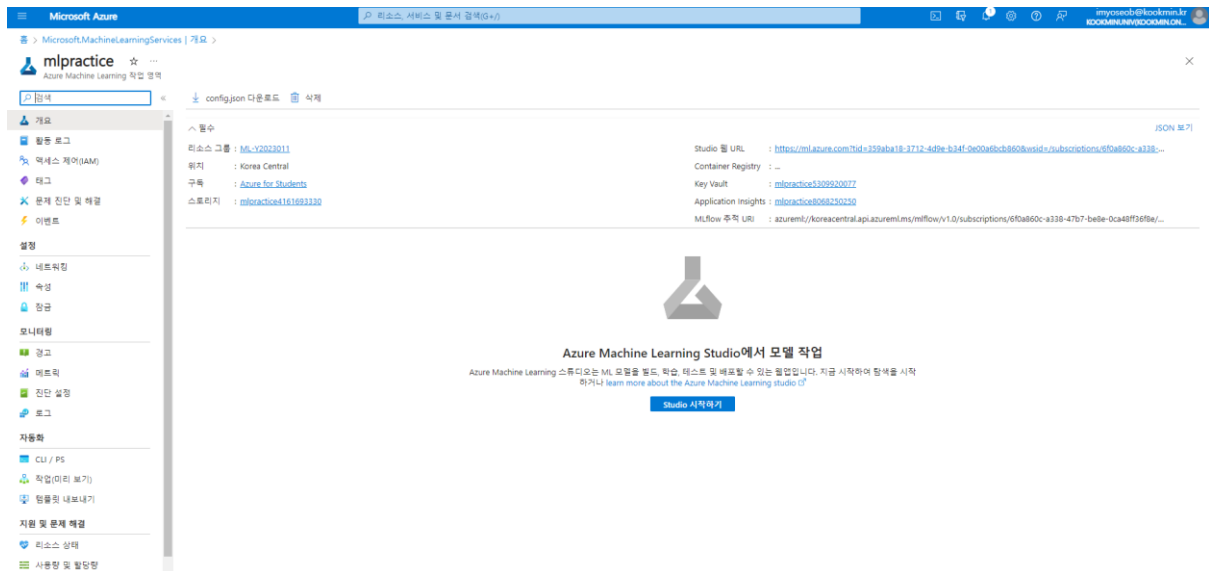


클라우드 데이터 & 인공지능 입문 10주차 과제: 윤요섭(Y2023011)

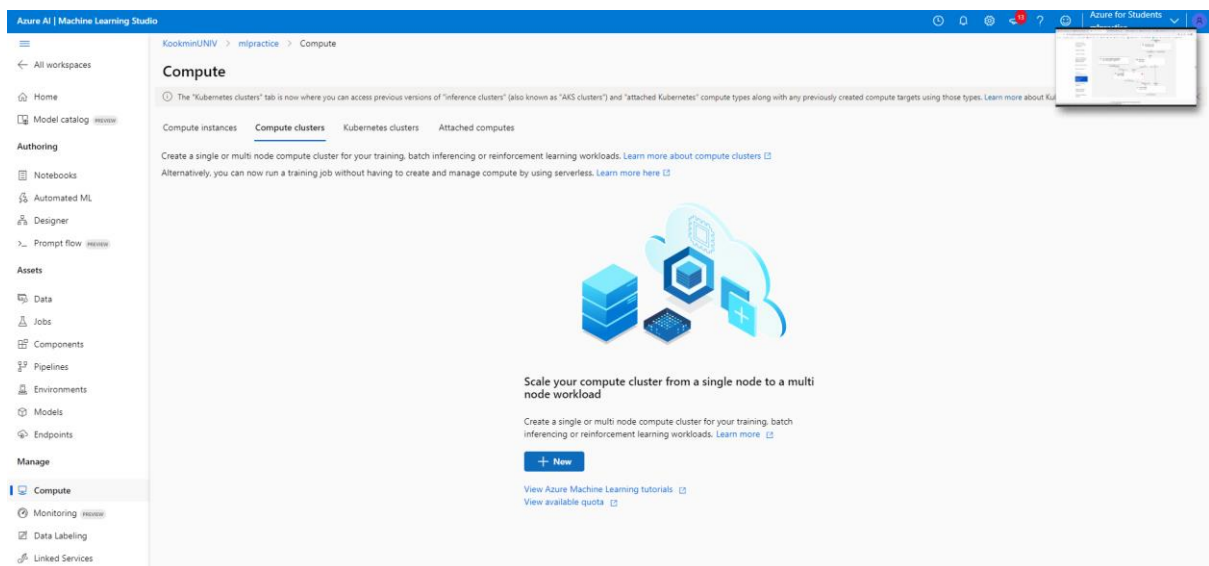
Azure Machine Learning 작업 영역 만들기

Azure Machine Learning 리소스를 만듭니다.



