IN THE NAME OF **ALLAH**

Linear Regression

**Implementing a linear regression for a real data set
(body measurements) with scikit-learn (sklearn) library in python**

Seyed Mahdi Mahdavi Mortazavi - 40030490

Numerical Computing Methods

July 2023

By theMHD

Introduction

What is the linear regression?

Linear regression analysis is used to predict value of a variable (dependent variable) based on the value of another variables (independent variables) in a data set.

Linear regression model takes this form:

$$y_i = a_0 + a_1x_1 + a_2x_2 + a_3x_3 + \ldots + a_nx_n + \varepsilon_i = a_0 + \sum_{i=[1,n]} a_ix_i + \varepsilon_i$$

y $\rightarrow$ dependent variable / x $\rightarrow$ independent variable / a_1 to a_n $\rightarrow$ coefficients of independent variables / a_0 $\rightarrow$ intercept of linear equation / ε_i $\rightarrow$ error of prediction

The goal of this project is to predict a person' age using his/her body measurements (with a real data set) …

```python
print("\n||| ------------------- Step 1: General file inf
print('1) The first 5 rows ---------------------------')
print(body_measures.head())

print('\n2) File information -------------------------')
print(body_measures.info())

print('\n3) Statistical information ------------------')
print(body_measures.describe())
```

```
||| ------------------- Step 1: General file information ------------------- |||

1) The first 5 rows ---------------------------
   Gender  Age  HeadCircumference  ...  WaistToKnee  LegLength  TotalHeight
0       1   30                 22  ...           25         22           52
1       1   28                 19  ...           25         20           56
2       2   27                 21  ...           14         18           53
3       1   29                 20  ...           20         21           45
4       2   28                 16  ...           32         13           47

[5 rows x 13 columns]
```

Let's go code!

Step 1

Get some general information from our data set file (part1)

.head() method and it's results (it shows the first 5 rows of data set file as it's header)

5 rows and 13 columns

More with code! …

```
2) File information -------------------------
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 716 entries, 0 to 715
Data columns (total 13 columns):
 #   Column            Non-Null Count  Dtype
---  ------            --------------  -----
 0   Gender            716 non-null    int64
 1   Age               716 non-null    int64
 2   HeadCircumference 716 non-null    int64
 3   ShoulderWidth     716 non-null    int64
 4   ChestWidth        716 non-null    int64
 5   Belly             716 non-null    int64
 6   Waist             716 non-null    int64
 7   Hips              716 non-null    int64
 8   ArmLength         716 non-null    int64
 9   ShoulderToWaist   716 non-null    int64
 10  WaistToKnee       716 non-null    int64
 11  LegLength         716 non-null    int64
 12  TotalHeight       716 non-null    int64
dtypes: int64(13)
memory usage: 72.8 KB
None
```

```
3) Statistical information ------------------
              Gender          Age  ...    LegLength  TotalHeight
count     716.000000   716.000000  ...   716.000000   716.000000
mean        1.452514    15.340782  ...    26.833799    48.118715
std         0.498088    11.831501  ...     7.925988    12.156722
min         1.000000     1.000000  ...     9.000000    19.000000
25%         1.000000     7.000000  ...    21.000000    40.000000
50%         1.000000    11.000000  ...    26.000000    48.000000
75%         2.000000    21.000000  ...    32.000000    55.000000
max         2.000000    68.000000  ...    50.000000    89.000000

[8 rows x 13 columns]
```

Step 2 (.info())
Get some general information
from our data set file (part2)

Step 3 (.describe())
Get some statistical information about our data set file …
Such as mean, std (root of variance), min, max and …

Create regression line

```python
print("\n||| --------------------- Step 2: Make regression line --------------------- |||")
x_list = body_measures[
    ["Gender", "HeadCircumference", "ShoulderWidth", "ChestWidth", "Belly", "Waist", "Hips",
     "ArmLength", "ShoulderToWaist", "WaistToKnee", "LegLength", "TotalHeight"]]
y_list = body_measures["Age"]
reg_res = LinearRegression()
reg_res.fit(x_list, y_list)

i = 0   # index of coefficients
print("The coefficient of variables are shown in bottom...")
for header in body_measures.columns:
    if header != "Age":
        print(f"{header}: {reg_res.coef_[i]}")
        i += 1
print("-----------------------------------")
print("Intercept of regression line is: ", reg_res.intercept_)

# Predict Ages and calculate error -----------------------------------------------
predicted_y = reg_res.predict(x_list)  # predicated y list (y is list Ages)
rmse = np.sqrt(np.mean(np.square(predicted_y - y_list)))  # root mean square error
print("Average prediction error: ", rmse, " ~= ", round(rmse))
```

With .coef_ attribute, we gain the coefficients of independent variables (x's) and with .intercept_ attribute, we gain the intercept of our regression line
...
Ok; the regression line is created for our dependent variable (Age → y) and independent variables (other body measurements → x_0 to x_{11}).

To create regression line, at first, we should consider one of variables as dependent variable (y) and other variables as independent variables (x)
...

In this project, Age is y and other variables considered as x
...

