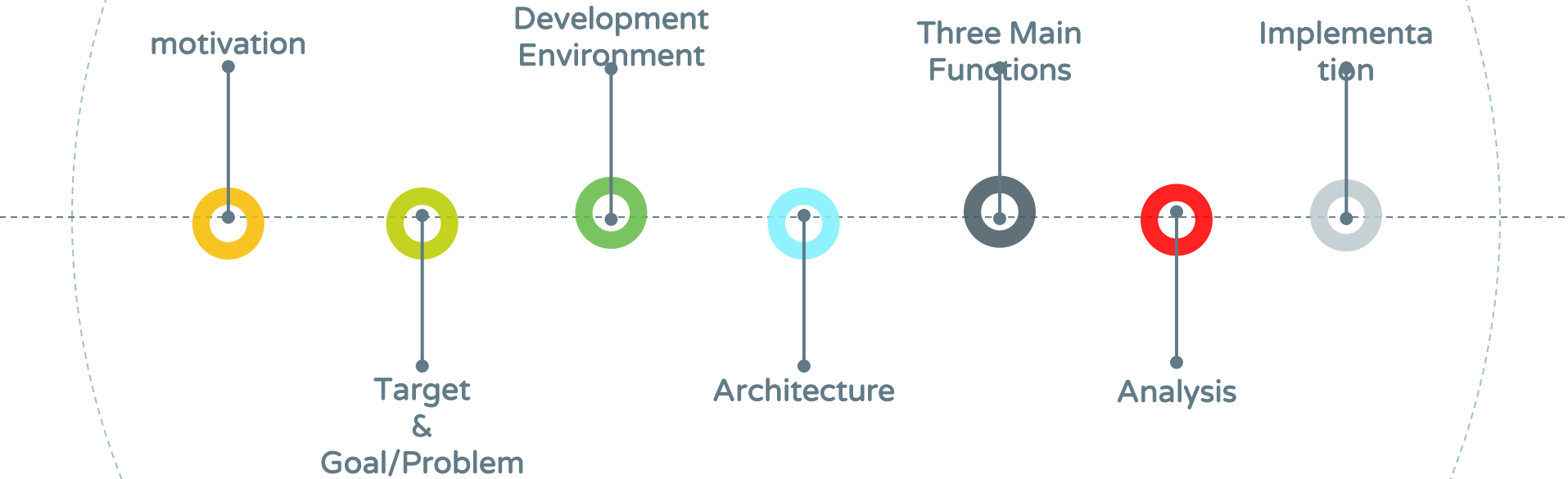
The background is white with several large, colorful circles in teal, lime green, orange, and pink. Some circles have smaller circles inside them, creating a nested effect. Dashed lines in various colors (teal, green, yellow) arc across the slide, connecting different points. The text is centered in a light blue, sans-serif font.

Prepare for
coding test
efficiently!

2019110627 CSE Kim MyeongJu
2016104131 CSE An YoungMin
2019110634 CSE Lee ChangRyeol

Index



Motivation



BEGINNERS FACE VARIOUS
DIFFICULTIES IN
PREPARING FOR A CODING
TEST.



MOST PROBLEM-SOLVING
SITES SHOW OVERALL
INFORMATION, SO IT IS
DIFFICULT TO FIND
INFORMATION THAT SUITS
THE USER AT A GLANCE.

Team Presentation

Kim MyeongJu

Team Leader
Scrum Master
Data Acquisition
Data Visualization

An YoungMin

Data Analysis
Data Storage
Optimizing
performance

Lee ChangRyeol

Data Acquisition
Data Analysis
Project Document

Milestone

	Week 1	Week 2	Week 3	Week 4	Week 5	Week 6
Data Acquisition						
Data Storage						
Data Analysis						
Data Visualization						

Target & Goal/Problem



For Anyone preparing for a coding test

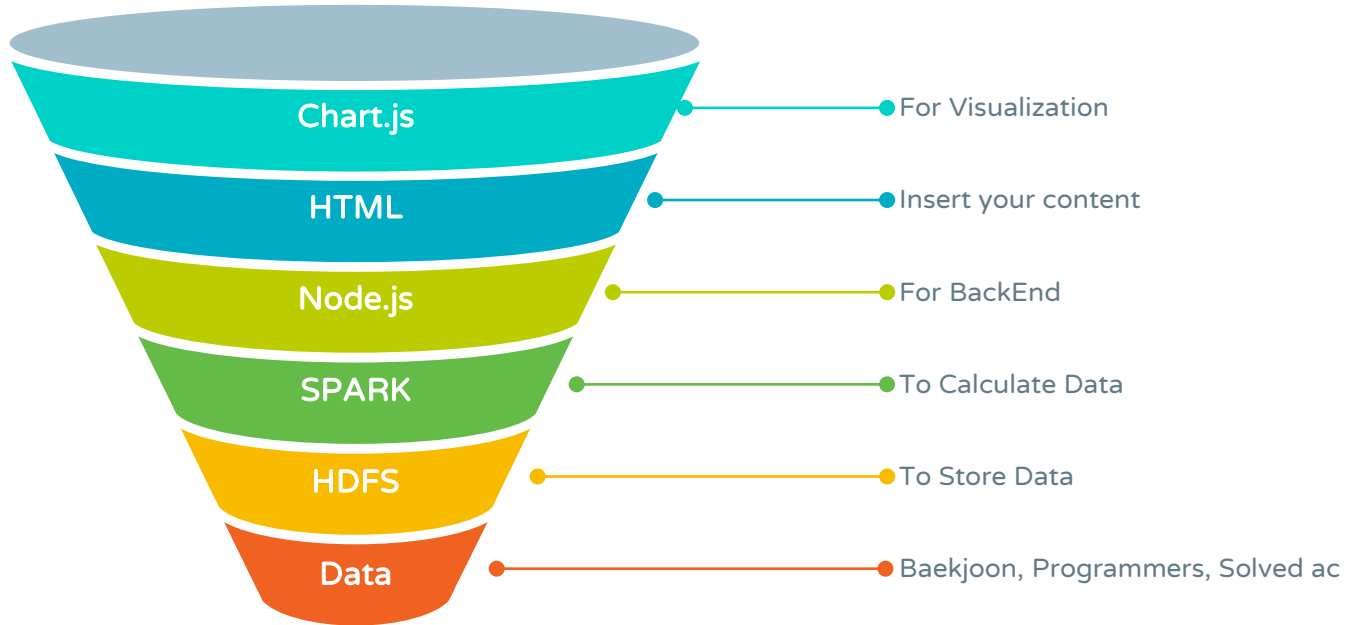


Problem : There is little information on companies for coding test

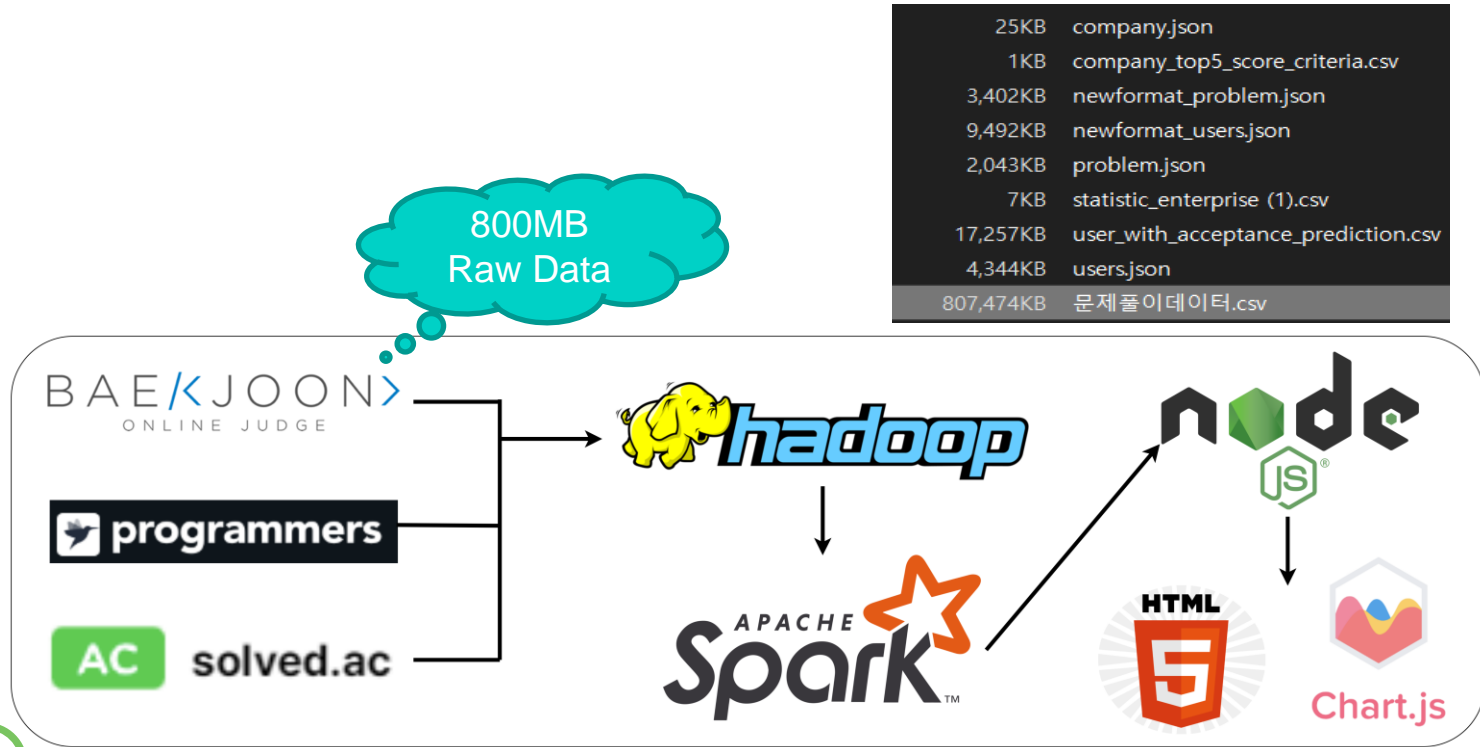


Goal : Show what types of problems do users need to solve more in order to go to a certain company

Development Environment



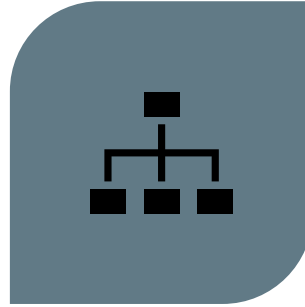
Architecture



Main Functions



WHAT TYPES OF
PROBLEMS USERS HAVE
SOLVED THE MOST



STUDY STRUCTURE
WITH PEOPLE OF
SIMILAR LEVEL



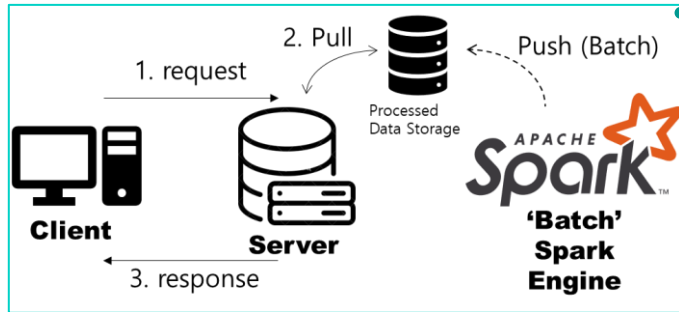
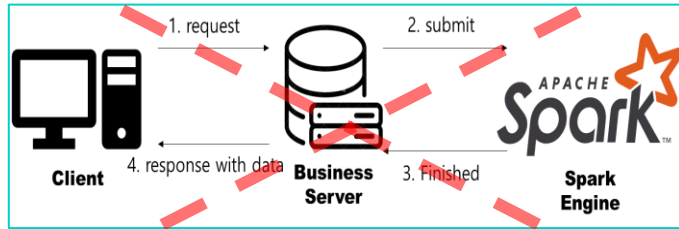
PASS PREDICTION

Analysis



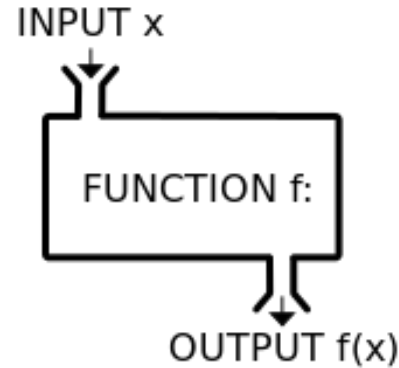
The Process we Analyzed

Batch Processing



Why we took the batch processing?

because all needed output is **deterministic!!!**



Raw Data

user.json

	userId	rank
type	String	Enum(RANK)
ex	abc123	마스터

problem.json

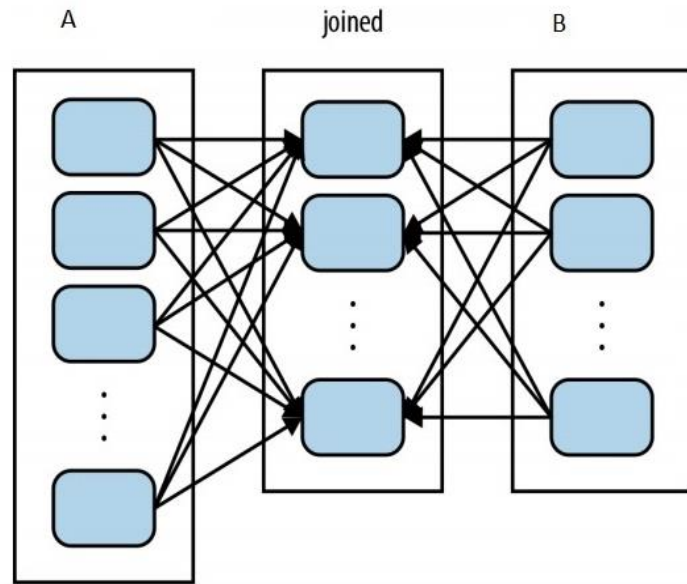
	questionId	title	difficulty	type
type	Integer	String	Enum(RANK)	Slash-Seperated-Value
ex	1000	A+B	브론즈 5	수학/구현/사칙연산

company.json

	company	title	difficulty	type
type	String	String	Enum(RANK)	Slash-Seperated-Value
ex	kakao	추석 트래픽	실버 2	브루트포스/구현/자료구조

trials.csv

	user	user_rank*	questionId	title	result	memory	time	language	volume	date
ex	abc123	마스터	1000	A+B	틀렸습니다!!	N/A	N/A	C++ 17	730	2022-05-14 00:00:00 PM



Monthly Top Algorithm Analysis

trial_df

	user	type	...	result	date
ex	abc123	수학/구현/사칙연산		틀렸습니다!!	2022-05-14 00:00:00 PM

```
withColumn("month", split("date", "-")[1])  
col("type").split("/").explode("type")
```

```
groupBy("month", "type")
```

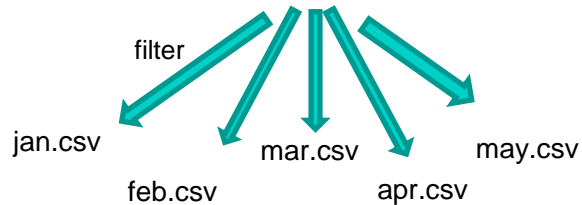
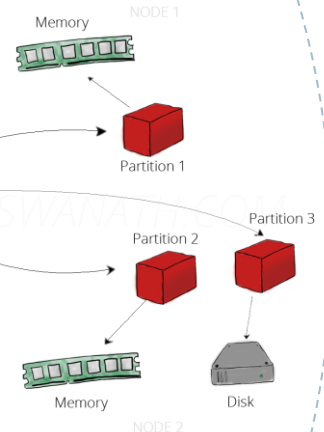
trial_per_type_df

	month	type	...	count
ex	02	수학		510

trial_per_type_df
.persist(MEMORY_AND_DISK)

RDD

rdd.persist(MEMORY_AND_DISK)

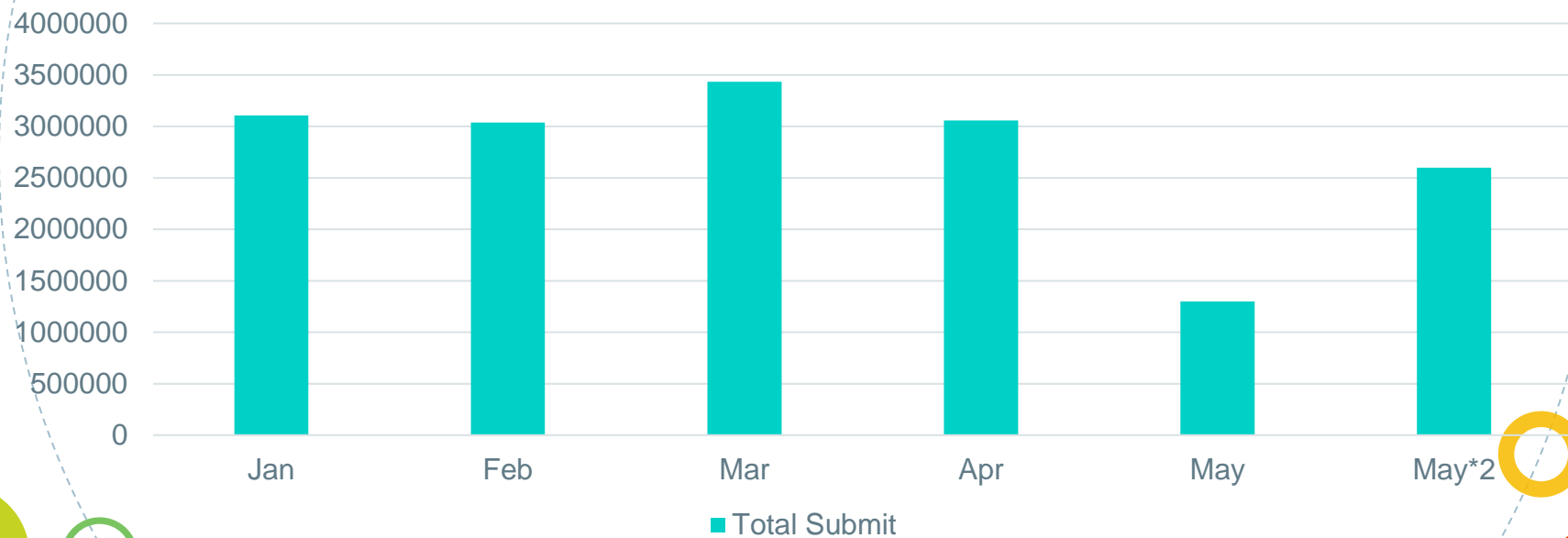


Monthly Top Algorithm Analysis

순위	1월	제출합	2월	제출합	3월	제출합	4월	제출합	5월	제출합
1	구현	573055	구현	502249	구현	601114	구현	557489	구현	223313
2	수학	537963	수학	454039	수학	554561	수학	487943	수학	207147
3	사칙연산	275737	사칙연산	232013	사칙연산	305384	사칙연산	266794	사칙연산	108169
4	그래프 이론	132869	그래프 이론	157594	그래프 이론	182649	그래프 이론	163517	그래프 이론	62522
5	자료 구조	129680	자료 구조	137495	그래프 탐색	143376	그래프 탐색	129326	자료 구조	59980
6	문자열	126303	그래프 탐색	121482	다이나믹 프로그래밍	138367	자료 구조	112730	문자열	52726
7	다이나믹 프로그래밍	117649	다이나믹 프로그래밍	120728	자료 구조	137610	다이나믹 프로그래밍	108485	다이나믹 프로그래밍	48256
8	정렬	103045	문자열	116597	문자열	123693	문자열	108205	그래프 탐색	48046
9	그래프 탐색	101461	정렬	103484	너비 우선 탐색	114599	너비 우선 탐색	105267	정렬	44291
10	브루트포스	96642	브루트포스	99322	정렬	109605	브루트포스	101127	브루트포스	38536

Monthly Top Algorithm Analysis

Total Submit



Top Algorithm per user and tier group

trial_df

	user	type	...	result	date
Ex)	abc123	수학/구현/사칙연산		틀렸습니다!!	2022-05-14 00:00:00 PM

['맞았습니다!!', '맞았습니다!! (2/3점)',...] -> **correct**
['틀렸습니다', '메모리 초과', '시간 초과'] -> **wrong**
['런타임 에러(..)', '컴파일 에러(..)', '출력 오류 ..'] -> **error**
etc -> **etc**

categorize_udf

categorized_df

	user	type	...	result	date
Ex)	abc123	수학		wrong	2022-05-14 00:00:00 PM

groupBy('user', 'type', 'result')
agg(count)
sort(count.desc)

correct_stat_user.csv

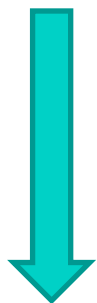
groupBy('user_rank', 'type', 'result')
agg(count)
sort(count.desc)

1correct_stat_rankgroup.csv

Top Algorithm per company

company_df

	company	difficulty	title	type
Ex)	kakao	실버 5	문자열 압축	문자열/구현



브론즈 5 = 1.0
브론즈 4 = 2.0

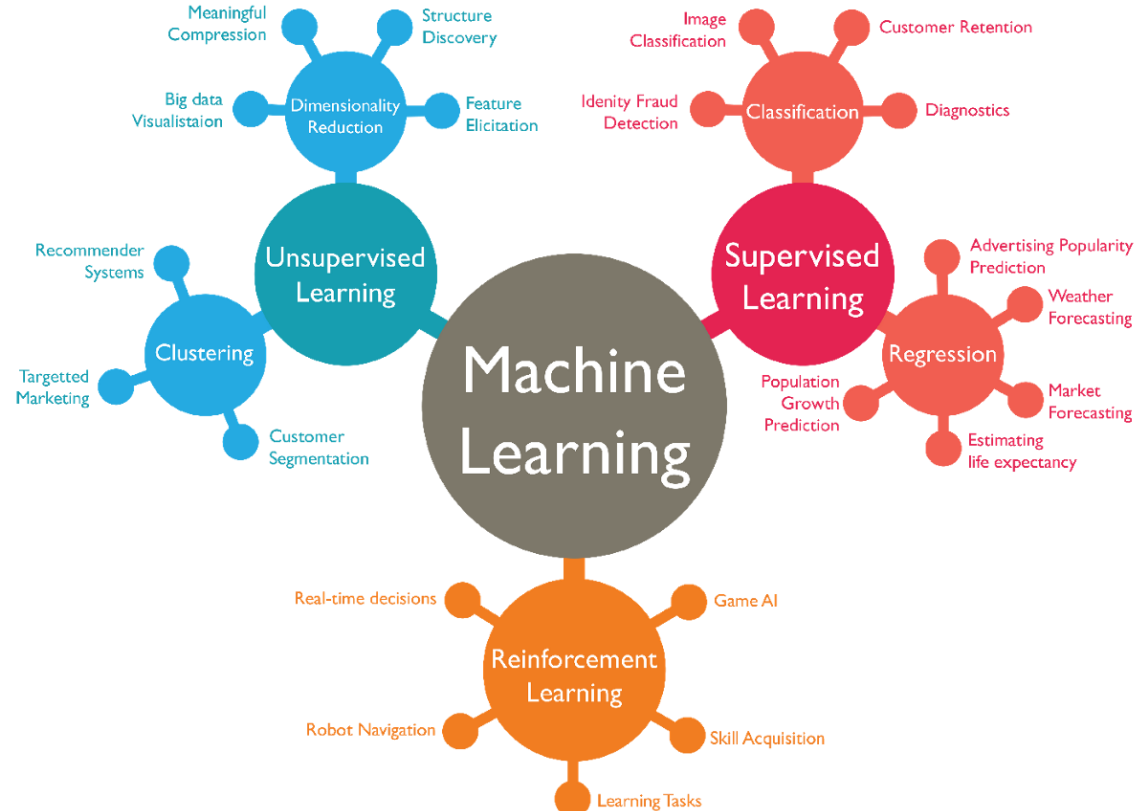
...
다이아몬드 2 = 24.0
다이아몬드 2 = 25.0

company_stat_df

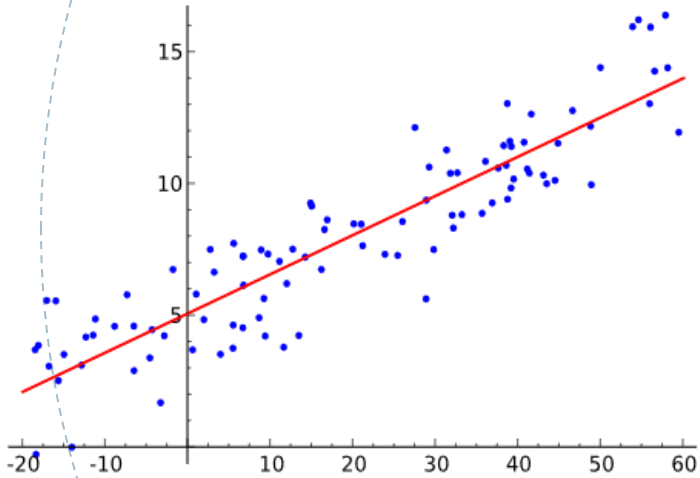
	company	type	count	avg_of_difficulty
Ex)	coupang	문자열	22	10.36

순위	Naver	KaKao	Line	Coupang	Samsung
1	문자열	구현	구현	구현	구현
2	구현	문자열	브루트 포스	자료 구조	시뮬레이션
3	자료 구조	브루트 포스	문자열	문자열	브루트 포스
4	수학	그래프 이론	수학	수학	그래프 이론
5	그래프 탐색	다이나믹 프로그래밍	자료 구조	브루트 포스	그래프 탐색
6	그래프 이론	깊이 우선 탐색	그래프 탐색	백트래킹	백트래킹
7	해시	너비 우선 탐색	그래프 이론	두 포인터	자료 구조

Predicting the probability of acceptance

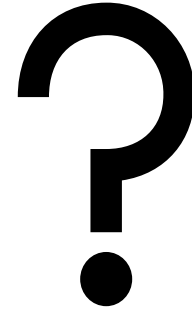


Predicting the probability of acceptance



Typical regression

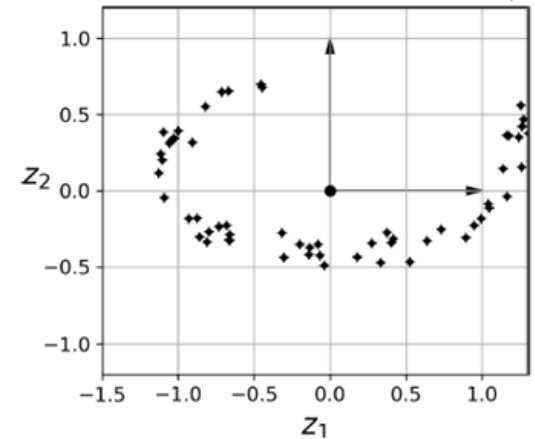
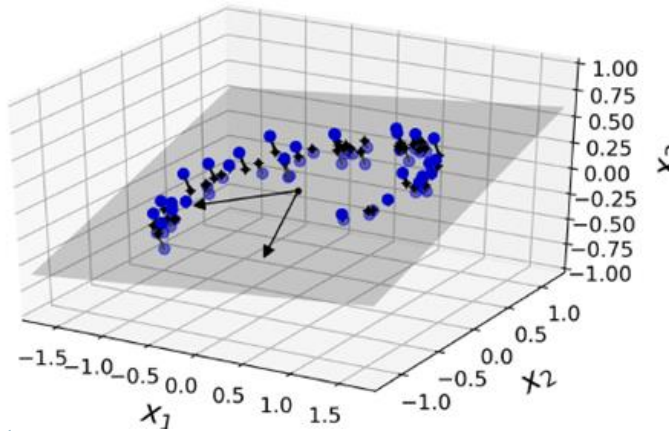
But we do not
have y value



Let's Create statistics-based
Probabilistic Model

Predicting the probability of acceptance

Dimensionality Reduction



Predicting the probability of acceptance

Dimensionality Reduction – Feature Selection

3] :

type	다익스트라	다이나믹 프로그래밍	그래프 탐색	구현	파싱	수학	깊이 우선 탐색	너비 우선 탐색	그래프 이론	그리디 알고리즘	두 포인터	문자열	백트래킹	자료 구조	시뮬레이션
다익스트라	1.000000	0.988020	0.993974	0.920017	0.989428	0.945196	0.987721	0.992501	0.994437	0.990049	0.999600	0.957273	0.963842	0.992050	0.977442
다이나믹 프로그래밍	0.988020	1.000000	0.998983	0.969471	0.999956	0.984260	0.999998	0.999475	0.998779	0.999905	0.991992	0.990433	0.993419	0.999586	0.998326
그래프 탐색	0.993974	0.998983	1.000000	0.957429	0.999363	0.975290	0.998894	0.999919	0.999991	0.999508	0.996678	0.983204	0.987244	0.999867	0.994703
구현	0.920017	0.969471	0.957429	1.000000	0.967122	0.997546	0.969942	0.961020	0.956176	0.966009	0.930737	0.994033	0.991177	0.962017	0.982029
파싱	0.989428	0.999956	0.999363	0.967122	1.000000	0.982554	0.999936	0.999736	0.999200	0.999991	0.993136	0.989091	0.992297	0.999813	0.997738
수학	0.945196	0.984260	0.975290	0.997546	0.982554	1.000000	0.984598	0.978019	0.974329	0.981737	0.954056	0.999231	0.998024	0.978769	0.992833
깊이 우선 탐색	0.987721	0.999998	0.998894	0.969942	0.999936	0.984598	1.000000	0.999411	0.998682	0.999877	0.991747	0.990697	0.993637	0.999529	0.998436

Predicting the probability of acceptance

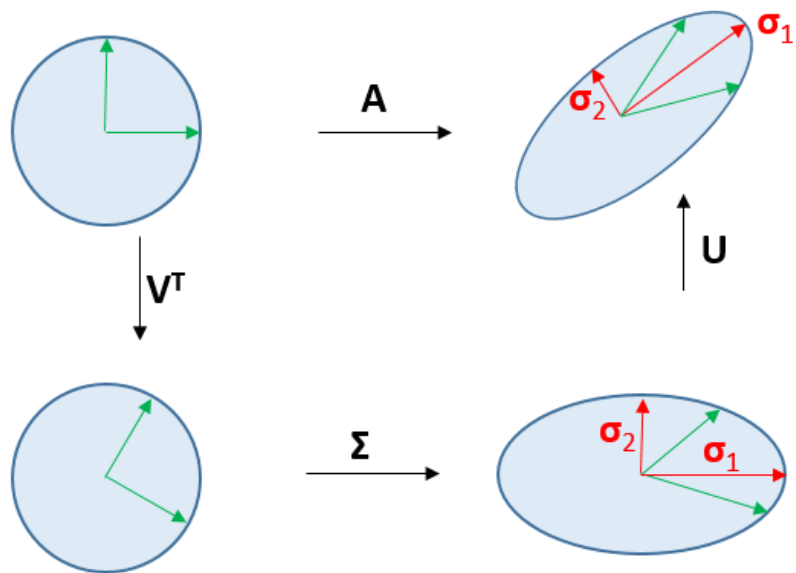
Dimensionality Reduction – Feature Selection

Select the correct ratio of each company's Top 5 algorithm

user	구현	브루트포스 알고리즘	문자열	자료 구조	수학
1223hyo	0.51428574	0.0	0.0	0.0	0.5
1998phy	0.75	0.5	0.5	0.25	0.45454547
1dillum0	0.5652174	0.37209302	0.39130434	0.39215687	0.6
1x2x257	0.0	0.0	0.0	0.0	0.83333333
20201785	0.71428573	1.0	0.0	0.0	0.5
2david2	0.80487806	0.0	0.54545456	0.0	0.61764705
4vm89092n7890	1.0	0.0	1.0	0.0	1.0
5gkfka5	0.83333333	0.33333334	0.0	0.26666668	0.6
aa4060	0.33333334	0.0	0.0	0.0	0.3
aaabbb4202	0.0	0.0	0.0	0.0	0.0
chuyee0710	0.66035106	0.66666671	0.71428573	0.76170501	0.71207121

Predicting the probability of acceptance

Dimensionality Reduction – Truncated Singular Value Decomposition (5D -> 2D)



Full SVD

$$A = U \Sigma V^T$$

Truncated SVD

$$A' = U_t \Sigma_t V_t^T$$

Predicting the probability of acceptance

Dimensionality Reduction – Truncated Singular Value Decomposition (5D -> 2D)

```
from pyspark.mllib.linalg import Vectors
from pyspark.mllib.linalg.distributed import RowMatrix, DenseMatrix
from pyspark.ml.feature import VectorAssembler
```

```
mat = RowMatrix(temp_rdd)
svd = mat.computeSVD(2, computeU=True)
```

(m=NumOfUser, n=5(top5 correct))

(m=NumOfUser, t=2(score1, score2))

Truncated SVD
using Spark MLlib

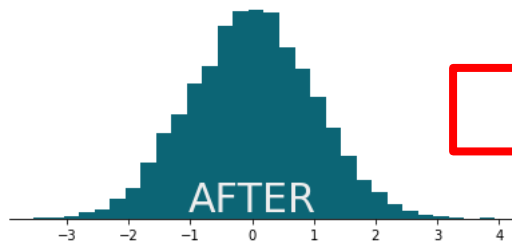
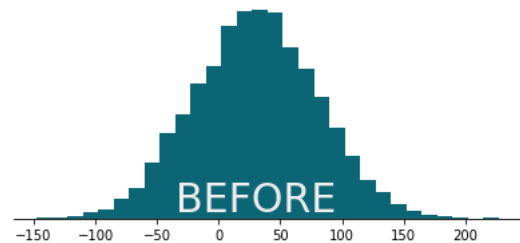
vectored_score1	vectored_score2
[-0.0041346723130510875]	[-1.8365840019368479E-4]
[-0.004060063896791902]	[-4.608305899533307E-4]
[-0.004966066530773475]	[-9.107137065034617E-4]
[0.0]	[0.0]
[-0.0026479857131488493]	[0.004824286092871272]
[-0.003262813488919546]	[0.005989079767576516]
[-0.0035749415369504337]	[0.0065604114561784305]
[-0.003624507813647184]	[0.004064016054085769]
[-0.005084874603735277]	[-0.003433330878991991]
[-0.005174017866012666]	[-0.002254957141646251]

Predicting the probability of acceptance

Standardization

```
from pyspark.ml.feature import StandardScaler
```

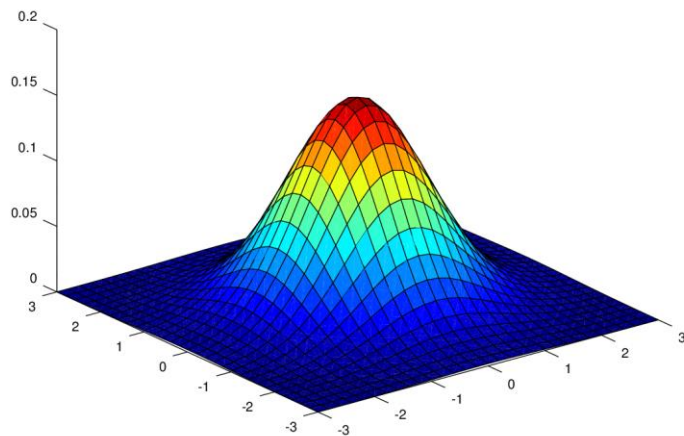
$$z = \frac{x - \mu}{\sigma}$$



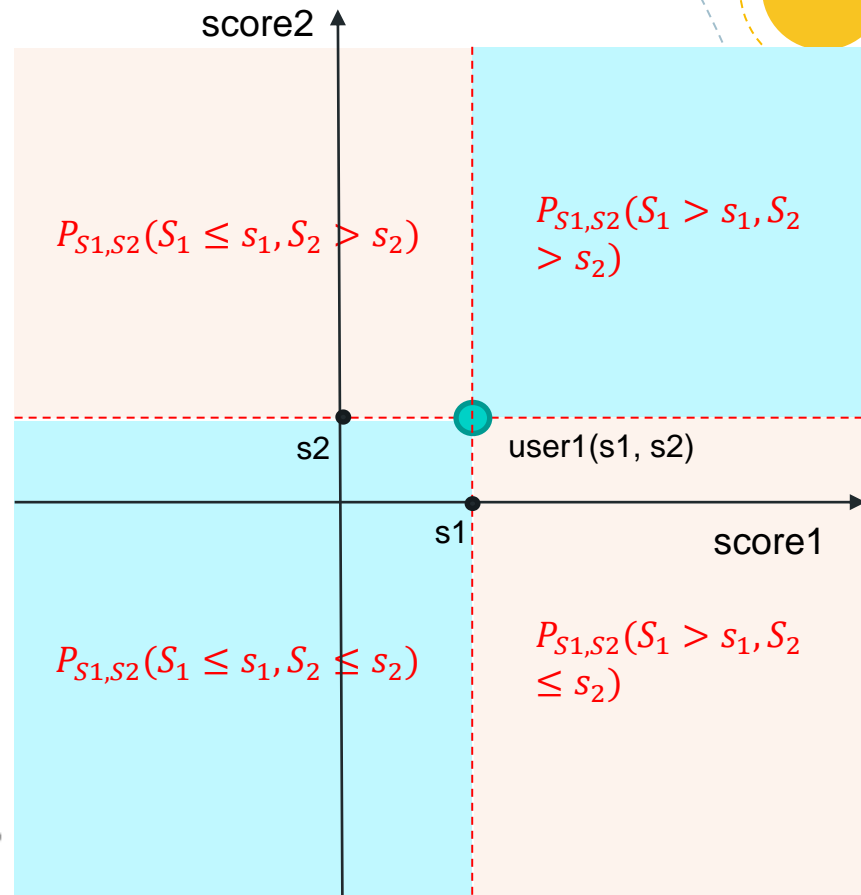
summary	scaled_score1	scaled_score2
nt	64337	64337
n	2.687631130613707E-9	-8.237840748820714E-10
dev	1.000000000811591	0.9999999997615333
	-2.6177993	-3.6584785
	2.1589222	1.8754575

Predicting the probability of acceptance

Bivariate Standard Normal Distribution



$$P(x_1, x_2) = \frac{1}{2\pi\sigma_1\sigma_2\sqrt{1-\rho^2}} \exp\left[-\frac{z}{2(1-\rho^2)}\right],$$



Predicting the probability of acceptance

Binomial Distribution Formula

If Company C have n applicants, and choose $(n-k)$ applicants

The probability that k applicants are **inferior** to User A and $(n-k)$ are **superior** to User B

$$B(n, p) = \binom{n}{k} p^k (1 - p)^{n-k}$$

Then we can simply estimate the User A's **probability of acceptance** for Company C

$$P(\text{Acceptance}) = \sum_{i=k}^n \binom{n}{i} p^i (1 - p)^{n-i}$$

Predicting the probability of acceptance

Bivariate Standard Normal Distribution

$$Score2 - s_2 = -(Score1 - s_1)$$

$$Score2 = -Score1 + (s_1 + s_2)$$

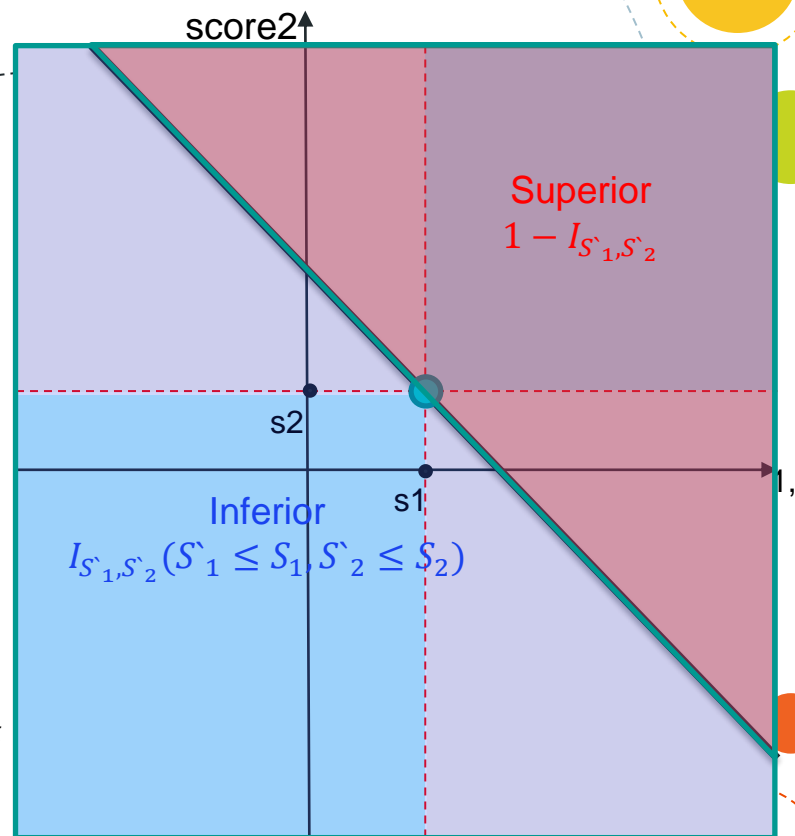
$$Score1 + Score2 = (s_1 + s_2)$$

```
test_df = stat_df.withColumn("Z", F.col("scaled_score1") + F.col("scaled_score2")).select("Z").describe()
```

summary Z	
count	64337
mean	4.713122354412475E-9
stddev	1.6222895088638696

$$Score1 + Score2 \leq (s_1 + s_2)$$

Assume $Z = Score1 + Score2$ and $Score1, Score2$ are independent
 then Z is also Normal distribution with $\sqrt{1^2 + 1^2}stddev$



Predicting the probability of acceptance

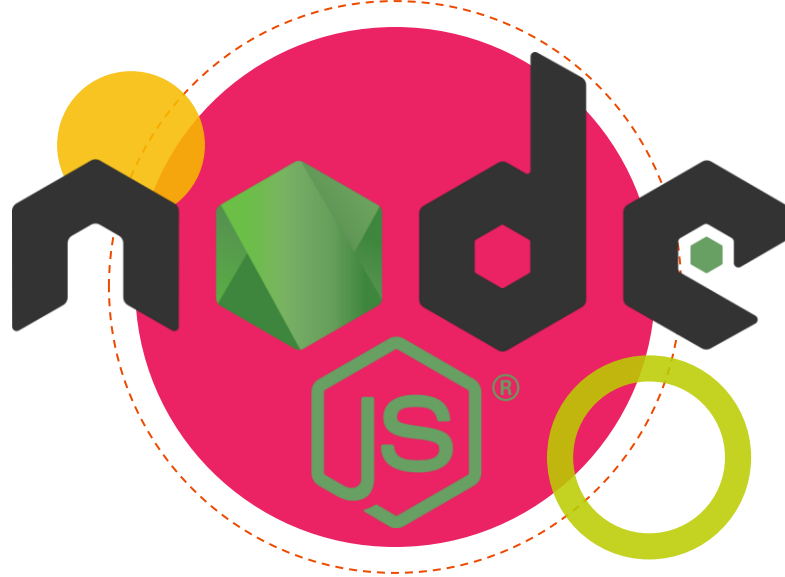
Calc each user's Probability of acceptance

$$P(\text{Acceptance}) = \sum_{i=k}^n \binom{n}{i} p^i (1-p)^{n-i}, p = N(\text{score1} + \text{score2}; 0, \sqrt{2})$$

```
from scipy.stats import norm
square_root_two = 2 ** (1/2)
inferior_cdf_udf = F.udf(lambda x, y: norm.cdf((x+y) / square_root_two).tolist(), FloatType())
```

user	scaled_score1	scaled_score2	inferior	superior	acceptance_probability_line
beth_shan	-0.33633533	-0.19340631	0.35398498	0.64601505	7.9419286E-16
black_203	-0.29130945	-0.26445845	0.34716445	0.65283555	2.3850674E-16
dongkum0417	-0.83807826	-0.3797844	0.19457525	0.80542475	5.6589176E-33
bokunoeiyuu20	2.1589222	-0.14632612	0.9226494	0.07735062	1.0
ty0603ty	0.5608738	1.0903629	0.8785161	0.12148392	0.99987984
dozinguy	0.18982783	1.3889537	0.86786747	0.13213253	0.999541
dntlr03	0.0014598591	1.5354125	0.86142254	0.13857746	0.9990533
ech913	-0.028453182	0.89547026	0.7300862	0.2699138	0.37440872
byj9402	-0.9097784	-1.0264486	0.085481	0.914519	0.0
tiaeldl	-0.963576	-0.72437656	0.11632454	0.88367546	0.0

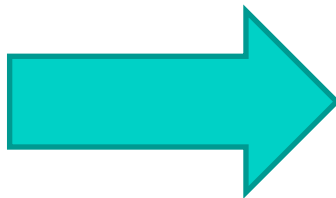
Implementation



The Process we Visualized

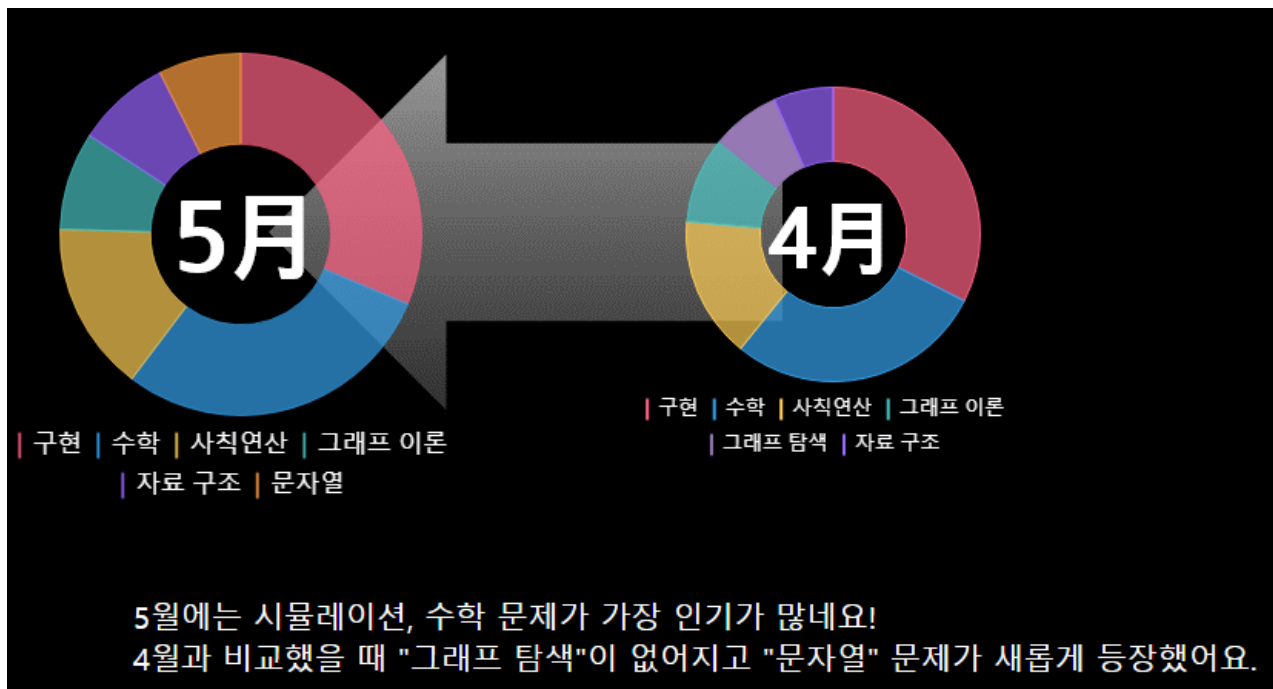
BackEnd - Trend for Month

```
Client
GET /trend
data : {
  "cur": "CURRENT MONTH",
  "bef": "BEFORE MONTH"
}
```



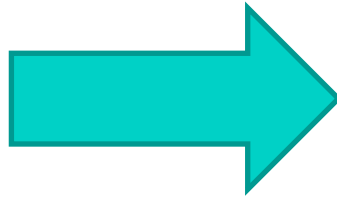
```
Server
response
[
  {
    "type": [
      "구현",
      "수학",
      "사칙연산",
      "그래프 이론",
      "자료 구조",
      "문자열"
    ]
  },
  {
    "type": [
      "구현",
      "수학",
      "사칙연산",
      "그래프 이론",
      "그래프 탐색",
      "자료 구조"
    ]
  }
]
```

FrontEnd - Trend for Month



BackEnd – Types of Problems

```
Client
GET /analysis
data : {
  "name": "user id"
}
```

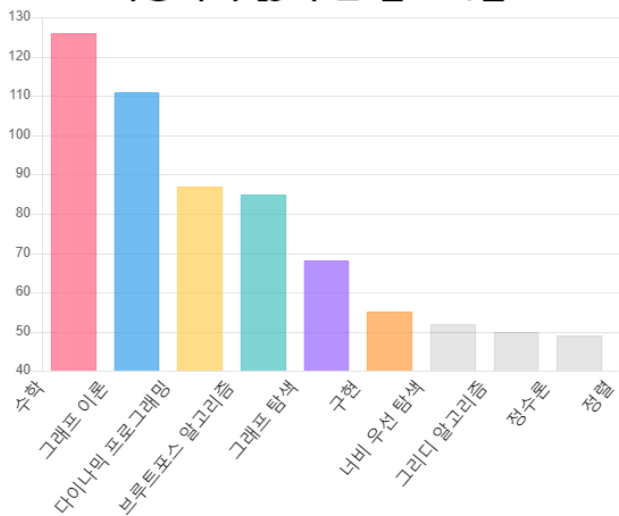


```
Server
Response
[
{
  "type": [
    "수학",
    ...(생략) 10 data
  ],
  "try": [
    "120",
    ...(생략) 10 data
  ],
  "ans": [
    "0.35",
    ...(생략) 10 data
  ],
},
{
  "type": [
    "구현",
    ...(생략) 10 data
  ],
  "try": [
    "85672",
    ...(생략) 10 data
  ],
  "ans": [
    "0.65",
    ...(생략) 10 data
  ],
},
]
```

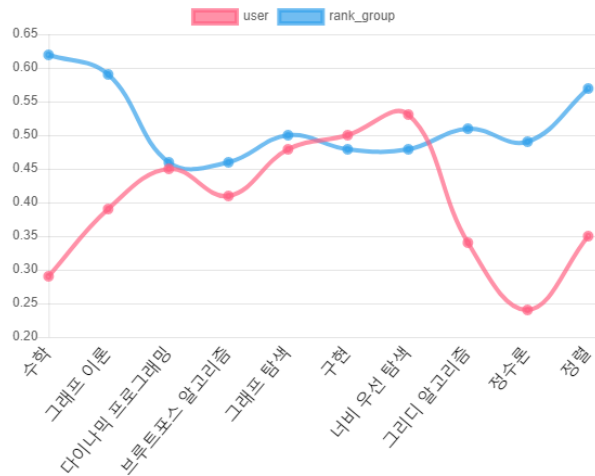
```
{
  "user_ans": "39",
  "tier_ans": "52",
  "diff": "-13"
  "weak": [
    "정확도"
  ],
  "strong": [
    "유형",
    "정답률"
  ]
},
```

FrontEnd - Types of Problems

사용자가 많이 푼 알고리즘



많이 푼 알고리즘 정답률



FrontEnd - Types of Problems

사용자 정답률



| 정답률 | 오답률

동일 티어 정답률

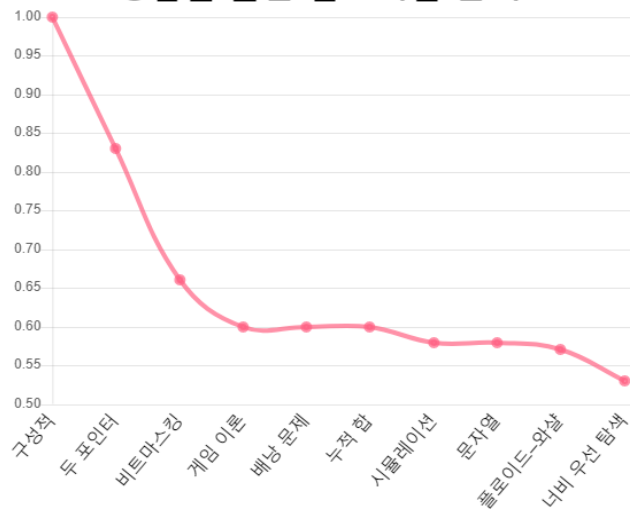


| 정답률 | 오답률

- 13%

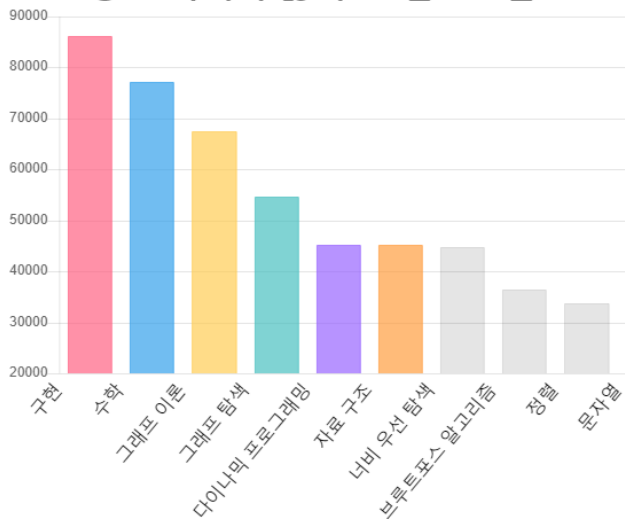
정확도가 필요할 시점입니다!
같은 수준의 사람들 보다 현재 정답률이 낮습니다.

정답률 높은 알고리즘 순위



FrontEnd - Types of Problems

동일 티어가 많이 풀 알고리즘

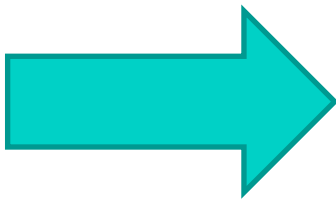


주의

정확도가 동일 티어 보다 떨어집니다.
알고리즘 유형이 동일 티어와 유사합니다.
상위 알고리즘의 정답률이 높습니다.

BackEnd – Study Group

```
Client
GET /study
data : {
  "name": "user id"
}
```



```
Server
response
{
  "weak": ["자료구조", "완전탐색"],
  "level": "골드 2",
  "user_name": [
    "0321minji",
    "054679860",
    "0913vision",
    "0h328",
    "0xe82de"
  ]
}
```

mjoo1106

“알고리즘 능력”



“취약 알고리즘”

자료구조
완전탐색

클릭하면 자동으로
쪽지가 전송됩니다.

0321minji

054679860

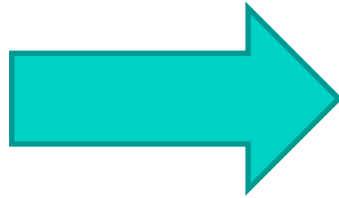
0913vision

0h328

0xe82de

BackEnd - Enterprise Acceptance Prediction Rate

```
Client
GET /company
data : {
  "name": "user id",
  "company": "회사"
}
```



```
Server
response

{
  "percent": 0.66,
  "weak": [
    "구현",
    "자료구조"
  ]
}
```

FrontEnd - Enterprise Acceptance Prediction Rate



합격 가능성



상위 35% 이내



구현, 브루트포스 문제를
풀어보세요!



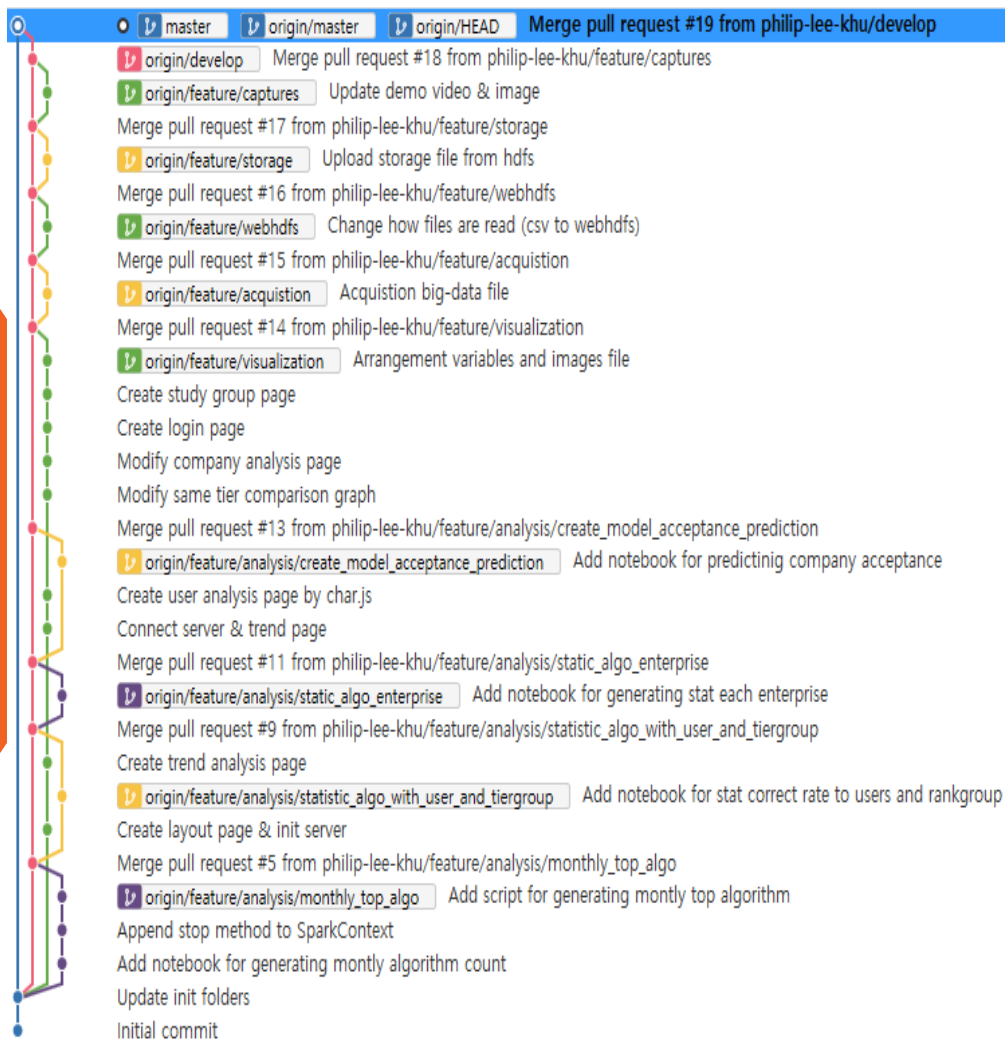
합격 가능성



하위 10% 이내



전반적인 코딩 공부가 필요합니다.
구현, 자료구조, 문자열 추천



A decorative graphic featuring a large, light blue dashed circle that frames the central text. Various smaller circles in teal, yellow, green, and orange are scattered around the perimeter of the dashed circle, some solid and some hollow.

<https://bigdata-server.herokuapp.com/>

You can Access with this Address!

Thanks!



Any questions?