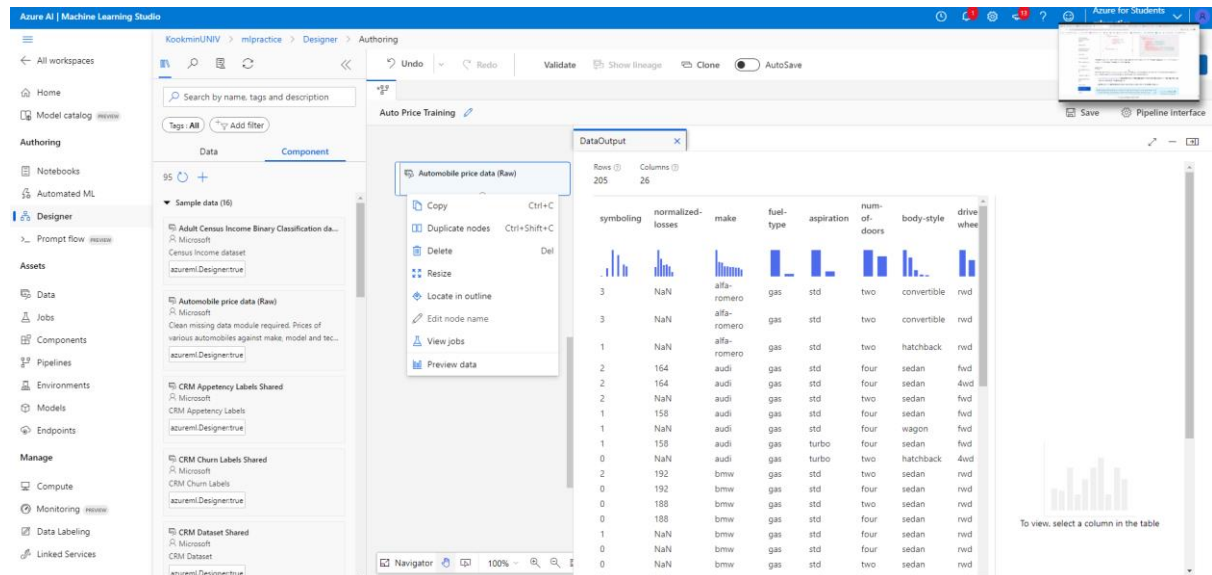
Machine Learning Studio에서 Compute Clusters를 만듭니다.

좌측 하단에 Compute를 클릭한 뒤, Compute Clusters를 클릭하여 새로운 Compute Clusters를 만듭니다.



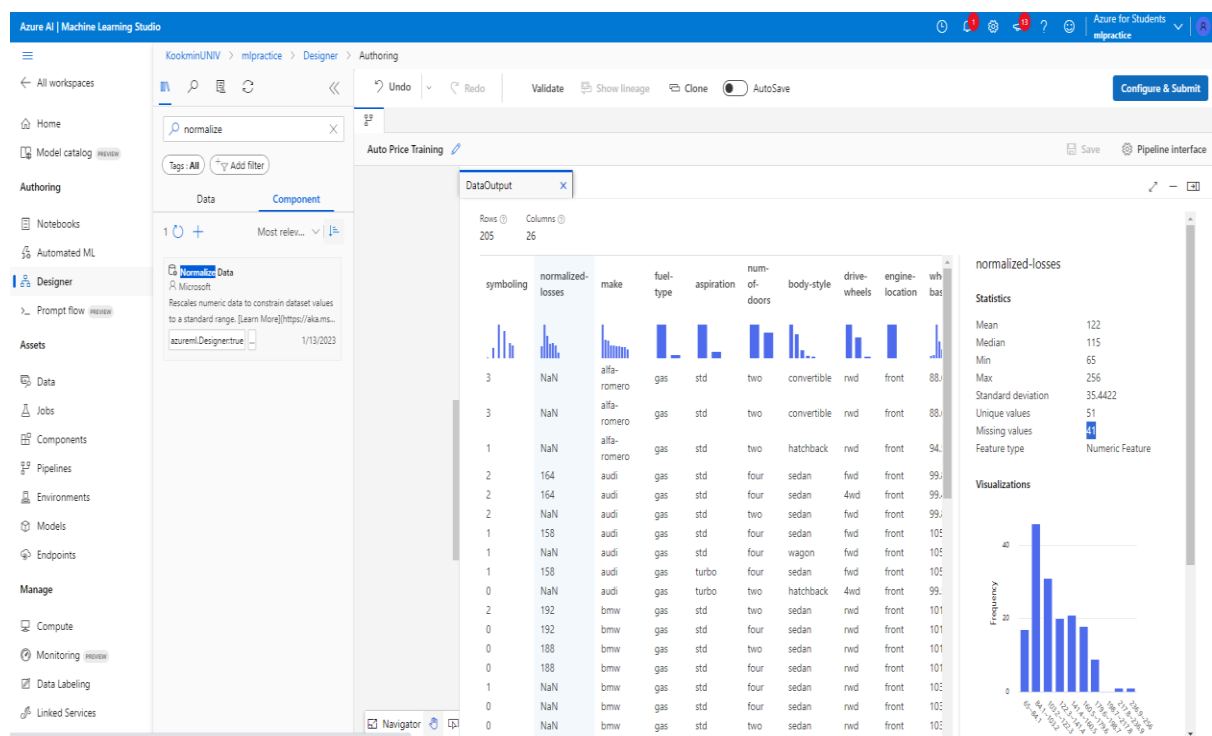
Designer에서 파이프라인 생성을 생성하고 데이터 세트를 추가합니다.

