Batch: A3 **Experiment Number:7**

Roll Number: 16010421073 **Name:Keyur Patel**

Aim of the Experiment:

Describe the following points with respect to the business under consideration,

---

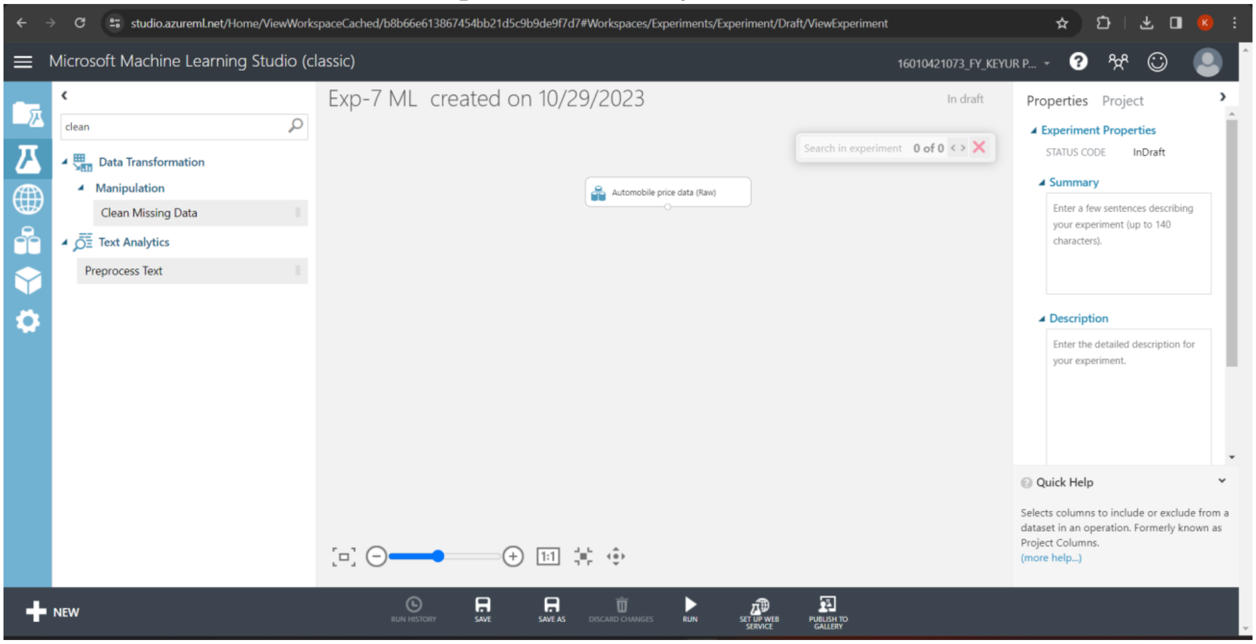
Program/ Steps:

Step 1. Sign-in using Microsoft account on studio.azureml.net
Step 2. Creating workspace for our Machine Learning project.
Step 3. Select New option on bottom right:
Step 4. Click on Blank experiment and write name and summary of experiment
Step 5. Select From Saved Datasets-> Samples-> dataset of your choice
Step 6. Now, search 'Select columns in dataset' from items and drag it
Step 7. Now, click on launch column selector-> with rules->exclude column normalized-losses as that column contains many rows/records with empty values.
Step 8. Search and select 'Clean Missing Data' from items list
Step 9. Now, select cleaning mode -> Remove entire row as it will remove the entire row wherever missing value is found
Step 10. Again choose 'select columns in dataset'
Step 11. Now, launch column selector and include all the columns based on which prediction is to be done: make, body-style, wheel-base, engine-size, horsepower, peak-rpm, highway-mpg, price
Step 12. Now, select 'split data' from list and drag it
Step 13. For Split data, enter the fraction of data which is needed for training while rest will be used for testing
Step 14. Now, Select 'Linear Regression' as the algorithm to be used and 'Train Model' from list
Step 15. For training model, click on launch column selector, include price column as Price is what is to be predicted
Step 16. Add Score Model from list drag it and make connections
Step 17. Now, Add Evaluate Model from list and make connections
Step 18. Now, Click on Run
Step 19. To check prediction results, right click on Score Model, select visualize
Step 20. To check Evaluation results, right click on Evaluation Model, select visualize

