

Deploy a predictive service

5 minutes

After you've created and tested an inference pipeline for real-time inferencing, you can publish it as a service for client applications to use.

ⓘ Note

In this exercise, you'll deploy the web service to an Azure Container Instance (ACI). This type of compute is created dynamically, and is useful for development and testing. For production, you should create an *inference cluster*, which provides an Azure Kubernetes Service (AKS) cluster that provides better scalability and security.



Deploy a service


1. View the **Predict Auto Price** inference pipeline you created in the previous unit.
2. At the top right, select **Deploy**, and deploy a new real-time endpoint, using the following settings:
 - **Name:** predict-auto-price
 - **Description:** Auto price regression.
 - **Compute type:** Azure Container Instance
3. Wait for the web service to be deployed - this can take several minutes. The deployment status is shown at the top left of the designer interface.

Test the service

Now you can test your deployed service from a client application - in this case, you'll use the code in the cell below to simulate a client application.

1. On the **Endpoints** page, open the **predict-auto-price** real-time endpoint.
2. When the **predict-auto-price** endpoint opens, view the **Consume** tab and note the following information there. You need this to connect to your deployed service from a client application.

- The REST endpoint for your service
 - The Primary Key for your service
3. Observe that you can use the  link next to these values to copy them to the clipboard.
 4. With the **Consume** page for the **predict-auto-price** service page open in your browser, open a new browser tab and open a second instance of [Azure Machine Learning studio](#) . Then in the new tab, view the **Notebooks** page (under **Author**).
 5. In the **Notebooks** page, under **My files**, use the  button to create a new file with the following settings:
 - **File location:** Users/*your user name*
 - **File name:** Test-Autos
 - **File type:** Notebook
 - **Overwrite if already exists:** Selected
 6. When the new notebook has been created, ensure that the compute instance you created previously is selected in the **Compute** box, and that it has a status of **Running**.
 7. Use the << button to collapse the file explorer pane and give you more room to focus on the **Test-Autos.ipynb** notebook tab.
 8. In the rectangular cell that has been created in the notebook, paste the following code:

Python  Copy

```
endpoint = 'YOUR_ENDPOINT' #Replace with your endpoint
key = 'YOUR_KEY' #Replace with your key

import urllib.request
import json
import os

# Prepare the input data
data = {
    "Inputs": {
        "WebServiceInput0":
            [
                {
                    'symboling': 3,
                    'normalized-losses': None,
                    'make': "alfa-romero",
                    'fuel-type': "gas",
                    'aspiration': "std",
                    'num-of-doors': "two",
                    'body-style': "convertible",
                    'drive-wheels': "rwd",
                    'engine-location': "front",
```

```

        'wheel-base': 88.6,
        'length': 168.8,
        'width': 64.1,
        'height': 48.8,
        'curb-weight': 2548,
        'engine-type': "dohc",
        'num-of-cylinders': "four",
        'engine-size': 130,
        'fuel-system': "mpfi",
        'bore': 3.47,
        'stroke': 2.68,
        'compression-ratio': 9,
        'horsepower': 111,
        'peak-rpm': 5000,
        'city-mpg': 21,
        'highway-mpg': 27,
    },
],
},
"GlobalParameters": {
}
}
body = str.encode(json.dumps(data))
headers = {'Content-Type': 'application/json', 'Authorization': ('Bearer ' +
key)}
req = urllib.request.Request(endpoint, body, headers)

try:
    response = urllib.request.urlopen(req)
    result = response.read()
    json_result = json.loads(result)
    y = json_result["Results"]["WebServiceOutput0"][0]
    print(y)

except urllib.error.HTTPError as error:
    print("The request failed with status code: " + str(error.code))

    # Print the headers to help debug the error
    print(error.info())
    print(json.loads(error.read().decode("utf8", 'ignore')))
```

❗ Note

Don't worry too much about the details of the code. It just submits details of a car and uses the **predict-auto-price** service you created to get a predicted price.

- Switch to the browser tab containing the **Consume** page for the **predict-auto-price** service, and copy the REST endpoint for your service. Then switch back to the tab containing the notebook and paste the key into the code, replacing **YOUR_ENDPOINT**.

10. Switch to the browser tab containing the **Consume** page for the **predict-auto-price** service, and copy the Primary Key for your service. Then switch back to the tab containing the notebook and paste the key into the code, replacing YOUR_KEY.
 11. Save the notebook. Then use the ▶ button next to the cell to run the code.
 12. Verify that predicted price is returned.
-

Next unit: Knowledge check

Continue >
