**Lesson 5: Classification with Decision Tree & Random Forest**

1. Decision Tree là gì ?

* Cây quyết định, hay decision tree là 1 thuật toán phân lớp dựa vào 1 dãy các thuộc tính được cho sẵn. Từ đó, mô hình sẽ rút ra được quy luật từ các thuộc tính đó và sẽ sử dụng quy luật đó để dự đoán class chưa biết dựa vào các thuộc tính cho trước.

**Ví dụ:**

A picture containing graphical user interface

Description automatically generated

Dựa vào 3 yếu tố outlook, wind, và humidity, có thể rút ra các quy luật:

* nếu trời nắng, độ ẩm bình thường => đi chơi
* nếu trời mưa, gió yếu => đi chơi
* nếu trời âm u, không nắng không mưa => đi chơi

→ Như vậy, yếu tố quan trọng nhất là bầu trời như thế nào, tiếp đó mới là các yếu tố như độ ẩm, gió, ...

A picture containing text

Description automatically generated

* Nhìn một cách đơn giản, nó chính là một tập các IF-ELSE, là kết quả của quá trình đào tạo dữ liệu bằng cây quyết định
* Có thể thấy, nhiều việc trong cuộc sống cũng được chúng ta thực thi dựa trên mô hình cây quyết định này.

1. **Nguyên lý hoạt động**

* Decision tree hoạt động bằng cách phân tích các features thành các nodes

A picture containing diagram

Description automatically generated

* Mỗi node sẽ có tác động tương ứng với các đầu ra khác nhau, vì vậy các đầu ra đó được ký hiệu thành 2 màu xanh và đỏ. Một node sẽ có chứa 1 điều kiện hoặc 1 giá trị. Mỗi node có thể có nhiều node con hoặc không có node con nào.
* Root Node : là node ở thứ tự cao nhất trong decision tree, nơi các node khác bắt đầu xuất hiện
* Branch node : là node có các child node theo sau nó
* Leaf Node : là node không có child node nào.
* Mô hình sẽ tính toán và phân loại đầu ra dựa vào dữ liệu đầu vào rơi vào các node nào.

1. **Entropy**

* Hàm số Entropy là một hàm số thể hiện phân phối xác suất của một biến rời rạc với nhiều giá trị khác nhau (vì 1 class có thể có nhiều giá trị dẫn tới class đó)
* Công thức :

