

C# Machine Learning application

C# MACHINE LEARNING APPLICATION



C# Machine Learning application

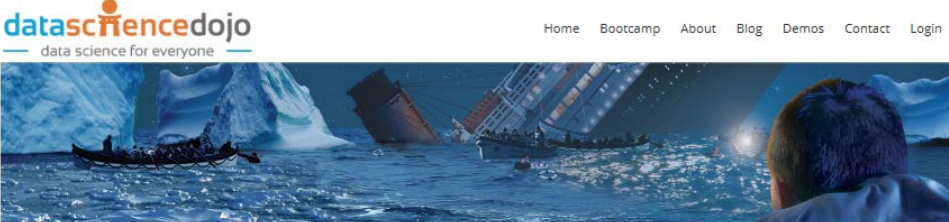
In this session

- Titanic Survival Predictor Web app
- Titanic Survival Predictor C# Win App
- Create User Interface
- Add class Titanic
- Titanic unit test
- Adding code to Class Form1

C# Machine Learning application

Titanic Survival Predictor Web app

<http://demos.datasciencedojo.com/demo/titanic/>



The image shows the Titanic Survival Predictor web application interface. It features a header with the 'datasciencedojo' logo and navigation links. Below the header is a large image of the Titanic ship. The main content area is titled 'Titanic Survival Predictor' and contains a form for predicting survival based on various factors. The form includes dropdown menus for passenger class and gender, sliders for age, number of siblings/spouses, number of parents/children, and cruise ticket price, and a dropdown for the embarkation port. A 'Submit' button is at the bottom of the form. To the right of the form is a 'PREDICTION TABLE' showing two entries, both predicting 'Survived' with a 79.71% chance of survival.

Titanic Survival Predictor

Would you survive the Titanic disaster? Given such information as your gender, age, and accommodation class, we'll be able to figure out your odds of survival from the doomed liner.

Would you survive the Titanic disaster?

What passenger class are you?

First Class Second Class Third Class

What is your gender?

Male Female

How old are you?

0 9 80

How many siblings and spouses were with you?

0 8

How many parents and children were aboard with you?

0 6

How much did you pay for your cruise ticket? (in 1910 USD)

\$0 \$359 \$512

Which port did you embark from?

Cherbourg Queenstown Southampton

Reset Submit

PREDICTION TABLE

Show 10 entries Search:

Chances of Survival	Prediction	Message
79.71%	Survived	Congratulations you made it out!
79.71%	Survived	Congratulations you made it out!

Showing 1 to 2 of 2 entries Previous 1 Next

Clear Tables

Loy Vanich (laploy@gmail.com 084 007 5544)

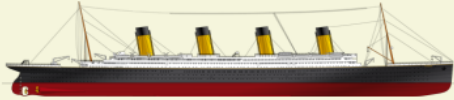
102

C# Machine Learning application

Titanic Survival Predictor C# Win App

Would you survive the Titanic disaster?

THE TITANIC DISASTER



What passenger class are you?

☒ First Class ☐ Second Class ☐ Third Class

What is your gender?

☒ Male ☐ Female

How old are you?

How many siblings and spouses were with you?

How many parents and children were aboard with you?

How much did you pay for your cruise ticket?

Which port did you embark from?

