# Creating the service

* Log into the Azure Portal at portal.azure.com
* Click on “Create a Resource”
* Search for “Anomaly Detector”
* Click Create
* Enter a unique name, subscription, etc. You can use the free pricing tier. When all information is entered click on confirm and create
* When the service is created go to the quick start tab and take note of the key and endpoint

# Calling the service to detect if the last data point is an outlier

* Open the Lab5 project
* Go to the Constants\AzureML.cs file and copy in the key and endpoint from the service into the AnomalyApiKey and AnomalyEndpoint variables
* Ensure the following Nuget packages are installed:
  + Microsoft.Azure.CognitiveServices.AnomalyDetector (this is in preview)
  + Newtonsoft.Json
* In the HomeController’s CheckLatestSat method add the following code:

// Create a client object to communicate with our service

var client = new AnomalyDetectorClient(new ApiKeyServiceClientCredentials(AzureML.AnomalyApiKey))

{

Endpoint = AzureML.AnomalyEndpoint

};

// Get a predefined list of datapoints

var dataPoints = GetAverageMathSatScore();

// Add a new datapoint for the next year with the entered value

dataPoints.Add(new Point { Timestamp = DateTime.Parse("1/1/2020"), Value = mathSat });

// Create our request with all the data

var request = new Request(dataPoints, Granularity.Yearly, 1);

// Get the result, note it is dynamically creating the model

var result = await client.LastDetectAsync(request);

var jsonResults = JsonConvert.SerializeObject(result);

// Create our model to display results to the user

return new IndexModel

{

MathSat = mathSat,

LastGpaResult = jsonResults

};

* Try running the project, enter a 2020 average SAT score and press the “Check Math SAT” button

If you try different numbers, do you notice that the number that will kick off an anomaly seems to change from what you think it should be? If so, why do you think that is happening?

# Looking at a Historical Series to Find Anomalies

* In the HomeController’s CheckLatestSat method add the following code:

// Create a client object to communicate with our service

var client = new AnomalyDetectorClient(new ApiKeyServiceClientCredentials(AzureML.AnomalyApiKey))

{

Endpoint = AzureML.AnomalyEndpoint

};

// Get the NOLA extreme snowfall datapoints for Albany

var dataPoints = GetExtremeSnowDepthDaysInAlbany();

// Create our request with all the data

var request = new Request(dataPoints, Granularity.Yearly);

// Get the result, note it is dynamically creating the model

var result = await client.EntireDetectAsync(request);

var jsonResults = JsonConvert.SerializeObject(result);

// Create our model to display results to the user

return new IndexModel

{

AllAnomoliesResult = jsonResults

};

* Run the project and press the “Snow Day Anomolies” button

Notice that each year is calculated with the expected value, and expected deviations. There is one year that it found an anomaly, 1969 with 36 days of extreme snow depth. 1970 has the same extreme maximum snow depth of 36 inches but isn’t being flagged as an anomaly. Why do you think that is?