

ML 모델 개발

Today: ML 모델 개발

1. ML 기본모델(회귀, 분류)
2. 교차 검증, 성능 평가지표

1.0 ML 모델

1. ML 기본모델: 상황에 알맞은 모델 선택 필요

- a. Regression (회귀)
 - i. Linear Regression
 - ii. Polynomial Regression
 - iii. Random Forest Regressor
- b. Classification (분류)
 - i. Logistic Regression
 - ii. Decision Tree
 - iii. Random Forest Classifier
- c. Neural Networks
 - i. Simple Multi-layer Perceptron
 - ii. Deep Learning (향후 고급 과정)



1.1 회귀 모델

- Linear Regression (선형 회귀)

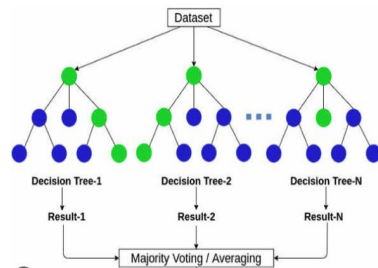
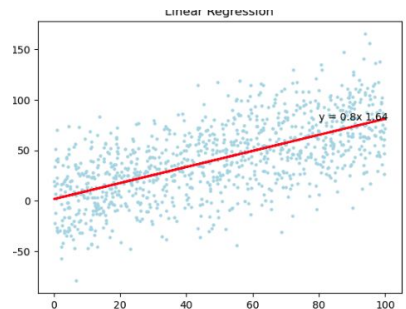
- 목표: 입력과 출력 간 선형 관계 찾기
- 수식: $y = ax_1 + bx_2 + \dots + c$
- 장점: 빠름, 해석 쉬움, 과적합 위험 낮음
- 단점: 비선형 관계 포착 어려움

- Random Forest Regressor

- 원리: 여러 결정트리 조합하여 예측
- 장점: 비선형 관계 포착, 이상치에 강함
- 단점: 해석 어려움, 메모리 사용량 많음

- 공항 데이터 적용

- Team 1: 수하물 도착 시간 예측 (연속값)
- Team 2: 체크인 대기 인원 예측 (연속값)



1.2. 분류 모델

- Logistic Regression

- 이진 분류 (0 또는 1)
- 확률로 결과 해석 가능
- 선형 결정 경계

- Decision Tree

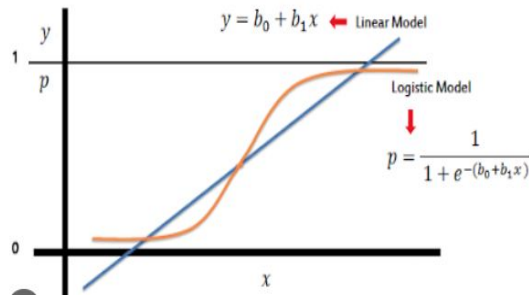
- 규칙 기반 분류
- 매우 해석하기 쉬움
- 과적합 위험 있음

- Random Forest Classifier

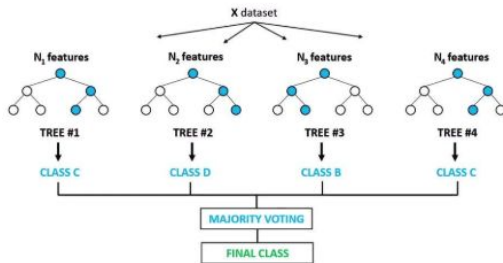
- 여러 트리의 다수결
- 높은 성능, 안정적
- Feature importance 제공

- 공항 데이터 적용 예시

- 지연 여부 분류 (정시/지연)
- 탑승객 타입 분류 (일반/VIP)



Random Forest Classifier



1.3. 구현

- scikit-learn (추천)

```
from sklearn.linear_model import LinearRegression
from sklearn.ensemble import RandomForestRegressor
```

```
model = LinearRegression()
model.fit(X_train, y_train)
y_pred = model.predict(X_test)
```

- from scratch using numpy

- 학습 목적으로 라이브러리 쓰지 않고 ML 알고리즘 직접 구현 추천
- 수학적 이해 향상, **leaky abstraction** 하지 않기
- <https://karpathy.medium.com/yes-you-should-understand-backprop-e2f06eab496b> 참고

- PyTorch, TensorFlow

- 딥러닝이 필요한 경우
- 복잡한 모델

2.1 교차 성능

- K-Fold Cross Validation

- 데이터를 K개 폴드로 분할
- K-1개로 학습, 1개로 검증
- K번 반복하여 평균 성능 계산

- 장점

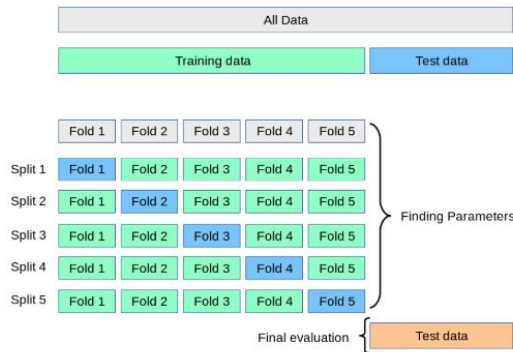
- 과적합 방지
- 안정적인 성능 평가
- 모든 데이터 활용

- 구현

```
from sklearn.model_selection import cross_val_score
```

```
scores = cross_val_score(model, X, y, cv=5, scoring='neg_mean_absolute_error')  
print(f"평균 MAE: {-scores.mean():.3f}")
```

- 일반적으로 K=5 또는 K=10 사용



2.2 성능 평가 지표

- 회귀 지표
 - MAE (Mean Absolute Error): 절댓값 평균 오차
→ 직관적, 이상치에 덜 민감
 - MSE (Mean Squared Error): 제곱 평균 오차
→ 큰 오차에 더 민감
 - R^2 (R-squared): 결정계수 (0~1)
→ 모델의 설명력 나타냄
- 분류 지표
 - Accuracy: 정확도 (전체 중 맞춘 비율)
 - Precision: 정밀도 (예측한 것 중 맞춘 비율)
 - Recall: 재현율 (실제 것 중 찾은 비율)
 - F1-Score: Precision과 Recall의 조화평균
- 공항 프로젝트 선택
 - Team 1 (수하물): MAE, R^2
 - Team 2 (체크인): MAE, R^2

3. 코드 실습