**Advances in Data Science & Architecture**

**Assignment 3**

# Index

Goal: 3

Methodology 3

Regression 5

Classification: 14

Clustering: 24

Web Application: 31

# Goal:

To build out a web service for energy models for Finland. There will be one web service which will invoke the rest api of all the models. We have the following functions: -

1.Regression with 4 models (Linear regression, Neural Network, Random Forest and KNN) to predict the Normalized energy consumption.

2.Classification with 4 models (Logistic regression, Neural Network, Random Forest and KNN) to predict the Normalized energy consumption.

3.Clustering with 2 models (K means and Hierarchical) to predict the Building ID cluster.

We have trained the data without the year so the trained model can be used to for any year and can also be retrained for the future data. With our current capacity we will be taking the weather input and the building parameter input from the user.

For all the three we have a retraining model and a prediction model. The prediction model has been made into a web service.

# Methodology

1. We have our data from Midterm

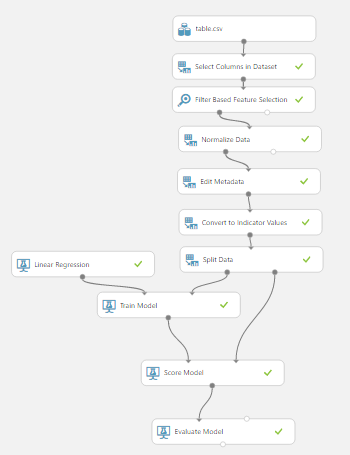
|  |  |  |
| --- | --- | --- |
| **Variables** | **Class** | **Comment** |
| Time | integer | Weather Data |
| TemperatureF | numeric |
| Dew.PointF | numeric |
| Humidity | integer |
| Sea.Level.PressureIn | numeric |
| VisibilityMPH | numeric |
| Wind.SpeedMPH | numeric |
| Gust.SpeedMPH | numeric |
| Events | integer |
| WindDirDegrees | integer |
| sta\_code | integer | Building Data |
| day | integer |
| month | integer |
| BuildingID | integer |
| meternumb | integer |
| type | integer |
| Consumption | integer |
| area\_floor \_m.sqr | integer |
| DayofWeek | integer |
| Weekday | integer |
| Peakhour | integer |
| mont | integer |
| Normalised | numeric | Dependent Variable for regression |
| BaseHourRate | integer | Dependent Variable for classification |

We have added this dataset to Azure ML

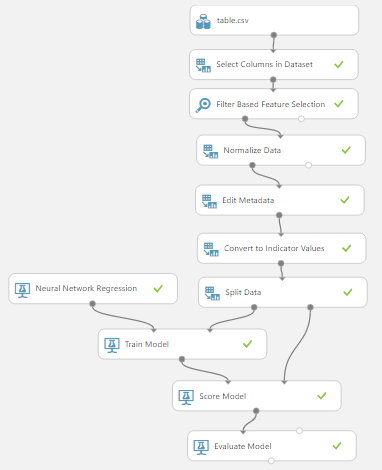
1. The next step is to make 4 predictive model

# Regression

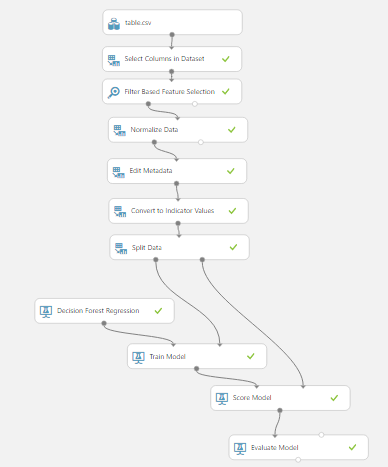
1. Linear Regression



1. Neural Network



1. Random Decision Forest



1. KNN

There is no module for KNN so we use the R script

Below mentioned: - # Map 1-based optional input ports to variables

dataset1 <- maml.mapInputPort(1) # class: data.frame

dataset2 <- maml.mapInputPort(2) # class: data.frame

library(kknn)

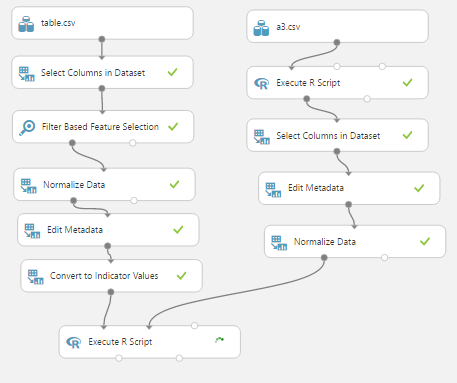
model <- train.kknn(Normalised ~ ., data = dataset1, kmax = 9)

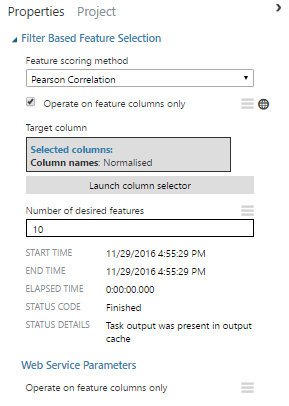
prediction <- predict(model, dataset2)

model <- train.kknn(Normalised ~ ., data = dataset1, kmax = 9)

prediction<-as.data.frame(prediction)

madel<-as.data.frame(model)

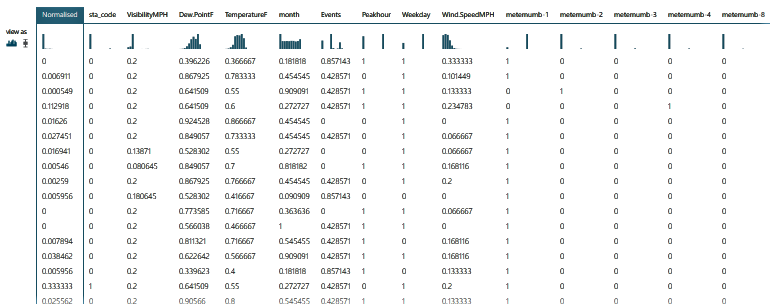
maml.mapOutputPort("prediction")



Perform feature selection for each model to reduce the independent variable and to get high co efficient of Determination for regression and Area under the curve (AUC) for Classification.

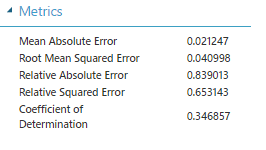
*Pearson Correlation has been chosen to eliminate collinearity.*

*The number of variables have been chosen by iteration (starting from 25 and reducing it to 10)*

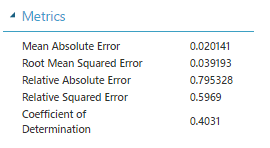
Our Model after selection is

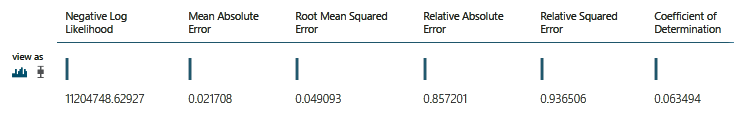
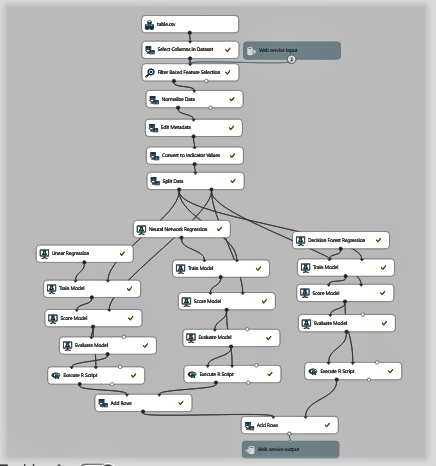
And the Coefficient of determination for

1. Linear regression

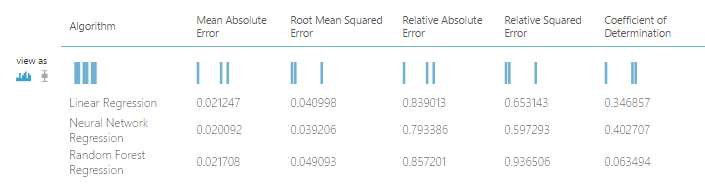


1. Neural Network



1. Random Forest
2. Next merge the 3 models into one predictive web service

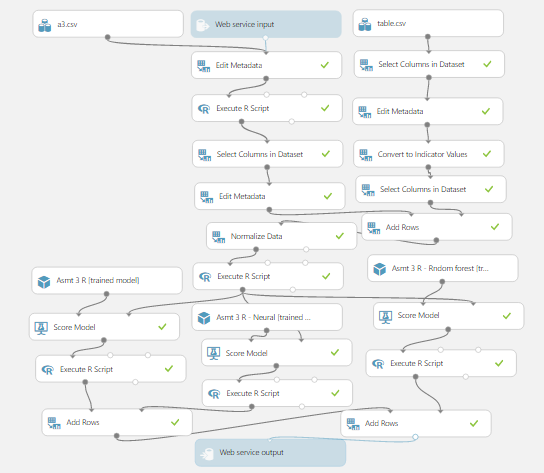
you can retrain it with new data and we get a comparative result of accuracy for all three



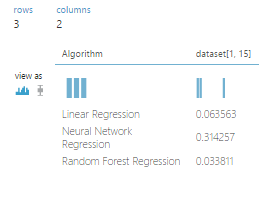
The best regression model is Neural Network as you can see from the co-efficient of Determination.

