Supervised Learning: Classification + Regression

Saturday, July 18, 2020 1:23 AM

- Supervised Learning
  - Classification: outputs are categorical or discrete
    - Classification of tabular data
      - Data available in form of rows and columns
    - Classification on image or sound data
      - Data available in images or sound whose categories are known
      - Data transformed into numerical vectors accepted by algorithms
    - Classification on text data
      - Data available in text those categories are known
      - Data transformed into numerical vectors accepted by algorithms
    - Ex. Computer vision, speech recognition, biometric identification, document analysis, sentiment analysis, credit scoring, anomaly detection
  - Categories of Algorithms - 3 main types
    - Two-class Classification (binary)
      - Two categories
      - Algorithms

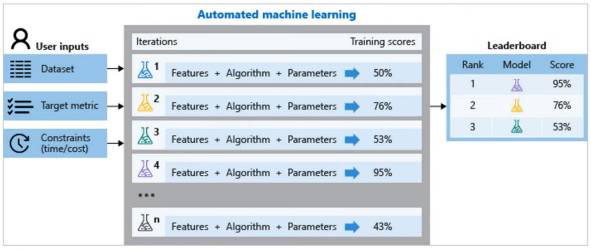
Algorithm	Characteristics
Two-Class Support Vector Machine	Under 100 features, linear model
Two-Class Averaged Perceptron	Fast training times, linear model
Two-Class Decision Forest	Accurate, fast training times
Two-Class Logistic Regression	Fast training times, linear model
Two-Class Boosted Decision Tree	Accurate, fast training, large memory footprint
Two-Class Neural Network	Accurate, long training times

    - Multi-class Single-Label Classification
      - Multiple categories, output belongs to single category
      - Algorithms

Algorithm	Characteristics
Multi-Class Logistic Regression	Fast training times, linear model
Multi-Class Neural Network	Accurate, long training times
Multi-Class Decision Forest	Accurate, fast training times
Multi-Class Boosted Decision Tree	Non-parametric, fast training times, and scalable
One-vs-All Multiclass	Depends on the underlying two-class classifier

    - Multi-class Multi-label Classification
      - Multiple categories, output belongs to one or more categories
  - Regression : outputs are numerical / continuous
    - Regression on tabular data
      - Data is available in the form of rows and columns
    - Regression on image of data
      - Training data consists of images/sounds whose numerical scores are already known
      - Data transformed into numerical vectors accepted by algorithms
    - Regression on text data
      - Training data consists of texts whose numerical scores are already known
      - Data transformed into numerical vectors accepted by algorithms
    - Ex. Housing prices, customer churn, customer lifetime value, forecasting, anomaly detection
  - Categories of Algorithms
    - Linear Regression
      - Fast training, linear model
      - Linear relationship between independent variables and a numeric outcome (dependent variables)
      - Approaches:
        - Ordinary Least Squares method
          - Computes error as sum of squares of error from actual value to predicted line, fits model by minimizing square error
          - Assumes strong linear relationship between input and dependent variable
        - Gradient descent
          - Minimize error at each step of model training process
      - Decision Forest Regression
        - Accurate, fast training times
        - Ensemble learning method that builds multiple decision trees and each tree outputs a distribution as a prediction
        - Aggregation performed over ensemble of trees to find distribution closest to the combined distribution
        - Same hyperparameters as the classification version
      - Neural Network Regression
        - Accurate, long training times
        - Supervised learning, requires tagged dataset and a label column
        - Label column must be numerical data type
        - Default architecture: input layer, one hidden layer, output later
        - Same hyperparameters as classification version
  - Automating Training of Regressors
    - Conventional ML Process
      - Features available in data sets
      - Algorithms that are suitable for the task
      - Hyperparameter tuning
      - Evaluation metrics
    - Automated ML
      - Automate testing the combinations need to produce a successful and accurate model
      - Intelligently tset multiple algorithms and hyper-parameteres in parallel
        - Deploy into production
        - Further customization and refinement
      - Azure ML Automated ML Process

The Automated ML Process



    - Default behavior for AutoML with missing values for categorical features is to ***impute with most frequent value***

