Commercial Building Energy Consumption modeling

1. Problem Statement:

Energy Consumption data is obtained for 78 buildings and the goal is to monitor and reduce energy consumption so that the buildings can be made more efficient. Data science skills have to be applied and models have to be built for prediction, classification and clustering. The goal for this report is to deploy these models as a web service using Azure Machine learning studio and build a user interface to invoke the web service.

2. Build Web Services:

A. Regression model:
   a) Linear Regression
   b) Decision Forest
   c) Decision Tree
   d) Neural Network

B. Classification model:
   a) Logistic Regression
   b) Decision Forest
   c) KNN
   d) Neural Network

C. Clustering model:
   a) K-means
   b) Hierarchical

Steps to create a Web Service:

Prerequisite: Train and evaluate the models

Step 1: Save and run the model trained

Step 2: Click on Set up Web Service.

Step 3: Web service input and Web service output modules are added.

Step 4: Define where the Web service will accept input and where it generates the output

Step 5: Click on deploy web service below the canvas and select deploy web service (classic).

Machine learning studio deploys the experiment as a web service and takes to the dashboard for that web service.

Assignment 3 - To build out a web service for the models

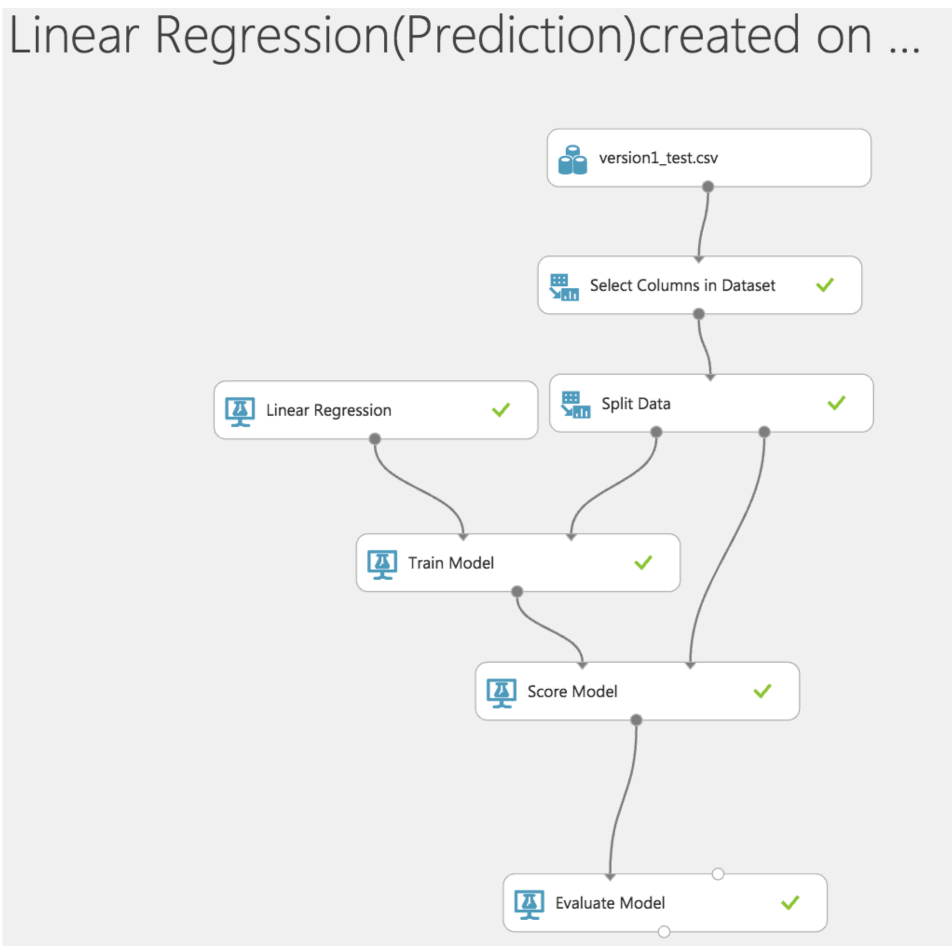
Regression Model – Linear Regression

Regression is a machine learning used to predict a numeric outcome. Linear regression attempts to establish a linear relationship between independent variables and an outcome variable, or dependent variable that is also numeric. The trained model can then be used to make predictions.

<u>Advantages/Limitations of linear model</u>:

- ➔ Linear regression implements a statistical model that, when relationships between the independent variables and the dependent variable are almost linear, shows optimal results.
- ➔ Linear regression is often inappropriately used to model non-linear relationships.
- ➔ Linear regression is limited to predicting numeric output.

Training a model:



Assignment 3 - To build out a web service for the models

Properties of linear Regression:

Linear Regression

Solution method

Ordinary Least Squares

L2 regularization weight

0.001

☑ Include intercept term

Prediction:

Linear Regression(Prediction)created on 2016/... › Score Model › Scored dataset

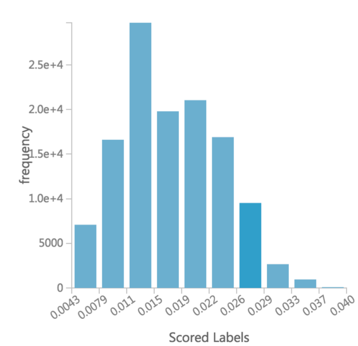
rows 124409

columns 9

month	Normalized_Consumption	TemperatureF	Dew_PointF	Humidity	WindDirDegrees	Scored Labels
7	0.001061	55.4	53.6	94	0	0.012345
2	0.01648	28.4	23	80	300	0.020089
4	0.039702	46.4	26.6	46	130	0.021222
6	0.0044	53.6	48.2	82	320	0.015332
1	0.016863	17.6	14	86	360	0.02506
9	0.006957	60.8	50	68	110	0.008553
6	0.00268	50	44.6	82	320	0.016506
2	0.001358	24.8	19.4	80	350	0.026837
5	0.003268	71.6	55.4	57	90	0.011323
8	0.033824	57.2	57.2	100	160	0.007066
5	0.018341	50	33.8	54	180	0.019092
10	0.022806	46.4	44.6	93	250	0.015773

Scored Labels

Histogram



Performance metrics:

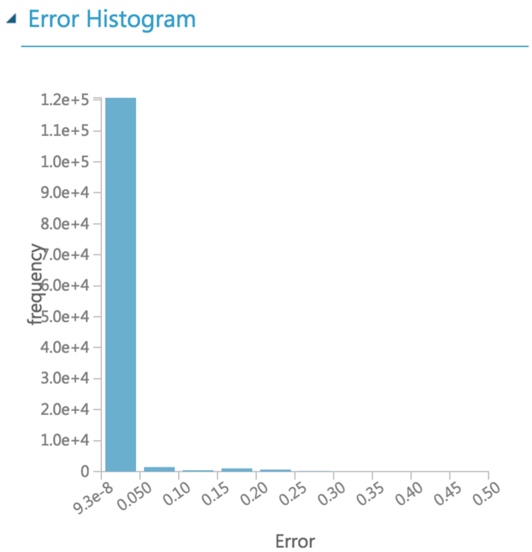
Metrics

Mean Absolute Error	0.01445
Root Mean Squared Error	0.030205
Relative Absolute Error	0.939439
Relative Squared Error	0.956155
Coefficient of Determination	0.043845

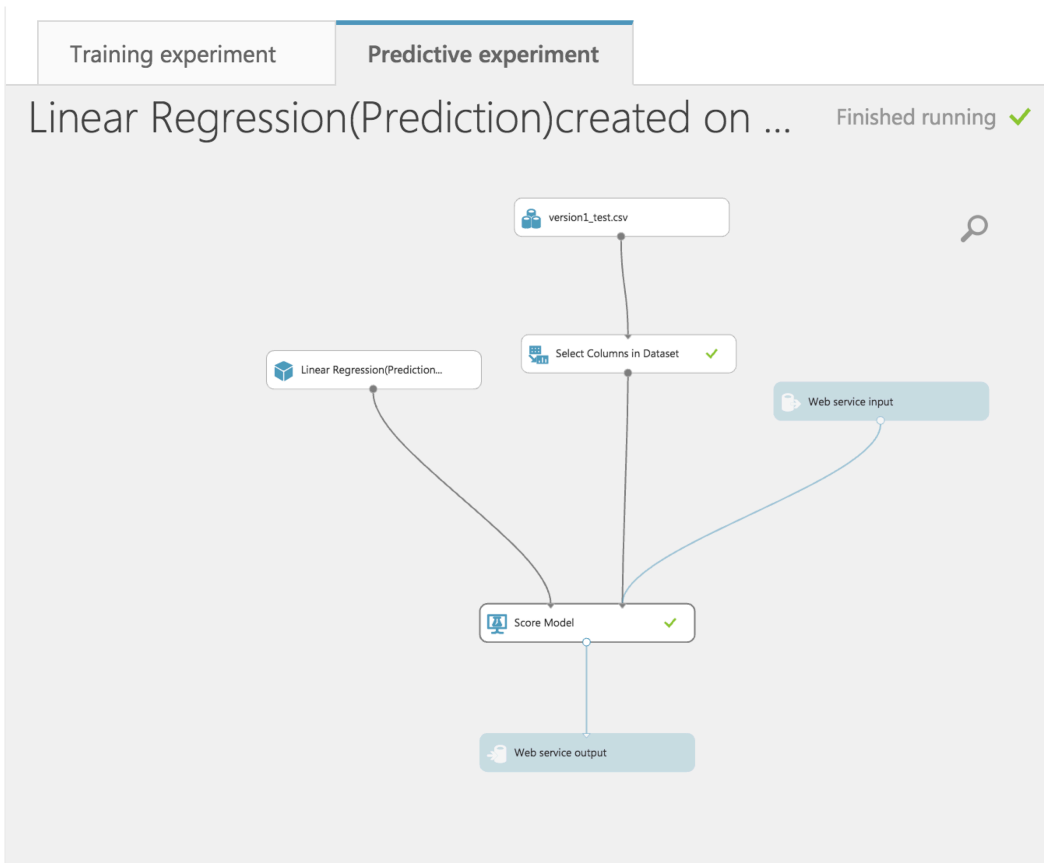
The below graph can be used to assess the quality of regression.

The term "error" here represents the difference between the predicted value and the true value. The error metrics measure the predictive performance of a regression model in terms of the mean deviation of its predictions from the true values. Lower error values mean the model is more accurate in making predictions. The coefficient of determination, which is also known as R squared, is also a standard way of measuring how well the model fits the data.

Assignment 3 - To build out a web service for the models



Setting up a web service and deploying it:



Regression Model – Decision Forest

Assignment 3 - To build out a web service for the models

Decision Forest Regression module to create a regression model using an ensemble of decision trees. Decision trees are non-parametric models that perform a sequence of simple tests for each instance, traversing a binary tree data structure until a leaf node (decision) is reached.

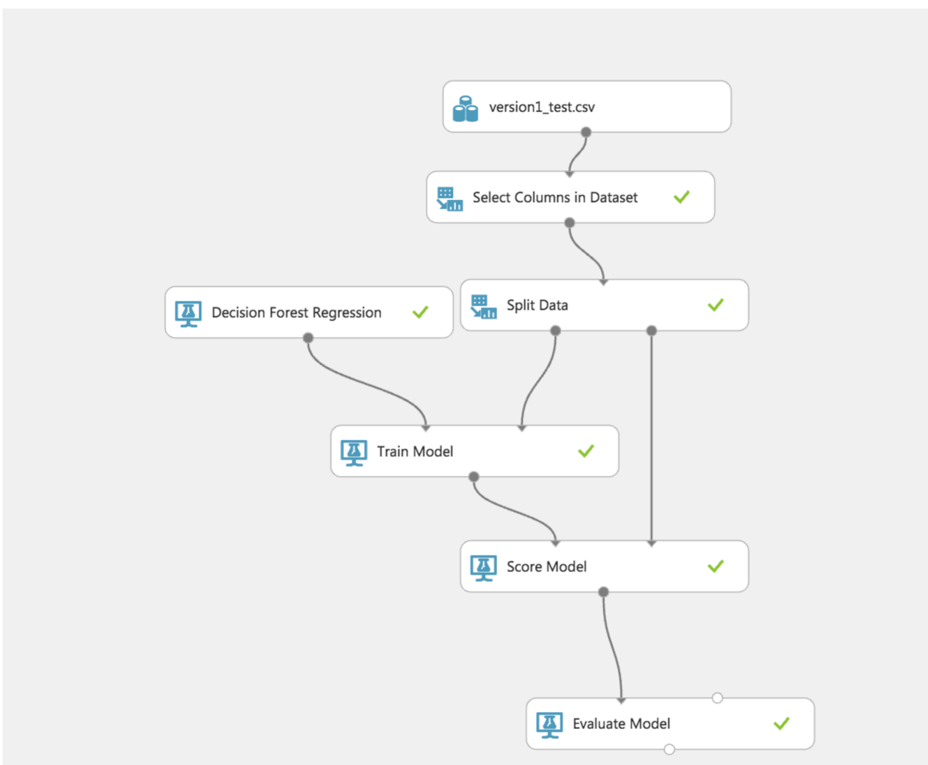
Advantages:

- ➔ They are efficient in both computation and memory usage during training and prediction.
- ➔ They can represent non-linear decision boundaries.
- ➔ They perform integrated feature selection and classification and are resilient in the presence of noisy features.

Disadvantage:

- ➔ Random forests have been observed to overfit for some datasets with noisy classification/regression tasks.

Trained Model:



Properties:

Bagging: Bagging is also called *bootstrap aggregating*. Each tree in a regression decision forest outputs a Gaussian distribution by way of prediction. The aggregation is to find a Gaussian whose first two moments match the moments of the mixture of Gaussians given by combining all Gaussians returned by individual trees.

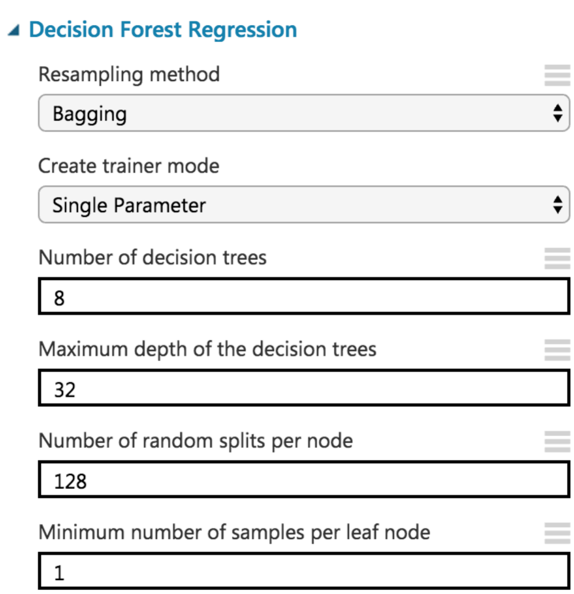
Number of decision trees, indicate the total number of decision trees to create in the ensemble. By creating more decision trees, you can potentially get better coverage, but training time will increase.

For Maximum depth of the decision trees, type a number to limit the maximum depth of any decision tree. Increasing the depth of the tree might increase precision, at the risk of some overfitting and increased training time.

For Number of random splits per node, type the number of splits to use when building each node of the tree. A *split* means that features in each level of the tree (node) are randomly divided.

For Minimum number of samples per leaf node, indicate the minimum number of cases that are required to create any terminal node (leaf) in a tree.

By increasing this value, you increase the threshold for creating new rules.



