



전자공학과 2020142001 곽종근



7주차 모델 검증 및 평가, 가우시안 실습과제4, chap.2

제출일: 2024.05.02.

실습 과제 4.

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt

# 데이터 불러오기
fold_dir = "C:\\Users\\pc\\Desktop\\3학년\\1학기\\머리\\lin_regression_data_01.csv"
temp_data = pd.read_csv(fold_dir, header=None)
temp_data = temp_data.to_numpy() # 데이터를 numpy로 변환

# 데이터 분리
x_data = temp_data[:, 0] # 무게 데이터를 Weizer장
y_data = temp_data[:, 1] # 길이 데이터를 Len에 저장

# 가우시안 기저 함수 정의
def gaussian_basis_function(X, K, k):
    x_min = X.min() # 데이터의 최소값
    x_max = X.max() # 데이터의 최대값
    mu = x_min + ((x_max - x_min) / (K - 1)) * k # 각 가우시안 함수의 평균 계산

    v = (x_max - x_min) / (K - 1) # 모든 가우스 함수의 분산

    simple = (X - mu) / v
    G = np.exp((-1/2) * (simple ** 2))

    return G

# 가중치 계산 함수
def calculate_weights(X, Y, K):
    # k의 배열 생성
    k_values = np.arange(K).reshape(-1, 1)
    # k에 따른 가우시안 기저 함수 계산
    X_b = np.column_stack([gaussian_basis_function(X, K, k) for k in k_values])
    # bias 추가
    X_b = np.hstack([X_b, np.ones((len(X), 1))])
    # 가중치 계산 (K+1개의 가중치)
    weights = np.linalg.inv(X_b.T @ X_b) @ X_b.T @ Y
    return weights

# MSE 계산 함수
def mse(X, Y, K):
    # k의 배열 생성
    k_values = np.arange(K).reshape(-1, 1)
    # k에 따른 가우시안 기저 함수 계산
    X_b = np.column_stack([gaussian_basis_function(X, K, k) for k in k_values])
    # bias 추가
    X_b = np.hstack([X_b, np.ones((len(X), 1))])
    # 가중치 계산
    weights = calculate_weights(X, Y, K)
    # MSE 계산
    mse_value = np.mean(((X_b @ weights) - Y) ** 2)
    return mse_value
```

```
# K_values에 따른 MSE 계산
K_values = np.arange(3, 11) # K 값 범위 설정
mse_values = [mse(x_data, y_data, K) for K in K_values]

# MSE 그래프 그리기
plt.figure(figsize=(10, 6))
plt.plot(K_values, mse_values, marker='o')
plt.xlabel('Number of Basis Functions (K)')
plt.ylabel('Mean Squared Error (MSE)')
plt.title('MSE vs. Number of Basis Functions')
plt.grid(True)
plt.show()

