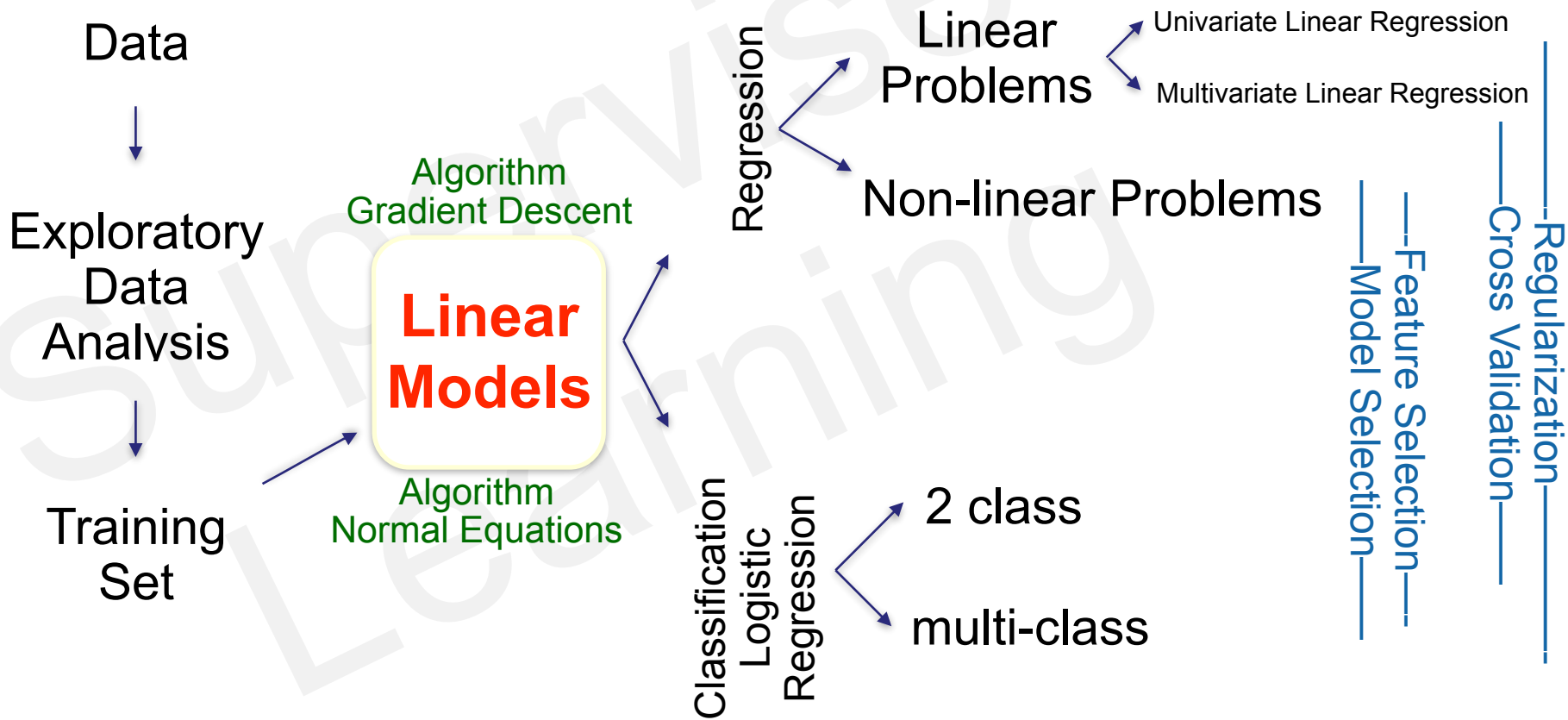
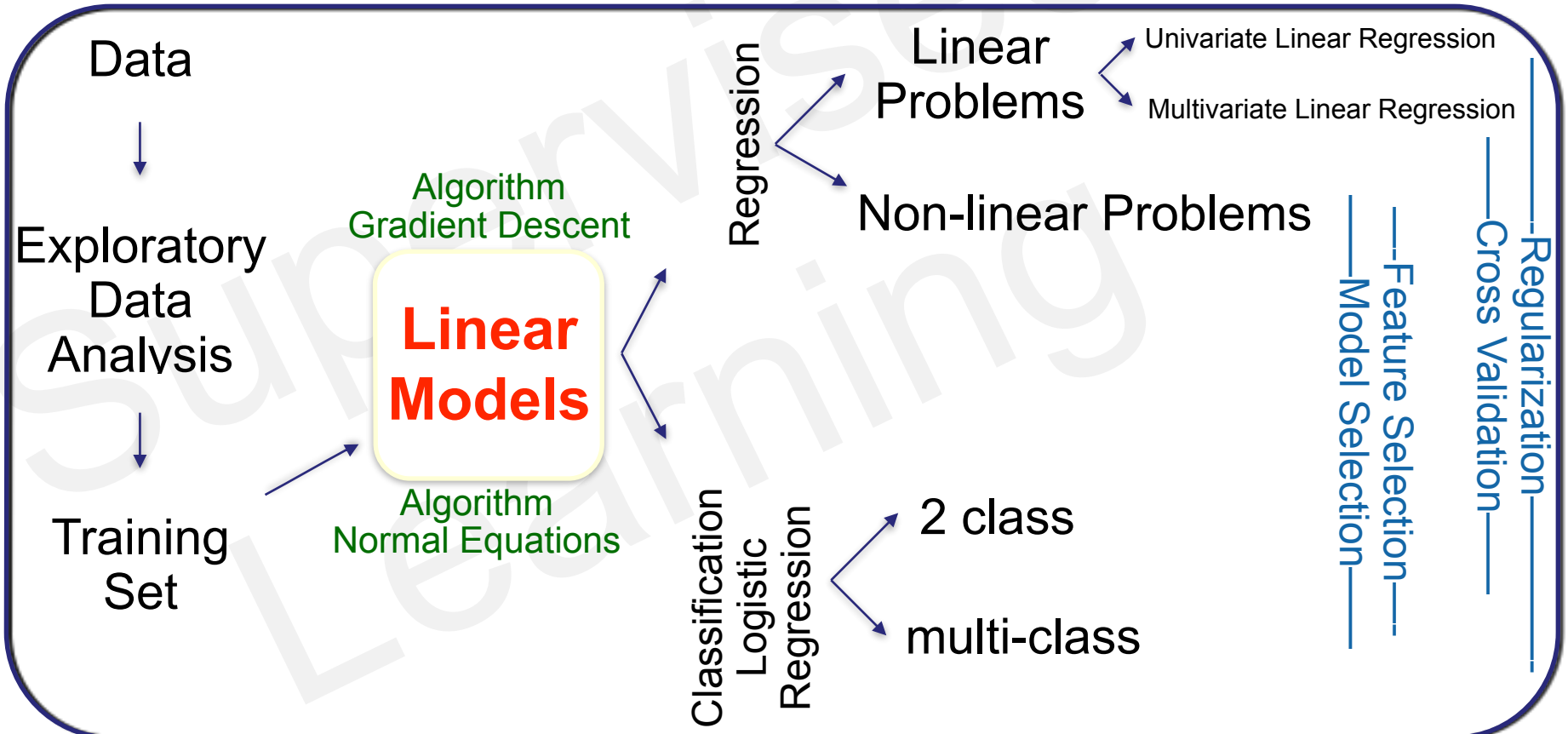
GENERAL ASSEMBLY

