Azure ML Studio

What is Machine Learning?

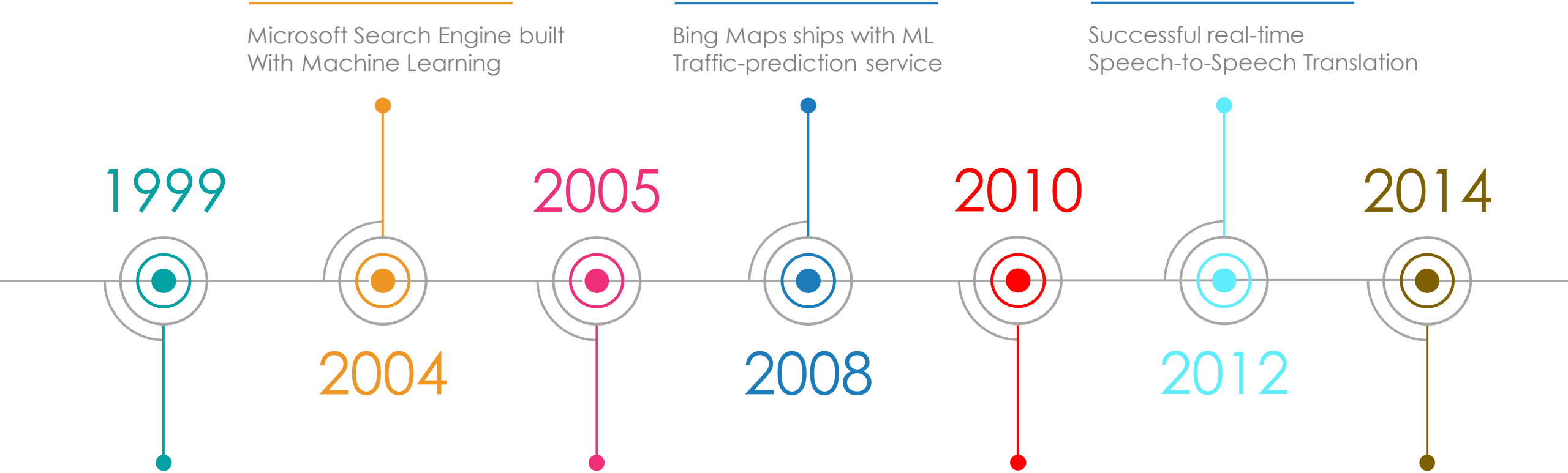
- Enables computers to adapt and learn from their experience.
- *Originally,* field of scientific study that concentrates on the construction and study of algorithms that can learn from data.
- Two main categories:
  - Supervised
  - Un-Supervised

Feed the Model Data and It "Learns"

Variables			What You Want to be Able to Predict
A	B	C	
1.2	90	28	N
1.4	86	32	N
1.1	134	6	Y
1.2	110	14	Y
1.2	60	26	N
1.3	89	35	Should get N
1.1	130	6	Should get Y

MICROSOFT MACHINE LEARNING HISTORY

proschool
An ims Initiative



https://blogs.technet.microsoft.com/machinelearning/2014/07/08/twenty-years-of-machine-learning-at-microsoft/

Machine Learning on Azure

Accelerate the end-to-end machine learning lifecycle

Empower developers and data scientists with a wide range of productive experiences for building, training and deploying machine learning models faster.

Boost productivity and access ML for all skills

Productivity for all skill levels, with code-first and drag-and-drop designer and automated machine learning.

Operationalize at scale with robust MLOps

Robust MLOps capabilities that integrate with existing DevOps processes and help manage the complete ML lifecycle.

Machine Learning on Azure..

Build responsible AI solutions

State-of-the-art model interpretability to build responsible AI solutions, with enhanced security and cost management.

Innovate on an open and flexible platform

Best-in-class support for open-source frameworks and languages including MLflow, Kubeflow, ONNX, PyTorch, TensorFlow, Python and R.

Microsoft Azure Cloud

- Benefits:
  - Access the cloud-based apps from anywhere
  - Cloud based disaster recovery for minimizing downtime
  - Scale up or down the applications as needed
  - Pay only for the resources you use
  - Back up your files and data to the cloud

Azure ML – Pros & Cons

proschool
An ims Initiative

Advantage

- Low or no coding involved.
- People without data science background can also build data models through drag-and-drop gestures and simple data flow diagrams.
- Easy to understand how a classification pipeline (data transformation, pre-processing, feature extraction etc.) works as it is visual.
- As It operates on the Azure public cloud, users no longer need to worry about maintenance and deployment.
- Once practiced and tested, model can be created and publish as the web service via clicking a single button. Use the web service from any device by passing valid credentials.

Disadvantage

- Price
- Limited storage with free account
- High-level tool, therefore less control.

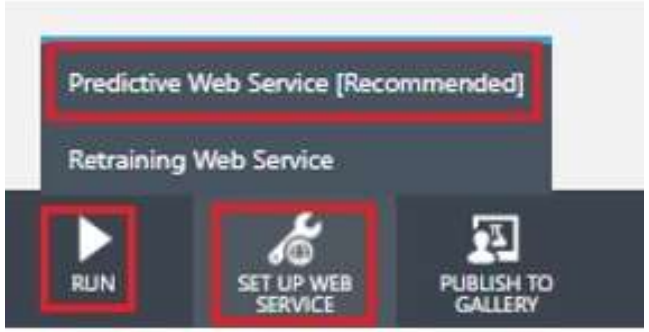
Example – Code Based vs. Azure ML

- For example, model deployment in Python requires you to build templates, RESTful web services & API which requires lots of coding, however the same can be done in ML Studio with only one button.

Python (A small portion of model deployment code)

```
from flask import Flask, request, jsonify
from ludwig import LudwigModel
import pandas as pd

app = Flask(__name__)

model = LudwigModel.load('model')

@app.route('/predict', methods=['POST'])
def predict():
    data = request.get_json()
    df = pd.DataFrame([str(data['text'])], columns=['content'])
    print(df.head())
    pred = model.predict(data_df=df)
    print(pred)
    return jsonify(pred['customer_sentiment_predictions'][0])

if __name__ == '__main__':
    app.run(port=3000, debug=True)
```

Azure ML Studio



Azure Machine Learning

ML is far too complex -

Mathematics, CS, Statistics, Domain Knowledge

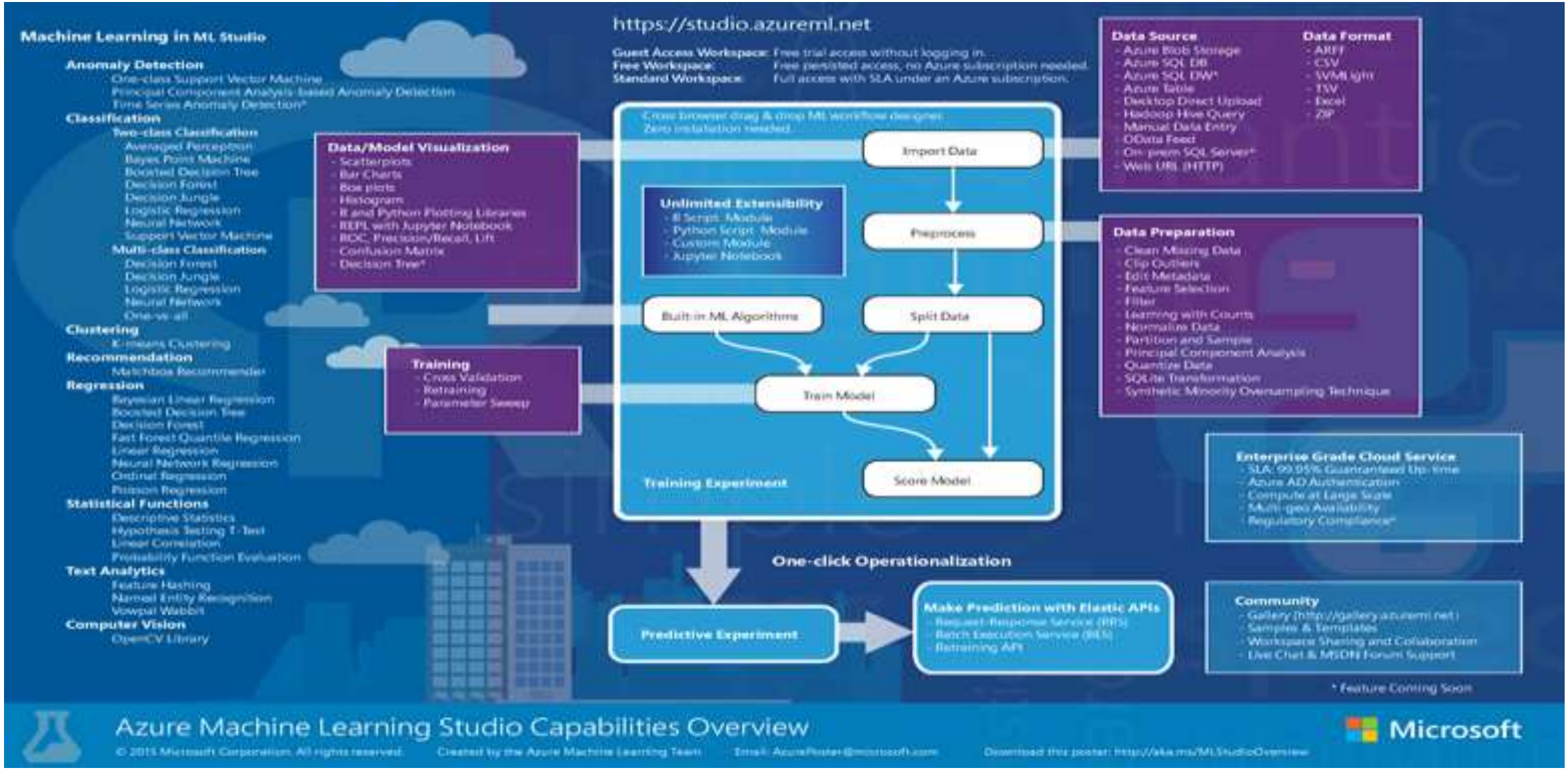
Fully managed cloud service for building predictive analytics solutions

No Hardware/Software to buy, only a web browser and an Azure account

Possibility of developing ML models without writing lines of code

ML Studio Overview

proschool
An ims Initiative



*** https://docs.microsoft.com/en-us/azure/machine-learning/studio/what-is-ml-studio**

Azure ML Studio - Prerequisites

- Azure ML Studio Account at https://studio.azureml.net
- An Hotmail or Outlook email address would be required to signup or login.

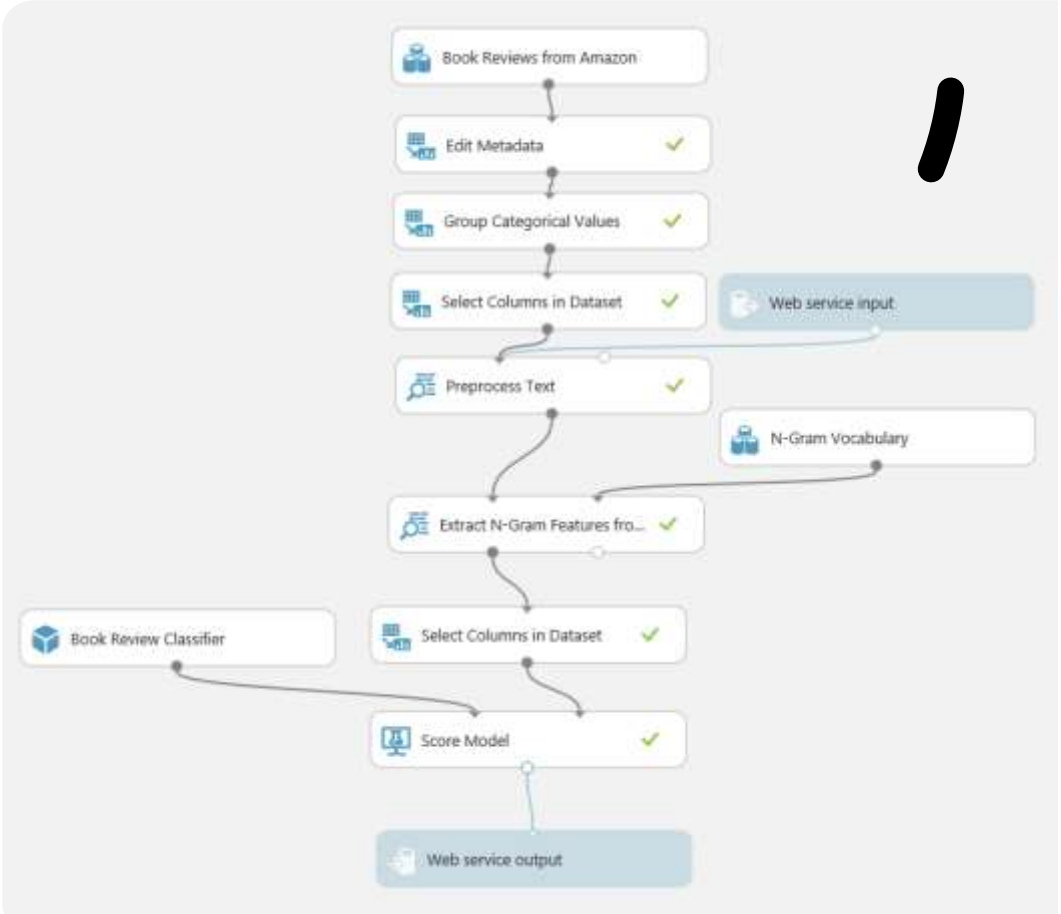
Azure ML Studio - Pricing

- Azure ML Studio is offered in two tiers—Free and Standard.
- Features by tier are compared in the table.