LinearRegression() class, prerforms the regression operation for us;

With fit() method, we can fit our x's and y for regression operation as dependent and independent variables.

```
||| -------------------- Step 2: Make regression line -------------------- |||
The coefficient of variables are shown in bottom...
Gender: -2.1600681758606353

HeadCircumference: -0.29175212771841047

ShoulderWidth: 0.4913502191932724

ChestWidth: -0.2637862587951137

Belly: -0.043528822433128
Waist: 0.15838333238679603

Hips: 0.1655076168425505

ArmLength: 0.23630518352852511

ShoulderToWaist: 0.589786941413992

WaistToKnee: 0.34539183792808575

LegLength: -0.06615631219156848

TotalHeight: 0.12851226045314823
---------------------------------
Intercept of regression line is:  -9.222430674200167
Average prediction error:  8.365045897383036  ~=  8
```

Create regression line (results)

The results obtained from performing linear regression operations to predict age;

…

Coefficients of independent variables and intercept of regression line.

…

With predict() method, we can predict our desired option (which is Age here) based on the different independent variables;
(shown in the previous slide)

…

To calculate error of our prediction, we use RMSE formula;
<< Root mean square error >>
← It's formula (predicted Ages – actual Ages)
N is the number of y's (from first age to the last).

$$RMSE = \sqrt{\frac{\sum_{i=1}^{N}(Predicted_i - Actual_i)^2}{N}}$$

Solve an example

In this part, we solve an example to check our prediction using the created equation (to predict age of a person with his/her body measurements);

…

In general we want to check obtained result of linear regression. ☺

```python
print("\n||| ----------------------- Step 3: Solve an example ----------------------- |||")
# You can use this numbers for example: 1, 20, 25, 24, 25, 36, 17, 23, 12, 9, 19, 48
x_for_predict = [[]]
for header in x_list:
    x_for_predict[0].append(int(input(f"Enter a number for input var <{header}>: ")))
predicted_example = reg_res.predict(x_for_predict)[0]
print("An example to prediction: ", abs(predicted_example), " ~= ", round(abs(predicted_example)), "years old")
```

```
||| ----------------------- Step 3: Solve an example ----------------------- |||
Enter a number for input var <Gender>: 1
Enter a number for input var <HeadCircumference>: 20
Enter a number for input var <ShoulderWidth>: 25
Enter a number for input var <ChestWidth>: 24
Enter a number for input var <Belly>: 25
Enter a number for input var <Waist>: 36
Enter a number for input var <Hips>: 17
Enter a number for input var <ArmLength>: 23
Enter a number for input var <ShoulderToWaist>: 12
Enter a number for input var <WaistToKnee>: 9
Enter a number for input var <LegLength>: 19
Enter a number for input var <TotalHeight>: 48
An example to prediction:  16.695157418056837  ~=  17 years old
```

Let's go to draw the plots …

Here is the code for the charts that will be shown in the next slides;

Note: the regression line in the graphs, created for Age based on **one** variable; that is, **Age based on the per independent variables**; also actual age values and predicted values (that obtained from regression operation (with all independent variables)) will be show along with the regression line.

The Charts and Plots ☺

```python
print("\n||| ------------------------ Step 4: Draw the plots ------------------------ |||")
for header in x_list:
    plt.grid()
    # New regression for per input
    reg_res = LinearRegression()
    reg_input = x_list[header].values.reshape(-1, 1)
    reg_res.fit(reg_input, y_list)
    y_reg_line = reg_res.coef_ * x_list[header] + reg_res.intercept_
    # The plot inputs
    plt.scatter(x_list[header], y_list, color="purple")
    plt.scatter(x_list[header], predicted_y, color="orange")
    plt.plot(x_list[header], y_reg_line, color="red")
    # The plot styles
    plt.xlabel(header, fontsize=15)
    plt.ylabel("Age", fontsize=15)
    plt.legend(["Original", "Predicted", "Regression line"])
    plt.show()
```
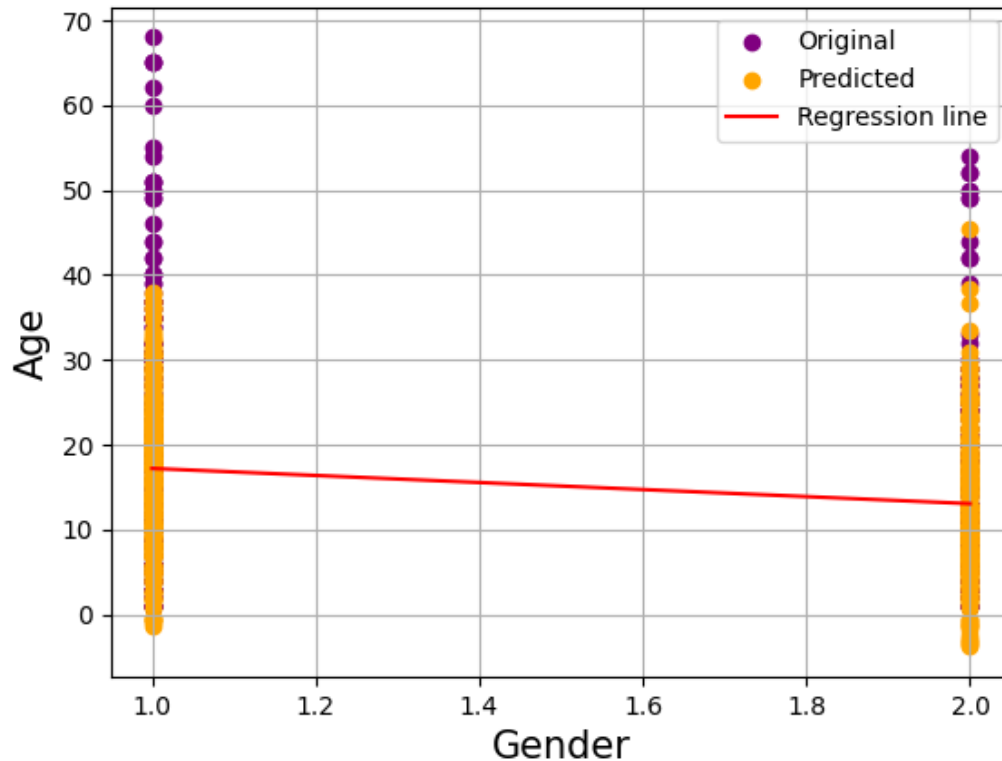
Age based on the Gender

