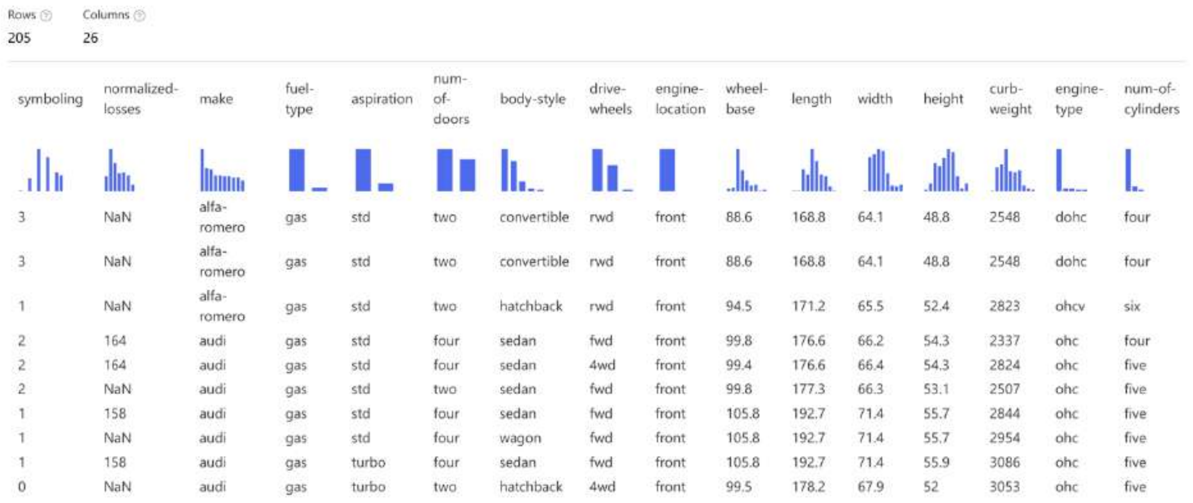
Automobile Price Prediction with Azure Machine Learning Studio

Overview

In this project, we utilized Microsoft Azure Machine Learning Designer to create and deploy a regression model for predicting automobile prices. The process involved setting up the Azure workspace and compute resources, developing and evaluating a regression model pipeline, and creating an inference pipeline for real-time predictions. The predictive service was then deployed to an Azure Container Instance and tested to ensure accurate price predictions based on automobile characteristics.

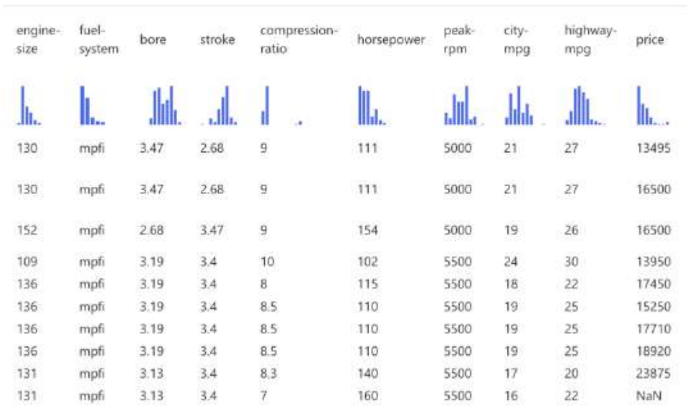
Dataset

The dataset used in this project is the "Automobile price data (Raw)" provided in Azure Machine Learning Designer's sample datasets. This dataset includes various features related to automobile characteristics, which are essential for training the regression model to predict vehicle prices based on these attributes.



Rows 205 Columns 26

symboling	normalized-losses	make	fuel-type	aspiration	num-of-doors	body-style	drive-wheels	engine-location	wheel-base	length	width	height	curb-weight	engine-type	num-of-cylinders
3	NaN	alfa-romero	gas	std	two	convertible	rwd	front	88.6	168.8	64.1	48.8	2548	dohc	four
3	NaN	alfa-romero	gas	std	two	convertible	rwd	front	88.6	168.8	64.1	48.8	2548	dohc	four
1	NaN	alfa-romero	gas	std	two	hatchback	rwd	front	94.5	171.2	65.5	52.4	2823	ohcv	six
2	164	audi	gas	std	four	sedan	fwd	front	99.8	176.6	66.2	54.3	2337	ohc	four
2	164	audi	gas	std	four	sedan	4wd	front	99.4	176.6	66.4	54.3	2824	ohc	five
2	NaN	audi	gas	std	two	sedan	fwd	front	99.8	177.3	66.3	53.1	2507	ohc	five
1	158	audi	gas	std	four	sedan	fwd	front	105.8	192.7	71.4	55.7	2844	ohc	five
1	NaN	audi	gas	std	four	wagon	fwd	front	105.8	192.7	71.4	55.7	2954	ohc	five
1	158	audi	gas	turbo	four	sedan	fwd	front	105.8	192.7	71.4	55.9	3086	ohc	five
0	NaN	audi	gas	turbo	two	hatchback	4wd	front	99.5	178.2	67.9	52	3053	ohc	five