Assignment 3 - To build out a web service for the models

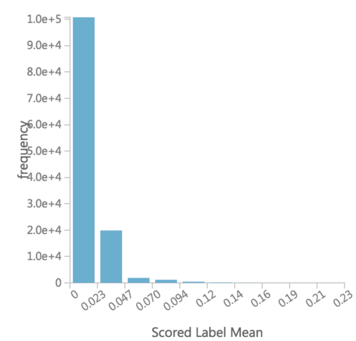
Prediction:

Decision Forest(Prediction)created on 2016/12... › Score Model › Scored dataset

rows 124409 columns 10

Normalized_Consumption	TemperatureF	Dew_PointF	Humidity	WindDirDegrees	Scored Label Mean	Scored Label Standard Deviation
0.001061	55.4	53.6	94	0	0.004981	0.005539
0.01648	28.4	23	80	300	0.021173	0.028643
0.039702	46.4	26.6	46	130	0.021029	0.015603
0.0044	53.6	48.2	82	320	0.009776	0.010232
0.016863	17.6	14	86	360	0.024368	0.035778
0.006957	60.8	50	68	110	0.007708	0.009456
0.00268	50	44.6	82	320	0.008888	0.008868
0.001358	24.8	19.4	80	350	0.189131	0.199703
0.003268	71.6	55.4	57	90	0.013693	0.015556
0.033824	57.2	57.2	100	160	0.006207	0.005273
0.018341	50	33.8	54	180	0.018052	0.019832

Scored Label Mean
Histogram



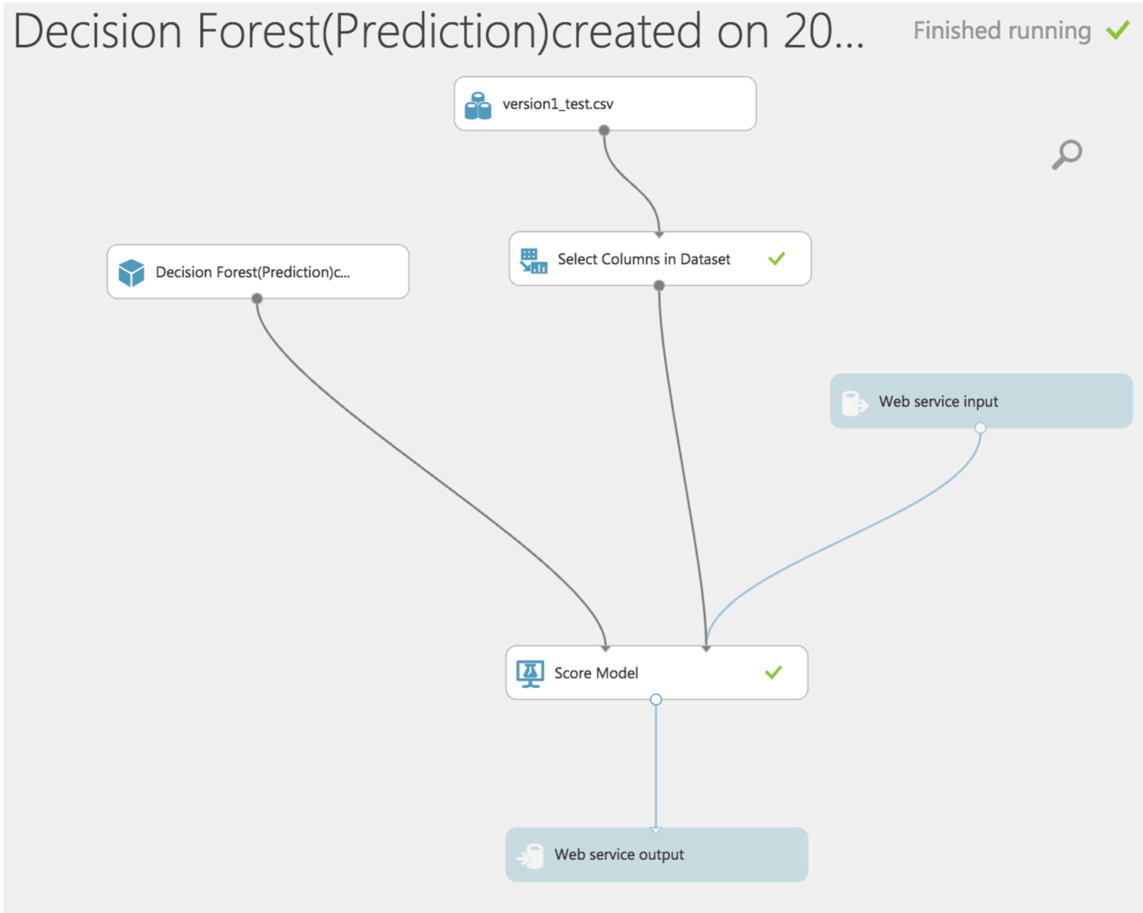
Performance Matrix:

view as

Negative Log Likelihood	Mean Absolute Error	Root Mean Squared Error	Relative Absolute Error	Relative Squared Error	Coefficient of Determination
2461026.888498	0.015397	0.033193	1.000958	1.154679	-0.154679

Setting up a web service and deploying it:

Assignment 3 - To build out a web service for the models



Regression Model – Decision Tree

Boosted Decision Tree Regression module to create an ensemble of regression trees using boosting. Boosting means that each tree is dependent on prior trees, and learns by fitting the residual of the trees that preceded it. Thus, boosting in a decision tree ensemble tends to improve accuracy with some small risk of less coverage.

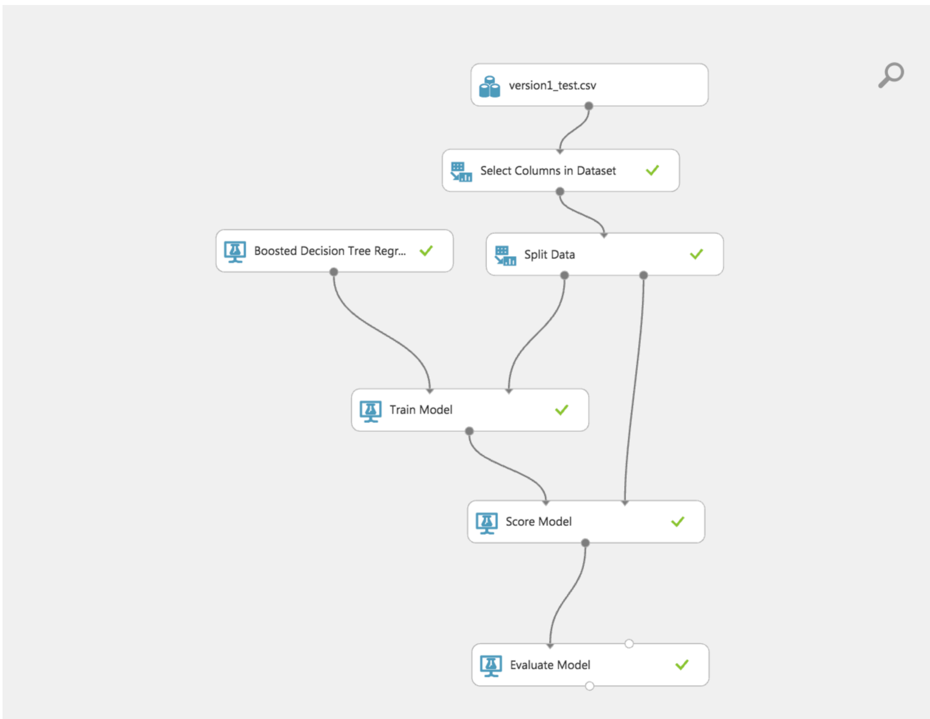
Advantages:

- ➔ Decision trees implicitly perform variable screening or feature selection
- ➔ Decision trees require relatively little effort from users for data preparation
- ➔ Nonlinear relationships between parameters do not affect tree performance
- ➔ The best feature of using trees for analytics - easy to interpret and explain

Disadvantages:

- ➔ They can be extremely sensitive to small perturbations in the data: a slight change can result in a drastically different tree.
- ➔ They can easily overfit.
- ➔ They can have problems out-of-sample prediction (this is related to them being non-smooth).

Trained model:



Properties:

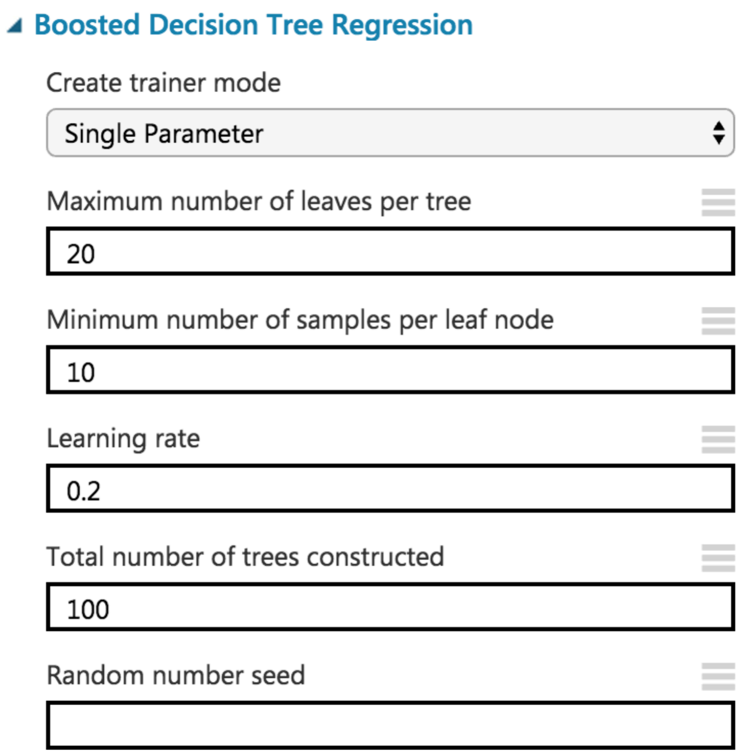
Maximum number of leaves per tree, indicate the maximum number of terminal nodes (leaves) that can be created in any tree.
By increasing this value, you potentially increase the size of the tree and get better precision, at the risk of overfitting and longer training time.

For **Minimum number of samples per leaf node**, indicate the minimum number of cases required to create any terminal node (leaf) in a tree.
By increasing this value, you increase the threshold for creating new rules.

For **Learning rate**, type a number between 0 and 1 that defines the step size while learning.

The learning rate determines how fast or slow the learner converges on the optimal solution.

For **Number of trees constructed**, indicate the total number of decision trees to create in the ensemble. By creating more decision trees, you can potentially get better coverage, but training time will increase.



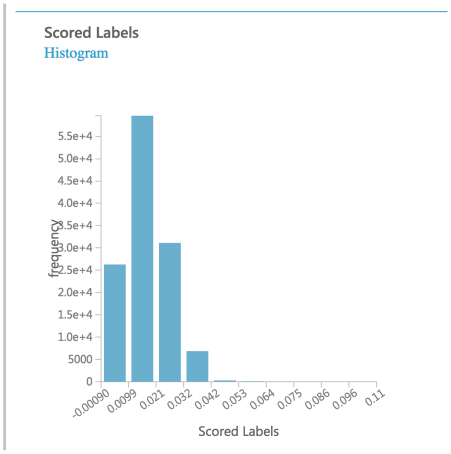
Prediction:

Assignment 3 - To build out a web service for the models

Decision Tree(Prediction)created on 2016/12/1 › Score Model › Scored dataset

rows	columns
124409	9

month	Normalized_Consumption	TemperatureF	Dew_PointF	Humidity	WindDirDegrees	Scored Labels
7	0.001061	55.4	53.6	94	0	0.007814
2	0.01648	28.4	23	80	300	0.020629
4	0.039702	46.4	26.6	46	130	0.02085
6	0.0044	53.6	48.2	82	320	0.012123
1	0.016863	17.6	14	86	360	0.024516
9	0.006957	60.8	50	68	110	0.008712
6	0.00268	50	44.6	82	320	0.013531
2	0.001358	24.8	19.4	80	350	0.045114
5	0.003268	71.6	55.4	57	90	0.015177
8	0.033824	57.2	57.2	100	160	0.007644
5	0.018341	50	33.8	54	180	0.019749
10	0.022806	46.4	44.6	93	250	0.019324

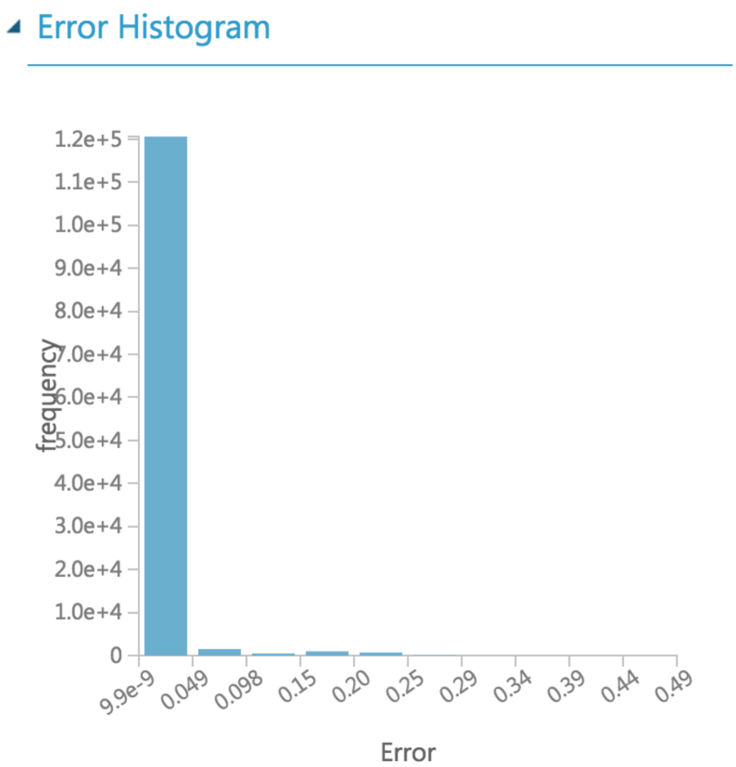


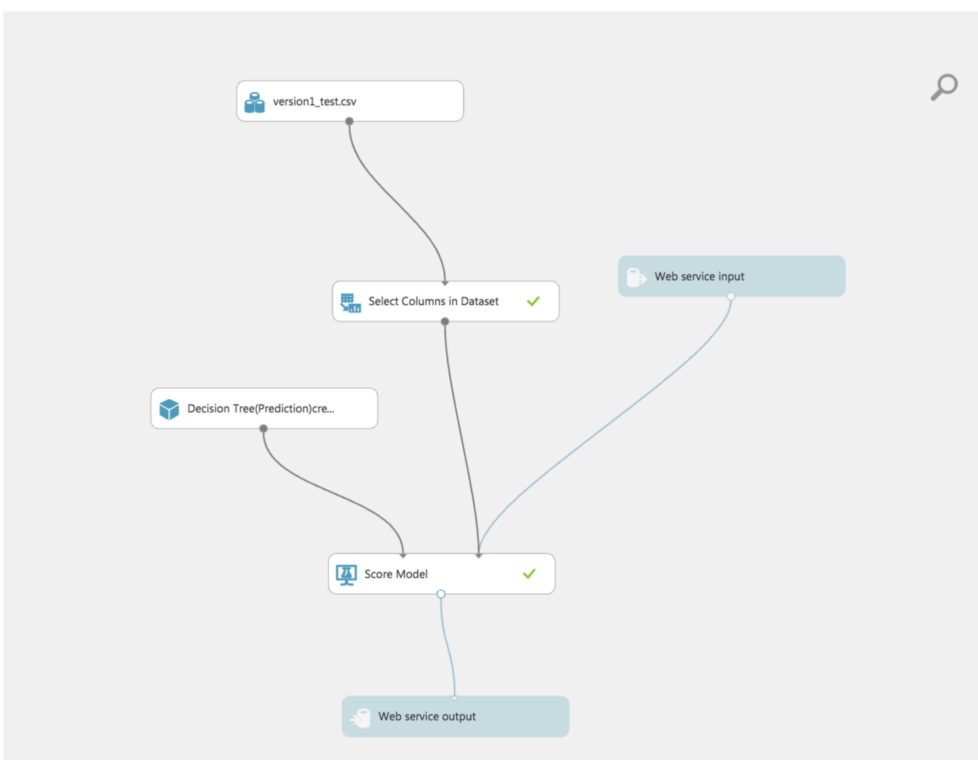
Performance Matrix:

Metrics

Mean Absolute Error	0.014053
Root Mean Squared Error	0.029949
Relative Absolute Error	0.913587
Relative Squared Error	0.939987
Coefficient of Determination	0.060013

Assignment 3 - To build out a web service for the models





Team 12 – Priti Agarwal, Yalin Li, Zhi Li

Assignment 3 - To build out a web service for the models

Regression Model – Neural Network

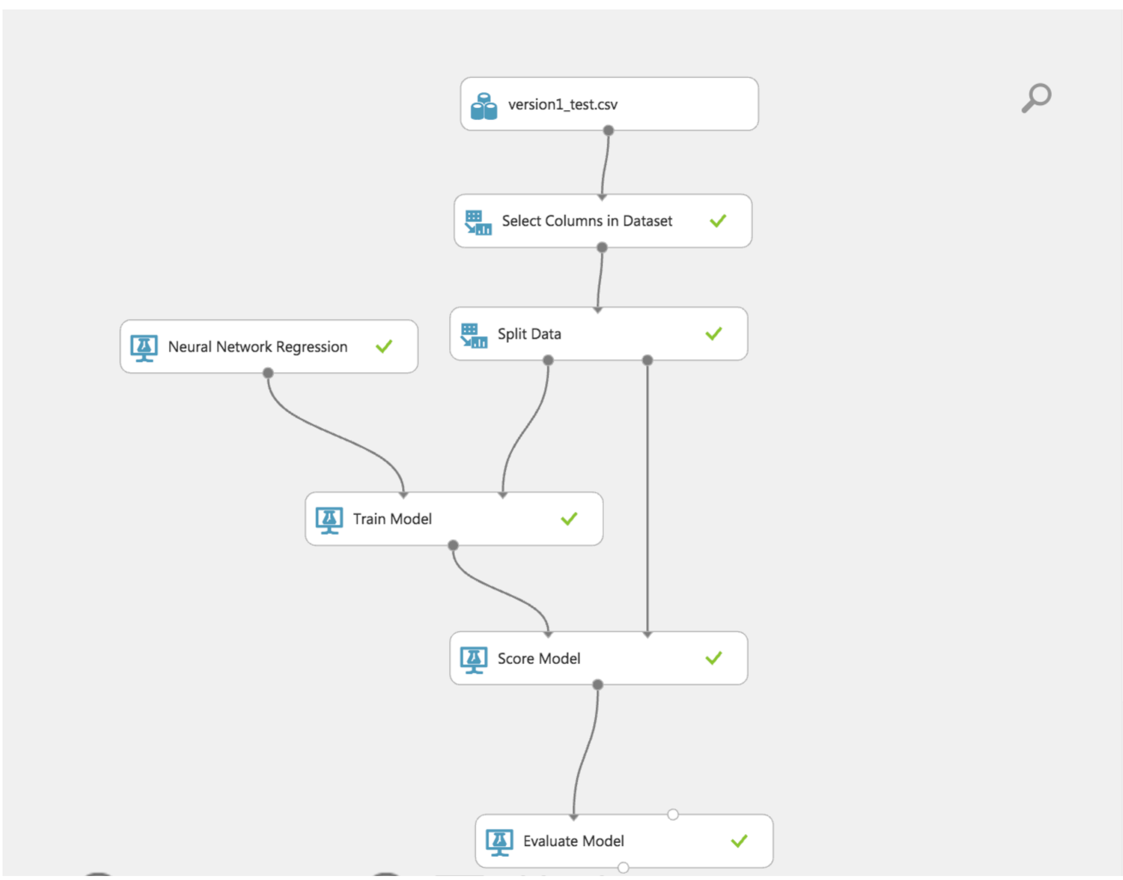
Neural Network Regression module used to create a regression model using a customizable neural network algorithm.
Neural network regression is a supervised learning method, and therefore requires a *tagged dataset*, which includes a label column. Because a regression model predicts a numerical value, the label column must be a numerical data type.

