

Unit 10 : In-Class Hands-On Lab

Sample Machine Learning model deployed as a Rest Service

Context

As data scientists, you will spend considerable time in training , validating and refining your machine learning models in a development environment. After you build your models, you still need to ensure your model is available to the end users / applications.

The Rest Service framework is typically used to deploy machine model as a consumable service. The consumers / client applications can leverage your model by simply calling the Rest Service without being burdened by the complexity involved in building the underlying model.

The In-Class Hands-On lab for Unit 10 will focus on creating such a rest service around a sample machine learning model.

Data Set and Source Code

We will use the Ames Housing data set for the exercise. A simple machine learning model has already been trained for this exercise. The model is trained in the Jupyter Notebook environment and saved to the file system using the "pickle" library. In Python, pickling is a standard way to store objects and retrieve them as their original state.

We will use Python Flask.

The exercise has 2 files:

1. main.py : the code to create a rest service which renders the machine learning model
2. PostRequest.py : code to simulate client applications calling the rest service for the input contained in NewHouse.json

The following python packages have to be installed as dependencies

requests	(pip install requests)
Flask	pip install flask
numpy	pip install numpy
sklearn	pip install sklearn
scipy	pip install scipy

Start the rest service by running
`python main.py`

once the service is successfully started post a client request using the command

`python PostRequest.py`

Online version of the Rest Service

A version of this service also running on the internet , deployed on the IBM Watson Cloud @
<http://amespricing.mybluemix.net/api/getSalesPrice>

The following online rest client can be used for testing this service on the internet :

<https://client.restlet.com/>
content-type : application/json