머신러닝 파이프라인

데이터 검증

TFDV TensorFlow Data Validation

송호연

목차

데이터 검증 TFDV

학습목표

데이터 검증 TFDV

01. 데이터 검증이 필요한 이유를 이해한다.

데이터 검증이 필요한 이유에 대해서 이해한다.

02. 스키마 추론과 스키마 환경을 이해한다.

스키마 추론과 스키마 환경에 대해서 이해한다.

03. 데이터 드리프트 및 스큐에 대해서 이해한다.

데이터 드리프트와 스큐가 어떤 의미인지 이해한다.

데이터 검증
TFDV

데이터 검증이 필요한 이유

TFDV 소개

TFDV 사용법

01

데이터 검증 TFDV TensorFlow Data Validation

데이터 검증이 필요한 이유

머신러닝 시스템에서 데이터로 인한 장애는 파악하기가 어렵다.

왜냐하면, 데이터가 잘못 들어와도 예측은 정상적으로 수행되기 때문이다.
그래서 잘못된 예측값을 늦게서야 인지하는 경우가 많다.

데이터를 사용하기 전에 미리 데이터가 정상적인지 확인하는 과정을
거쳐야 한다.

출처 : https://link.chris-chris.ai/ai-lecture-12

데이터 검증 TFDV TensorFlow Data Validation

데이터 검증이 필요한 이유

TFDV에는 기술 통계보기, 스키마 추론, 이상 항목 확인 및 수정, 데이터 세트의 드리프트 및 왜곡 확인이 포함됩니다.

프로덕션 파이프 라인에서 시간이 지남에 따라 변경 될 수 있는 방법을
포함하여 데이터 세트의 특성을 이해하는 것이 중요합니다.
또한 데이터에서 이상한 점을 찾고 훈련, 평가 및 제공 데이터 세트를
비교하여 일관성이 있는지 확인하는 것도 중요합니다.

출처 : https://link.chris-chris.ai/ai-lecture-12

데이터 검증 TFDV TensorFlow Data Validation

데이터 검증이 필요한 이유

데이터 세트의 열은 다음과 같습니다.

pickup_community_area	**fare**	**trip_start_month**
trip_start_hour	**trip_start_day**	**trip_start_timestamp**
pickup_latitude	**pickup_longitude**	**dropoff_latitude**
dropoff_longitude	**trip_miles**	**pickup_census_tract**
dropoff_census_tract	**Payment_type**	**company**
trip_seconds	**dropoff_community_area**	**tips**

출처 : https://link.chris-chris.ai/ai-lecture-12

데이터 검증 TFDV TensorFlow Data Validation

Pip 업그레이드

```
try:
    import colab
    !pip install --upgrade pip
except:
    pass
```

출처 : https://link.chris-chris.ai/ai-lecture-12

데이터 검증 TFDV TensorFlow Data Validation

TensorFlow 설치

```
$ pip install tensorflow==2.2.0
```

출처 : https://link.chris-chris.ai/ai-lecture-12

데이터 검증 TFDV TensorFlow Data Validation

Python 버전 확인

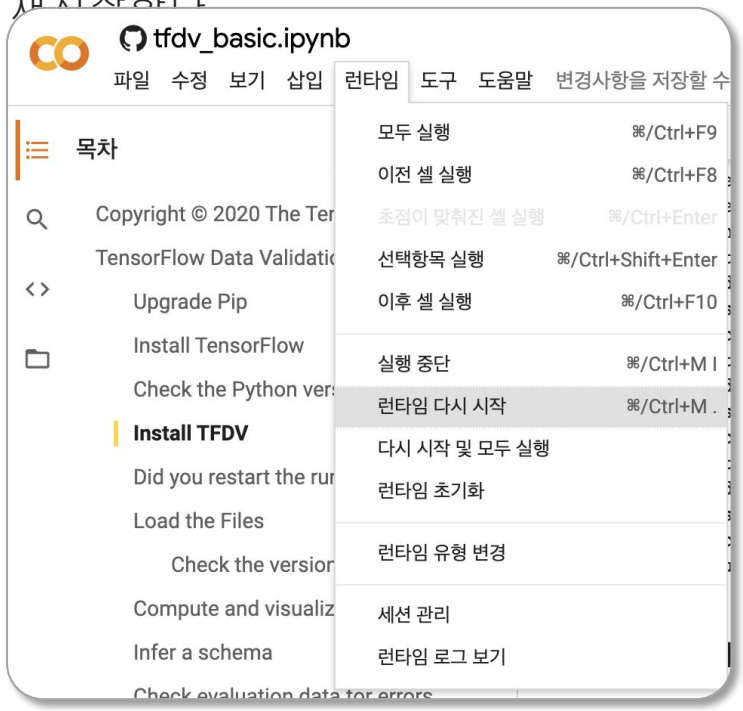
```
import sys

# Confirm that we're using Python 3
assert sys.version_info.major is 3, 'Oops, not running Python 3. Use
Runtime > Change runtime type'
```

출처 : https://link.chris-chris.ai/ai-lecture-12

데이터 검증 TFDV TensorFlow Data Validation

런타임 재시작

런타임 > 런타임 재시작
메뉴를 클릭해서 런타임을
재시작한다



출처 : https://link.chris-chris.ai/ai-lecture-12

데이터 검증 TFDV TensorFlow Data Validation

TFDV 설치

```
import tensorflow as tf

print ( 'Installing TensorFlow Data Validation' )
!pip install -q tensorflow_data_validation[visualization]
```

출처 : https://link.chris-chris.ai/ai-lecture-12

데이터 검증 TFDV TensorFlow Data Validation

Google Cloud Storage에서 데이터세트를 로드

```
import os
import tempfile, urllib, zipfile

#  Set up some globals for our file paths
BASE_DIR = tempfile.mkdtemp()
DATA_DIR = os.path.join(BASE_DIR, 'data')
OUTPUT_DIR = os.path.join(BASE_DIR, 'chicago_taxi_output')
TRAIN_DATA = os.path.join(DATA_DIR, 'train', 'data.csv')
EVAL_DATA = os.path.join(DATA_DIR, 'eval', 'data.csv')
SERVING_DATA = os.path.join(DATA_DIR, 'serving', 'data.csv')
```

출처 : https://link.chris-chris.ai/ai-lecture-12

데이터 검증 TFDV TensorFlow Data Validation

데이터 검증 TFDV 소개

```
# Download the zip file from GCP and unzip it
zip, headers =
urllib.request.urlretrieve('https://storage.googleapis.com/artifacts.
tfx-oss-public.appspot.com/datasets/chicago_data.zip')
zipfile.ZipFile(zip).extractall(BASE_DIR)
zipfile.ZipFile(zip).close()

print("Here's what we downloaded:")
!ls -R {os.path.join(BASE_DIR, 'data')}
```

출처 : https://link.chris-chris.ai/ai-lecture-12

데이터 검증 TFDV TensorFlow Data Validation

Google Cloud Storage에서 데이터세트를 로드

```
Here's what we downloaded:
/tmp/tmp481nnxcj/data:
eval        serving        train

/tmp/tmp481nnxcj/data/eval:
data.csv

/tmp/tmp481nnxcj/data/serving:
data.csv

/tmp/tmp481nnxcj/data/train:
data.csv
```

출처 : https://link.chris-chris.ai/ai-lecture-12

데이터 검증 TFDV TensorFlow Data Validation

버전을 체크

```
import tensorflow_data_validation as tfdv
print('TFDV version: {}'.format(tfdv.version.__version__))
```

TFDV version: 0.27.0

출처 : https://link.chris-chris.ai/ai-lecture-12

데이터 검증 TFDV TensorFlow Data Validation

통계 계산 및 시각화

먼저 tfdv.generate_statistics_from_csv 를 사용하여
훈련 데이터에 대한 통계를 계산한다.

TFDV는 존재하는 특징과 가치 분포의 형태 측면에서 데이터의 빠른 개요를
제공하는 기술 통계를 계산할 수 있다.
내부적으로 TFDV는 Apache Beam의 데이터 병렬 처리 프레임 워크를
사용하여
대규모 데이터 세트에 대한 통계 계산을 확장한다. TFDV와 더 깊이
통합하려는
애플리케이션의 경우 (예 : 데이터 생성 파이프 라인 끝에 통계 생성 연결)
API는
통계 생성을 위해 Beam PTransform도 노출한다.

출처 : https://link.chris-chris.ai/ai-lecture-12

데이터 검증 TFDV TensorFlow Data Validation

통계 계산 및 시각화

```
train_stats =
tfdv.generate_statistics_from_csv(data_location=TRAIN_DATA)
```

Instructions for updating:

Use eager execution and:

`tf.data.TFRecordDataset(path)`

Instructions for updating:

Use eager execution and:

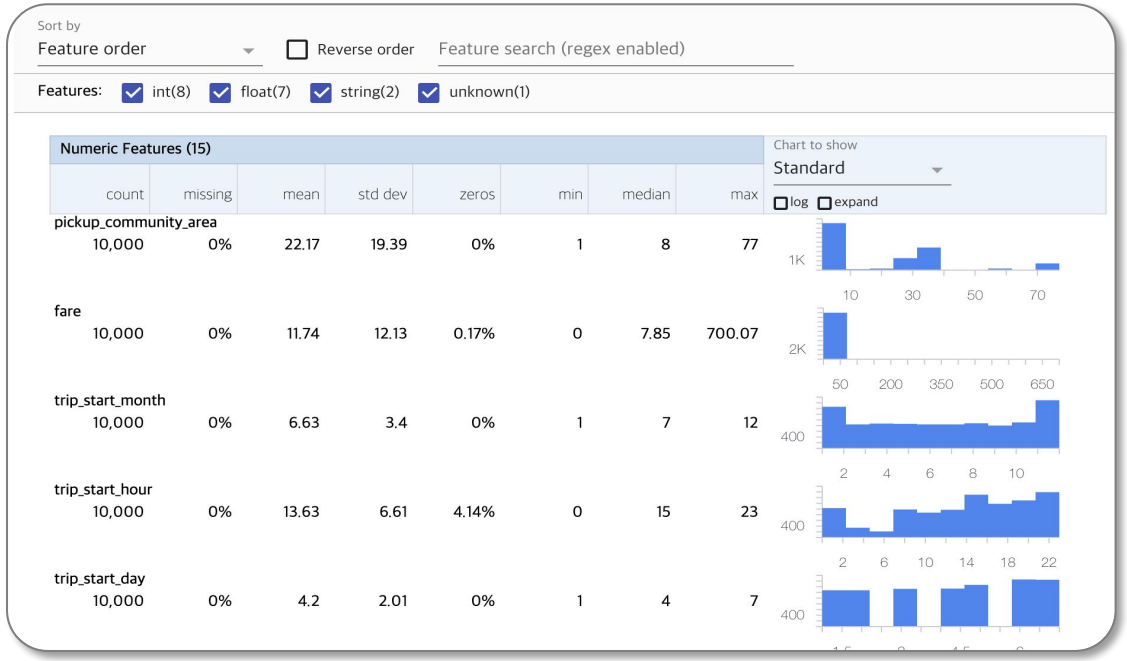
`tf.data.TFRecordDataset(path)`

출처 : https://link.chris-chris.ai/ai-lecture-12

데이터 검증 TFDV TensorFlow Data Validation

통계 계산 및 시각화

```
tfdv.visualize_statistics(train_stats)
```



출처 : https://link.chris-chris.ai/ai-lecture-12

스키마 추론과 스키마 환경

스키마 추론

평가 데이터의 오류 확인

평가 데이터의 이상 데이터Anomaly 확인

스키마의 평가 이상 수정

스키마 환경

02

스키마 추론과 스키마 환경

스키마 추론

이제 tfdv.infer_schema 를 사용하여 데이터에 대한 스키마를 생성 해 보겠습니다.
스키마는 ML과 관련된 데이터에 대한 제약 조건을 정의합니다. 제약 조건의 예에는
각 피쳐의 데이터 유형 (숫자 형이든 범주 형이든 상관없이) 또는 데이터에 존재하는
빈도가 포함됩니다. 범주 형 피쳐의 경우 스키마는 허용되는 값 목록 인 도메인도 정의합니다.

출처 : https://link.chris-chris.ai/ai-lecture-12

스키마 추론과 스키마 환경

스키마 추론

스키마 작성은 특히 많은 피쳐에있는 데이터 세트의 경우 지루한 작업이 될 수 있으므로
TFDV는 기술 통계를 기반으로 스키마의 초기 버전을 생성하는 방법을 제공합니다.

나머지 프로덕션 파이프 라인은 TFDV가 올바른 스키마를 생성하기 때문에 스키마를
올바르게 가져 오는 것이 중요합니다. 스키마는 데이터에 대한 문서도 제공하므로
여러 개발자가 동일한 데이터를 작업 할 때 유용합니다. tfdv.display_schema 를 사용하여 추론 된 스키마를 표시하여 검토 할 수 있도록 하겠습니다.

출처 : https://link.chris-chris.ai/ai-lecture-12

스키마 추론과 스키마 환경

스키마 추론

```
schema = tfdv.infer_schema(statistics=train_stats)
tfdv.display_schema(schema=schema)
```

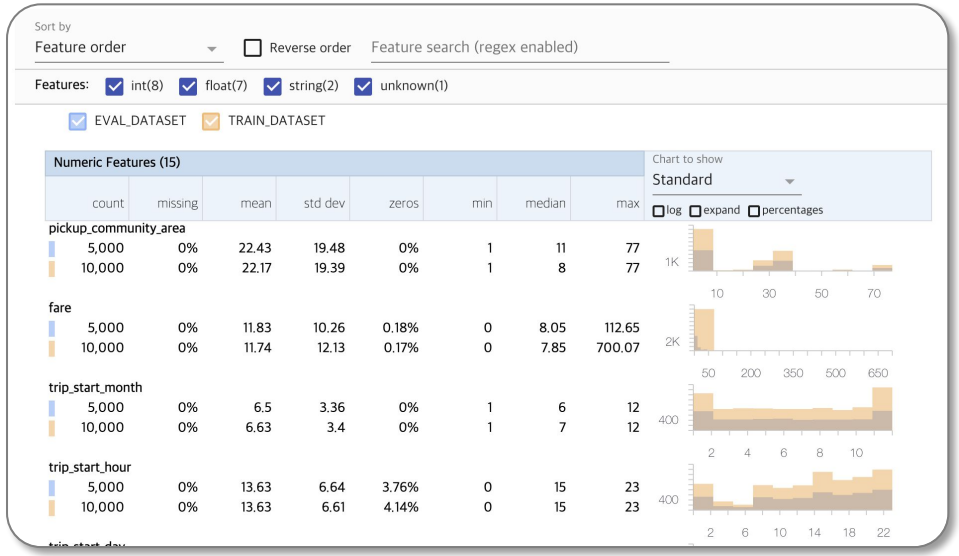
Feature name	Type	Presence	Valency	Domain
'pickup_community_area'	INT	required		-
'fare'	FLOAT	required		-
'trip_start_month'	INT	required		-
'trip_start_hour'	INT	required		-
'trip_start_day'	INT	required		-
'trip_start_timestamp'	INT	required		-
'pickup_latitude'	FLOAT	required		-
'pickup_longitude'	FLOAT	required		-
'dropoff_latitude'	FLOAT	optional	single	-
'dropoff_longitude'	FLOAT	optional	single	-

출처 : https://link.chris-chris.ai/ai-lecture-12

스키마 추론과 스키마 환경

평가 데이터의 오류 확인

```
# Compute stats for evaluation data
eval_stats = tfdv.generate_statistics_from_csv(data_location=EVAL_DATA)
# Compare evaluation data with training data
tfdv.visualize_statistics(lhs_statistics=eval_stats, rhs_statistics=train_stats,
                          lhs_name='EVAL_DATASET', rhs_name='TRAIN_DATASET')
```



출처 : https://link.chris-chris.ai/ai-lecture-12

스키마 추론과 스키마 환경

평가 데이터의 이상 데이터Anomaly 확인

```
# Check eval data for errors by validating the eval data stats using the previously
 inferred schema.
anomalies = tfdv.validate_statistics(statistics=eval_stats, schema=schema)
tfdv.display_anomalies(anomalies)
```

Feature name	Anomaly short description	Anomaly long description
'company'	Unexpected string values	Examples contain values missing from the schema: 2092 - 61288 Sbeih company (<1%), 2192 - 73487 Zeymane Corp (<1%), 2192 - Zeymane Corp (<1%), 2823 - 73307 Seung Lee (<1%), 3094 - 24059 G.L.B. Cab Co (<1%), 3319 - CD Cab Co (<1%), 3385 - Eman Cab (<1%), 3897 - 57856 Ilie Malec (<1%), 4053 - 40193 Adwar H. Nikola (<1%), 4197 - Royal Star (<1%), 585 - 88805 Valley Cab Co (<1%), 5874 - Sergey Cab Corp. (<1%), 6057 - 24657 Richard Addo (<1%), 6574 - Babylon Express Inc. (<1%), 6742 - 83735 Tasha ride inc (<1%).
'payment_type'	Unexpected string values	Examples contain values missing from the schema: Prcard (<1%).

출처 : https://link.chris-chris.ai/ai-lecture-12

스키마 추론과 스키마 환경

스키마의 평가 이상 수정

이런! 평가 데이터에는 Company 에 대한 새로운 값이 있지만 학습 데이터에는 없는 것 같습니다.
또한 Payment_Type 대한 새로운 값이 있습니다. 이것들은 비정상적인 것으로 간주되어야
하지만 이에 대해 우리가 결정하는 것은 데이터에 대한 도메인 지식에 달려 있습니다.
이상이 실제로 데이터 오류를 나타내는 경우 기본 데이터를 수정해야 합니다. 그렇지 않으면
평가 데이터 세트에 값을 포함하도록 스키마를 업데이트 할 수 있습니다.

평가 데이터 세트를 변경하지 않는 한 모든 것을 수정할 수는 없지만, 수용하기 편한
스키마를 수정할 수 있습니다. 여기에는 특정 피쳐의 이상 현상Anomaly이 무엇인지,
무엇이 아닌지에 대한 우리의 견해를 완화하고 범주 형 피쳐에 대한 누락 된 값을 포함하도록
스키마를 업데이트하는 것이 포함됩니다. TFDV를 통해 수정해야 할 사항을 발견 할 수
있었습니다.

지금 수정 한 다음 한 번 더 검토하겠습니다.

출처 : https://link.chris-chris.ai/ai-lecture-12

스키마 추론과 스키마 환경

스키마의 평가 이상 수정

```
# Relax the minimum fraction of values that must come from the domain for feature company.
company = tfdv.get_feature(schema, 'company')
company.distribution_constraints.min_domain_mass = 0.9

# Add new value to the domain of feature payment_type.
payment_type_domain = tfdv.get_domain(schema, 'payment_type')
payment_type_domain.value.append('Prcard')

# Validate eval stats after updating the schema
updated_anomalies = tfdv.validate_statistics(eval_stats, schema)
tfdv.display_anomalies(updated_anomalies)
```

No anomalies found.

출처 : https://link.chris-chris.ai/ai-lecture-12

스키마 추론과 스키마 환경

스키마 환경

또한 이 예제에서 'Serving' 데이터 세트를 분리 했으므로 이것도 확인해야합니다.
기본적으로 파이프 라인의 모든 데이터 세트는 동일한 스키마를 사용해야 하지만
예외가 있는 경우가 많습니다.

예를 들어 지도 학습에서 데이터 세트에 레이블을 포함해야 하지만
추론을 위해 모델을 제공 할 때는 레이블이 포함되지 않습니다.

어떤 경우에는 약간의 스키마 변형을 도입해야 합니다.

출처 : https://link.chris-chris.ai/ai-lecture-12

스키마 추론과 스키마 환경

스키마 환경

환경을 사용하여 이러한 요구 사항을 표현할 수 있습니다.
특히 Schema의 피쳐는 default_environment, in_environment 및 not_in_environment 사용하여 환경 세트와 연관 될 수 있습니다.

예를들어 이 데이터 세트에서 tips 피쳐는 학습용 라벨로 포함되어 있지만 제공 데이터에는 없습니다. 환경을 지정하지 않으면 예외로 표시됩니다.

```
serving_stats = tfdv.generate_statistics_from_csv(SERVING_DATA)
serving_anomalies = tfdv.validate_statistics(serving_stats, schema)

tfdv.display_anomalies(serving_anomalies)
```

Feature name	Anomaly short description	Anomaly long description
'tips'	Column dropped	Column is completely missing

출처 : https://link.chris-chris.ai/ai-lecture-12

스키마 추론과 스키마 환경

스키마 환경

```
options = tfdv.StatsOptions(schema=schema, infer_type_from_schema=True)
serving_stats = tfdv.generate_statistics_from_csv(SERVING_DATA, stats_options=options)
serving_anomalies = tfdv.validate_statistics(serving_stats, schema)

tfdv.display_anomalies(serving_anomalies)
```

Feature name		
'tips'	Column dropped	Column is completely missing

출처 : https://link.chris-chris.ai/ai-lecture-12

스키마 추론과 스키마 환경

스키마 환경

이제 비정상 ('열 삭제')으로 표시되는 tips 피쳐(라벨)가 있습니다.
물론 제공 데이터에 레이블이 있을 것으로 예상하지 않으므로 TFDV에 이를 무시하도록 지시합시다.

```
# All features are by default in both TRAINING and SERVING environments.
schema.default_environment.append('TRAINING')
schema.default_environment.append('SERVING')

# Specify that 'tips' feature is not in SERVING environment.
tfdv.get_feature(schema, 'tips').not_in_environment.append('SERVING')

serving_anomalies_with_env = tfdv.validate_statistics(
    serving_stats, schema, environment='SERVING')

tfdv.display_anomalies(serving_anomalies_with_env)
```

No anomalies found.

출처 : https://link.chris-chris.ai/ai-lecture-12

데이터 드리프트 및 스큐

데이터 드리프트

평가 데이터의 오류 확인

평가 데이터의 이상 데이터Anomaly 확인

스키마의 평가 이상 수정

스키마 환경

03

데이터 드리프트 및 스큐

드리프트 및 스큐 확인

데이터 세트가 스키마에 설정된 기대치를 준수하는지 확인하는 것 외에도
TFDV는 드리프트 및 스큐를 감지하는 기능도 제공합니다.

TFDV는 스키마에 지정된 드리프트 / 스큐 비교를 기반으로
여러 데이터 세트의 통계를 비교하여 이 검사를 수행합니다.

출처 : https://link.chris-chris.ai/ai-lecture-12

데이터 드리프트 및 스큐

드리프트

드리프트 감지는 범주 형 특성 및 데이터의 연속 범위 (즉, 범위 N과 범위 N + 1 사이)
(예 : 다른 훈련 데이터 날짜 사이)에 대해 지원됩니다.

L-Infinity Distance로 드리프트를 표현하며, 드리프트가 허용 가능한 것보다 높을 때 경고를 받을 수 있도록 임계 거리를 설정할 수 있습니다.

올바른 범위를 설정하는 것은 일반적으로 도메인 지식과 실험이 필요한 반복적 인 프로세스입니다.

출처 : https://link.chris-chris.ai/ai-lecture-12

데이터 드리프트 및 스큐

스큐

TFDV는 데이터에서 스키마 편향, 특성 편향 및 분포 편향의 세 가지 다른 종류의 편향을 감지할 수 있습니다.

- 스키마 스큐
- 특성 스큐
- 분포 스큐

출처 : https://link.chris-chris.ai/ai-lecture-12

데이터 드리프트 및 스큐

스키마 스큐 Schema Skew

스키마 스큐(Schema Skew)는 학습 및 서빙 데이터가 동일한 스키마를 따르지 않을 때 발생합니다. 학습 데이터와 서빙 데이터는 모두 동일한 스키마를 준수해야합니다.

둘 사이의 예상 편차 (예 : 학습 데이터에만 존재하지만 제공에는 없는 라벨)는
스키마의 환경 필드를 통해 지정해야 합니다.

출처 : https://link.chris-chris.ai/ai-lecture-12

데이터 드리프트 및 스큐

특성 스큐 Feature Skew

특성 스큐(Feature Skew)는 모델이 학습하는 특성 값이 서빙 시에 표시되는
특성 값과 다를 때 발생합니다. 예를 들어 다음과 같은 경우에 발생할 수 있습니다.

- 일부 특성 값이 학습 및 서빙 중간에 수정됩니다.
- 학습과 서빙 시에 특성을 전처리하는 로직이 다릅니다.
- 예를 들어, 학습 혹은 서빙 중 하나에만 일부 전처리를 적용하는 경우입니다.

출처 : https://link.chris-chris.ai/ai-lecture-12

데이터 드리프트 및 스큐

분포 스큐 Distribution Skew

분포 스큐(Distribution Skew)는 학습 데이터 세트의 분포가 제공 데이터 세트의 분포와 크게 다를 때 발생합니다.

분포 스큐의 주요 원인 중 하나는 다른 코드 또는 다른 데이터 소스를 사용하여
학습 데이터 세트를 생성하는 것입니다. 또 다른 이유는 학습 할 제공 데이터의
대표적이지 않은 하위 샘플을 선택하는 잘못된 샘플링 메커니즘입니다.

출처 : https://link.chris-chris.ai/ai-lecture-12

데이터 드리프트 및 스큐

드리프트 및 스큐

스키마가 검토되고 선별되었으므로 이제 "고정"상태를 반영하도록 파일에 저장합니다.

```
from tensorflow.python.lib.io import file_io
from google.protobuf import text_format

file_io.recursive_create_dir(OUTPUT_DIR)
schema_file = os.path.join(OUTPUT_DIR, 'schema.pbtxt’)
tfdv.write_schema_text(schema, schema_file)

!cat {schema_file}
```

출처 : https://link.chris-chris.ai/ai-lecture-12

데이터 드리프트 및 스큐

실습

https://link.chris-chris.ai/ai-lecture-12

출처 : https://link.chris-chris.ai/ai-lecture-12

짚어보기

데이터 검증 TFDV

01. 데이터 검증이 필요한 이유를 이해한다.

데이터 검증이 필요한 이유에 대해서 이해한다.

02. 스키마 추론과 스키마 환경을 이해한다.

스키마 추론과 스키마 환경에 대해서 이해한다.

03. 데이터 드리프트 및 스큐에 대해서 이해한다.

데이터 드리프트와 스큐가 어떤 의미인지 이해한다.

머신러닝 파이프라인

데이터 검증

TFDV TensorFlow Data Validation

송호연

감사합니다.

THANKS FOR WATCHING