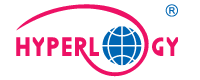


|  |
| --- |
| **Internship report** **2021** |
|  |
| JUNE - 18  Hyperlogy technology company  Authored by: Lang Duc Hai |



# ĐỀ TÀI

Đánh giá các mô hình học có giám sát   
sử dụng đặc trưng HOG

Contents

[ĐỀ TÀI 2](#_Toc75479706)

[I. GIỚI THIỆU 3](#_Toc75479707)

[1. Các mô hình học có giám sát (Supervised learning): 3](#_Toc75479708)

[2. Đặc trưng HOG: 3](#_Toc75479709)

[II. Phương pháp xây dựng: 4](#_Toc75479710)

[1. Xử lý Input: 4](#_Toc75479711)

[1.1. Tạo đặc trưng HOG trong hàm create\_HOG\_description: 4](#_Toc75479712)

[1.2. Xử lý tập dữ liệu đầu vào: 4](#_Toc75479713)

[1.3. Hàm chuyển đổi data và labels về dữ liệu để huấn luyện và kiểm tra: 6](#_Toc75479714)

[1.4. Lưu tập huấn luyện và đào tạo vào package Files: 7](#_Toc75479715)

[2. Lấy dữ liệu từ package: 7](#_Toc75479716)

[3. Xây dựng các mô hình học có giám sát: 8](#_Toc75479717)

[3.1. Mô hình Logistic Regression (Hồi quy Logistic) 8](#_Toc75479718)

[3.2. Mô hình Decision Tree (Cây quyết định) 8](#_Toc75479719)

[3.3. Mô hình Naïve Bayes classifiers (Bộ phân loại bayes) 9](#_Toc75479720)

[3.4. Mô hình Support Vector Machines (SVM) 9](#_Toc75479721)

[4. Khởi chạy kết quả: 9](#_Toc75479722)

[III. Đánh giá mô hình: 11](#_Toc75479723)

[1. Kết quả 11](#_Toc75479724)

[2. Đánh giá 11](#_Toc75479725)

[3. BẢNG ĐÁNH GIÁ TỔNG QUÁT CÁC MÔ HÌNH PHÂN LỚP 12](#_Toc75479726)

# 

# I. GIỚI THIỆU

## 1. Các mô hình học có giám sát (Supervised learning):

- Là dạng học máy trong đó cho trước tập dữ liệu huấn luyện dưới dạng các ví dụ cùng với giá trị đầu ra hay giá trị đích. Dựa trên dữ liệu huấn luyện, thuật toán học cần xây dựng mô hình hay hàm đích để dự đoán giá trị đầu ra (giá trị đích) cho các trường hợp mới.

- Ở đây chúng ta sẽ đề cập so sánh giữa 4 mô hình cơ bản gồm:

Logistic Regression (Hồi quy logistic)

Decision Tree (Cây quyết định)

Naïve Bayes classifiers (Bộ phân loại bayes)

Support Vector Machines (SVM)

## 2. Đặc trưng HOG:

- Là phương pháp chủ yếu để mô tả đặc trưng của hình ảnh.

- Ngày nay HOG không còn dược sử dụng nhiều nữa. Tuy nhiên, đối với các bộ dữ liệu kích thước nhỏ, HOG có thể được sử dụng để tạo đặc trưng đầu vào cho các thuật toán học có giám sát cổ điển ở trên mà vẫn mang lại độ chính xác cao, quá trình huấn luyện nhanh và yêu cầu ít tài nguyên tính toán.

# II. Phương pháp xây dựng:

Sau đây chúng ta cùng thực hành xây dựng mộ mô hình phân loại nhãn hiệu xe dựa trên ảnh logo thông qua việc trích suất đặc trưng HOG. Dữ liệu gồm trên bức ảnh logo của 10 thương hiệu xe được chia làm 2 tập train và test. Tập train gồm 1400 ảnh với 140 ảnh/mỗi thương hiệu và tập test gồm 100 ảnh với 10 ảnh/mỗi thương hiệu để phân loại. Số lượng nhãn sẽ giảm dần trong lúc đánh giá chi tiết ở bên dưới để đánh giá đầu ra của bài toán một cách tổng quát.

