신체 조건을 통한 신장 예측하기

A반 김채현

CONTENTS

1. 문제 정의

국방의 의무를 위해 받는

신체검사 데이터를

활용할 수 있는 방법이 없을까?

신체 조건을 통해 사이즈 예측 모델 생성

2. 데이터분석 - 수집

공공데이터 포탈 -국방부_육군 신체측정정보

CSV 국방부_육군 신체측정정보

육군 신병 신체치수 측정정보(키, 몸무게, 허리둘레, 머리둘레 등)

0 0 관심

바로가기

오류신고 및 담당자 문의

* 2014년 1월 ~ 2017년 1월 (4년간)

* 167,983건의 데이터

파일데이터 정보 메타데이터 다운로드

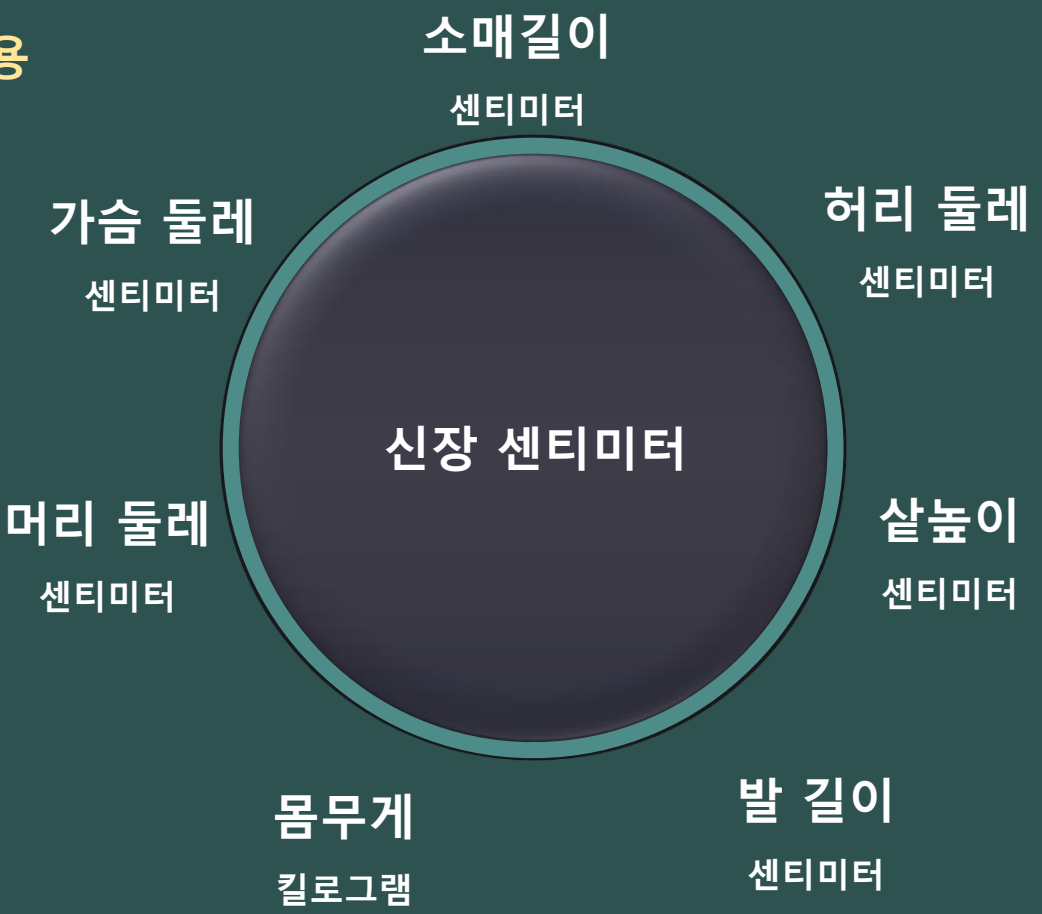
파일데이터명	국방부_육군 신체측정정보_20200930		
분류체계	국방 - 병무행정	제공기관	국방부
관리부서명	육군본부 군수참모부 보급근무과	관리부서 전화번호	042-550-4228
보유근거		수집방법	
업데이트 주기	수시	차기 등록 예정일	
매체유형	텍스트	전체 행	16983
확장자	CSV	다운로드(바로가기)	1997
데이터 한계		키워드	신병신체치수,신체치수측정,신체치수측정정보
등록	2014-11-20	수정	2020-10-27

