

Introdução ao Aprendizado de Máquina em C#

Aprendizado de Máquina até em .NET.



Luiz Fernando Ramos

Sócio e Gerente de Contas da Dinamio Tecnologia
Joinville, Santa Catarina, Brasil

Profissional graduado em desenvolvimento e análise de sistemas da informação com mais de 15 anos de experiência em empresas de tecnologia. Experiência na condução de equipes de desenvolvimento e suporte a aplicações e sistemas complexos. Participação em inúmeros projetos em empresas de grande representatividade nacional e internacional. Trabalhos realizados como analista e desenvolvedor de soluções em ambiente Microsoft além de implantação de portais SharePoint para padronização e organização de processos empresariais.

Amplo conhecimento em linguagens de programação .Net, SharePoint, Windows Server e Microsoft SQL Server.



Por que .NET C#?



Accord.NET Framework

Iniciando

Visual Studio

Enterprise 2017

This program is protected by U.S. and international copyright laws as described in Help/About.

© 2017 Microsoft Corporation.
All rights reserved.



SUA DEDICAÇÃO,
COMPROMETIMENTO
E PROFISSIONALISMO
SÃO OS INGREDIENTES
QUE FAZEM A DIFERENÇA
NA NOSSA HISTÓRIA.

MAIO DIA DO
TRABALHADOR

DINAMIO

Potencializando Empresas e suas Pessoas.