INTRO TO DATA SCIENCE
LECTURE 8: LOGISTIC REGRESSION

WHERE ARE WE ON THE DATA SCIENCE ROAD-MAP?



WHERE ARE WE ON THE DATA SCIENCE ROAD-MAP?



KEY CONCEPTS - CLASSIFICATION

Examples of classification:

- Binary Classification - 2 classes
  - Tumor: Benign or Malignant
  - Online Transaction: Fraudulent or Not-fraudulent

- Multi-class Classification
  - Image Classification: Cat or Dog or Horse or Deer

KEY CONCEPTS - CLASSIFICATION

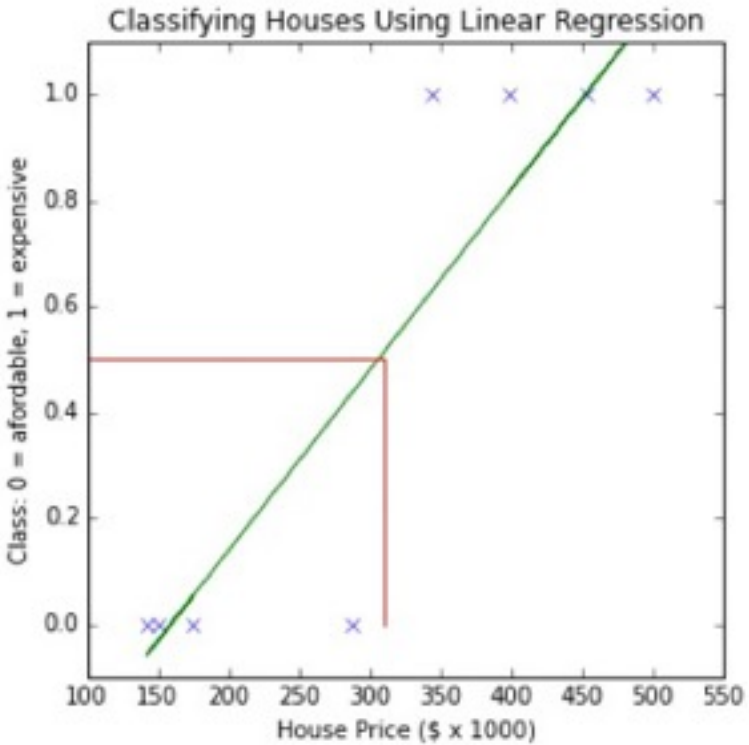
- The output of a classifier is y
- In a Binary Classification model y takes on 1 of 2 values:
  - 0 and 1, where 0 represents the ‘negative class’, and 1 represents the ‘positive class’
  - 0 might represent benign tumors, 1 might represent malignant tumors
- Intuitively the negative class conveys ‘the absence of something’
- But it really does NOT matter to which class is assigned 0 and to which class is assigned 1

