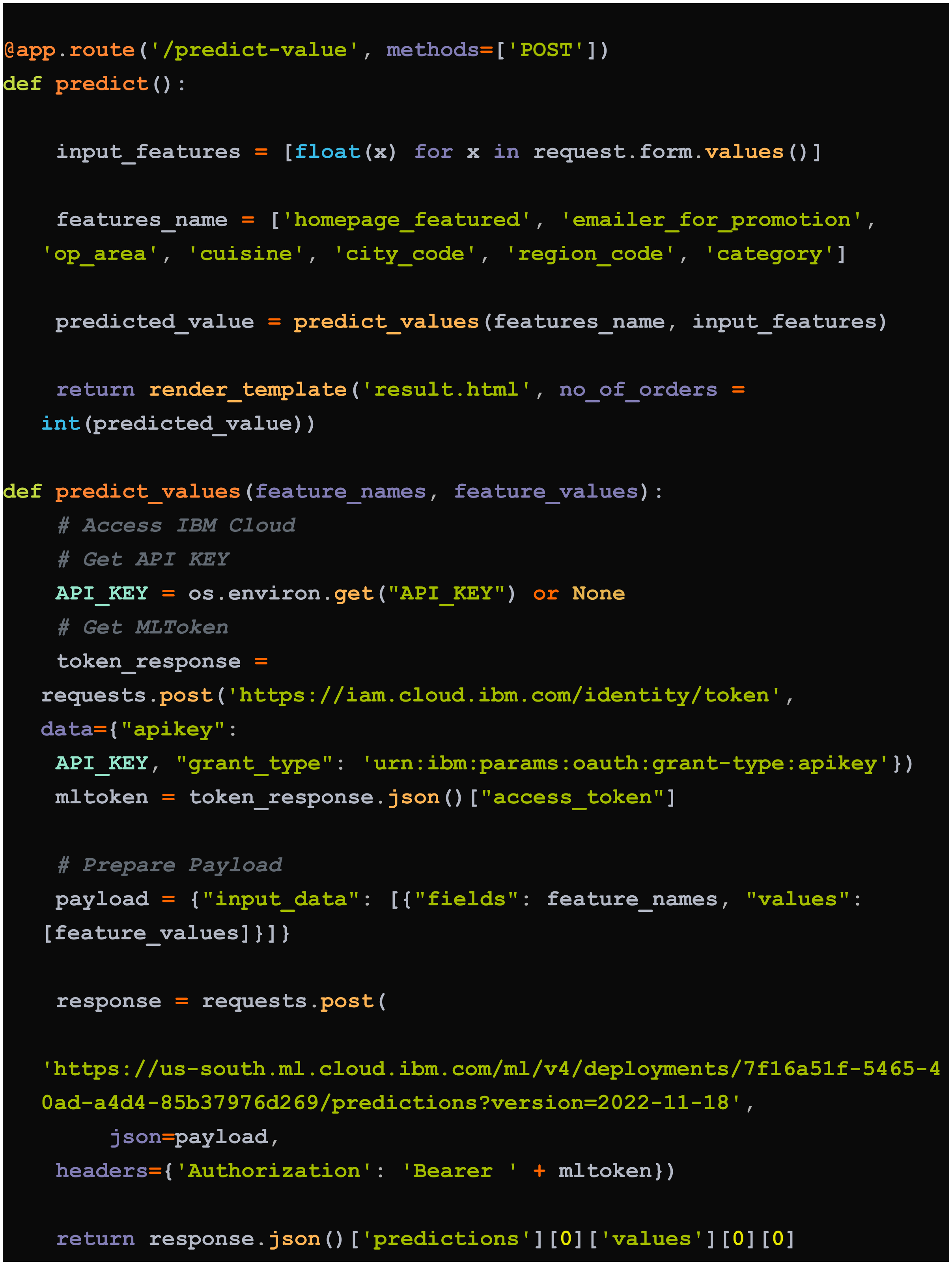
Model ID: 71d9ca75-9ff6-45f3-9bf5-836bb3...

Software specification: runtime-22.1-py3.9

Description: No description provided.

Tags: Add tags to make assets easier to find.

Integrate Flask with Scoring Endpoint



```
@app.route('/predict-value', methods=['POST'])
def predict():

    input_features = [float(x) for x in request.form.values()]

    features_name = ['homepage_featured', 'emailer_for_promotion',
   'op_area', 'cuisine', 'city_code', 'region_code', 'category']

    predicted_value = predict_values(features_name, input_features)

    return render_template('result.html', no_of_orders =
   int(predicted_value))

def predict_values(feature_names, feature_values):
    # Access IBM Cloud
    # Get API KEY
    API_KEY = os.environ.get("API_KEY") or None
    # Get MLToken
    token_response =
   requests.post('https://iam.cloud.ibm.com/identity/token',
   data={"apikey":
    API_KEY, "grant_type": 'urn:ibm:params:oauth:grant-type:apikey'})
    mltoken = token_response.json()["access_token"]

    # Prepare Payload
    payload = {"input_data": [{"fields": feature_names, "values":
   [feature_values]}]}

    response = requests.post(

   'https://us-south.ml.cloud.ibm.com/ml/v4/deployments/7f16a51f-5465-4
   0ad-a4d4-85b37976d269/predictions?version=2022-11-18',
        json=payload,
    headers={'Authorization': 'Bearer ' + mltoken})

    return response.json()['predictions'][0]['values'][0][0]
```