☒ Cherbourg ☐ Queenstown ☐ Southampton

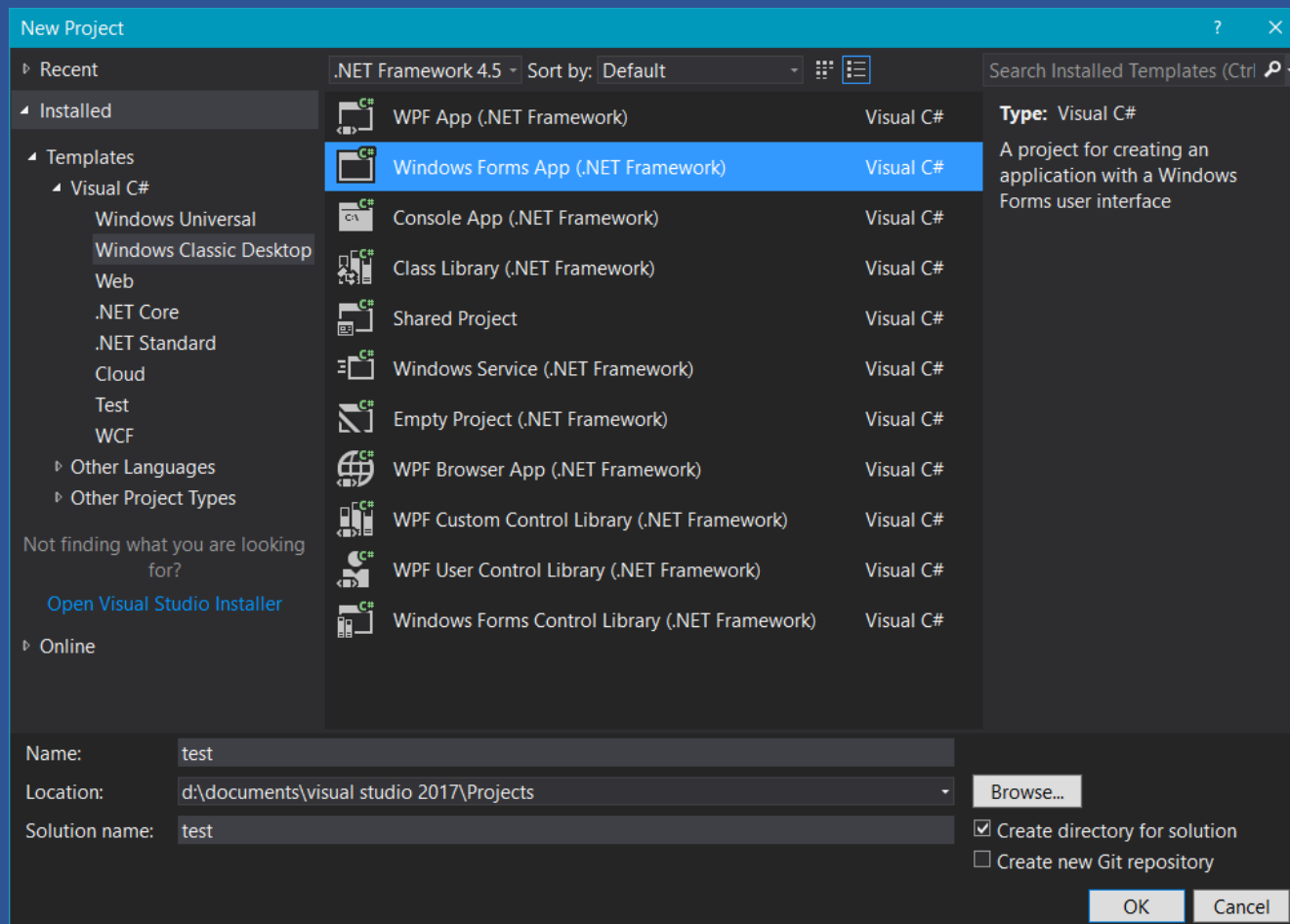
Ready

C# Machine Learning application

Create Visual C# Win form

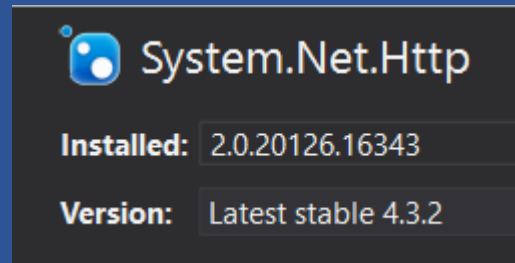
Name = **Test**

Location = **d:\temp**



C# Machine Learning application

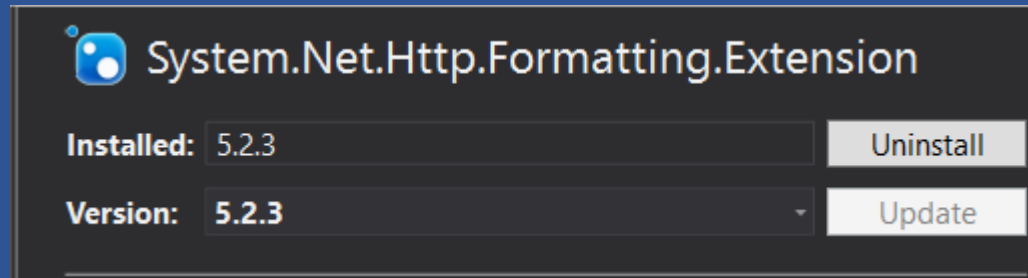
Add package **System.Net.Http**



1. Right click at project name
2. Click Manage NuGet Packages
3. Click Browse
4. Enter **System.Net.Http** in to the search box
5. Click Install