	FREE	STANDARD
Price	Free	$9.99/₹660.302 per ML studio workspace per month. $1/₹66.097 per studio experimentation hour.
Azure subscription	Not required	Required
Max number of modules per experiment	100	Unlimited
Max experiment duration	1 hour per experiment	Up to 7 days per experiment with a maximum of 24 hours per module
Max storage space	10GB	Unlimited - BYO
Execution/performance	Single node	Multiple nodes
Production Web API	No	Yes
SLA	No	Yes

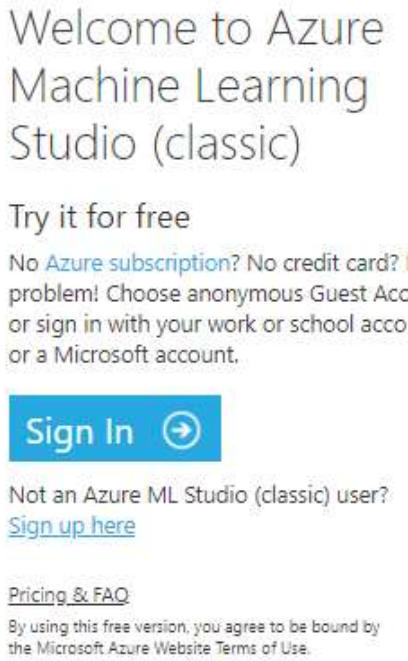
Azure ML Studio

- A collaborative, drag-and-drop visual workspace where you can –
  - Build,
  - Test, and
  - Deploy ML solutions

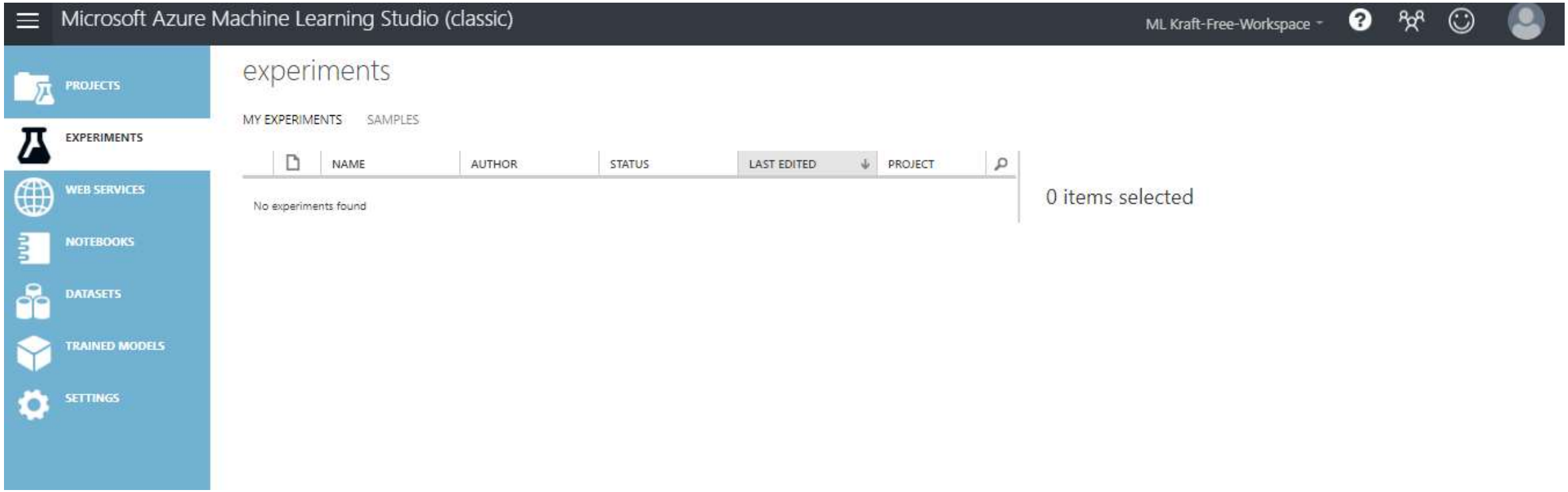


Azure ML Studio

- login to Azure ML studio by using the https://studio.azureml.net
- If you've signed into Machine Learning Studio before, click **Sign In**.
- If you don't have a Microsoft Azure account, create a free account without providing a credit card.
- This is the welcome screen of Azure ML studio and you can sign into it by clicking 'Sign In' button and providing with the necessary credentials required by Azure ML studio.

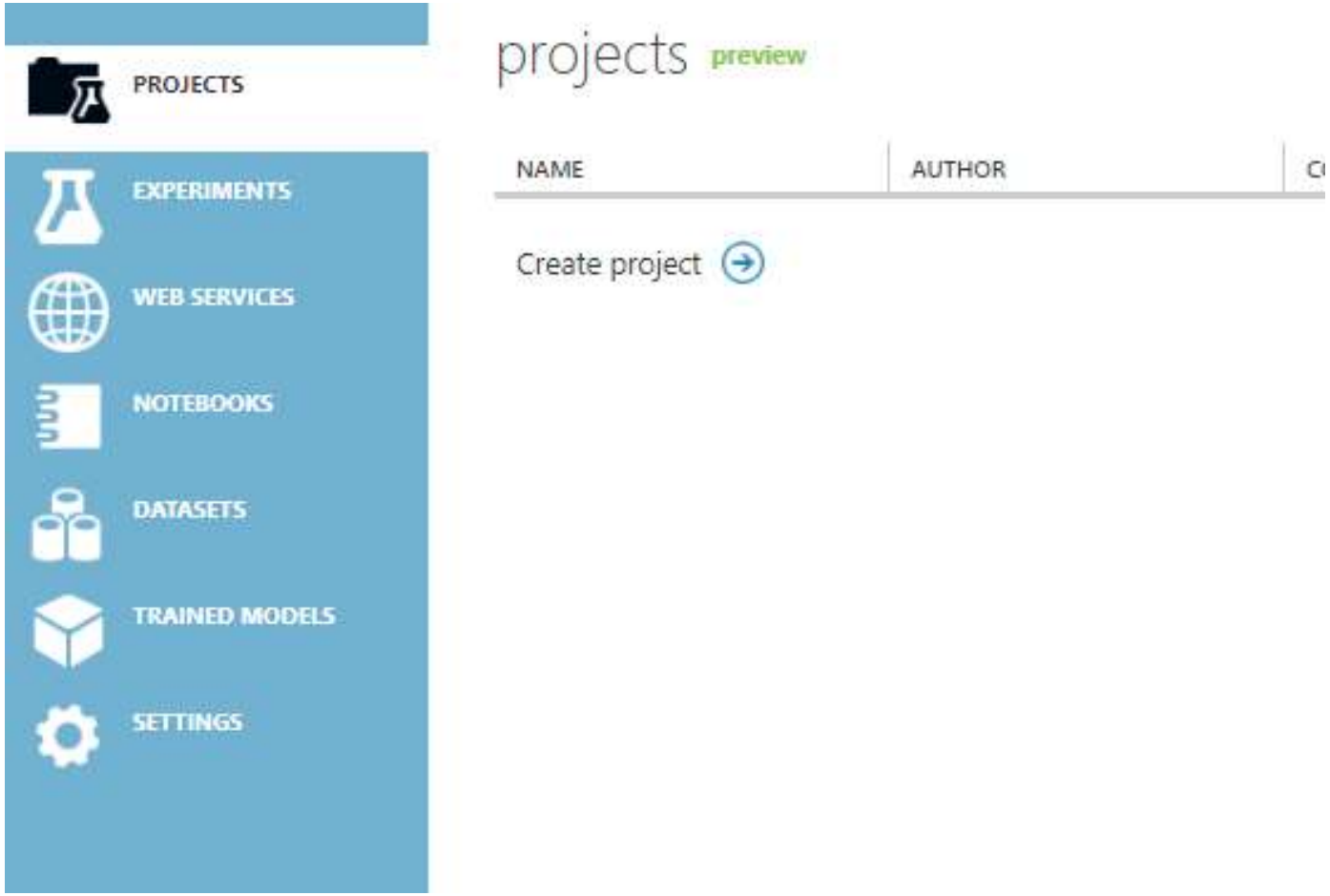


Azure ML Studio

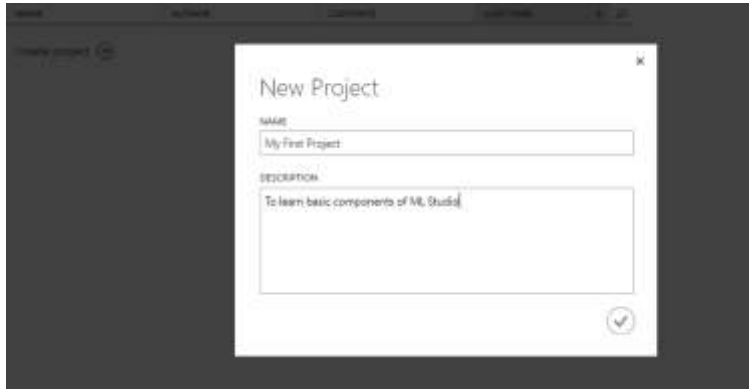


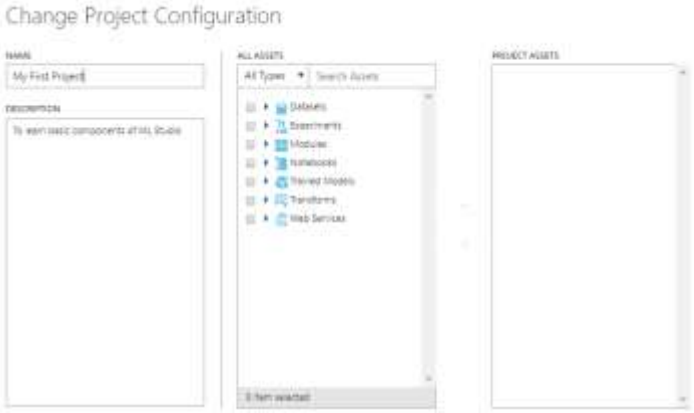
Azure ML Studio Components - Projects

Projects are collections of experiments, datasets, notebooks, and other resources representing a single project.



Azure ML Studio Components – Projects..



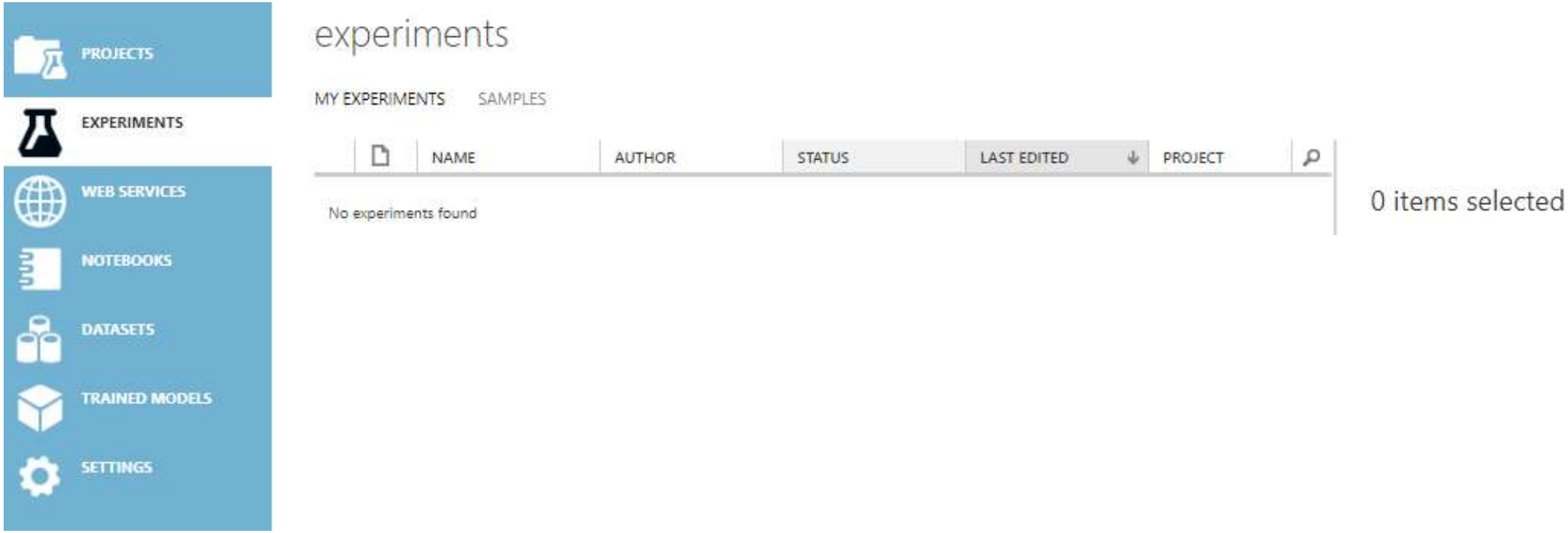


- When a New Project is created, you can give it a name and description, and add an assets which can be Notebooks, Datasets or Experiments.

Azure ML Studio Components - Experiments

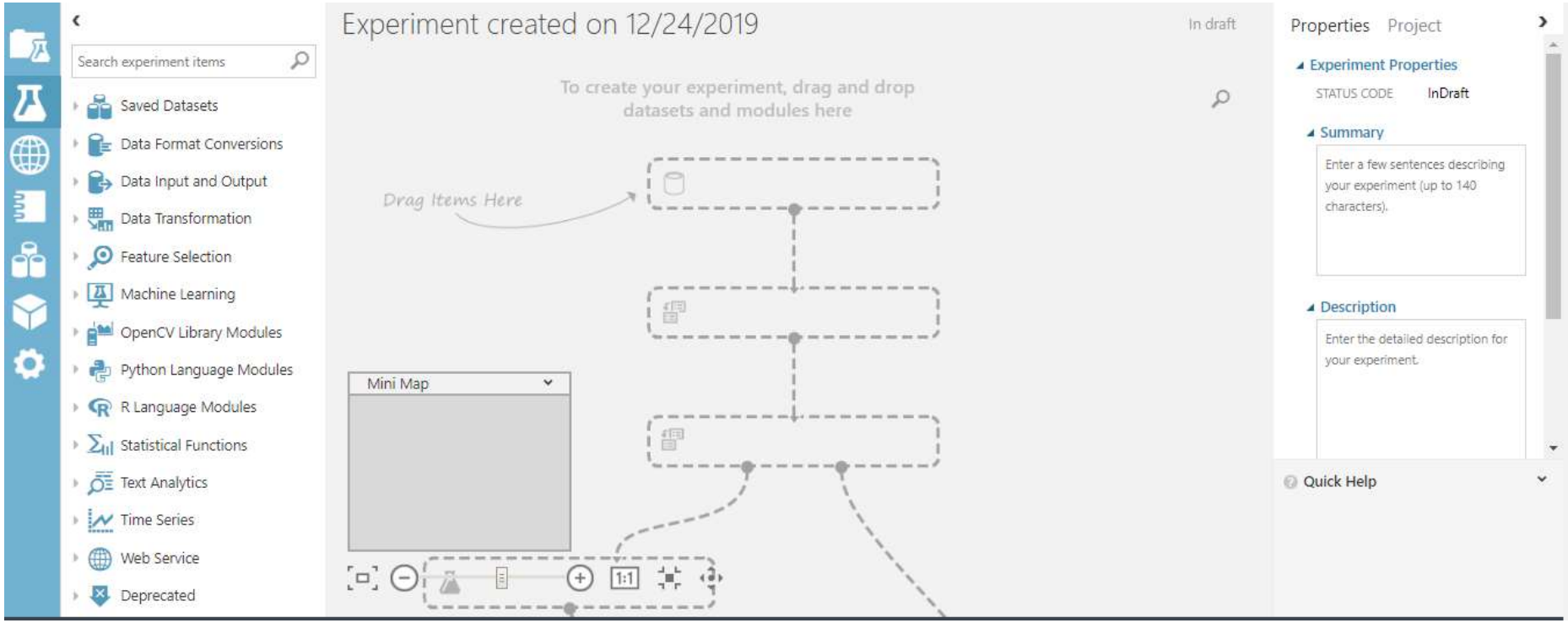
proschool An ims Initiative

Experiments are what you create with the drag-and-drop tool, which is experiments that you have created and run or saved as drafts.



Azure ML Studio Components – Experiments..

- A blank experiment would look like this in which you can drag and drop datasets and modules from the left sidebar.

