Jericho McLeod
Chapter 3 Review

The chapter was again primarily review until the ‘more expressive rules’ section. At this point the author was rather exhaustive in explaining how a simple rule based on an input variable was insufficient to explain whether an instance was ‘standing’ or ‘lying’, but didn’t offer what I found to be a relatively simple way to explain the relationship between the input variables as being the deciding factor; by visualizing it. I wrote a short snippet of python to do so, shown below.

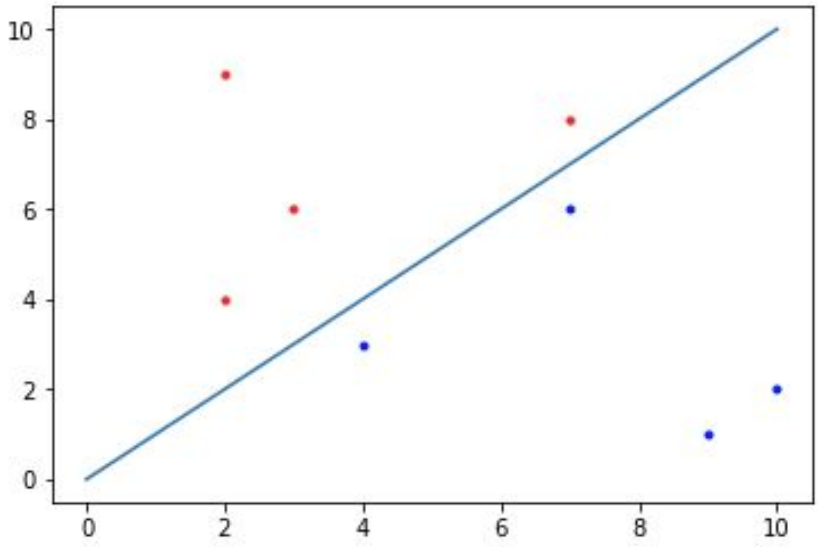
```
import matplotlib.pyplot as plt
import numpy as np

data = [[2,4,4,'Standing'],\
        [3,6,4,'Standing'],\
        [4,3,4,'Lying'],\
        [7,8,3,'Standing'],\
        [7,6,3,'Lying'],\
        [2,9,4,'Standing'],\
        [9,1,4,'Lying'],\
        [10,2,3,'Lying']]

for obs in data:
    if obs[3] == 'Standing':
        tick = 'r.'
    if obs[3] == 'Lying':
        tick = 'b.'
    plt.plot(obs[0],obs[1],tick)

x = np.linspace(0,10,1000)
plt.plot(x,x)
plt.show()
```

The visualization obtained from this is, in my opinion, straightforward. Blue = lying, red = standing, the light blue line is f(x) = x, width is taken as the X value, and height is the Y value in this visualization.



This can translate into measuring parts of a system, such as the tower described in the text, with no additional issues, and in my opinion offers a better way to communicate the point than the set of rules describing whether shapes are standing or lying down. I must admit, however, that context is important. Given both the text and my visualization, I appreciate the visualization, but either absent the other means that the text immediately tells you what is being measured, and how, without any additional data. My visualization alone is not contextualized like descriptive rules.

Regarding instance based representation, the author cautions against generalizing nominal data. This has been something of an area of focus of mine for the past few months, as I've had to do this with text. I have found that creating a histogram of all words in a corpus, then selecting the top *n* words, then creating vectors of *n* length for each document with a binary value showing whether each word in the original vector is present or absent is sufficient to measure similarity between documents. It does create an n-dimensional space that can be computationally expensive, but working with small datasets of less than 250k observations it is still achievable using off the shelf components, and allows for generalizations to be made on data that is otherwise nominal.

The next section being on the topic of clusters is a bit amusing, as that is part of what I was trying to do in text processing, and the similarity measures are what I used to create clusters. My algorithm was hierarchical, so that trimming from the top could be used to obtain more and more specific clusters, and also for ease of calculating. It was a great way to explore a massive amount of text, and to segregate things into categories so that topics could be extracted that were not dependent on overall presence in the corpus, but instead may be well represented only within a subset of the data.