Advantages:

- Easy to conceptualize
- Lots of libraries /implementations available

Disadvantages:

- There are alternatives that are simpler, faster, easier to train, and provide better performance (svm, decision trees, regression)
- Multi-layer neural networks are usually hard to train, and require tuning lots of parameters

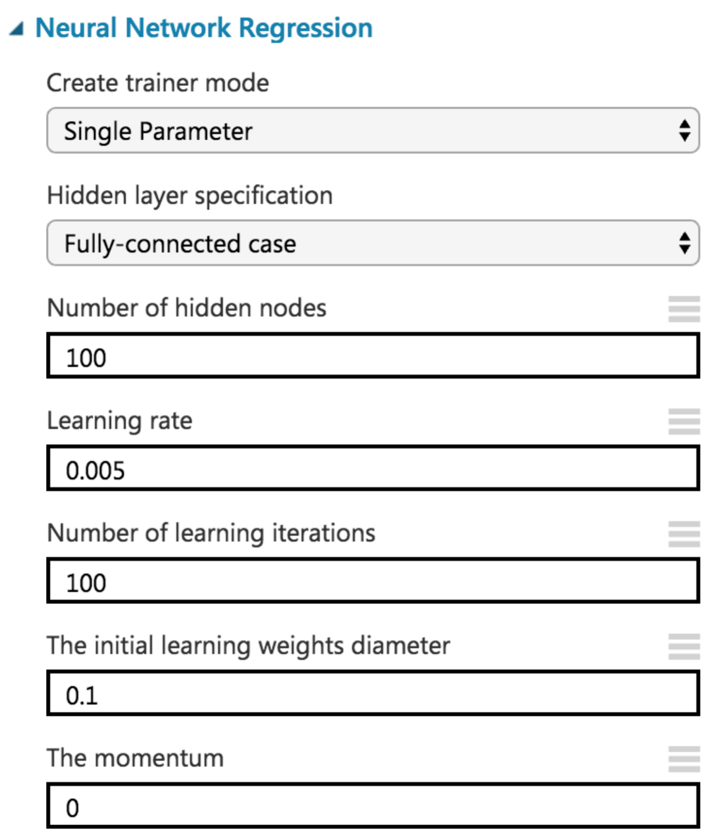


Properties:

Fully connected case: This option creates a model using the default neural network architecture, which for a neural network regression model, has these attributes:

- ➔ one hidden layer
- ➔ output layer is fully connected to the hidden layer and hidden layer is fully connected to the input layer
- ➔ no. of node: 100

Learning rate: A larger value for learning rate can cause the model to converge faster, but it can overshoot local minima.
Number of learning iterations, specify the maximum number of times the algorithm processes the training cases.
For **The initial learning weights diameter** , type a value that determines the node weights at the start of the learning process.
For **The momentum**, type a value to apply during learning as a weight on nodes from previous iterations.



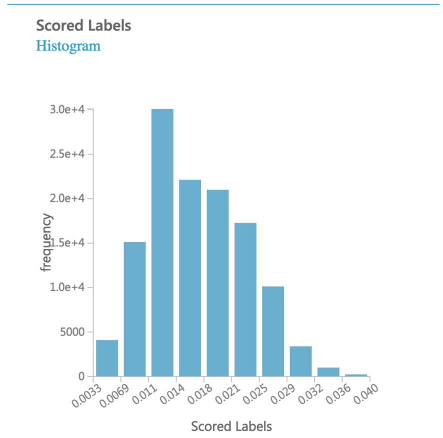
Assignment 3 - To build out a web service for the models

Prediction:

Neural Network (Prediction)created on 2016/1... › Score Model › Scored dataset

rows	columns
124409	13

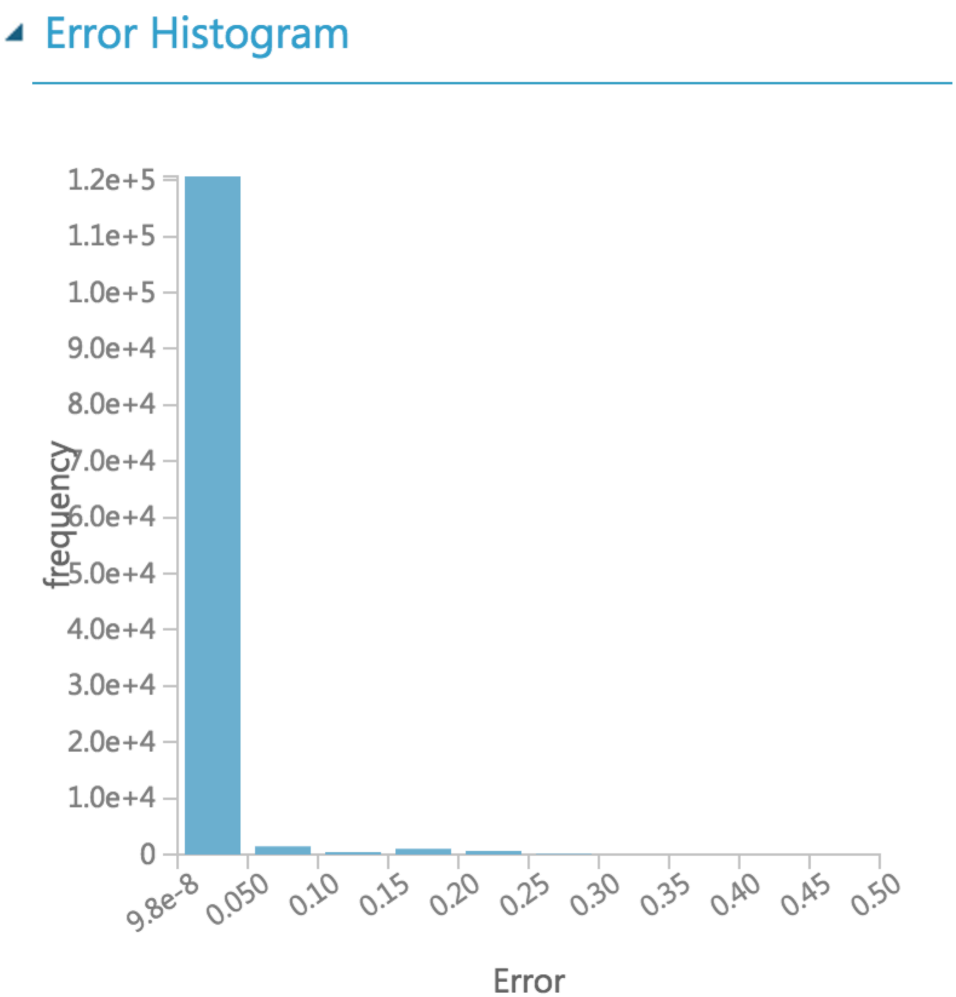
TemperatureF	Dew_PointF	Humidity	Sea_Level_PressureIn	VisibilityMPH	WindDirDegrees	Scored Labels
55.4	53.6	94	30.06	3.1	0	0.011747
28.4	23	80	29.53	6.2	300	0.020955
46.4	26.6	46	30.15	6.2	130	0.02104
53.6	48.2	82	29.95	3.1	320	0.014647
17.6	14	86	29.8	6.2	360	0.025827
60.8	50	68	29.89	6.2	110	0.008665
50	44.6	82	30.09	6.2	320	0.015956
24.8	19.4	80	30.33	6.2	350	0.025776
71.6	55.4	57	29.89	6.2	90	0.011436
57.2	57.2	100	30.09	6.2	160	0.007065
50	33.8	54	30.27	6.2	180	0.018651
46.4	44.6	93	29.47	6.2	250	0.015723



Performance Metrics:

Metrics

Mean Absolute Error	0.014339
Root Mean Squared Error	0.030207
Relative Absolute Error	0.932223
Relative Squared Error	0.956274
Coefficient of Determination	0.043726


Error Histogram
1.2e+5
1.1e+5
1.0e+5
9.0e+4
8.0e+4
7.0e+4
6.0e+4
5.0e+4
4.0e+4
3.0e+4
2.0e+4
1.0e+4
0
frequency
9.8e-8
0.050
0.10
0.15
0.20
0.25
0.30
0.35
0.40
0.45
0.50
Error

Assignment 3 - To build out a web service for the models

Comparison :

Model	Performance Metrics
a) **Linear Regression**	Mean Absolute Error 0.01445 Root Mean Squared Error 0.030205 Relative Absolute Error 0.939439 Relative Squared Error 0.956155 Coefficient of Determination 0.043845
b) **Decision Forest**	Mean Absolute Error 0.015397 Root Mean Squared Error 0.033193 Relative Absolute Error 1.000958 Relative Squared Error 1.154679 Coefficient of Determination -0.154679
c) **Decision Tree**	Mean Absolute Error 0.014053 Root Mean Squared Error 0.029949 Relative Absolute Error 0.913587 Relative Squared Error 0.939987 Coefficient of Determination 0.060013
d) **Neural Network**	Mean Absolute Error 0.014339 Root Mean Squared Error 0.030207 Relative Absolute Error 0.932223 Relative Squared Error 0.956274 Coefficient of Determination 0.043726

Result: The model with least root mean squared error and mean absolute error is Decision Tree and it has the highest coefficient of determination, 60

Classification: Classification algorithms predict the class or category for a single instance of data.

Classification Model – Logistic Regression

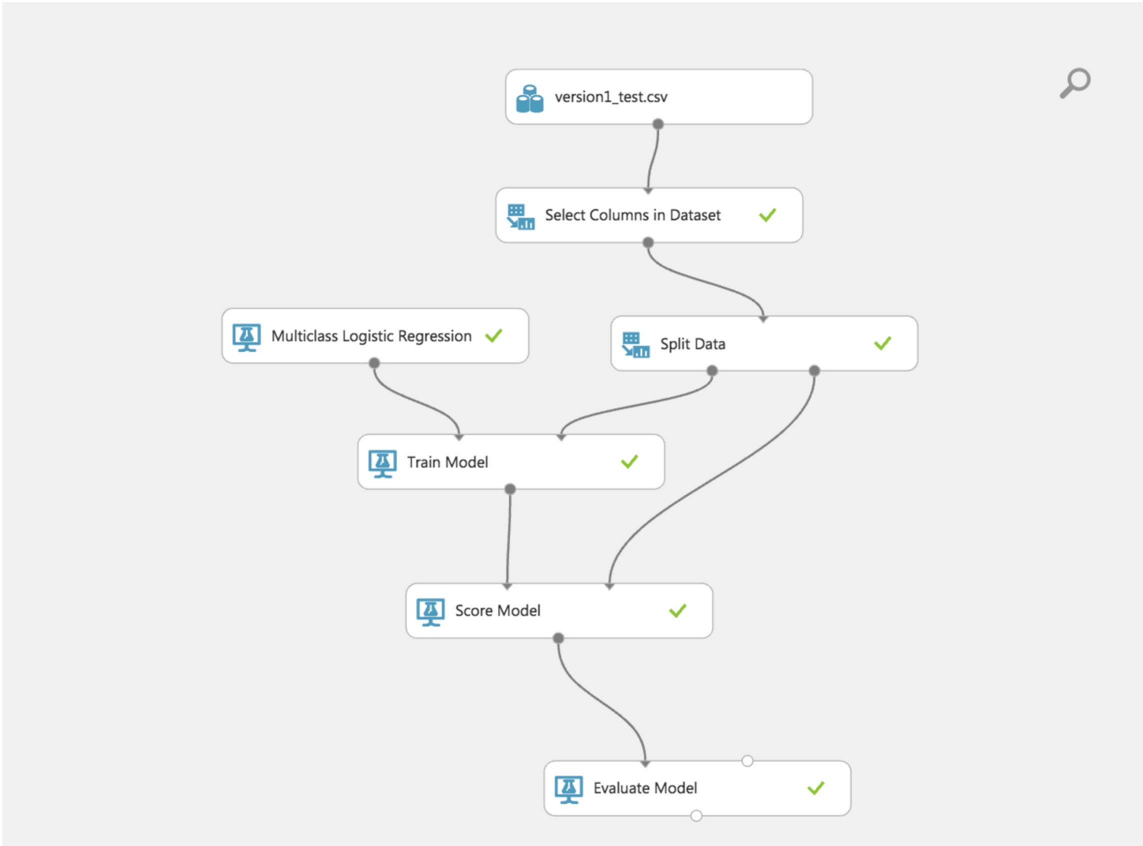
Logistic regression is a well-known method in statistics that is used to predict the probability of an outcome, and is particularly popular for classification tasks. The algorithm predicts the probability of occurrence of an event by fitting data to a logistic function.

Advantages/Disadvantages:

- ➔ Logistic regression will work better if there's a single decision boundary, not necessarily parallel to the axis.
- ➔ Logistic regression is intrinsically simple, it has low variance and so is less prone to over-fitting.
- ➔ requires large sample size to achieve stable results.

Trained model:

Assignment 3 - To build out a web service for the models



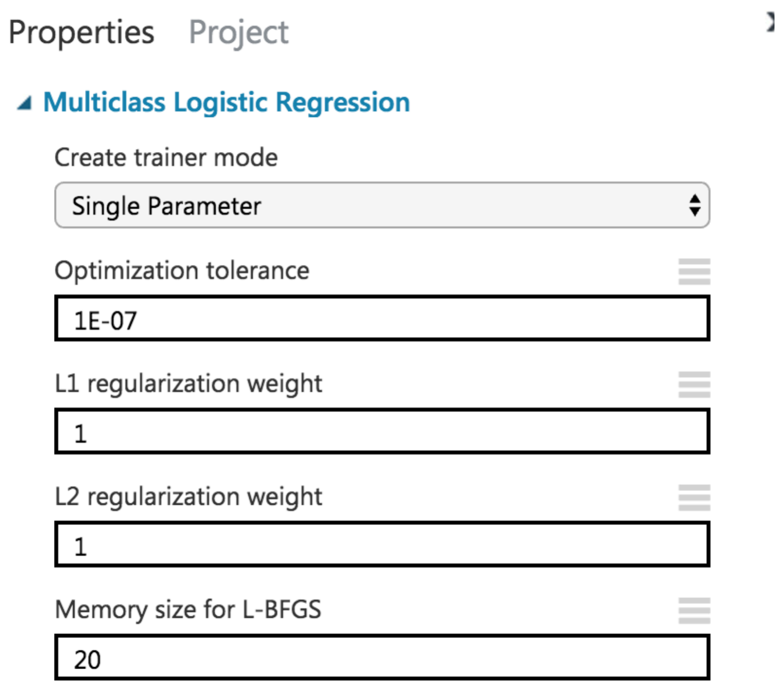
Properties:

For **Optimization tolerance**, specify the threshold value for optimizer convergence. If the improvement between iterations is less than the threshold, the algorithm stops and returns the current model.

L1 regularization weight: Regularization is a method for preventing overfitting by penalizing models with extreme coefficient values. Regularization works by adding the penalty that is associated with coefficient values to the error of the hypothesis. An accurate model with extreme coefficient values would be penalized more, but a less accurate model with more conservative values would be penalized less. L1 can be applied to sparse models, which is useful when working with high-dimensional data.

Memory size for L-BFGS: specify the amount of memory to use for L-BFGS optimization. This parameter indicates the number of past positions and gradients to store for the computation of the next step.

Assignment 3 - To build out a web service for the models



Classification Results:

ratureF	Dew_PointF	Humidity	WindDirDegrees	Base_Hour_Class	Scored Probabilities for Class "High"	Scored Probabilities for Class "Low"	Scored Labels
	53.6	94	0	High	0.582434	0.417566	High
	23	80	300	Low	0.444583	0.555417	Low
	26.6	46	130	High	0.601383	0.398617	High
	48.2	82	320	High	0.59981	0.40019	High
	14	86	360	High	0.5388	0.4612	High
	50	68	110	Low	0.382244	0.617756	Low
	44.6	82	320	High	0.606656	0.393344	High
	19.4	80	350	Low	0.693707	0.306293	High
	55.4	57	90	High	0.53602	0.46398	High
	57.2	100	160	Low	0.380866	0.619134	Low
	33.8	54	180	High	0.599717	0.400283	High

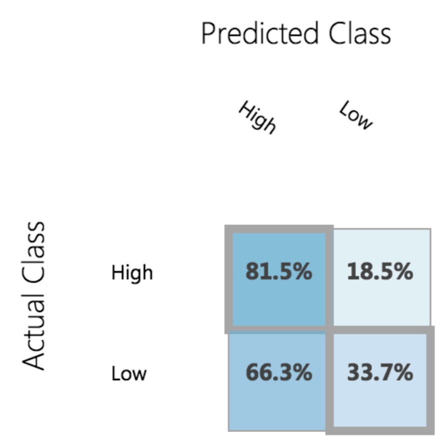
Performance Metric and Confusion matrix:

Assignment 3 - To build out a web service for the models

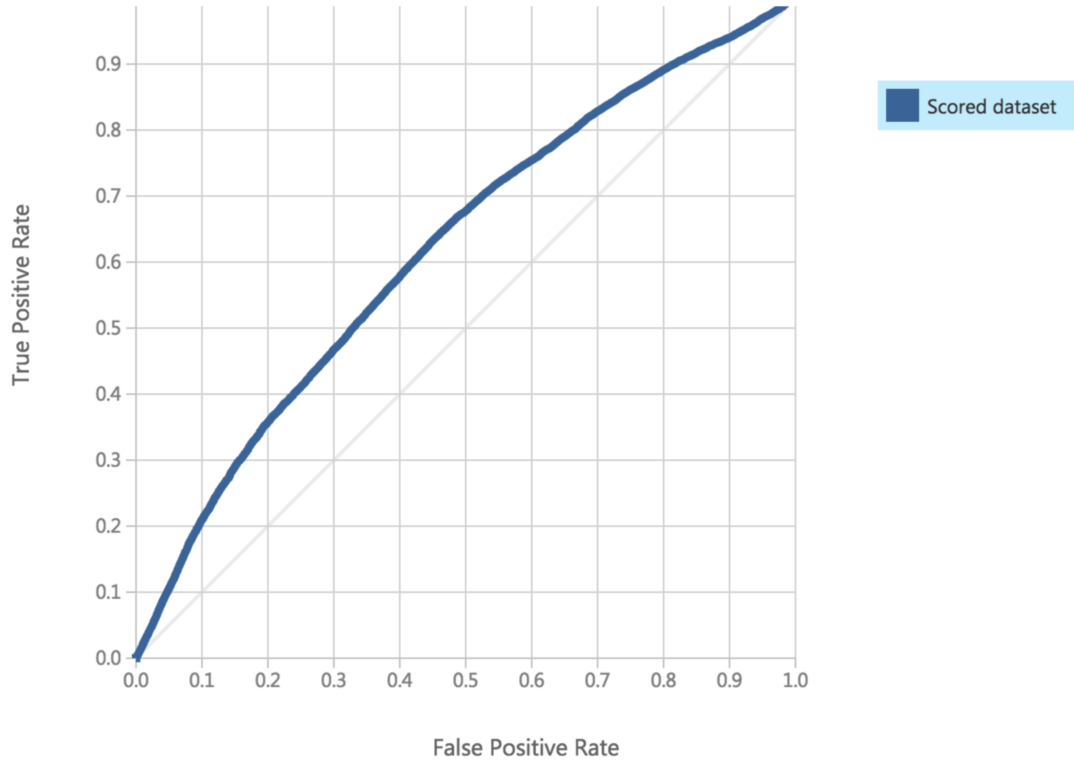
Metrics

Overall accuracy	0.609289
Average accuracy	0.609289
Micro-averaged precision	0.609289
Macro-averaged precision	0.599055
Micro-averaged recall	0.609289
Macro-averaged recall	0.575892

