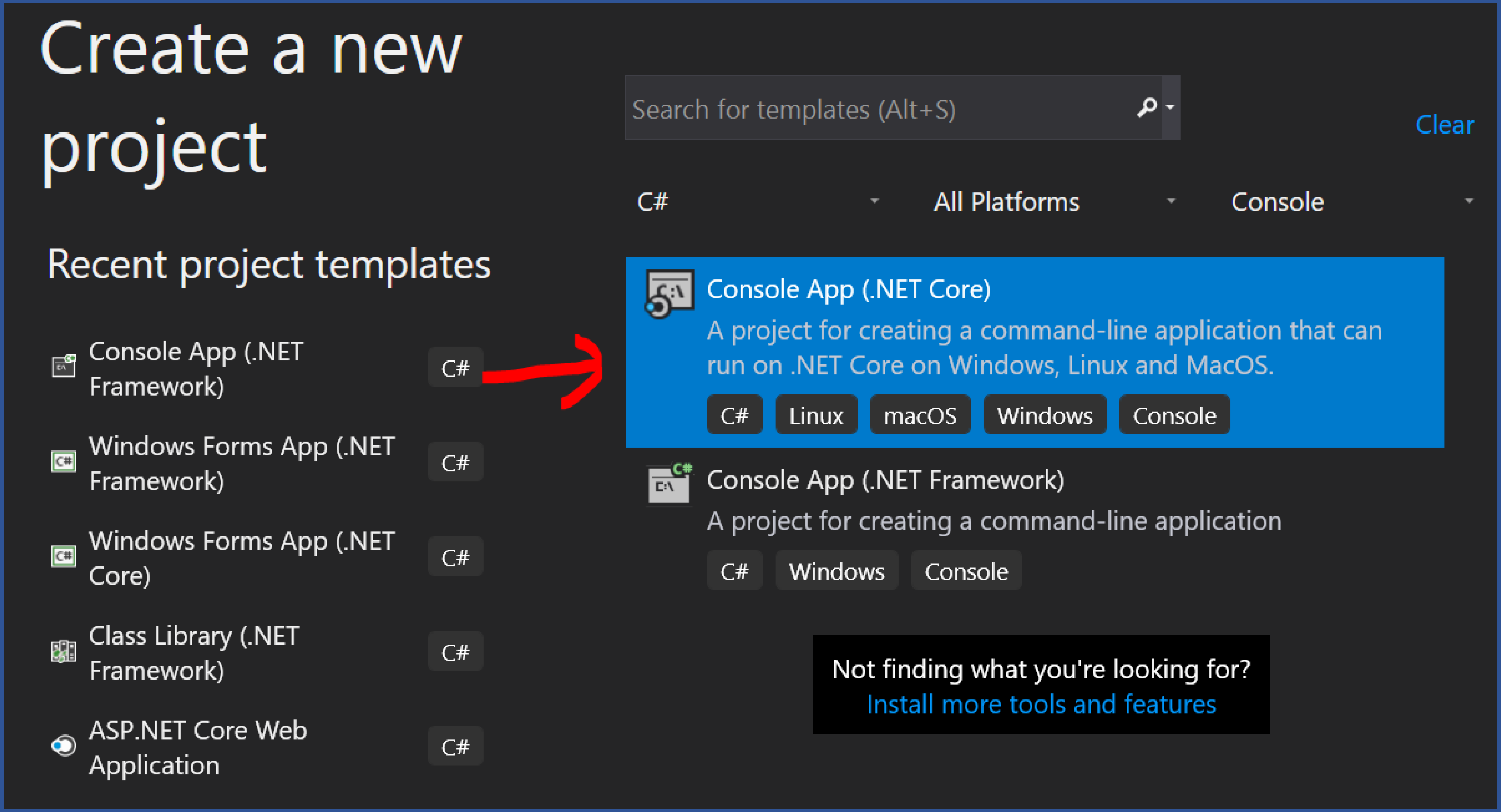
Predict from IoT

What's in this session?

1. Create new .NET CORE console project
2. Add NuGet
3. Copy data models to project
4. Add base class reference
5. Add using to Program
6. Add ML Model and test data to mib folder
7. Add class fields
8. Add method GetPrediction
9. Add method GetD2CMessage
10. Add code to Main
11. Run program and verify the result

Create new .NET CORE console project



Create a new project

Search for templates (Alt+S)

