COMPUTING SUBJECT:	Machine Learning
TYPE:	WORK ASSIGNMENT
IDENTIFICATION:	Regression Performance
COPYRIGHT:	*Michael Claudius*
DEGREE OF DIFFICULTY:	Easy
TIME CONSUMPTION:	1 hours
EXTENT:	< 60 lines
OBJECTIVE:	Basic understanding of RMSE regression

COMMANDS:

IDENTIFICATION: Regression Performance/MICL

The Mission

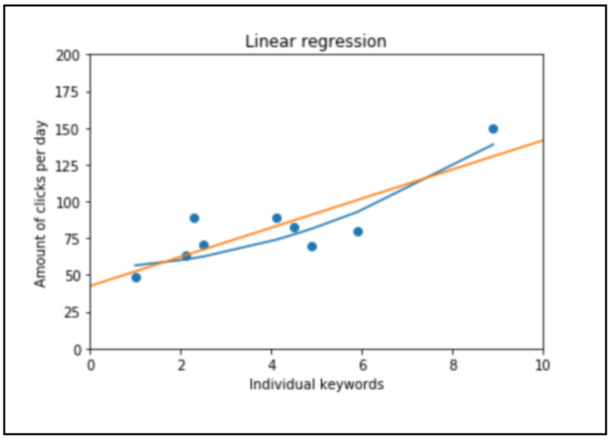
To understand the idea behind linear regression and Root Square Mean Error (RSME).
The context is limited to one variable, y, depending on the independent variable, x,

Precondition

You must have done the exercise on Linear Regression.
The first 4 questions are similar to previous assignments on linear regression so a copy and paste is just fine!.

The problem

Given a data list with values for y, and another data list with corresponding values for, x, you are to investigate the performance of linear regression: y = b*x + a, as well as polynomial regression: $y = A*x^2 + B*x + C$. As an example we will use the data given in Appendix A and end up with



As performance measure for the regression, we use the Root Mean Square Error (RMSE):

Equation 2-1. Root Mean Square Error (RMSE)

$$\mathrm{RMSE}(\mathbf{X}, h) = \sqrt{\frac{1}{m}\sum_{i=1}^{m}\left(h\left(\mathbf{x}^{(i)}\right) - y^{(i)}\right)^2}$$

Maybe Maybe Not Useful links

https://en.wikipedia.org/wiki/Root-mean-square_deviation
https://www.statisticshowto.datasciencecentral.com/rmse/

https://matplotlib.org/3.1.0/tutorials/introductory/pyplot.html

Assignment 1: Math behind Root Mean Square Error

Read the 1.5 pages (p. 39-41) in "Aurélien Géron Hands-on Machine Learning" Chapter 2 about "Performance measure".
Discuss the formula for calculating RMSE:

Equation 2-1. Root Mean Square Error (RMSE)

$$\mathrm{RMSE}(\mathbf{X}, h) = \sqrt{\frac{1}{m}\sum_{i=1}^{m}\left(h\left(\mathbf{x}^{(i)}\right) - y^{(i)}\right)^2}$$

Before the serious calculations, we will play a little and try to guess the correct linear regression values.

Assignment 2: Application program, define data and hypothesis
Start Jupyter and create a new file, *RegressionPerformance.*
First, import libraries numpy, pandas and matplotlib.pyplot and math.
In second cell, declare two lists *x* & *y* of same length

```
#Cost per click of individual keywords
x = [1.0, 2.1, 2.3, 2.5, 4.1, 4.5, 4.9, 5.9, 8.9]

#Total amount of clicks per day
y = [48.2, 63.0, 89.0, 71.0, 89.0, 82.2, 70.0, 80.0, 150.0]
```

In next cell declare two global values for slope and intersection:

```
b = 12  # try later 8  9  9.8
a = 50  # try later 50 40 44.5
```

and the hypothesis function, h:

```
def h(x):
   return b*x + a
```

Try to call and print h(2).

Assignment 3: Application plot of data and line

Use the plot library and plot the diagram and the data points like you have done before.

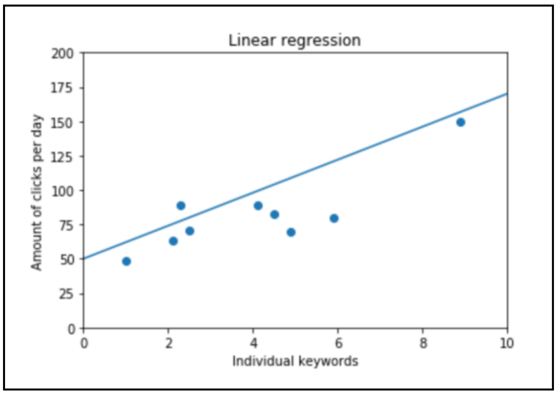
```
plt.axis([0, 10, 0, 200])
plt.scatter(x, y)
```

Then, use plt.title, plt.xlabel and plt.ylabel to apply text according to the plot on page 2.

BUT this time utilize the hypothesis function, h, to plot the regression line:

```
regression_line = [h(item)for item in [0, 10]]
```

and hopefully you see:



Try to change the values of a and b and run the code again.
Can you manually find a line that fits better by the look.

Now we move on to the RMSE

Assignment 4: Application program, the data

We are still using the data:

```
x: Cost per click of individual keywords
x = [1.0, 2.1, 2.3, 2.5, 4.1, 4.5, 4.9, 5.9, 8.9]

y: Total amount of clicks per day
y = [48.2, 63.0, 89.0, 71.0, 89.0, 82.2, 70.0, 80.0, 150.0]
```

Assignment 5: Function sum of squares

Look at the formula and the inner part. First declare a function, Sum_Of_Squares, to calculate and return the sum of the squares: $(h(x) - y)^2$ of elements in in two lists:

```
def Sum_Of_Squares(x, y, hFunc):
. . . . .
    dif = hFunc(numX) - numY
    xy2.append(dif**2)
......
```

Make the rest yourself….

Call the function with h as parameter:

```
result = Sum_Of_Squares(x, y, h)
```

and print the value.
Tip: Similar to xySum_Prod from previous assignment.

Assignment 6: RMSE function
Declare a function for calculating and returning the value of RMSE.
You just need to utilize *Sum_Of_Squares*, a square root and division by the number of data points:

```
def RMSE(x, y, hFunc):
. . . . . .
```

Print out the error for different values of a and b.

Afterthoughts
Probably you already used your previous program to find the best fit !
BUT is linear regression best ?
Let´s investigate polynomial regression of degree 2.

Assignment 7: Polynomial regression
We shall investigate a polynomial regression for the data set.
Thus the hypothesis function is:

$$A*x^2 + B*x + C$$

Instead of

$$b*x + a$$

Step back to definition cell for h (second cell).
Declare 3 variables A, B, C with values 2.0, 1.0, 60.0
And change the h(x)-function to return:

```
def h(x):
   # return b*x + a
   return A*x**2 + B*x + C
```

Run!
Can you find some values giving a lower RMSE than the linear regression ?

Assignment 8: Discussion in the class
So what is best linear or polynomial regression ?
Can we conclude? Ready to launch ? What shall we do ?

This ends your own mathematical programming, hopefully you got an idea of regression and understand some of the libraries to be used.

Appendix A

x: Cost per click of individual keywords
x = [1.0, 2.1, 2.3, 2.5, 4.1, 4.5, 4.9, 5.9, 8.9]

y: Total amount of clicks per day
y = [48.2, 63.0, 89.0, 71.0, 89.0, 82.2, 70.0, 80.0, 150.0]