Confusion Matrix



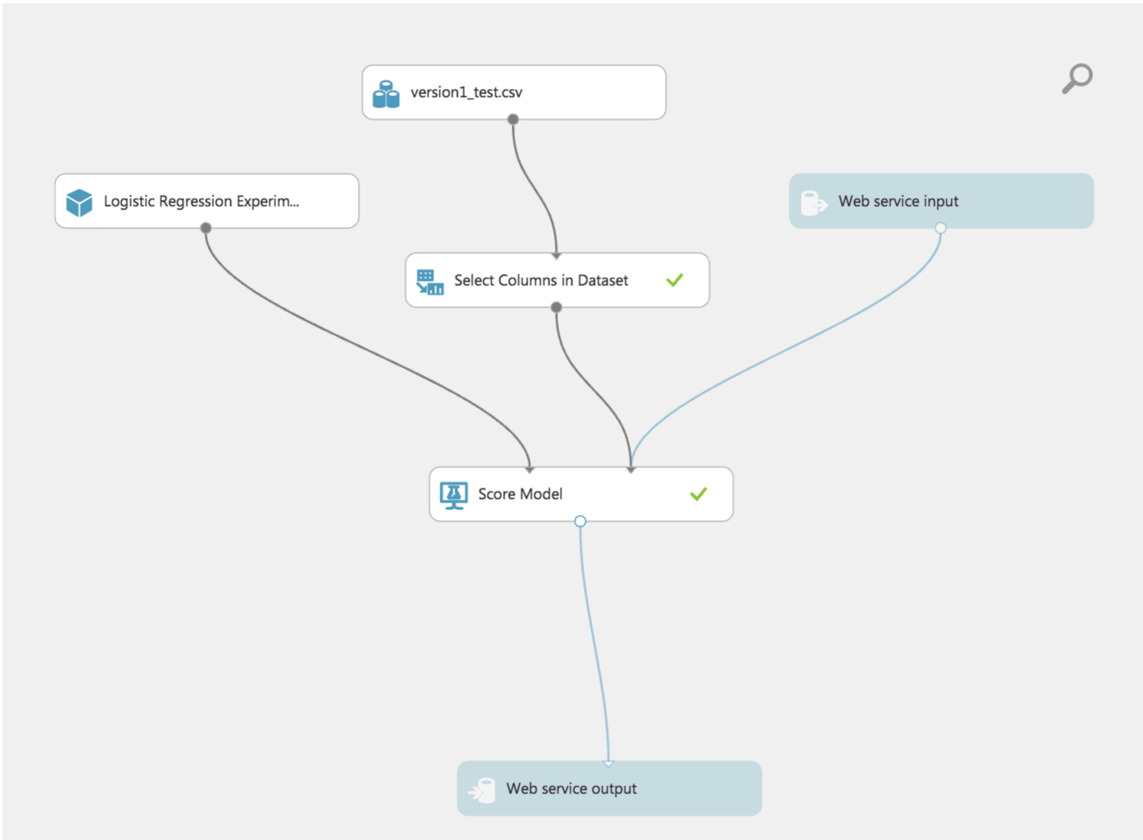
ROC curve:



Accuracy	Precision	Threshold	AUC
0.610	0.579	0.5	0.621

Web-Service:

Assignment 3 - To build out a web service for the models



Team 12 – Priti Agarwal, Yalin Li, Zhi Li

Assignment 3 - To build out a web service for the models

Classification Model – Decision Forest

The decision forest algorithm is an ensemble learning method for classification. The algorithm works by building multiple decision trees and then voting on the most popular output class. The aggregation process sums these histograms and normalizes the result to get the “probabilities” for each label. The trees that have high prediction confidence will have a greater weight in the final decision of the ensemble.

Advantages:

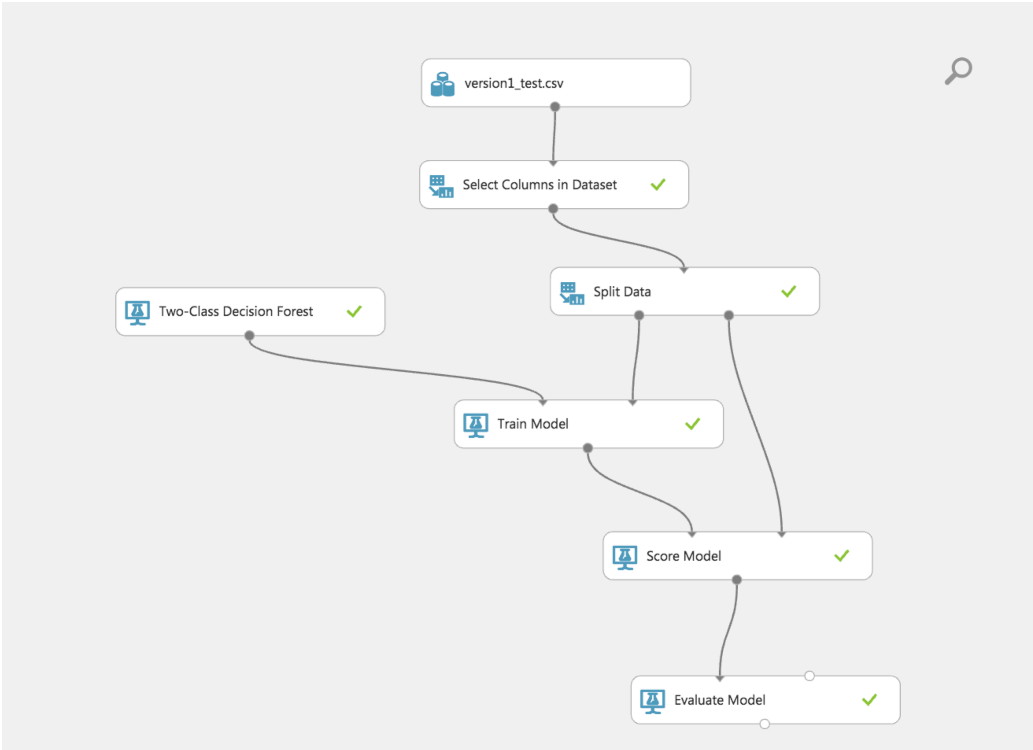
- ➔ They can represent non-linear decision boundaries.
- ➔ They are efficient in computation and memory usage during training and prediction.
- ➔ They perform integrated feature selection and classification.
- ➔ They are resilient in the presence of noisy features.

Disadvantages:

- ➔ May overfit data
- ➔ May get stuck in local minima so need ensembles to help reduce the variance

Trained Model:

Assignment 3 - To build out a web service for the models



Properties: Explained in Regression notes

Two-Class Decision Forest

Resampling method: Bagging

Create trainer mode: Single Parameter

Number of decision trees: 8

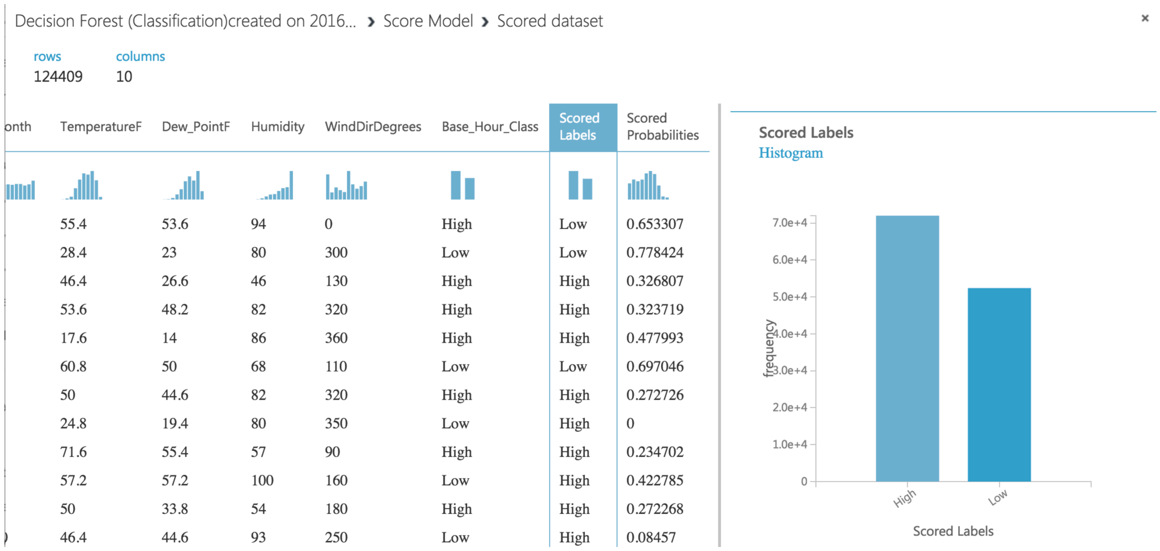
Maximum depth of the decision trees: 32

Number of random splits per node: 128

Minimum number of samples per leaf node: 1

Classification Results:

Assignment 3 - To build out a web service for the models



Decision Forest (Classification)created on 2016... › Score Model › Scored dataset

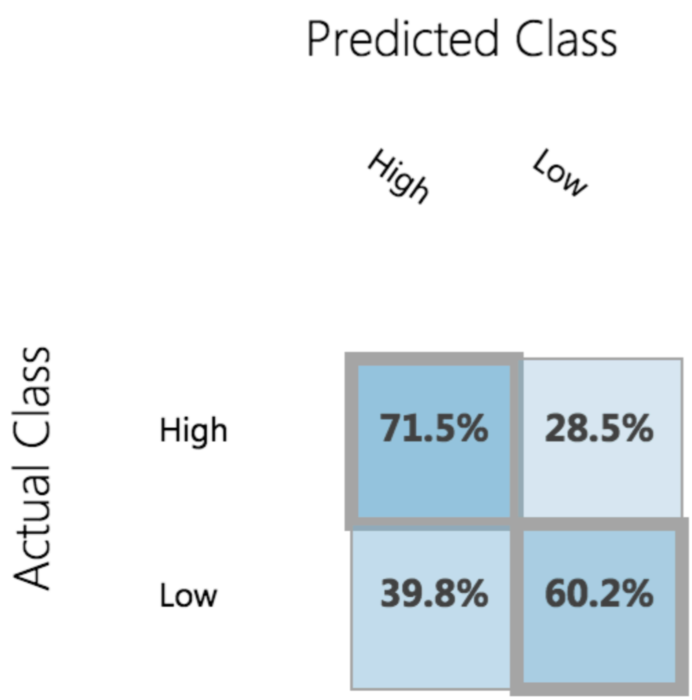
rows 124409 columns 10

onth	TemperatureF	Dew_PointF	Humidity	WindDirDegrees	Base_Hour_Class	Scored Labels	Scored Probabilities
	55.4	53.6	94	0	High	Low	0.653307
	28.4	23	80	300	Low	Low	0.778424
	46.4	26.6	46	130	High	High	0.326807
	53.6	48.2	82	320	High	High	0.323719
	17.6	14	86	360	High	High	0.477993
	60.8	50	68	110	Low	Low	0.697046
	50	44.6	82	320	High	High	0.272726
	24.8	19.4	80	350	Low	High	0
	71.6	55.4	57	90	High	High	0.234702
	57.2	57.2	100	160	Low	High	0.422785
	50	33.8	54	180	High	High	0.272268
)	46.4	44.6	93	250	Low	High	0.08457

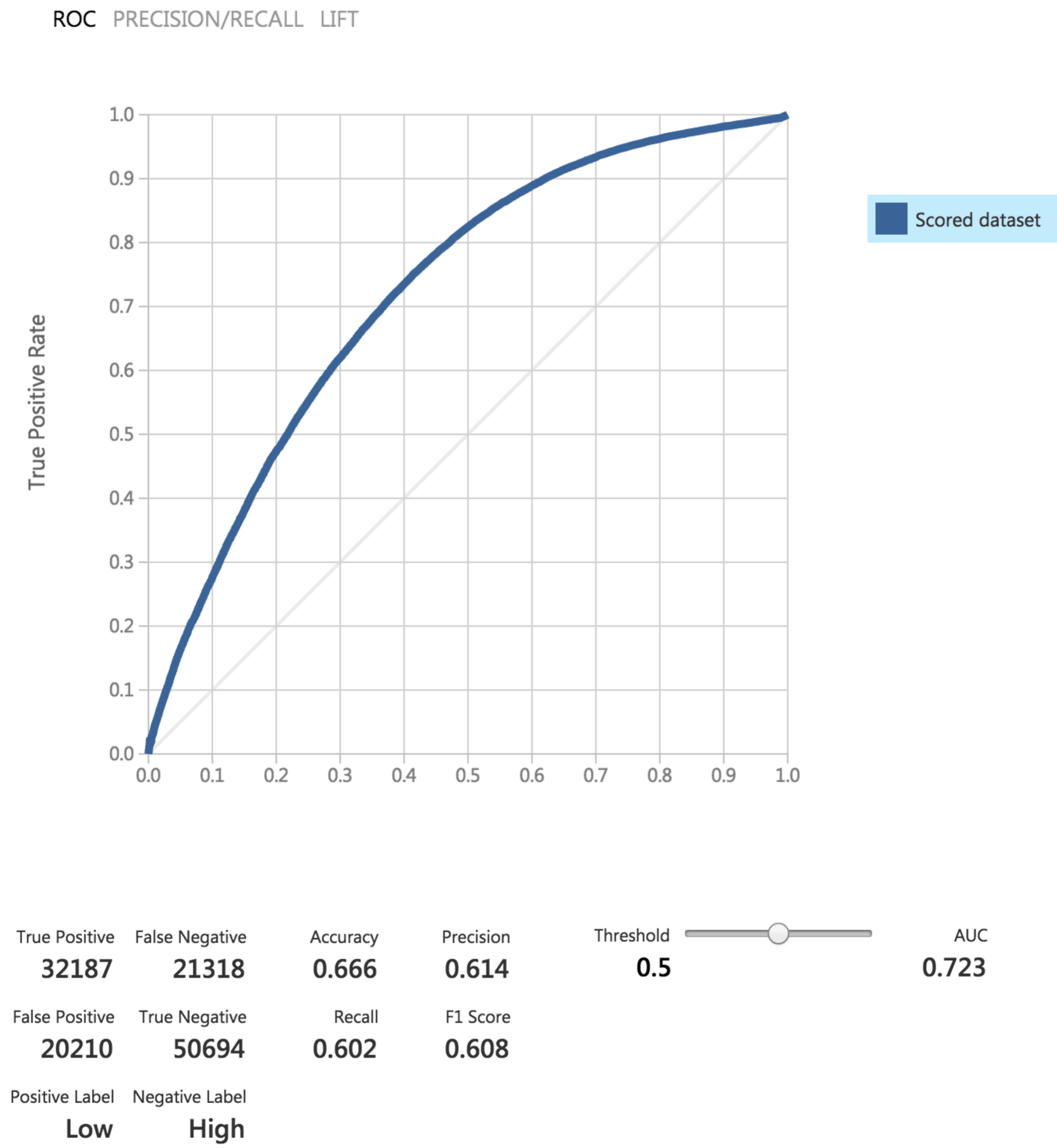
Confusion matrix:

Overall accuracy	0.66619
Average accuracy	0.66619
Micro-averaged precision	0.66619
Macro-averaged precision	0.659121
Micro-averaged recall	0.66619
Macro-averaged recall	0.658264

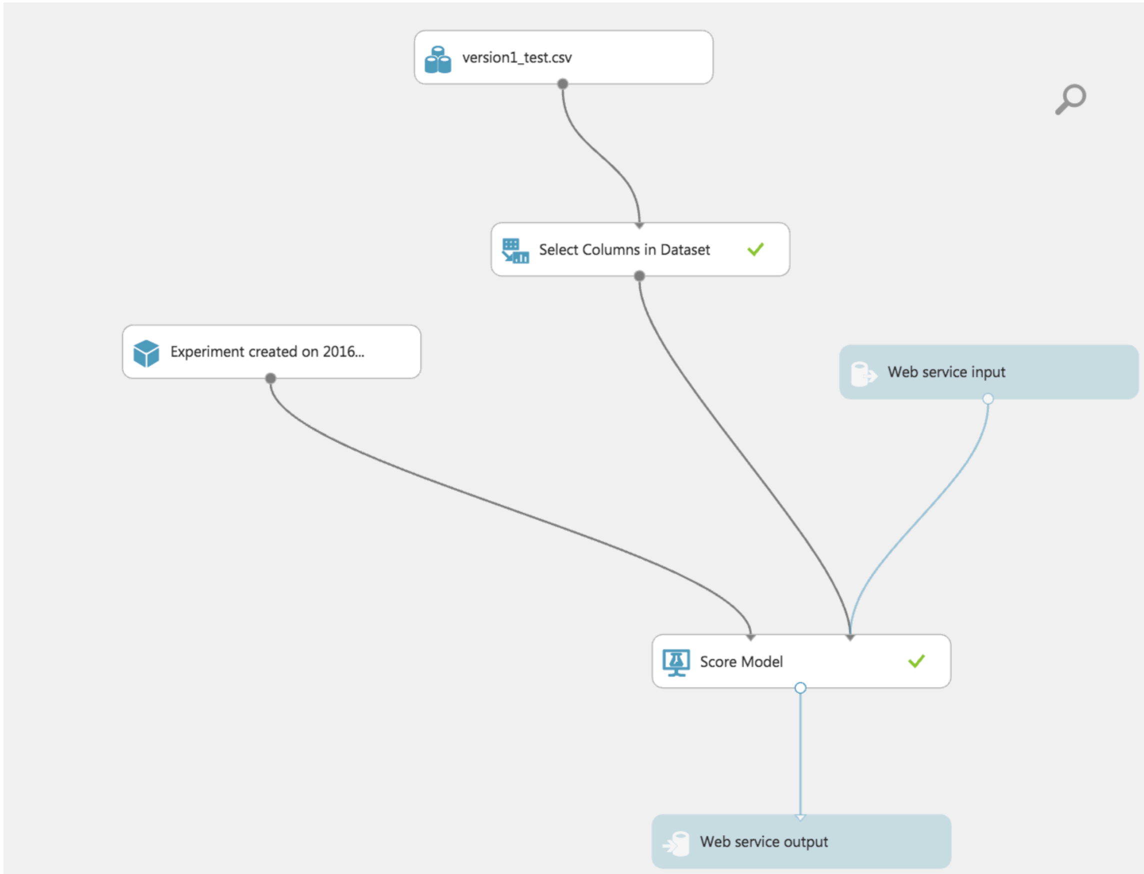
Confusion Matrix



Assignment 3 - To build out a web service for the models



Assignment 3 - To build out a web service for the models



Classification Model – K-Nearest Neighbor

K nearest neighbors is a simple algorithm that stores all available cases and classifies new cases based on a similarity measure (e.g., distance functions). KNN has been used in statistical estimation and pattern recognition already in the beginning of 1970's as a non-parametric technique.
A case is classified by a majority vote of its neighbors, with the case being assigned to the class most common amongst its K nearest neighbors measured by a distance function. If K = 1, then the case is simply assigned to the class of its nearest neighbor.

