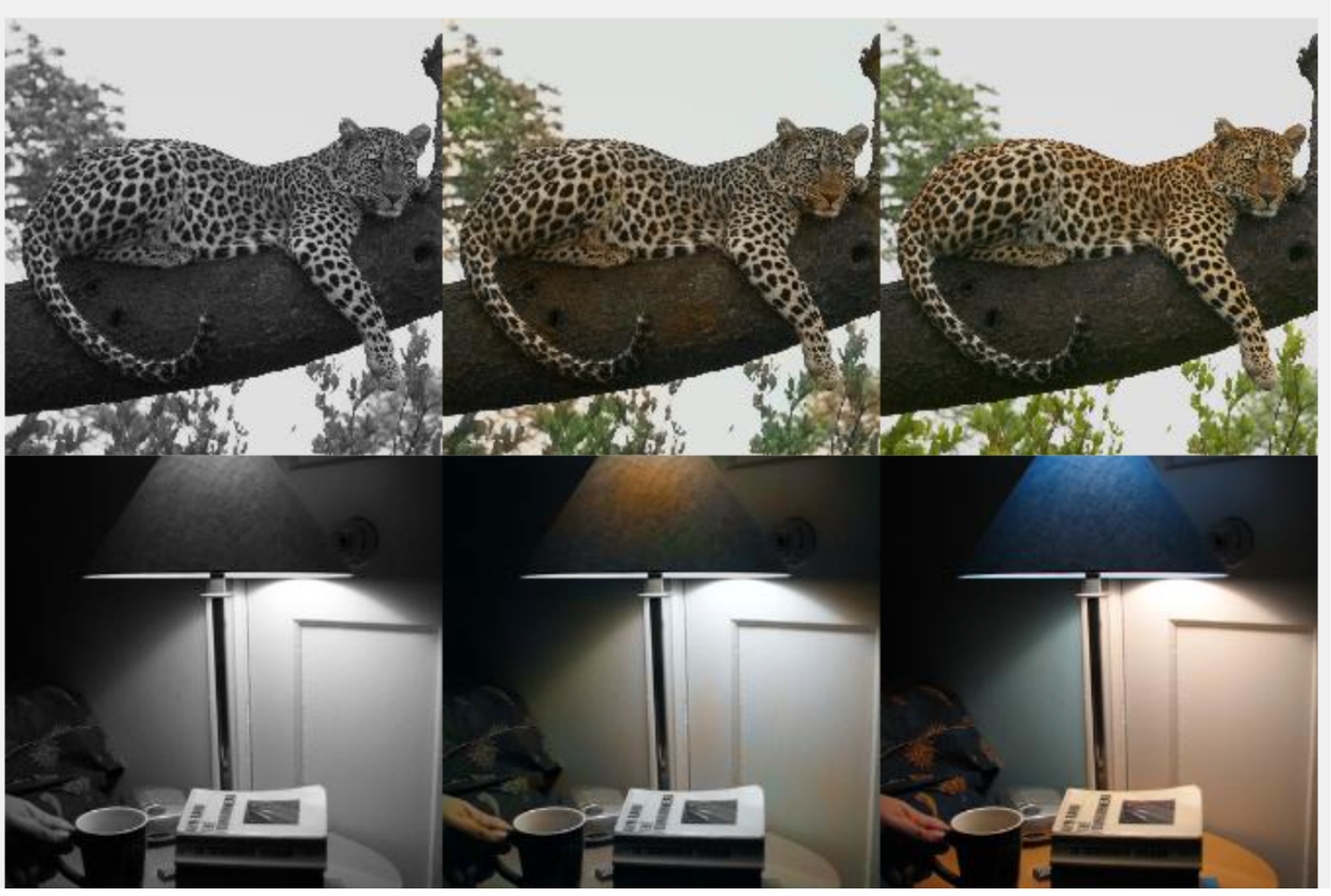
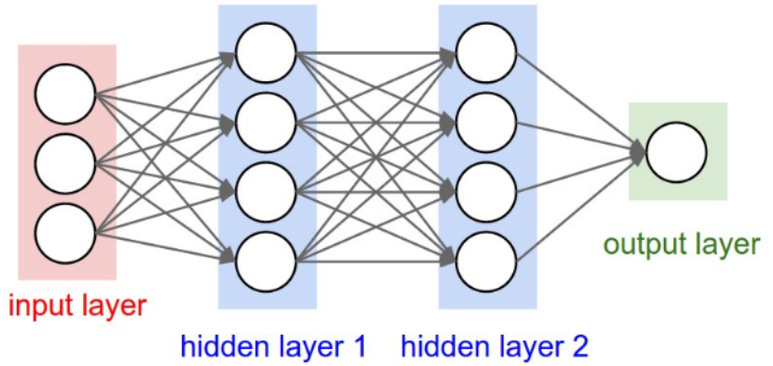
GENERAL ASSEMBLY

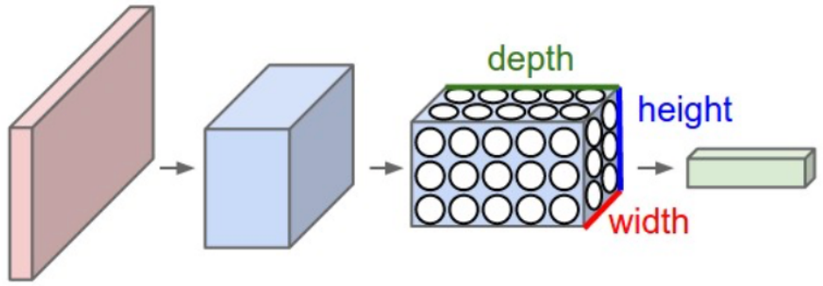
LOGISTIC REGRESSION

Brian Chung

COOL STUFF – AUTOMATIC COLORIZATION







http://tinyclouds.org/colorize/

RECAP NAIVE BAYES

Questions from last time:

- Frequentist vs Bayesian Thoughts
- Marginal Probabilities P(xly) vs P(xly')
- MAP vs ML estimate
- Log Transformations of the Multinomial Likelihood model
- GaussianNB()

$$P(\theta \mid X) = \frac{P(X \mid \theta)P(\theta)}{P(X)}$$

Multinomial PMF

$$\frac{n!}{x_1! \cdots x_k!} p_1^{x_1} \cdots p_k^{x_k}$$

LOGISTIC REGRESSION AGENDA

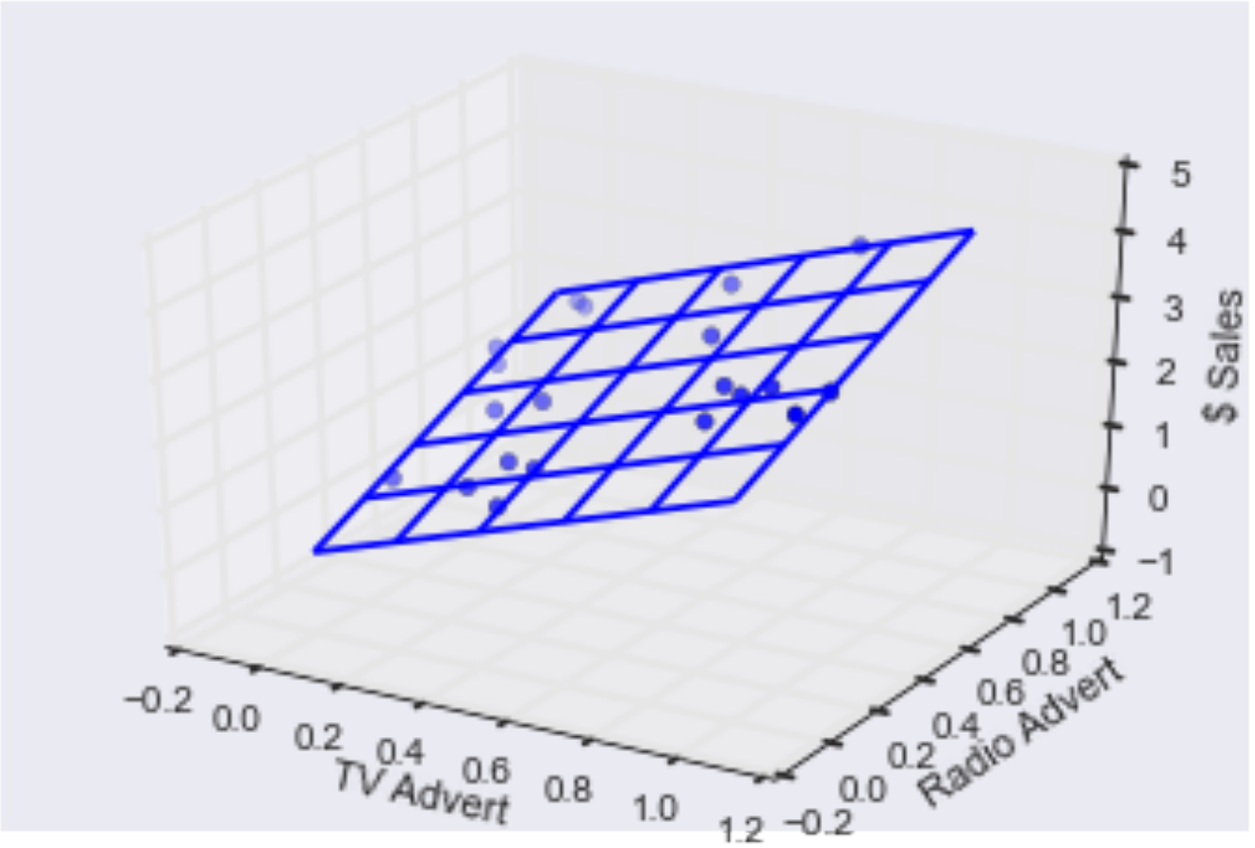
I. Precursor
II. Logistic Regression
III. Interpreting Results
IV. Decision Boundaries
V. Evaluating Classifiers

I. PRECURSOR

REGRESSION RECAP

$$y = \alpha + \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_3 + \beta_4 x_4 + \ldots + \beta_n x_n + \varepsilon$$

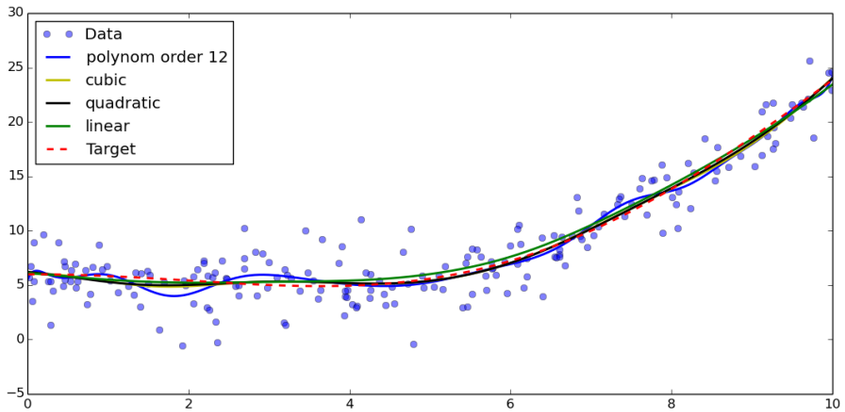
y: Response Variable
α: Intercept Value
β_1: Regression Coefficient (Beta1)
β_2: Regression Coefficient (Beta2)
...
ε: Residual (Error)

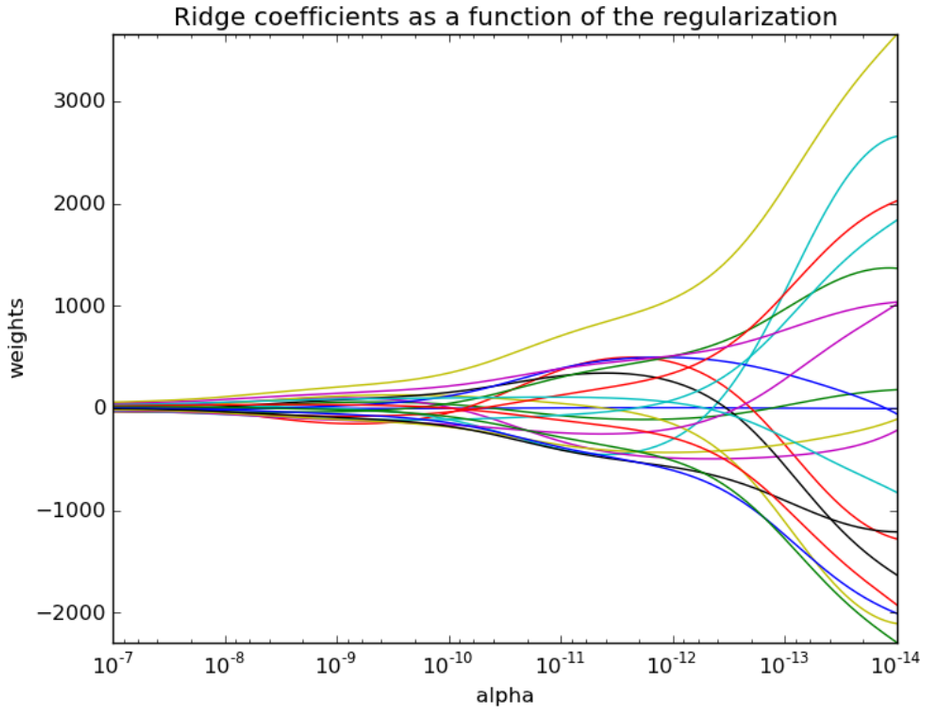


LINEAR REGRESSION RECAP

We've seen linear regression can do some power things

- Polynomial terms
- Transformed terms
- Categorical terms
- L1 Regularization (LASSO)
- L2 Regularization (Ridge Regression)





