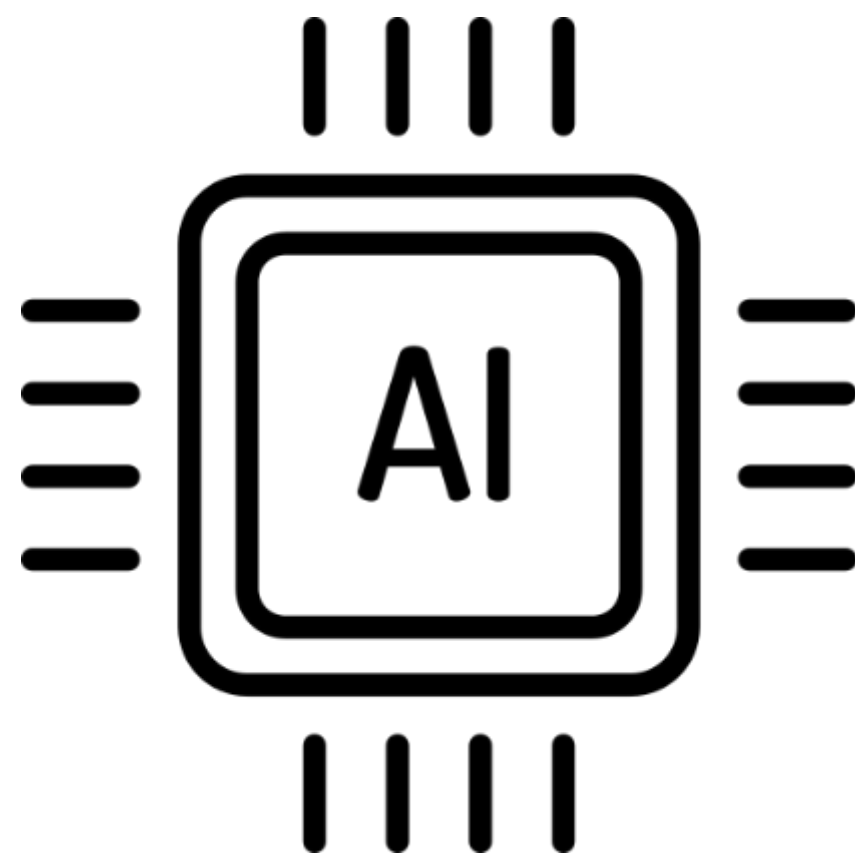

PORTFOLIO



이헌형
2020.03.03

CONTENTS



[1. ML Experience]

Kaggle - Predicting Future Sales
Space Weather - AI Contest
Big Contest - Champions League



[2. Projects]

3D Game - OpenGL
Raspberry Pi - CNN



[3. Etc]

Mediation of Color in Odor – Emotion
Association (accepted by Kiise)
Algorithm

1. ML Experience

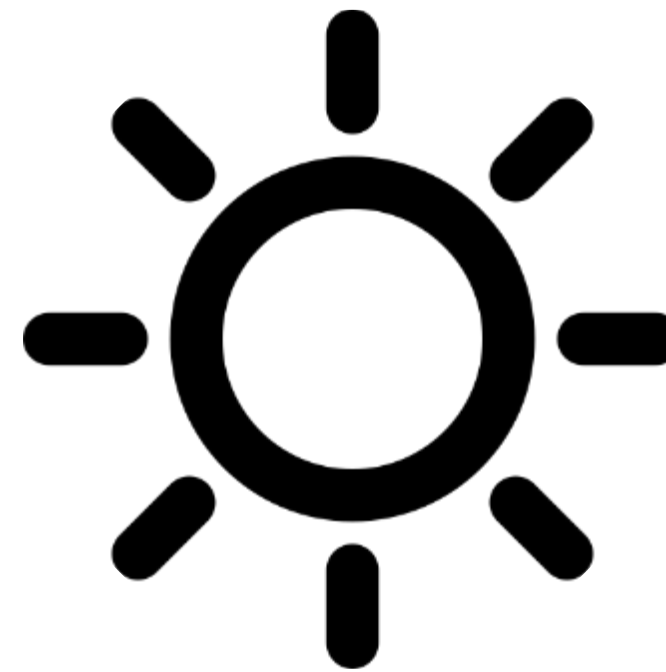


[Kaggle] Predicting Future Sales

Predicting total sales of items in a certain months

Python – XGBoost, LightGBM

Time Series Data



[Space Weather] AI Contest

Predicting space radio calamity

Python – LightGBM

Geomagnetic Disturbance Data



[Big Contest] Champions League

Predicting users who quit the game soon

Python - LightGBM

Game Data Log



Kaggle - Predict Future sales

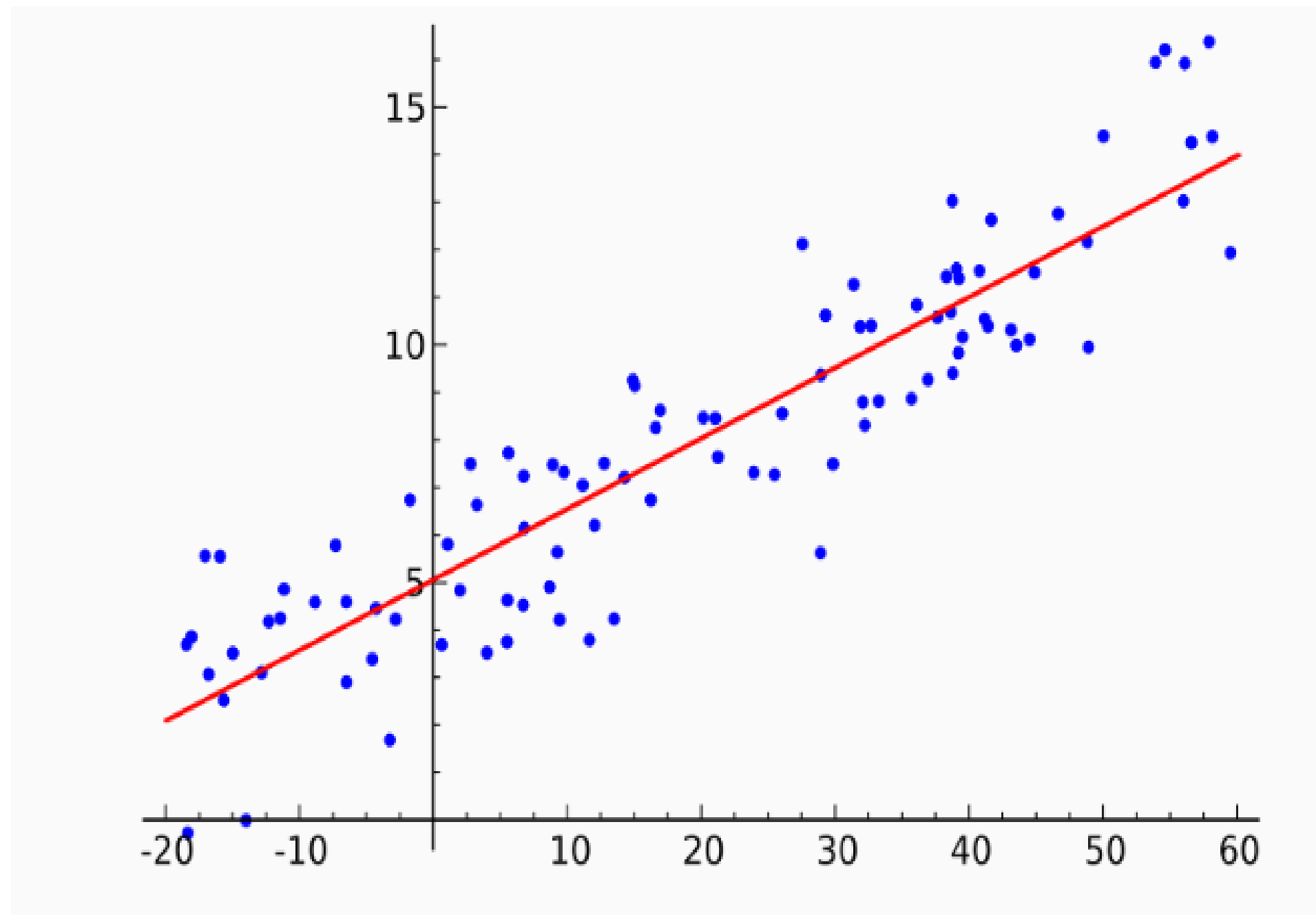
Mission

The task is to forecast the total amount of products sold in every shop for the test set.

Data Set : Russian software firms - 1C Company

<https://drive.google.com/drive/folders/1N1QyCkOQW1fPncfWZOcNehhRr2ZDc8Zo?usp=sharing>

