

# 안전 관련 머신러닝 모델 개발 보고서

2018301 강도훈

- 안전 관련 머신러닝 모델 개발의 목적

데이터의 독립 변수를 활용하여 안전 상태를 예측하고, 작업 환경에서의 위험을 사전에 감지하여 작업자의 안전을 보장하기 위함.

- 개발의 의의

학습 모델을 개발함으로써 사고 예방 및 작업 환경 개선에 기여하여 작업자의 안전을 보장하고 생산성을 향상시킴.

- 개발 계획

- 데이터에 대한 요약 정리 및 시각화

데이터의 분포, 클래스별 분포, 이미지 시각화 등을 통해 데이터의 특성을 파악함.

- 데이터 전처리 계획

이미지 데이터를 모델에 적합한 형태로 전처리하는 방안을 수립함.

- 어떠한 머신러닝 모델을 사용할 것인지

RandomForest 또는 다른 적절한 머신러닝 모델을 선택함.

- 머신러닝 모델 예측 결과가 어떠한 지

학습된 모델의 성능 평가 지표 설정 (예: 정확도, 정밀도, 재현율 등).

- 사용할 성능 지표

<https://www.kaggle.com/datasets/andrewmvd/hard-hat-detection/data>

- 성능 검증 방법 계획 등

모델의 일반화 성능을 확인하기 위한 검증 방법을 수립함.

- 개발 과정 및 후기

- 학습 모델 개발 과정 정리 (\*개발 과정 캡처 필수)

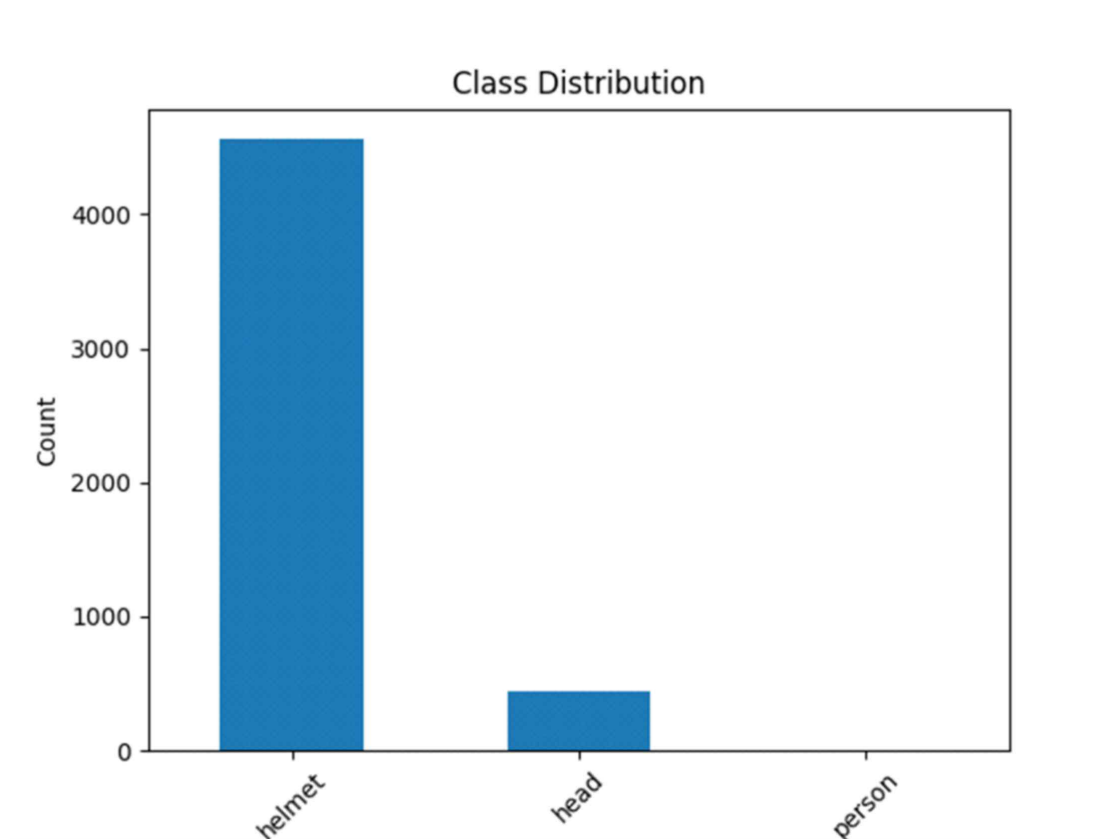
```
LIFE.py ×
1 import os
2 import pandas as pd
3 import cv2
4 import numpy as np
5 from xml.etree import ElementTree as ET
6 import matplotlib.pyplot as plt
7 from sklearn.model_selection import train_test_split
8 from sklearn.ensemble import RandomForestClassifier
9 from sklearn.metrics import accuracy_score, confusion_matrix, classification_report
10
11 # 1. 데이터 로딩
12 data_directory = 'C:/Users/home/Desktop/Sheet'
13 annotations_path = os.path.join(data_directory, 'annotations')
14 images_path = os.path.join(data_directory, 'images')
15
16 # CSV 파일 생성
17 labels_list = []
18 for filename in os.listdir(annotations_path):
19     labels_list.append({'filename': filename.replace(_old: '.xml', _new: '.png'), 'class': 'safe'})
20
21 labels = pd.concat(objs: [pd.DataFrame(labels_list)], ignore_index=True)
22
23 # 2. 데이터 전처리
24 X = [] # 이미지를 저장할 리스트
25 y = labels['class']
```

```
64
65 # 빈 DataFrame 생성
66 labels = pd.DataFrame(columns=['filename', 'class'])
67
68 # XML 파일 읽어와서 DataFrame에 추가
69 for filename in os.listdir(annotations_path):
70     if filename.endswith('.xml'):
71         tree = ET.parse(os.path.join(annotations_path, filename))
72         root = tree.getroot()
73         class_label = root.find('object/name').text
74         image_filename = filename.replace(_old: '.xml', _new: '.png')
75         labels = pd.concat(objs: [labels, pd.DataFrame({'filename': [image_filename], 'class': [class_label]})], ignore_index=True)
76
77 # 5. 데이터 요약 정리 및 시각화 그래프
78 print("Data Summary:")
79 print(labels.head())
80
81 # 시각화 그래프 (클래스별 분포)
82 class_distribution = labels['class'].value_counts()
83 class_distribution.plot(kind='bar', rot=45)
84 plt.xlabel('Class')
85 plt.ylabel('Count')
86 plt.title('Class Distribution')
87 plt.show()
```

-학습 모델의 성능 평가

데이터셋에서 정밀도와 재현율, 정확도가 높은 값을 나타내고 있음.

-데이터 시각화



-개발 후 느낀 점

데이터셋을 사용하여 머신러닝을 제작하는 것은 쉽지 않다는 것을 느낌.

- 개발한 학습 모델의 효과

현장에서 안전모 착용 여부를 감지함으로써 안전 감시를 할 수 있음.