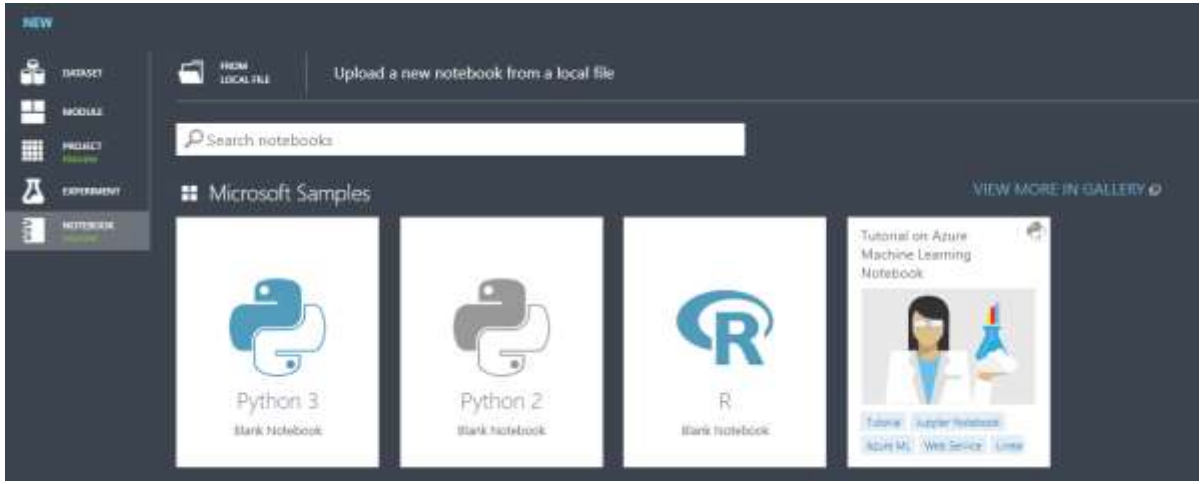


Azure ML Studio Components – Web Services

Web services are that you have deployed from your experiments.

Azure ML Studio Components – Notebooks

Notebooks are Jupyter notebooks where you collect code snippets, equations, links, and figures.



Azure ML Studio Components - Datasets

proschool An ims Initiative

Datasets are important in machine learning, since your predictions are only as good as the data you work with. Fortunately, ML Studio gives you access to lots of interesting datasets or you can also upload datasets in studio.

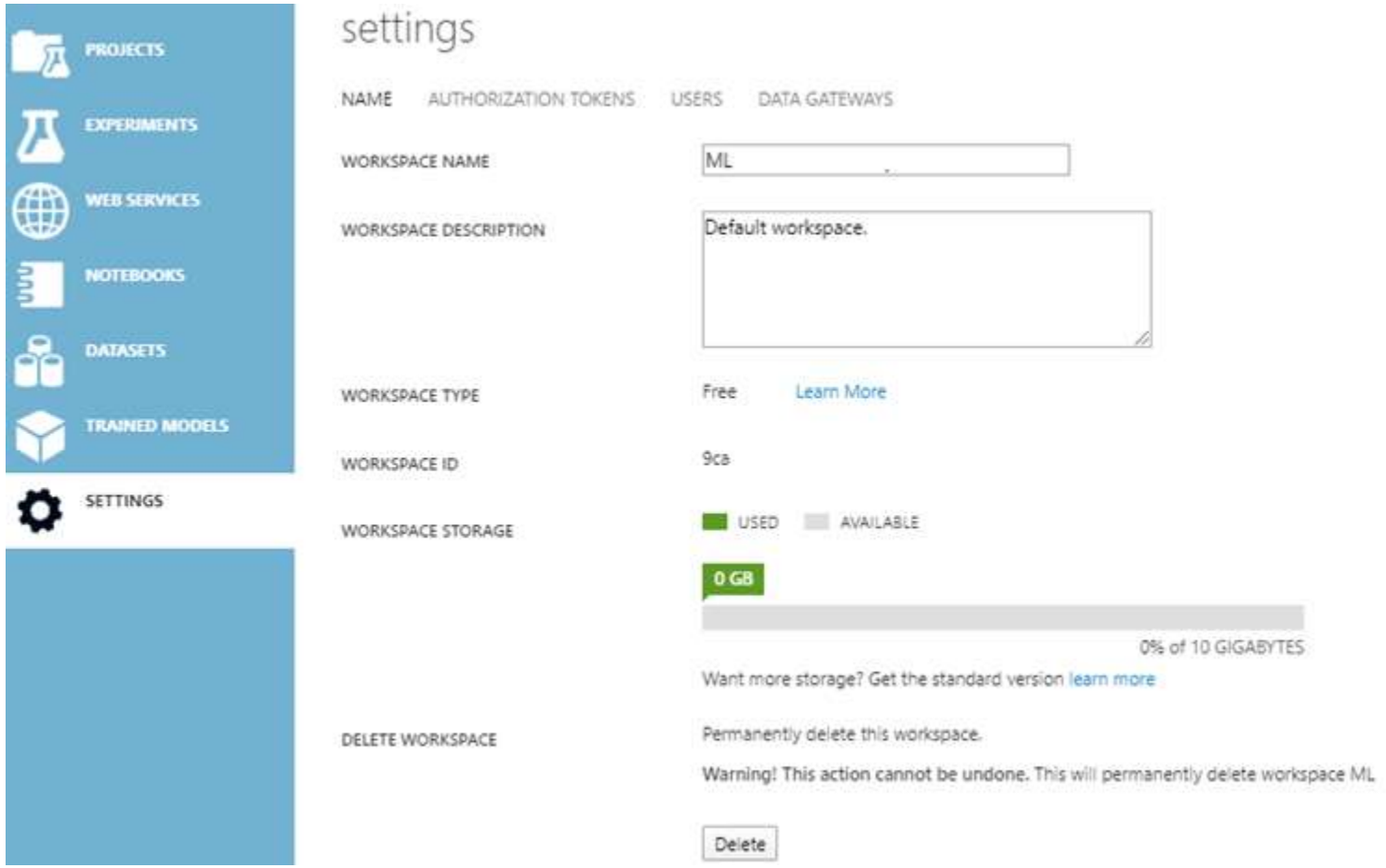
Azure ML Studio Components – Trained Models

Trained models are your machine learning output. In other words, models that you have trained in experiments and saved in Studio.

Azure ML Studio Components - Settings

proschool An ims Initiative

Settings is a collection of settings that you can use to configure your account.



Creating an Experiment

To **Create an Experiment in Machine Learning**, there are **five basic steps**.

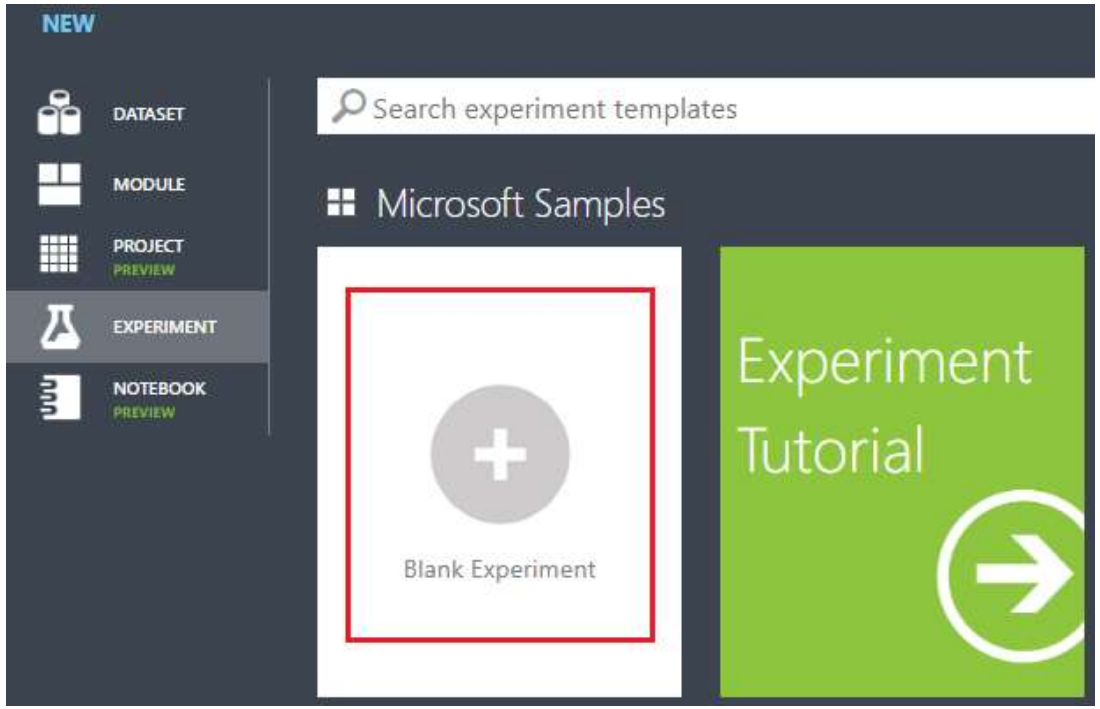
We will examine each of **these steps through developing our own prediction model**.

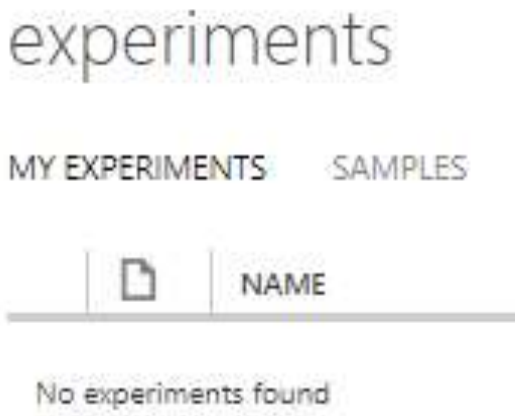
Sample Dataset – **Automobile Price Data (Raw)** which is available in Sample dataset

1. Obtaining the Data

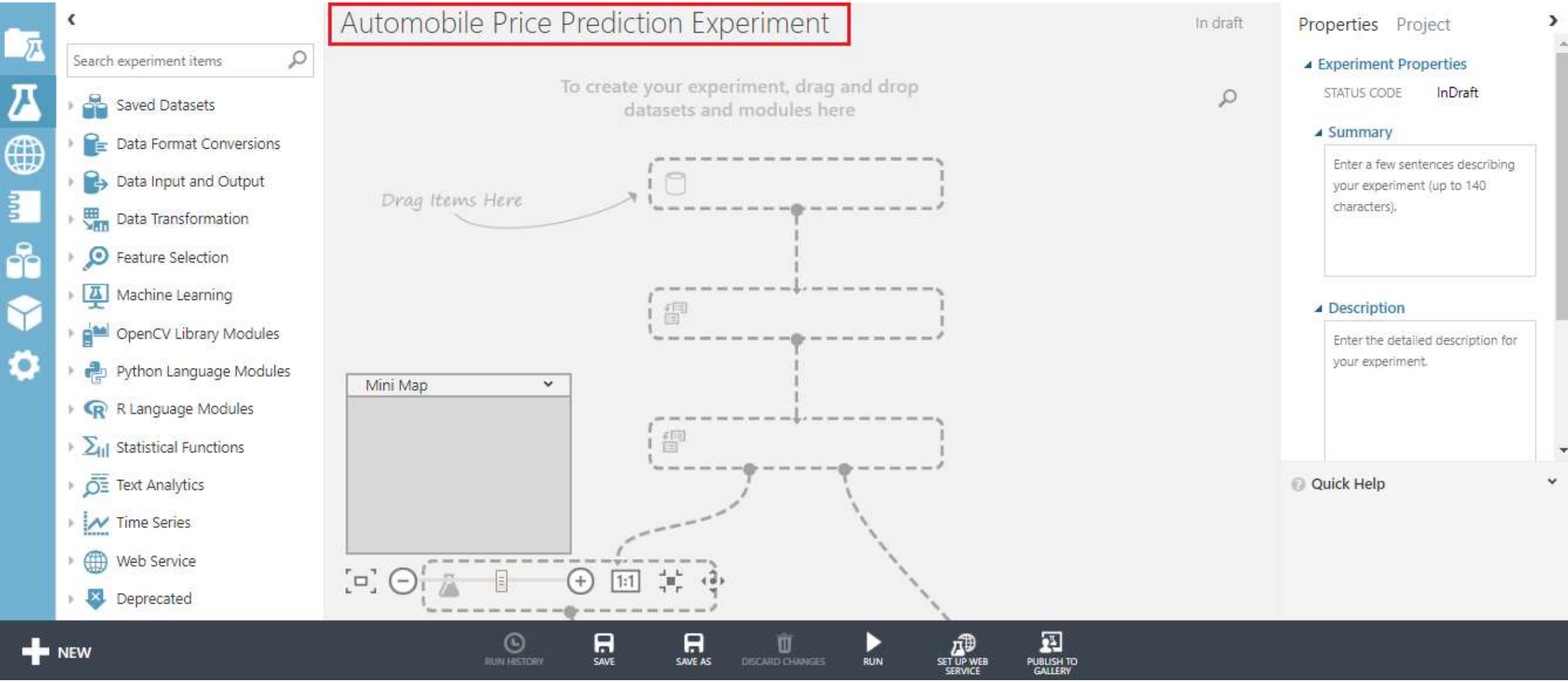
- Collecting data is one of the most important step in this process.
- Relevance of the data is the basis for creating good prediction models.
- ML Studio provides several datasets.
- For the experiment, we will use Automobile price data (Raw)
- Dataset includes information about automobiles by make and model, including the price, features such as the number of cylinders and MPG, as well as an insurance risk score.
- We will try to predicts price of automobile based on different variables

In the EXPERIMENTS pane, click on the NEW option.
In the New option select the Blank Experiment.

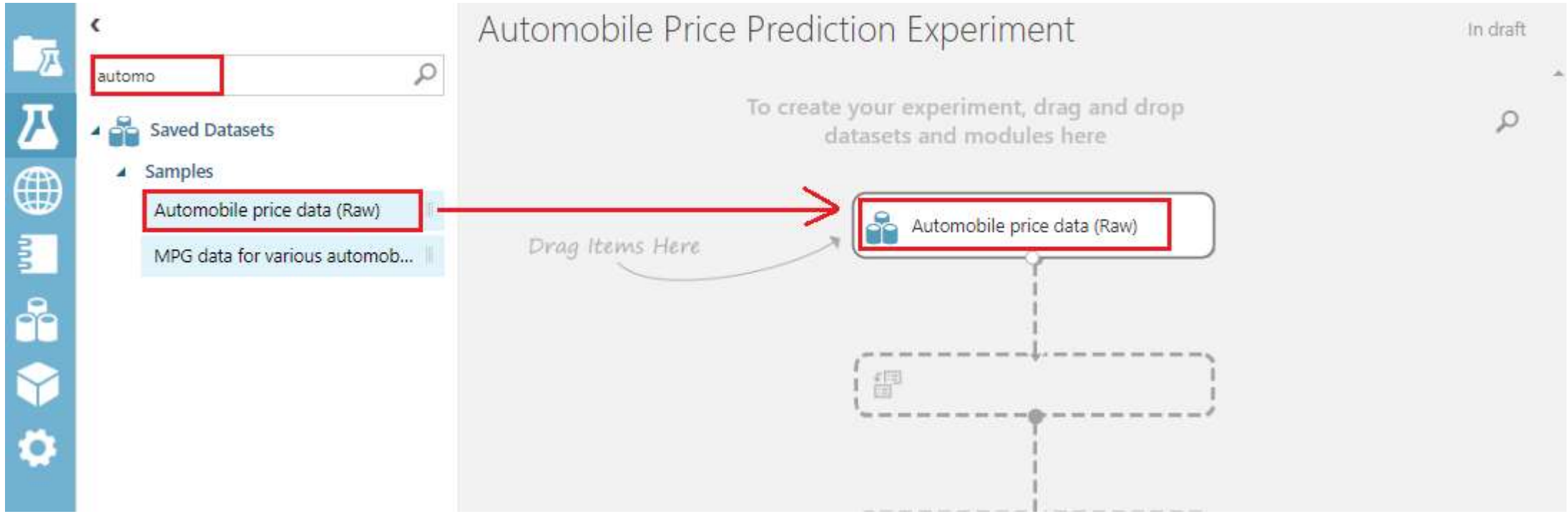




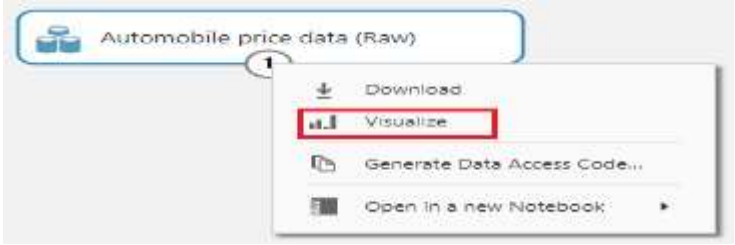
Change the Experiment name something meaningful.