Clear

C# | All Platforms | Console

Recent project templates

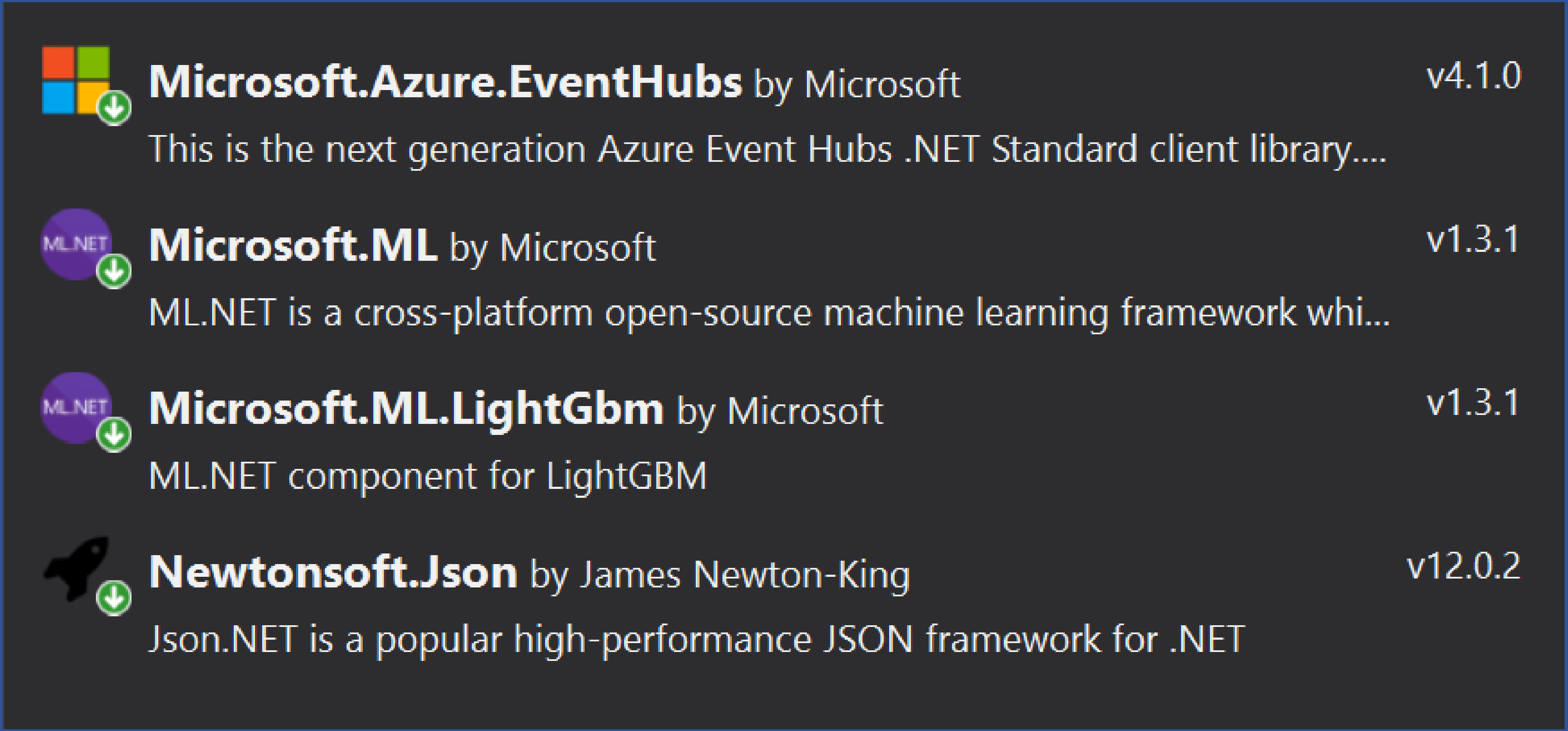
- Console App (.NET Framework) C#
- Windows Forms App (.NET Framework) C#
- Windows Forms App (.NET Core) C#
- Class Library (.NET Framework) C#
- ASP.NET Core Web Application C#

Console App (.NET Core)
A project for creating a command-line application that can run on .NET Core on Windows, Linux and MacOS.
C# Linux macOS Windows Console

Console App (.NET Framework)
A project for creating a command-line application
C# Windows Console

Not finding what you're looking for?
Install more tools and features

Add NuGet



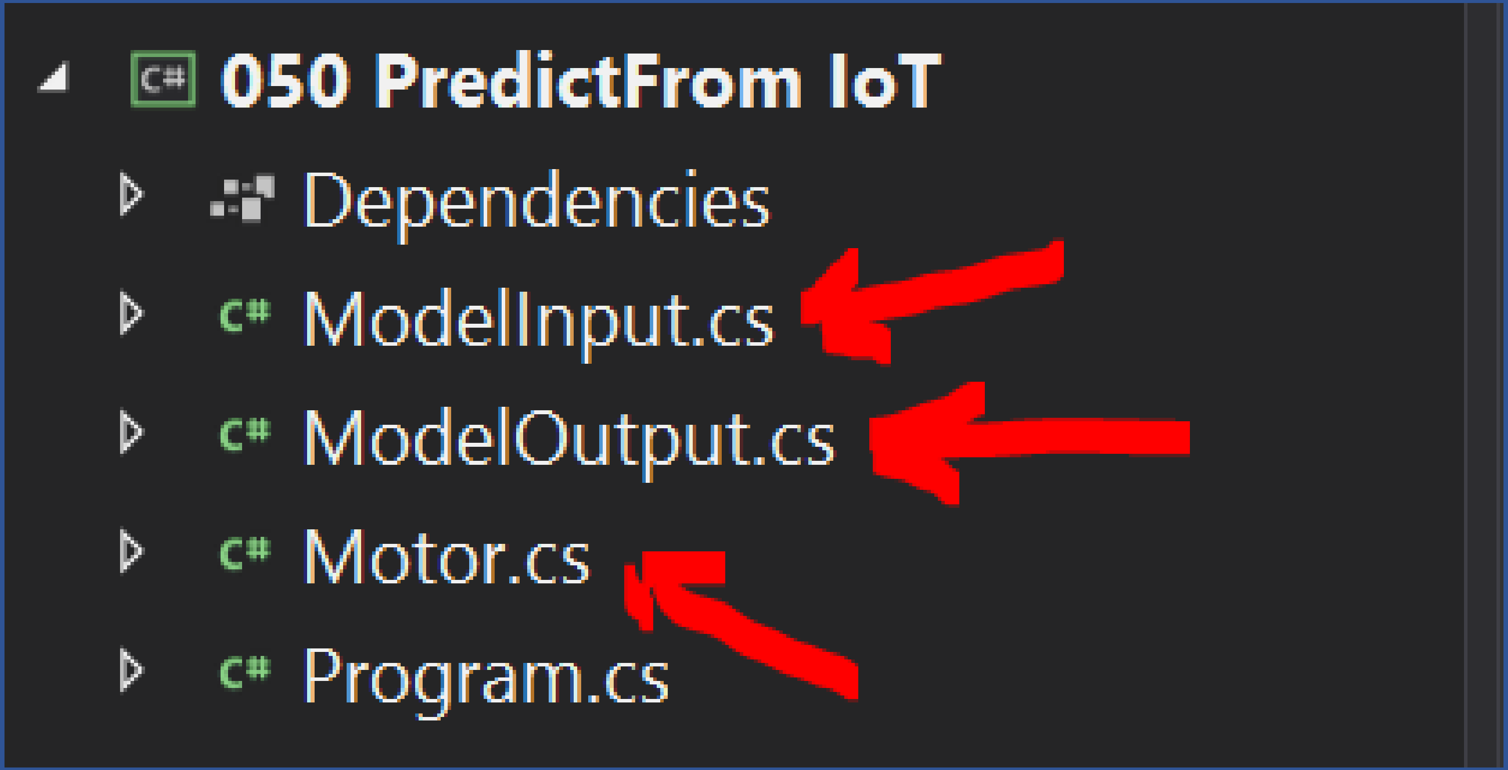
Microsoft.Azure.EventHubs by Microsoft v4.1.0
This is the next generation Azure Event Hubs .NET Standard client library....

