**TRƯỜNG ĐẠI HỌC ĐIỆN LỰC**

**KHOA CÔNG NGHỆ THÔNG TIN**



**BÁO CÁO CHUYÊN ĐỀ HỌC PHẦN**

**HỌC MÁY NÂNG CAO**

**Đề tài: TÌM HIỂU MICROSOFT ML.NET   
VÀ ỨNG DỤNG**

**Sinh viên thực hiện : LÊ THANH TÙNG**

**ĐINH DOÃN VIỆT**

**LÊ CHÍ CÔNG TRƯỜNG**

**Giảng viên hướng dẫn : PHẠM THỊ KIM DUNG**

**Ngành : CÔNG NGHỆ THÔNG TIN**

**Chuyên ngành : CÔNG NGHỆ PHẦN MỀM**

**Lớp : D12CNPM2**

**Khóa : 2017 – 2022**

***Hà Nội, tháng 1 năm 2021***

**PHIẾU CHẤM ĐIỂM**

Sinh viên thực hiện:

|  |  |  |
| --- | --- | --- |
| **Họ và tên** | **Chữ ký** | **Điểm** |
| Lê Thanh Tùng  (1781310181) |  |  |
| Đinh Doãn Việt  (1781310183) |  |  |
| Lê Chí Công Trường  (1781310076) |  |  |
|  |  |  |

Giảng viên chấm:

|  |  |  |
| --- | --- | --- |
| **Họ và tên** | **Chữ ký** | **Ghi chú** |
| Giảng viên chấm 1: |  |  |
| Giảng viên chấm 2: |  |  |

**LỜI CẢM ƠN**

Trên thực tế không có sự thành công nào mà không gắn liền với những sự hỗ trợ, sự giúp đỡ dù ít hay nhiều, dù là trực tiếp hay gián tiếp của người khác. Trong suốt thời gian từ khi bắt đầu học tập, chúng em đã nhận được rất nhiều sự quan tâm, giúp đỡ của Thầy Cô, gia đình và bạn bè.

Em xin chân thành gửi lời cảm ơn sâu sắc tới các thầy, các cô đã tận tình giảng dạy và truyền đạt cho chúng em những kiến thức, kinh nghiệm quý báu trong suốt thời gian học tập và rèn luyện.

Sau cùng, chúng em xin kính chúc các thầy cô trong Khoa Công Nghệ Thông Tin dồi dào sức khỏe, niềm tin để tiếp tục thực hiện sứ mệnh của mình là truyền đạt kiến thức cho thế hệ mai sau.

Chúng em xin chân thành cảm ơn.

***Sinh viên thực hiện***

***Lê Thanh Tùng***

***Đinh Doãn Việt***

***Lê Trí Công Trường***

**MỤC LỤC**

[**LỜI MỞ ĐẦU 2**](#_heading=h.1fob9te)

[**CHƯƠNG 1: THƯ VIỆN MACHINE LEARNING .NET 1**](#_heading=h.2et92p0)

[1.1. Khái niệm ML.NET 1](#_heading=h.tyjcwt)

[1.2. Các bước cài đặt Machine Learning .NET 1](#_heading=h.3dy6vkm)

[1.3. Cài đặt thuật toán trên ML.NET 3](#_heading=h.1t3h5sf)

[1.3.1. Cài đặt thuật toán phân cụm Kmeans trên ML.NET 3](#_heading=h.4d34og8)

[1.3.2. Cài đặt thuật toán hồi quy tuyến tính Linear Regression trên ML.NET 6](#_heading=h.2s8eyo1)

[1.4. Cài đặt bài toán sử dụng ML.NET Model Builder 9](#_heading=h.17dp8vu)

[1.4.1. Xử lý dữ liệu 9](#_heading=h.3rdcrjn)

[1.4.2. Thực hiện cài đặt bài toán sử dụng ML.NET Model Builder 10](#_heading=h.26in1rg)

[**CHƯƠNG 2. CÀI ĐẶT THUẬT TOÁN 16**](#_heading=h.lnxbz9)

[2.1. Bài toán dự đoán giá bán xe ô tô sử dụng hồi quy tuyến tính. 16](#_heading=h.35nkun2)

[2.1.1. Tập dữ liệu 16](#_heading=h.1ksv4uv)

[2.1.2. Mô tả bài toán 19](#_heading=h.44sinio)

[2.1.3. Tiền xử lý dữ liệu 19](#_heading=h.2jxsxqh)

[2.1.4. Khởi tạo mô hình học máy 20](#_heading=h.z337ya)

[s2.1.5. Thực hiện dự đoán và đánh giá thuật toán 20](#_heading=h.3j2qqm3)

[2.2. Bài toán dự đoán giá bán xe ô tô sử dụng hồi quy tuyến tính. 22](#_heading=h.1y810tw)

[2.2.1. Tập dữ liệu 22](#_heading=h.4i7ojhp)

[2.2.2. Mô tả bài toán 22](#_heading=h.2xcytpi)

[2.2.3. Tiền xử lý dữ liệu 22](#_heading=h.1ci93xb)

[2.2.4. Khởi tạo mô hình học máy 23](#_heading=h.3whwml4)

[**CHƯƠNG 3. ĐÁNH GIÁ THUẬT TOÁN 24**](#_heading=h.2bn6wsx)

[3.1. Đánh giá bài toán dự đoán giá bán xe ô tô sử dụng thuật toán học máy Decision Tree Regressor và thuật toán học máy Random Forest Regressor 24](#_heading=h.qsh70q)

[**Tài liệu tham khảo 25**](#_heading=h.3as4poj)

**Danh mục từ viết tắt**

|  |  |
| --- | --- |
| AI | Artificial Intelligence |
| CNPM | Công nghệ phần mềm |
| EPU | Đại học điện lực |
| ML.NET | Machine Learning D ot Net |

# 

# LỜI MỞ ĐẦU

Đã từ rất lâu khi máy tính bắt đầu phát triển, kéo theo sự trỗi dậy của ngành công nghệ phần mềm để phục vụ cho vô vàn những bài toán hóc búa, để nâng cao hơn nữa chất lượng cuộc sống con người. Nhũng tưởng nhhững chiếc máy tính vô hồn cũng chỉ có thể làm những công việc đơn thuần như tính toán phức tạp là cùng, thế rồi Machine learning ra đời, mang lại một cuộc cách mạng công nghệ mới cho loài người.

Bắt đầu là những thuật toán hết sức đơn giản, chỉ là dự đoán những thứ không mấy phức tạp trong cuộc sống hay chỉ là những game tự động dễ dàng. Nhưng rồi mechine learning bắt đầu bùng nổ, với những thứ được gọi là “ bộ não nhân tạo”, chúng dần làm nên những điều mà trước đó chỉ con người làm được, thậm chí vượt qua cả con người: chơi cờ vua, nhận dạng vật thể, chiến thắng cờ vây, xe tự lái,… Nhận thấy đây là một lĩnh vực mới mẻ nhưng cũng vô cùng thú vị, nhóm chúng em bao gồm Đinh Doãn Việt, Lê Thanh tùng, Lê Chí Công Trường cùng với sự giúp đỡ tận tình của cô Phạm Thị kim Dung, đã quyết định tìm hiểu về nó, thông qua bài báo cáo nhỏ này: **TÌM HIỂU MICROSOFT ML.NET VÀ ỨNG DỤNG** cùng với một số bài tập nhỏ,với mong muốn có thêm kiến thức, cũng như được tiếp cận gần hơn với Machine Learning. Với lượng kiến thức có hạn, và lĩnh vực tuy mới nhưng lại vô cùng rộng lớn và khó khăn, chúng em khó tránh khỏi sai sót, mong thầy cô có thể nhìn nhận bài báo cáo một cách nhẹ nhàng hơn, cũng như có thể góp ý cho chúng em để có thể rút kinh nghiệm cho những lần sau. Một lần nữa chúng em xin cảm ơn cô Phạm Thị Kim Dung đã nhiệt tình giúp đỡ chúng em trong thời gian qua để có thể hoàn thiện bài báo cáo này.

# 

# CHƯƠNG 1: THƯ VIỆN MACHINE LEARNING .NET

## 1.1. Khái niệm ML.NET

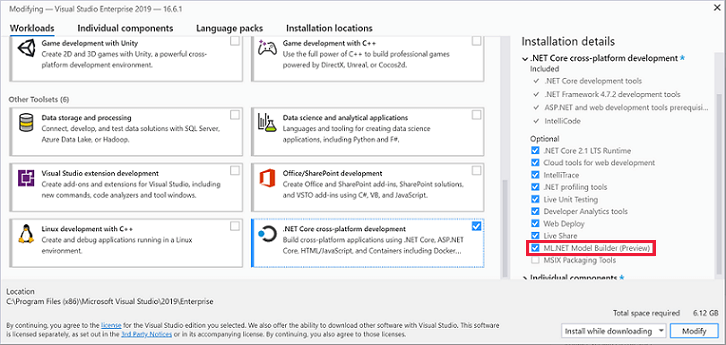
ML.NET là một thư viện Học máy chủ yếu được phát triển cho các nhà phát triển.

