Introduction to XGBoost

2018 Fall CS376 Machine Learning
2018. 9. 20.

Joonyoung Yi
joonyoung.yi@kaist.ac.kr
** Slide heavily adopted from the XGBoost author's slide.*

KAIST

CONTENTS

CONTENTS

KAIST

MLILAB
Machine Learning & Intelligence

Why XGBoost is Important?

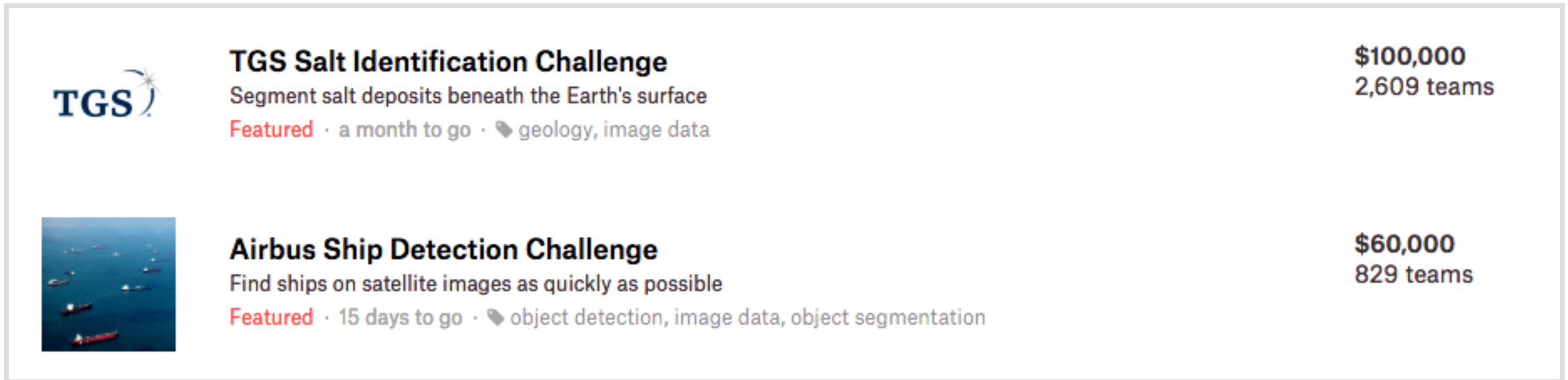
- XGBoost is a machine learning library like numpy, tensorflow, pytorch.
  - https://xgboost.readthedocs.io/en/latest/index.html

- XGBoost is a useful tool to achieve good performance in the Kaggle or data science competitions.

- If you don't know how to start the term project,
  I recommend you using the XGBoost without any reasons.
  I'll explain in the later slides.

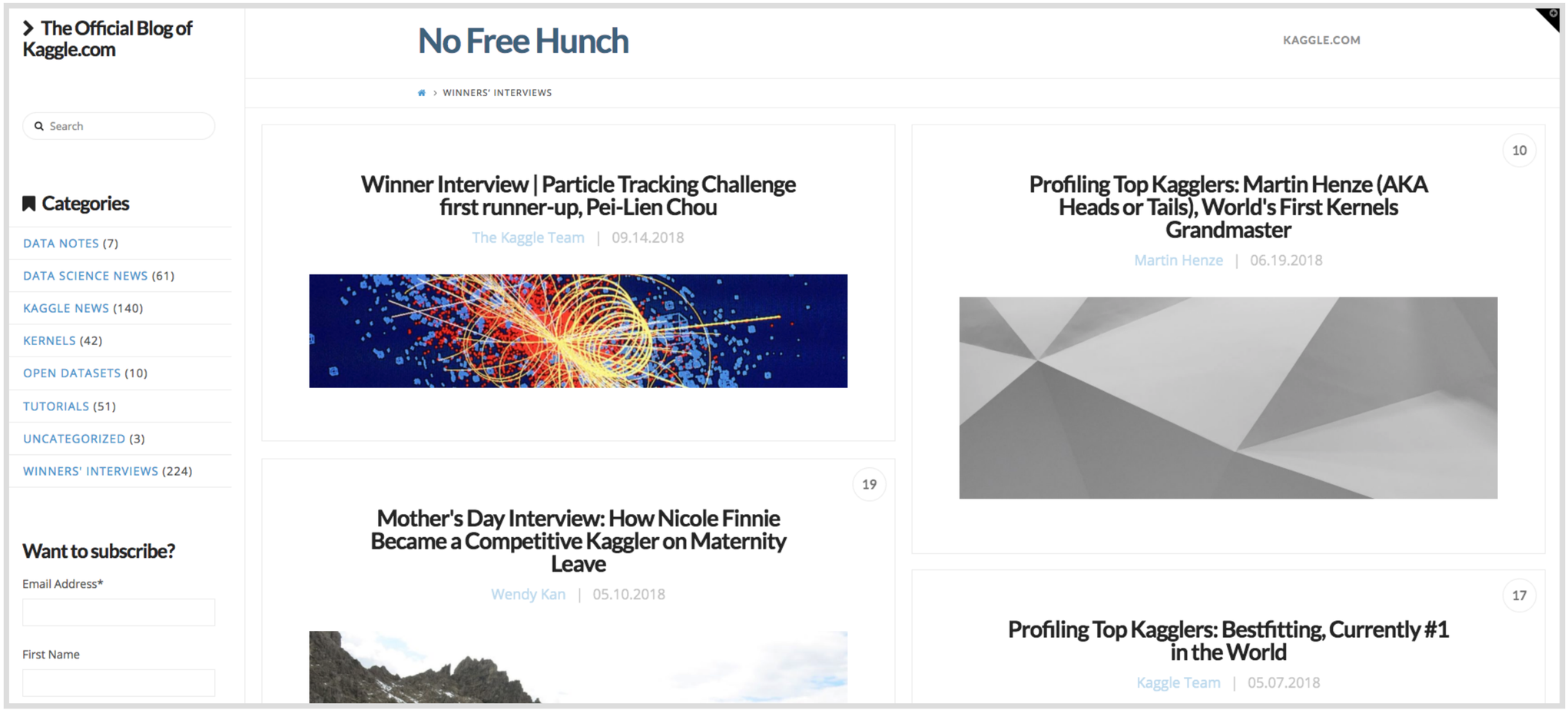
KAGGLE

- Do you know the startup named **Kaggle**?
  - Invested $12.76M in 2 rounds and acquired by Google in 2017.
- A site where numerous machine learning competitions have been held.
  - When the organizer (usually companies like Amazon, Netflix) provides the data (usually real-life dataset),
    the team who best predicts the correct answer with the provided data wins.
- The winners will get a prize or get a chance to join the company depends on the competition.



One Important Rule in Kaggle

- The winners have to disclose how they won.
  - http://blog.kaggle.com/category/winners-interviews/



- **One of the Popular Tools of Winners is XGBoost.**

CONTENTS

Quick Start

- I would like to show you
  how easy and powerful XGBoost is with this Quick start.
- Let's solve a real problem.

- 1. Problem description
- 2. Code (XGBoost Solution)
- 3. Results
- 4. Installation

Problem Description

- Pima Indians Diabetes Prediction
  - Predict the onset of diabetes based on diagnostic measures.
  - https://www.kaggle.com/uciml/pima-indians-diabetes-database
  - Tabular Data : 768 rows x 9 columns
    - 768 people
    - 8 input features and 1 output

- Input features (diagnostic measures) : $X \in R^{768 \times 8}$
  - Pregnancies, glucose, blood pressure, skin thickness, insulin, BMI, diabetes pedigree function, age

- Output : $y \in R^{768 \times 1}$
  - Whether he / she has diabetes (0 or 1).

Code

- https://github.com/JoonyoungYi/KAIST-2018-Fall-CS376-Machine-Learning-Intro-XGBoost/tree/master
  - Total 20 lines.

- The core part of the code.

```
from xgboost import XGBClassifier
model = XGBClassifier()
model.fit(train_X, train_y)
test_y_hat = model.predict(test_X)
```

- **Very simple.** Isn't it?

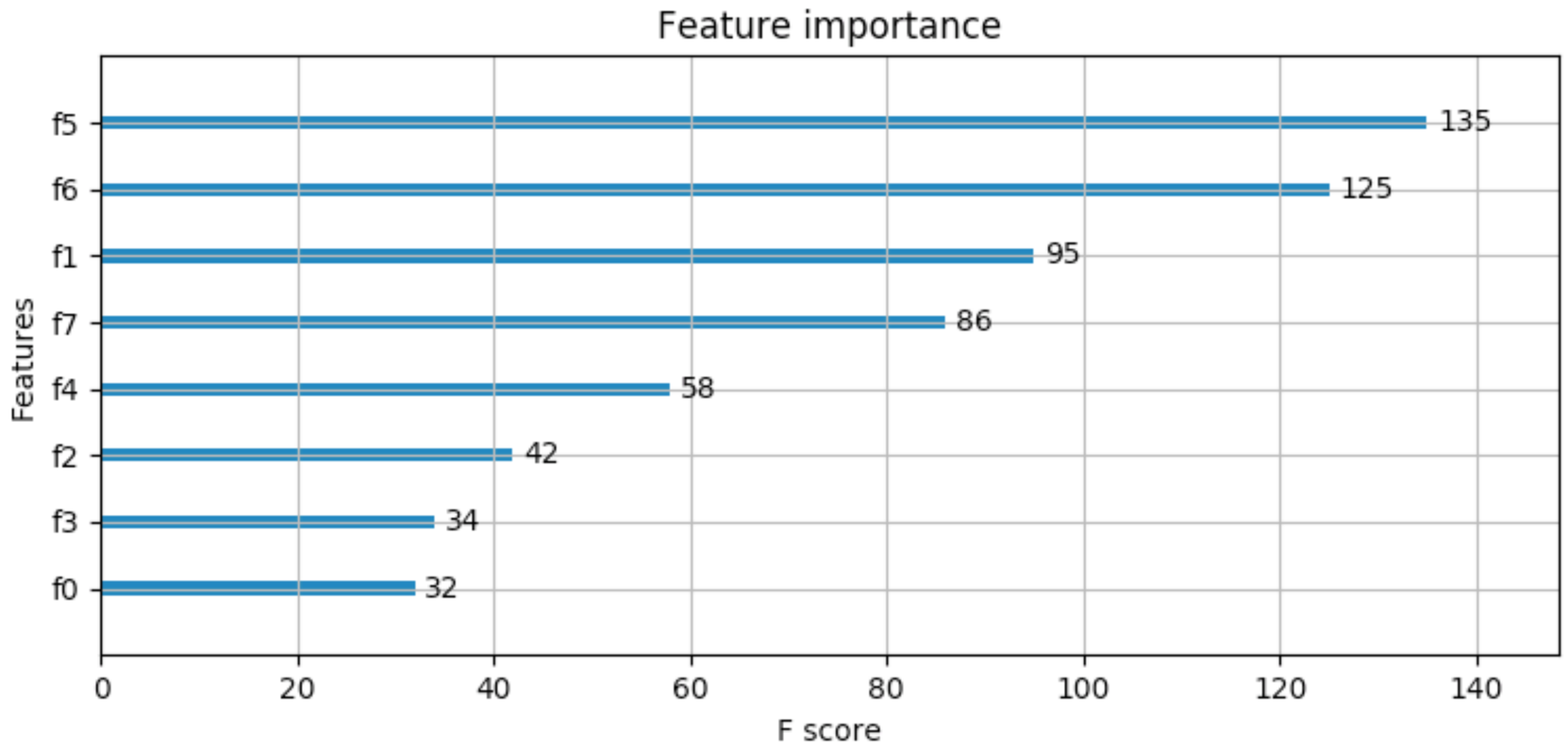
RESULTS

- Train error: 12.2 % / Test error: 19.7 %
  - Quite good performance.
  - Vary depending on the trial.

- Also, this library is really fast.
  - It takes < 10 seconds to fit the model in my lab-top computer.

Results and Further Usage

- XGBoost also tells you how important each feature is.



- 5-th feature is most important and 0-th feature is least important.
  - 5-th feature: BMI / 0-th feature: Pregnancies
- It can be used as a basis when using other models (such as Deep Learning to learn later).

INSTALLATION

- Also, it is easy to install by PIP.
  - https://en.wikipedia.org/wiki/Pip_(package_manager)

- Install commands

```
pip install xgboost
pip install sklearn
```

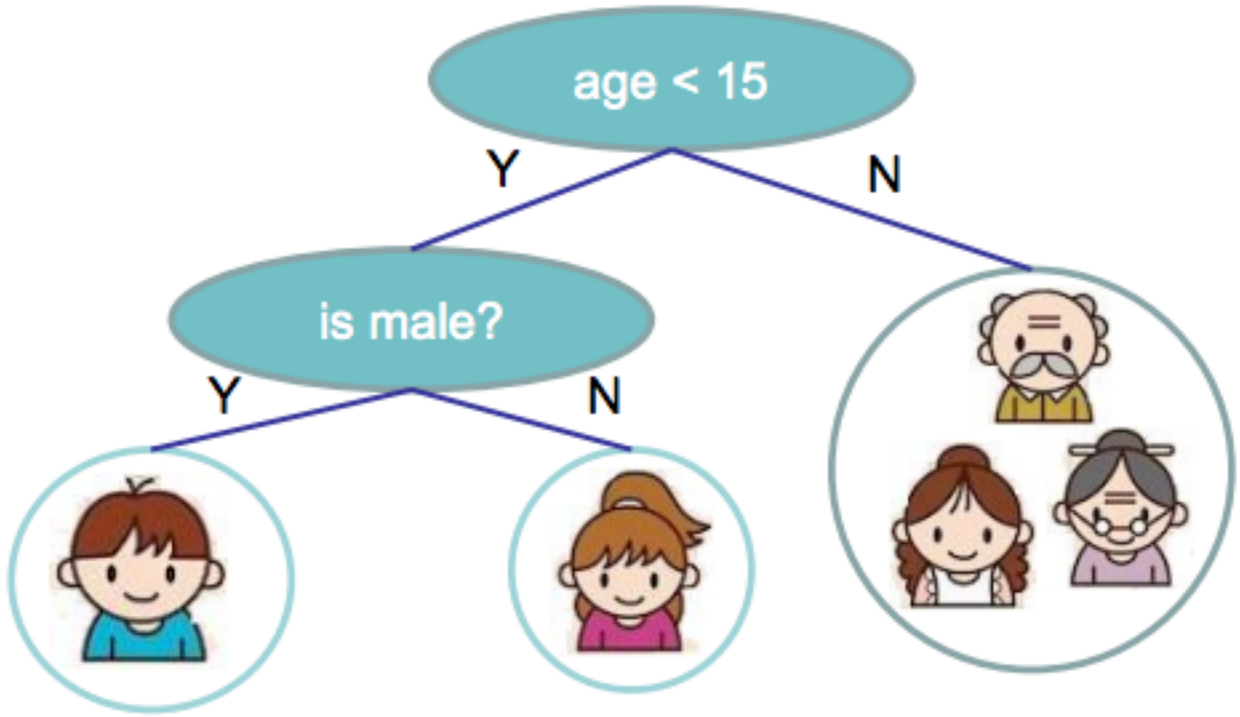
- If your machine needs sudo privileges, you can install it with sudo privileges.
- You can also install it using the python virtualenv (or conda).
  - virtualenv: https://virtualenv.pypa.io/en/stable/

CONTENTS

Decision Tree

- Input: BMI, age, sex, …

- Output: Whether he / she has diabetes



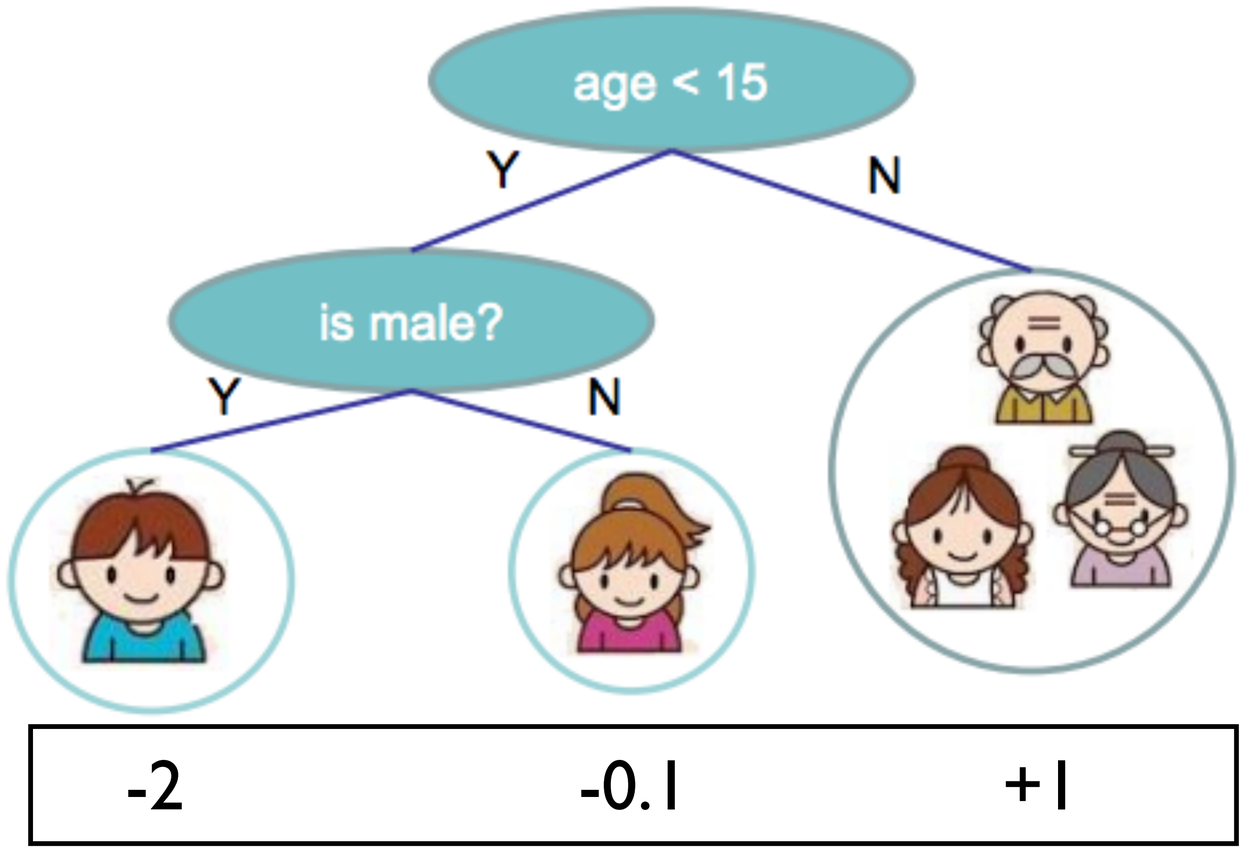
False	True	True

True or False (1 or 0) in each leaf

CART

- Classification and regression tree (CART)
  - Decision rules same as in decision tree.
- Input: BMI, age, sex, …

- Output: Whether he / she has diabetes



prediction score in each leaf

- Contains one score in each leaf value.
- Recall: A function that maps the attributes to the score.

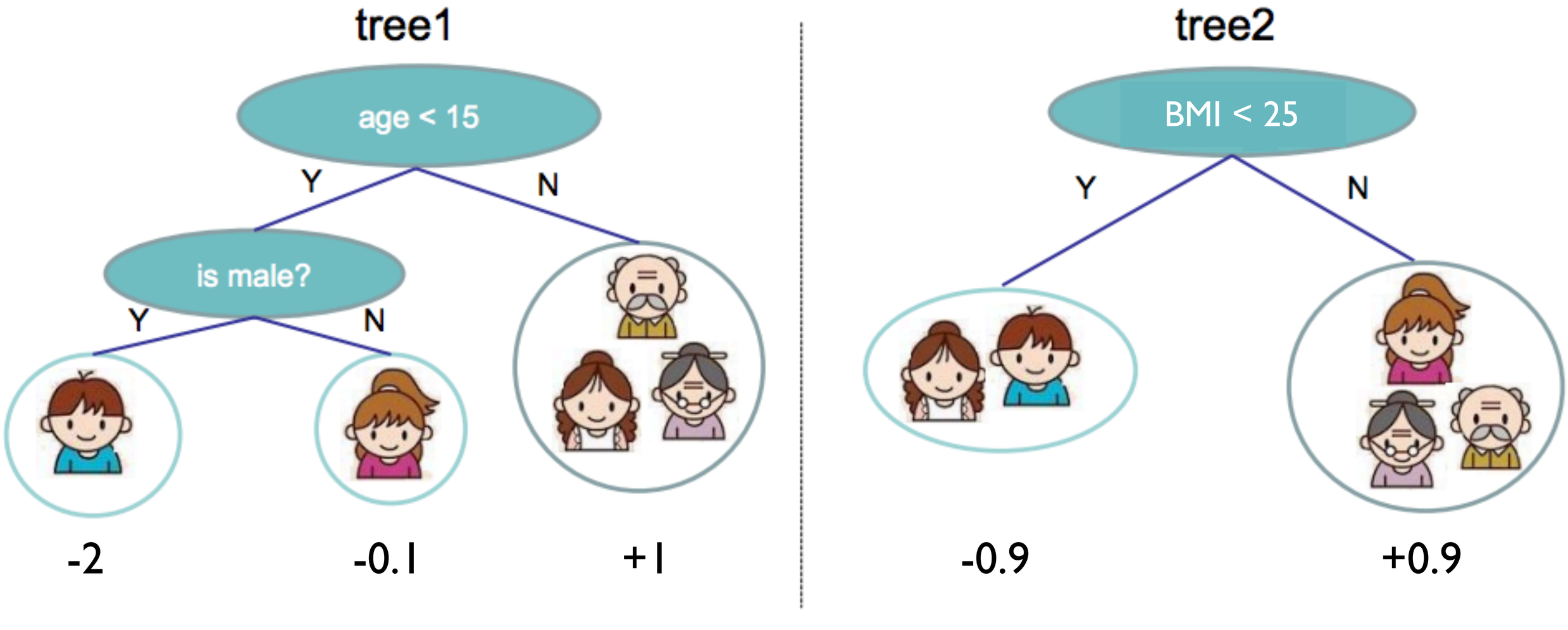
Ensemble Methods

- In wikipedia,
  - Ensemble methods use multiple learning algorithms to obtain better predictive performance than could be obstained from any of the constituent learning algorithms alone.
  - https://en.wikipedia.org/wiki/Ensemble_learning

- Any algorithms that integrate multiple models (algorithms) to better performance.
  - ex. bagging and boosting.

CART Ensemble

- CART makes ensemble more easy.
  - Prediction of is sum of scores predicted by each of the tree.



f() = -2 - 0.9 = -2.9

f() = +1+0.9 = +1.9

Contents

KAIST

MLILAB
Machine Learning & Intelligence

Principle of XGBoost

- Focus on high level concepts.
  - Focus on better using the library.
  - I think detailed algorithm is beyond the scope of this course.
  - If you curious, refer to the paper for detail algorithm.
    - XGBoost: A Scalable Tree Boosting System
      - https://arxiv.org/pdf/1603.02754.pdf

- You might have noticed, XGBoost is a CART ensemble model.

Model and Parameters

- Model: assuming we have K trees.

space of functions containing all regression trees

$$\hat{y} = \sum_{k=1}^{K} f_k(x_i), f_k \in \mathcal{F}$$

  - Recall: regression tree is a function that maps the attributes to the score.

- Parameters
  - Including structures of each tree, and the score in the leaf.
  - Or simply use function as parameters:

$$\Theta = \{f_1, f_2, ..., f_K\}$$

  - Instead learning weights in R^d, we are learning functions (trees).

Objective for Tree Ensembles of XGBoost

- Optimization form:

$$\min \sum_{i=1}^{N} \mathcal{L}(y_i, \hat{y}_i) + \sum_{k=1}^{K} \Omega(f_k)$$

Training loss Complexity of the trees: Regularizer

- What is the possible ways to define Ω ?
  - The number of nodes in the tree, depth.
  - L2 norm of the leaf weights.
  - L1 norm of the lear weights.
  - Think about the role of L1 norm and L2 norm.

- **How do we learn?**

How Do We Learn?

- We can't apply Stochastic Gradient Descent (SGD). **Why?**
  - The variables we should optimize are trees instead of just numerical vectors.

- Solution: **Boosting (Additive Training)**
  - Start from constant prediction, add a new function each time.

$$
\begin{aligned}
\hat{y}_i^{(0)} &= 0 \\
\hat{y}_i^{(1)} &= f_1(x_i) = \hat{y}_i^{(0)} + f_1(x_i) \\
\hat{y}_i^{(2)} &= f_1(x_i) + f_2(x_i) = \hat{y}_i^{(1)} + f_2(x_i) \\
&\dots \\
\hat{y}_i^{(t)} &= \sum_{k=1}^{t} f_k(x_i) = \hat{y}_i^{(t-1)} + f_t(x_i)
\end{aligned}
$$

New function

Model at training round t

Keep functions added in previous round

How Do We Learn?

- As I already mentioned,
  detailed learning algorithm is beyond the scope of this course.

- The XGBoost library will take care of learning instead of you.

CONTENTS

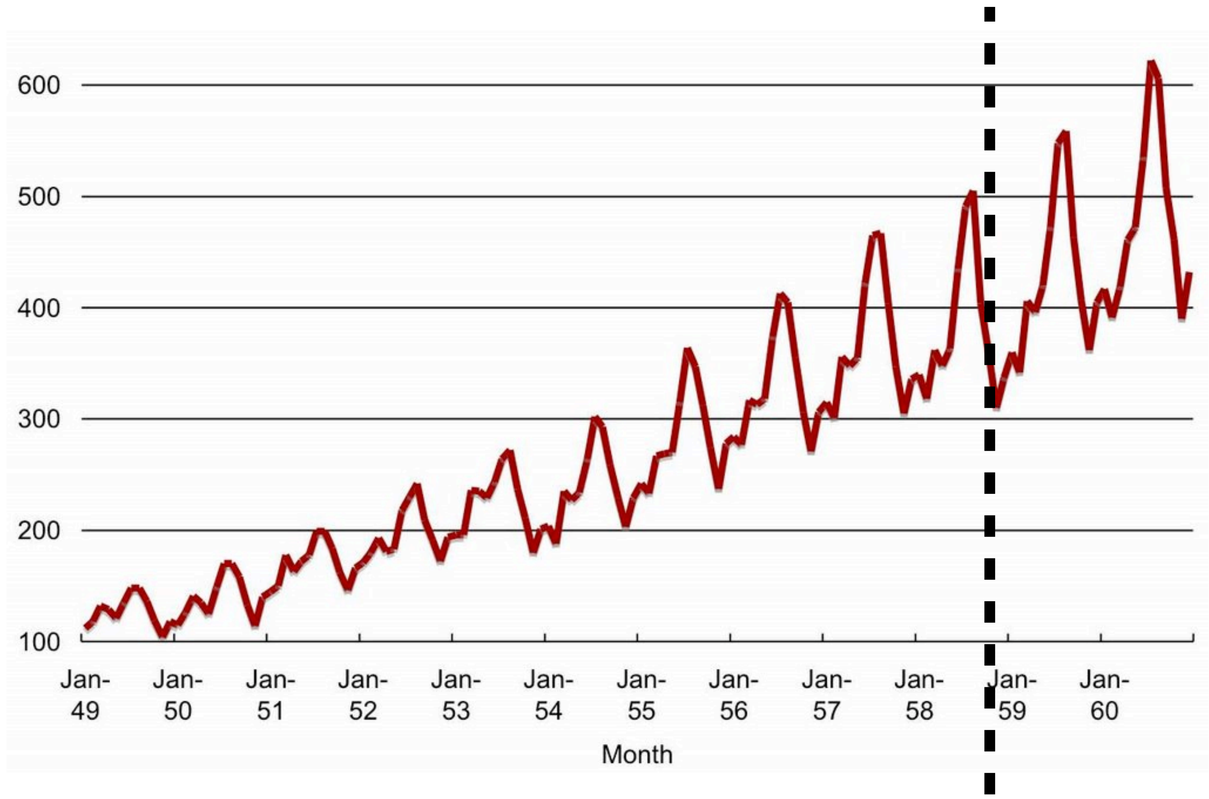
Solving Regression Problem

- The quick start example only shows that the classification problem can be solved by XGBoost, but it can also be used for the regression problem.

```
from xgboost import XGBRegressor
model = XGBRegressor()
model.fit(train_X, train_y)
test_y_hat = model.predict(test_X)
```

