9/17: Linear and Logistic Regression

Discord: https://discord.gg/68VpV6

Zoom Poll

Prefer Discord, Email, or Both?

Popular Project Suggestions

- **Chat bot** (custom data?)
- Sentiment Analysis
- Neural Networks
- Summarizer Bot
- Language Translations/Identifier

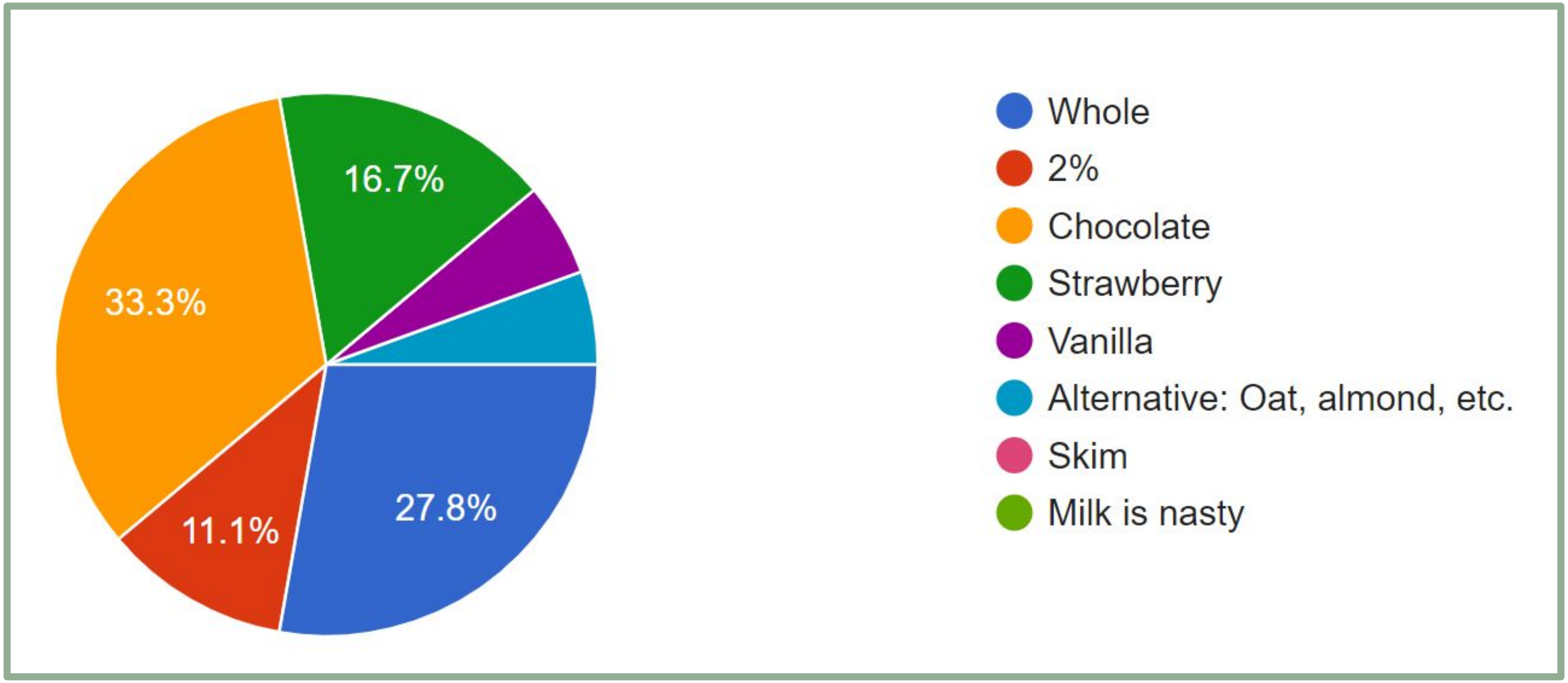
Current Agenda

- Week 2: Linear/Logistic Regression, Sentiment Analysis Preview
- Week 3: In-depth Sentiment Analysis (Tweets?)
- Week 4: Neural Networks (Handwritten Digits)
- Week 5: Chatbot Pt 1 (If ready!)