Chúng ta có thể sử dụng C# hoặc F# để phát triển các ứng dụng ML.NET. ML.NET là mã nguồn mở và đa nền tảng và có thể chạy trên Windows, Linux và macOS.

ML.NET vẫn đang được phát triển và bây giờ chúng ta có thể sử dụng phiên bản xem trước để làm việc và chơi với ML.NET.

## 1.2. Các bước cài đặt Machine Learning .NET

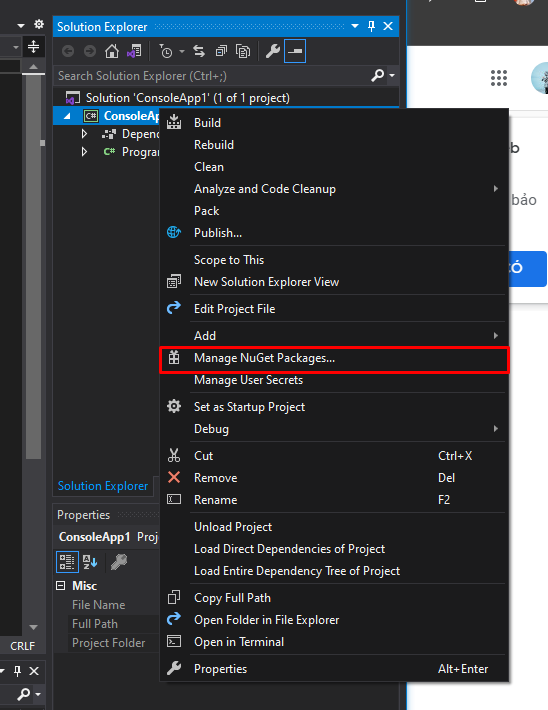
* Bước 1. Truy cập vào đường dẫn: <https://visualstudio.microsoft.com/downloads/>
* Bước 2. Thực hiện tải về Visual Studio Community
* Bước 3. Thực hiện cài đặt như trong hình:



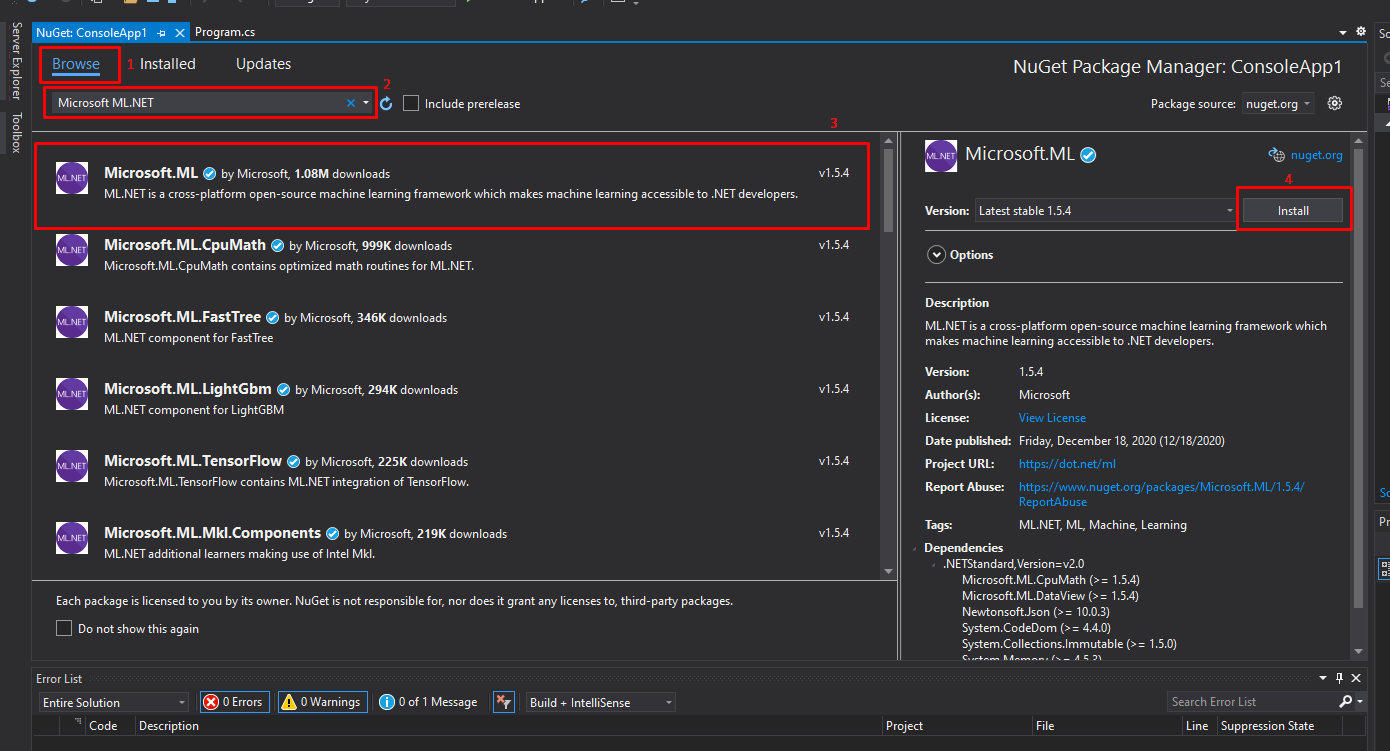
* Bước 4. Kích hoạt tính năng xây dựng ML.NET Model

Người dung truy cập vào phần **Tools > Options > Environment > Preview Features** và kích hoạt **ML.NET Model Builder.**

* Bước 5. Thực hiện tạo Project Console App và chọn Manage NuGet Packages để thực hiện thêm thư viện ML.NET



* Bước 6. Chọn vào Tab Browse, tìm kiếm thư viện Microsoft ML.NET và thực hiện cài đặt thư viện. Accept các quy định và chứng chỉ do Microsoft đưa ra để thực hiện cài đặt.



## 1.3. Cài đặt thuật toán trên ML.NET

### 1.3.1. Cài đặt thuật toán phân cụm Kmeans trên ML.NET

#### 1.3.1.1. Xử lý dữ liệu

Tập dữ liệu: Dữ liệu phân tích hoa Iris

Link: <https://archive.ics.uci.edu/ml/datasets/iris>

Mô tả dữ liệu:

|  |  |  |  |
| --- | --- | --- | --- |
| STT | Tên cột | Kiểu dữ liệu | Mô tả |
| 1 | sepal length in cm | Float | chiều dài đài hoa tính bằng cm |
| 2 | sepal width in cm | Float | chiều rộng đài hoa tính bằng cm |
| 3 | petal length in cm | Float | chiều dài cánh hoa tính bằng cm |
| 4 | petal width in cm | Float | chiều rộng cánh hoa tính bằng cm |
| 5 | class | String | loài hoa |

Dữ liệu:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Sepal length | Sepal width | Petal length | Petal width | class |
| 5.1 | 3.5 | 1.4 | 0.2 | Iris-setosa |
| 4.9 | 3 | 1.4 | 0.2 | Iris-setosa |
| 4.7 | 3.2 | 1.3 | 0.2 | Iris-setosa |
| 4.6 | 3.1 | 1.5 | 0.2 | Iris-setosa |
| 5 | 3.6 | 1.4 | 0.2 | Iris-setosa |
| 5.4 | 3.9 | 1.7 | 0.4 | Iris-setosa |
| 4.6 | 3.4 | 1.4 | 0.3 | Iris-setosa |
| … | … | … | … | … |
| 6.2 | 3.4 | 5.4 | 2.3 | Iris-virginica |
| 5.9 | 3 | 5.1 | 1.8 | Iris-virginica |

Kết luận:

* Số dữ liệu: 150
* Số thuột tính: 5
  + Sepal length
  + Sepal width
  + Petal length
  + Petal width
  + Class
* Code đọc dữ liệu bằng sử dụng ML.NET

static readonly string \_dataPath = Path.Combine(Environment.CurrentDirectory, "Data", "iris.data");

IDataView dataView = mlContext.Data.LoadFromTextFile<IrisData>(\_dataPath, hasHeader: false, separatorChar: ',');

#### 1.3.1.2. Huẩn luyện Model

* Thuật toán phân cụm: KMeans
* Số cụm: 3
* Code:

string featuresColumnName = "Features";

var pipeline = mlContext.Transforms.Concatenate  
(featuresColumnName, "SepalLength", "SepalWidth", "PetalLength", "PetalWidth").Append(mlContext.Clustering.Trainers.KMeans(featuresColumnName, numberOfClusters: 3));

var model = pipeline.Fit(dataView);

* Thực hiện lưu model Kmeans:

using (var fileStream = new FileStream(\_modelPath, FileMode.Create, FileAccess.Write, FileShare.Write))

{

mlContext.Model.Save(model, dataView.Schema, fileStream);