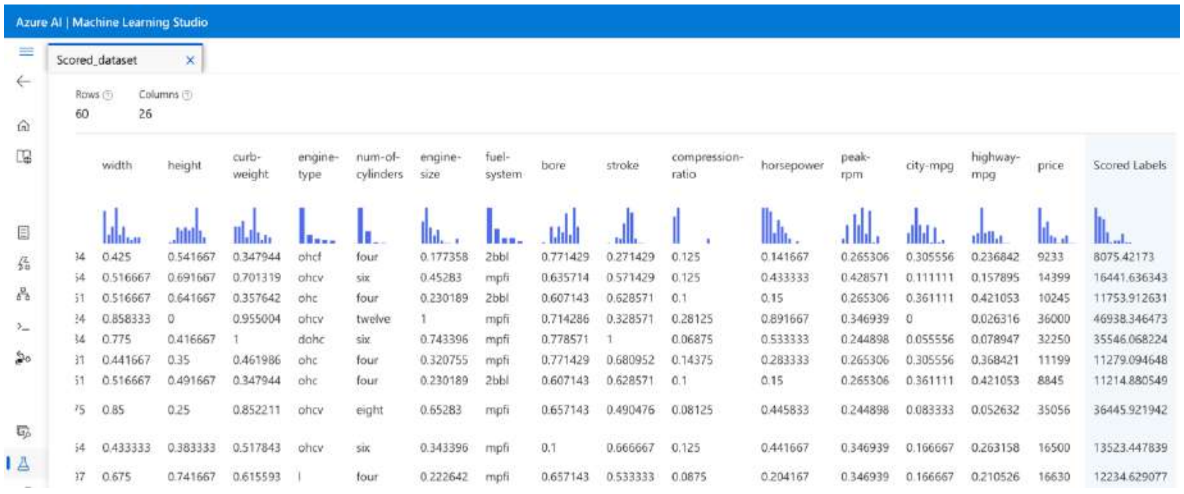
engine-size	fuel-system	bore	stroke	compression-ratio	horsepower	peak-rpm	city-mpg	highway-mpg	price
130	mpfi	3.47	2.68	9	111	5000	21	27	13495
130	mpfi	3.47	2.68	9	111	5000	21	27	16500
152	mpfi	2.68	3.47	9	154	5000	19	26	16500
109	mpfi	3.19	3.4	10	102	5500	24	30	13950
136	mpfi	3.19	3.4	8	115	5500	18	22	17450
136	mpfi	3.19	3.4	8.5	110	5500	19	25	15250
136	mpfi	3.19	3.4	8.5	110	5500	19	25	17710
136	mpfi	3.19	3.4	8.5	110	5500	19	25	18920
131	mpfi	3.13	3.4	8.3	140	5500	17	20	23875
131	mpfi	3.13	3.4	7	160	5500	16	22	NaN

1- Setting Up Azure Machine Learning Workspace and Compute Resources

Established a Microsoft Azure Machine Learning workspace to manage data, compute resources, and models efficiently. Following this, compute targets were created to provide the necessary processing power for model training and deployment. A compute instance was set up as a development workstation, and a compute cluster was configured to handle the scalable processing of machine learning tasks, ensuring a robust environment for data science activities.