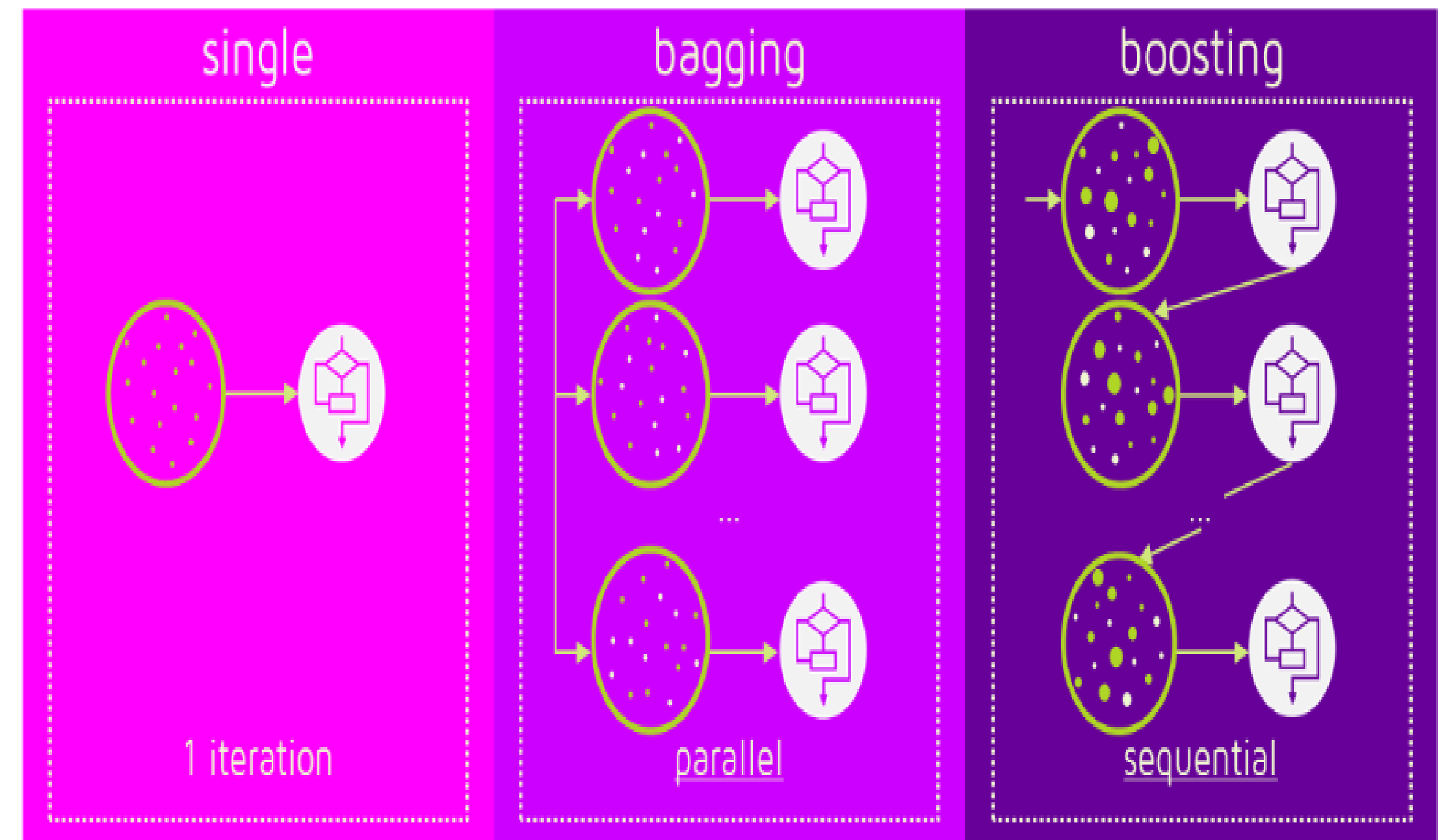
ALGORITHMS



WORKS

Normal Regression

Gradient Descent Algorithm



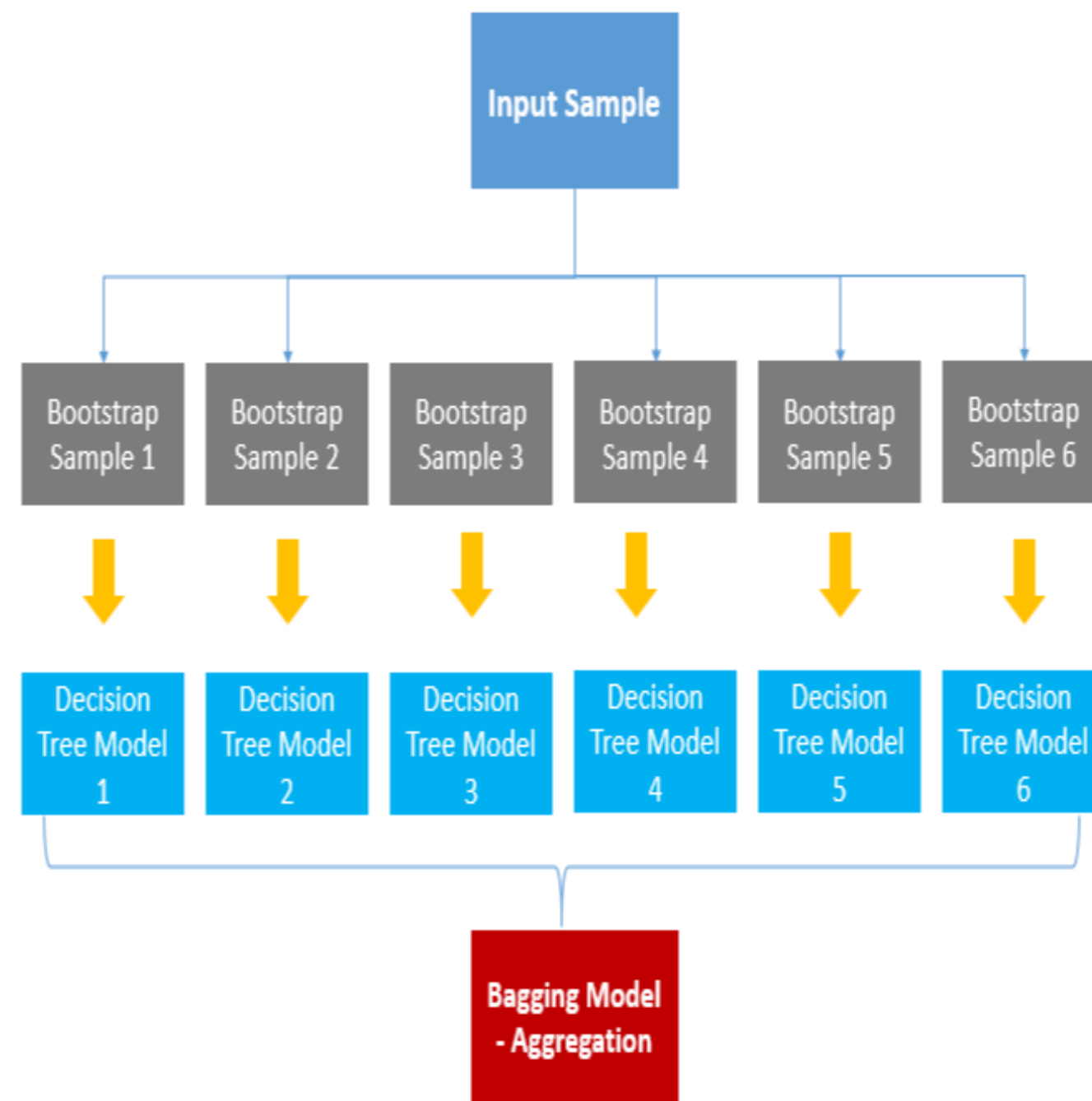
WORKS

ENSEMBLE

Bagging, Boosting, Bootstrap

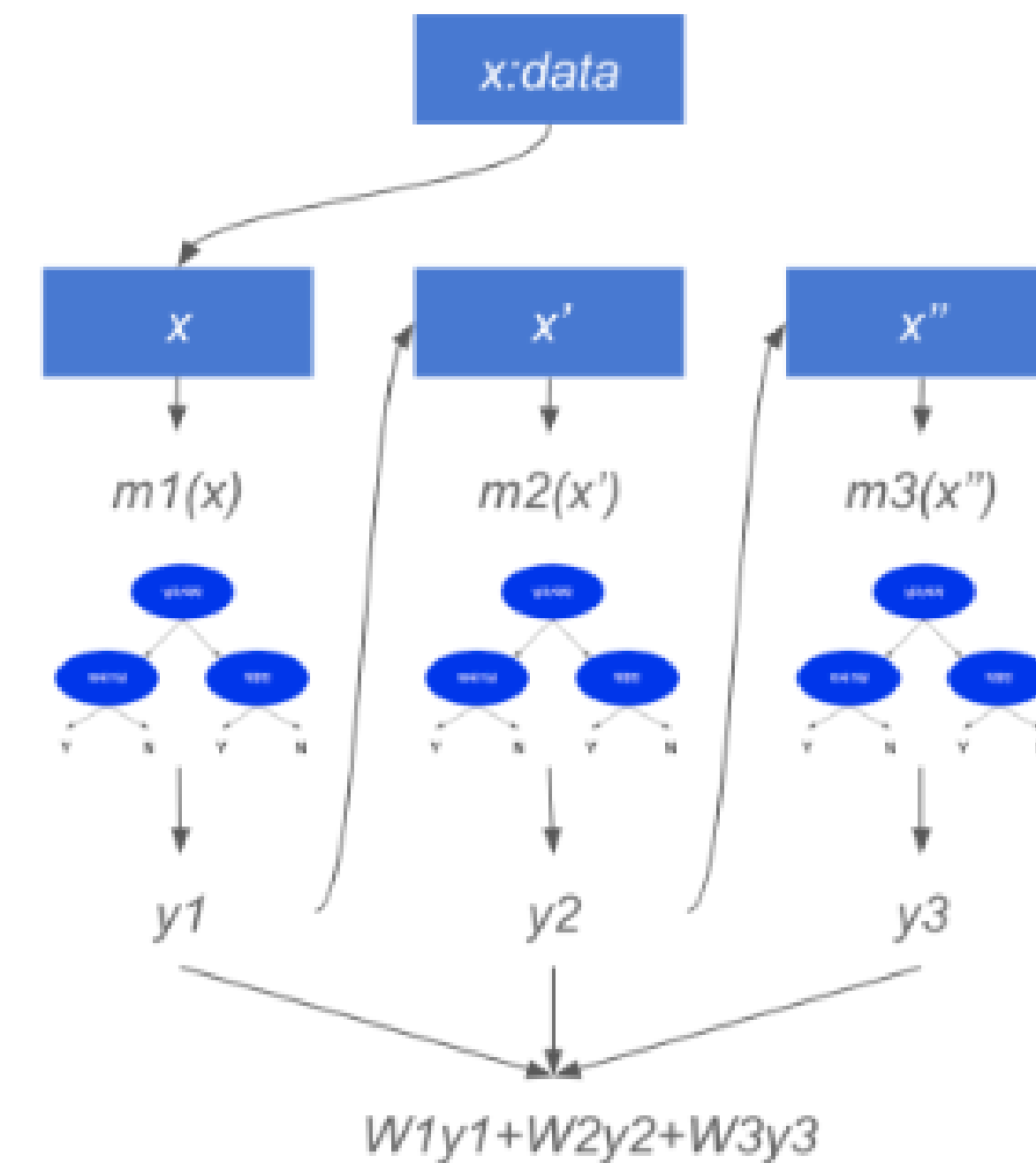
Then, What is Ensemble exactly?

ENSEMBLE



Bagging

1. Create several bootstrap data from Input sample.
2. Train and make a model using each bootstrap samples repectively.
3. Combine for making a final churn prediction model.
4. Target value
 - Nominal : Voting which model is best
 - Numeric : Taking their average

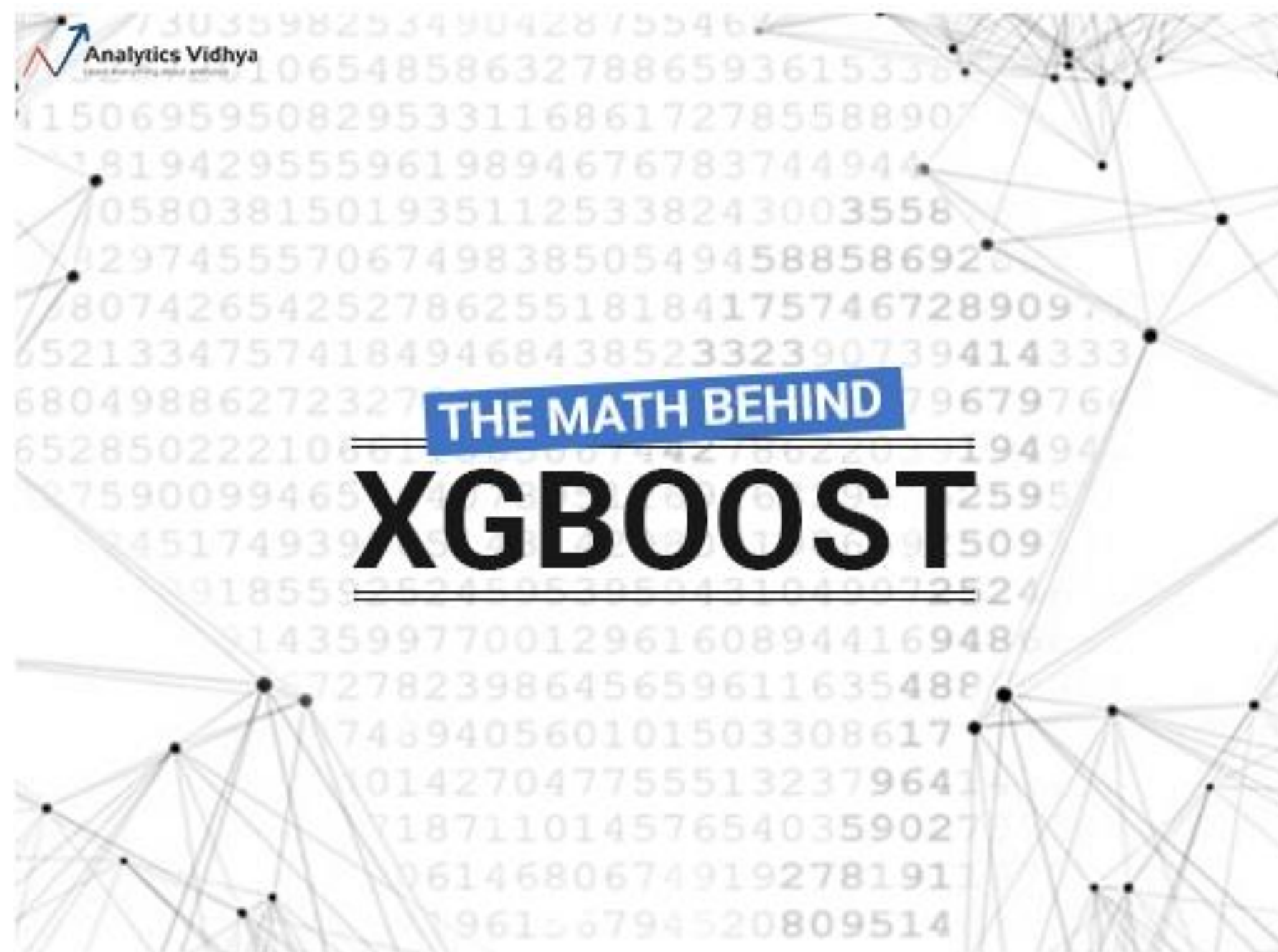


Boosting

1. Create several bootstrap data from Input sample.
2. Focus on wrong classified data and make new classification rule to make up for the data.
3. That is, Boosting concentrates problems which are hard to solve.

**Boosting outperforms
but is vulnerable to outlier.**

Boosting Methods.



LightGBM

1. Having better performance at regression and classification
2. Faster than normal gradient boosting machine using parallel work
3. Having own overfitting-proof function

1. Even faster than XGBoost
2. Less memory, faster prediction
3. Automatically convert categorical feature (auto one hot encoding)

So we used normal regression, XGBoost, LightGBM.

Preprocessing

It was time series data.
We had to use lag feature!

