

Energy Efficiency Regression Data

Contents

Table of Contents

[Objectives 3](file:////Users/apple/Desktop/Company%20-%20Do%20Not%20Remove/group-i/Energy%20Efficiency%20Regression%20Document.docx#_Toc3737080)

[Descriptive Summary 4](file:////Users/apple/Desktop/Company%20-%20Do%20Not%20Remove/group-i/Energy%20Efficiency%20Regression%20Document.docx#_Toc3737081)

[Subtask - a 4](file:////Users/apple/Desktop/Company%20-%20Do%20Not%20Remove/group-i/Energy%20Efficiency%20Regression%20Document.docx#_Toc3737082)

[Problem Statement : 4](#_Toc3737083)

[Understanding Data: 4](#_Toc3737084)

[Records 5](#_Toc3737085)

[Statistics 6](#_Toc3737086)

[AML Studio Notebook Summary 6](#_Toc3737087)

[Magnitude of Coefficients 8](#_Toc3737088)

[AML Studio Steps 9](file:////Users/apple/Desktop/Company%20-%20Do%20Not%20Remove/group-i/Energy%20Efficiency%20Regression%20Document.docx#_Toc3737089)

[& 9](file:////Users/apple/Desktop/Company%20-%20Do%20Not%20Remove/group-i/Energy%20Efficiency%20Regression%20Document.docx#_Toc3737090)

[Visualisation . 9](file:////Users/apple/Desktop/Company%20-%20Do%20Not%20Remove/group-i/Energy%20Efficiency%20Regression%20Document.docx#_Toc3737091)

[Subtasks – b,c 9](file:////Users/apple/Desktop/Company%20-%20Do%20Not%20Remove/group-i/Energy%20Efficiency%20Regression%20Document.docx#_Toc3737092)

[Dataset 9](#_Toc3737093)

[Split Data 11](#_Toc3737094)

[Model Selection 11](#_Toc3737095)

[Score Model 13](#_Toc3737096)

[Evaluate Model 13](#_Toc3737097)

[Visualizing Output : 14](#_Toc3737098)

[Whole Scheme & Each Step 15](file:////Users/apple/Desktop/Company%20-%20Do%20Not%20Remove/group-i/Energy%20Efficiency%20Regression%20Document.docx#_Toc3737099)

[API Key 17](file:////Users/apple/Desktop/Company%20-%20Do%20Not%20Remove/group-i/Energy%20Efficiency%20Regression%20Document.docx#_Toc3737100)

[**API KEY :** 17](#_Toc3737101)

[**Secondary Key :** 17](#_Toc3737102)

[**Request Response :** 18](#_Toc3737103)

[**Sample Code to Test :** 18](#_Toc3737104)

# Objectives

1. Use the sample data “Energy Efficiency Regression Data” from Microsoft Azure Machine Learning Studio.
2. Use linear regression to predict the “Heating Load”.
3. Evaluate your model and provide the relevant statistics.
4. The report related to this part should include:
   1. Descriptive summary of the data
   2. Steps taken during the process of implementing the project and why you decided to take these steps
   3. Visualized output of each step.
   4. Include output of whole scheme + each step
   5. Produce an API Key or your project.

# Descriptive Summary

# Subtask - a

### Problem Statement :

In order to have a better design on structural buildings , Heating and Cooling Load play a Vitol role for maintaining better air circulations . Other than that architects should also evaluate different parameters while designing the structures like characteristics of the building . Below are few of the characters which are considered for this dataset.

*Relative Compactness, Surface Area, Wall Area, Roof Area, Overall Height, Orientation, Glazing Area, and Glazing Area Distribution.*

In order to evaluate the model, we have to consider R Squared which helps in determining how close is the data to our linear regression equation line. We will be using Linear Regression for this requirement.

### Understanding Data:

The dataset consists of 768 records and 10 features which helps in predicting the better heating load.

