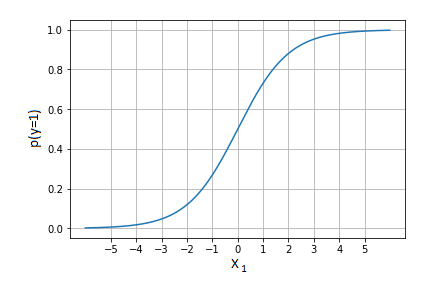
**CSCE 623 Spring 2020 - Machine Learning: In Class Work, Day 7**

**(solutions)**

From Chapter 4, part 1: Logistic Function **Thinking**



1. Thought Question [Answer this before you code]: Given the logistic function for computing a probability from an input based on 2 beta coefficients and *X*1, consider the graph to the right:

1. What happens to the shape of the graph as *β*0 increases or decreases?

The s-curve’s x=0.5 value moves to the left for an increase or to the right for a decrease

1. What happens to the shape of the graph when a non-zero *β*0 switches signs (e.g. positive to negative)?

The s-curve’s values are shifted such that the x =0.5 crossing is on the other side of the y axis by an equal distance

1. What happens to the shape of the graph as *β*1 increases or decreases?

As *β*1 increases, the effect from *X*1 is magnified, thus the S-curve is compressed width-wise and becomes steeper (greater slope). As *β*1 decreases, the curve becomes shallower (less slope)

1. What happens to the shape of the graph when a non-zero *β*1 switches signs (e.g. positive to negative)?

The slope of the s-curve flips from positive to negative

1. Suppose you wanted to make a classifier that classified the target (*y*) as class 1 when *X*1 < -3 and class 0 when *X*1 > -3. What beta values would you choose?

Select *β*0 to be 3 and *β*1 to be a large number (e.g. 1000)

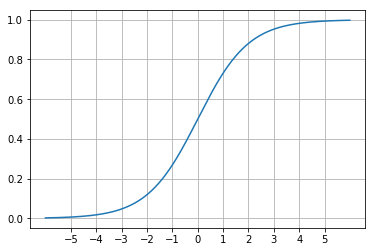
From Chapter 4, part 1: Logistic Function **Coding**

2. Python Practice: Your goal is to build a function to compute the logistic regression equation (don’t use a built-in function). You will need numpy and matplotlib.pyplot



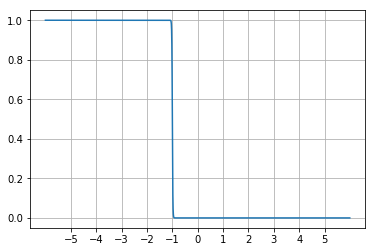
a) Define a python function to compute P(*y*=1) from an input signature (*β*0, *β*1, *X*1) where each beta is a scalar, *X*1 is a (*n* by 1) matrix and *P*(*y*=1) is a (*n* by 1) matrix

b) Write code to plot *X*1 versus *P*(y=1). Extra Feature: Design your code to ensure that most of the probability mass shows up on the plot (e.g. the part of the curve containing 0.01 < *P*(*y*=1) < 0.99). Define an input matrix *X*1 to contain many observed values of *X*1 on the needed range.

c) Try to pick betas to recreate the graph to the right, and plot the results. What values did you use for your betas?

d) Alter *β*0 as described in question 1a above. What happens? (don’t forget the axes scale may have shifted if you implemented the extra feature)

e) Reset *β*0 to its setting in 2c. Alter *β*1 as described in question 1b above. What happens?



f) Pick a set of beta’s to approximate a 1-to-0 Boolean transition centered at  
 *X*1 = -1 (when *X*1 is less than -1, *y*=1, when *X*1 is greater than -1, at *y*=0, and the  
line at *X*1 = -1 is as close as possible to vertical)