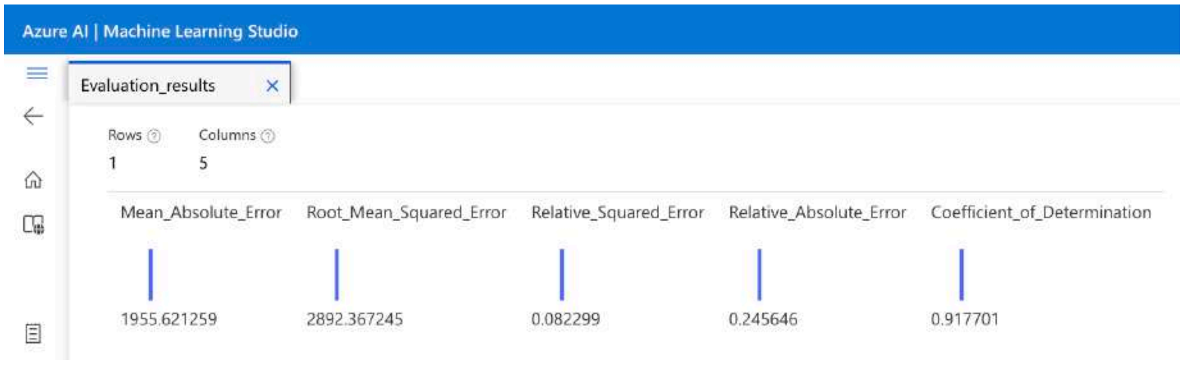
2- Regression Model Pipeline Creation and Evaluation

- **Pipeline Setup:** Created a new pipeline in Azure Machine Learning named "Auto Price Training" and selected the appropriate compute target.
- **Data Preparation:** Added and explored the "Automobile price data (Raw)" dataset, addressing missing values and normalizing numeric columns using data transformations.
- **Model Training:** Configured the pipeline to split the data into training and validation sets, trained a Linear Regression model, and scored the model's predictions.
- **Model Evaluation:** Added an evaluation module to assess the model's performance using metrics such as MAE, RMSE, RSE, RAE, and R-Squared.