A picture containing text, watch

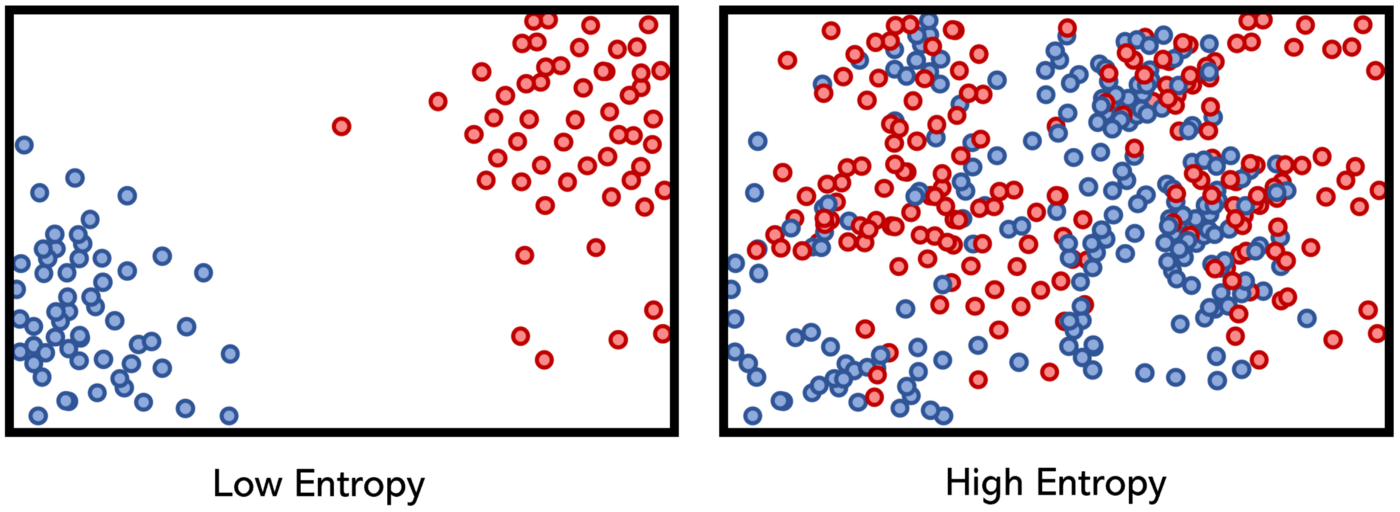
Description automatically generated

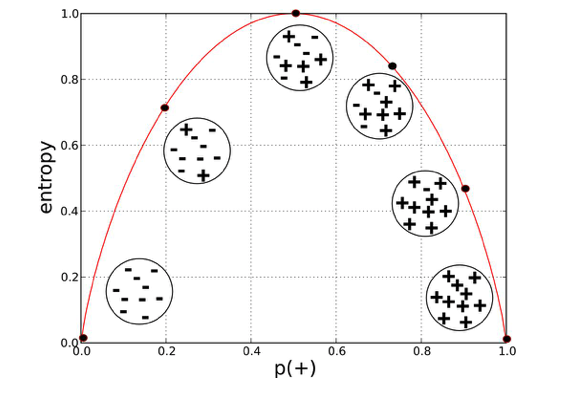
khi và , log là logarit tự nhiên

* Lưu ý: Một số tài liệu dùng logarit cơ số 2 và quy ước

Text

Description automatically generated





* Lưu ý: Vì mỗi features sẽ là một node, mỗi node lại có thể có nhiều child node hoặc mang giá trị khác nhau nên mô hình decision tree rất dễ bị overfitting nếu model có quá nhiều features.
* Vì vậy, cần phải xác định các features chính để xây decision tree. Có thể xác định các features chính bằng cách tính chỉ số GINI ( information gain).

1. **GINI Index**

* Gini index tương tự như information gain, dùng để đánh giá xem việc phân chia ở node điều kiện có tốt hay không
* Chỉ số gini thấp nhất (=0) khi node đó chỉ chứa dữ liệu của 1 lớp duy nhất, chỉ số gini cao nhất khi dữ liệu các lớp ở trong node đó cân bằng (đã chứng minh ở trên) => Ta mong muốn khi phân tách thì chỉ số gini ở những lớp con thấp
* Ngoài ra, còn có thể hạn chế overfitting bằng cách giới hạn độ sâu của cây (dừng việc insert thêm các node có nhiều child node) , hoặc giới hạn min value các leaf node (nếu leaf node có tổng value <= n value thì sẽ không phân tách tiếp nữa)
* Pruning (cắt tỉa các branch node, chỉ lấy các branch node dẫn đến mô hình có accuracy cao, hoặc giảm thiểu loss function) (kỹ thuật cao).

1. **Practice** 1 ([Link)](https://colab.research.google.com/drive/1JB8H5CSuLUvLqzOmtKh6EF65RkhpBDGL)
2. **Các thuật toán dựa trên Decision Tree**
3. **Iterative Dichotomiser 3**

* ID3 là một thuật toán decision tree được áp dụng cho các bài toán binary classification mà tất cả các thuộc tính đều ở dạng categorical

1. **Classification and Regression Tree (CART)**

* CART là một thuật toán decision tree được áp dụng cho các bài toán classification mà tất cả các thuộc tính đều có thể ở dạng biến categorical hoặc biến liên tục

