수질 데이터 셋을 활용한 사람이 마실 수 있는 물 여부 예측 머신러닝 모델 개발 보고서

1. 개발의 목적:

- 학습 모델 활용 대상: 수질 데이터를 기반으로 사람이 마실 수 있는 물 여부를 예측하는 것이 목적이다.

- 데이터의 독립 변수 및 종속 변수: 주어진 데이터셋에서는 pH, Hardness, Solids, Chloramines, Sulfate, Conductivity, Organic_carbon, Trihalomethanes, Turbidity 등 9개의 독립 변수를 사용하여 물의 마시기 적합 여부(Potability)를 예측한다.

- 개발의 의의: 이 모델을 개발함으로써, 주어진 수질 데이터를 활용하여 물의 품질을 예측하고 마시는 물의 안전성을 판단하는 가치를 창출한다.

1) 학습 모델의 활용:

- 수도관 및 우물 관리: 모델은 주로 수도관이나 우물에서 나오는 물의 품질을 예측하는 데 사용된다.

- 수질 감시 기관: 수질 감시 기관은 이 모델을 활용하여 물의 안전성을 빠르게 평가할 수 있다.

- 일반 사용자: 일반 사용자들은 이 모델을 통해 자신이 사용하는 물이 안전한지 여부를 확인할 수 있다.

2) 개발의 의의:

- 데이터 분석을 통한 물의 품질 예측: 주어진 다양한 수질 지표를 분석하여 물의 품질을 예측하는 모델은 물의 소비자들에게 더 안전한 마실 물을 제공한다.

- 사용자의 건강 보호: 정확한 물의 품질 예측은 사용자의 건강을 보호하고 물 관련 질병을 예방하는 데 기여한다.

- 데이터 기반 의사 결정: 이 모델은 데이터 기반의 의사 결정을 가능케 하여 수질에 대한 정확한 정보를 제공한다.

2. 모델의 네이밍의 의미:

- 모델 명칭: AquaSafePredictor

- 의미 설명: AquaSafePredictor는 "물의 안전성을 예측하는 모델"을 의미한다. 이 모델은 주

어진 수질 데이터를 기반으로 물의 품질을 분석하고, 사람이 해당 물을 마실 수 있는지 여부를 예측하는 데 사용된다.

3. 개발 계획

1) 데이터에 대한 요약 정리 및 시각화

- 데이터 셋의 행렬 크기: (행, 열) = (데이터 수, 변수 수)

- 데이터 요약: 데이터의 주요 통계량, 중요 변수 간의 상관 관계 등

- 데이터 시각화: 주요 변수들 간의 관계를 나타내는 산점도 및 상자 그림 작성

2) 데이터 전처리 계획

- 결측값 처리: 결측값이 있는 행 제거

- 독립 변수 및 종속 변수 설정: 특정 변수를 독립 변수로 설정하고, "Potability" 변수를 종속 변수로 설정

3) 머신러닝 모델 선택

- 모델 선택: RandomForestClassifier를 사용

- 이론: RandomForest는 여러 결정 트리를 사용하여 데이터를 학습하고, 안정적이며 강력한 예측 성능을 제공하는 알고리즘

4) 머신러닝 모델 예측 결과

- 모델 예측 결과: AquaSafePredictor 모델은 주어진 수질 데이터를 기반으로 해당 물이 마실 수 있는지 여부를 예측

- 예측 결과 해석: 0 또는 1로 나오는 예측 결과를 해석하여 해당 수질이 안전한지를 나타냄

5) 사용할 성능 지표

- 정확도 (Accuracy): 모델이 정확하게 예측한 비율

- 평균 제곱 오차 (MSE): 예측 값과 실제 값 간의 평균 제곱 오차

- 평균 절대 오차 (MAE): 예측 값과 실제 값 간의 평균 절대 오차

- 오차 행렬 (Confusion Matrix): 모델의 분류 성능을 나타내는 행렬

- 정밀도 (Precision): 긍정으로 예측한 것 중에서 실제로 긍정인 비율

- 재현율 (Recall): 실제로 긍정인 샘플 중에서 모델이 긍정으로 예측한 비율

- F1 스코어 (F1 Score): 정밀도와 재현율의 조화 평균

6) 성능 검증 방법 계획

- 데이터 분할: 학습 데이터와 테스트 데이터로 데이터를 분할

- 데이터 전처리: 표준화(Standardization)를 통해 데이터를 전처리

- 모델 학습: RandomForestClassifier를 사용하여 모델 학습

- 성능 평가: 정확도, MSE, MAE, Confusion Matrix, Precision, Recall, F1 Score를 계산하여 모델의 성능을 평가