C# Machine Learning application

Add package **System.Net.Http.Formatting.Extension**



1. Right click at project name
2. Click Manage NuGet Packages
3. Click Browse
4. Enter **System.Net.Http.Formatting.Extension** in to the search box
5. Click Install

C# Machine Learning application

Create User Interface

Control naming convention

- radioButtonFirstClass
- radioButtonSecondClass
- radioButtonThirdClass

- radioButtonFemale
- radioButtonMale

- trackBarAge
- labelAge

- trackBarSib
- labelSib

- trackBarPar
- labelPar

- trackBarPay
- labelPay

- radioButtonCher
- radioButtonSout
- radioButtonQueens

- buttonReset
- buttonSubmit

- labelResult

C# Machine Learning application

Add class Input

1. Add new **Class**
2. Name = **Input**

```
10  ///</remarks>
11  namespace test
12  {
13      5 references
      public class Input...
47      2 references
      public class StringTable
48      {
49          1 reference
          public string[] ColumnNames { get; set; }
50          1 reference
          public string[,] Values { get; set; }
51      }
52  }
```

```
13      5 references
      public class Input
14      {
15          3 references
          public int PassengerId { get; set; }
16          3 references
          public int Survived { get; set; }
17          6 references
          public int Pclass { get; set; }
18          3 references
          public string Name { get; set; }
19          5 references
          public string Sex { get; set; }
20          4 references
          public int Age { get; set; }
21          4 references
          public int SibSp { get; set; }
22          4 references
          public int Parch { get; set; }
23          3 references
          public string Ticket { get; set; }
24          4 references
          public int Fare { get; set; }
25          3 references
          public string Cabin { get; set; }
26          6 references
          public string Embarked { get; set; }
          1 reference
```

C# Machine Learning application

Add class Input

Add **Constructor** and **Clear** methods

```
27 public Input()
28 {
29     Clear();
30 }
31 public void Clear()
32 {
33     PassengerId = 0;
34     Survived = 0;
35     Pclass = 0;
36     Name = "value";
37     Sex = "0";
38     Age = 1;
39     SibSp = 0;
40     Parch = 0;
41     Ticket = "value";
42     Fare = 0;
43     Cabin = "value";
44     Embarked = "value";
45 }
46 }
```

C# Machine Learning application

Add class Titanic

```
12 using System;
13 using System.Collections.Generic;
14 using System.Threading.Tasks;
15 using System.Net.Http;
16 using System.Net.Http.Headers;
17
18 namespace test
19 {
20     public class Titanic
21     {
22         public string resultMessage { get; set; }
23         const string apiKey = "key";
24         // Replace above with your API key for the web service
25         const string baseAddress = "url";
26         private string[] inputColumnName = ...;
27
28         private void CreateRequestBody(ref string[] cn, ref string[,] v, Input input) ...
29
30         private string GetResult(string result) ...
31
32         public async Task GetPrediction(Input input) ...
33     }
34 }
```

C# Machine Learning application

Add class Titanic

Copy and paste **API key** from Web Service page **Titanic 2 [predictive exp.]**

The screenshot shows the Microsoft Azure Machine Learning Studio interface. The top navigation bar includes the title 'Microsoft Azure Machine Learning Studio', a user profile dropdown 'Laploy V. Angkul-Free-Wor...', and icons for help, chat, groups, and a smiley face. The left sidebar contains icons for various ML Studio components. The main content area displays the details for the 'titanic 2 [predictive exp.]' web service. The 'General' tab is selected, showing the 'New Web Services Experience' preview. The 'Description' section states 'No description provided for this web service.' The 'API key' section shows a masked key with a yellow arrow pointing to it. Below the API key, the 'Default Endpoint' section is visible. At the bottom, there is a table with columns for 'API HELP PAGE', 'TEST', 'APPS', and 'LAST UPDATED'.