2. 데이터분석 - 수집

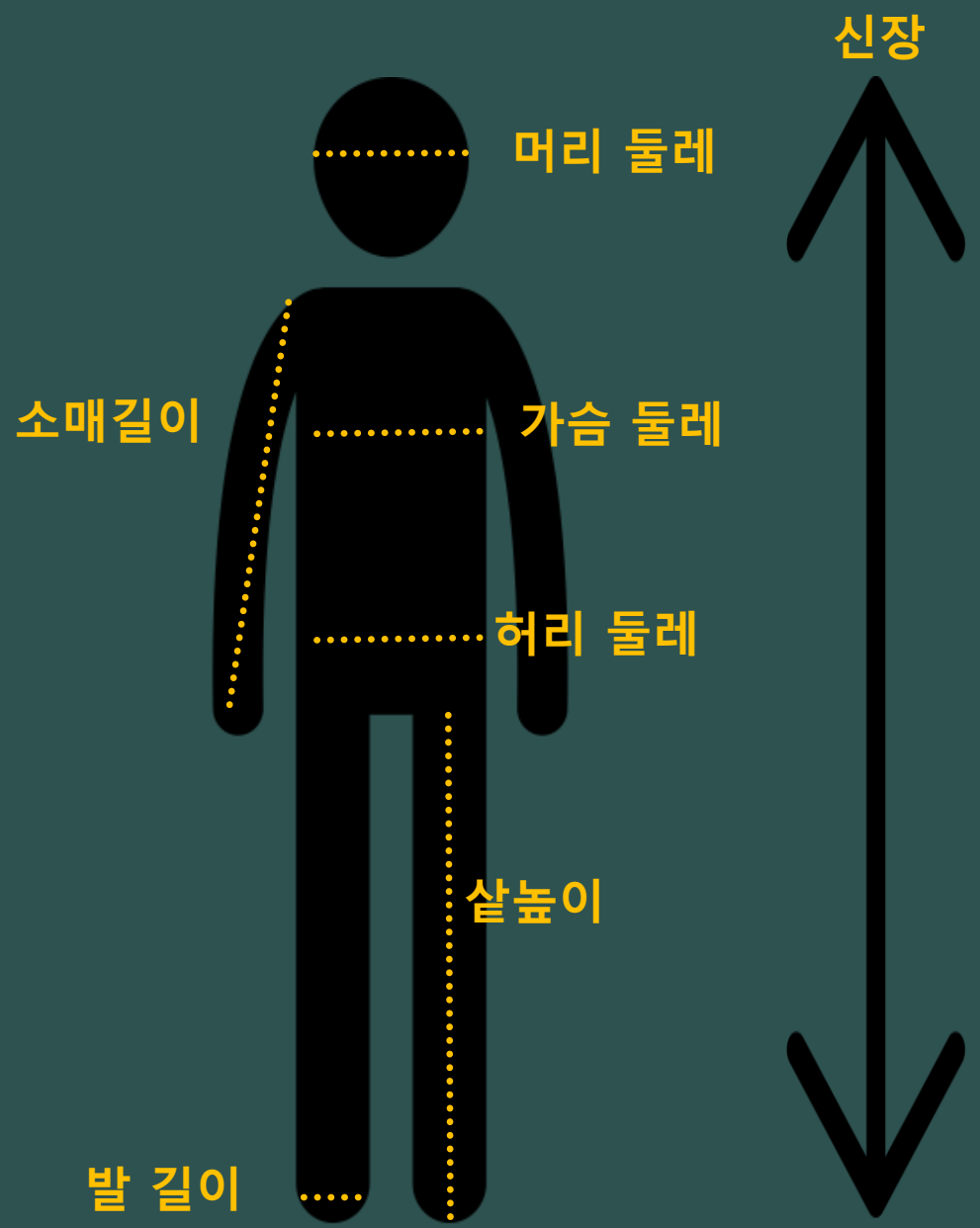
육군 신체 측정 데이터(수시 업데이터).csv

8개 columns만 사용

	순번	측정 일자	가슴 둘레	소매길이 /	신장 센티	허리 둘레	샅높이 센	머리 둘레	발 길이 센	몸무게 킬로그램
2	1	20140106	96.3	92.6	185.5	82.5	83.8	57.1	28.5	73.2
3	2	2013	101.9	83.8	167.2	81.2	74.4	55	24.2	65.1
4	3	20140106	99.5	89.6	179.9	99.2	84.2	56.3	28.1	93.2
5	4	2013	98.6	85.5	171.4	85.9	76.4	54.5	27.4	71.6
6	5	20140106	94.7	88.1	176.9	78.3	79.7	58.5	26.5	67.1
7	6	2013	122.3	90.2	180.5	118.4	79.7	61.5	29	114.2
8	7	20140106	98	89.7	180.4	93.6	83	59.9	27.1	76.5
9	8	2013	89.6	87.3	173.9	71.1	76.5	53.9	26.8	53
10	9	20140106	93.8	89.3	178.7	82.9	83.8	57.7	28.6	69.7
11	10	2013	122.3	90.2	180.5	118.4	79.7	61.5	29	114.2
12	11	20140106	115.4	86.5	173.1	112.6	80.6	59.1	27.7	95.3
13	12	2013	98	89.6	178.5	95.4	78.4	57.1	27.2	78.7
14	13	20140106	83.1	86	172.6	75.7	75.4	57.9	26.2	58.6
15	14	2013	94.7	85.1	170.7	81.7	75.3	56.4	26.2	66.7
16	15	20140106	97.1	87.7	175	87.3	78.9	56.4	27.4	77.8
17	16	2013	98.8	87.7	175.8	89.6	77.6	58.5	26.2	77.8
18	17	20140106	95.9	91.3	182.9	85.8	86	58.4	29.2	80.4