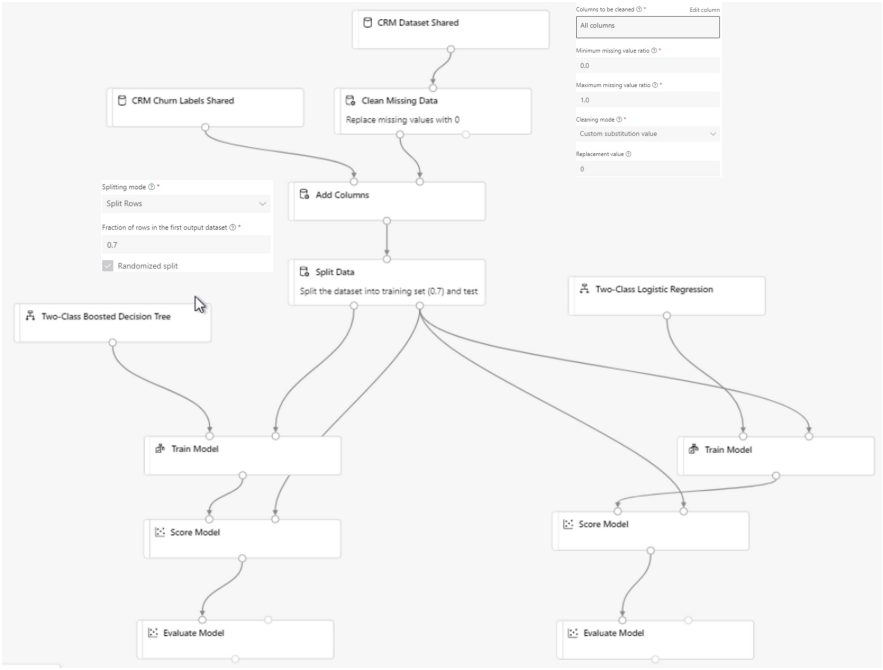
Lab: Two-Class Classifiers Performance

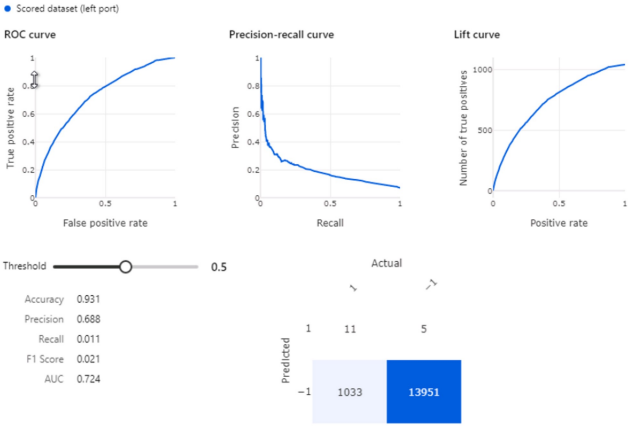
Monday, July 20, 2020 1:21 PM

Lab Overview

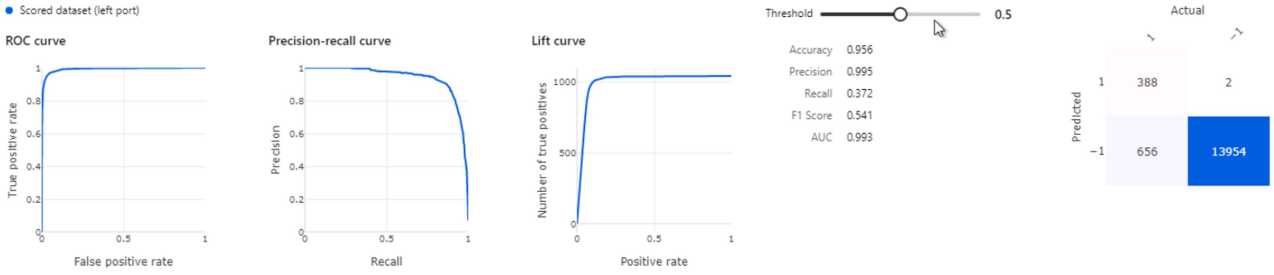
Azure Machine Learning designer (preview) gives you a cloud-based interactive, visual workspace that you can use to easily and quickly prep data, train and deploy machine learning models. It supports Azure Machine Learning compute, GPU or CPU. Machine Learning designer also supports publishing models as web services on Azure Kubernetes Service that can easily be consumed by other applications.

In this lab, we will be compare the performance of two binary classifiers: Two-Class Boosted Decision Tree and Two-Class Logistic Regression for predicting customer churn. The goal is to run an expensive marketing campaign for high risk customers; thus, the **precision** metric is going to be key in evaluating performance of these two algorithms. We will do all of this from the Azure Machine Learning designer without writing a single line of code.





