기초 이론부터 실무 실습까지

머신 러닝 익히기

programmers

Part 04. 선형 회귀와 선형 분류

정 정 민

programmers

Chapter 10. 선형 회귀 실습

1. 의료비 개인 데이터셋
2. EDA, 탐색적 데이터 분석
3. 데이터 전처리
4. 모델 구축 및 결과 확인

의료비 개인 데이터셋

programmers

의료비 개인 데이터셋 (Medical Cost Personal Datasets)

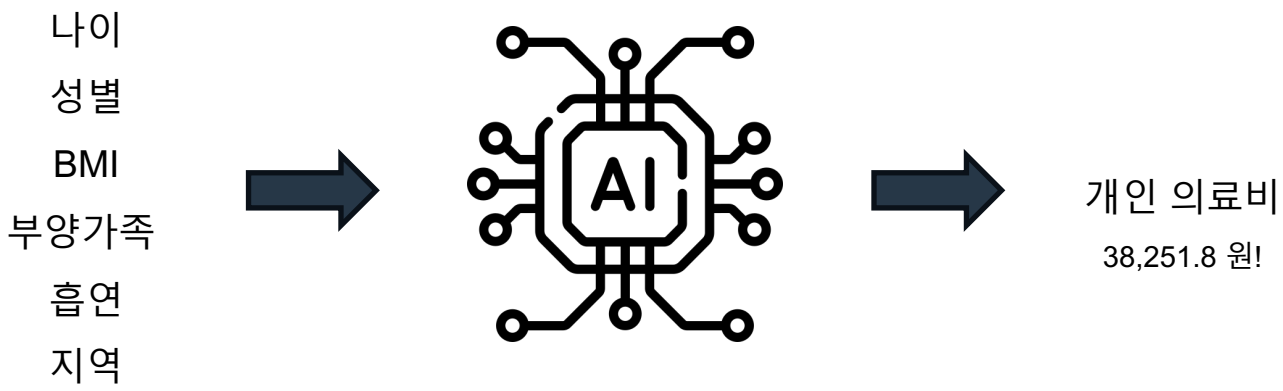
- 이번 실습에서 사용할 데이터로 Kaggle의 공개 데이터 (링크)
  - 다운로드 받아주세요!

- 건강 및 인구통계학적 정보와 개인의 의료비 정보를 모아 둔 데이터
- 아래의 변수를 포함
  - 나이
  - 성별
  - 체지방 지수 (BMI)
  - 부양가족 수
  - 흡연 유무
  - 사는 지역 : 미국 내 북동 / 남동 / 남서 / 북서
  - 개인 의료비 (charges)

문제 정의

- 풀어야 하는 문제
  - 주어진 건강 및 인구통계학적 정보를 바탕으로 개인의 연간 의료 보험료를 예측
    독립 변수 종속 변수

- 머신 러닝 모델의 입, 출력 정의
  - 입력 : 앞선 독립 변수들
  - 출력 : 개인 의료비 (종속 변수)



EDA, 탐색적 데이터 분석

programmers

강의 실습 코드 : 링크

EDA(Exploratory Data Analysis), 탐색적 데이터 분석

- 데이터 분석의 초기 단계에서 진행하는 과정
- 데이터를 여러 각도에서 살피며 데이터의 특징, 구조, 패턴, 이상치, 변수 간의 관계 등을 이해

- 이를 통해
  - 데이터에 대한 직관을 얻거나
  - 후속 분석에 필요한 모델링 전략 수립의 통찰을 얻을 수 있음

- 아래의 작업들이 수행됨
  - 기초 통계 분석 : 평균, 중앙값, 표준편차, 최솟/최댓값 등
  - 시각화 : 데이터 패턴, 이상치, 경향성 식별
  - 변수간 관계 파악 : 서로 다른 변수 간 상관관계 분석
  - 이상치 탐지 : 다른 데이터의 특성에서 벗어난 비정상적 데이터 식별
  - 결측치 분석 : 누락 데이터 확인