Các bước xử lý bao gồm:

## 1. Xử lý Input:

- Input gồm:

+ folder Carlogo chứa tập ảnh train và test

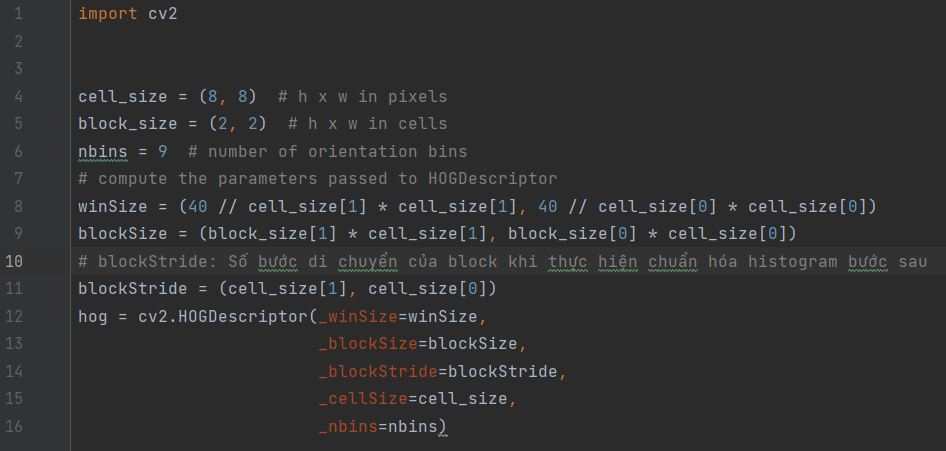
+ package Files chứa 2 tập đặc trưng (X\_train, y\_train) và (X\_test, y\_test) để huấn luyện và kiểm tra

+ package process để xử lý dữ liệu

### 1.1. Tạo đặc trưng HOG trong hàm create\_HOG\_description:

- Tính toán các tham số truyền vào: cell\_size, block\_size, nbins, winSize, blockSize, blockStride (đọc thêm về HOG để biết thêm ý nghĩa các tham số). Đặc biệt tham số 40 trong winSize chính là kích thước chiều dài và chiều rộng của logo sau khi xử lý.

- Tính toán đặc trưng HOG, sử dụng hàm cv2.HOGDescriptor()



### 1.2. Xử lý tập dữ liệu đầu vào:

Xây dựng hàm \_preprocessing như sau:

- Đọc từng ảnh trong train folder để xử lý, ở đây mỗi ảnh có kích thước 70x70

- Xử dụng phương pháp Canny để phát hiện biên cho ảnh đầu vào

- Trích xuất logo của xe:

+ Đầu tiên, sử dụng hàm findContour() để tìm contours trong ảnh edged

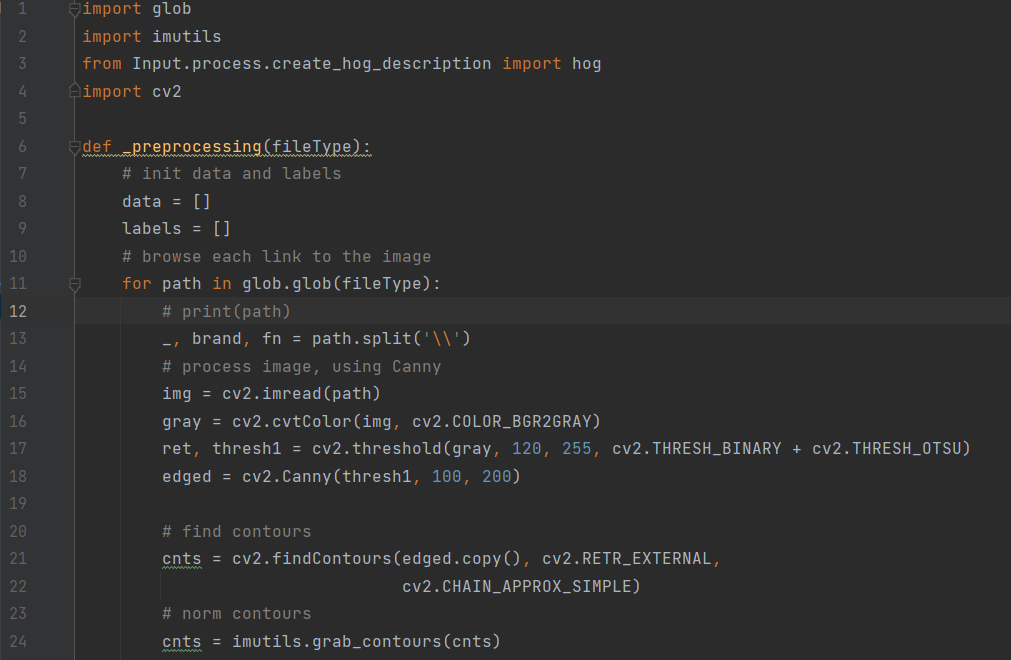
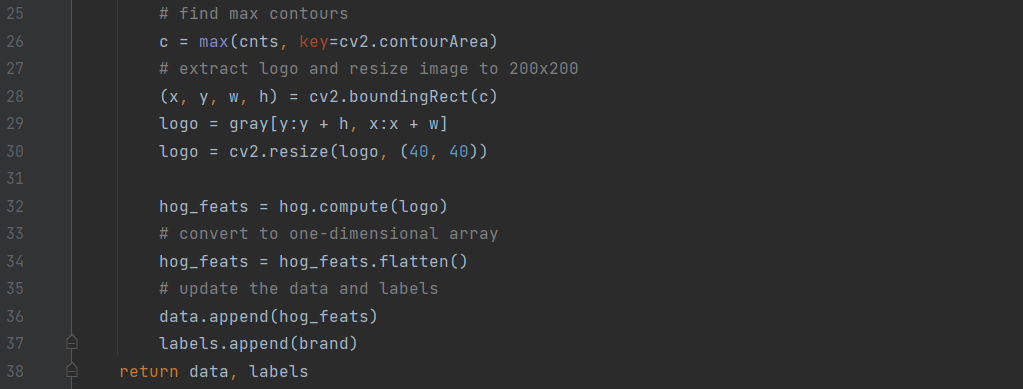
+ Chuẩn hóa tập tập contours tìm được phù hợp với từng phiên bản openCV

+ Lấy ra đường bao có diện tích lớn nhất bao hết được logo

+ Xác định đường bao lớn nhất đó rồi cắt ra, resize để các ảnh về một kích thước ta thu được vùng chưa logo tối thiểu để phục vụ cho thao tác sau. Ở đây ta chọn kích thước của logo là 40x40

