목적 및 데이터 소개

'Toyota 서비스센터 이용 고객 데이터'

Elasticsearch의
Machine Learning에 학습시켜 'EFA' 옵션 사용을 예측
- EFA : 고객의 유료서비스 사용 여부

데이터
- 학습 데이터 : **train_rosa_pre_data.csv (2020.08~2022.06) - 290585건**
- 예측 대상 데이터 : **test_rosa_pre_data.csv (2022.07) - 12544건**

학습 데이터	
CENTER	서비스센터
RO	제안서번호
CUST_SEQ	고객번호
CUST_TYPE	고객유형
CPS_3YR_PROPO	3년유상금액
VISIT_3YR_PROPO	3년방문횟수
VIN	차대번호
SALES_TYPE	판매유형
BRAND	브랜드
HEV	하이브리드여부
HP	엔진출력
TAKING_FIX	승차정원
VEHICLE_AGE_PROPO	차령(제안서)
MILEAGE_PROPO	주행거리(제안서)
GR_BP	수리작업구분
KPI_CATEGORY	KPI구분
12V	
TIRE	
EFA	
Air Care	

추론 데이터	
VIN	차대번호
BRAND	브랜드
HEV	하이브리드여부
VEHIC_AGE_SYS	차령
MILEAGE_SYS	현재주행거리
HP	원동기마력
TAKING_FIX	승차정원
CPS_3YR_SYS	기준3년 유상서비스액
VISIT_3YR_SYS	기준3년간 현재 입고회수

데이터 정제

1. 날짜 필드 정제

입고 순서 제거, Python의 datetime 형식으로 변경

2020-08-01**(1)** > 2020-08-01

2. 차대번호 필드 분리

- 차량 특성(CHARACTERISTICS)
- 생산년도(MODEL_YEAR)

> 2가지 필드 생성

3. target 변수 0, 1로 바꿔줌

Null -> 0
'EFA' -> 1

4. 필요없는 필드 제거

12V, TIRE, Air Care, RO (처음 날짜가 있던 필드) 필드 삭제

12V, TIRE, Air Care : Null 값이 많고 유료고객 예측에 필요없다고 판단

모델

1. Elastic에 데이터 import 후 Data Frame Analytics > Jobs > Create job
2. Classification model train & 결과 확인

Normalized confusion matrix for testing dataset

Predicted class

Columns

Actual class	0	1
0	74%	26%
1	27%	73%

Evaluation quality metrics

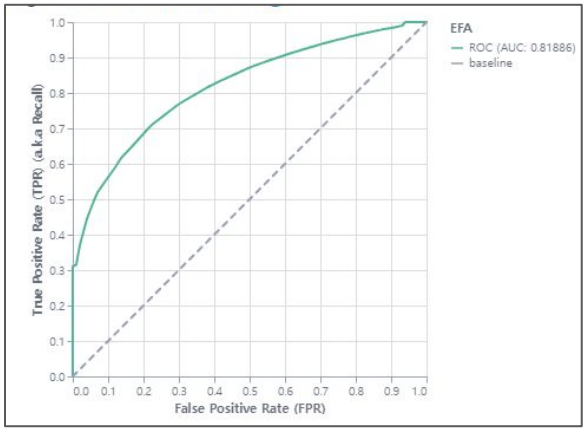
0.738 Overall accuracy

0.733 Mean recall

Per class recall and accuracy

Class	Accuracy	Recall
0	0.738	0.74
1	0.738	0.727

분류기 성능 - ROC (Receiver Operating Characteristics)



AUC : 0.82

모델 학습 결과		학습 데이터의 'EFA' Value	
		0	1
ML(Confusion matrix)	0	**0.74%**	0.26%
	1	0.27%	**0.73%**

분류를 통한 클래스 학습 참고 링크

예측 결과

1. 방법

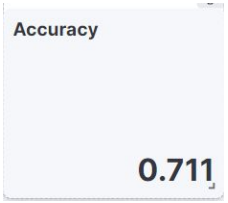
- 학습 결과 Deploy -> ingest pipeline 형성
- 미리 import 한 test data를 pipeline 통과

```
PUT /test_rosa_p1/_settings
{
  "index": {
    "default_pipeline": "new-ml-inference-indexed_rosa"
  }
}

POST _reindex
{
  "source": {
    "index": "test_rosa_pre_data_index"  // 원본 인덱스
  },
  "dest": {
    "index": "test_rosa_p1"  // 새로운 인덱스 (선택 사항)
  }
}
```

2. 정확도

- **dash board**




```
(count(kql='EFA :0 and
ml.inference.EFA_prediction.EFA_prediction :0')+
count(kql='EFA :1 and
ml.inference.EFA_prediction.EFA_prediction:1'))/
count()
```

- **Confusion Matrix**

		EFA_prediction	
		0	1
EFA	0	7926	3188
	1	386	1043