

신라대학교

머신러닝 100분 정복

마이크로소프트 Azure Machine Learning

김대우 부장

Microsoft

daewoo.kim@microsoft.com

- **Machine Learning overview**
- **기본 예측 모델 생성 및 예측 수행**
- **모바일 게임에서의 고객 이탈**
- **이탈 예측 모델링 리서치**
- **데이터 탐색**
- **Azure ML Studio 모델링**
- **실제 게임에 적용하기**
- **적용 결과**
- **이후 진행 상황**

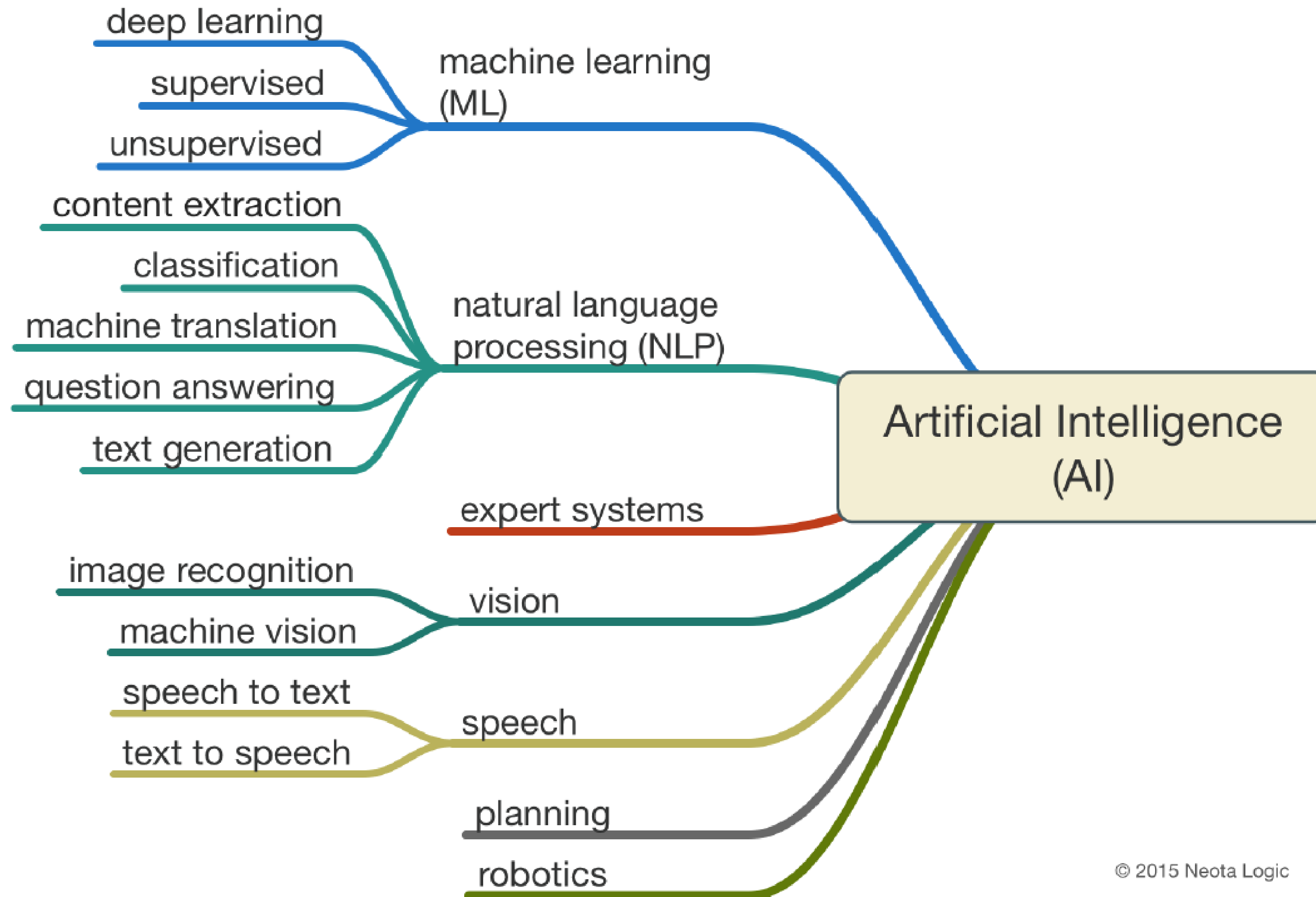


“

머신러닝? 인공지능?
인지서비스?

”

머신러닝과 인공지능, 인지서비스(Cognitive Service)



Video :

Starship commander

Microsoft & Human Interact:

Players control “Voice”

지도학습

Video

넥슨 - 마비노기 듀얼

Azure Machine Learning 기반

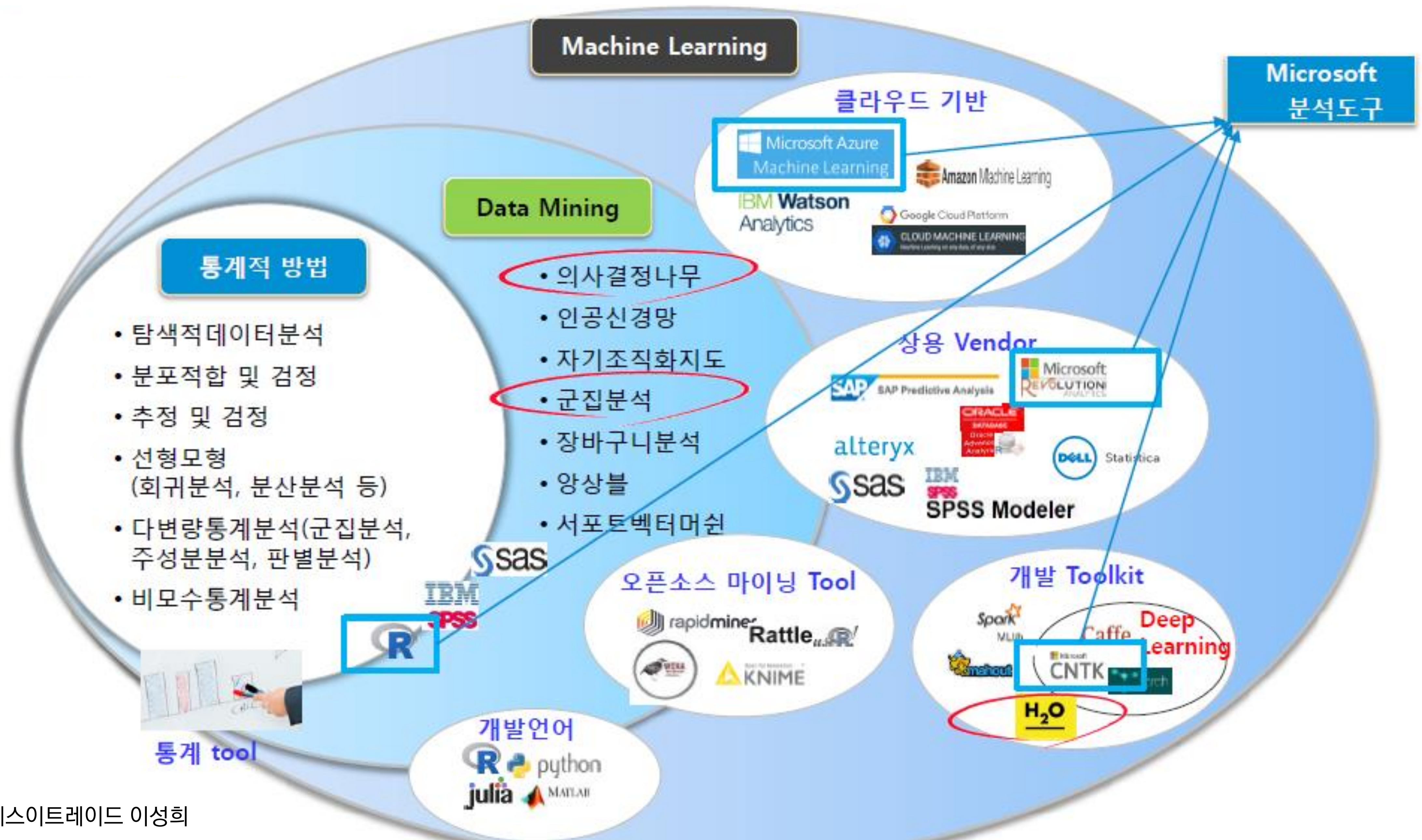
재방문 유도 시스템

“

R, SAS, Python?
Tensorflow, CNTK, H2O, keras?
Scikit, Caret, fastcluster, party?
Azure ML, Google ML, Amazon ML?

”

Machine Learning의 영역은 통계적 방법, 데이터마이닝 등 기존 분석기법들을 포괄하고 있음



예측 분석을 이용한 기술은 향후 모든 산업에 필요 충분 조건



이탈 예측 분석



소셜 네트워크 분석



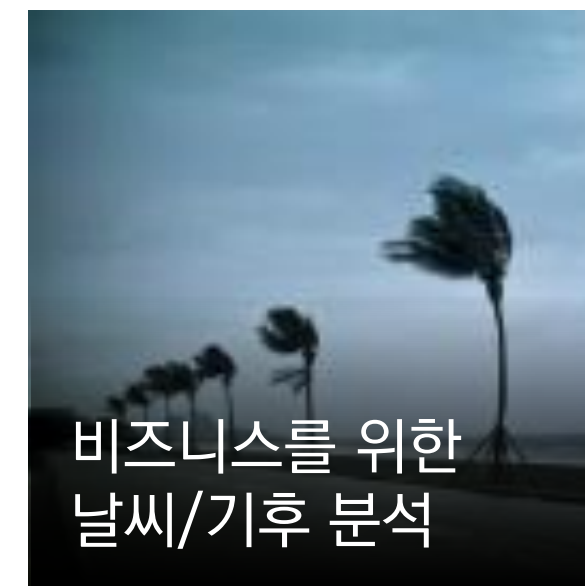
추천 엔진



지역 기반 추적과 서비스



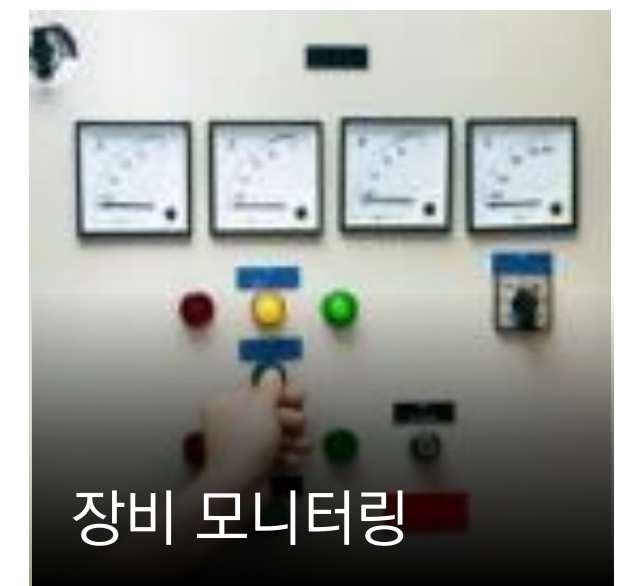
IT 인프라 및 웹
어플리케이션 최적화



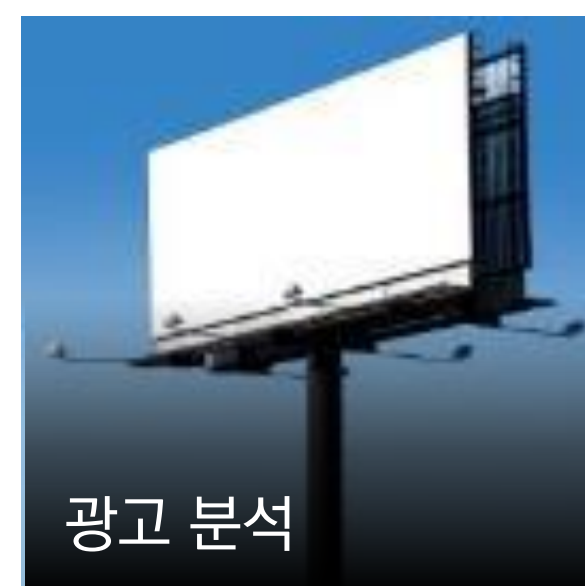
비즈니스를 위한
날씨/기후 분석



문서 탐색



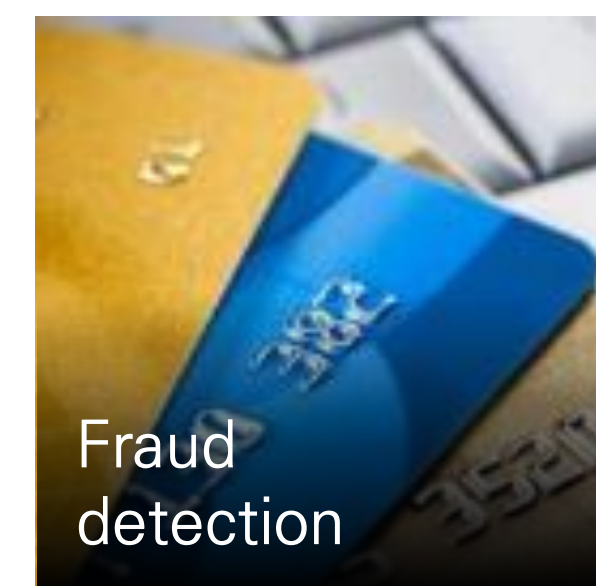
장비 모니터링



광고 분석



가격 분석



Fraud
detection



개인화된 상품 예측
제공

개발자가 머신러닝을 다루는 패턴

IRIS Data

(본사와 무관한 통계업계의 “Hello World”)

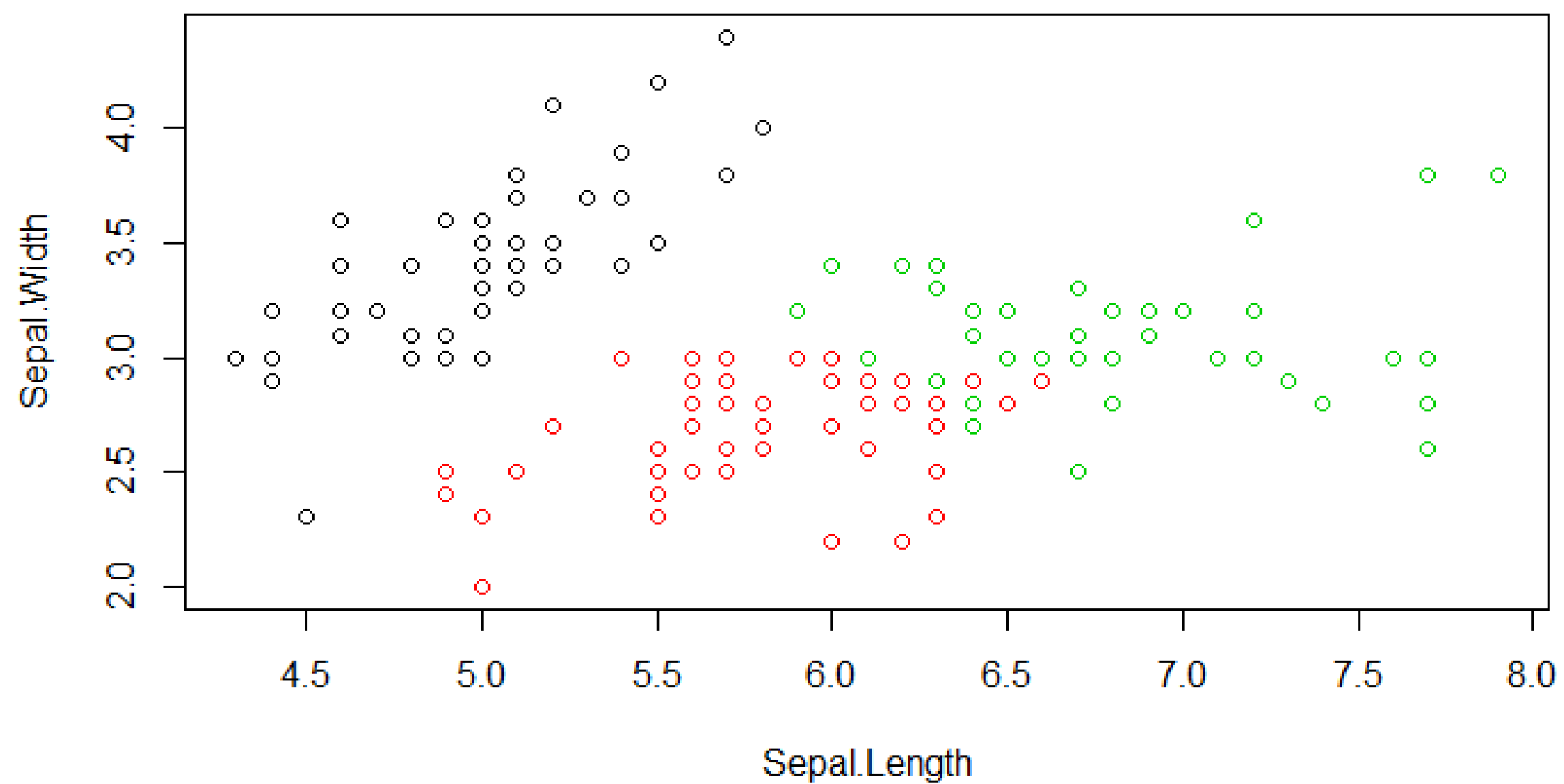
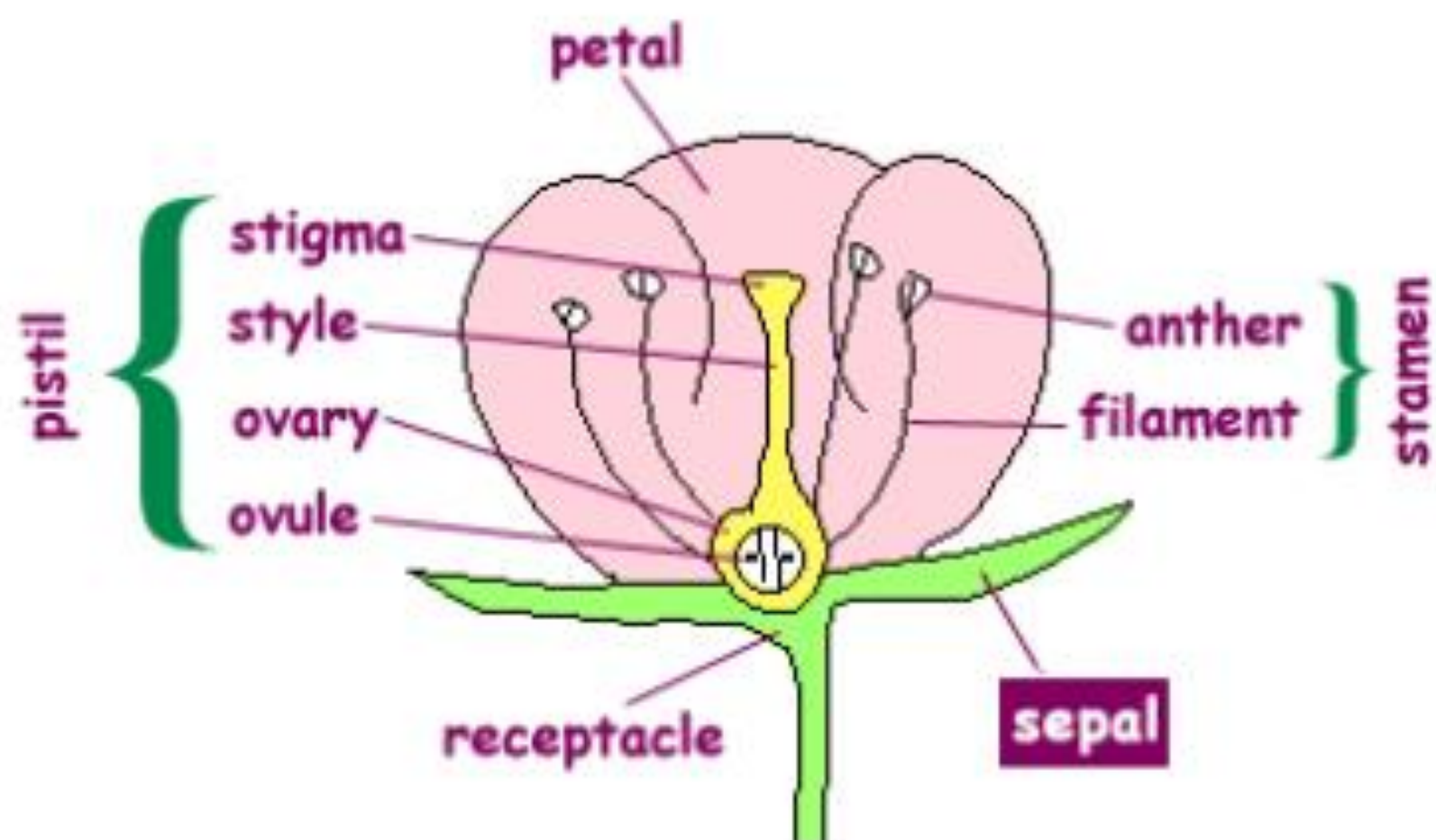
Iris setosa



Iris versicolor

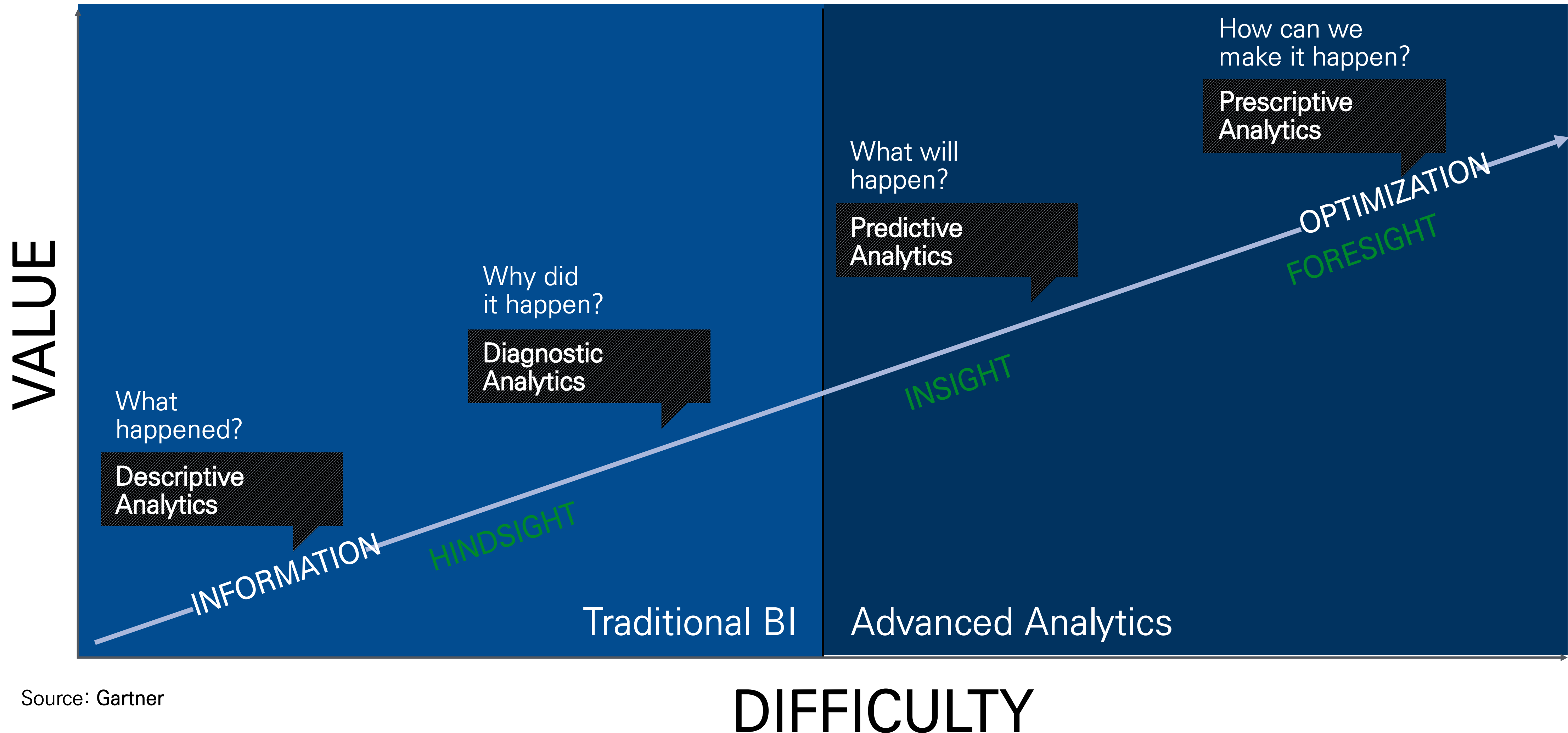


Iris virginica



Advanced Analytics

Business Intelligence와의 비교



구분	Reinforcement Learning	Machine Learning (Supervised Learning)	비고
목적함수	보상을 최대화 (또는 손실을 최소화)	오차를 최소화 (오차 = 추정 - 실제)	
산출방식	순차적으로 현재 스테이지의 보상과 총 보상을 산출하여 "총 보상"이 최대화 되도록 함	실제 사례를 기반으로 사례와 가장 유사하게 모사하도록 함수를 구성하도록 함	
산출 방법론	Optimization	분류문제와 예측문제로 구분되며 다양한 알고리즘 존재	
데이터 구성	State 별 Action Matrix (모든 가능한 State 각각에 대한 모든 실행 가능한 Action과 확률)	State vs. Action에 대한 성공과 실패 사례	
특징	규칙기반으로의 설계가 용이함 (Heuristic 설계 용이)	- 데이터마이닝: 규칙(if/else) 기반 설계 용이 - 기계학습: 규칙 파악이 어려움	기계학습은 정확도 향상이 주 목표임
구현 방법	최적화 엔진 (Dynamic LP 등)	통계 소프트웨어 또는 기계학습 엔진	

Unsupervised Learning

Supervised Learning

Reinforcement Learning

(강화학습)

Video : [베카 러닝](#)

학습모델 / 예측모델

“

바보(머신)에게 공부할 기회를
= 학습모델

”

“

바보(머신)에게 배운거 물어볼까
= 예측모델

”

예측모델 생성 데모

API로 노출

Hands on Lab

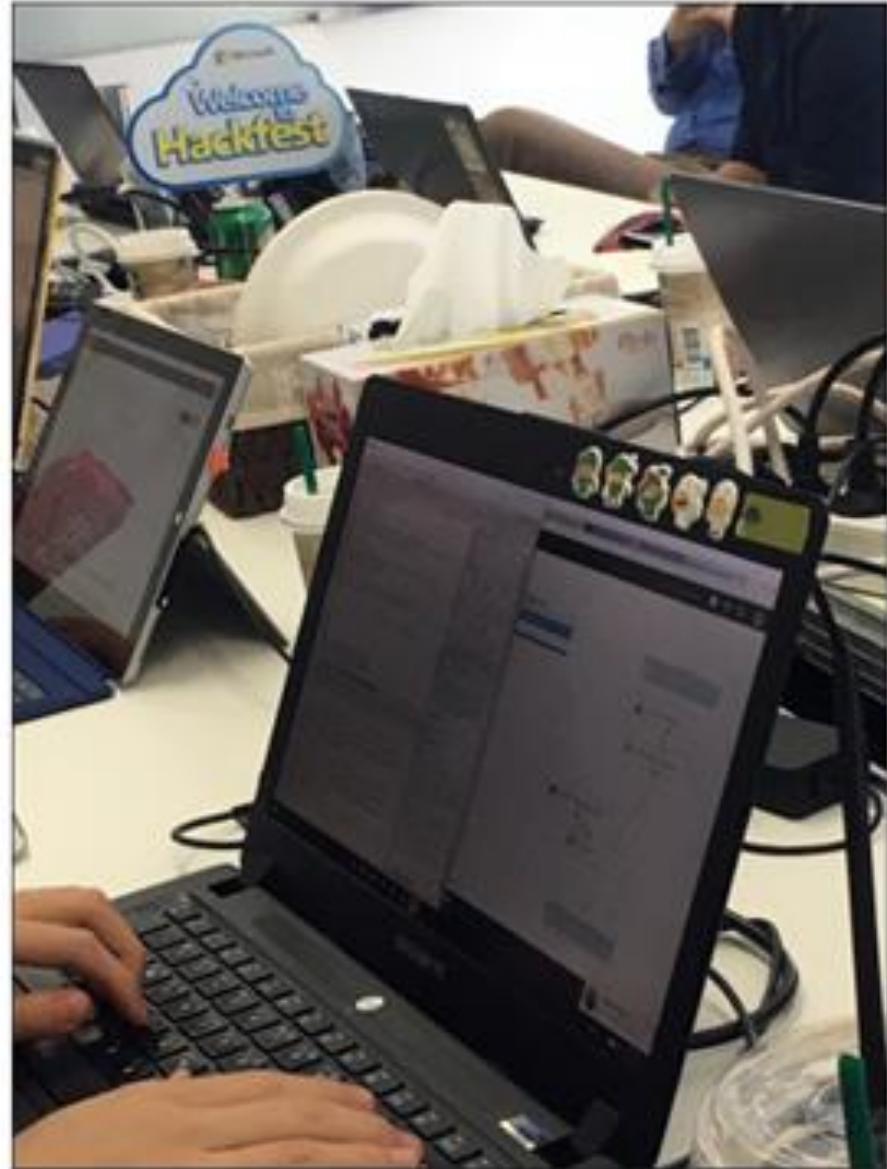
Machine Learning 모델 구성

REST API 개발자 환경 구성

Workbench를 이용한 개발

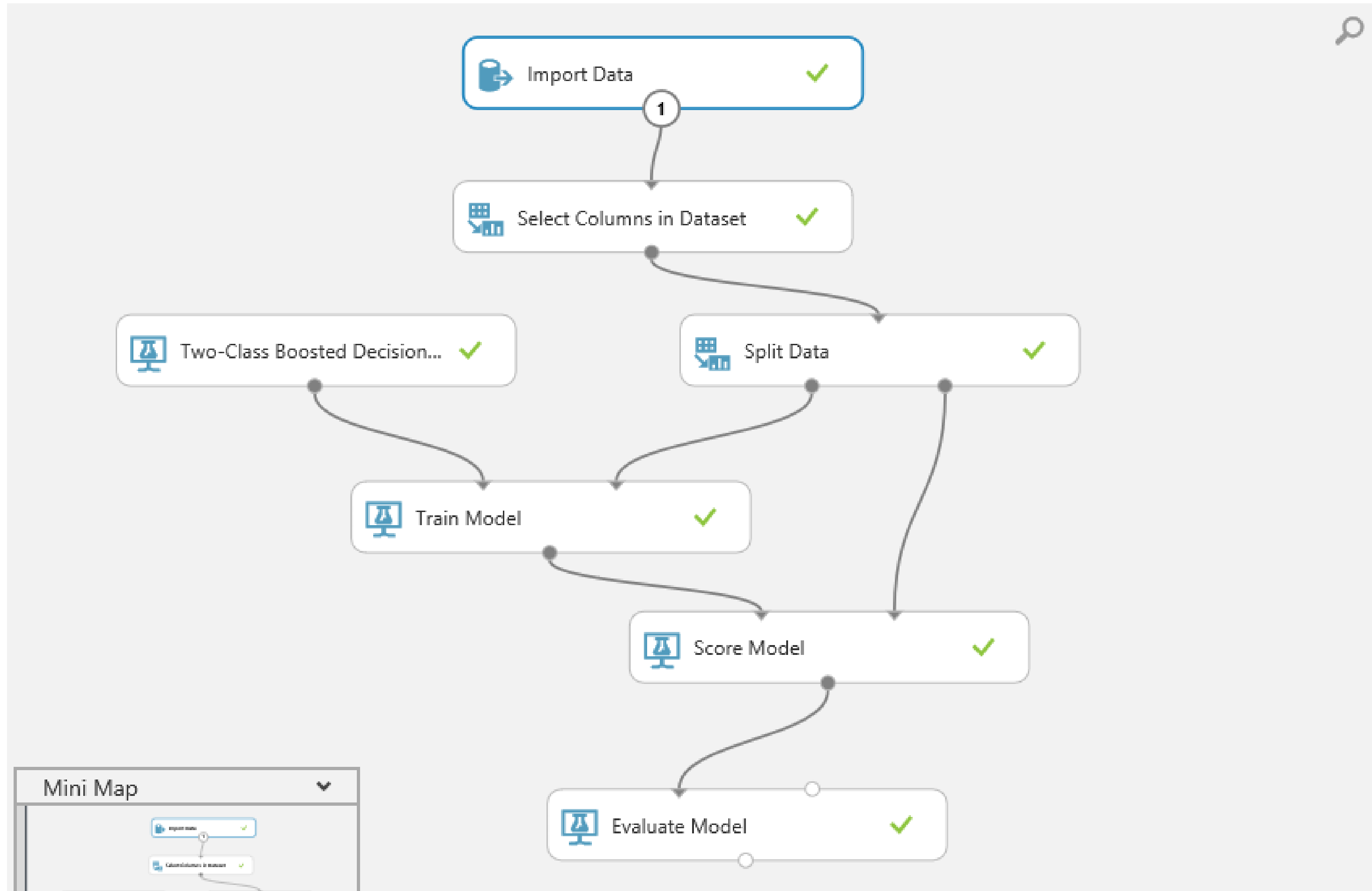
Q&A

감사합니다



Nexon 구성 사례

Azure ML Studio 모델링

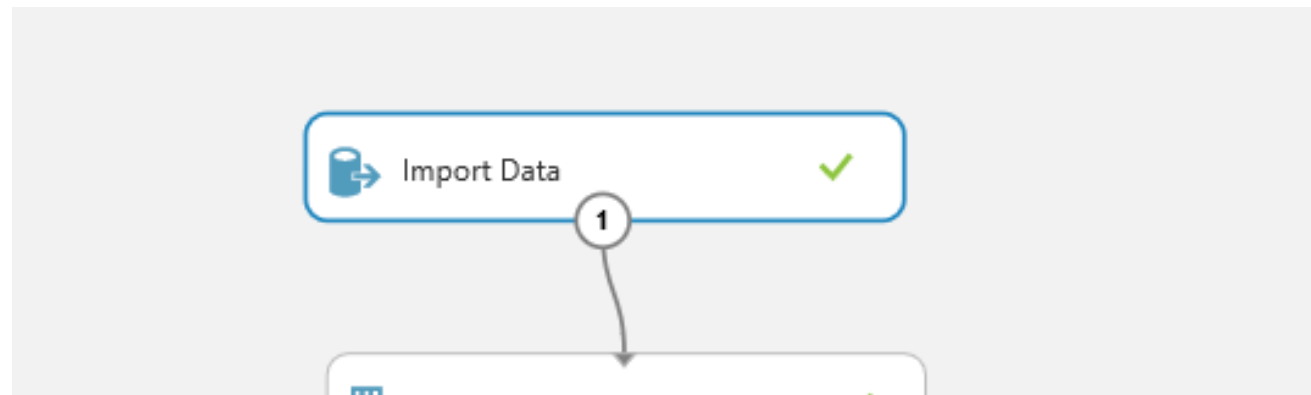


웹사이트에서 손쉽게 사용가능
<https://studio.azureml.net/>

Azure ML Studio 모델링

1. Import Data

ML 교육자료 입력 : 최근 30일간 유저 접속 데이터

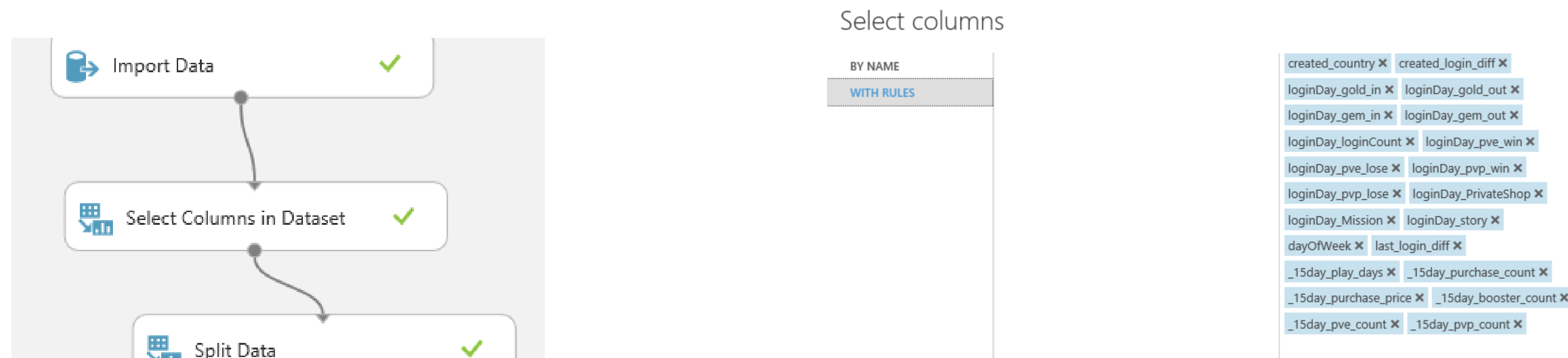


user_idx	(PK, bigint, Null이 아님)
loginDateKey	(PK, int, Null이 아님)
nextdayConnect	(bit, Null)
gold	(int, Null)
gem	(int, Null)
play_days	(int, Null)
purchase_count	(int, Null)
purchase_price	(int, Null)
booster_count	(int, Null)
pve_count	(int, Null)
pvp_count	(int, Null)
created_country	(nvarchar(50), Null)
created_login_diff	(int, Null)
loginDay_gold_in	(int, Null)
loginDay_gold_out	(int, Null)
loginDay_gem_in	(int, Null)
loginDay_gem_out	(int, Null)
loginDay_loginCount	(int, Null)
loginDay_pve_win	(int, Null)
loginDay_pve_lose	(int, Null)
loginDay_pvp_win	(int, Null)
loginDay_pvp_lose	(int, Null)
loginDay_PrivateShop	(int, Null)
loginDay_Mission	(int, Null)
loginDay_story	(int, Null)
dayOfWeek	(nvarchar(10), Null)
last_login_diff	(int, Null)

Azure ML Studio 모델링

2. Select Columns in Dataset 학습에 사용할 컬럼 지정

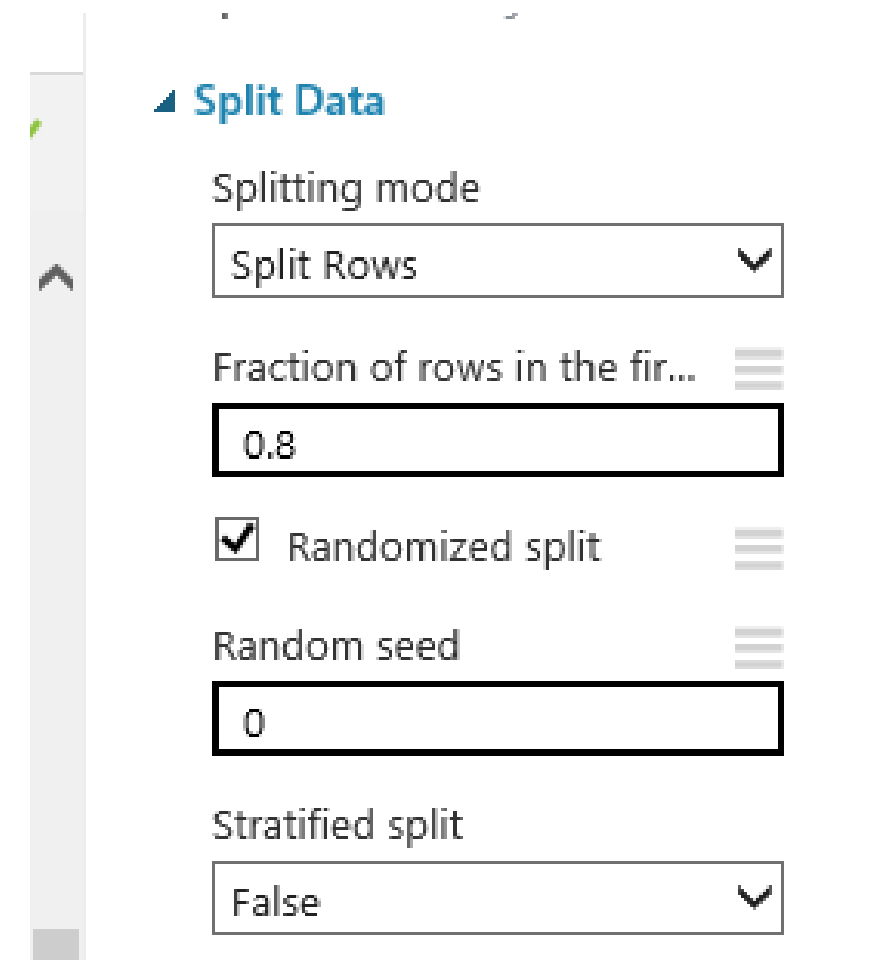
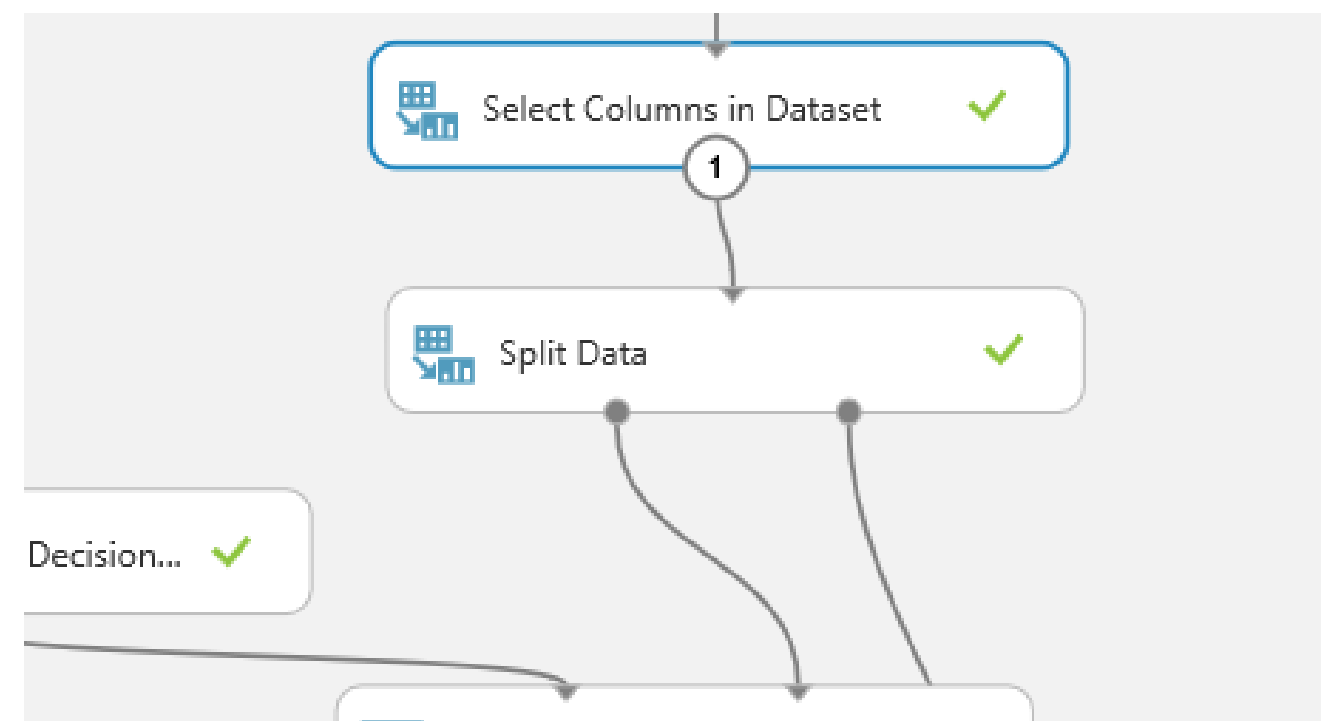
다음날접속여부, 접속일보유골드, 접속일보유보석, 가입후접속일까지접속일수, 접속일까지구매횟수, 접속일까지구매금액, 접속일까지부스터개봉횟수, 접속일까지PVE횟수, 접속일까지PVP횟수, 국가, 가입일부터접속일까지일수, 접속당일골드입수, 접속당일골드사용, 접속당일보석입수, 접속당일보석사용, 접속당일로그인횟수, 접속당일PVE승리횟수, 접속당일PVE패배횟수, 접속당일PVP승리, 접속당일PVP패배, 접속당일개인상점거래, 접속당일미션, 접속당일스토리, 접속일일자, 접속일요일, 최근로그인일자



Azure ML Studio 모델링

3. Split Data

학습 데이터를 8:2로 나눠서, 80%는 학습을 시키고 나머지 20%는 검증에 사용



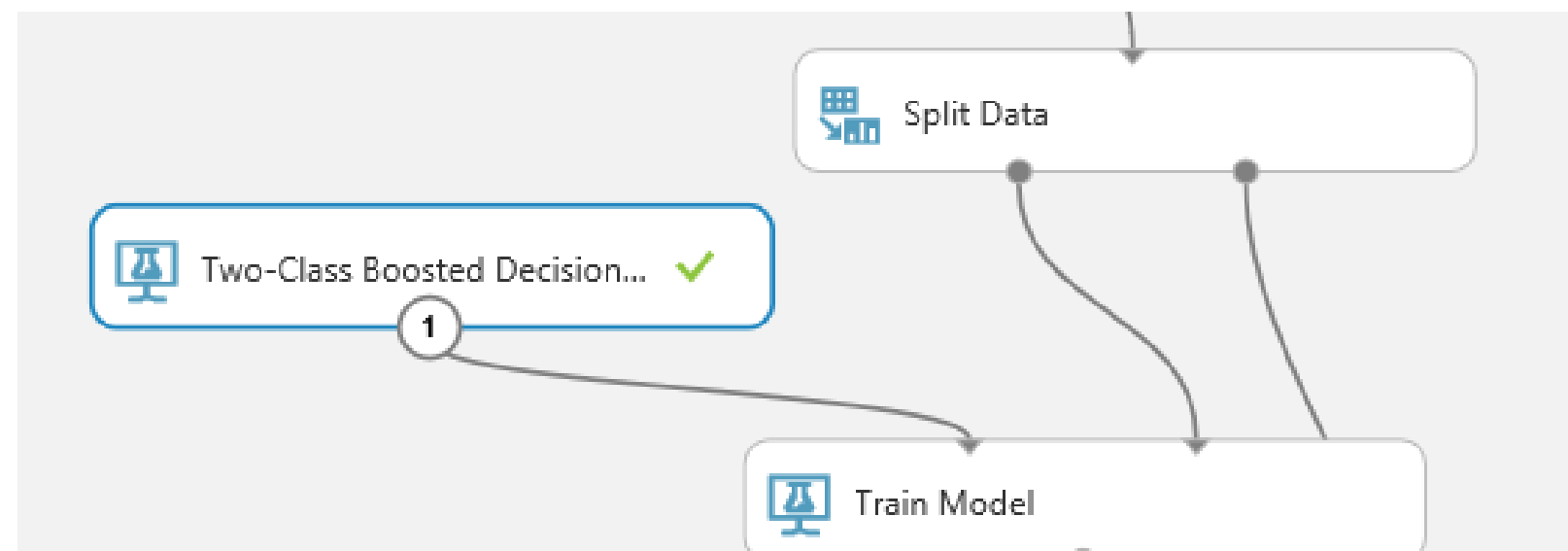
Azure ML Studio 모델링

4. Select Machine Learning Algorithm

교육에 사용될 알고리즘을 선택한다.

선택한 알고리즘은 Two-Class Boosted Decision Tree.

(<https://msdn.microsoft.com/library/en-us/Dn906025.aspx>)



Two-Class Boosted Decision...

Create trainer mode

Single Parameter

Maximum number of l...

20

Minimum number of s...

10

Learning rate

0.2

Number of trees const...

100

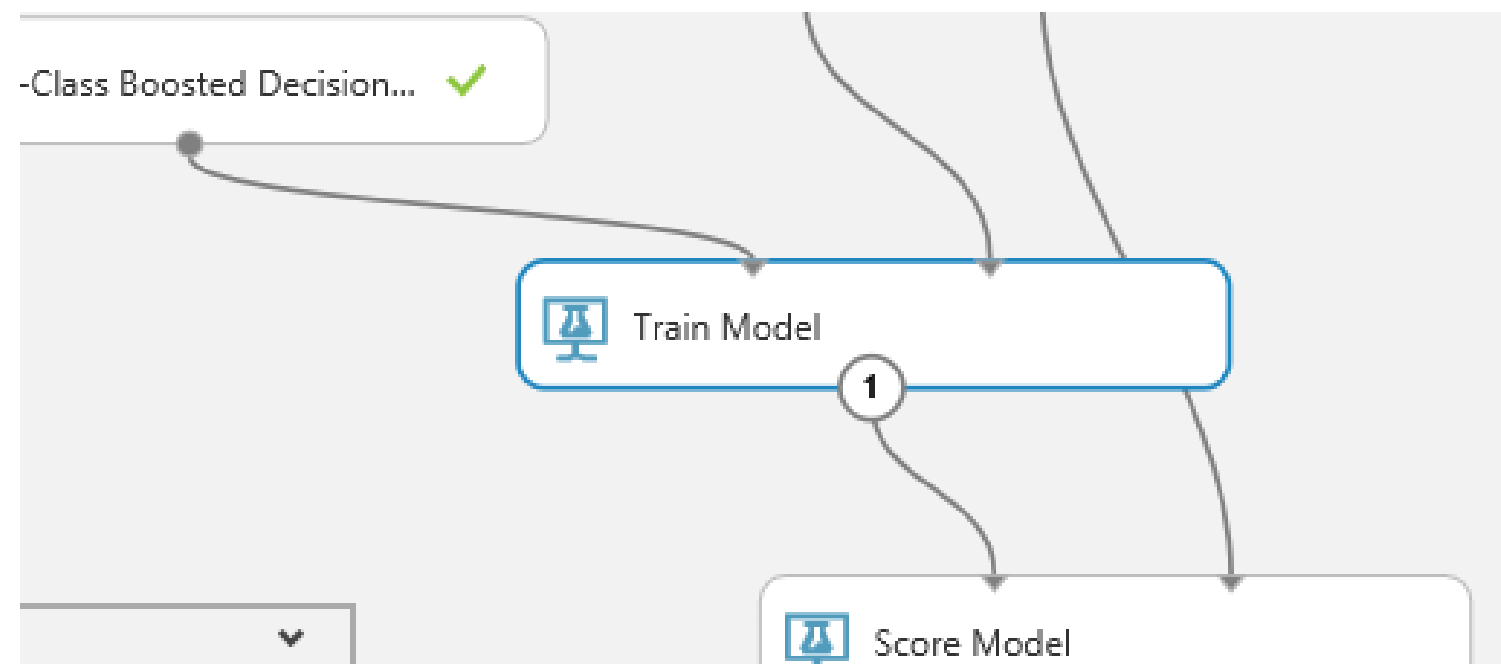
Random number seed

☒ Allow unknown ca...