좌측 탭에서 Designer를 선택 한 뒤, 컴포넌트에서 Automobile price data(Raw) 데이터를 사용하기를 클릭한 뒤, Preview Data를 클릭하여 Data Output을 미리봅니다.



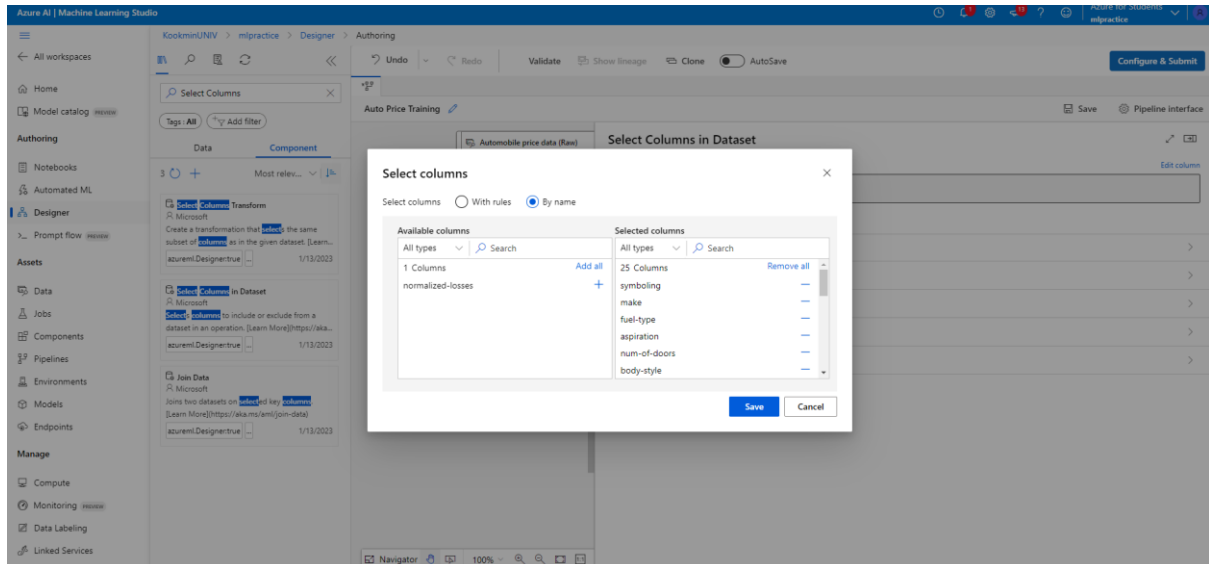
데이터 살펴보기

Automobile price data (Raw)의 normalized-losses를 보면, 결측치가 많은 것을 확인 할 수 있습니다. 데이터 전처리 시에 결측치를 제거하거나, 결측치가 많은 해당 열을 제거하는 것을 고려해볼 수 있습니다.



데이터 변환하기

Asset library에서 Select Coluns in Dataset Component를 선택하여, 캔버스에 추가한 뒤, 결측치가 많았던, normalized-losses 컬럼을 제거해주고 저장합니다.



Clean Missing Data module

결측치가 많은 데이터의 열을 제거한 이후로도, 결측치가 있는지 확인해봅니다.

1) 결측치 있는 컬럼 정리

결측치가 있는 컬럼 이름	결측치 개수
num-of-doors	2
bore	4
stroke	4
horsepower	2
peak-rpm	2

Clean Missing Data module에서 Columns to be cleaned를 클릭하여, 결측치가 있는 항목의 경우, 해당 row를 제거해줍니다.