2. 데이터분석 - 수집


신장
머리 둘레
소매길이
가슴 둘레
허리 둘레
샅높이
발 길이

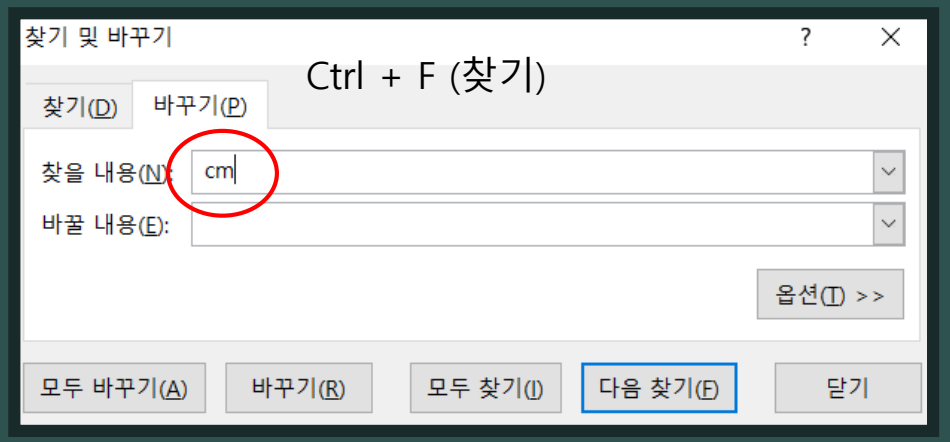

소매 길이
센티미터
가슴 둘레
센티미터
허리 둘레
센티미터
신장 센티미터
머리 둘레
센티미터
샅높이
센티미터
몸무게
킬로그램
발 길이
센티미터

2. 데이터분석 – 데이터셋 생성하기

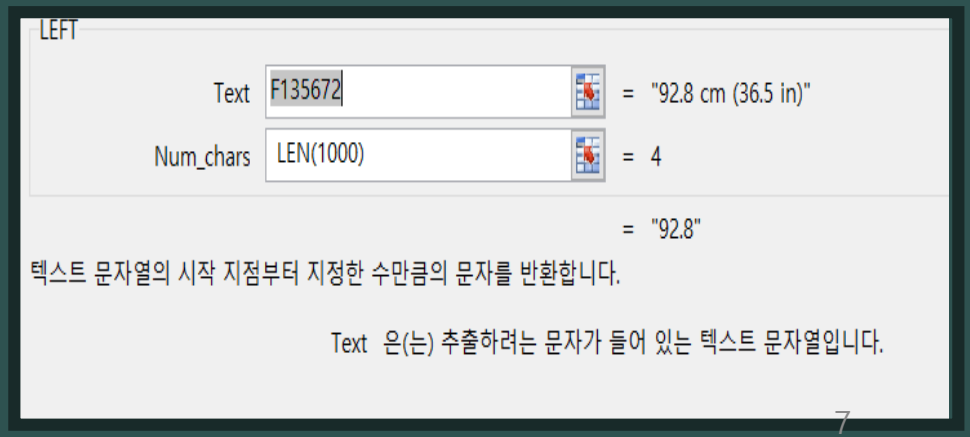
단위 삭제하기

	측정 일자	가슴 둘레 센티미터	소매길이 센티미터	신장 센티미터	허리 둘레 센티미터	샅높이 센티미터	머리 둘레 센티미터	발 길이 센티미터	몸무게 킬로
167967	20170131	88.2 cm	89.3 cm	177.6 cm	76.4 cm	(30.1 in)	78.9 cm	57.6 cm	30.0 cm	59.4 kg
167968	20170131	84.8 cm	86.2 cm	173.0 cm	75.6 cm	(29.8 in)	78.5 cm	59.6 cm	29.3 cm	57.4 kg
167969	20170131	91.1 cm	89.3 cm	179.2 cm	83.1 cm	(32.7 in)	77.9 cm	58.2 cm	28.8 cm	67.3 kg
167970	20170131	95.9 cm	86.1 cm	172.3 cm	86.0 cm	(33.8 in)	76.1 cm	58.4 cm	27.6 cm	64.4 kg
167971	20170131	97.1 cm	85.1 cm	169.9 cm	91.7 cm	(36.1 in)	78.1 cm	59.0 cm	28.9 cm	73.7 kg
167972	20170131	114.1 cm	85.3 cm	171.6 cm	115.1 cm	(45.3 in)	80.2 cm	62.1 cm	28.0 cm	104.4 kg
167973	20170131	96.7 cm	85.0 cm	169.4 cm	86.5 cm	(34.1 in)	74.3 cm	60.7 cm	28.1 cm	60.7 kg
167974	20170131	84.6 cm	82.8 cm	166.5 cm	78.5 cm	(30.9 in)	78.2 cm	58.3 cm	27.2 cm	60.8 kg
167975	20170131	97.2 cm	84.2 cm	169.2 cm	90.4 cm	(35.6 in)	75.0 cm	61.7 cm	29.1 cm	72.4 kg
167976	20170131	109.6 cm	84.6 cm	169.5 cm	106.4 cm	(41.9 in)	78.6 cm	60.4 cm	25.7 cm	89.8 kg
167977	20170131	94.6 cm	89.2 cm	177.8 cm	76.0 cm	(29.9 in)	78.7 cm	59.6 cm	29.5 cm	65.7 kg