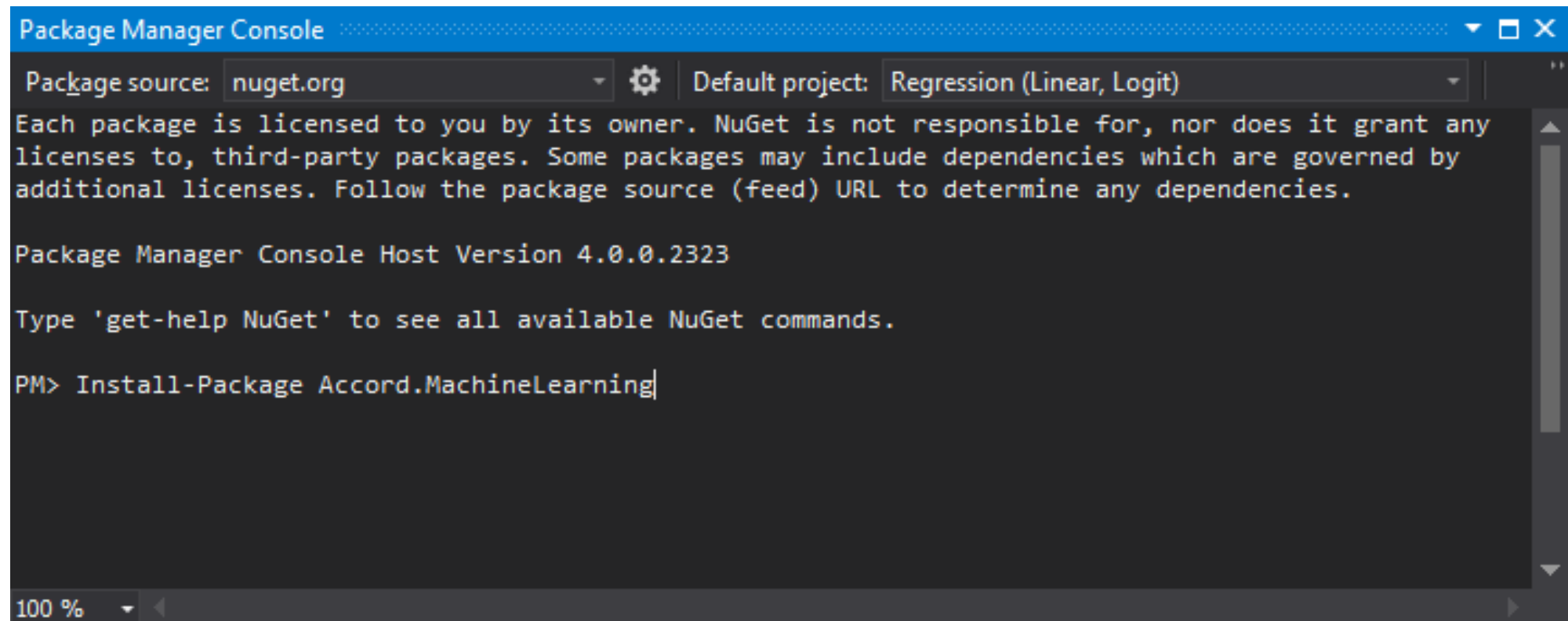
○ Digite aqui para pesquisar



09:00

12/05/2017

Iniciando



```
Package Manager Console
Package source: nuget.org
Default project: Regression (Linear, Logit)

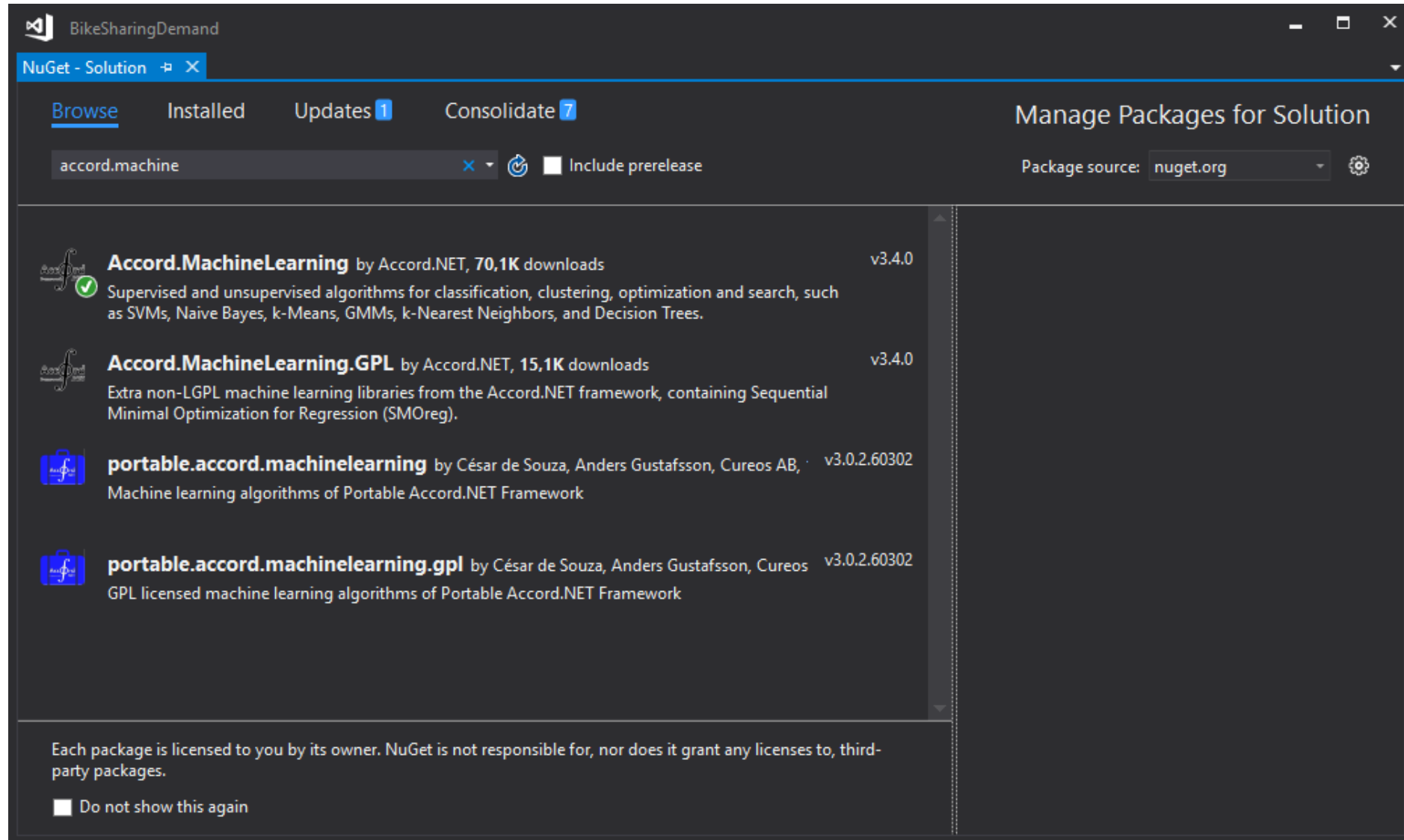
Each package is licensed to you by its owner. NuGet is not responsible for, nor does it grant any
licenses to, third-party packages. Some packages may include dependencies which are governed by
additional licenses. Follow the package source (feed) URL to determine any dependencies.

Package Manager Console Host Version 4.0.0.2323

Type 'get-help NuGet' to see all available NuGet commands.

PM> Install-Package Accord.MachineLearning
```