Normalize Data

engine-size

Statistics

Mean	126.9073
Median	120
Min	61
Max	326
Standard deviation	41.6427
Unique values	44
Missing values	0
Feature type	Numeric Feature

width

Statistics

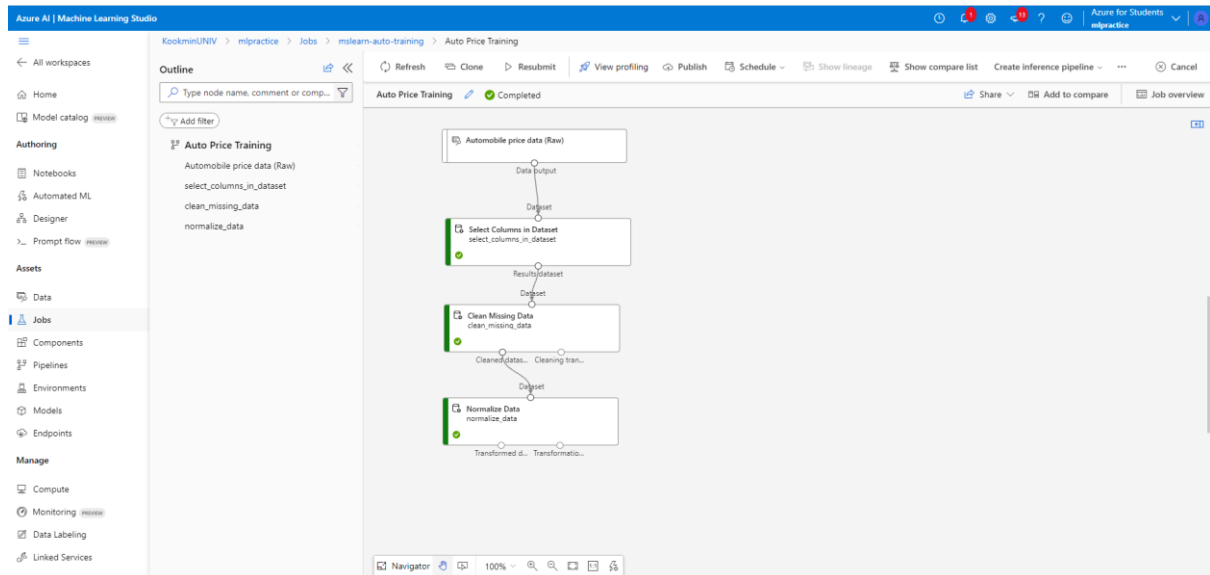
Mean	65.9078
Median	65.5
Min	60.3
Max	72.3
Standard deviation	2.1452
Unique values	44
Missing values	0
Feature type	Numeric Feature

engine-size의 컬럼과 width의 컬럼을 보면 min값과 max값이 다른 것을 확인할 수 있습니다. 머신러닝 학습시 알고리즘의 특정 컬럼에 영향을 과도하게 받지 않도록 하기 위해 0~1 사이의 값으로 값들을 Scaling(Min-Max Scaling)을 하여, Normalize를 합니다.

Parameters	
Transformation method	MinMax
Use 0 for constant columns when checked	True
Columns to transform	Column names: symboling, wheel-base, length, width, height, curb-weight, engine-size, bore, stroke, compression-ratio, horsepower, peak-rpm, city-mpg, highway-mpg