Microsoft.ML by Microsoft v1.3.1
ML.NET is a cross-platform open-source machine learning framework whi...

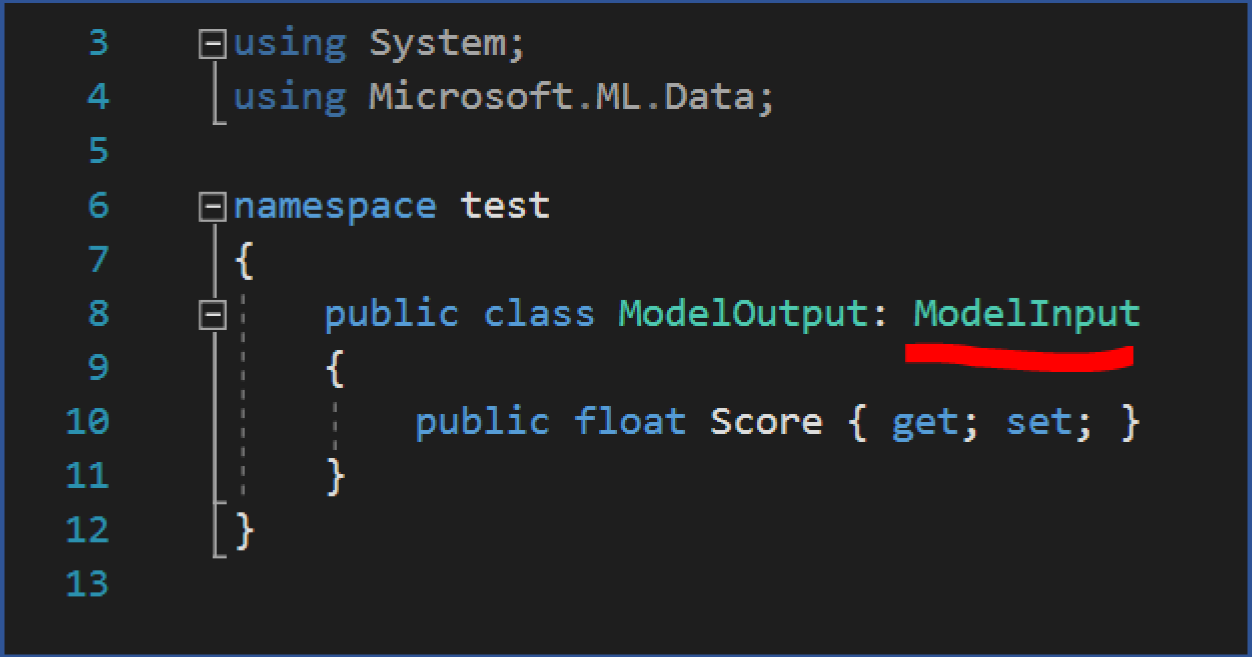
Microsoft.ML.LightGbm by Microsoft v1.3.1
ML.NET component for LightGBM

Newtonsoft.Json by James Newton-King v12.0.2
Json.NET is a popular high-performance JSON framework for .NET