Other events

- We can definitely do professional development
- Social event ideas?
- Try to get speakers
- **Independent/Group Projects**

Some more interesting data



Linear and Logistic Regression

Two of the most fundamental algorithms for Natural Language Processing

Supervised Learning

“

Using already existing (labeled) data to predict something based off of future inputs

Linear vs Logistic Regression

Linear Regression

- Uses data to predict <u>numerical</u> outputs
- Essentially creates a line of best fit for our data (just like graphing calculators!)
- Ex. Predict a Yelp review score based on the review

Logistic Regression

- Uses data to predict <u>categorical</u> outputs
- Divides our data into a predetermined categories
- Ex. Classifying emails as Spam or Not Spam given the email

Data Management Steps

Before we can create a model, we must format our data using the following steps:

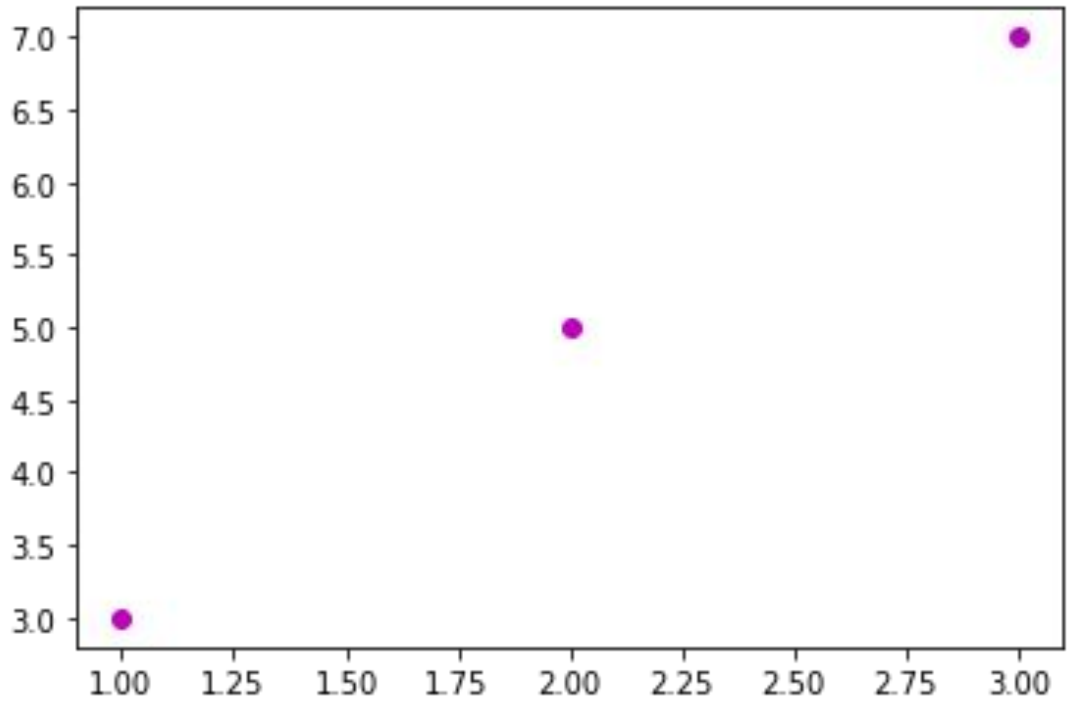
1) Categorize data into inputs (x) and outputs (y)

2) Split data into train and test sets (around 80/20)

3) Train our model using the train data, test with either the test data or our own inputs