167978	20170131	95.9 cm	86.8 cm	173.3 cm	88.7 cm	(34.9 in)	80.7 cm	57.3 cm	26.7 cm	73.8 kg
167979	20170131	100.6 cm	86.4 cm	173.2 cm	86.7 cm	(34.1 in)	77.1 cm	56.9 cm	27.1 cm	73.7 kg
167980	20170131	97.9 cm	82.7 cm	166.3 cm	90.5 cm	(35.6 in)	73.4 cm	61.5 cm	28.8 cm	70.7 kg
167981	20170131	97.9 cm	87.2 cm	175.2 cm	94.3 cm	(37.1 in)	83.1 cm	57.1 cm	27.6 cm	78.7 kg
167982	20170131	85.3 cm	86.8 cm	173.6 cm	75.9 cm	(29.9 in)	76.6 cm	57.6 cm	24.3 cm	57.4 kg
167983	20170131	86.6 cm	88.3 cm	175.8 cm	73.0 cm	(28.7 in)	77.8 cm	56.3 cm	26.1 cm	51.7 kg

1. cm, kg 삭제



2. Inch 변환 삭제



2. 데이터분석 – 데이터셋 생성하기

이상치 삭제하기

```
1 df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 167983 entries, 0 to 167982
Data columns (total 10 columns):
 #   Column       Non-Null Count   Dtype
---  ------       --------------   -----
 0   순번           167983 non-null  int64
 1   측정 일자        167983 non-null  int64
 2   가슴 둘레 센티미터  167982 non-null  float64
 3   소매길이 센티미터   167983 non-null  float64
 4   신장 센티미터      167983 non-null  float64
 5   허리 둘레 센티미터  167983 non-null  float64
 6   샅높이 센티미터     167983 non-null  float64
 7   머리 둘레 센티미터  167981 non-null  float64
 8   발 길이 센티미터    167983 non-null  float64
 9   몸무게 킬로그램     167983 non-null  float64
dtypes: float64(8), int64(2)
memory usage: 12.8 MB
```