To the left of the canvas, type in search box Automobile and fine the dataset Automobile price data. Then Drag the dataset to the canvas as shown in the picture.



Once uploaded, we can preview the data. Click on the **Visualize**.



Automobile Price Prediction Experiment › Automobile price data (Raw) › dataset

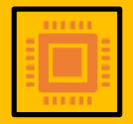
rows	columns
205	26

view as

symboling	normalized-losses	make	fuel-type	aspiration	num-of-doors	body-style	drive-wheels	engine-location	wheel-base	length	width	height	curb-weight	e t
3		alfa-romero	gas	std	two	convertible	rwd	front	88.6	168.8	64.1	48.8	2548	c
3		alfa-romero	gas	std	two	convertible	rwd	front	88.6	168.8	64.1	48.8	2548	c
1		alfa-romero	gas	std	two	hatchback	rwd	front	94.5	171.2	65.5	52.4	2823	c
2	164	audi	gas	std	four	sedan	fwd	front	99.8	176.6	66.2	54.3	2337	c
2	164	audi	gas	std	four	sedan	4wd	front	99.4	176.6	66.4	54.3	2824	c
2		audi	gas	std	two	sedan	fwd	front	99.8	177.3	66.3	53.1	2507	c
1	158	audi	gas	std	four	sedan	fwd	front	105.8	192.7	71.4	55.7	2844	c
1		audi	gas	std	four	wagon	fwd	front	105.8	192.7	71.4	55.7	2954	c

Statistics and Visualizations

2. Preparing the Data



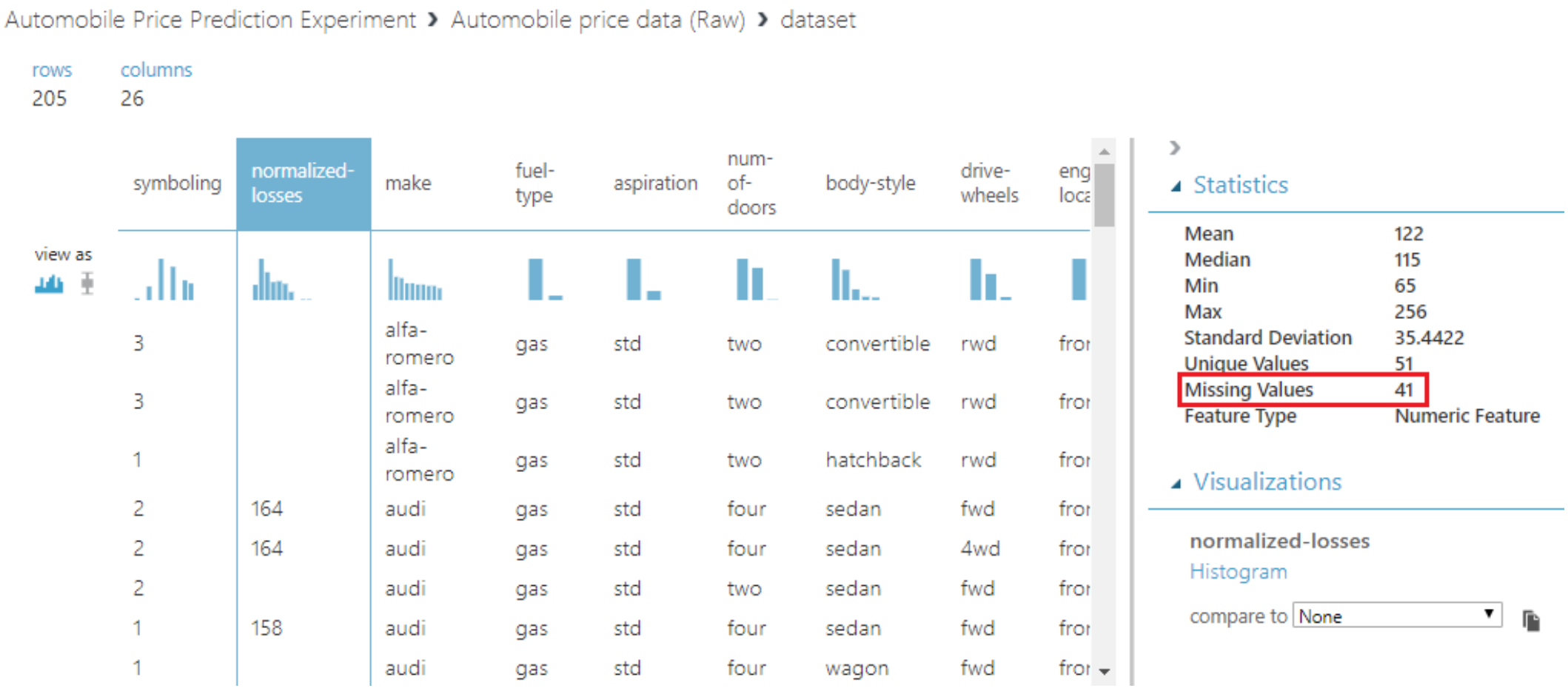
It involves adjusting the available data to your needs.

Datasets can contain lots of missing values which requires replacing with some other value or by removing them entirely.

Descriptive Statistics module can be used here to computes statistical data from the available data.

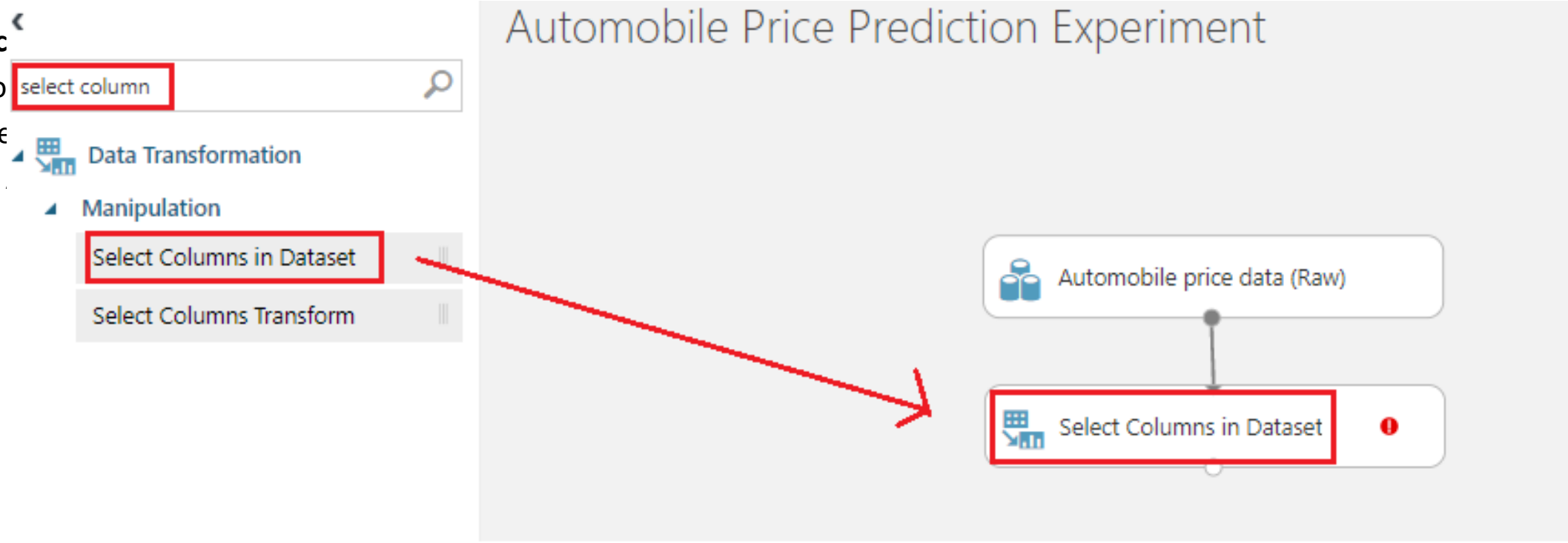
Other commonly used module is **Clean Missing Data**.

normalized-losses column has many missing values, so we'll removes the normalized-losses column completely. To do this, we will add a module to remove the column.

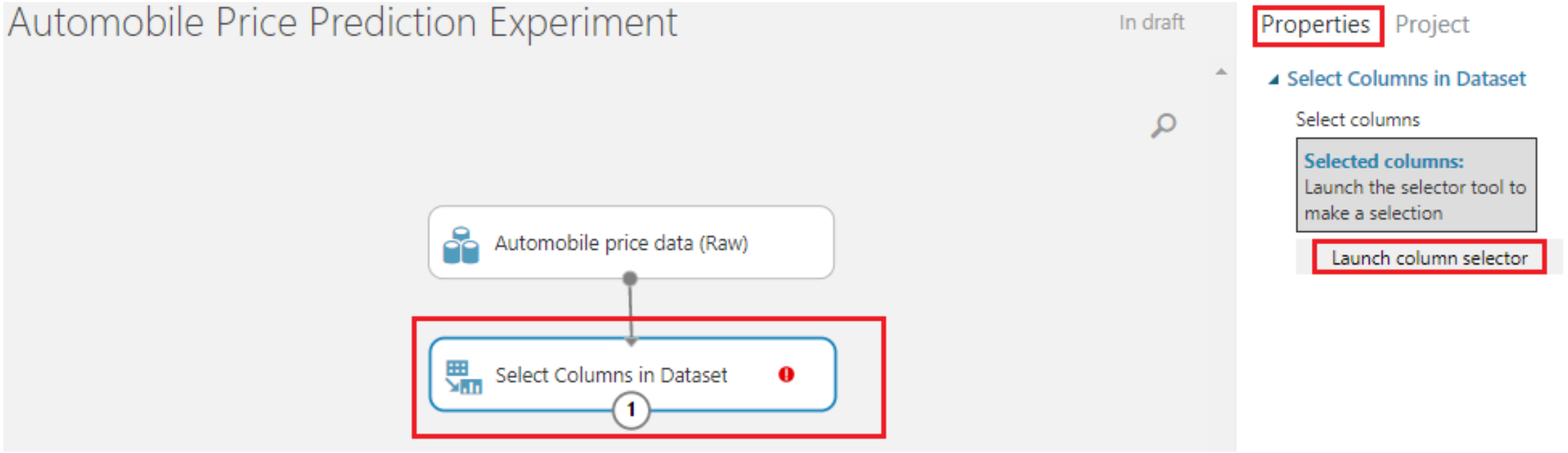


Removing the normalized-losses column

- Type **select c**
- It allow us to
- Drag the Sele
- Connect the

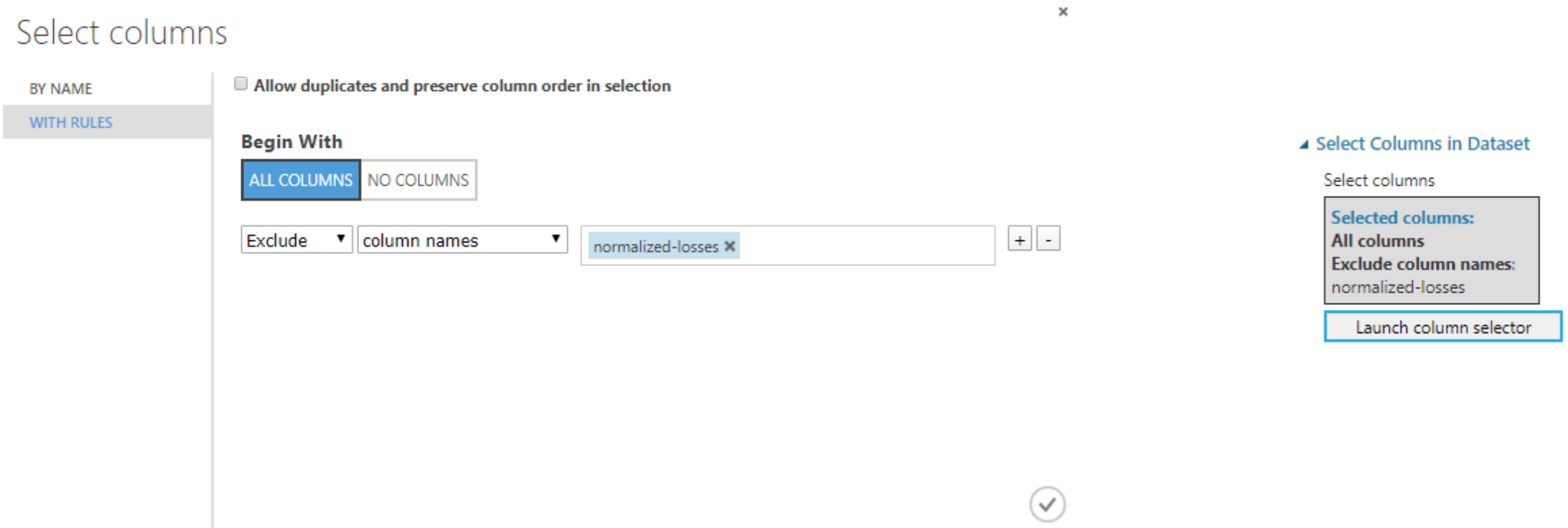


Removing the normalized-losses column ..

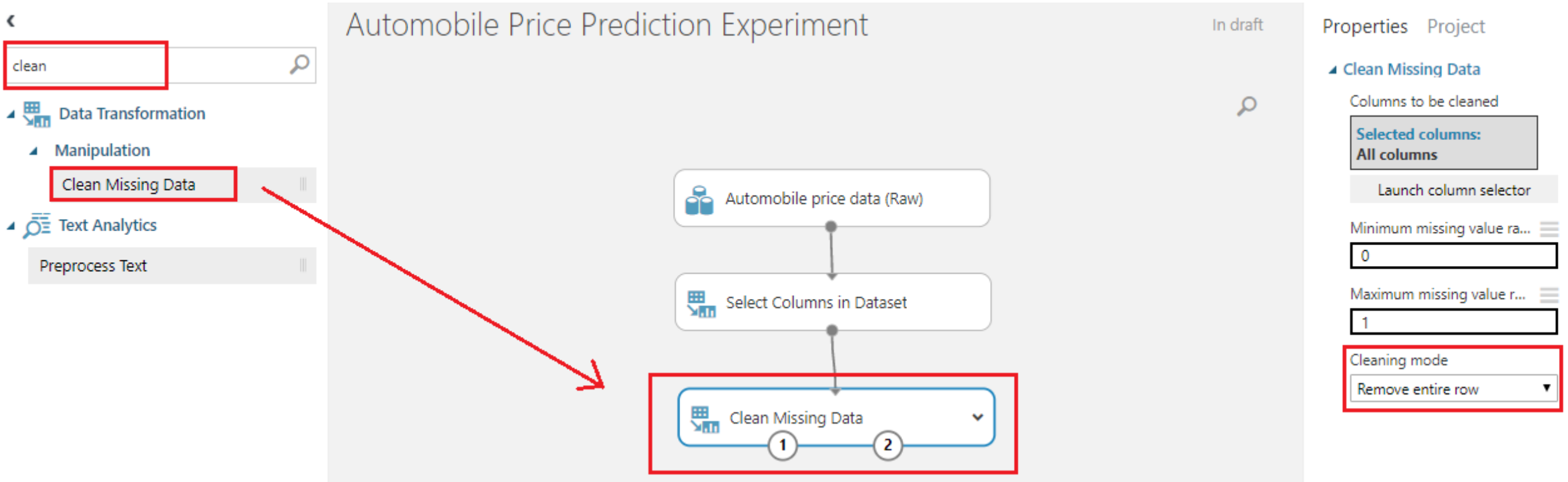


Removing the normalized-losses column ..