Score bin ↓	Positive exam...	Negative exam...	Fraction above thresh...	Accuracy	F1 Score	Precisi...	Recall	Negative precisi...	Negative recall	Cumulative AUC
(0.900,1.000]	0	0	0.000	0.930	0.000	1.000	0.000	0.930	1.000	0.000
(0.800,0.900]	1	0	0.000	0.930	0.002	1.000	0.001	0.930	1.000	0.000
(0.700,0.800]	1	0	0.000	0.931	0.004	1.000	0.002	0.931	1.000	0.000
(0.600,0.700]	4	1	0.000	0.931	0.011	0.857	0.006	0.931	1.000	0.000



Score bin ↓	Positive exam...	Negative exam...	Fraction above thresh...	Accuracy	F1 Score	Precisi...	Recall	Negative precisi...	Negative recall	Cumulative AUC
(0.900,1.000]	2	0	0.000	0.931	0.004	1.000	0.002	0.931	1.000	0.000
(0.800,0.900]	9	0	0.001	0.931	0.021	1.000	0.011	0.931	1.000	0.000
(0.700,0.800]	58	0	0.005	0.935	0.124	1.000	0.066	0.935	1.000	0.000
(0.600,0.700]	106	0	0.012	0.942	0.287	1.000	0.168	0.941	1.000	0.000
(0.500,0.600]	213	2	0.026	0.956	0.541	0.995	0.372	0.955	1.000	0.000
(0.400,0.500]	242	19	0.043	0.971	0.743	0.968	0.603	0.971	0.998	0.001
(0.300,0.400]	239	57	0.063	0.983	0.873	0.918	0.832	0.988	0.994	0.004
(0.200,0.300]	102	214	0.084	0.976	0.842	0.769	0.930	0.995	0.979	0.018
(0.100,0.200]	57	1070	0.159	0.908	0.599	0.430	0.985	0.999	0.902	0.092
(0.000,0.100]	16	12594	1.000	0.070	0.130	0.070	1.000	1.000	0.000	0.993

Multi-Class Algorithms

Monday, July 20, 2020 1:56 PM

- Multi-class Algorithms and Hyperparameters
  - Multi-Class Logistic Regression
    - Predict probability of an outcome
    - Hyperparameters to configure
      - Optimization tolerance -> when to stop iterations
        - If improvement between iterations is less than threshold, algorithm stops and returns model
      - Regularization Weight -> penalized models with extreme coefficient values
        - Determines how to penalize in each iteration
  - Multi-Class Neural Network
    - Simple example -> input layer, hidden layer, output layer
    - Hyperparameters
      - Number of hidden nodes -> customize # of hidden nodes
      - Learning rate -> size of step taken at each iteration before correction
      - Number of learning iterations -> max # of times the algorithm should process training cases
  - Multi-Class Decision Forest
    - Ensemble of decision trees
    - Building multiple decision trees then voting on most popular outcome class
    - Hyperparameters
      - Resampling method
        - controls method used to create individual trees
      - # of decision trees
        - max # of decision trees that can be created
      - Maximum depth
        - limit max depth of ANY decision tree
      - # of random splits per node
        - # of splits to used when building each node of the tree
      - Min # of samples per leaf node
        - min. # of cases required to create a terminal node in a tree

Lab: Multi-Class Classifiers Performance

Monday, July 20, 2020 2:35 PM

Lab Overview

Azure Machine Learning designer (preview) gives you a cloud-based interactive, visual workspace that you can use to easily and quickly prep data, train and deploy machine learning models. It supports Azure Machine Learning compute, GPU or CPU. Machine Learning designer also supports publishing models as web services on Azure Kubernetes Service that can easily be consumed by other applications.

In this lab, we will be compare the performance of two different multiclass classification approaches: `Two-Class Support Vector Machine` used with `One-vs-All Multiclass` module vs `Multiclass Decision Forest`. We will apply the two approaches for the letter recognition problem and compare their performance. We will do all of this from the Azure Machine Learning designer without writing a single line of code.

1. The `Two-Class Support Vector Machine` algorithm is extended for multiclass classification problem by using the `One-vs-All Multiclass` module.
2. As you can observe that the native multiclass algorithm `Multiclass Decision Forest` outperforms the `Two-Class Support Vector Machine` across all key performance metrics.
3. One recommendation for next steps is to increase the `Number of iterations` parameter for the `Two-Class Support Vector Machine` module to an higher value like **100** and observe its impact on the performance metrics.

Evaluate Model result visualization

Rows 2 Columns 5

Overall_Accuracy	Micro_Precision	Macro_Precision	Micro_Recall	Macro_Recall
0.699	0.699	0.71433	0.699	0.700227
0.8972	0.8972	0.900638	0.8972	0.89763

Overall_Accuracy

Statistics

Mean	0.7981
Median	0.7981
Min	0.699
Max	0.8972
Standard deviation	0.1401
Unique values	2
Missing values	0
Feature type	Numeric Feature

Visualizations

Lab: Train a Classifier Using Automated Machine Learning

Monday, July 20, 2020 3:24 PM

```
https://introtomlsampledata.blob.core.windows.net/data/crm-churn/crm-churn.csv
```