# 주어진 코드에서의 그래프 그리기
# mse 그래프와 같이 그려질 수 있도록 위의 그래프 그리기 코드 아래에 추가합니다..

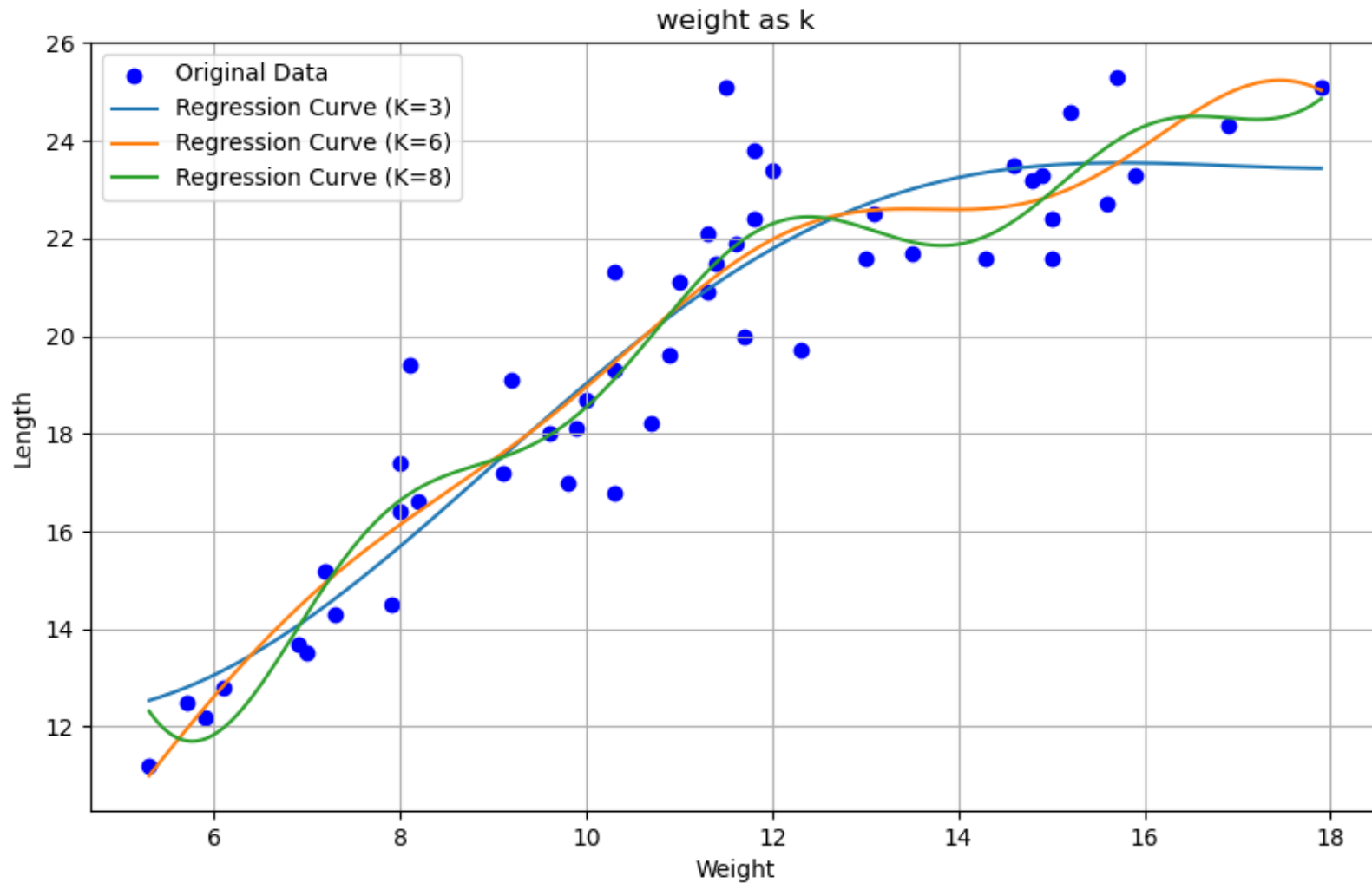
# 가중치 계산
K_values = [3, 6, 8]
weights_list = [calculate_weights(x_data, y_data, K) for K in K_values]

# 그래프 그리기
plt.figure(figsize=(10, 6))
plt.scatter(x_data, y_data, color='blue', label='Original Data') # 원본 데이터
for K, weights in zip(K_values, weights_list):
    x_range = np.linspace(x_data.min(), x_data.max(), 1000)
    # bias 추가하여 예측값 계산
    y_pred = np.column_stack([gaussian_basis_function(x_range, K, k) for k in range(K)]) @ weights[:-1] + weights[-1]
    plt.plot(x_range, y_pred, label=f'Regression Curve (K={K})') # 회귀 곡선 그리기

