Statistics and Regression

Ryan Amos

Slides adapted from Wells Santo (AI4ALL) with content from Becca Roelofs from BAIR AI4ALL

Recap

- **Supervised Learning:**

Recap

- **Supervised Learning:** learning from labelled examples

Recap

- **Supervised Learning:** learning from labelled examples
- **Classification**:

Recap

- **Supervised Learning:** learning from labelled examples
- **Classification**: a type of supervised learning problem where we want to determine what class each example belongs to

Recap

- **Supervised Learning:** learning from labelled examples

- **Classification**: a type of supervised learning problem where we want to determine what class each example belongs to

- **Naive Bayes**:

Recap

- **Supervised Learning:** learning from labelled examples

- **Classification**: a type of supervised learning problem where we want to determine what class each example belongs to

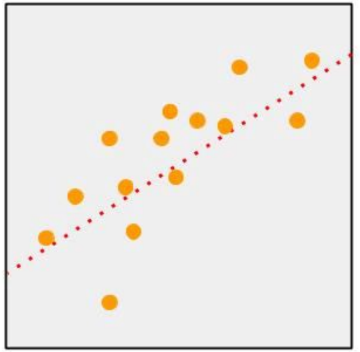
- **Naive Bayes**: a classification algorithm that uses Bayes' Theorem from probability to determine how likely an example belongs to a class, with an independence assumption

Recap

- **Supervised Learning:** learning from labelled examples
- **Classification**: a type of supervised learning problem where we want to determine what class each example belongs to
- **Naive Bayes**: a classification algorithm that uses Bayes' Theorem from probability to determine how likely an example belongs to a class, with an independence assumption
- But what if we want to predict something more specific, like an actual real number output?

The Regression Task

- **Regression**: A type of supervised learning problem where given an input we want to predict a specific output value

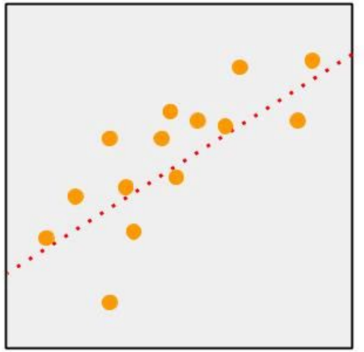


Regression

The Regression Task

- **Regression**: A type of supervised learning problem where given an input we want to predict a specific output value

- Unlike classification, our possible predictions are *any real number*



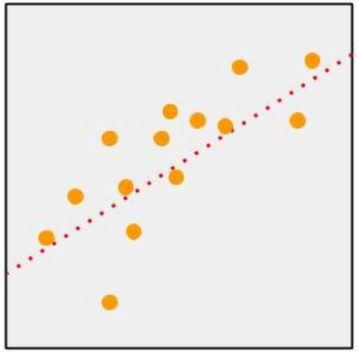
Regression

The Regression Task

- **Regression**: A type of supervised learning problem where given an input we want to predict a specific output value

- Unlike classification, our possible predictions are *any real number*

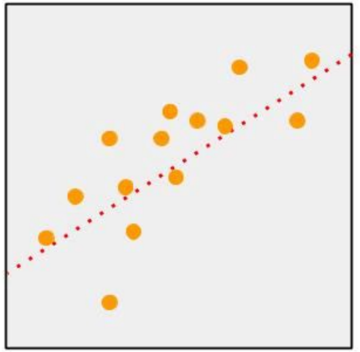
- Examples:
    - What will the value of a home in California be in 2020?
    - What will the temperature be tomorrow?
    - How likely will someone click on an ad on a website?