API HELP PAGE	TEST	APPS	LAST UPDATED
REQUEST/RESPONSE	Test Test preview	Excel 2013 or later Excel 2010 or earlier	6/4/2017 1:20:12 PM
BATCH EXECUTION	Test Test preview	Excel 2013 or later workbook	6/4/2017 1:20:12 PM

C# Machine Learning application

Add class Titanic

Copy and paste **Base Address** from **Request Response API Documentation for Titanic 2 [Predictive Exp.]**

```
var scoreRequest = new
{
    Inputs = new Dictionary<string, StringTable> () {
        {
            "input1",
            new StringTable()
            {
                ColumnNames = new string[] {"PassengerId", "Survived", "Pclass", "Name", "Sex", "Age", "SibSp",
                Values = new string[,] { { "0", "0", "0", "value", "value", "0", "0", "0", "value", "0", "value"
            }
        },
    },
    GlobalParameters = new Dictionary<string, string>() {
    };
    const string apiKey = "abc123"; // replace this with the API key for the web service
    client.DefaultRequestHeaders.Authorization = new AuthenticationHeaderValue( "Bearer", apiKey);

    client.BaseAddress = new Uri("https://ussouthcentral.services.azureml.net/workspaces/ede12cb3aaf24c7e826493f4e36

// WARNING: The 'await' statement below can result in a deadlock if you are calling this code from the UI thread
// One way to address this would be to call ConfigureAwait(false) so that the execution does not attempt to resu
// For instance, replace code such as:
//     result = await DoSomeTask()
// with the following:
//     result = await DoSomeTask().ConfigureAwait(false)
```



C# Machine Learning application

Add class Titanic

inputColumnName array class member

```
26 private string[] inputColumnName = {  
27     "PassengerId", "Survived", "Pclass", "Name", "Sex", "Age",  
28     "SibSp", "Parch", "Ticket", "Fare", "Cabin", "Embarked" };
```

C# Machine Learning application

Add class Titanic

Method CreateRequestBody

```
30 private void CreateRequestBody(ref string[] cn, ref string[,] v, Input input)
31 {
32     cn = input.ColumnName;
33     v = new string[,]
34     {
35         {
36             input.PassengerId.ToString(),
37             input.Survived.ToString(),
38             input.Pclass.ToString(),
39             input.Name.ToString(),
40             input.Sex.ToString(),
41             input.Age.ToString(),
42             input.SibSp.ToString(),
43             input.Parch.ToString(),
44             input.Ticket.ToString(),
45             input.Fare.ToString(),
46             input.Cabin.ToString(),
47             input.Embarked.ToString()
48         },
49         {
50             input.PassengerId.ToString(),
51             input.Survived.ToString(),
52             input.Pclass.ToString(),
53             input.Name.ToString(),
54             input.Sex.ToString(),
55             input.Age.ToString(),
56             input.SibSp.ToString(),
57             input.Parch.ToString(),
58             input.Ticket.ToString(),
59             input.Fare.ToString(),
60             input.Cabin.ToString(),
61             input.Embarked.ToString()
62         }
63     };
64 }
```

C# Machine Learning application

Add class Titanic

Method **GetResult**

```
"{"Results":{"output1":{"type":"table","value":{"ColumnNames":["Survived","PassengerClass","Gender","Age","SiblingSpouse","ParentChild","FarePrice","PortEmbarkation","Scored Labels","Scored Probabilities"],"ColumnTypes":["Int32","Int32","String","Double","Int32","Int32","Double","String","Int32","Double"],"Values":[["0","1","0","1","0","0","0","C","1","0.997975647449493"],["0","1","0","1","0","0","0","C","1","0.997975647449493"]]]}}}
```

```
65 private string GetResult(string result)
66 {
67     string s = string.Empty;
68     var cleaned = result.Replace("\"", string.Empty);
69     cleaned = cleaned.Replace("[", string.Empty);
70     cleaned = cleaned.Replace("]", string.Empty);
71     cleaned = cleaned.Replace("}", string.Empty);
72     string[] ra = cleaned.Split(",".ToCharArray());
73     if (ra[39] == "0")
74         s += "You Dead.";
75     else
76         s += "You Servived.";
77     s += " Possibility = " + ra[40].Substring(0, 5) + "%.";
78     return s;
79 }
```