The dataset is considered from the AML studio sample datasets as below:

A screenshot of a cell phone

Description automatically generated

Note: Overall understanding on the data shows that we couldn’t find any missing values nor any Skewness in the dataset. Below are few of the approaches implemented to confirm the statement:

1. Created a project, added the dataset into the project and visualized: This shows we have 768 \* 10 features in this dataset.

### Records

A screenshot of a social media post

Description automatically generated

1. Descriptive analysis overall: We have analyzed the overall statistics on this dataset using Summarize Data component which helps in providing detailed analysis about each feature.

### Statistics

A screenshot of a cell phone

Description automatically generated

Note: The above information shows that all the variables are not skewed and has normalized data as both the mean and median are almost similar.

1. As AML studio doesn’t by default provide much visualization, we went ahead with creating many visualizations in understanding the data better using Notebook creation inside the AML Studio.
2. This helps in understanding the details of each feature and its correlation in much better way.

### AML Studio Notebook Summary

A screenshot of a social media post

Description automatically generated

A skyscraper and a window

Description automatically generated

The above image provides an understanding on how the dataset features are correlated. This clearly shows that Relative Compactness & Surface Area are Negatively correlated & Cooling, Heating Load are Positively Correlated.

1. It is also important to understand which features are really important and contributing for the regression line. In this process we have analyzed through AML Studio Notebooks with following Code:

### Magnitude of Coefficients

A screenshot of a cell phone

Description automatically generated

Output :

A screenshot of a cell phone

Description automatically generated

Note: The above graph clearly demonstrates that Orientation is not a required feature to consider as its impact is very minimal. So we have considered not to use Orientation feature in the model creation.

Based on all the above Descriptive Analysis below are the steps followed for implementing the model

# AML Studio Steps

# &

# Visualisation .

# Subtasks – b,c

1. Create a Project and append the dataset from the sample datasets provided by microsoft.

### Dataset

A screenshot of a cell phone

Description automatically generated

Once done with adding the dataset , create a Python Notebook with Python 3 for the dataset to analysis and visualizations. This can be implemented by clicking on the dataset and Python 3 Notebook as below :

A picture containing screenshot

Description automatically generated

This helps in understanding the descriptive statistics and summary with better visualizations.

1. Select all the columns expect the Orientation as this features is not impacting the model much as seen in the above Descriptive Analysis.

A picture containing object

Description automatically generated

A screenshot of a cell phone

Description automatically generated

1. As the dataset is clean and not much of preprocessing is required we can directly split the data into Training and Testing Datasets using Split Data Component.

### Split Data

A close up of a womans face

Description automatically generated

Note : While splitting the data the most important aspect is to consider selecting target variable as Stratified Column which helps in creating a balanced training and testing datasets while splitting

A screenshot of a cell phone

Description automatically generated

1. As this project requires to use Linear Regression model , the linear regression model component is included into the project.

### Model Selection

A screenshot of a cell phone

Description automatically generated

Considering the Hypertuning Parameter components and its results below are the indices we considered for the linear regression .

A screenshot of a cell phone

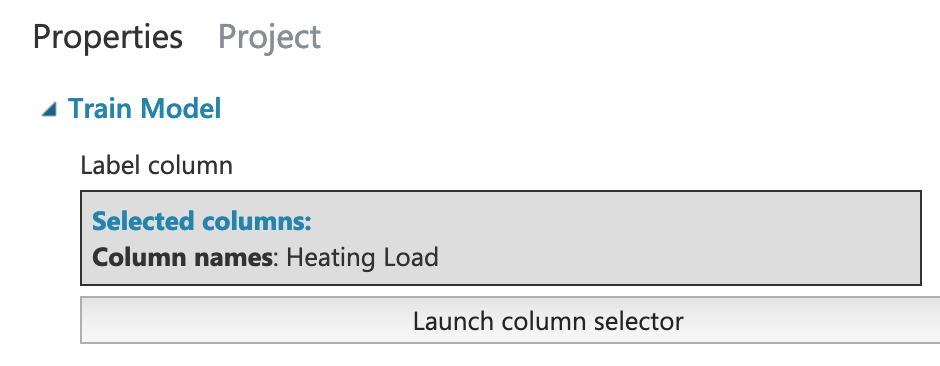
Description automatically generated

1. Considering Linear Regression Model Component and Train Dataset from split data as Input sources to Train a Model , Train Model component will be integrated.