기본 정보

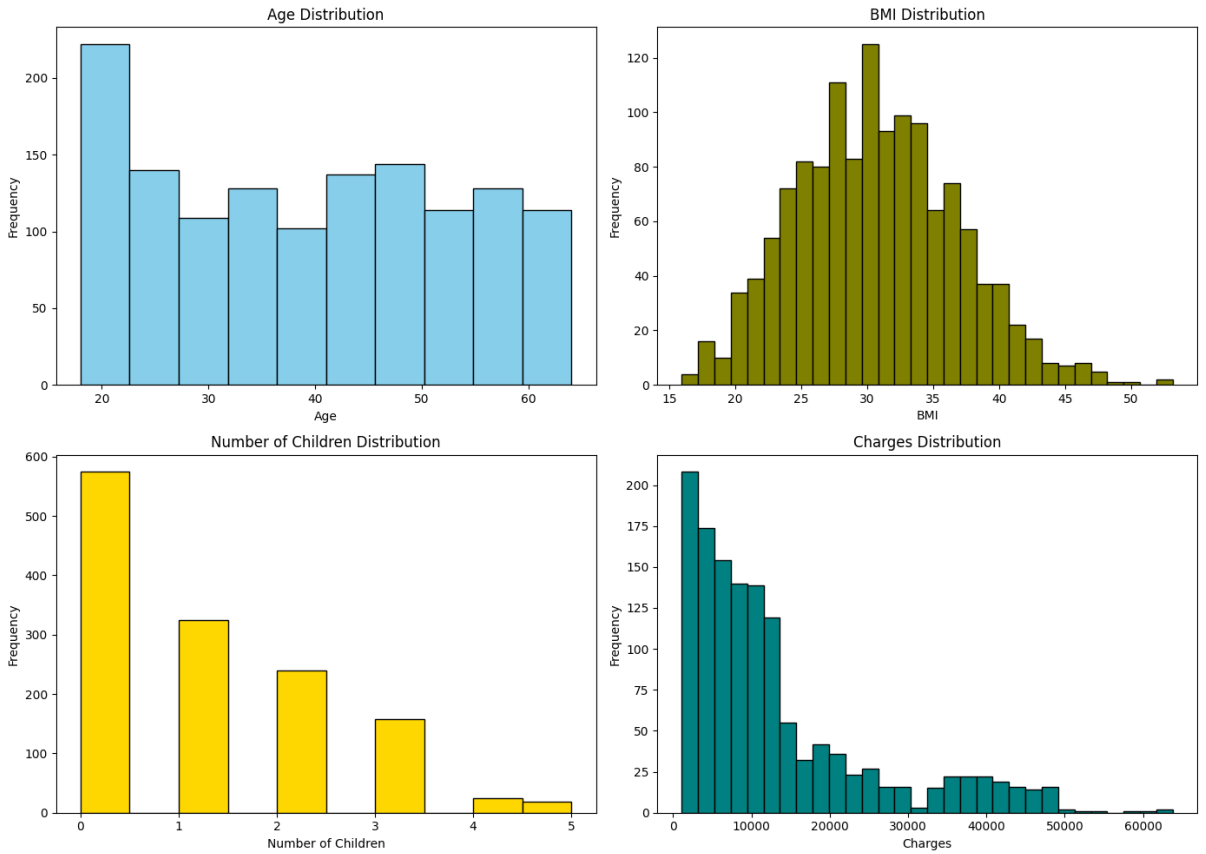
- 전체 데이터셋 크기
  - 총 1338개의 개별 데이터
  - 총 7개 특성
- 데이터 타입
  - 정수형 (int) : ‘age’, ‘children’
  - 실수형 (float) : ‘bmi’, ‘charges’
  - 문자열 (object) : ‘sex’, ‘smoker’, ‘region’
- 누락 : 없음

```python
# 기본 정보
print('#'*20, '기본 정보', '#'*20)
insurance_data.info() # info() 안에서 자동으로 print를 진행

# 기초 통계량
summary_statistics = insurance_data.describe(include='all')
print('#'*20, '기초 통계량', '#'*20)
print(summary_statistics)
```

```
RangeIndex: 1338 entries, 0 to 1337
Data columns (total 7 columns):
 #   Column    Non-Null Count  Dtype
---  ------    --------------  -----
 0   age       1338 non-null   int64
 1   sex       1338 non-null   object
 2   bmi       1338 non-null   float64
 3   children  1338 non-null   int64
 4   smoker    1338 non-null   object
 5   region    1338 non-null   object
 6   charges   1338 non-null   float64
dtypes: float64(2), int64(2), object(3)
memory usage: 73.3+ KB
```