KEY CONCEPTS - CLASSIFICATION

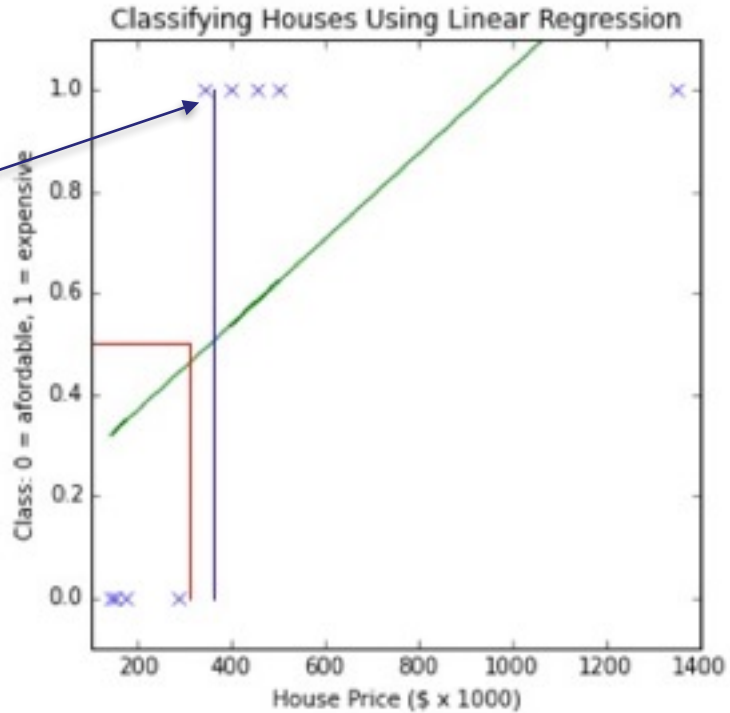
- In a Multi-class classification model y takes on integer values, starting at 0, and increasing with the number of classes in the problem

- For a 4 image classification problem, y would take on values 0 for cat, 1 for dog, 2, for horse, and 3 for deer

- If h(x) >= 0.5 might classify house as expensive
- If h(x) < 0.5 might classify house as affordable
- The decision threshold is at 0.5, i.e. when the hypothesis function returns values above we classify to one class, and when it returns values below we classify to the other class



- By adding a single example, however, linear regression will fail
- The new line moves the decision threshold, to the extent that one house previously classified as expensive would now be classified as affordable
- Linear Regression can also output values >1 and <0
- Don't use linear regression for classification problems

