HelloWorld!

ASP.NET Core Web App

What's in this session?

1. Question and Data

2. Create project

3. Add NuGet packages

4. Add using name space

5. Create data set input/output scheme

6. Set data set path

7. Load data

8.   Add algorithm

9.   Train the model
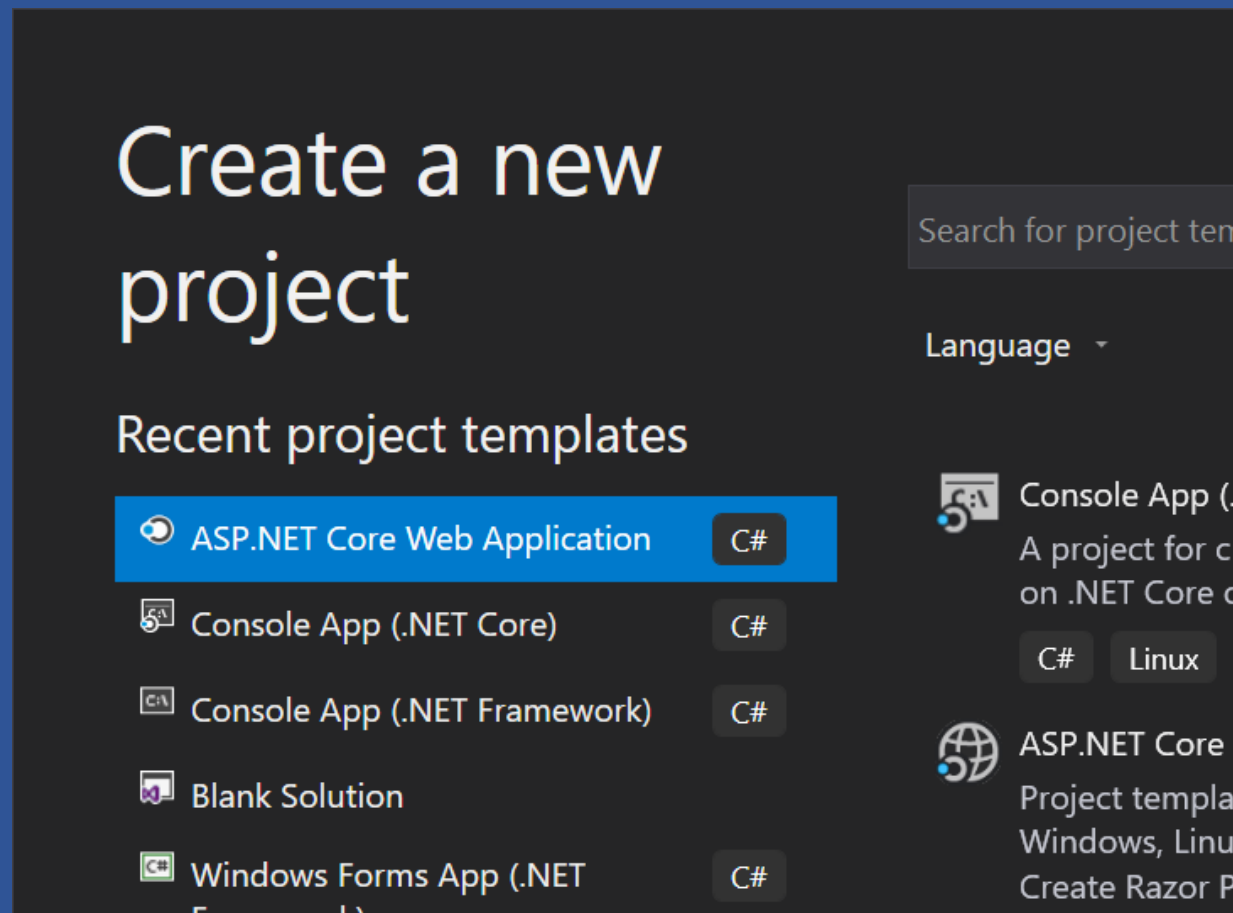
10.   Predict single item

Question and Data

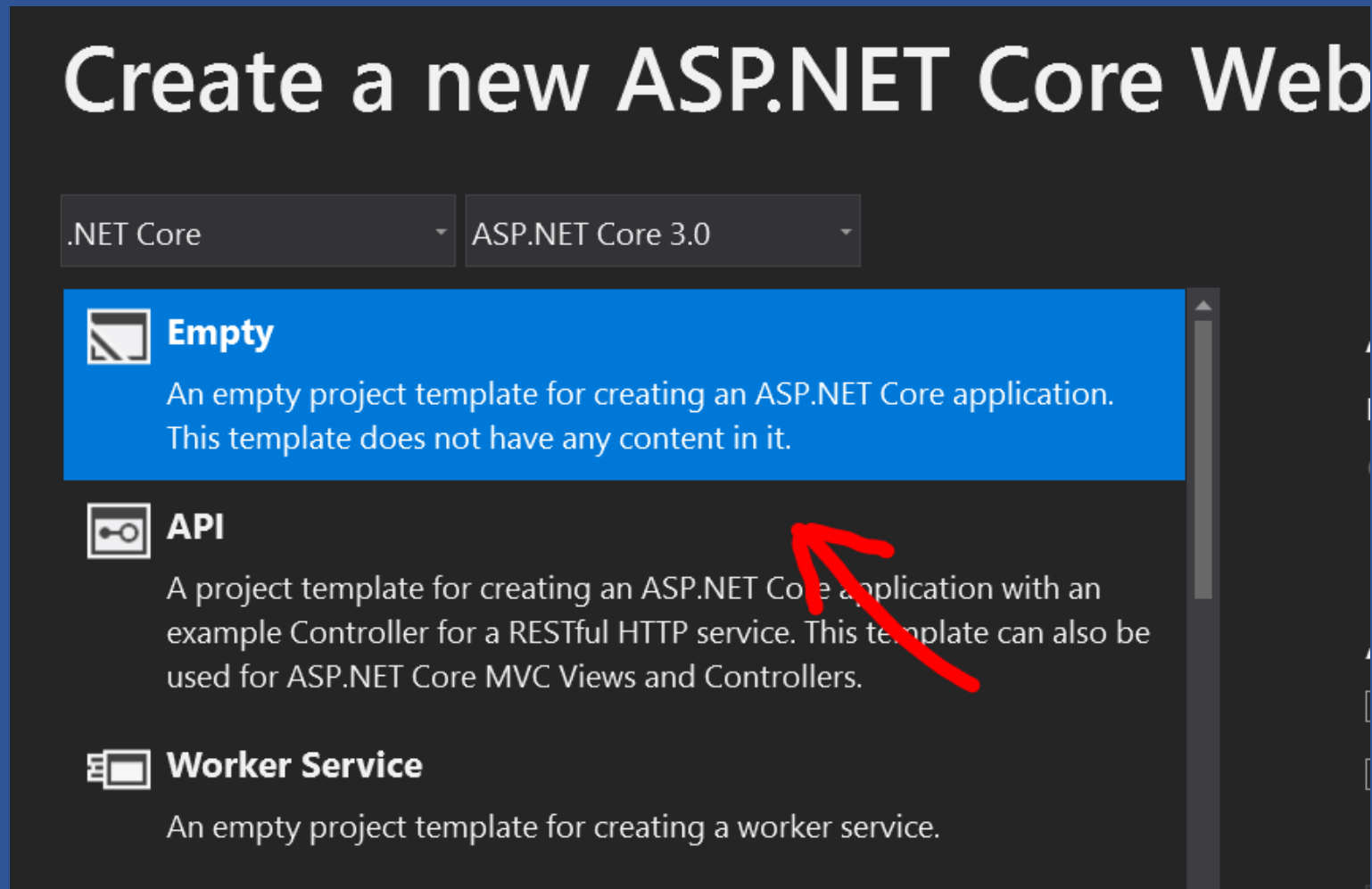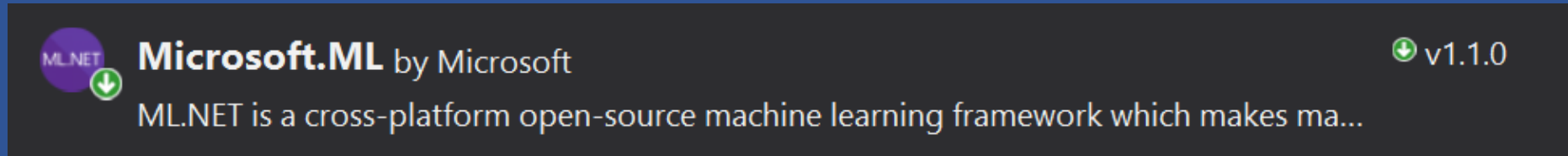Question: How much is my diamond costed?