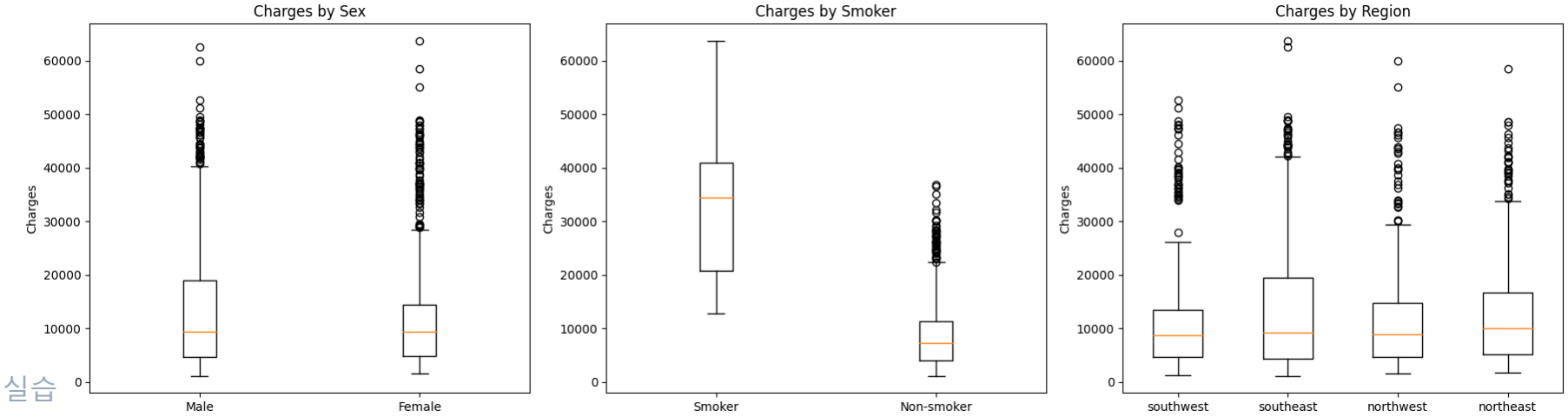
시각화 – 수치형 데이터

- 시각화를 통해 전체 데이터의 분포를 볼 수 있음
  - 나이 분포 : 상대적으로 균일하게 분포로 큰 편중이 없음
  - BMI 분포 : 정규분포와 유사한 형태를 보임
  - 부양 가족 수 분포 : 대부분 0~2명의 자녀를 갖고 있음
  - 의료비 분포 : 오른쪽으로 꼬리가 긴 분포
- 한글 입력 시 프린트가 안되는 문제가 있음
  - 한글 폰트 설치 후 사용하면 가능
- 각 특성 간 상관관계 분석도 수행할 수 있음
  - Pandas 객체에서 바로 상관관계 분석 가능



시각화 – 카테고리 데이터

- 카테고리 데이터는 카테고리가 정해져 있어서 종속 변수와의 관계를 살펴볼 수 있음
  - 성별
    - 성별에 따른 의료비용 분포에 약간의 차이가 있음
    - 남성의 경우 여성보다 의료비용을 좀 더 많이 냄
  - 흡연 여부
    - 흡연 유무는 매우 두드러지는 차이를 보임
    - 종속 변수에 영향을 미치는 큰 요인으로 보임
  - 지역별
    - 차이는 보이지만 흡연 유무만큼은 아님