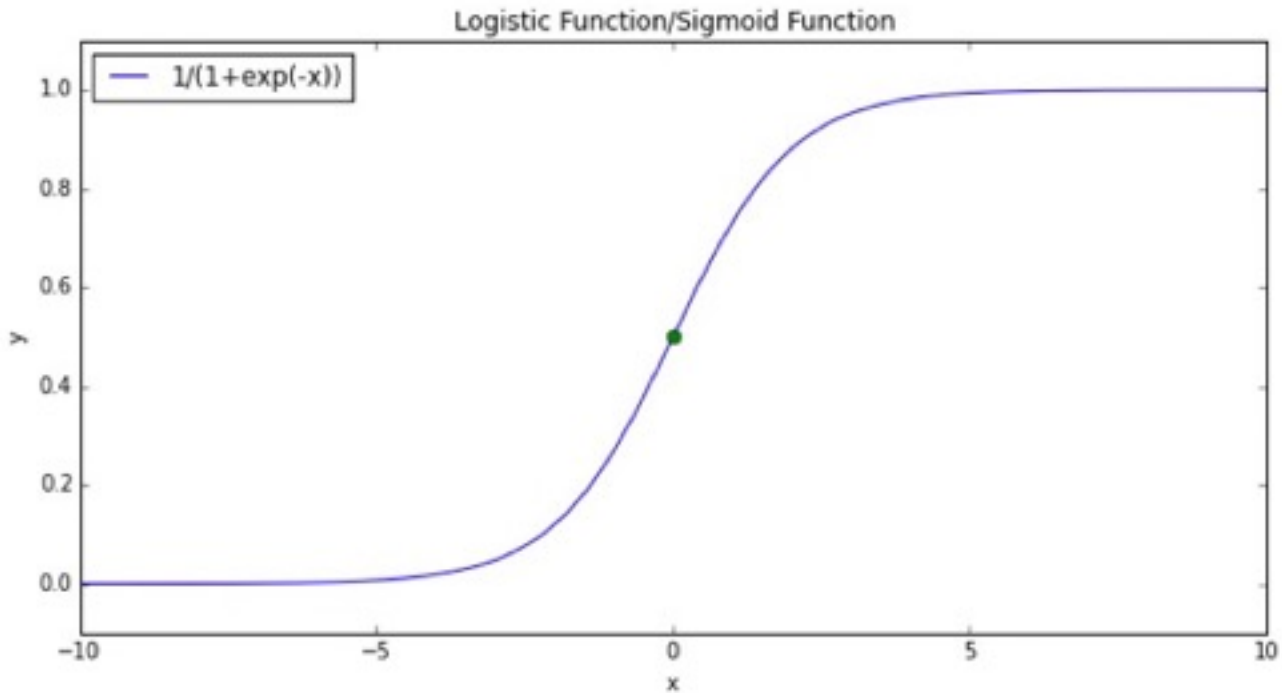


KEY CONCEPTS - LOGISTIC REGRESSION

- We are going to use Logistic Regression, where 0 <= h(x) <= 1

- To obtain the property of constraining all values between 0 and 1, we utilize the sigmoid or logistic function

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



KEY CONCEPTS - LOGISTIC REGRESSION

$$h(x) = \theta_0 + \theta_1 x_1 + \theta_2 x_2$$

$$h'(x) = \frac{1}{1+e^{-h(x)}}$$

$$h'(x) = \frac{1}{1+e^{-(\theta_0+\theta_1 x_1+\theta_2 x_2)}}$$

- This satisfies the constraint that we want - that h(x) returns values between 0 and 1

KEY CONCEPTS - LOGISTIC REGRESSION

- Since the new hypothesis function outputs values between 0 and 1 we can treat the output as a probability
- For example, if our house price is $1000000.00 then our hypothesis function might return a value of 0.7
- This is interpreted as a 70% chance that the house belongs to the class of expensive houses
- You are measuring P(y=1|x), i.e. the probability that y belongs to class 1, given x

KEY CONCEPTS - LOGISTIC REGRESSION

- In a binary classification task there are only 2 values that y can take on - namely 0 and 1

- Therefore, $P(y=0|x) + P(y=1|x) = 1$

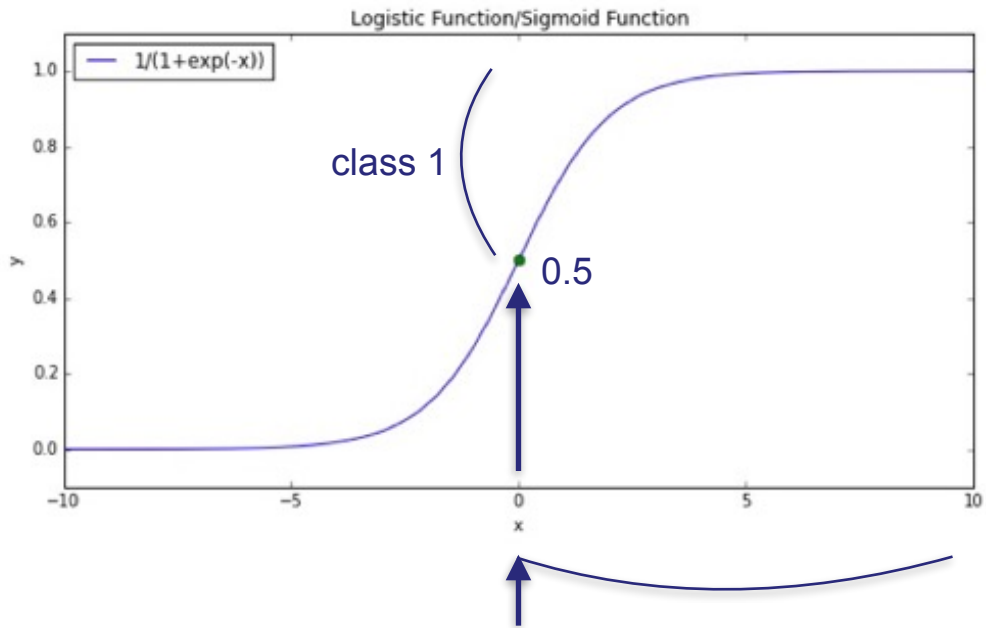
  and hence,

  $P(y=0|x) = 1 - P(y=1|x)$

- So in our example, there is a 30% chance that the house will belong to the class of houses we are calling ‘affordable’

KEY CONCEPTS - DECISION BOUNDARY

- Note carefully that y is classified as 1 if the logistic function returns greater than 0.5
- This occurs when y is greater than or equal to 0
- The logistic regression algorithm will find values for θ that satisfy these requirements