plt.xlabel('Weight')
plt.ylabel('Length')
plt.title('weight as k')
plt.legend()
plt.grid(True)
plt.show()
```

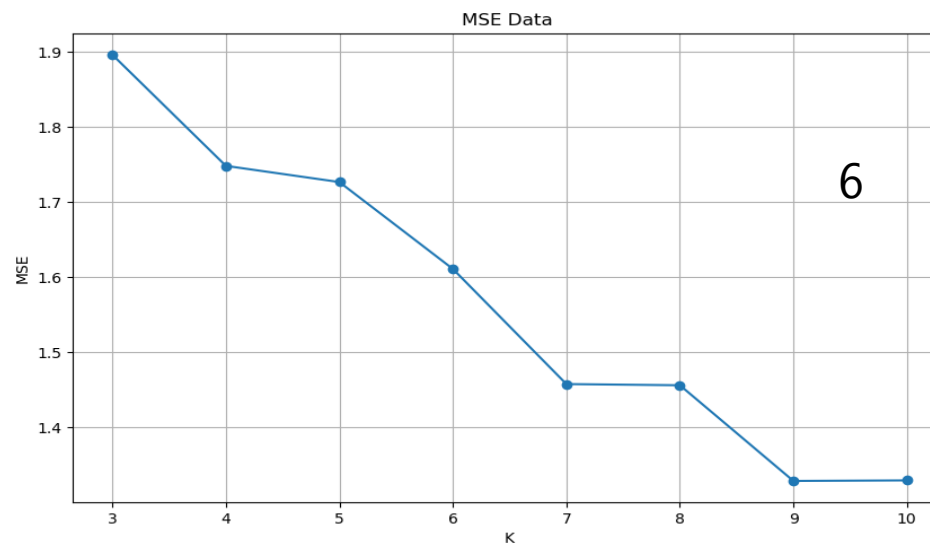
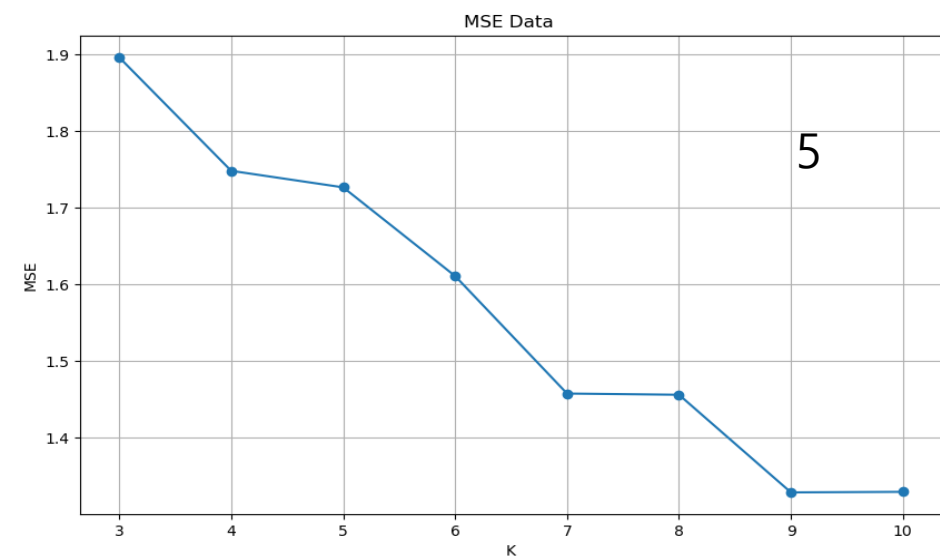
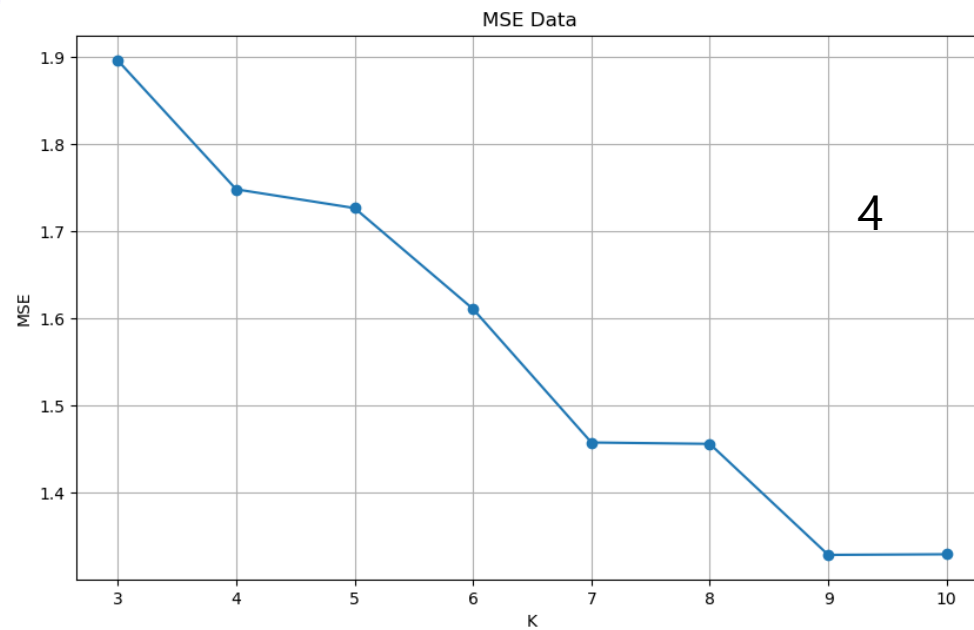
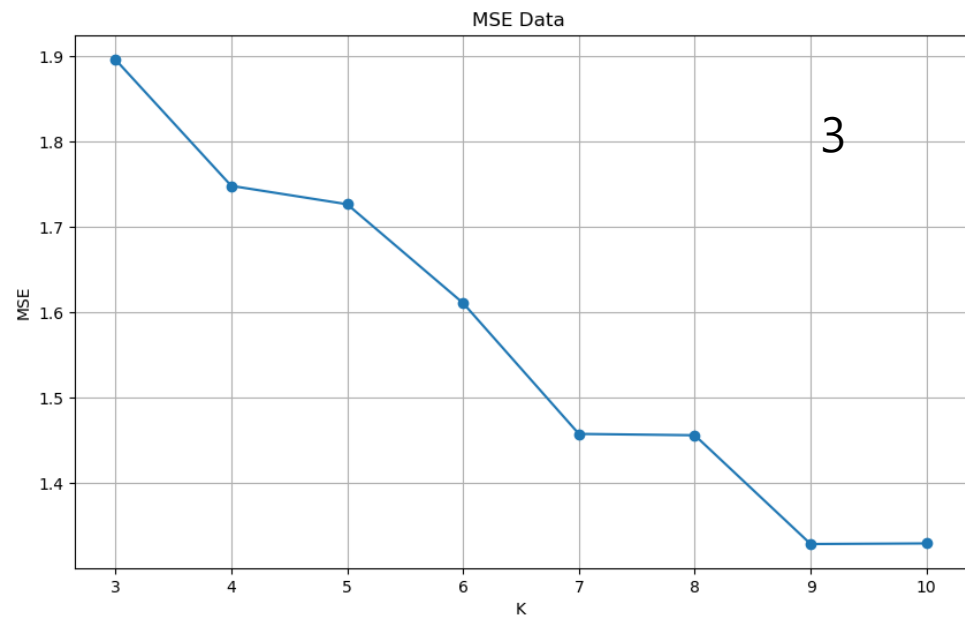