Diagram

Description automatically generated

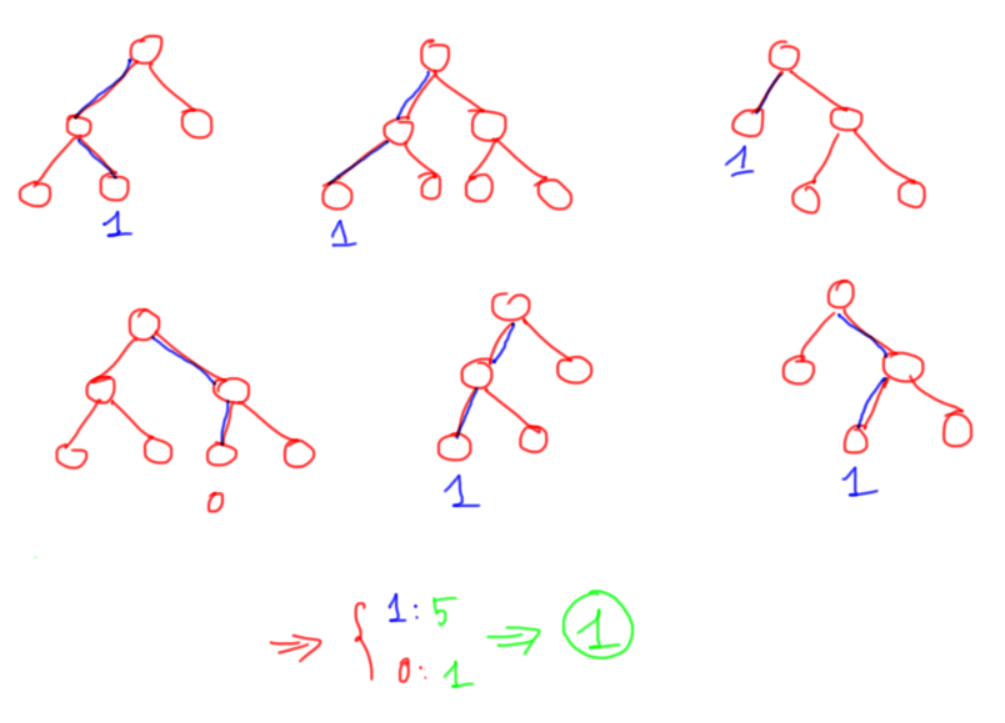
1. **C4.5**

* C4.5 là một phiên bản decision tree không bị giới hạn ở khả năng phân nhánh (decision tree bình thường sẽ phân 2 nhánh 1 lần)

1. **Random forest**

* Random là ngẫu nhiên, Forest là rừng. Random Forest là một thuật toán xây dựng nhiều cây quyết định bằng thuật toán Decision Tree, tuy nhiên, mỗi cây quyết định này là khác nhau (lấy random). Sau đó kết quả dự đoán sẽ được tổng hợp từ các cây quyết định này.

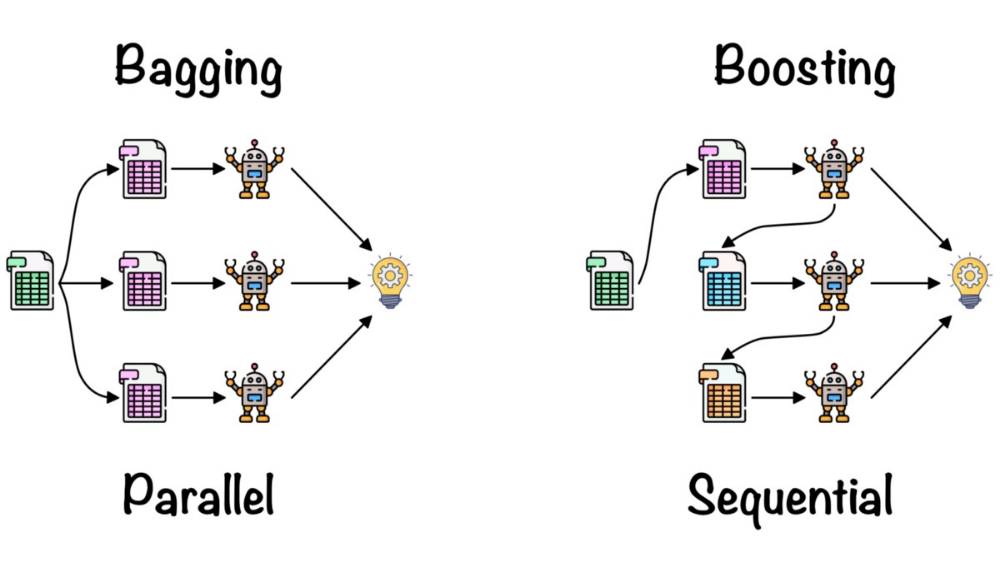
→ Các cây quyết định có thể vận hành khác nhau để trả ra class giống nhau