Type of relation: indirect

Gender 1 is male / Gender 2 is female
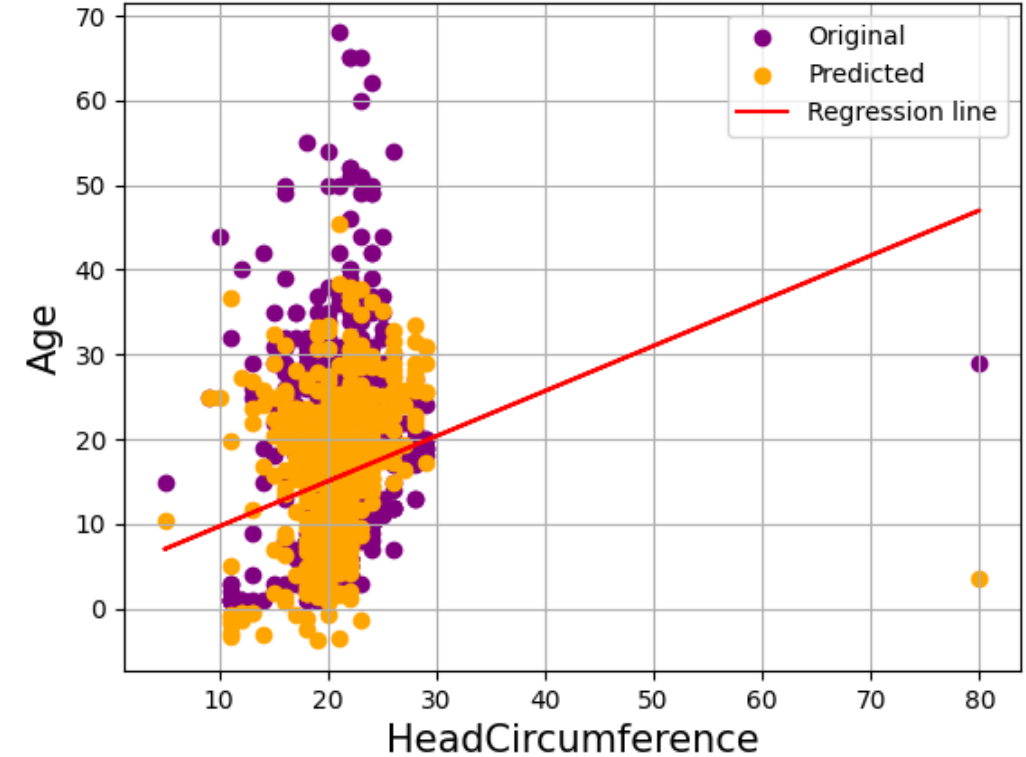
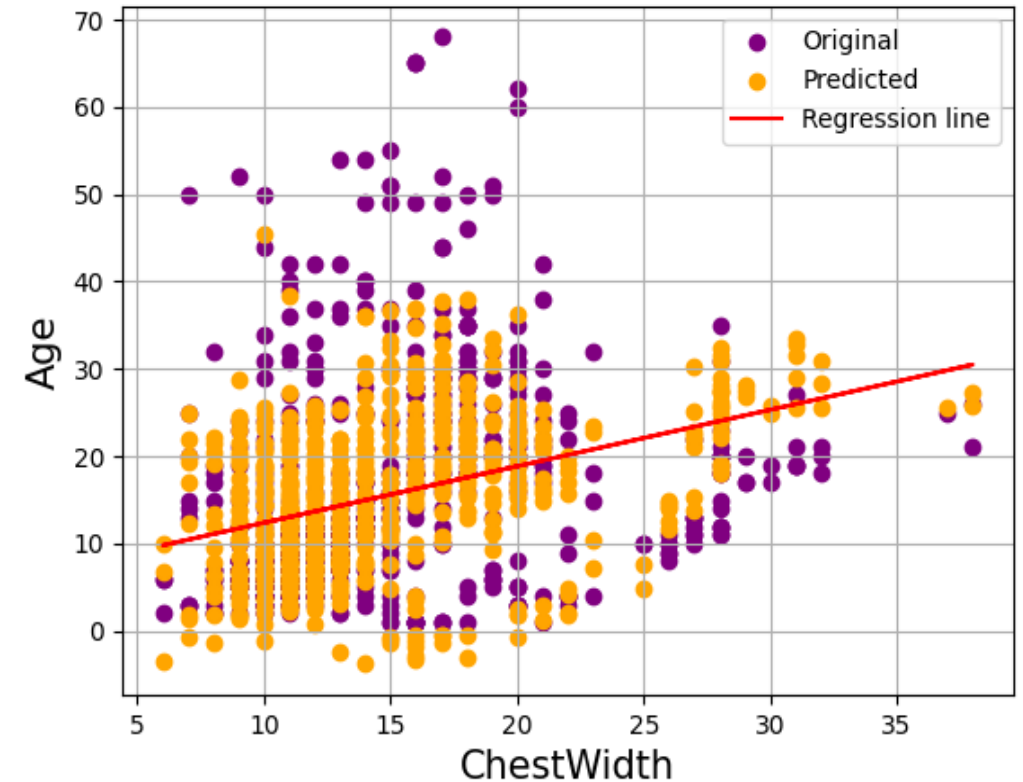**Age based on the Head circumference**