LIMITATIONS OF XGBOOST

- Can you guess of the limitations of the XGBoost?
  - 1. Appropriate algorithm for supervised learning.
  - 2. The more complex the data, the more likely it will not work properly.
    - Because, it is based on Decision Tree.
  - 3. Inappropriate for time-series data. **Why?**

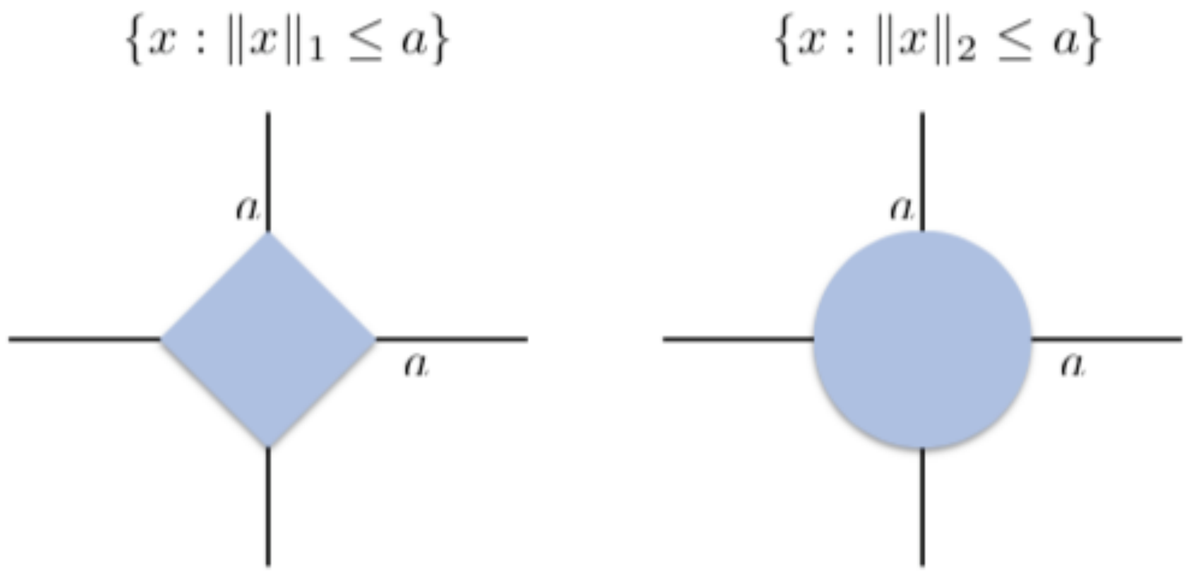


Not randomly splitting
training and test data,
But using historical data as training
and future data as test data.
Since XGBoost is based on
a decision tree,
it will have difficulty in predicting.
Do you have any idea
to handle this issue in XGBoost?

Figure: http://oracledmt.blogspot.com/2006/03/time-series-forecasting-2-single-step.html

ADJUSTING REGULARIZER

- By setting L1 and L2 regularization constants, we can adjust the weights to have a specific trend.



- Including the L1 and L2 regularizers, there are many options we can adjust.
  - https://xgboost.readthedocs.io/en/latest/index.html
  - The document sometimes said what option is proper in some kind of data.
  - Read the documents **carefully**!

Any Questions?

KAIST

REFERENCES

1. https://kaggle.com/
2. http://blog.kaggle.com/category/winners-interviews/
3. https://homes.cs.washington.edu/~tqchen/pdf/BoostedTree.pdf
4. https://brunch.co.kr/@snobberys/137
5. https://www.slideshare.net/rahuldausa/introduction-to-machine-learning-38791937