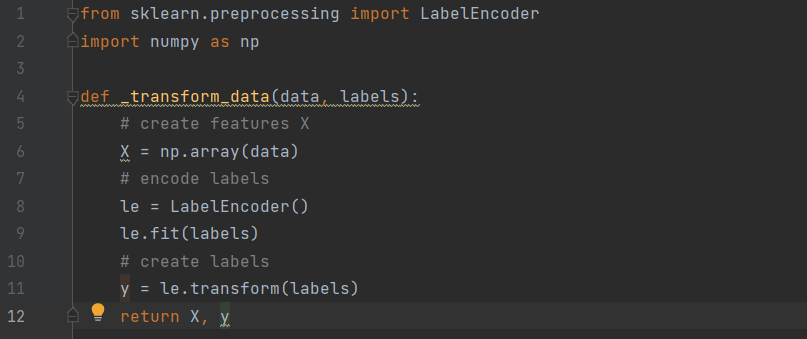
- Tạo đặc trưng HOG

- Update tập đầu ra data và labels

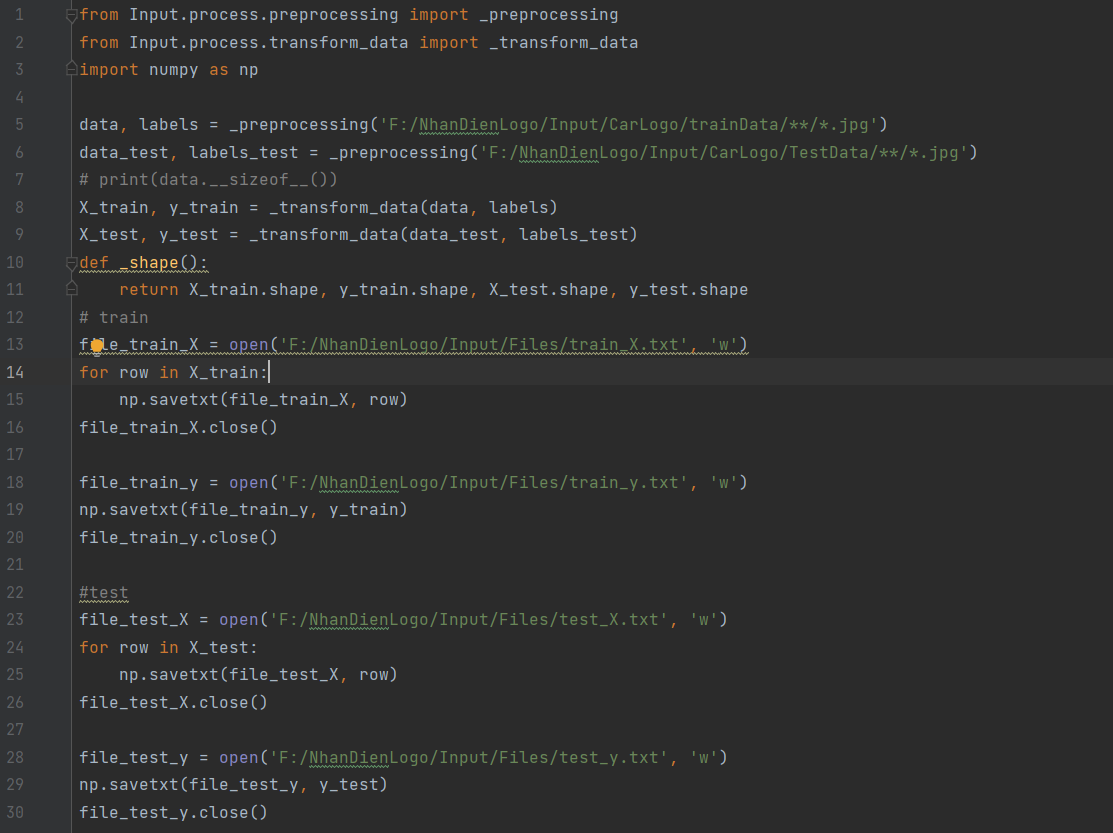
  


### 1.3. Hàm chuyển đổi data và labels về dữ liệu để huấn luyện và kiểm tra:

- Xây dựng hàm \_transform\_data() để chuyển đổi tập đặc trưng và nhãn tương ứng thành tập dữ liệu đào tạo.



### 1.4. Lưu tập huấn luyện và đào tạo vào package Files:



## 2. Lấy dữ liệu từ package:

- Lấy 4 tập X\_train, y\_train, X\_test, y\_test từ các files đã lưu rồi định dạng lại về numpy array của chúng như ban đầu.

