

HelloWorld!

Write Code ML.NET

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

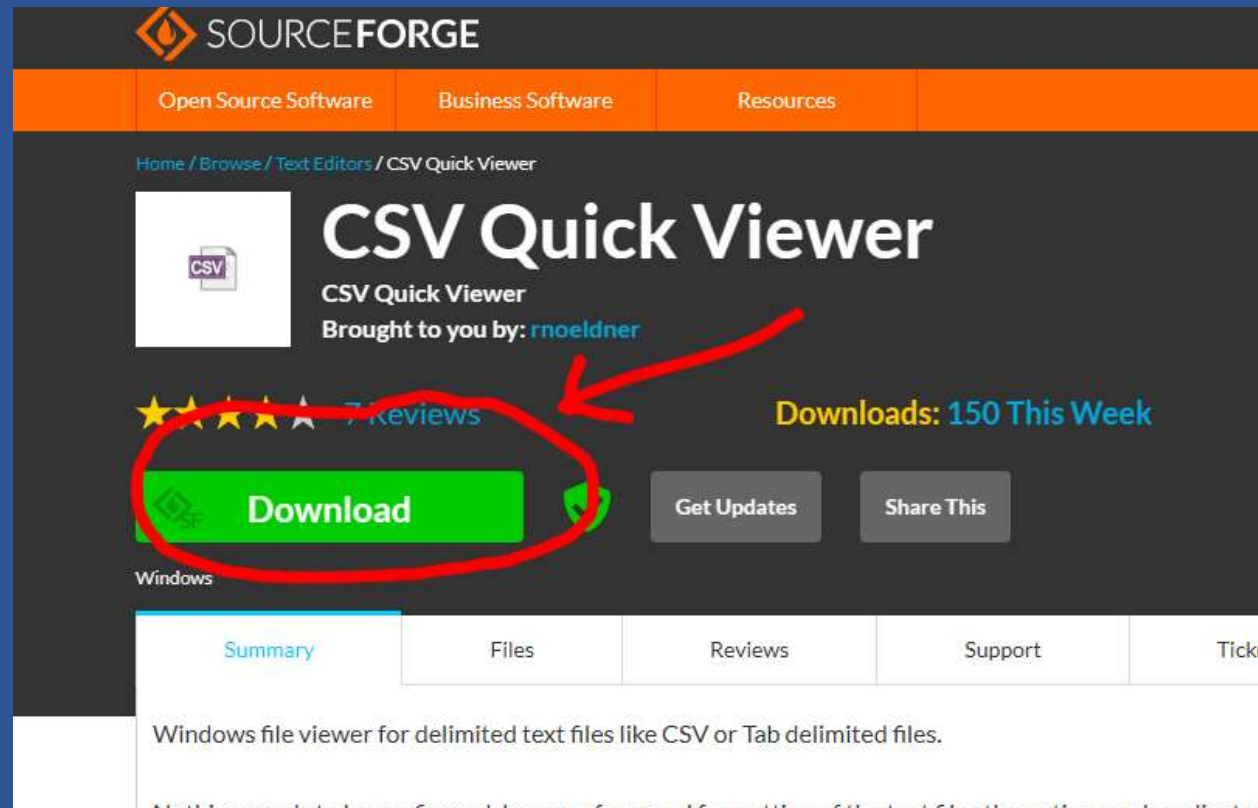
Question: How much is my diamond costed?

Dataset:

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

CSV Quick Viewer

<https://sourceforge.net/projects/csvquickviewer/>



The screenshot displays the SourceForge project page for "CSV Quick Viewer". The page header includes the SourceForge logo and navigation links for "Open Source Software", "Business Software", and "Resources". The breadcrumb trail reads "Home / Browse / Text Editors / CSV Quick Viewer". The project name "CSV Quick Viewer" is prominently displayed, along with the text "Brought to you by: rnoeldner". Below the project name, there are five yellow stars and the text "7 Reviews". A red circle highlights the "Download" button, and a red arrow points from the reviews section towards the download button. To the right of the reviews, it says "Downloads: 150 This Week". Below the project name, there are three buttons: "Download" (green), "Get Updates" (grey), and "Share This" (grey). The "Download" button is circled in red. Below the buttons, there are tabs for "Summary", "Files", "Reviews", "Support", and "Tickets". The "Summary" tab is selected, showing the text "Windows file viewer for delimited text files like CSV or Tab delimited files."