Dataset:

https://github.com/laploy/ML.NET/blob/master/HelloWorld/diamondSmall.csv

GreatFriends.Biz

Create New Project

Create ASP.NET Core web app name = "HelloWeb"

Empty project template

Add NuGet Packages

Microsoft.ML by Microsoft   v1.1.0

ML.NET is a cross-platform open-source machine learning framework which makes ma...

Add Namespace

```
3    using System;
4        using Microsoft.ML;
5        using Microsoft.ML.Data;
```

Change this line in Start up

```
app.UseEndpoints(endpoints =>
{
    endpoints.MapGet("/", async context =>
    {
        await context.Response.WriteAsync("Hello World!");
    });
});
```

to this

```
app.UseEndpoints(endpoints =>
{
    endpoints.MapGet("/", async context => { test(context);});
});
```

Create data scheme

```csharp
public class DiamondScheme
{
    [LoadColumn(0)]
    public float Size { get; set; }
    [LoadColumn(1)]
    public float Price { get; set; }
}
public class DiamonPredic: DiamondScheme
{
    [ColumnName("Score")]
    public float PricePredict { get; set; }
}
```
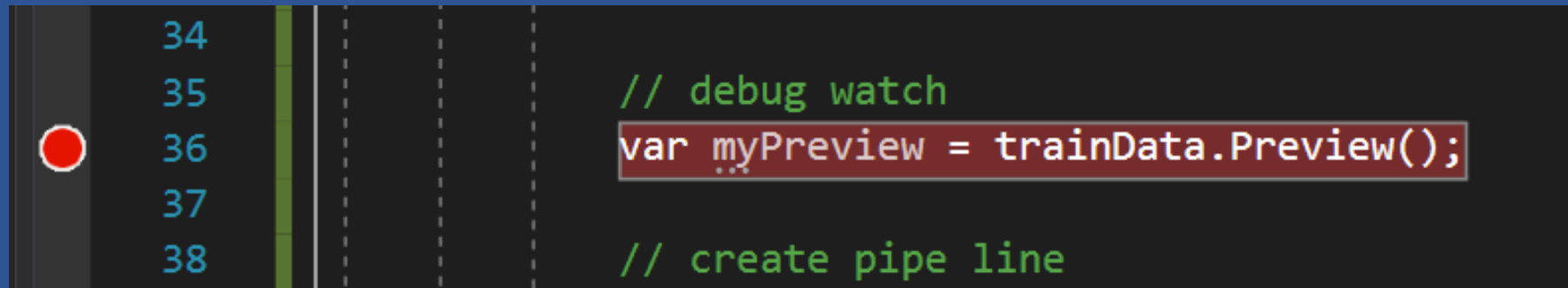
Add data path, context, IDataView, and Debug

```
string trainDataPath = @"E:\ml\diamondSmall.csv";

// create context
MLContext mlContext = new MLContext();

// Load train data
IDataView trainData = mlContext.Data.LoadFromTextFile<DiamondScheme>
    (trainDataPath, hasHeader: true, separatorChar: ',');

// debug watch
var myPreview = trainData.Preview();
```

GreatFriends.Biz

Set break point / run / add watch


```
34
35        // debug watch
36        var myPreview = trainData.Preview();
37
38        // create pipe line
```