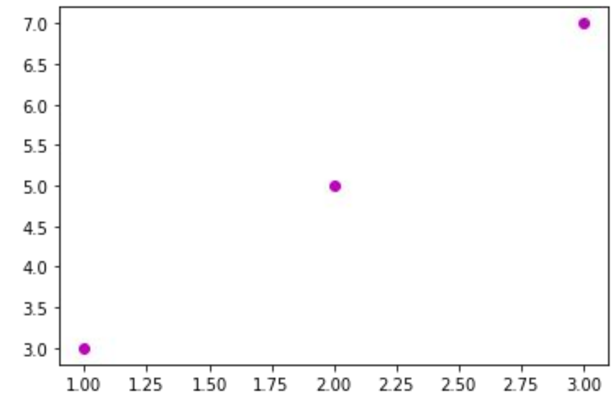
Linear Regression

Say we have the following data:



We need to find an equation that best fits this plot

Matrix Representation



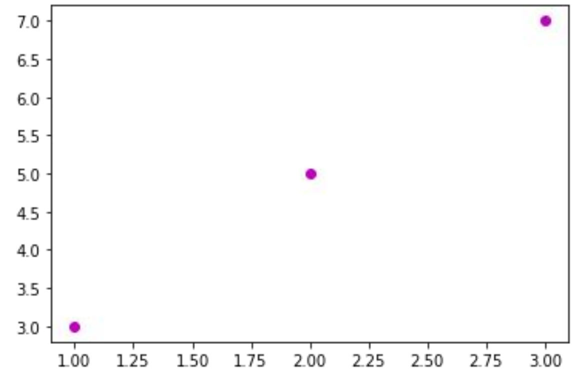
It's safe to assume this model follows:

$$y = \Theta_0 + \Theta_1 x$$

If we represent x and our thetas as matrices, we can calculate each observation's y with this equation:

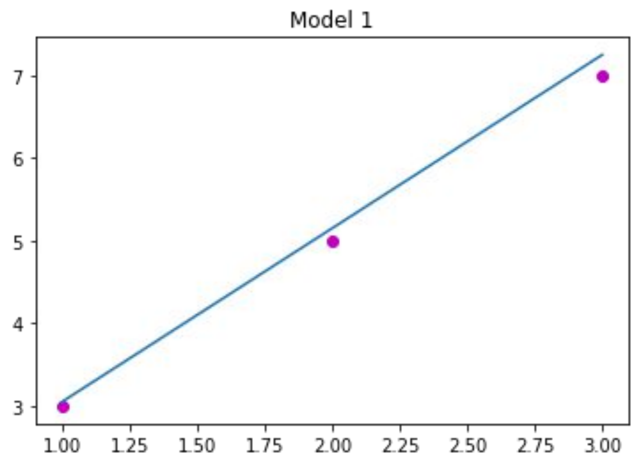
$$y = \begin{Bmatrix} y_0 \\ y_1 \\ y_2 \end{Bmatrix} = \begin{Bmatrix} 1 & x_0 \\ 1 & x_1 \\ 1 & x_2 \end{Bmatrix} \cdot \{\Theta_0 \;\; \Theta_1\}$$

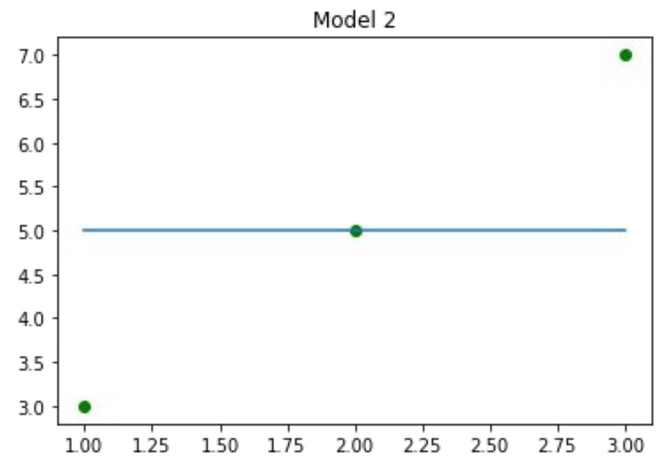
What makes a model good?