Drawback: One major drawback in calculating distance measures directly from the training set is in the case where variables have different measurement scales or there is a mixture of numerical and categorical variables.

Advantages:

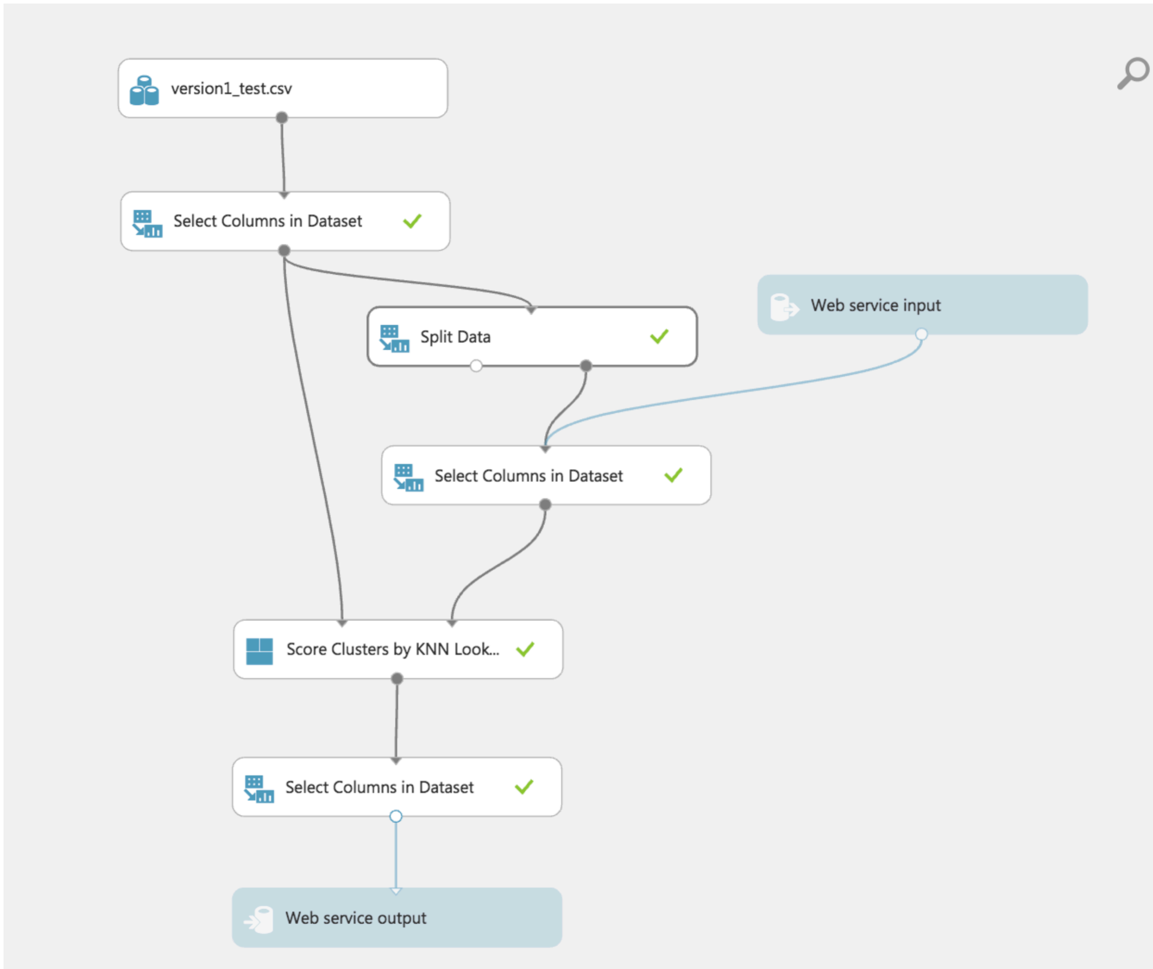
- ➔ Robust to noisy training data
- ➔ Effective if the training data is large

Disadvantages:

- ➔ Need to determine value of parameter K
- ➔ Distance based learning is not clear which type of distance to use and which attribute to use to produce the best results
- ➔ Computation cost is high

Assignment 3 - To build out a web service for the models

Trained model:

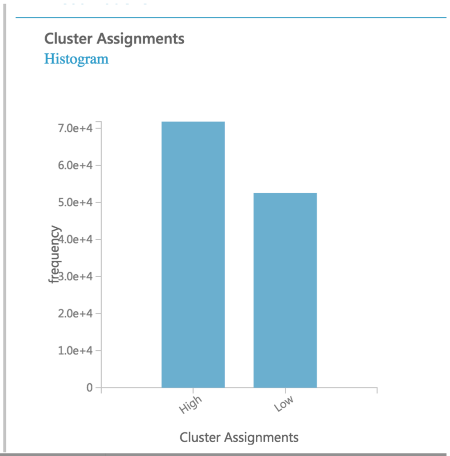


Classification Results:

K-Nearest Neighbor Model › Score Clusters by KNN Lookup › Result

rows: 124409   columns: 8

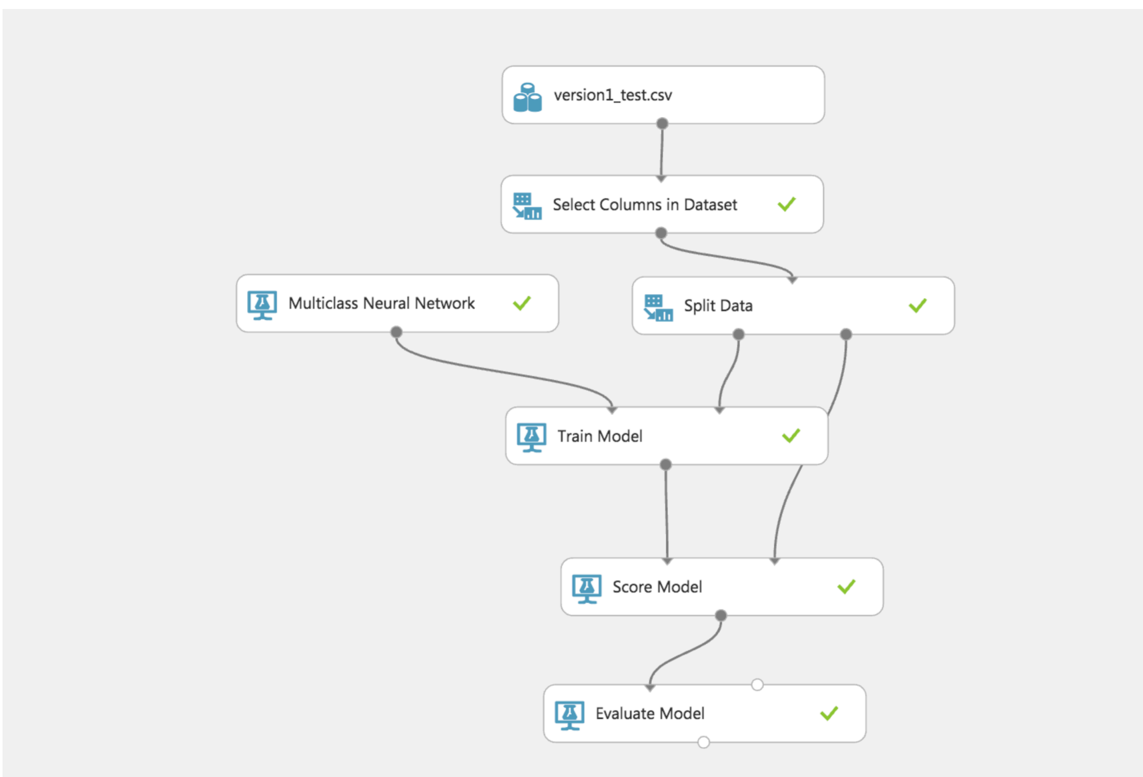
hour	Weekday	month	TemperatureF	Dew_PointF	Humidity	WindDirDegrees	Cluster Assignments
4	1	7	55.4	53.6	94	0	Low
3	0	2	28.4	23	80	300	Low
13	1	4	46.4	26.6	46	130	High
7	1	6	53.6	48.2	82	320	High
21	0	1	17.6	14	86	360	High
16	0	9	60.8	50	68	110	Low
6	1	6	50	44.6	82	320	High
15	1	2	24.8	19.4	80	350	High
12	1	5	71.6	55.4	57	90	High
3	0	8	57.2	57.2	100	160	High
13	1	5	50	33.8	54	180	High
7	1	10	46.4	44.6	93	250	High



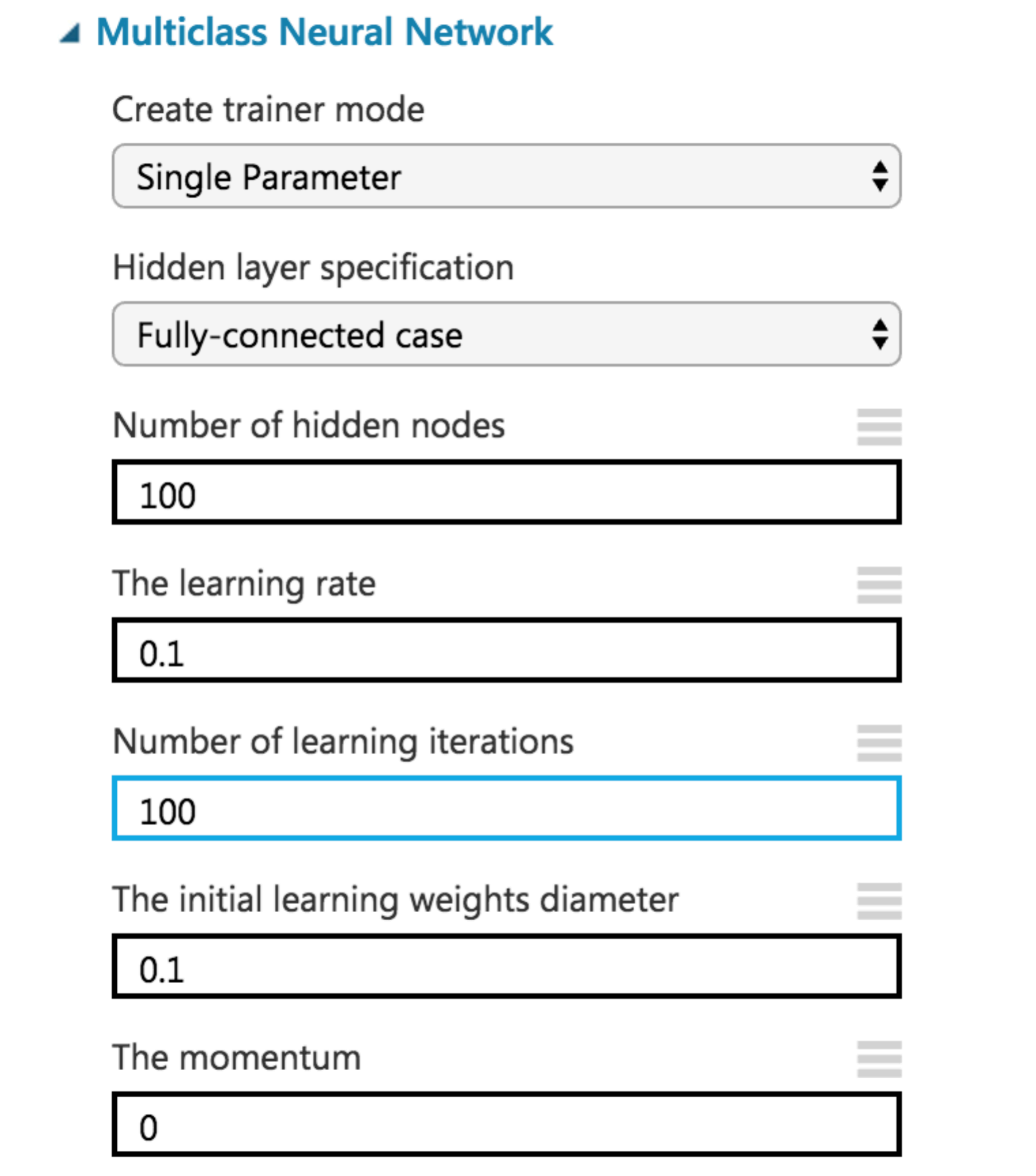
Classification Model – Neural Network

A neural network is a set of interconnected layers, in which the inputs lead to outputs by a series of weighted edges and nodes. The weights on the edges are learned when training the neural network on the input data. The direction of the graph proceeds from the inputs through the hidden layer, with all nodes of the graph connected by the weighted edges to nodes in the next layer.

To compute the output of the network for any given input, a value is calculated for each node in the hidden layers and in the output layer. For each node, the value is set by calculating the weighted sum of the values of the nodes in the previous layer and applying an activation function to that weighted sum.



Properties:


Multiclass Neural Network
Create trainer mode
Single Parameter
Hidden layer specification
Fully-connected case
Number of hidden nodes
100
The learning rate
0.1
Number of learning iterations
100
The initial learning weights diameter
0.1
The momentum
0

Assignment 3 - To build out a web service for the models

Classification Results:

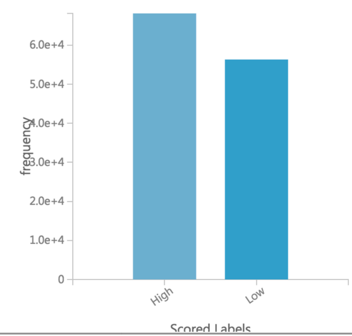
Neural Network (Classification) created on 201... › Score Model › Scored dataset

rows 124409 columns 11

ratureF	Dew_PointF	Humidity	WindDirDegrees	Base_Hour_Class	Scored Probabilities for Class "High"	Scored Probabilities for Class "Low"	Scored Labels
	53.6	94	0	High	0.455678	0.544314	Low
	23	80	300	Low	0.330335	0.669623	Low
	26.6	46	130	High	0.637899	0.362097	High
	48.2	82	320	High	0.793107	0.206904	High
	14	86	360	High	0.660779	0.339216	High
	50	68	110	Low	0.425338	0.574684	Low
	44.6	82	320	High	0.785519	0.214492	High
	19.4	80	350	Low	0.847238	0.152757	High
	55.4	57	90	High	0.586336	0.413659	High
	57.2	100	160	Low	0.431511	0.568479	Low
	33.8	54	180	High	0.636197	0.363801	High