Lab Overview

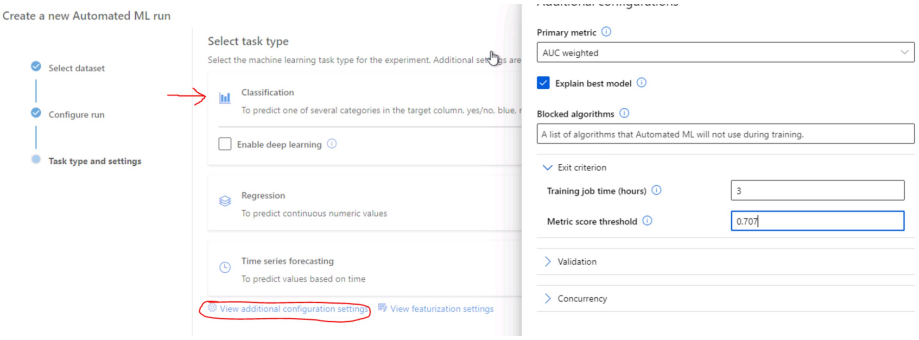
Automated machine learning picks an algorithm and hyperparameters for you and generates a model ready for deployment. There are several options that you can use to configure automated machine learning experiments.

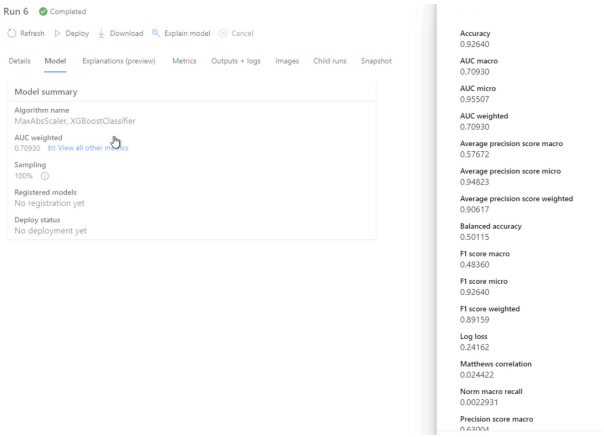
Configuration options available in automated machine learning:

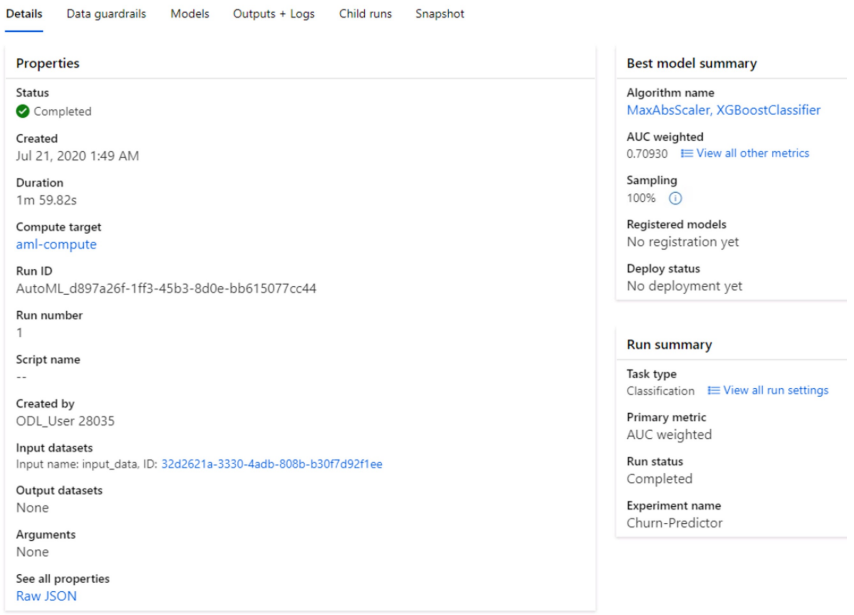
- Select your experiment type: Classification, Regression or Time Series Forecasting
- Data source, formats, and fetch data
- Choose your compute target
- Automated machine learning experiment settings
- Run an automated machine learning experiment
- Explore model metrics
- Register and deploy model

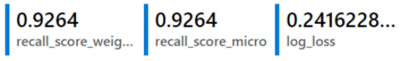
You can create and run automated machine learning experiments in code using the Azure ML Python SDK or if you prefer a no code experience, you can also create your automated machine learning experiments in Azure Machine Learning Studio.

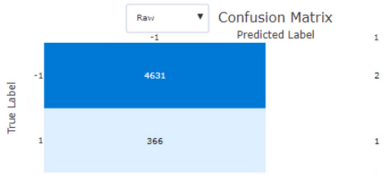
In this lab, we will use Automated Machine Learning to find the best performing binary classification model for predicting customer churn. We will do all of this from the Azure Machine Learning Studio without writing a single line of code.