KEY CONCEPTS - LINEAR DECISION BOUNDARY

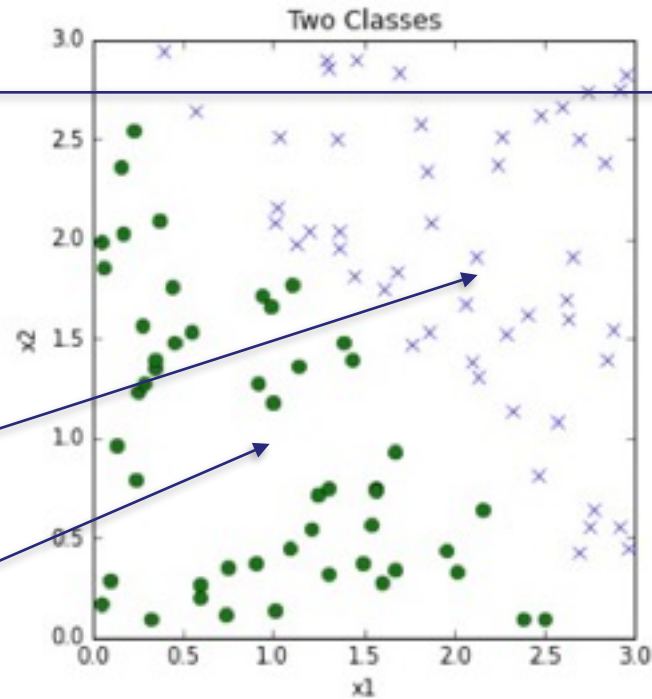
Setting the parameters to:

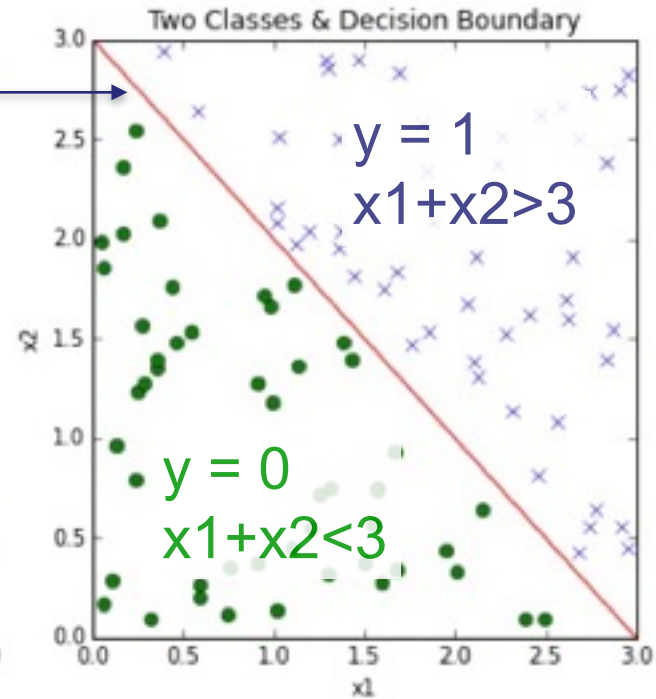
- $\theta_0 = -3$, $\theta_1 = 1$, $\theta_2 = 1$

- means that the original hypothesis function returns 0, such that both classes can be separated.

e.g. x1 = 2, x2 = 2
h(x1, x2) = 1.0 > 0.0
y=h'(1) > 0.5

e.g. x1 = 1, x2 = 1
h(x1, x2) = -1.0 < 0.0
y=h'(-1) < 0.5





KEY CONCEPTS - LINEAR DECISION BOUNDARY

The parameters of our Linear Model completely define the decision boundary

KEY CONCEPTS - NON-LINEAR DECISION BOUNDARY

$$h(x) = \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \theta_3 x_1^2 + \theta_4 x_2^2$$

$$h'(x) = \frac{1}{1+e^{-h(x)}}$$

$$h'(x) = \frac{1}{1+e^{-(\theta_0+\theta_1 x_1+\theta_2 x_2+\theta_3 x_1^2+\theta_4 x_2^2)}}$$

