Machine Learning Notes

January 21, 2023

P.McDowell

Chapter 1 - Data Pre-Processing

In this section we will look at some basic data analysis tools, namely basic statistics. It is understood that for many this is a review. Initially the discussion will be focused on the calculation of basic measures of statistics. Following that we will use those measures to characterize data and find outliers.

We will take the time to go over some concepts, relate those concepts to equations, and finally hammer out some code to implement the ideas. Many of the examples in this section use the python interpreter.

As mentioned above, the python interpreter is often used. There are many nice IDEs for python; the ones used here are usually the Spyder IDE from Anaconda, IDLE from standard python download, python from jdoodle, or python from repl. The first two require downloading and installation, but Spyder in particular provides some nice tools that can be used in more complex endeavors, IDLE is nice because it works with some libraries that can fussy with Spyder, and the last two are nice because they require no installation, and in some cases offer program storage online with the user's account.

1.1 Average, Variance, Standard Deviation, Median, Mode

Creating an array called x and giving some values.
```
x = [1, 3, 5, 7, 9]
```

Mean or Average

The average of the data set x is found by summing the individual numbers and dividing them by total number of numbers.

$$\mu = \sum x_i / n$$

Summing the values of x:
```
mySum = sum(x)
```

If we type the name of the variable the interpreter will output the value:
```
mySum

Out[12]: 25
```

Finding the average is easy too:
```
av = mySum/len(x)

av
Out[14]: 5.0
```

This is all very straightforward, however the average of a data set does not tell us how the data is distributed. For example, look at the next data set.

xStretch = [-5, 0, 5, 10, 15]

This data set has the same average as the previous one, x, but they are obviously different. The data in xStretch is on average further from the mean value of the data set.

av = sum(xStretch)/len(xStretch)

av
Out[17]: 5.0

Variance and Standard Deviation

The statistical measures that indicate how spread out a data set is are called variance and standard deviation. They reflect the average of the squared distance from the center of data set, and average distance from the center of the data set. The square is used so that negative distances from the average do not cancel out positive distances from the average. Looking at the two data sets:

x = [1, 3, 5, 7, 9]

and

xStretch = [-5, 0, 5, 10, 15]

It is easy to see that x is more closely centered about the mean than xStretch by looking at the sorted data point to data point differences. For the x data set the difference is 2, for xStretch the difference is 5, but they are both centered around 5, giving them the same average. The formula for variance reflects this.

$$\sigma^2 = \sum(x_i - \mu)^2/n$$

Standard deviation is the square root of the variance.

$$\sigma = \sqrt{\sum(x_i - \mu)^2/n}$$

For some the standard deviation is a bit more intuitive than variance because it is easier to visualize the average displacement from the mean of the data than the square of that value. Another consideration is that outliers have a larger influence on the results because of the square of the distance from the central point. If there are several outliers, this effect can skew results.

Median and Mode

The median value of a data set is the one that falls in the middle of the data, after it is sorted. If the data set has an odd number of items the median is middle item, if there is even number of items, the median value is average of the two middle items.

biff = [12, 3, 8, 2, 9, 17, 21]

We need to sort these. Python has some niceties.

biff.sort()

[2, 3, 8, 9, 12, 17, 21]

Now we can find the median value by counting the data items. 9 is our median value.

We can see that the median and the average are slightly different.

sum(biff)/len(biff)

Out[26]: 10.285714285714286

If the data set had an even number of values, we would have needed to take the average of the two centrally located values.

1.2 Descriptive Measures of Multi-Dimensional Data

If the data has more than one 1-dimension we still use most of the tools described in the previous section.

To find the central point of a data set with multiple coordinates, we take the average of each of the coordinates. For example if we a data set of (x, y) points, the central point of the data set is:

$(x_c, y_c) = (\sum x_i / n , \sum y_i / n)$

Variance and Standard Deviation in this case measure how tightly the points are clustered about the central point. Our formulas from the earlier section are not quite what we need because they operate in a single dimension. However, they are close to what is needed; we can still use them by calculating the distance of each point from the central point of the data set and then taking the variance and standard deviation using the conventional equations.

To find how tightly grouped points are about the centroid (the middle of the data) we:

- Find the centroid by finding the mean value of each component of the data. If it the data is composed of x-y pairs, we use the formula from above.
- Once we have found the centroid we use the distance formula to generate a set of distances from every point in the data set to the centroid.
- We then find the variance and standard deviation of this data.

There are other methods of doing this – below is an example from Stack Exchange:

From the calculations I did, this is what I came up with:

$$\sigma xy = \sqrt{(SxxSyy - (Sxy)(Sxy)}$$

In this case, Sxx and Syy are the variances of x and of y respectively, whereas Sxy is kinda like the mixed variance of x and y.