LINEAR REGRESSION RECAP

We’ve seen linear regression can do some power things…

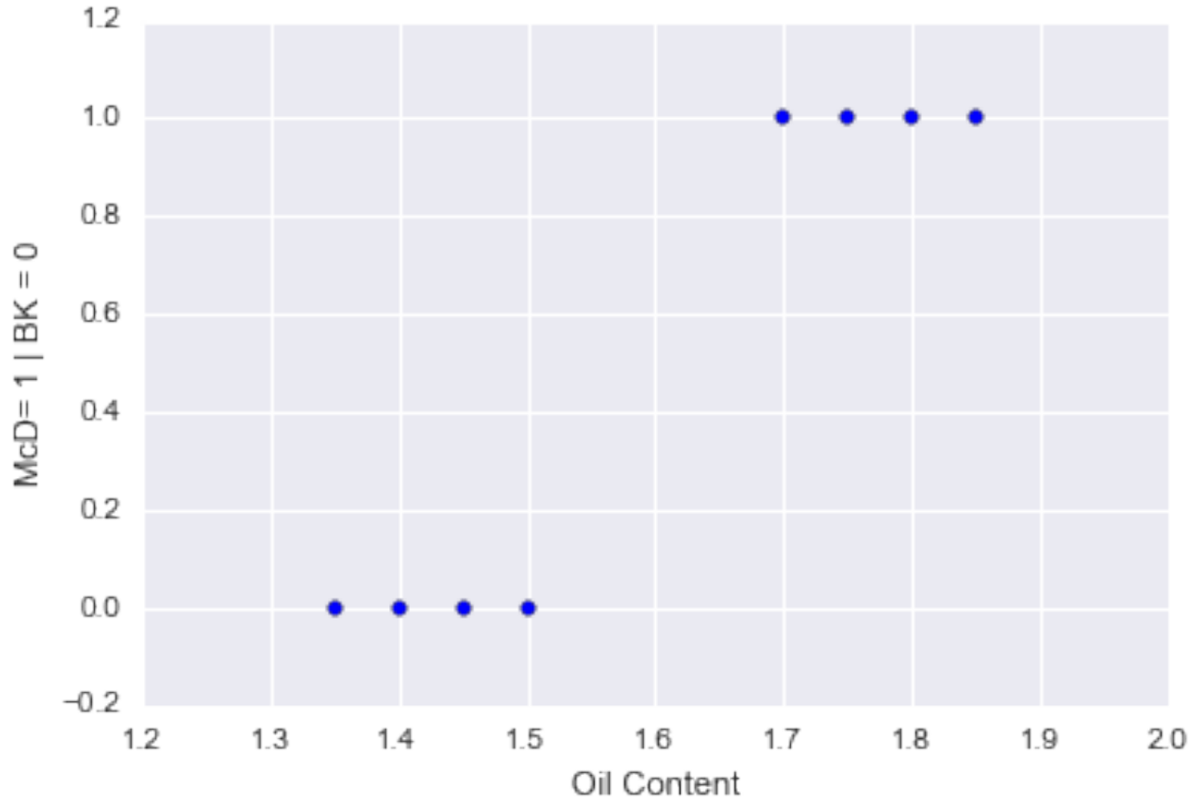
…But, will it blend? **CLASSIFY?**

LOGISTIC REGRESSION

Let’s say we want to classify class C, that has two labels: “McDonalds”, and “Burger King” based on a single feature x: “oil content”

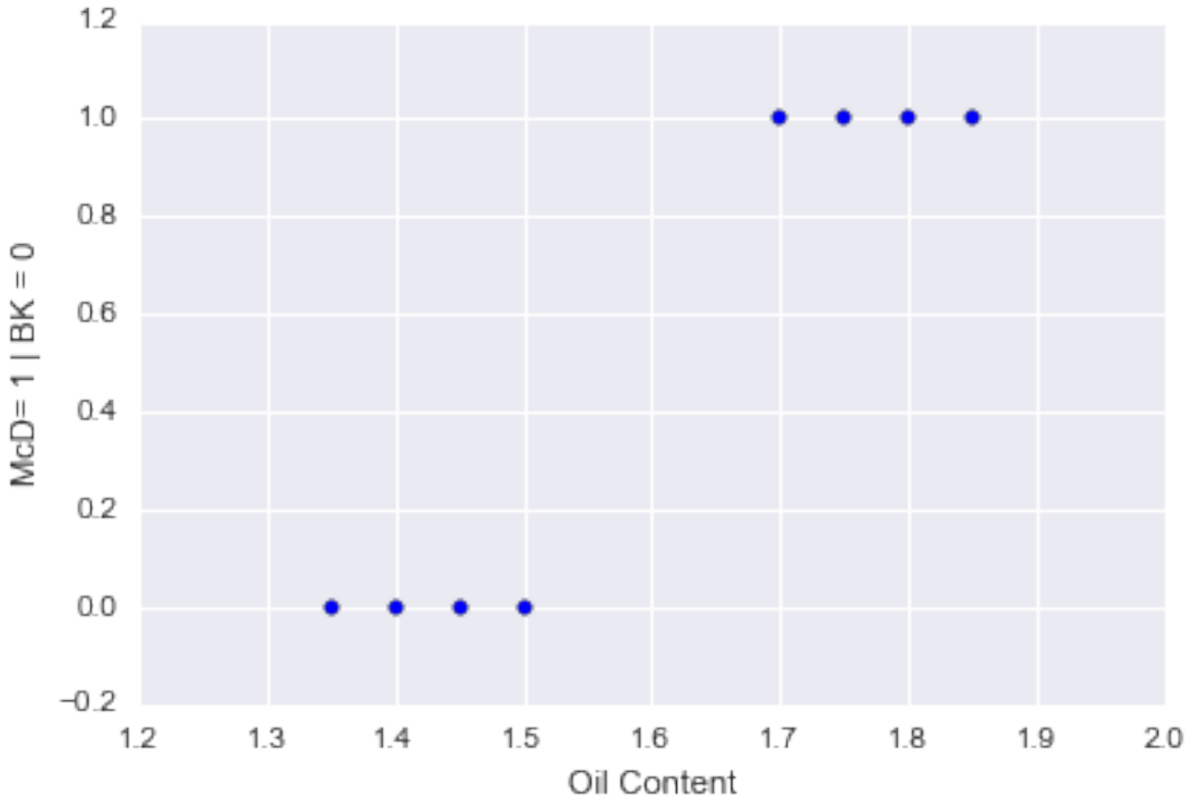
LOGISTIC REGRESSION

Let's say we want to classify a number of french fries as "McDonalds" or "Burger King" based on a single feature x: "oil content" **What if we code McDonalds as "1", and "Burger King" as "0"?**



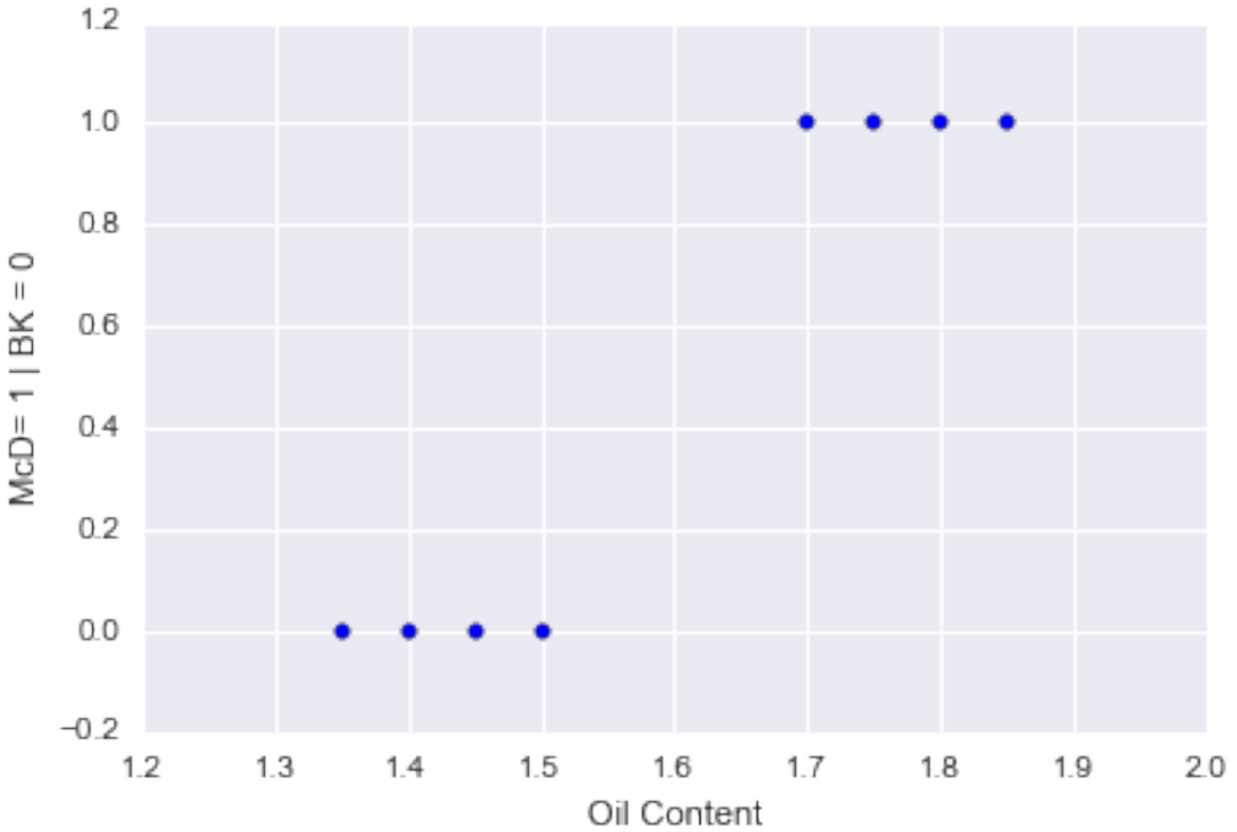
LOGISTIC REGRESSION

Seems McD fries have higher oil content. What if we use a linear regression line, and make a rule. Any time the line is greater than 0.5, we'll classify as McD. Any time the line is less than 0.5, we'll classify as BK



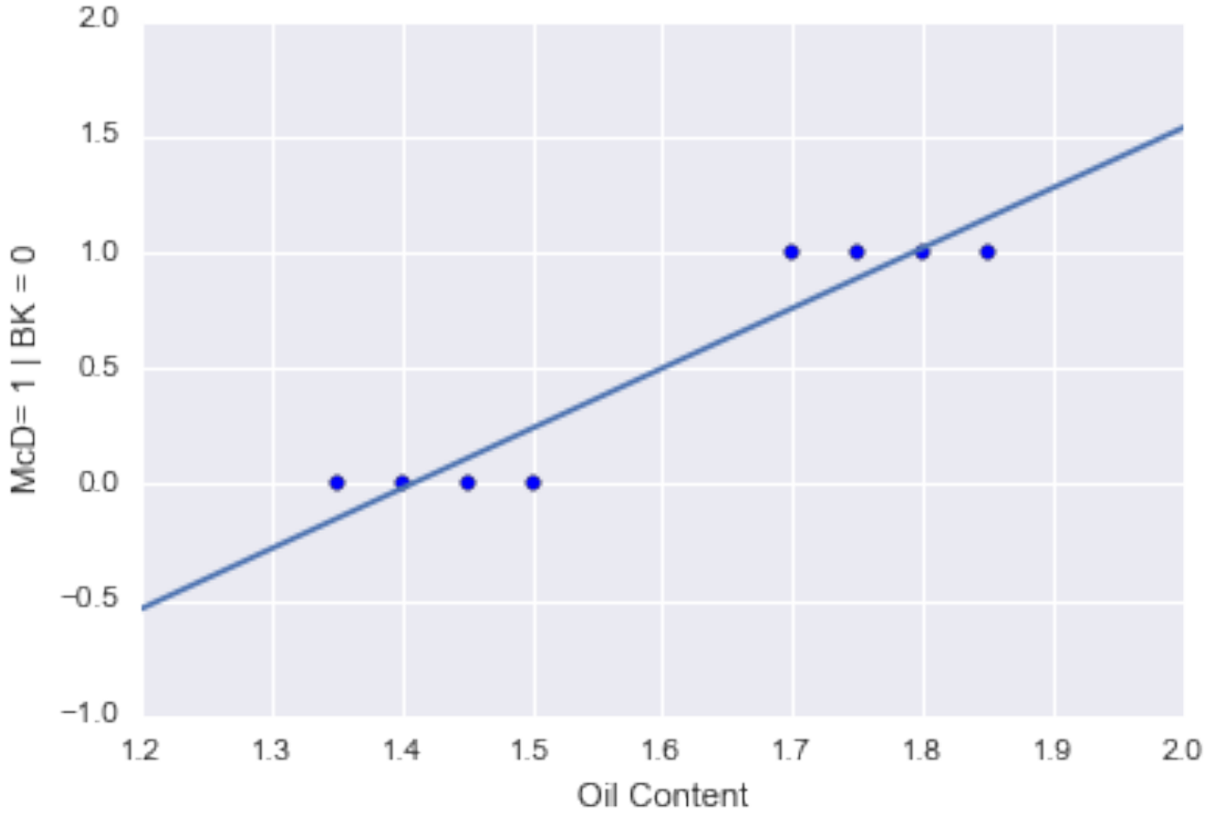
LOGISTIC REGRESSION

Any possible issues with that scheme?



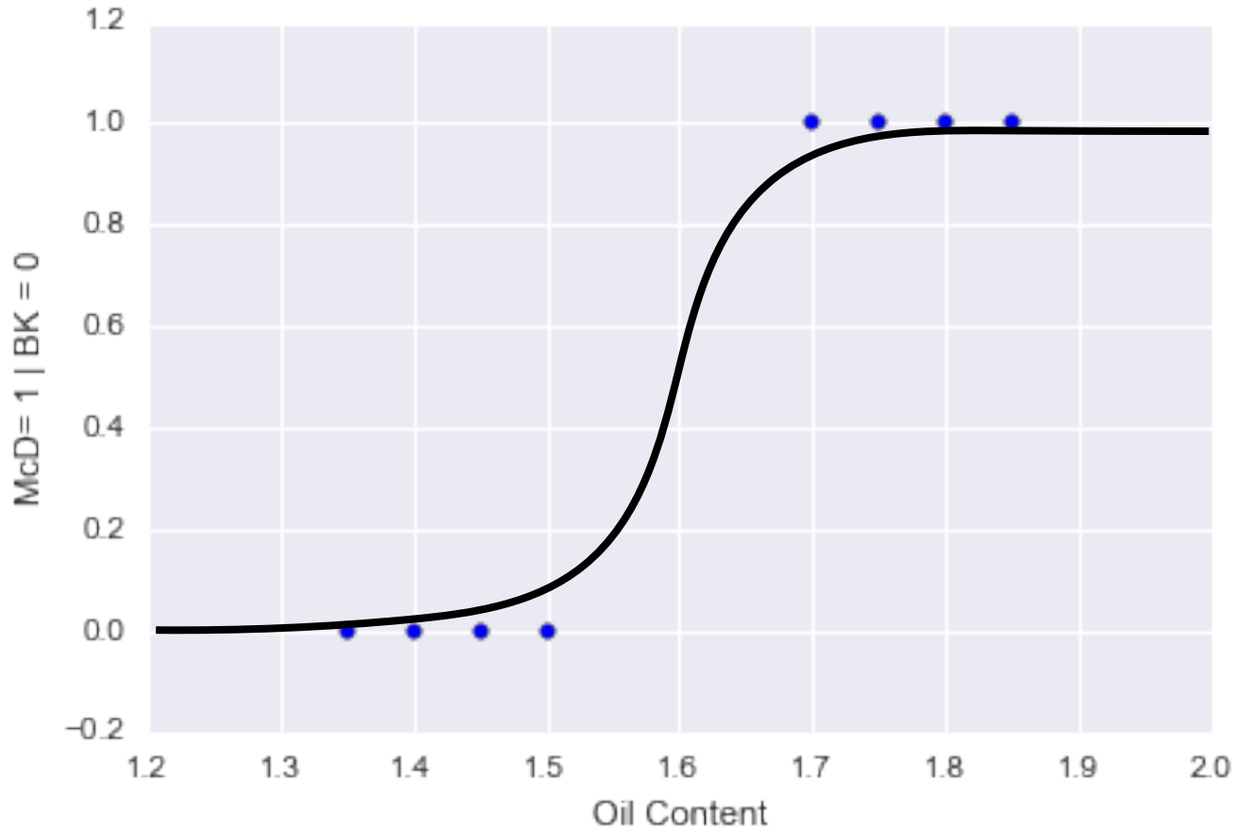
LOGISTIC REGRESSION

Any possible issues with that scheme?
* What does -0.5 probability mean? What does 1.5 probability mean?