C# Machine Learning application

Add class Titanic

Method `GetPrediction`

```
80 public async Task GetPrediction(Input input)
81 {
82     string[] cn = new string[12];
83     string[,] v = new string[12, 12];
84     CreateRequestBody(ref cn, ref v, input);
85     var myRequest = new
86     {
87
88         Inputs = new Dictionary<string, StringTable>() { { "input1", new StringTable() { ColumnNames = cn, Values = v } }, },
89         GlobalParameters = new Dictionary<string, string>() { }
90     };
91     var client = new HttpClient();
92     client.DefaultRequestHeaders.Authorization = new AuthenticationHeaderValue("Bearer", apiKey);
93     client.BaseAddress = new Uri(baseAddress);
94     HttpResponseMessage response = await client.PostAsJsonAsync("", myRequest);
95     if (response.IsSuccessStatusCode)
96     {
97         string result = await response.Content.ReadAsStringAsync();
98         Console.WriteLine("Result: {0}", result);
99         resultMessage = GetResult(result);
100     }
101     else
102     {
103         Console.WriteLine(string.Format("The request failed with status code: {0}", response.StatusCode));
104         // Print the headers - they include the request ID and the timestamp, which are useful for debugging the failure
105         Console.WriteLine(response.Headers.ToString());
106         string responseContent = await response.Content.ReadAsStringAsync();
107         Console.WriteLine(responseContent);
108     }
109 }
```

C# Machine Learning application

Titanic unit test

1. Open Form1 class code
2. Add test method at the bottom of the class

```
131 private async void test()  
132 {  
133     Input myInput = new Input();  
134     myInput.Age = 8;  
135     myInput.Cabin = "value";  
136     myInput.Embarked = "C";  
137     myInput.Fare = 100;  
138     myInput.Name = "value";  
139     myInput.Parch = 1;  
140     myInput.PassengerId = 0;  
141     myInput.Pclass = 1;  
142     myInput.Sex = "1";  
143     myInput.SibSp = 1;  
144     myInput.Survived = 0;  
145     myInput.Ticket = "value";  
146  
147     Titanic myTitanic = new Titanic();  
148     await myTitanic.GetPrediction(myInput);  
149     Console.WriteLine(myTitanic.resultMessage);  
150 }
```