Visualizations

Scored Labels
Histogram

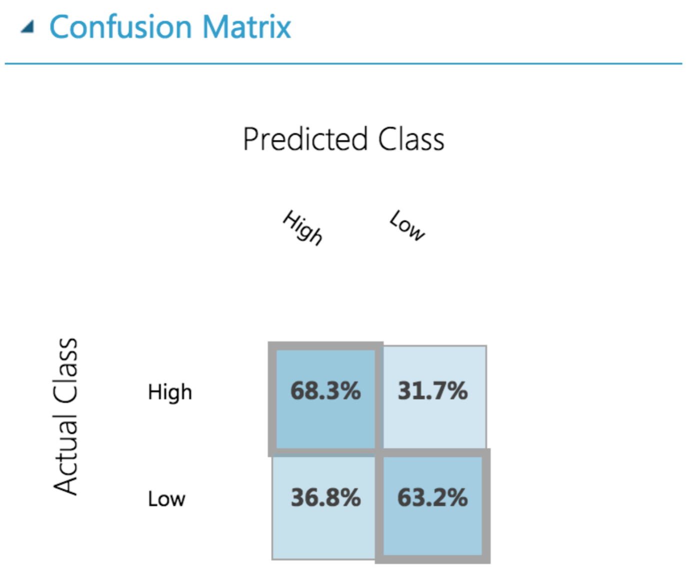


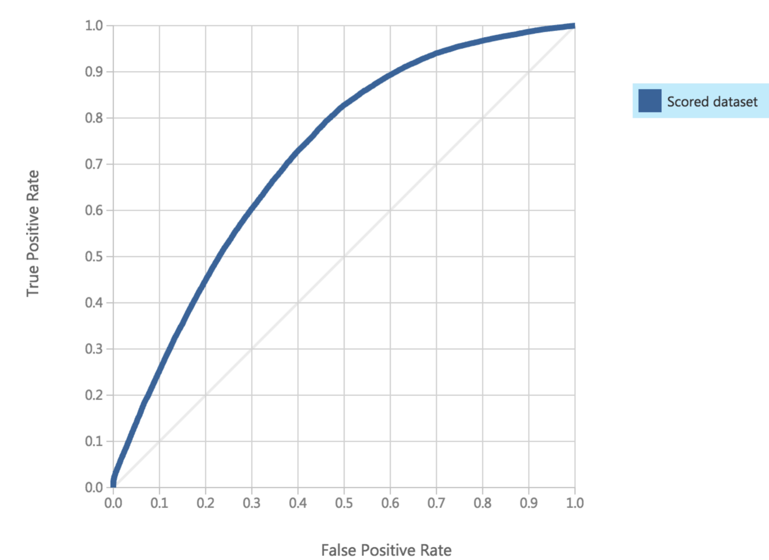
Performance Metric:

Metrics

Overall accuracy	0.660804
Average accuracy	0.660804
Micro-averaged precision	0.660804
Macro-averaged precision	0.655574
Micro-averaged recall	0.660804
Macro-averaged recall	0.657251

Assignment 3 - To build out a web service for the models

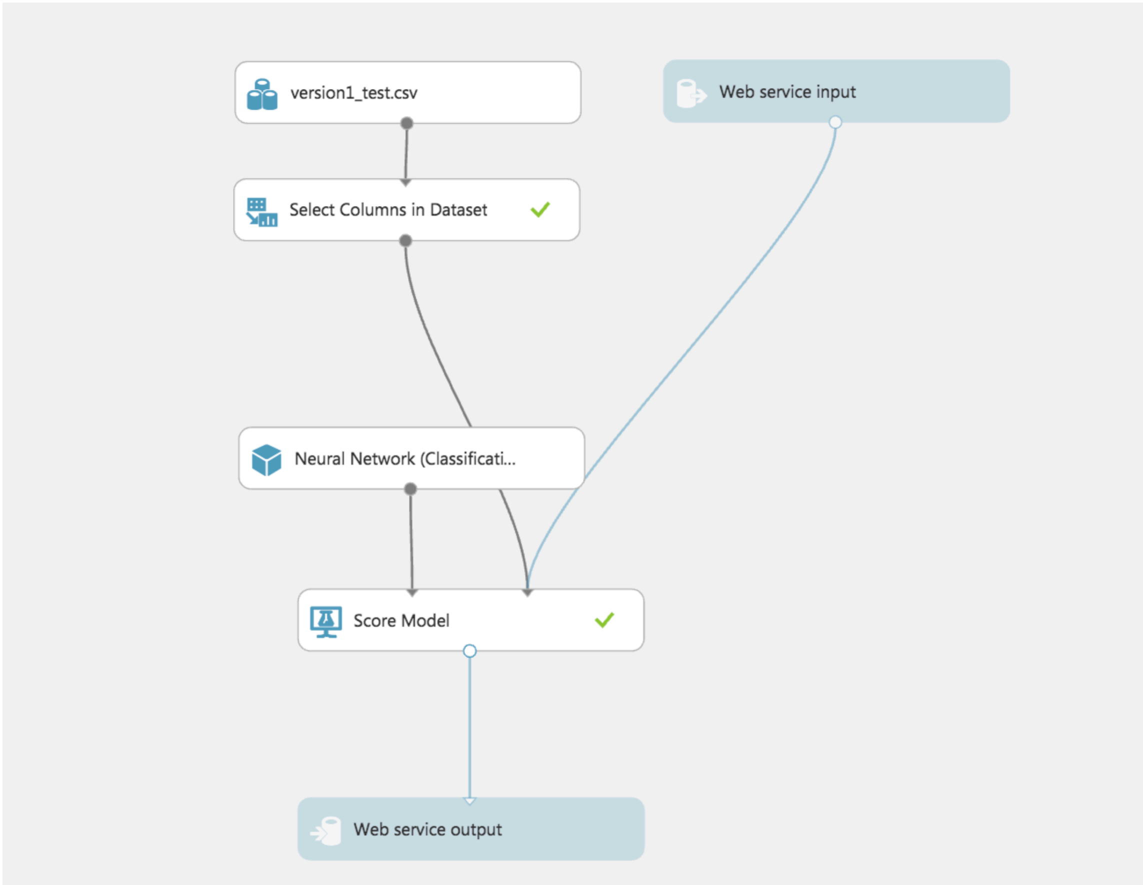




Accuracy	Precision	Threshold	AUC
0.659	0.594	0.5	0.718

Team 12 – Priti Agarwal, Yalin Li, Zhi Li

Assignment 3 - To build out a web service for the models



Assignment 3 - To build out a web service for the models

Model	Performance Metrics
a) **Logistic Regression**	Accuracy 0.610 Precision 0.579 AUC 0.621
b) **Decision Forest**	Accuracy 0.666 Precision 0.614 AUC 0.723
c) **Neural Network**	Accuracy 0.659 Precision 0.594 AUC 0.718

Result: The model with maximum accuracy is Decision Forest and it has the highest AUC, 71.8%

Assignment 3 - To build out a web service for the models

Clustering Modeling

Tools Used: Microsoft Azure ML Studio

Building the K-Mean and Hierarchical Clustering Model

Steps (K-Mean Model):

1. Upload the data.csv into Datasets of Azure Machine Learning Studio.
2. Input data from My Datasets
3. Select Columns from Dataset which are “meternumb”, “Consumption”, “area_floor._m.sqr”, “BuildingID”, “type”.
4. Using SQL code to Avg each column to make sure each column has same row.
5. Edit Metadata which change the type of column “type” to categorical.
6. Convert to indicator values about ‘type’ and make elect and dist_heating to 1 and 0; and then normalize data.
7. Select new columns for model use which are: “meternumb”, “area_floor._m.sqr”, “Comsumption”, “type-Dist_Heating”, “type-elect” without “type”.
8. Use K-Mean Clustering of Azure Machine Learning Studio to train the model, based on the algorithm.
9. From train clustering model we can see two cluster which is 0 and 1.
10. Select Column again which include new column which called “Assignments” and present in 0 and 1.
11. Score the model in cluster by KNN to get “Cluster Assignments”.
12. Select “Cluster Assignments” and Evaluate the model.

Assignment 3 - To build out a web service for the models

K-Means Clustering:

Clustering uses iterative techniques to group cases in a dataset into clusters that contain similar characteristics. These groupings are useful for exploring data, identifying anomalies in the data, and eventually for making predictions. Clustering models also can help you identify relationships in a dataset that you might not logically derive by browsing or simple observation.

- **First N**. Some initial number of data points are chosen from the data set and used as the initial means.
    - Also called the *Forgy method*.
- **Random**. The algorithm randomly places a data point in a cluster and then computes the initial mean to be the centroid of the cluster's randomly assigned points.
  Also called the *random partition* method.
- **K-Means++**. This is the default method for initializing clusters.
  The **K-means ++** algorithm was proposed in 2007 by David Arthur and Sergei Vassilvitskii to avoid poor clustering by the standard k-means algorithm. **K-means ++** improves upon standard K-means by using a different method for choosing the initial cluster centers.
- **K-Means++Fast**. A variant of the **K-means ++** algorithm that was optimized for faster clustering.
- **Evenly**. Centroids are located equidistant from each other in the d-Dimensional space of n data points.
- **Use label column**. The values in the label column are used to guide the selection of centroids.

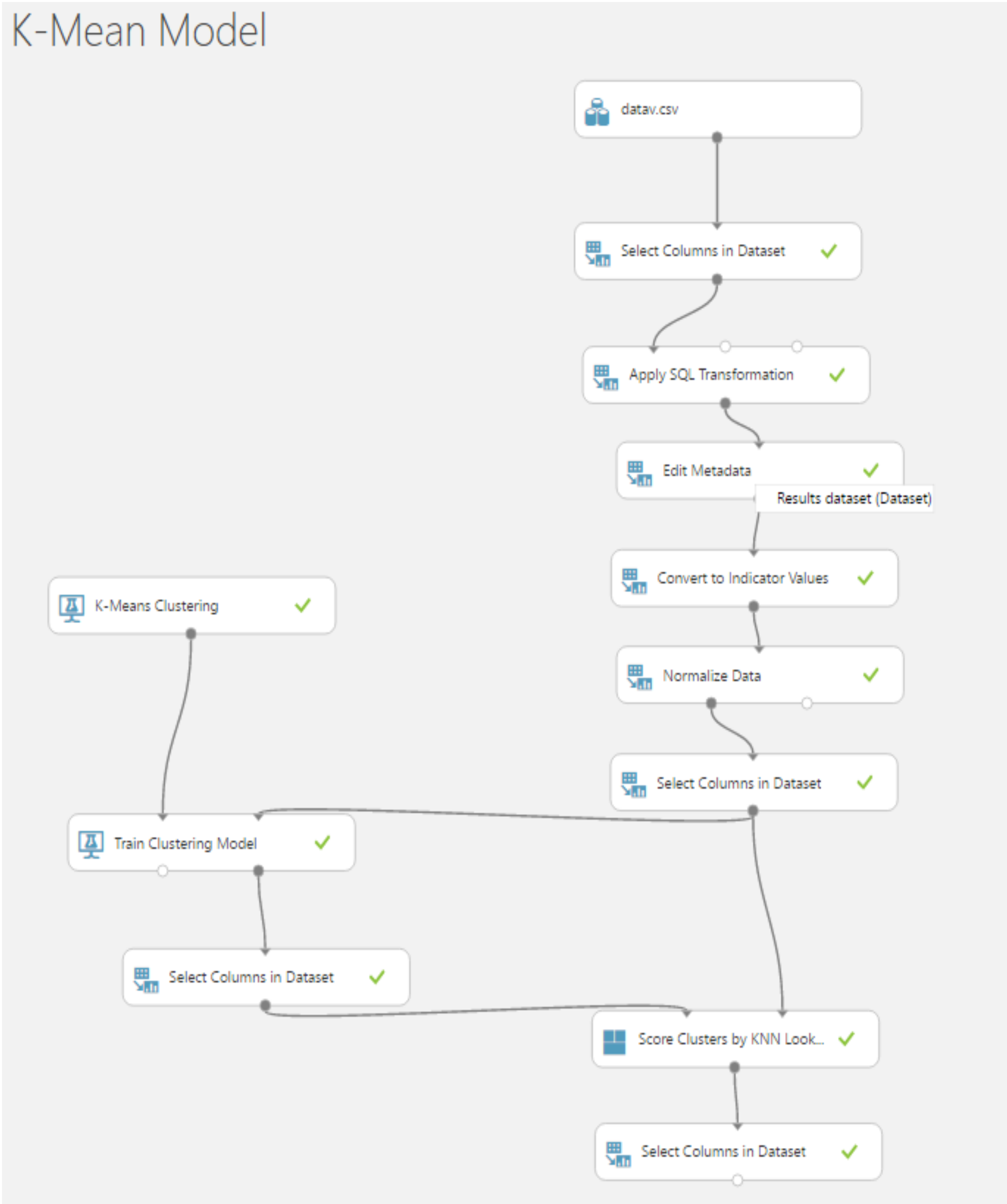
Advantages:

- Faster, because order of time complexity is linear with the number of data
- Works great if clusters are spherical
- Simple, easy to implement

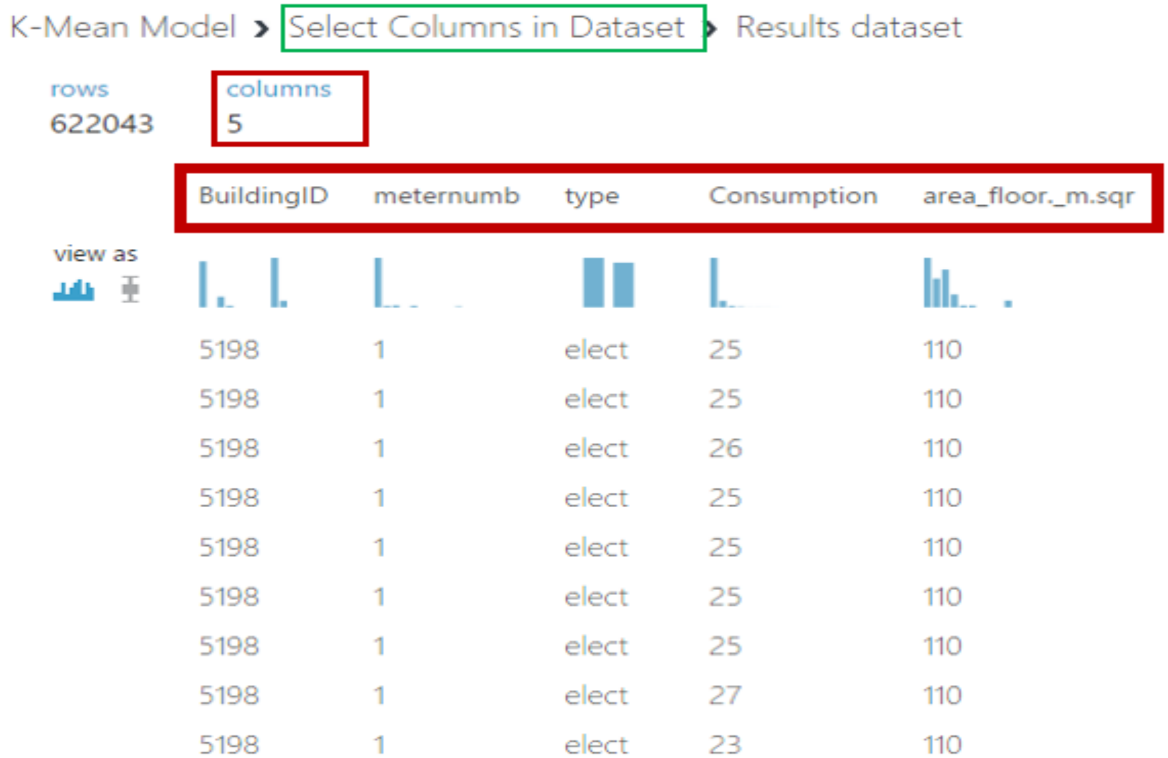
Disadvantages:

- Different densities may work poorly with clusters with different densities but spherical shape.
- Sensitive to outliers

Assignment 3 - To build out a web service for the models



Assignment 3 - To build out a web service for the models



K-Mean Model › Select Columns in Dataset › Results dataset

rows 622043

columns 5

view as

BuildingID	meternumb	type	Consumption	area_floor._m.sqr
5198	1	elect	25	110
5198	1	elect	25	110
5198	1	elect	26	110
5198	1	elect	25	110
5198	1	elect	25	110
5198	1	elect	25	110
5198	1	elect	25	110
5198	1	elect	27	110
5198	1	elect	23	110

K-Mean Model › Apply SQL Transformation › Results dataset

rows 78 columns 5

view as

BuildingID	meternumb	area_floor._m.sqr	type	Comsumption
5198	1	110	elect	24.693582
5199	1	11697	elect	343.204116
5286	1	469	elect	2.544513
5288	1	8766	elect	0
5290	1	11068	elect	128.91828
5290	3	11068	elect	0
5290	8	11068	elect	0
5304	1	41339	elect	5.612967
5304	3	41339	elect	600.215573

Assignment 3 - To build out a web service for the models

K-Mean Model › Convert to Indicator Values › Results dataset

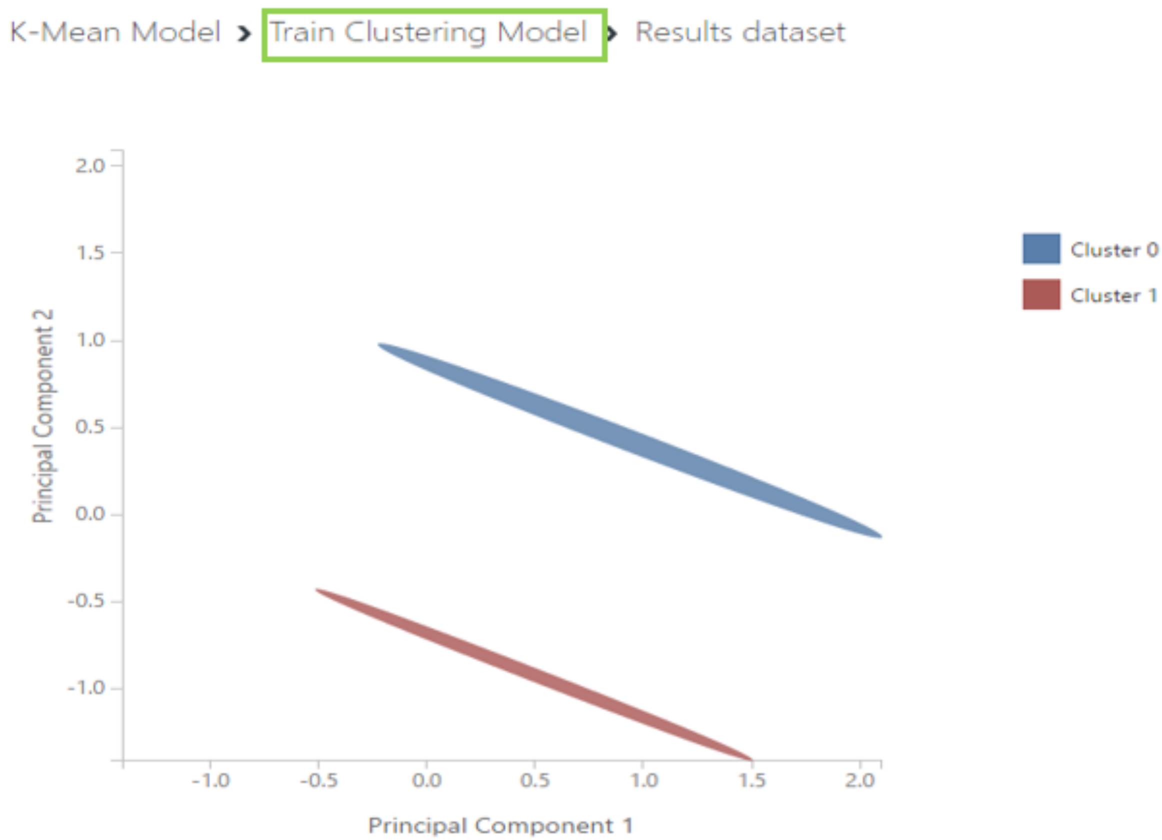
rows	columns
78	7

BuildingID	meternumb	area_floor._m.sqr	type	Comsumption	type-Dist_Heating	type-elect
5198	1	110	elect	24.693582	0	1
5199	1	11697	elect	343.204116	0	1
5286	1	469	elect	2.544513	0	1
5288	1	8766	elect	0	0	1
5290	1	11068	elect	128.91828	0	1
5290	3	11068	elect	0	0	1
5290	8	11068	elect	0	0	1