Type of relation: direct

The highest density is between 10 and 30
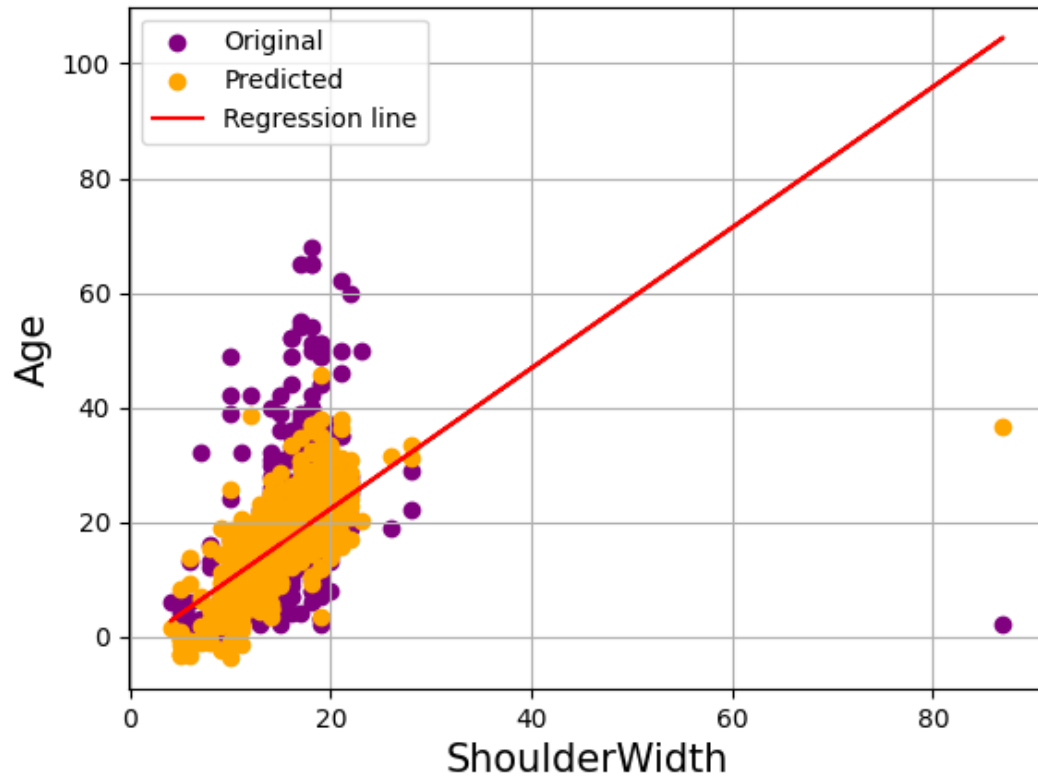
Age based on the Shoulder width
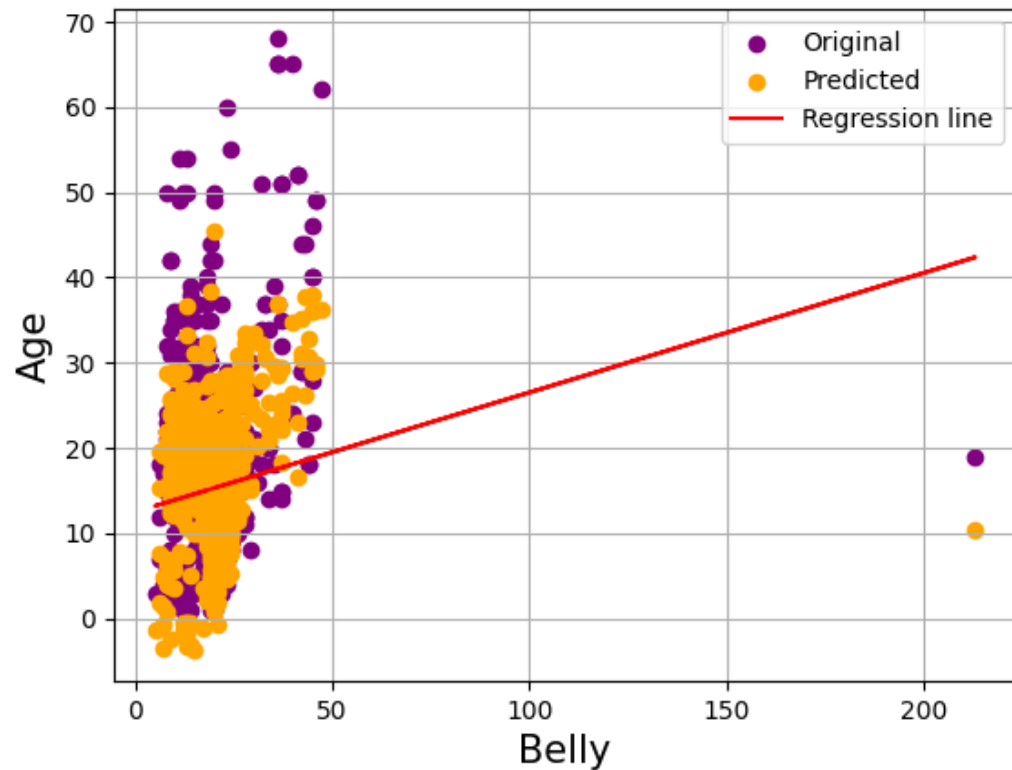
Type of relation: direct