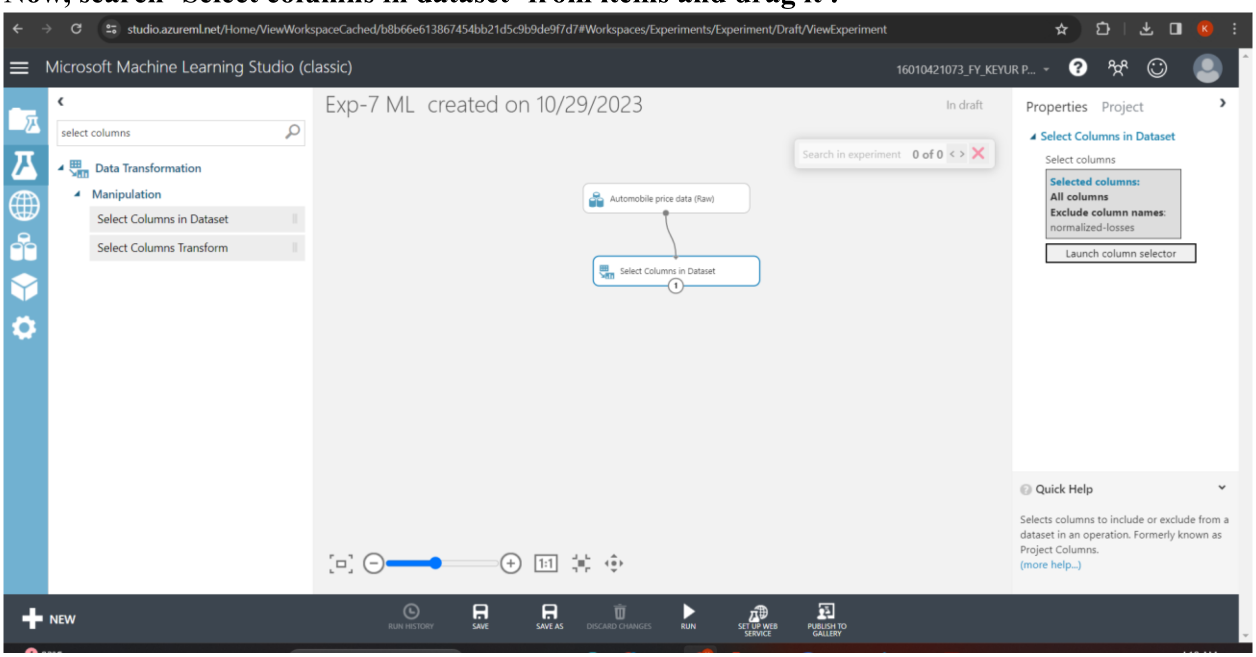
---

Output/Result:

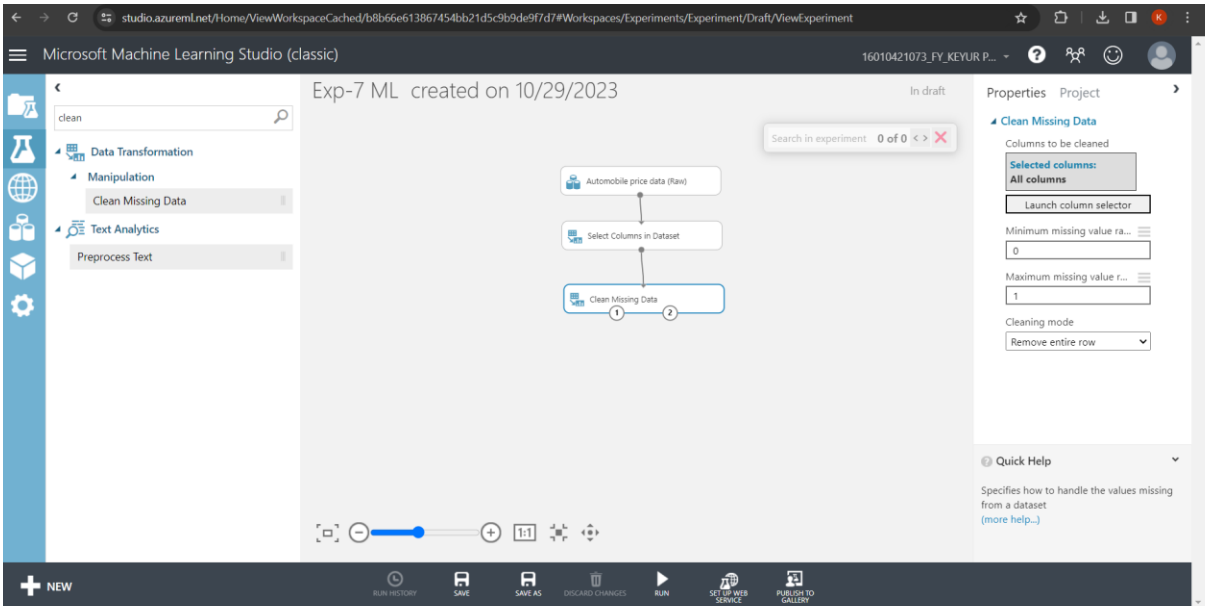
- **Select From Saved Datasets-> Samples-> dataset of your choice .**



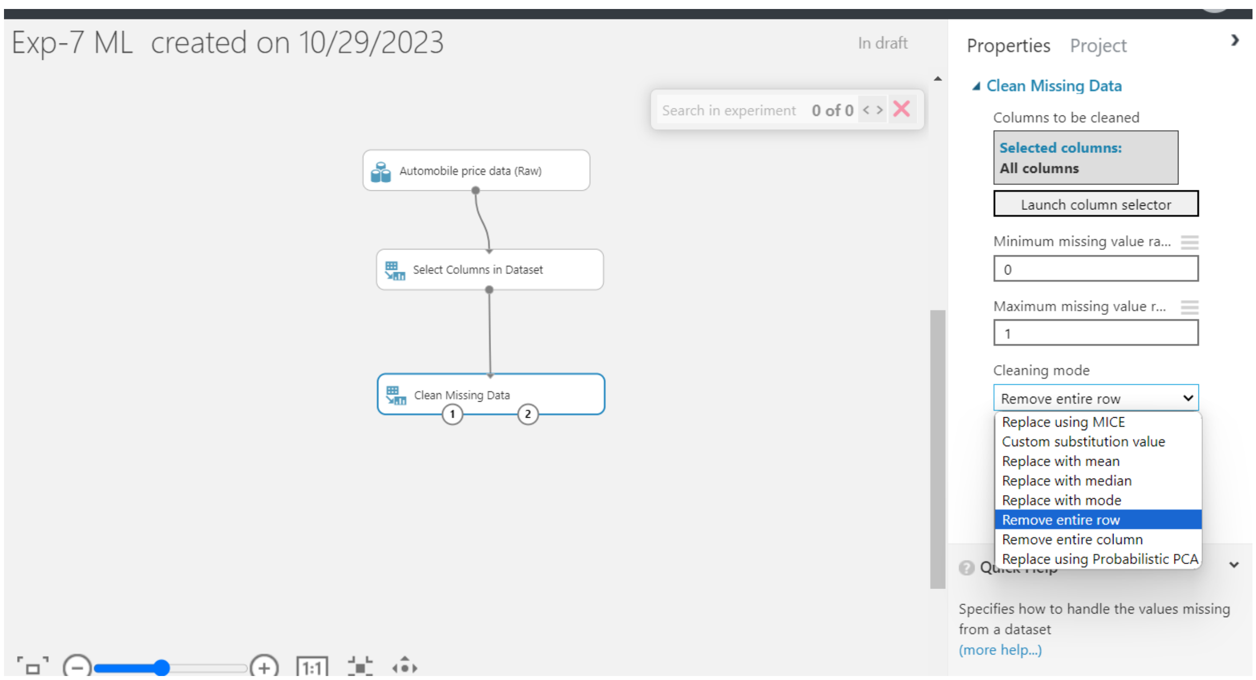
- **Now, search ‘Select columns in dataset’ from items and drag it .**