Copy data models to project

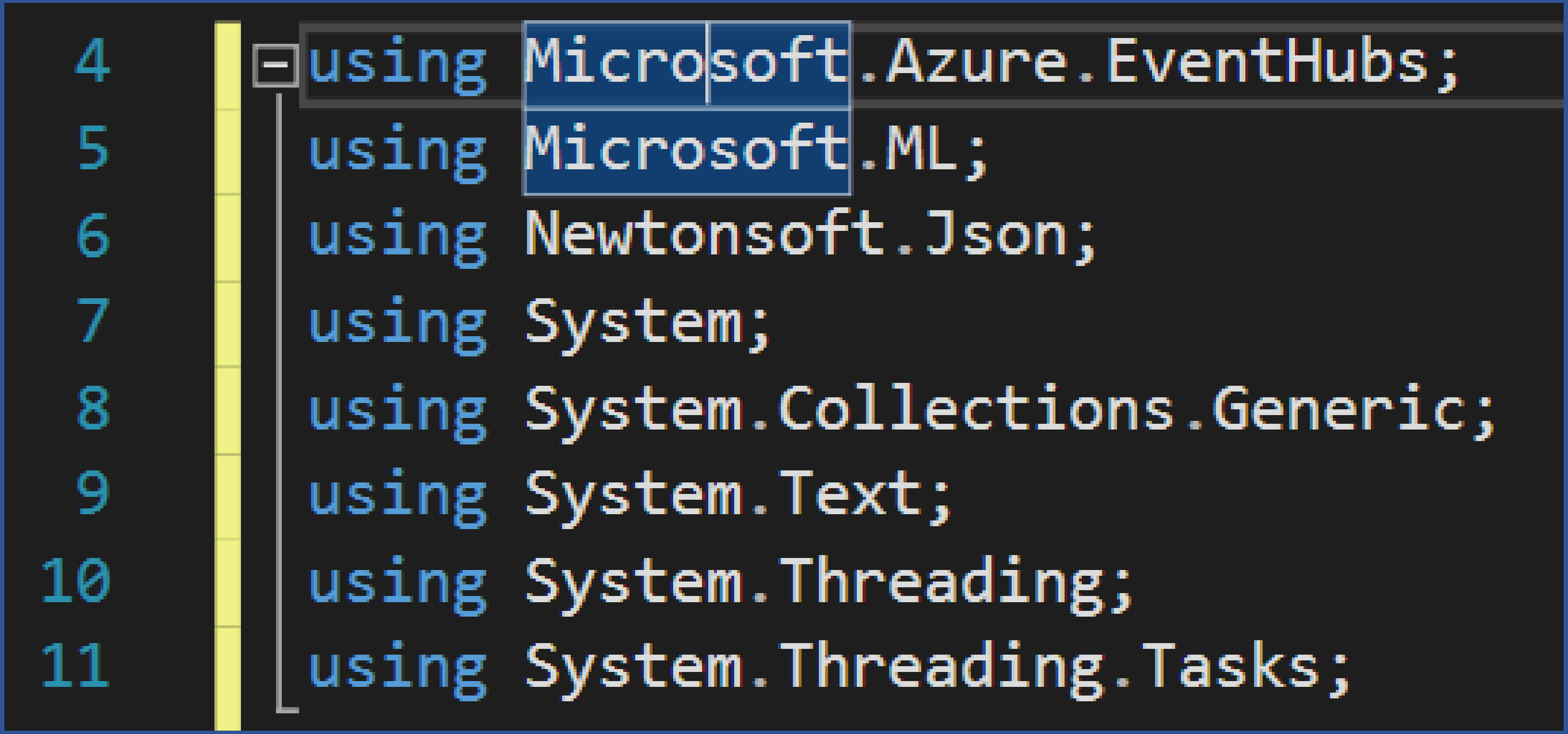


Add base class reference



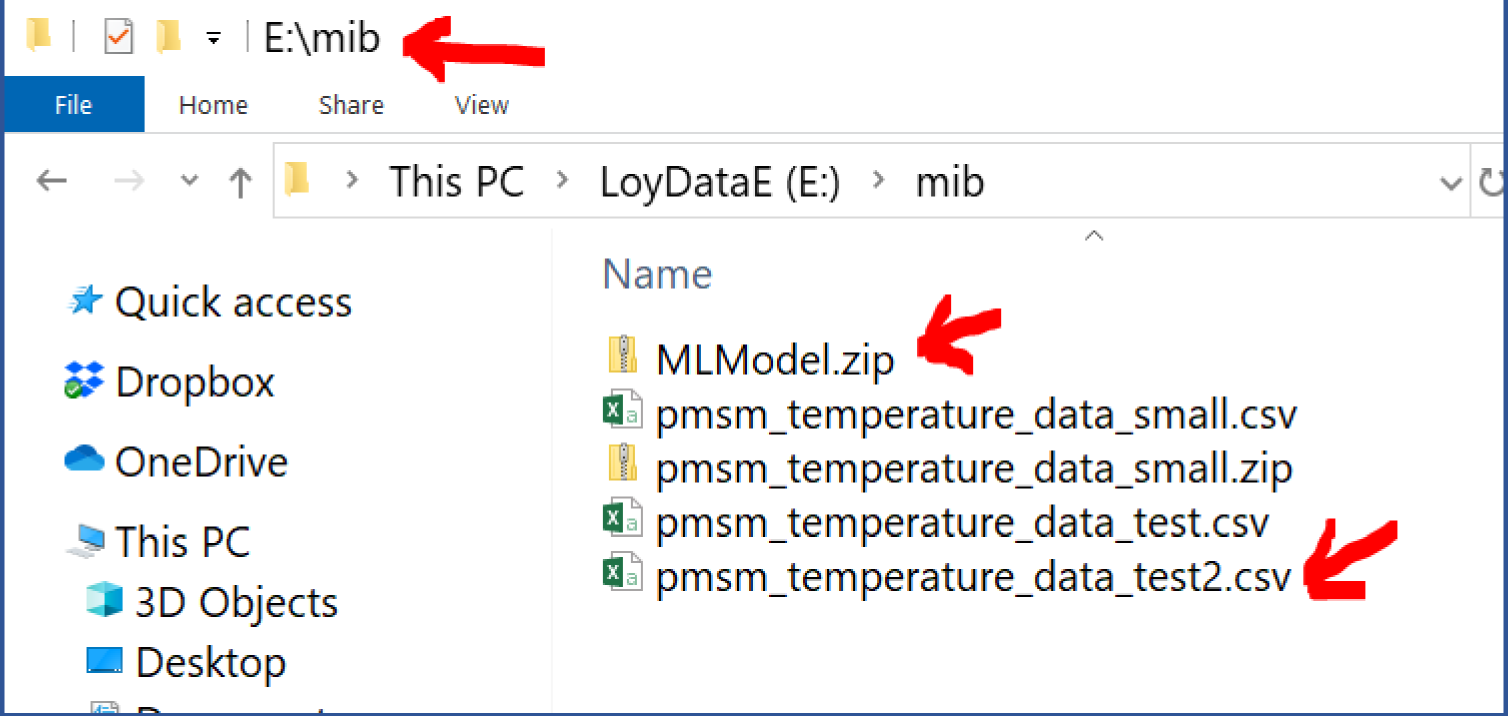
```
using System;
using Microsoft.ML.Data;

namespace test
{
    public class ModelOutput: ModelInput
    {
        public float Score { get; set; }
    }
}