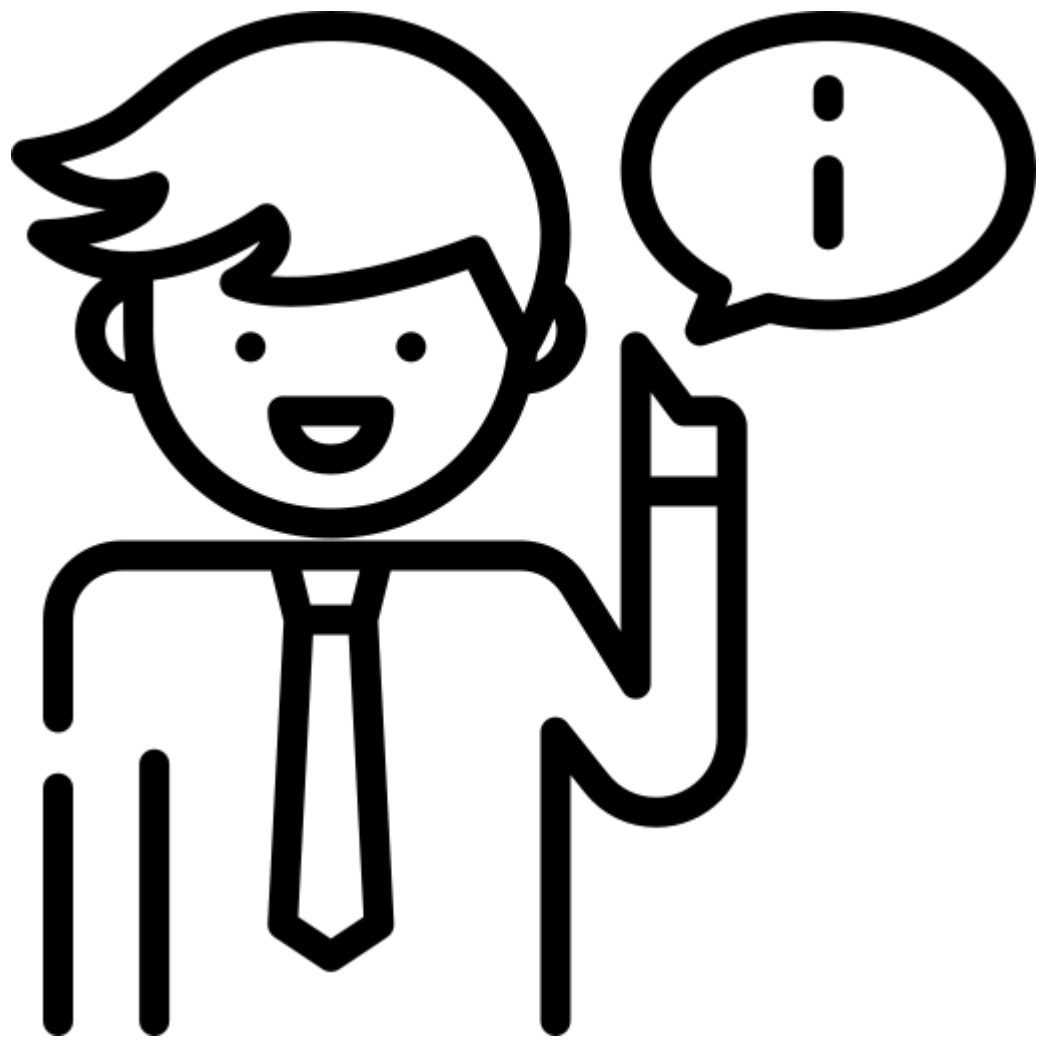


Then... what is lag?

| | date | date_block_num | shop_id | item_id | item_price | item_cnt_day |
|----|------------|----------------|---------|---------|------------|--------------|
| 0 | 2013-01-02 | 0 | 59 | 22154 | 999.00 | 1.0 |
| 1 | 2013-01-03 | 0 | 25 | 2552 | 899.00 | 1.0 |
| 2 | 2013-01-05 | 0 | 25 | 2552 | 899.00 | -1.0 |
| 3 | 2013-01-06 | 0 | 25 | 2554 | 1709.05 | 1.0 |
| 4 | 2013-01-15 | 0 | 25 | 2555 | 1099.00 | 1.0 |
| 5 | 2013-01-10 | 0 | 25 | 2564 | 349.00 | 1.0 |
| 6 | 2013-01-02 | 0 | 25 | 2565 | 549.00 | 1.0 |
| 7 | 2013-01-04 | 0 | 25 | 2572 | 239.00 | 1.0 |
| 8 | 2013-01-11 | 0 | 25 | 2572 | 299.00 | 1.0 |
| 9 | 2013-01-03 | 0 | 25 | 2573 | 299.00 | 3.0 |
| 10 | 2013-01-03 | 0 | 25 | 2574 | 399.00 | 2.0 |
| 11 | 2013-01-05 | 0 | 25 | 2574 | 399.00 | 1.0 |
| 12 | 2013-01-07 | 0 | 25 | 2574 | 399.00 | 1.0 |
| 13 | 2013-01-08 | 0 | 25 | 2574 | 399.00 | 2.0 |
| 14 | 2013-01-10 | 0 | 25 | 2574 | 399.00 | 1.0 |
| 15 | 2013-01-11 | 0 | 25 | 2574 | 399.00 | 2.0 |
| 16 | 2013-01-13 | 0 | 25 | 2574 | 399.00 | 1.0 |
| 17 | 2013-01-16 | 0 | 25 | 2574 | 399.00 | 1.0 |
| 18 | 2013-01-26 | 0 | 25 | 2574 | 399.00 | 1.0 |
| 19 | 2013-01-27 | 0 | 25 | 2574 | 399.00 | 1.0 |
| 20 | 2013-01-09 | 0 | 25 | 2593 | 279.00 | 1.0 |

Preprocessing - lag

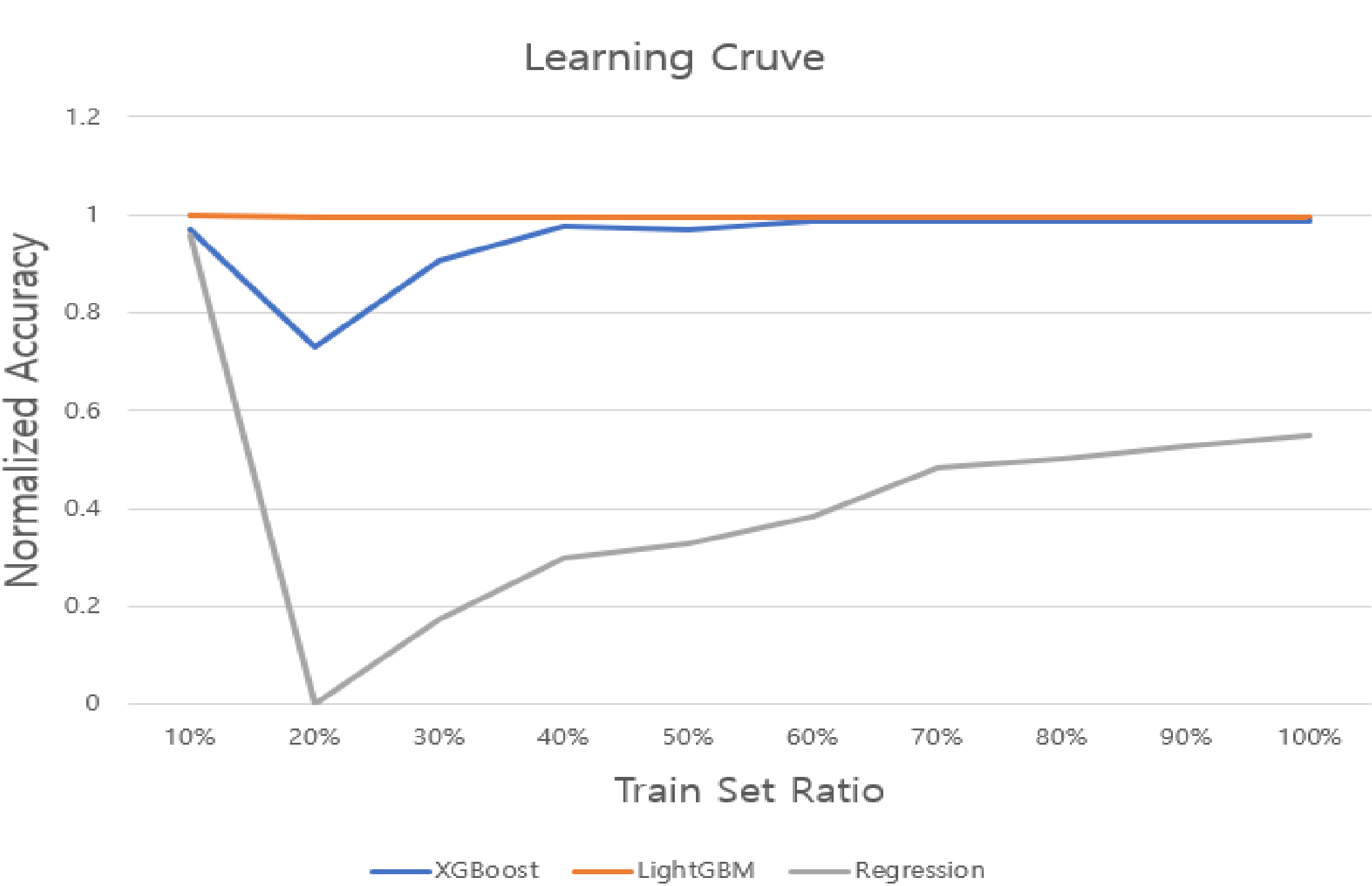
Lag is...



1. Lag is used in prediction of time series data.
2. Consider previous date so as to use current date as a label.
3. That is, we can use current date's attributes as a label to train!

| month | net_sales | previous_month_sales |
|-------|-----------|----------------------|
| 1 | 381430 | NULL |
| 2 | 200657 | 381430 |
| 3 | 363990 | 200657 |
| 4 | 817920 | 363990 |
| 6 | 189 | 817920 |
| 7 | 11338 | 189 |
| 8 | 8378 | 11338 |
| 9 | 8964 | 8378 |
| 10 | 3781 | 8964 |
| 11 | 11362 | 3781 |
| 12 | 6517 | 11362 |

Result



RMSE of Learning Curve

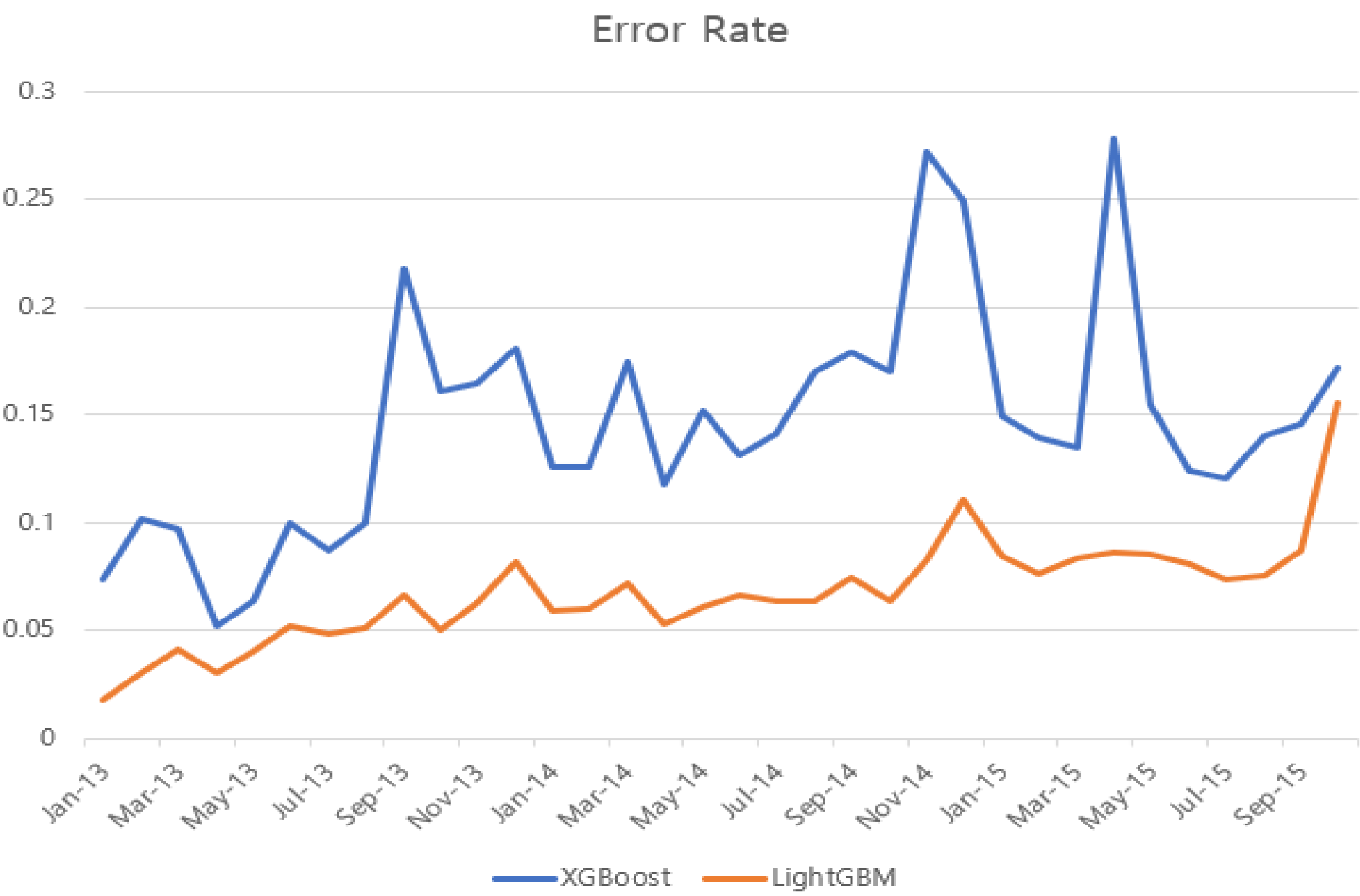
| Ratio(%) | 10 | 20 | 30 | 40 | 50 | 60 | 70 | 80 | 90 | 100 |
|------------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|
| XGBoost | 0.536 | 4.139 | 1.474 | 0.473 | 0.574 | 0.307 | 0.288 | 0.273 | 0.307 | 0.257 |
| LightGBM | 0.102 | 0.176 | 0.154 | 0.167 | 0.187 | 0.169 | 0.168 | 0.163 | 0.157 | 0.155 |
| Regression | 0.705 | 15.14 | 12.54 | 10.63 | 10.21 | 9.376 | 7.877 | 7.623 | 7.214 | 6.894 |

