

학력

학부 : 부산대학교 화공생명공학부 졸업 (2008.03 ~ 2014.02)

석사: KAIST 생명화학공학과 졸업 (2014.03 ~ 2016.02)

박사: KAIST 생명화학공학과 박사과정 재학중 (2016.03 ~ 현재)

실험실: Molecular Simulation Laboratory (Prof. Jihan Kim)

세부 전공: Molecular simulation, computational chemistry,
Machine learning, deep learning

활동

KaggleKorea 페이스북 온라인 그룹(현재 약4,000명) 운영자

대전 캐글스터디(50명), 부산 캐글스터디(40명) 운영자



이유한



YouHan Lee

Ph.D student at KAIST

대전광역시, 대전광역시, 대한민국

Joined 2 years ago · last seen in the past day



Followers 176

Following 29



Competitions
Expert

[Home](#)

[Competitions \(19\)](#)

[Kernels \(32\)](#)

[Discussion \(278\)](#)

[Datasets](#) ...

[Edit Profile](#)

Competitions Expert



Rank

715

of 102,298



0



3



3

[Quick, Draw! Doodle Recog...](#)

🕒 4 months ago · Top 2%

24th

of 1316

[Microsoft Malware Prediction](#)

🕒 7 days ago · Top 2%

40th

of 2426

[Elo Merchant Category Rec...](#)

🕒 22 days ago · Top 3%

86th

of 4129

Kernels Expert



Current Rank

38

of 88,266

Highest Rank

36



4



5



16

[My EDA - I want to see all!](#)

🕒 2 months ago

173

votes

[Simple quant features usin...](#)

🕒 6 months ago

117

votes

[Which encoding is good for...](#)

🕒 6 months ago

77

votes

Discussion Expert



Current Rank

128

of 87,597

Highest Rank

123



3



7



85

[My prize is always what I've...](#)

🕒 5 months ago

18

votes

[중요한 공지입니다. 확인하세...](#)

🕒 2 months ago

17

votes

[Insight or Dodgy](#)

🕒 5 months ago

13

votes

What is Kaggle?

캐글 as a company

- 2010년 설립된 빅데이터 솔루션 대회 플랫폼 회사
- 2017 년 3월에 구글에 인수

kaggle

캐글 as a community

- 현재 200만명의 회원 보유
- Data science, ML, DL 을 주제로 모인 community

kaggle

Competition - Data Race for 데이터 과학자!

기업, 정부기관, 단체, 연구소, 개인

**Dataset
With Prize**

kaggle

**Dataset & Prize
개발 환경(kernel)
커뮤니티(follow, discussion)**

전 세계 데이터 사이언티스트

Dataset - Data Playground for 데이터 과학자!

기업, 정부기관, 단체, 연구소, 개인

**Dataset
With or without Prize**

kaggle

**Dataset & Prize
개발 환경(kernel)
커뮤니티(follow, discussion)**

전 세계 데이터 사이언티스트

Dataset

2018/11/02 기준 캐글에 등록되어 있고,
다운받을 수 있는 데이터셋 숫자는
11,883 개

Kernel!

- 캐글에서 제공하는 가상환경.
 - 컴퓨터 수십, 수백 대 제공해 줍니다.
 - With GPU!!!
- 검증된 캐글러들이 자신이 분석한 것을 공유합니다.
 - 좋은 reference, 공부 자료!

Why do kaggle?

머신러닝으로 할 수 있는 대부분의 문제 유형을 담고 있는 컴퍼티션들

지금까지 302개의
competition 이 치뤄짐.



머신러닝으로 풀 수 있는
대부분의 문제가 담겨있다

개인적 측면

- 실력
 - ML certificate
 - Portfolio
 - 경험

이유한 씨는
ML, DL, DS 에 대한
경험이 있으신가요?

개인적 측면 – 캐글 프로필 관리

넵

See this

YouHan Lee
Ph.D student at KAIST
대전광역시, 대전광역시, 대한민국
Joined 2 years ago · last seen in the past day

Followers 176
Following 29

Competitions Expert

Home Competitions (19) Kernels (32) Discussion (278) Datasets ... Edit Profile

Competitions Expert	
Rank	715 of 102,298
0	3
Quick, Draw! Doodle Recog... 4 months ago · Top 2%	24 th of 1316
Microsoft Malware Prediction 7 days ago · Top 2%	40 th of 2426
Elo Merchant Category Rec... 22 days ago · Top 3%	86 th of 4129

Kernels Expert	
Current Rank	Highest Rank
38 of 88,266	36
4	5
My EDA - I want to see all! 2 months ago	173 votes
Simple quant features usin... 6 months ago	117 votes
Which encoding is good for... 6 months ago	77 votes

Discussion Expert	
Current Rank	Highest Rank
128 of 87,597	123
3	7
My prize is always what I've... 5 months ago	18 votes
중요한 공지입니다. 확인하세... 2 months ago	17 votes
Insight or Dodgy 5 months ago	13 votes

여러 사례들

Kaggle (company) Contests and Competitions +2 

Does participating in Kaggle competitions open doors in machine learning jobs?

 Answer  Follow 163  Request      

- 답변: 캐글 자체가 job을 주지 않는다. 하지만 캐글을 해서, ML, DL, DS 실력을 엄청 쌓아서, 그것이 job 을 얻게한다.

여러 사례들



- Porto competition 을 진행
- 이 컴퍼티션을 통해, 한 그룹의 DS team 의 chief 와 연결 됨.
- 입사!!!

개인적 측면 – 경험 실력 FOR 정형 데이터

Porto: 고객이 내년에 자동차 보험금 청구를 할 것인가?

Home Credit: 고객이 앞으로 대출 상환을 할 것인가?

Costa rican: 고객의 소득 수준을 ML 로 구분하라

Elo: 거래 내역 데이터를 가지고, 고객 충성도를 예측하라

New York taxi: Taxi 탑승 시간을 예측하라

직방: 아파트 거래가격 예측하라

INFOCARE: 아파트 경매가격 예측해라

개인적 측면 – 경험 실력 FOR 정형 데이터

- Exploratory data analysis
 - Data visualization
 - Matplotlib, Seaborn, Plotly
 - Data mining
 - Pandas, numpy
- Feature engineering
 - Time series features
 - Categorical features
 - Numerical features
 - Aggregation features
 - Ratio features
 - Product features
- Data preparation
 - Data augmentation (imbalance)
 - Upsampling
 - Downsampling
 - SMOTE
- Model development
 - Sklearn
 - Linear model
 - Non-linear model
 - Tree-model
 - Not sklearn
 - Xgboost
 - Lightgbm
 - Catboost
 - LibFFM

개인적 측면 – 경험 실력 FOR 딥러닝

- Model evaluation
 - Various metrics
 - Accuracy
 - Precision
 - Recall
 - F1-score
 - Etc.
- Other technique
 - Machine learning pipeline
 - My pipeline code
 - Feature management

정형 데이터 위해 학습한 모델만,,,
천단위 이상으로 만들어 봤을 겁니다.

개인적 측면 – 경험 실력 FOR 딥러닝

Tensorflow: 30개 단어를 구분하는 AI 만들어라

Quora: 성실한, 불성실한 질문을 구분해내라

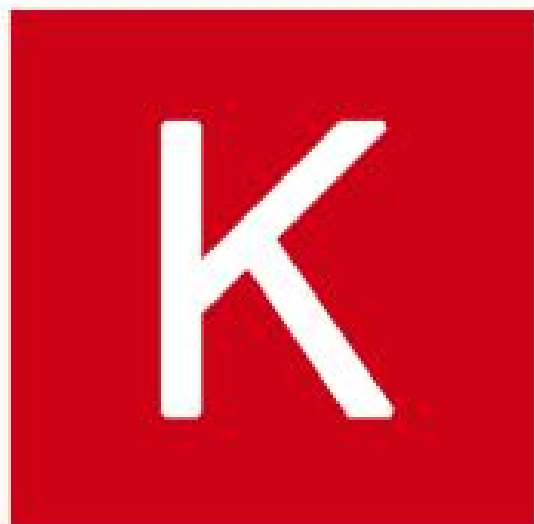
Doodle: 340개의 클래스 별 낙서를 ANN 으로 구분하라.

Protein: 28개의 클래스 별 Protein 을 ANN 으로 구분하라.

Airbus: 바다 위 배를 찍은 위성 사진에서 배의 위치를 찾아내라

Statoil: 바다 위 빙산과, 배를 구분하라

Keras: The Python Deep Learning library



Keras

개인적 측면 – 경험 실력 FOR 딥러닝

- Model Development
 - Not pretrained
 - CNN
 - RNN
 - Simese network
 - Pretrained
 - Fine-tunning
- Learning technique
 - Cyclic learning
 - Generator
 - Data augmentation

딥러닝 학습한 구조만,
몇백개 이상 만든 거 같네요.

My prize is always what I have learned

여지껏 배운 것이
언제나
저의 prize 입니다.

생생한 캐글 대회 후기

Home Credit Default Risk competition - Overview

A banner for the Home Credit Default Risk competition. It features a background image of a large stack of US dollar bills. In the top left corner, there is a small icon of a trophy and the text "Featured Prediction Competition". The main title "Home Credit Default Risk" is in a large, bold, white font. Below it, a subtitle asks "Can you predict how capable each applicant is of repaying a loan?". In the bottom left corner, there is a red Home Credit logo and the text "Home Credit Group · 7,198 teams · 24 days ago". In the top right corner, the prize money "\$70,000" is displayed in a large, bold, white font, with "Prize Money" written below it in a smaller font.

Featured Prediction Competition

Home Credit Default Risk

Can you predict how capable each applicant is of repaying a loan?

Home Credit Group · 7,198 teams · 24 days ago

\$70,000
Prize Money

Many people struggle to get loans due to insufficient or non-existent credit histories. And, unfortunately, this population is often taken advantage of by untrustworthy lenders.

Home Credit strives to broaden financial inclusion for the unbanked population by providing a positive and safe borrowing experience. In order to make sure this underserved population has a positive loan experience, Home Credit makes use of a variety of alternative data--including telco and transactional information--to predict their clients' repayment abilities.

While Home Credit is currently using various statistical and machine learning methods to make these predictions, they're challenging Kagglers to help them unlock the full potential of their data. Doing so will ensure that clients capable of repayment are not rejected and that loans are given with a principal, maturity, and repayment calendar that will empower their clients to be successful.



Home Credit Default Risk competition - Evaluation

Submissions are evaluated on [area under the ROC curve](#) between the predicted probability and the observed target.