$$\Theta = \{0.95 \;\; 2.1\}$$

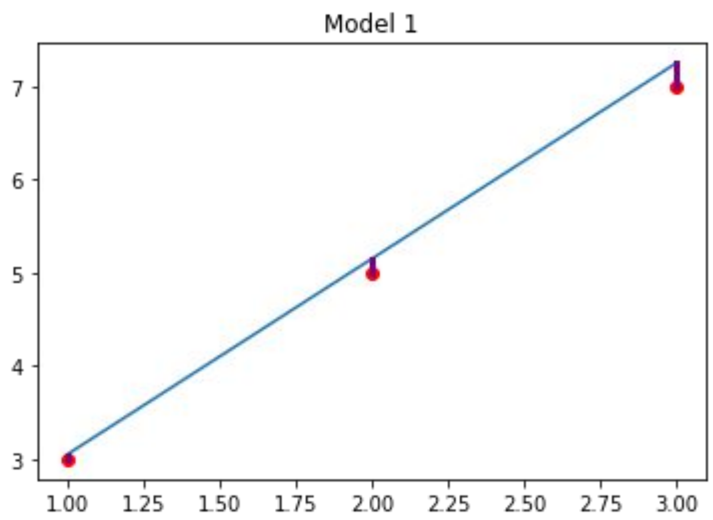
$$\Theta = \{5 \;\; 0\}$$

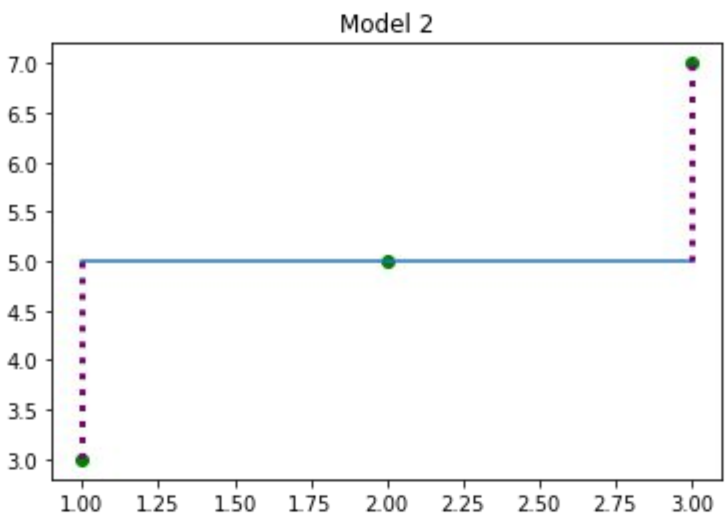




Which model is better? Why? (type in chat)