1. The next step is to convert the models into trained modules to be used for prediction.

We have modules for Linear regression, Random Forest and Neural network.



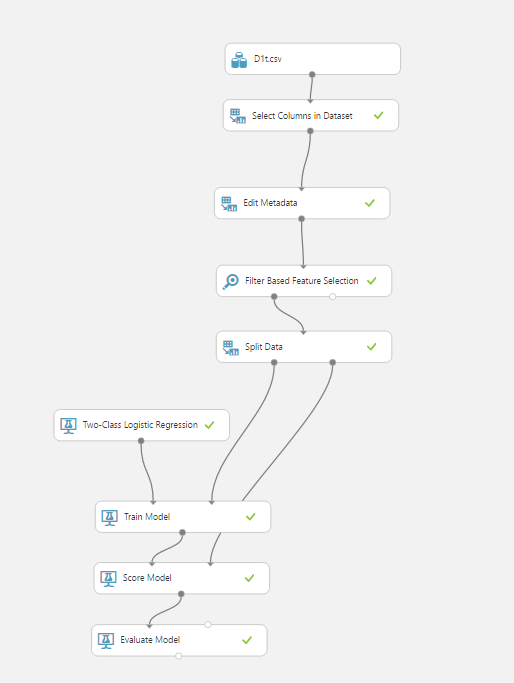
The Web Service Output will have the following output



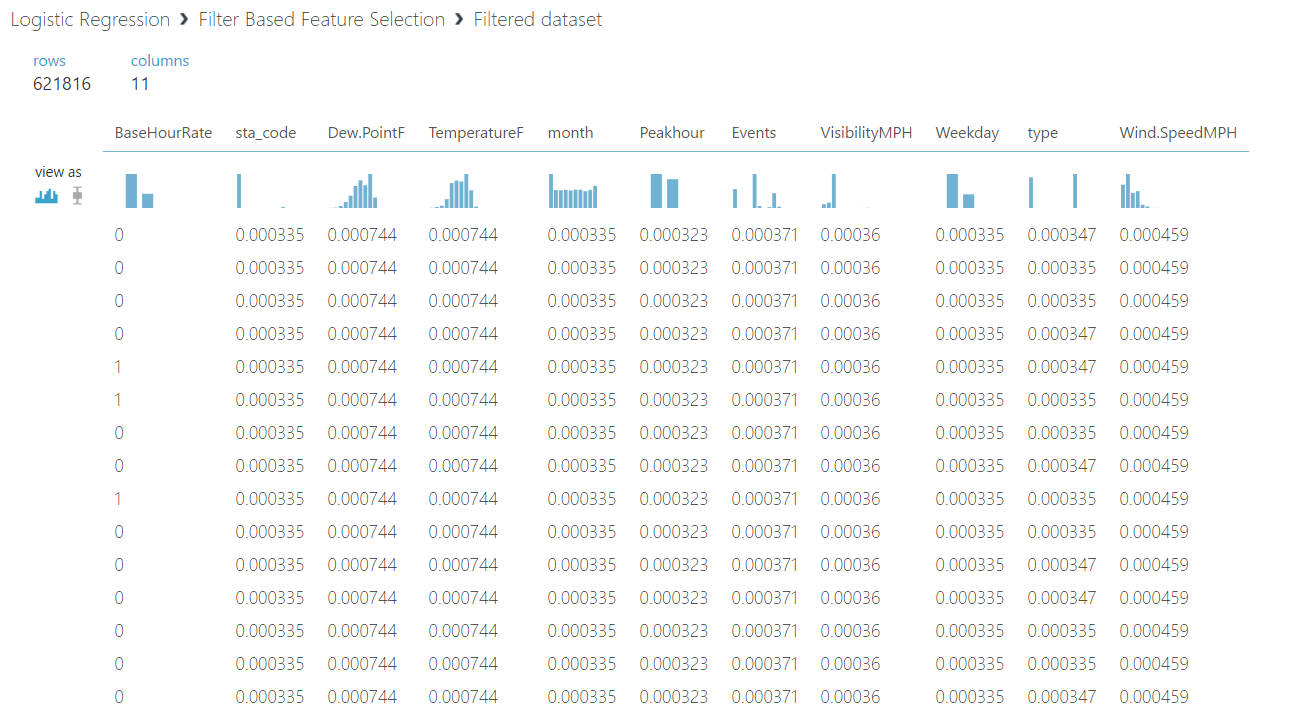
Here the data set is the predicted Normalised (consumption in KWh/Sq mt)

# Classification:

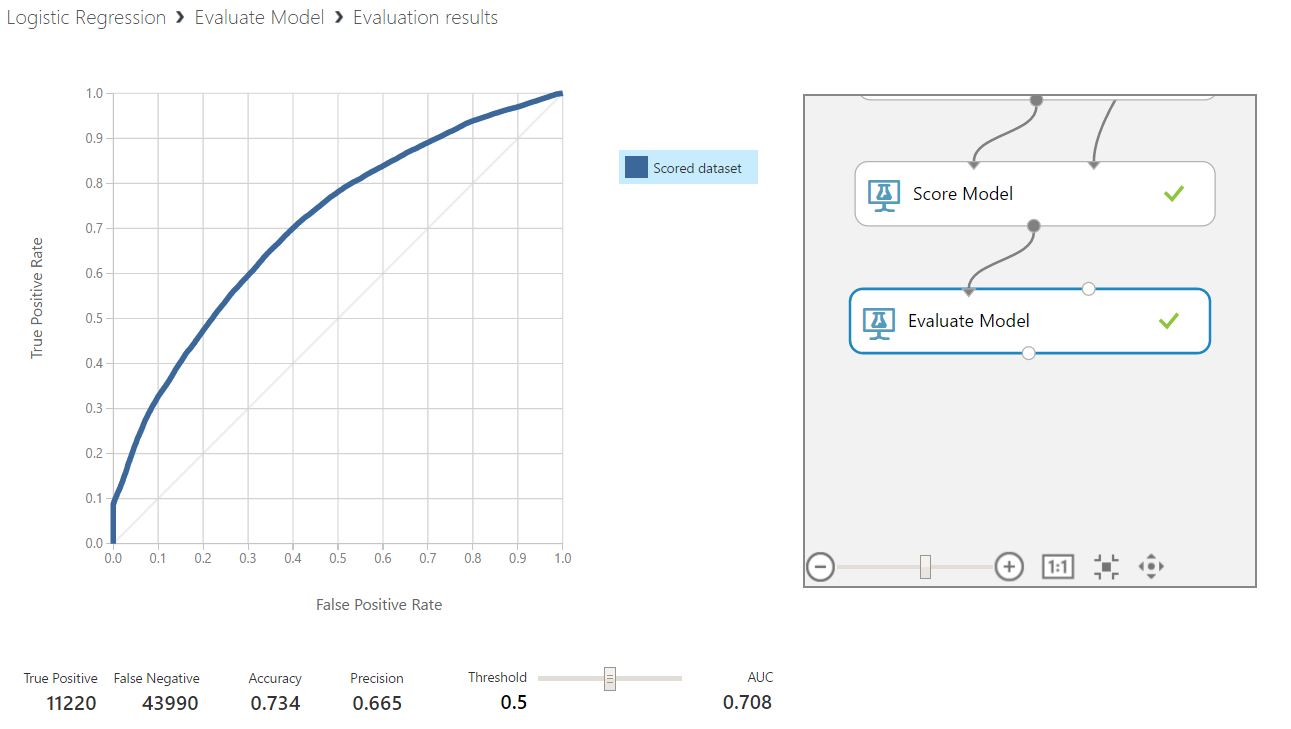
1. Logistic Regression:



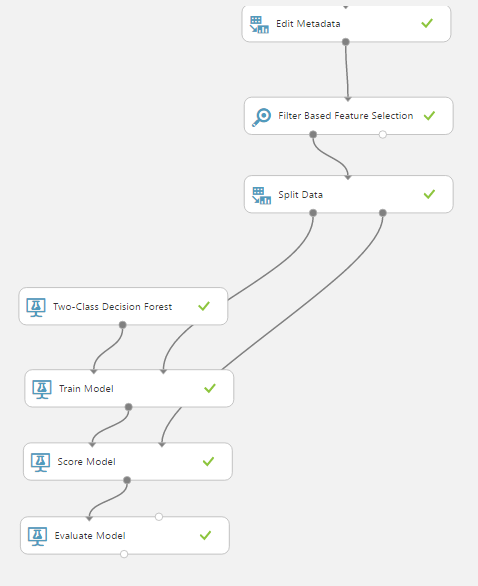
1a. Feature selection: Perform feature selection for each model to reduce the independent variable and to get high Area under the curve (AUC) for Classification. *Pearson Correlation has been chosen to eliminate collinearity. The number of variables have been chosen by iteration (starting from 25 and reducing it to 10)*