데이터 전처리

강의 실습 코드 : 링크

카테고리형 변수 인코딩

- 카테고리형(범주형) 변수는 선형 모델에 입력으로 사용하기 위해 수치형으로 변경해야 함
  - 이번 데이터에서는 성별, 흡연 유무, 지역이 이에 해당

- 일반적으로 원-핫 인코딩(one-hot encoding)방식을 사용
- 예를 들어, 성별이라면
  - 성별_남성과 성별_여성 이라는 별도의 열을 만들고
  - 각각을 0과 1의 값으로 표현

```
insurance_encoded = pd.get_dummies(insurance_data,
                                   drop_first=True)
```

- Pandas의 get_dummies 함수를 사용
  - drop_first 옵션은 첫 카테고리를 제거하는 역할
  - 새롭게 생겨난 변수들의 강한 상관관계가 나타나서(다중공선성) 보통은 제거하는 것이 좋음

학습 및 평가 데이터 분리

- 사용할 데이터를 확정했다면 학습 및 평가 데이터로 분리
- 원래는 학습 / 검증 / 평가 과정으로 나누어야 하지만
  실제 서비스 모델을 개발하는 과정이 아니므로 학습과 평가 데이터로만 분리

- 먼저, 독립 변수와 종속 변수를 분리
  - Pandas dataframe에서 특정 열만 뽑아서 정의

- 이후, 학습과 평가 데이터로 분리하기 위해
  - sklearn 의 내장 함수인 train_test_split이라는 함수를 사용
  - test_size 변수는 학습과 평가 데이터 사이의 비율을 의미

- 결과적으로 학습 데이터 1070개, 평가 데이터 268개 구성

```
from sklearn.model_selection import train_test_split

X = insurance_encoded.drop('charges', axis=1)
y = insurance_encoded['charges']
X_train, X_test, y_train, y_test = train_test_split(X, y,
                                                    test_size=0.2,
                                                    random_state=42)
```

특성 스케일링

- 필수 단계는 아니지만 매우 권장되는 과정
- 서로 다른 수치형 데이터 특성 사이의 값 범위를 비슷하게 맞춰주는 과정

- 효과
  - 특히 경사 하강법을 사용하는 과정에서 수렴 속도를 높일 수 있음
  - 규제 모델을 사용한다면 일부 특성에 강하게 규제가 걸리는 과정을 회피할 수 있음

- 방법
  - StandardScaler
    - 평균 0, 표준편차 1로 조정
    - 데이터의 분포가 정규분포일 경우 사용하면 be
    - 일반적으로 많이 사용
  - MinMaxScaler
    - 최댓값 1, 최솟값 0이 되도록 조정
    - 이상치가 큰 영향을 미치는 경우 사용

```
continuous_columns = list(set(insurance_encoded.columns) -
                          set(encoded_columns) -
                          set(y_column)) # ['bmi', 'age', 'children']

scaler = StandardScaler()

# 수치형 데이터만 스케일링 진행
X_train_continuous = scaler.fit_transform(X_train[continuous_columns])
X_test_continuous = scaler.fit_transform(X_test[continuous_columns])
```