```

Add using to Program



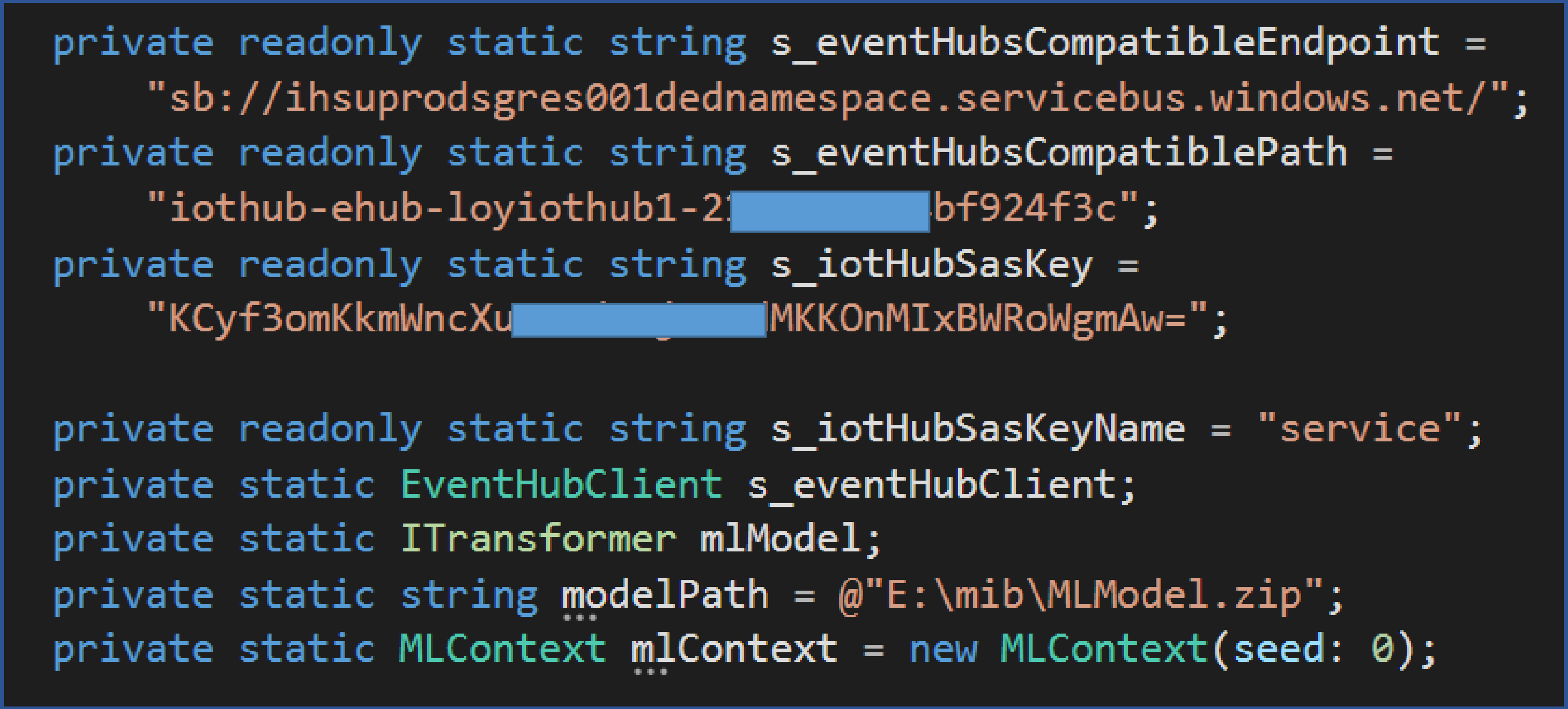
```
using Microsoft.Azure.EventHubs;
using Microsoft.ML;
using Newtonsoft.Json;
using System;
using System.Collections.Generic;
using System.Text;
using System.Threading;
using System.Threading.Tasks;
```

Add ML Model and test data to mib folder



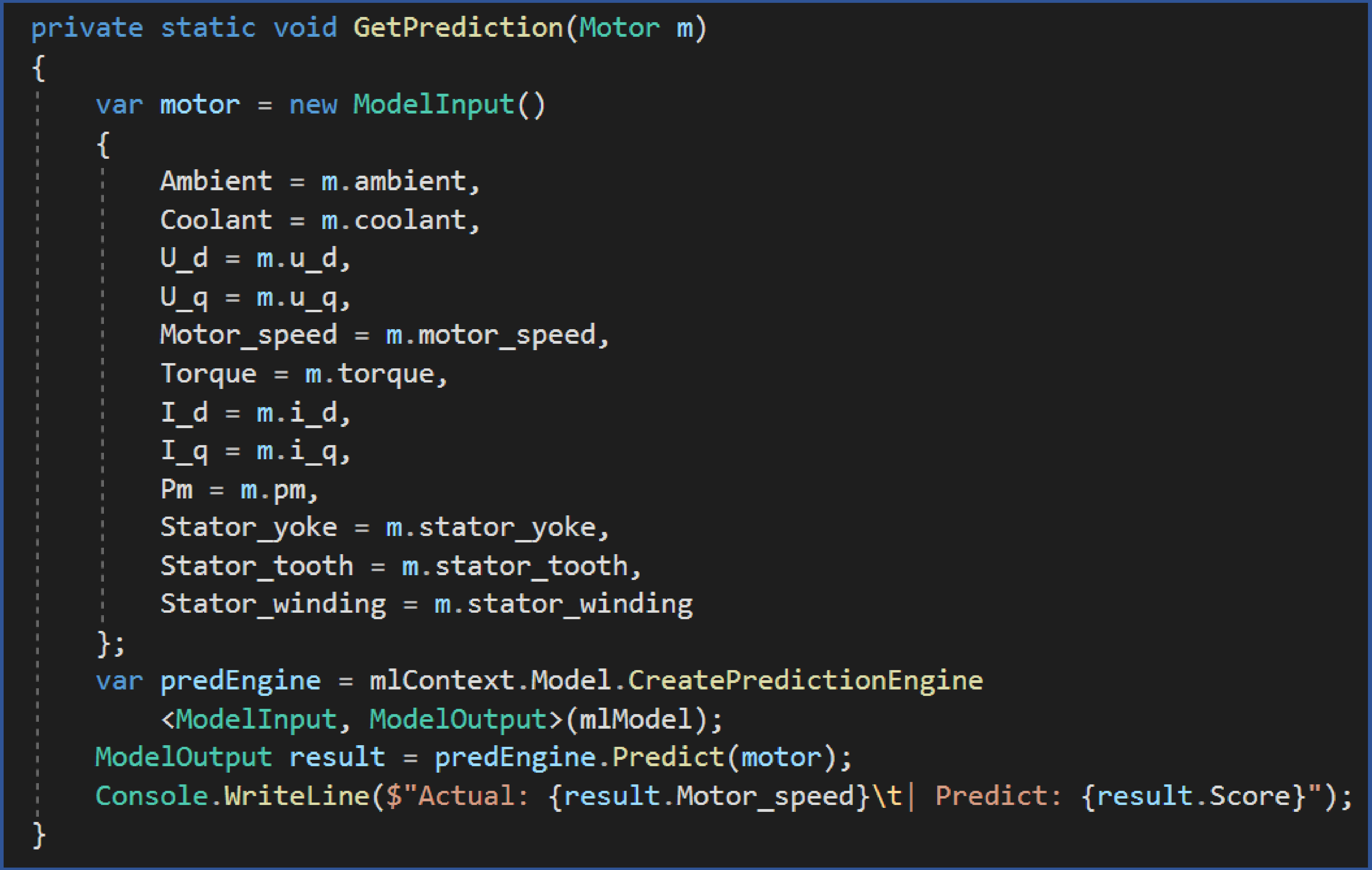
MLModel.zip
pmsm_temperature_data_small.csv
pmsm_temperature_data_small.zip
pmsm_temperature_data_test.csv
pmsm_temperature_data_test2.csv