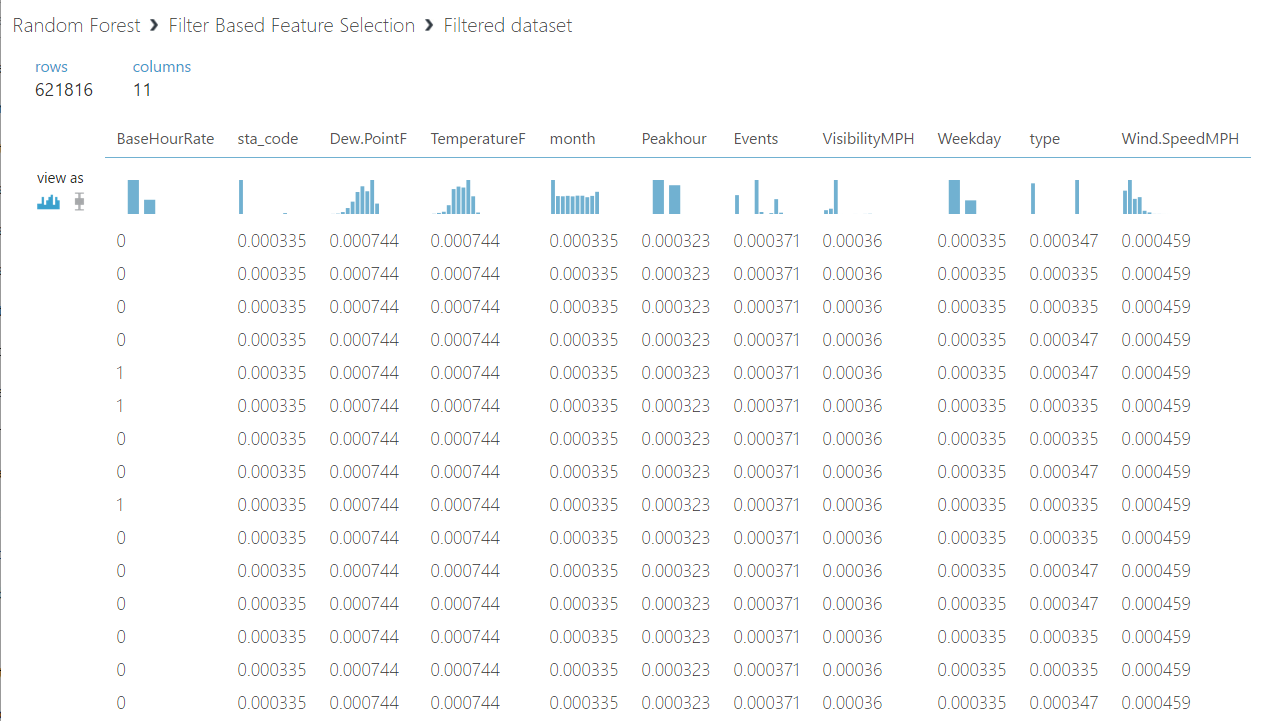
1b. Evaluation Results: A model with an AUC above 70% is considered to be a good model.



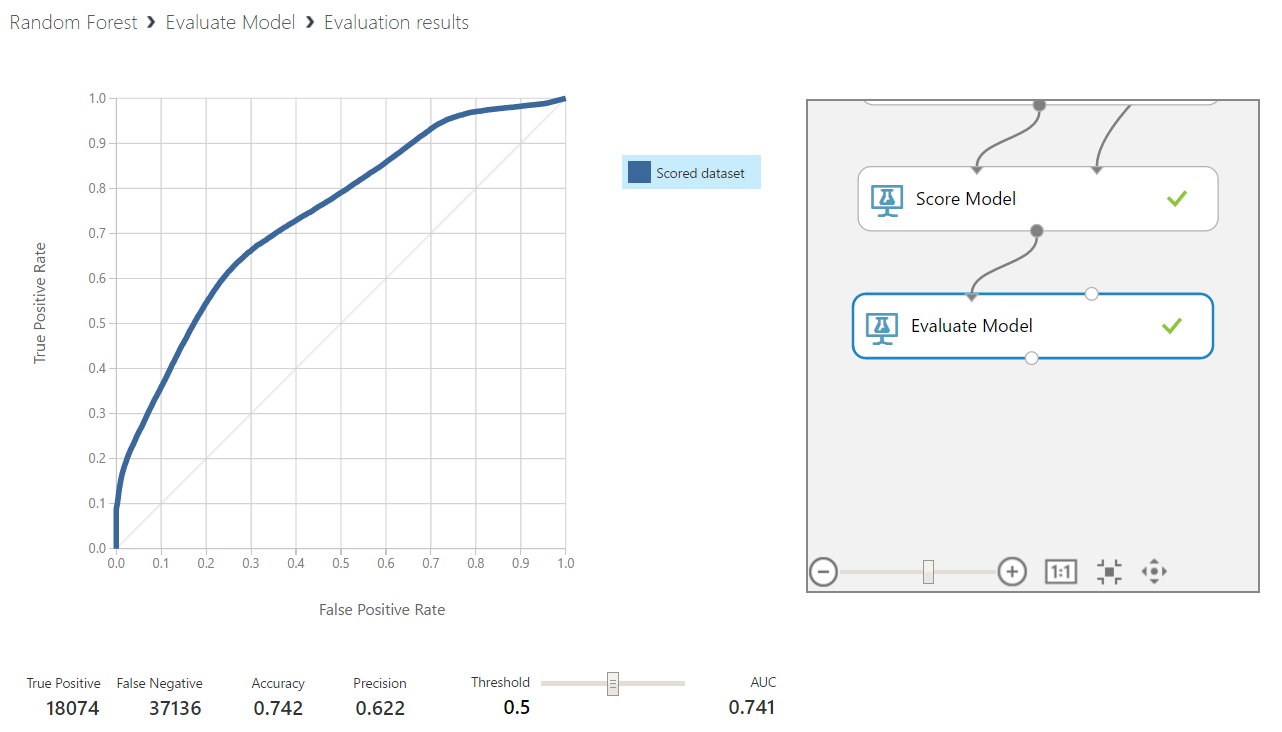
2.Decision forest:



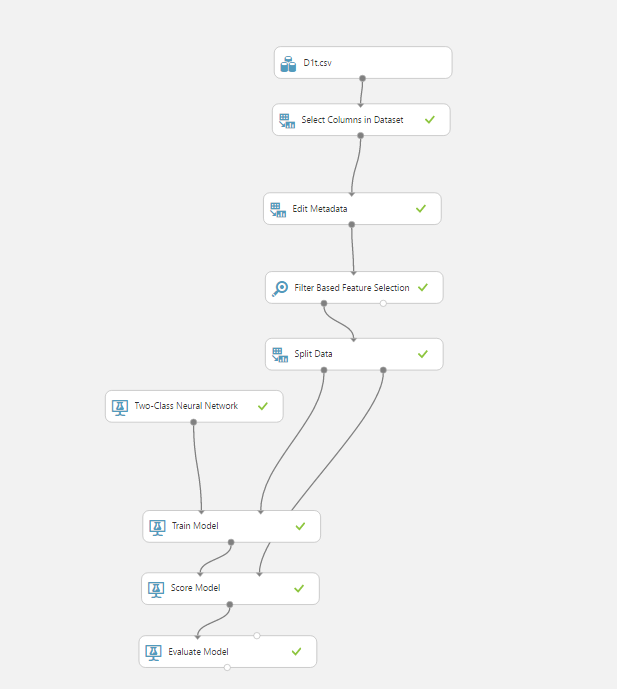
2a. Feature Selection: Perform feature selection for each model to reduce the independent variable and to get high Area under the curve (AUC) for Classification. *Pearson Correlation has been chosen to eliminate collinearity. The number of variables have been chosen by iteration (starting from 25 and reducing it to 10)*