모델 구축 및 결과 확인

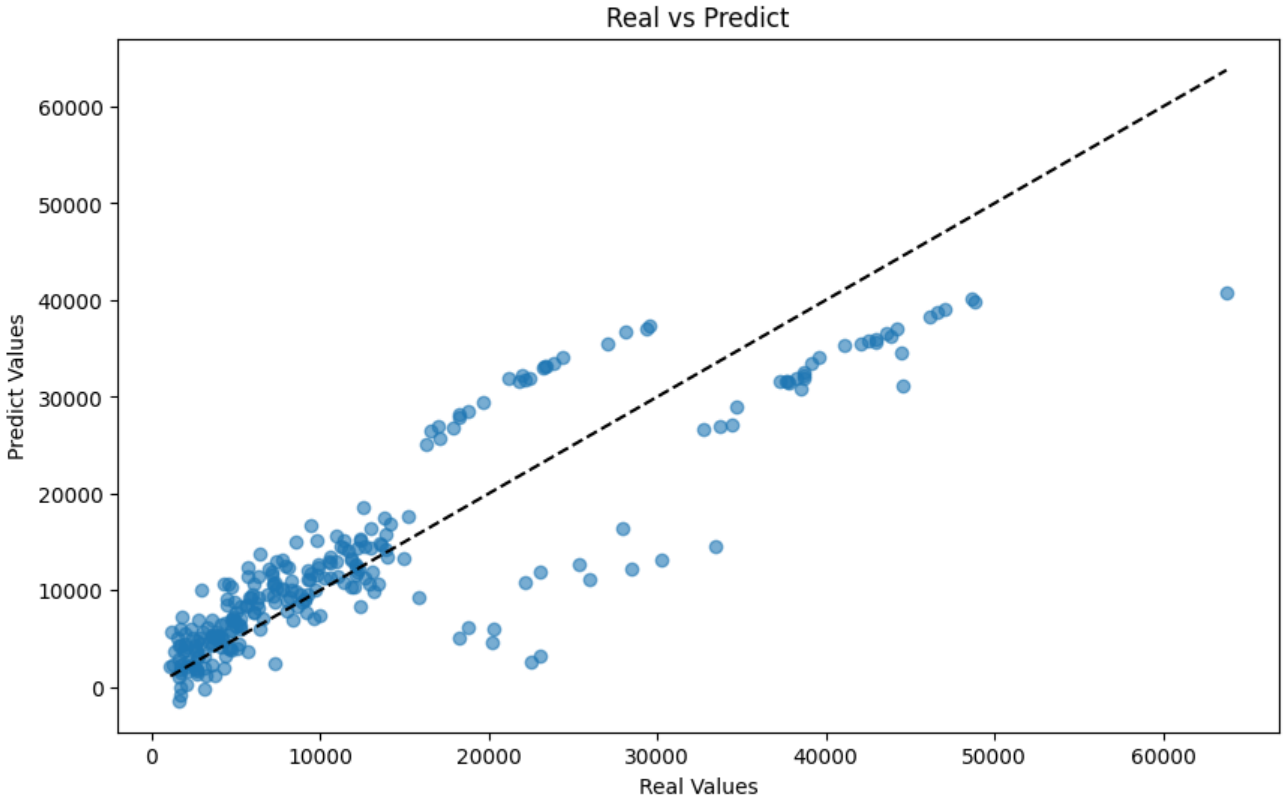
강의 실습 코드 : 링크

평가 진행

- MSE 값을 이용해 평균적인 예측 실패 정도를 판단할 수 있음
- 하지만 MSE 값으로 잘 한건지 / 그렇지 않은지를 판단하기가 어려움
- R^2 방법도 있지만 이는 추후에 설명

- 산점도 시각화 방법을 사용
  - $y = x$ 그래프와 가까울수록 좋은 예측

```python
from sklearn.metrics import mean_squared_error

# 예측 수행
y_train_pred = linear_reg.predict(X_train_final)
y_test_pred = linear_reg.predict(X_test_final)

# 평가 지표 계산: MSE
mse_train = mean_squared_error(y_train, y_train_pred)
mse_test = mean_squared_error(y_test, y_test_pred)
```