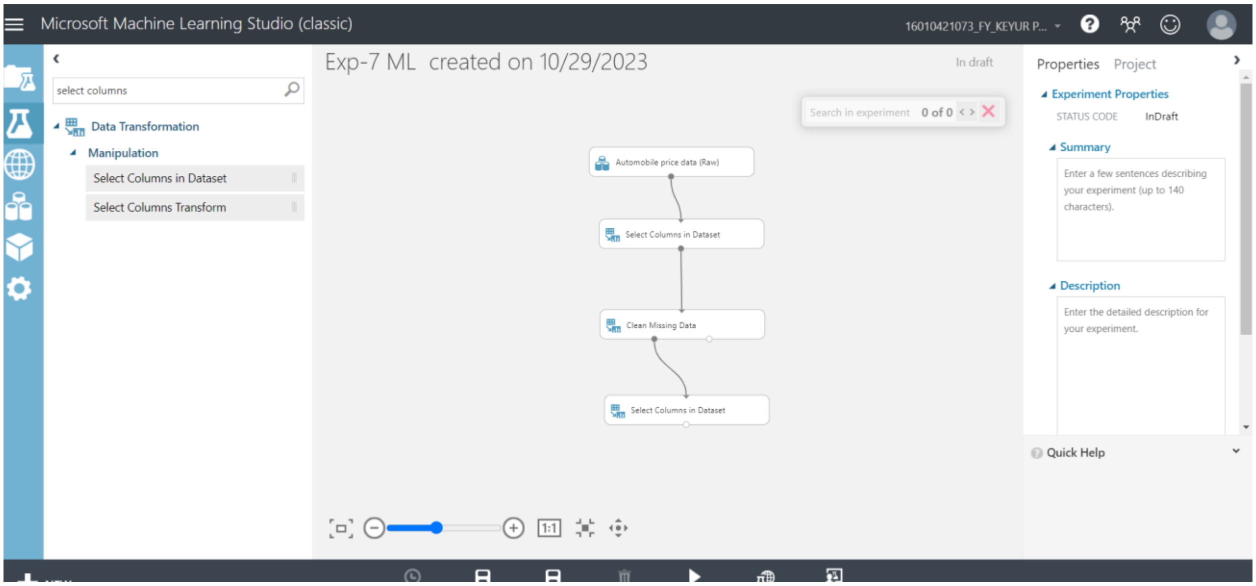
- **Now, click on launch column selector-> with rules->exclude column normalized-losses as that column contains many rows/records with empty values.**



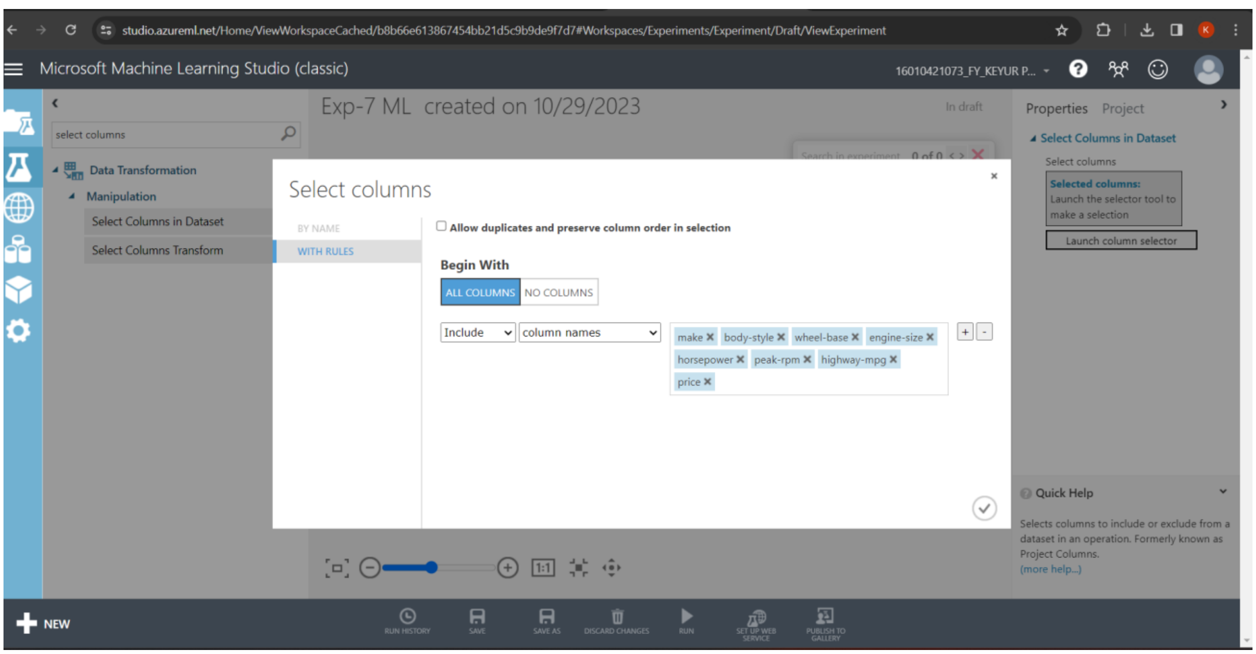
- **Now, select cleaning mode -> Remove entire row as it will remove the entire row wherever missing value is found**