}

#### 1.3.1.3. Dự đoán và đánh giá

* Thực hiện dự đoán loài hoa bằng một đối tượng tùy chọn
  + Khởi tạo đối tượng Test gồm các giá trị sau:
    - SepalLength = 5.1f,
    - SepalWidth = 3.5f,
    - PetalLength = 1.4f,
    - PetalWidth = 0.2f
  + Thực hiện dự đoán sử dụng Model Kmean đã có:

var predictor = mlContext.Model.CreatePredictionEngine<IrisData, ClusterPrediction>(model);

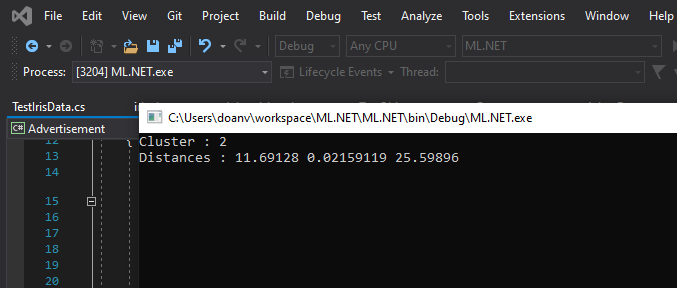
var prediction = predictor.Predict(TestIrisData.Setosa);

Console.WriteLine($"Cluster : {prediction.PredictedClusterId}");

Console.WriteLine($"Distances : {string.Join(" ", prediction.Distances)}");

Console.ReadLine();

* Kết quả dự đoán:



s

Dựa vào kết quả dự đoán: Loài hoa thuộc cụm số 2 với khoảng cách lần lượt là

* Cụm 1: 11.69128
* Cụm 2: 0.021519119
* Cụm 3: 25.59896

### 1.3.2. Cài đặt thuật toán hồi quy tuyến tính Linear Regression trên ML.NET

#### 1.3.2.1. Xử lý dữ liệu:

Tập dữ liệu: Dữ liệu quảng cáo và Sales

Link: <https://www.kaggle.com/purbar/advertising-data>

Mô tả dữ liệu:

|  |  |  |  |
| --- | --- | --- | --- |
| STT | Tên cột | Kiểu dữ liệu | Mô tả |
| 1 | TV | Float | Television (Truyền hình) |
| 2 | Radio | Float | Radio (Thông qua đài Radio) |
| 3 | Newspaper | Float | Báo |
| 4 | Sales | Float | Bán hàng |

Dữ liệu:

|  |  |  |  |
| --- | --- | --- | --- |
| TV | Radio | Newspaper | Sales |
| 230.1 | 37.8 | 69.2 | 22.1 |
| 44.5 | 39.3 | 45.1 | 10.4 |
| 17.2 | 45.9 | 69.3 | 9.3 |
| 151.5 | 41.3 | 58.5 | 18.5 |
| 180.8 | 10.8 | 58.4 | 12.9 |
| … | … | … | … |
| 293.6 | 42 | 66.2 | 25.5 |
| 232.1 | 8.6 | 8.7 | 13.4 |

* Kết luận:
  + Số dữ liệu: 200 bản ghi.
  + Số thuộc tính: 4 thuộc tính
    - TV
    - Radio
    - Newspaper
    - Sales
  + Số trường NULL, N/A : 0 trường.

- Code đọc dữ liệu từ file CSV

static readonly string \_dataPath = Path.Combine(Environment.CurrentDirectory, "Data", "Advertising.data");

IDataView dataView = mLContext.Data.LoadFromTextFile<AdverData>(\_dataPath, hasHeader: false, separatorChar: ',');

#### 1.3.2.2. Huấn luyện Model

- Thuật toán học máy hồi quy tuyến tính (Linear Regression).

- Tìm X: TV, Radio, Newspaper

- Tìm Y: Sales

- Code:

var pipeline = mLContext.Transforms.Concatenate("Features", new[] { "TV", "Radio", "Newspaper" }).Append(mLContext.Regression.Trainers.Sdca(labelColumnName: "Sales", maximumNumberOfIterations: 201));

var model = pipeline.Fit(dataView);

using (var fileStream = new FileStream(\_modelPath, FileMode.Create, FileAccess.Write, FileShare.Write))

{

mLContext.Model.Save(model, dataView.Schema, fileStream);

}

#### 1.3.2.3. Dự đoán và đánh giá

- Đối tượng dự đoán: Sales

- Khởi tạo 1 đối tượng dự đoán gồm các thuộc tính TV, Radio, Newspaper

Code:

internal static readonly AdverData Setosa = new AdverData

{

TV = 220.7f,

Radio = 27f,

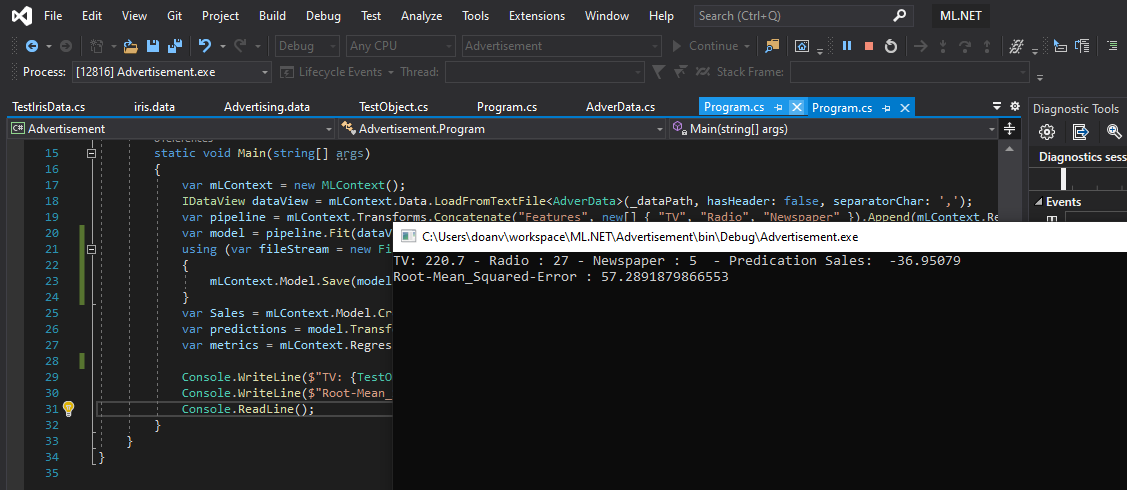
Newspaper = 5f,

};

* Thực hiện dự đoán Sales dựa trên Model đã có trước đó:

var Sales = mLContext.Model.CreatePredictionEngine<AdverData, Prediction>(model).Predict(TestObject.Setosa);

Kết quả dự đoán:



## 1.4. Cài đặt bài toán sử dụng ML.NET Model Builder

### 1.4.1. Xử lý dữ liệu

Dữ liệu được sử dụng trong ví dụ dưới đây là taxi-fare.csv

Link tải dữ liệu: <https://raw.githubusercontent.com/dotnet/machinelearning/master/test/data/taxi-fare-train.csv>

* Mô tả thuộc tính:

|  |  |  |  |
| --- | --- | --- | --- |
| ds | Tên thuộc tính | Kiểu dữ liệu | Mô tả thuộc tính |
| 1 | vendor\_id | Object | Mã tài xế |
| 2 | rate\_code | Numberic | Mã đánh giá |
| 3 | passenger\_count | Numberic | Số lượng khách hàng |
| 4 | trip\_time\_in\_secs | Numberic | Thời gian chuyến đi (giây) |
| 5 | trip\_distance | Numberic | Độ dài chuyến đi |
| 6 | payment\_type | Objects | hình thức thanh toán |
| 7 | fare\_amount | Numberic | Số tiền vé |

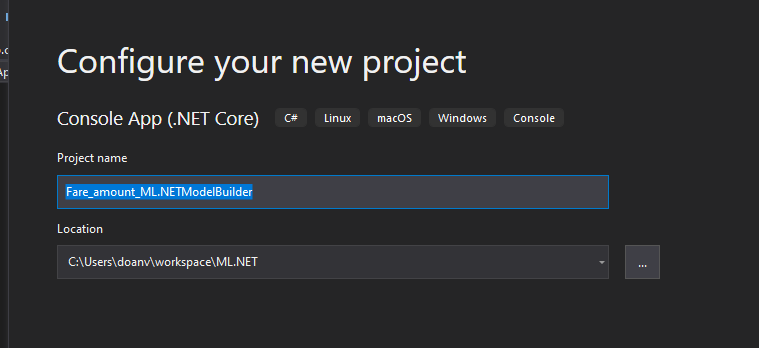
* Mô tả dữ liệu

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| vendor\_id | rate\_code | passenger\_count | trip\_time\_in\_secs | trip\_distance | payment\_type | fare\_amount |
| CMT | 1 | 1 | 1271 | 3.8 | CRD | 17.5 |
| CMT | 1 | 1 | 474 | 1.5 | CRD | 8 |
| CMT | 1 | 1 | 637 | 1.4 | CRD | 8.5 |
| CMT | 1 | 1 | 181 | 0.6 | CSH | 4.5 |
| … | … | … | … | … | … | … |
| VTS | 1 | 1 | 2340 | 4.7 | CRD | 24.5 |
| VTS | 1 | 1 | 1800 | 10.99 | CSH | 33 |

