index

January 23, 2022

1 Simple Linear Regression - Lab

1.1 Introduction

In this lab, you'll get some hand-on practice developing a simple linear regression model. You'll also use your model to make a prediction about new data!

1.2 Objectives

You will be able to:

- Perform a linear regression using self-constructed functions
- Interpret the parameters of a simple linear regression model in relation to what they signify for specific data

1.3 Let's get started

The best-fit line's slope $\hat{m}$ can be calculated as:

$$\hat{m} = \rho \frac{S_Y}{S_X}$$

With ρ being the correlation coefficient and S_Y and S_X being the standard deviation of x and y, respectively. It can be shown that this is also equal to:

$$\hat{m} = \frac{\overline{x} * \overline{y} - \overline{xy}}{(\overline{x})^2 - \overline{x^2}}$$

You'll use the latter formula in this lab. First, break down the formula into its parts. To do this, you'll import the required libraries and define some data points to work with. Next, you'll use some pre-created toy data in NumPy arrays. Let's do this for you to give you a head start.

[ ]:

```
# import necessary libraries

import numpy as np
import matplotlib.pyplot as plt
from matplotlib import style
style.use('ggplot')
%matplotlib inline
```

```
# Initialize arrays X and Y with given values
# X = Independent Variable
X = np.array([1,2,3,4,5,6,8,8,9,10], dtype=np.float64)
# Y = Dependent Variable
Y = np.array([7,7,8,9,9,10,10,11,11,12], dtype=np.float64)
```

1.4 Create a scatter plot of X and Y and comment on the output

[ ]:

```
# Scatter plot
```

[ ]:

```
# Your observations about the relationship between X and Y




#
```

1.5 Write a function `calc_slope()`

Write a function `calc_slope()` that takes in X and Y and calculates the slope using the formula shown above.

[ ]:

```
# Write the function to calculate slope as:
# (mean(x) * mean(y) - mean(x*y)) / ( mean (x)^2 - mean( x^2))
def calc_slope(xs,ys):

    pass

calc_slope(X,Y)

# 0.5393518518518512
```

Great, so we have our slope. Next we calculate the intercept.

As a reminder, the calculation for the best-fit line's y-intercept is:

$$\hat{c} = \overline{y} - \hat{m}\overline{x}$$

1.6 Write a function best_fit()

Write a function `best_fit()` that takes in X and Y, calculates the slope and intercept using the formula. The function should return slope and intercept values.

[ ]:

```
# use the slope function with intercept formula to return calculate slope and␣
 ↪intercept from data points

def best_fit(xs,ys):

    pass
```

```

# Uncomment below to test your function

#m, c = best_fit(X,Y)
#m, c

# (0.5393518518518512, 6.379629629629633)
```

We now have a working model with `m` and `c` as model parameters. We can create a line for the data points using the calculated slope and intercept:

- Recall that $y = mx + c$. We can now use slope and intercept values along with X data points (features) to calculate the Y data points (labels) of the regression line.

1.7 Write a function reg_line()

Write a function `reg_line()` that takes in slope, intercept and X vector and calculates the regression line using $y = mx + c$ for each point in X

[ ]:

```
def reg_line (m, c, xs):

    pass

# Uncomment below
#regression_line = reg_line(m,c,X)
```

1.8 Plot the (x,y) data points and draw the calculated regression line for visual inspection

[ ]:

```
# Plot data and regression line
```

So there we have it, our least squares regression line. This is the best fit line and does describe the data pretty well (still not perfect though).

1.9 Describe your Model Mathematically and in Words

[ ]:

```
# Your answer here
```

1.10 Predicting new data

So, how might you go about actually making a prediction based on this model you just made?

Now that we have a working model with m and b as model parameters. We can fill in a value of x with these parameters to identify a corresponding value of $\hat{y}$ according to our model. Recall the formula:

$$\hat{y} = \hat{m}x + \hat{c}$$

Let's try to find a y prediction for a new value of $x = 7$, and plot the new prediction with existing data

[ ]:

```
x_new = 7
y_predicted = None
y_predicted

# 10.155092592592592
```

1.11 Plot the prediction with the rest of the data

[ ]:

```
# Plot as above and show the predicted value
```

You now know how to create your own models, which is great! Next, you'll find out how to determine the accuracy of your model!

1.12 Summary

In this lesson, you learned how to perform linear regression for data that are linearly related. You first calculated the slope and intercept parameters of the regression line that best fit the data. You then used the regression line parameters to predict the value ($\hat{y}$-value) of a previously unseen feature (x-value).