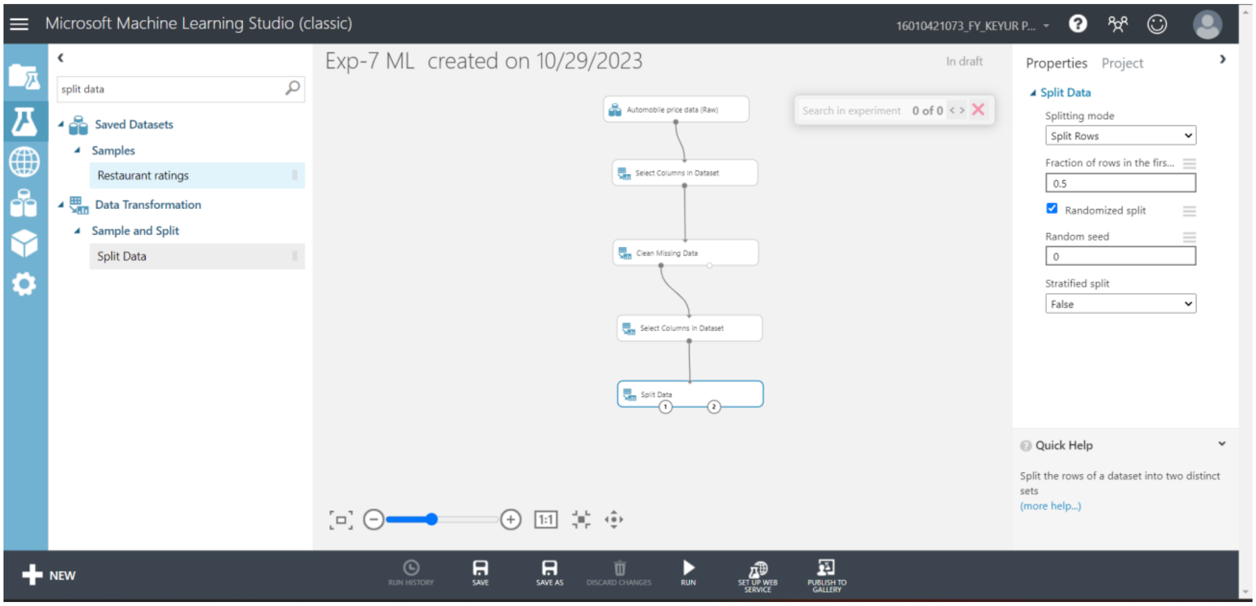
- **Again choose 'select columns in dataset'**



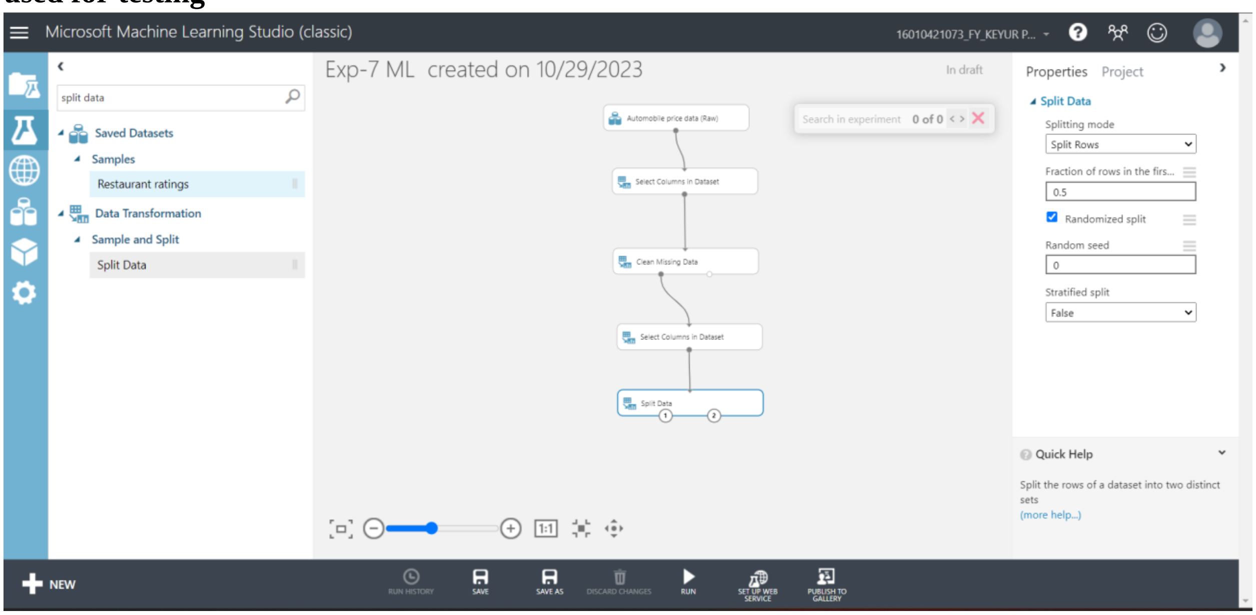
- **Now, launch column selector and include all the columns based on which prediction is to be done: make, body-style, wheel-base, engine-size, horsepower, peak-rpm, highway-mpg, price**