Minimum RMSE

| Algorithms | XGBoost | LightGBM | Regression |
|------------|---------|----------|------------|
| RMSE(100%) | 0.257 | 0.155 | 6.894 |

1. Before 40%, LightGBM has the greater accuracy.
But after that, LightGBM and XGBoost is almost same.
2. When we submit a result using test set, XGBoost has more accuracy.
3. So we chose Xgboost as a final model.

Result



Submit Result

| Algorithms | Benchmark | Lag Feature | Flatten |
|------------|-----------|-------------|---------|
| RMSE | 0.906 | 1.061 | 1.17 |

- 1. Benchmark was that we copied a best voted kernel in Kaggle.
 - 2. Lag Feature and Flatten are what we have done.
 - 3. Flatten is another preprocessing.
- Simply, it use all the months before last monts as attributes.

- 1. From Jan.2013 to Oct.2015, we took each month as a label.
- 2. XGBoost’s average error rate is 0.14592
- 3. LightGBM’s average error rate is 0.06726

But, currently we are... 153/5884 :)

153

Taker

0.88794

40

3mo

Your Best Entry ↑

Your submission scored 0.92266, which is not an improvement of your best score. Keep trying!

Discussion

Statistical Significance

| | ID | item_cnt_month |
|-------|---------------|----------------|
| count | 214200.000000 | 214200.000000 |
| mean | 107099.500000 | 0.207311 |
| std | 61834.358168 | 0.936287 |

Total instances are 214200.

so we assume that it follows binomial distribution.

Confidence Interval

At a statistical power of 80%, $0.206 < P < 0.208$

At a statistical power of 90%, $0.205 < P < 0.209$

Algorithms' Optimum RMSE

| Size of Sets | 125950 | 127235 | 128520 | 128520 | 131090 |
|--------------|---------|---------|---------|---------|---------|
| XGBoost | 0.25584 | 0.25775 | 0.25564 | 0.25553 | 0.24799 |
| LightGBM | 1.41482 | 0.3369 | 1.08032 | 0.58752 | 0.30169 |
| Regression | 6.9952 | 5.8847 | 6.8155 | 6.7161 | 6.894 |

ANOVA TEST

| Source of variation | SS | df | MS | F | P-value | F crit |
|---------------------|----------|----|----------|----------|----------|----------|
| Between Groups | 127.1546 | 2 | 63.57732 | 437.3196 | 6.15E-12 | 3.885294 |
| Within Groups | 1.744554 | 12 | 0.14538 | | | |
| Total | 128.8992 | 14 | | | | |

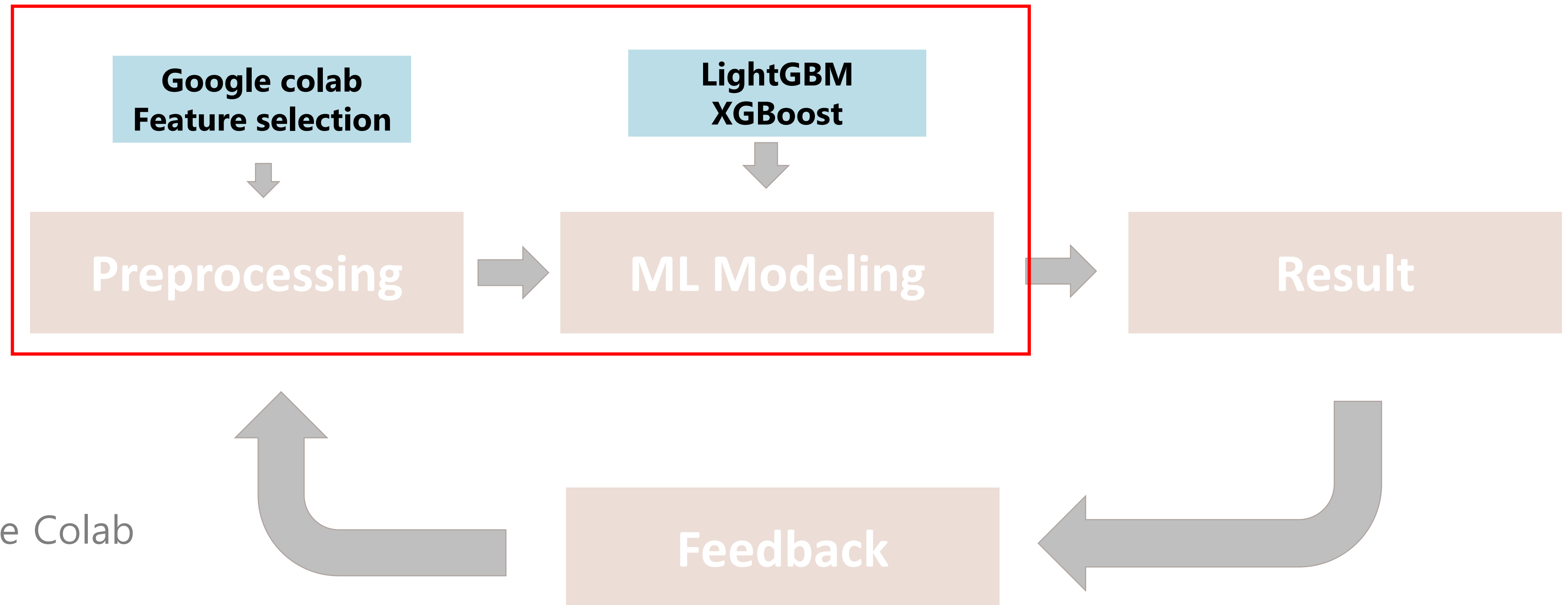
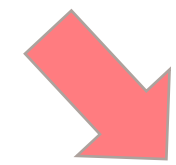
we can reject null hypothesis, due to

$F : 437.3196 > F \text{ crit} : 3.885294$.

Therefore, there are meaningful differences between algorithms.

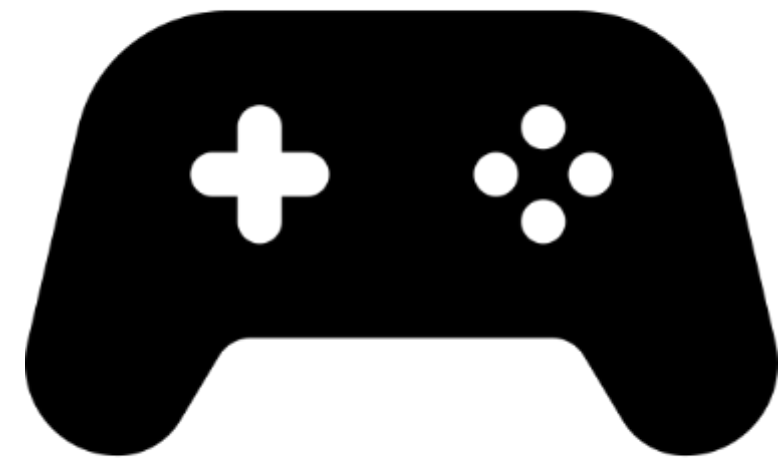
My Work

My Work



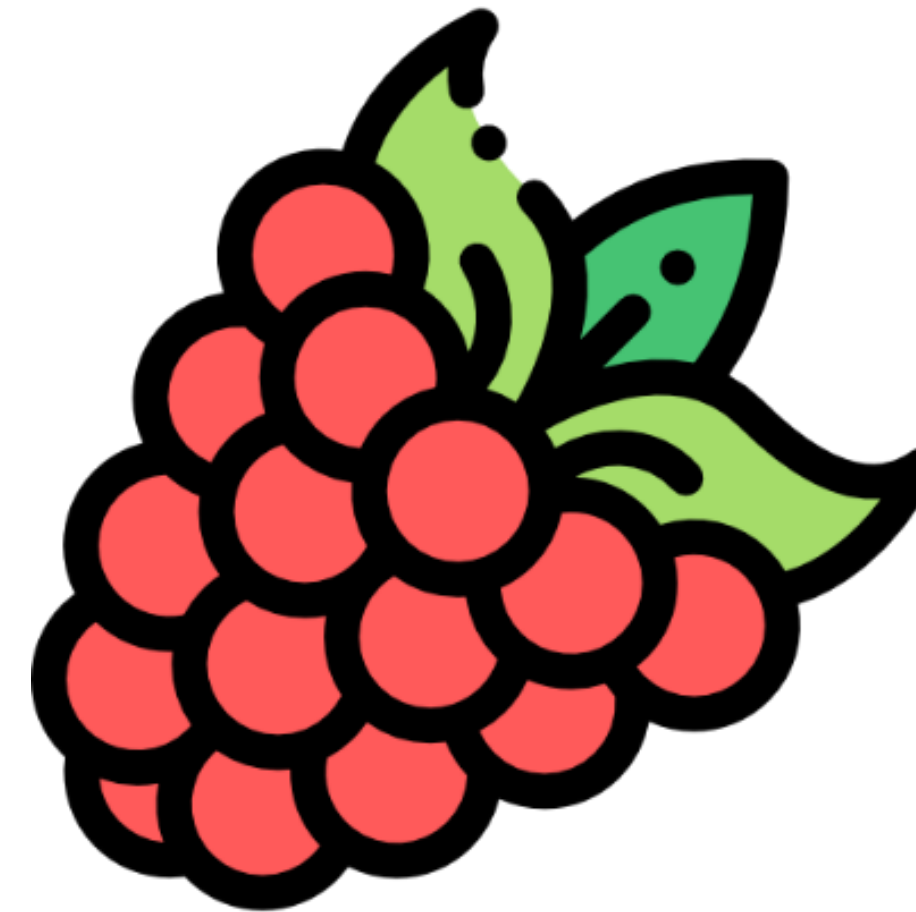
- Enviornment : google Colab
- Language : python
- Libraries : pandas, sklearn, lightgbm, xgboost
- Duration : Oct.2019 ~ Dec.2019
- Member : 김영준, 남현석, 윤인식, 이헌형, 박상수