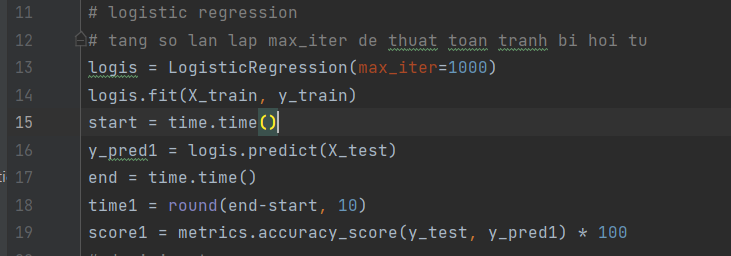
- Hàm \_\_data() trả về 4 tập lần lượt theo thứ tự trên



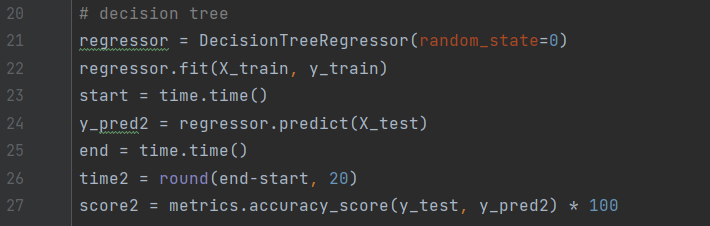
## 3. Xây dựng các mô hình học có giám sát:

- Từ tập dữ liệu đào tạo và huấn luyện ở trên ta lần lượt đào tạo các mô hình và đánh giá thời gian dự đoán, độ chính xác của thuật toán:

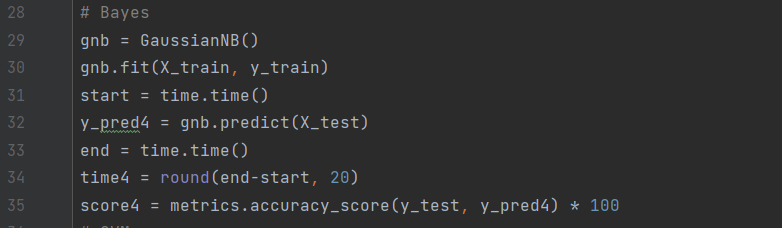
### 3.1. Mô hình Logistic Regression (Hồi quy Logistic)



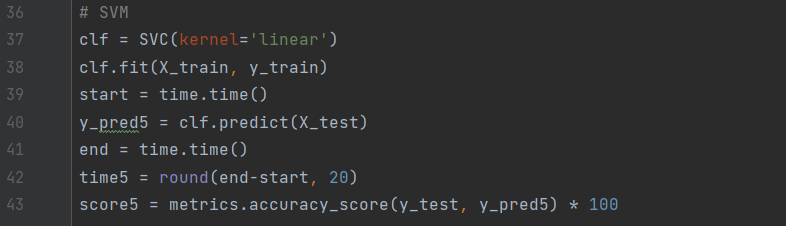
### 3.2. Mô hình Decision Tree (Cây quyết định)