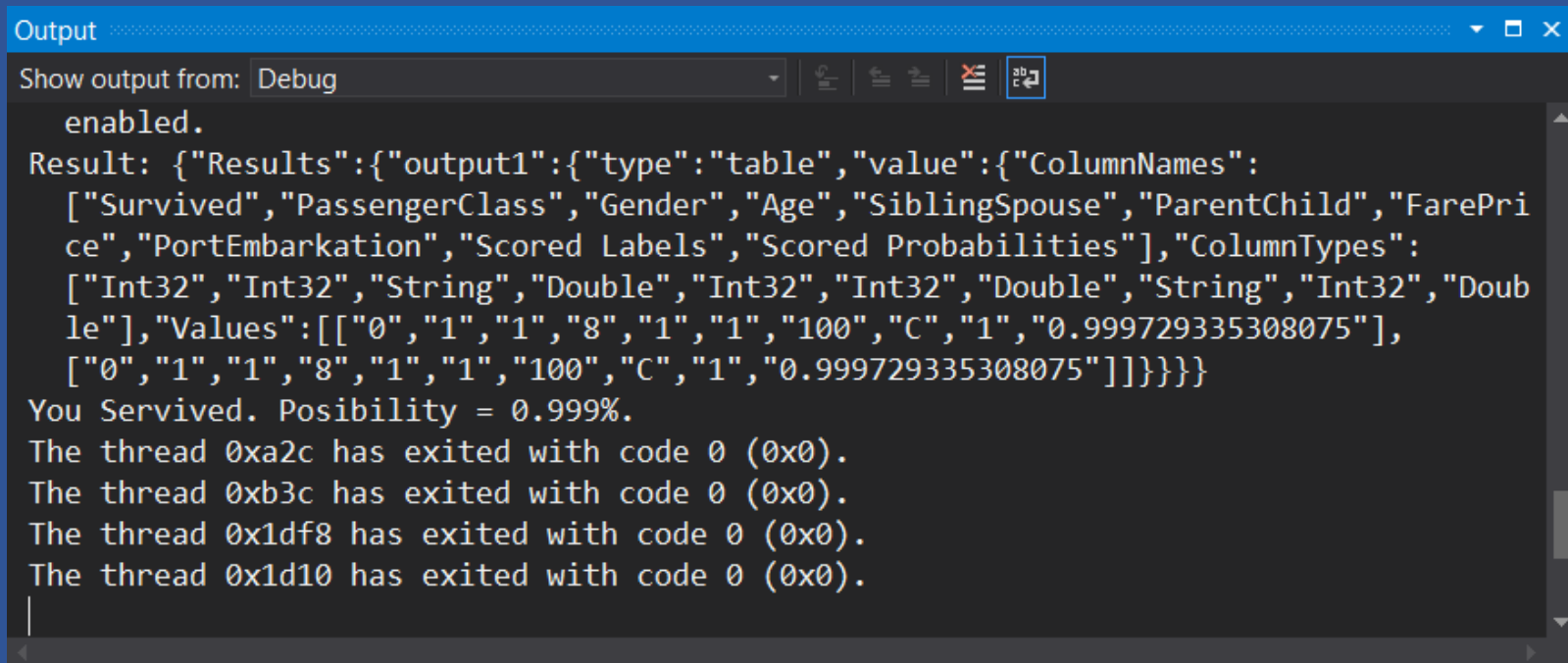
C# Machine Learning application

Titanic unit test

3. Add this code to Form Load method

```
31 private void Form1_Load(object sender, EventArgs e)
32 {
33     test();
34 }
```

4. Run program and watch Output



```
Output
Show output from: Debug
enabled.
Result: {"Results":{"output1":{"type":"table","value":{"ColumnNames":
["Survived","PassengerClass","Gender","Age","SiblingSpouse","ParentChild","FarePri
ce","PortEmbarkation","Scored Labels","Scored Probabilities"],"ColumnTypes":
["Int32","Int32","String","Double","Int32","Int32","Double","String","Int32","Doub
le"],"Values":[["0","1","1","8","1","1","100","C","1","0.999729335308075"],
["0","1","1","8","1","1","100","C","1","0.999729335308075"]]]}}}
You Servived. Posibility = 0.999%.
The thread 0xa2c has exited with code 0 (0x0).
The thread 0xb3c has exited with code 0 (0x0).
The thread 0x1df8 has exited with code 0 (0x0).
The thread 0x1d10 has exited with code 0 (0x0).
```

C# Machine Learning application

Adding code to Class Form1

Class overview

```
1  /// <summary> Titanic-Survival-Predictor
11
12  using ...
22
23  namespace test
24  {
25      3 references
26      public partial class Form1 : Form
27      {
28          private Input myInput = new Input();
29          private Titanic myTitainc = new Titanic();
30
31          1 reference
32          public Form1(...)
33
34          1 reference
35          private void Form1_Load(object sender, EventArgs e)
36
37          2 references
38          private void Reset()
39
40          Buttons
41
42          81
43          82
44          105
45          106
46          140
47          141
48      }
49  }
```

C# Machine Learning application

Adding code to Class Form1

Constructor & Form load

```
30 1 reference public Form1()  
31  {  
32      InitializeComponent();  
33      trackBarAge.Minimum = 0;  
34      trackBarAge.Maximum = 80;  
35      trackBarSib.Minimum = 0;  
36      trackBarSib.Maximum = 8;  
37      trackBarPar.Minimum = 0;  
38      trackBarPar.Maximum = 6;  
39      trackBarPay.Minimum = 0;  
40      trackBarPay.Maximum = 512;  
41  }  
42 1 reference private void Form1_Load(object sender, EventArgs e)  
43  {  
44      Reset();  
45  }
```

C# Machine Learning application

Adding code to Class Form1

Method **Reset**

```
46 private void Reset()  
47 {  
48     myInput.Clear();  
49  
50     labelResult.Text = "Ready";  
51     radioButtonFirstClass.Checked = true;  
52     radioButtonMale.Checked = true;  
53  
54     labelAge.Text = "20";  
55     trackBarAge.Value = 20;  
56  
57     labelSib.Text = "0";  
58     trackBarSib.Value = 0;  
59  
60     labelPar.Text = "0";  
61     trackBarPar.Value = 0;  
62  
63     labelPay.Text = "1";  
64     trackBarPay.Value = 1;  
65  
66     radioButtonCher.Checked = true;  
67 }
```