KEY CONCEPTS - NON-LINEAR DECISION BOUNDARY

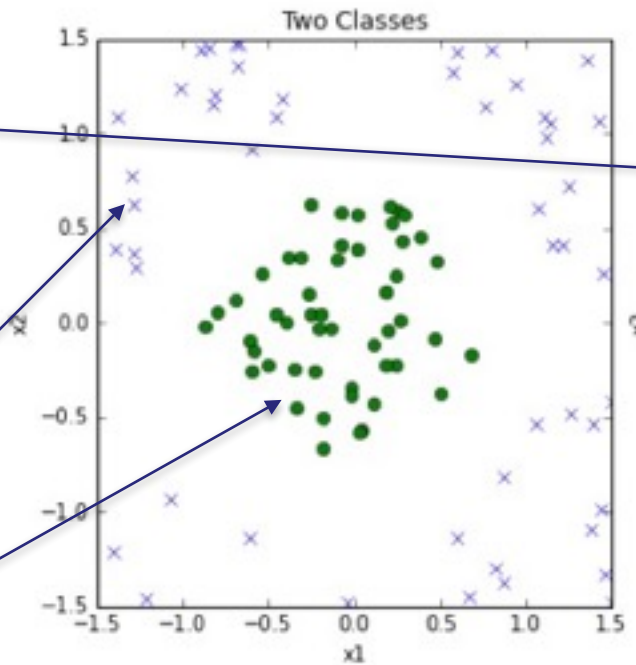
- Setting the parameters to:

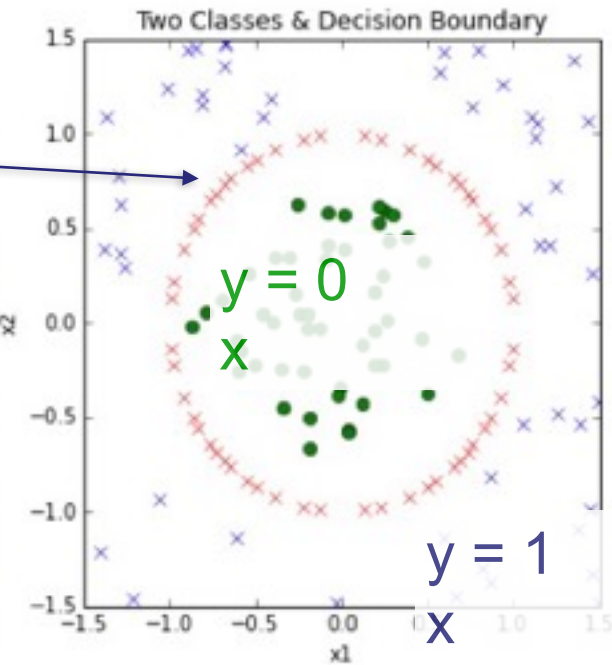
$\theta_0 = -1$, $\theta_1 = 0$, $\theta_2 = 0$,
$\theta_3 = 1$, $\theta_4 = 1$

- means that the original hypothesis function returns 0, such that both classes can be separated.

e.g. x1 = -1, x2 = 1
h(x1, x2) = 1.0 > 0.0
y=h'(1) > 0.5

e.g. x1 = 0.5, x2 = 0.5
h(x1, x2) = -0.5 < 0.0
y=h'(-1) < 0.5





KEY CONCEPTS - NON-LINEAR DECISION BOUNDARY

Higher order polynomials lead to even more complex decision boundaries

KEY CONCEPTS - FINDING THE PARAMETERS

- The original cost function non longer works, however.
- Although the shape of the cost function is still bowl shaped the non-linearity has introduced local minima.
- There is no longer a global minimum for gradient descent to find.

KEY CONCEPTS - A MODIFIED COST FUNCTION

Remember h'(x) returns 0 or 1

$$Cost(h'_\theta(x), y) = \begin{cases} -log(h'_\theta(x)) & if\, y = 1 \\ -log(1 - h'_\theta(x)) & if\, y = 0 \end{cases}$$

This can be expressed as:

$$Cost(h'(x), y) = -y \log(h'(x)) - (1 - y) \log(1 - h'(x))$$

Let $y = 1$

$$Cost(h'(x), y) = -\log(h'(x))$$

Let $y = 0$

$$Cost(h'(x), y) = -\log(1 - h'(x))$$

KEY CONCEPTS - A MODIFIED COST FUNCTION

$$J(\theta) = \frac{1}{m} \sum_{i=1}^{m} Cost(h'(x), y)$$

$$= -\frac{1}{m} \Big[\sum_{i=1}^{m} y \log(h'(x)) + (1 - y) \log(1 - h'(x)) \Big]$$

KEY CONCEPTS - A MODIFIED COST FUNCTION

Consider 4 examples:

$h'(x) = 1$ AND $y = 1$ => $J(\theta) = 0$

$h'(x) = 0$ AND $y = 1$ => $J(\theta)$ -> infinity

$h'(x) = 1$ AND $y = 0$ => $J(\theta)$ -> infinity

$h'(x) = 0$ AND $y = 0$ => $J(\theta) = 0$

KEY CONCEPTS - A MODIFIED COST FUNCTION

- So if $P(y=1|x) = 0$, but $y = 1$, then J is very high

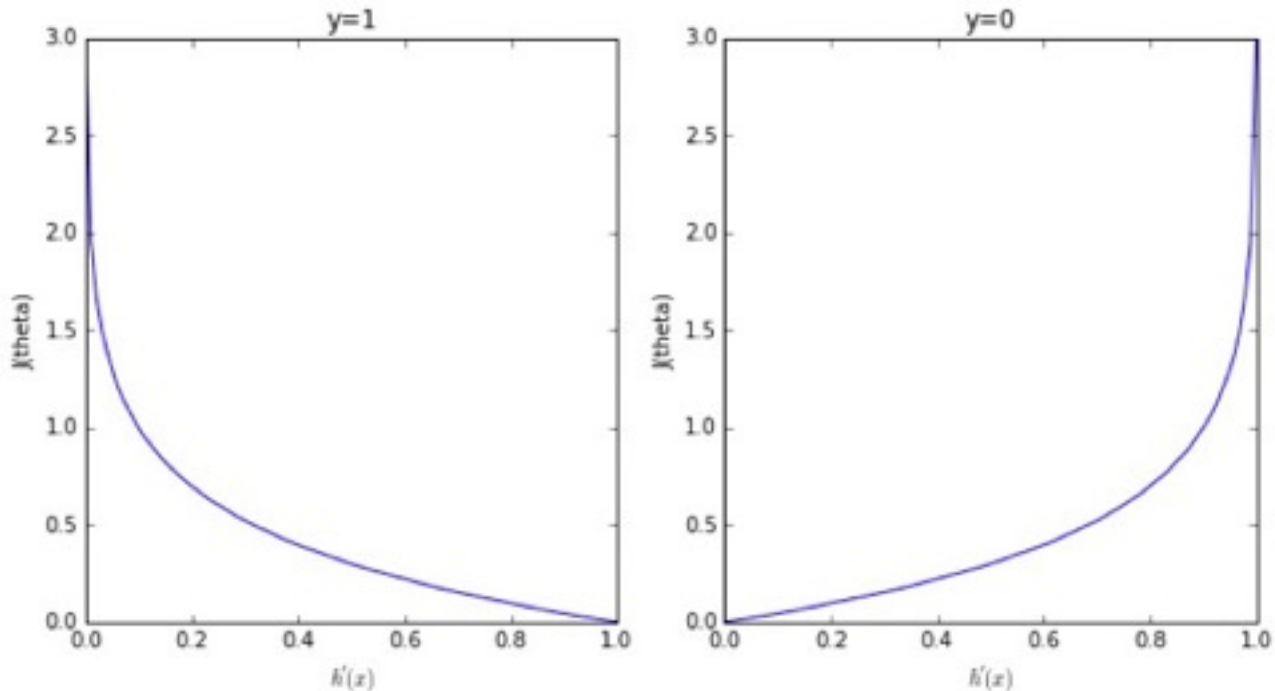
and if $P(y=0|x) = 0$, but $y = 0$, then J is, again, very high

AND LIKEWISE

- If the $P(y=1|x) = 1$, and $y = 1$, then J is 0

and if $P(y=0|x) = 0$, and $y = 0$, then J is, again, 0

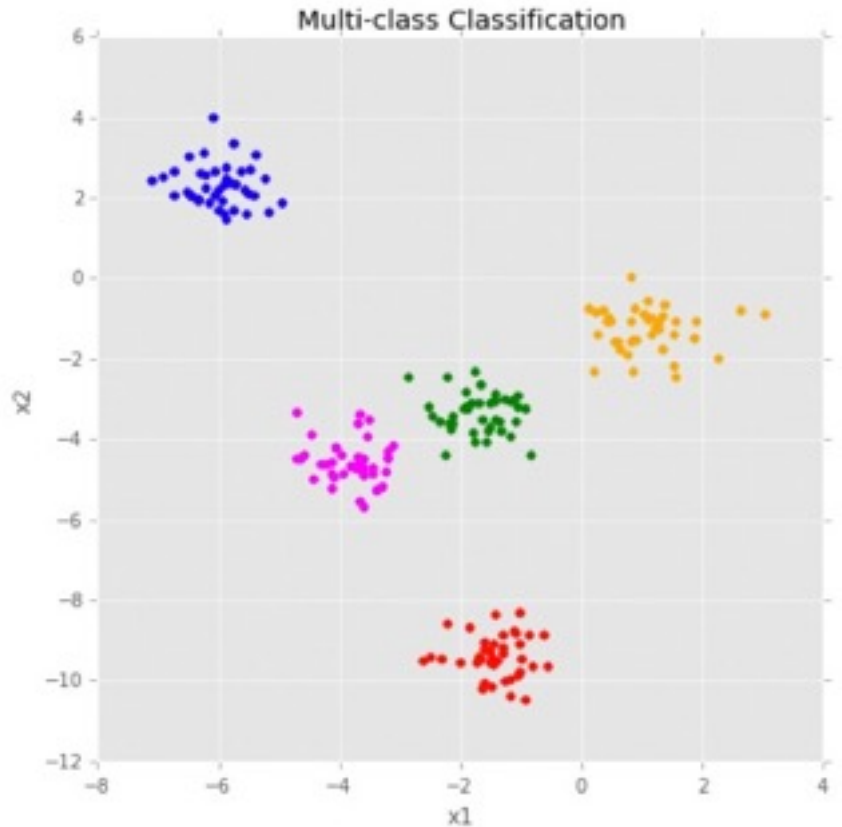
KEY CONCEPTS - THE MODIFIED COST FUNCTION - A SINGLE GLOBAL MINIMUM!