K개의 가우스 함수를 이용한 선형 기저함수 모델의 Analytic Solution을 구하는 사용자 지정함수를 구현하라.

실습 과제 4.2) $k=3,6,8$ 일 때에 대한 weight

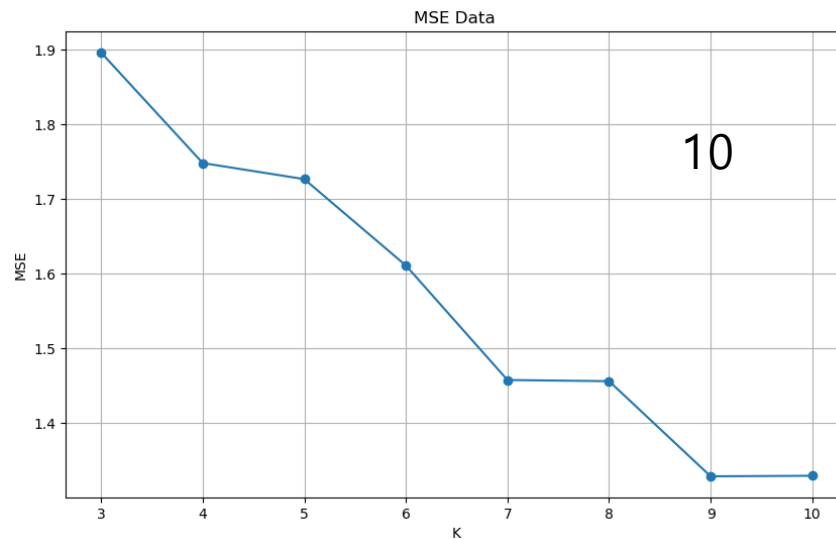
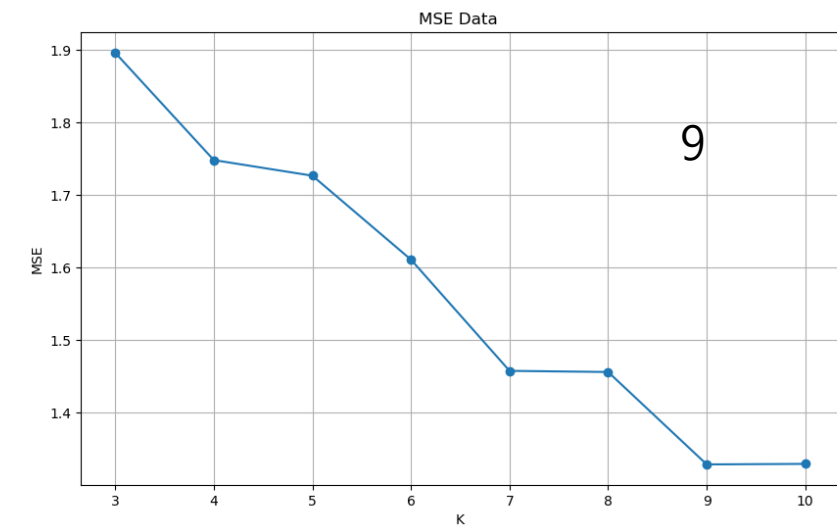
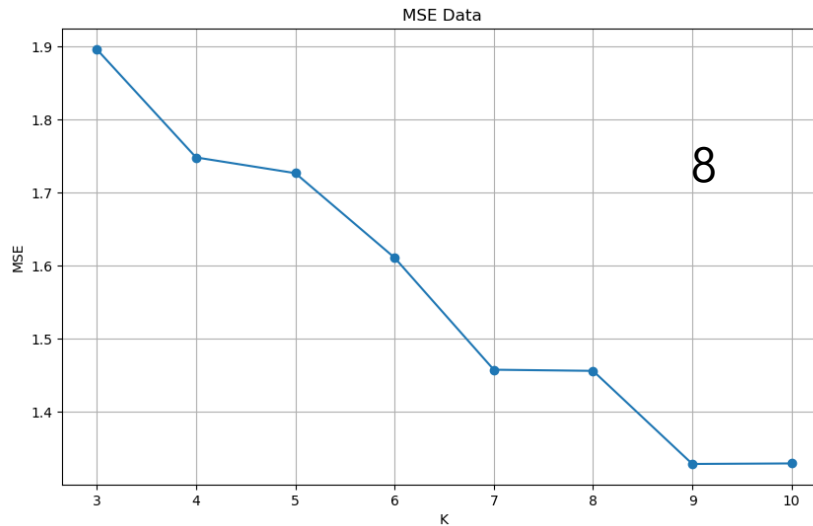
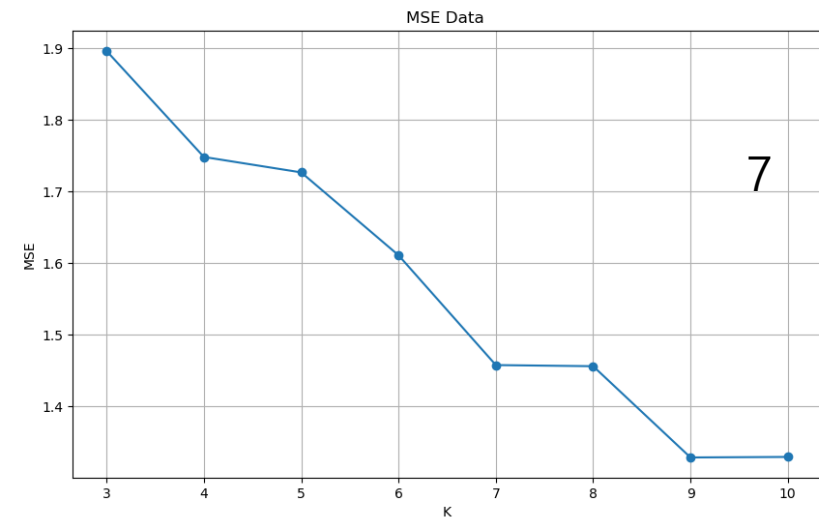