Iniciando



Lendo dados

```
1  using Accord.IO;
2  using System.Data;
3  using System.Threading;
4
5  namespace ConsoleApp1
6  {
7      class Program
8      {
9          static void Main(string[] args)
10         {
11             Thread.CurrentThread.CurrentCulture = new System.Globalization.CultureInfo("en-US");
12
13             DataTable tableInputs = new CsvReader("csvfile.csv", true).ToTable();
14
15             DataTable table = new ExcelReader("worksheet.xlsx").GetWorksheet("Sheet1");
16         }
17     }
18 }
19
```


Iris Species

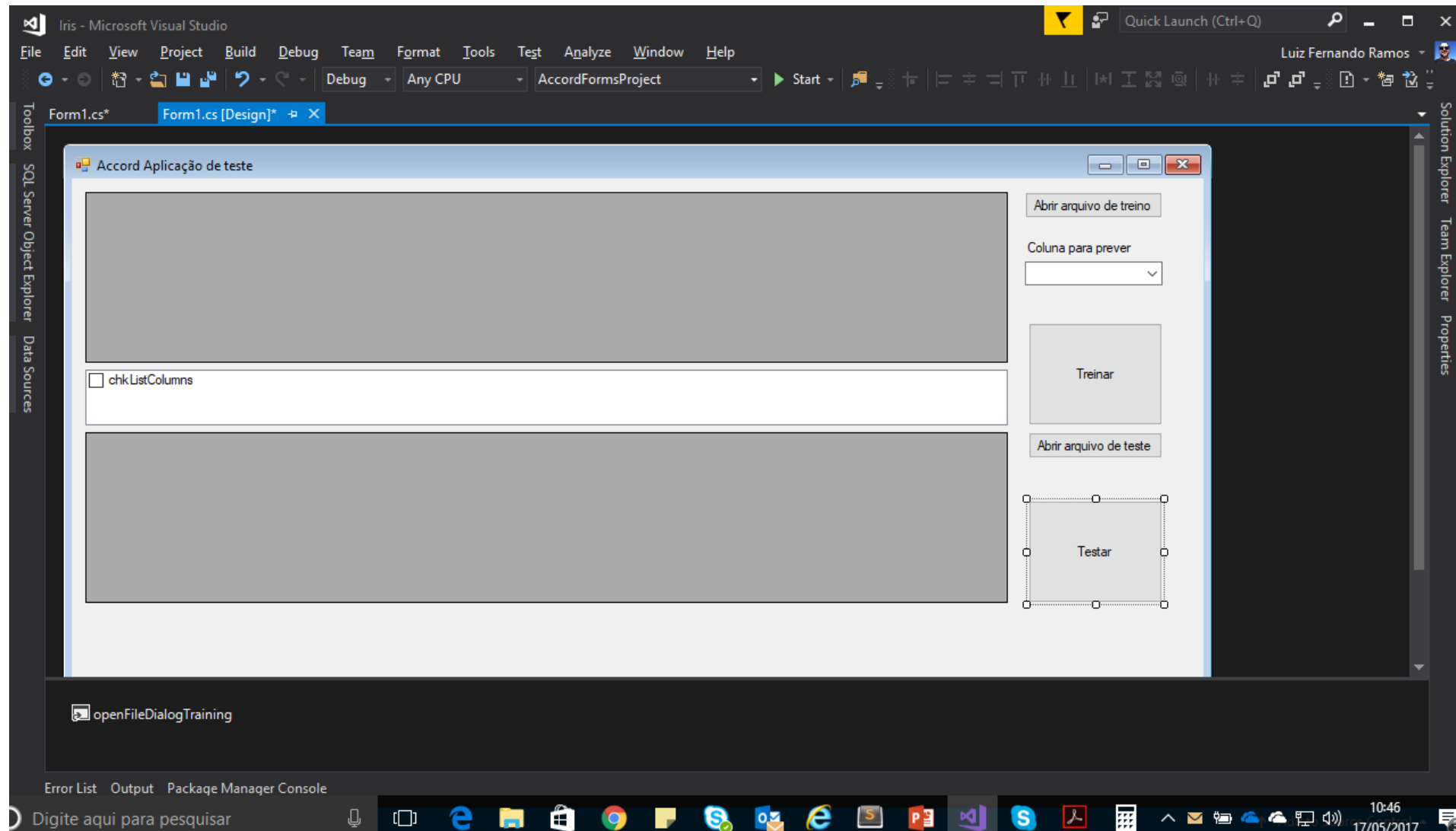


O Problema

Classificação da planta Íris

- 3 classes (tipos de planta) e 50 instâncias cada
- Um dos tipos é linearmente separado das outras, mas as outras 2 não
- Atributos:
 - Comprimento da sépala
 - Largura da sépala
 - Comprimento da pétala
 - Largura da pétala
- Classes:
 - Íris Setosa
 - Íris Versicolor
 - Íris Virginica

Nosso aplicativo



Treino

1 reference

```
private void btnTrain_Click(object sender, EventArgs e)
{
    string[] selectedColumns = chkListColumns.CheckedItems.Cast<string>().ToArray<string>();

    // Create a new codification codebook to
    // convert strings into integer symbols
    Codification codebook = new Codification(tableInputs, cmbColumns.SelectedItem.ToString());
    DataTable tableInputs1 = codebook.Apply(tableInputs);

    double[][] inputs = tableInputs.ToJagged<double>(selectedColumns);
    int[] outputs = tableInputs1.ToArray<int>(cmbColumns.SelectedItem.ToString());

    List<DecisionVariable> decisionVariables = new List<DecisionVariable>();
    for (int n = 0; n < selectedColumns.Length; ++n)
    {
        decisionVariables.Add(
            new DecisionVariable(selectedColumns[n], DecisionVariableKind.Continuous)
        );
    }

    // Create a new Decision Tree classifier.
    DecisionTree ClassificationDecisionTree = new DecisionTree(decisionVariables, codebook[cmbColumns.SelectedItem.ToString()].NumberOfSymbols);

    // Create a new instance of the C45 algorithm to be learned by the tree.
    C45Learning C45LearningTree = new C45Learning(ClassificationDecisionTree);

    double decisionTreeL = C45LearningTree.Run(inputs, outputs);
}
```

Teste

```
1 reference
private void btnOpenTestFile_Click(object sender, EventArgs e)
{
    if (openFileDialogTraining.ShowDialog() == DialogResult.OK)
    {
        tableInputsTest = new CsvReader(openFileDialogTraining.FileName, true).ToTable();
        tableInputsTest.Columns.Add(String.Format("result_{0}", cmbColumns.SelectedItem.ToString()));

        gridTest.DataSource = tableInputsTest;

        btnTest.Enabled = true;
    }
}

1 reference
private void btnTest_Click(object sender, EventArgs e)
{
    DataRow dataRow;
    double[] inputs;
    int result;
    foreach (DataGridViewRow row in gridTest.Rows)
    {
        dataRow = ((DataRowView)row.DataBoundItem).Row;
        inputs = dataRow.ToArray<double>(selectedColumns);
        result = ClassificationDecisionTree.Decide(inputs);

        row.Cells[String.Format("result_{0}", cmbColumns.SelectedItem.ToString())].Value = codebook.Revert(cmbColumns.SelectedItem.ToString(), result);
    }
}
```



DEMO



Thank You