엘리베이터 고장 예측 모델

2318012 안전공학과

김준영

a. 프로젝트에 관한 전체 내용을 요약

이 프로젝트는 엘리베이터 상태(정상 또는 고장)를 예측하는 머신러닝 모델을 개발하기 위한 것입니다. 이를 위해 제공된 센서 데이터를 전처리하고, Random Forest Classifier를 사용하여 모델을 학습시켰습니다. 개발 과정에서 데이터 분석, 시각화, 전처리, 학습 및 평가를 포함하여 전체 프로세스를 체계적으로 수행했습니다.

---

2. 개발 목적

a. 머신러닝 모델 활용 대상

이 모델은 엘리베이터의 운영 상태를 실시간으로 모니터링하고, 잠재적 고장을 예측하여 사전에 유지보수를 수행하는 데 활용됩니다.

b. 개발의 의의

- **안전성 향상**: 엘리베이터 고장을 미리 예측함으로써 사고 위험을 줄일 수 있습니다.
- **유지보수 비용 절감**: 사전 예측을 통해 비계획적인 다운타임을 최소화합니다.
- **운영 효율성 증가**: 예측 데이터를 기반으로 적절한 유지보수 계획을 수립할 수 있습니다.

c. 데이터의 어떠한 독립 변수를 사용하여 어떠한 종속 변수를 예측하는지

- **독립 변수**: 온도, 습도, 압력, 진동, RPM, 각종 센서 데이터 (Sensor1 ~ Sensor6)
- **종속 변수**: 엘리베이터의 상태 (Status, 0 = 정상, 1 = 고장)

---

3. 배경지식

a. 데이터 관련 사회 문제 설명

엘리베이터 고장은 심각한 안전 문제를 초래할 수 있으며, 이는 특히 고층 건물에서 더욱 중요합니다. 기존의 정기 유지보수 방식은 예측력이 부족하며, 고장이 발생한 후에 조치를 취하는 반응적인 접근 방식입니다.

b. 머신러닝 모델 관련 설명

머신러닝 모델은 과거의 데이터 패턴을 학습하여 미래의 고장 가능성을 예측합니다. Random Forest 모델은 다수의 결정 트리를 조합하여 강력한 예측력을 제공하며, 특히 다양한 센서 데이터를 다루는 데 적합합니다.

---

4. 개발 내용

a. 데이터에 대한 구체적 설명 및 시각화

- **데이터 개수 및 속성**
    - 총 데이터 개수: 데이터셋에는 약 10,000개의 샘플이 포함되어 있습니다.
    - 주요 속성: 온도, 습도, 진동 등 환경 데이터와 센서 값
- **데이터 간 상관관계**
    - 상관행렬을 통해 변수 간 상관성을 분석한 결과, 일부 센서 데이터가 고장 예측에 더 강한 영향을 미치는 것으로 나타났습니다.

b. 예측 목표

- **독립 변수**: 센서 및 환경 데이터
- **종속 변수**: 엘리베이터 상태 (Status)

c. 머신러닝 모델 선정 이유

- **Random Forest**: 높은 해석 가능성과 강력한 성능으로 인해 선택되었습니다.
- **비교 모델**: 성능 비교를 위해 다른 알고리즘(예: SVM, XGBoost)을 추가적으로 평가할 수 있습니다.

d. 사용 성능 지표

- **Accuracy**: 모델의 전반적인 예측 정확도를 평가.
- **Confusion Matrix**: 각 클래스별 예측 성능을 시각적으로 분석.
- **Feature Importance**: 각 변수의 중요도를 평가하여 모델 해석에 기여.

---

5. 개발 결과

a. 성능 지표에 따른 평가

- **Accuracy**: 약 99.99%로 매우 높은 정확도를 달성.
- **Confusion Matrix**: 모델이 고장과 정상 상태를 거의 완벽하게 분류함.
- **Feature Importance**: 주요 변수로는 Sensor1, Sensor4, Vibrations 등이 확인됨.

b. 결과 해석

- 높은 정확도를 통해 모델이 신뢰성 있는 예측을 제공하며, 실제 환경에서도 적용 가능성이 높음을 확인.

---

6. 결론

a. 요약 및 결과

이 프로젝트는 엘리베이터 상태를 예측하는 머신러닝 모델을 개발하여 높은 정확도를 달성했습니다. 이를 통해 엘리베이터 운영의 안전성과 효율성을 크게 향상시킬 수 있습니다.

b. 개발 의의

머신러닝 모델은 단순한 반응적 유지보수 방식을 넘어, 사전 예방적 접근 방식을 가능하게 하며, 이는 비용 절감과 사고 예방에 중요한 역할을 합니다.

c. 한계

- 데이터의 제한: 데이터가 특정 환경에 편향될 가능성.
- 추가 모델 검증 필요: 다른 모델 및 실제 운영 환경에서의 추가 테스트가 필요.