이 데이터에서 점은 원래 데이터이고 k 가 3,6,8에 대한 회귀곡선을 그려보았을 때 k 가 증가함에 따라 원래 데이터와 차이가 적게 나타난다는 것을 알 수 있다.

실습 과제 4.3) 3~10에 대한 mse값

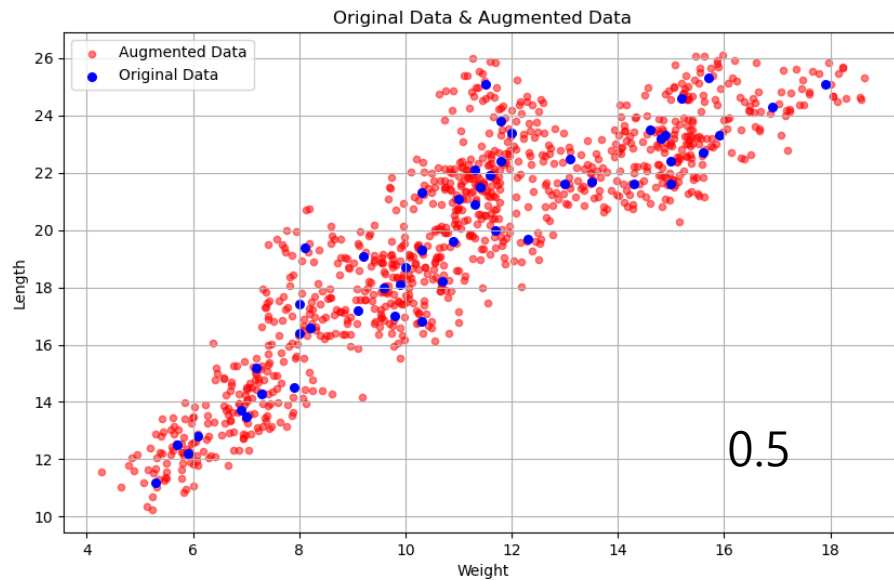
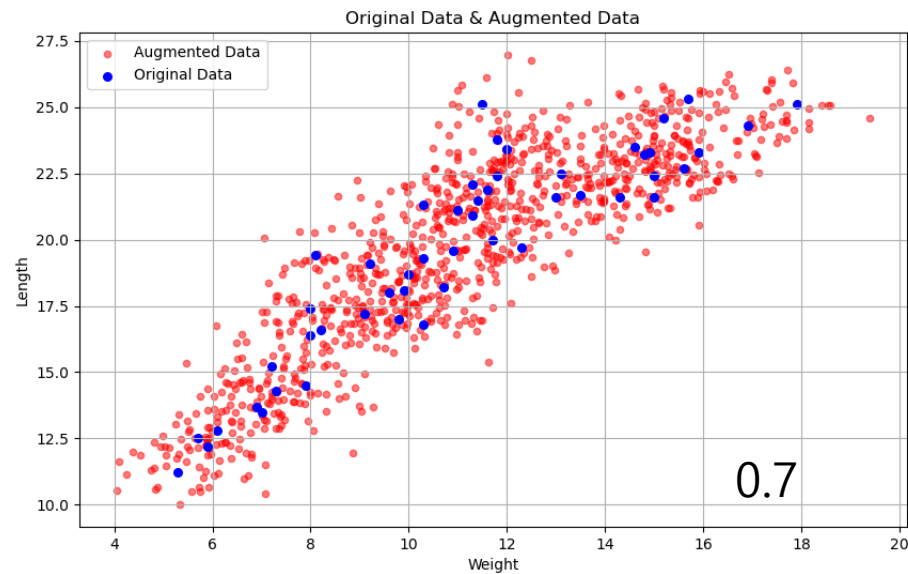
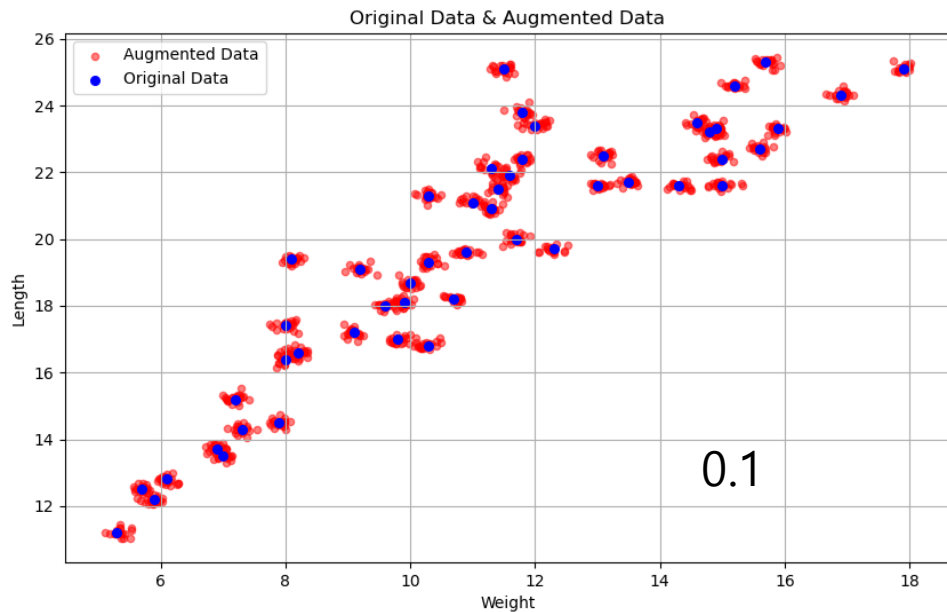


실습 과제 4.3) 3~10에 대한 mse값



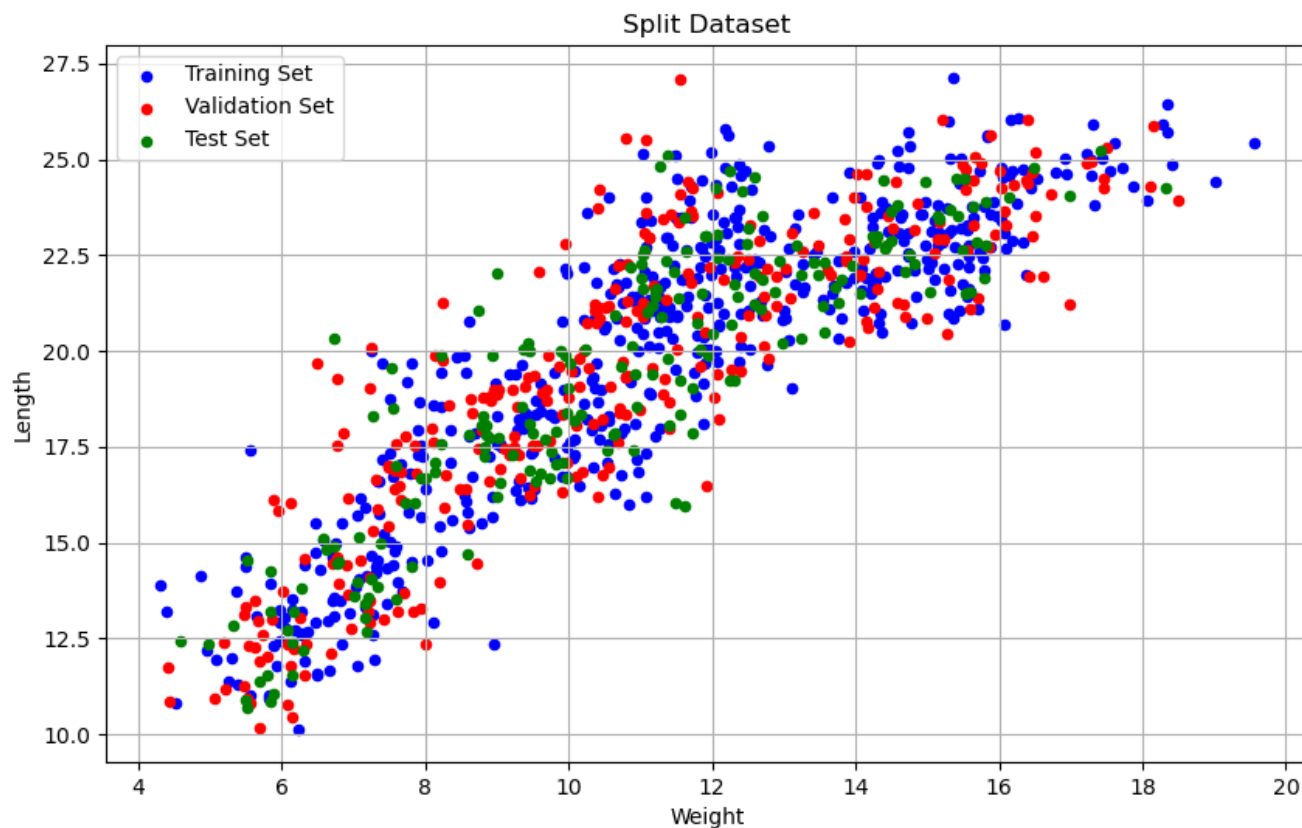