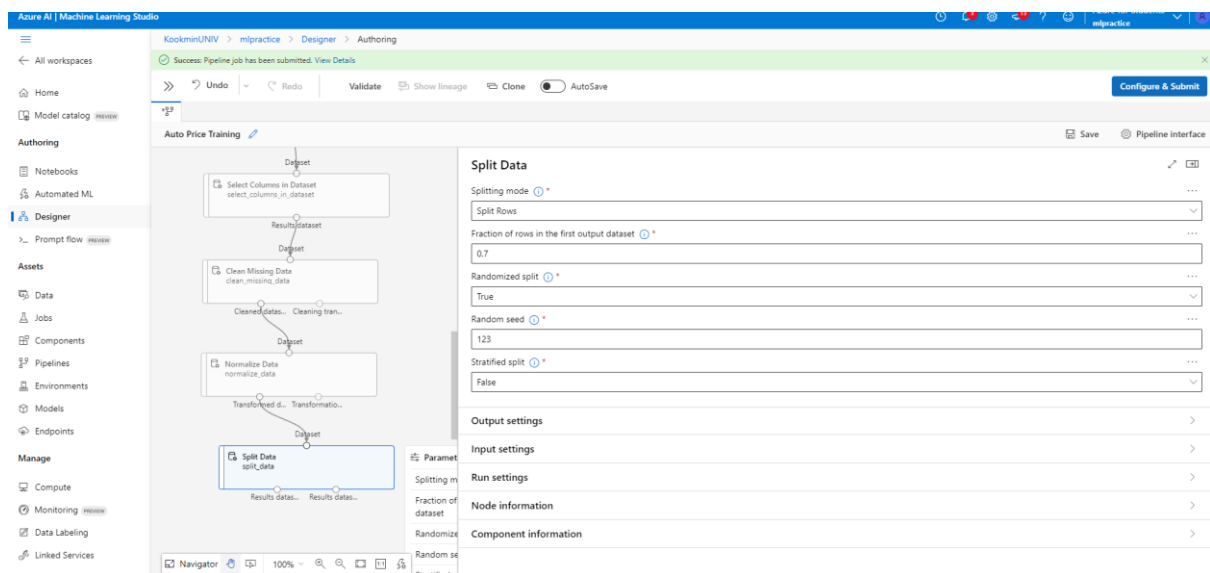
파이프라인 실행해보기

우측 상단의 Configure & Submit을 클릭 한뒤, Runtime settings을 마친 뒤, training pipeline을 제출하여 실행시킵니다.