To elaborate, assuming there are n elements, and x_μ

represents the mean value of x and y_μ

represents the mean of y:

$$Sxx = \sum_{i=1}^{n} \frac{(x - x_\mu)^2}{n}$$

$$Syy = \sum_{i=1}^{n} \frac{(y - y_\mu)^2}{n}$$

$$Sxy = \sum_{i=1}^{n} \frac{[\![(y - y_\mu)(x - x_\mu)}{n}$$

1.3 I/O and Sample and Sample Programs

Below are some sample programs that will help with getting proficient in Python.

Generating Data from an Equation

```
def generateXYDataFromEquation(xmin, xmax):
    x = []
    y = []
    for myX in range(xmin, xmax):
        x.append(myX)
        myY = myX
        y.append(myY)

    return x, y
```

Writing x-y Data to a File:

```
def writeXYData(filename, x, y):
    out = open(filename, "w")
    for j in range(len(x)):
        myString = str(x[j])+" "+str(y[j])+" \n"
        out.write(myString)
    out.close()

    return
```

Reading x-y Data from a File:

```
def readXYData(filename):
    inFile = open(filename, "r")
    x = []
    y = []
    for myLine in inFile:
        print(myLine)
        xs, ys = myLine.split()
        x.append(int(xs))
        y.append(int(ys))
    inFile.close

    return x, y
```

Basic Plot:

```
def basicPlot(x, y, title):
    plt.plot(x, y, 'bo')
    plt.title(title)
    plt.show()

    return
```

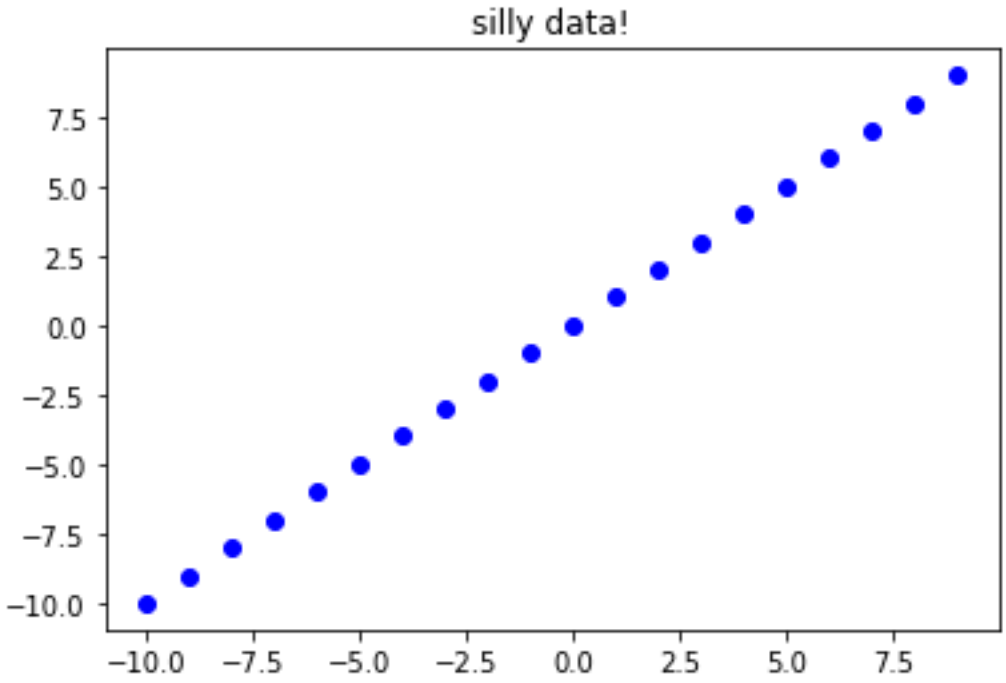
For the plot to work, the plot library must be imported:

```
import matplotlib.pyplot as plt
```

Putting it all together:

```
# Testing the functions!
x, y = generateXYDataFromEquation(-10, 10)
writeXYData("mySillyData.txt", x, y)
x, y = readXYData("mySillyData.txt")
basicPlot(x, y, "silly data!")
```

Here is the result:



1.4 Exercises

The purpose of these exercises is to get accustomed to using Python and the associated environments. Here are some links at which various Python resources are available:

- https://replit.com/languages/python3
- https://www.anaconda.com/products/distribution
- https://www.python.org/downloads/

1. Write a program that writes the values of the equation $y = ½ x^2 + 7$ to a file named "myNumbers.txt". Start the x values at -5 and go to 5, increment by 0.1
2. Make sure that the numbers are in the file. Read them and print them out programmatically.
3. Using the plot library in python plot the numbers.
4. Using Python, read the data in the file "myData.txt". Find the values of the descriptive statistics for the y values of the data. Also, find the average point to point difference. To this, write functions for the following:
    a. getMean
    b. getVariance
    c. getStd
    d. getMedian
    e. getMode
5. Plot the data in a graph.
6. Do the same for the file "myNuttyData.txt"