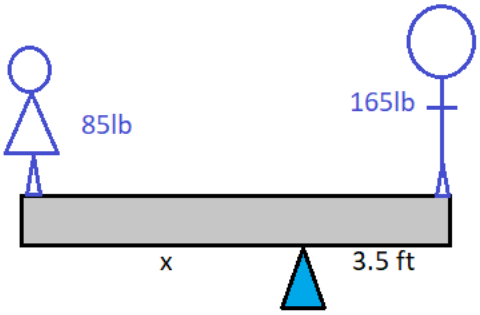
Week 1: Teeter-Totter ICA
CS 429/529

Simple ML example

Let's stick with the teeter totter problem:



Best Solution so far:

The model we used to generalize the experiences that were recorded (in the dataset of X=weight and Y=distance) for the distance where people should sit from the fulcrum given their weight.

The best performing linear regression model defined as $x = a_1 w_1 + a_0 Y$ where x is the distance from fulcrum w_1 is the mass, has parameters $a_1 = -0.1$ and $a_0 = 20$. The error between the actual distances and what model predicted was E= 559.17.

Now, let's use this model. Two new people come to the park: John and Marry. John is 179 lb and Mary is 134 lb. Use the model to determine where on the teeter totter they should sit:

John:

Mary:

Can we have a better model?:

Let's see: we have a model definition, and we have the model's parameters. Any given model instance (definition + parameters) produced Y_hat outputs that do not 'replicate' the original experiences, but generalize the relationship between the inputs (weight) and the output (distance). The error is then defined as a mean square error between the measured/actual distances (Y) and the distances that model output. RMSE = $|Y\text{-}Y\_hat|^2$.

To better understand the relationship between the model parameters and the error, let's first create a 3D scatter plot between the model parameters (axis x = a_0, y=a_1) and the error (axis z=e).

Code along to make the 3D scatter plot using the data we have so far:

a_0, a_1, e

0.1,3,2373.77

0.2,13,19255.73

-0.1,20,559.17

Copy and paste the above data into data.csv file in your project directory

```
from mpl_toolkits import mplot3d

import matplotlib.pyplot as plt

import numpy as np


fig = plt.figure()

ax = plt.axes(projection ='3d')
```

Preconditions: before you can run the code, please install matplotlib for python on your VM. “pip install matplotlib” Unfortunately, this will be installed on your local profile, so the next time you log in you will have to reinstall it.

Copy and paste the starter code into plot.py file in your project directory

a) Make a new guess for the model parameters that will produce lower error. Don’t just shoot into the dark, look at the plot of which model parameters and error, then make a guess what parameter values produce lower error.

b) Plug the parameters to Excel, calculate the resulting error, add the new parameters and error as a new line to the data.txt file.

c) Replot the data.

d) Repeat a-c. Are you getting closer?

Questions:

- What is Josh who weights 65lb would want to sit on the teeter-totter? Will the model produce ‘valid’ answer? In other words, would it be fair to rely on the model to accurately predict the distance from the fulcrum?
- How were you making your ‘educated’ guesses?
- What was the pattern in the scatter plot of parameters vs. error that you followed to select new parameter values to test?

•