Add class fields



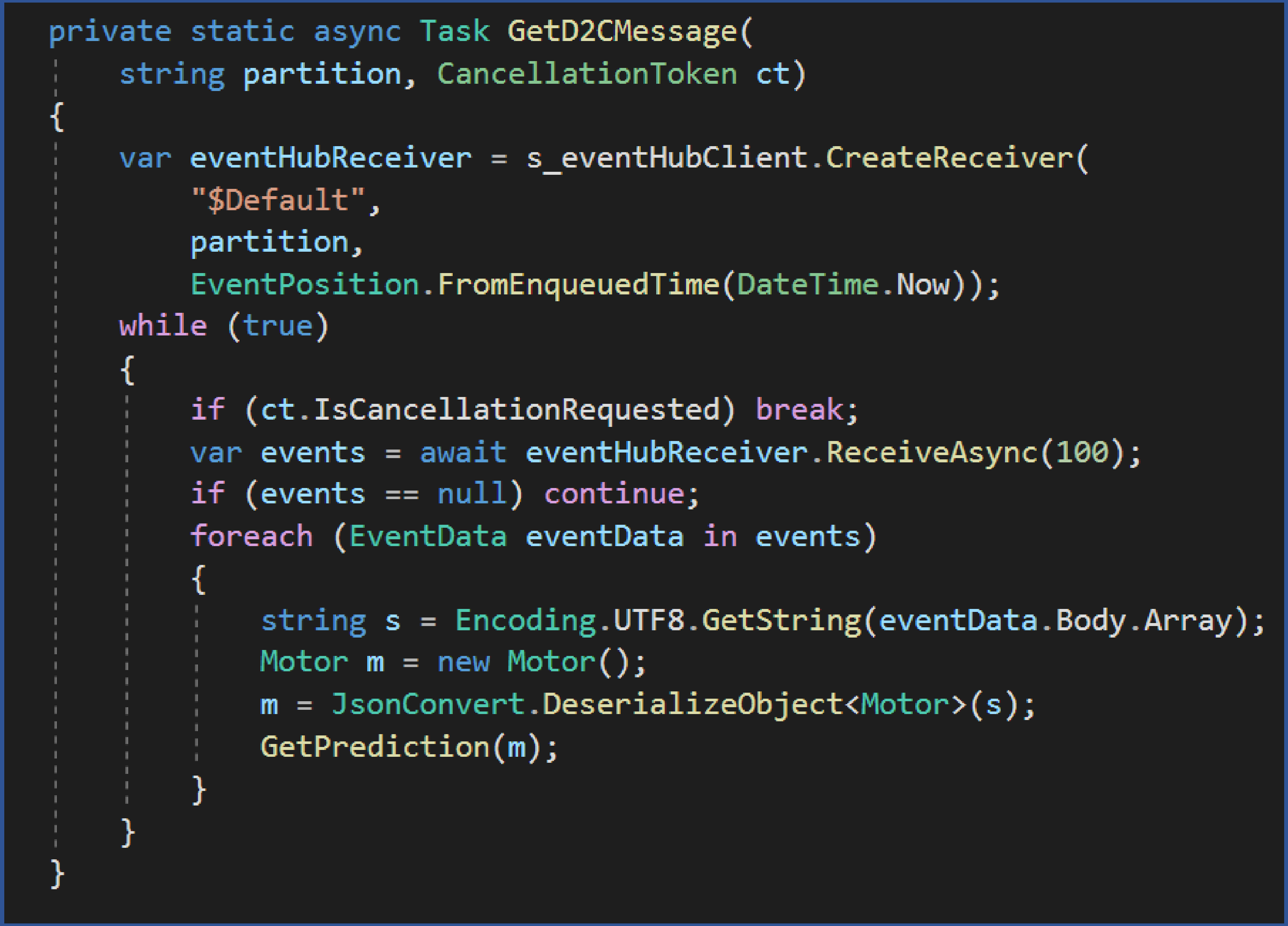
```
private readonly static string s_eventHubsCompatibleEndpoint =
    "sb://ihsuprodsgres001dednamespace.servicebus.windows.net/";
private readonly static string s_eventHubsCompatiblePath =
    "iothub-ehub-loyiothub1-2[redacted]bf924f3c";
private readonly static string s_iotHubSasKey =
    "KCyf3omKkmWncXu[redacted]MKKOnMIxBWRoWgmAw=";

private readonly static string s_iotHubSasKeyName = "service";
private static EventHubClient s_eventHubClient;
private static ITransformer mlModel;
private static string modelPath = @"E:\mib\MLModel.zip";
private static MLContext mlContext = new MLContext(seed: 0);
```