2. Extra Experience



**[3D Game]
Escape Room**

Open GL



**[Raspberry Pi]
CNN**

Smart Glasses
For Facial Recognition Disorder



3D Game – Escape Room

Mission

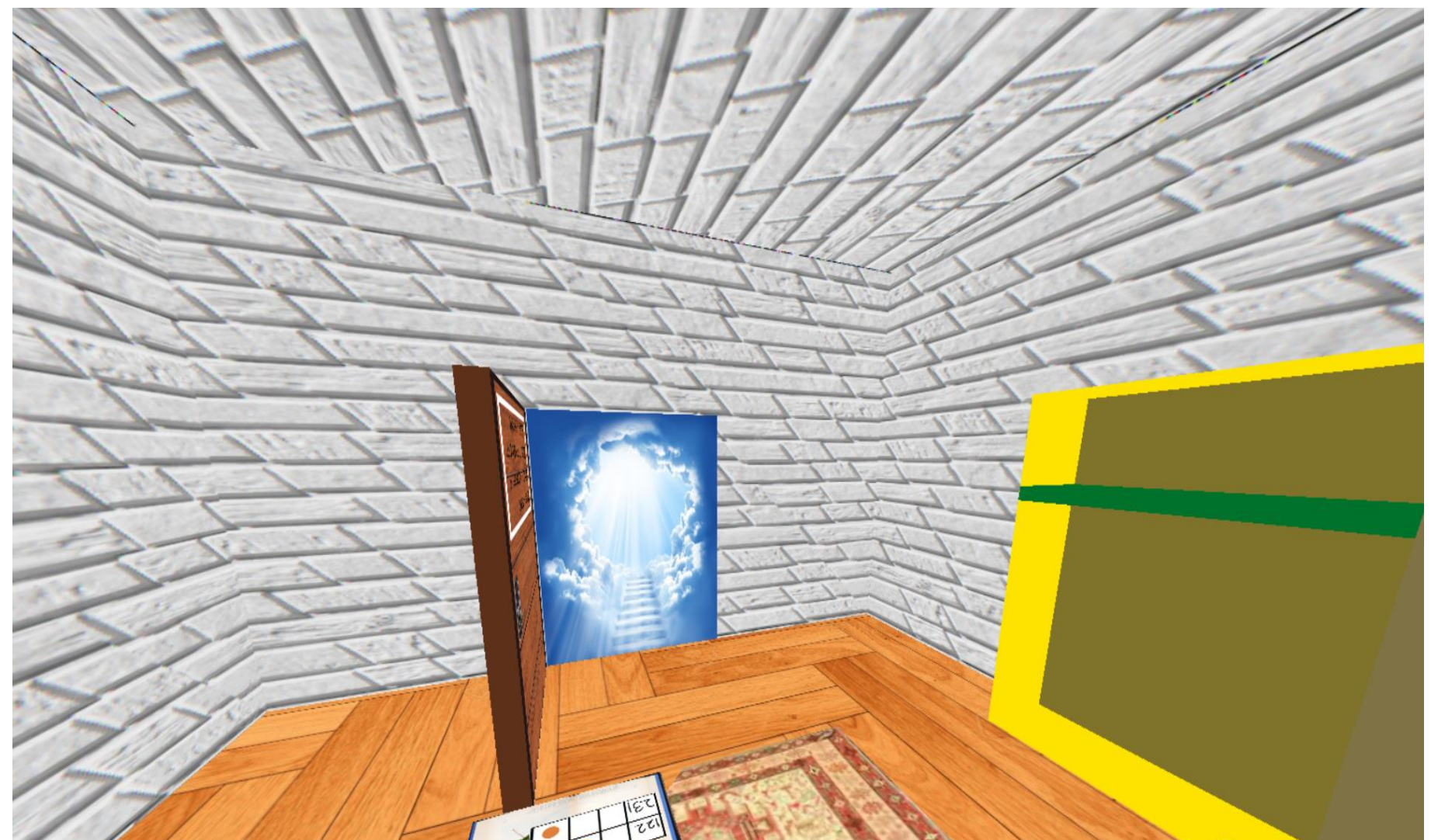
The task is to Implement a 3d game using openGL.

Goal of the game : Escape the room using given hints.

Download - <https://github.com/suyeony0/EscapeRoom>

Motivation

1. It's fun!
2. OpenGL has lots of call back functions and the structures are easy.
To optimize these merits, 3D game is best fit.
3. The genre : Escape room looks easy.
So nobody has a hesitation to try.



Function Descriptions - 1



[Moving]

Keyboard callback

Press w,a,s,d to go, left, back, right.

Press space to jump, c to sit



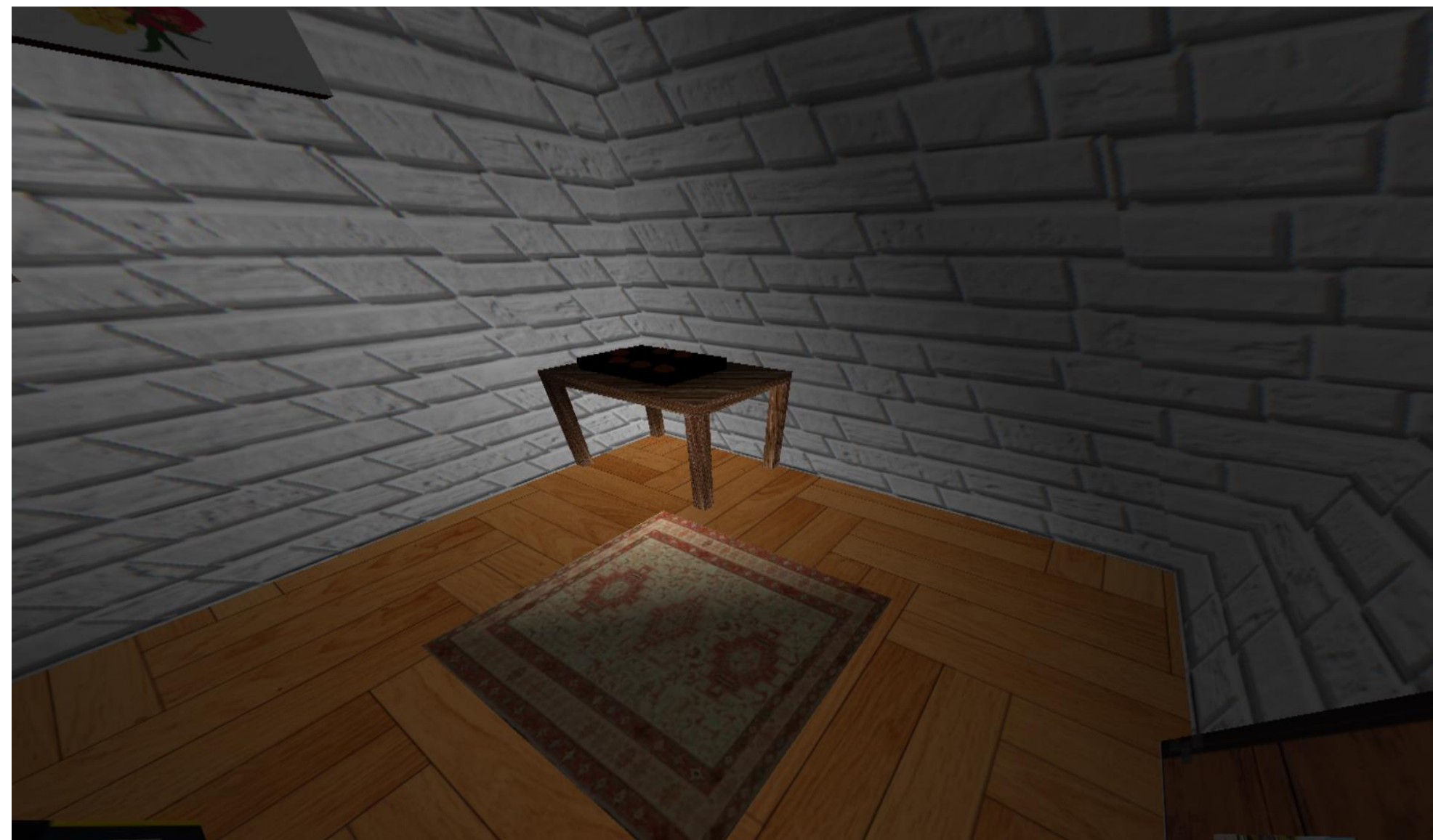
[Eye-tracking]

PassivMotion callback

First person view(like FPS)

Move mouse to rotate viewpoint

Function Descriptions - 2



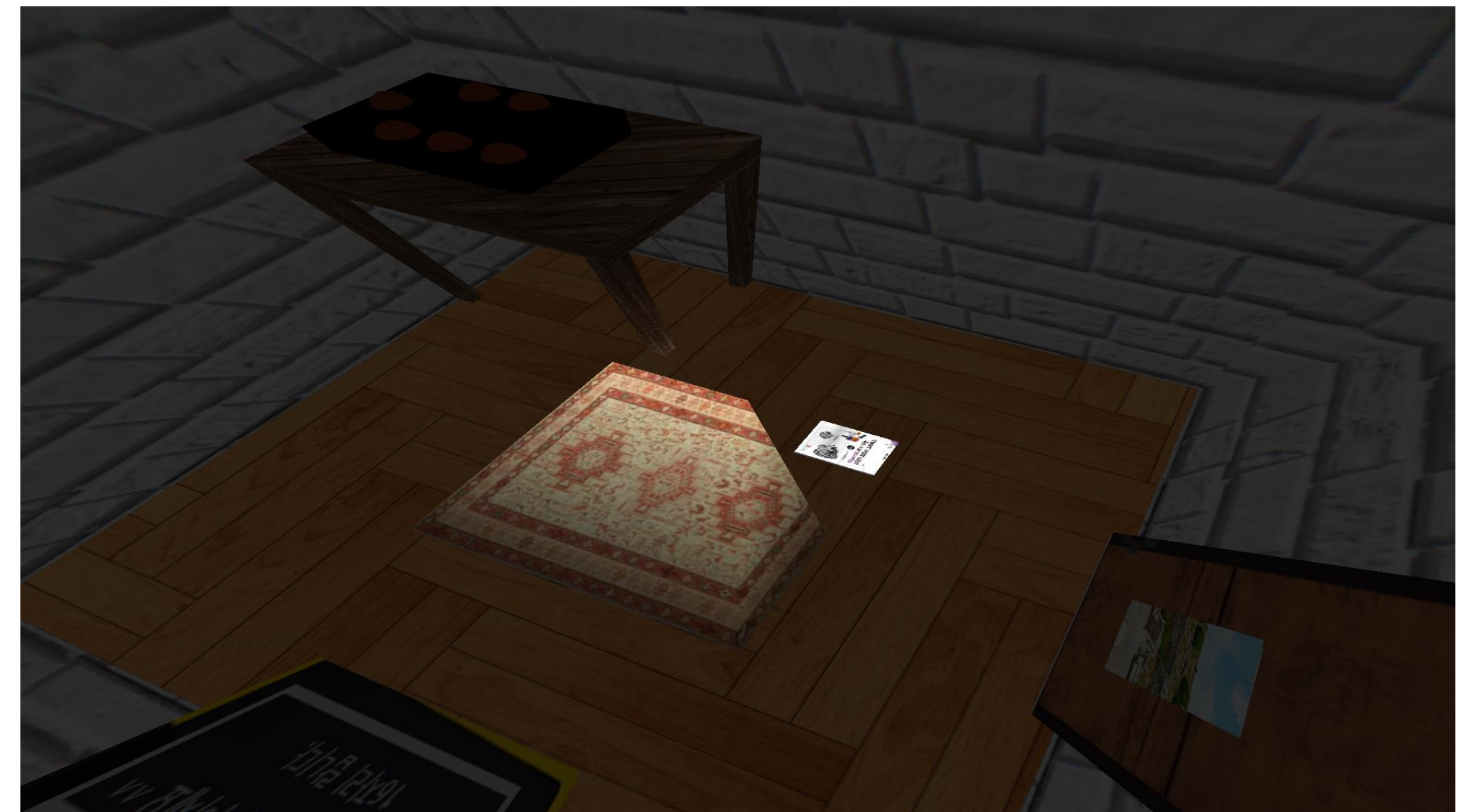
[Texture Mapping]

OpenGL doesn't provide texture mapping.

AUX.h was needed.

Using LoadTexture() to read bitmap.

Then the bitmap will be bound
once an object is drawn.



[Object Seletion]

DEPTH_TEST was needed to get z_index and distance.

1. Once x,y coordinates were read, selectMode is implemented.
2. Take the object name and act a designed rule.

Function Descriptions - 3



[Light Handling]
glLight callback
Synchronize viewpoint and light.
Once PostRedisplay() occurs,
change the direction and initial point of light
so as to be acted like flash-light.



Hope you can get out!



Raspberry Pi - CNN

Mission

The task is to make a facial recognition equipment using Raspberry-pi.
For facial recognition disorder.

