[분석 개요]

1. 사용 언어 : Python
2. 분석 환경 : [CPU] Apple M2, [RAM] 8GB, [GPU] Colab T4
3. 분석 데이터 정보
   1. 수집 기관 : Ford 자동차 제조사
   2. 데이터 내용 : 엔진 상태와 관련 있는 센서 500개의 측정 값
   3. 데이터 파일 : Train Dataset 1file, Test Dataset 1file
4. 분석 데이터 구조
   1. Features : 500 columns (Sensor values)
   2. Samples : 4921 raw (Train 3601, Test 1320)
   3. Labels : 1 or 0 (Normal or Abnormal)
5. 학습 및 예측 모델
   1. LogisticRegression
   2. TabularPredictor (Auto ML)
   3. XGBClassifier
   4. LGBMClassifier
   5. CatBoostClassifier
   6. RandomForestClassifier
   7. CNN
   8. RNN
6. 예측 결과

[데이터 분석]

1. 데이터를 불러와서 DataFrame 형식으로 변환

텍스트, 폰트, 스크린샷이(가) 표시된 사진

자동 생성된 설명

1. Train, Test dataset info.

텍스트, 스크린샷, 폰트이(가) 표시된 사진

자동 생성된 설명

* + train\_df : 3601 samples, 501 columns (Include target 1)
  + test\_df : 1320 samples, 501 columns (Include target 1)
  + 500 feature type : float64, target type : object

1. Target value 1과 0으로 변환

텍스트, 폰트, 스크린샷이(가) 표시된 사진

자동 생성된 설명

1. Target value 빈도의 균형이 맞는지 확인

텍스트, 스크린샷, 소프트웨어, 도표이(가) 표시된 사진

자동 생성된 설명

* + train, test dataset 의 target value 빈도는 균형이 맞다.

1. features value 분포 확인

텍스트, 스크린샷, 폰트, 번호이(가) 표시된 사진

자동 생성된 설명

1. 0과 1 클래스 각 샘플의 센서 값 분포 비교 (각 3개 샘플 비교)

텍스트, 스크린샷, 폰트, 라인이(가) 표시된 사진

자동 생성된 설명

* + 클래스별 센서 값에 차이가 있음 -> 학습 가능할 것으로 보임
  + 전체 샘플의 센서 별 평균도 볼 필요 있음 (클래스 구분)

1. 센서 별 클래스 평균 값 분포

텍스트, 스크린샷, 폰트, 소프트웨어이(가) 표시된 사진

자동 생성된 설명

도표, 그래프, 스크린샷이(가) 표시된 사진

자동 생성된 설명

* + 센서 후반부 (s403 ~) 클래스에 따라 센서 값이 양수와 음수로 나눠짐 -> 후반부 센서만 학습시켜 볼 필요 있음

1. 전체 샘플에 대한 센서 값들의 분포 (클래스별)

텍스트, 스크린샷, 라인, 폰트이(가) 표시된 사진

자동 생성된 설명

* + 각 샘플 값이 시계열 데이터일 경우 시간의 흐름에 따른 패턴은 보이지 않음
  + RNN 모델로 학습 해볼 필요는 있음.
  + 클래스별 평균 값으로 다시 시각화 해볼 필요 있음 (다음 장)

1. 샘플별 500개 센서 값의 평균

텍스트, 스크린샷, 폰트, 번호이(가) 표시된 사진

자동 생성된 설명

* + 출력 결과 다음 페이지

스크린샷, 텍스트, 라인, 도표이(가) 표시된 사진

자동 생성된 설명

* + 0클래스에 노이즈 샘플이 있음.
  + 노이즈 샘플 제거 후 학습 해볼 필요 있음.

1. 센서 별 상관관계 분석

텍스트, 스크린샷, 다채로움, 디자인이(가) 표시된 사진

자동 생성된 설명

인접한 센서와 상관계수가 높음

* CNN으로 학습 가능 할

것으로 보임

* kernel\_size를 3으로하면

학습이 잘될 것으로 예상

1. Target에 영향을 주는 센서 확인

텍스트, 스크린샷, 디자인이(가) 표시된 사진

자동 생성된 설명

* + 앞서 확인 한대로 후반부(s403 ~)에 target에 영향을 주는 센서가 있음.
  + Target과 상관계수가 높은 센서들만 학습해볼 필요 있음.

1. Auto ML(AutoGluon)을 사용하여 최적의 머신러닝 모델 확인하기

텍스트, 폰트, 스크린샷이(가) 표시된 사진

자동 생성된 설명

[학습 및 예측 결과]

AutoGluon training complete, total runtime = 90.28s ... Best model: "WeightedEnsemble\_L2"

TabularPredictor saved. To load, use: predictor = TabularPredictor.load("AutogluonModels/ag-20230816\_071132/")

Best model을 test 데이터셋으로 평가했을 때 정확도 : 0.8469696969696969

[Leaderboard 확인]



텍스트, 스크린샷, 폰트, 문서이(가) 표시된 사진

자동 생성된 설명

* + 14개 학습 모델 중 WeightedEnsemble\_L2 모델의 정확도가 가장 높음 (84.7%)
  + train, test 데이터셋 라벨 분리 (x, y) – 개별 ML 학습을 위해 (다음 페이지)

텍스트, 폰트, 스크린샷이(가) 표시된 사진

자동 생성된 설명

1. 머신러닝 4가지 모델로 학습 및 예측

텍스트, 스크린샷, 문서, 폰트이(가) 표시된 사진

자동 생성된 설명

텍스트, 스크린샷, 문서, 폰트이(가) 표시된 사진

자동 생성된 설명

텍스트, 폰트, 영수증, 대수학이(가) 표시된 사진

자동 생성된 설명

* + CatBoost모델의 성능이 가장 좋음
    - 기본 threshold(0.50) 에서 Accuracy 84.55%
    - 최적의 threshold(0.46) 에서 Accuracy 85.30%