이전 슬라이드와 지금 슬라이드에서
k값에 따라서 mse 그래프를 비교했더니
그 값은 달라지지 않는다

실습 과제 chap.2-1



noise 크기를 이렇게 0.1, 0.5, 0.7로 설정해서 그래프를 그렸을 때 작게 하면 원래의 데이터에 별로 차이 없는 값으로 나오고 크게 하면 원래 데이터에서 차이가 많이 나는 값으로 그려지므로 적당한 값을 설정하는 것이 중요하다

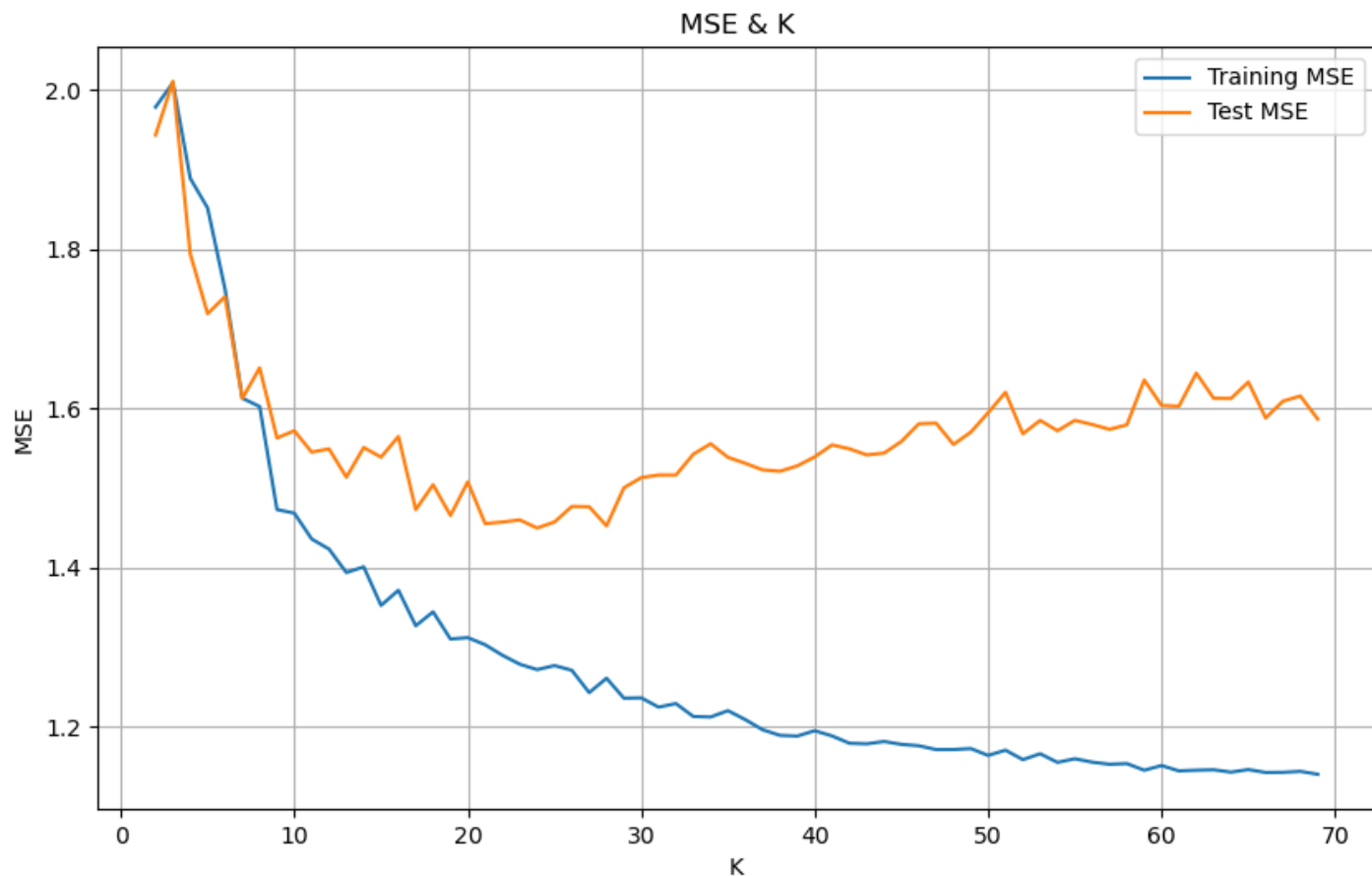
실습 과제 chap.2-2



test_set	Array of float64	(200, 2)
train_set	Array of float64	(500, 2)
val_set	Array of float64	(300, 2)

증강된 총 1000개의 데이터를 5:3:2로 분할했더니 각 데이터가 500,300,200개로 분리되었고 그 분리된 데이터는 실행마다 랜덤하게 분배됨을 알 수 있다.

실습 과제 chap.2-3



```
In [4]: runfile('C:/Users/user/OneDrive - 한국공학대학교/바탕 화면/3학년 1학기/머신러닝실습/Machine-Learning/7주차/train_set,test_set에 대한 mse그래프 추가.py', wdir='C:/Users/user/OneDrive - 한국공학대학교/바탕 화면/3학년 1학기/머신러닝실습/Machine-Learning/7주차')  
최적의 K 값: 41
```

이렇게 test_mse와, training_mse의 그래프를 그려보았고, test_mse의 그래프는 감소하다가 어느 순간에 증가함을 알 수 있고 이때의 증하가는 그 시작점이 최적의 k값임을 알 수 있다.