데이터 수: 167,980개

```
1 rdf = df.dropna(subset=['머리 둘레 센티미터', '가슴 둘레 센티미터'], how='any', axis=0)
2 rdf.info()
3 # 머리 둘레, 가슴 둘레 null값 삭제
4 # null값 지우고 167980개 통일
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 167980 entries, 0 to 167982
Data columns (total 10 columns):
 #   Column       Non-Null Count   Dtype
---  ------       --------------   -----
 0   순번           167980 non-null  int64
 1   측정 일자        167980 non-null  int64
 2   가슴 둘레 센티미터  167980 non-null  float64
 3   소매길이 센티미터   167980 non-null  float64
 4   신장 센티미터      167980 non-null  float64
 5   허리 둘레 센티미터  167980 non-null  float64
 6   샅높이 센티미터     167980 non-null  float64
 7   머리 둘레 센티미터  167980 non-null  float64
 8   발 길이 센티미터    167980 non-null  float64
 9   몸무게 킬로그램     167980 non-null  float64
dtypes: float64(8), int64(2)
memory usage: 14.1 MB
```

2. 데이터분석 – 데이터셋 생성하기

정규화 후 검증 데이터셋 만들기

```
# 분석에 활용할 속성을 선택 (가슴 둘레, 소매길이, 허리 둘레, 샅높이, 머리둘레, 발길이, 몸무게)
X = rdf.iloc[:, [2,3,5,6,7,8,9]] # 독립 변수
y = rdf['신장 센티미터']         # 종속 변수
```

정규화

```
# 숫자의 수와 차이가 크기 때문에 정규화 작업
from sklearn import preprocessing

X = preprocessing.StandardScaler().fit(X).transform(X)
```

검증 데이터셋 형성

```
# 검증 데이터셋을 만듭니다
from sklearn.model_selection import train_test_split

X_train, X_test, y_train, y_test  = train_test_split(X, y, test_size=0.30, random_state=777) # 랜덤 추출 값

```

Train: Test=7:3

Validation 추가

```
# 검증 데이터셋을 만듭니다
from sklearn.model_selection import  train_test_split

X_train, X_val, y_train, y_val = train_test_split(X_train,          # 독립 변수
                                                  y_train,          # 종속 변수
                                                  test_size=0.33,   # 검증 33%
                                                  random_state=777) # 랜덤 추출 값

```

np.array

```
import numpy as np
np.set_printoptions(suppress=True)

X_train=np.array(X_train)
X_test=np.array(X_test)
y_train=np.array(y_train)
y_test=np.array(y_test)
X_val=np.array(X_val)
y_val=np.array(y_val)
```

```
import numpy as np
np.set_printoptions(suppress=True)
```

3. 모델 구성 및 설정

모델 구성하기

```
[20]  1 from tensorflow.keras.models import Sequential
      2 from tensorflow.keras.layers import Dense
      3
      4 model = Sequential()
      5 #model.add(Dense(activation='relu', input_shape=(7,)))
      6
      7 model.add(Dense(32, activation='relu', input_shape=(7,)))
      8 model.add(Dense(1)) # 하나의 값을 출력 -> 신장 길이
      9
     10 # 활성화 함수를 안 적은 이유는 리니어가 디폴트이기 때문에 안 적어도 작동 됨
```

모델 설정하기

```
1 model.compile(optimizer = 'adam', loss='mse', metrics=['mae', 'mse'])
```