- To remove **normalized-losses** from dataset, click **With Rules** on the left side.
- Under **Begin With**, click **All columns**.
- From the drop-downs, select **Exclude** and **column names**, and then click inside the text box. Select **normalized-losses** and Click on the check mark button at the bottom.
- Now in the Properties pane, it shows that it will select all the column except normalized-losses

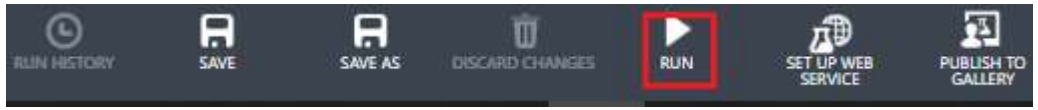


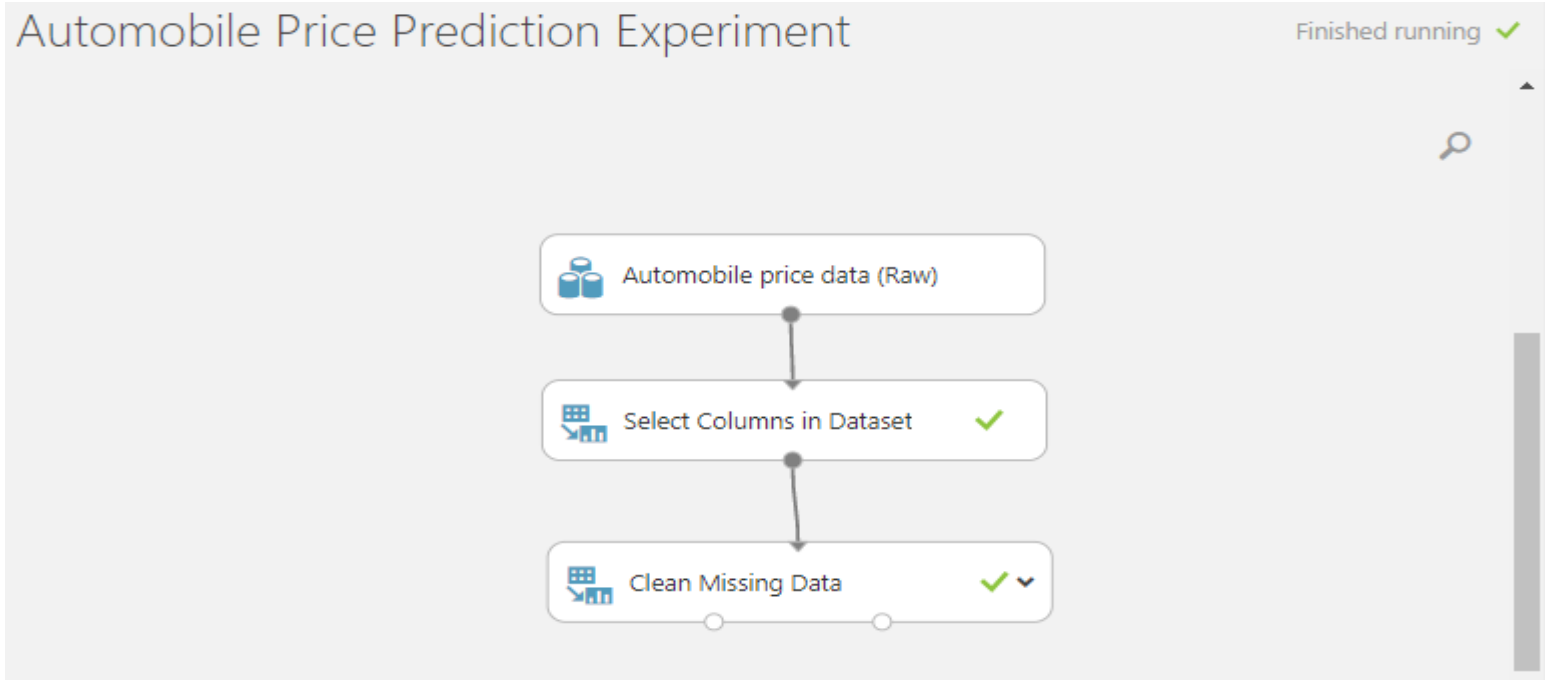
Removing rows with missing data



Run the Experiment

- At the bottom of the page click **Run** to the run the experiment.
- When the experiment has finished running, all the modules have a green check mark to indicate that they finished successfully.





3. Defining Features

In the dataset, each row represents one automobile, and each column is a feature of that automobile.

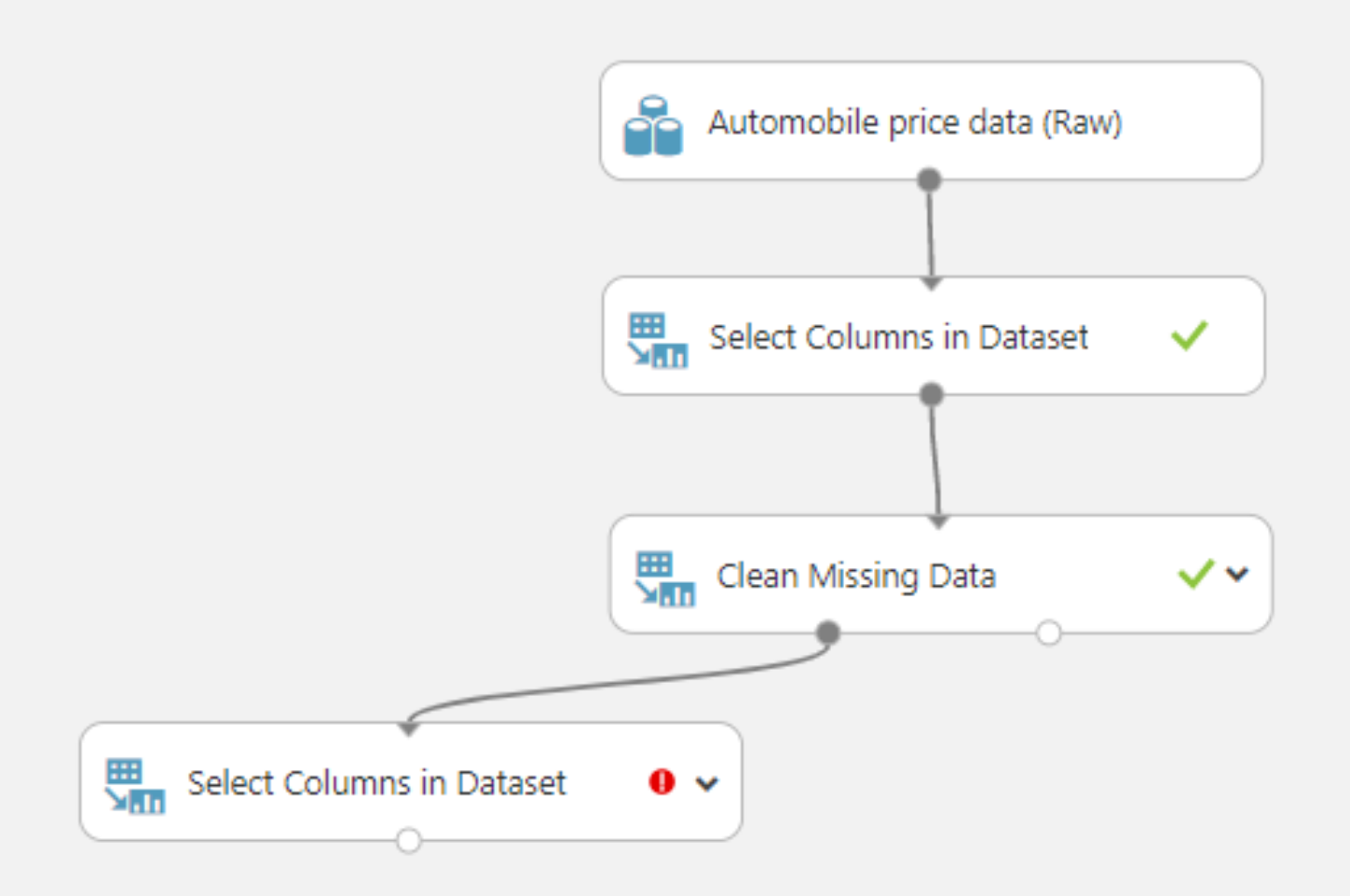
Finding a good set of features for predictive modelling requires experimentation and knowledge about the problem you want to solve.

Some features are better for predicting the target than others.

Also, some features have a strong correlation with other features and can be removed. For example, city-mpg and highway-mpg are closely related so we can keep one and remove the other without significantly affecting the prediction.

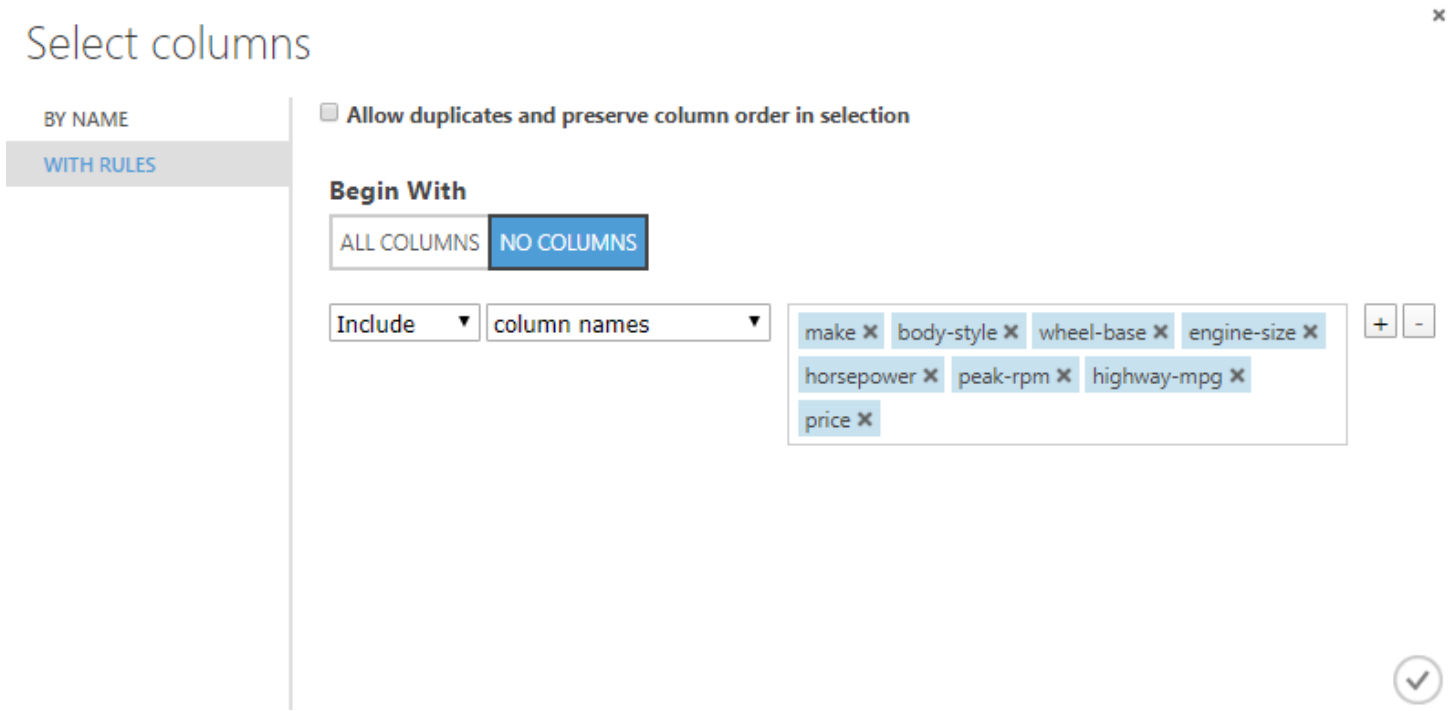
Defining Feature..

- Drag another **Select Columns in Dataset** module to the experiment canvas.
- Connect the left output port of the **Clean Missing Data** module to the input of the **Select Columns in Dataset** module.



Defining Feature..

- Click **Launch column selector** in the Properties pane and click **With rules**.
- Under **Begin With**, click **No columns**.
- In the filter row, select **Include** and **column names** and select our list of column names in the text box.
- We will be using following columns: **make, body-style, wheel-base, engine-size, horsepower, peak-rpm, highway-mpg, price**
- Click the check mark (OK) button.



4. Model Training

Now we will construct a predictive model to see how close it's able to predict prices.

We'll use a simple linear regression model a supervised learning algorithm..