LOGISTIC REGRESSION

Ideally, we'd want our function to somehow be limited to the range of [0,1] regardless of the value of "x"



I. PRECURSOR
II. LOGISTIC REGRESSION

TYPES OF ML SOLUTIONS

	Continuous	Categorical
Supervised	Regression	Classification
Unsupervised	Dimension Reduction	Clustering

LOGISTIC REGRESSION

Q: What *is* **logistic regression?**

LOGISTIC REGRESSION

Q: What *is* **logistic regression?**

A: A generalization of linear regression to classification problems

LOGISTIC REGRESSION

In **linear regression**, features predict a continuous outcome variable

LOGISTIC REGRESSION

In **linear regression**, features predict a continuous outcome variable

In **logistic regression**, features predict **probabilities** of binary class membership (though multiple class variants exist)

LOGISTIC REGRESSION

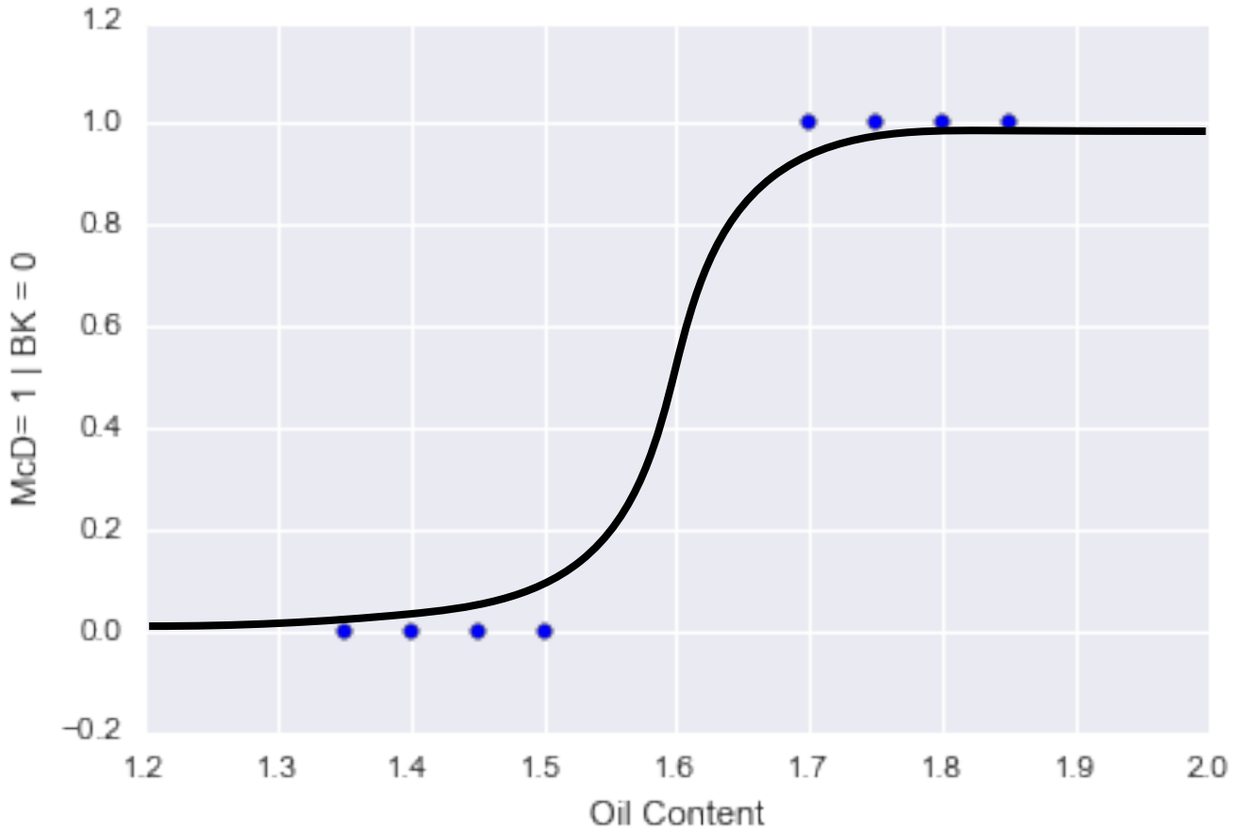
In **linear regression**, features predict a continuous outcome variable

In **logistic regression**, features predict **probabilities** of binary class membership (though multiple class variants exist)

These probabilities are then mapped to **class labels** (i.e. McD/BK), thus solving the classification problem.

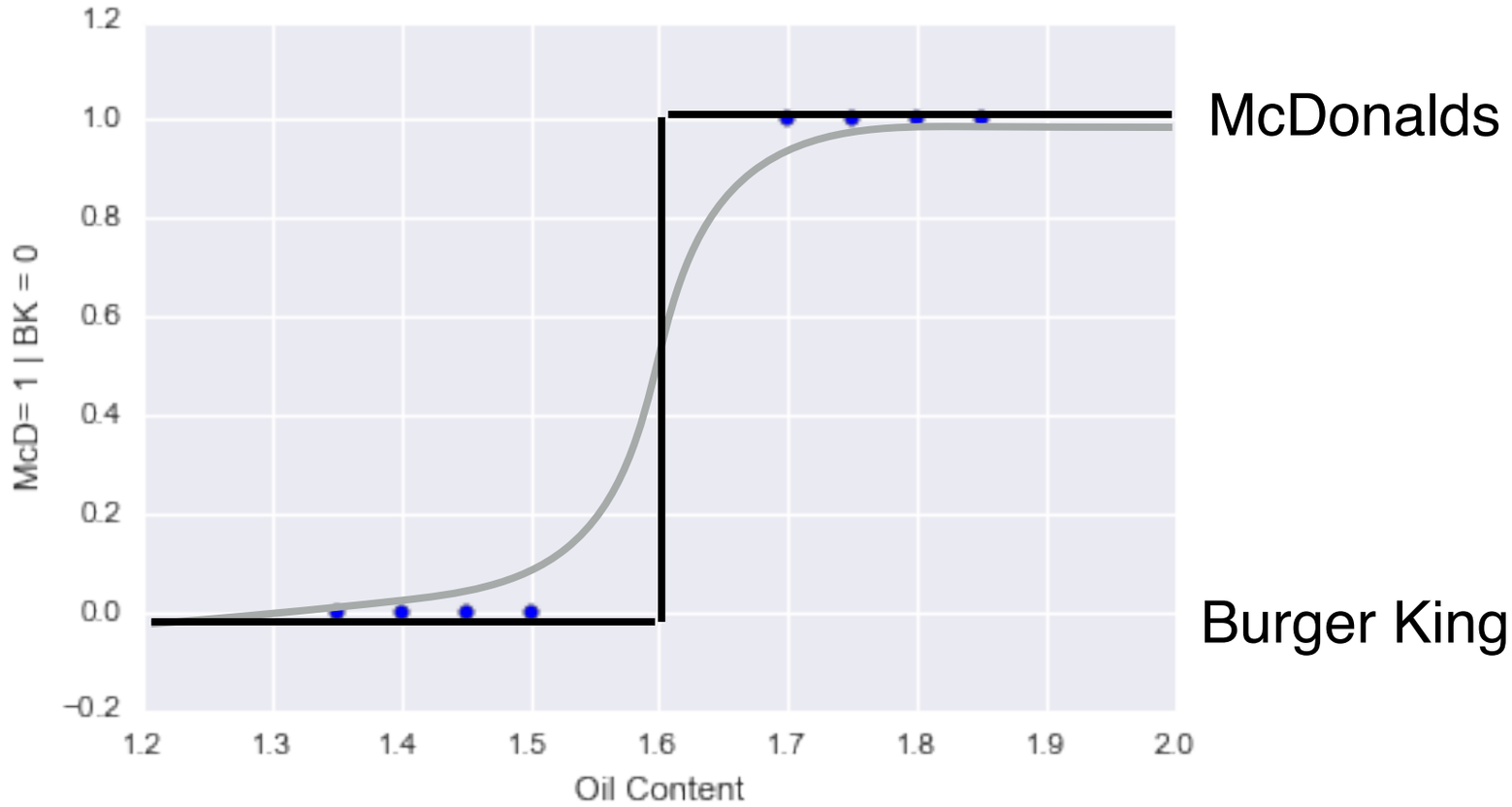
LOGISTIC REGRESSION

Logistic regression gives us predicted probabilities (i.e. between 0 and 1)



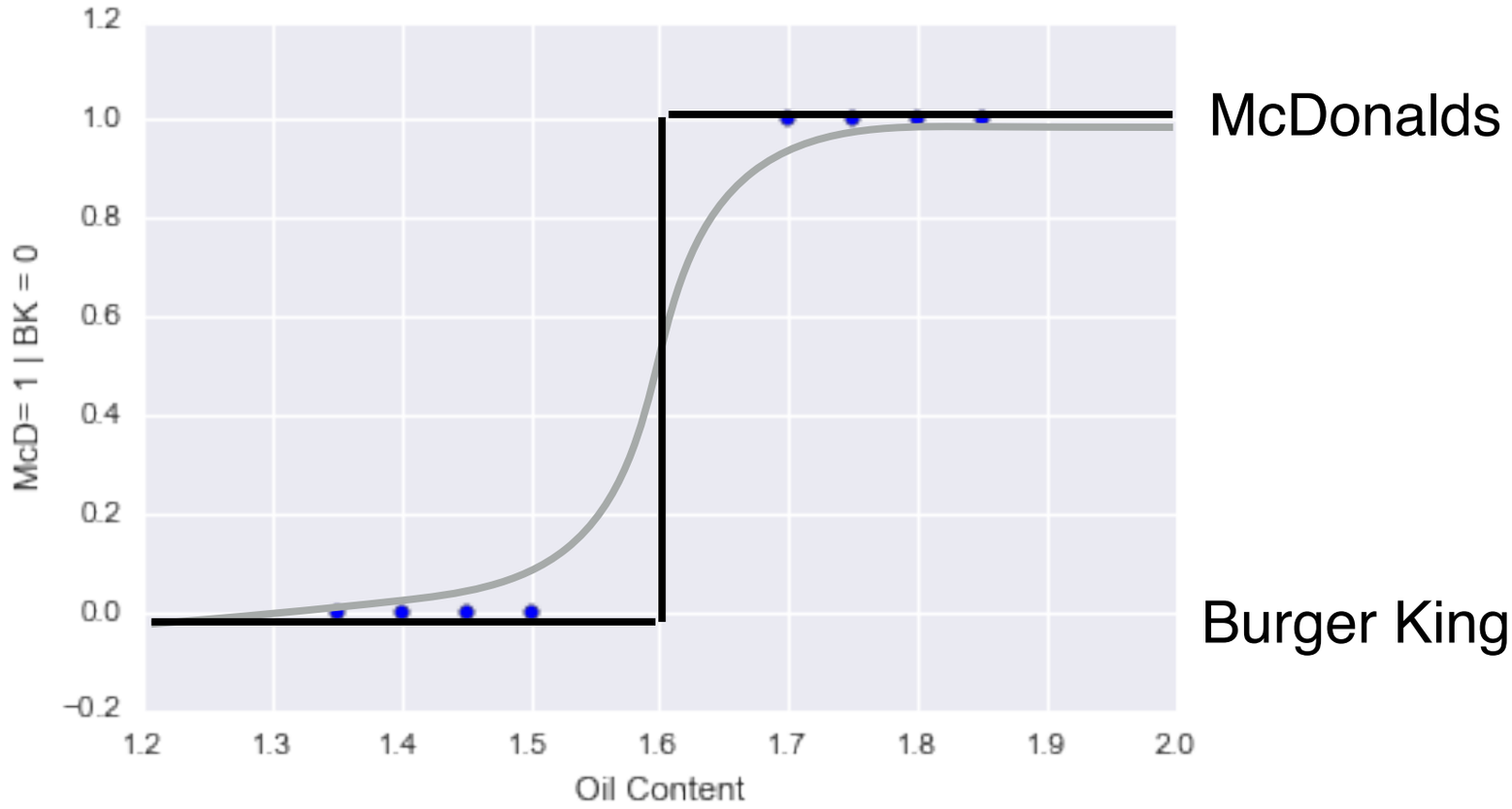
LOGISTIC REGRESSION

Logistic regression gives us predicted probabilities (i.e. between 0 and 1), which can then be ‘snapped’ to class labels



LOGISTIC REGRESSION

Very cool! We originally specified 0 and 1 as class labels, but now we get the benefit of predicting *probabilities of classes*, not only class labels



LOGISTIC REGRESSION

The logistic regression model is an extension of the linear regression model, with a couple of important differences.

LOGISTIC REGRESSION

The logistic regression model is an extension of the linear regression model, with a couple of important differences.

The first difference is in what the outcome (or predicted) variable is

LOGISTIC REGRESSION

The logistic regression model is an extension of the linear regression model, with a couple of important differences.

