HUTECH UNIVERSITY OF TECHNOLOGY

****

Artificial Intelligence
Final Test Report

TOPIC NAME: Gradient Boosting Decision Tree Algorithm

Group:

1. **2011064187_Do Hoang Cong Danh**
2. **2011060554_Nguyen Thi Kim Linh**

Ho Chi Minh City, 12/2022

Table of Contents

PART 1:

1. **Definition:**

Gradient boosting is a machine learning technique used in regression and classification tasks, among others. It gives a prediction model in the form of an ensemble of weak prediction models, which are typically decision trees. When a decision tree is the weak learner, the resulting algorithm is called gradient-boosted trees; it usually outperforms random forest. A gradient-boosted trees model is built in a stage-wise fashion as in other boosting methods, but it generalizes the other methods by allowing optimization of an arbitrary differentiable loss function.

When the target column is continuous, we use Gradient Boosting Regressor whereas when it is a classification problem, we use Gradient Boosting Classifier.

The idea of gradient boosting originated in the observation by Leo Breiman that boosting can be interpreted as an optimization algorithm on a suitable cost function. Explicit regression gradient boosting algorithms were subsequently developed, by Jerome H. Friedman.

2. **Algorithm:**

A. Gradient Boosting Classification Details

Input: Data $\{(x_i, y_i)\}_{i=1}^{n}$, and a differentiable Loss Function $L\big(y, F(x)\big)$

The loss function for the classification problem is given below:

$$L =- \sum_{i=1}^{n} y_i \log(p) + (1-p)\log(1-p)$$

Note: Differentiable Loss Function = $-Observed + p$

p: predicted probability

Step 1: Initialize model with a constant value:

$$F_0(x) = \underset{\gamma}{\operatorname{argmin}} \sum_{i=1}^{n} L(y_1, \gamma)$$

Step 2:for m = 1 to M:

(A) Compute $r_{im} = -\left[\frac{\partial L(y_i, F(x_i))}{\partial F(x_i)}\right]_{F(x)=F_{m-1}(x)}$ for i = 1,…,n

(B) Fit a regression tree to the r_{im} values and create terminal region r_{jm}, for j = 1,…, J_m

(C) For j = 1,..., J_m compute

$$\gamma_{im} = \underset{\gamma}{\operatorname{argmin}} \sum_{x_i \in R_{ij}} L(y_1, F_{m-1}(x_i) + \gamma)$$

(D) Update

$$F_m(x) = F_{m-1}(x) + \nu \sum_{j=1}^{J_m} \gamma_{jm} I(x \in R_{jm})$$

Step 3: Output $F_M(x)$

3. Example:

Example 1:

Input: Training Data

Likes Popcorn	Age	Favorite Color	Loves Troll 2
Yes	12	Blue	Yes
No	87	Green	Yes
No	44	Blue	No

Step 1: Initialize model with a constant value:

$$F_0(x) = \underset{\gamma}{\operatorname{argmin}} \sum_{i=1}^{n} L(y_1, \gamma)$$

Initial Prediction — We start with a leaf which represents an initial prediction for every individual. For classification, this will be equal to log(odds) of the dependent variable. Since there are 3 people with and 1 not Love Troll and 2 Love Troll, log(odds) is equal to –

$$Log(odds) = \log\left(\frac{2}{1}\right) = 0.69$$

$$F_0(x) = \log\left(\frac{2}{1}\right)$$

$$p = \frac{e^{\log\left(\frac{2}{1}\right)}}{1 + e^{\log\left(\frac{2}{1}\right)}} = 0.67$$

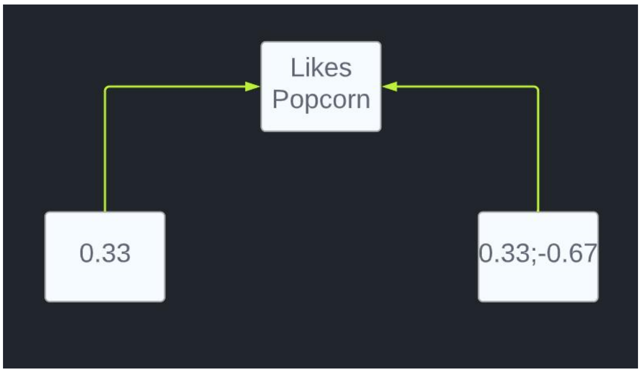
Step 2: for m = 1 to M:

Step 2 (A): Compute $r_{im} = -\left[\frac{\partial L(y_i, F(x_i))}{\partial F(x_i)}\right]_{F(x)=F_{m-1}(x)}$ for i = 1,…,n

$$r_{1,1} = (1 - 0.67) = 0.33$$
$$r_{2,1} = (1 - 0.67) = 0.33$$
$$r_{3,1} = (0 - 0.67) =- 0.67$$

Step 2 (B): Fit a regression tree to the r_{im} values and create terminal region r_{jm}, for j = 1,…, J_m

Likes Popcorn	Age	Favorite Color	Loves Troll 2	$r_{i,1}$
Yes	12	Blue	Yes	0.33
No	87	Green	Yes	0.33
No	44	Blue	No	-0.67



Here is the new tree

Step 2 (C): For j = 1,..., J_m compute

$$\gamma_{im} = \underset{\gamma}{\operatorname{argmin}} \sum_{x_i \in R_{ij}} L(y_1, F_{m-1}(x_i) + \gamma)$$

$$\gamma_{1,1} = \frac{Residual}{p * (1-p)} = \frac{0.33}{0.67 * (1 - 0.67)} = 1.5$$

$$\gamma_{2,1} = \frac{Residual_2 + Residual_3}{[p_2 * (1-p_2)] + [p_3 * (1-p_3)]} = \frac{0.33 + (-0.67)}{[0.67 * (1-0.67)] + [0.67 * (1-0.67)]} = -0.77$$

Step 2 (D): Update

$$F_m(x) = F_{m-1}(x) + \nu \sum_{j=1}^{J_m} \gamma_{jm} I(x \in R_{jm})$$

The new predicion for first sample x_1:

$F_1(x) = 0.69 + 0.8 * 1.5 = 1.89$

The new predicion for second sample x_2:

$F_1(x) = 0.69 + 0.8 * -0.77 = 0.07$

The new predicion for third sample x_3:

$F_1(x) = 0.69 + 0.8 * -0.77 = 0.07$

Now we set m = 2 and do everything over again:

Step 2:

Step 2 (A): Compute $r_{im} = -\left[\frac{\partial L(y_i, F(x_i))}{\partial F(x_i)}\right]_{F(x)=F_{m-1}(x)}$ for i = 1,…,n

$$p(x_1) = \frac{e^{1.89}}{1 + e^{1.89}} = 0.87$$

$$r_{1,1} = (1 - 0.87) = 0.13$$

$$p(x_2) = \frac{e^{0,07}}{1 + e^{0.07}} = 0.52$$

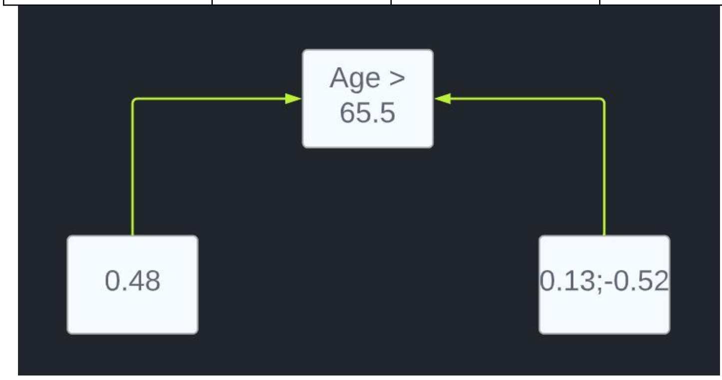
$$r_{2,1} = (1 - 0.52) = 0.48$$

$$p(x_3) = \frac{e^{0,07}}{1 + e^{0.07}} = 0.52$$

$$r_{3,1} = (0 - 0.52) =- 0.52$$

Step 2 (B): Fit a regression tree to the r_{im} values and create terminal region r_{jm}, for j = 1,…, J$_m$

Likes Popcorn	Age	Favorite Color	Loves Troll 2	$r_{i,1}$	$r_{i,2}$
Yes	12	Blue	Yes	0.33	0.13
No	87	Green	Yes	0.33	0.48
No	44	Blue	No	-0.67	-0.52