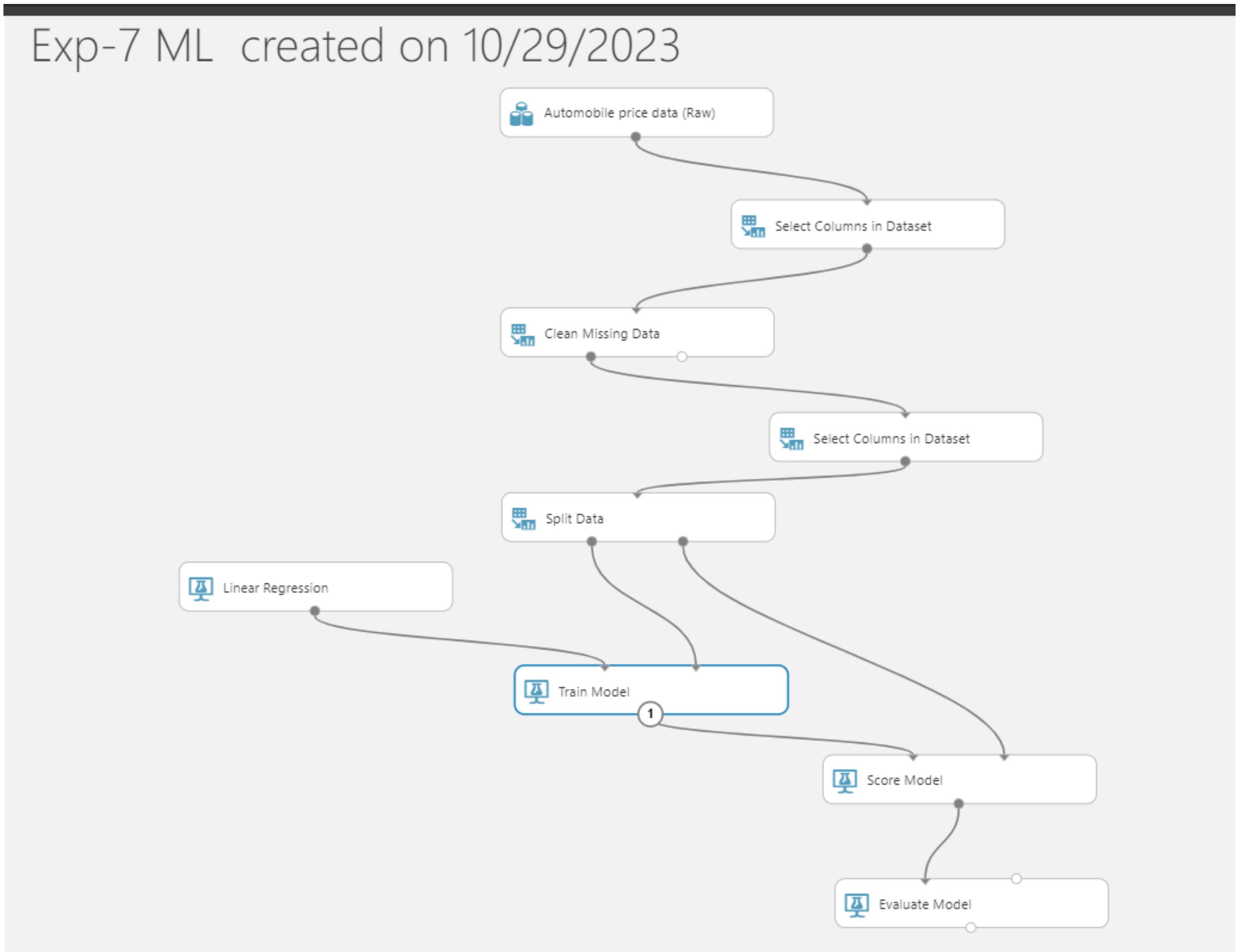
- **Now, select 'split data' from list and drag it**



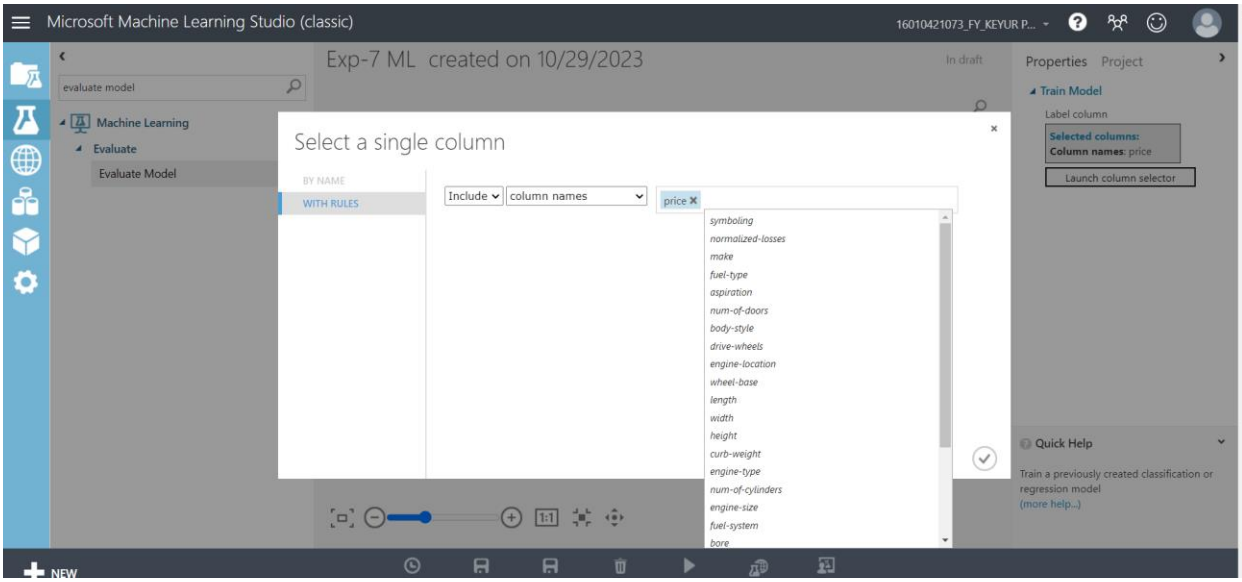
- **For Split data, enter the fraction of data which is needed for training while rest will be used for testing**