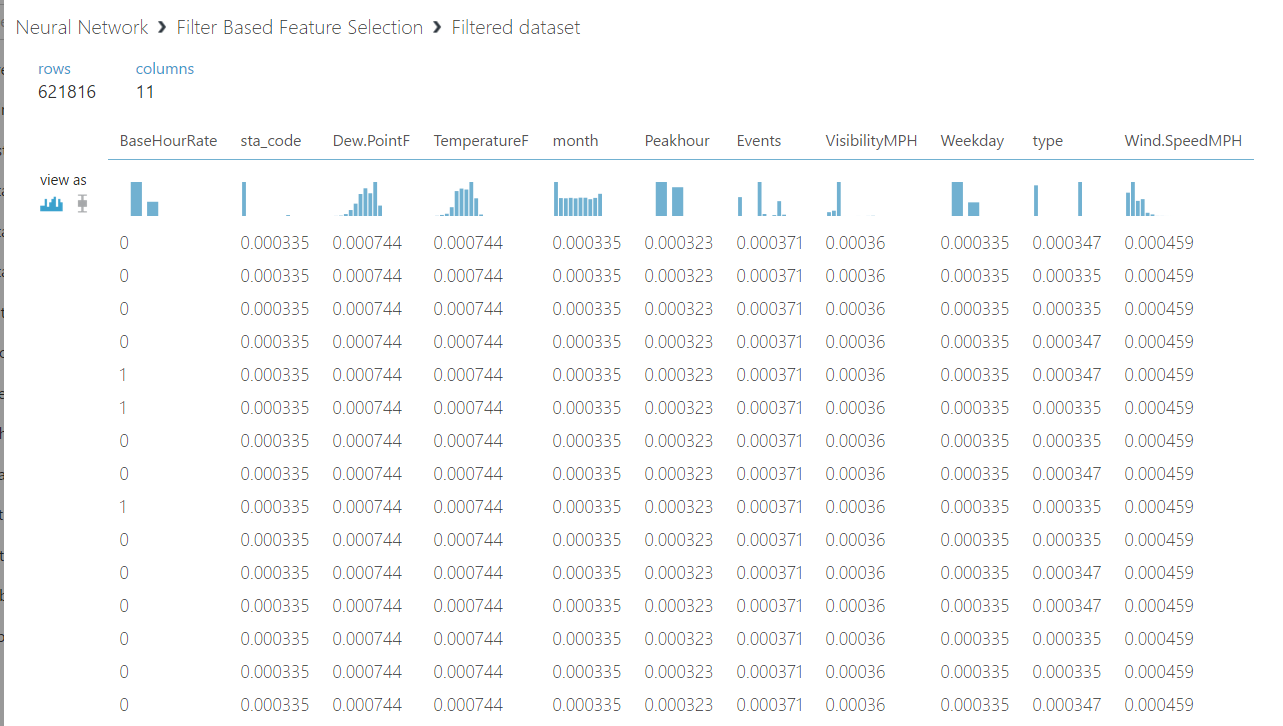
2b. Evaluation Results: A model with an AUC above 70% is considered to be a good model.



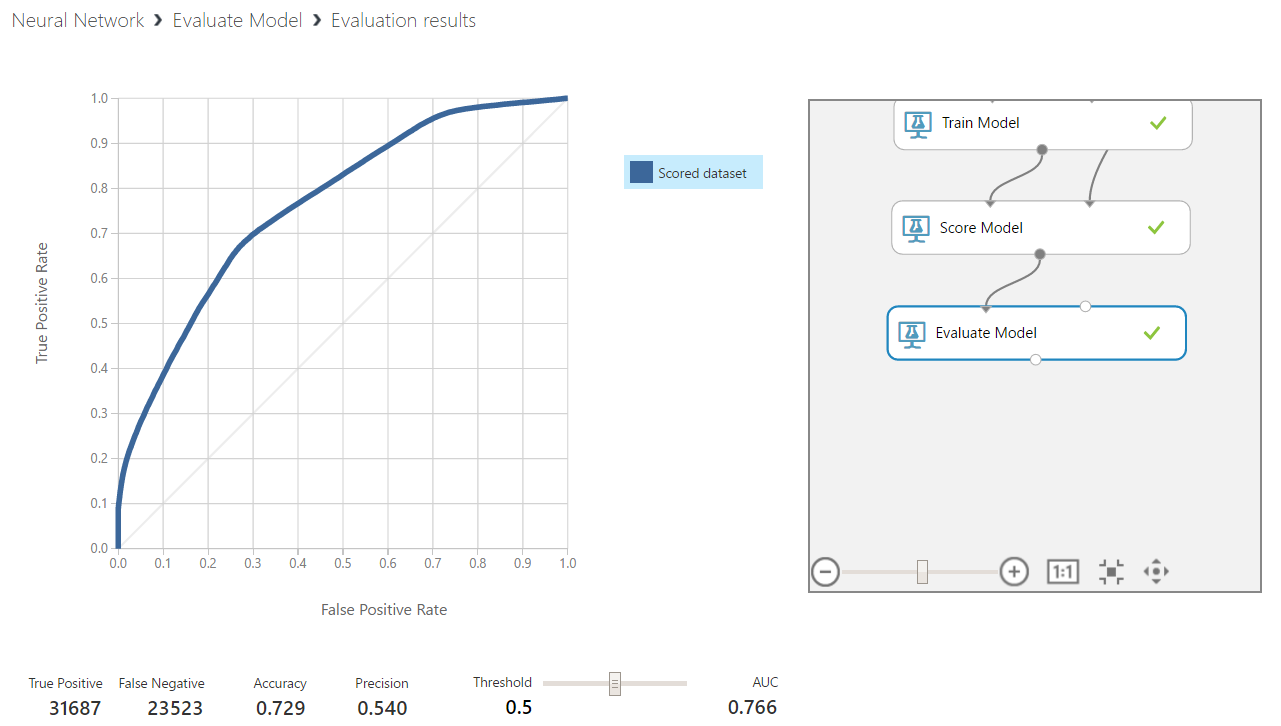
3.Neural Networks:



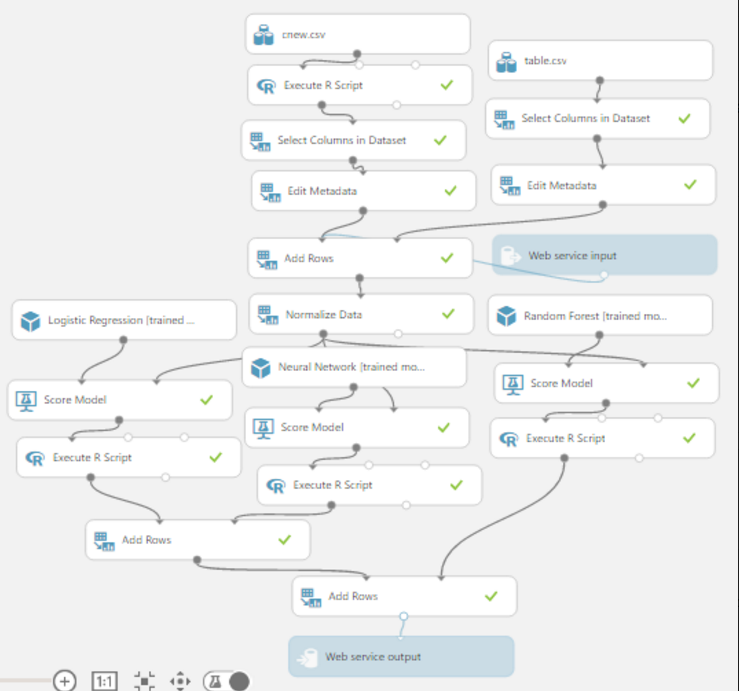
3a. Feature Selection: Perform feature selection for each model to reduce the independent variable and to get high Area under the curve (AUC) for Classification. *Pearson Correlation has been chosen to eliminate collinearity. The number of variables have been chosen by iteration (starting from 25 and reducing it to 10)*



3b. Evaluation Results: A model with an AUC above 70% is considered to be a good model.

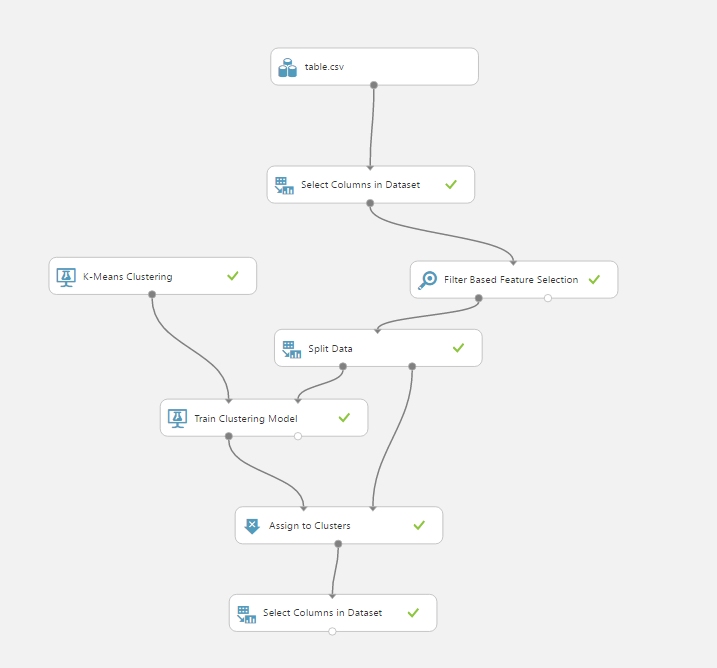


4. The next step is to convert the models into trained modules to be used for classification. We have modules for Linear regression, Random Forest and Neural network.