Step 2 (C): For j = 1,..., J$_m$ compute

$$\gamma_{im} = \underset{\gamma}{\operatorname{argmin}} \sum_{x_i \in R_{ij}} L(y_1, F_{m-1}(x_i) + \gamma)$$

$$\gamma_{1,2} = \frac{Residual}{p*(1-p)} = \frac{0.48}{0.52*(1-0.52)} = 1.9$$

$$\gamma_{2,2} = \frac{Residual_2 + Residual_3}{[p_2 * (1 - p_2)] + [p_3 * (1 - p_3)]}$$

$$= \frac{0.13 + (-0.52)}{[0.87 * (1 - 0.87)] + [0.52 * (1 - 0.52)]} = -1.1$$

Step 2 (D):

The new prediction for first sample x_1

$F_2(x) = 0.69 + 0.8 * 1.5 + 0.8 * 1.9 = 3.41$

The new prediction for second sample x_2

$F_2(x) = 0.69 + 0.8 *- 0.77 + 0.8 * (-1.1) = 0.806$

The new prediction for third sample x_3

$F_2(x) = 0.69 + 0.8 *- 0.77 + 0.8 * (-1.1) = 0.806$

Step 3:

If m = 2 then $F_2(x)$ is the output from the Gradient Boost Algorithm

Now if we have the new data

Likes Popcorn	Age	Favorite Color	Loves Troll 2
Yes	90	Green	???

The predicted log(odds) that this person Love Troll 2:

$$\log\left(\frac{2}{1}\right) + (0.8 * 1.5) + (0.8 * 1.9) = 3.4$$

The predicted probability that this person Love Troll 2:

$$\frac{e^{3.4}}{1 + e^{3.4}} = 0.97$$

If we use a threshold of 0.5 for deciding if someone Loves Troll 2, then since 0.97 > 0.5, this person Love Troll 2.

Example 2:

Input: Training Data

Chest Pain	Good Blood Circulation	Blocked Arteris	Heart Disease
No	No	No	No
Yes	Yes	Yes	Yes
Yes	Yes	No	Yes

Step 1: Initialize model with a constant value:

$$F_0(x) = \underset{\gamma}{\operatorname{argmin}} \sum_{i=1}^{n} L(y_1, \gamma)$$

Initial Prediction — We start with a leaf which represents an initial prediction for every individual. For classification, this will be equal to log(odds) of the dependent variable. Since there are 2 people with and 1people without Heart Disease, log(odds) is equal to –

$$Log(odds) = \log\left(\frac{2}{1}\right) = 0.69$$

$$F_0(x) = \log\left(\frac{2}{1}\right)$$

$$p = \frac{e^{\log\left(\frac{2}{1}\right)}}{1 + e^{\log\left(\frac{2}{1}\right)}} = 0.67$$

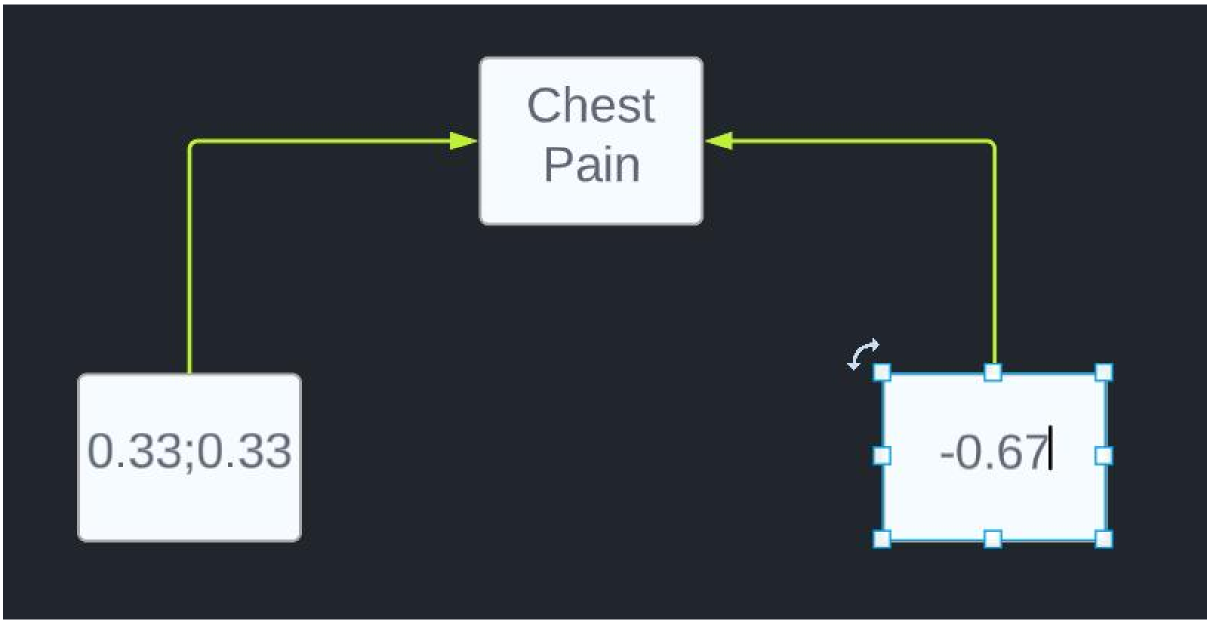
Step 2: for m = 1 to M:

Step 2 (A): Compute $r_{im} = -\left[\frac{\partial L(y_i, F(x_i))}{\partial F(x_i)}\right]_{F(x)=F_{m-1}(x)}$ for i = 1,…,n

$$r_{1,1} = (0 - 0.67) =- 0.67$$
$$r_{2,1} = (1 - 0.67) = 0.33$$
$$r_{3,1} = (1 - 0.67) = 0.33$$

Step 2 (B): Fit a regression tree to the r_{im} values and create terminal region r_{jm}, for j = 1,…, J_m

Chest Pain	Good Blood Circulation	Blocked Arteris	Heart Disease	$r_{i,1}$
No	No	No	No	-0.67
Yes	Yes	Yes	Yes	0.33
Yes	Yes	No	Yes	0.33



Step 2 (C): For j = 1,..., J_m compute

$$\gamma_{im} = \underset{\gamma}{\operatorname{argmin}} \sum_{x_i \in R_{ij}} L(y_1, F_{m-1}(x_i) + \gamma)$$

$$\gamma_{1,1} = \frac{Residual_2 + Residual_3}{[p_2 * (1 - p_2)] + [p_3 * (1 - p_3)]} = \frac{0.33 + 0.33}{[0.67 * (1 - 0.67)] + [0.67 * (1 - 0.67)]} = 1.49$$

$$\gamma_{2,1} = \frac{Residual_1}{[p_1 * (1 - p_1)]} = \frac{-0.67}{[0.67 * (1 - 0.67)]} = -3.03$$

Step 2 (D): Update

$$F_m(x) = F_{m-1}(x) + \nu \sum_{j=1}^{J_m} \gamma_{jm} I(x \in R_{jm})$$

The new predicion for first sample x_1:

$F_1(x) = 0.69 + 0.8 * (-3.03) =- 1.73$

The new predicion for second sample x_2:

$F_1(x) = 0.69 + 0.8 * 1.49 = 1.88$

The new predicion for third sample x_3:

$$F_1(x) = 0.69 + 0.8 * 1.49 = 1.88$$

Now we set m = 2 and do everything over again:

Step 2:

Step 2 (A): Compute $r_{im} = -\left[\frac{\partial L(y_i, F(x_i))}{\partial F(x_i)}\right]_{F(x)=F_{m-1}(x)}$ for i = 1,…,n