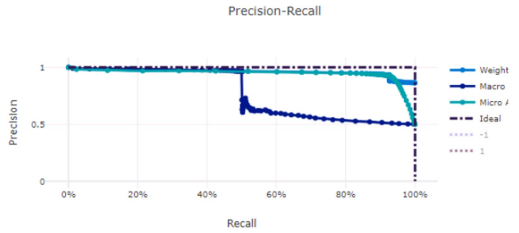


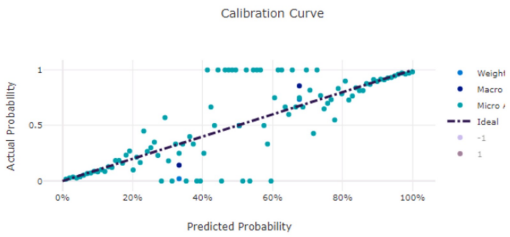


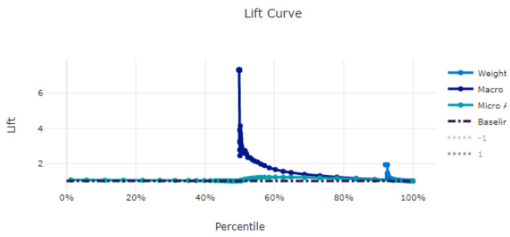


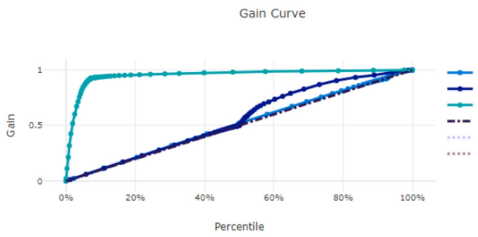












Lab: Regressors Performance

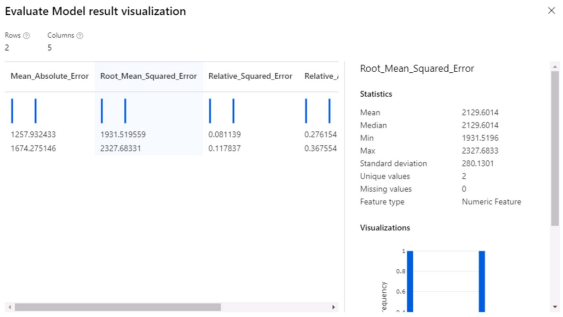
Monday, July 20, 2020 7:15 PM

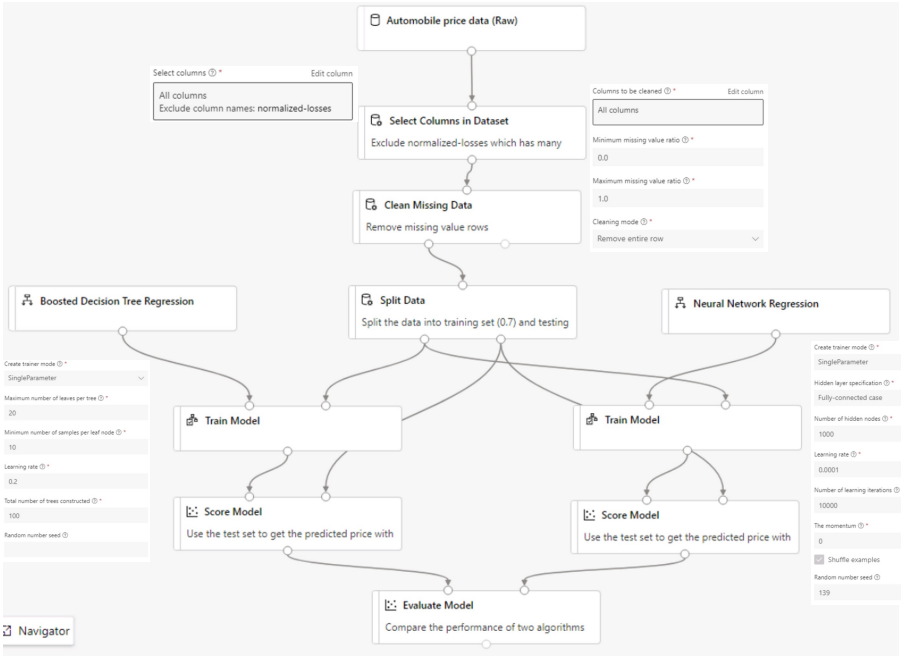
Lab Overview

Azure Machine Learning designer (preview) gives you a cloud-based interactive, visual workspace that you can use to easily and quickly prep data, train and deploy machine learning models. It supports Azure Machine Learning compute, GPU or CPU. Machine Learning designer also supports publishing models as web services on Azure Kubernetes Service that can easily be consumed by other applications.