4. 개발 과정

1) 계획 후 실제 학습 모델 개발 과정 정리

① 데이터 로딩 및 요약:

- 데이터를 불러와서 주요 통계량 및 데이터의 구조를 확인

- 결측값이 있을 경우 해당 행 제거

```
# 데이터 읽어오기
data = pd.read_csv(filename, names=column_names)
```

```
# 결측값 제거 (만약 결측값이 있다면)
data = data.dropna()
```

② 데이터 시각화:

- 각 독립 변수와 종속 변수 간의 관계를 시각화하여 데이터를 탐색

```
# 시각화 그래프 저장
plt.figure(figsize=(10, 6))
for i in range(X.shape[1]):
    plt.scatter(X[:, i], y, alpha=0.5, label=column_names[i])
plt.legend()
plt.title('Feature vs Potability')
plt.xlabel('Feature Values')
plt.ylabel('Potability')
plt.savefig('feature_vs_potability.png')
```

③ 데이터 분할 및 전처리:

- 학습 데이터와 테스트 데이터로 분할

- 표준화(Standardization)를 통해 데이터 전처리

```
# 데이터 분할 (학습 데이터와 테스트 데이터)
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

# 데이터 전처리: 표준화(Standardization)
scaler = StandardScaler()
X_train = scaler.fit_transform(X_train)
X_test = scaler.transform(X_test)
```

④ 모델 학습:

- RandomForestClassifier 모델을 사용하여 학습

```
# 머신러닝 모델 생성 및 학습
model = RandomForestClassifier(random_state=42)
model.fit(X_train, y_train)
```

⑤ 성능 평가:

- 학습된 모델을 사용하여 테스트 데이터에 대한 예측 수행 및 성능 평가

```
# 학습된 모델을 사용하여 테스트 데이터에 대한 예측 수행
y_pred = model.predict(X_test)
```

```
# 성능 평가
accuracy = accuracy_score(y_test, y_pred)
mse = mean_squared_error(y_test, y_pred)
mae = mean_absolute_error(y_test, y_pred)
conf_matrix = confusion_matrix(y_test, y_pred)
precision, recall, f1, _ = precision_recall_fscore_support(y_test, y_pred, average='binary')
```

```
# 결과 출력
print("\n성능 평가:")
print("정확도 (Accuracy):", accuracy)
print("평균 제곱 오차 (MSE):", mse)
print("평균 절대 오차 (MAE):", mae)
print("오차 행렬 (Confusion Matrix):\n", conf_matrix)
print("정밀도 (Precision):", precision)
print("재현율 (Recall):", recall)
print("F1 스코어 (F1 Score):", f1)
```

2) 각 함수는 어떻게 동작하는 지 구체적으로 설명

① 데이터 로딩 및 요약

- 데이터 로딩:

- `pd.read_csv(filename, names=column_names)`: 주어진 파일에서 데이터를 읽어와 DataFrame으로 저장

- 결측값 제거:

- `data.dropna()`: DataFrame에서 결측값이 있는 행을 제거

② 데이터 시각화

- 산점도 그래프 작성:

- 주어진 독립 변수와 종속 변수를 산점도로 표현하여 변수 간의 관계를 시각화

- `plt.scatter(X[:, i], y, alpha=0.5, label=column_names[i])`

③ 데이터 분할 및 전처리

- 데이터 분할:

- `train_test_split(X, y, test_size=0.2, random_state=42)`: 전체 데이터를 학습 데이터와 테스트 데이터로 분할

- 데이터 표준화:

- `scaler.fit_transform(X_train)`, `scaler.transform(X_test)`: 학습 데이터로 표준화를 학습하고, 테스트 데이터를 표준화

④ 모델 학습

- RandomForestClassifier 모델 학습:

- `RandomForestClassifier(random_state=42)`: 랜덤 포레스트 분류기 모델 생성

- `model.fit(X_train, y_train)`: 학습 데이터를 사용하여 모델 학습

⑤ 성능 평가

- 테스트 데이터 예측:

- `model.predict(X_test)`: 학습된 모델을 사용하여 테스트 데이터에 대한 예측 수행

- 성능 평가 지표 계산:

- `accuracy_score`, `mean_squared_error`, `mean_absolute_error`, `confusion_matrix`, `precision_recall_fscore_support` 함수를 사용하여 정확도 및 기타 성능 지표 계산

3) 에러 발생 지점 및 해결 과정

- 에러 발생 지점:

- 예외 처리 및 오류 핸들링이 필요한 부분에서 에러 발생

- 해결 과정:

- 예외 처리를 통해 오류에 대한 메시지를 출력하고, 사용자에게 안내

4) 학습 모델의 성능 평가

① 성능 평가 결과

- 정확도 (Accuracy): XX

- 평균 제곱 오차 (MSE): XX

- 평균 절대 오차 (MAE): XX

- 오차 행렬 (Confusion Matrix): XX

- 정밀도 (Precision): XX

- 재현율 (Recall): XX

- F1 스코어 (F1 Score): XX

5) 결과 시각화

- Feature vs Potability 그래프: `feature_vs_potability.png` 파일로 저장하여 시각화

5. 개발 후기

프로젝트를 시작하기 전에는 수질 데이터를 활용하여 마실 수 있는 물의 여부를 예측하는 과제에 대한 이해가 부족했다. 그러나 프로젝트를 진행하면서 데이터 전처리, 시각화, 머신러닝 모델 학습 등의 다양한 단계를 경험하며 데이터 과학 프로젝트의 전반적인 흐름을 이해할 수 있었다.

6. 개발 후 느낀 점

1) 데이터의 중요성

프로젝트를 진행하면서 데이터의 품질이 모델의 성능에 큰 영향을 미친다는 것을 몸소 깨달았다. 초기에는 데이터의 특성을 이해하는 데 어려움을 겪었지만, 데이터를 시각화하고 분석함으로써 머신러닝 모델이 데이터의 특징을 어떻게 학습하는지 이해할 수 있었다.

2) 모델 성능 평가

머신러닝 모델의 성능을 평가하는 지표를 선택하고 해석하는 것이 중요하다는 것을 깨달았다. 정확도 외에도 정밀도, 재현율, F1 스코어 등 다양한 지표를 활용하여 모델의 강점과 약점을 파악하고 개선 방향을 찾을 수 있었다.

7. 개발한 학습 모델의 효과

프로젝트에서 개발한 AquaSafePredictor 모델은 주어진 특성을 기반으로 마실 수 있는 물의 여부를 예측하는 데 일정 수준의 성능을 보였다. 그러나 현재 모델은 개선의 여지가 있으며, 추가적인 데이터 수집, 특성 엔지니어링, 모델 튜닝 등을 통해 성능을 더욱 향상시킬 수 있을 것으로

기대된다.

8. 향후 계획

향후에는 다음과 같은 방향으로 프로젝트를 발전시킬 계획이다.

1) 데이터 추가 수집: 현재 사용된 데이터 이외에도 다양한 출처에서 데이터를 추가로 수집하여 모델의 학습 데이터를 확장할 예정이다.

2) 특성 엔지니어링: 현재 사용된 특성들을 조합하여 새로운 특성을 생성하거나, 특성의 스케일을 조절하여 모델의 학습에 도움이 될 수 있는 새로운 특성을 도출할 예정이다.

3) 다양한 알고리즘 적용: 현재는 Random Forest Classifier를 사용했으나, 다른 머신러닝 알고리즘을 시도하여 어떤 알고리즘이 가장 효과적인지 비교할 예정이다.

4) 모델 튜닝: 현재의 모델 파라미터를 조정하고, 그리드 서치와 같은 기술을 사용하여 최적의 하이퍼파라미터를 찾을 예정이다.

이번 프로젝트를 통해 데이터 과학 및 머신러닝 프로젝트의 일련의 과정을 경험하고, 모델의 개선 방향에 대한 인사이트를 얻을 수 있었다. 앞으로의 프로젝트에서는 이러한 경험을 활용하여 보다 높은 품질의 모델을 개발하고 실제 활용 가능성을 탐색할 계획이다.