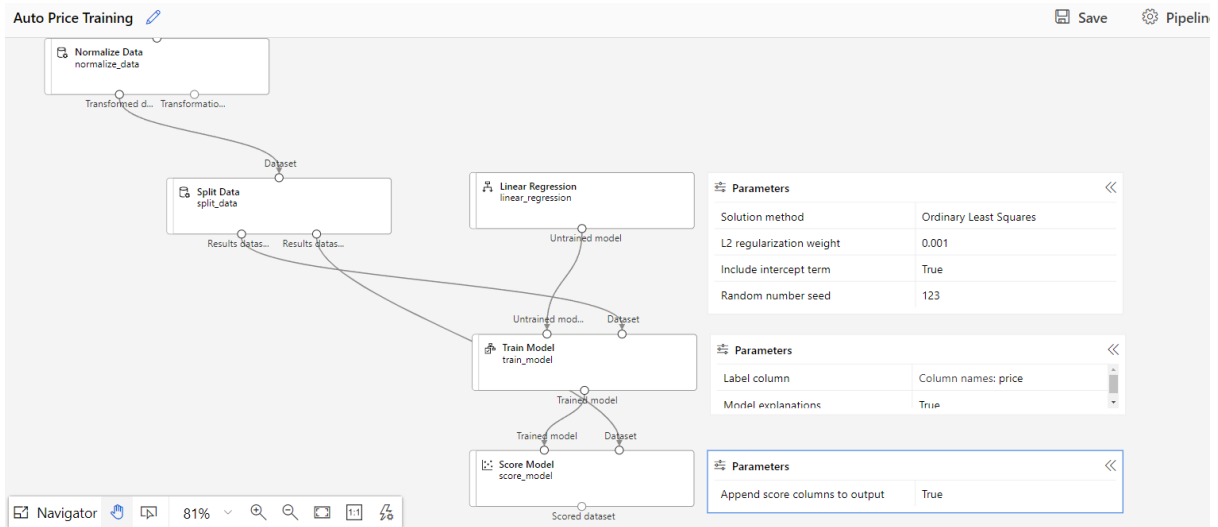


Train / Test 데이터로 나누기

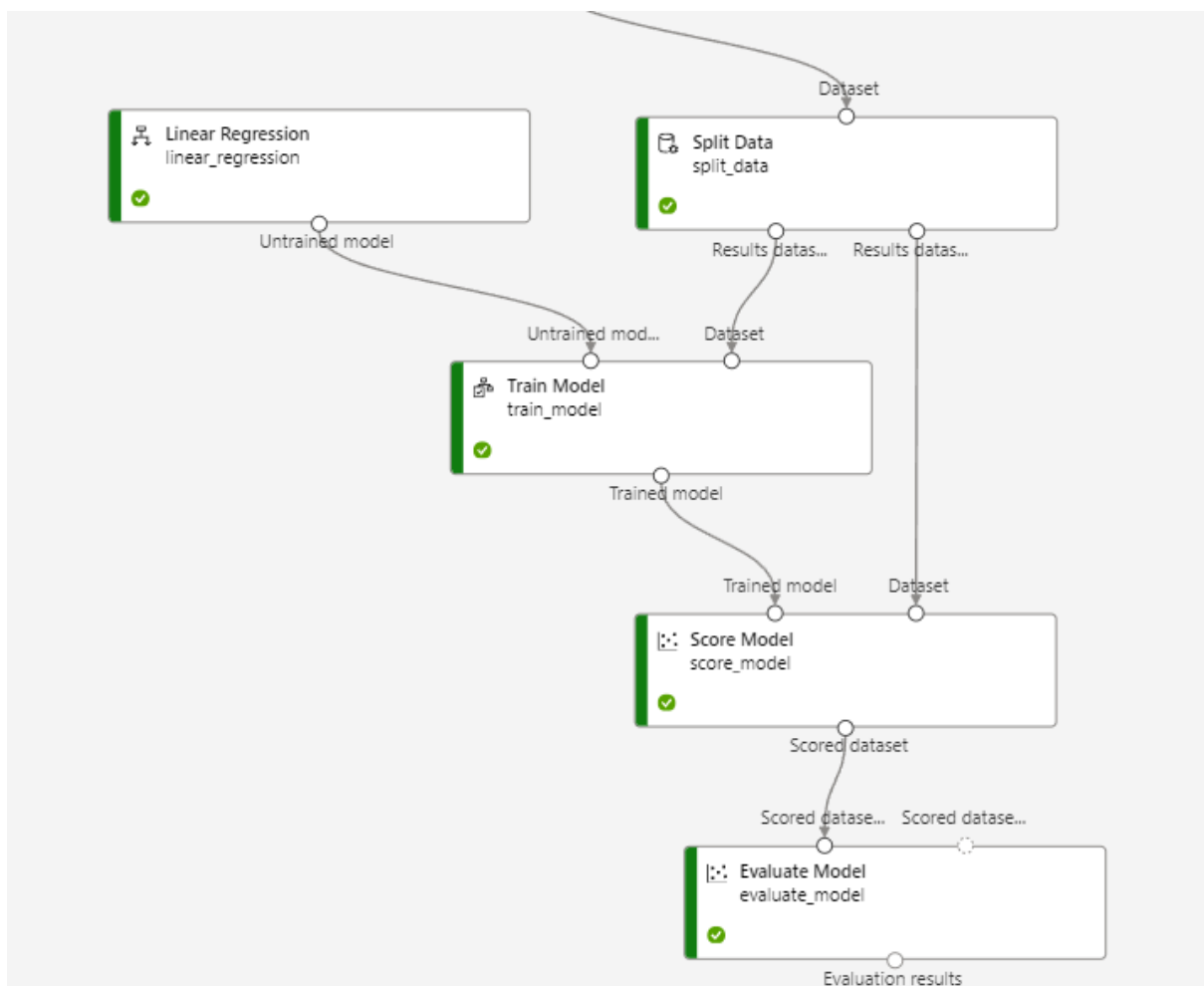
학습용 데이터와 테스트용 데이터를 7대 3으로 나눕니다.



Linear Regression 모델 Train / Test 데이터와 연결한 뒤, 모델 평가하기



Linear Regression과 Train model, Score Model Component의 각각의 설정을 마친 후, Scored Dataset을 Evaluate Model 컴포넌트와 연결 한 뒤, 다시 한 번, Configure & Submit을 진행합니다.



Evaluate Model

Overview Parameters Outputs + logs **Metrics** Child jobs Images Code Explanations (preview) Fairness (preview) Monitoring

