

HelloWorld!

Windows form 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?

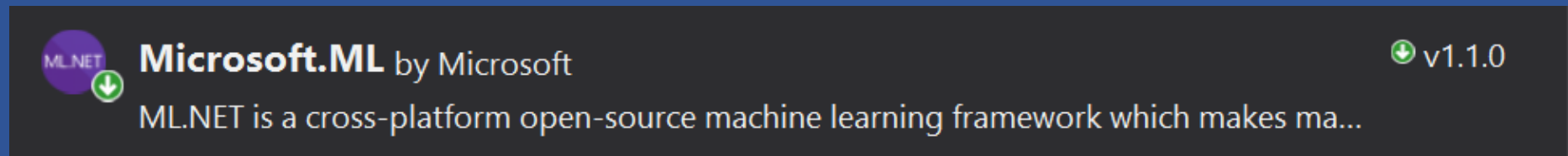
Dataset:

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

Create New Project

Create new Windows form app project name = “HelloWin”

Add NuGet Packages



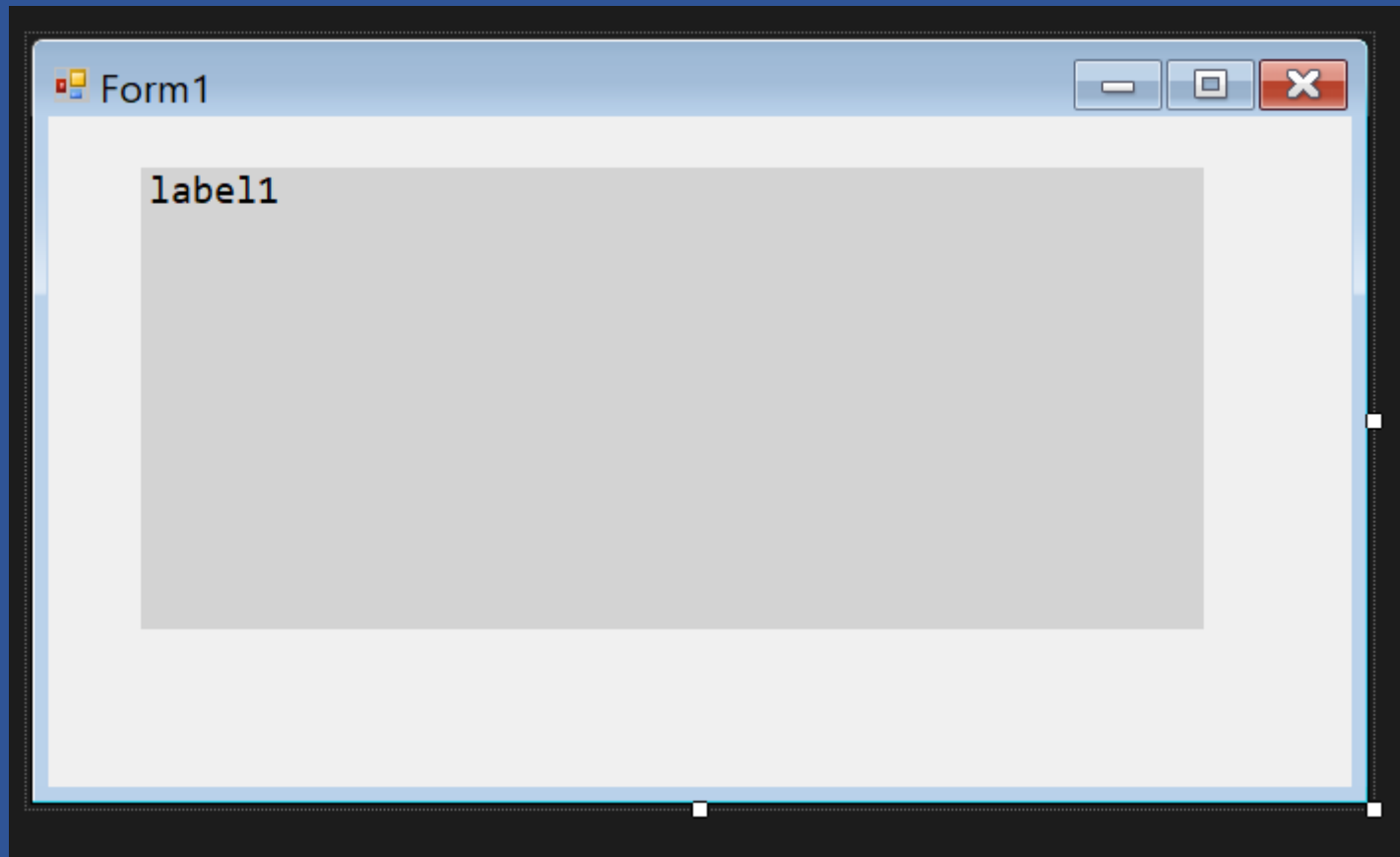
Add Namespace

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

Create data scheme

```
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; }
}
```