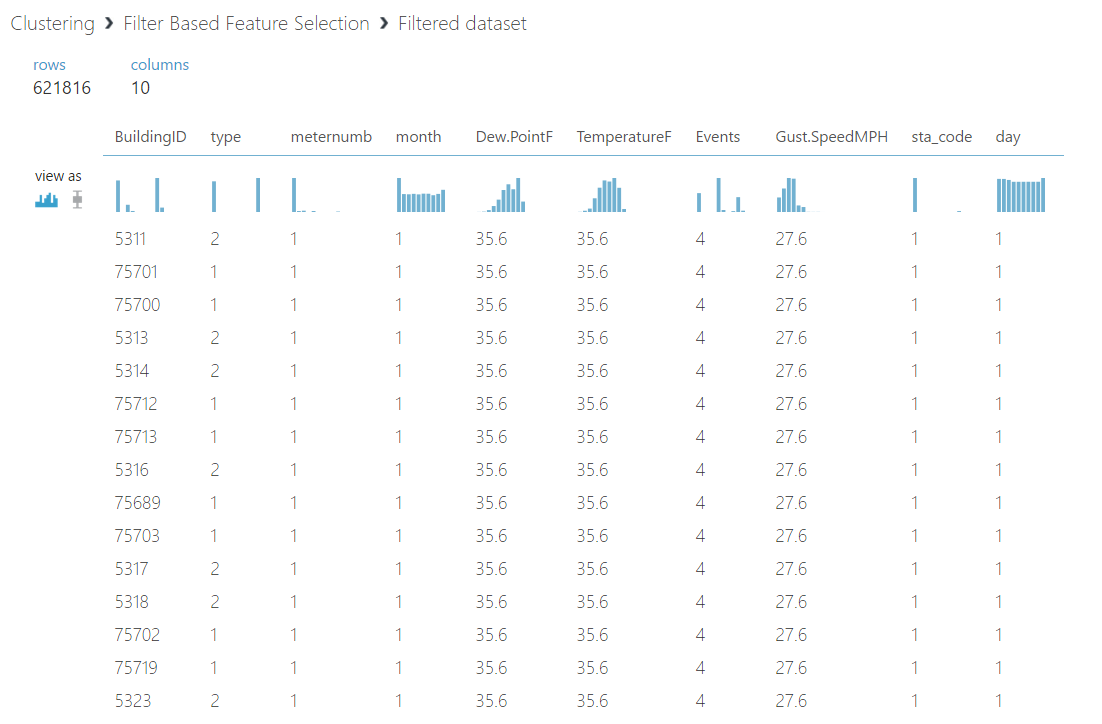


# Clustering:

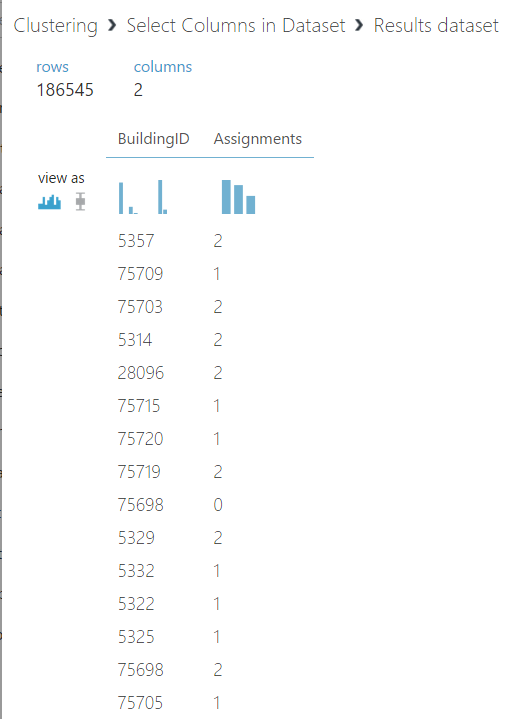
5. K means Clustering: Training Experiment



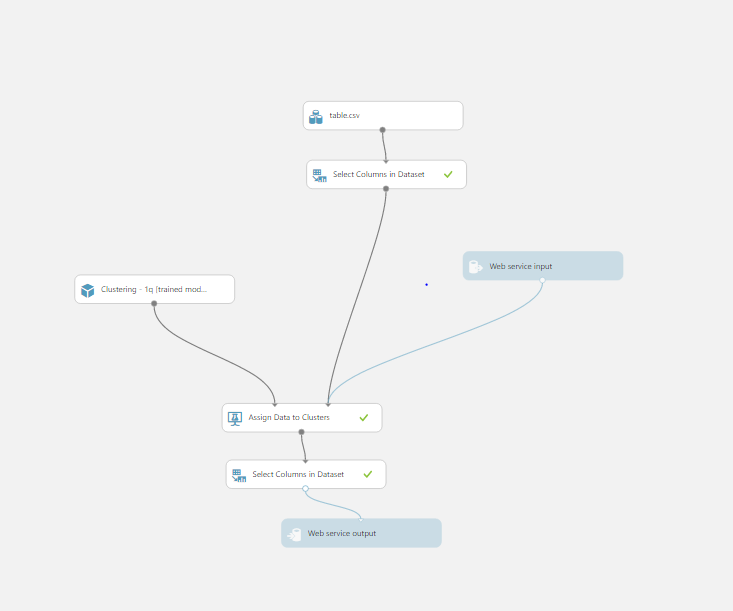
5a. Feature Selection: Perform feature selection for each model to reduce the independent variable and to get high Area under the curve (AUC) for Classification. *Pearson Correlation has been chosen to eliminate collinearity. The number of variables have been chosen by iteration (starting from 25 and reducing it to 10)*



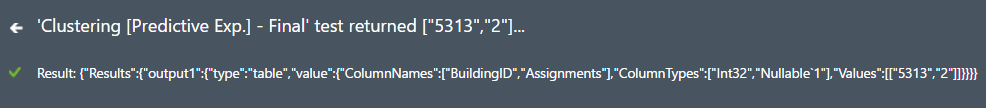
5b. Evaluation Results: The buildings are assigned to their respective clusters based on the building data.



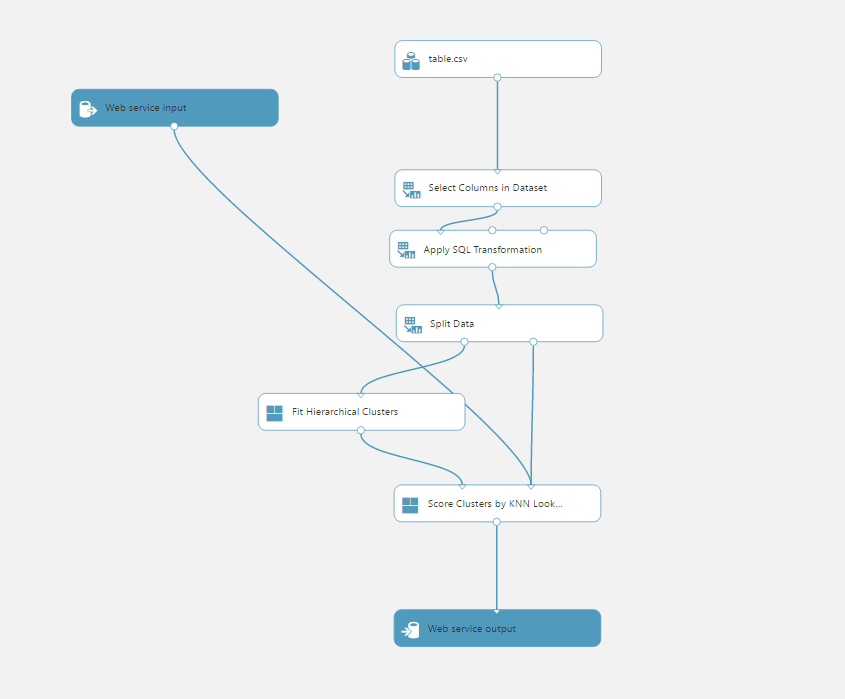
5c. K means Clustering: Predictive Experiment



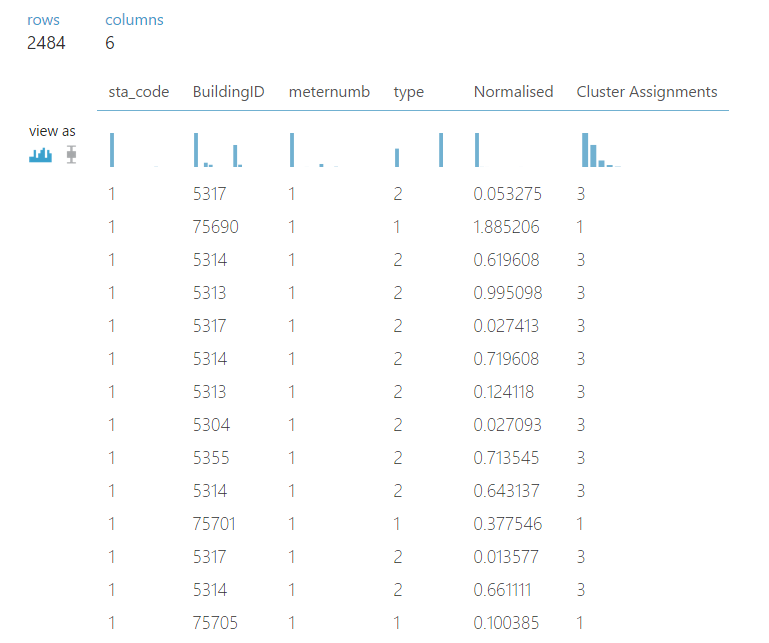
5d. We deploy the predictive experiment as a web service and enter inputs as part of the building data and the result is as shown below.