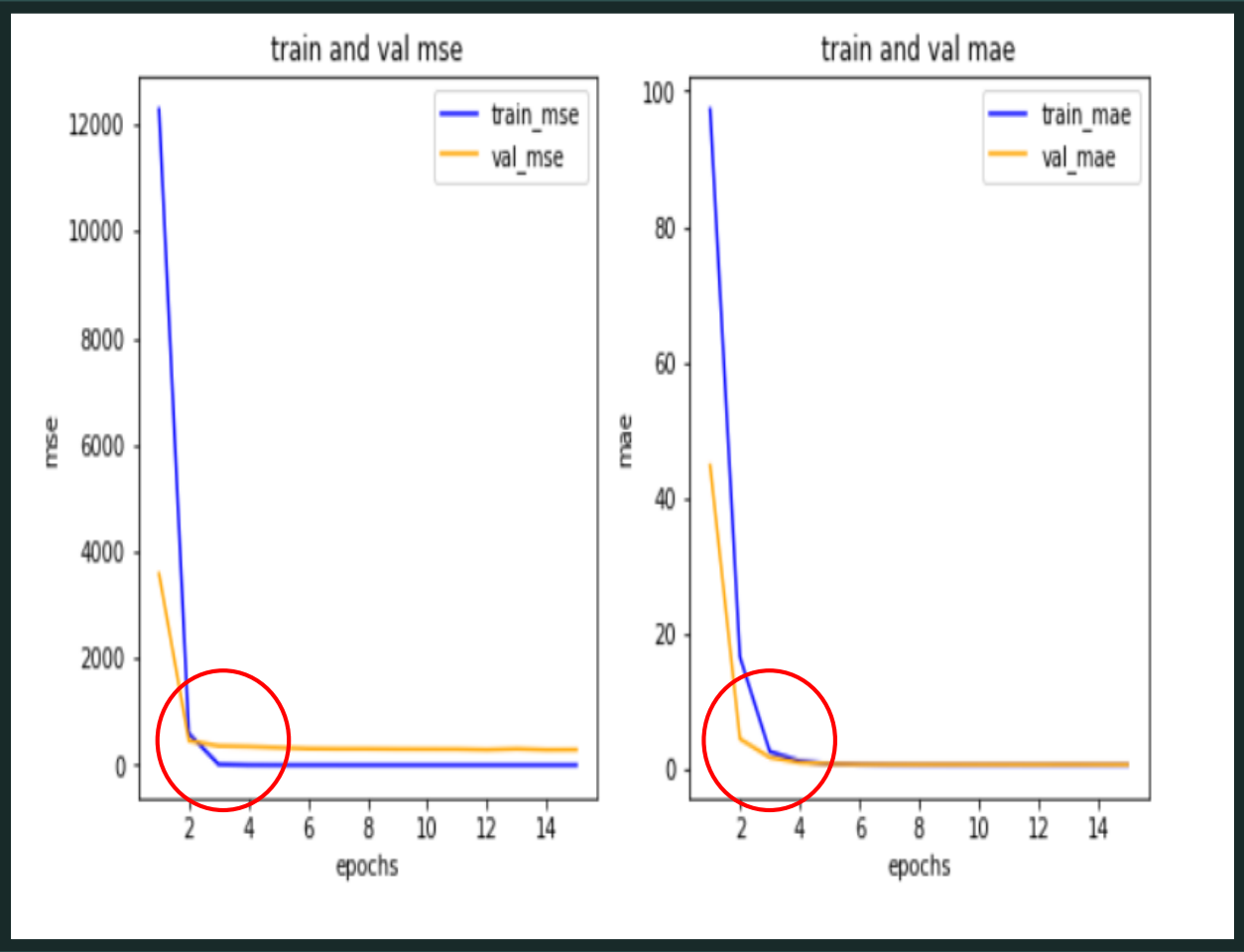
4. 모델 학습 및 예측 검증

```python
history = model.fit(X_train, y_train,
                    epochs=15,                    # 15번 공부시키기
                    validation_data = (X_val, y_val))
```

```
Epoch 1/15
2462/2462 [==============================] - 9s 4ms/step - loss: 12283.9873 - mae: 97.3091 - mse: 12283.9873 - val_loss: 3584.5686 - val_mae: 44.7307 - val_mse: 3584.5686
Epoch 2/15
2462/2462 [==============================] - 9s 3ms/step - loss: 592.3273 - mae: 16.4668 - mse: 592.3273 - val_loss: 459.8058 - val_mae: 4.3577 - val_mse: 459.8058
Epoch 3/15
2462/2462 [==============================] - 9s 4ms/step - loss: 13.4206 - mae: 2.4921 - mse: 13.4206 - val_loss: 359.3657 - val_mae: 1.6180 - val_mse: 359.3657
Epoch 4/15
2462/2462 [==============================] - 9s 4ms/step - loss: 2.3189 - mae: 1.0670 - mse: 2.3189 - val_loss: 348.5815 - val_mae: 0.8623 - val_mse: 348.5815
Epoch 5/15
2462/2462 [==============================] - 9s 4ms/step - loss: 0.7593 - mae: 0.6493 - mse: 0.7593 - val_loss: 328.8500 - val_mae: 0.6656 - val_mse: 328.8500
Epoch 6/15
2462/2462 [==============================] - 9s 4ms/step - loss: 0.4423 - mae: 0.5301 - mse: 0.4423 - val_loss: 309.2267 - val_mae: 0.5893 - val_mse: 309.2267
Epoch 7/15