The highest density is almost between 4 and 25

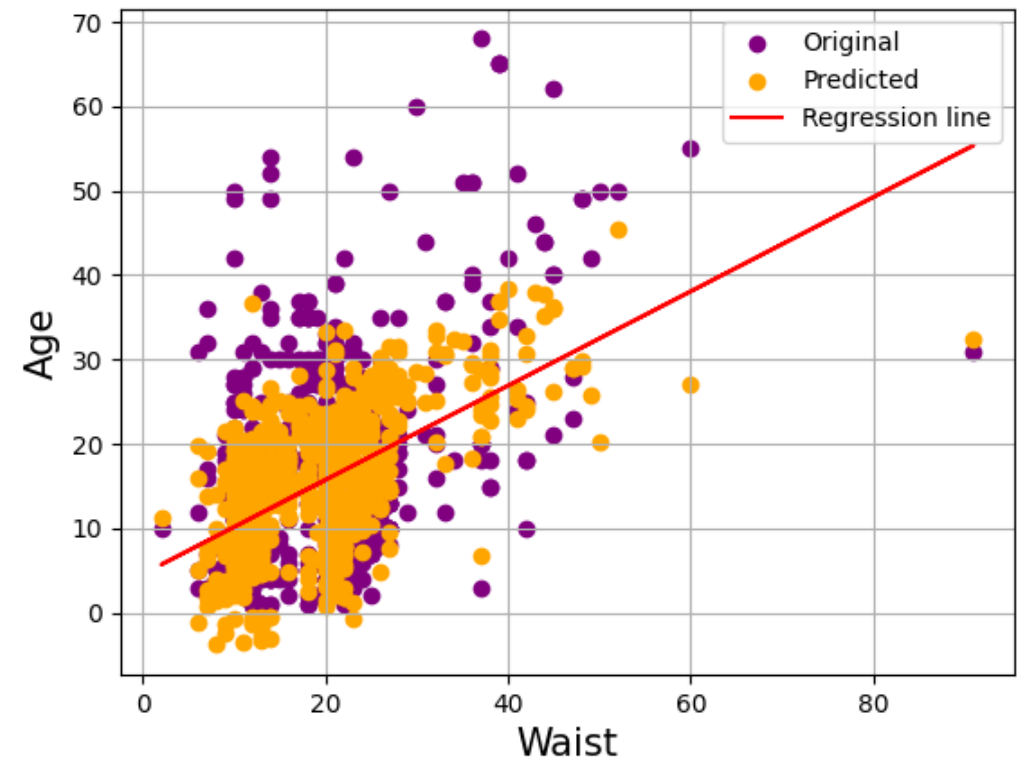**Age based on the Head Chest width**

Type of relation: direct

The highest density is between 5 and 25

(and a weaker density between 25 and 35)

In general, density is a little weak in this chart.