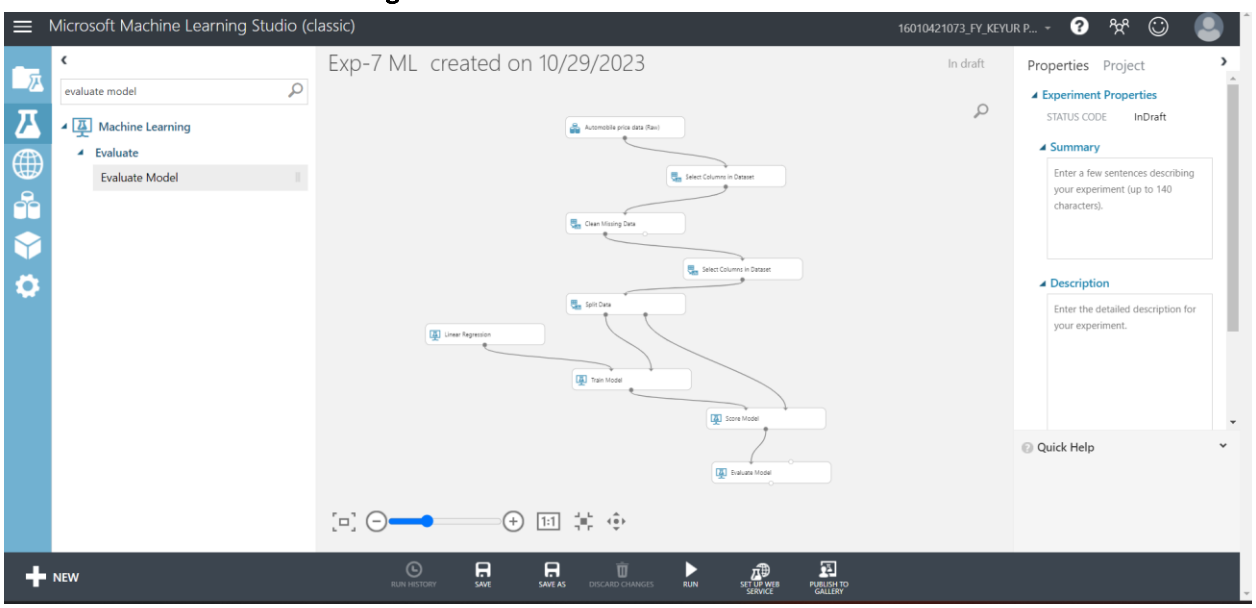


- **Now, Select ‘Linear Regression’ as the algorithm to be used and ‘Train Model’ from list**

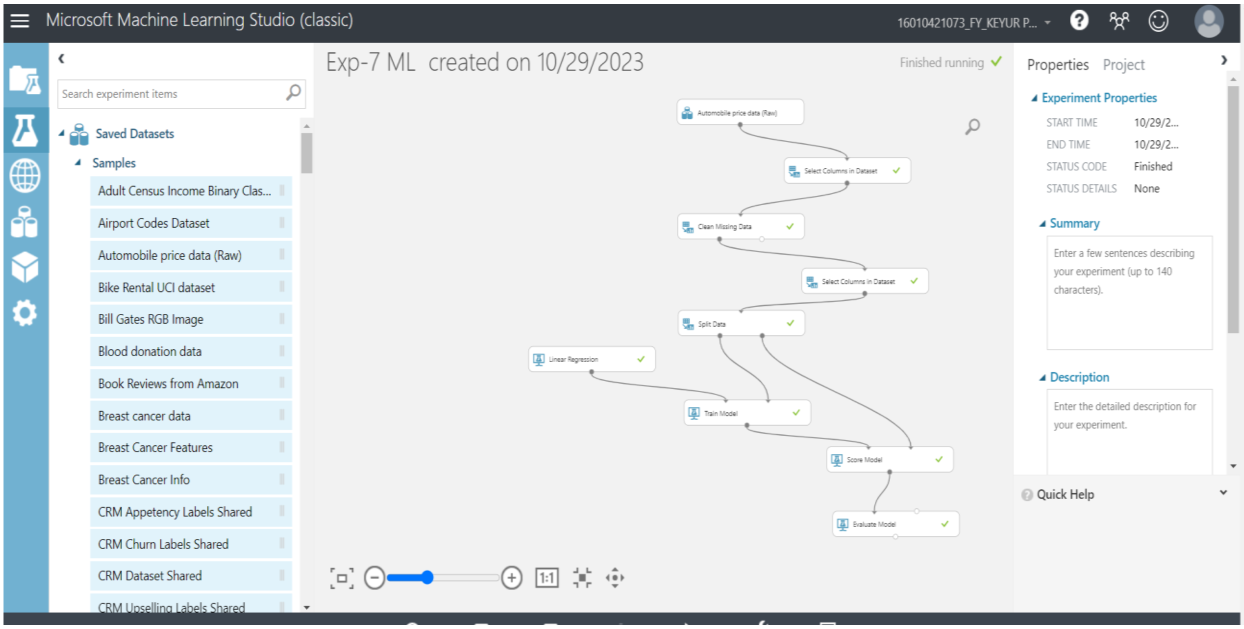


- **For training model, click on launch column selector, include price column as Price is what is to be predicted**