Examine data

H:\ml\diamondSmall.csv - CP 65001 - Unicode (UTF-8) without BOM - CSV Quick Vie... — □ ×

Open File Setting **All Records** Unique Values Duplicates Hierarchy

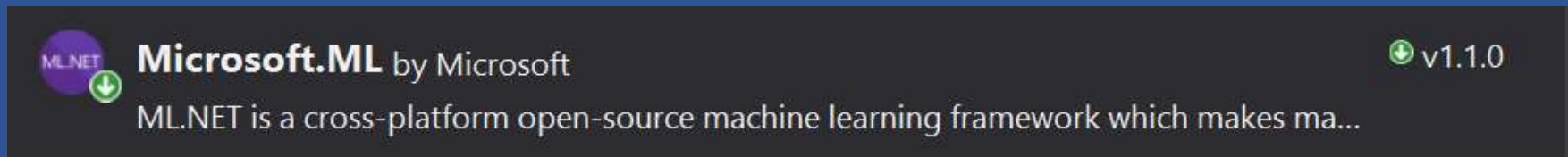
	Size	Price	#Line
▶	1.01	7366	2
	0.49	985	3
	0.31	544	4
	1.51	9140	5
	0.37	493	6
	0.73	3011	7
	1.53	11413	8
	0.56	1814	9
	0.41	876	10
	0.74	2690	11
	0.63	1190	12
	0.6	4172	13
	2.06	11764	14
	1.1	4682	15
	1.31	6171	16
*			

1 of 15

Create New Project

Create new .NET CORE console app project name = "HelloWorld"