선형 회귀 모델 학습 진행

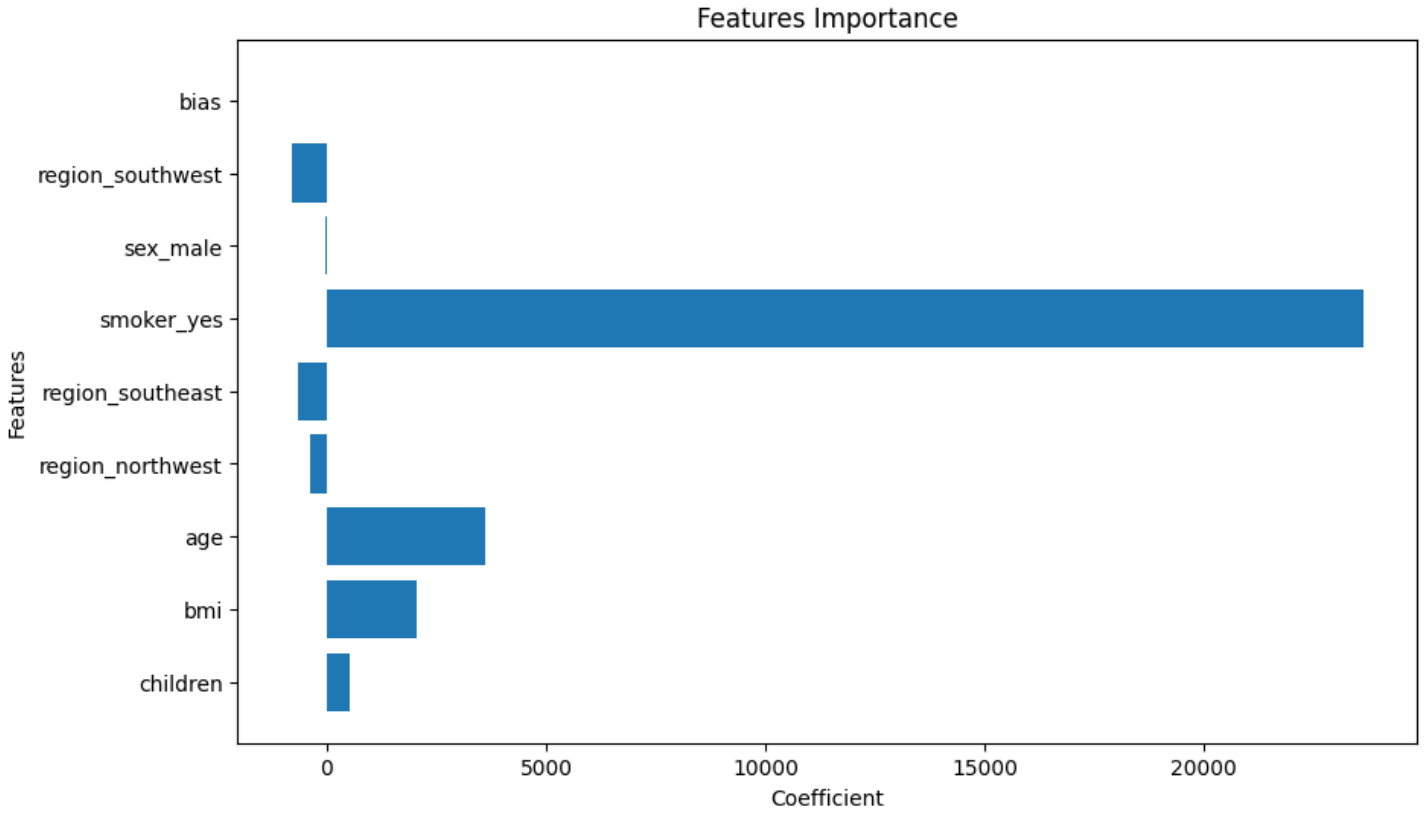
- $w0$ 값을 위해 bias를 추가해줌
  - 이전에도 결과를 봤지만 내장 함수를 이용하면
  - 자동으로 추가해서 결과를 보여줌