Rows 60 Columns 26

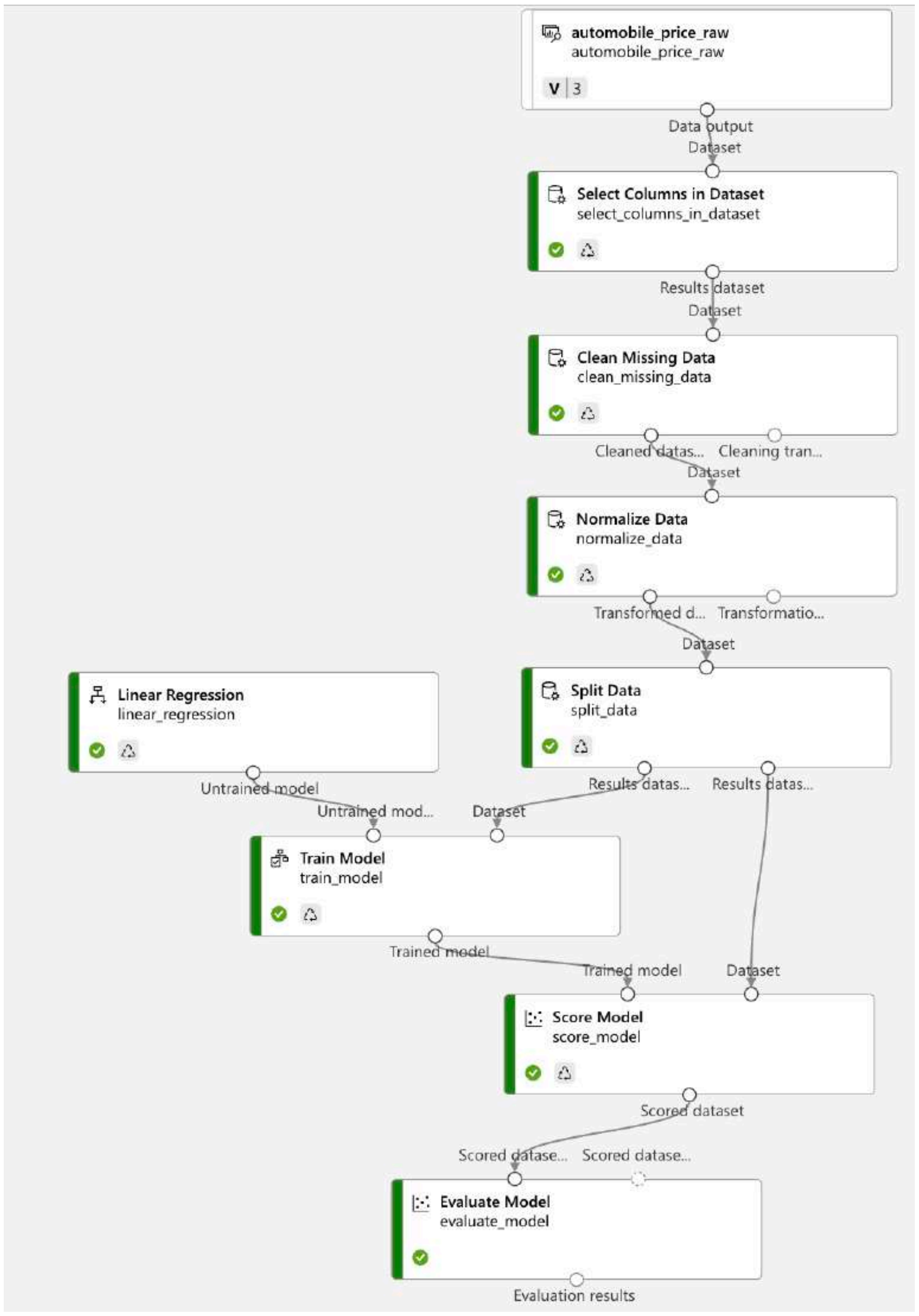
	width	height	curb-weight	engine-type	num-of-cylinders	engine-size	fuel-system	bore	stroke	compression-ratio	horsepower	peak-rpm	city-mpg	highway-mpg	price	Scored Labels
34	0.425	0.541667	0.347944	ohcf	four	0.177358	2bbl	0.771429	0.271429	0.125	0.141667	0.265306	0.305556	0.236842	9233	8075.42173
54	0.516667	0.691667	0.701319	ohcv	six	0.45283	mpfi	0.635714	0.571429	0.125	0.433333	0.428571	0.111111	0.157895	14399	16441.636343
51	0.516667	0.641667	0.357642	ohc	four	0.230189	2bbl	0.607143	0.628571	0.1	0.15	0.265306	0.361111	0.421053	10245	11753.912631
24	0.858333	0	0.955004	ohcv	twelve	1	mpfi	0.714286	0.328571	0.28125	0.891667	0.346939	0	0.026316	36000	46938.346473
34	0.775	0.416667	1	dohc	six	0.743396	mpfi	0.778571	1	0.06875	0.533333	0.244898	0.055556	0.078947	32250	35546.068224
31	0.441667	0.35	0.461986	ohc	four	0.320755	mpfi	0.771429	0.680952	0.14375	0.283333	0.265306	0.305556	0.368421	11199	11279.094648
51	0.516667	0.491667	0.347944	ohc	four	0.230189	2bbl	0.607143	0.628571	0.1	0.15	0.265306	0.361111	0.421053	8845	11214.880549
75	0.85	0.25	0.852211	ohcv	eight	0.65283	mpfi	0.657143	0.490476	0.08125	0.445833	0.244898	0.083333	0.052632	35056	36445.921942
54	0.433333	0.383333	0.517843	ohcv	six	0.343396	mpfi	0.1	0.666667	0.125	0.441667	0.346939	0.166667	0.263158	16500	13523.447839
37	0.675	0.741667	0.615593	l	four	0.222642	mpfi	0.657143	0.533333	0.0875	0.204167	0.346939	0.166667	0.210526	16630	12234.629077



Rows 1 Columns 5

Mean_Absolute_Error	Root_Mean_Squared_Error	Relative_Squared_Error	Relative_Absolute_Error	Coefficient_of_Determination
1955.621259	2892.367245	0.082299	0.245646	0.917701