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; }
}
```

Add data path, context, IDataView, and Debug

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

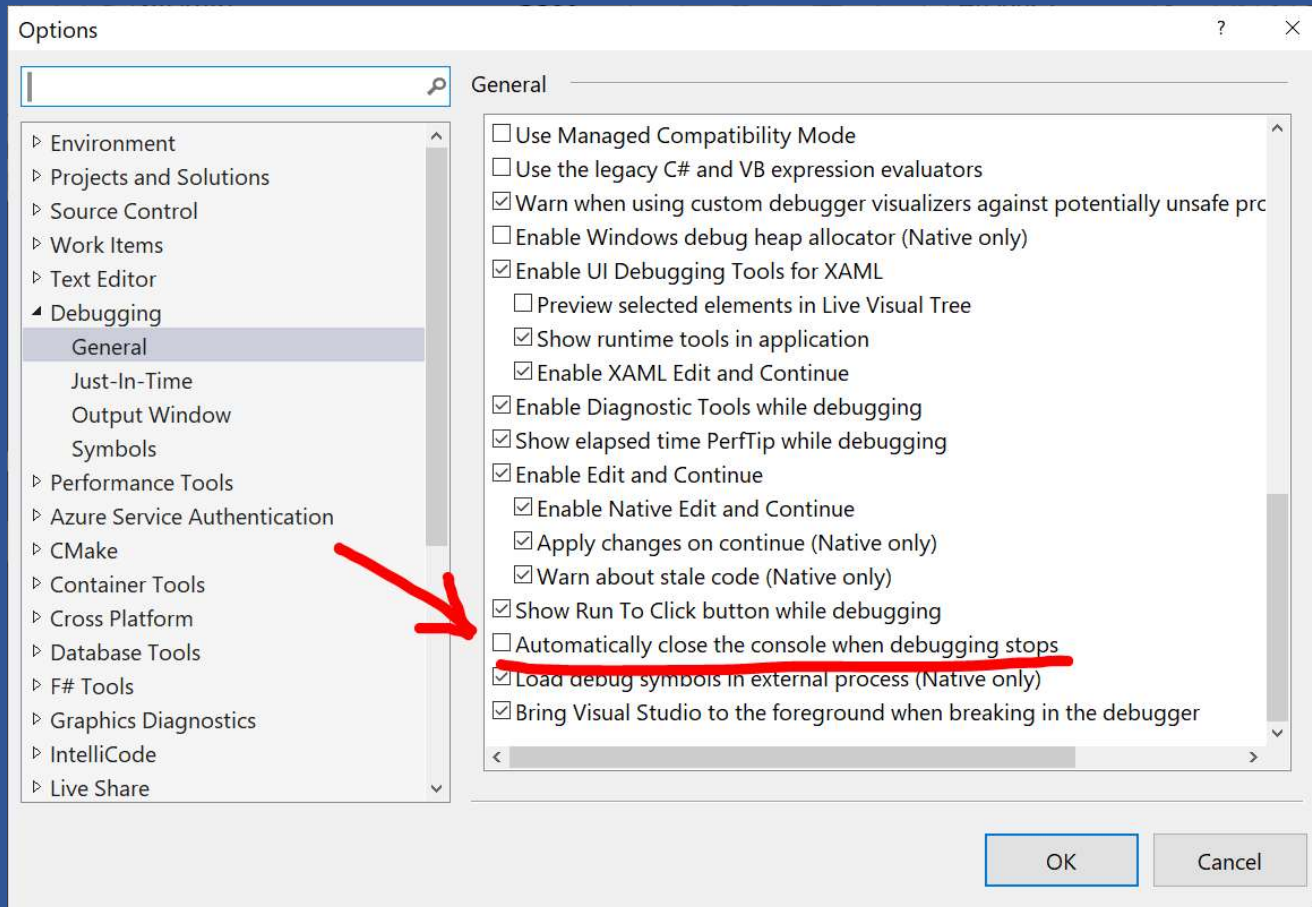
// create context
MLContext mlContext = new MLContext();

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

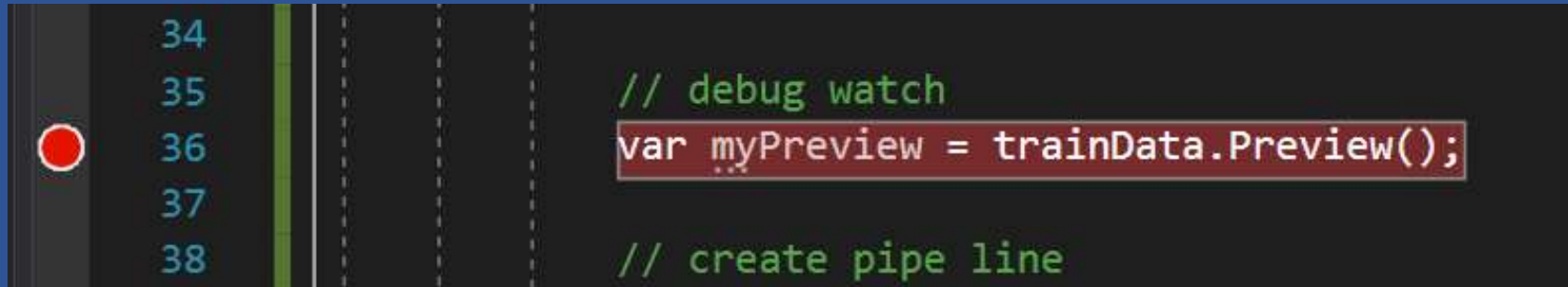
// debug watch
var myPreview = trainData.Preview();
```

Set the console to remain open

Tools->Options->Debugging->Automatically close the console when debugging stops



Set break point / run / add watch



A screenshot of a code editor with a dark background. On the left, line numbers 34, 35, 36, 37, and 38 are listed. A red circle with a white center, representing a break point, is positioned to the left of line 36. A vertical green line is positioned to the right of line 36. The code on line 36 is `var myPreview = trainData.Preview();`, which is highlighted with a red background. The code on line 35 is `// debug watch` and the code on line 38 is `// create pipe line`.

```
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);
```


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
Console.WriteLine("\n=====");
Console.WriteLine("Predicted result");
Console.WriteLine("-----");
Console.WriteLine($"Size = {mySize.Size} " +
    $": Price = {myPrice.PricePredict:C}");
Console.WriteLine("-----");
```

Examine the result

```
=====
Predicted result
-----
Size = 1.35 : Price = $7,715.11
-----
```

What's next?