A screenshot of a cell phone

Description automatically generated

Train model takes the target feature to get trained upon.



1. Inorder to score the dataset , Score Model component takes the Test Data to get the results on trained model

### Score Model

A screenshot of a cell phone

Description automatically generated

1. Once the model is scored this can be evaluated using Evaluate Model Component

### Evaluate Model

A screenshot of a cell phone

Description automatically generated

### Visualizing Output :

Root Mean Squared Error gives the clarity on deviation of your data with respect to the Regression Line.

A screenshot of a cell phone

Description automatically generated

# Whole Scheme & Each Step

Subtask - d

whole scheme + each step

The whole scheme design looks as below :

A close up of a map

Description automatically generated

After generating the Webservice :

A screenshot of a cell phone

Description automatically generated

# API Key

Subtask - e

Once the deployment is completed the AML Studio generates an API Key which can be consumed through any of the services and can be testing using C#, Python …

A screenshot of a social media post

Description automatically generated

### **API KEY :**

VZDWr3hxmItmxw5OEEVS2O/aa+sJnyWEINC3Y+0odB/PitlAleuLUP7NRFqizcveP+FwkT1PH5vnIuocYUMcXQ==

### **Secondary Key :**

r86QB0BH1tagi5rwy749965R9xb8Qh4rlBhEjBKht6zTqq3/iA5Ptv+GjErL5H3XDW7H0N2fKuuRzRF9D4AdtA==

### **Request Response :**

<https://ussouthcentral.services.azureml.net/workspaces/0791fbe1c5254bca93cbfb90611752c5/services/e0927203ba4641a38898150a57043c7a/execute?api-version=2.0&format=swagger>

**Batch Requests :**

<https://ussouthcentral.services.azureml.net/workspaces/0791fbe1c5254bca93cbfb90611752c5/services/e0927203ba4641a38898150a57043c7a/jobs?api-version=2.0>

### **Sample Code to Test :**

Pass the API Key into this code .

*import urllib.request*

*import json*

*data = {*

*"Inputs": {*

*"input1":*

*[*

*{*

*'Relative Compactness': "1",*

*'Surface Area': "1",*

*'Wall Area': "1",*

*'Roof Area': "1",*

*'Overall Height': "1",*

*'Orientation': "1",*

*'Glazing Area': "1",*

*'Glazing Area Distribution': "1",*

*'Heating Load': "1",*

*'Cooling Load': "1",*

*}*

*],*

*},*

*"GlobalParameters": {*

*}*

*}*

*body = str.encode(json.dumps(data))*

*url = 'https://ussouthcentral.services.azureml.net/workspaces/0791fbe1c5254bca93cbfb90611752c5/services/e0927203ba4641a38898150a57043c7a/execute?api-version=2.0&format=swagger'*

*api\_key = 'abc123' # Replace this with the API key for the web service*

*headers = {'Content-Type':'application/json', 'Authorization':('Bearer '+ api\_key)}*

*req = urllib.request.Request(url, body, headers)*

*try:*

*response = urllib.request.urlopen(req)*

*result = response.read()*

*print(result)*

*except urllib.error.HTTPError as error:*

*print("The request failed with status code: " + str(error.code))*

*# Print the headers - they include the requert ID and the timestamp, which are useful for debugging the failure*

*print(error.info())*

*print(json.loads(error.read().decode("utf8", 'ignore')))*