Introducing Cost





Cost: The square difference between what we expected and what we actually got

Our Goal: Minimize the cost of our model

Official Linear Regression Cost Formula

$$\frac{1}{2m}\sum_{i=1}^{m}(predicted - actual)^2$$

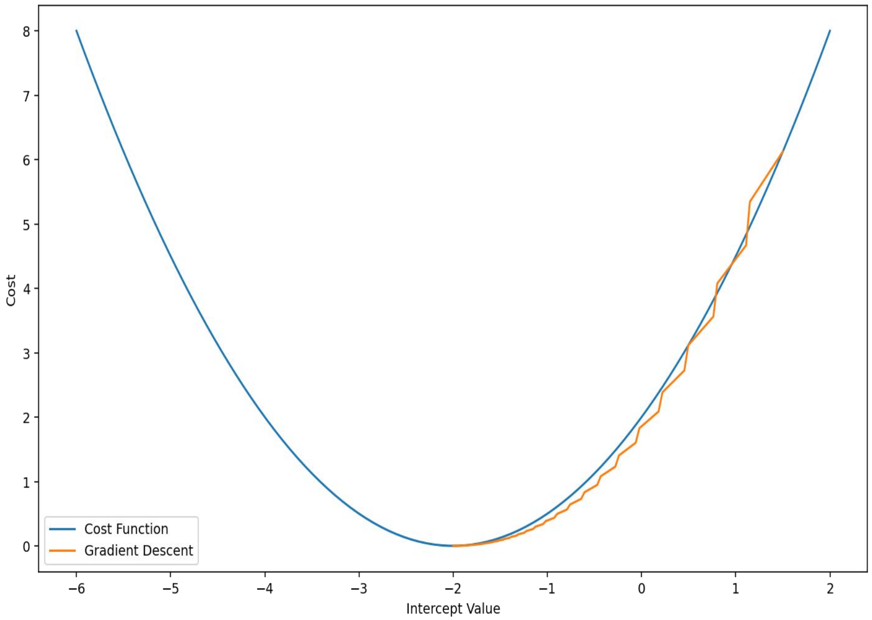
We need to minimize this value

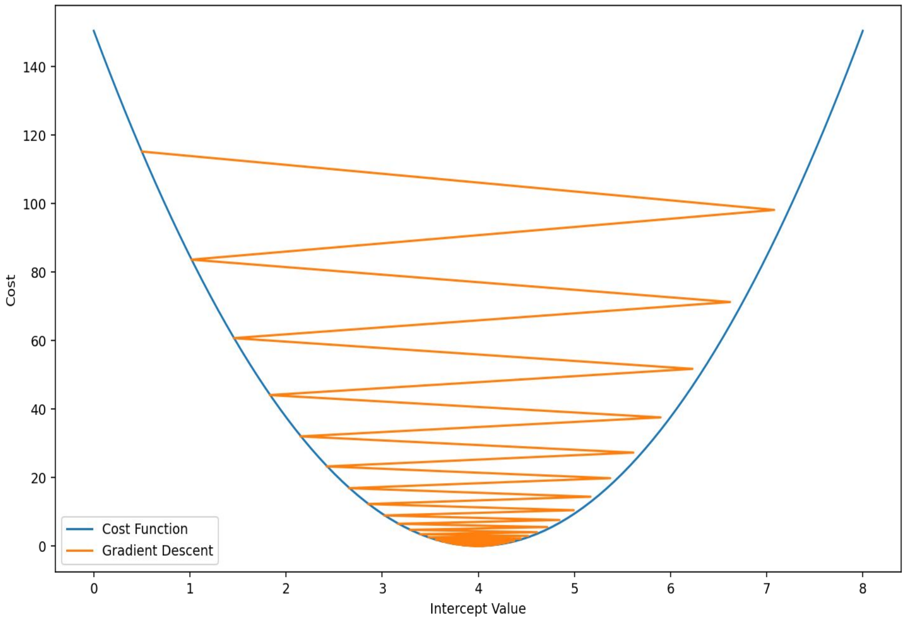
How do we normally find the minimum value of a function? (type in chat)

Introducing Gradient Descent

The derivative tells us if the function is increasing or decreasing

Since we want to reach the minimum value, we will want to travel in the opposite direction of the derivative





If we take a small step in the opposite direction, we will slowly travel down the cost curve until we reach the minimum cost

Official Derivative Formula

$$\frac{1}{m} * [X^T \cdot (predicted - actual)]$$

We multiply this value by a learning rate (around 0.001), to take a small step

Then, we keep subtracting the learning rate times the gradient from theta until we reach the minimum cost

General Algorithm

1) Predict a value for y from our thetas

2) Calculate the derivative of our cost function with that value of y

3) Take a small step in the direction of the negative derivative and adjust our thetas

4) Repeat steps 1 - 3 until we reach a minimum cost

Logistic Regression

What do we have to do if we need to split our data into two categories?