Age based on the Belly

Type of relation: direct

The highest density is between 0 and 50

Age based on the Waist

Type of relation: direct

The highest density is between 0 and 25

(with a weaker density between 25 and 50 (to 60))
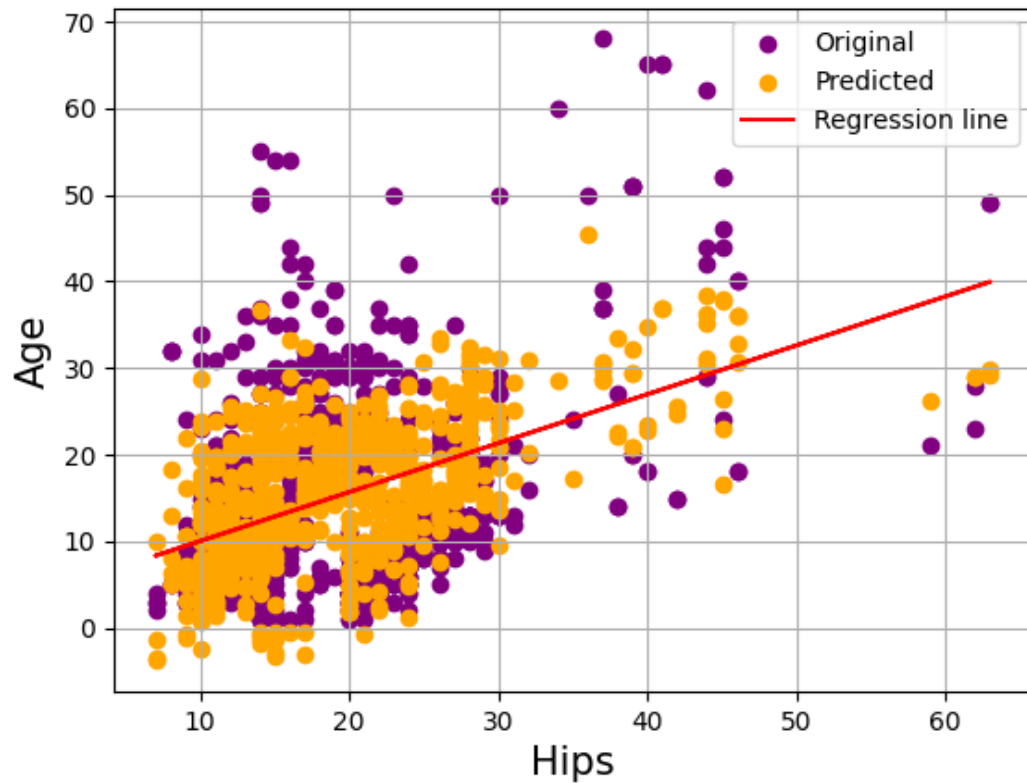
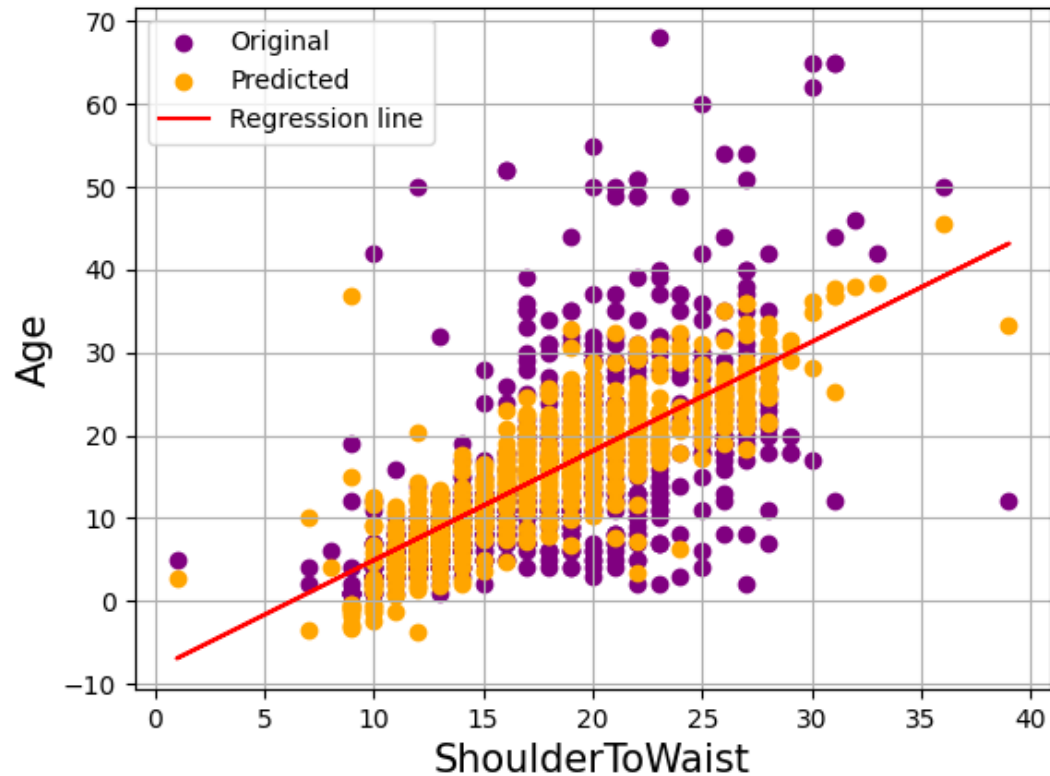**Age based on the Hips**