2462/2462 [==============================] - 9s 3ms/step - loss: 0.3635 - mae: 0.4953 - mse: 0.3635 - val_loss: 304.8917 - val_mae: 0.5807 - val_mse: 304.8917
Epoch 8/15
2462/2462 [==============================] - 9s 3ms/step - loss: 0.3403 - mae: 0.4856 - mse: 0.3403 - val_loss: 304.6197 - val_mae: 0.5729 - val_mse: 304.6197
Epoch 9/15
2462/2462 [==============================] - 10s 4ms/step - loss: 0.3327 - mae: 0.4824 - mse: 0.3327 - val_loss: 300.8052 - val_mae: 0.5768 - val_mse: 300.8052
Epoch 10/15
2462/2462 [==============================] - 9s 4ms/step - loss: 0.3309 - mae: 0.4821 - mse: 0.3309 - val_loss: 299.5128 - val_mae: 0.5708 - val_mse: 299.5128
Epoch 11/15
2462/2462 [==============================] - 9s 4ms/step - loss: 0.3290 - mae: 0.4817 - mse: 0.3290 - val_loss: 298.9938 - val_mae: 0.5662 - val_mse: 298.9938
Epoch 12/15
2462/2462 [==============================] - 10s 4ms/step - loss: 0.3249 - mae: 0.4793 - mse: 0.3249 - val_loss: 289.0409 - val_mae: 0.5623 - val_mse: 289.0409
Epoch 13/15
2462/2462 [==============================] - 9s 3ms/step - loss: 0.3267 - mae: 0.4808 - mse: 0.3267 - val_loss: 304.3545 - val_mae: 0.5775 - val_mse: 304.3545
Epoch 14/15
2462/2462 [==============================] - 9s 4ms/step - loss: 0.3263 - mae: 0.4809 - mse: 0.3263 - val_loss: 287.2889 - val_mae: 0.5673 - val_mse: 287.2889
```