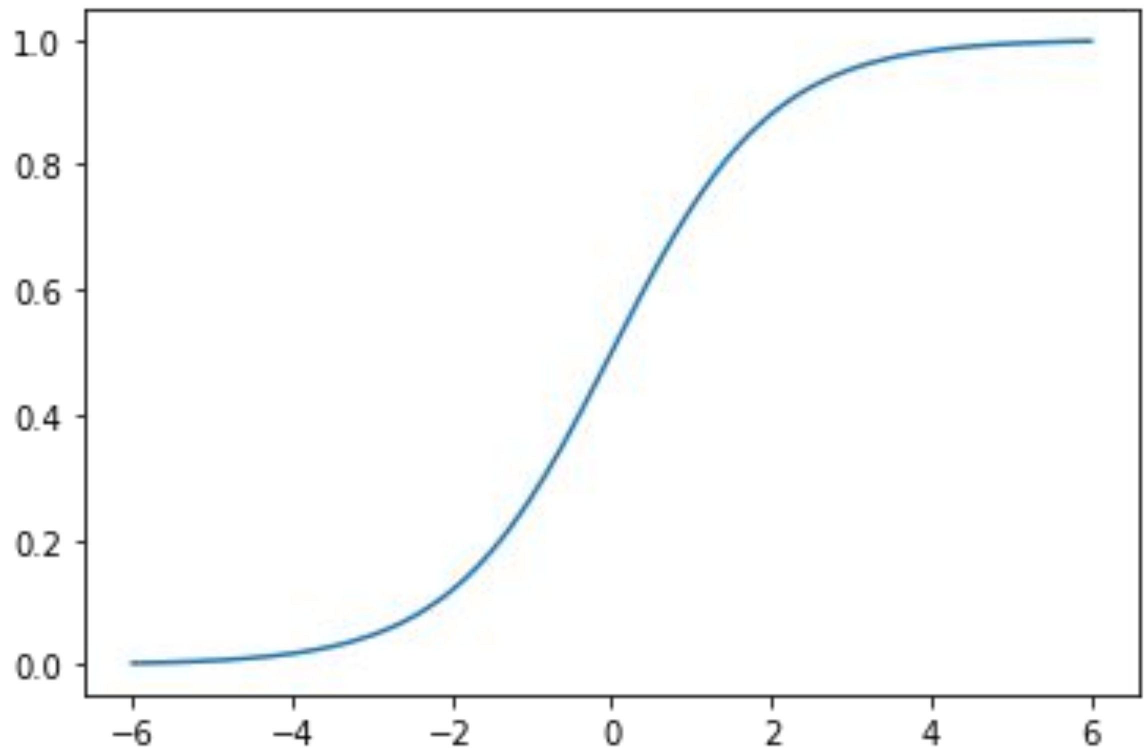
Fortunately, the algorithm for logistic regression is almost the same as linear regression!

But first, any ideas on how we can determine a category from a numerical output (think positive and negative numbers)?

Difference 1: Sigmoid Function

The sigmoid curve will map our very positive values to 1, and very negative values to zero

$$\frac{1}{1+e^{-x}}$$



Predicting Logistic Values

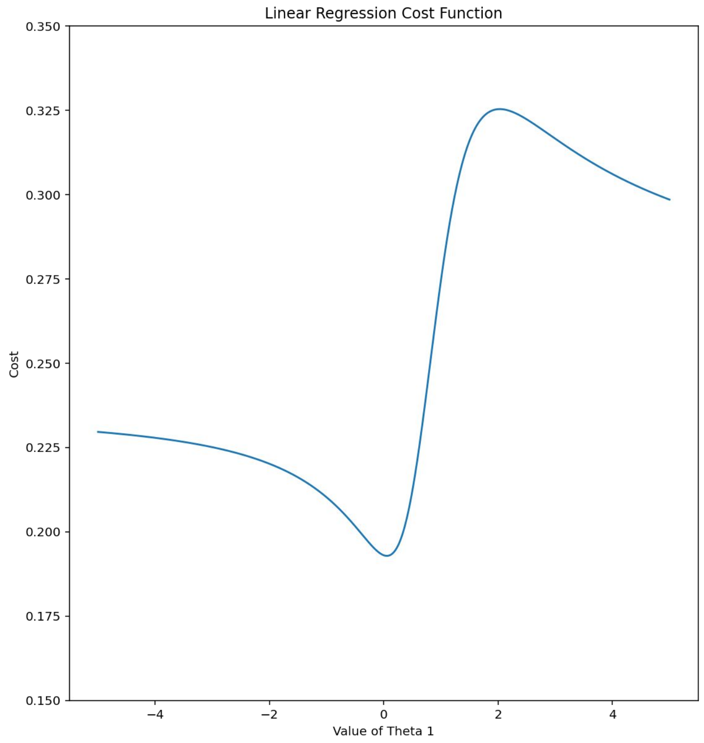
Now that we have a way to get an output between 0 and 1, we need to change our gradient descent algorithm to reflect this:

$$y = \begin{Bmatrix} y_0 \\ y_1 \\ y_2 \end{Bmatrix} = \begin{Bmatrix} 1 & x_0 \\ 1 & x_1 \\ 1 & x_2 \end{Bmatrix} \cdot \{\Theta_0 \ \Theta_1\}$$

$$y = sigmoid(y)$$

Difference 2: Cost Function

Here's what happens when we use our linear regression cost function in logistic regression:



In more complicated examples, there are many more minimum points. We need a cost function with only one minimum

Logistic Cost Function

Here is the general outline for our logistic cost function:

```
if (actual_value == 0):
   predicted_value = 0  -> 0 cost
   predicted_value = 1  -> very high cost

if (actual_value == 1):
   predicted_value = 0  -> very high cost
   predicted_value = 1  -> 0 cost
```

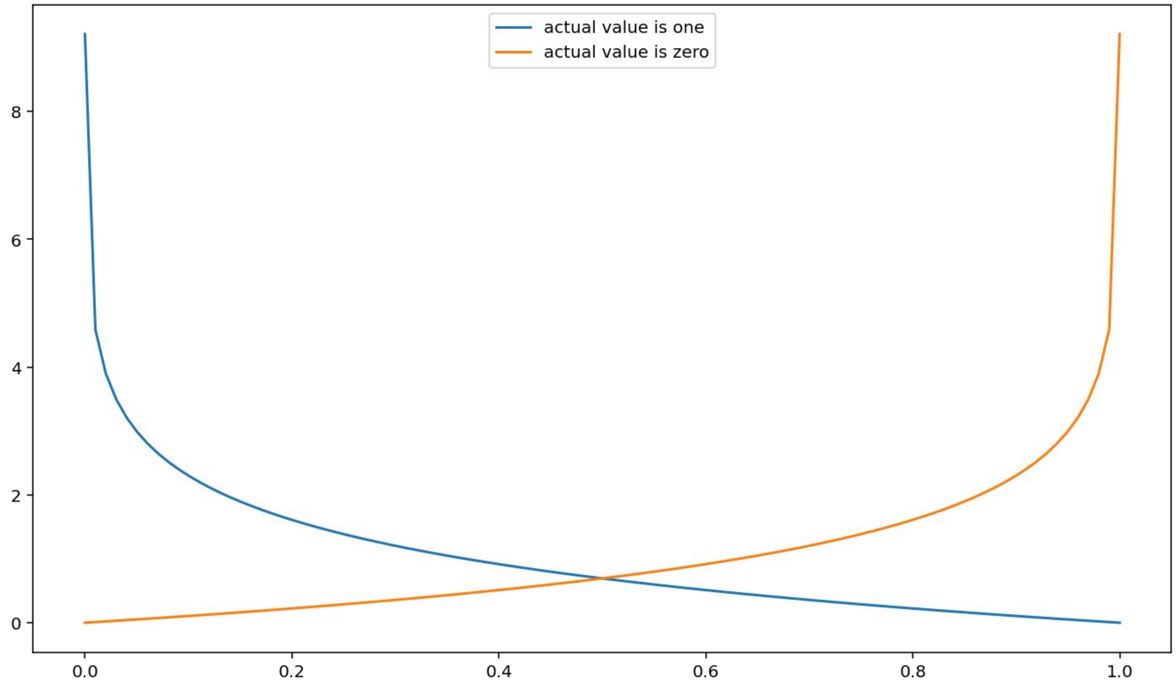
Logistic Cost Function (cont)

```
if (actual_value == 0):
```

