



시계열 예측을 위한 AutoML 만들어보기

김태영

CEO

AIFactory

tykim@aifactory.page



AutoML

작동원리
이용방법
문제유형과 평가
계산 컴퓨팅 리소스
작업 형식
시계열 예측 미리보기

시계열 예측을 위한 AutoML 만들어보기

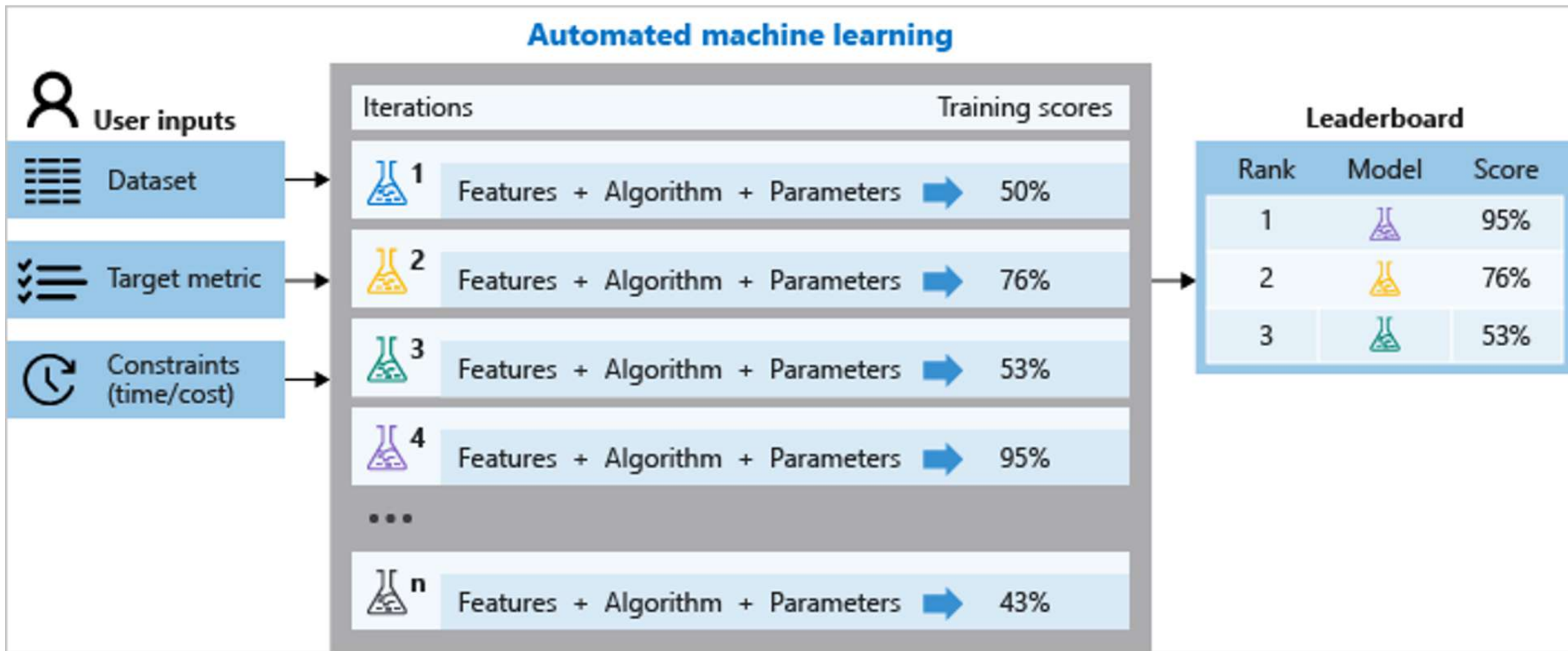


AutoML

Azure 기계 학습에서 지정한 대상 메트릭을 사용하여 모델을 학습하고 조정하려는 경우 AutoML을 적용합니다. AutoML은 기계 학습 모델 개발 프로세스를 보편화하고, 데이터 과학 전문 지식에 관계없이 사용자의 역량을 강화하여 모든 문제에 대해 엔드투엔드 기계 학습 파이프라인을 식별합니다.

분류	타임시리즈 예측	회귀
사기 탐지	판매 예측	CPU 성능 예측
마케팅 예측	수요 예측	
뉴스 그룹 데이터 분류	음료 생산 예측	

AutoML - 작동원리



AutoML - Azure Machine Learning Studio 이용



미리 보기

Microsoft Azure Machine Learning

☰

새로 만들기

홈

작성자

Notebooks

자동화된 ML

디자이너

자산

데이터 세트

실험

파이프라인

모델

엔드포인트

관리

컴퓨팅

데이터 저장소

데이터 레이블 지정

AutoML_TS > 자동화된 ML > 실행 시작

새 자동화된 ML 실행 만들기

데이터 세트 선택

실행 구성

작업 형식 및 설정

데이터 세트 선택

아래 목록에서 데이터 세트를 선택하거나 새 데이터 세트를 만드세요. 자동화된 ML은 현재 실행 작성용 테이블 형식 데이터만 지원합니다.

+ 데이터 세트 만들기

...

검색하여 항목 필터링...

데이터 세트 이름	데이터 세트 형식	만든 날짜:
<div></div> <div>표시할 데이터 세트 없음</div>		

뒤로

다음

취소

시계열 예측을 위한 AutoML 만들어보기

#gav2020kr

AutoML - Python SDK 이용



실험 시간 초과 분을 30분으로 설정하고 2개의 교차 유효성 검사 접기로 설정된 AUC를 기본 메트릭으로 가중치를 사용하는 분류 실험.

Python

복사

```
automl_classifier=AutoMLConfig(  
    task='classification',  
    primary_metric='AUC_weighted',  
    experiment_timeout_minutes=30,  
    blacklist_models=['XGBoostClassifier'],  
    training_data=train_data,  
    label_column_name=label,  
    n_cross_validations=2)
```

AutoML - Python SDK 이용



다음은 5개의 유효성 검사 교차 접기로 60분 후에 종료하도록 설정된 회귀 실험의 예입니다.

Python

복사

```
automl_regressor = AutoMLConfig(  
    task='regression',  
    experiment_timeout_minutes=60,  
    whitelist_models=['KNN'],  
    primary_metric='r2_score',  
    training_data=train_data,  
    label_column_name=label,  
    n_cross_validations=5)
```


AutoML - 문제유형과 평가



분류	회귀	시계열 예측
accuracy	spearman_correlation	spearman_correlation
AUC_weighted	normalized_root_mean_squared_error	normalized_root_mean_squared_error
average_precision_score_weighted	r2_score	r2_score
norm_macro_recall	normalized_mean_absolute_error	normalized_mean_absolute_error
precision_score_weighted		

AutoML - 계산 컴퓨팅 리소스 : 로컬

로컬 컴퓨터를 이용하여 AutoML을 학습하는 경우

Python

복사

```
from azureml.train.automl import AutoMLConfig

automl_config = AutoMLConfig(task='regression',
                             debug_log='automated_ml_errors.log',
                             X=x_train.values,
                             y=y_train.values.flatten(),
                             **automl_settings)
```

AutoML - 계산 컴퓨팅 리소스 : 로컬



Python

복사

```
from azureml.core.experiment import Experiment
experiment = Experiment(ws, "taxi-experiment")
local_run = experiment.submit(automl_config, show_output=True)
```

복사

Running on local machine

Parent Run ID: AutoML_1766cdf7-56cf-4b28-a340-c4ae15b12b

Current status: DatasetFeaturization. Beginning to featurize the dataset.

Current status: DatasetEvaluation. Gathering dataset statistics.

Current status: FeaturesGeneration. Generating features for the dataset.

Current status: DatasetFeaturizationCompleted. Completed featurizing the dataset.

Current status: DatasetCrossValidationSplit. Generating individually featurized datasets.

Current status: ModelSelection. Beginning model selection.

ITERATION: The iteration being evaluated.

PIPELINE: A summary description of the pipeline being evaluated.

AutoML - 계산 컴퓨팅 리소스 : 원격

- 원격 컴퓨팅을 사용하면 AutoML 실험 반복이 비동기적으로 실행됨
- 원격으로 학습하려면 AmlCompute와 같은 원격 계산 대상 필요

Python

복사

```
from azureml.core.compute import AmlCompute
from azureml.core.compute import ComputeTarget
import os

# choose a name for your cluster
compute_name = os.environ.get("AML_COMPUTE_CLUSTER_NAME", "cpu-cluster")
compute_min_nodes = os.environ.get("AML_COMPUTE_CLUSTER_MIN_NODES", 0)
compute_max_nodes = os.environ.get("AML_COMPUTE_CLUSTER_MAX_NODES", 4)

# This example uses CPU VM. For using GPU VM, set SKU to STANDARD_NC6
vm_size = os.environ.get("AML_COMPUTE_CLUSTER_SKU", "STANDARD_D2_V2")
```


AutoML - 계산 컴퓨팅 리소스 : 원격

- 원격 컴퓨팅을 사용하면 AutoML 실험 반복이 비동기적으로 실행됨
- 원격으로 학습하려면 AmlCompute와 같은 원격 계산 대상 필요

```
print('creating a new compute target...')
provisioning_config = AmlCompute.provisioning_configuration(vm_size = vm_size,
                                                            min_nodes = compute
                                                            max_nodes = compute

# create the cluster
compute_target = ComputeTarget.create(ws, compute_name, provisioning_config)

# can poll for a minimum number of nodes and for a specific timeout.
# if no min node count is provided it will use the scale settings for the clust
compute_target.wait_for_completion(show_output=True, min_node_count=None, timec

# For a more detailed view of current AmlCompute status, use get_status()
print(compute_target.get_status().serialize())
```


AutoML - 계산 컴퓨팅 리소스 : 원격

- 원격 컴퓨팅을 사용하면 AutoML 실험 반복이 비동기적으로 실행됨
- 원격으로 학습하려면 AmlCompute와 같은 원격 계산 대상 필요

```
Python 복사

from azureml.core.experiment import Experiment
experiment = Experiment(ws, 'automl_remote')
remote_run = experiment.submit(automl_config, show_output=True)
```

다음 예제와 비슷한 출력이 표시됩니다.

→

```
Running on remote compute: mydsvmParent Run ID: AutoML_015ffe76-c331-406d-9bfc
*****
ITERATION: The iteration being evaluated.
PIPELINE: A summary description of the pipeline being evaluated.
DURATION: Time taken for the current iteration.
METRIC: The result of computing score on the fitted pipeline.
BEST: The best observed score thus far.
*****
```

AutoML - 작업 형식





새 자동화된 ML 실행 만들기


- ✓ 데이터 세트 선택
- ✓ 실행 구성
- 작업 형식 및 설정

작업 형식 선택

실험을 위한 기계 학습 작업 형식을 선택합니다. 필요한 경우 추가 설정을 사용하여 실험을 미세 조정할 수 있습니다.

 **분류**
대상 열에서 예/아니요, 파랑, 빨강, 녹색 등의 여러 범주 중 하나를 예측하려는 경우

 **회귀**
연속 숫자 값을 예측하려는 경우

 **시계열 예측**
시간을 기준으로 값을 예측하려는 경우



시계열 예측 방법에는 몇 가지 추가 정보가 필요합니다.

AutoML - 시계열 예측 미리보기

열렬 forecasting 작업에는 구성 개체에 추가 매개 변수가 필요합니다.

1. `time_column_name`: 유효한 시간계를 포함하는 학습 데이터의 열 이름을 정의하는 필수 매개 변수입니다.
2. `max_horizon`: 학습 데이터의 주기에 따라 예측할 시간을 정의합니다. 예를 들어 일일 시간 그레인 이 있는 학습 데이터가 있는 경우 모델이 학습할 기간을 정의하는 것입니다.
3. `grain_column_names`: 학습 데이터에 개별 열렬 데이터가 포함된 열의 이름을 정의합니다. 예를 들어 매장별로 특정 브랜드의 매출을 예측하는 경우 매장 및 브랜드 열을 그레인 열로 정의합니다. 각 그레인/그룹화에 대해 별도의 타임시리즈 및 예측이 생성됩니다.

아래 에서 사용되는 설정의 예는 [샘플 전자 필기장](#)을 참조하십시오.

Python

복사

```
# Setting Store and Brand as grains for training.
grain_column_names = ['Store', 'Brand']
nseries = data.groupby(grain_column_names).ngroups

# View the number of time series data with defined grains
print('Data contains {0} individual time-series.'.format(nseries))
```

AutoML - 시계열 예측 미리보기



Python

복사

```
time_series_settings = {
    'time_column_name': time_column_name,
    'grain_column_names': grain_column_names,
    'drop_column_names': ['logQuantity'],
    'max_horizon': n_test_periods
}

automl_config = AutoMLConfig(task = 'forecasting',
                             debug_log='automl_oj_sales_errors.log',
                             primary_metric='normalized_root_mean_squared_error',
                             experiment_timeout_minutes=20,
                             training_data=train_data,
                             label_column_name=label,
                             n_cross_validations=5,
                             path=project_folder,
                             verbosity=logging.INFO,
                             **time_series_settings)
```

AutoML 시계열 예측

문제정의
AutoML 시계열 시작하기
실습 - 에너지 수요 예측



문제정의



상점 매출 예측

유형: 복수 시계열 데이터 예측
평가: 정규화된 제곱 평균 오차

맥주 생산량 예측

유형: 단일 시계열 데이터 예측
평가: 정규화된 제곱 평균 오차

공유 자전거 수요 예측

유형: 복수 시계열 데이터 예측
평가: 정규화된 제곱 평균 오차

에너지 수요 예측


유형: 단일 시계열 데이터 예측
평가: 정규화된 제곱 평균 오차

AutoML 시계열 시작하기

- 순서
 - 데이터 준비
 - AutoMLConfig을 이용해서 시계열 파라미터 설정
 - 시계열 예측 수행
- 모델
 - Prophet : accurate & fast, robust to outliers, missing data, and dramatic changes
 - Auto-ARIMA : a popular statistical method
 - ForecastTCN : Given larger data, deep learning models

AutoML 시계열 시작하기

- 데이터 준비하기
 - 데이터형태 살펴보기

 복사

```
day_datetime,store,sales_quantity,week_of_year
9/3/2018,A,2000,36
9/3/2018,B,600,36
9/4/2018,A,2300,36
9/4/2018,B,550,36
9/5/2018,A,2100,36
9/5/2018,B,650,36
9/6/2018,A,2400,36
9/6/2018,B,700,36
9/7/2018,A,2450,36
9/7/2018,B,650,36
```

AutoML 시계열 시작하기

- 데이터 준비하기
 - datetime 타입 정의

Python

복사

```
import pandas as pd
data = pd.read_csv("sample.csv")
data["day_datetime"] = pd.to_datetime(data["day_datetime"])
```

AutoML 시계열 시작하기

- 데이터 준비하기
 - 데이터셋 구성

Python

복사

```
train_data = data.iloc[:950]
test_data = data.iloc[-50:]

label = "sales_quantity"

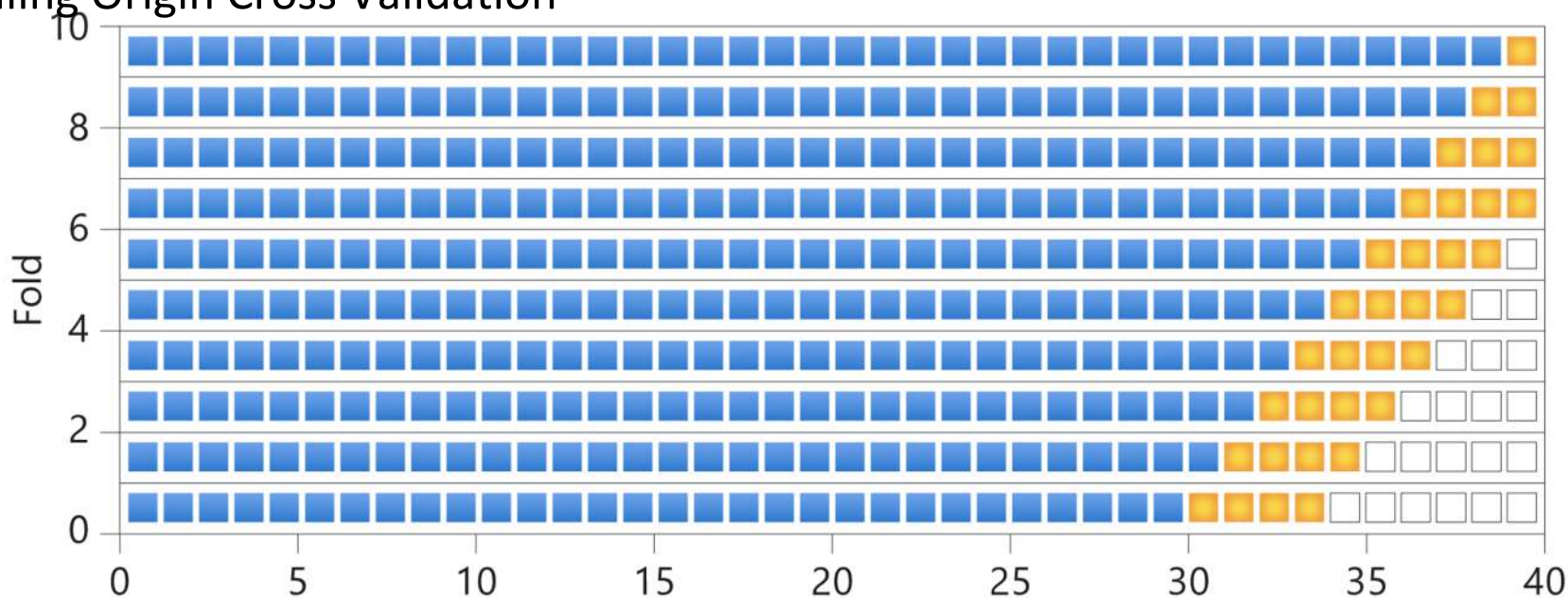
test_labels = test_data.pop(label).values
```


AutoML 시계열 시작하기



- 데이터 준비하기

- Rolling Origin Cross Validation



Python

Copy

```
automl_config = AutoMLConfig(task='forecasting',  
                             n_cross_validations=3,  
                             ...  
                             **time_series_settings)
```

AutoML 시계열 시작하기

- AutoMLConfig 설정하기

Python

복사

```
time_series_settings = {  
    "time_column_name": "day_datetime",  
    "grain_column_names": ["store"],  
    "max_horizon": "auto",  
    "target_lags": "auto",  
    "target_rolling_window_size": "auto",  
    "preprocess": True,  
}
```

AutoML 시계열 시작하기

- AutoMLConfig 설정하기

- 전처리 수행 내용

- Detect time-series sample frequency (for example, hourly, daily, weekly) and create new records for absent time points to make the series continuous.
 - Impute missing values in the target (via forward-fill) and feature columns (using median column values)
 - Create grain-based features to enable fixed effects across different series
 - Create time-based features to assist in learning seasonal patterns
 - Encode categorical variables to numeric quantities

AutoML 시계열 시작하기

- AutoMLConfig 설정하기

Python

복사

```
from azureml.core.workspace import Workspace
from azureml.core.experiment import Experiment
from azureml.train.automl import AutoMLConfig
import logging

automl_config = AutoMLConfig(task='forecasting',
                             primary_metric='normalized_root_mean_squared_error',
                             experiment_timeout_minutes=15,
                             enable_early_stopping=True,
                             training_data=train_data,
                             label_column_name=label,
                             n_cross_validations=5,
                             enable_ensembling=False,
                             verbosity=logging.INFO,
                             **time_series_settings)
```

AutoML 시계열 시작하기

- AutoMLConfig 설정하기
 - 딥러닝 적용

Python

복사

```
automl_config = AutoMLConfig(task='forecasting',  
                             enable_dnn=True,  
                             ...  
                             **time_series_settings)
```


AutoML 시계열 시작하기



- AutoMLConfig 설정하기

- 딥러닝 적용

Create a new Automated ML run

- ✓ Select dataset
- ✓ Configure run
- Task type and settings

Select task type

Select the machine learning task type for the experiment. Additional settings are available to fine tune the experiment if needed.

Classification

To predict one of several categories in the target column, yes/no, blue, red, green.

Regression

To predict continuous numeric values

Time series forecasting

To predict values based on time

The time series forecasting method requires some additional information.

Time column * ⓘ

Select a time column...

Group by column(s) ⓘ

Select column(s)...

Forecast horizon * ⓘ

☐ Autodetect

☒ Enable deep learning (preview) ⓘ

[View additional configuration settings](#) [View featurization settings](#)

Back

Finish

Cancel

AutoML 시계열 시작하기

- 실험 실행

Python

복사

```
from azureml.core.workspace import Workspace
from azureml.core.experiment import Experiment
from azureml.train.automl import AutoMLConfig
import logging

automl_config = AutoMLConfig(task='forecasting',
                             primary_metric='normalized_root_mean_squared_error',
                             experiment_timeout_minutes=15,
                             enable_early_stopping=True,
                             training_data=train_data,
                             label_column_name=label,
                             n_cross_validations=5,
                             enable_ensembling=False,
                             verbosity=logging.INFO,
                             **time_series_settings)
```

```
ws = Workspace.from_config()
experiment = Experiment(ws, "forecasting_example")
local_run = experiment.submit(automl_config, show_output=True)
best_run, fitted_model = local_run.get_output()
```

AutoML 시계열 시작하기

- 실험 실행

Python

복사

```
from azureml.core.workspace import Workspace
from azureml.core.experiment import Experiment
from azureml.train.automl import AutoMLConfig
import logging

automl_config = AutoMLConfig(task='forecasting',
                             primary_metric='normalized_root_mean_squared_error',
                             experiment_timeout_minutes=15,
                             enable_early_stopping=True,
                             training_data=train_data,
                             label_column_name=label,
                             n_cross_validations=5,
                             enable_ensembling=False,
                             verbosity=logging.INFO,
                             **time_series_settings)
```

```
ws = Workspace.from_config()
experiment = Experiment(ws, "forecasting_example")
local_run = experiment.submit(automl_config, show_output=True)
best_run, fitted_model = local_run.get_output()
```

AutoML 시계열 시작하기

- 실험 실행
 - 피처 엔지니어링 요약 보기

Python

복사

```
fitted_model.named_steps['timeseriestransformer'].get_featurization_summary()
```

AutoML 시계열 시작하기

- 예측 및 평가

- 예측

Python

복사

```
label_query = test_labels.copy().astype(np.float)
label_query.fill(np.nan)
label_fcst, data_trans = fitted_pipeline.forecast(
    test_data, label_query, forecast_destination=pd.Timestamp(2019, 1, 8))
```

- 평가

Python

복사

```
from sklearn.metrics import mean_squared_error
from math import sqrt

rmse = sqrt(mean_squared_error(actual_labels, predict_labels))
rmse
```

실습



상점 매출 예측

유형: 복수 시계열 데이터 예측

평가: 정규화된 제곱 평균 오차

맥주 생산량 예측

유형: 단일 시계열 데이터 예측

평가: 정규화된 제곱 평균 오차

공유 자전거 수요 예측

유형: 복수 시계열 데이터 예측

평가: 정규화된 제곱 평균 오차

에너지 수요 예측

유형: 단일 시계열 데이터 예측

평가: 정규화된 제곱 평균 오차

당신의 목소리를 들려주세요!

Global Azure Virtual 2020는 여러분의 목소리를 기다립니다.
가감 없는 목소리가 발표자 분에게 매우 큰 힘이 됩니다.
앞으로 더 좋은 행사가 될 수 있도록 목소리를 내주세요.
감사합니다!

세션에 대한 목소리:

<https://sv.krazure.com/>(세션명)_(순서)

파트너사 행사:

<https://bit.ly/2RvJQzR>

