

# Techtonic 2018

-  
Thu . Nov 15

-  
SAMSUNG SDS Tower  
West Campus B1F  
Magellan Hall /Pascal Hall



머신러닝이 쉬워진다!

# 오픈소스 AI분석 플랫폼 Brightics Studio

삼성SDS 민승재 마스터



Techtonic 2018

---

# Agenda

- Data Analytics Trend
- Brightics Studio
- Demo
- Brightics AI

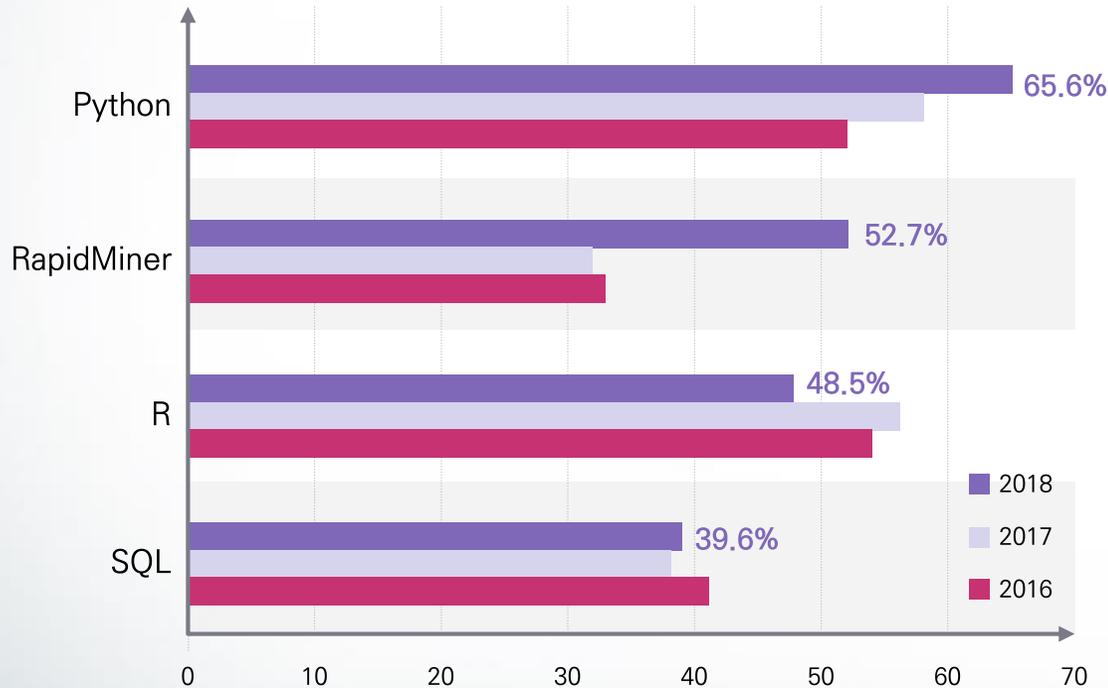
오픈소스 AI분석 플랫폼 Brightics Studio

# Data Analytics Trend

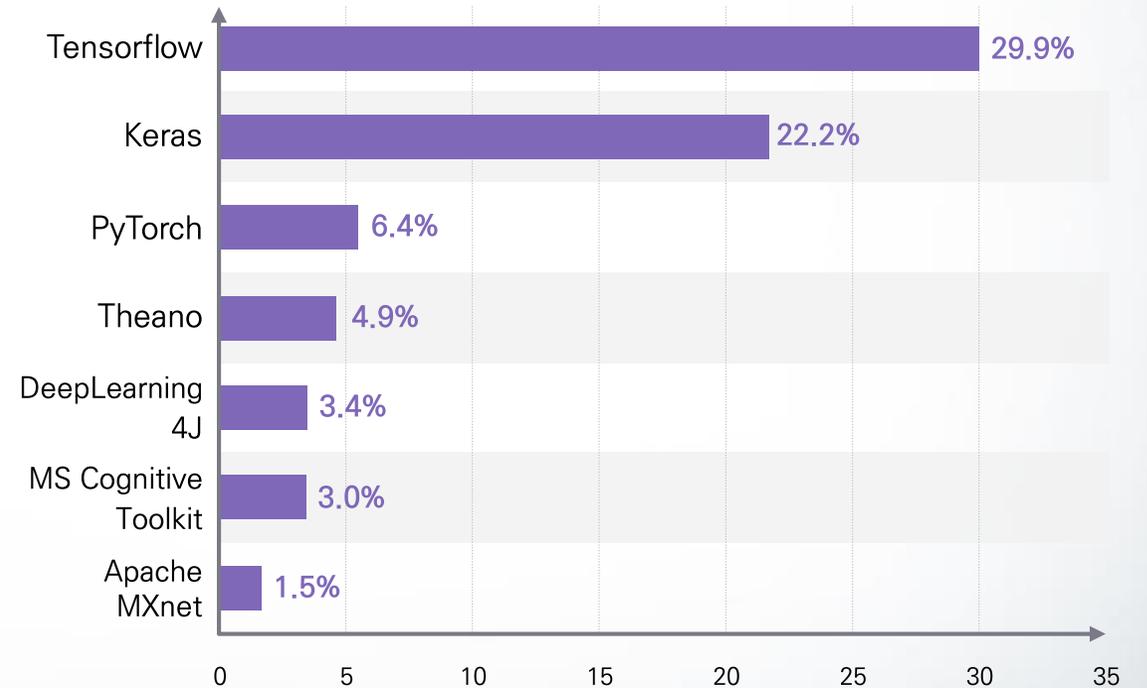
# 데이터분석 및 머신러닝 관련 S/W 선호도 조사

응답자 평균 7.0 개의 선호 S/W를 응답하였으며, 2017년의 6.75개 대비 소폭 상승됨

## Machine Learning SW Poll, 2016-2018



## Deep Learning Tools



※ Source : KDnuggets – Top Software for Machine Learning in 2018: Trends and Analysis

# 업종전문가를 위한 쉬운 Visual 분석환경

## Professional Data Scientist 用 (분석 전문가)

### R Code

```
#Load Train and Test datasets
```

```
#Identify feature and response variable(s) and values must be numeric and numpy arrays
```

```
x_train <- input_var
```

```
y_train <- target_v
```

```
x_test <- input_var
```

```
x <- cbind(x_train,y
```

```
# Train the model us
```

```
linear <- lm(y_train
```

```
summary(linear)
```

```
#Predict Output
```

```
predicted= predict(
```

### Python Code

```
import matplotlib.pyplot as plt
```

```
import numpy as np
```

```
from sklearn import datasets, linear_model
```

```
from sklearn.metrics import mean_squared_error, r2_score
```

```
diabetes = datasets.load_diabetes()
```

```
diabetes_X = diabetes.data[:, np.newaxis, 2]
```

```
# Split the data into training/testing sets
```

```
diabetes_X_train = diabetes_X[:-20]
```

```
diabetes_X_test = diabetes_X[-20:]
```

```
regr = linear_model.LinearRegression()
```

```
# Train the model using the training sets
```

```
regr.fit(diabetes_X_train, diabetes_y_train)
```

```
# Make predictions using the testing set
```

```
diabetes_y_pred = regr.predict(diabetes_X_test)
```

```
# Plot outputs
```

```
plt.scatter(diabetes_X_test, diabetes_y_test, color='black')
```

```
plt.plot(diabetes_X_test, diabetes_y_pred, color='blue', linewidth=3)
```

## Citizen Data Scientist 用 (업종 전문가)



# 오픈소스 AI분석 플랫폼의 필요성

※ SDS AI분석플랫폼인 Brightics AI 사용자들의 Feedback기반



업종 전문가

“ Coding을 할 줄 몰라도  
내 업무에 AI를 적용해 보고 싶다. ”



개발자

“ 목적별로 적합한 머신러닝,  
딥러닝 툴을 연계할 수 있고  
분석모델을 운영에 쉽게  
Deploy할 수 있는 기능이 필요 ”



Student / MBA

“ 라이선스 제한있는 상용제품보다  
학기후에도 활용가능한 오픈소스 플랫폼으로  
AI실습강의가 이루어졌으면 좋겠다 ”

오픈소스 AI분석 플랫폼 Brightics Studio

# Brightics Studio

# 오픈소스 Brightics Studio의 주요 특징

## 1 Workflow Modeler

시각화된 분석 모델링 환경

## 3 Ready to USE

분석모델의 입력/공유/배포기능  
기본제공

## 5 3rd Party I/F

Public Cloud의 ML/DL tool 연계  
분석모델링 환경

## 2 User defined Function

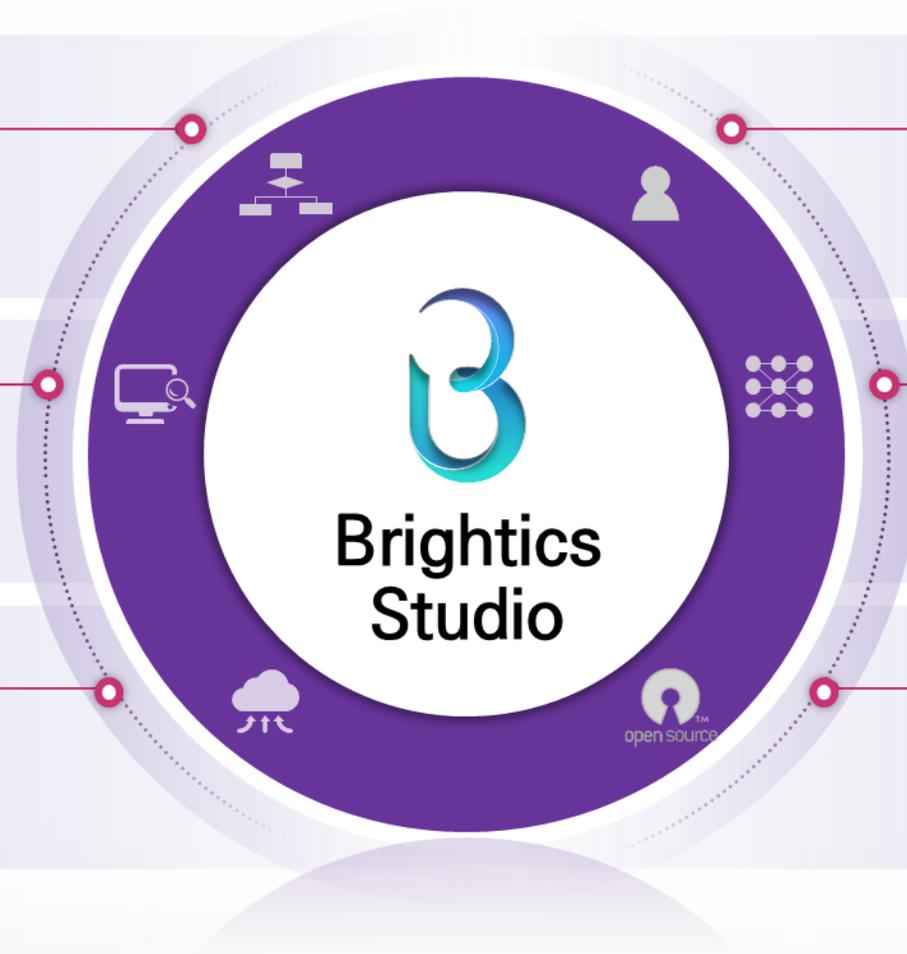
사용자 정의함수로 개인화된  
분석환경 제공 및 생태계 확산

## 4 Deep Learning

시각화 된 Deep Learning  
모델링/분석 환경

## 6 Open Source

Apache License,  
Version 2.0 으로 배포



# Feature 1 Workflow Modeler - 시각화된 분석 모델링 환경

**1 분석 workflow 시각화**

**2 입력데이터 시각화**

**3 분석함수 즉시실행**

**4 출력데이터 시각화**

Workflow steps: One Hot Encode (21:39:11.671) → Train Data (21:39:11.978) → XGB Regression Train (21:39:13.795) → XGB Regression Predict (21:39:13.983) → Evaluate Regression (21:39:14.6) → Python Script: Filter - ITEM\_0 (21:39:14.60) → Melt (21:39:14.230) → Python Script: Filter - ITEM\_14 (21:39:14.112) → Melt (21:39:14.112)

Table view (Table):

No.	ITEM_ID	YEARWEEK	WEEK	WEEK_SEQ	PROMOTI...	SALES_QTY	SALES_Q...	TEMPERA...	W
1	item_0	201801	1	130	0	677	-0.89661	6.54	
2	item_0	201802	2	131	0	493	-1.10212	5.56	
3	item_0	201803	3	131	0	412	-1.19259	7.73	
4	item_0	201804	4	133	0	1615	0.15107	5.81	
5	item_0	201805	5	134	0	1938	0.51183	3.9	
6	item_0	201806	6	135	0	1793	0.34988	0.6	
7	item_0	201807	7	136	0	1618	0.15442	2.03	
8	item_0	201808	8	137	0	1516	0.04049	-1.77	
9	item_1	201801	1	123	0	1026	-0.79606	6.54	
10	item_1	201802	2	124	0	715	-0.881270...	5.56	

XGB Regression Predict configuration:

- Inputs: table, Test Data, out\_table, model, XGB Regression Train, model
- Prediction Column Name: prediction

Box plot: prediction vs ITEM\_ID (item\_0, item\_12, item\_16, item\_5, item\_9)

Pie chart: prediction vs ITEM\_ID (item\_0 to item\_16)

# Feature 1 List of Machine Learning Functions

## Descriptive Analytics

### Manipulation

- Replace Missing Number
- Replace Missing String
- Simple Filter
- Sort

### Transform

- Concatenate
- Melt
- Merge
- Pivot Table
- Random Sample
- Delete Missing Data
- Train Test Split
- Select Column

### Statistics

- Bartlett's Test
- Correlation
- One Way ANOVA
- Pairplot
- Profile Table
- Tukey's Range Test

### Extraction

- Add Shift
- Label Encode (with Model)
- One Hot Encode (with Model)
- PCA (with Model)
- Scale (with Model)
- Add Column
- Add Column If

## Predictive Analytics

### Regression

- Decision Tree Regression Train
- Decision Tree Regression Predict
- GLM Train
- GLM Predict
- Linear Regression Train
- Linear Regression Predict
- XGB Regression Train
- XGB Regression Predict

### Clustering

- Kmeans Train & Predict
- Kmeans Train & Predict (Silhouette)
- Kmeans Predict

### Classification

- Decision Tree Classification Train
- Decision Tree Classification Predict
- Logistic Regression Train
- Logistic Regression Predict
- Support Vector Classification Train
- Support Vector Classification Predict
- XGB Classification Train
- XGB Classification Predict

### Evaluation

- Evaluate Classification
- Evaluate Regression
- Plot ROC Curve and PR Curve

# Feature ② User Defined Function (UDF)

분석전문가의 전문 알고리즘을 쉽게 Workflow로 통합하여 시각화된 분석환경으로 제공

## Write a Python Script

The screenshot shows the 'Spec Generator' interface. It has fields for 'Label' (User Defined Function Sample 01), 'Description' (This is User Defined Function Sample 01), 'Tags' (+ New Tag), 'Script Type' (Python), 'Inputs Configuration' (Key: table, Type: Table), and 'Output Configuration' (Key: model, Type: Model). A Python script is shown in a text area, with a red dashed box highlighting the OLS fit and summary lines.

```
import numpy as np
import statsmodels.api as sm
import statsmodels.formula.api as smf
in_table = table[0]
result = sm.OLS(in_table[label_col], in_table[feature_cols]).fit()
model = result.summary()
```

## Set Parameters for your UDF

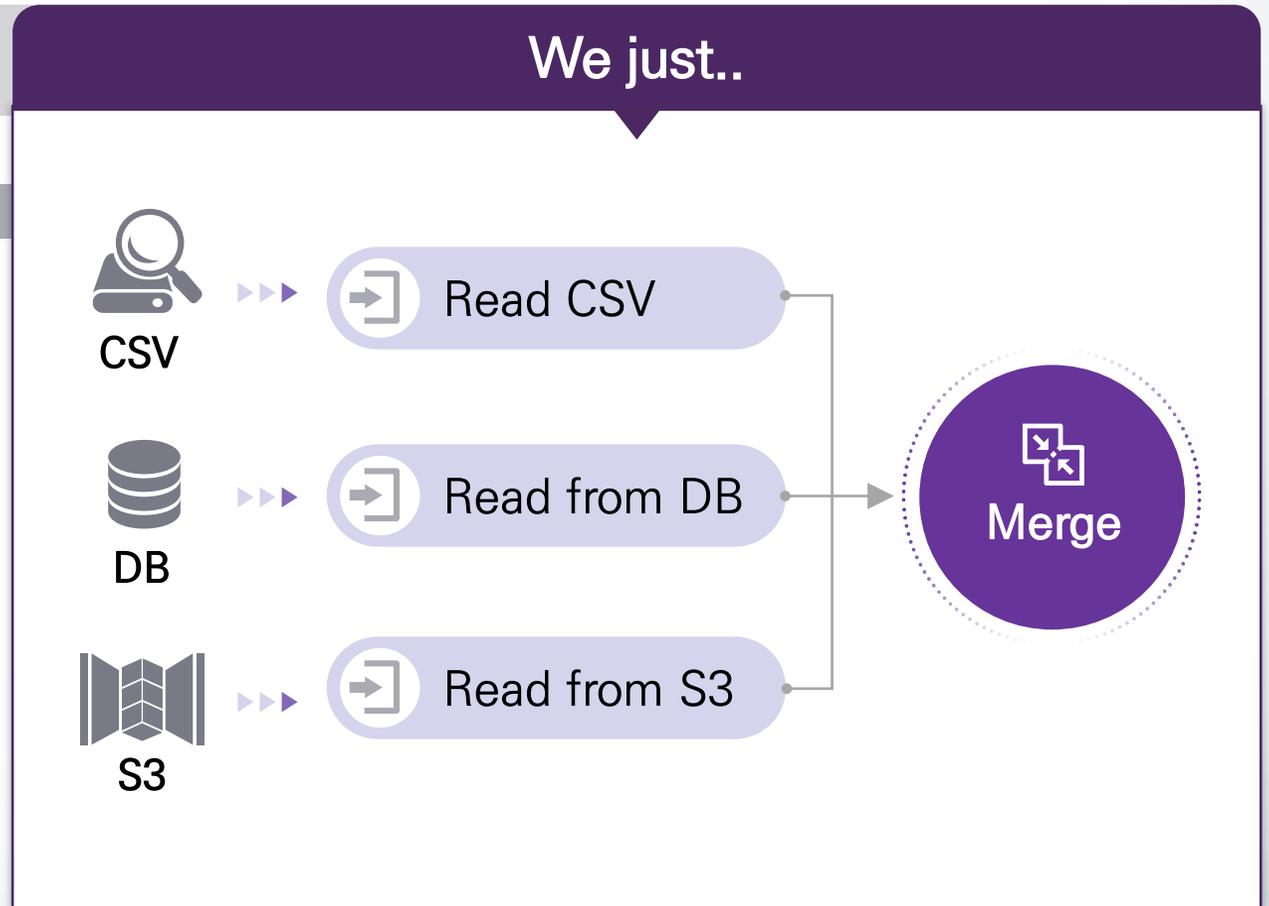
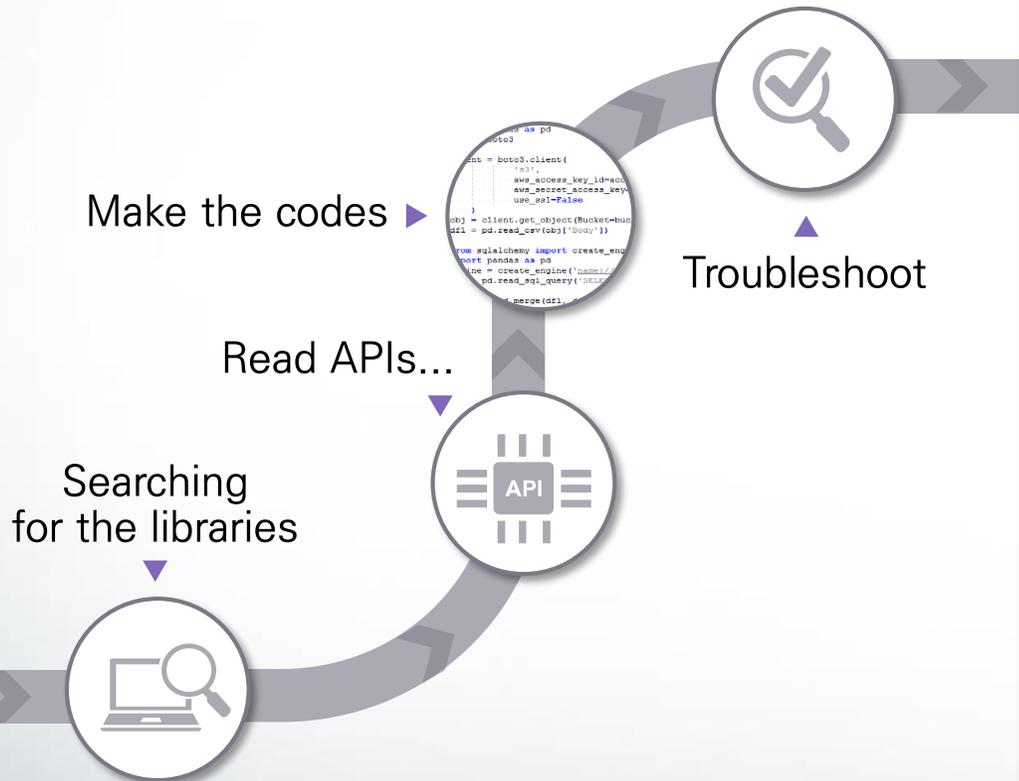
The screenshot shows the parameter configuration interface for a UDF. It has a dropdown for 'label\_col' (label\_col) and a 'Delete' button. Below are fields for 'Label' (Label Column), 'Description' (This is Label Column), 'Visible When' (+ Add Condition), 'Validation' (+ Add Validation), 'Control' (Column Selector), 'Mandatory' (True), 'Placeholder', 'Selection Mode' (Single), and 'Column Type(s)' (Check all, Integer, Long, Float, Double, Decimal, String, Array(String), Array(Integer), Array(Double), Date, Boolean).

# Feature ③ Connection – easy to connect

데이터 준비를 위해 필요한 주요 데이터 connection을 단일 함수(기능)로 제공

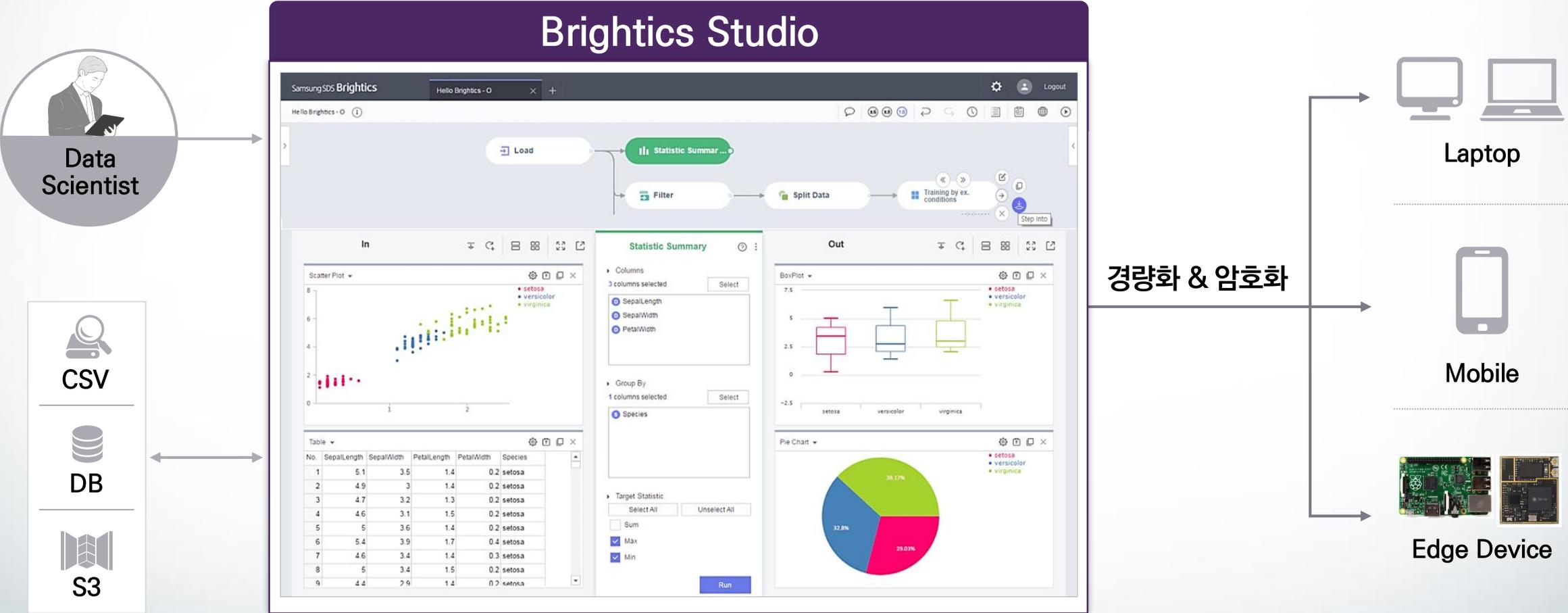
You need..

We just..



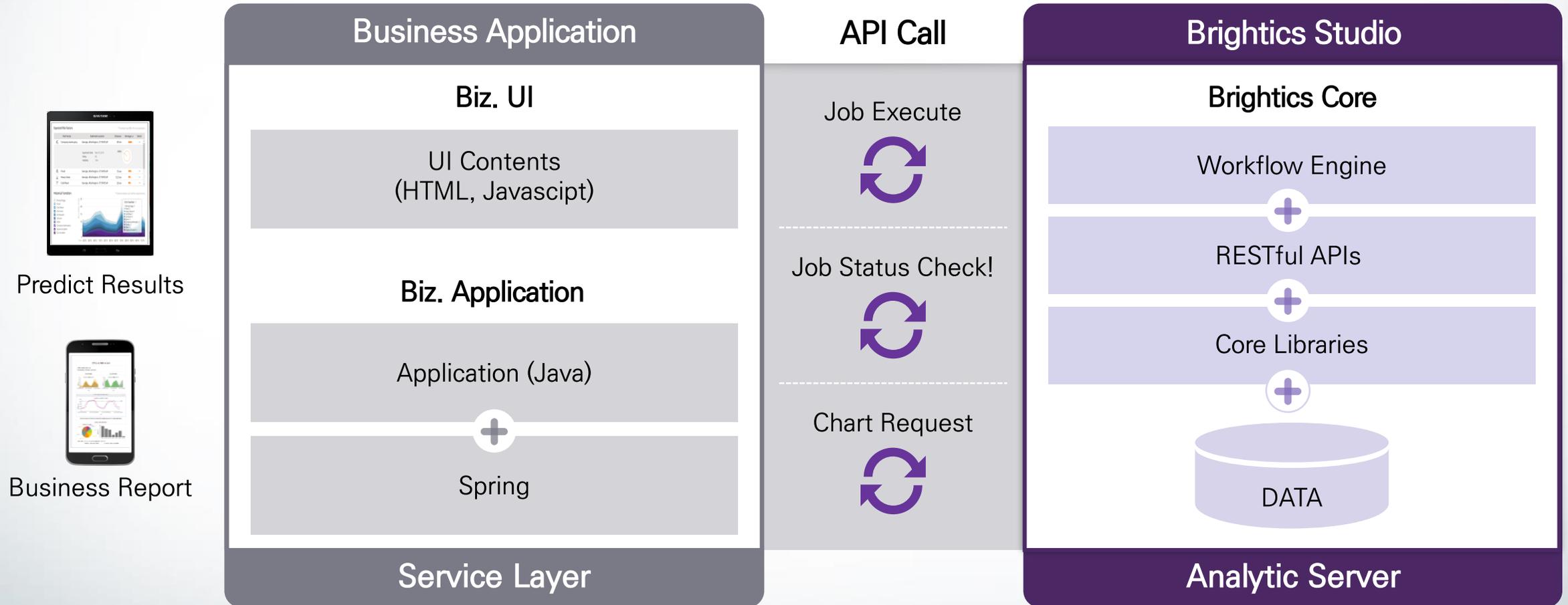
# Feature ③ Deploy – Lightweight deployable model

Edge Analytics를 지원하는 경량화 분석모델 배포 제공



# Feature ③ Share – RESTful API

분석결과 (차트/데이터)의 활용을 위한 RESTful API 제공



# Feature ④ Deep Learning 연계

시각화 된 Deep Learning 모델링/분석 환경 제공

### Deep Learning Work-flow시각화

The screenshot shows the Samsung SDS Brightics interface for a Deep Learning workflow. On the left is a 'PALETTE' with 'Function' and 'Parameters' tabs, listing components like Loss (categorical\_crossentropy), Metrics (accuracy), Batch Size (32), Epochs (10), Optimizer (Adam), and Checkpoint Group Name. The central workspace shows a vertical flow of components: DL Load, Conv2D, Conv2D (highlighted with a red dashed box), MaxPooling2D, and Dropout. On the right is a 'PROPERTIES' panel for the selected Conv2D layer, showing Filters (32), Kernel Size (3,3), and Activation (relu). A 'Minimap' is visible in the top left of the workspace.

### 머신러닝 + 딥러닝 통합분석

The screenshot shows the Samsung SDS Brightics interface for machine learning analysis. It features a flow diagram with 'DL Predict' and 'Statistic Summary' components, both highlighted with red dashed boxes. Below the flow are two pink callout boxes labeled 'Deep Learning' and 'Machine Learning'. The interface also includes a 'Variables' panel on the left, a 'Table' view at the bottom with columns for 'No.', 'predict\_0', 'predict\_1', 'predict\_2', 'predict\_3', and 'predict\_4', and a 'Statistic Sum' panel on the right.

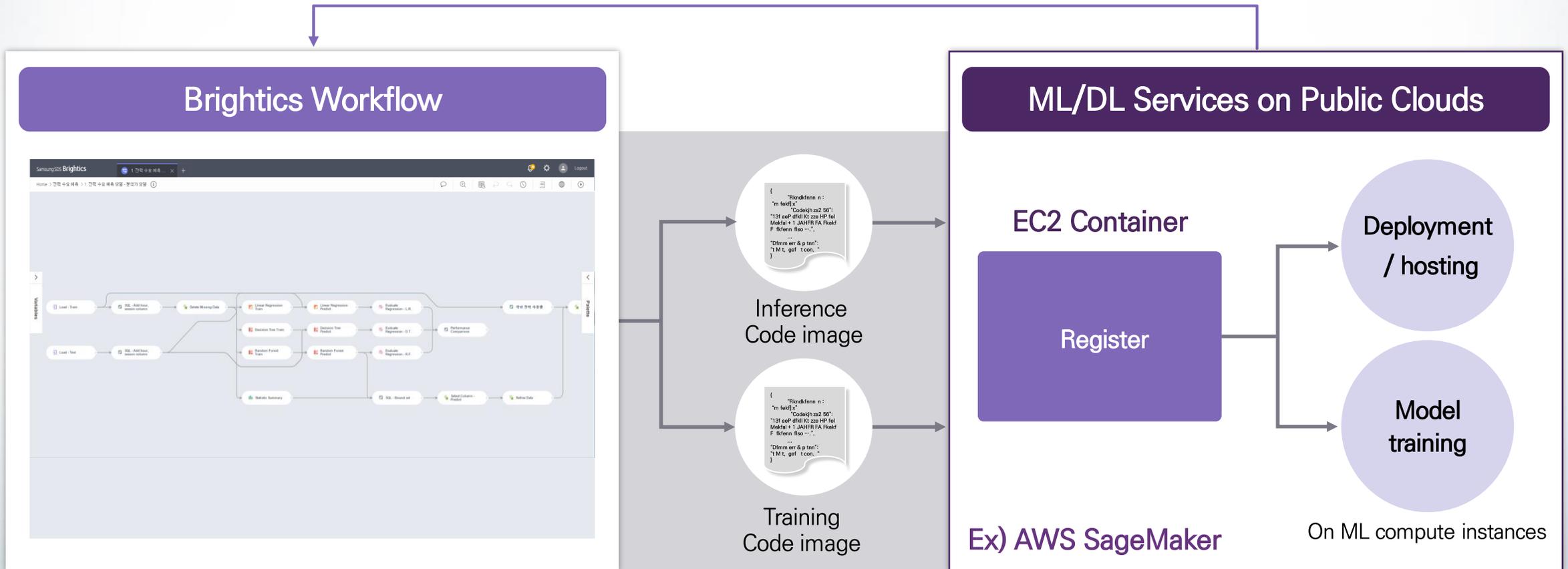
### 3rd Party Tool 지원

Python code Export

```
model.add(Conv2D(filters=32, kernel
model.add(MaxPooling2D(pool_size=(2
model.add(Dropout(rate=0.5))
model.add(Flatten())
model.add(Dense(units=128,activation
model.add(Dropout(rate=0.5))
```

# Feature ⑤ 3<sup>rd</sup> Party Tool 연계 – Public Cloud DL/ML Tool

각 Public Cloud(AWS, MS Azure, GC 등) 에서 제공하는 DL/ML Tool 을  
Brightics 의 시각화 된 workflow 와 연동하여 분석모델링 환경 제공



## Feature ⑥ Open Source

# Brightics Studio Download

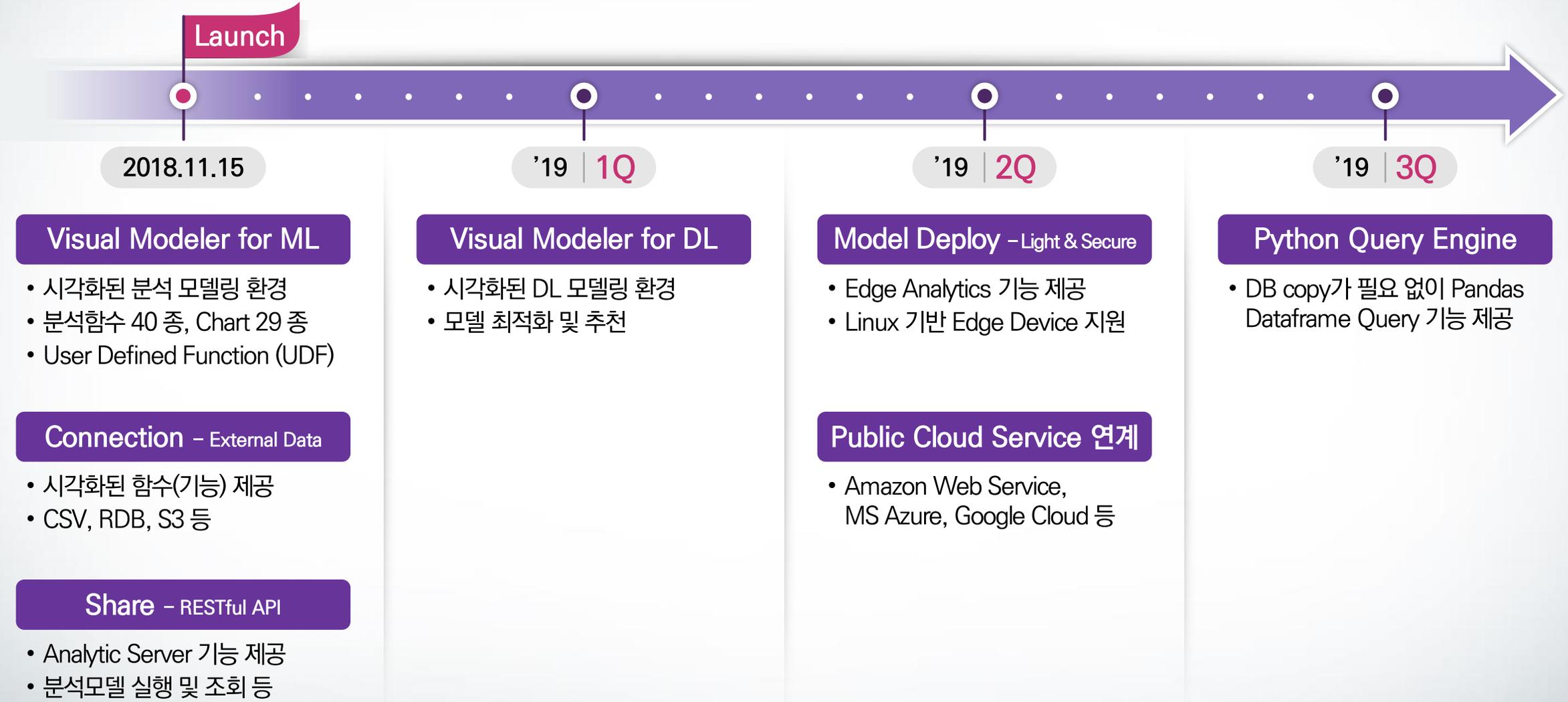
[www.brightics.ai](http://www.brightics.ai)

Github : [github.com/brightics/studio](https://github.com/brightics/studio)

※ Licensed under the Apache License, Version 2.0



# Roadmap



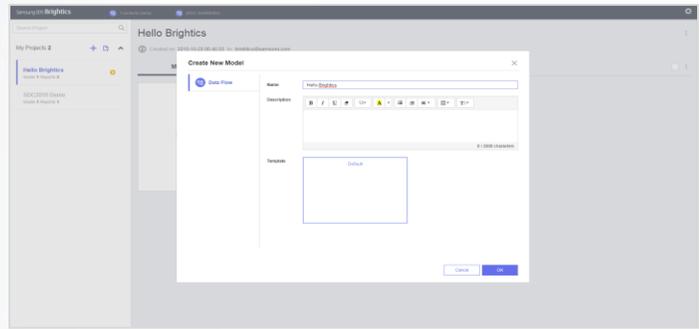
오픈소스 AI분석 플랫폼 Brightics Studio

# Demo

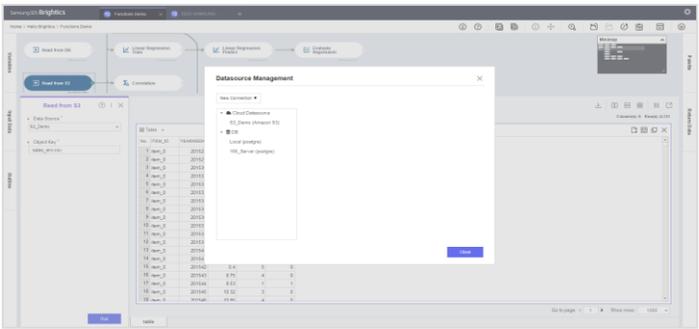
# Brightics Studio Demo Scenario

플랫폼에서 분석모델 생성, 시각화, Report, Export

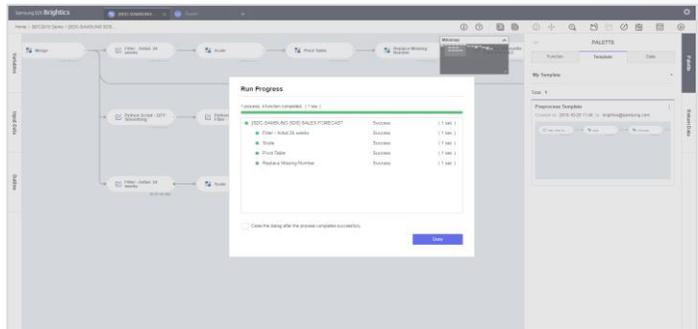
## Create a model



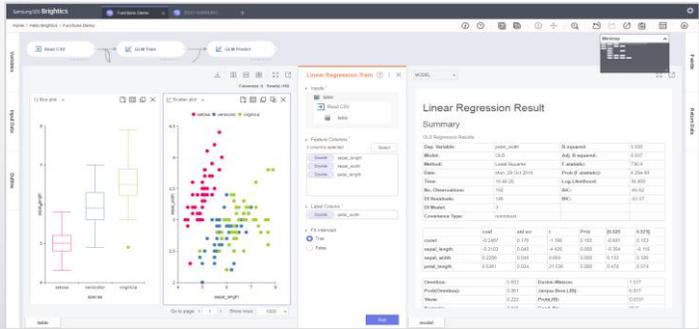
## Select data sources



## Build a model & run



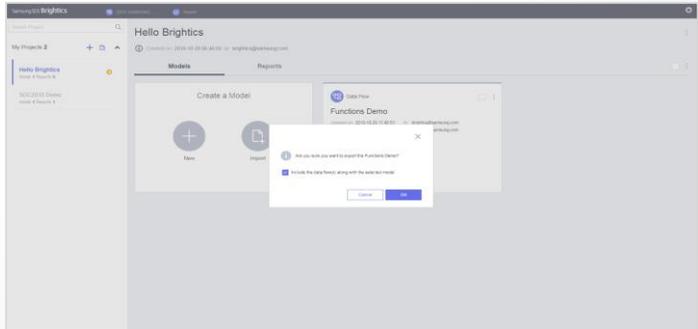
## Visualization & summary



## Create a report



## Export a model

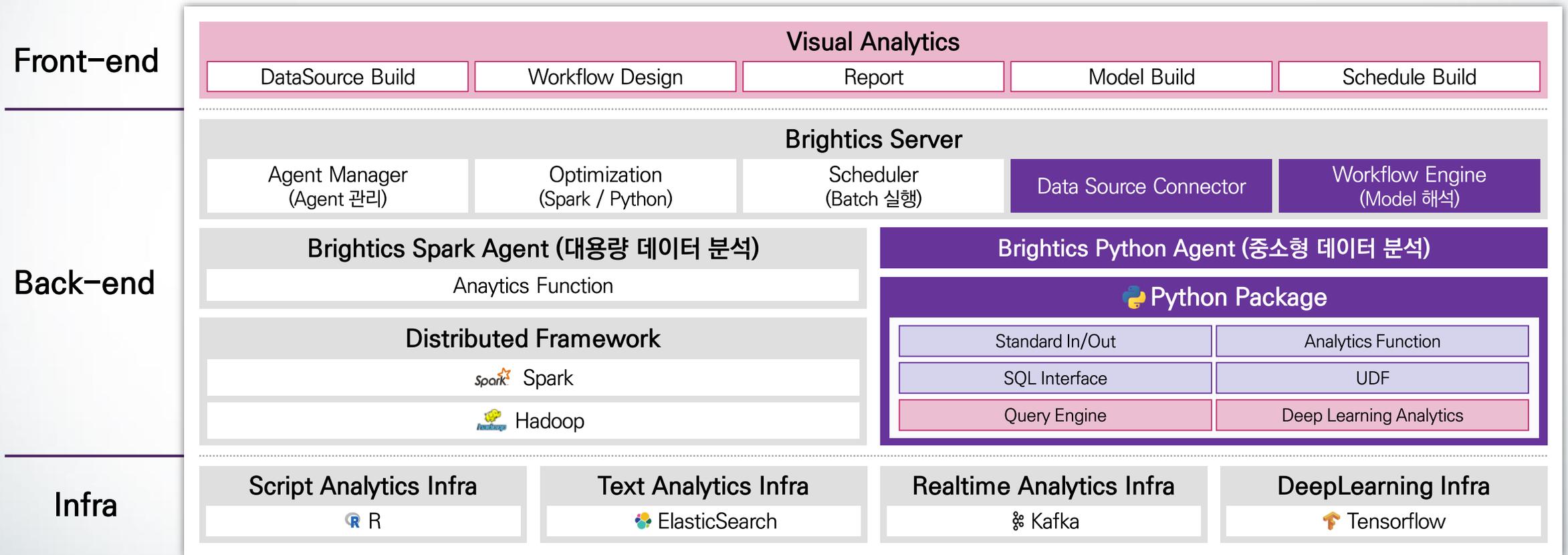


오픈소스 AI분석 플랫폼 Brightics Studio

# Brightics AI

# Brightics AI

다양한 기업데이터를 가공, 분석하여 Business Layer에 AI 서비스를 제공하는 플랫폼



※'18 Open Binary 제공

'18 Open Source '19 Open Source

# 업종별 주요 분석사례

제조, 마케팅, 물류, 보안, IoT, 헬스 분야 등 90개 레퍼런스 확보



# Q & A

Partner

Disrupt

Foresee



Thank you