The first difference is in what the outcome (or predicted) variable is

The second difference is in the error term

LOGISTIC REGRESSION

We’ve seen that for logistic regression, the conditional mean E[Y|X] of the outcome takes values **only** in the unit interval [0,1]

LOGISTIC REGRESSION

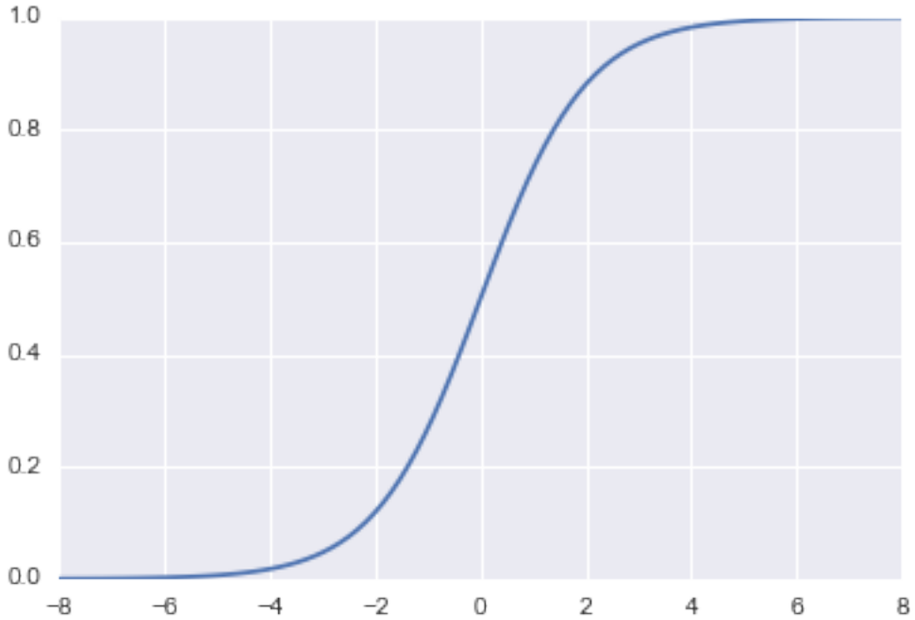
We've seen that for logistic regression, the conditional mean E[Y|X] of the outcome takes values **only** in the unit interval [0,1]

So our first step is to be able to transform the outcome variable into the unit interval.

LOGISTIC REGRESSION

Q: How do we map the outcome into the unit variable?
A: Through the **logistic (sigmoid) function:**

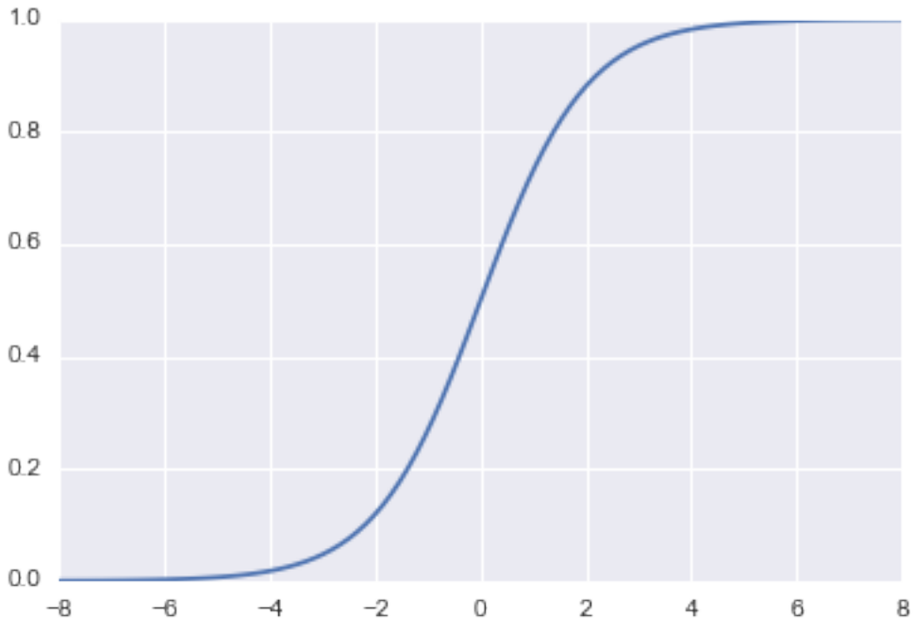
$$sigmoid(z) = \frac{1}{1+e^{-z}}$$



LOGISTIC REGRESSION

Q: How do we map the outcome into the unit variable?
A: Through the **logistic (sigmoid) function:**

$$sigmoid(z) = \frac{1}{1+e^{-z}}$$

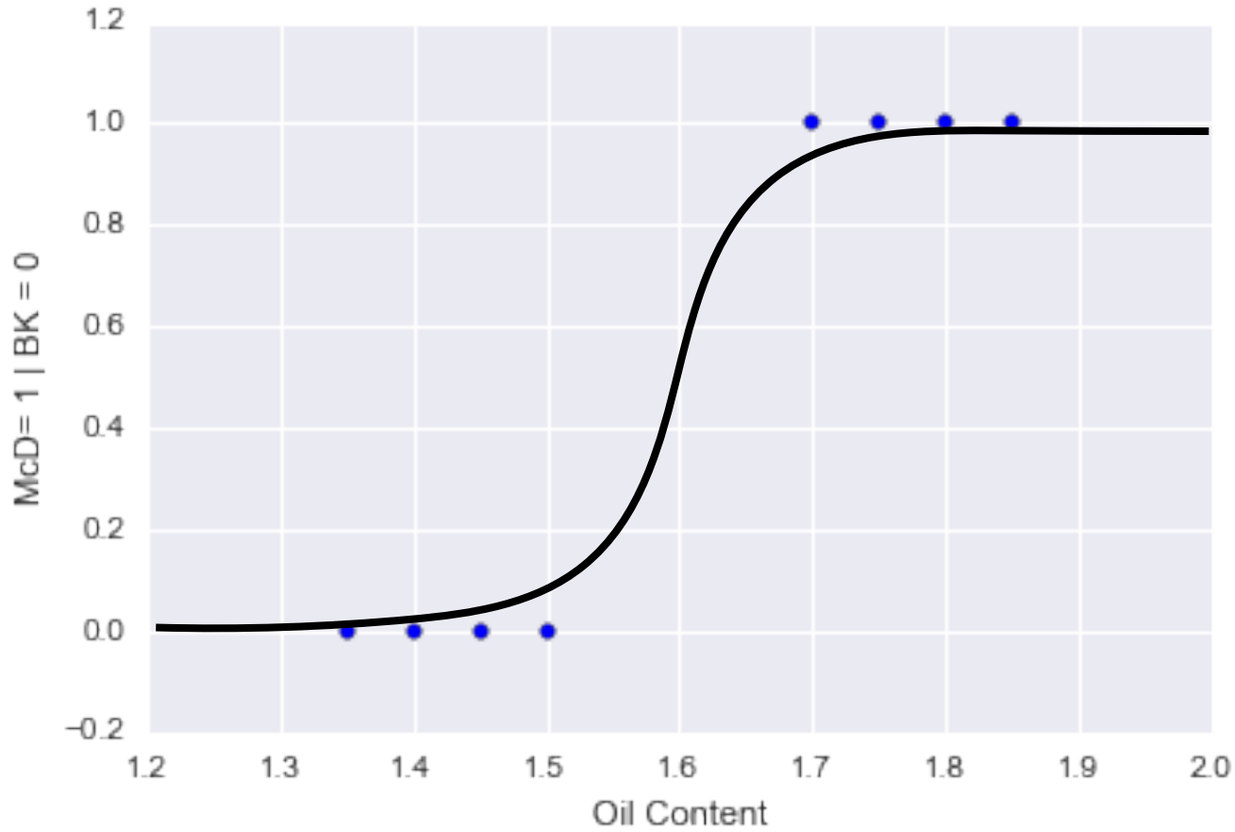


When z —> infinity
sigmoid(z) —> 1

When z —> -infinity
sigmoid(z) —> 0

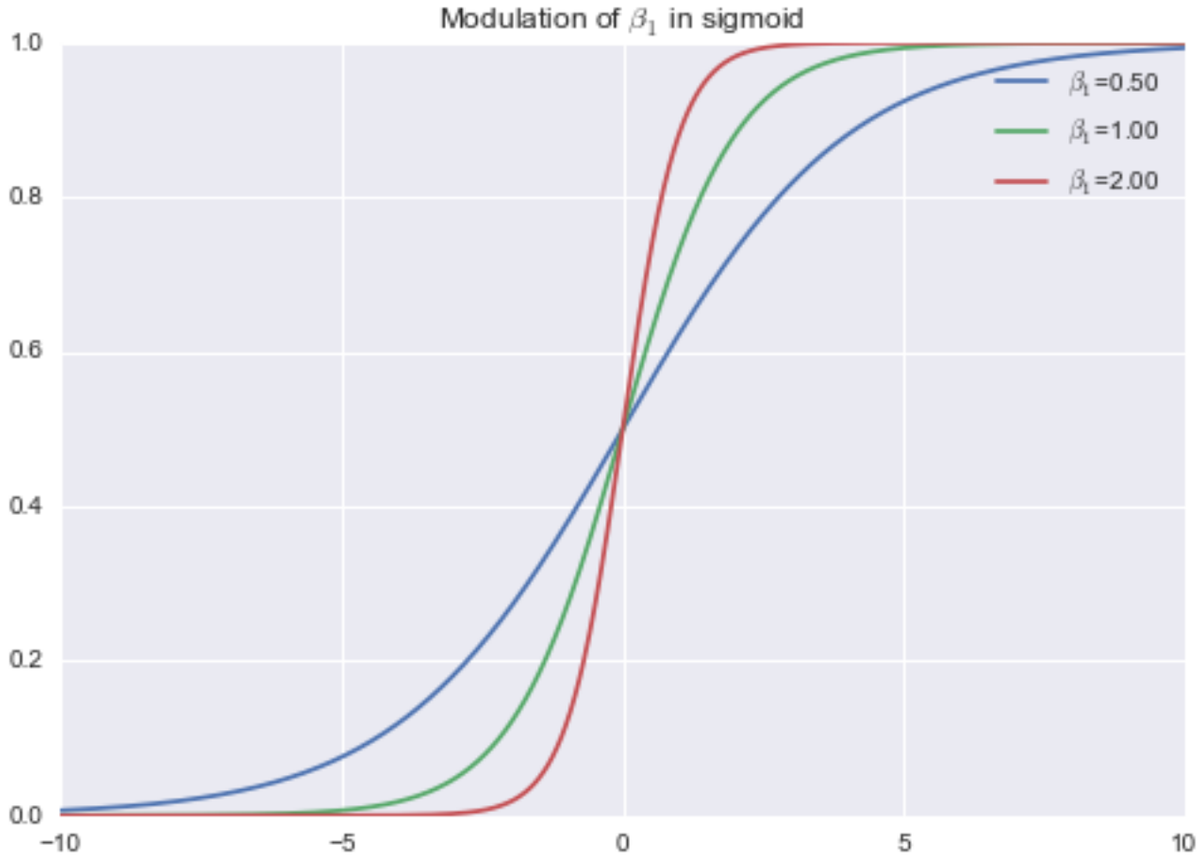
LOGISTIC REGRESSION

New Prediction Scheme: $$P(y) = \frac{1}{1 + e^{-(\alpha + \beta_1 x_1 + \ldots + \beta_n x_n)}}$$



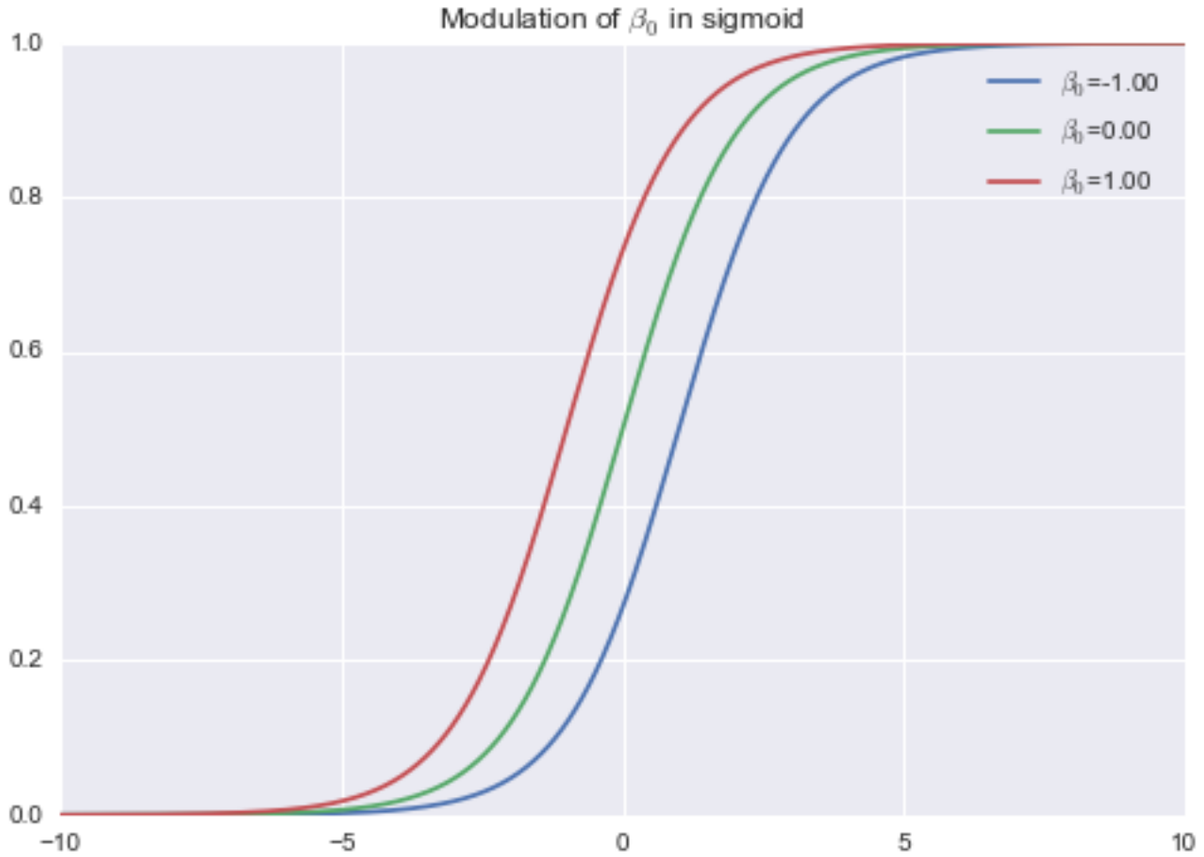
LOGISTIC REGRESSION

New Prediction Scheme: $$P(y) = \frac{1}{1 + e^{-(\alpha + \beta_1 x_1 + \ldots + \beta_n x_n)}}$$