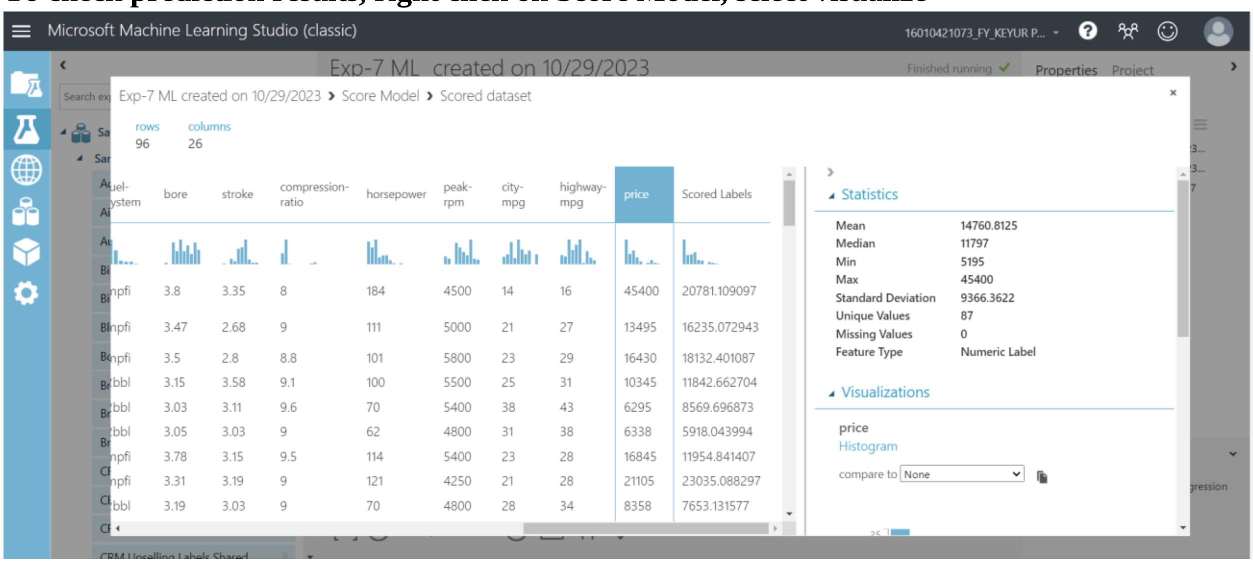


- **Add Score Model from list drag it and make connections**
- 

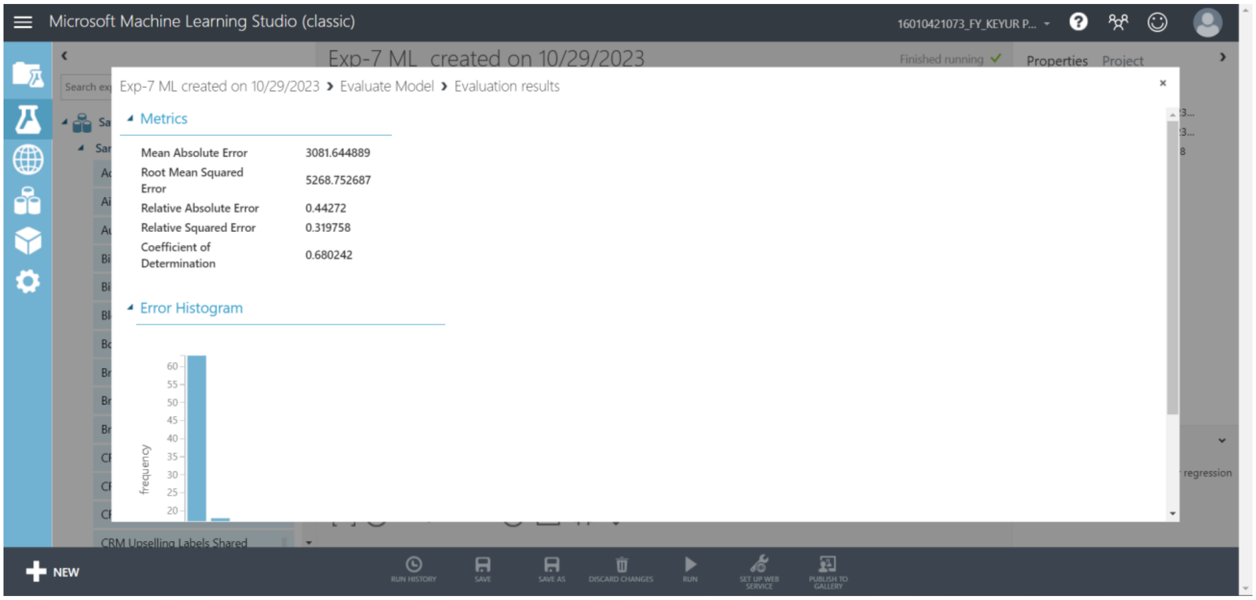
- **Now, Click on Run**

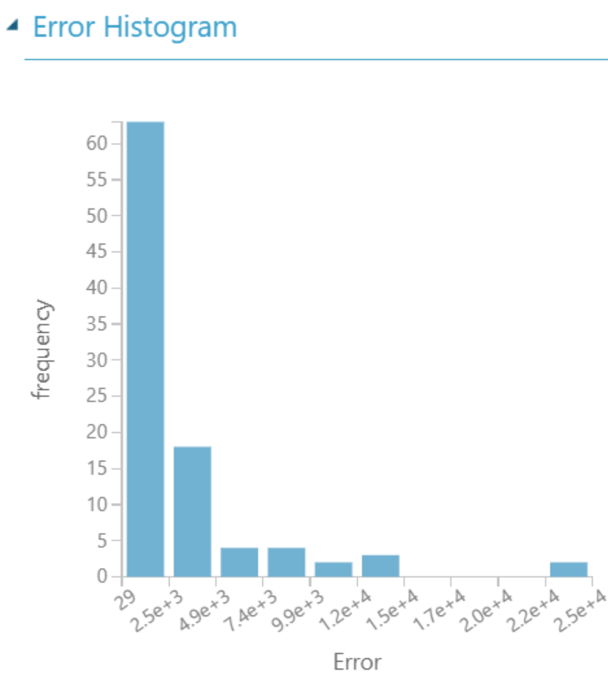


- **To check prediction results, right click on Score Model, select visualize**



- **To check Evaluation results, right click on Evaluation Model, select visualize**





Post Lab Question-Answers:

1) Differentiate between linear and nonlinear regression.
Ans:

Aspect	**Linear Regression**	**Nonlinear Regression**
Mathematical	$Y = \beta0 + \beta1X + \varepsilon$	$Y = f(\beta0, \beta1, \beta2, ..., \beta n, X) + \varepsilon$

Formulation		
Linearity Assumption	Assumes a linear relationship between the predictor(s) and the response variable.	Does not assume a linear relationship; the relationship can be any functional form.
Parameter Estimation	Typically involves estimating coefficients (β0, β1, etc.) using methods like Ordinary Least Squares (OLS).	Involves estimating the parameters (β0, β1, β2, etc.) of a chosen nonlinear function, which may require more complex optimization techniques.
Model Complexity	Generally simpler models with fewer parameters.	Can involve complex and highly flexible functional forms, allowing for a better fit to data but potentially leading to overfitting.
Model Interpretability	Coefficients (β0, β1, etc.) are interpretable as the change in the response variable associated with a unit change in the predictor(s).	Parameter interpretation can be challenging, as the relationship is nonlinear and may not have simple interpretations.
Residual Analysis	Linear regression often assumes homoscedasticity (constant variance of errors) and independence of residuals.	Residuals may not exhibit homoscedasticity or independence, and model assumptions may need to be checked carefully.
Use Cases	Well-suited for modeling linear relationships, such as simple trends, correlations, and linear patterns.	Appropriate for modeling more complex, nonlinear relationships, such as exponential growth, sigmoidal curves, or polynomial functions.
Assumption Violations	Sensitivity to violations of linearity and other assumptions, which can lead to biased estimates and unreliable predictions.	More robust to violations of linearity, but the choice of the nonlinear function should be guided by domain knowledge.
Model Selection	Typically involves selecting the predictor(s) and checking for multicollinearity.	Involves selecting both the functional form and the predictor(s), which can be more challenging.
Examples	Simple linear regression, multiple linear regression.	Logistic regression, exponential growth models, polynomial regression.

2) Write a note on converting non-linear model into linear model
Ans:

Converting a non-linear model into a linear model is a common technique in machine learning and statistics when dealing with complex data or improving model interpretability. This transformation simplifies the model while sacrificing some of the expressiveness of non-linearity. Here are some common methods for converting non-linear models into linear ones:

- **Feature Engineering:**
  One of the simplest approaches is to engineer features to make the model more linear. For example, if you have a non-linear relationship between two variables, you can create

interaction terms or polynomial features to capture the non-linear behavior. This can sometimes make the data more amenable to linear modeling techniques like linear regression.

- **Logarithmic or Exponential Transformations:**
  Applying logarithmic or exponential transformations to the features or the target variable can help linearize relationships. For instance, when dealing with exponential growth in data, taking the logarithm can often make it linear.
- **Piecewise Linearization:**
  For complex non-linear relationships, you can divide the data into segments and fit separate linear models to each segment. This approach is known as piecewise linearization and is particularly useful when you have domain knowledge that suggests different linear behaviors in different regions of the data.
- **Spline Models:**
  Splines are piecewise-defined polynomial functions that can approximate non-linear data by combining several linear segments. There are various types of splines, such as cubic splines or B-splines, that can be used to approximate non-linear relationships.
- **Kernel Methods:**
  Kernel methods transform non-linear data into a higher-dimensional space, where it might be more linear. Support Vector Machines (SVMs) use the kernel trick to map data into a high-dimensional feature space and then find a linear separator. Kernelized models can approximate non-linear functions effectively.
- **Neural Network with Linear Activation:**
  While neural networks are inherently non-linear models, you can create a network with only linear activation functions, effectively making it a linear model. However, this approach might not be as expressive as a true non-linear neural network.
- **Regularization:**
  Regularization techniques like Lasso (L1) or Ridge (L2) can help make non-linear models behave more like linear models by penalizing the magnitude of coefficients. This can promote sparsity or shrink coefficients, making the model simpler and more linear.
- **Principal Component Analysis (PCA):**
  PCA is a dimensionality reduction technique that can be used to project high-dimensional data onto a lower-dimensional space while preserving as much variance as possible. In some cases, this projection can make the data more amenable to linear modeling.

Outcomes:

CO2 : Apply concepts of different types of Learning and Neural Network.

Conclusion (based on the Results and outcomes achieved):
Thus we successfully developed a regression model using Microsoft Azure Machine Learning Studio.

References:

Books/ Journals/ Websites: