11. 세이지메이커 Autopilot

2강. Autopilot을 이용한 모델 생성 및 배포

학습목표

- 세이지메이커 스튜디오에서 데이터 준비 및 Autopilot 실험 단계를 설명할 수 있다.
- 세이지메이커 스튜디오에서 Autopilot을 통해 생성된 모델 배포 및 정리단계를 설명할 수 있다.

학습내용

- 데이터 세트 준비
- Autopilot 실험 생성
- Autopilot 실험
- 최적 모델 배포 및 정리

■ 세상을 잇(IT)다!

- 은행 마케팅 데이터 세트
  - ✓ 고객이 정기 예금(변수 y)을 가입할지 예측하기 위한 포르투갈 금융 기관의 마케팅(전화통화) 데이터 세트

1. 나이	2. 직업	3. 결혼 여부
4. 교육	5. 신용정보	6. 주택 대출 여부
7. 개인 대출 여부	8. 연락 통신 유형	9. 마지막 연락 월
10. 마지막 연락 요일	11. 마지막 통화 시간(초)	12. 마케팅 관련 고객과 통화한 연락처 수
13. 이전 마케팅 이후 경과한 일수	14. 마케팅 이전에 고객과 통화한 연락처 수	15. 이전 마케팅 결과
16. 고용 변동률	17. 소비자 물가 지수	18.소비자 신뢰 지수
19. 3개월 금리	20. 고용인 수	21. (target) 정기예금 신청 여부

1. 데이터 세트 준비

- 데이터 세트 준비 단계 SageMaker Autopilot의 역할

① 새 노트북 생성

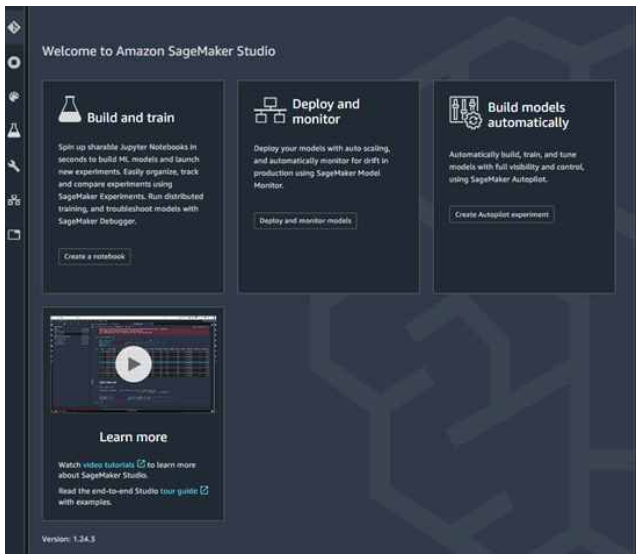
② 데이터 세트 다운로드 코드 실행

③ 데이터 확인

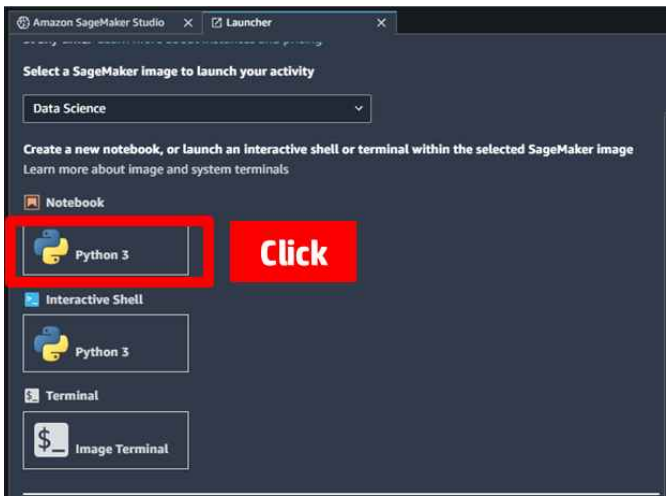
④ S3 버킷에 업로드

✓ 새 노트북 생성

- SageMaker 콘솔에 로그인 후 Studio 열기



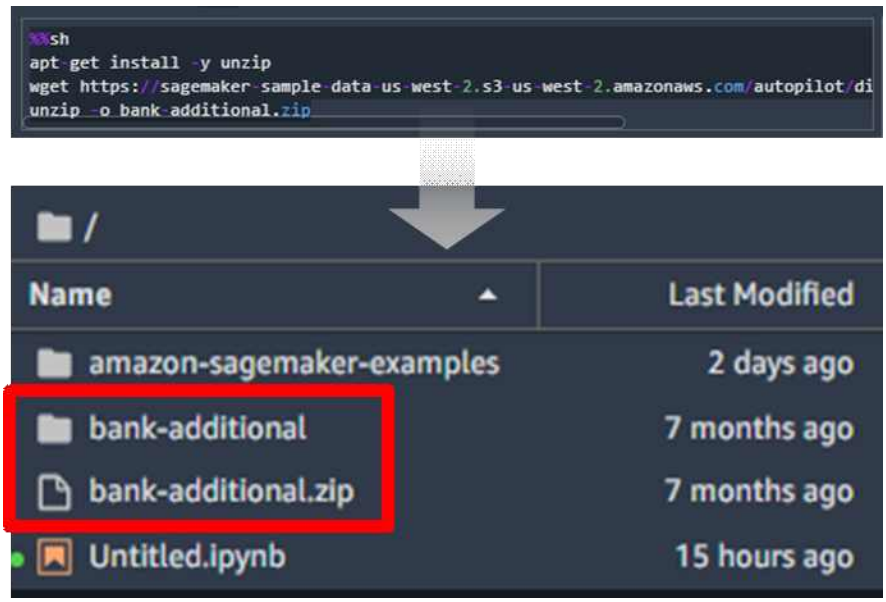
- Launcher 탭에서 [Notebook] 클릭



- 노트북 오른쪽 상단에서 커널 및 인스턴스 선택
- 커널 : Python 3(Data Science)
- 인스턴스 : ml.t3.meduim

✓ 데이터 세트 다운로드 코드 실행

- 데이터 코드를 다운로드하기 위하여 아래 코드 실행



- unzip 라이브러리를 활용하여 다운로드 한 데이터 압축 해제

```
%%sh

apt-get install -y unzip

wget https://sagemaker-sample-data-us-west-2
.s3-us-west-2.amazonaws.com/autopilot/
direct_marketing/bank-additional.zip

unzip -o bank-additional.zip
```

✓ 데이터 확인

- CSV 데이터 세트를 로드하여 첫 10줄이 표시

```
import pandas as pd
data = pd.read_csv('./bank-additional/bank-additional-full.csv')
data[:10]
```

	age	job	marital	education	default	housing	loan	contact
0	56	housemaid	married	basic.4y	no	no	no	telephone
1	57	services	married	high.school	unknown	no	no	telephone
2	37	services	married	high.school	no	yes	no	telephone
3	40	admin.	married	basic.6y	no	no	no	telephone
4	56	services	married	high.school	no	no	yes	telephone
5	45	services	married	basic.9y	unknown	no	no	telephone
6	59	admin.	married	professional.course	no	no	no	telephone
7	41	blue-collar	married	unknown	unknown	no	no	telephone
8	24	technician	single	professional.course	no	yes	no	telephone
9	25	services	single	high.school	no	yes	no	telephone

10 rows × 21 columns

- bank-additional-full.csv 파일을 로드하여 첫 10줄을 출력

```
import pandas as pd
data = pd.read_csv('./bank-additional/bank-additional-full.csv')
data[:10]
```

✓ S3 버킷에 업로드

- csv 형태의 데이터 세트를 S3 버킷에 업로드
- 코드 출력 시 표시되는 S3 버킷 URI를 보관하기

```
import sagemaker

prefix = 'sagemaker/tutorial-autopilot/input'
sess   = sagemaker.Session()

uri = sess.upload_data(path="./bank-additional/bank-additional-full.csv", k
print(uri)
```

```
s3://sagemaker-us-east-1-[redacted]/sagemaker/tutorial-autopilot/input/bank-additional-full.csv
```

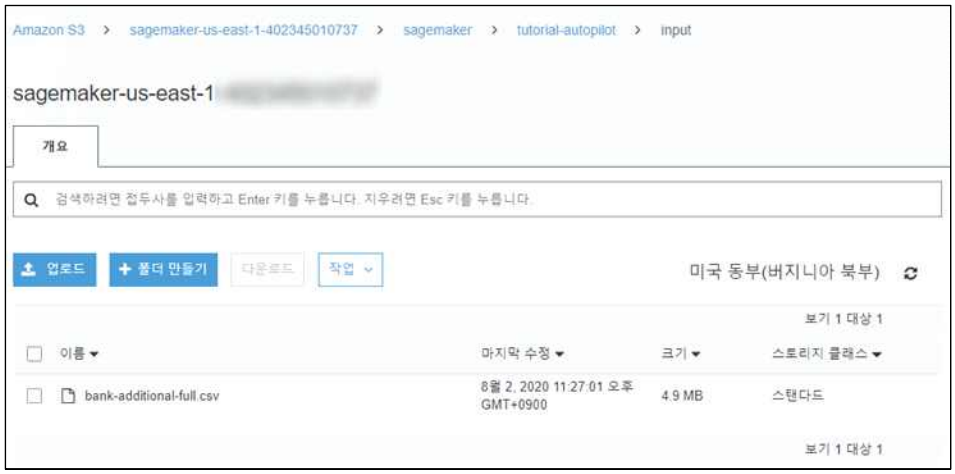
✓ 데이터를 업로드하면 세이지메이커에서 계정에 기본 버킷을 자동으로 생성하므로 S3 버킷을 직접 생성할 필요가 없음

```
import sagemaker

prefix = 'sagemaker/tutorial-autopilot/input'
sess   = sagemaker.Session()

uri = sess.upload_data(path="./bank-additional/bank-additional-full.csv",
key_prefix=prefix)
print(uri)
```

- 데이터세트 업로드 경로
- S3/버킷/sagmemaker/tutorial-autopilot/input/bank-additional-full.csv



2. Autopilot 실험 생성

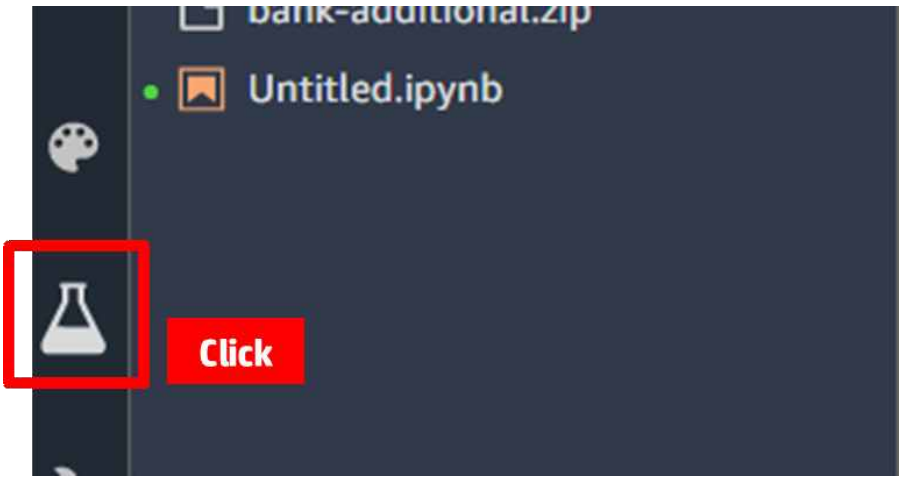
- Autopilot 실험 생성 단계

① 실험생성

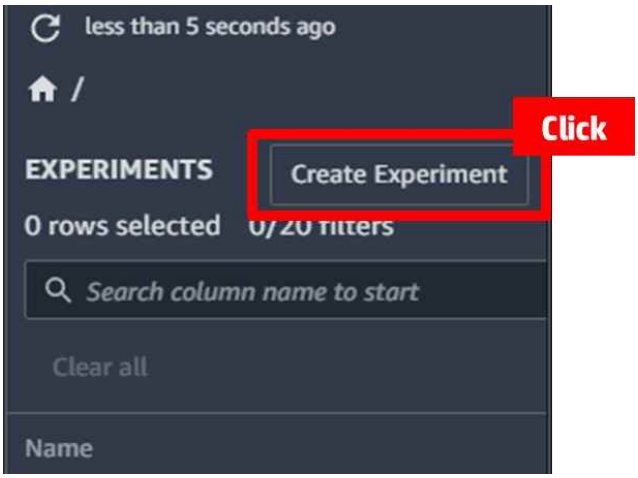
② 실험 작업 설정

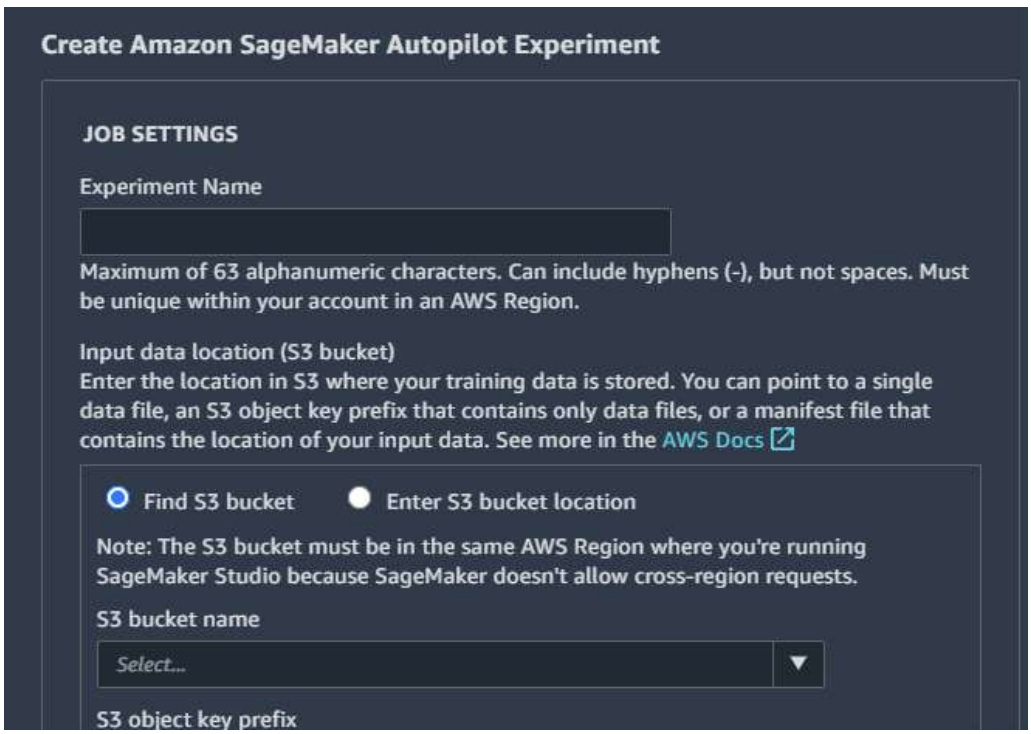
✓ 실험 생성

- Studio 왼쪽 창에서 실험 아이콘 클릭



- 실험 생성 클릭





✓ 실험 작업 설정

- 작업 설정 필드에 다음과 같은 정보를 입력

① 실험 이름

② 입력 데이터의 S3 위치

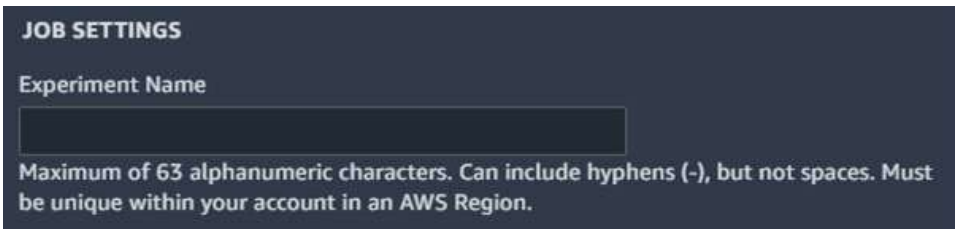
③ 대상 속성

④ 출력 데이터의 S3 위치

⑤ 머신러닝 문제 유형

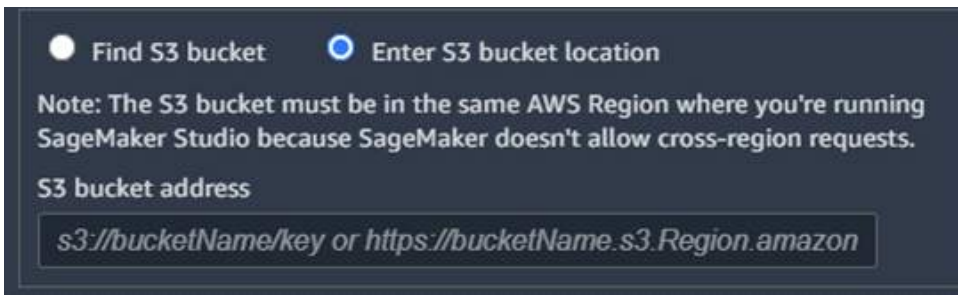
- 실험 이름

- 현재 AWS 리전에서 이 실험에 대한 이름은 고유해야 함
- 영·숫자 및 하이폰(-)을 포함하여 최대63 자 입력 가능
- 공백은 포함 안 됨



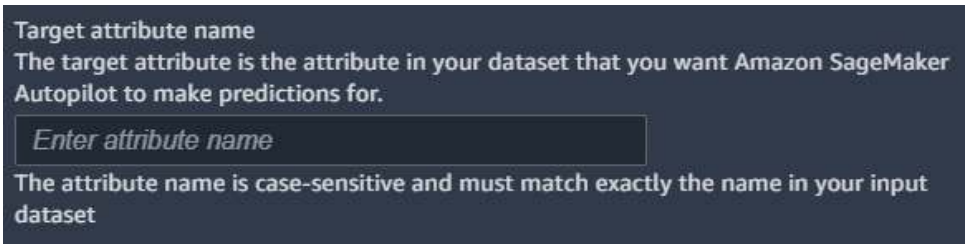
- 실험 작업 설정

- 세이지메이커가 읽기 권한을 가지고 있는 S3:// 형식의 URL
- 입력 데이터는 CSV 형식이어야 하며 1000개 이상의 행을 포함해야 함



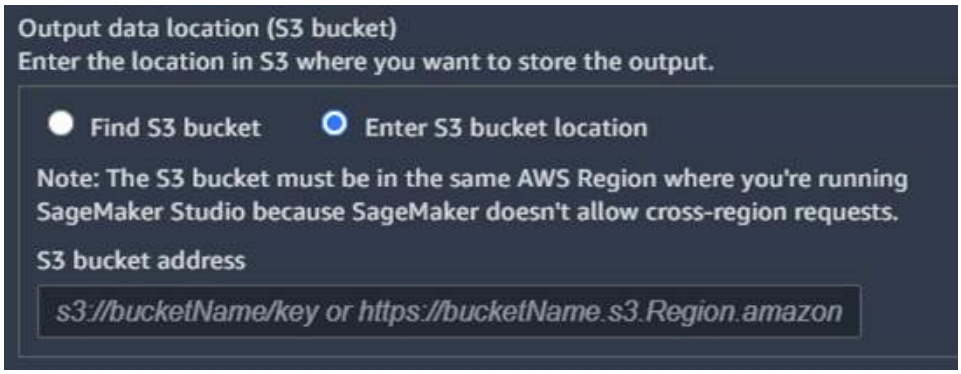
- 대상 속성

- 모델이 대상(타겟)으로 삼을 데이터의 열
- 속성 이름은 대소문자를 구분하며 입력 데이터 세트의 이름과 정확히 일치해야 함



- 출력 데이터의 s3 위치

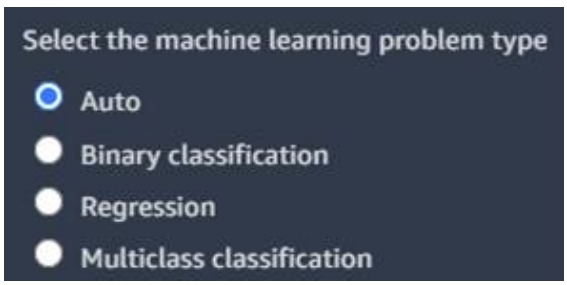
■ 세이지메이커가 쓰기 권한을 가지고 있는 s3://형식의 URL



- 머신러닝 문제 유형

■ 회귀(예:주택 가격), 이진 분류(예:핫도그인지 아닌지), 멀티클래스 분류(고양이vs개) 중 원하는 모델을 지정

■ 모델을 평가하는데 사용할 지표를 설정 가능



- 작업 설정 입력 후 [Creat Experiment] 선택

■ 실험 이름 : bdu-test-autopilot

■ 입력 데이터의 s3 위치:s3://sagemaker-us-east-1-[ACCOUNT-NUMBER]/sagemaker/tutorial-autopilot/input/bank-additional-full.csv

■ 대상 속성 이름 : y

■ 출력할 데이터의 위치 : s3://sagemaker-us-east-1-[ACCOUNT-NUMBER]/sagemaker/tutorial-autopilot/output/

■ [ACCOUNT-NUMBER]는 실제 계정 번호로 바꿔야 함

■ 머신러닝 문제 유형 : Auto

■ 나머지 설정은 모두 기본값

3. Autopilot 실험

- Autopilot 실험 단계

① 데이터 분석

② 기능(Feature) 엔지니어링

③ 모델 튜닝

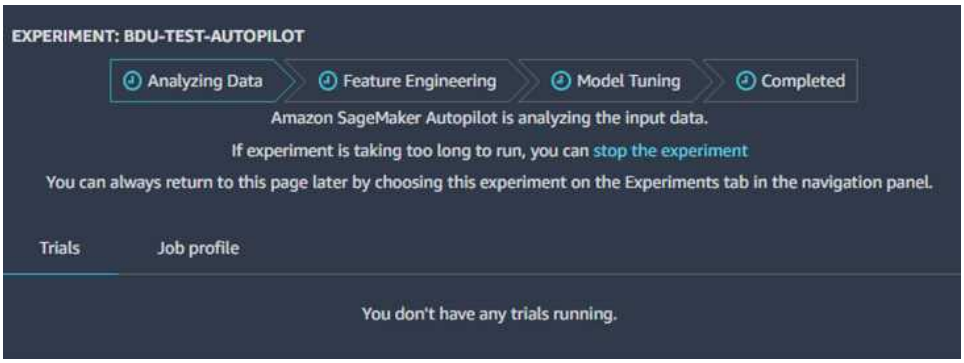
✓ 데이터 분석

- 데이터 분석 단계에서는 해결할 문제 유형을 식별
- 후보 파이프라인 10개가 제시
- 파이프라인에 포함된 기능

① 누락된 값 처리/새 기능 엔지니어링 등의 데이터 사전 처리 단계

② 문제 유형과 일치하는 ML알고리즘을 사용하는 모델 훈련 단계가 포함

- 세이지메이커 Autopilot이 입력 데이터 분석 단계



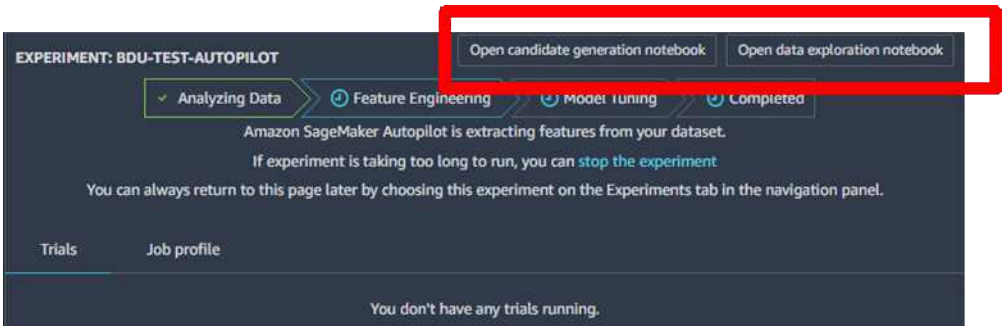
✓ 기능 엔지니어링

- 각 후보 파이프라인용 훈련 및 검증 데이터 세트가 생성되며 s3 버킷에 모든 아티팩트가 저장
- 기능 엔지니어링 단계에서는 자동 생성 노트북 두 개를 열어서 확인할 수 있음

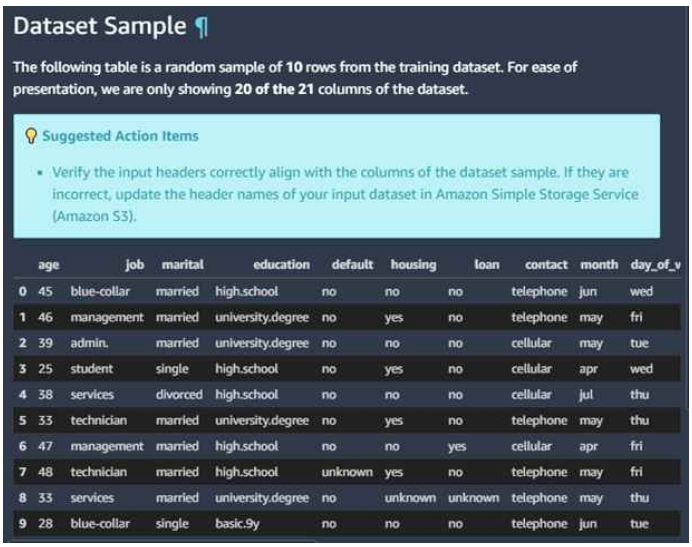
① 후보 생성 노트북 (Open candidate generation notebook) : 파이프라인 10개의 정의가 포함

② 데이터 탐색 노트북 (Open data exploration notebook) : 데이터 세트 관련 정보와 통계가 포함

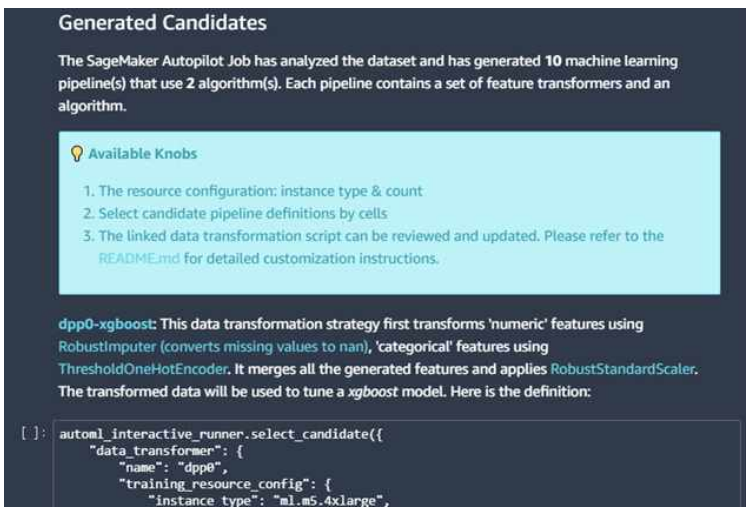
- 두 노트북을 활용하여 데이터가 사전 처리되는 방식 및 모델 구축/최적화 방식을 자세히 파악할 수 있음



- 데이터 탐색 노트북

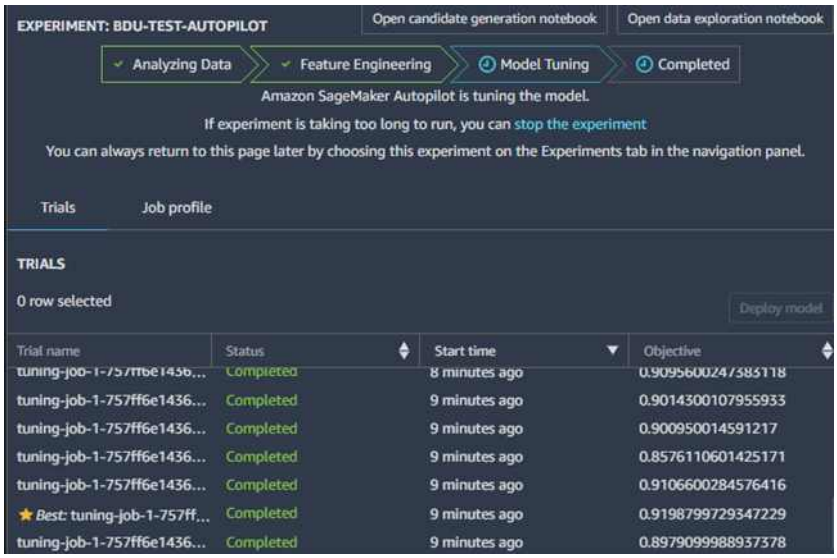


- 후보 생성 탐색 노트북



✓ 모델 튜닝

- 각 후보 파이프라인과 사전 처리된 데이터 세트에 대해 하이퍼파라미터 최적화 작업



4. 최적 모델 배포 및 정리

- 최적 모델 배포 및 정리 단계

① 실험 시도 목록 정렬

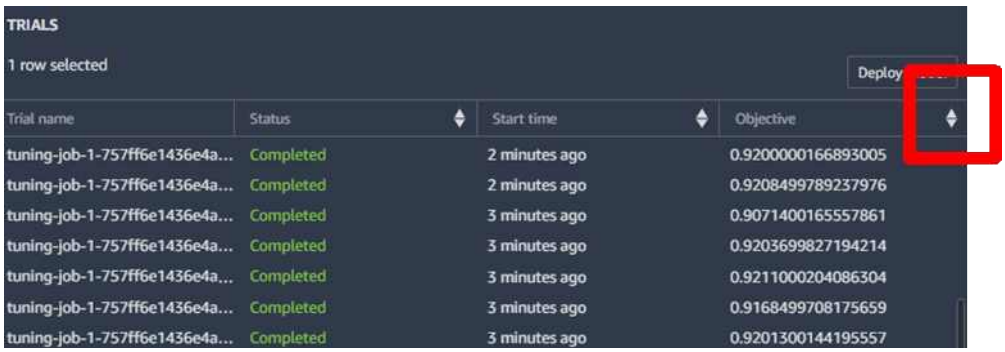
② 모델 배포

③ 배포된 모델을 사용한 예측

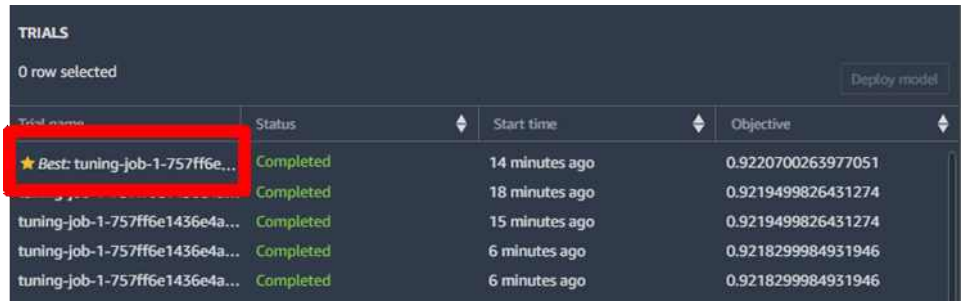
④ 리소스 정리

✓ 실험 시도 목록 정렬

- 실험의 시도 목록에서 목록 옆의 정렬 아이콘 ()선택

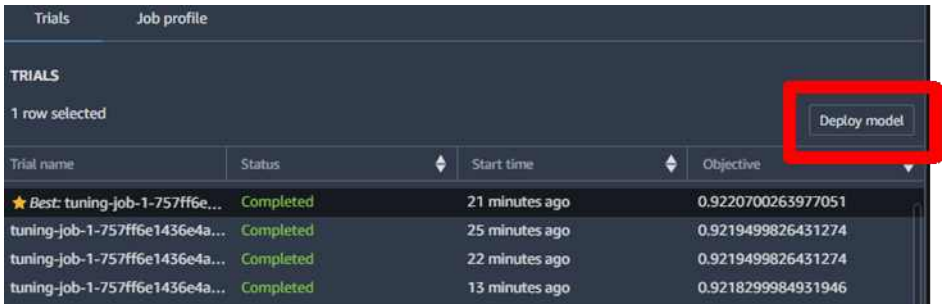


- 최적의 튜닝 작업이 별표로 강조 표시됨

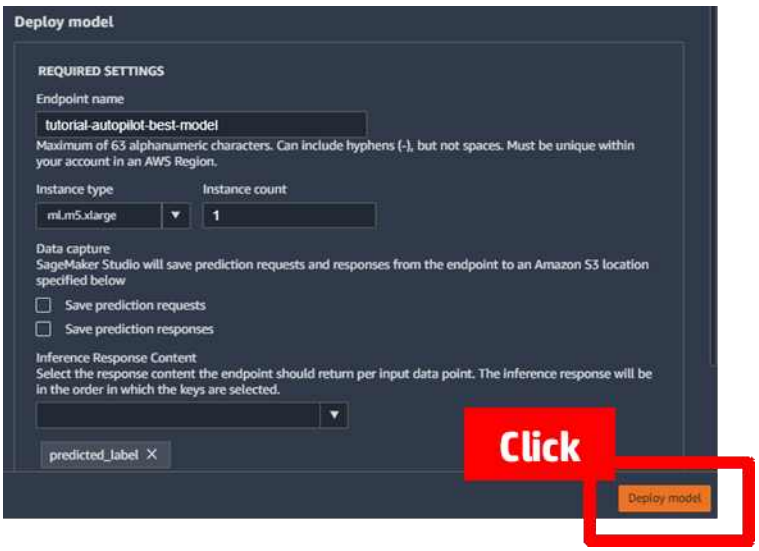


✓ 모델 배포

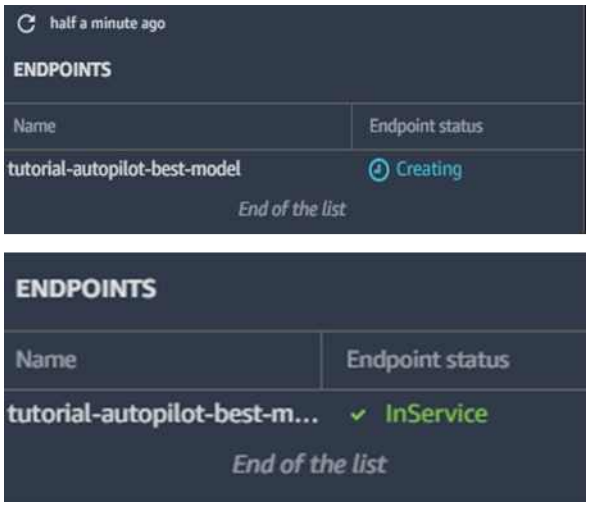
- 별표가 표시된 최적 튜닝 작업을 선택하고 모델 배포 선택



- 모델 배포 상자에 엔드포인트 이름을 입력하고 모든 설정은 기본값으로 유지 후 모델 배포 선택



- 왼쪽 도구 모임에서 엔드포인트 아이콘 ( )클릭 후 엔드포인트 상태가 InService로 바뀌면 데이터를 전송하고 예측 정보를 수신할 수 있음



✓ 배포된 모델을 사용한 예측

- 모델을 배포 했으므로 데이터 세트의 첫 2,000개 샘플을 예측해보기
- 예측에는 boto3 SDK의 Invoke_endpoint API를 사용
- 주요 기계 학습 지표인 정확도(accuracy), 정밀도(precision), 회수(recall), f1 점수를 계산

```
import boto3, sys
ep_name = 'end point name'
sm_rt = boto3.Session().client('runtime.sagemaker')
tn=tp=fn=fp=count=0
with open('bank-additional/bank-additional-full.csv') as f:
    lines = f.readlines()
    for l in lines[1:2000]:   # Skip header
        l = l.split(',')      # Split CSV line into features
        label = l[-1]         # Store 'yes'/'no' label
        l = l[:-1]            # Remove label
        l = ','.join(l)       # Rebuild CSV line without label
```

```
response =
sm_rt.invoke_endpoint(EndpointName=ep_name,
                                    ContentType='text/csv',
                                    Accept='text/csv', Body=l)
        response = response['Body'].read().decode("utf-8")
        #print ("label %s response %s" %(label,response))
  if 'yes' in label:  # Sample is positive
            if 'yes' in response:  # True positive
                tp=tp+1
            else:  # False negative
                fn=fn+1
```

```
 else:  # Sample is negative
            if 'no' in response:
                # True negative
                tn=tn+1
            else:  # False positive
                fp=fp+1
        count = count+1
        if (count % 100 == 0):
            sys.stdout.write(str(count)+' ')

print ("Done")
```

```
accuracy  = (tp+tn)/(tp+tn+fp+fn)
precision = tp/(tp+fp)
recall    = tn/(tn+fn)
f1        = (2*precision*recall)/(precision+recall)

print ("%.4f %.4f %.4f %.4f" % (accuracy, precision, recall,
f1))
```

- 출력결과

- accuracy : 0.9815
- precision : 0.6190
- recall : 0.9853
- f1 : 0.7604

생각해보기

- 분류성능평가 지표
  - ✓ Accuracy(정확도) = (tp+tn)/(tp+tn+fp+fn)
  - ✓ Precision(정밀도) = tp/(tp+fp)
  - ✓ Recall(재현율) = tn/(tn+fn)
  - ✓ f1(조화평균) = (2*precision*recall)/(precision+recall)

		예측결과	
		TRUE	FALSE
실제 정답	TRUE	TP (True Positive)	FN (False Negative)
	FALSE	FP (False Positive)	TN (True Negative)

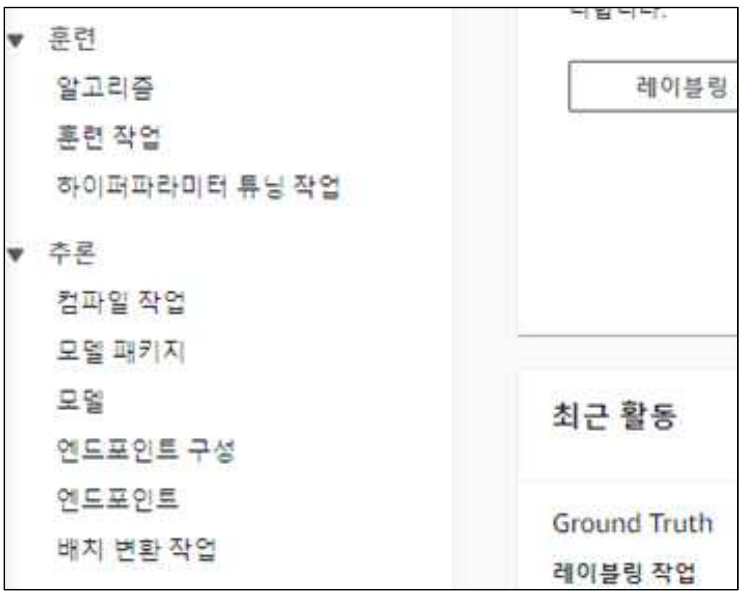
✓ 리소스 정리

- 모델, 사전 처리한 데이터 세트 등의 모든 훈련 아티팩트 등 사용한 리소스를 종료하기 위하여 아래 코드 실행

```
sess.delete_endpoint(endpoint_name=ep_name)

%%sh
aws s3 rm --recursive s3://sagemaker-us-east-1-ACCOUNT_NUMBER/sagemaker/tutorial-autopilot/
```

- 세이지메이커 콘솔 창에서 왼쪽 대시보드에서 모델, 엔드포인트 구성 등을 선택하여 삭제
- s3 콘솔 접속 후 버킷 삭제



평가하기

1. Autopilot 실험 생성을 위하여 작업 설정 필드에서 입력해야 할 정보가 아닌 것은?

① 입력 데이터 S3 위치

② 실험 이름

③ 대상 속성

④ 알고리즘

- 정답 : ④번

해설 : Autopilot 실험 생성을 위한 작업 설정필드에서는 실험이름, 입력 데이터의 S3위치,대상 속성, 머신러닝 문제 유형이 있습니다.

2. Autopilot 실험에서 각 후보 파이프라인과 사전처리 된 데이터 세트에 대하여 하이퍼파라미터 최적화 작업을 하는 단계는?

① 기능 엔지니어링

② 작업 설정

③ 모델 튜닝

④ 데이터 분석

- 정답 : ③번

해설 : 모델 튜닝 단계에서 세이지메이커 Autopilot은각 후보 파이프라인과 사전 처리된 데이터 세트에대해 하이퍼파라미터 최적화 작업을 수행합니다.

학습정리

1. 데이터세트 준비

- 새 노트북 생성
- 데이터 세트 다운로드 코드 실행
- 데이터 확인
- S3 버킷에 업로드

2. Autopilot 실험 생성

- 실험 생성
- 실험 작업 설정

3. Autopilot 실험

- 데이터 분석
- 기능 엔지니어링
- 모델 튜닝

3. 최적 모델 배포 및 정리

- 실험 시도 목록 정렬
- 모델 배포
- 배포된 모델을 사용한 예측
- 리소스 정리