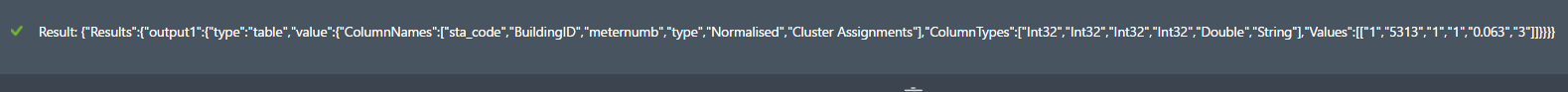
6. Hierarchical Clustering:



6a. Evaluation Results: The results produced by the Score Clusters by KNN lookup is as follows,



6b. We deploy the predictive experiment as a web service and enter inputs as part of the building data and the result is as shown below.



# Web Application:

Applications Used:

* Sublime Text 3 - Text Editor (Mac)
* Postman – Chrome Extension App (Mac)
* Terminal – Command Prompt (Mac)

Languages:

* HTML
* JavaScript
* jQuery – AJAX

Work Flow:

In ML Azure Studio,

Deploy Web Service in Azure ML Studio

Get the API Key and URL link

We have created 3 Regression Models which are deployed as a single web service. The API key and URL generated are feed in to the jQuery – Ajax function in HTML Code to get back the JSON object from ML Studio.

Code Modules:

Take input from user for the provided Text Fields

The AJAX Function – Returns a JSON object of the results from ML Azure

Select the Model

Pass the variable through the AJAX function

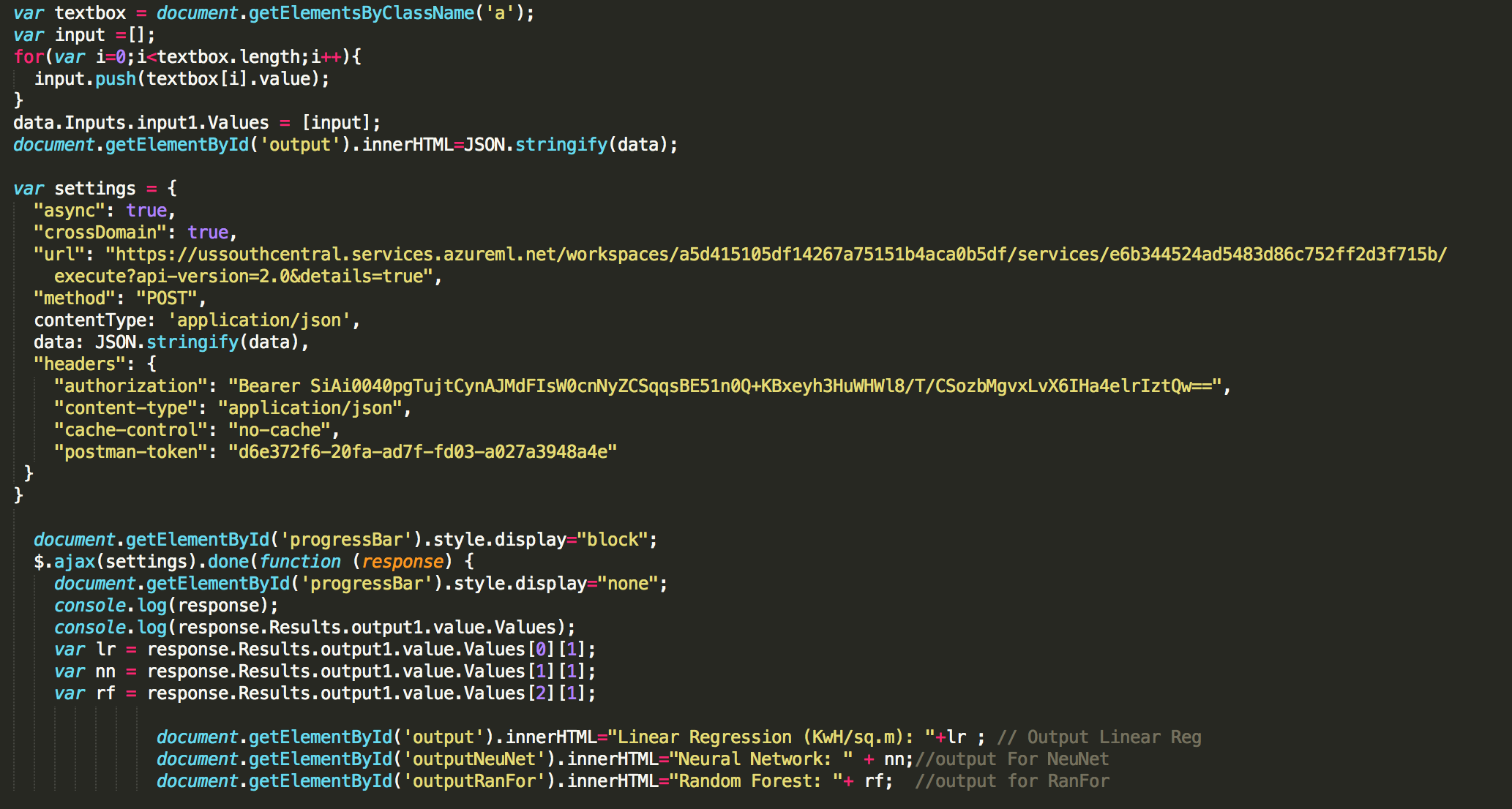
Using jQuery – AJAX Function

Make an API – Call

Points to Note:

* There are 4 HTML files (Prediction, Classification, K-means, Hierarchical Clustering).
* The functionality of all the 4 files remain the same.

CODE BREAKDOWN:



In the above code,

* var settings is the variable that is passed to the $.ajax(settings).done(function(response){}

Setting async to false means that the statement you are calling should complete before the next statement in your function can be called. If you set async: true, then that statement will begin its execution and the next statement will be called regardless of whether the async statement has completed yet.

CrossDomain is set to true for server-side redirection to another domain.

URL: From ML Azure studio

Method: Post (Denotes that it’s a Post API Request) which sends the content body (data) and content type: application / json.

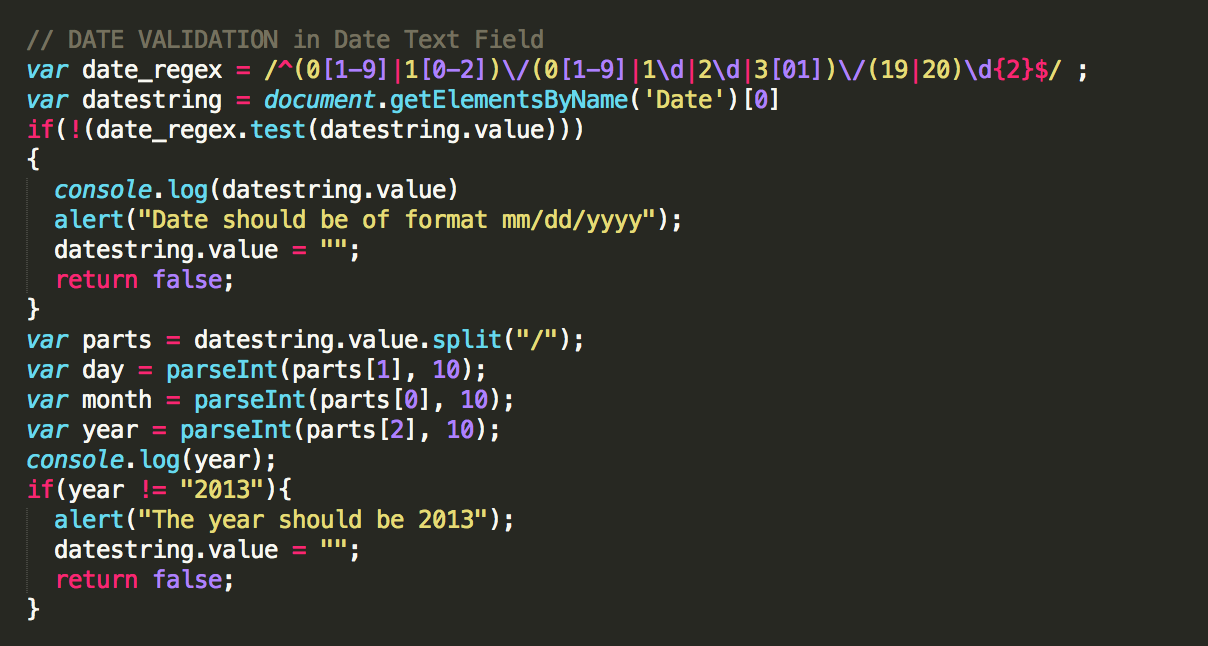
Response variable gets back the JSON object after hitting the API.