LOGISTIC REGRESSION

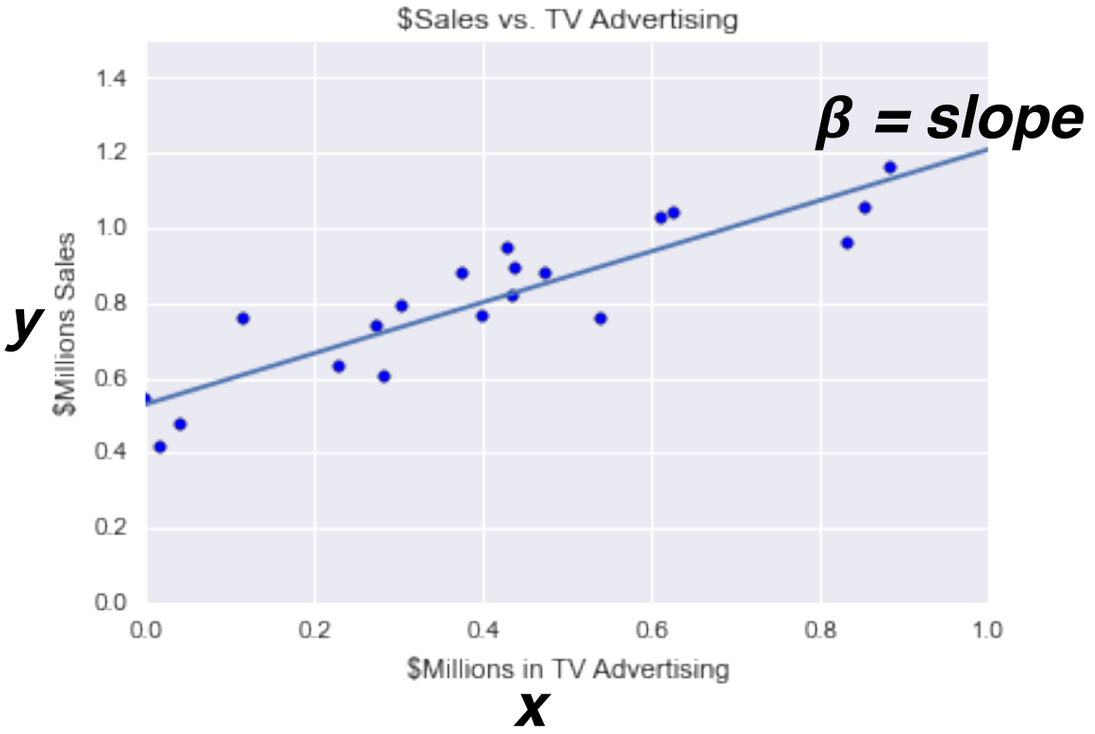
New Prediction Scheme: $$P(y) = \frac{1}{1 + e^{-(\alpha + \beta_1 x_1 + \ldots + \beta_n x_n)}}$$



I. PRECURSOR
II. LOGISTIC REGRESSION
III. INTERPRETING RESULTS

INTERPRETATIONS

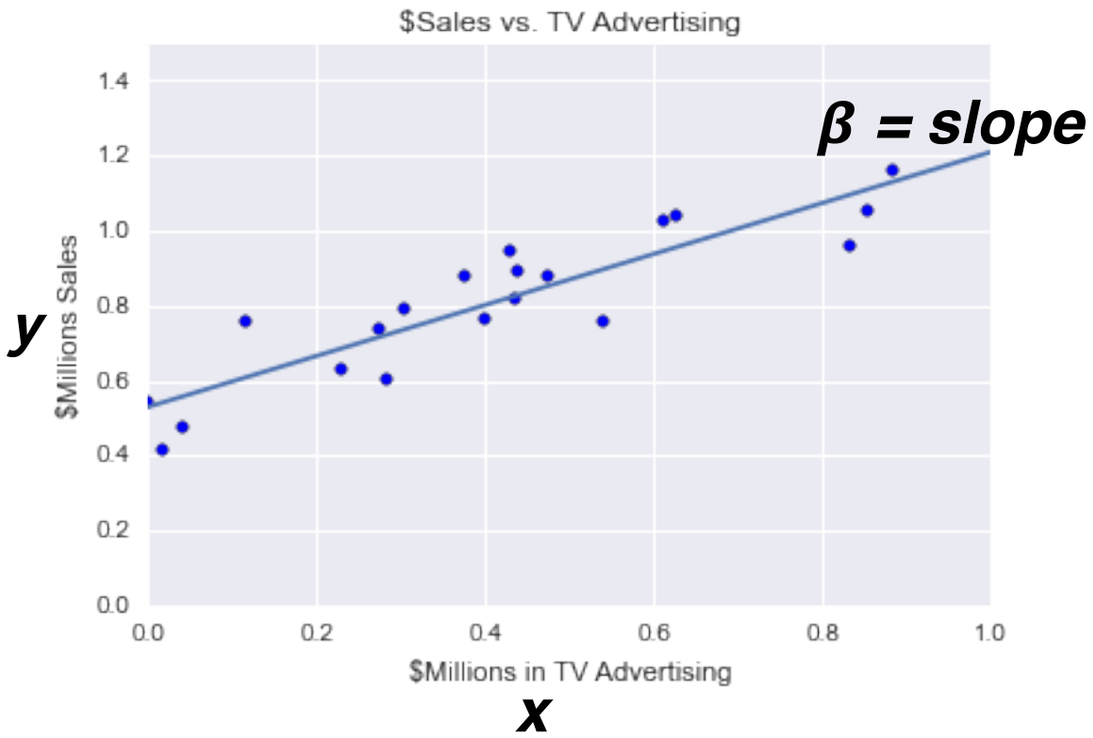
In **linear regression**, the parameter β represented the change in the response variable for a unit change in **X**



$$y = \alpha + \beta_1 x_1$$

INTERPRETATIONS

In **linear regression**, the parameter β represented the change in the response variable for a unit change in **X**



With each additional mil spent in TV advertising, $Sales goes up β

$$y = \alpha + \beta_1 x_1$$

INTERPRETATIONS

For a logistic regression, we can understand what β represents by rearranging our equation

$$P(y) = \frac{1}{1 + e^{-(\alpha+\beta_1 x_1+...+\beta_n x_n)}}$$

$$P(y)[1 + e^{-(\alpha+\beta_1 x_1+...+\beta_n x_n)}] = 1$$

$$P(y) + P(y)e^{-(\alpha+\beta_1 x_1+...+\beta_n x_n)} = 1$$

$$e^{-(\alpha+\beta_1 x_1+...+\beta_n x_n)} = \frac{1 - P(y)}{P(y)}$$

$$-(\alpha + \beta_1 x_1 + ... + \beta_n x_n) = \ln\left(\frac{1 - P(y)}{P(y)}\right)$$

$$-(\alpha + \beta_1 x_1 + ... + \beta_n x_n) = \ln\left(\frac{1 - P(y)}{P(y)}\right)$$

$$\alpha + \beta_1 x_1 + ... + \beta_n x_n = \ln\left(\frac{P(y)}{1 - P(y)}\right)$$

INTERPRETATIONS

β represents the change in the **logit function** for a unit change in x.
logit(p) = log(p / 1 - p)

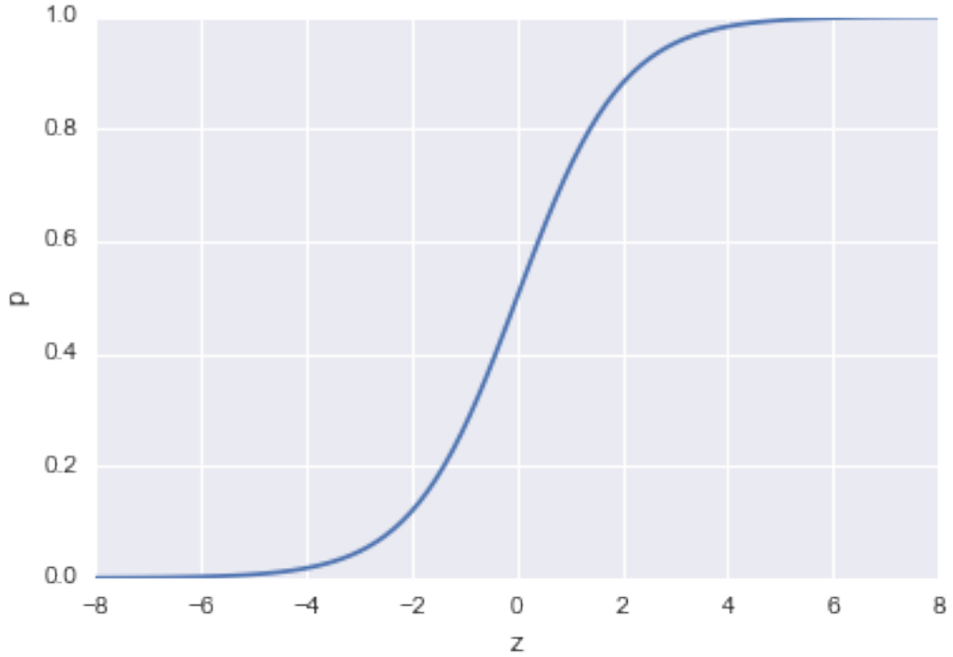
$$\alpha + \beta_1 x_1 + \ldots + \beta_n x_n = \ln\left(\frac{P(y)}{1 - P(y)}\right)$$

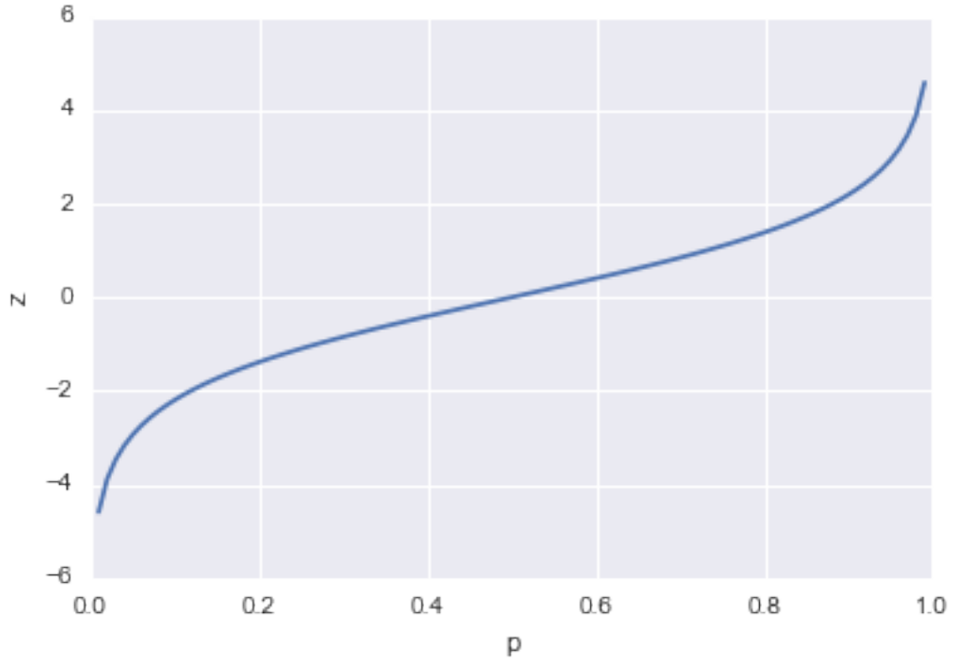
Logit function for P(y)

INTERPRETATIONS

β represents the change in the **logit function** for a unit change in x. The **logit** function is the inverse of the **logistic** function

$$p = \log istic(z) = \frac{1}{1+e^{-z}} \quad \longleftrightarrow \quad z = \log it(p) = \ln\left(\frac{p}{1-p}\right)$$

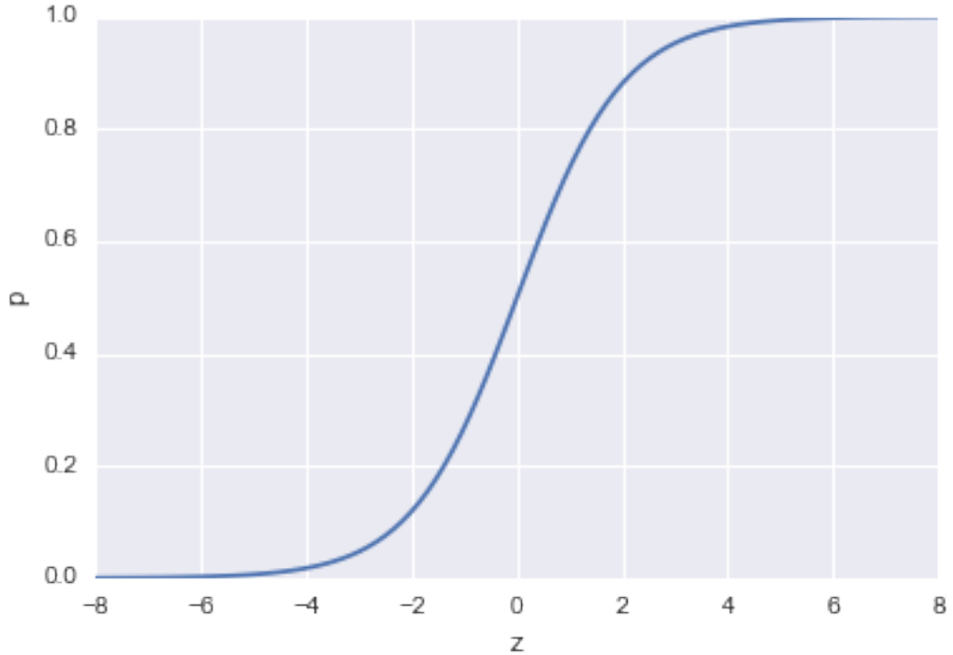


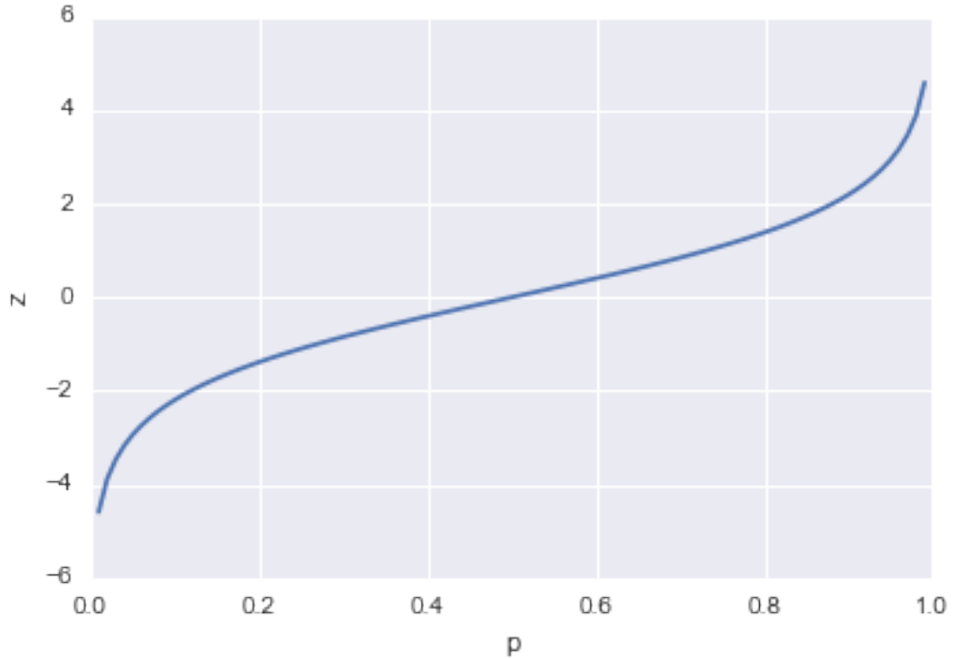


INTERPRETATIONS

The **logit** function is also called the **log odds function**

$$p = \log istic(z) = \frac{1}{1+e^{-z}} \quad \longleftrightarrow \quad z = \log it(p) = \ln\left(\frac{p}{1-p}\right)$$