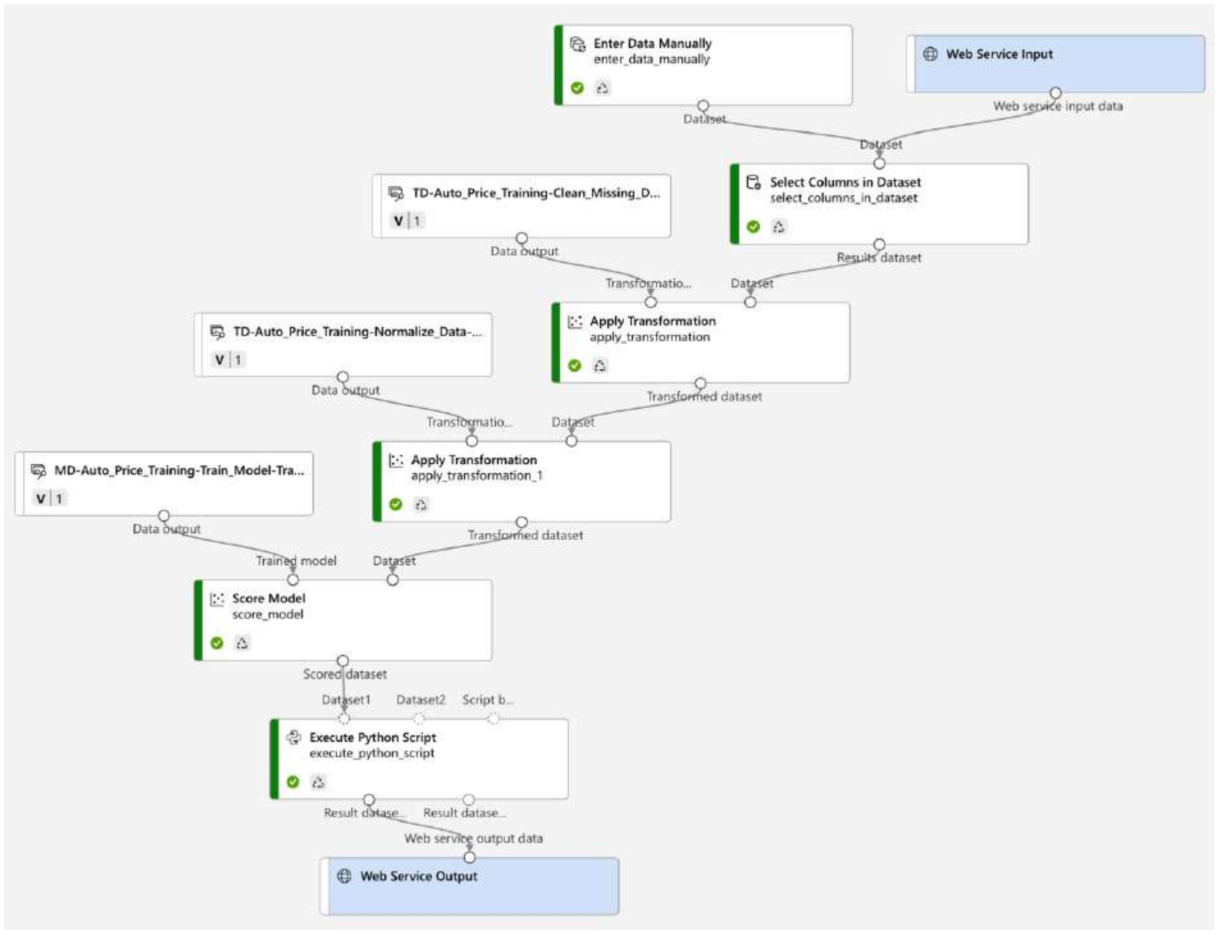
The pipeline successfully prepared data, trained a regression model, and evaluated its performance.