In this lab, we will be compare the performance of two regression algorithms: `Boosted Decision Tree Regression` and `Neural Net Regression` for predicting automobile prices. We will do all of this from the Azure Machine Learning designer without writing a single line of code.

Based on the performance metric, `Root_Mean_Squared_Error`, it shows that the `Boosted Decision Tree Regression` algorithm outperforms the `Neural Net Regression` algorithm. One recommendation for next steps is to tune the hyperparameters for the `Neural Net Regression` module to see if we can improve its performance.





Lab: Train a Regressor using Automated Machine Learning

Tuesday, July 21, 2020 11:16 PM

Lab Overview

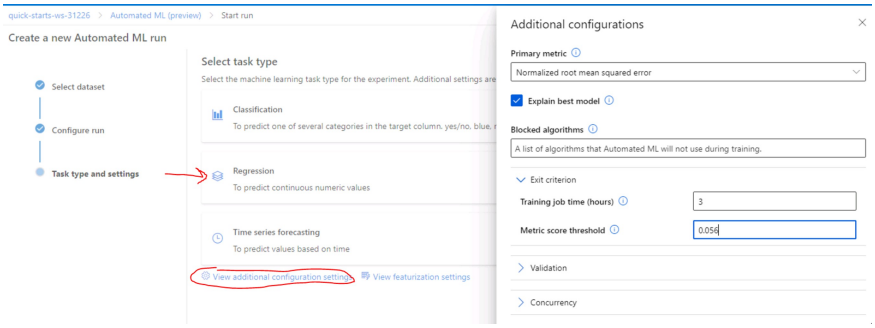
Automated machine learning picks an algorithm and hyperparameters for you and generates a model ready for deployment. There are several options that you can use to configure automated machine learning experiments.

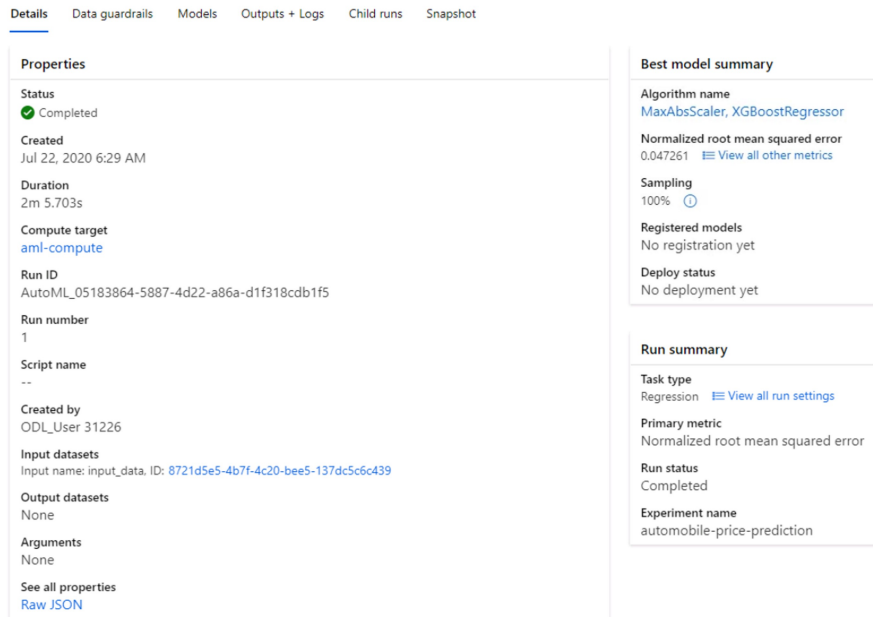
Configuration options available in automated machine learning:

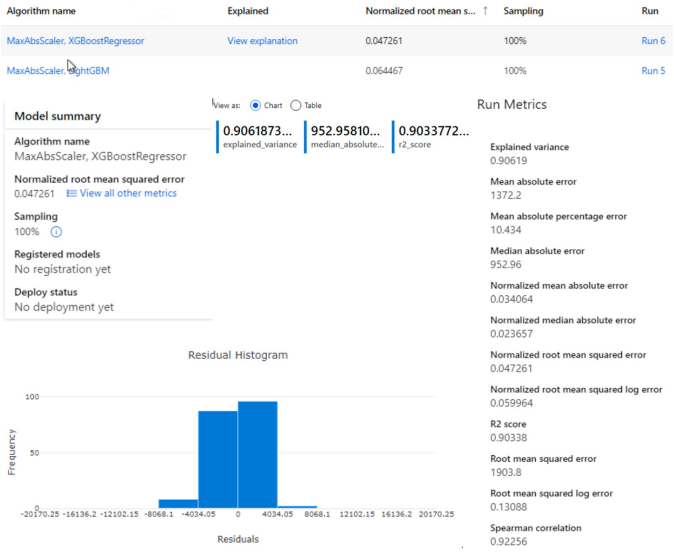
- Select your experiment type: Classification, Regression or Time Series Forecasting
- Data source, formats, and fetch data
- Choose your compute target
- Automated machine learning experiment settings
- Run an automated machine learning experiment
- Explore model metrics
- Register and deploy model

You can create and run automated machine learning experiments in code using the Azure ML Python SDK or if you prefer a no code experience, you can also create your automated machine learning experiments in Azure Machine Learning Studio.