텍스트, 스크린샷, 도표, 평행이(가) 표시된 사진

자동 생성된 설명

* + 혼동 행렬의 4가지 항목 (TP, TN, FP, FN) 모두 CatBoost 성능이 가장 좋음
  + 모델별 ROC Curve에서 AUC 값
    - XGBoost : 0.85, LGBM : 0.86, CatBoost : 0.91, RandomForest : 0.81
  + Feature\_importances\_ 값은 각 특성의 중요도를 나타냄
    - 각 특성이 모델의 예측에 얼마나 기여했는지
  + Featrue\_importances\_ 값은 target 상관관계 값과 관련 없음
    - 각 상위 10개 센서 중 중복 되는 센서 없음

텍스트, 스크린샷, 폰트, 번호이(가) 표시된 사진

자동 생성된 설명텍스트, 스크린샷, 폰트, 그래프이(가) 표시된 사진

자동 생성된 설명

1. CNN 학습 및 결과 예측

텍스트, 스크린샷, 폰트, 문서이(가) 표시된 사진

자동 생성된 설명

텍스트, 스크린샷, 문서, 폰트이(가) 표시된 사진

자동 생성된 설명

[CNN 학습 결과]

하이퍼 파라미터 최적 조건

텍스트, 스크린샷, 번호, 폰트이(가) 표시된 사진

자동 생성된 설명

**🡪 최적 조건으로 테스트 데이터셋 정확도 97.2 ~ 97.5%**

* Conv 1D Layer
  + 최적 조건 : 3 Layer
  + 2Layer 또는 4Layer일때 정확도 떨어짐(95% 이하), 5Layer일때 과적합 발생
* 활성 함수
  + 최적 함수 : LeakyReLU
  + Relu, swish, softplus에 비해 과적합이 덜 일어나고, 정확도도 가장 좋게 나타남
* Optimizer
  + 최적 함수 : adam
  + SGD 일때 400epoch기준 정확도 떨어짐(92%), 2000epoch 까지 늘려도 97%에 도달하지 못함.
* 필터 수
  + 최적 필터 수 : 32
  + 필터 수가 많다고 해서 성능이 좋아지는 것은 아님. 최적의 필터 수는 실험 결과 32가 가장 적합함.
* Batch\_size
  + 최적 배치 수 : 16~32
  + 배치 수가 많을 수록 학습 속도는 빨라지나 정확도가 떨어짐
* 최적모델 저장 기준(monitor)
  + 최적 기준 : ‘val\_sparse\_categorical\_accuracy’
  + ’val\_loss’ 를 기준으로 할 경우 정확도가 0.5% 정도 떨어짐
* 검증 데이터 비중
  + 최적 비중 : 15~20% 일때 성능이 가장 좋게 나타남
* Kernel\_size
  + 최적 커널 수 : 3
  + 커널 수가 2, 4, 5일 경우 정확도가 떨어짐 (94% 이하)
  + 센서 간 상관계수에서 살펴본대로 인접 센서와 연관성이 높아 길이 3의 패턴이 의미있는 특성을 가지고 있는 것으로 보임
* 데이터 정규화(StandardScaler)
  + 최적 조건 : 데이터 정규화 불필요
  + 정규화를 진행했을 경우 머신러닝, CNN모델 모두 전/후 정확도가 비슷하거나 약간 낮아지는 정도
* 배치 정규화(BatchNormalization)
  + 최적 조건 : 배치 정규화 필요
  + 배치 정규화 안했을 경우 성능 크게 떨어짐 (92% 이하)
* Epoch
  + 최적 조건 : 400회
  + Epoch 400회 초과 되어도 성능이 더 이상 개선 되지 않음

(단, SDG optimizer 사용했을 경우는 예외)

[훈련/검증 정확도, 훈련/검증 손실 그래프]

* + 하이퍼파라미터 최적 조건으로 학습 했을 때

텍스트, 스크린샷, 그래프, 라인이(가) 표시된 사진

자동 생성된 설명

[CNN 혼동행렬과 ROC Curve]

텍스트, 스크린샷, 폰트, 문서이(가) 표시된 사진

자동 생성된 설명

도표, 라인, 그래프, 스크린샷이(가) 표시된 사진

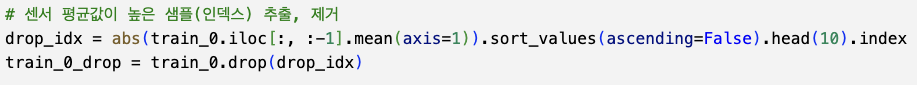
자동 생성된 설명

* + CNN 모델 정확도 97.2% 이상, AUC = 0.99 로 학습이 잘 됨

1. 데이터 전처리 후 재학습(1)

--- 500개 전체 센서의 평균 값이 높은 샘플 데이터 제거 후 학습

* + 평균 값 상위 10개 샘플 제거



* + 상위 10개 샘플 제거 후 시각화

텍스트, 스크린샷, 라인, 도표이(가) 표시된 사진

자동 생성된 설명 스크린샷, 텍스트, 도표, 디자인이(가) 표시된 사진

자동 생성된 설명

* + 상위 10개 샘플 제거 후 학습 및 예측 결과

1. 데이터 전처리 후 재학습 (2)
   * S403 센서 이후 데이터만 학습
2. 데이터 전처리 후 재학습 (3)
   * Target과 상관계수 값 상위 센서로만 학습