Create pipe line / set algorithm / train

```
// create pipe line
var pipeline = mlContext.Transforms.Concatenate
    ("Features", new[] { "Size" })

// set training algorithm
    .Append(mlContext.Regression.Trainers.Sdca
        (labelColumnName: "Price",
        maximumNumberOfIterations:100));

// Train model
var myModel = pipeline.Fit(trainData);
```
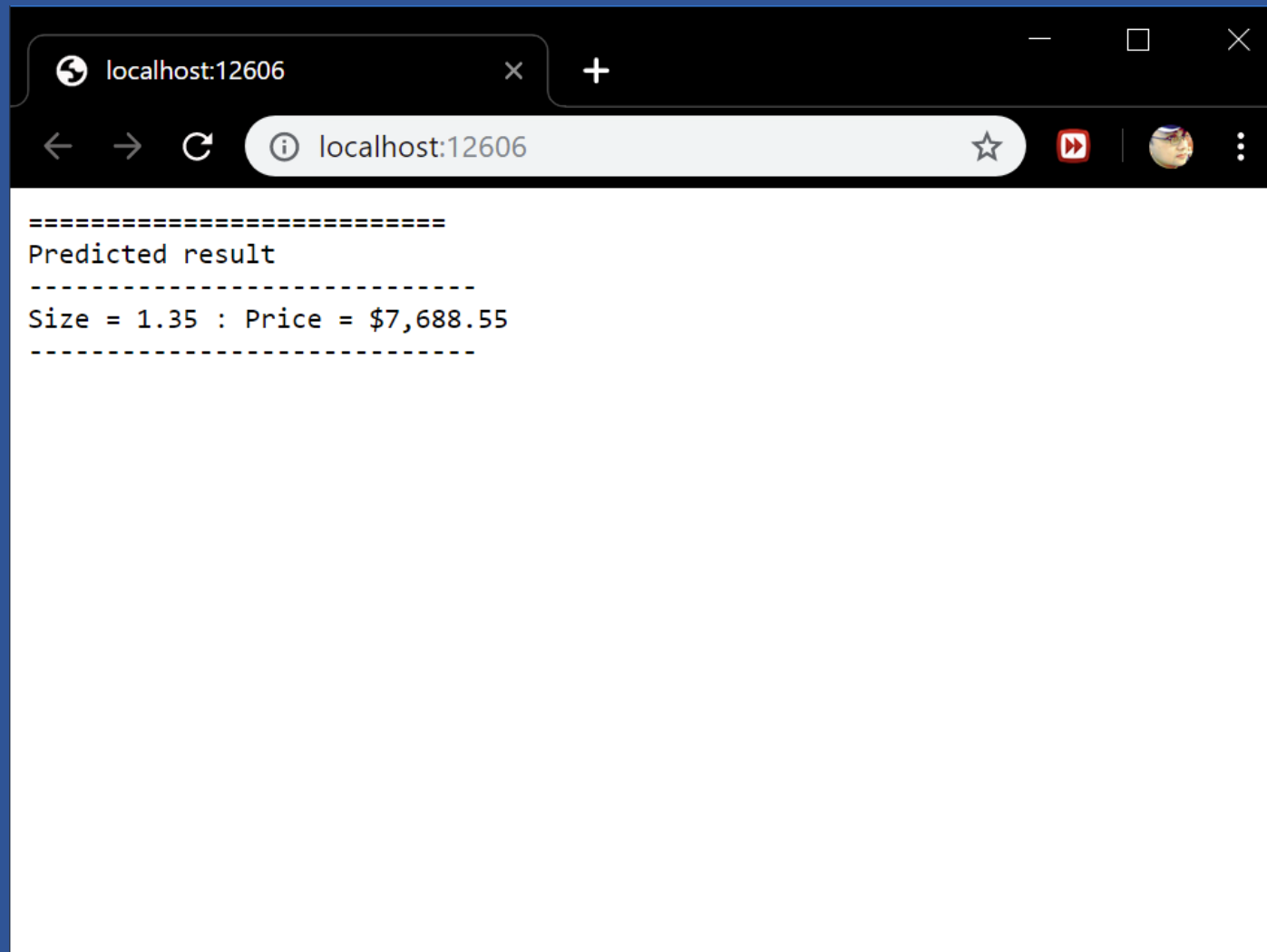
GreatFriends.Biz

Make predict / Show result

```
// Make a prediction
var mySize = new DiamondScheme() { Size = 1.35F };
var myPrice = mlContext.Model.CreatePredictionEngine
    <DiamondScheme, DiamonPredic>
    (myModel).Predict(mySize);

// Show result
await context.Response.WriteAsync("============================\n");
await context.Response.WriteAsync("Predicted result\n");
await context.Response.WriteAsync("----------------------------\n");
await context.Response.WriteAsync($"Size = {mySize.Size} " +
    $": Price = {myPrice.PricePredict:C}\n");
await context.Response.WriteAsync("----------------------------");
```

Examine the result

What's next?