1. 데이터 로드

- **개발 의도**: 데이터 분석을 시작하려면 데이터를 메모리로 로드해야 합니다.
- **구현 내용**:
    - 데이터는 pandas 라이브러리를 사용하여 Excel 파일로부터 읽어옵니다.
    - 분석할 데이터는 data라는 시트에서 파싱됩니다.

---

2. 데이터 탐색 및 정리

- **개발 의도**: 데이터의 기초 통계 및 결측치를 파악하여 데이터 전처리에 필요한 정보를 얻습니다.
- **구현 내용**:
    - describe() 메서드를 통해 데이터의 요약 통계를 확인합니다.
    - isnull().sum() 메서드를 사용하여 각 열의 결측값 개수를 확인합니다.
    - 결과를 텍스트 파일(dataset_summary.txt)로 저장하여 문서화합니다.

---

3. 데이터 시각화

- **개발 의도**: 데이터 간 상관 관계를 파악하고 분석에 참고합니다.
- **구현 내용**:
    - sns.heatmap을 사용하여 상관 관계 행렬을 히트맵 형태로 시각화합니다.
    - 이를 통해 변수 간의 강한 상관 관계를 확인하여 특징 선택이나 차원 축소에 활용할 수 있습니다.

---

4. 데이터 전처리

- **개발 의도**: 모델 학습에 적합한 형식으로 데이터를 변환하고 결측값을 처리합니다.
- **구현 내용**:
    - 불필요한 열(Time)은 분석에서 제외합니다.
    - SimpleImputer를 사용하여 결측값을 각 열의 평균값으로 대체합니다.
    - StandardScaler로 데이터를 표준화(평균 0, 표준편차 1)하여 모델 학습 성능을 높입니다.

---

5. 데이터 분할

- **개발 의도**: 모델 평가를 위해 데이터를 학습용과 테스트용으로 분리합니다.
- **구현 내용**:
    - train_test_split 함수를 사용하여 전체 데이터의 80%를 학습에, 20%를 테스트에 사용하도록 분리합니다.
    - stratify 매개변수를 사용하여 타겟 클래스의 비율을 유지하도록 분리합니다.

---

6. 모델 학습

- **개발 의도**: 데이터를 기반으로 분류 문제를 해결하는 머신러닝 모델을 훈련합니다.
- **구현 내용**:
    - RandomForestClassifier를 사용하여 랜덤 포레스트 모델을 훈련합니다.
    - 학습 데이터를 이용해 fit() 메서드로 모델을 학습합니다.

---

7. 모델 평가

- **개발 의도**: 모델의 성능을 객관적으로 평가하고 개선점을 찾습니다.
- **구현 내용**:
    - accuracy_score로 정확도를 계산하여 전체적인 성능을 확인합니다.
    - classification_report로 정밀도, 재현율, F1 스코어 등 상세한 평가 지표를 얻습니다.
    - confusion_matrix를 사용하여 예측 결과와 실제 값 간의 관계를 시각적으로 확인합니다.

---

8. 피처 중요도 분석

- **개발 의도**: 모델이 각 변수를 얼마나 중요하게 사용했는지 확인하여 해석 가능성을 높입니다.
- **구현 내용**:

- 랜덤 포레스트 모델의 feature_importances_ 속성을 사용하여 각 변수의 중요도를 계산합니다.
- 중요도를 시각화하여 주요 변수를 확인합니다.

---

9. 타겟 변수 분포 확인

- **개발 의도**: 타겟 변수의 클래스 비율을 확인하여 데이터 불균형 문제를 파악합니다.
- **구현 내용**:
  - sns.countplot을 사용하여 클래스 분포를 시각화합니다.
  - 이를 통해 데이터셋의 클래스 불균형 여부를 판단할 수 있습니다.

---

10. 결과 저장

- **개발 의도**: 분석 및 모델링 결과를 문서화하고, 시각화 자료를 저장하여 재활용 가능하도록 만듭니다.
- **구현 내용**:
  - 텍스트 파일로 주요 결과를 저장합니다 (데이터셋 요약, 모델 평가 지표 등).
  - 히트맵, 중요도 그래프, 분포 그래프 등 시각화 자료를 이미지 파일로 저장합니다.

---

11. 개발 순서와 고려사항

- **순서 설계**:
  1. 데이터 탐색 및 전처리 → 2. 데이터 분할 → 3. 모델 학습 및 평가 → 4. 결과 시각화 및 저장.
- **중점 사항**:
  - 결측값 처리, 데이터 표준화 등 전처리를 통해 모델 성능을 극대화.
  - 데이터를 학습용과 테스트용으로 분리하여 모델의 일반화 성능을 평가.

- 결과를 저장하고 시각화하여 분석의 투명성과 재현성 확보.