Submission File

For each `SK_ID_CURR` in the test set, you must predict a probability for the `TARGET` variable. The file should contain a header and have the following format:

```
SK_ID_CURR, TARGET
100001, 0.1
100005, 0.9
100013, 0.2
etc.
```

Home Credit Default Risk competition - Prize

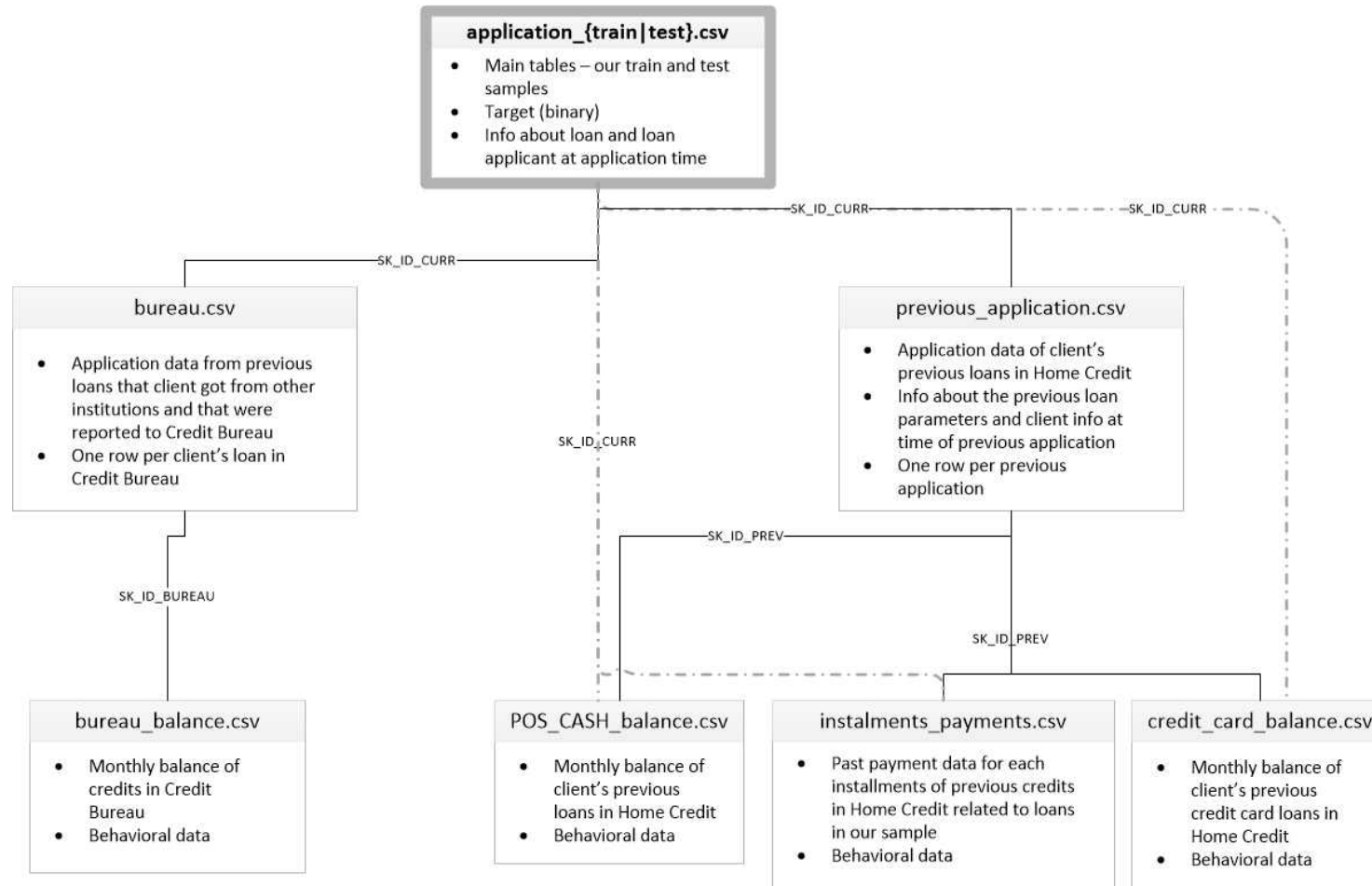
- 1st Place - \$ 35,000
- 2nd Place - \$ 25,000
- 3rd Place - \$ 10,000

Home Credit Default Risk competition - Timeline

- **August 22, 2018** - Entry deadline. You must accept the competition rules before this date in order to compete.
- **August 22, 2018** - Team Merger deadline. This is the last day participants may join or merge teams.
- **August 29, 2018** - Final submission deadline.

All deadlines are at 11:59 PM UTC on the corresponding day unless otherwise noted. The competition organizers reserve the right to update the contest timeline if they deem it necessary.

Home Credit Default Risk competition - Data



무엇부터
해야 할까요?

무엇부터 해야 할까요?

데이터 분석

알고리즘 선택

머신 러닝 ?

Pattern recognition

머신 러닝 ?

Make **general function(conditions)**
to obtain goal(minimize loss)

머신 러닝 ?

Learn **statistics(correlations)** between
feature vs feature/
feature vs target

Pattern, correlation 이 있어서 AI 가 찾았다

AI 가 Pattern, correlation을 만들어냈다

그렇다면
데이터 사이언스,
머신, 딥러닝에서
가장 중요한 것은?

DATA

Data

Data

Dataaaaaaaaaaaaaa!!!!!!

Your neural network is only as good as the data you feed it.

당신의 모델은,
당신이 input으로 준 것 만큼 좋다!

Garbage in, Garbage out!

<https://medium.com/nanonets/how-to-use-deep-learning-when-you-have-limited-data-part-2-data-augmentation-c26971dc8ced>

무엇부터 해야 할까요?

데이터 분석
EDA
(Exploratory data analysis)

Home Credit Default Risk competition – Data description

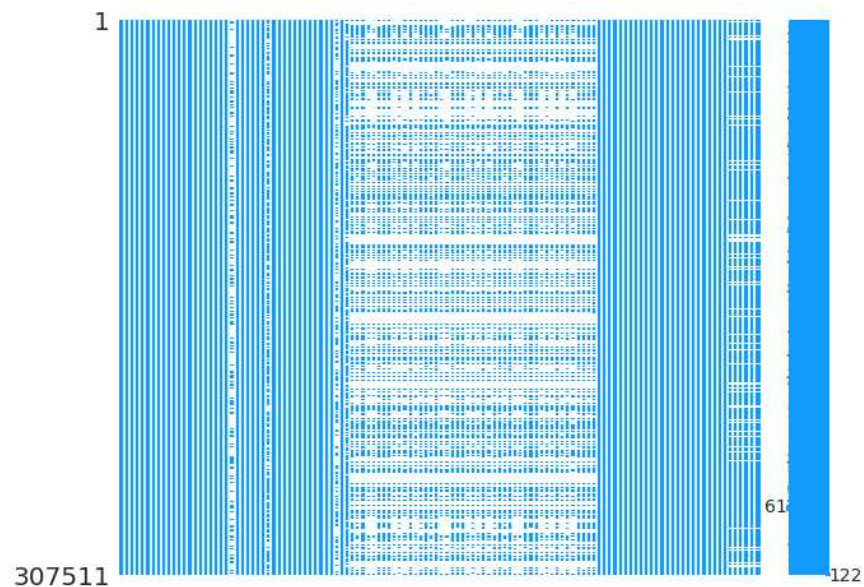
- bureau_balance.csv
 - Monthly balances of previous credits in Credit Bureau.
 - This table has one row for each month of history of every previous credit reported to Credit Bureau – i.e the table has (#loans in sample * # of relative previous credits * # of months where we have some history observable for the previous credits) rows.
- POS_CASH_balance.csv
 - Monthly balance snapshots of previous POS (point of sales) and cash loans that the applicant had with Home Credit.
 - This table has one row for each month of history of every previous credit in Home Credit (consumer credit and cash loans) related to loans in our sample – i.e. the table has (#loans in sample * # of relative previous credits * # of months in which we have some history observable for the previous credits) rows.
- credit_card_balance.csv
 - Monthly balance snapshots of previous credit cards that the applicant has with Home Credit.
 - This table has one row for each month of history of every previous credit in Home Credit (consumer credit and cash loans) related to loans in our sample – i.e. the table has (#loans in sample * # of relative previous credit cards * # of months where we have some history observable for the previous credit card) rows.
- previous_application.csv
 - All previous applications for Home Credit loans of clients who have loans in our sample.
 - There is one row for each previous application related to loans in our data sample.
- installments_payments.csv
 - Repayment history for the previously disbursed credits in Home Credit related to the loans in our sample.
 - There is a) one row for every payment that was made plus b) one row each for missed payment.
 - One row is equivalent to one payment of one installment OR one installment corresponding to one payment of one previous Home Credit credit related to loans in our sample.

1. Data check – Feature check

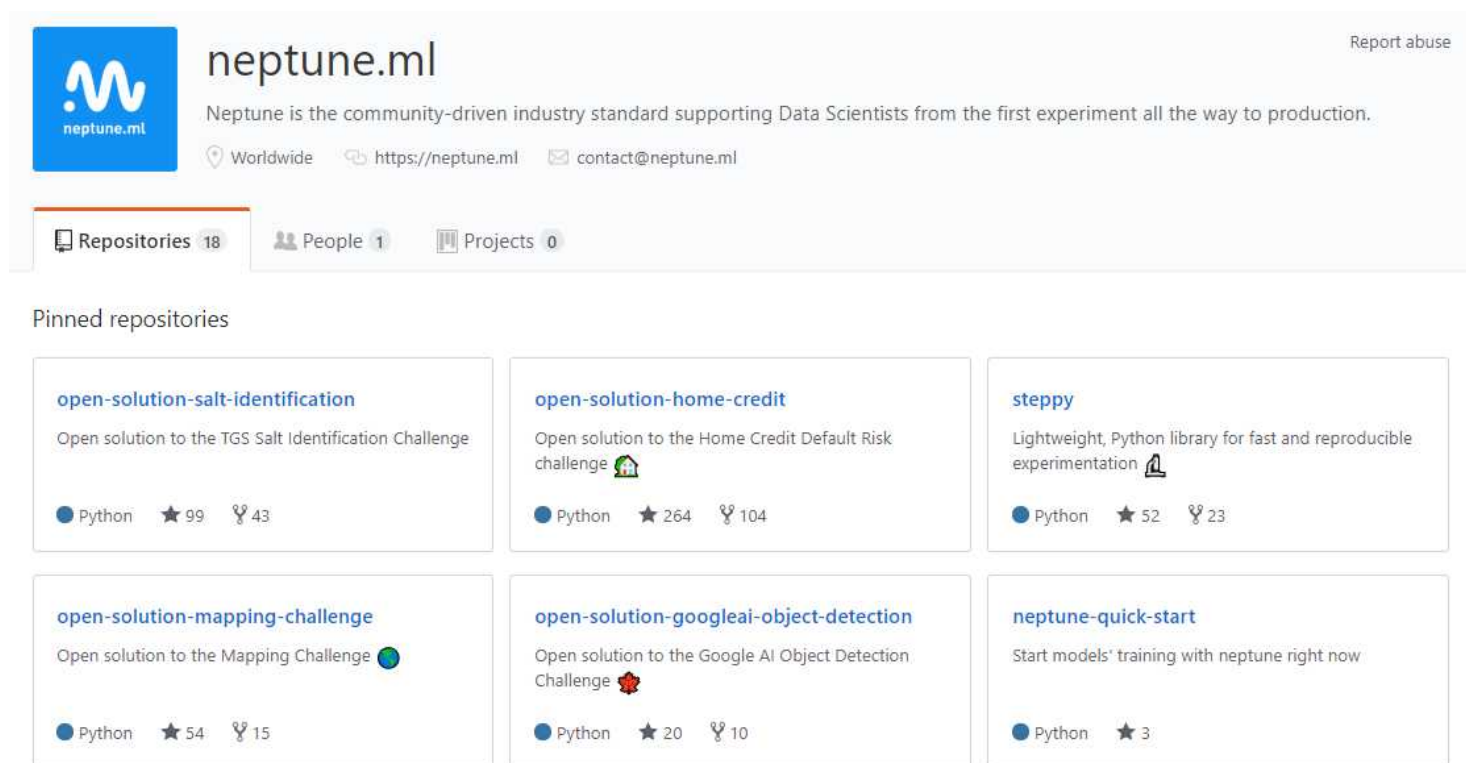
	role	level	dtype	response_rate
varname				
SK_ID_CURR	id	ordinal	int64	100.000000
TARGET	target	ordinal	int64	100.000000
NAME_CONTRACT_TYPE	input	categorical	object	100.000000
CODE_GENDER	input	categorical	object	100.000000
FLAG_OWN_CAR	input	categorical	object	100.000000
FLAG_OWN_REALTY	input	categorical	object	100.000000
CNT_CHILDREN	input	ordinal	int64	100.000000
AMT_INCOME_TOTAL	input	interval	float64	100.000000
AMT_CREDIT	input	interval	float64	100.000000
AMT_ANNUITY	input	interval	float64	99.996098
AMT_GOODS_PRICE	input	interval	float64	99.909597
NAME_TYPE_SUITE	input	categorical	object	99.579852
NAME_INCOME_TYPE	input	categorical	object	100.000000
NAME_EDUCATION_TYPE	input	categorical	object	100.000000
NAME_FAMILY_STATUS	input	categorical	object	100.000000
NAME_HOUSING_TYPE	input	categorical	object	100.000000
REGION_POPULATION_RELATIVE	input	interval	float64	100.000000
DAYS_BIRTH	input	ordinal	int64	100.000000
DAYS_EMPLOYED	input	ordinal	int64	100.000000
DAYS_REGISTRATION	input	interval	float64	100.000000
DAYS_ID_PUBLISH	input	ordinal	int64	100.000000
OWN_CAR_AGE	input	interval	float64	34.009190
FLAG_MOBIL	input	ordinal	int64	100.000000
FLAG_EMP_PHONE	input	ordinal	int64	100.000000
FLAG_WORK_PHONE	input	ordinal	int64	100.000000
FLAG_CONT_MOBILE	input	ordinal	int64	100.000000
FLAG_PHONE	input	ordinal	int64	100.000000
FLAG_EMAIL	input	ordinal	int64	100.000000
OCCUPATION_TYPE	input	categorical	object	88.654455
CNT_FAM_MEMBERS	input	interval	float64	99.999350

1. Data check – Null data check

	Total	Percent
COMMONAREA_MEDI	214865	69.872297
COMMONAREA_AVG	214865	69.872297
COMMONAREA_MODE	214865	69.872297
NONLIVINGAPARTMENTS_MODE	213514	69.432963
NONLIVINGAPARTMENTS_MEDI	213514	69.432963
NONLIVINGAPARTMENTS_AVG	213514	69.432963
FONDKAPREMONT_MODE	210295	68.386172
LIVINGAPARTMENTS_MEDI	210199	68.354953
LIVINGAPARTMENTS_MODE	210199	68.354953
LIVINGAPARTMENTS_AVG	210199	68.354953
FLOORSMIN_MEDI	208642	67.848630
FLOORSMIN_MODE	208642	67.848630
FLOORSMIN_AVG	208642	67.848630
YEARS_BUILD_MEDI	204488	66.497784
YEARS_BUILD_AVG	204488	66.497784
YEARS_BUILD_MODE	204488	66.497784
OWN_CAR_AGE	202929	65.990810
LANDAREA_MODE	182590	59.376738
LANDAREA_AVG	182590	59.376738
LANDAREA_MEDI	182590	59.376738



1. Data check – Outlier check



The screenshot displays the GitHub profile for **neptune.ml**. The profile header includes the repository icon, the name **neptune.ml**, a description: "Neptune is the community-driven industry standard supporting Data Scientists from the first experiment all the way to production.", location "Worldwide", website "https://neptune.ml", and email "contact@neptune.ml". Below the header, statistics show 18 repositories, 1 person, and 0 projects. The "Pinned repositories" section lists six repositories:

Repository Name	Description	Language	Stars	Forks
open-solution-salt-identification	Open solution to the TGS Salt Identification Challenge	Python	99	43
open-solution-home-credit	Open solution to the Home Credit Default Risk challenge	Python	264	104
steppy	Lightweight, Python library for fast and reproducible experimentation	Python	52	23
open-solution-mapping-challenge	Open solution to the Mapping Challenge	Python	54	15
open-solution-googleai-object-detection	Open solution to the Google AI Object Detection Challenge	Python	20	10
neptune-quick-start	Start models' training with neptune right now	Python	3	0

1. Data check – Outlier check

```
X['CODE_GENDER'].replace('XNA', np.nan, inplace=True)
X['DAYS_EMPLOYED'].replace(365243, np.nan, inplace=True)
X['NAME_FAMILY_STATUS'].replace('Unknown', np.nan, inplace=True)
X['ORGANIZATION_TYPE'].replace('XNA', np.nan, inplace=True)

bureau['AMT_CREDIT_SUM'].fillna(self.fill_value, inplace=True)
bureau['AMT_CREDIT_SUM_DEBT'].fillna(self.fill_value, inplace=True)
bureau['AMT_CREDIT_SUM_OVERDUE'].fillna(self.fill_value, inplace=True)
bureau['CNT_CREDIT_PROLONG'].fillna(self.fill_value, inplace=True)
```

2. Feature engineering – Ratio features

Ratio feature : A per B

```
X['annuity_income_percentage'] = X['AMT_ANNUITY'] / X['AMT_INCOME_TOTAL']
X['car_to_birth_ratio'] = X['OWN_CAR_AGE'] / X['DAYS_BIRTH']
X['car_to_employ_ratio'] = X['OWN_CAR_AGE'] / X['DAYS_EMPLOYED']
```

- 연금 / 전체 수입
- 차 소유한 나이 / 총 생애 일수
- 차 소유하 나이 / 총 근무 일수

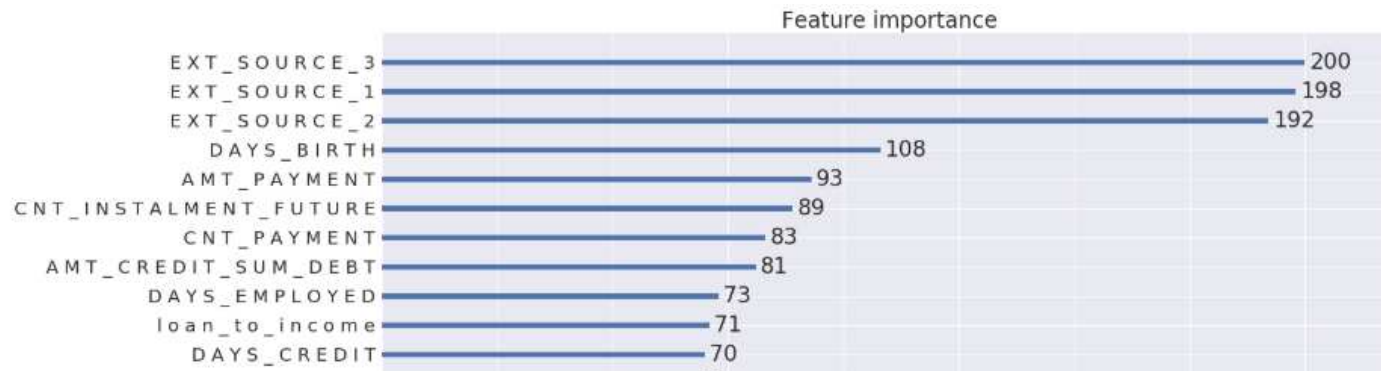
```
X['annuity_income_percentage'] = X['AMT_ANNUITY'] / X['AMT_INCOME_TOTAL']
X['car_to_birth_ratio'] = X['OWN_CAR_AGE'] / X['DAYS_BIRTH']
X['car_to_employ_ratio'] = X['OWN_CAR_AGE'] / X['DAYS_EMPLOYED']
X['children_ratio'] = X['CNT_CHILDREN'] / X['CNT_FAM_MEMBERS']
X['credit_to_annuity_ratio'] = X['AMT_CREDIT'] / X['AMT_ANNUITY']
X['credit_to_goods_ratio'] = X['AMT_CREDIT'] / X['AMT_GOODS_PRICE']
X['credit_to_income_ratio'] = X['AMT_CREDIT'] / X['AMT_INCOME_TOTAL']
X['days_employed_percentage'] = X['DAYS_EMPLOYED'] / X['DAYS_BIRTH']
X['income_credit_percentage'] = X['AMT_INCOME_TOTAL'] / X['AMT_CREDIT']
X['income_per_child'] = X['AMT_INCOME_TOTAL'] / (1 + X['CNT_CHILDREN'])
X['income_per_person'] = X['AMT_INCOME_TOTAL'] / X['CNT_FAM_MEMBERS']
X['payment_rate'] = X['AMT_ANNUITY'] / X['AMT_CREDIT']
X['phone_to_birth_ratio'] = X['DAYS_LAST_PHONE_CHANGE'] / X['DAYS_BIRTH']
X['phone_to_employ_ratio'] = X['DAYS_LAST_PHONE_CHANGE'] / X['DAYS_EMPLOYED']
X['external_sources_weighted'] = X.EXT_SOURCE_1 * 2 + X.EXT_SOURCE_2 * 3 + X.EXT_SOURCE_3 * 4
X['cnt_non_child'] = X['CNT_FAM_MEMBERS'] - X['CNT_CHILDREN']
X['child_to_non_child_ratio'] = X['CNT_CHILDREN'] / X['cnt_non_child']
X['income_per_non_child'] = X['AMT_INCOME_TOTAL'] / X['cnt_non_child']
X['credit_per_person'] = X['AMT_CREDIT'] / X['CNT_FAM_MEMBERS']
X['credit_per_child'] = X['AMT_CREDIT'] / (1 + X['CNT_CHILDREN'])
X['credit_per_non_child'] = X['AMT_CREDIT'] / X['cnt_non_child']
for function_name in ['min', 'max', 'sum', 'mean', 'nanmedian']:
    X['external_sources_{}'.format(function_name)] = eval('np.{}'.format(function_name))(
        X[['EXT_SOURCE_1', 'EXT_SOURCE_2', 'EXT_SOURCE_3']], axis=1)

X['short_employment'] = (X['DAYS_EMPLOYED'] < -2000).astype(int)
X['young_age'] = (X['DAYS_BIRTH'] < -14000).astype(int)
```

2. Feature engineering – Product features

Product feature : A x B

- Feature importance 를 봤을 때, 상위 feature 들 중 numerical feature 끼리 곱하여 추가 함.



```
df['EXT_SOURCE_1_x_EXT_SOURCE_2'] = df['EXT_SOURCE_1'] * df['EXT_SOURCE_2']  
df['EXT_SOURCE_1_x_EXT_SOURCE_3'] = df['EXT_SOURCE_1'] * df['EXT_SOURCE_3']  
df['EXT_SOURCE_2_x_EXT_SOURCE_3'] = df['EXT_SOURCE_2'] * df['EXT_SOURCE_3']  
df['AMT_PAYMENT_x_EXT_SOURCE_3'] = df['AMT_PAYMENT'] * df['EXT_SOURCE_3']
```


2. Feature engineering – Addition or Subtraction features

Addition feature : $A + B$

Subtraction feature : $A - B$

- 중요한 feature 끼리 더하거나 빼서 새로운 feature 생성 .

```
# External sources
X['external_sources_weighted'] = X.EXT_SOURCE_1 * 2 + X.EXT_SOURCE_2 * 3 + X.EXT_SOURCE_3 * 4
for function_name in ['min', 'max', 'sum', 'mean', 'nanmedian']:
    X['external_sources_{}'.format(function_name)] = eval('np.{}'.format(function_name))(
        X[['EXT_SOURCE_1', 'EXT_SOURCE_2', 'EXT_SOURCE_3']], axis=1)
```

2. Feature engineering – New categorical features

특정 기준을 만족하느냐,
만족하지 않느냐로
Binary category 를 만들 수 있음.

```
bureau['bureau_credit_active_binary'] = (bureau['CREDIT_ACTIVE'] != 'Closed').astype(int)  
bureau['bureau_credit_enddate_binary'] = (bureau['DAYS_CREDIT_ENDDATE'] > 0).astype(int)
```


2. Feature engineering – Aggregation features

Category 와 numerical feature 의 조합으로 생성하며,
Category 각 그룹당 mean, median, variance, standard
deviation 을 feature 로 사용

```
group_object = credit_card.groupby(by=['SK_ID_CURR'])['AMT_DRAWINGS_ATM_CURRENT'].agg('sum').reset_index()
```

말그대로 조합이기 때문에,
엄청나게 만들어 낼 수 있음.

$$C(n, r) = \frac{n!}{r! (n - r)!}$$

2. Feature engineering – Categorical features

One-hot
encoding

Category 개수 만큼 column 이 늘어난다.
너무 오래 걸린다.

Label
encoding

자칫 bias ordering 문제가 생길 수 있다.

Lightgbm
Built-in

데이터셋이 크니 학습이 빠른 Lightgbm 을
쓰건데, 마침 카테고리를 처리하는 내장
알고리즘이 있구나! 이거다

2. Feature engineering – Fill missing values and infinite values

Numerical features

Lightgbm 은 missing value 를 빼고 tree split 을 한 다음, missing value 를 각 side 에 넣어봐서 loss 가 줄어드는 쪽으로 missing value 를 분류

- 0 로 채우지 말고, 내장 알고리즘 쓰기로 결정.
- (+) Infinite value 는 $1.2 * \text{max value}$
- (-) Infinite value 는 $1.2 * \text{min value}$

Categorical features

‘NAN’ 이라는 새로운 category 를 만들어서 채움

3. Feature selection – Use various approaches

이리저리 만들다 보니 약 2300개 features 생성됨.
이걸 다 쓸 것인가?

Feature importance, target과의
correlation 으로 거르자

-> 1000개 정도 추려냄.

4. Model development – Use LGBM

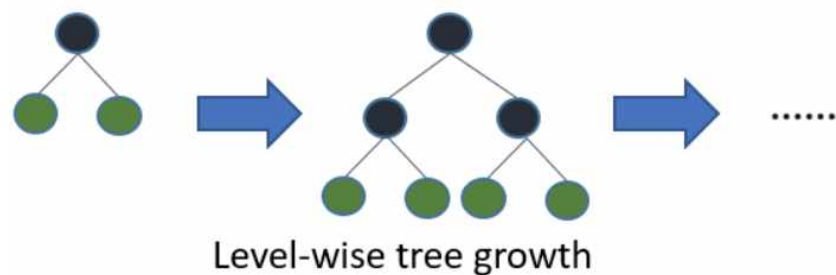
Why lightgbm?

- Faster training speed and higher efficiency
- Lower memory usage
- Better accuracy
- Parallel and GPU learning supported
- Capable of handling large-scale data

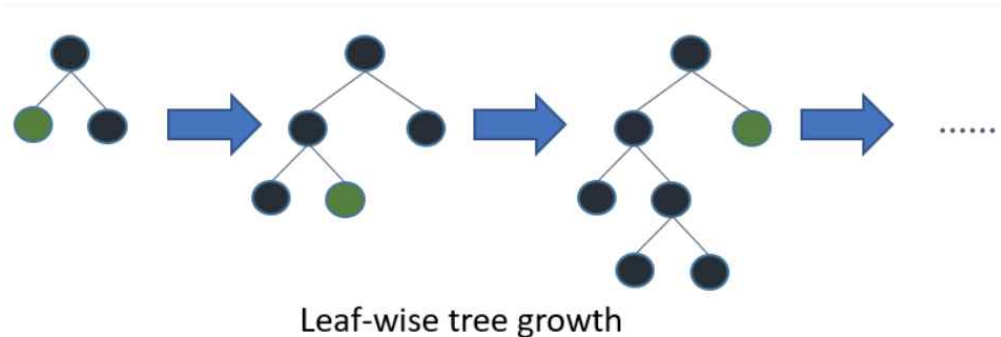
4. Model development – Use LGBM

What is different?

Most decision tree algorithm

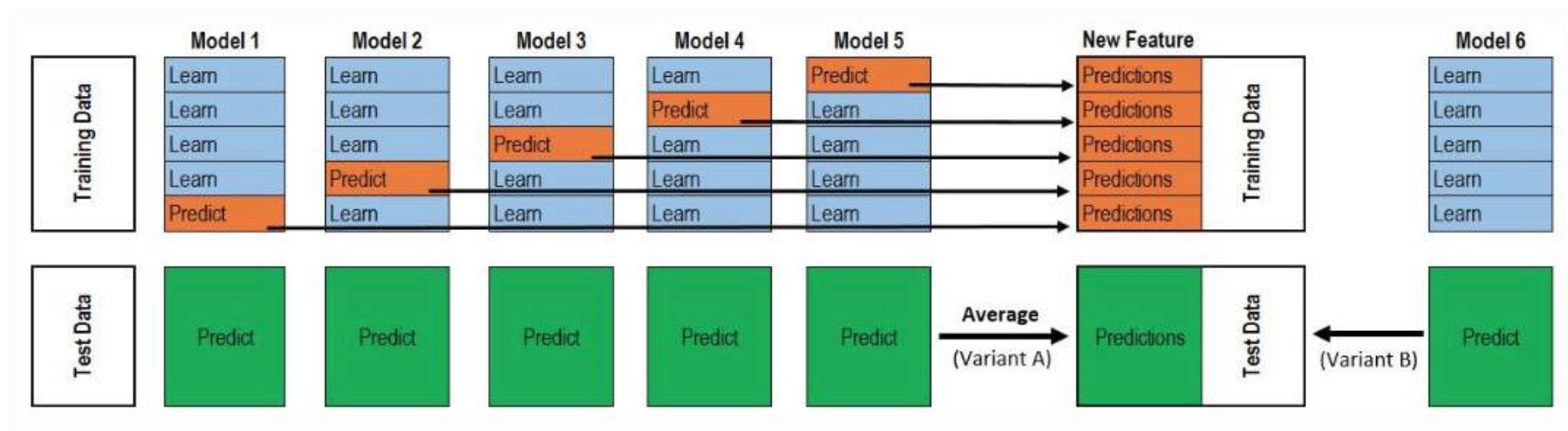


Lightgbm



종고 빠르다 쓰자

5. Training strategy – Ensemble is always answer

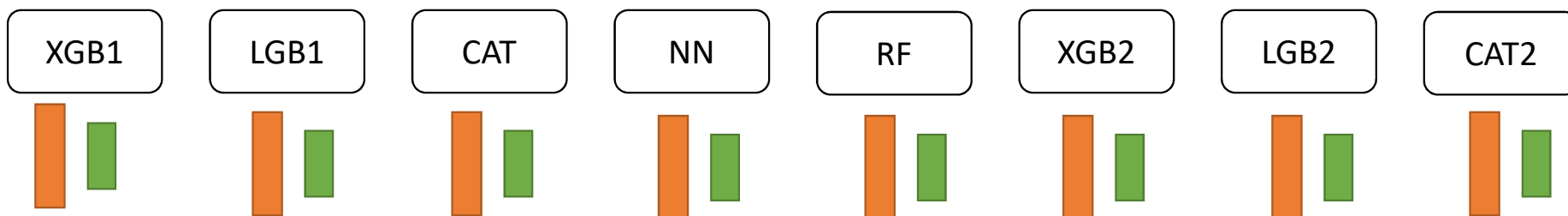
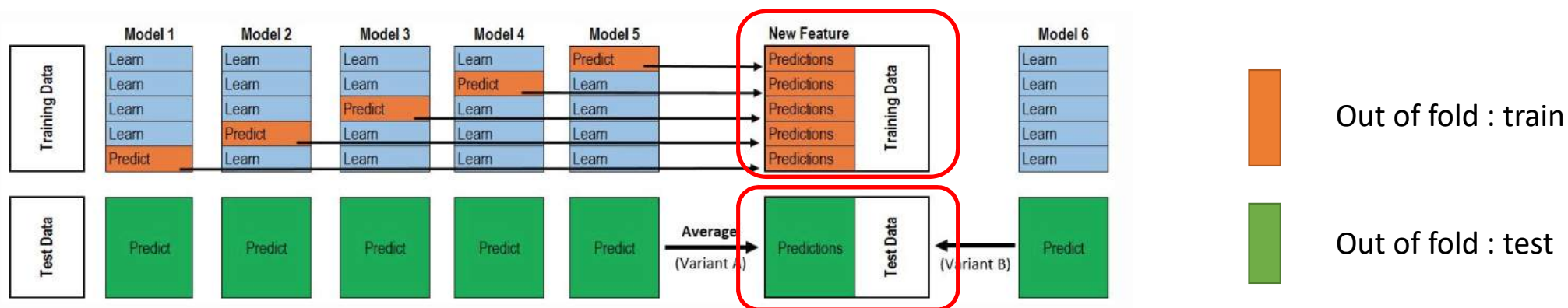


Sum of Weak learner is stronger than One strong learner.

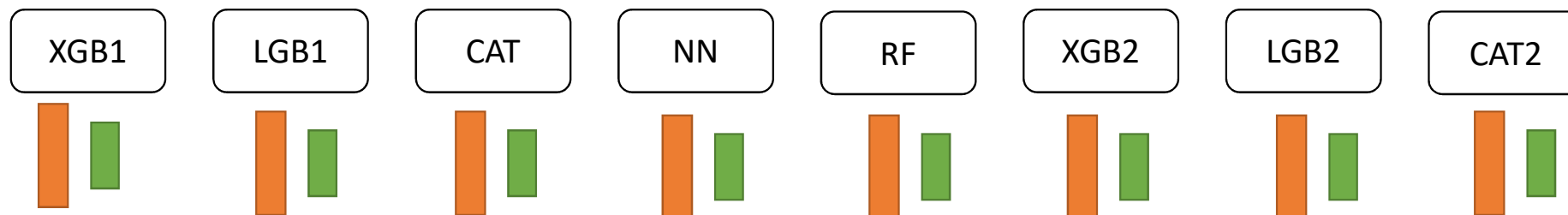
이렇게 해서 제출한
성적은?

동메달은 무슨...
그래도 TOP 15%
정도는 갔어요

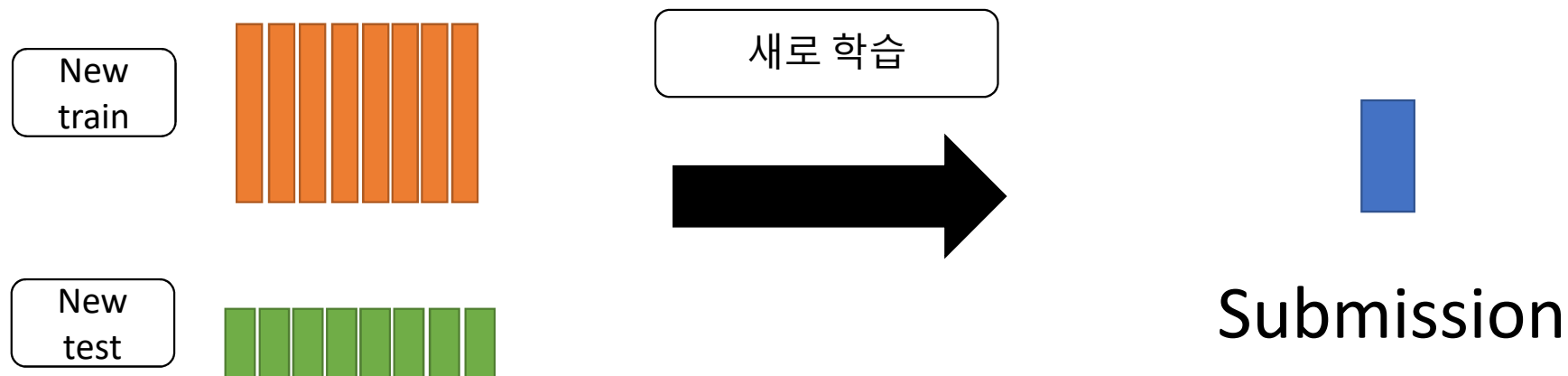
6. Stacking and Averaging



6. Stacking and Averaging

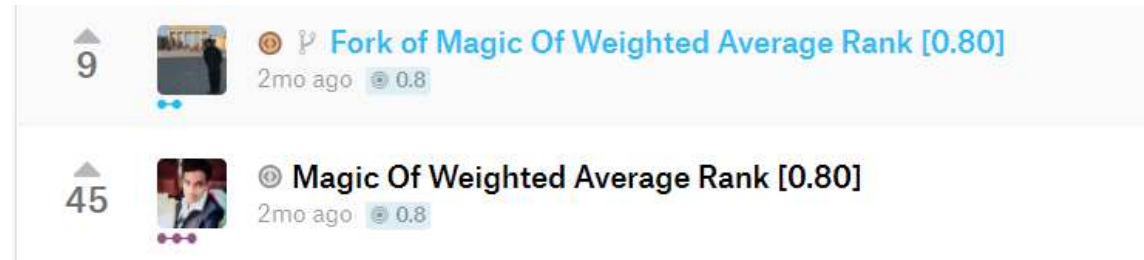
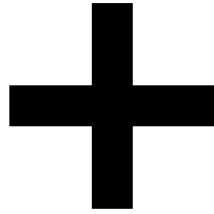


애네들을 feature 로 사용함



6. Stacking and Averaging


우리
Submission



다른 캐글러
Submissions

Simple average or weighted average

이렇게 해서 제출한
성적은?

동메달권
들어갔습니다.

자, 이제 더 성능을
올리려면 뭘
해야할까요?

Feature
generation

Parameter
tuning

More stacking

자, 이제 더 성능을
올리려면 뭘
해야할까요?

Feature
generation

Parameter
tuning

More stacking

다 해야합니다 ^^

저는
Parameter tuning 만
선택했습니다.

이것이 동메달로 끝내게 한
main reason 입니다.

7. Hyper parameter tuning

For Faster Speed

- Use bagging by setting `bagging_fraction` and `bagging_freq`
- Use feature sub-sampling by setting `feature_fraction`
- Use small `max_bin`
- Use `save_binary` to speed up data loading in future learning
- Use parallel learning, refer to [Parallel Learning Guide](#)

For Better Accuracy

- Use large `max_bin` (may be slower)
- Use small `learning_rate` with large `num_iterations`
- Use large `num_leaves` (may cause over-fitting)
- Use bigger training data
- Try `dart`

Deal with Over-fitting

- Use small `max_bin`
- Use small `num_leaves`
- Use `min_data_in_leaf` and `min_sum_hessian_in_leaf`
- Use bagging by set `bagging_fraction` and `bagging_freq`
- Use feature sub-sampling by set `feature_fraction`
- Use bigger training data
- Try `lambda_l1`, `lambda_l2` and `min_gain_to_split` for regularization
- Try `max_depth` to avoid growing deep tree

7. Hyper parameter tuning

Grid Search

- Parameters Grid space 를 만들어서 성능 확인
- 규칙적으로 optimum 을 찾아갈 수 있다.
- 하지만 grid size 에 따라 combination 이 너무 많아진다. Computational cost!
- 내가 어떤 영역을 잡느냐에 따라 optimum을 제대로 못 찾을 수 있다.

Randomized Search

- 각 Parameter 들의 range 를 만들어 그 안에서 임의로 숫자를 뽑아서 성능 확인
- 규칙적이지 않다. 하지만 trial 의 수는 많이 줄어 들 수 있다.
- 마찬가지로 내가 잡아주는 range 에 따라 optimum 을 제대로 못 찾을 수 있다.




Bayesian Optimization

- 성능 = $F(\text{parameters})$ 라는 함수를 가정하여, 그 함수의 형태를 추정하면서 global optimization 을 찾아가는 것.
- Bayesian 의 prior 개념이 들어감.
- 여러 trial 을 시도하면서, prior 를 이용해 global optimum 을 찾아가는 형태.
- Python package 가 있어서 사용하기 쉬움.
- 여러 trial 을 하면서 찾아가므로 기본 computational cost 가 있음.

물론 성능은 조금
좋아졌으나,
엄청난 Boost 는
없었습니다.

그리고 끝이 났습니다.

537 ▼ 29 Kaggle Korea Facebook onli... 0.79520 103 1mo

Competitions Contributor		
Unranked		
 0	 0	 1
Home Credit Default Risk a month ago · Top 8%		537 th of 7198
TensorFlow Speech Recog... 9 months ago · Top 43%		565 th of 1315
Statoil/C-CORE Iceberg Cl... 8 months ago · Top 20%		639 th of 3343

제 첫 컴퍼티션 메달입니다!

그 이후는..??

상위권
솔루션을 읽어봤습니다.

Study together, share together

write-ups

1st Bojan Tunguz: [I am speechless](#)

1st Bojan Tunguz: [1st Place Solution](#)

2nd Maxwell: [2nd place solution \(team ikiri_DS \)](#) [Github]

2nd Giba: [Congratulations, Thanks and Finding!!!](#)

3rd alijs: [3rd place solution](#)

4th Shubin: [4th place sharing and tips about having a good teamwork experience](#)

5th narsil: [Overview of the 5th solution \[+0.004 to CV/Private LB of your model\]](#)

7th Abdelwahed Assklou: [7th solution - **Not great things but small things in a great way.**](#)

8th Xuan Cao: [8th Solution Overview](#)

9th MichaelP: [#9 Solution](#)

10th nlgn: [10th place writeup](#)

12th zr: [#12 solution](#)

13th Μάριος Μιχαηλίδης KazAnova: [13th place - time series features](#)

14th seagull: [#14 solution](#)

16th propower: [The 16th Solution](#)

17th Qinghui Ge: [17th place mini writeup](#)

19th AllMight: [# 19th solution](#)

24th Arthur Llau: [24th place - Simple Solution with 7 Models.](#)

27th nyanp: [Pseudo Data Augmentation \(27th place short writeup\)](#) [Github]

32th arnowaczynski: [Story behind the 32th place \(top 1%\) with 2 submissions](#)

48th James Davis: [Simple feature that made public kernels top 50 \(and thanks!\)](#)

아직도 뇌리에 남는
글귀가 있었습니 다.

Silogram
 Freelance data scientist at Self-employed
 Zurich, Switzerland
 Joined 6 years ago · last seen in the past day
 Followers 829
 Following 40
 Competitions Grandmaster

Home Competitions (71) Kernels (3) Discussion (129) Datasets (1) Followers (829) Contact User Follow User

Competitions Grandmaster		Kernels Contributor		Discussion Expert	
Current Rank	Highest Rank	Unranked		Current Rank	Highest Rank
13 of 91,032	7			17 of 69,506	16
12	28	0	0	18	20
Home Credit Default Risk a month ago · Top 1%	1st of 7198	sentiment_analysis 2 years ago	3 votes	A few notes... 4 months ago	376 votes
Grupo Bimbo Inventory De... 2 years ago · Top 1%	1st of 1989	XGBoost Starter Code (pyt... 3 years ago	3 votes	Notes from sub-0.5 solution a year ago	136 votes
Quora Question Pairs a year ago · Top 1%	2nd of 3307	Random Forest Starter wit... 2 years ago	0 votes	Overview of 2nd-Place Solu... a year ago	99 votes

5. As in all Kaggle competitions (and all machine learning problems, for that matter), the most important first step is to get a validation set-up that matches the test set. There's no point in spending time on feature-engineering before your validation system is trustworthy.

Feature engineering 보다 먼저 할 것은
 stable, trustworthy validation system 을 만드는 것!

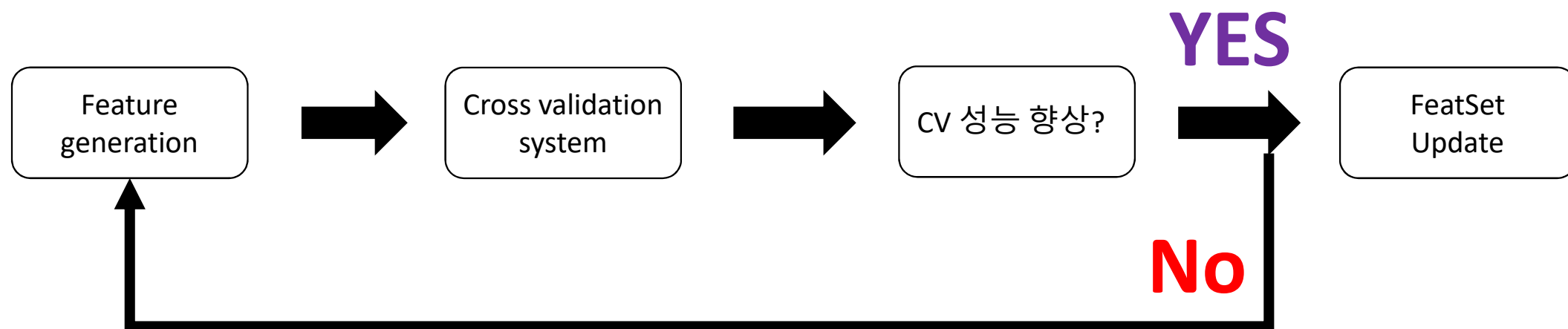
당연했습니다.

Feature 를 만들어도,
그것이 정말 좋은 것인지
판단해야 하는데,
그 판단 시스템이
믿을 만 해야 하잖아요?

그 이후는..??

캐글에 깊~게 빠졌습니다.
그리고 접목했습니다.

Lesson1 - Feature generation 과 Cross-validation 은 함께 해야 한다.

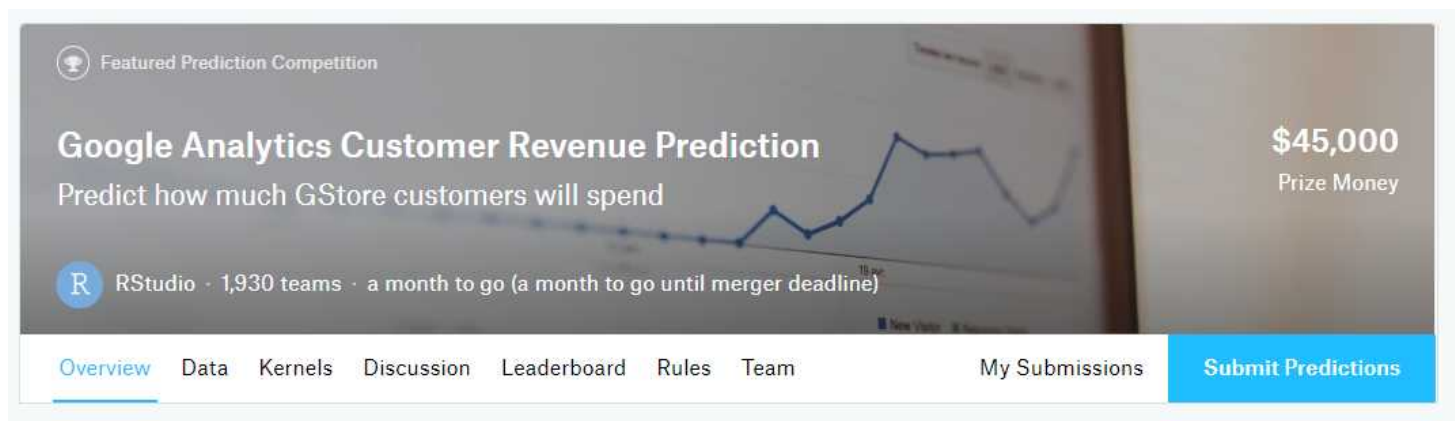


Bottom-Up Feature selection

Lesson2 - 모든 결과를 기록해야 한다.








날짜	Featset version	Feature 설명	성능(ROC)	Choose?
2018/09/09	FeatSet 1	Initial features	0.660	
2018/09/11	FeatSet 2	FeatSet1 + Time features	0.680	O
2018/09/14	FeatSet 3	FeatSet2 + Ratio features	0.700	O
2018/09/15	FeatSet 4	FeatSet3 + New A category	0.690	X
2018/09/16	FeatSet 5	FeatSet3 + New B category	0.720	O
2018/09/17	FeatSet 6	FeatSet5 + other encoding	0.760	O
....

이 두 교훈을 바탕으로 새로이 시작한 google competition!















Gstore customer 의
revenue 를 예측하라!

Feature selection 은 Bottom – Up 이다.

 Featset8_User_level_feature_selection.ipynb	
 Featset8_User_level.ipynb	
 Featset7_User_Level.ipynb	
 Featset7+classification_features.ipynb	
 Featset7_User_Level_Adding_classification_feature.ipynb	
 timeFeat+(browser, OS)+(visit, hits, page_per_ID)+(visit, hits_per_session)+ratio(hit, visit, page)+peller+source(only_cat)+ref(interaction).ipynb	7일 전
 timeFeat+(browser, OS)+(visit, hits, page_per_ID)+(visit, hits_per_session)+ratio(hit, visit, page)+peller+source(only_cat)+ref(interaction)-Copy1.ipynb	8일 전
 Encoding_comparision_final.ipynb	8일 전
 timeFeat+(browser, OS)+(visit, hits, page_per_ID)+(visit, hits_per_session)+ratio(hit, visit, page)+peller+source(cat+num).ipynb	8일 전
 timeFeat+(browser, OS)+(visit, hits, page_per_ID)+(visit, hits_per_session)+ratio(hit, visit, page)+peller+source(only_cat).ipynb	8일 전
 timeFeat+(browser, OS)+(visit, hits, page_per_ID)+(visit, hits_per_session)+ratio(hit, visit, page)+peller.ipynb	8일 전
 timeFeat+(browser, OS)+(visit, hits, page_per_ID)+(visit, hits_per_session)+ratio(hit, visit, page)+geoNetworkDomain.ipynb	8일 전
 timeFeat+(browser, OS)+(visit, hits, page_per_ID)+(visit, hits_per_session)+ratio(hit, visit, page).ipynb	8일 전
 timeFeat+(browser, OS)+(visit, hits, page_per_ID)+(visit, hits_per_session).ipynb	8일 전
 timeFeat+(browser, OS)+(visit, hits, page_per_ID)+(visit, hits, page_per_day).ipynb	8일 전
 timeFeat+(browser, OS)+(visit, hits, page_per_ID).ipynb	8일 전
 timeFeat+browser+os.ipynb	8일 전
 featset6.ipynb	8일 전

세상에나!!!!!!! 너무 행복합니다.

13	new	Jordan Meyer		1.4085	22	2d
14	▼ 12	Herra Huu		1.4086	31	8h
15	▲ 8	Yiemon		1.4089	73	4h
16	▼ 13	Sergei Fironov		1.4090	92	17h
17	▼ 10	GolemMaking		1.4120	67	10h
18	new	lights		1.4120	34	10h
19	▲ 209	Jane Lee		1.4122	50	11h
20	▲ 4	Manuel Campos		1.4124	109	3h
21	▼ 11	Roger Gou	   +4	1.4126	10	2d
22	▲ 24	YouHan Lee		1.4127	98	2h

Your Best Entry ↑

Your submission scored 1.4172, which is not an improvement of your best score. Keep trying!



Google Analytics Customer Revenue Prediction







Predict how much GStore customers will spend

Featured - a month to go - tabular data, regression









22/2020
Top 2%

지금 등수의 비결은?

- 52  Which encoding is good for time-validation?-1.4417
7d ago 1.4417
- 68  Stratified sampling for regression LB: 1.4627
8d ago 1.4627
- 41  EDA + LGBM for starter LB: 1.6878
19d ago 1.5725
- 106  user_level_lightgbm LB 1.4480
7d ago 1.448
- 121  Teach_LightGBM_to_Sum_Predictions
4d ago 1.4285
- 44  Mean (likelihood) encodings: a comprehensive study
13d ago tutorial, classification, feature engineering, gradient boosting

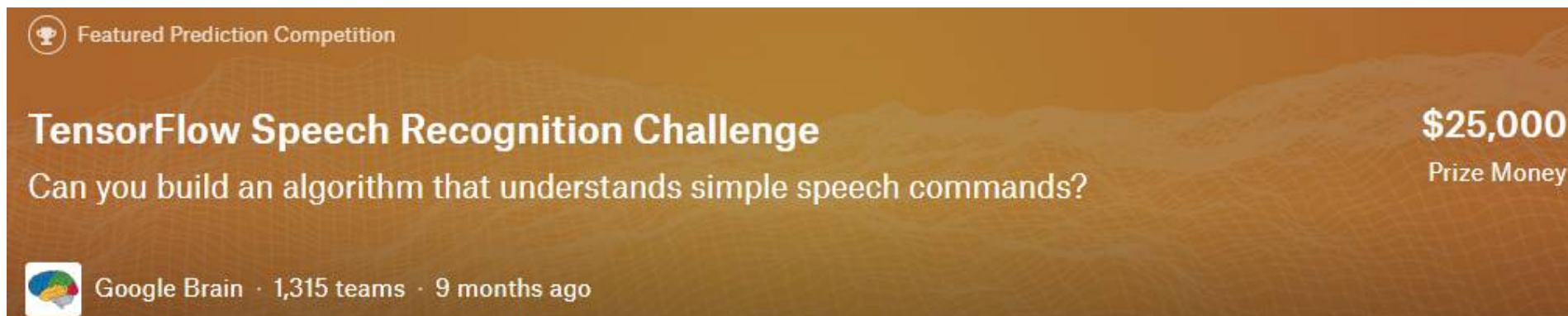
이미
길은 다
공유되어
있습니다 ^^

지금 등수의 비결은?

52		🔒 Which encoding is good for time-validation?-1.4417 7d ago 1.4417
68		🔒 Stratified sampling for regression LB: 1.4627 8d ago 1.4627
41		🔒 EDA + LGBM for starter LB: 1.6878 19d ago 1.5725
106		🏆 user_level_lightgbm LB 1.4480 7d ago 1.448
121		🏆 Teach_LightGBM_to_Sum_Predictions 4d ago 1.4285
44		🔒 Mean (likelihood) encodings: a comprehensive study 13d ago tutorial, classification, feature engineering, gradient boosting

캐글은
함께 발전하는
커뮤니티입니다.
여러분은
혼자가
아닙니다

딥러닝 컴퍼티션들 – Image recognition and classification




Featured Prediction Competition

TensorFlow Speech Recognition Challenge

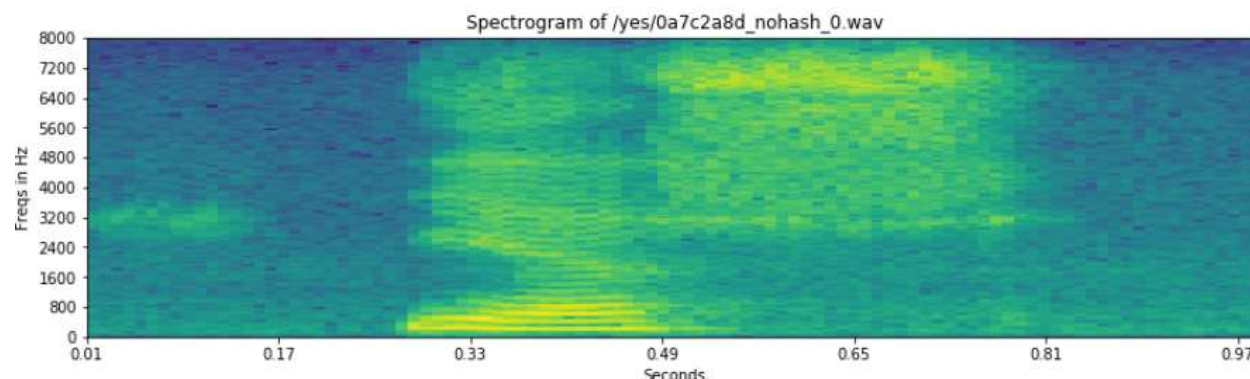
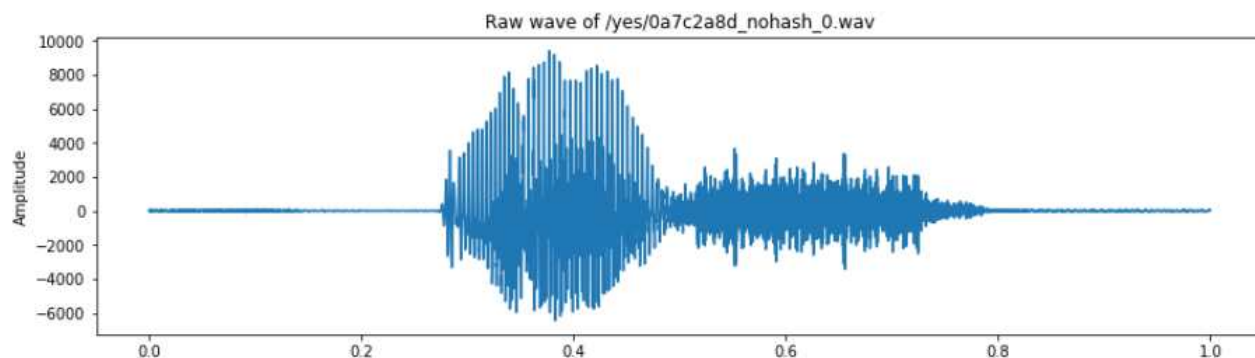
Can you build an algorithm that understands simple speech commands?

\$25,000
Prize Money

 Google Brain · 1,315 teams · 9 months ago

Yes, no, up, down, left, right, on, off, stop, go, silence, others
위 단어들을 구분할 수 있는 machine 을 만들어라!

Solution process (1) 더 많은 feature 을 위한 1D -> 2D 변환



Solution process (2) 2D-CNN 을 사용한 이미지 학습

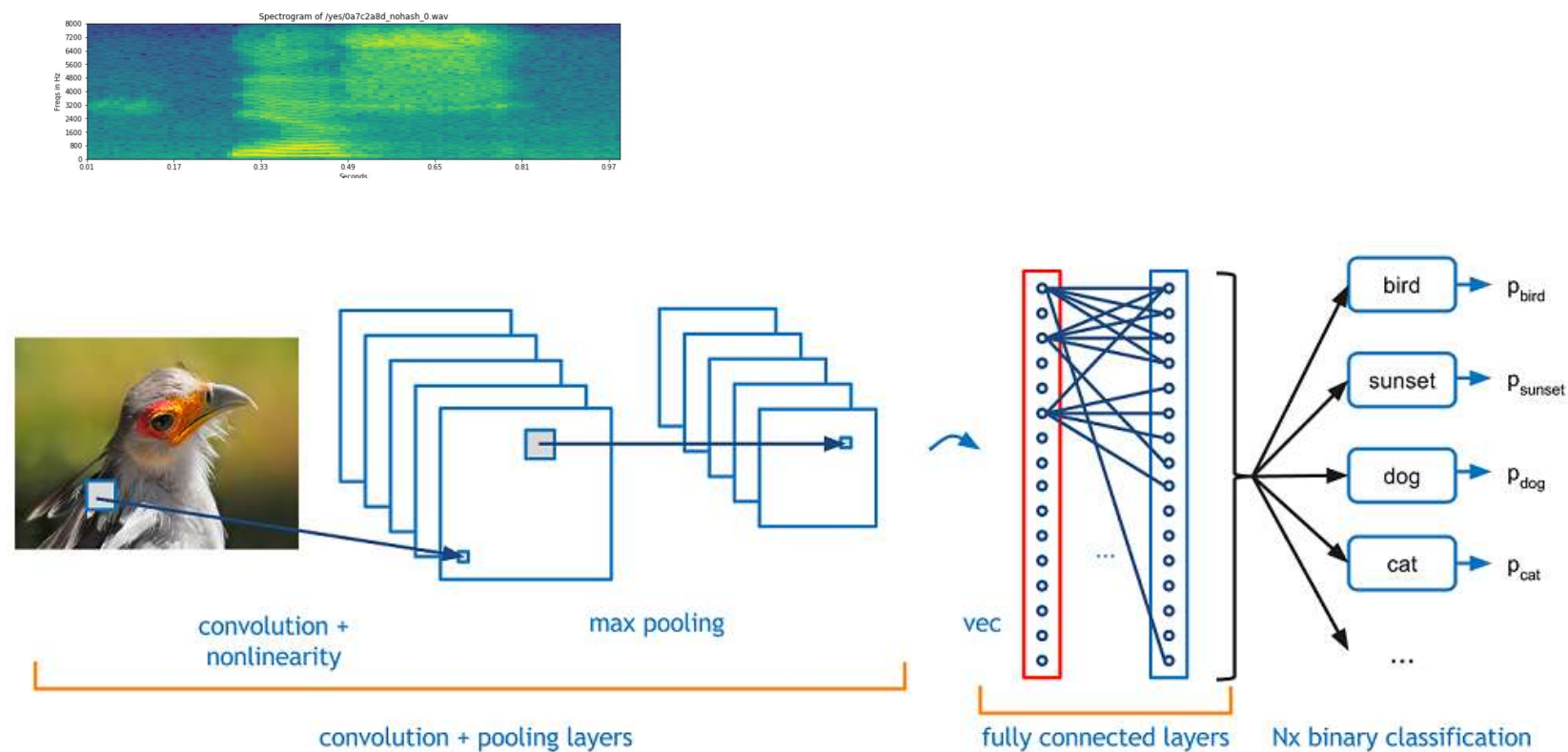


Image feature extraction

Integration of features

2D-CNN은 어떻게 만드나?

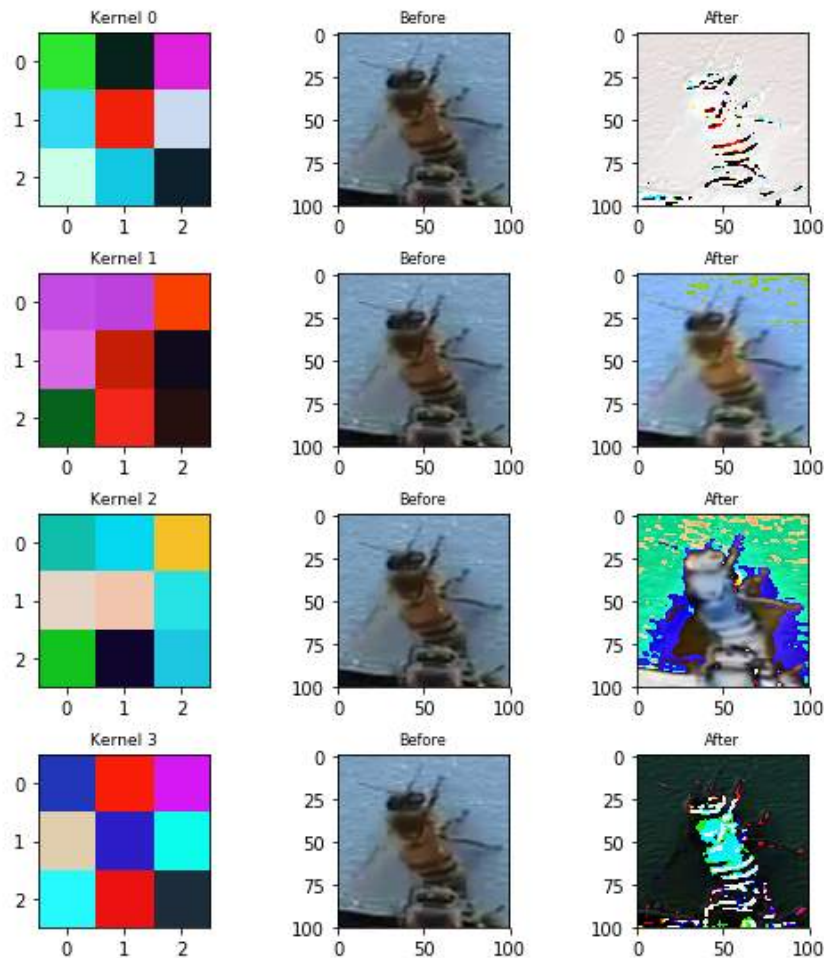


+



Keras

최신 동향 – Pre-trained model 사용

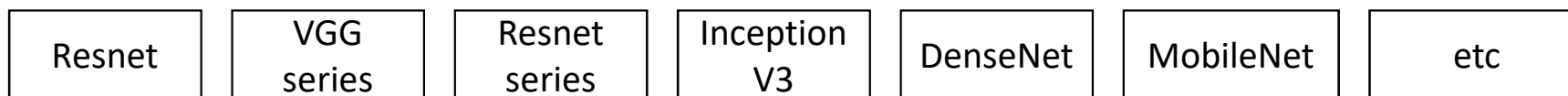
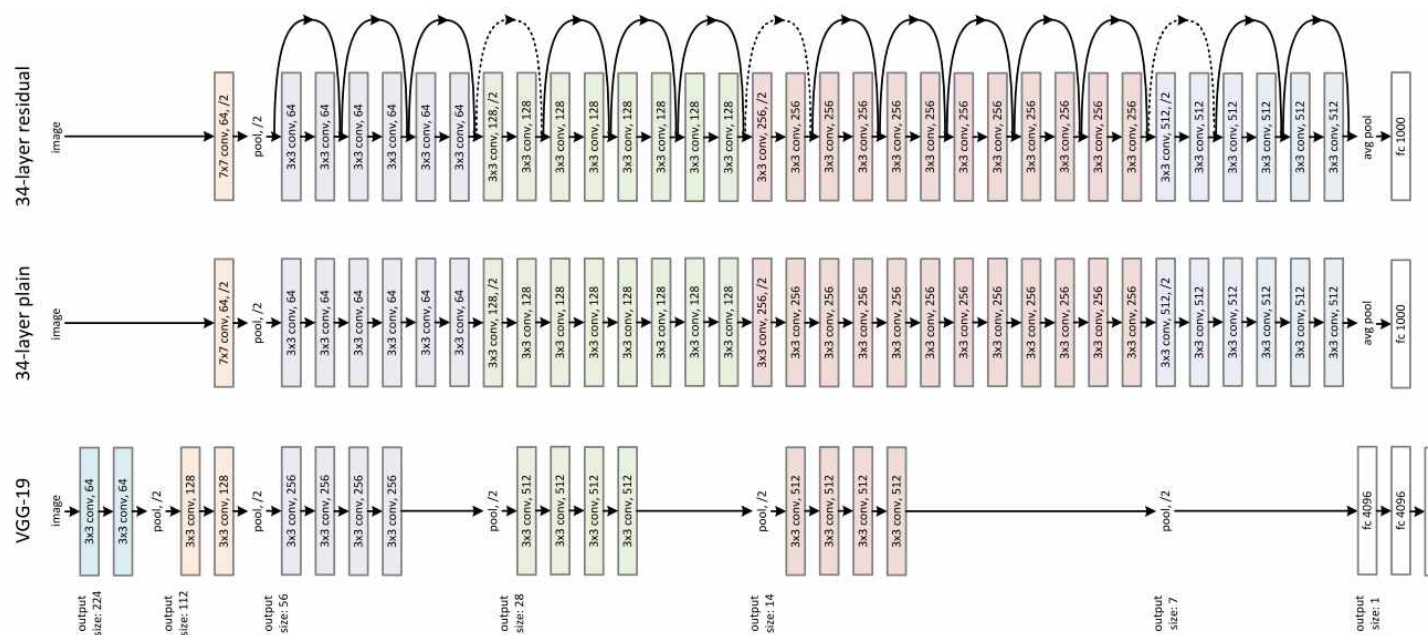


CNN 은 Image 에서 중요한 feature
를 뽑아내는 하나의 도구

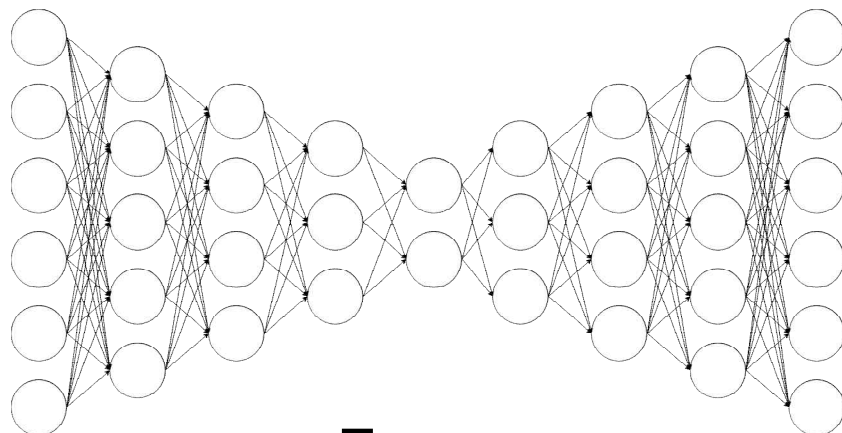


Image 에서 feature 를 잘
뽑아내게 학습된 model 을
가져와 쓰자

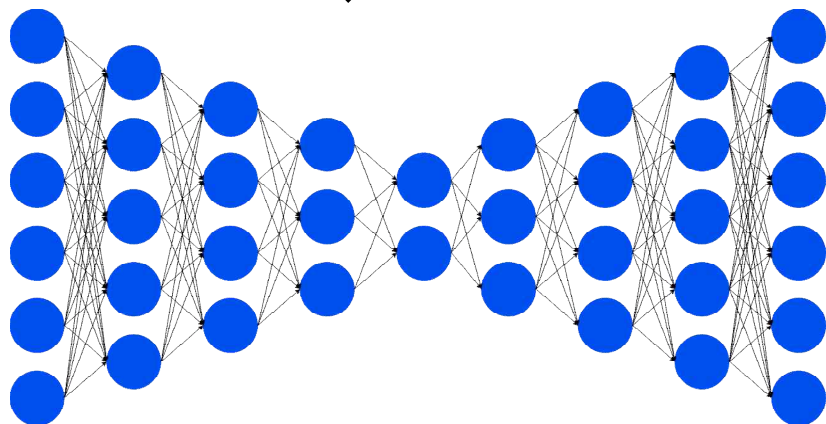
최신 동향 – Pre-trained model 사용



최신 동향 – Pre-trained model 사용



Training



Non-trained
-> random numbered(initialized)
weights

Trained
-> Optimized weights

사용방법? – Use built-in function

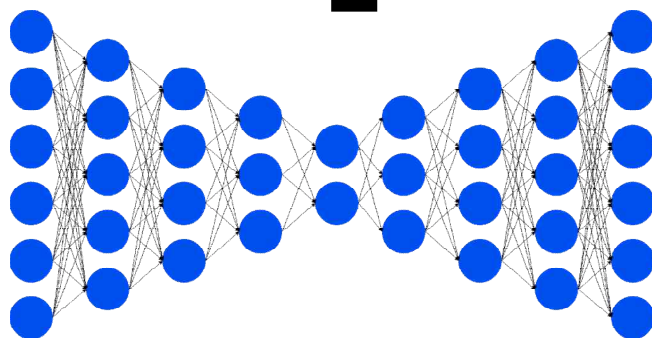
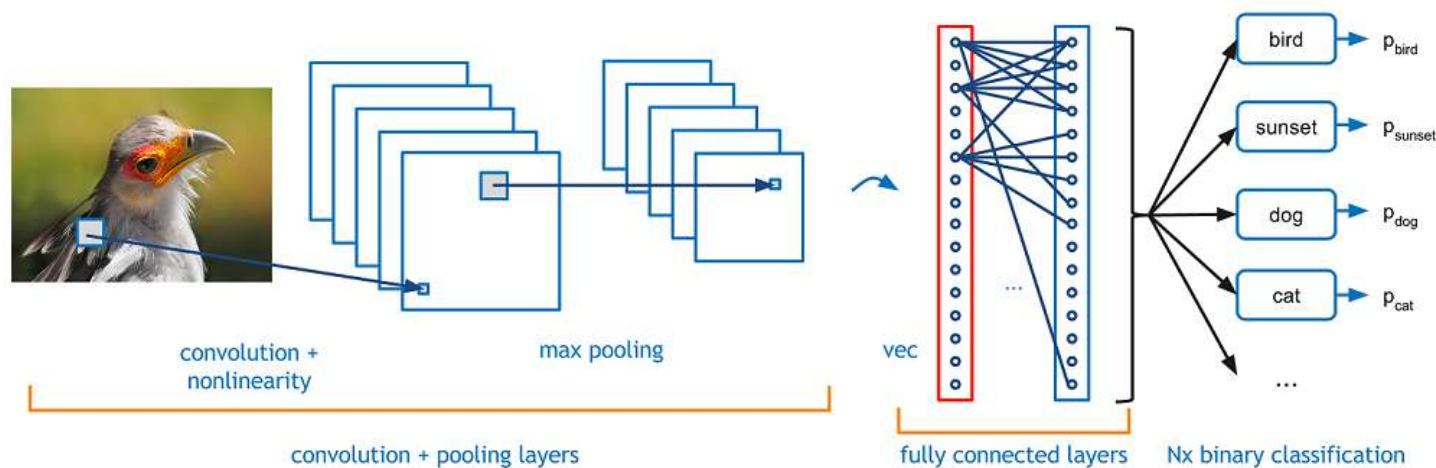
```
from keras.applications.resnet50 import ResNet50
from keras.preprocessing import image
from keras.applications.resnet50 import preprocess_input, decode_predictions
import numpy as np

model = ResNet50(weights='imagenet')

img_path = 'elephant.jpg'
img = image.load_img(img_path, target_size=(224, 224))
x = image.img_to_array(img)
x = np.expand_dims(x, axis=0)
x = preprocess_input(x)

preds = model.predict(x)
# decode the results into a list of tuples (class, description, probability)
# (one such list for each sample in the batch)
print('Predicted:', decode_predictions(preds, top=3)[0])
# Predicted: [(u'n02504013', u'Indian_elephant', 0.82658225), (u'n01871265', u'tusker', 0.1122357), (u'n02504458
```


사용방법? – Use built-in function



Yes, no, up, down, left,
right, on, off, stop, go,
silence, others

딥러닝 컴퍼티션들 – Object detection and segmentation

Featured Prediction Competition

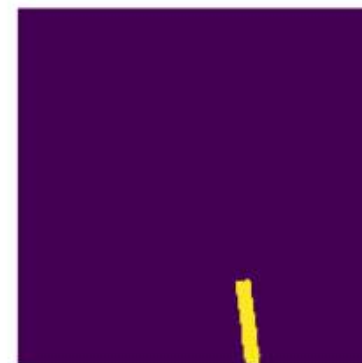
Airbus Ship Detection Challenge

Find ships on satellite images as quickly as possible

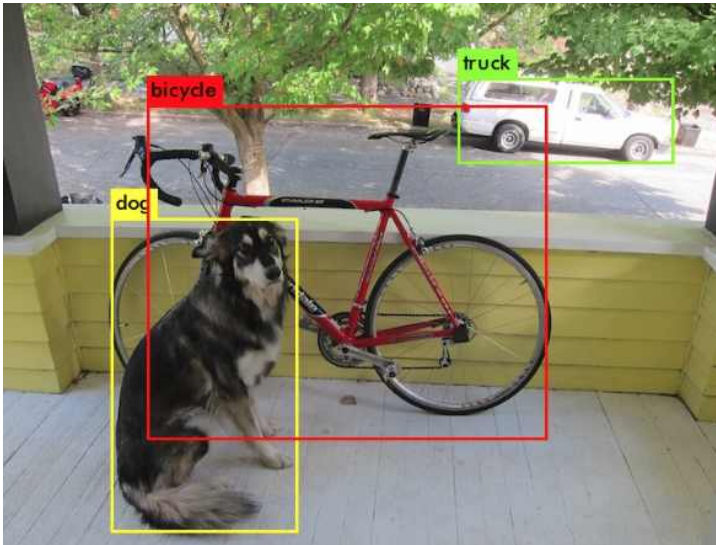
\$60,000
Prize Money

A Airbus · 616 teams · 13 days to go (6 days to go until merger deadline)

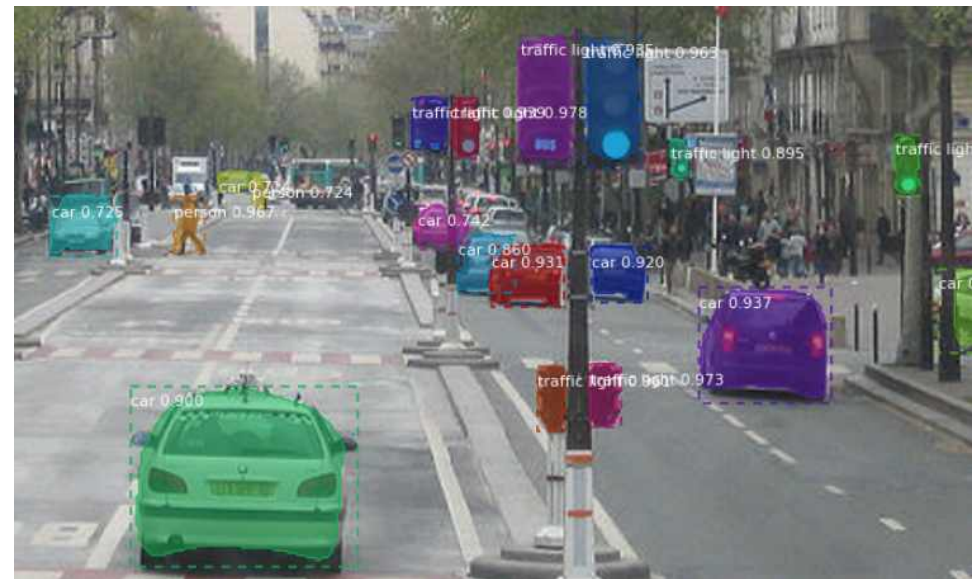
위성 사진 속 배들을 찾아라!



검증된 많은 모델들이 존재.



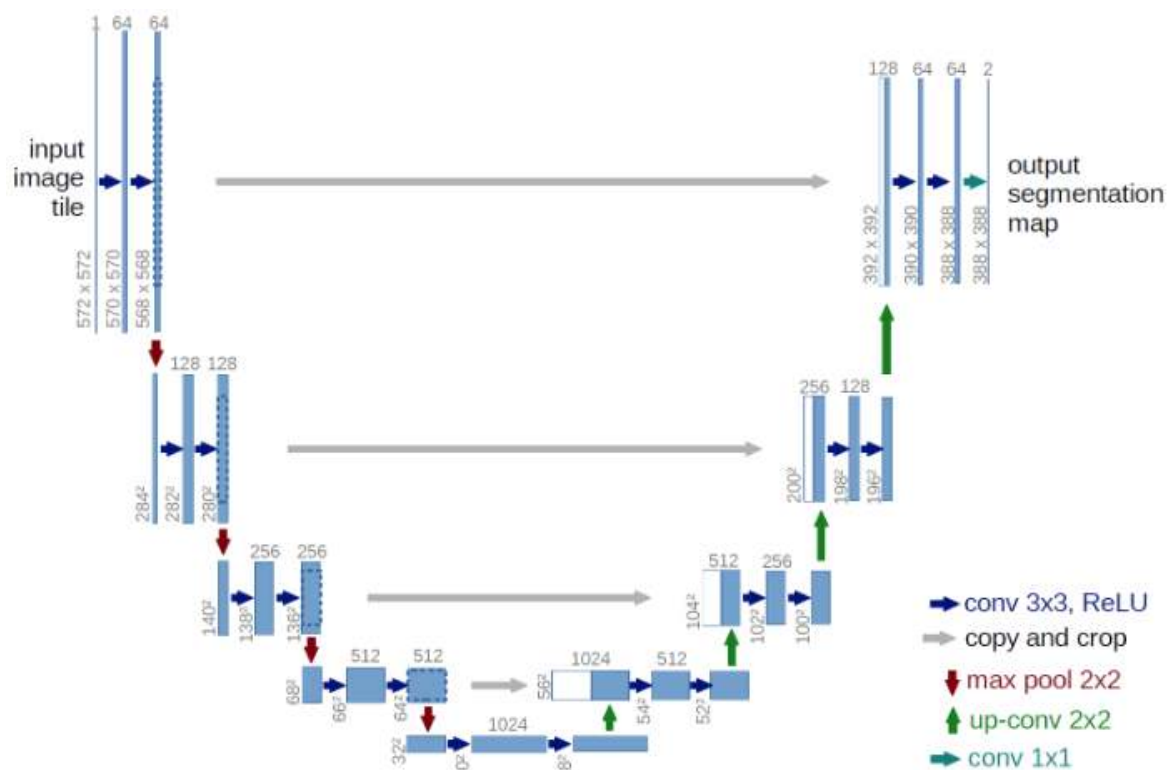
Mask R-CNN



검증된 많은 모델들이 존재.

U-net

- Biomedical image segmentation 에 검증됨
- Transfer 가능



이런 지식을 캐글에서
지식을 얻으려면?

캐글의 위대한 유산 – 공유 정신

write-ups

1st Bojan Tunguz: [I am speechless](#)

1st Bojan Tunguz: [1st Place Solution](#)

2nd Maxwell: [2nd place solution \(team ikiri_DS \) \[Github\]](#)

2nd Giba: [Congratulations, Thanks and Finding!!!](#)

3rd alijs: [3rd place solution](#)

4th Shubin: [4th place sharing and tips about having a good teamwork experience](#)

5th narsil: [Overview of the 5th solution \[+0.004 to CV/Private LB of your model\]](#)

7th Abdelwahed Assklou: [7th solution - **Not great things but small things in a great way.**](#)

8th Xuan Cao: [8th Solution Overview](#)

9th MichaelP: [#9 Solution](#)

10th nlgn: [10th place writeup](#)

12th zr: [#12 solution](#)

13th Μαρκος Μιχαηλιδης KazAnova: [13th place - time series features](#)

14th seagull: [#14 solution](#)

16th propower: [The 16th Solution](#)

17th Qinghui Ge: [17th place mini writeup](#)

19th AllMight: [# 19th solution](#)

24th Arthur Llau: [24th place - Simple Solution with 7 Models.](#)

27th nyanp: [Pseudo Data Augmentation \(27th place short writeup\) \[Github\]](#)

32th arnowaczynski: [Story behind the 32th place \(top 1%\) with 2 submissions](#)

48th James Davis: [Simple feature that made public kernels top 50 \(and thanks!\)](#)

Register with just one click:

We won't share anything without your permission

Google

Facebook

Yahoo

Manually create an account:

Email

Password

Register

구글,
페이스북,
야후 아이디로
가입만 하시면
됩니다

몇가지 천기 누설!!!!

- 투자한 시간은 배신하지 않는다.(from 김현우)
 - Keep going!
- 데이터 탐색 >>> 파라미터 튜닝 (from 김연민)
- Garbage in, Garbage out
 - Show me the feature!
- See kernel, See discussion
- Many weak learner win one strong learner
 - ensemble
- Make Cross-validation system along with LB

캐글 코리아

Kaggle Korea

Non-Profit Facebook Group Community

함께 공부해서, 함께 나눕시다
Study Together, Share Together

들어주셔서
감사합니다!