In this lab, we will use Automated Machine Learning to find the best performing regression model for predicting automobile prices. We will do all of this from the Azure Machine Learning Studio without writing a single line of code.







Un/Semi-supervised Learning

Monday, July 20, 2020 6:55 PM

- Unsupervised ML
  - Algorithms train on unlabeled data
  - Without the expected outputs, the algorithm attempts to find, on its own, hidden structures in the data
  - Training process aims to identify common aspects between entities then use them / absence of them to predict results
  - Cost of labeled data is high and sometimes not scalable
  - Unsupervised learning can find:
    - Clustering
    - Association
    - Dimensionality reduction
    - Feature extraction
    - Feature learning
    - Anomaly Detection
    - Neural Networks
    - Principal Component Analysis
    - Matrix Factorization
  - Three Main Types of Approaches
    - Clustering
      - Organizes entities from input data into a finite number of subsets of clusters
      - Ex. Cluster/tag similar documents based on contents of documents
    - Feature Learning (Representation Learning)
      - Transforms sets of inputs into other inputs that are potentially more useful in solving a given problem
      - Ex. Useful in uncovering more and useful features in high level models
    - Anomaly Detection
      - Identifies two major groups of entities
        - Normal
        - Abnormal
      - Ex. Fraud detection
  - Autoencoders are an example of unsupervised learning

- Semi-Supervised Learning
  - **Semi-supervised learning uses data that is partially labeled**
  - Labeled data can be hard or expensive to acquire, unlabeled data is usually much cheaper
  - Combines unsupervised and supervised approaches, typically small amounts of labeled data and large amounts of unlabeled data
    - Use labeled data to determine how to make unlabeled data useful
  - Three Major approaches
    - Self-Training
      - Train using labeled data then use to predict unlabeled data
    - Multi-View Training
      - Train multiple models on different views of the data with different views, include various feature selection, parts of training data, various model architectures
    - Self-Ensemble Training
      - Similar to multi-view but you use single base model but different hyperparameter settings

Clustering

Tuesday, July 21, 2020 11:40 PM

- Clustering
  - Organizing entities from input data into a finite number of subsets or clusters
    - Maximize similarity of entities in the same cluster
    - Maximize differences between cluster groups
  - Applications
    - Personalization and target marketing
      - Group customers based on similar characteristics
    - Document classification
      - Cluster similar documents based on content
    - Fraud Detection
      - Isolate new cases based on proximity with historical clusters of fraudulent behavior
    - Medical Imaging
      - Differentiate between different types of tissue in a 3D image
    - City Planning
      - Identify groups of houses depending on house type, value, geographical location
  - 4 Types of Algorithms
    - Centroid-based Clustering
      - Organizes data into clusters based on proximity to cluster centroids
      - K-Means Clustering
        - Centroid-based, unsupervised
        - Creates target (K) number of clusters, grouping similar members
        - Minimize intra-cluster distances (squared error between members of cluster and its center)
        - Algorithm Process
          - Initialize centroids
          - Cluster assignment
            - Assigns each member to its closest centoid
          - Move centroids
            - Computes new centroids based on current cluster membership
            - Re-centering the centroid so it better represents its members
          - Check for Convergence
            - Convergence criteria: how much do centroid locations change due to new cluster membership?
            - Can redirect back to cluster assignment to better place centroids until exit criteria is met
        - Module Configurations
          - Number of centroids
            - Ideal number of centroids to start with, may end up with less
          - Initialization approach (to select initial centroids)
            - First N
            - Random
            - K-Means ++ Algorithm
          - Distance Metric
            - Euclidian
          - Normalize Features
            - Uses min-max feature to scale data from 0 to 1
          - Assign Label Mode
            - Can only be used if label column exists
            - Can use label values to guide cluster selection
            - Can fill in missing values

        - Can replace label column with predictor values
    - Number of iterations
        - Number of times it should iterate before finalizing centroid selection

- Density-based Clustering
    - Clusters members that are closely packed together
    - Can learn clusters of arbitrary shape
- Distribution-based Clustering
    - Assumes data has natural distributions in it, algorithm groups members based on how they fall into different distributions
- Hierarchical Clustering
    - Builds a tree of clusters
    - Better suited for stuff like taxonomy where there is a hierarchy