Regression

The Regression Task

- Just like classification, there are many different algorithms that attempt to solve the regression task, in their own particular ways



Regression

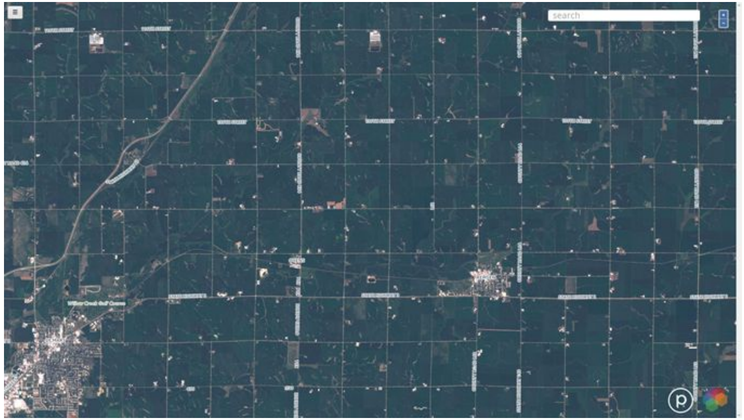
Regression Application

REPORT SCIENCE TECH

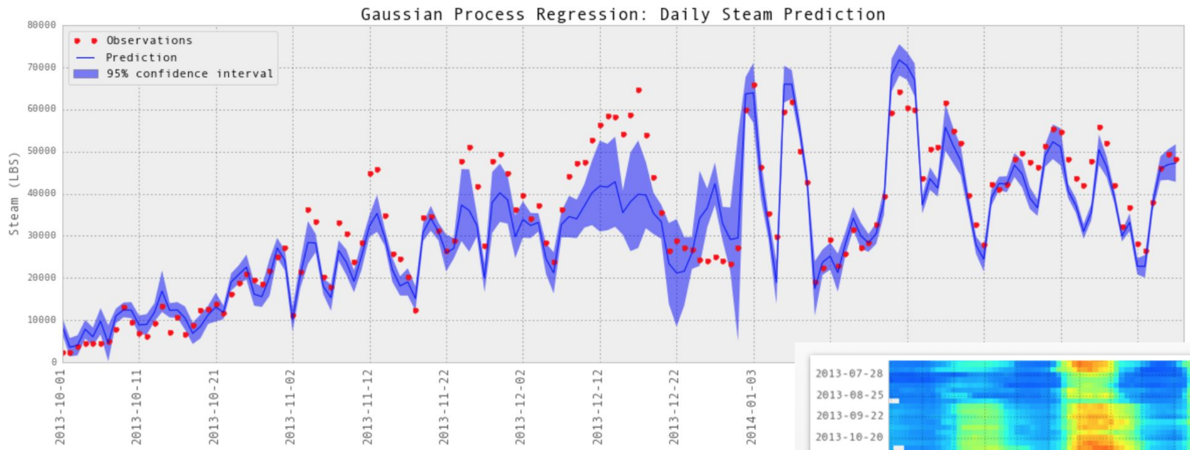
This startup uses machine learning and satellite imagery to predict crop yields

Artificial intelligence + nanosatellites + corn

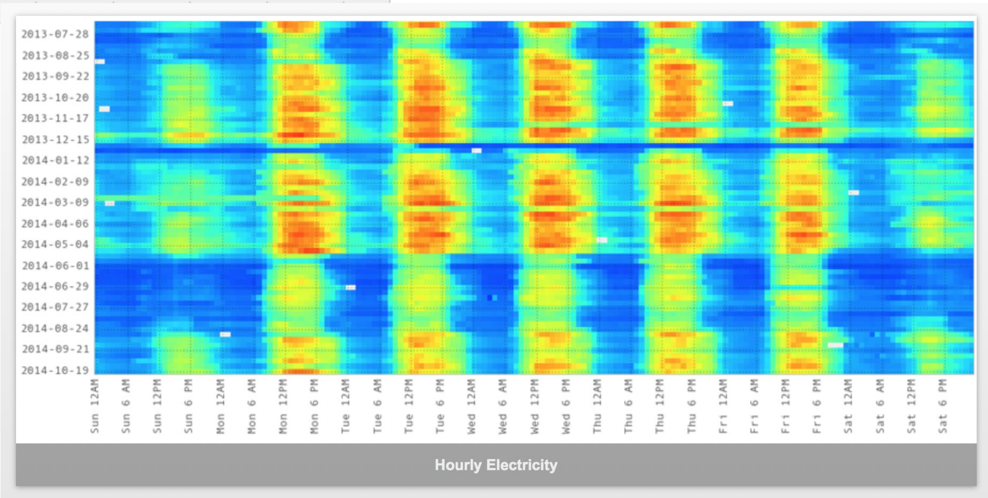
By Alex Brokaw | Aug 4, 2016, 10:22am EDT