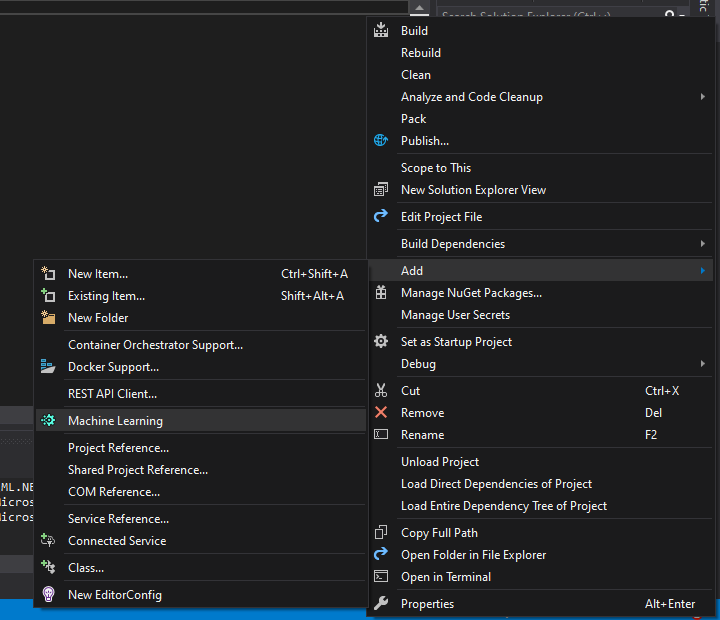
* Kết luận:
  + Tập dữ liệu gồm 1.048.575 bản ghi
  + Tập dữ liệu có 7 thuộc tính
  + Tập dữ liệu có 0 bản ghi Null/NA

### 1.4.2. Thực hiện cài đặt bài toán sử dụng ML.NET Model Builder

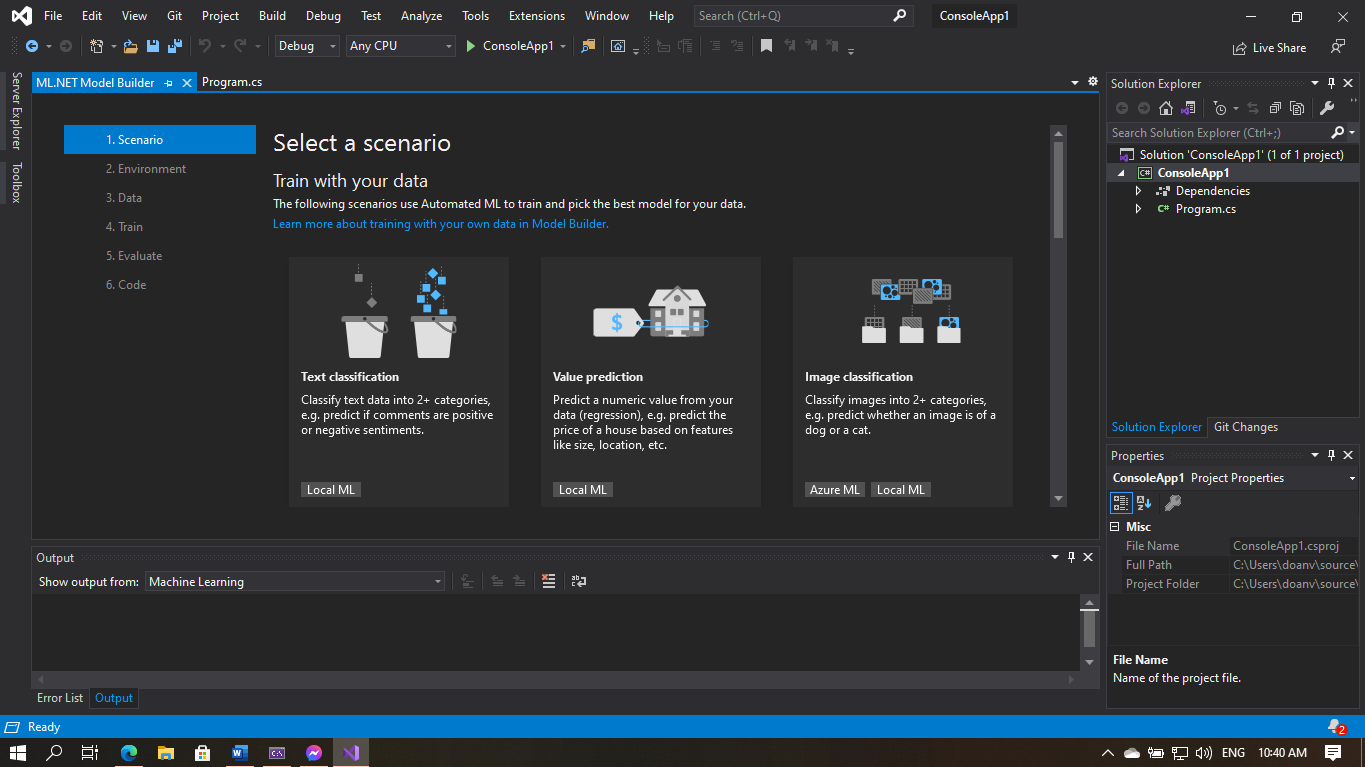
Đầu tiên, khởi tạo Project App sử dụng C# .NET Core



Chuột phải vào project -> Add -> Machine Learning



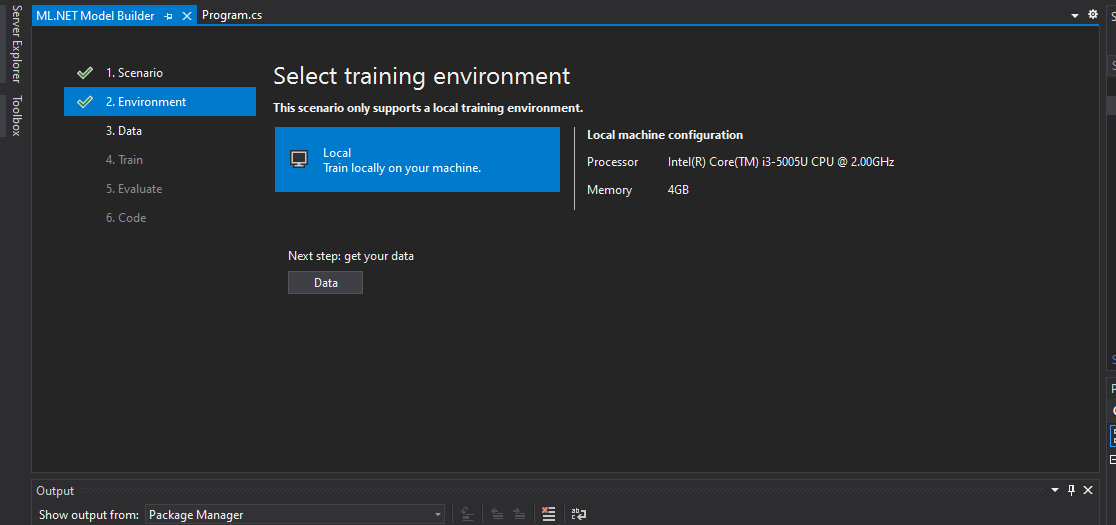
Khi đó, cửa sổ ML.NET Model Builder sẽ hiển thị lên màn hình với giao diện lựa chọn bài toán học máy



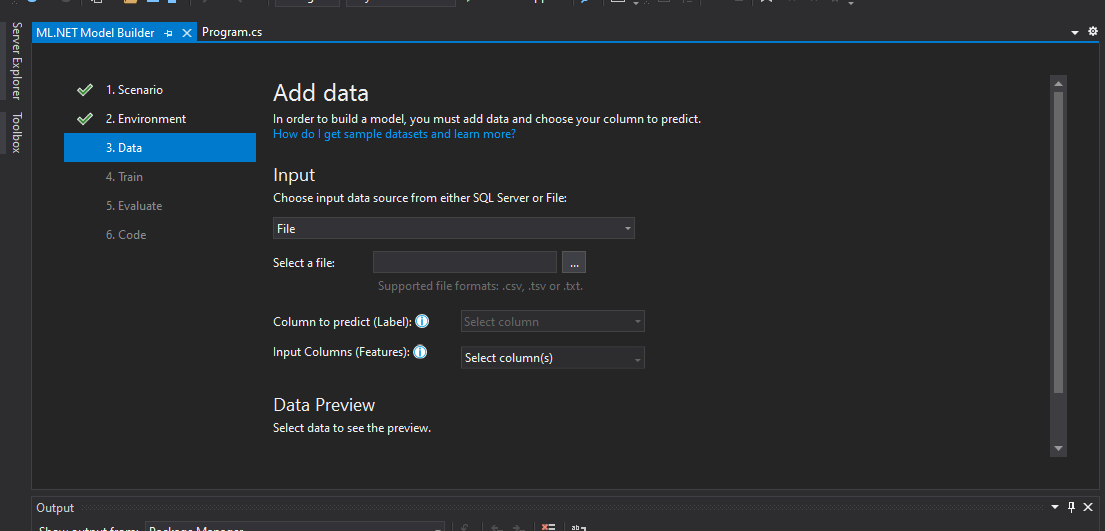
Với tập dữ liệu taxi-fare.csv, chúng ta có thể xây dựng bài toán hồi quy dự đoán tiền phí di chuyển fare\_amount sử dụng các thuộc tính trước đó.