Refresh Create custom chart View as... Current view: Local Edit view

Select metrics

Select to view as visualization or table of the data

Search

☐ Select all

☒ Metrics (5)

☐ Coefficient_of_Determination

☐ Mean_Absolute_Error

☐ Relative_Absolute_Error

☐ Relative_Squared_Error

☒ Root_Mean_Squared_Error

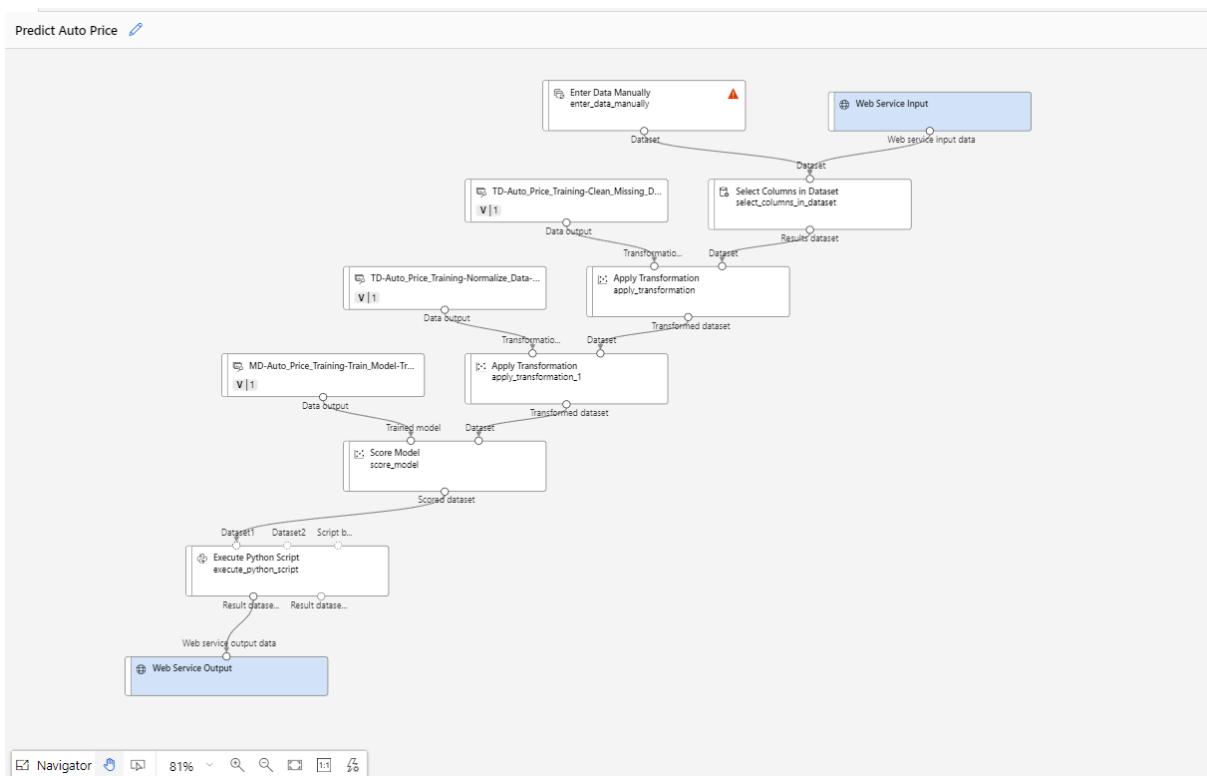
Root_Mean_Squared_E...