KEY CONCEPTS - MULTI-CLASS CLASSIFICATION

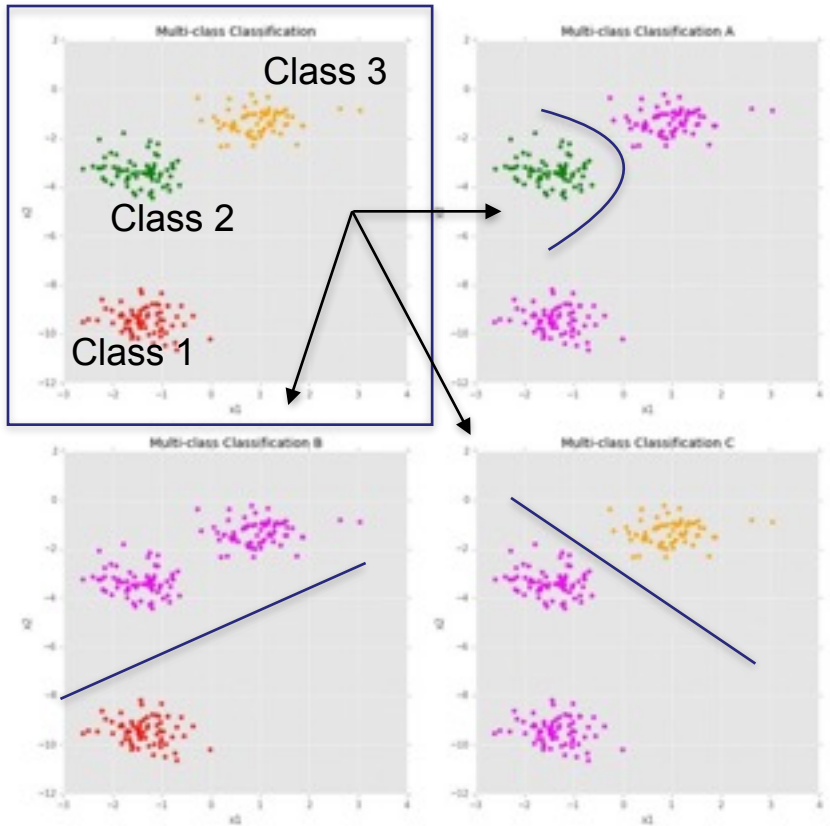
- y takes on a discrete number of outputs

e.g. In this examples y is 0, 1, 2, 3, or 4



KEY CONCEPTS - ONE-VS-ALL OR ONE-VS-REST

- Turn this problem in 3 binary classification problems
- To predict you pick which of the binary classifiers produces the largest probability



Sklearn - LogisticRegression()

- Remember to feature scale
- Use the pipeline for non-linear features
- Logistic Regression needs to be regularized

Sklearn - Logistic Regression Regularizer

- L1-norm, L2-norm available
- The regularization parameter is ‘C’
- It works slightly differently.
- C is a positive float
  - Small values of C provide stronger regularization

KEY CONCEPTS - ASSESSING PERFORMANCE - THE CONFUSION MATRIX

	Prediction		
Actual		***Negative***	***Positive***
	Negative	***TP***	***FP***
	Positive	***FN***	***TN***

- **True positive**: correctly classifying a malignant tumor
- **True negative**: correctly classifying a benign tumor
- **False positive**: a benign tumor that is incorrectly classifier as being malignant
- **False negative**: a malignant tumor that is incorrectly classifier as being benign

The confusion matrix:

In [3]:
```
#Fit a standard logistic regression model
clf = LogisticRegression()
clf.fit(X, y)

#Let's look at the confusion matrix
cm = pd.crosstab(y, clf.predict(X), rownames=["Actual"], colnames=["Predicted"])
cm
```

Out[3]:

Predicted	0	1	2	3	4
Actual					
0	40	0	0	0	0
1	0	40	0	0	0
2	0	0	40	0	0
3	0	0	0	40	0
4	0	0	0	0	40

KEY CONCEPTS - ASSESSING PERFORMANCE

Accuracy: the proportion of instances classified correctly

Accuracy = (TP+TN)/(TP + TN + FP + FN)

Be aware of using these metrics for classification when your training set is not ‘balanced’

http://www2.cs.uregina.ca/~dbd/cs831/notes/confusion_matrix/confusion_matrix.html