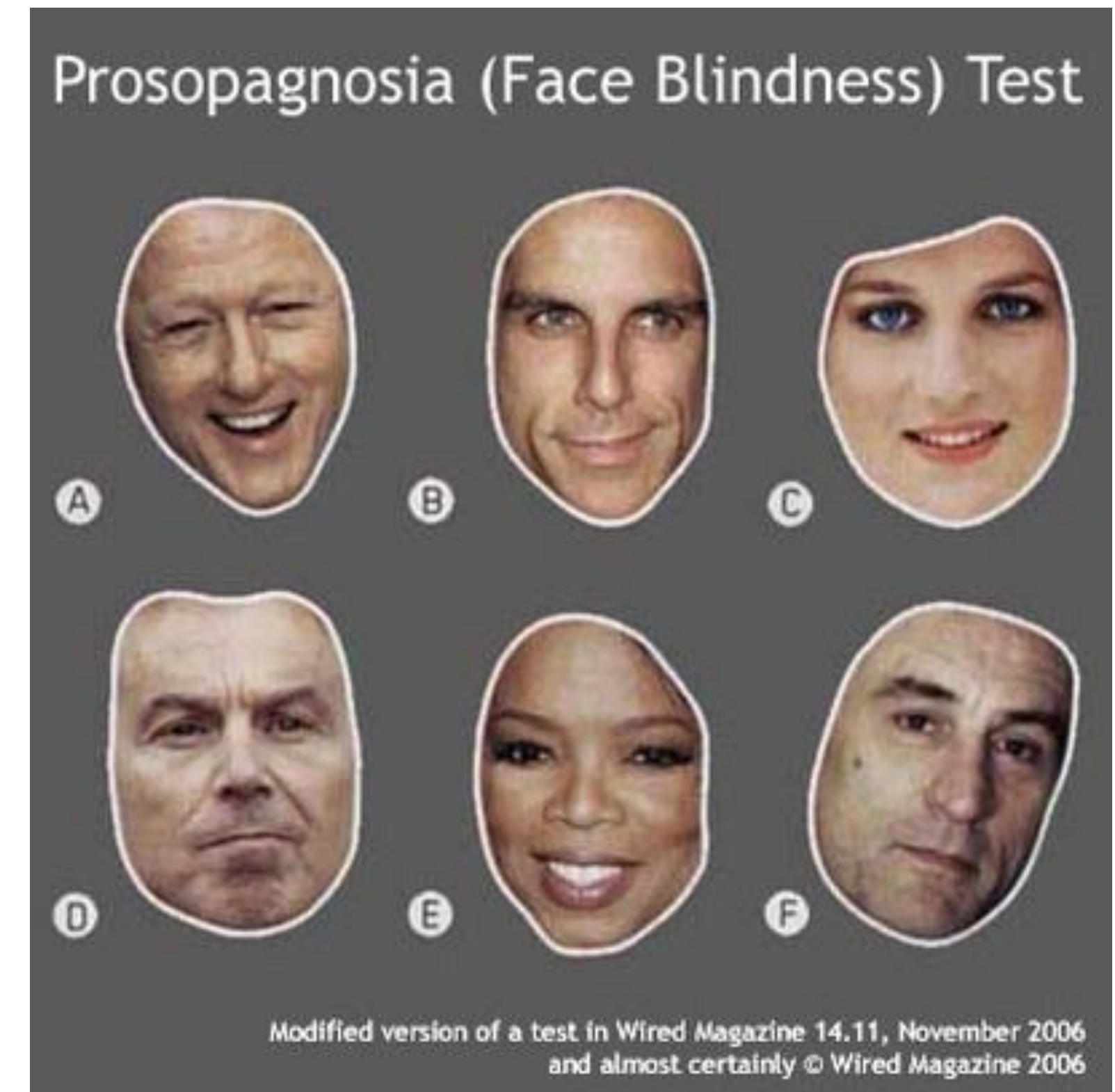
Problem Description

What is facial recognition disorder?

- Face Blindness : Prosopagnosia
- Who sees a face and says "It's apple."
- Who sees family and says "Nice to meet you."
- Both of congenital and acquired are possible.
- They distinguish people using hair, clothes, voice.
→ such a big problem to get into society.

Detail

- 2 out of 100 have the disorder.
- if Acquired : can be cured by curing the cause of the disease such as Stroke.
- if Congenital : No solution.
- Is there any Medical Supplementary Device? : No.



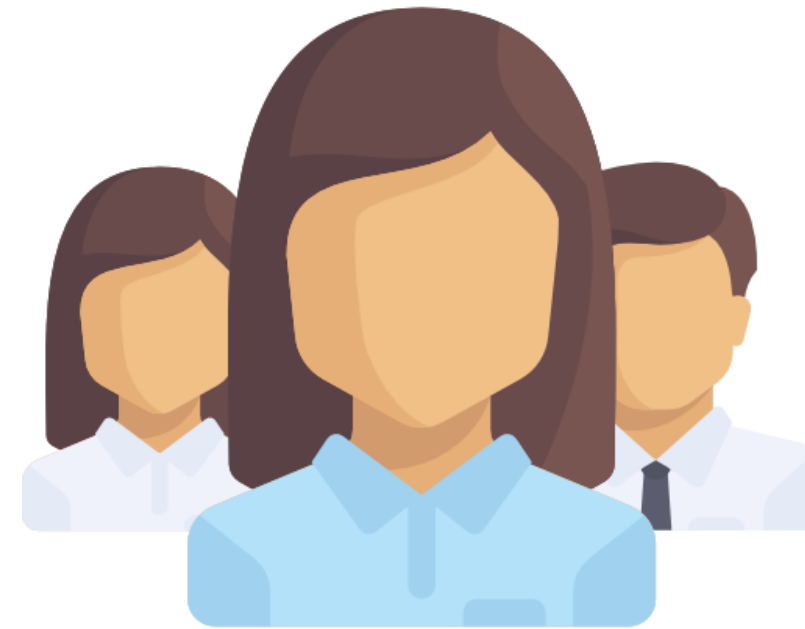
Solution



Scenario



Wearing the glasses
starting the app.



Nothing happens
for unregistered person.



Name appears
If the person is registered.