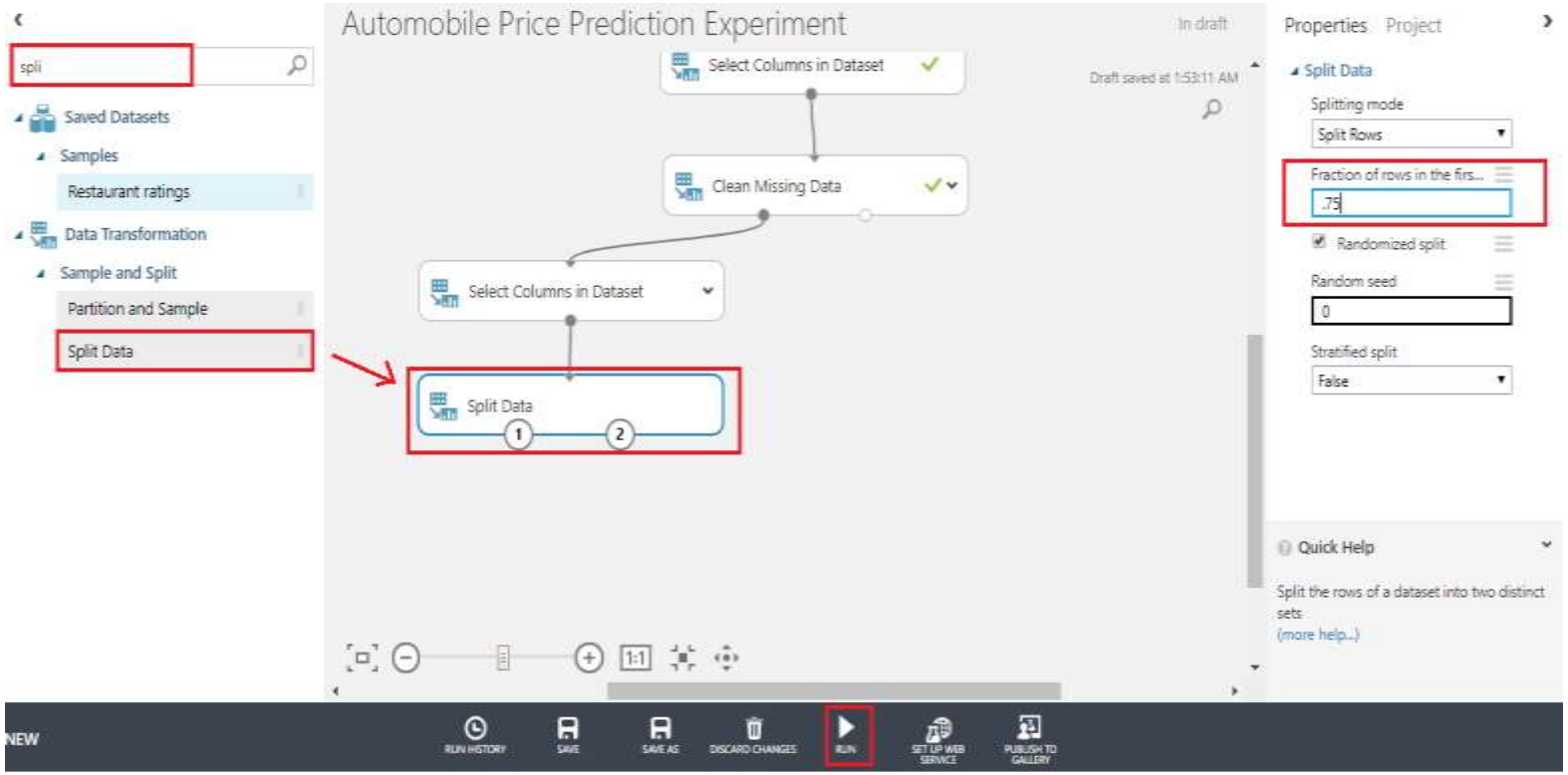
Regression is used to make a prediction from a continuous set of values.

We can use our data for both training and testing by splitting it into separate training and testing sets.

The models analyse the data and finds the relation between *prices* and *automobile features*. We, then, test our model with the training data. We give model set of features for automobiles and then see how closely our model was able to predict the known price.

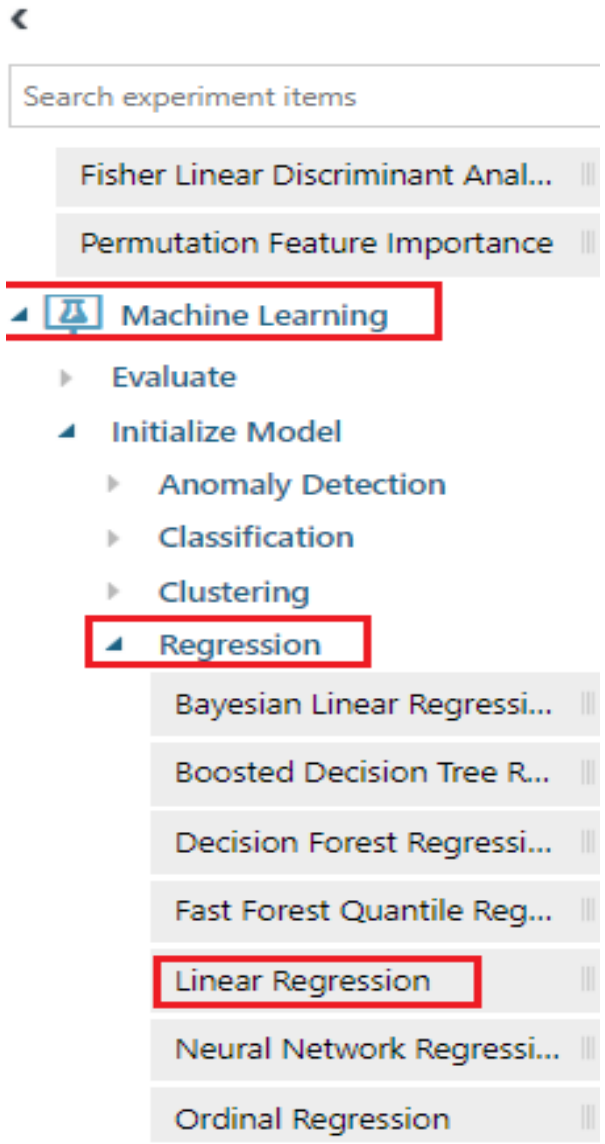
Training Data

- Select and drag the **Split Data** module to the experiment canvas and connect it to the last Select Columns in Dataset module.
- Click the Split Data module to select it, and in the Properties pane - the Fraction of rows in the first output dataset and set it to 0.75.
- We'll use 75 percent of the data to train the model, and 25 percent for testing.
- Run the experiment.



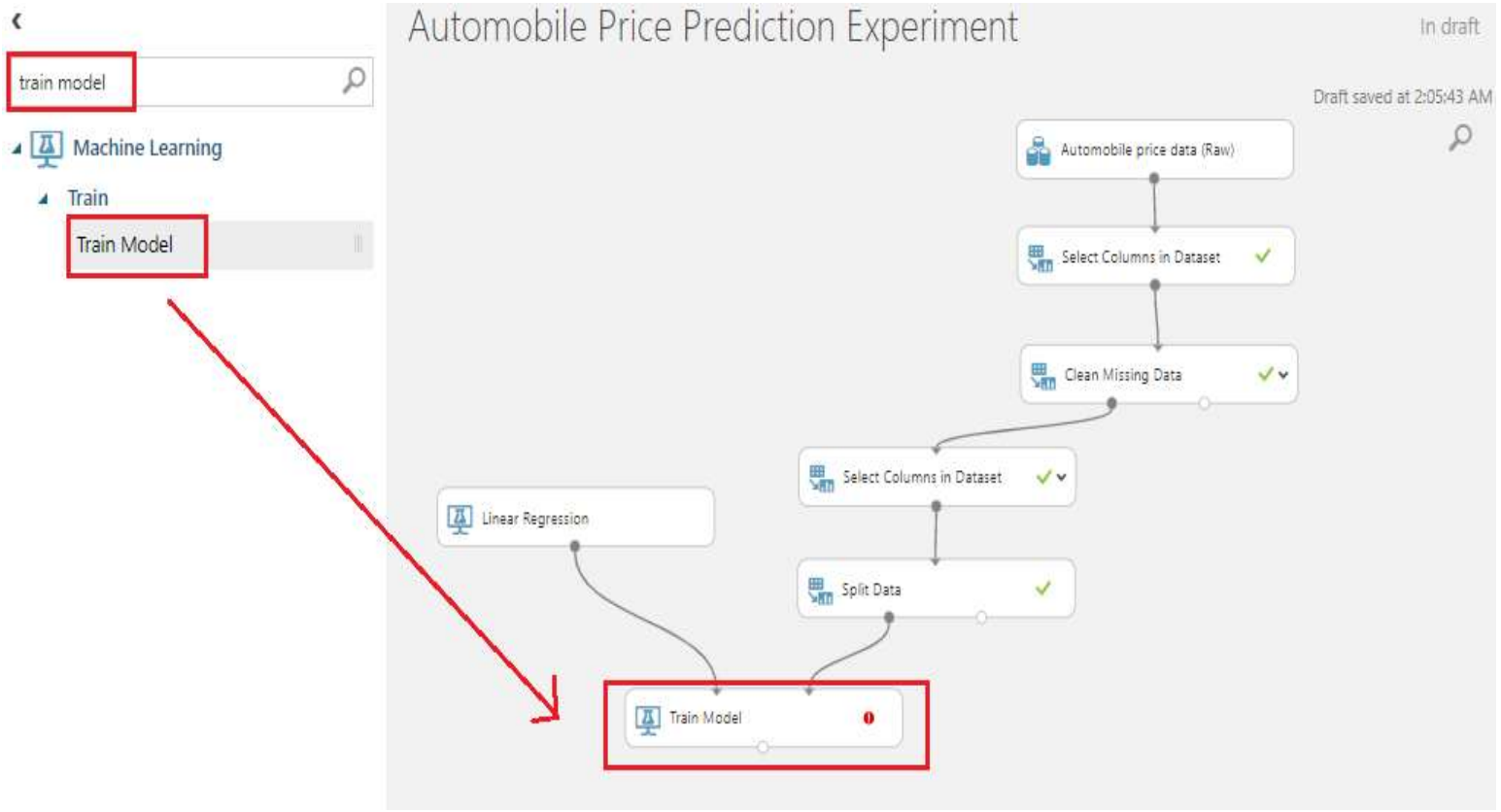
Selecting learning Algorithms

- Expand the **Machine Learning category** in the module palette to the left of the canvas, and then expand **Initialize Model**.
- Select the **Linear Regression** module under the **Regression** category and drag it to the experiment canvas.
- Other way to select Linear Regression module is by typing in the palette Search box.
- Run the experiment.



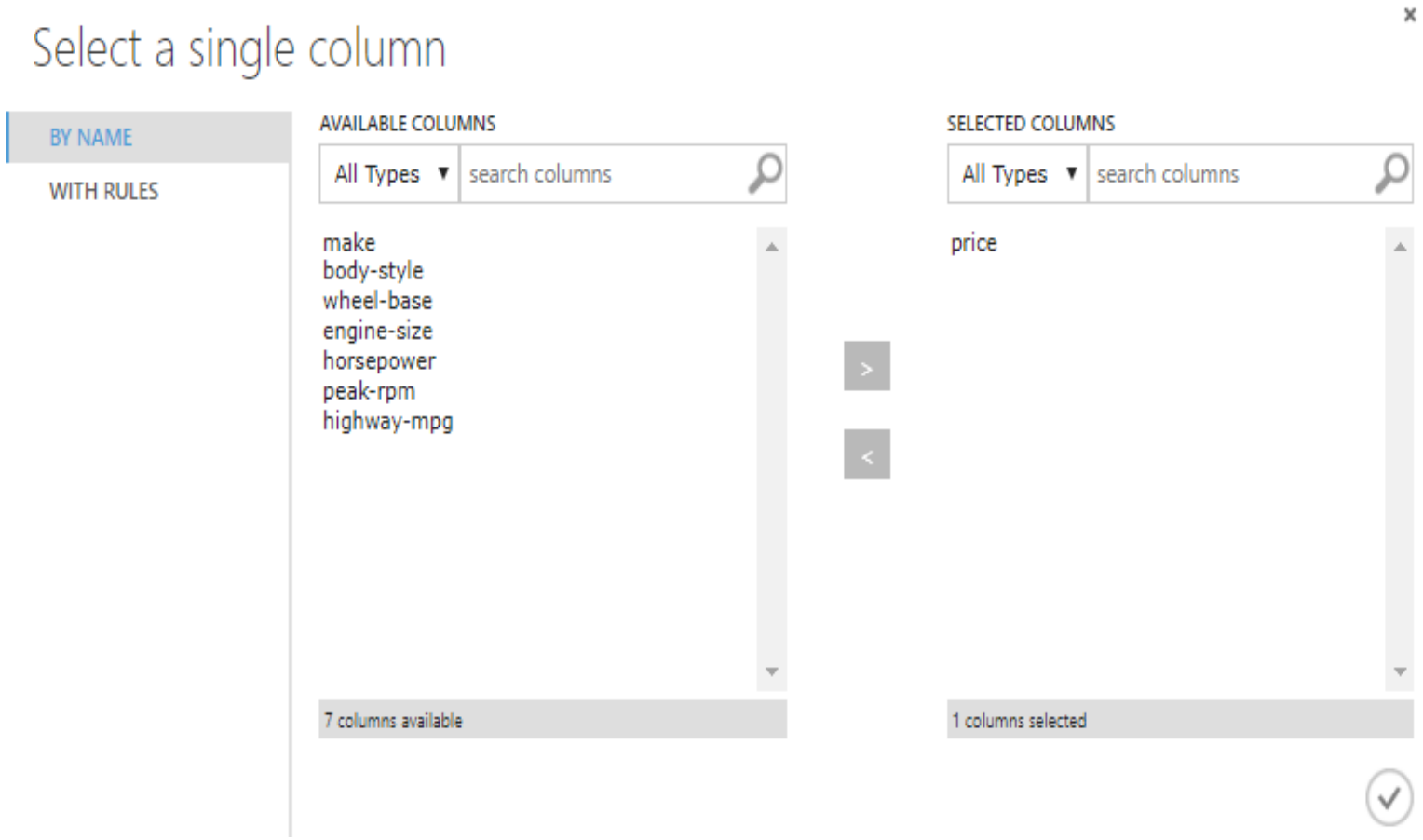
Selecting learning Algorithms

- Drag the **Train Model** module to the experiment canvas.
- Connect the output of the **Linear Regression** module to the left input of the **Train Model** module
- Connect the **training data output** of the **Split Data** module to the right input of the **Train Model** module.
- Run the experiment.



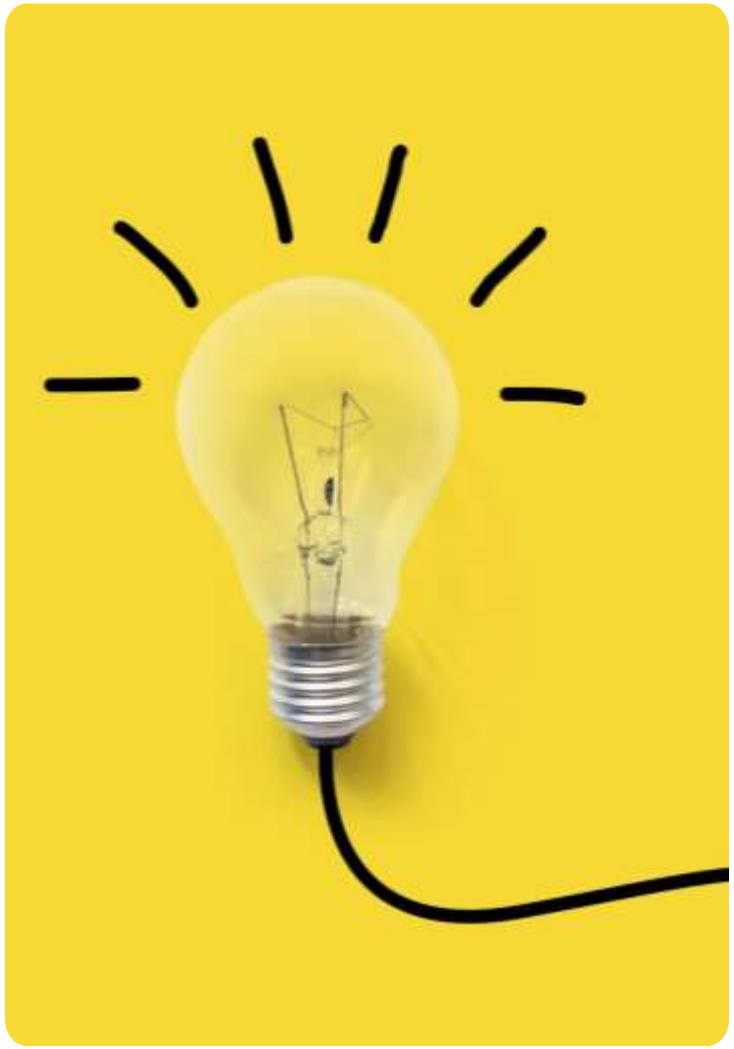
Training the Model

- Click the **Train Model** module, click **Launch column selector** in the **Properties** pane, and then select the **price** column.
- This is the value that our model is going to predict.
- Run the experiment.