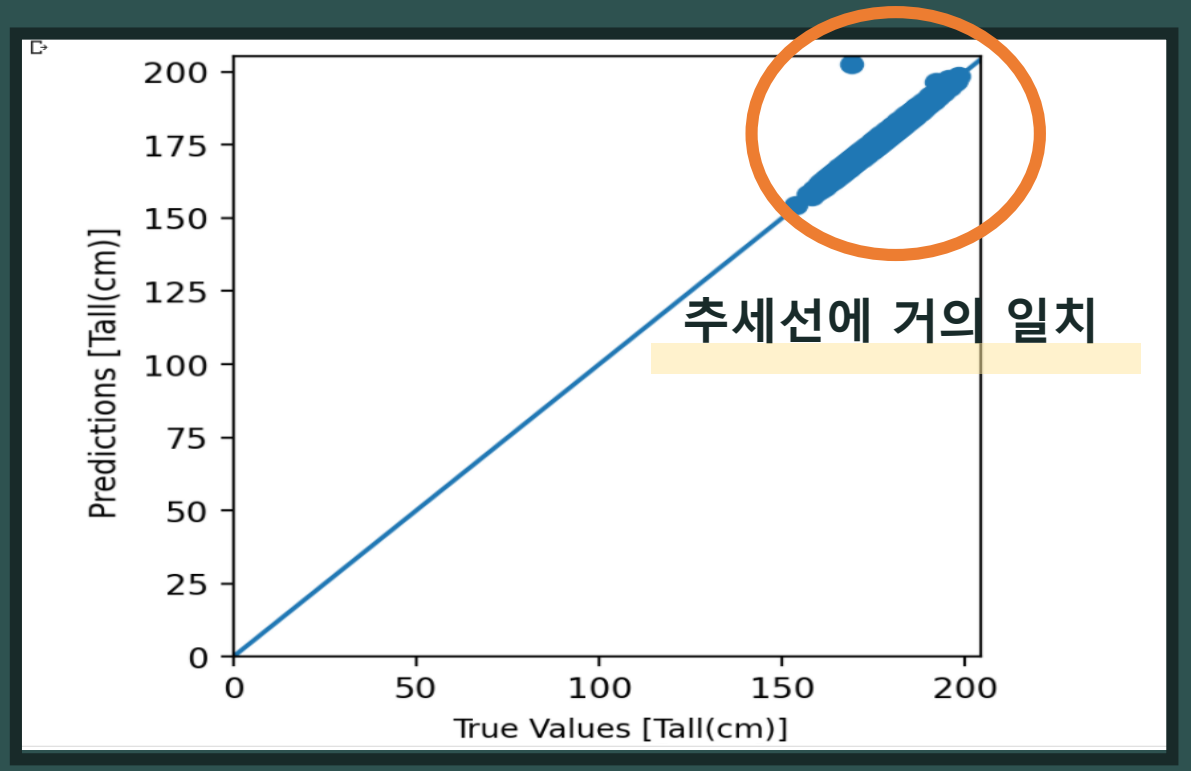
감소

4. 모델 학습 및 예측 검증



훈련 및 검증 손실

훈련 및 검증 정확도



mae: 0.4703

```
1 model.evaluate(X_test, y_test)
```

mae의 단위는?

```
1575/1575 [==============================] - 3s 2ms/step - loss: 0.3293 - mae: 0.4703 - mse: 0.3293
[0.329255610704422, 0.47029227018356323, 0.329255610704422]
```

4. 모델 학습 및 예측 검증

검증하기

머리 둘레 57.1cm

가슴 둘레 93.2cm

소매 길이 92.6cm

허리 둘레 82.5cm

샅높이 83.8cm

발 길이 28.5cm

몸무게 73.2kg

```
1 man_a = np.array([ 0.07696313,  1.93356124, -0.35272211,  1.17775554, -0.1031317 ,1.19240925,  0.07602794]).reshape(1,7)
2 #가슴 둘레 96.3cm , 소매길이 92.6cm, 허리 둘레 82.5cm, 샅높이 83.8cm, 머리 둘레 57.1cm, 발 길이 28.5cm, 몸무게 73.2kg
3 model.predict(man_a)
4
```

```
array([[185.05519]], dtype=float32)
```

```
array([[185.05519]], dtype=float32)
```

이런 신체조건을 가지고 있는

사람의 실제 키는?

185.5cm

실제 185.5cm vs 185.1cm 모델 예측값

4. 모델 학습 및 예측 검증

검증하기

머리 둘레 55cm

가슴 둘레 101.9cm

소매 길이 83.8cm

허리 둘레 81.2m

샅높이 74.4cm

발 길이 24.2cm

몸무게 65.1kg

이 사람의 실제 키는? **167.2cm**

모델은 어떻게 예측했을까?

```
1 man_b = np.array([ 0.68633539, -1.2816482 , -0.46795998, -1.30406627, -0.31699406, -1.88407218, -0.51341162]).reshape(1,7)
2 #가슴 둘레 101.9cm , 소매길이 83.8cm, 허리 둘레 81.2cm, 샅높이 74.4cm, 머리 둘레 55cm, 발 길이 24.2cm, 몸무게 65.1kg
3 model.predict(man_b)
```

```
array([[167.77908]], dtype=float32)
```

```
array([[167.77908]], dtype=float32)
```

실제 **167.2cm** vs 모델 예측값 **167.8cm**

4. 모델 학습 및 예측 검증

mae: 0.4703

```
1 model.evaluate(X_test, y_test)
```

```
1575/1575 [==============================] - 3s 2ms/step - loss: 0.3293 - mae: 0.4703 - mse: 0.3293
[0.329255610704422, 0.4702922701835632 3, 0.329255610704422]
```

mae: 0.4703 => 오차 범위 약 ± 0.5cm

낮은 오차 범위 => 높은 예측률

5. 시사점 및 한계점

생활관 리뉴얼시, 가구 제작에 활용

군복의 사이즈 수요를
미리 파악하여 공급 원활에 도움

신체 사이즈 예측 프로그램을 통해
신체 검사 시간 및 측정 비용 절약 가능

일반 의류산업의 맞춤형 사이즈 추천 프로그램으로 활용 가능

한계점

- 성별이 명시되어 있지 않음
- 성장기인 청소년층에 적용하기 어려움

6. 느낀점

한계의 직면

한계에 많이 부딪히며 배움
=> 오류 극복을 통한 짜릿한 성취감

깨달음

포기하는 법도 배워야 한다
=> 포기하는 법을 아는 자는 커다란 한계에 부딪혀 앞이 캄캄해 졌을 때, 평정을 되찾고 좌표를 새로이 세울 수 있다

책 '담백한 인생이 행복하다'-작가 무무

7. 참조 링크

[데이터 수집] - 국방부_육군 신체측정정보
https://www.data.go.kr/data/15083227/fileData.do

[픽토그램]
https://thenounproject.com/

[사진 자료]
옥택연 사진
@Twitter ‘Baque1227’

임시완 사진
https://entertain.naver.com/read?oid=421&aid=0003903570

Thank you!