Methods

MTCNN

- Extract face's landmark



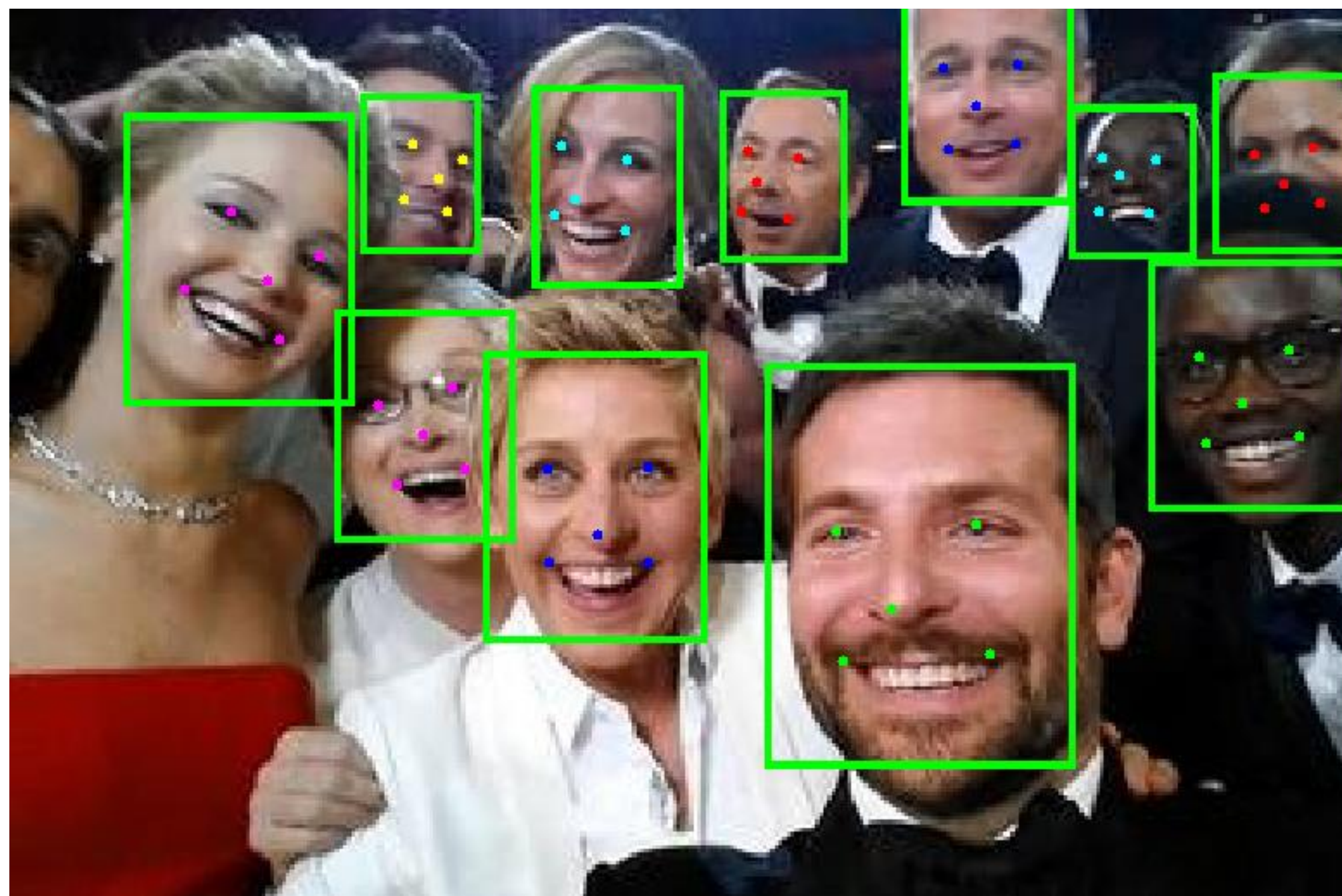
Preprocessing
OpenCV lib

- Crop the extracted region of face



FaceNet

A model for face recognition



MTCNN

FaceNet

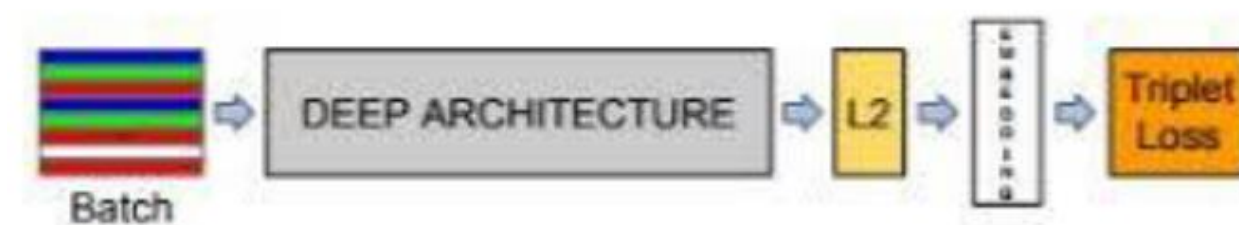
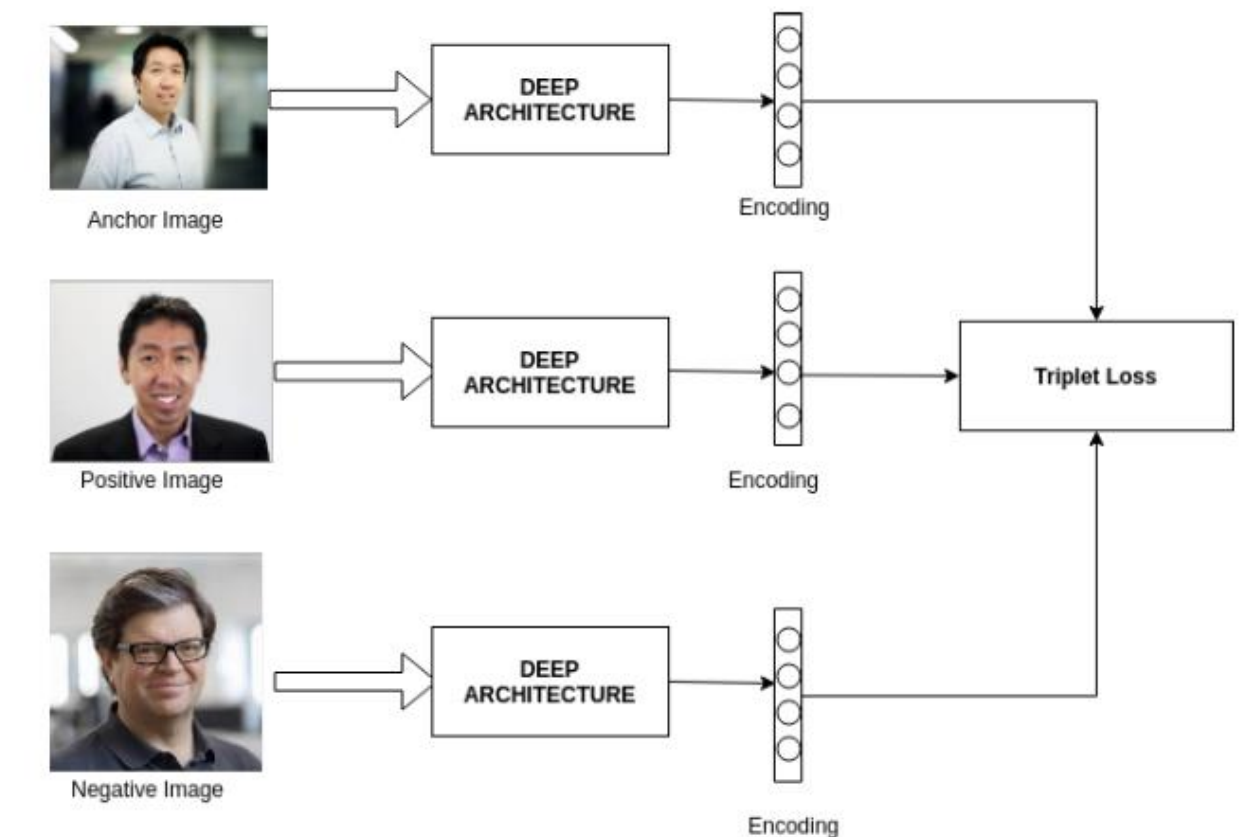
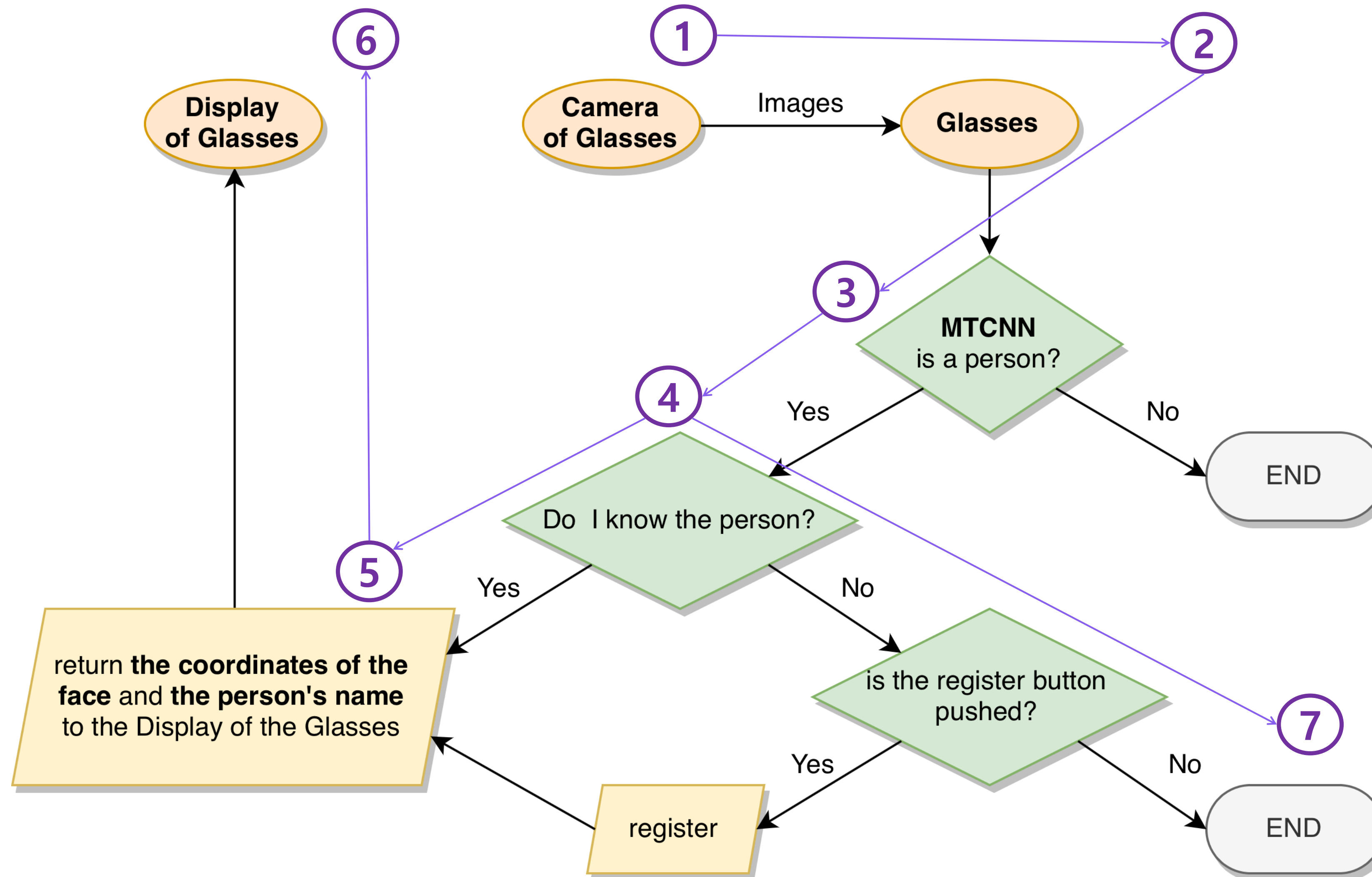


Figure 2. **Model structure.** Our network consists of a batch input layer and a deep CNN followed by L_2 normalization, which results in the face embedding. This is followed by the triplet loss during training.



FaceNet Architecture

Flowchart

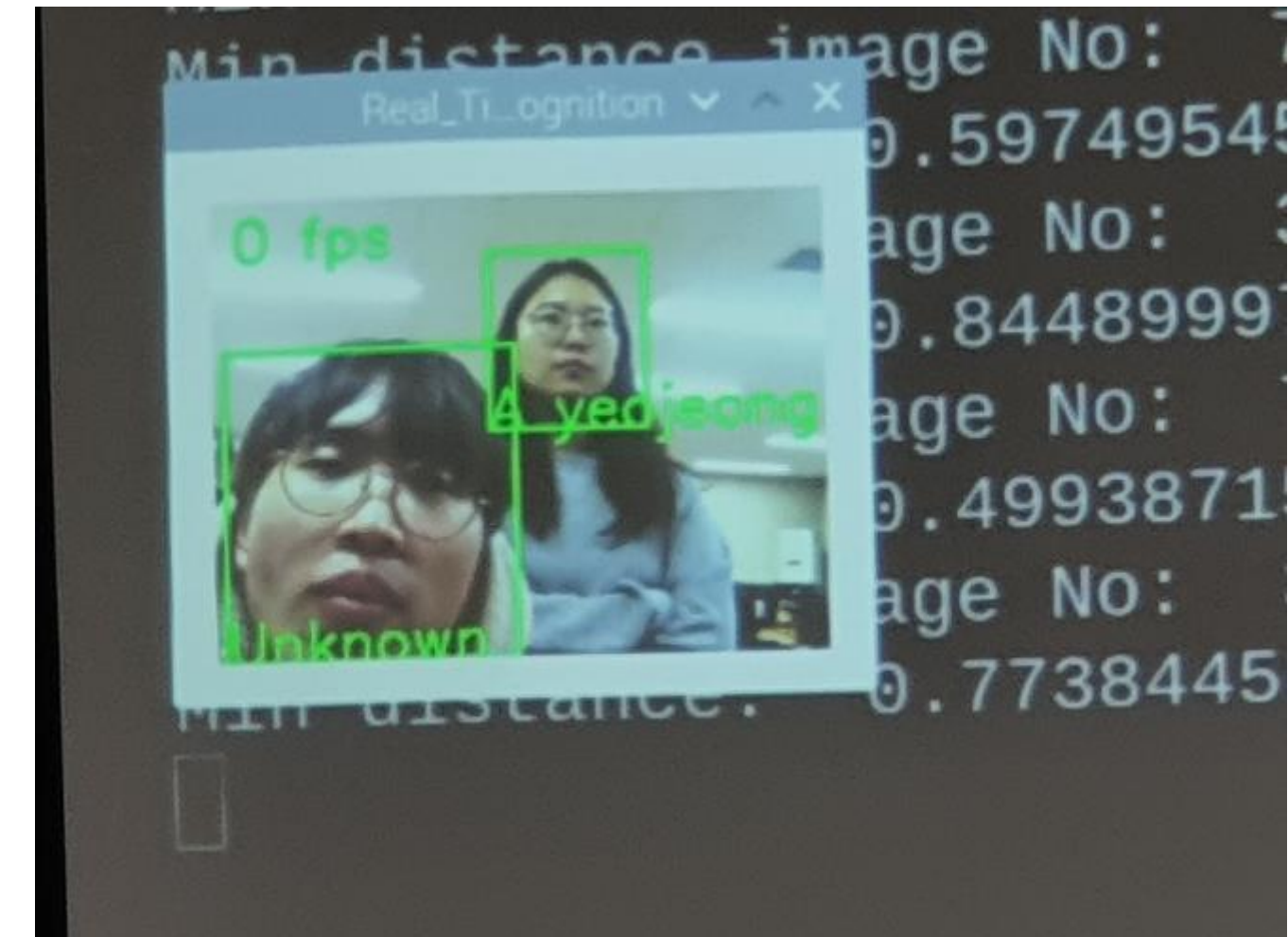


My work and Demo

My work is ...

1. Set up Raspberry pi – Linux
2. Configure all the required tools and libraries in the Raspberry pi
3. Put the model into the Raspberry pi
4. Make the model run

Honestly, the CNN part was not my job :(



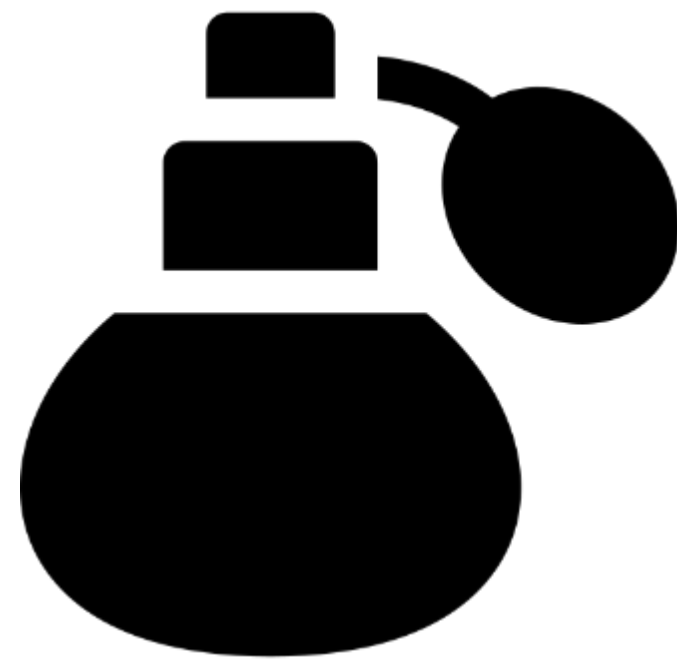
It's me!