The HTML body consists of all the Input we require from the user that should be passed on to the ML azure model to get back the result for Regression.

Error Handling is done on the text field to prevent the user from typing junk data and to instruct the user to input valid data to feed it into the model.

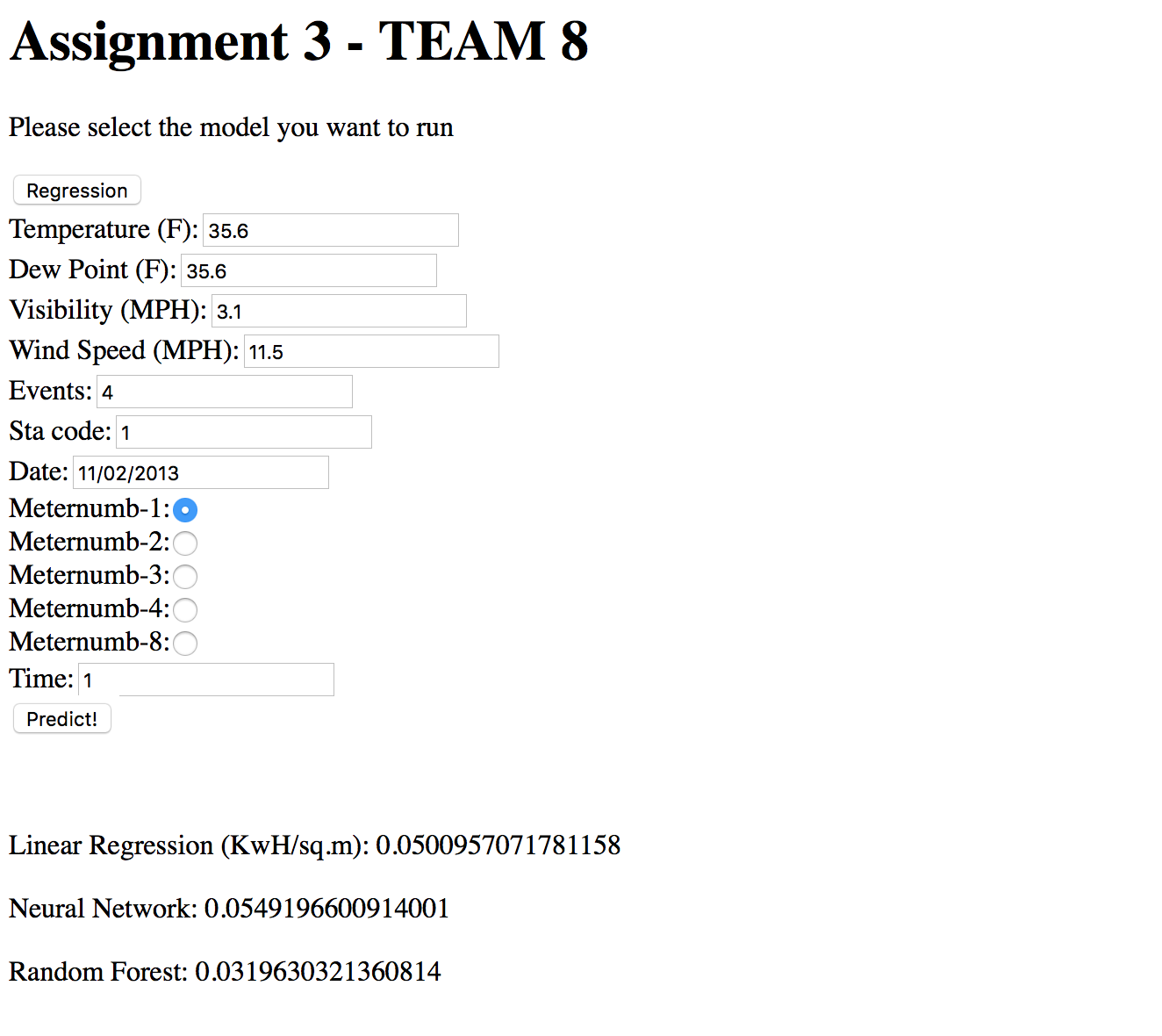


For Date Field Validation (Instructing user to use the right format and the Year being 2013) ---- OPTIONAL 

PredictionRegression() – Function that contains the data that gets passed to the ML Azure Models.

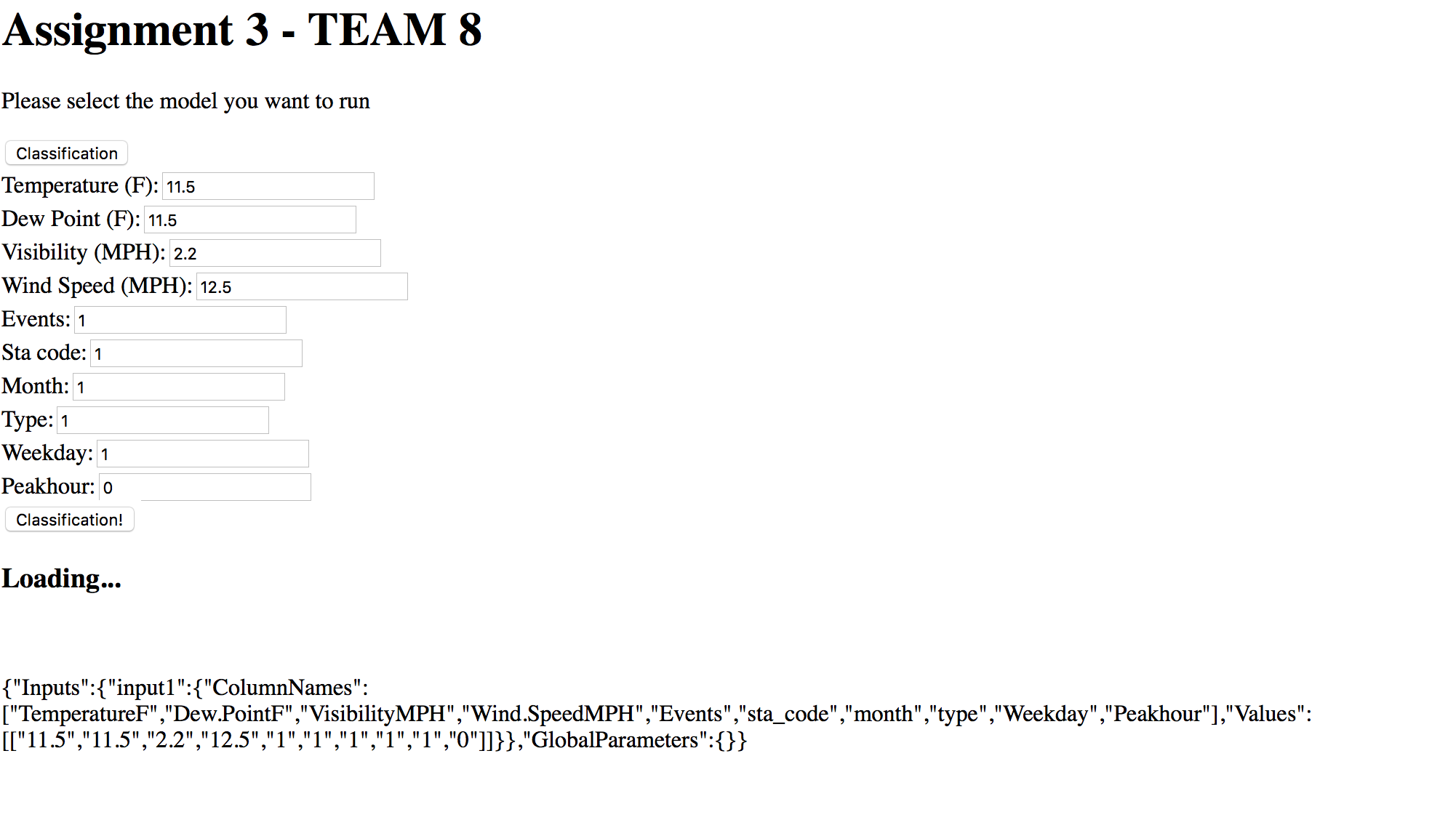
OUTPUT SNIPPETS:

PREDICTION:



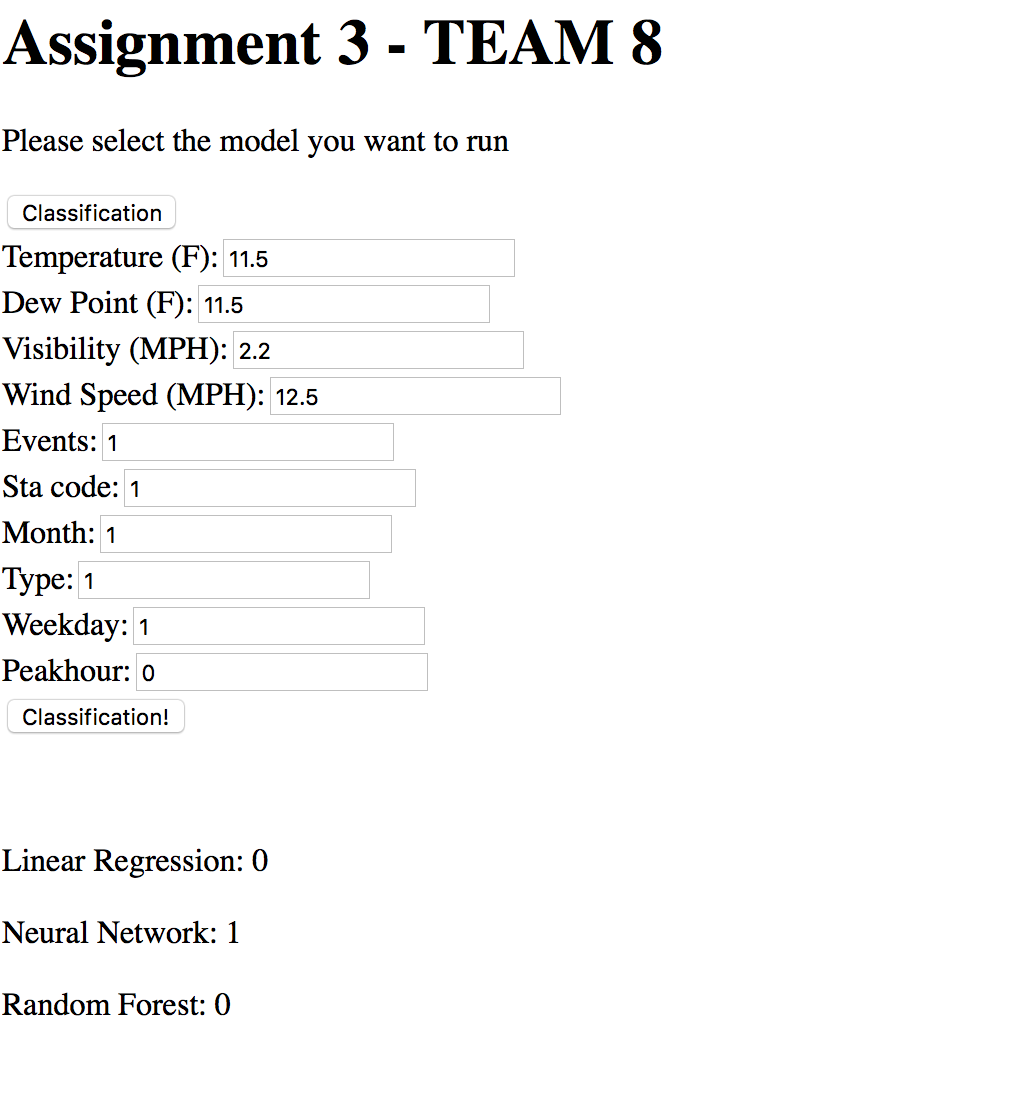
The code takes in the input and returns the three Regression models containing the dependent variable (Normalized Consumption/sq.m) for the input feed.

CLASSIFICATION:



The Loading … indicates that the HTML code is passing the parameters to the web URL to get the results back.

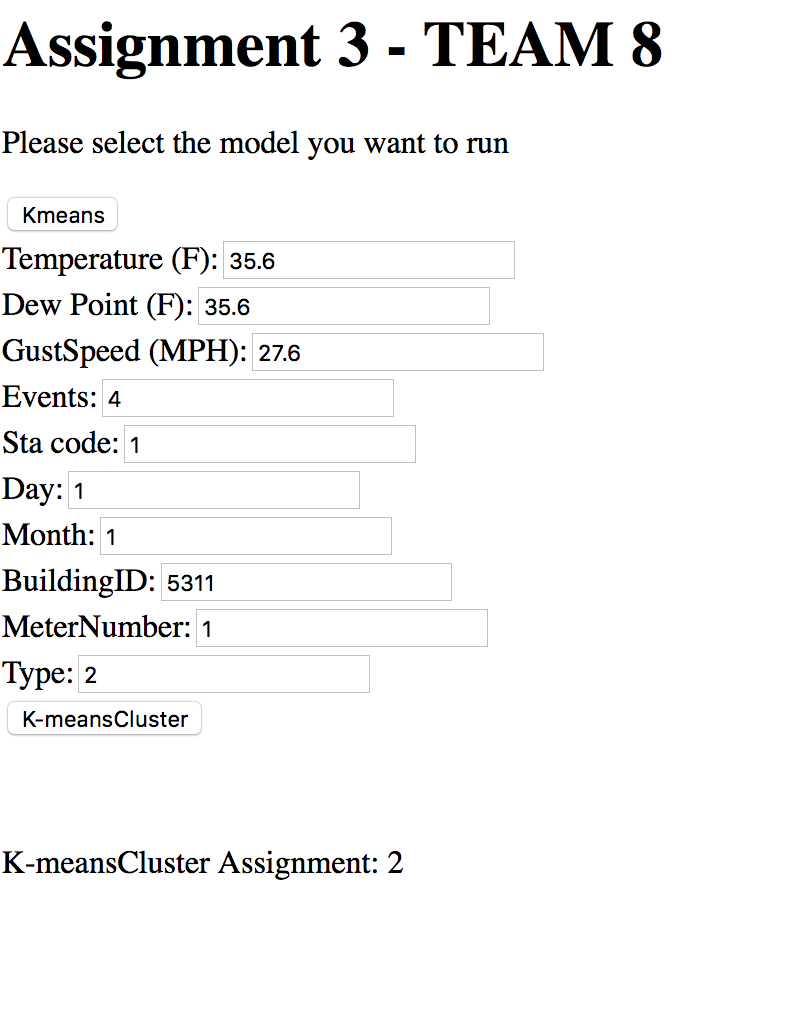
Clustering output:



As accuracy of none of the model is 100%, there will be few times when the output of all three will not be same. So, depending on AUC (area under the curve),

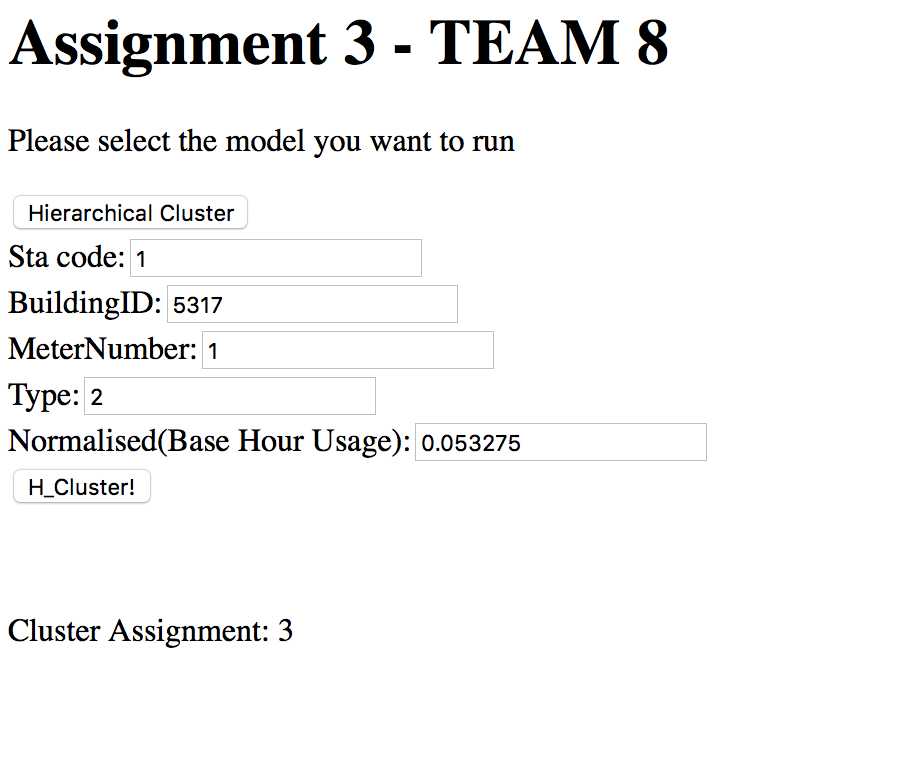
Neural Network will be accurate most of the times.

K-Means Clustering OUTPUT :



Depending on the input, it will assign the Buildings to the respective clusters based on the building data entered.

Hierarchical Cluster Output:



Team Contribution:

Ankita – Regression Models & Web service Deployment

Sriniketan – Classification & Clustering Models

Vignesh – Web application development