INTERPRETATIONS

In **linear regression**, the parameter β represented the change in the response variable for a unit change in **X**

In **logistic regression**, the parameter β represents the change in the **log odds** for a unit change in **X**

$$\alpha + \beta_1 x_1 + ... + \beta_n x_n = \ln\left(\frac{P(y)}{1 - P(y)}\right)$$

Interpreting this change in the logic function requires us to define what odds are

INTERPRETATIONS

The **odds** of an event are given by the ratio of the probability by its compliment:

$$\text{odds}(p) = \frac{p}{1-p}$$

The **odds** tell you how much more likely an event is to happen compared to the event not happening

$$\text{odds of P(A happens)} = \frac{P(\text{A happens})}{P(\text{A doesn't happen })}$$

INTERPRETATIONS

In **linear regression**, the parameter β represented the change in the response variable for a unit change in **X**

In **logistic regression**, the parameter β represents the change in the **log odds** for a unit change in **X**

$$e^{\alpha+\beta_1 x_1+\ldots+\beta_n x_n} = \frac{P(y)}{1-P(y)}$$

Rearranged from earlier

$$e^{\alpha} e^{\beta_1 x_1} \ldots e^{\beta_n x_n} = \frac{P(y)}{1-P(y)}$$

Expanded exponent

INTERPRETATIONS

Let's say β = log(2).

This means that a unit increase in x leads to double increase in the odds of P(y)! P(y) is now twice as likely than the compliment.

$$e^{\alpha} e^{\beta_1 x_1} \ldots e^{\beta_n x_n} = \frac{P(y)}{1 - P(y)}$$

INTERPRETATIONS

Let's say β = log(2).

This means that a unit increase in x leads to a double increase in the odds of P(y)! P(y) is now twice as likely than the compliment.

$$e^{\alpha} e^{\beta_1 x_1} \ldots e^{\beta_n x_n} = \frac{P(y)}{1 - P(y)}$$

EX: For β = log(2), a Fry is TWICE as likely to be from McDonalds than Burger King for a unit increase in oil content

INTERPRETATIONS

Let's say β = log(2).

This means that a unit increase in x leads to a double increase in the odds of P(y)! P(y) is now twice as likely than the compliment.

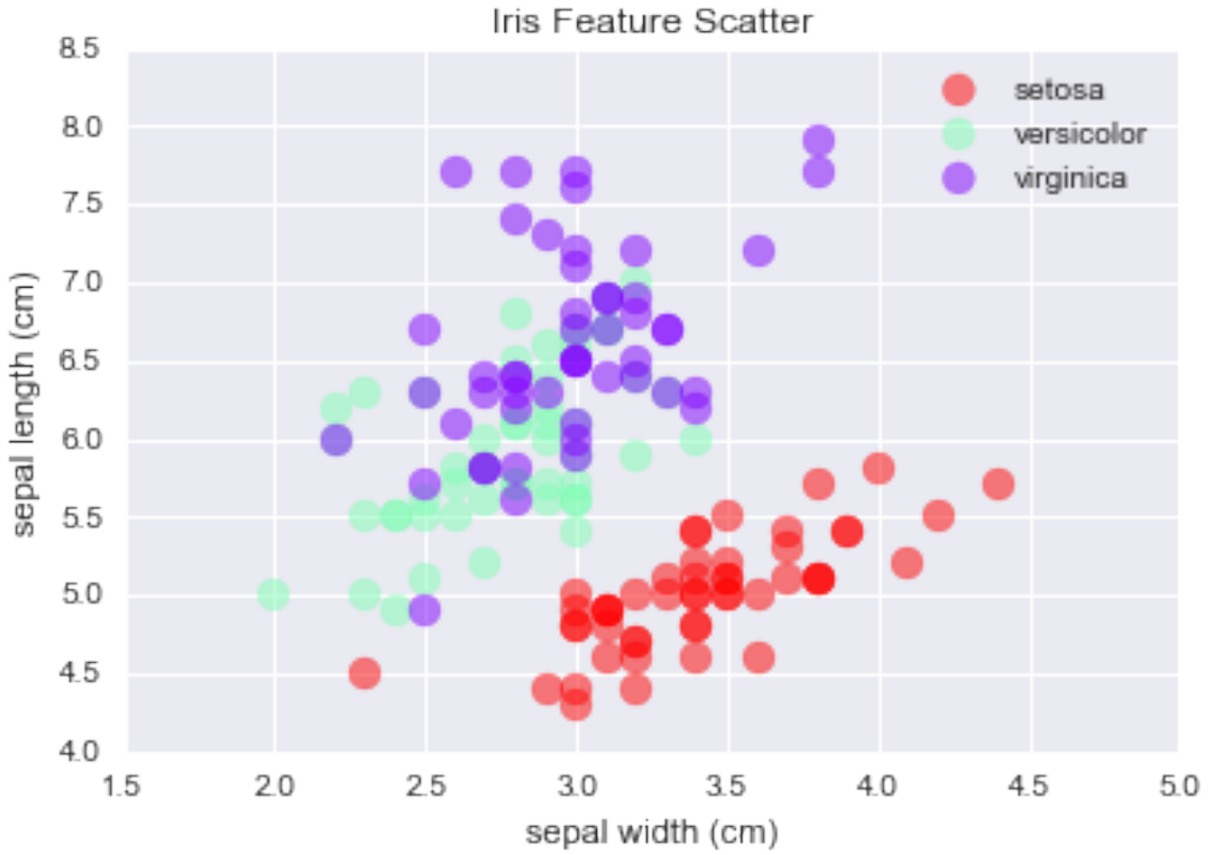
$$e^{\alpha} e^{\beta_1 x_1} \ldots e^{\beta_n x_n} = \frac{P(y)}{1 - P(y)}$$

EX: For β = 5, a Fry is e^5 times as likely to be from McDonalds than Burger King for a unit increase in oil content. (i.e. A LOT MORE LIKELY)

I. PRECURSOR
II. LOGISTIC REGRESSION
III. INTERPRETING RESULTS
IV. DECISION BOUNDARIES

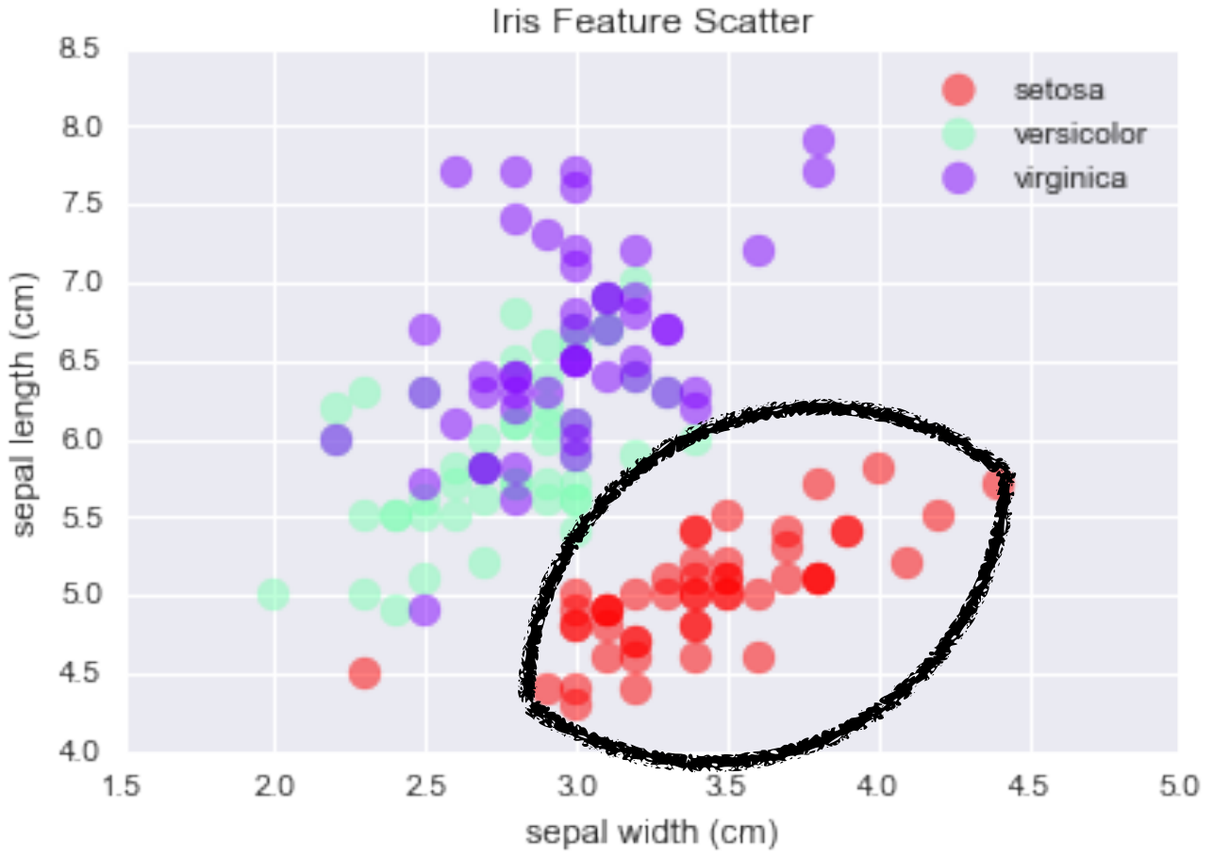
DECISION BOUNDARIES

Coming back to the Iris dataset…



DECISION BOUNDARIES

Let's classify using ***logistic regression,*** whether a iris is a "*setosa*" or "*other*", based on *sepal_width* and *sepal_length*.



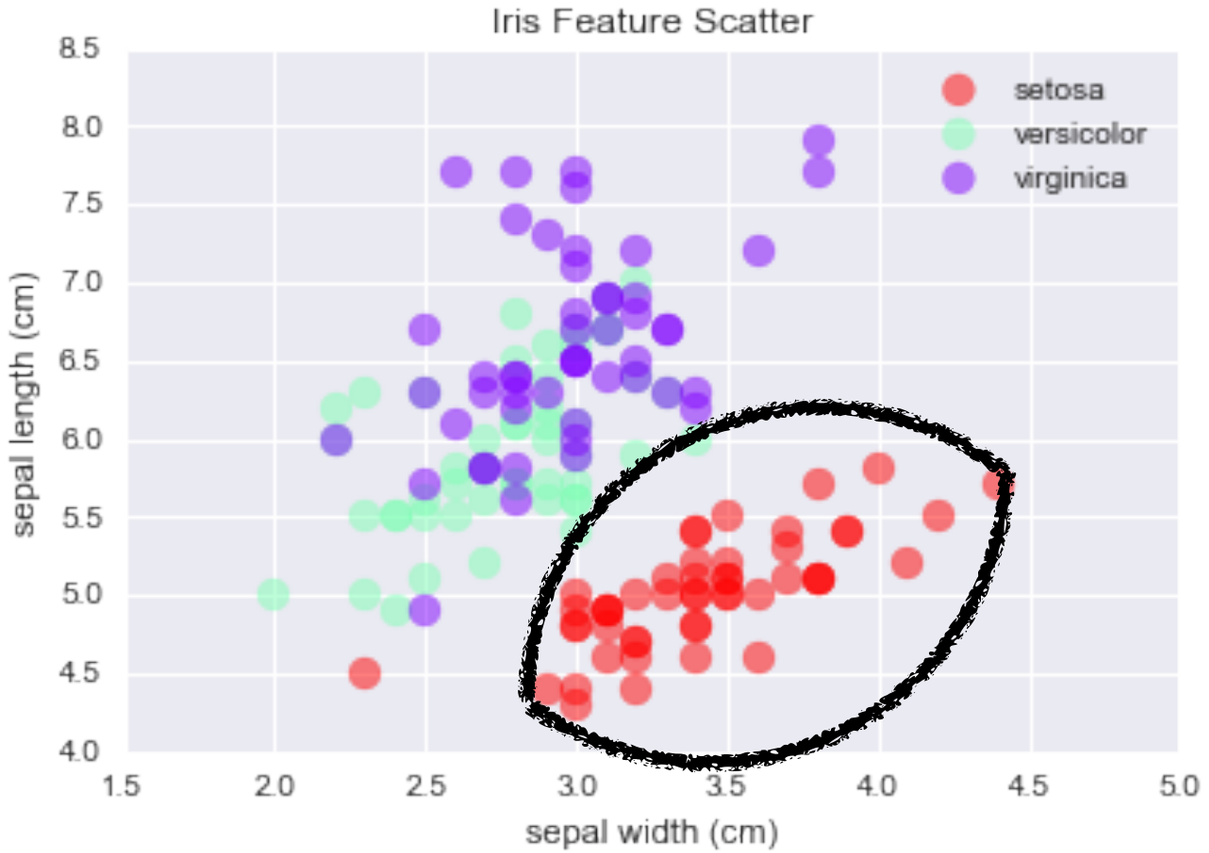
DECISION BOUNDARIES

Let's classify using ***logistic regression,*** whether a iris is a "*setosa*" or "*other*", based on *sepal_width* and *sepal_length*.