3401.861

모델 학습까지 완료하였고 평가지표로 RMSE가 3401.861이 나온 것을 확인 할 수 있습니다.

추론 파이프라인 생성 및 실행

캔버스 우측 상단 Create inference pipeline을 클릭한 뒤, Real-time inference pipeline를 클릭합니다.



다음과 같이 새 파이프라인을 만들어서, 웹 서비스의 입력과 결과를 반환하기 위한 웹 서비스 출력을 포함시킵니다.

수동으로 데이터 입력 모듈에 자동차에 대한 레이블이 없는 기능 값을 포함하는 CSV 데이터를 입력합니다.

Enter Data Manually

Data format

Has header

Data

```
1 -of-doors,body-style,drive-wheels,engine-location,wheel-base,length
2 168.8,64.1,48.8,2548,dohc,four,130,mpfi,3.47,2.68,9,111,5000,21,27
3 168.8,64.1,48.8,2548,dohc,four,130,mpfi,3.47,2.68,9,111,5000,21,27
4 1.2,65.5,52.4,2823,ohcv,six,152,mpfi,2.68,3.47,9,154,5000,19,26
```

price를 예측하는 데이터이므로, Select Columns in Dataset 모듈에서 price field를 제거하여 줍니다.

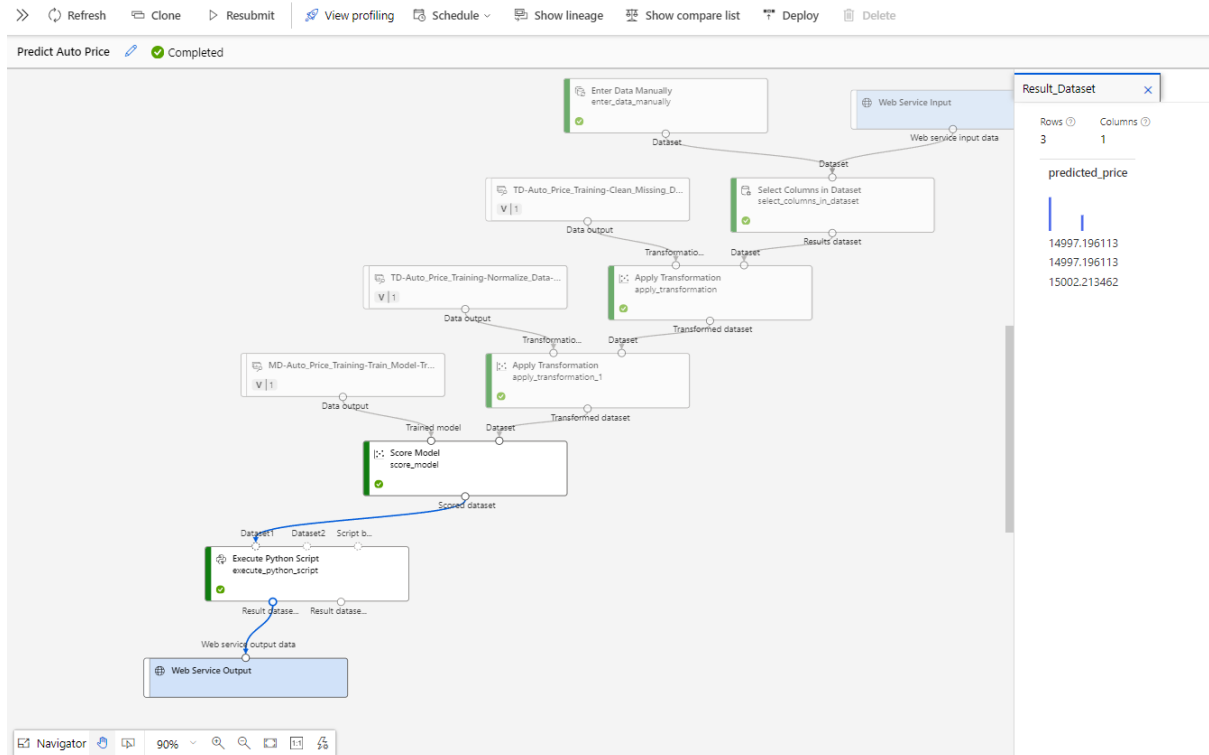
Execute Python Script

Python script

```
1 # The script MUST contain a function named acurem_main
2 # which is the entry point for this module.
3 import pandas as pd
4
5 def acurem_main(dataframe1 = None, dataframe2 = None):
6     scored_results = dataframe1[['Scored Labels']]
7     scored_results.rename(columns={'Scored Labels':'predicted_price'},
8                           inplace=True)
9     return scored_results
```

파이썬 스크립트 코드를 작성하여 예측 가격 시리즈를 반환하도록 설정 한 뒤, 컴퓨터 클러스터에 mslearn-auto-inference라는 새 실험으로 파이프라인에 제출합니다.

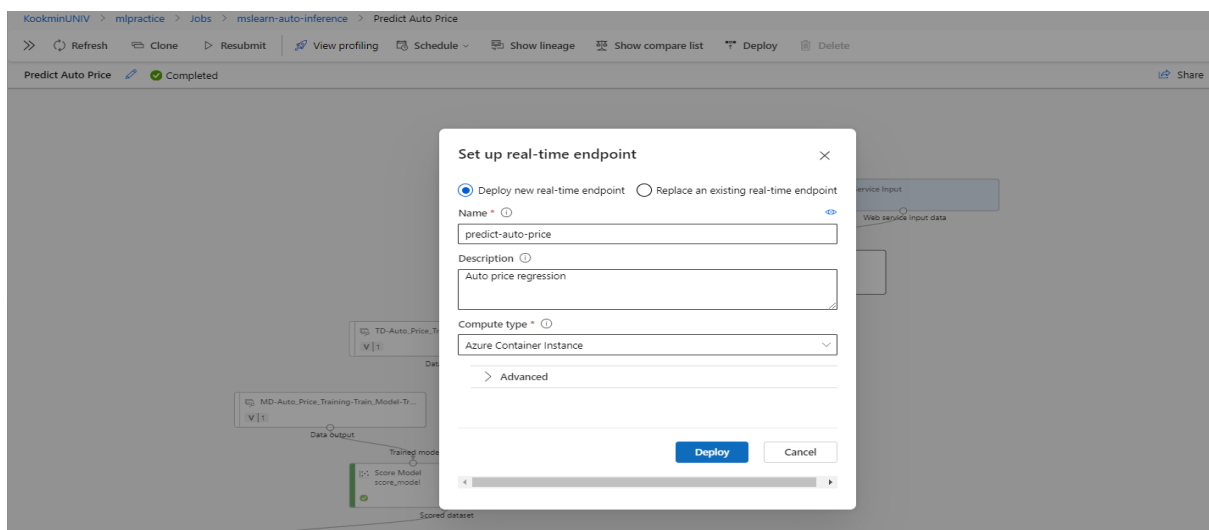
추론 파이프라인 생성 후, 서비스로 배포하기



Execute Python Script 모듈 미리보기 데이터로 자동차 세 대에 대한 예측 가격을 확인합니다.

	실제값	예측값
자동차1	13495	14997.196113
자동차2	16500	14997.196113
자동차3	16500	15002.213462

배포를 진행합니다.



서비스 테스트 해보기

엔드포인트 페이지에서 Test에 새로운 데이터로 모델을 테스트 합니다.

The screenshot displays the 'predict-auto-price' API interface. It features a 'Test' tab and a 'Test' button. The 'Input data to test endpoint' section shows a JSON object with various car attributes. The 'Test result' section shows the output, which includes a 'predicted_price' of 14997.19611345198.

```
Inputs: {
  "WebServiceInput0": {
    "symboling": 3,
    "normalized-losses": 1.0,
    "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,
    ...
  }
}
```

```
Results: { 1 item
  WebServiceOutput0: { 1 item
    0: { 1 item
      predicted_price: float 14997.19611345198
    }
  }
}
```

화면 오른쪽에 데이터에 지정된 특정 입력 기능을 갖춘 차량의 예측가격을 볼 수 있습니다.