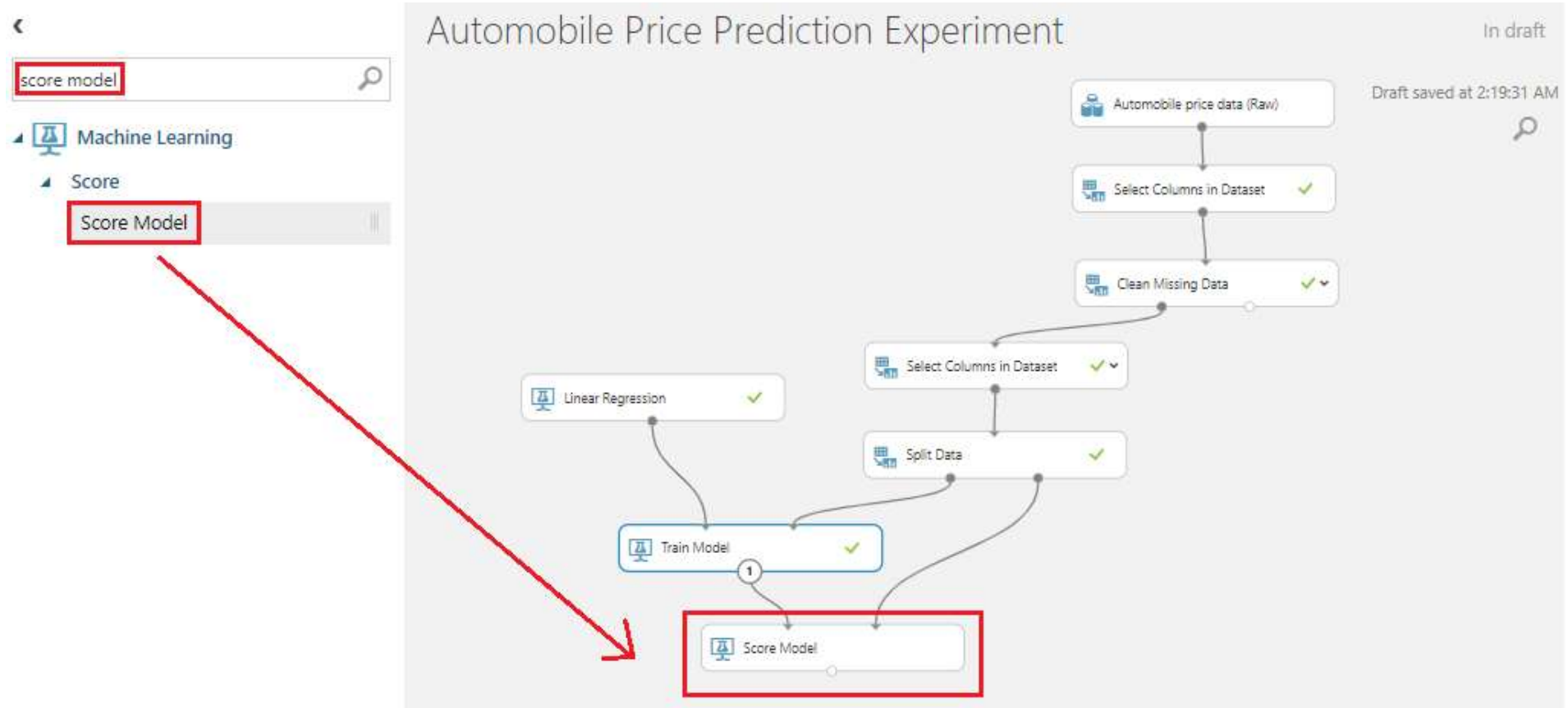
5. Prediction

- We have trained the model using 75 percent of data, we can use it to score the other 25 percent of the data to see how well our model functions.



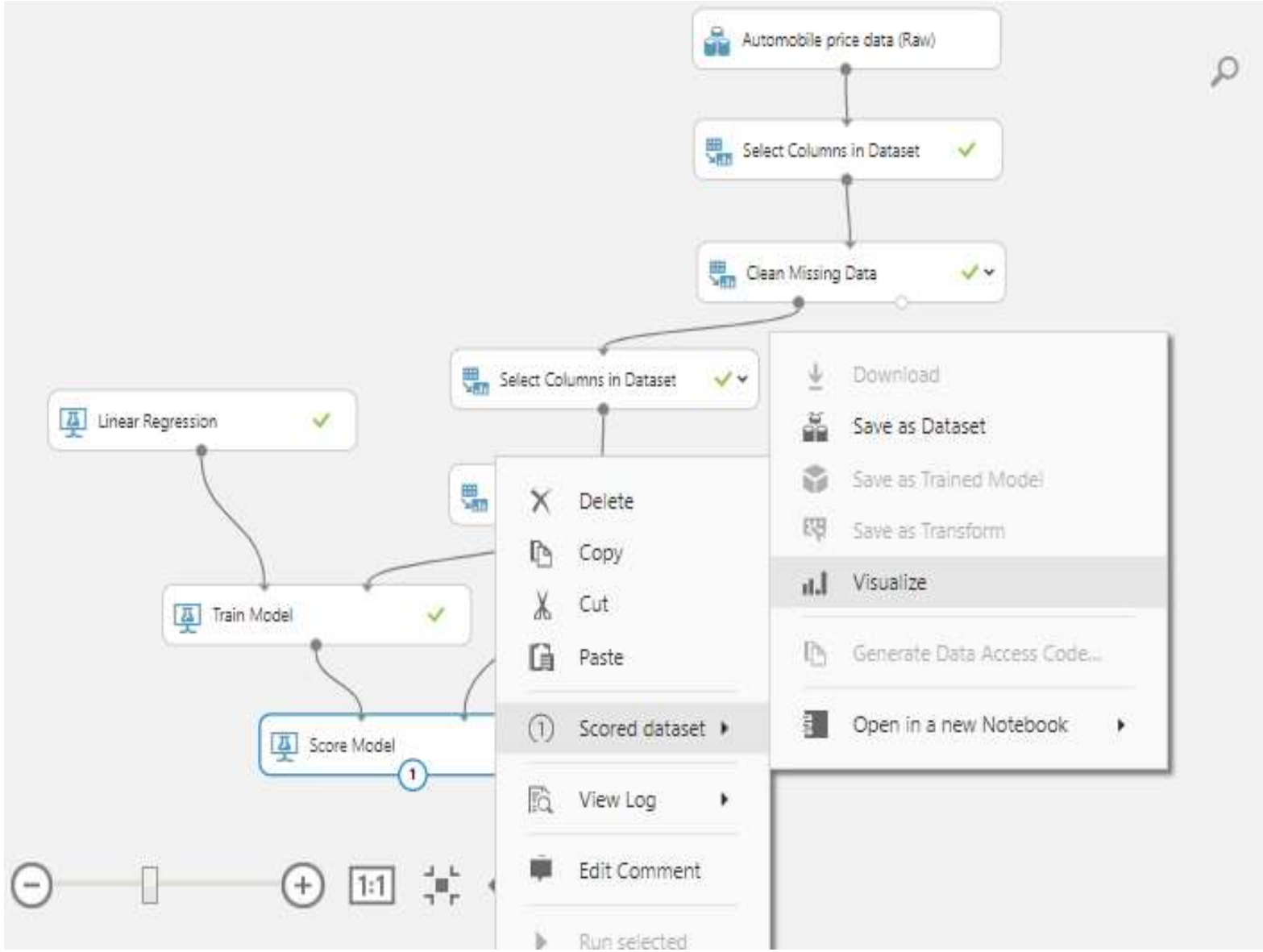
Predict Price

- Find and drag the **Score Model** module to the experiment canvas.
- Connect the output of the **Train Model** module to the left input port of **Score Model**.
- Connect the **test data output** (right port) of the **Split Data** module to the **right input port** of **Score Model**.



Predicted Price

- Run the experiment.
- View the output from the **Score Model** module.



Predicted Price..

- The output shows the predicted values for price and the known values from the test data.

Automobile Price Prediction Experiment › Score Model › Scored dataset

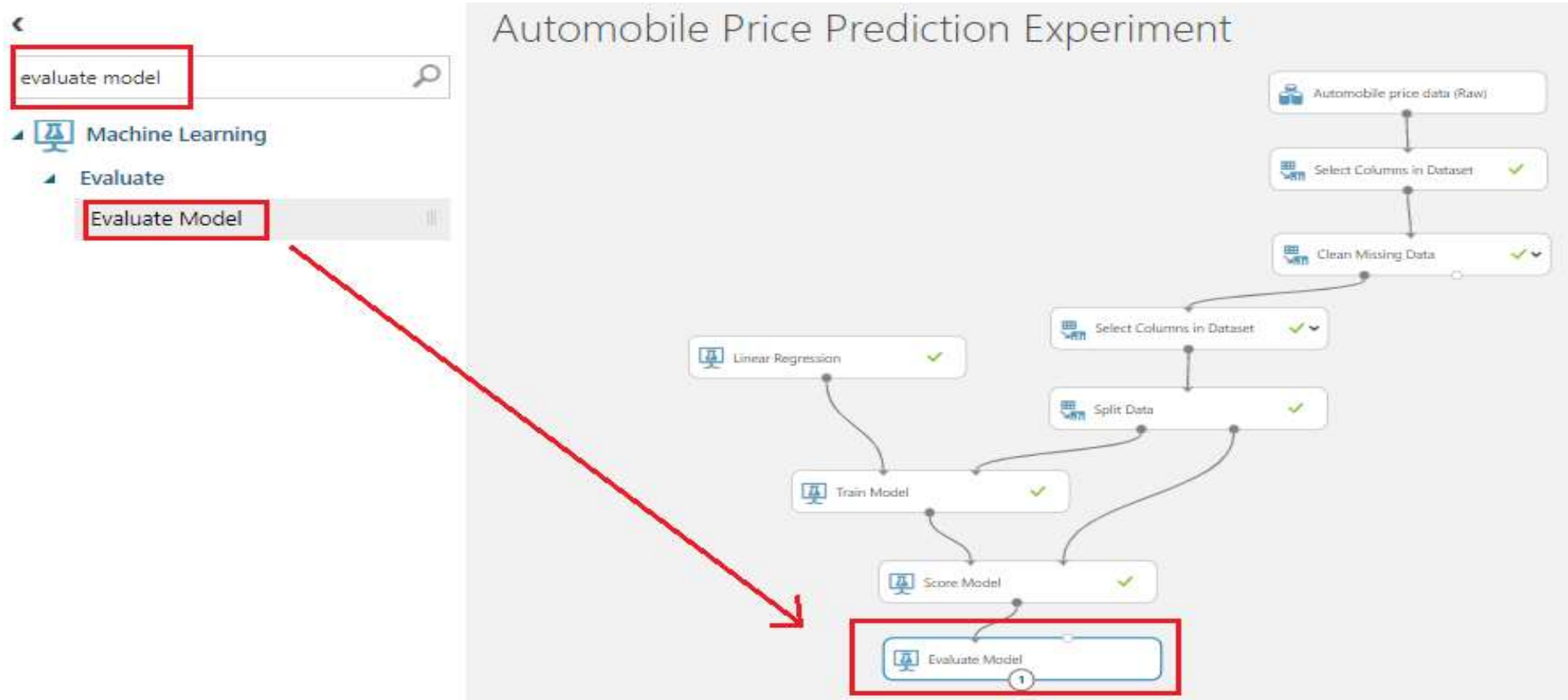
rows 48 columns 9

view as

make	body-style	wheel-base	engine-size	horsepower	peak-rpm	highway-mpg	price	Scored Labels
subaru	sedan	97	108	111	4800	29	11259	10286.204819
mitsubishi	hatchback	93.7	92	68	5500	38	6669	5446.847864
dodge	hatchback	93.7	90	68	5500	38	6229	6344.800711
honda	hatchback	86.6	92	76	6000	38	6855	5528.302953
alfa-romero	convertible	88.6	130	111	5000	27	16500	13498.476233
volvo	wagon	104.3	141	114	5400	28	16515	16097.608038
isuzu	hatchback	96	119	90	5000	29	11048	8315.257218
dodge	hatchback	93.7	90	68	5500	41	5572	6630.154608
bmw	sedan	101.2	108	101	5800	29	16430	19913.408695
mitsubishi	hatchback	93.7	92	68	5500	41	5389	5732.201761
bmw	sedan	103.5	209	182	5400	22	41315	30548.819502

Evaluate Model

- Select and drag the **Evaluate Model** module to the experiment canvas and connect the output of the **Score Model** module to the left input of **Evaluate Model**.
- Run the experiment.
- To view the output from the Evaluate Model module, click the output port, and then select Visualize.



Evaluation Result

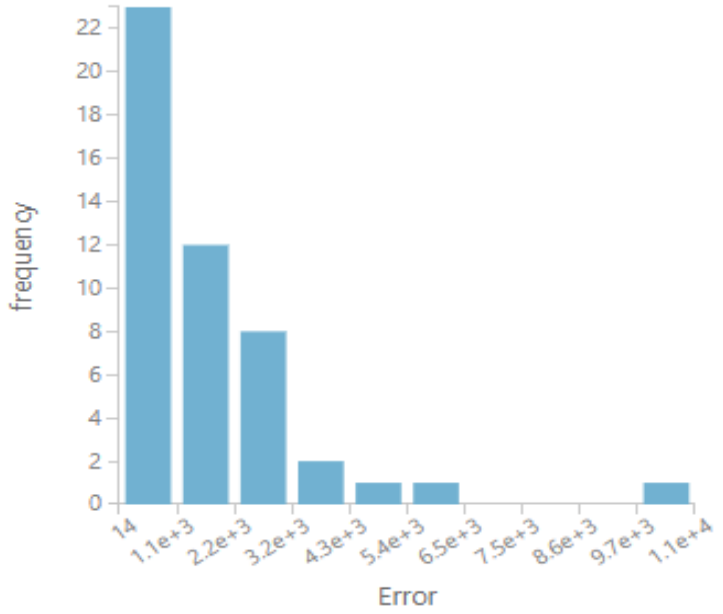
- A smaller value indicates that the predictions more closely match the actual values.
- For Coefficient of Determination, the closer its value is to one (1.0), the better the predictions.