Vậy chúng ta sẽ chọn bài toán Value Prediction.

Tiếp đó, phần mềm sẽ chuyển sang giao diện chọn môi trường học máy và thực hiện chọn tập dữ liệu.



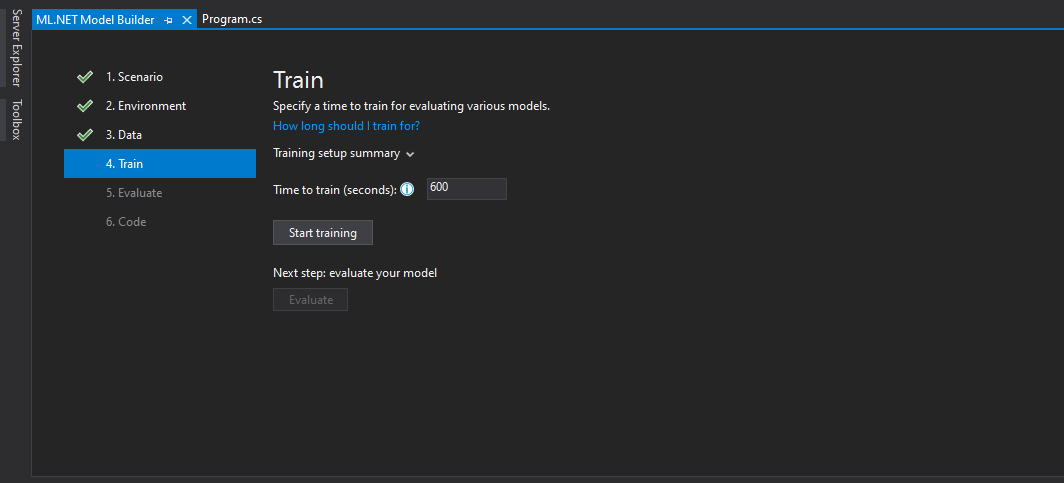
Ta chọn Local Machine và nhấm nút Data để thực hiện chọn dữ liệu.



Khi đó, giao diện dữ liệu (Data) sẽ hiện lên. Ta chọn Select a File và chọn nguồn dữ liệu Dataset cho bài toán.

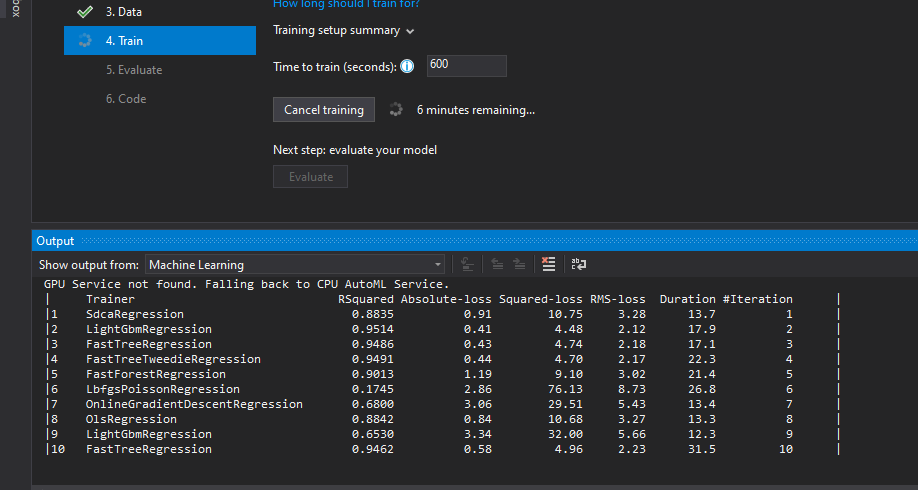
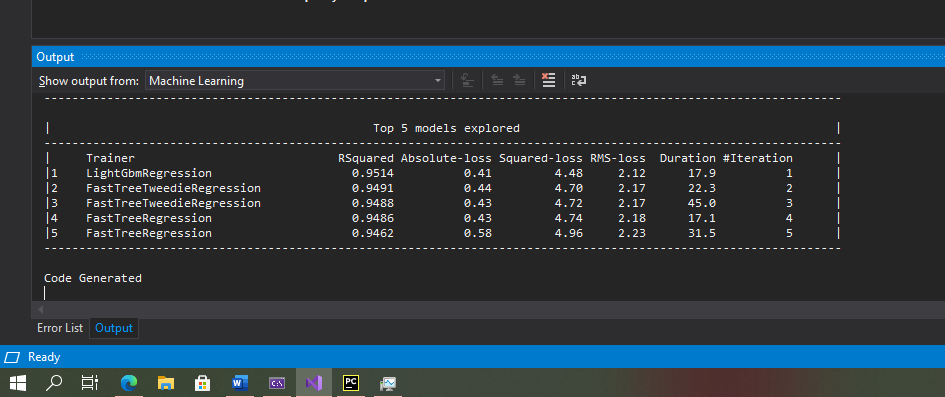
Sau đó ta chọn cột cần để dự đoán, đối với tập dữ liệu taxi-fare.csv, ta chọn cột fare\_amount để thực hiện dự đoán giá cước. Khi đó ở cột Input Column sẽ tự động tích chọn các cột còn lại để quy thành các cột thuộc tính dự đoán.

Sau đó, ta chọn vào mục 4. Train trên thanh menu.

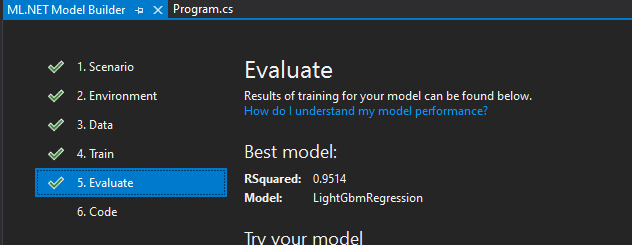


Đối với giao diện 4. Train, ta có thể cài đặt thời gian học của bài toán, thời gian càng lớn thì xác xuất chọn thuật toán học máy càng chính xác.  
Đơn vị thời gian: seconds (giây).

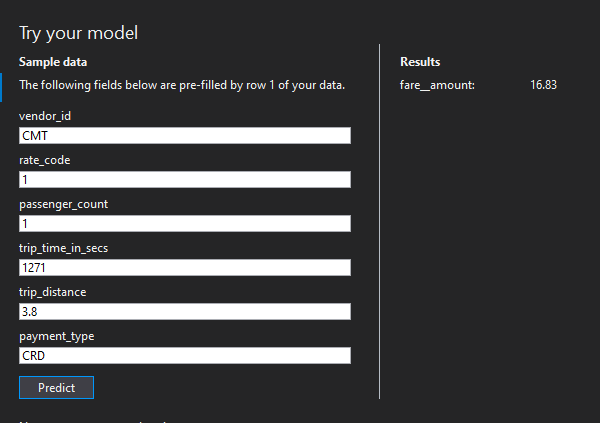
Sau khi chọn được thời gian Trainning. Ta thực hiện Chọn Start Trainning để thực hiện việc học của bài toán. Trong lúc này hệ thống sẽ tự động chọn bài toán tối ưu nhất cho tập dữ liệu và đối tượng dự đoán.

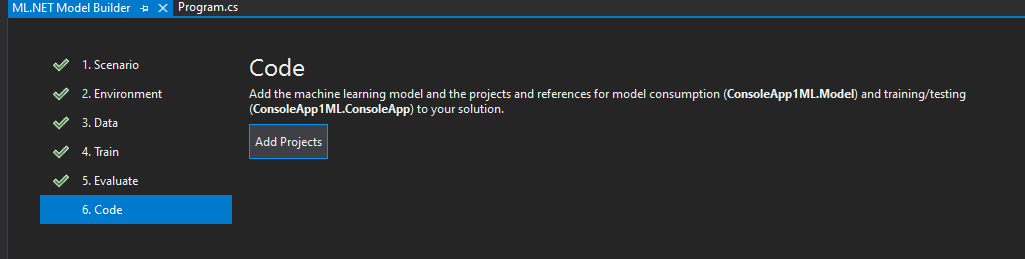
Sau khi thực hiện Trainning thành công, bài toán tối ưu nhất sẽ được hệ thống chọn.