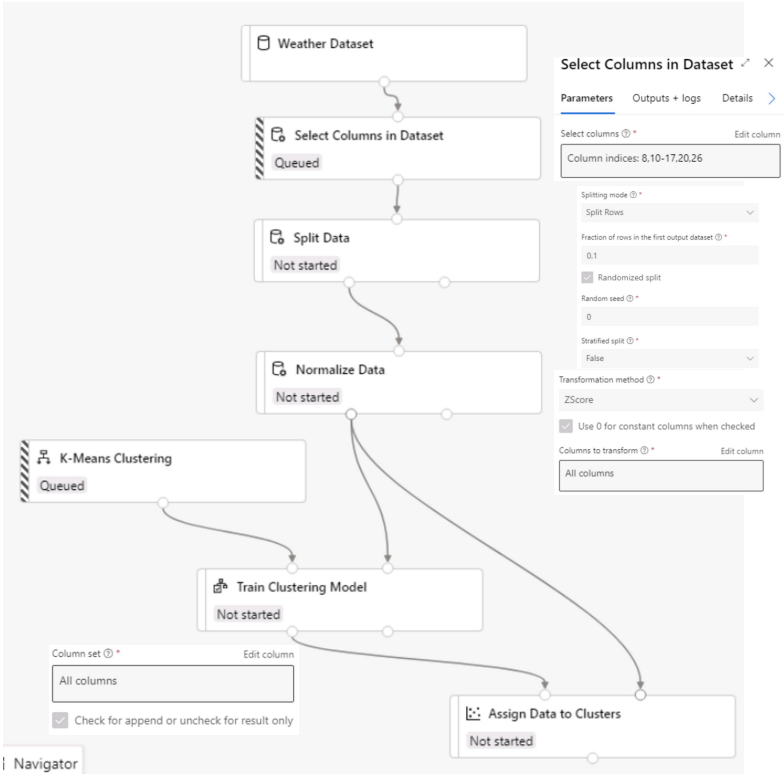
Lab: Simple Clustering Model

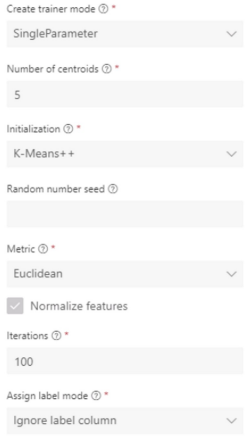
Tuesday, July 21, 2020 11:40 PM

Lab Overview

Azure Machine Learning designer (preview) gives you a cloud-based interactive, visual workspace that you can use to easily and quickly prep data, train and deploy machine learning models. It supports Azure Machine Learning compute, GPU or CPU. Machine Learning designer also supports publishing models as web services on Azure Kubernetes Service that can easily be consumed by other applications.

In this lab, we will be using the `Weather Dataset` that has weather data for 66 different airports in the USA from April to October 2013. We will cluster the dataset into 5 distinct clusters based on key weather metrics, such as visibility, temperature, dew point, wind speed etc. The goal is to group airports with similar weather conditions. We will do all of this from the Azure Machine Learning designer without writing a single line of code.





Assign Data to Clusters result visualization

Rows 28,456 Columns 17

WindSpeed	StationPressure	Altimeter	Assignments	DistancesToClusterCenter no.0	DistancesToClusterCenter no.1	DistancesToClusterCenter no.2	DistancesToClusterCenter no.3
1.017665	0.492638	-0.640816	1	1.043761	0.248115	0.485734	0.738018
-0.505009	0.684469	1.234543	4	0.624923	0.928306	0.595271	1.054183
0.446662	0.617328	0.26453	2	0.67998	0.406787	0.138439	0.743919
-1.456679	-0.284277	-0.382146	1	0.868134	0.31572	0.320039	0.693531
0.256328	0.713243	1.363878	2	0.828748	0.447536	0.17172	0.701901
-0.885677	-0.677531	-1.028821	0	0.310712	0.764818	0.609973	1.05576
0.065994	0.166525	-0.705483	1	0.867219	0.235992	0.567249	0.882484
-0.314674	0.646102	0.393865	1	0.985481	0.145122	0.514127	0.80665
-0.885677	0.502229	-0.317478	2	0.833548	0.429738	0.159848	0.68089
1.017665	-0.399376	0.00586	3	0.989102	0.636432	0.561921	0.224928
1.017665	-0.437742	-1.481494	2	0.754271	0.559745	0.280868	0.72195
NaN	-2.34646	-0.705483	NaN	NaN	NaN	NaN	NaN
-0.885677	0.425497	-0.964154	2	0.915879	0.475018	0.351797	0.384997

Unique values 5
Missing values 98
Feature type Categorical Score

Visualizations