Add method GetPrediction



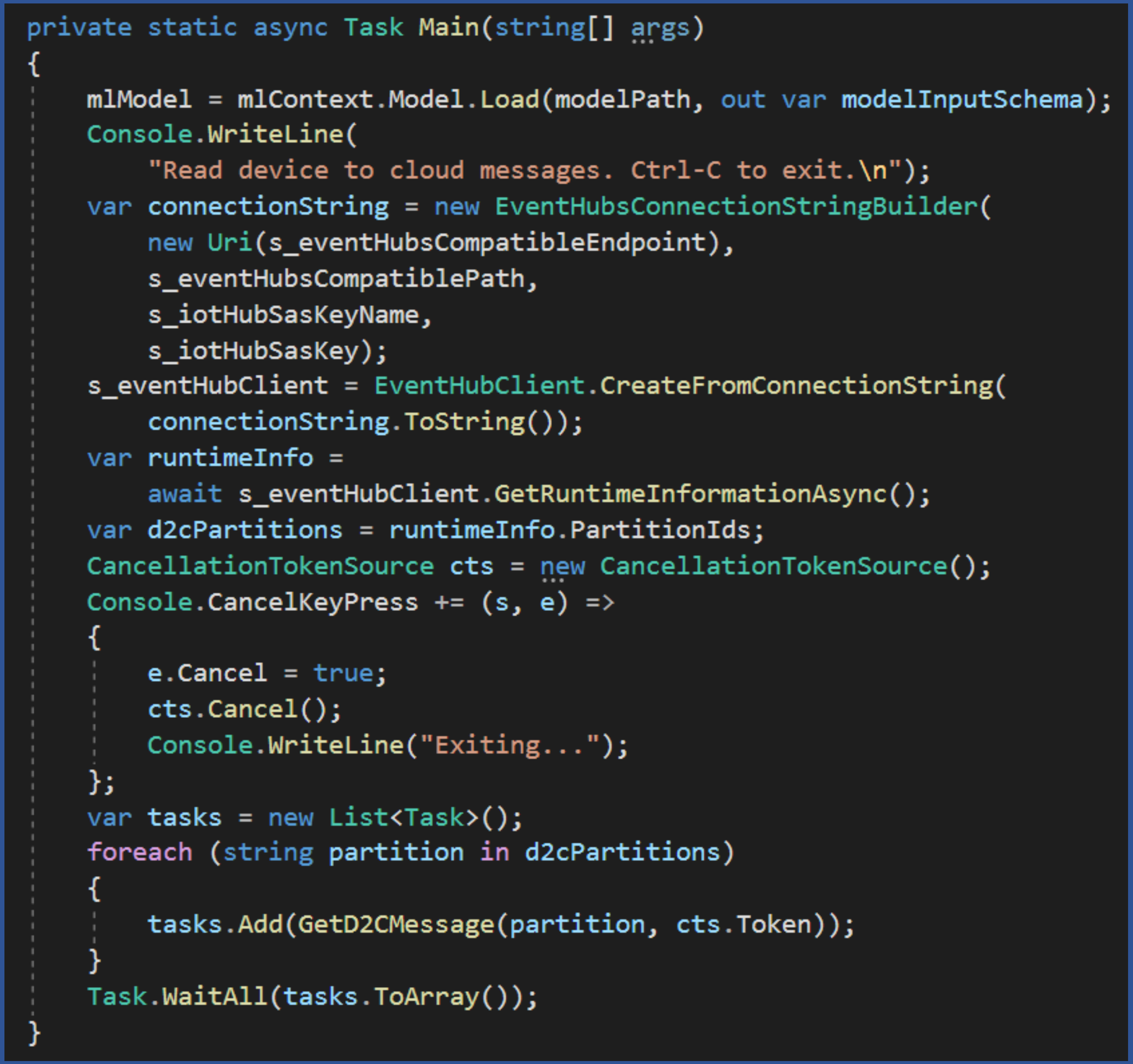
```
private static void GetPrediction(Motor m)
{
    var motor = new ModelInput()
    {
        Ambient = m.ambient,
        Coolant = m.coolant,
        U_d = m.u_d,
        U_q = m.u_q,
        Motor_speed = m.motor_speed,
        Torque = m.torque,
        I_d = m.i_d,
        I_q = m.i_q,
        Pm = m.pm,
        Stator_yoke = m.stator_yoke,
        Stator_tooth = m.stator_tooth,
        Stator_winding = m.stator_winding
    };
    var predEngine = mlContext.Model.CreatePredictionEngine
        <ModelInput, ModelOutput>(mlModel);
    ModelOutput result = predEngine.Predict(motor);
    Console.WriteLine($"Actual: {result.Motor_speed}\t| Predict: {result.Score}");
}
```