- LinearRegression() 객체를 생성 후 학습 진행

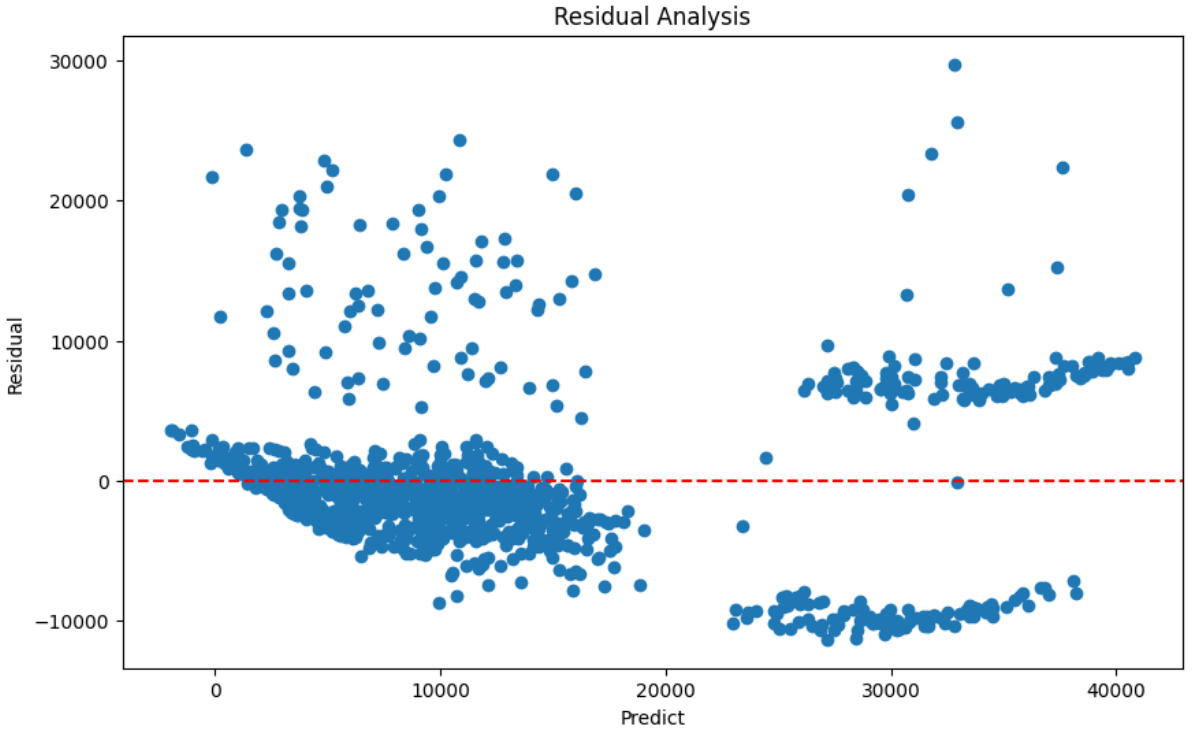
```
from sklearn.linear_model import LinearRegression

X_train_final['bias'] = 1

linear_reg = LinearRegression()
linear_reg.fit(X_train_final, y_train)

coefficients = linear_reg.coef_
intercept = linear_reg.intercept_
```

결과 해석 – 변수의 중요도

- 회귀 모델이 사용한 변수 중 중요한 변수를 추출
- 선형 모델의 장점으로 파라미터를 시각화해 이것의 중요도를 알 수 있음



결과 해석 – 잔차 분포



- 정답과의 차이인 잔차(residual)의 분포를 확인
  - 무작위 분포 : 좋은 분포
  - 특정 패턴이 존재 : 데이터를 완전히 파악하지 못함
- 결과 확인
  - 예측값이 커질수록 분포가 넓어짐
  - 큰 예측값에서 2개의 그룹이 존재
- 해석 및 추후 필요 행동
  - 의료비가 큰 경우 해석력이 떨어지므로 비선형적 특성이 있을 수 있음
    - 비선형 모델을 선택하거나,
    - 로그 혹은 제곱근 변환 등의 비선형 근사법 적용
  - EDA에서 살펴본 이상치의 영향이 있을 수 있으므로 이상치 제거

[숙제] 더 해보기!

강사가 먼저 해본 결과를 공유합니다 : 링크
꼭 스스로 먼저 해보고 강사의 결과와 비교해보세요!

- 학습시킨 모델을 SGD 방식으로 학습해보세요
  - SGDRegressor 클래스 활용 (링크)
  - 초기값
    - 학습률 : 0.001
    - 학습량 : 1000
    - 규제 : None

- 그 결과를 분석!!
  - SVD-OLS 방식의 풀이와 MSE 어떤 차이를 보이는지 확인
  - 수업에서 다룬 분석 과정을 반복
  - 변경 상수(하이퍼파라미터) 조정으로 여러 SGD를 학습
  - 학습 시간 비교
  - 등등

E.O.D