3. etc

Mission

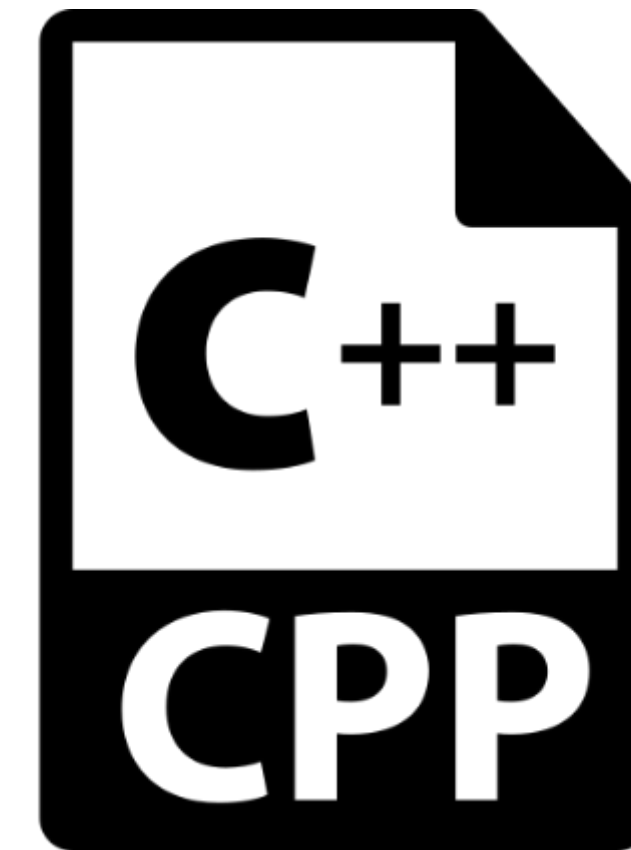
ExtraCurricular



[Paper]

Mediation of Color in Odor – Emotion
Association (accepted by Kiise)

<http://www.dbpia.co.kr/journal/articleDetail?nodeId=NODE08763620>



[Study Algorithm]

Leetcode, BaekJoon, Programmers

<https://github.com/suyeony0/Leetcode-Algorithm>

Paper

냄새와 감정의 색깔에 의한 매개성 연구

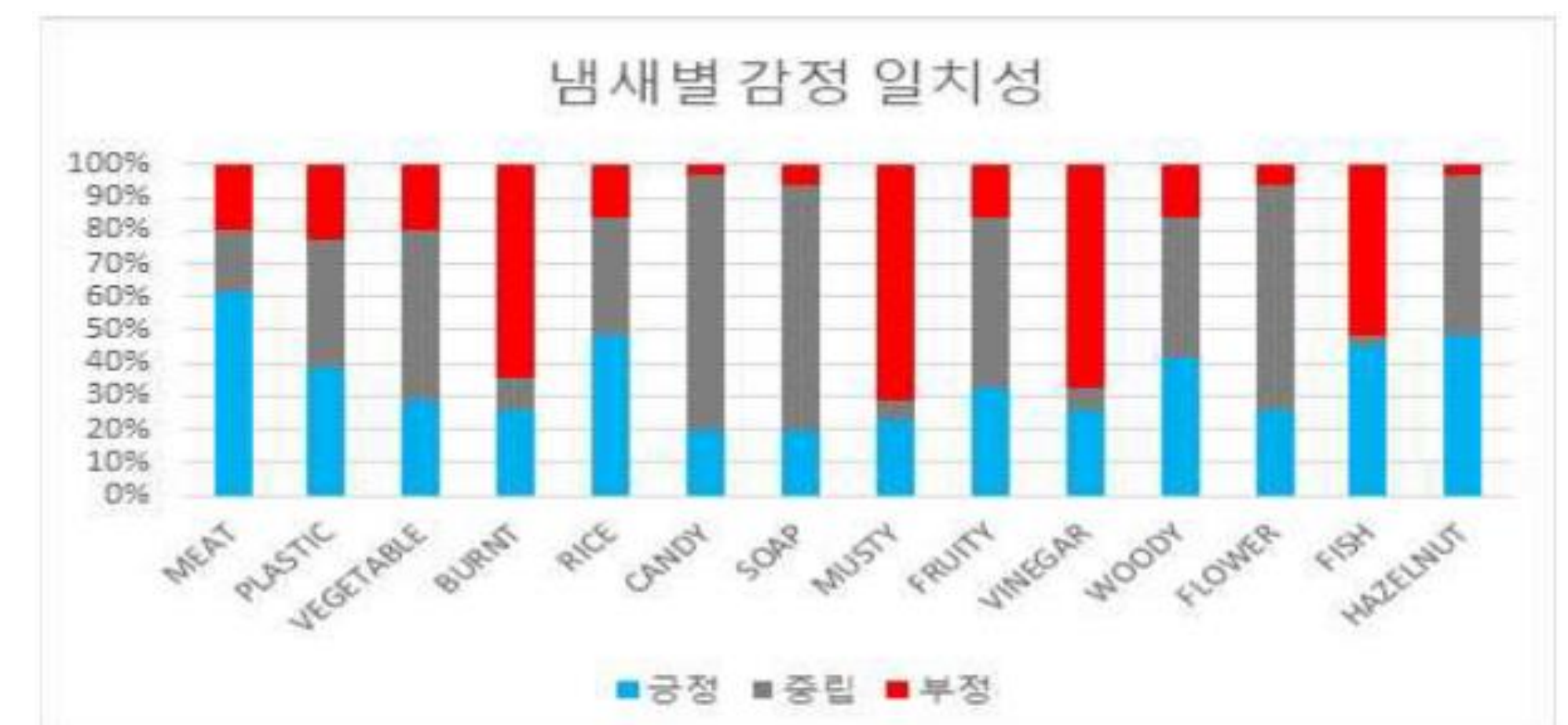
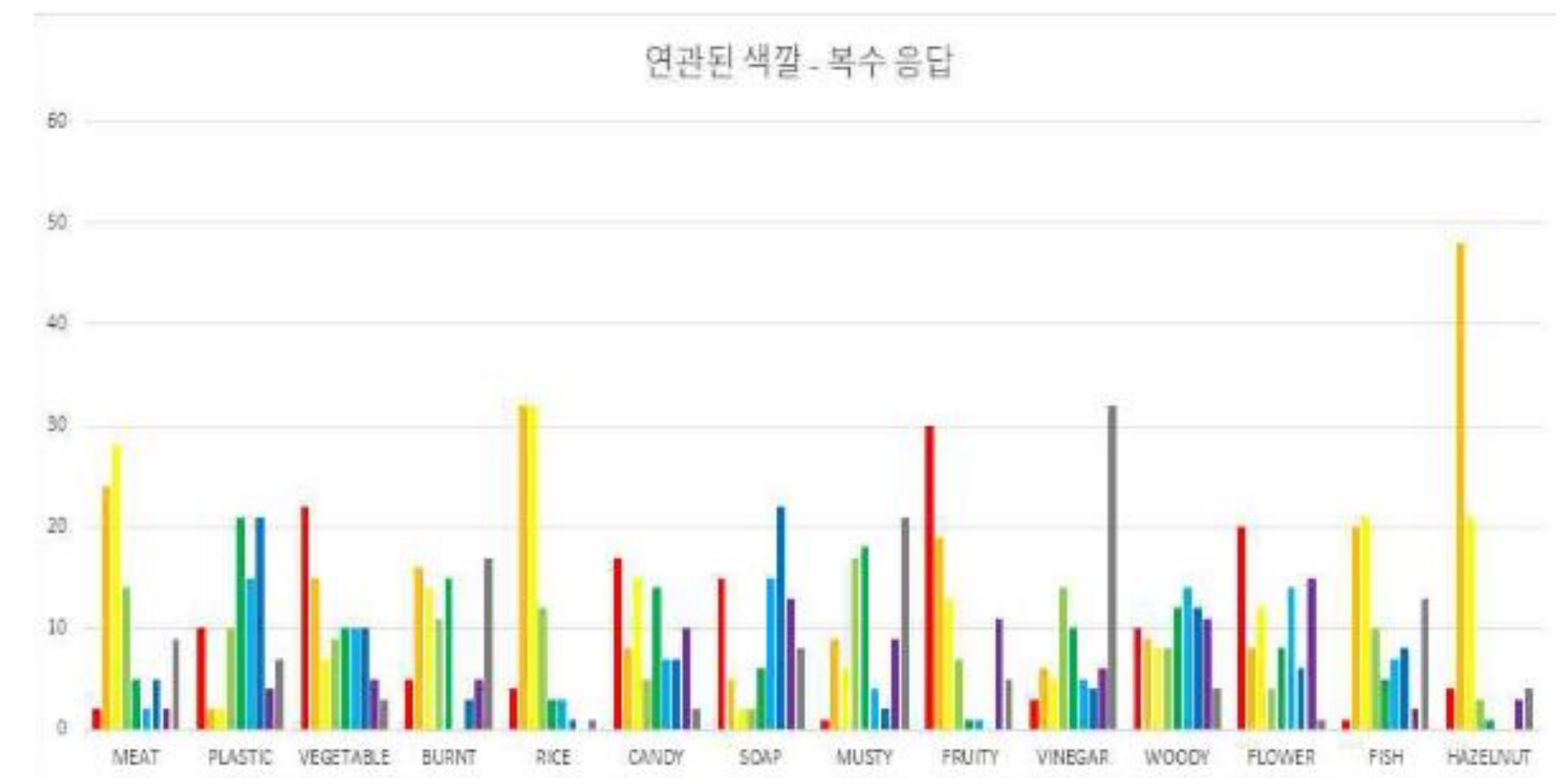
요약

인간의 감각적 양상은 심리학, 의학 등의 분야에서 치료 목적으로의 지속적인 활용과 연구가 진행되어왔다.

색깔과 소리등 연구가 활발한 다른 양상연구에 비해 냄새는 쉽게 적응되고 그 효과가 특정되지 않는다는 이유로 연구가 활발히 진행되지 못한 편이다.

본 연구는 냄새와 색깔의 연관성이 깊고 문화권내에 한정된다는 기존 연구를 바탕으로 한국 문화권내의 특정된 냄새-색깔, 냄새-감정의 존재유무와 색깔의 매개성 분석에 목적을 두고 연구를 진행하였다.

일상생활에서 맡을 수 있는 공통적인 냄새와 Berkeley color project의 색깔을 이용하여, 각 양상에 대한 감정을 피 실험자들의 주관적 피드백과 심박수와 근전도를 척도로한 객관적 수치의 연계를 통해 위의 가정을 분석하고자 하였다. 분석 결과 냄새, 색깔에 대한 피 실험자들의 감정은 높은 일치성을 보인 반면, 두 양상의 감정 비교의 일치성은 떨어지는 결과를 보였다.



About Me

Profile

- Birth : 1994.04.02
- Tel. : (+82)-10-6653-8277
- Email : gjsgud2@gmail.com
- Github : <https://github.com/suyeony0/>

Skills

- English : TOEIC(940), OPIC(AL), TEPS(374)
- Programming Language : C++, python

Honors

- Summa cum laude (the highest honor)
- GPA : 4.45 / 4.5



HeonHyung Lee
Computer Engineer

Education Histroy

- Catholic University of Korea (graduated) *FEB.2020*
- Ogeum High School (graduated) *FEB.2013*
- Language Study (London - UK) *FEB.2018 ~ AUG.2018*

ExtraCurricular

- [Big Contest] - Champions League *Jul.2019 – Sep.2019*
- [Space Weather] – AI Contest *Aug.2019 – Sep.2019*
- [Kaggle] - Predicting Future Sales *Nov.2019 – Dec.2019*
- <https://drive.google.com/drive/folders/1N1QyCkOQW1fPncfWZOcNehhRr2ZDc8Zo?usp=sharing>
- Undergraduate Research Student *Feb.2019 ~ Feb.2020*
- Mediation of Color in Odor – Emotion Association (accepted by Kiise)
- <http://www.dbpia.co.kr/journal/articleDetail?nodeId=NODE08763620>
- [3D game] – Escape Room(openGL) *Oct.2019~Dec.2019*
- <https://suyeony0.github.io/EscapeRoom/>
- [Algorithm] - <https://github.com/suyeony0/Leetcode-Algorithm>

References

Icon

[AI](#) : Icongeek26
[gps](#) : Freepik
 Thinking person : [Freepik](#)
 Money pocket : [CC Attribution](#)
 Sun : Freepik
 Warrior : Freepik
 느낌표 : [Freepik](#)
 Cpp : Freepik
 Python : www.python.org
 Ensemble : <https://swalloow.github.io/bagging-boosting>
 Bagging : <https://swalloow.github.io/bagging-boosting>
 Regression : https://en.wikipedia.org/wiki/Regression_analysis
 XGBoost : <https://www.analyticsvidhya.com/blog/2018/09/an-end-to-end-guide-to-understand-the-math-behind-xgboost/>
 lightGBM : <https://techsupportallbu>

Paper

[1] Tianqi Chen, Carlos Guestrin, 10 Jun 2016 , XGBoost: A Scalable Tree Boosting System, Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pages 785-794

 [2] Guolin Ke1 , Qi Meng2 , Thomas Finley3 , Taifeng Wang1 , Wei Chen1 , Weidong Ma1 , Qiwei Ye1 , Tie-Yan Liu, 2017-12-04, LightGBM: A Highly Efficient Gradient Boosting Decision Tree, Proceeding NIPS'17 Proceedings of the 31st International Conference on Neural Information Processing Systems Pages 3149-3157
gs.wordpress.com/2018/05/26/install-lightgbm-on-mac-osx-10-9/

Web

Lag : <https://www.sqlservertutorial.net/sql-server-window-functions/sql-server-lag-function/>
 Ensemble : <https://learningcarrot.wordpress.com/2015/11/12/%EB%B6%80%ED%8A%B8%EC%8A%A4%ED%8A%B8%EB%9E%A9%EC%97%90-%EB%8C%80%ED%95%98%EC%97%AC-bootstrapping/>
 Boosting : <https://dailyheumsi.tistory.com/111>