$$p(x_1) = \frac{e^{-1.73}}{1 + e^{-1.73}} = 0.15$$

$$r_{1,1} = (0 - 0.15) =- 0.15$$

$$p(x_2) = \frac{e^{1.88}}{1 + e^{1.88}} = 0.87$$

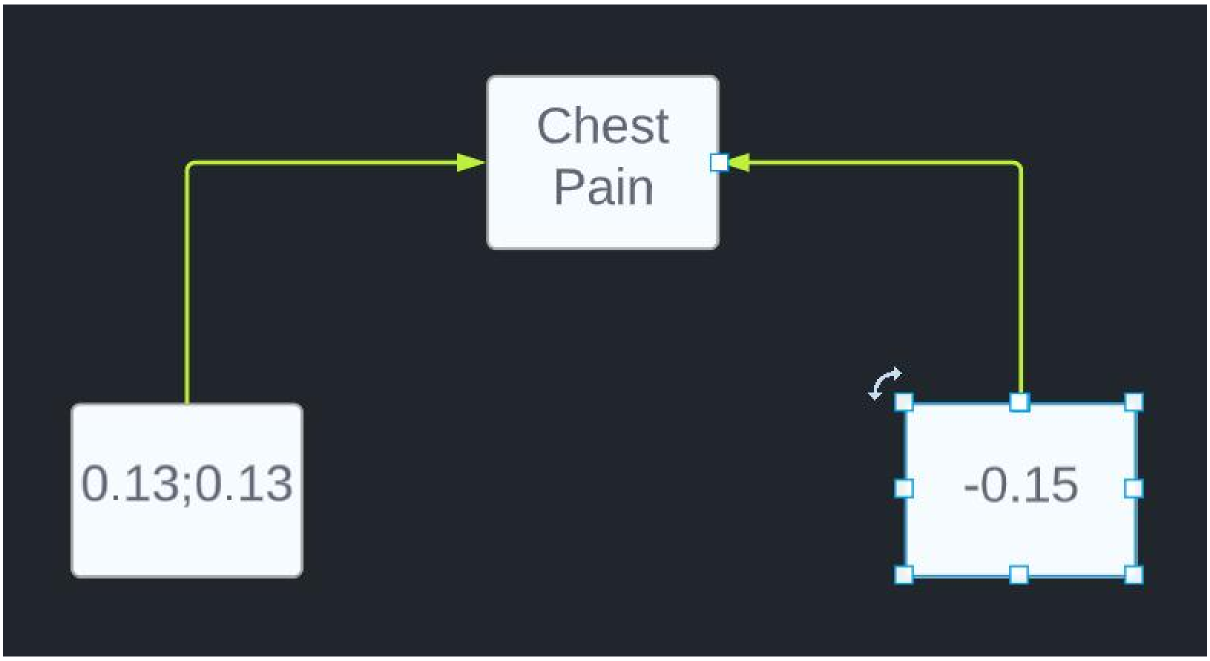
$$r_{2,1} = (1 - 0.87) = 0.13$$

$$p(x_3) = \frac{e^{1.88}}{1 + e^{1.88}} = 0.87$$

$$r_{3,1} = (1 - 0.87) = 0.13$$

Step 2 (B): Fit a regression tree to the r_{im} values and create terminal region r_{jm}, for j = 1,…, J_m

Chest Pain	Good Blood Circulation	Blocked Arteris	Heart Disease	$r_{i,1}$	$r_{i,2}$
No	No	No	No	-0.67	-0.15
Yes	Yes	Yes	Yes	0.33	0.13
Yes	Yes	No	Yes	0.33	0.13



Step 2 (C): For j = 1,..., J_m compute

$$\gamma_{im} = \underset{\gamma}{\operatorname{argmin}} \sum_{x_i \in R_{ij}} L(y_1, F_{m-1}(x_i) + \gamma)$$

$$\gamma_{1,1} = \frac{Residual_2 + Residual_3}{[p_2 * (1 - p_2)] + [p_3 * (1 - p_3)]}$$

$$= \frac{0.13 + 0.13}{[0.87 * (1 - 0.87)] + [0.87 * (1 - 0.87)]} = 1.15$$

$$\gamma_{2,1} = \frac{Residual_1}{[p_1 * (1 - p_1)]} = \frac{-0.15}{[0.15 * (1 - 0.15)]} = -1.18$$

Step 2 (D):

The new prediction for first sample x_1

$F_2(x) = 0.69 + 0.8 * (-3.03) + 0.8 * (-1.18) =- 2.68$

The new prediction for second sample x_2

$F_2(x) = 0.69 + 0.8 * 1.49 + 0.8 * 1.15 = 2.8$

The new prediction for third sample x_3

$F_2(x) = 0.69 + 0.8 * 1.49 + 0.8 * 1.15 = 2.8$

Step 3:

If m = 2 then $F_2(x)$ is the output from the Gradient Boost Algorithm

Now if we have the new data

Chest Pain	Good Blood Circulation	Blocked Arteris	Heart Disease
Yes	Yes	Yes	???

The predicted log(odds) that this person Love Troll 2:

$$\log\left(\frac{2}{1}\right) + (0.8 * 1.49) + (0.8 * 1.15) = 2.8$$

The predicted probability that this person Love Troll 2:

$$\frac{e^{2.8}}{1 + e^{2.8}} = 0.94$$

If we use a threshold of 0.5 for deciding if someone Heart Disease, then since 0.94 > 0.5, this person Heart Disease.