C# Machine Learning application

Adding code to Class Form1

Method **button click**

```
69  #region Buttons
70  1 reference
71  private void buttonReset_Click(object sender, EventArgs e)
72  {
73      Reset();
74  }
75  1 reference
76  private async void buttonSubmit_Click(object sender, EventArgs e)
77  {
78      labelResult.Text = "Processing....";
79      await myTitainc.GetPrediction(myInput);
80      labelResult.BeginInvoke(new Action(() => { labelResult.Text = myTitainc.resultMessage; }));
    }
    #endregion
```

C# Machine Learning application

Adding code to Class Form1

Method **TrackBar**

```
82  #region TrackBar
83      1 reference
84      private void trackBarAge_Scroll(object sender, EventArgs e)
85      {
86          labelAge.Text = trackBarAge.Value.ToString();
87          myInput.Age = trackBarAge.Value;
88      }
89      1 reference
90      private void trackBarSib_Scroll(object sender, EventArgs e)
91      {
92          labelSib.Text = trackBarSib.Value.ToString();
93          myInput.SibSp = trackBarSib.Value;
94      }
95      1 reference
96      private void trackBarPay_Scroll(object sender, EventArgs e)
97      {
98          labelPay.Text = trackBarPay.Value.ToString();
99          myInput.Fare = trackBarPay.Value;
100      }
101      1 reference
102      private void trackBarPar_Scroll(object sender, EventArgs e)
103      {
104          labelPar.Text = trackBarPar.Value.ToString();
105          myInput.Parch = trackBarPar.Value;
106      }
107  #endregion
```


C# Machine Learning application

Adding code to Class Form1

Method **radio button** / **passenger class**

```
105 #region RadioButtons
106 1 reference
106 private void radioButtonFirstClass_CheckedChanged(object sender, EventArgs e)
107 {
108     myInput.Pclass = 1;    //1 = 1st, 2 = 2nd, 3 = 3rd
109 }
110 1 reference
110 private void radioButtonSecondClass_CheckedChanged(object sender, EventArgs e)
111 {
112     myInput.Pclass = 2;    //1 = 1st, 2 = 2nd, 3 = 3rd
113 }
114 1 reference
114 private void radioButtonThirdClass_CheckedChanged(object sender, EventArgs e)
115 {
116     myInput.Pclass = 3;    //1 = 1st, 2 = 2nd, 3 = 3rd
117 }
118 1 reference
118 private void radioButtonMale_CheckedChanged(object sender, EventArgs e)
122 1 reference
122 private void radioButtonFemale_CheckedChanged(object sender, EventArgs e)
126 1 reference
126 private void radioButtonCher_CheckedChanged(object sender, EventArgs e)
130 1 reference
130 private void radioButtonQueens_CheckedChanged(object sender, EventArgs e)
134 1 reference
134 private void radioButtonSout_CheckedChanged(object sender, EventArgs e)
138 #endregion
```

C# Machine Learning application

Adding code to Class Form1

Method **radio button / Sex & Embark**

```
118 private void radioButtonMale_CheckedChanged(object sender, EventArgs e)
119 {
120     myInput.Sex = "0";    // male = 0, female = 1
121 }
122 private void radioButtonFemale_CheckedChanged(object sender, EventArgs e)
123 {
124     myInput.Sex = "1";    // male = 0, female = 1
125 }
126 private void radioButtonCher_CheckedChanged(object sender, EventArgs e)
127 {
128     myInput.Embarked = "C";    // C = Cherbourg, Q = Queenstown, S = Southampton
129 }
130 private void radioButtonQueens_CheckedChanged(object sender, EventArgs e)
131 {
132     myInput.Embarked = "Q";    // C = Cherbourg, Q = Queenstown, S = Southampton
133 }
134 private void radioButtonSout_CheckedChanged(object sender, EventArgs e)
135 {
136     myInput.Embarked = "S";    // C = Cherbourg, Q = Queenstown, S = Southampton
137 }
138 #endregion
```

C# Machine Learning application

More Information

Source code (MSVS 2017 Solution)

<https://github.com/laploy/Titanic-Survival-Predictor>

More information on Microsoft Azure ML Web Service

Azure Machine Learning Web Services: Deployment and consumption

<https://docs.microsoft.com/en-us/azure/machine-learning/machine-learning-deploy-consume-web-service-guide>