$x_1 = \text{sepal width}$

$x_2 = \text{sepal length}$

$y = \text{P(setosa)}$



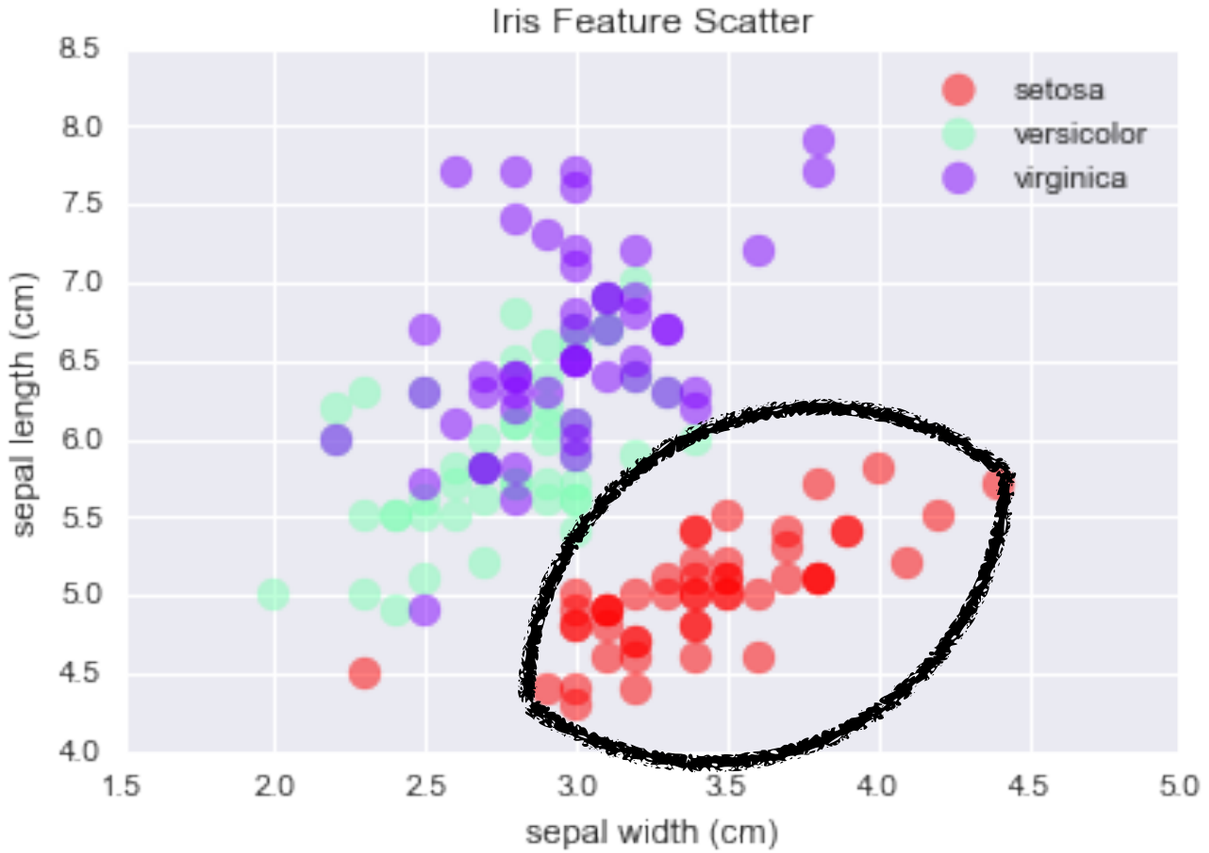
DECISION BOUNDARIES

$$\text{P(setosa)} = \text{sigmoid}(\ \alpha + \beta_1 x_1 + \beta_2 x_2\ )$$

$x_1 = \text{sepal width}$

$x_2 = \text{sepal length}$

$y = \text{P(setosa)}$



DECISION BOUNDARIES

$$\text{P(setosa)} = \text{sigmoid}(\ \alpha + \beta_1 x_1 + \beta_2 x_2\ )$$

If we set our cutoff at 50%, we predict P(setosa) if and only if P >= 0.5 (i.e. if sigmoid(…) >= 0.5

The case when P = 0.5 denotes the **decision boundary** between our predictions

DECISION BOUNDARIES

$$1/2 = \text{sigmoid}(\ \alpha + \beta_1 x_1 + \beta_2 x_2\ )$$

If we set our cutoff at 50%, we predict P(setosa) if and only if P >= 0.5 (i.e. if sigmoid(…) >= 0.5

The case when P = 0.5 denotes the **decision boundary** between our predictions

DECISION BOUNDARIES

$$1/2 = \text{sigmoid}(\ \alpha + \beta_1 x_1 + \beta_2 x_2\ )$$

If we set our cutoff at 50%, we predict P(setosa) if and only if P >= 0.5 (i.e. if sigmoid(…) >= 0.5

The case when P = 0.5 denotes the **decision boundary** between our predictions.

Remember the equation for a sigmoid. If we set the sigmoid above to 0.5, this implies that z (or our linear equation) has to equal 0

$$\text{sigmoid(z)} = \frac{1}{1+e^{-z}} \longrightarrow 0.5 = \frac{1}{1+e^{-z}} \longrightarrow 0.5 = \frac{1}{1+e^{(z==0)}}$$

DECISION BOUNDARIES

In general, for a *logistic regression model* given by:

$$P(y) = \frac{1}{1 + e^{-(\alpha + \beta_1 x_1 + ... + \beta_n x_n)}}$$

It's **decision boundary** is given by the equation

$$\alpha + \beta_1 x_1 + ... + \beta_n x_n = 0$$

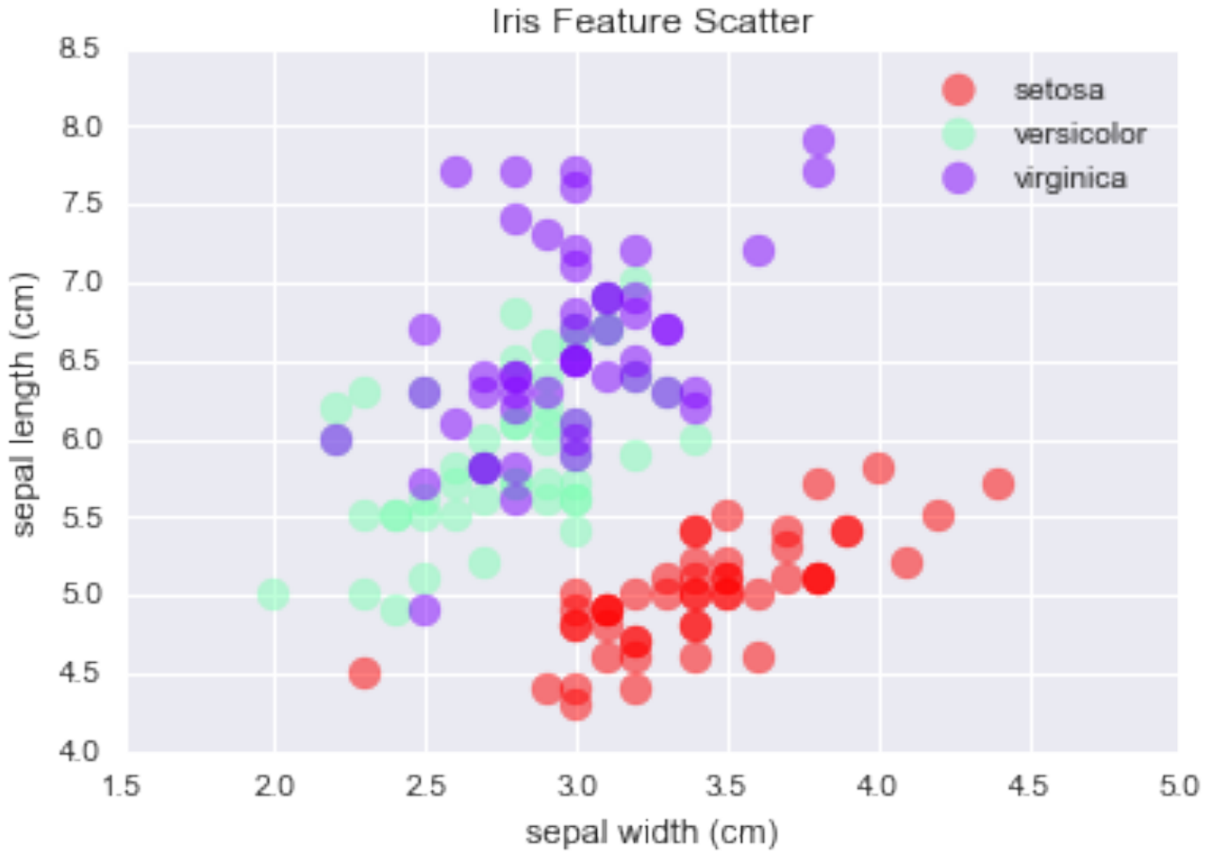
DECISION BOUNDARIES

$$\text{P(setosa)} = \text{sigmoid}(\ \alpha + \beta_1 x_1 + \beta_2 x_2\ )$$

$x_1 = \text{sepal width}$

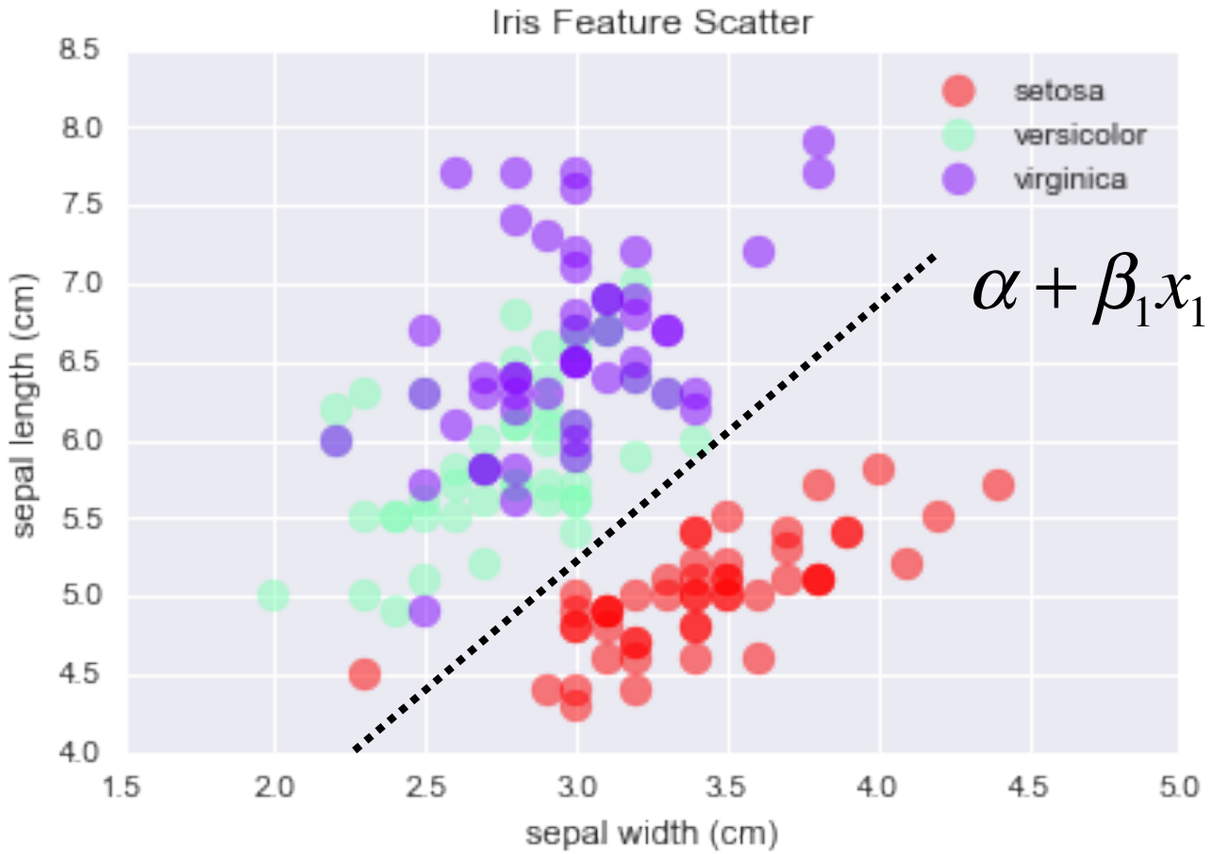
$x_2 = \text{sepal length}$

$y = \text{P(setosa)}$



DECISION BOUNDARIES

$$\text{P(setosa)} = \text{sigmoid}(\ \alpha + \beta_1 x_1 + \beta_2 x_2\ )$$



$x_1 = \text{sepal width}$

$x_2 = \text{sepal length}$

$y = \text{P(setosa)}$

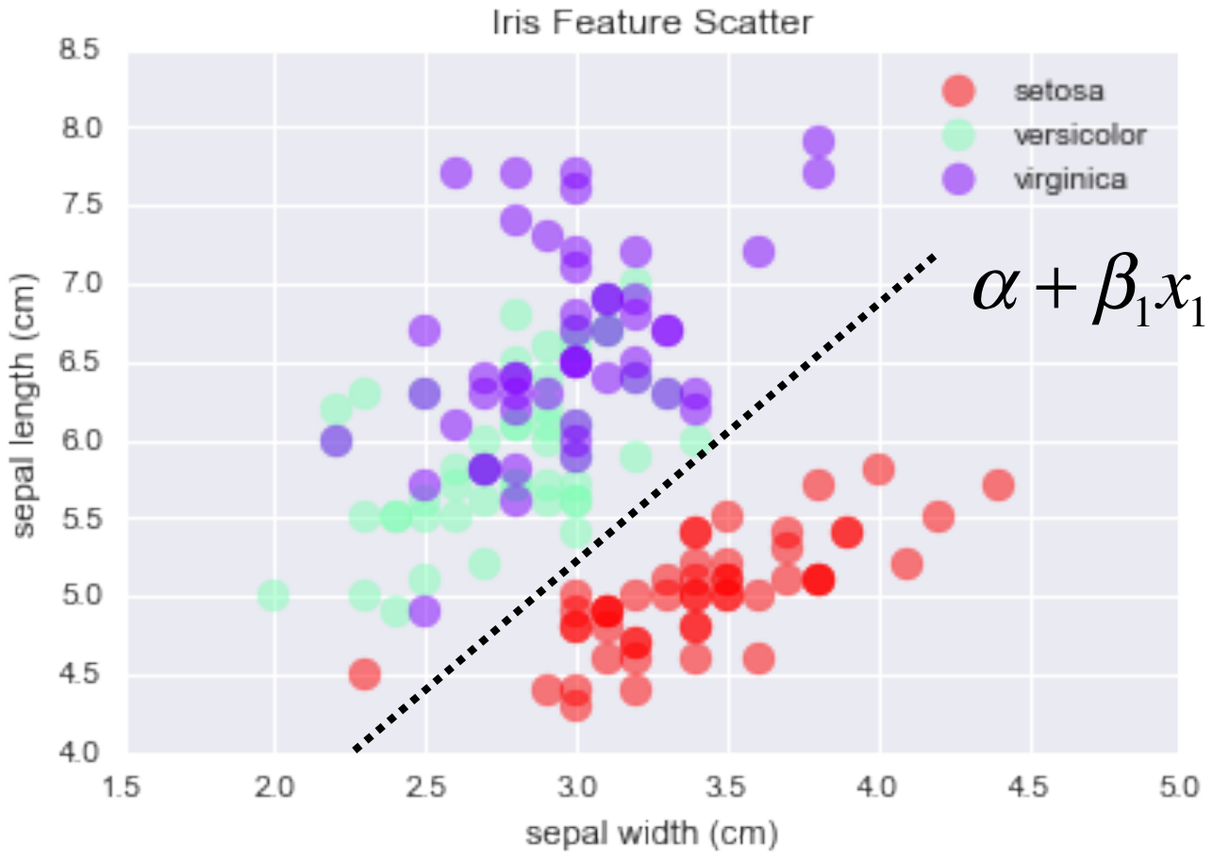
DECISION BOUNDARIES

The line **separates** the two classes!