### **3.3. Mô hình Naïve Bayes classifiers (Bộ phân loại bayes)**

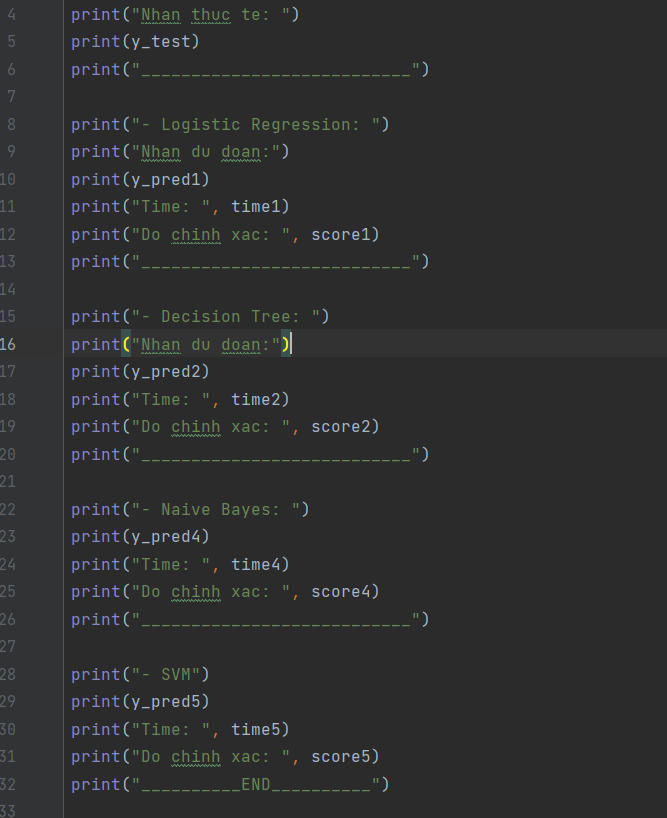


### 3.4. Mô hình Support Vector Machines (SVM)



## 4. Khởi chạy kết quả:

- Chạy các mô hình ở trên, lần lượt thay đổi số lượng ảnh. Thống kê và ghi lại kết quả.



# III. Đánh giá mô hình:

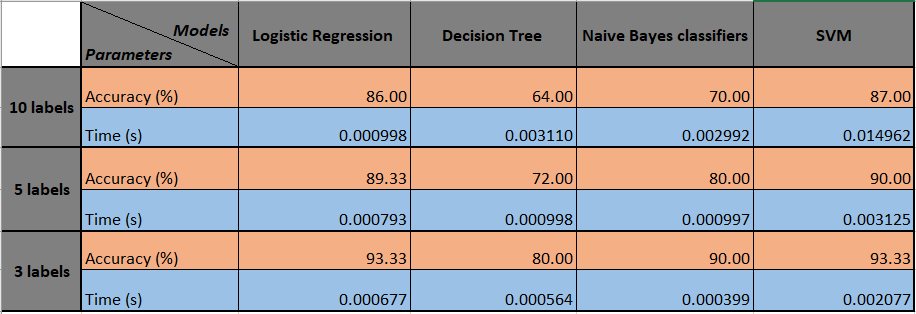
## 1. Kết quả

- Tập dữ liệu sử dụng tuy không lớn nhưng cũng đủ để đánh giá tương đối các mô hình. Ở đây ta bỏ qua ảnh hưởng của dung lượng bộ nhớ bởi các hệ thống hiện tại được xây dựng với bộ nhớ rất lớn.

- Chúng ta sẽ sử dụng 2 yếu tố để đánh giá: Độ chính xác và thời gian

- Ở trên ta có nói đầu vào sử dụng là 10 nhãn để dự đoán. Ta sẽ giảm dần con số này để xem thay đổi của các yếu tố trên đối với từng model. 3 mức được sử dụng: 10 nhãn, 5 nhãn và 3 nhãn. Data ở tập training và test cũng sẽ được loại bỏ theo các nhãn

- Với mỗi bộ test ta sẽ chạy code 3 lần và lấy trung bình độ chính xác và thời gian của từng mô hình. Kết quả thể hiện ở bảng đánh giá sau:



## 2. Đánh giá

\* Tổng quát:

- Các mô hình học có giám sát cho kết quả tốt khi số lượng nhãn nhỏ

- Thời gian thực thi tương đối thấp (cao hơn cả là SVM)

- Đối với các bộ dữ liệu kích thước nhỏ, HOG có thể được sử dụng để tạo đặc trưng đầu vào cho các thuật toán học có giám sát cổ điển ở trên. Khi thay đổi các tham số trong hàm cv2.HOGDescription() ở mục 1.1 sẽ làm ảnh hưởng tới tập các feature của dữ liệu. Ở đây khuyến nghị thay đổi 3 tham số: cellSize, blockSize, logo.shape (40x40). Giảm cellSize, blockSize hoặc tăng các kích thước trong logo.shape sẽ làm cho số lượng các feature sẽ tăng lên. Khi đó độ chính xác tăng nhưng sẽ làm tăng bộ nhớ và thời gian xử lý dữ liệu. Do vậy cần cân bằng 3 tham số trên để có kết quả phù hợp cho từng mô hình.