Add method GetD2CMessage



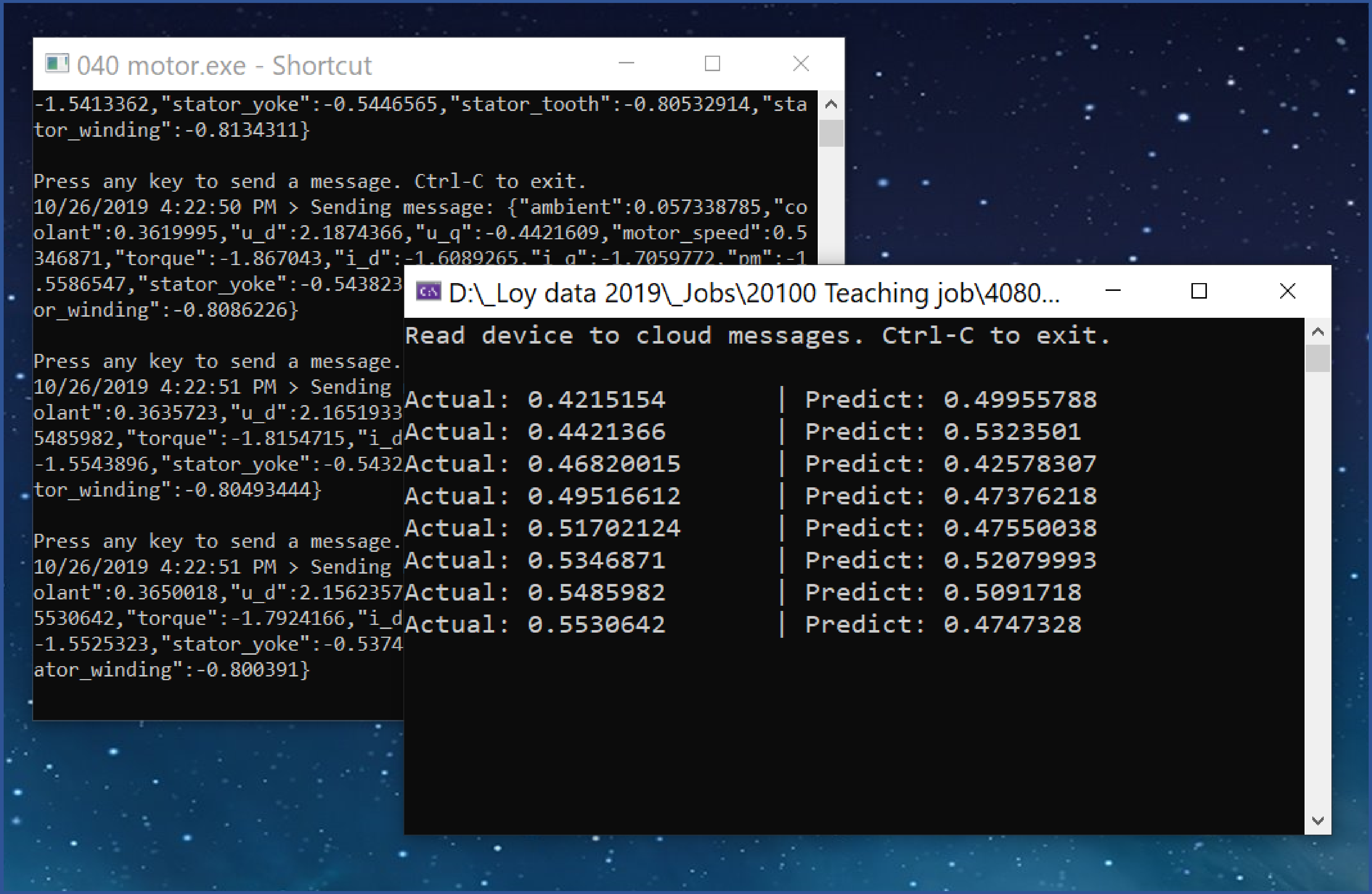
```csharp
private static async Task GetD2CMessage(
    string partition, CancellationToken ct)
{
    var eventHubReceiver = s_eventHubClient.CreateReceiver(
        "$Default",
        partition,
        EventPosition.FromEnqueuedTime(DateTime.Now));
    while (true)
    {
        if (ct.IsCancellationRequested) break;
        var events = await eventHubReceiver.ReceiveAsync(100);
        if (events == null) continue;
        foreach (EventData eventData in events)
        {
            string s = Encoding.UTF8.GetString(eventData.Body.Array);
            Motor m = new Motor();
            m = JsonConvert.DeserializeObject<Motor>(s);
            GetPrediction(m);
        }
    }
}
```

Add code to Main



```
private static async Task Main(string[] args)
{
    mlModel = mlContext.Model.Load(modelPath, out var modelInputSchema);
    Console.WriteLine(
        "Read device to cloud messages. Ctrl-C to exit.\n");
    var connectionString = new EventHubsConnectionStringBuilder(
        new Uri(s_eventHubsCompatibleEndpoint),
        s_eventHubsCompatiblePath,
        s_iotHubSasKeyName,
        s_iotHubSasKey);
    s_eventHubClient = EventHubClient.CreateFromConnectionString(
        connectionString.ToString());
    var runtimeInfo =
        await s_eventHubClient.GetRuntimeInformationAsync();
    var d2cPartitions = runtimeInfo.PartitionIds;
    CancellationTokenSource cts = new CancellationTokenSource();
    Console.CancelKeyPress += (s, e) =>
    {
        e.Cancel = true;
        cts.Cancel();
        Console.WriteLine("Exiting...");
    };
    var tasks = new List<Task>();
    foreach (string partition in d2cPartitions)
    {
        tasks.Add(GetD2CMessage(partition, cts.Token));
    }
    Task.WaitAll(tasks.ToArray());
}
```

Run program and verify the result



```
-1.5413362,"stator_yoke":-0.5446565,"stator_tooth":-0.80532914,"sta
tor_winding":-0.8134311}

Press any key to send a message. Ctrl-C to exit.
10/26/2019 4:22:50 PM > Sending message: {"ambient":0.057338785,"co
olant":0.3619995,"u_d":2.1874366,"u_q":-0.4421609,"motor_speed":0.5
346871,"torque":-1.867043,"i_d":-1.6089265,"i_q":-1.7059772,"pm":-1
.5586547,"stator_yoke":-0.543823
or_winding":-0.8086226}

Press any key to send a message.
10/26/2019 4:22:51 PM > Sending
olant":0.3635723,"u_d":2.1651933
5485982,"torque":-1.8154715,"i_d
-1.5543896,"stator_yoke":-0.5432
tor_winding":-0.80493444}

Press any key to send a message.
10/26/2019 4:22:51 PM > Sending
olant":0.3650018,"u_d":2.1562357
5530642,"torque":-1.7924166,"i_d
-1.5525323,"stator_yoke":-0.5374
ator_winding":-0.800391}
```

```
Read device to cloud messages. Ctrl-C to exit.

Actual: 0.4215154        | Predict: 0.49955788
Actual: 0.4421366        | Predict: 0.5323501
Actual: 0.46820015       | Predict: 0.42578307
Actual: 0.49516612       | Predict: 0.47376218
Actual: 0.51702124       | Predict: 0.47550038
Actual: 0.5346871        | Predict: 0.52079993
Actual: 0.5485982        | Predict: 0.5091718
Actual: 0.5530642        | Predict: 0.4747328
```

What's next?