Automobile Price Prediction Experiment › Evaluate Model › Evaluation results

Metrics

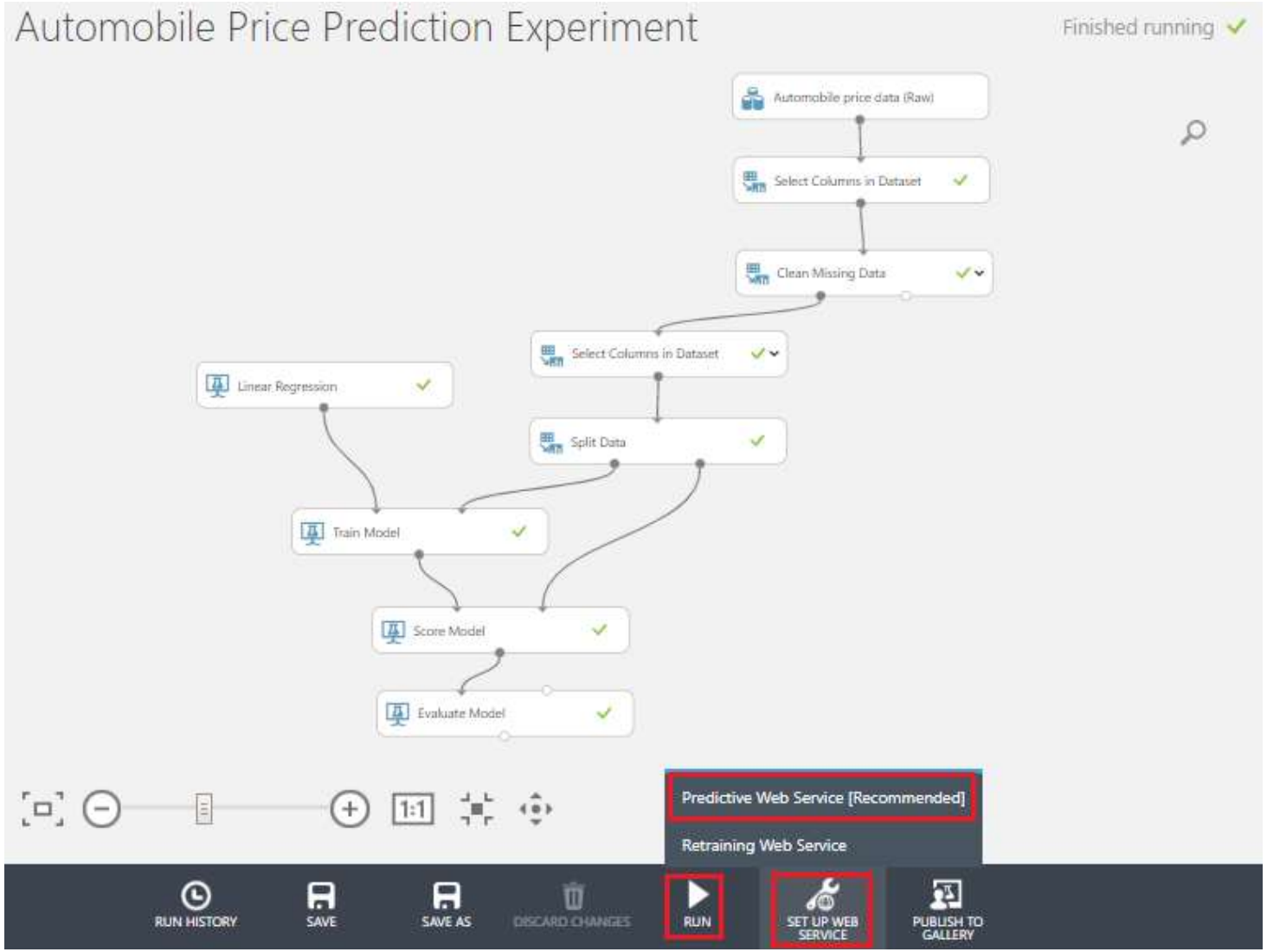
Mean Absolute Error	1656.147651
Root Mean Squared Error	2456.983209
Relative Absolute Error	0.276606
Relative Squared Error	0.089608
Coefficient of Determination	0.910392

Error Histogram

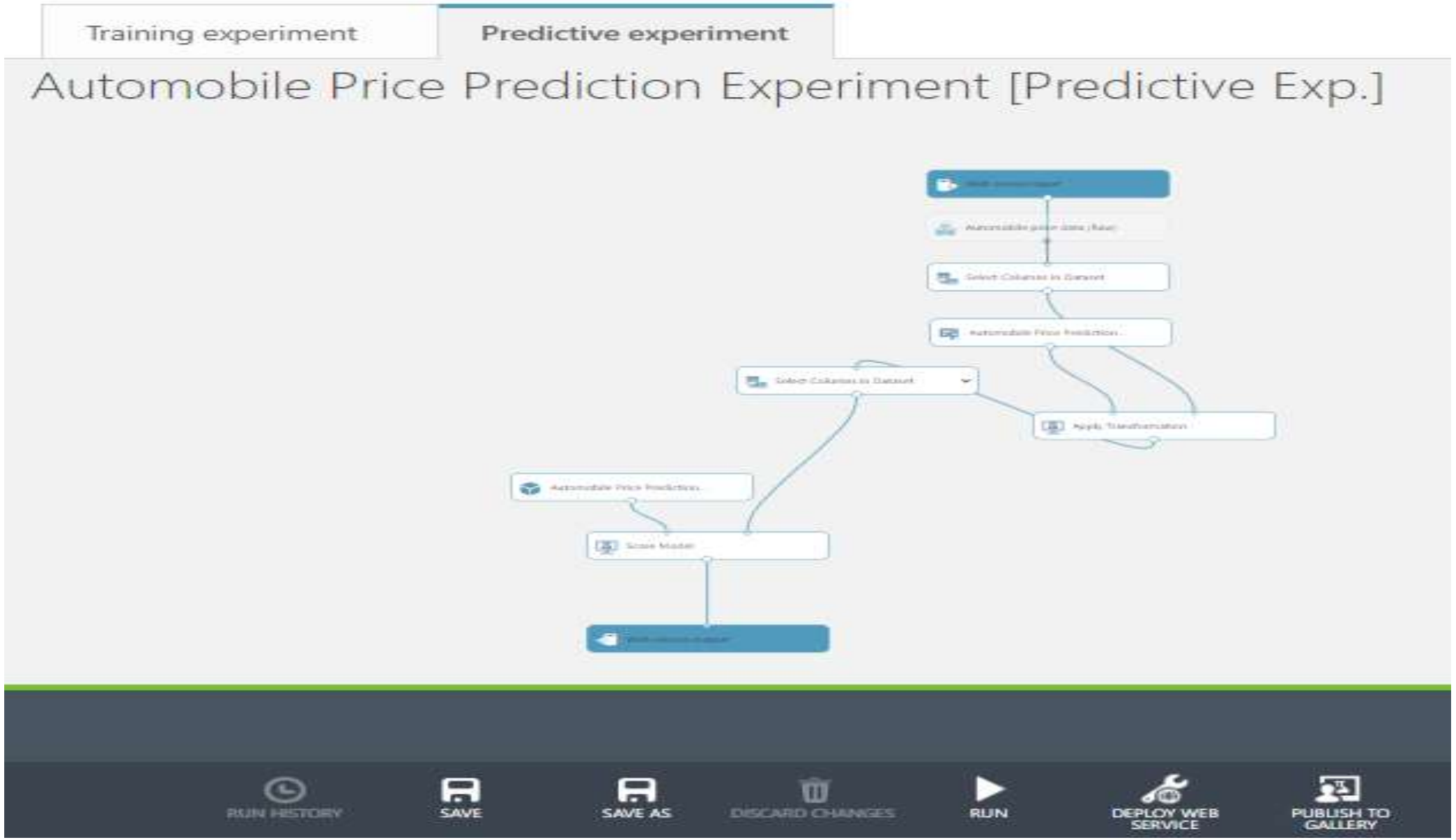


Deploy the model as a web service..

- When you're satisfied with your model, you can deploy it as a web service to be used to predict prices by using new data.
- Before we execute the next command, now is a good time to save the experiment.
- To convert training experiment to a predictive experiment, click **Run**, then click **Set Up Web Service**, then select **Predictive Web Service**.

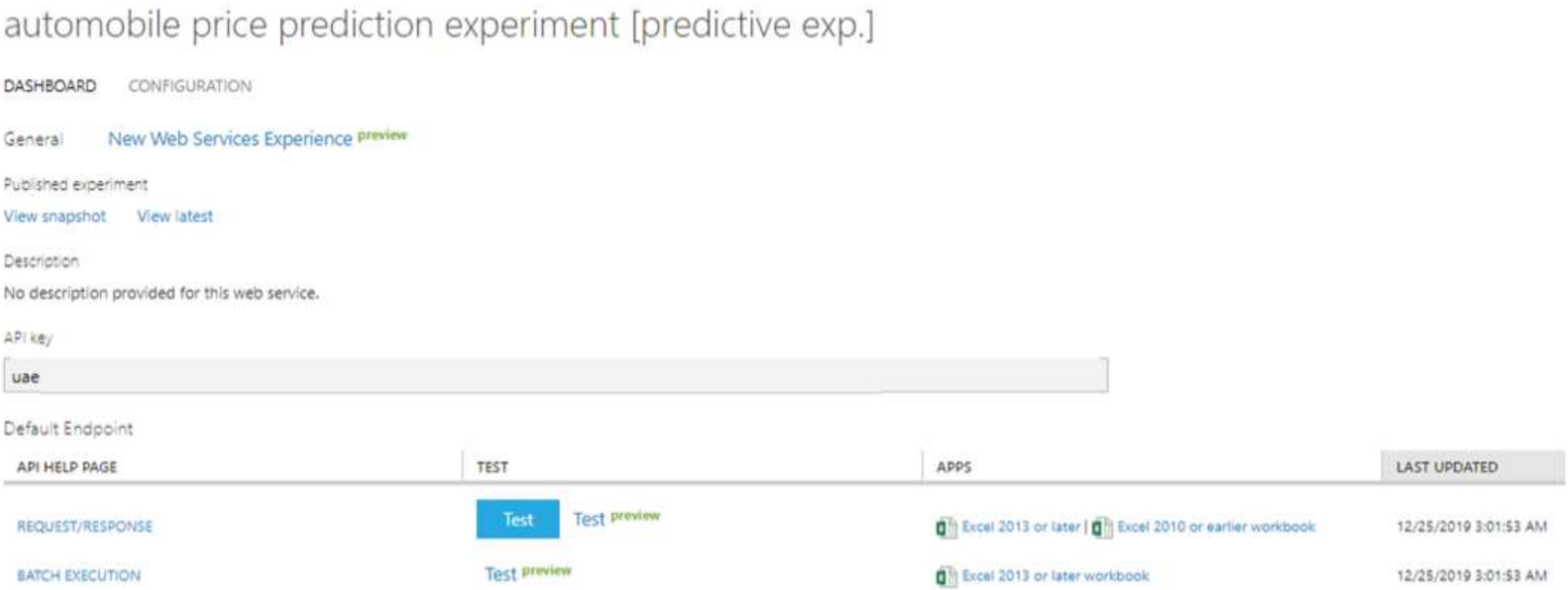


Deploy the model as a web service..



Deploy the model as a web service..

- To deploy predictive experiment, click **Run** at the bottom of the experiment canvas. Once the experiment has finished running, click **Deploy Web Service** and select **Deploy Web Service New**.
- The deployment page of the Machine Learning Web Service portal opens

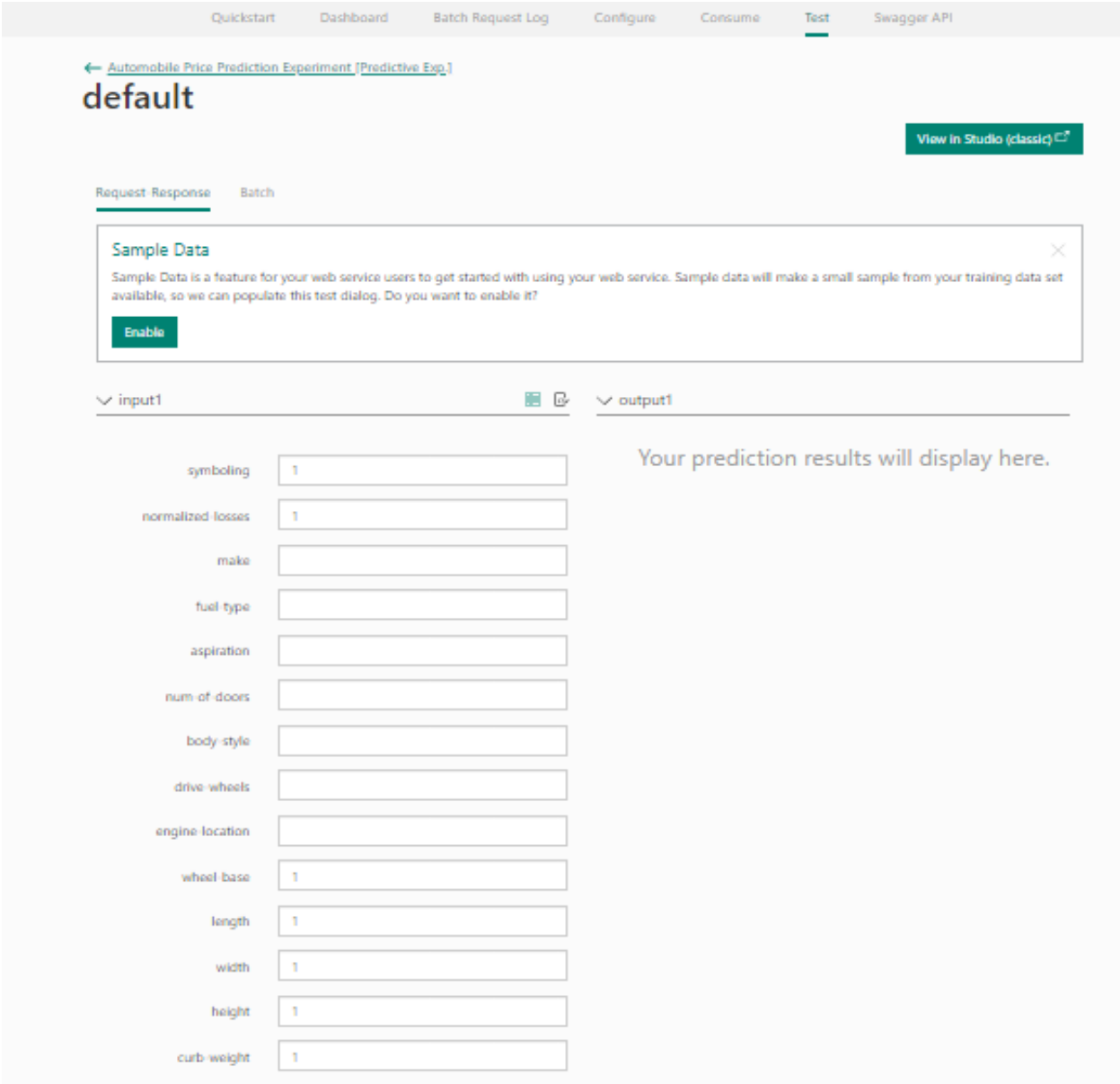


Deploy the model as a web service..

- To test your new web service, click **Test web service** under common tasks. On the Test page, you can test your web service as a Request-Response Service (RRS) or a Batch Execution service (BES).

- The RRS test page displays the inputs, outputs, and any global parameters that you have defined for the experiment. To test the web service, you can manually enter appropriate values for the inputs or supply a comma separated value (CSV) formatted file containing the test values.

- To test using RRS, from the list view mode, enter appropriate values for the inputs and click **Test Request-Response**. Your prediction results display in the output column to the left.



References

Start with Azure Home Page to know end-to-end ML lifecycle

Followed by Official documentation, blog-post, Microsoft Learn & videos.

And finally, Azure ML Cheat Sheet

Thank You.