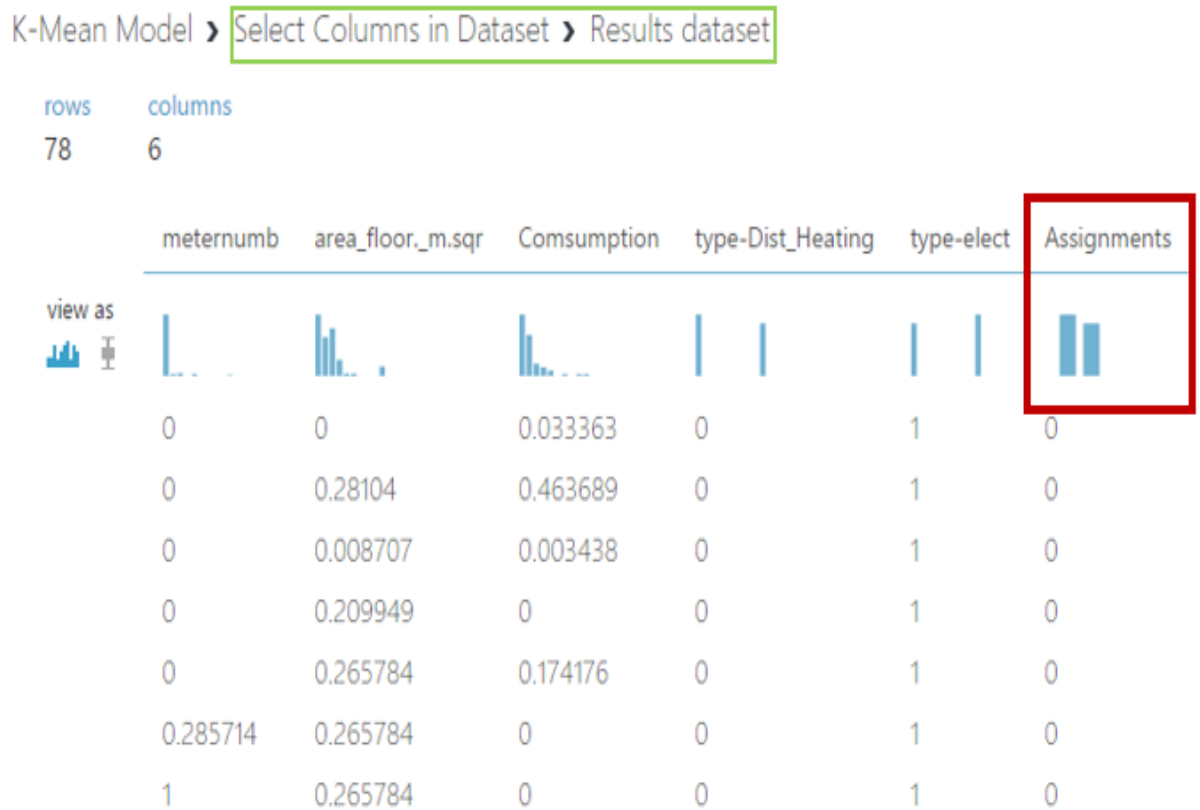
K-Mean Model › Select Columns in Dataset › Results dataset

rows	columns
78	5

meternumb	area_floor._m.sqr	Comsumption	type-Dist_Heating	type-elect
0	0	0.033363	0	1
0	0.28104	0.463689	0	1
0	0.008707	0.003438	0	1
0	0.209949	0	0	1
0	0.265784	0.174176	0	1
0.285714	0.265784	0	0	1


K-Mean Model
Train Clustering Model
Results dataset
Cluster 0
Cluster 1
Principal Component 2
Principal Component 1
2.0
1.5
1.0
0.5
0.0
-0.5
-1.0
-1.0
-0.5
0.0
0.5
1.0
1.5
2.0

Assignment 3 - To build out a web service for the models

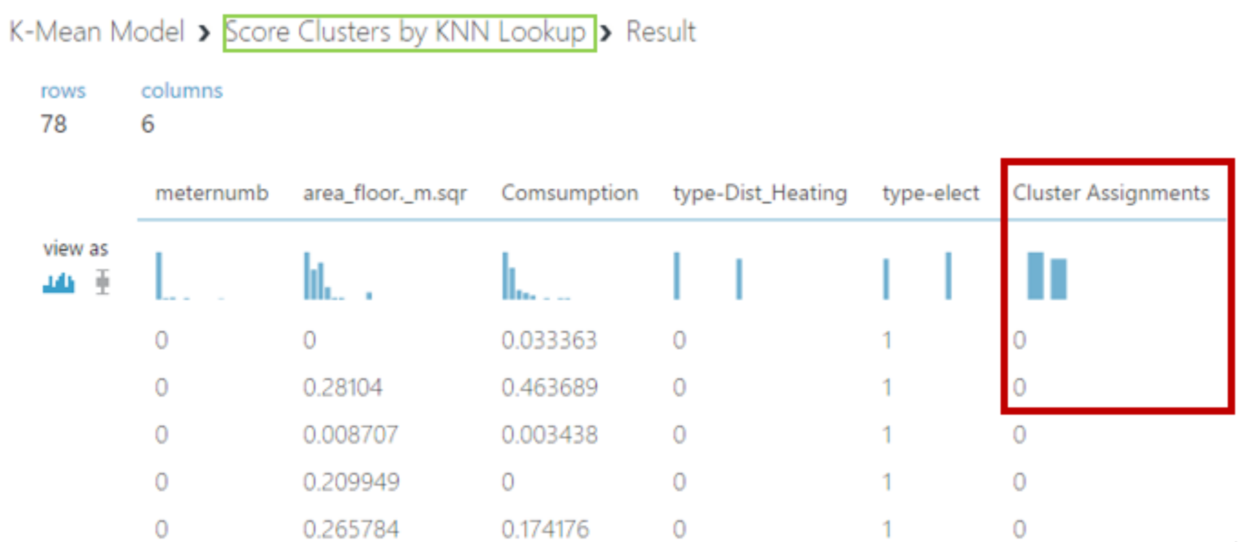


K-Mean Model › Select Columns in Dataset › Results dataset

rows 78

columns 6

meternumb	area_floor._m.sqr	Comsumption	type-Dist_Heating	type-elect	Assignments
0	0	0.033363	0	1	0
0	0.28104	0.463689	0	1	0
0	0.008707	0.003438	0	1	0
0	0.209949	0	0	1	0
0	0.265784	0.174176	0	1	0
0.285714	0.265784	0	0	1	0
1	0.265784	0	0	1	0



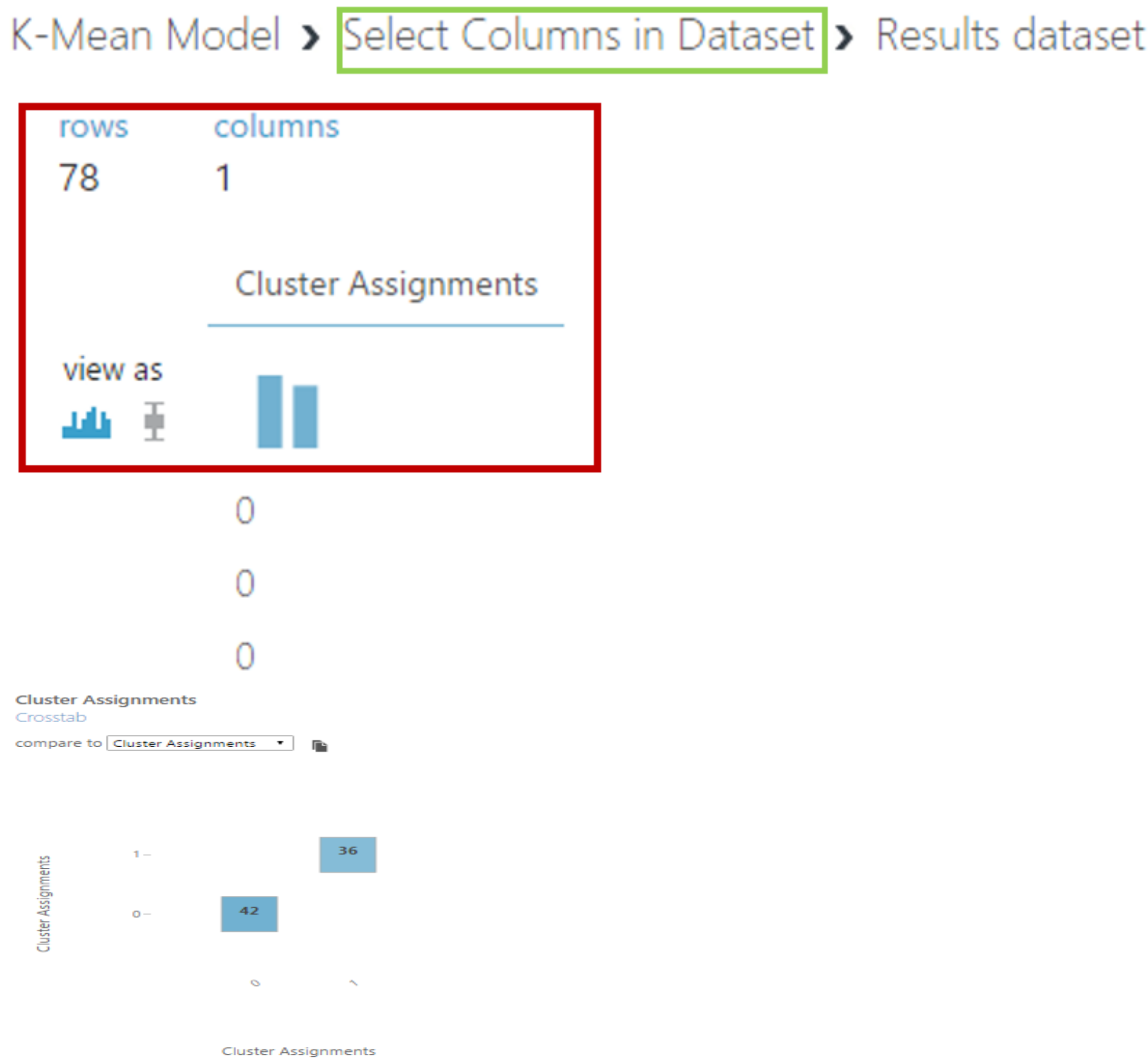
K-Mean Model › Score Clusters by KNN Lookup › Result

rows 78

columns 6

meternumb	area_floor._m.sqr	Comsumption	type-Dist_Heating	type-elect	Cluster Assignments
0	0	0.033363	0	1	0
0	0.28104	0.463689	0	1	0
0	0.008707	0.003438	0	1	0
0	0.209949	0	0	1	0
0	0.265784	0.174176	0	1	0

Assignment 3 - To build out a web service for the models



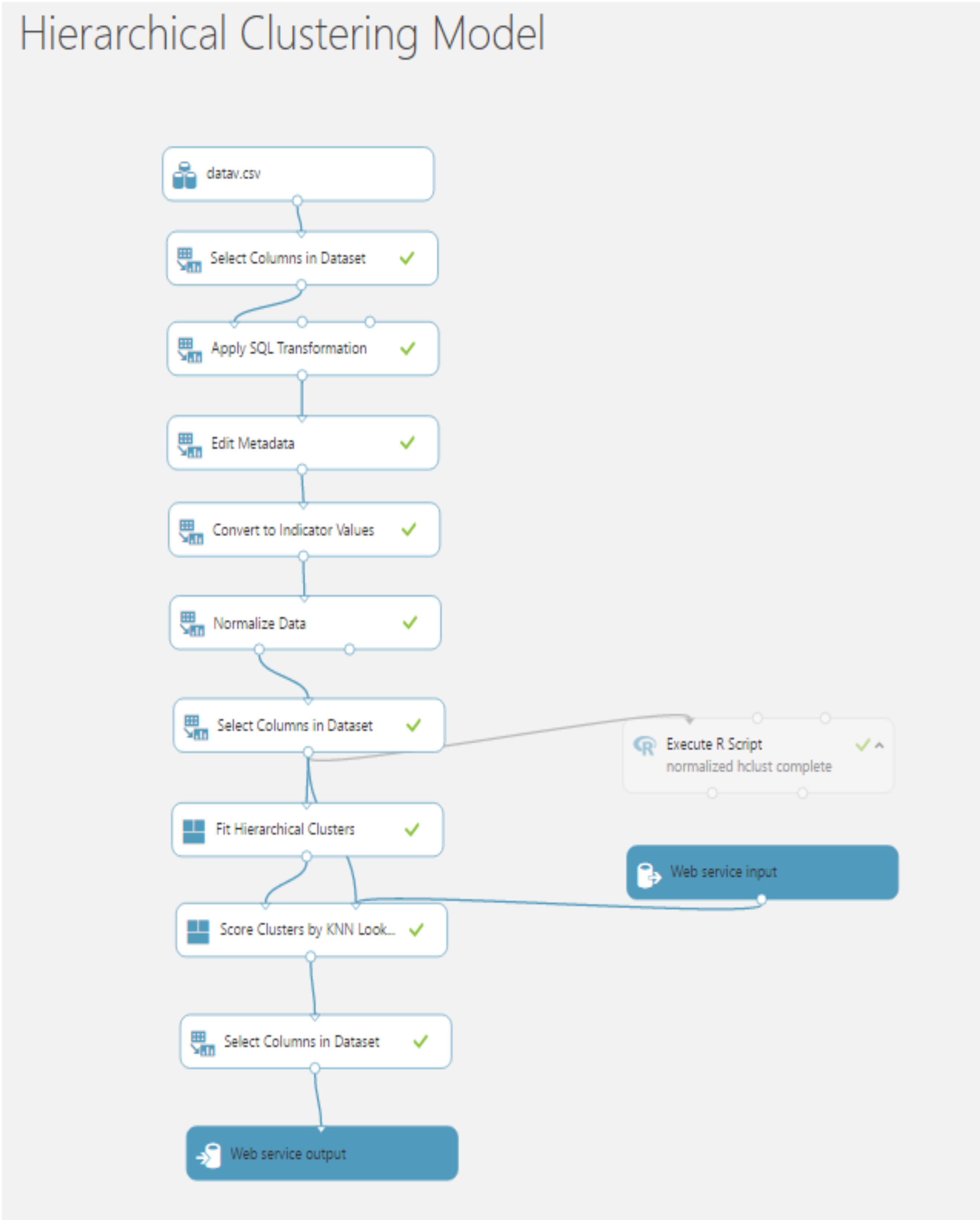
Assignment 3 - To build out a web service for the models

Steps (Hierarchical Clustering Model):

1. Upload the data.csv into Datasets of Azure Machine Learning Studio.
2. Input data from My Datasets
3. Select Columns from Dataset which are "meternumb", "Consumption", "area_floor._m.sqr", "BuildingID", "type".
4. Using SQL code to Avg each column to make sure each column has same row.
5. Edit Metadata which change the type of column "type" to categorical.
6. Convert to indicator values about 'type' and make elect and dist_heating to 1 and 0; and then normalize data.
7. Select new columns for model use which are: "meternumb", "area_floor._m.sqr", "Comsumption", "type-Dist_Heating", "type-elect" without "type".
8. Execute R Script to output Graphics.
9. Use Fit Hierarchical Cluster of Azure Machine Learning Studio to train the model, based on the algorithm, and get new column "Cluster Assignments".
10. From train clustering model we can see two cluster which is 1 and 2.
11. Score the model in cluster by KNN lookup based on the algorithm.
12. Select "Cluster Assignments" and Evaluate the model.

Hierarchical Clustering: It is a method of cluster analysis, which seeks to build a hierarchy of clusters. Hierarchical clustering only requires a measure of similarity between groups of data points.