$$cost = -log(x)$$

```
if (actual_value == 1):
```

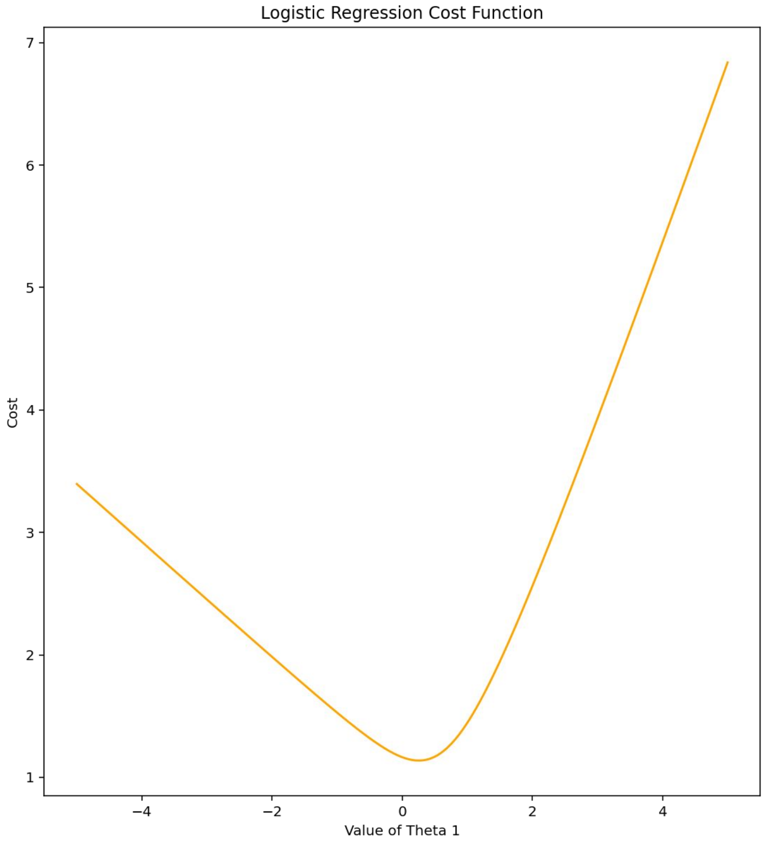
$$cost = -log(1 - x)$$



$$-\frac{1}{m} * sum(y^T \cdot log(y_{actual}) + (1 - y)^T * log(y_{actual}))$$

Logistic Cost Function vs Thetas

Now we only have one minimum point!



The question is why do we use this complicated function?

Logistic Cost Function Derivative

Taking the derivative of this function will leave us with:

$$\frac{1}{m} * [X^T \cdot (predicted - actual)]$$

Which is the same as our linear regression derivative!

Therefore, our general algorithm will be **almost** the same

General Algorithm

1) Predict a value for y from our thetas **<u>and take the sigmoid of that value</u>**

2) Calculate the derivative of our cost function with that value of y

3) Take a small step in the direction of the negative derivative and adjust our thetas

4) Repeat steps 1 - 3 until we reach a minimum cost

Tasks to Complete

1) Work on the notebook in the google drive folder
   https://drive.google.com/drive/folders/1EE8K1jO2b0X9sAuwohATK3DgY5endF68?usp=sharing

Try to work on it collaboratively! You might meet some people you could do a project with in the future

As always, let us know if you need any help!

SIG
NLL