Azure ML Studio 모델링

5. Train Model

예측할 컬럼을 지정하고, 데이터를 학습한다.
예측 컬럼은 nextdayConnect



Select a single column

BY NAME

WITH RULES

Include ▼

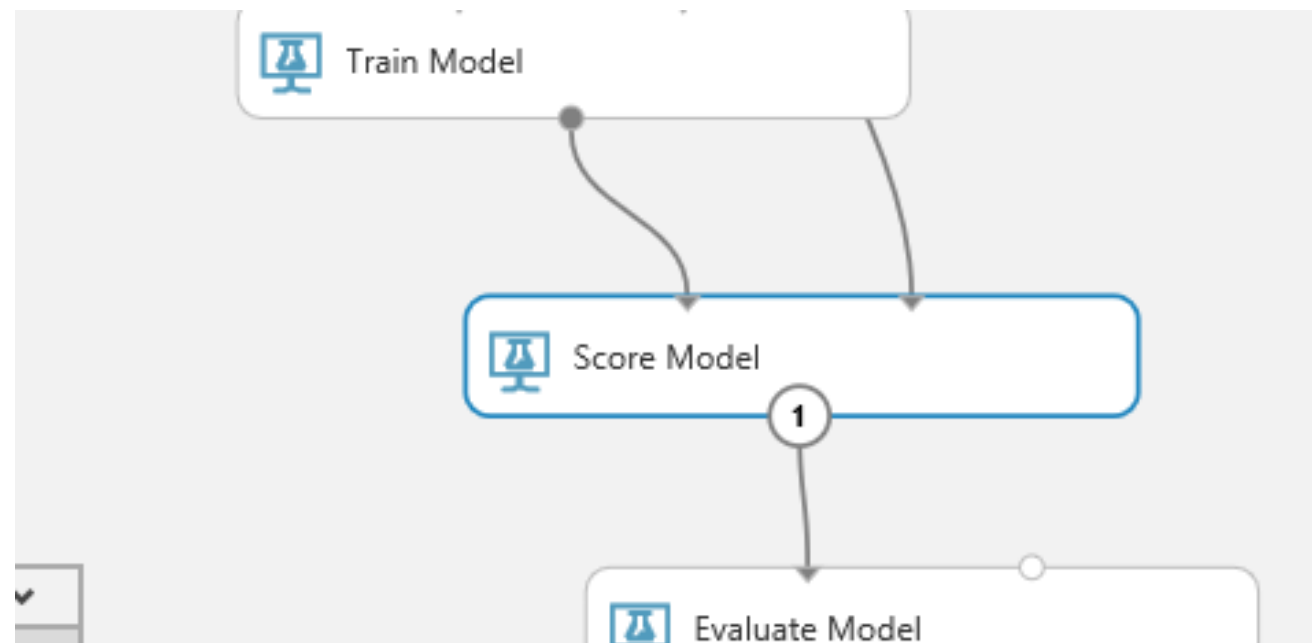
column names ▼



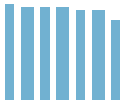


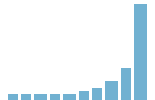
nextdayConnect ✕

Azure ML Studio 모델링

6. Score Model

학습 시킨 데이터를 수치화 시킨다. (확률화)

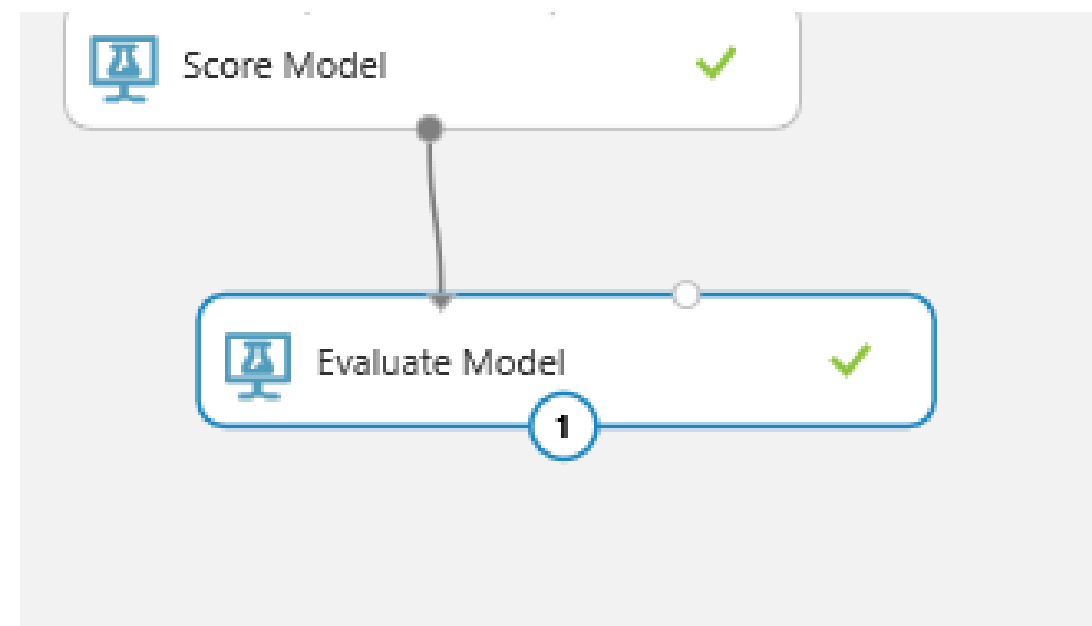


'privateShop	loginDay_Mission	loginDay_story	dayOfWeek	last_login_diff	Scored Labels	Scored Probabilities
						
0	2	Thursday	1	true	0.836902	
12	0	Friday	1	true	0.939345	
6	0	Saturday	1	true	0.960245	
0	0	Saturday	1	false	0.482898	
5	3	Friday	2	true	0.737602	
0	0	Monday	1	true	0.930039	
7	0	Monday	1	true	0.836035	

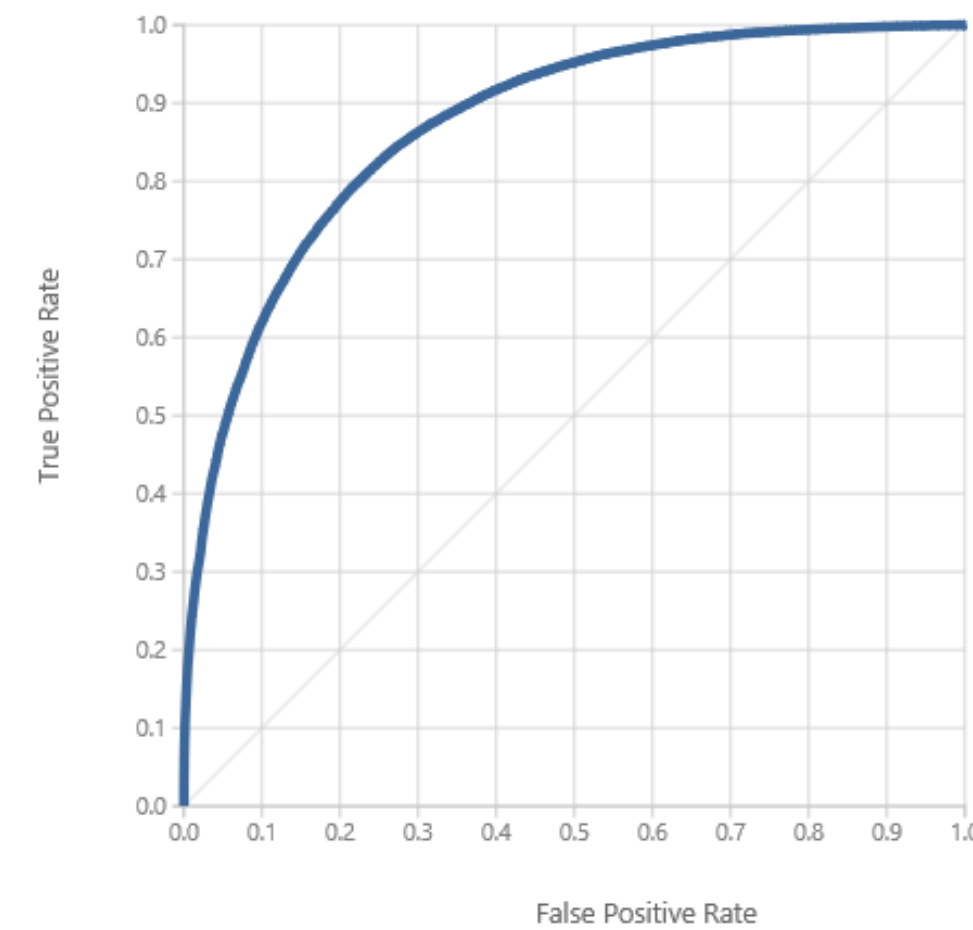
Azure ML Studio 모델링

7. Evaluate Model

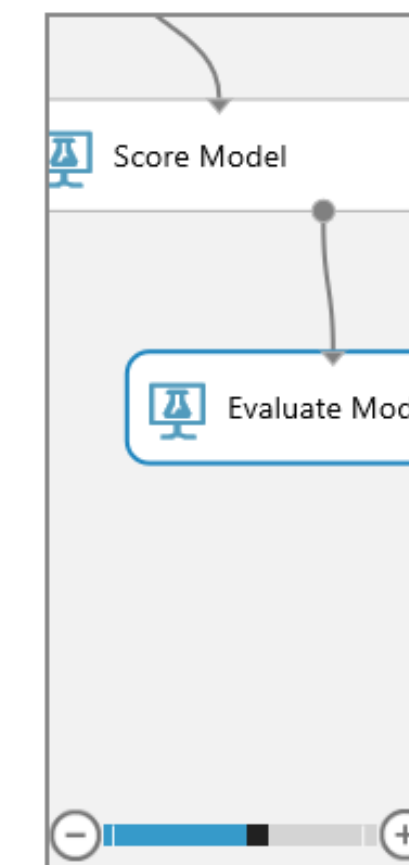
80%로 학습시킨 모델링의 결과가 정확한지 나머지 20%의 데이터로 검증한다.



ROC PRECISION/RECALL LIFT



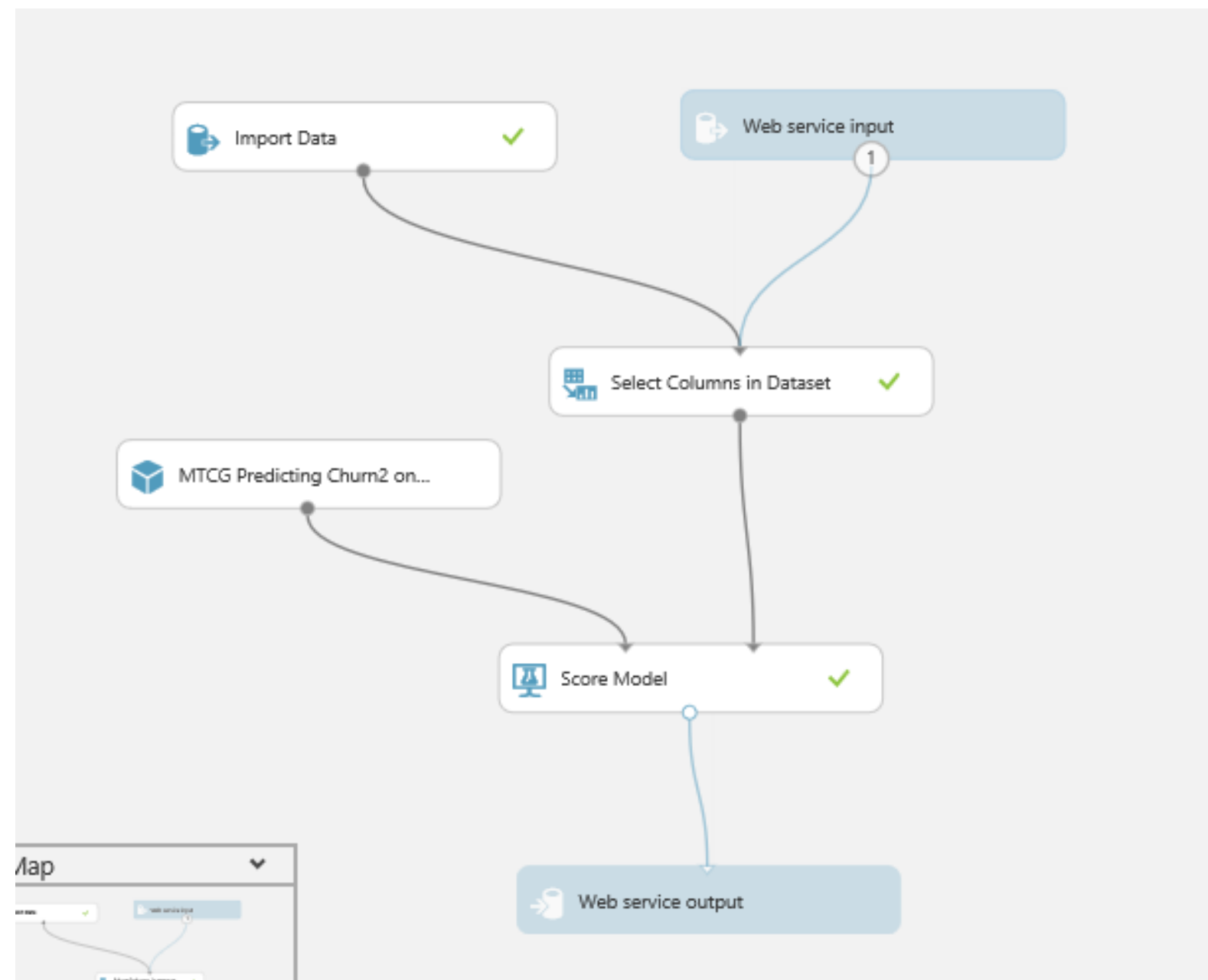
Scored dataset



True Positive	False Negative	Accuracy	Precision	Threshold	AUC
150760	9587	0.843	0.864	0.5	0.873
False Positive	True Negative	Recall	F1 Score		
23703	27848	0.940	0.901		
Positive Label	Negative Label				
True	False				

Azure ML Studio Web Service

- Web Service
만들어진 모델링을 Web Api로 배포.
Azure ML 의 가장 큰 장점.



```
import urllib2
import json

data = {
    "Inputs": {
        "input1":
            [
                {
                    'user_idx': "100001981",
                    'loginDateKey': "20161226",
                    'nextDayConnect': "false",
                    'gold': "16570",
                    'gem': "30",
                    '_15day_play_days': "1",
                    '_15day_purchase_count': "0",
                    '_15day_purchase_price': "0",
                    '_15day_booster_count': "0",
                    '_15day_pve_count': "0",
                    '_15day_pvp_count': "0",
                    'created_country': "KR",
                    'created_login_diff': "473",
                    'loginDay_gold_in': "2100",
                    'loginDay_gold_out': "0",
                    'loginDay_gem_in': "20",
                    'loginDay_gem_out': "0",
                    'loginDay_loginCount': "1",
                    'loginDay_pve_win': "0",
                    'loginDay_pve_lose': "0",
                    'loginDay_pvp_win': "0",
                    'loginDay_pvp_lose': "0",
                    'loginDay_PrivateShop': "0",
                    'loginDay_Mission': "0",
                    'loginDay_story': "0",
                    'dayOfWeek': "Monday",
                    'last_login_diff': "67",
                }
            ],
    },
    "GlobalParameters": {
    }
}

body = str.encode(json.dumps(data))

url = 'https://ussouthcentral.services.azureml.net/workspaces/a82dc1fda0a34a2cb155599614e4f203/services/4163bad846c44976916e9aacb99bce03/execute?api-version=2.0&format=swagger'
api_key = 'abc123' # Replace this with the API key for the web service
headers = {'Content-Type': 'application/json', 'Authorization': ('Bearer '+ api_key)}

req = urllib2.Request(url, body, headers)

try:
    response = urllib2.urlopen(req)
    result = response.read()
    print(result)
except urllib2.HTTPError, error:
    print("The request failed with status code: " + str(error.code))

# Print the headers - they include the request ID and the timestamp, which are useful for debugging the failure
print(error.info())
print(json.loads(error.read()))
```

Scored Labels	False
Scored Probabilities	0.224127575755119