Make a Form



Add data path, context, IDataView, and Debug

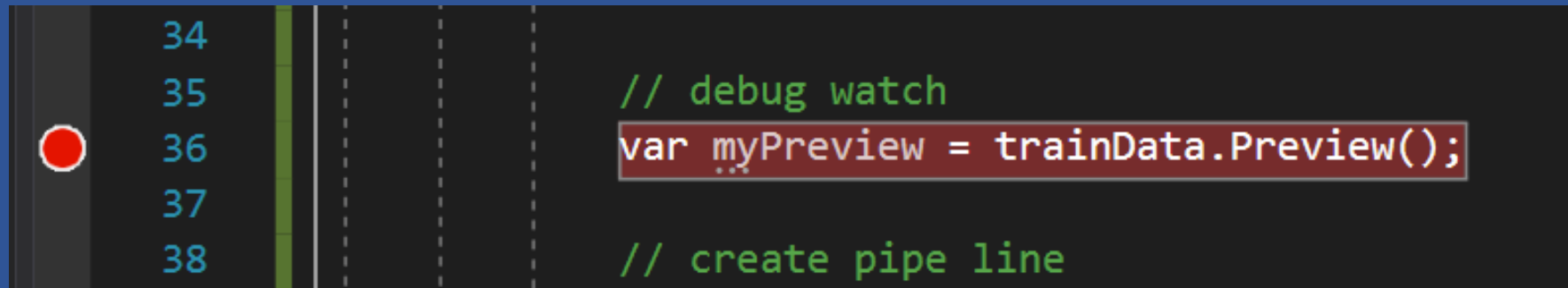
```
string trainDataPath = @"E:\ml\diamondSmall.csv";

// 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();
```

Set break point / run / add watch

A screenshot of a code editor with a dark background. On the left side, there is a vertical toolbar with a red circle icon, which is a common symbol for setting a break point in many IDEs. The code is written in a light green monospace font. Line 36 is highlighted with a red background, and the text 'var myPreview = trainData.Preview();' is highlighted with a red selection box. The code includes comments '// debug watch' and '// create pipe line'.

```
34  
35  
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));

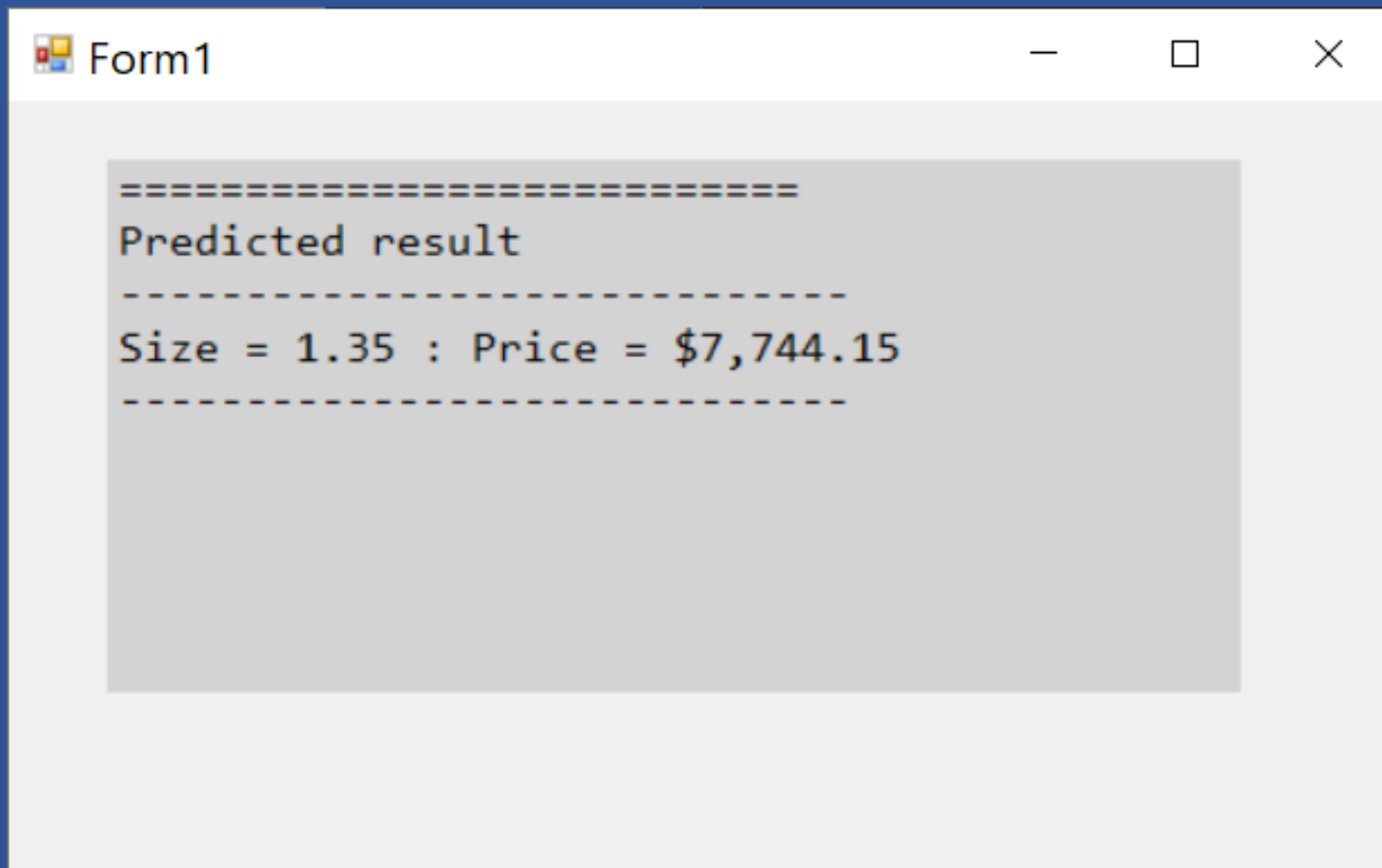
// Train model
var myModel = pipeline.Fit(trainData);
```

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
label1.Text = "=====\n";
label1.Text += "Predicted result\n";
label1.Text += "-----\n";
label1.Text += $"Size = {mySize.Size} " +
    $": Price = {myPrice.PricePredict:C}\n";
label1.Text += "-----";
```

Examine the result



What's next?