3- Inference Pipeline Creation

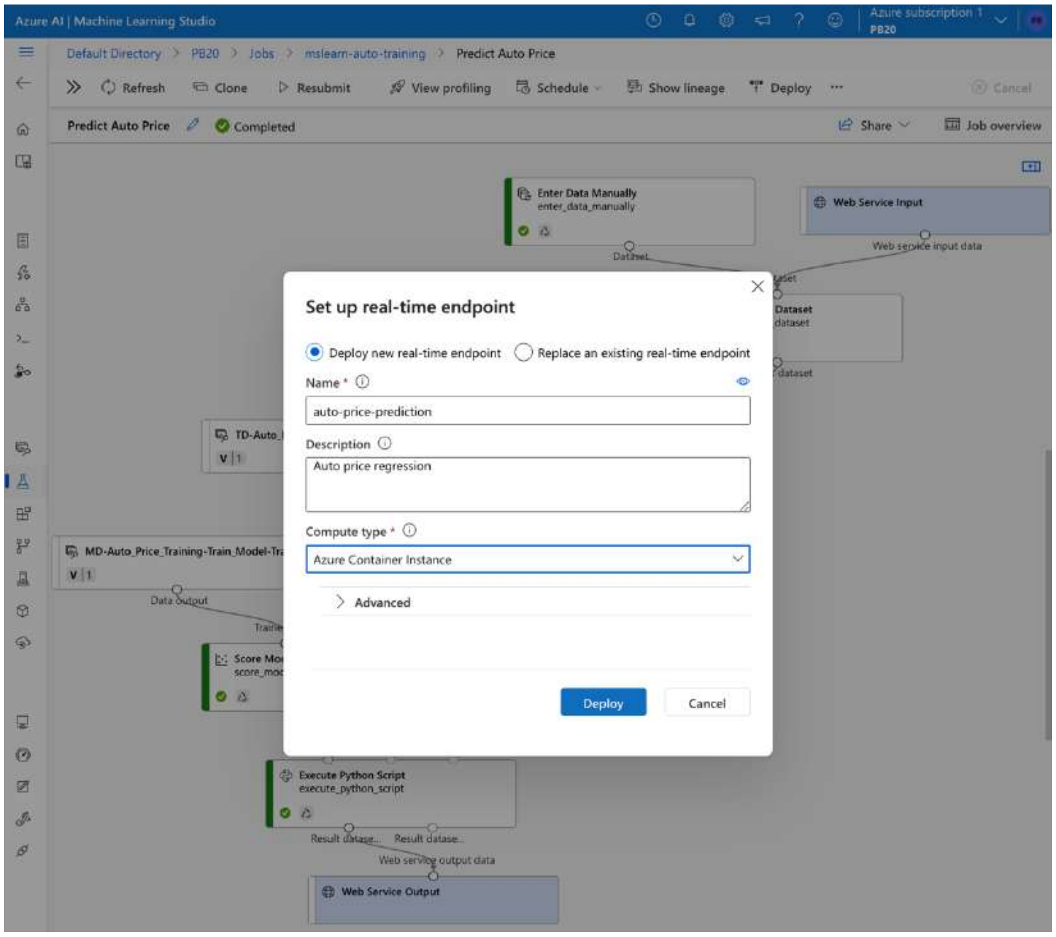
- **Pipeline Setup:** Opened the previously created "Auto Price Training" pipeline and selected the "Real-time inference pipeline" option, creating a new pipeline named "Predict Auto Price".
- **Data Transformation and Pipeline Modifications:** Replaced the dataset with an "Enter Data Manually" module for new data features, updated column selections, and removed unnecessary modules. Added an "Execute Python Script" module to extract and rename predicted labels, connecting it to the web service output.
- **Execution and Validation:** Submitted the pipeline as a new experiment on the compute cluster. Verified predictions by visualizing the output of the "Execute Python Script" module.

The inference pipeline prepares new data and applies the trained model to predict prices.



4- Deploying a Predictive Service

Deployed the "Predict Auto Price" inference pipeline by selecting "Deploy" and creating a new real-time endpoint named "auto-price-prediction" on Azure Container Instance (ACI). This deployment allows for development and testing purposes.



Set up real-time endpoint

(●) Deploy new real-time endpoint ( ) Replace an existing real-time endpoint

Name *: auto-price-prediction

Description: Auto price regression

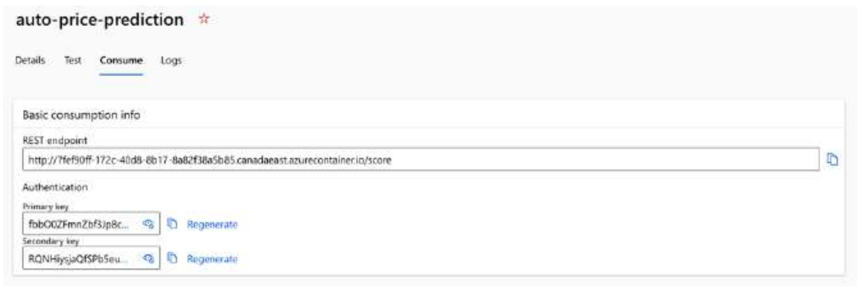
Compute type *: Azure Container Instance

Advanced

Deploy Cancel

5- Real-Time Endpoint Testing

Accessed the "auto-price-prediction" endpoint on the Endpoints page to retrieve the REST endpoint and Primary Key.