Assignment 3 - To build out a web service for the models



Team 12 – Priti Agarwal, Yalin Li, Zhi Li

Assignment 3 - To build out a web service for the models

K-Mean Model › Select Columns in Dataset › Results dataset

rows
622043

columns
5

view as

BuildingID	meternumb	type	Consumption	area_floor._m.sqr
5198	1	elect	25	110
5198	1	elect	25	110
5198	1	elect	26	110
5198	1	elect	25	110
5198	1	elect	25	110
5198	1	elect	25	110
5198	1	elect	25	110
5198	1	elect	27	110
5198	1	elect	23	110

Assignment 3 - To build out a web service for the models

K-Mean Model › Apply SQL Transformation › Results dataset

rows	columns
78	5

view as

BuildingID	meternumb	area_floor._m.sqr	type	Comsumption
5198	1	110	elect	24.693582
5199	1	11697	elect	343.204116
5286	1	469	elect	2.544513
5288	1	8766	elect	0
5290	1	11068	elect	128.91828
5290	3	11068	elect	0
5290	8	11068	elect	0
5304	1	41339	elect	5.612967
5304	3	41339	elect	600.215573

K-Mean Model › Convert to Indicator Values › Results dataset

rows	columns
78	7

view as

BuildingID	meternumb	area_floor._m.sqr	type	Comsumption	type-Dist_Heating	type-elect
5198	1	110	elect	24.693582	0	1
5199	1	11697	elect	343.204116	0	1
5286	1	469	elect	2.544513	0	1
5288	1	8766	elect	0	0	1
5290	1	11068	elect	128.91828	0	1
5290	3	11068	elect	0	0	1
5290	8	11068	elect	0	0	1

Assignment 3 - To build out a web service for the models

K-Mean Model › Select Columns in Dataset › Results dataset

rows	columns
78	5

meternumb	area_floor._m.sqr	Comsumption	type-Dist_Heating	type-elect
0	0	0.033363	0	1
0	0.28104	0.463689	0	1
0	0.008707	0.003438	0	1
0	0.209949	0	0	1
0	0.265784	0.174176	0	1
0.285714	0.265784	0	0	1

```
# Map 1-based optional input ports to variables
dataset1 <- maml.mapInputPort(1) # class: data.frame
require(stats)
require(flashClust)
require(graphics); require(grDevices)
d_dis <- dist(dataset1)
hc_dis_complete <- hclust(d_dis, method = "complete")

dend1 <- as.dendrogram(hc_dis_complete)
plot(dend1)
# Select data.frame to be sent to the output Dataset port
maml.mapOutputPort(hc_dis_complete);
```

Assignment 3 - To build out a web service for the models

Hierarchical Clustering Model › Execute R Script › R Device

Standard Output

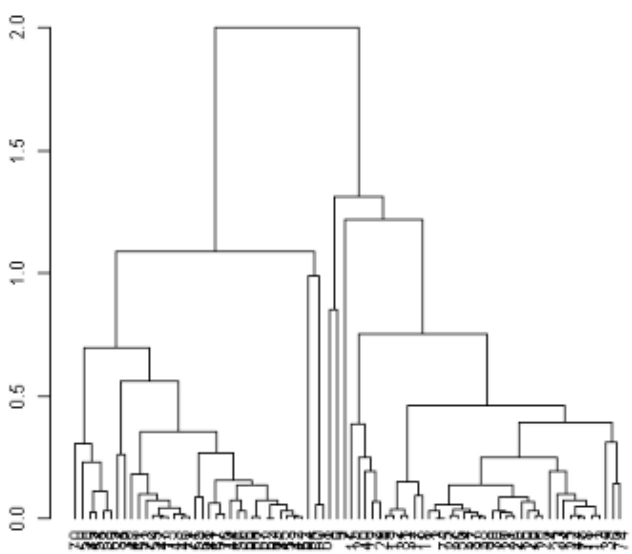
RWorker pushed "port1" to R workspace.
Beginning R Execute Script

[1] 56000
Loading objects:
  port1
[1] "Loading variable port1..."

Standard Error

R reported no errors.

Graphics



Assignment 3 - To build out a web service for the models

Hierarchical Clustering Model › Fit Hierarchical Clusters › Clustered dataset

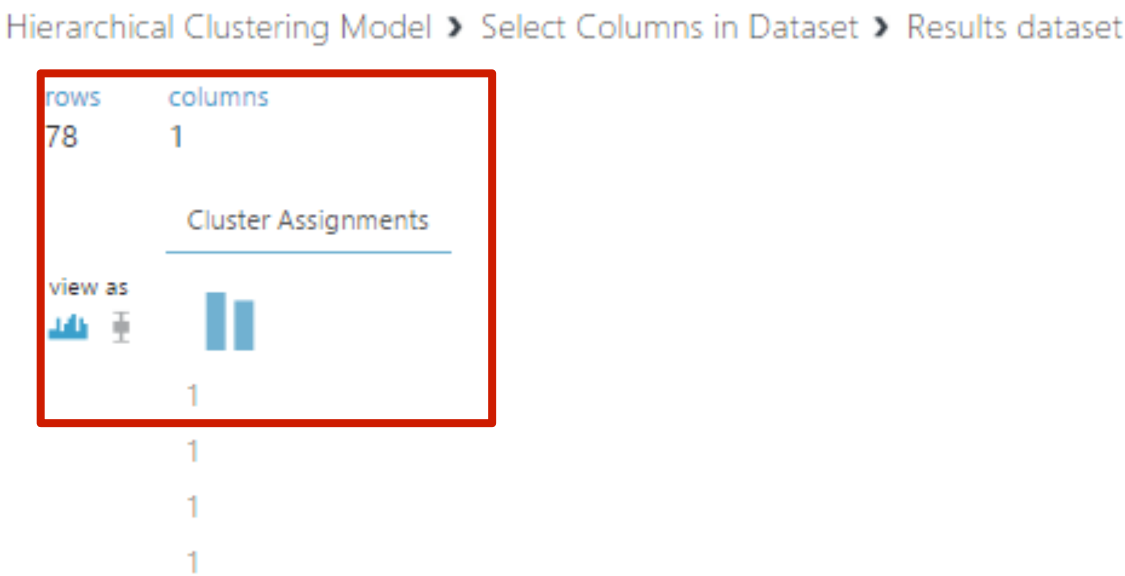
rows 78

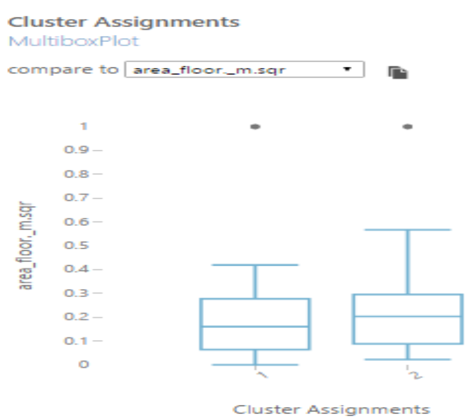
columns 6

view as

meternumb	area_floor._m.sqr	Comsumption	type-Dist_Heating	type-elect	Cluster Assignments
0	0	0.033363	0	1	1
0	0.28104	0.463689	0	1	1
0	0.008707	0.003438	0	1	1
0	0.209949	0	0	1	1
0	0.265784	0.174176	0	1	1
0.285714	0.265784	0	0	1	1
1	0.265784	0	0	1	1
0	1	0.007583	0	1	1

Assignment 3 - To build out a web service for the models





3. Invoke Rest API:

Tool used: Microsoft Visual Studio 2015
Language used: C#
Web technology used: asp.net, html, CSS

Prerequisite:

Services can be called with any programming language and from any device that satisfies three criteria:

- Has a network connection
- Has SSL capabilities to perform HTTPS requests
- Can parse JSON (by hand or support libraries)

Once the experiment has been deployed, there are four pieces of information that we need to call either the RRS or BES service.

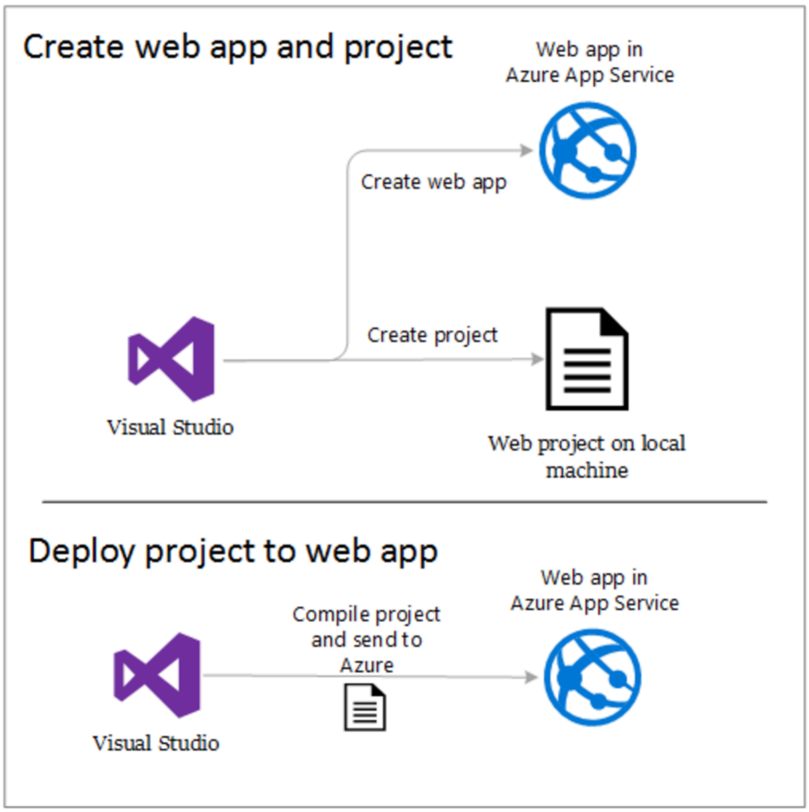
a. The service **API key** or **Primary key**
b. The service **request URI**
c. The expected API **request headers** and **body**
d. The expected API **response headers** and **body**

How to Invoke Request Response Service:

a. Create HttpClient object.
b. Set-up the input message by copying the input-json body and replacing the input values with the values obtained from UI.
c. Set the Authorization header based on the api key:
   "Bearer" + apikey
d. Setup the client base address
e. Call REST API and retrieve the response client using **'await'**
f. If the response is a success code, then store the result.
g. If the response is an error code, then show the error.

Assignment 3 - To build out a web service for the models

Deploy an ASP.NET web app to Azure App Service, using Visual Studio:



1. In the solution explorer, right click and choose Publish. In a few seconds, the Publish Web wizard appears.
2. On the Connection tab of the Publish Web wizard, click Next.
3. Next is the Settings tab. Here you can change the build configuration to deploy a debug build for remote debugging.
4. The tab also offers several File Publish Options.
5. On the Settings tab, click Next.
6. On the Preview tab, click Publish. Upon successful deployment, the default browser automatically opens to the URL of the deployed web app

Assignment 3 - To build out a web service for the models

C# code:

```
public static string PredictClassificationLR(Inputs classification)
{
    using (var client = new HttpClient())
    {
        var scoreRequest = new
        {

            Inputs = new Dictionary<string, StringTable>() {
                {
                    "input1",
                    new StringTable()
                    {
                        ColumnNames = new string[] {"hour", "Weekday", "month", "TemperatureF", "Dew_PointF", "Humidity", "WindDirDeg
                        Values = new string[,] {  { classification.hour, classification.weekday, classification.month, classification
                        { classification.hour, classification.weekday, classification.month, classification.temperatreF, classificati

                    }
                },
            },
            GlobalParameters = new Dictionary<string, string>()
            {
            }
        };
        const string apiKey = "QhiNfApR1fHRugbuFs6ZPJ9/XF/JdljtGzx1fAANnxK23b4DiN//Sehf9y+/Dtak1Cyys8upQBmDto96osTMnQ==";
        client.DefaultRequestHeaders.Authorization = new AuthenticationHeaderValue("Bearer", apiKey);

        client.BaseAddress = new Uri("https://ussouthcentral.services.azureml.net/workspaces/bd7ad4350c2245faab8796d385b1c5e5/service
```

Dynamic Show input:

```
switch (ModelType)
{
    case "logiticRegression":
        //hour
        form.Controls.Add(new LiteralControl("</br><h3>Hour: </h3>"));
        TextBox hour1 = new TextBox();
        hour1.ID = "hour1";
        form.Controls.Add(hour1);
        //weekday
        form.Controls.Add(new LiteralControl("</br><h3>weekday: </h3>"));
        TextBox weekday1 = new TextBox();
        weekday1.ID = "weekday1";
        form.Controls.Add(weekday1);
        //month
        form.Controls.Add(new LiteralControl("</br><h3>month: </h3>"));
        TextBox month1 = new TextBox();
        month1.ID = "month1";
        form.Controls.Add(month1);
        //temperature F
        form.Controls.Add(new LiteralControl("</br><h3>Temperature F: </h3>"));
        TextBox tf1 = new TextBox();
        tf1.ID = "tf1";
        form.Controls.Add(tf1);
        //drewpoint F
        form.Controls.Add(new LiteralControl("</br><h3>Drewpoint F: </h3>"));
        TextBox df1 = new TextBox();
        df1.ID = "df1";
```

Assignment 3 - To build out a web service for the models

Button clicked function:

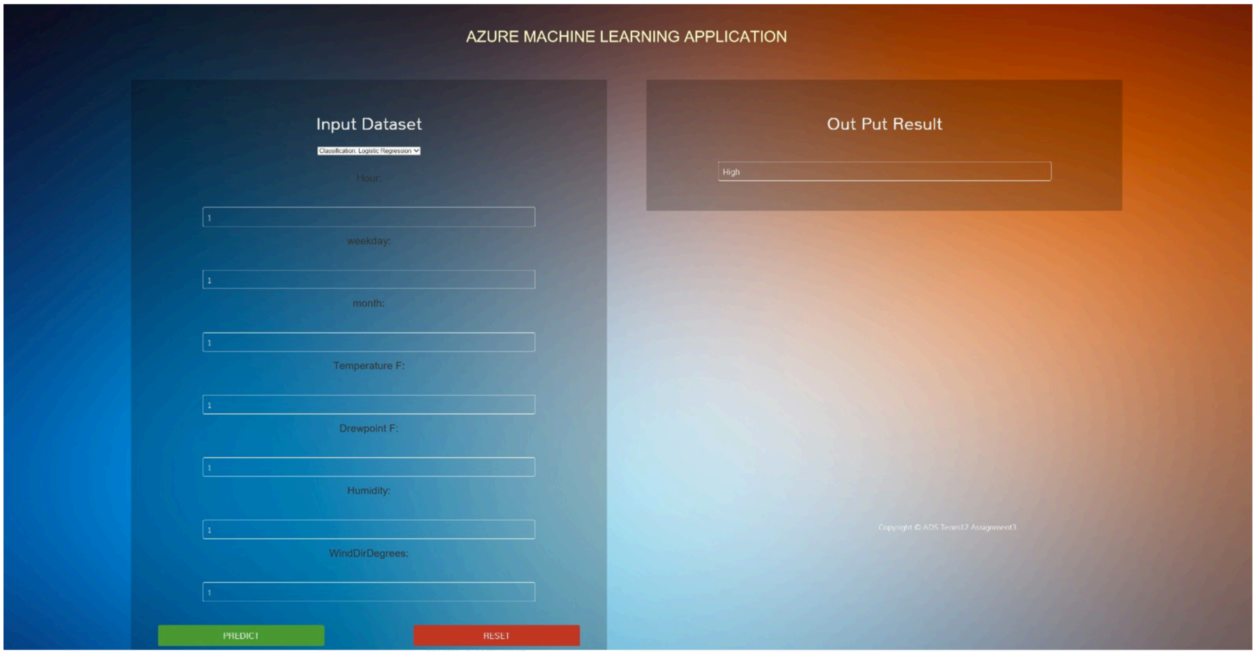
```
protected void btnPredictClicked(object sender, EventArgs e)
{
    switch (ModelType)
    {
        case "logiticRegression":
            Inputs logisticRegression = new Inputs();
            logisticRegression.hour = ((TextBox)div1.FindControl("hour1")).Text; ;
            logisticRegression.weekday = ((TextBox)div1.FindControl("weekday1")).Text;
            logisticRegression.month = ((TextBox)div1.FindControl("month1")).Text;
            logisticRegression.temperatreF = ((TextBox)div1.FindControl("tf1")).Text;
            logisticRegression.dewPointF = ((TextBox)div1.FindControl("df1")).Text;
            logisticRegression.humidity = ((TextBox)div1.FindControl("humidity1")).Text;
            logisticRegression.windDirDegrees = ((TextBox)div1.FindControl("wdd1")).Text;
            string otpt1 = Predict.PredictClassificationLR(logisticRegression);
            //Response.Write("<script language=javascript>alert('opt1 " + otpt1 + "');</script>");

            if (otpt1 != "Error")
            {
                txtResult.Text = otpt1;
                //resulta1.Text = otpt1;
            }
            break;
```

asp frontend code:

```
        <div class="content1">
            <h2>Input Dataset</h2>
            <br/>
            <asp:DropDownList name="state" ID="models" Runat="server" OnSelectedIndexChanged="Selection_Change" AutoPostBack="true">
                <asp:ListItem   value="0">Please Choose Model</asp:ListItem>
                <asp:ListItem   value="logiticRegression">Classification: Logistic Regression</asp:ListItem>
                <asp:ListItem   value="classificationNeuralNetwork">Classification: Neutral Network </asp:ListItem>
                <asp:ListItem   value="classificationDecisionForest">Classification: Decision Forest</asp:ListItem>
                <asp:ListItem   value="KNN">Classification: KNN </asp:ListItem>
                <asp:ListItem   value="predictionNeuralNetwork">Prediction: Neural Network</asp:ListItem>
                <asp:ListItem   value="decisionTree">Prediction: Decision Tree </asp:ListItem>
                <asp:ListItem   value="predictionDecisionForest">Prediction: Decision Forest</asp:ListItem>
                <asp:ListItem   value="linearRegression">Prediction: Linear Regression </asp:ListItem>
                <asp:ListItem   value="kmean">Clustering: K-mean</asp:ListItem>
                <asp:ListItem   value="hierarchialClustering">Clustering: Hierarchial Clustering </asp:ListItem>
            </asp:DropDownList>
            <div id="div1" runat="server">
                <div id="form" runat="server">
                </div>
                <div class="button-row">
                    <asp:Button ID="btnPredict" runat="server" Text="Predict" CssClass="sign-in"  OnClick="btnPredictClicked" />
                    <asp:Button runat="server" Text= "Reset" CssClass="reset"   />
                </div>
            </div>
        </div>
```

Assignment 3 - To build out a web service for the models

UI: http://adsassignment3.azurewebsites.net/



Team 12 – Priti Agarwal, Yalin Li, Zhi Li