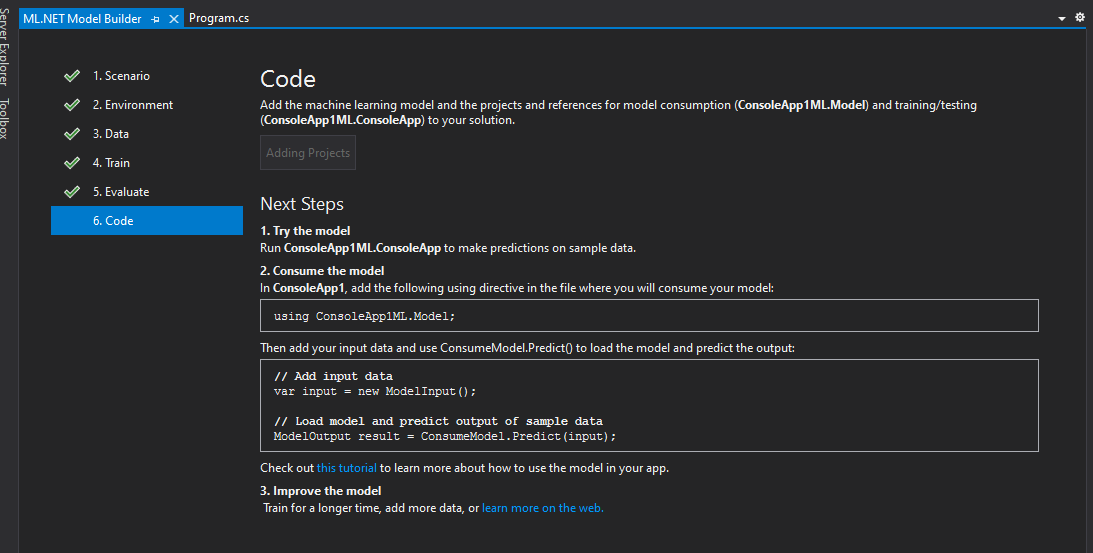


Sau đó, người dùng có thể tạo 1 đối tượng để dự đoán trong bài toán

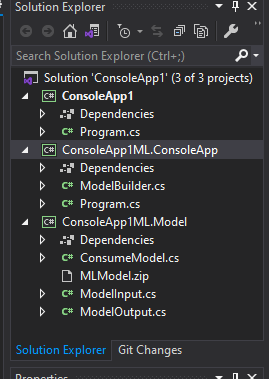


Sau đó người dùng có thể nhấn chọn nút Code để hệ thống genarate Code của bài toán.





Khi đó bên Solution Explorer sẽ hiển thị 2 Project mới do hệ thống tự động tạo ra



# CHƯƠNG 2. CÀI ĐẶT THUẬT TOÁN

## 2.1. Bài toán dự đoán giá bán xe ô tô sử dụng hồi quy tuyến tính.

### 2.1.1. Tập dữ liệu

* Tập dữ liệu sử dụng: CarPrice\_Assignment.csv
* Link đường dẫn tập dữ liệu: <https://www.kaggle.com/hellbuoy/car-price-prediction>
* Mô tả dữ liệu:
  + Các thuộc tính:

|  |  |  |  |
| --- | --- | --- | --- |
| STT | Tên Thuộc Tính | Kiểu dữ liệu | Mô Tả |
| 1 | Car\_ID | Numberic | ID của xe |
| 2 | Symboling | Numberic | Xếp hạng rủi ro (-3: có rủi ro, 3 an toàn) |
| 3 | CarCompany | Object | Hãng xe |
| 4 | fueltype | Object | Loại nhiên liệu |
| 5 | aspiration | Object | Mức nhả nhiên liệu |
| 6 | doornumber | Object | Số cửa của xe |
| 7 | Carbody | Object | Loại xe |
| 8 | drivewheel | Object | Loại bánh xe dẫn động |
| 9 | enginelocation | Object | Vị trí của động cơ |
| 10 | Wheelbase | Numberic | Chiều dài cơ sở |
| 11 | Carlength | Numberic | Chiều dài xe ô tô |
| 12 | Carwidth | Numberic | Chiều rộng xe ô tô |
| 13 | Carheight | Numberic | Chiều cao xe ô tô |
| 14 | Curbweight | Numberic | Cân nặng ô tô |
| 15 | Enginetype | Object | Loại động cơ |
| 16 | Cylindernumber | Object | Số xi-lanh động cơ |
| 17 | Enginesize | Numberic | Kích thước động cơ |
| 18 | Fuelsystem | Object | Hệ thống nhiên liệu |
| 19 | Boreratio | Numberic | Tỉ lệ lỗ khoan |
| 20 | Stroke | Numberic | Hệ số xi-lanh bên trong động cơ |
| 21 | Compressionratio | Numberic | Tỷ số nén của ô tô |
| 22 | Horsepower | Numberic | Mã lực |
| 23 | Peakrpm | Numberic | vòng / phút cao nhất của ô tô |
| 24 | Citympg | Numberic | Số dặm trong thành phố |
| 25 | Highwaympg | Numberic | Số dặm trêm đường cao tốc |
| 26 | price(Dependent variable) | Numberic | Giá cả |

* Dữ liệu:

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Tên Thuộc Tính |  |  |  |  |  |  |  |  |
| Car\_ID | 1 | 2 | 3 | 4 | … | … | 204 | 205 |
| Symboling | 3 | 3 | 1 | 2 | … | … | -1 | -1 |
| CarCompany | alfa-romero giulia | alfa-romero giulia | alfa-romero Quadrifoglio | audi 100 ls | … | … | volvo 246 | volvo 264gl |
| fueltype | gas | gas | gas | gas | … | … | Diesel | Gas |
| aspiration | std | std | std | std | … | … | Turbo | Turbo |
| doornumber | two | two | two | four | … | … | four | four |
| Carbody | convertible | convertible | hatchback | sedan | … | … | sedan | sedan |
| drivewheel | rwd | rwd | rwd | rwd | … | … | rwd | rwd |
| enginelocation | front | front | front | front | … | … | front | front |
| Wheelbase | 88.6 | 88.6 | 94.5 | 99.8 | … | … | 109.1 | 109.1 |
| Carlength | 168.8 | 168.8 | 171.2 | 176.6 | … | … | 188.8 | 188.8 |
| Carwidth | 64.1 | 64.1 | 65.5 | 66.2 | … | … | 68.9 | 68.9 |
| Carheight | 48.8 | 48.8 | 52.4 | 54.3 | … | … | 55.5 | 55.5 |
| Curbweight | 2548 | 2548 | 2823 | 2337 | … | … | 3217 | 3062 |
| Enginetype | dohc | dohc | ohcv | ohc | … | … | ohc | ohc |
| Cylindernumber | four | four | six | four | … | … | six | four |
| Enginesize | 130 | 130 | 152 | 109 | … | … | 145 | 141 |
| Fuelsystem | mpfi | mpfi | mpfi | mpfi | … | … | idi | mpfi |
| Boreratio | 3.47 | 3.47 | 2.68 | 3.19 | … | … | 3.01 | 3.78 |
| Stroke | 2.68 | 2.68 | 3.47 | 3.4 | … | … | 3.4 | 3.15 |
| Compressionratio | 9 | 9 | 10 | 8 | … | … | 23 | 9.5 |
| Horsepower | 111 | 111 | 154 | 102 | … | … | 105 | 114 |
| Peakrpm | 5000 | 5000 | 5000 | 5500 | … | … | 4800 | 5400 |
| Citympg | 21 | 21 | 19 | 24 | … | … | 26 | 19 |
| Highwaympg | 27 | 27 | 26 | 30 | … | … | 27 | 25 |
| price(Dependent variable) | 13495 | 16500 | 16500 | 13950 | … | … | 22470 | 22625 |

* sKết luận:
  + Số lượng bản ghi: 205 bản ghi.
  + Số lượng thuộc tính: 26 thuộc tính.
  + Số trường Null, N/A: 0.

### 2.1.2. Mô tả bài toán

* Dựa vào tập dữ liệu đã được biểu hiện ở trên, chúng ta có thể dự đoán Giá bán xe dựa trên 25 thuộc tính còn lại.
* Tìm kiếm X, Y:
  + X gồm: symboling, fueltype, aspiration, doornumber, carbody, drivewheel, enginelocation, wheelbase, carlength, carwidth, carheight, curbweight, enginetype, cylindernumber, enginesize, fuelsystem, boreratio, stroke, compressionratio, horsepower, peakrpm, citympg, highwaympg.
  + Y gồm: price
* Chia tập train/test với tỉ lệ 80/20.
  + Tập train: 164 bản ghi.
  + Tập test: 41 bản ghi.
* Bài toán học máy sử dụng trong bài: Regression
* Kỹ thuật học máy được sử dụng là: Linear Regression, Decision Tree Regressor, Random Forest Regressor.
* Phương pháp đánh giá: mean squared error (MSE) – so sánh sai số

### 2.1.3. Tiền xử lý dữ liệu

- Thực hiện đọc file dataset

dataset = pd.read\_csv('Data/CarPrice\_Assignment.csv')

* Xử lý dữ liệu bị thiếu N/A: Sử dụng fillna() hoặc dropna().

dataset.dropna() #Xóa giá trị N/A  
dataset=dataset.fillna(dataset.mean()) #giá trị trung bình

* Xử lý dữ liệu category: sử dụng LabelEncoder thông qua thư viện Sklearn
* X = dataset.apply(lambda col: preprocessing.LabelEncoder().fit\_transform(col))

### 2.1.4. Khởi tạo mô hình học máy

- Thực hiện chia tập train/test:

#Chia tập train test  
X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.2, random\_state=42)

* Decision Tree Regressor
* from sklearn.tree import DecisionTreeRegressor  
  regressor = DecisionTreeRegressor(random\_state=0)  
  regressor.fit(X\_train,y\_train)  
  y\_train\_pred = regressor.predict(X\_train)  
  y\_test\_pred = regressor.predict(X\_test)
* Random Forest Regressor
* from sklearn.ensemble import RandomForestRegressor  
  Rand\_frst = RandomForestRegressor(criterion = 'mse',  
   random\_state = 20,  
   n\_jobs = -1)  
  Rand\_frst.fit(X\_train,y\_train)  
  Rand\_frst\_train\_pred = Rand\_frst.predict(X\_train)  
  Rand\_frst\_test\_pred = Rand\_frst.predict(X\_test)  
  y\_test\_predict = Rand\_frst.predict(X\_test)

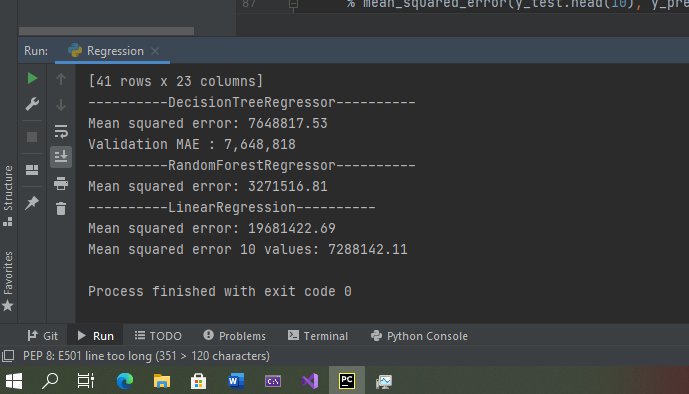
### 2.1.5. Thực hiện dự đoán và đánh giá thuật toán

* Thực hiện tính sai số và hiển thị sơ đồ:
  + Sai số của thuật toán Decision Tree Regressor:

#sai số MSE  
print('----------DecisionTreeRegressor----------');  
print('Mean squared error: %.2f'  
 % mean\_squared\_error(y\_test, y\_test\_pred))  
print("Validation MAE : {:,.0f}".format(val\_mae))

* + Sai số thuật toán Random Forest Regressor:

print('----------RandomForestRegressor----------');  
print('Mean squared error: %.2f'  
 % mean\_squared\_error(y\_test, Rand\_frst\_test\_pred))



Dựa vào kết quả ở trên, sai số MSE của Random Forest Regressor là 3.271.516 đạt giá trị nhỏ nhất so với Descision Tree Regressor (7.648.818).

Vậy, thuật toán Random Forest Regressor là thuật toán tối ưu trong bài toán dự đoán giá xe.

## 2.2. Bài toán dự đoán giá bán xe ô tô sử dụng hồi quy tuyến tính.

### 2.2.1. Tập dữ liệu

* Tập dữ liệu sử dụng: Car\_ad.csv
* Link đường dẫn tập dữ liệu: [Car Sale Advertisements | Kaggle](https://www.kaggle.com/antfarol/car-sale-advertisements?fbclid=IwAR1v3PiCLYRIYV0UaeXYVEII_rCHuTHGqqMaCiV4i9mARSrjuvbtZYuJrhU)
* Mô tả dữ liệu:
  + Các thuộc tính:

|  |  |  |  |
| --- | --- | --- | --- |
| STT | Tên Thuộc Tính | Kiểu dữ liệu | Mô Tả |
| 1 | Car | object | Tên xe |
| 2 | Price | float64 | giá của người bán trong quảng cáo (bằng USD) |
| 3 | Body | object | Mẫu mã |
| 4 | mileage | int64 | Số dặm |
| 5 | engV | float64 | Mã lực |
| 6 | engType | object | Nguyên liệu |
| 7 | registration | object | Đăng ký |
| 8 | year | int64 | năm sản xuất |
| 9 | model | object | Loại xe |
| 10 | drive | object | loại ổ đĩa |

### 2.2.2. Mô tả bài toán

* Dựa vào tập dữ liệu đã được biểu hiện ở trên, chúng ta có thể dự đoán Giá bán xe dựa trên 25 thuộc tính còn lại.
* Tìm kiếm X, Y:
  + X gồm: body, mileage, engV, engType, registration, year, model, drive.
  + Y gồm: price
* Bài toán học máy sử dụng trong bài: Regression
* Kỹ thuật học máy được sử dụng là: Linear Regression, Decision Tree Regressor
* Phương pháp đánh giá: mean squared error (MSE) – so sánh sai số

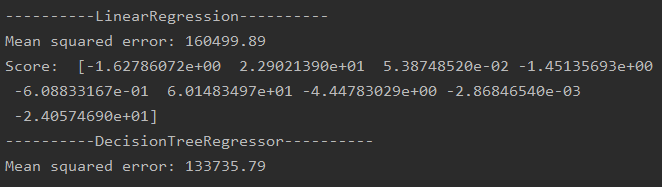
### 2.2.3. Tiền xử lý dữ liệu

* Xử lý dữ liệu bị thiếu N/A: Sử dụnghoặc dropna().
* import pandas as pd  
  import matplotlib.pyplot as plt  
  df = pd.read\_csv("Data/car\_ad.csv", sep=',', encoding='latin-1')  
    
  df.dropna()  
  df['car'] = df['car'].str.split(' ', expand=True)  
  df.head()
* Xử lý dữ liệu category: sử dụng LabelEncoder thông qua thư viện Sklearn
* from sklearn import preprocessing  
  df = df.apply(lambda col: preprocessing.LabelEncoder().fit\_transform(col.astype(str)))

### 2.2.4. Khởi tạo mô hình học máy

* Linear Regession:
* from sklearn.linear\_model import LinearRegression  
  model = LinearRegression()  
  model.fit(X\_train, y\_train)  
  y\_predict = model.predict(X\_test)
* Decision Tree Regressor
* from sklearn.tree import DecisionTreeRegressor  
  regressor = DecisionTreeRegressor(random\_state=0)  
  regressor.fit(X\_train, y\_train)  
  y\_predict = regressor.predict(X\_test)

- Kết quả trả ra :



Dựa vào kết quả ở trên, sai số MSE của Linear Regression là (160499.89) với Descision Tree Regressor (133735.79) do đó, là thuật toán tối ưu trong bài toán dự đoán so với bộ dữ liệu car\_ad.csv.

## 2.3. Bài toán dự đoán học sinh học cao học.

### 2.3.1. Tập dữ liệu

Bộ dữ liệu được tải về từ kaggle, link tải dữ liệu:

<https://www.kaggle.com/larsen0966/student-performance-data-set>

Dữ liệu này tiếp cận thành tích của học sinh trong giáo dục trung học của hai trường học ở Bồ Đào Nha. Các thuộc tính dữ liệu bao gồm điểm học sinh, các đặc điểm liên quan đến nhân khẩu học, xã hội và trường học) và nó được thu thập bằng cách sử dụng báo cáo của trường và bảng câu hỏi. Hai tập dữ liệu được cung cấp liên quan đến thành tích trong hai môn học riêng biệt: Toán học (mat) và ngôn ngữ Bồ Đào Nha (por).

Bộ dữ liệu gồm 649 bộ dữ liệu,với 33 cột thuộc tính:

school : (binary: 'GP' - Gabriel Pereira or 'MS' - Mousinho da Silveira): trường học của học sinh: 'GP' - Gabriel Pereira và 'MS' - Mousinho da Silveira

sex: (binary: 'F' - female or 'M' - male) : giới tính học sinh, F là nữ, M là nam

age (numeric: from 15 to 22): tuổi học sinh, từ 15 – 22 tuổi

address : (binary: 'U' - urban or 'R' - rural) vị trí nhà ở: 'U' - urban or 'R' - rural

famsize : (binary: 'LE3' - less or equal to 3 or 'GT3' - greater than 3): số thành viên trong nhà: LE3: nhỏ hơn 3 và GT3: lớn hơn 3

Pstatus : (binary: 'T' - living together or 'A' - apart): tình trạng quan hệ của bố mẹ: T: sống chung, A : li dị

Medu: (numeric: 0 - none, 1 - primary education (4th grade), 2 â€“ 5th to 9th grade, 3 â€“ secondary education or 4 â€“ higher education)Tình trạng giáo dục của mẹ

Fedu:father's education (numeric: 0 - none, 1 - primary education (4th grade), 2 â€“ 5th to 9th grade, 3 â€“ secondary education or 4 â€“ higher education): Tình trạng giáo dục của bố

Mjob: (nominal: 'teacher', 'health' care related, civil 'services' (e.g. administrative or police), 'at\_home' or 'other'): nghề của mẹ

Fjob: (nominal: 'teacher', 'health' care related, civil 'services' (e.g. administrative or police), 'at\_home' or 'other'): nghề của bố