4. Application:

Predict learning enjoyment based on age, gender and focus.

Predict tennis play base on weather, gender, age

PART 2:

Project: Graph Coloring: Optimize

File1: main.cpp

```
#include <stdio.h>
#define max 20

struct graph{
	int arr[max][max];
	int num;
};

void ReadMatrixFromFile(graph &g,char filename[50]);
void Displaymatrix(graph g);
int Out(graph g,int* level);
int SelectVertex(graph g,int level[max]);
int SelectColor(graph g,int S);
int* CalLevel(graph g);
int main(void) {

	graph mygraph;

	//Graph Coloring

	char filename[50] = "F:\\AI\\GraphColor\\G03.txt";


	printf("\nMatrix 1: \n");
	//Display MATRIX 1
	ReadMatrixFromFile(mygraph,filename);
	Displaymatrix(mygraph);
	printf("\n\nOptimized Algorithm");
	int* level = CalLevel(mygraph);
	Out(mygraph,level);
}
```

File2: Read and Display Matrix from file txt

```
	#include <stdio.h>
#define max 20
struct graph{
	int arr[max][max];
	int num;
};
//Read and display matrix
void ReadMatrixFromFile(graph &g,char filename[50]){
```

```
FILE *f;
f = fopen(filename,"r");
if(f == NULL){
	printf("ERROR: OPEN FILE FAIL!");
}else{
	fscanf(f,"%d",&g.num);
	for(int i=0;i<g.num;i++){
		for(int j=0;j<g.num;j++){
			fscanf(f,"%d ",&g.arr[i][j]);
		}
	}
	fclose(f);
	}
}
void Displaymatrix(graph g){
	for(int i=0;i<g.num;i++){
		for(int j=0;j<g.num;j++){
			printf("%d ",g.arr[i][j]);
		}
		printf("\n");
	}
}
```

File3: Optimize Algorithm

```
	#include <stdio.h>
#define max 20

struct graph{
	int arr[max][max];
	int num;
};

int ColorNumber = 1;
int ColorTable[max];

int* CalLevel(graph g){
	static int Cal[max];
	for(int i = 0;i < g.num;i++){
		int count = 0;
		for(int j = 0;j < g.num;j++){
			if(g.arr[i][j] > 0){
				count ++;
			}
		}
		*(Cal + i) = count;
```

```
        }
        return Cal;
}



int SelectVertex(graph g,int* level){
        int MaxLevel = *(level + 0);
        int Vertex = 0;
        for(int i = 0;i < g.num;i++){
                if(*(level + i) > MaxLevel){
                        MaxLevel = *(level + i);
                        Vertex = i;
                }
        }
        return Vertex;
}

void handle(graph g,int S,int* level){
        for(int i = 0;i < g.num;i++){
                if(g.arr[S][i] == 1){
                        *(level + i) = *(level + i) - 1;
                }
        }
}

int SelectColor(graph g,int S){
        int Check;
        int Color;
        for(Color = 1;Color <= ColorNumber;Color++){
                Check = 1;
                for(int i = 0;i < g.num;i++){
                        if(g.arr[S][i] == 1 && ColorTable[i] == Color){
                                Check = 0;
                        }
                }
                if(Check){
                        return Color;
                }
        }
        ColorNumber++;
        return  Color;
}
void Out(graph g,int* level){
```

```
        for(int i = 0;i < g.num;i++){
                int S = SelectVertex(g,level);
                int Color = SelectColor(g,S);
                ColorTable[S] = Color;
                level[S] = -1;
                handle(g,S,level);
        }

        printf("\nNumber of Color used: %d\n", ColorNumber);

        for(int i = 0;i < g.num;i++){
                printf("Vertex[%d]-Color[%d]\n",i,ColorTable[i] );
        }
}
```

Reference source:

https://en.wikipedia.org/wiki/Gradient_boosting#Usage
https://www.youtube.com/watch?v=3CC4N4z3GJc
https://www.youtube.com/watch?v=2xudPOBz-vs&t=370s
https://www.youtube.com/watch?v=jxuNLH5dXCs&t=149s
https://www.youtube.com/watch?v=StWY5QWMXCw

Link source code:
https://drive.google.com/drive/folders/1zqf8T25ASoVOC-EZ3-_vOslJiCz4pXGQ?usp=sharing