Type of relation: direct

The highest density is between 4 and 30 (to 33)
(there is a weaker density between ~35 to ~57)
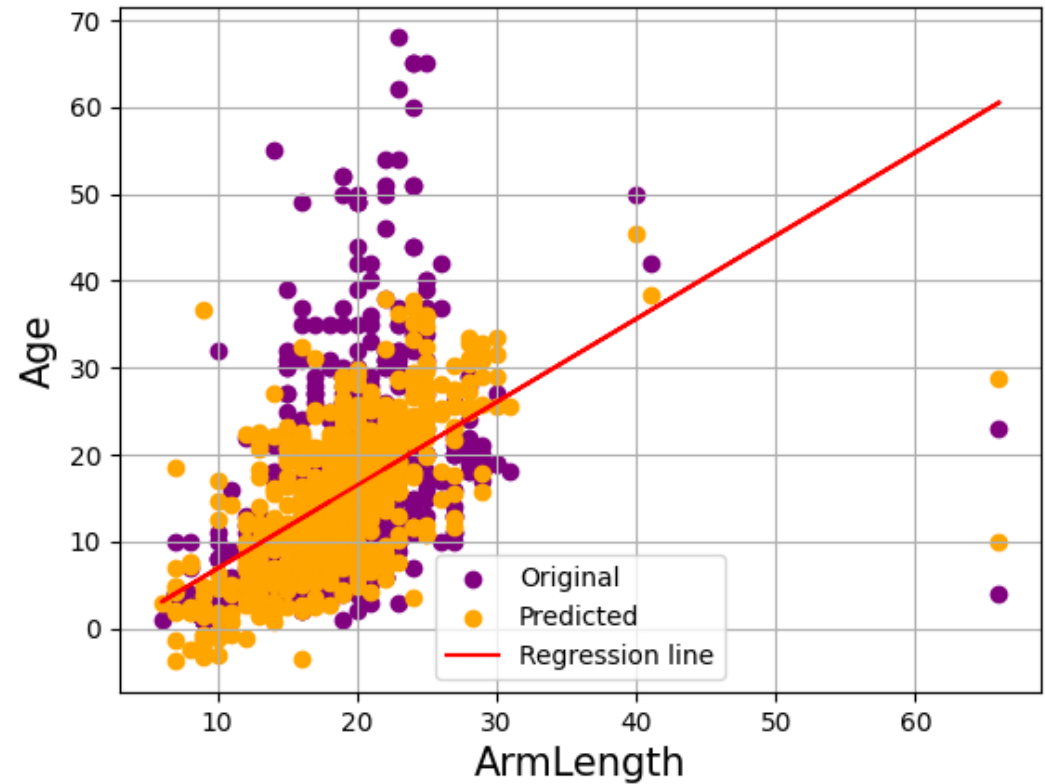
Age based on the Arm length
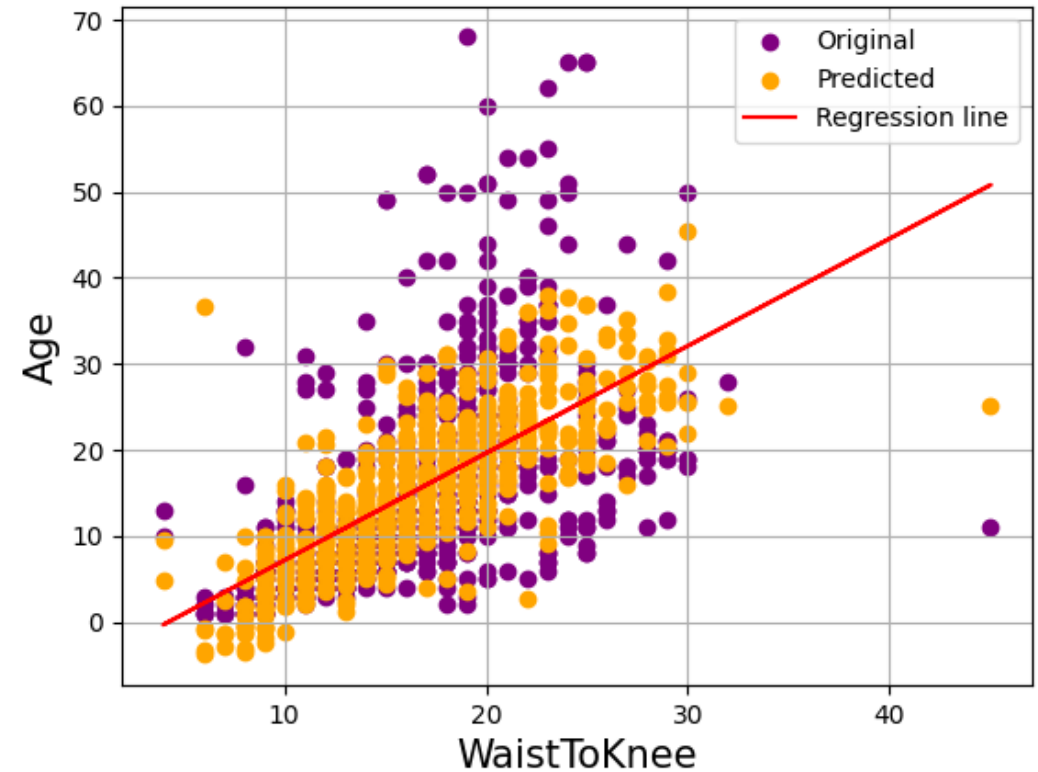
Type of relation: direct

The highest density is between 4 and 30 (to 32)