$x_1 = \text{sepal width}$

$x_2 = \text{sepal length}$

$y = \text{P(setosa)}$



$$\alpha + \beta_1 x_1 + \beta_2 x_2 = 0$$

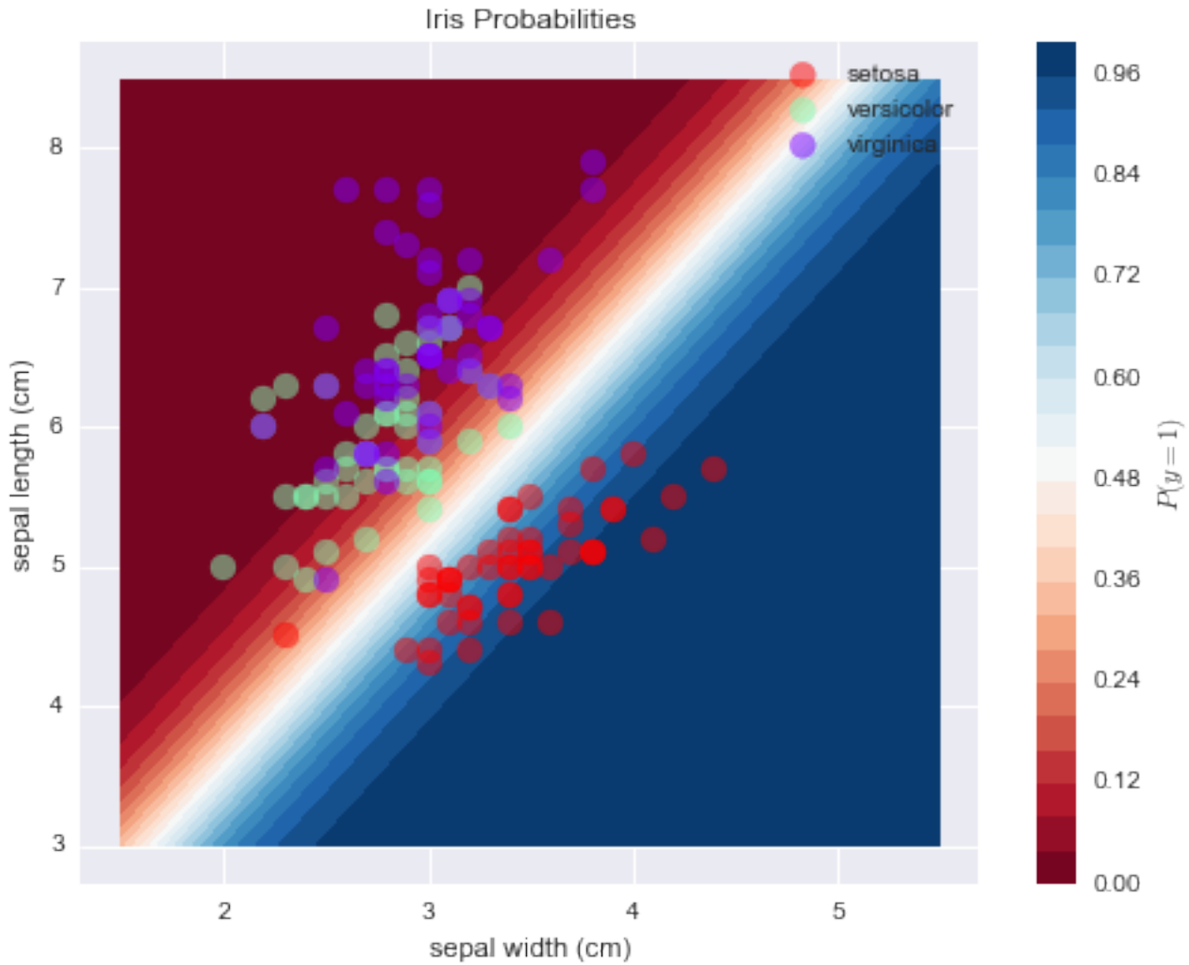
DECISION BOUNDARIES

Notice how quickly the probabilities increase

$x_1 = \text{sepal width}$

$x_2 = \text{sepal length}$

$y = \text{P(setosa)}$



LOGISTIC REGRESSION

The logistic regression model is an extension of the linear regression model, with a couple of important differences.

The first difference is in what the outcome (or predicted) variable is

The second difference is in the **error term**

LOGISTIC REGRESSION

In linear regression, our prediction was:

$$y = \alpha + \beta_1 x_1 + \varepsilon$$

The error was distributed according to a **gaussian** distribution

LOGISTIC REGRESSION

In logistic regression, our prediction is:

$$y = \frac{1}{1 + e^{-(\alpha + \beta_1 x_1)}} + \varepsilon$$

The error is now distributed according to a **binomial** distribution

(i.e. the same coin distribution representing a coin flip. Think about why this is)

LOGISTIC REGRESSION

In fact, depending on how we represent the error distribution, we can model many prediction problems using different **link** functions

https://en.wikipedia.org/wiki/Generalized_linear_model

LAB TIME

I. PRECURSOR
II. LOGISTIC REGRESSION
III. INTERPRETING RESULTS
IV. DECISION BOUNDARIES
V. EVALUATING CLASSES

EVALUATING CLASSES

So far, we've evaluated classification models, according to the following metric of **accuracy**:

$$accuracy(\text{clf}) = \frac{\#\text{ correct}}{\#\text{ total}}$$

EVALUATING CLASSES

So far, we've evaluated classification models, according to the following metric of **accuracy**:

$$accuracy(\text{clf}) = \frac{\#\ \text{correct}}{\#\ \text{total}}$$

Q: When is this a bad metric?

EVALUATING CLASSES

So far, we've evaluated classification models, according to the following metric of **accuracy**:

$$accuracy(\text{clf}) = \frac{\#\ \text{correct}}{\#\ \text{total}}$$

Q: When is this a bad metric?
A: When predicting rare or very likely events

EVALUATING CLASSES

So far, we've evaluated classification models, according to the following metric of **accuracy**:

$$accuracy(\text{clf}) = \frac{\#\ \text{correct}}{\#\ \text{total}}$$

Q: When is this a bad metric?
A: When predicting rare or very likely events

…

Q: What else can we use?

CONFUSION MATRIX

We’re going to introduce the concept of a *Confusion Matrix* (or contingency table) which is aptly named…

CONFUSION MATRIX

observations \ *predictions*	Yes	No
Yes		
No		

CONFUSION MATRIX

	predictions	
observations	Yes	No
Yes	***true positive***	
No		

CONFUSION MATRIX

observations \ ***predictions***	Yes	No
Yes	***true positive***	
No		***true negative***

CONFUSION MATRIX

		predictions	
	observations	Yes	No
	Yes	***true positive***	***false negative***
	No	***false positive***	***true negative***

CONFUSION MATRIX

Accuracy = (TP + TN) / all

	predictions	
observations	Yes	No
Yes	***TP***	*FN*
No	*FP*	***TN***

Of all the samples, what percentage of my predictions were correct?

CONFUSION MATRIX

Accuracy = (TP + TN) / all

Precision = TP / (TP + FP)

	predictions	
observations	Yes	No
Yes	***TP***	*FN*
No	*FP*	*TN*

Of all the times I said yes, what percentage was I actually correct?
i.e. If I cry wolf a million times, and the wolf comes once, I have bad precision

CONFUSION MATRIX

Accuracy = (TP + TN) / all

Precision = TP / (TP + FP)

Recall = TP / (TP + FN)
(aka **hit rate** or **sensitivity**)

	predictions	
observations	Yes	No
Yes	***TP***	*FN*
No	*FP*	*TN*

*"Of all the cases where the condition was true, what percentage did I **hit** or predict it correctly?"*
i.e. Every time an event happens, how often did I predict it would happen?

CONFUSION MATRIX

Accuracy = (TP + TN) / all

Precision = TP / (TP + FP)

Recall = TP / (TP + FN)
(aka **hit rate** or **sensitivity**)

F1 score = 2 * P * R / (P + R)

		predictions
observations	Yes	No
Yes	***TP***	***FN***
No	***FP***	***TN***

Weighted average of precision and recall, between 0 and 1

CONFUSION MATRIX

Accuracy = (TP + TN) / all

Precision = TP / (TP + FP) = % correct **cases** of all **positive predictions**

Recall = TP / (TP + FN) = % correct **predictions** of all **positive cases**

F1 score = 2 * P * R / (P + R)

CONFUSION MATRIX

Q: When do you want a high recall model?

A: When the cost of a false positive is low and the cost of a false negative is high

CONFUSION MATRIX

Q: When do you want a high recall model?

A: When the cost of a false positive is low and the cost of a false negative is high

i.e. if you predict catastrophic events. You definitely do not want to miss out on these potential events because the cost of missing them is high

CONFUSION MATRIX

Q: When do you want a high precision model?

A: When the cost of a false positive is high

CONFUSION MATRIX

Q: When do you want a high precision model?

A: When the cost of a false positive is high

i.e. Predicting who should receive a very expensive and complicated medical treatment.

Having a high number of false positives would be burdensome and expensive!

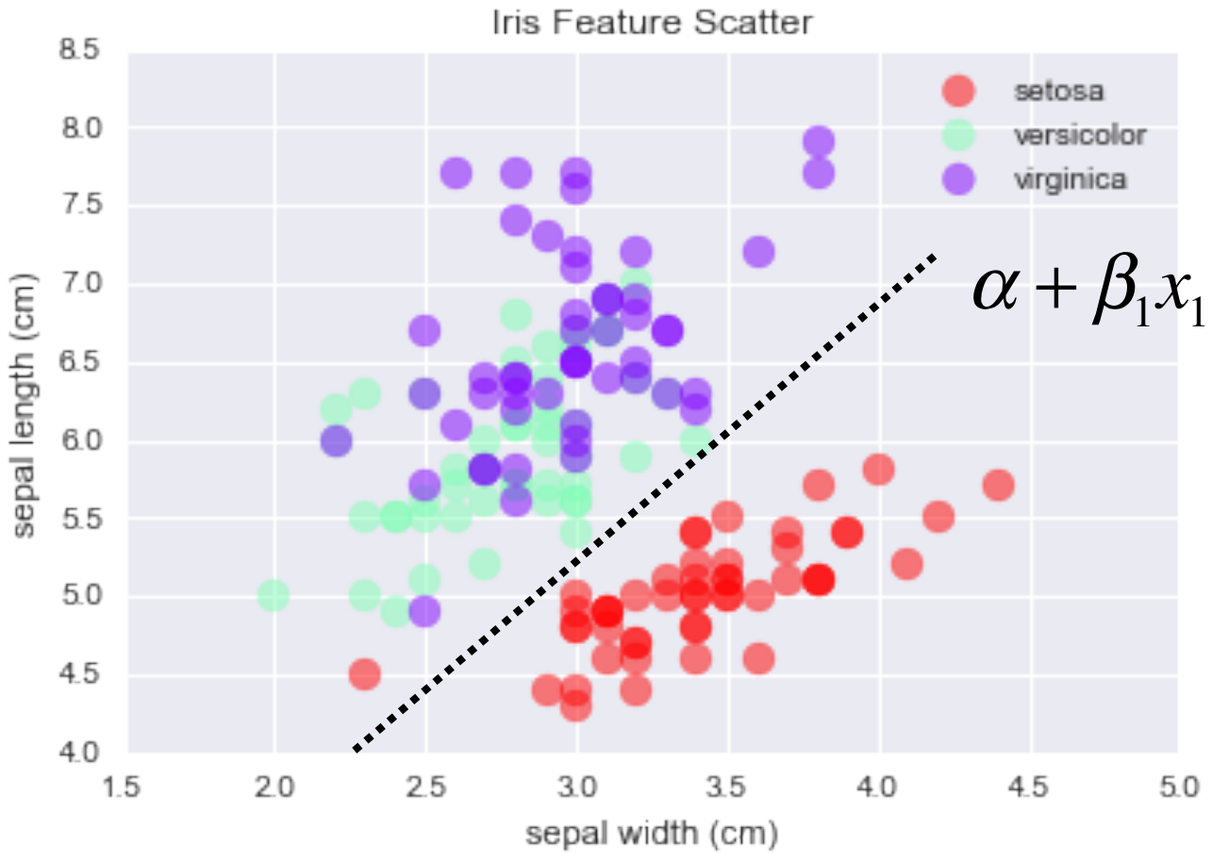
MODULATING THE PROBABILITY CUTOFF

The line **separates** the two classes at P(x) = 0.5
But…what if we reduced the threshold?

$x_1 = \text{sepal width}$

$x_2 = \text{sepal length}$

$y = \text{P(setosa)}$



$$\alpha + \beta_1 x_1 + \beta_2 x_2 = 0$$

LAB TIME

THAT'S IT!

- Exit Tickets: DAT1 - Lesson 8 - Logistic Regression
- Homework 5 is due Jan 20
- Milestone 2 is due Jan 25
- You'll have partial opportunity to work on it during class Jan 20th (as well as review)