Reason: (nominal: close to 'home', school 'reputation', 'course' preference or 'other'): lí do chọn trường

guardian (nominal: 'mother', 'father' or 'other'): người giám hộ

traveltime (numeric: 1 - <15 min., 2 - 15 to 30 min., 3 - 30 min. to 1 hour, or 4 - >1 hour): thời gian đến trường

studytime (numeric: 1 - <2 hours, 2 - 2 to 5 hours, 3 - 5 to 10 hours, or 4 - >10 hours): thời gian học trong 1 tuần

failures (numeric: n if 1<=n<3, else 4): số lần thi lên lớp bị trượt

schoolsup (binary: yes or no): hỗ trợ học tập từ phía nhà trường

famsup(binary: yes or no): hỗ trợ học tập từ phía gia đình

paid (Math or Portuguese) (binary: yes or no): các lớp học trả phí bổ sung trong môn học

activities (binary: yes or no) hoạt động ngoại khóa

nursery (binary: yes or no)học trường mẫu giáo

higher (binary: yes or no)muốn học cao hơn

internet (binary: yes or no): có kết nối internet tại nhà

romantic(binary: yes or no): đang có 1 mối quan hệ lãng mạn

famrel (numeric: from 1 - very bad to 5 - excellent): mối quan hệ gia đình

freetime(numeric: from 1 - very low to 5 - very high): thời gian rảnh rỗi sau giờ học

goout (numeric: from 1 - very low to 5 - very high)tần suất ra ngoài với bạn bè

Dalc (numeric: from 1 - very low to 5 - very high): uống rượu trong ngày làm việc

Walc(numeric: from 1 - very low to 5 - very high): uống rượu cuối tuần

health (numeric: from 1 - very bad to 5 - very good)trạng thái sức khỏe hiện tại

absences absences (numeric: from 0 to 93) số lần nghỉ học

G1 (numeric: from 0 to 20) khối 1

G2 (numeric: from 0 to 20) khối 2

G3 (numeric: from 0 to 20, output target) khối 3

### 2.3.2. Mô tả bài toán

Bài toán dự đoán một học sinh có quyết định học lên đại học không dựa trên các yếu tố nhân khẩu học của học sinh đó như nhà trường, gia đình và xã hội cùng với các hoạt động của họ tại trường và nhà.

### 2.3.3. Tiền xử lý dữ liệu

- Thực hiện đọc file dataset

* import pandas as pd
* # doc file du lieu csv
* df = pd.read\_csv('student-por.csv')
* # xu ly du lieu
* #1. Loại bỏ/fill Dữ liệu NaN
* df = df.dropna()
* # Xử lý dữ liệu Category (LabelEncoder, OneHotEncoder/dummy)
* from sklearn.preprocessing import LabelEncoder
* lb\_make = LabelEncoder()
* df\_encoder = df.apply(lb\_make.fit\_transform)

### 2.3.4. Khởi tạo mô hình học máy

# sap xep va chia train test

df\_encoder = df\_encoder.sort\_values('higher', ascending = True)

Y\_train = df\_encoder.iloc[0:69,20:21].append(df\_encoder.iloc[79:,20:21])

Y\_test = df\_encoder.iloc[69:79,20:21]

# print(yTrain)

df\_encoder = df\_encoder.drop('higher', axis = 1)

X\_train = df\_encoder.iloc[0:69].append(df\_encoder.iloc[79:])

X\_test = df\_encoder.iloc[69:79]

### 2.3.5. Thực hiện dự đoán và đánh giá thuật toán

Có hai thuật toán được sử dụng trong bài toán này, đó là hồi quy logistic và phân lớp nhị phân bằng thuật toán MVC của sklearn

########logictis

from sklearn.linear\_model import LogisticRegression

from sklearn.metrics import accuracy\_score

from sklearn.metrics import classification\_report

model = LogisticRegression(max\_iter = 10000).fit(X\_train, Y\_train)

Y\_hat = model.predict(X\_test)

accuracy = accuracy\_score(Y\_test, Y\_hat)

print("============== LR =====================")

print('Accuracy: %.2f' % (accuracy\*100))

print(classification\_report(Y\_test,Y\_hat))

######## MVC

from sklearn.svm import SVC

model\_SVC = SVC()

model\_SVC.fit(X\_train, Y\_train)

# test MVC

predict = model\_SVC.predict(X\_test)

print("============== MVC =====================")

ac\_score = accuracy\_score(Y\_test, predict)

cl\_report = classification\_report(Y\_test, predict)

print("Score = ", ac\_score)

print(cl\_report)

# CHƯƠNG 3. ĐÁNH GIÁ THUẬT TOÁN

## 3.1. Đánh giá bài toán dự đoán giá bán xe ô tô sử dụng thuật toán học máy Decision Tree Regressor và thuật toán học máy Random Forest Regressor

Bài toán dự đoán giá bán xe ÔTÔ sử dụng hồi quy (Random Forest Regressor) với tập dữ liệu gồm 205 bản ghi dữ liệu và 26 thuộc tính.

Dựa vào tập dữ liệu, ta chia được X, Y gồm các thuộc tính cần thiết để có thể giải quyết được bài toán.

Bài toán được chia tập Train/Test với tỉ lệ 80/20 ta có 164 bản ghi Train và 41 bản ghi Test.

Sử dụng bài toán Random Forest Regressor, ta huấn luyện được một Model bài toán để thực hiện dự đoán giá bán xe dựa trên tập huấn luyện trước đó. Để có thể đánh giá được tính tối ưu của bài toán, ta có thể so sánh Trung bình sai số bình phương (Mean Squared Error) và thực hiện so sánh với các thuật toán học máy khác (cụ thể trong bài là thuật toán Decision Tree Regressor).

Do vậy, thuật toán Random Forest Regressor có tính tối ưu, hiệu quả hơn.

Trong tương lai, tập dữ liệu sẽ được bổ sung và hoàn thiện hơn để phù hợp với thực tế xu hướng và được sử dụng cách kỹ thuật mới hiệu quả hơn thì bài toán sẽ được giải quyết hiệu quả hơn.

## 3.2. Đánh giá bài toán dự đoán giá bán xe ô tô sử dụng thuật toán học máy Linear Regression và thuật toán học máy Decision Tree Regressor

Bài toán dự đoán giá bán xe sử dụng hồi quy (Linear Regression) với tập dữ liệu gồm 9576  bản ghi dữ liệu và 10 thuộc tính.

Dựa vào tập dữ liệu, ta chia được X, Y gồm các thuộc tính cần thiết để có thể giải quyết được bài toán.

Bài toán được chia tập Train/Test với tỉ lệ 0,01 ta có 9480 bản ghi Train và 96 bản ghi Test.

Sử dụng bài toán Linear Regression, ta huấn luyện được một Model bài toán để thực hiện dự đoán giá bán xe dựa trên tập huấn luyện trước đó. Để có thể đánh giá được tính tối ưu của bài toán, ta có thể so sánh Trung bình sai số bình phương (Mean Squared Error) và thực hiện so sánh với các thuật toán học máy khác (cụ thể trong bài là thuật toán Decision Tree Regressor). Tuy nhiên, Linear Regression là nó rất **nhạy cảm với nhiễu** (sensitive to noise) và  Linear Regression là nó **không biễu diễn được các mô hình phức tạp.** Mà đối với bộ dữ liệu car\_ad.csv có độ nhiễu cao nên thuật toán Linear Regression không có độ đáng tin cao bằng thuật toán Decision Tree Regressor.

Do vậy, thuật toán Decision Tree Regressor có tính tối ưu, hiệu quả hơn.

Trong tương lai, tập dữ liệu sẽ được bổ sung và hoàn thiện hơn để phù hợp với thực tế xu hướng và được sử dụng cách kỹ thuật mới hiệu quả hơn thì bài toán sẽ được giải quyết hiệu quả hơn.

## . Đánh giá bài toán dự đoán chọn lên đại học của học sinh

## Bài toán với sự chênh lệch dữ liệu lớn, hiện chưa thể có 1 kết quả khả quan với bộ dữ liệu này.

## Tỷ lệ dự đoán của mô hình còn rất thấp, cụ thể :

## LogistícRegresion

## ============== LR =====================

## Accuracy: 55.00

## precision recall f1-score support

## 0 1.00 0.10 0.18 10

## 1 0.53 1.00 0.69 10

## accuracy 0.55 20

## macro avg 0.76 0.55 0.44 20

## weighted avg 0.76 0.55 0.44 20

## 

MVC

Score = 0.5

precision recall f1-score support

0 0.00 0.00 0.00 10

1 0.50 1.00 0.67 10

accuracy 0.50 20

macro avg 0.25 0.50 0.33 20

weighted avg 0.25 0.50 0.33 20

Như có thể thấy thuật toán gần như ko thể dự đoán được số học sinh chọn nghỉ học do dữ liệu này quá ít. Phương pháp xử lý: có thể xử lý lại dữ liệu bằng thuật toán encoder tốt hơn, thử dùng smote.

# Tài liệu tham khảo

1.Hang Nguyen Thuy ,Tutorial từ Qgis bản tiếng Việt .*2010*.

2.Xuân Cường,Website: https://cuongdx313.wordpress.com,2016.