Age based on the Shoulder to waist

Type of relation: direct
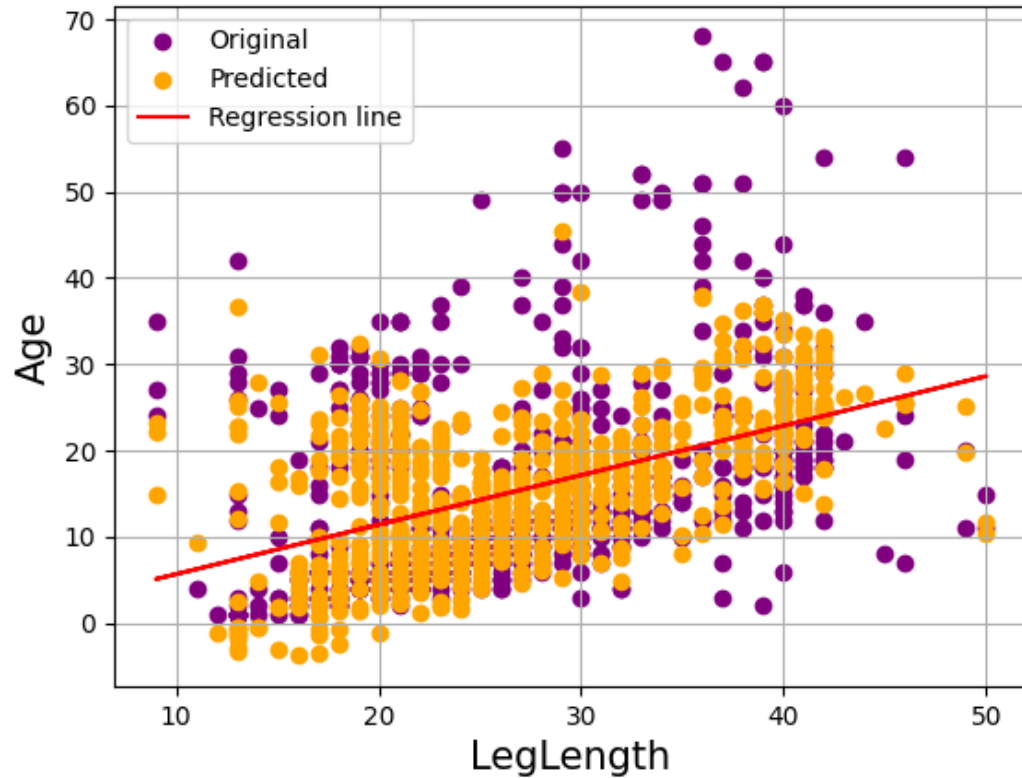
The highest density is between 5 (~6) and 30 (~28)

Age based on the Head Waist to knee

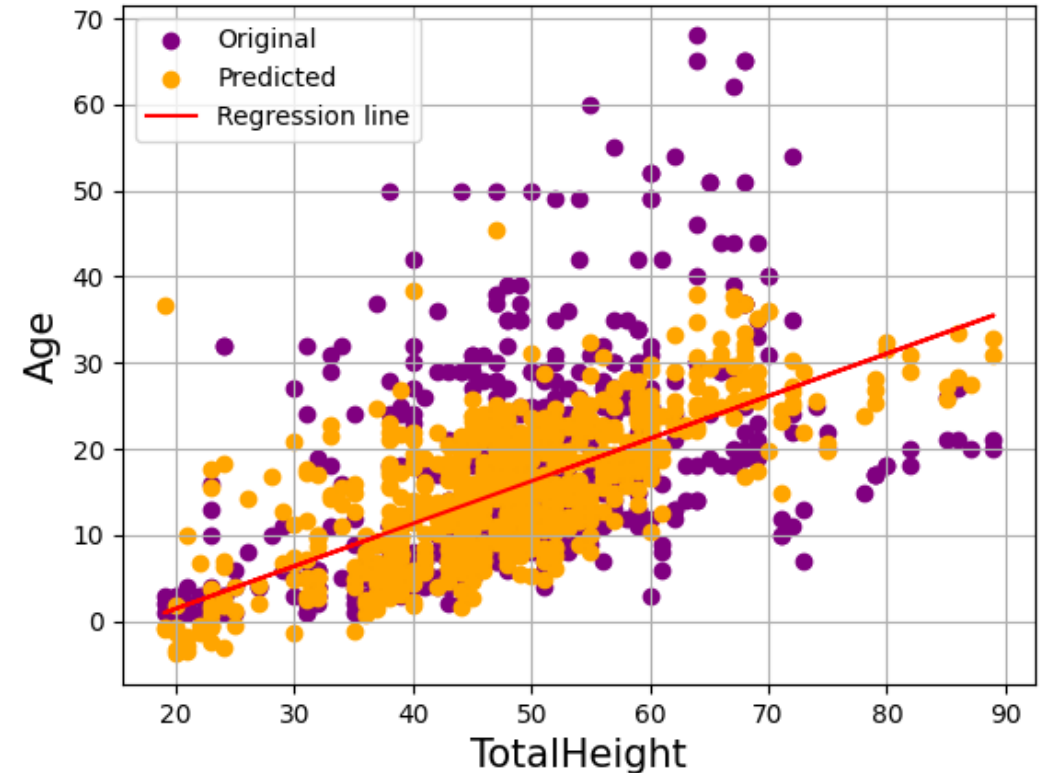Type of relation: direct

The highest density is between 4 and 30 (~31)

Age based on the Leg length

Type of relation: indirect

The highest density is between 10(~11) and 43

(with very weaker density to 50)

Age based on the Total height

Type of relation: direct

The highest density is almost between 15 and 75

(with very weaker density to 90)

The End

خدایا چنان کن سرانجام کار   تو خشنود باشی و ما رستگار

By theMHD