\* Chi tiết:

- Logistic regression: đây là mô hình rất hiệu quả với độ chính xác cao và thời gian tốt. Đây là kết quả hiển nhiên bởi nó được xây dựng riêng cho các bài toán phân loại.

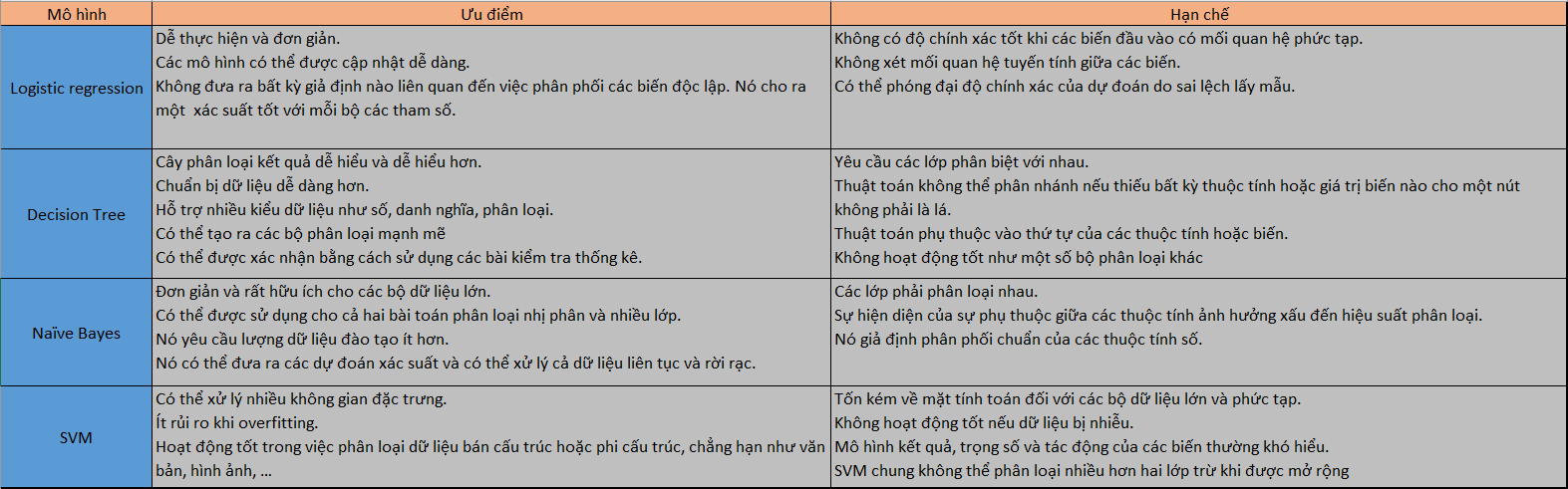
- Decision tree: cho kết quả dự đoán không tốt trong trường hợp này do các feature là đặc trưng của mỗi ảnh theo từng pixel. Do đó với số lượng nút khi xây dựng lớn và phức tạp như vậy sẽ khiến cho mô hình hoạt động không tốt.

=> Có thể cải tiến bởi mô hình Random Forest

- Naïve Bayes classifiers: cho độ chính xác cao khi số nhãn thấp. Tuy nhiên điểm nổi bật là thời gian xử lý cực nhanh ngay khi tập dữ liệu lớn. Nguyên nhân do khi xây dựng chỉ phải tính các giá trị xác suất là lưu trữ nên khi dự đoán chỉ cần thực hiện một phép toán số học là ra được kết quả.

- SVM: mô hình cho kết quả tốt với hầu hết các mẫu.

## 3. BẢNG ĐÁNH GIÁ TỔNG QUÁT CÁC MÔ HÌNH PHÂN LỚP

  
  
  
- Link code: <https://github.com/haild2k/BC2.git>