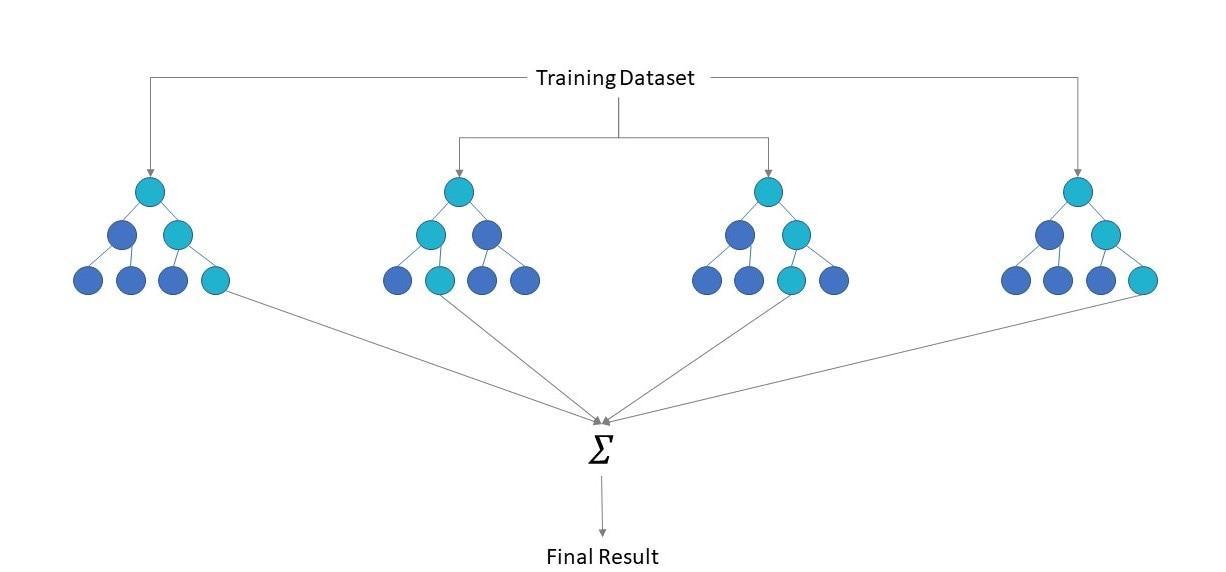


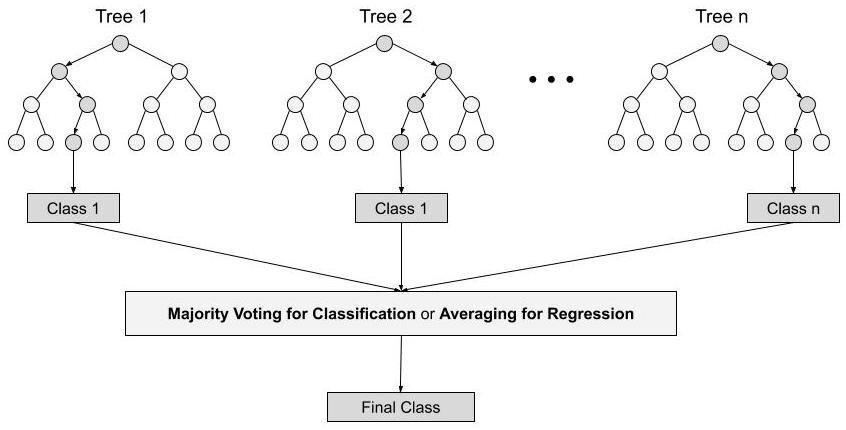
* Ví dụ cùng 1 vấn đề, nhưng 6 decision tree cho ra 5 dự đoán là class 1 trong khi chỉ có 1 cây cho dự đoán là class 0 => Vote dự đoán là class 1.
* **Ví dụ**: Mọi người muốn mua 1 sản phẩm trên tiki chẳng hạn, khi đọc review sản phẩm, nếu chỉ đọc 1 review thì có thể là ý kiến chủ quan của người đấy, hoặc sản phẩm người ấy mua không may bị lỗi gì; thông thường để có cái nhìn tốt về sản phẩm, mình hay đọc tất cả review rồi cho ra quyết định cuối cùng