auto-price-prediction

Basic consumption info

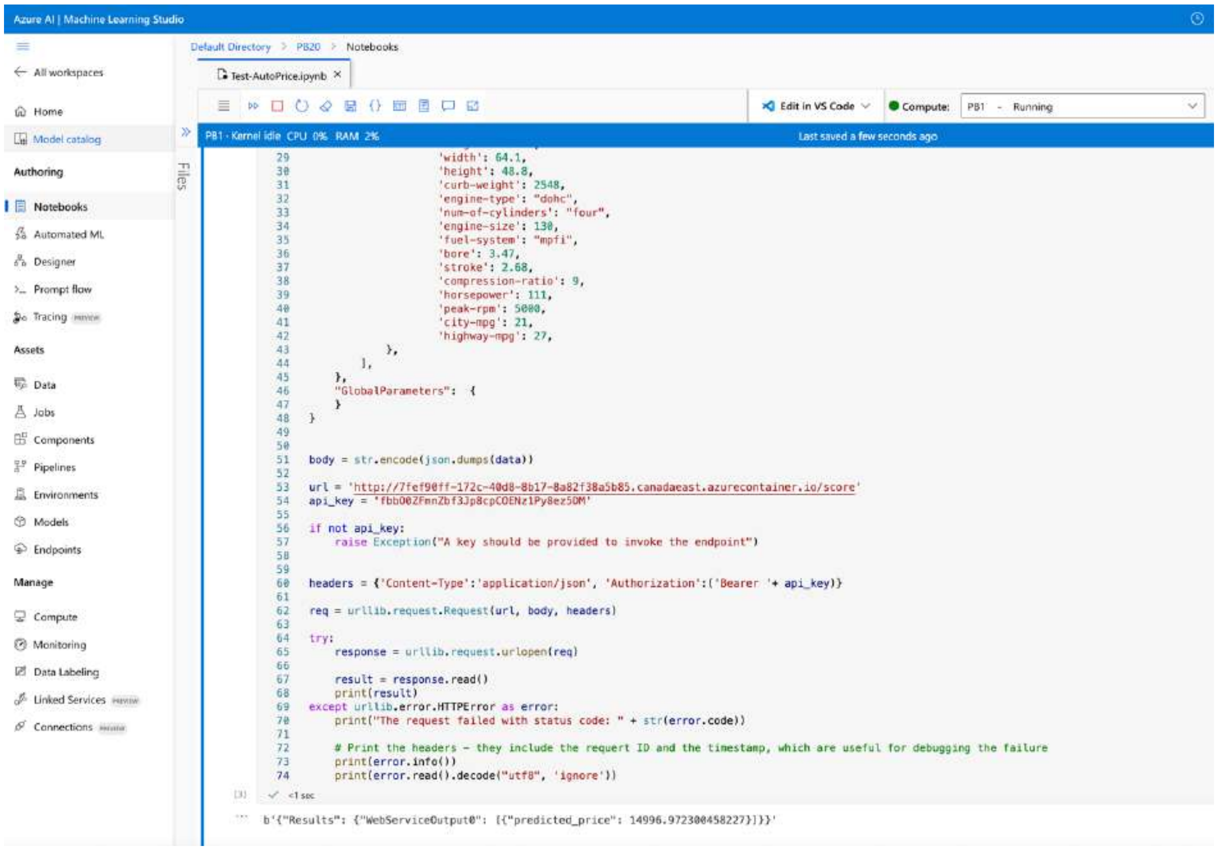
REST endpoint: http://7fef90ff-172c-40d8-8b17-8a82f38a5b85.canadaeast.azurecontainer.io/score

Authentication

Primary key: fbbO0ZFmnZbf3Jp8c...

Secondary key: RQNHiysjaQfSPb5eu...

Tested the deployed service by retrieving the REST endpoint and Primary Key, then using these details in a new notebook within Azure Machine Learning Studio to run a test and confirm that the service accurately predicts prices.



```
                        '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))

url = 'http://7fef90ff-172c-40d8-8b17-8a82f38a5b85.canadaeast.azurecontainer.io/score'
api_key = 'fbb00ZFmnZbf3Jp8cpCOENz1Py8ez5OM'

if not api_key:
    raise Exception("A key should be provided to invoke the endpoint")


headers = {'Content-Type':'application/json', 'Authorization':('Bearer '+ api_key)}

req = urllib.request.Request(url, body, headers)

try:
    response = urllib.request.urlopen(req)

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

    # Print the headers - they include the requert ID and the timestamp, which are useful for debugging the failure
    print(error.info())
    print(error.read().decode("utf8", 'ignore'))
```

```
b'{"Results": {"WebServiceOutput0": [{"predicted_price": 14996.972300458227}]}}'
```

The deployment and testing process ensures that the predictive service is operational and accessible for client applications, providing real-time price predictions based on the trained model.