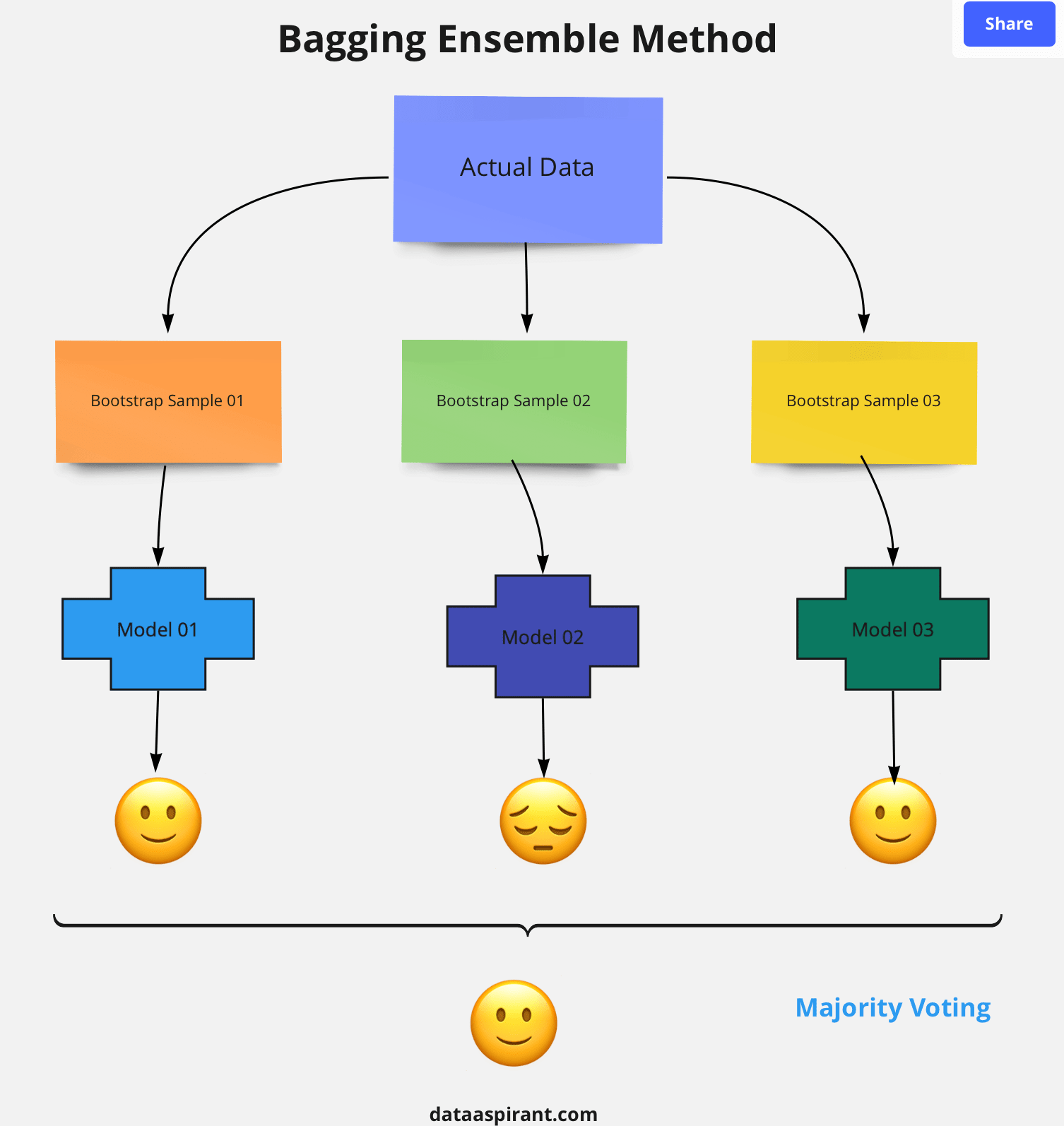
1. Nguyên lý hoạt động của Random forest classifier

* Random forest là một thuật toán bagging thay vì boosting









* Ưu điểm:
* Hạn chế được overfitting, thay vì 1 cây có quá nhiều features thì chúng ta tạo ra nhiều cây để chứa nhiều features.
* Cho kết quả tốt hơn
* Nhược điểm:
* Cần lượng data rất lớn để train
* Chạy lâu vì có nhiều decision tree
* Phức tạp hơn vì nhiều decision tree trong 1 model.
* Đôi khi số lượng cây sai lại ảnh hưởng đến kết quả từ cây đúng.

**Ví dụ:** có 6 cây, 4 cây cho kết quả 1, 2 cây cho kết quả 0 => Lấy kết quả 1, tuy nhiên thực tế kết quả đúng lại là kết quả 0)

* Đôi khi các cây trong forest có sự correlation với nhau sẽ dẫn đến kết quả không còn khách quan (vì mỗi cây nên là independent)

1. **Practice 2** ([Link](https://colab.research.google.com/drive/1JB8H5CSuLUvLqzOmtKh6EF65RkhpBDGL))