Regression Application



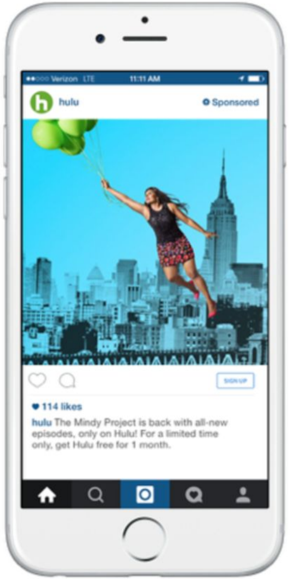
Building Energy Consumption Prediction



Regression Example

How many likes will my Instagram post get?

- What is the label?
- What are some features we might collect?



Small Group Activity

What are some uses of regression that you can think of?

In these cases:

- What are the examples?
- What are the labels?

But first… Statistics!

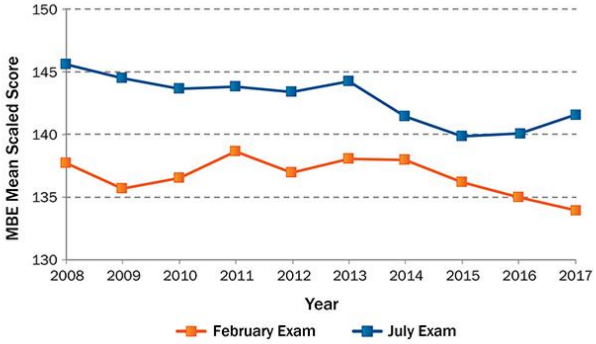
- Regression is a form of analysis that comes from the mathematical field of **statistics**

- Before getting to our machine learning algorithms for learning regression, let’s review some statistics!

Statistics

Statistics

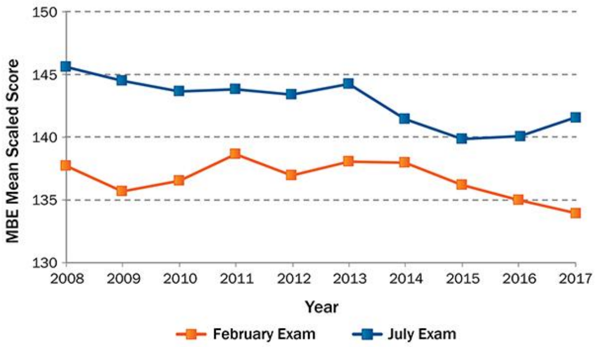
- A branch of mathematics that deals with collecting, analyzing, interpreting, presenting, and organizing data



Statistics

- A branch of mathematics that deals with collecting, analyzing, interpreting, presenting, and organizing data

- We use statistics to help understand trends in large quantities of data

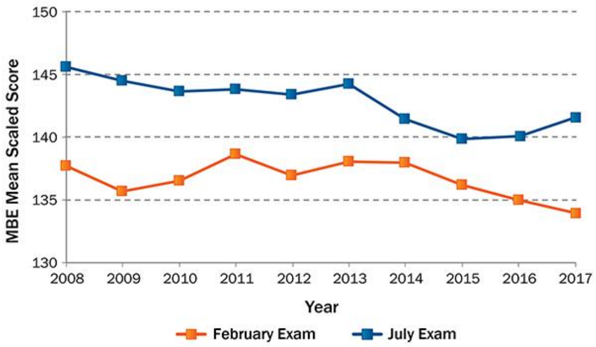


Statistics

- A branch of mathematics that deals with collecting, analyzing, interpreting, presenting, and organizing data

- We use statistics to help understand trends in large quantities of data

- Statistics is based on probability, but applied to large quantities of related data (often called a **population**)



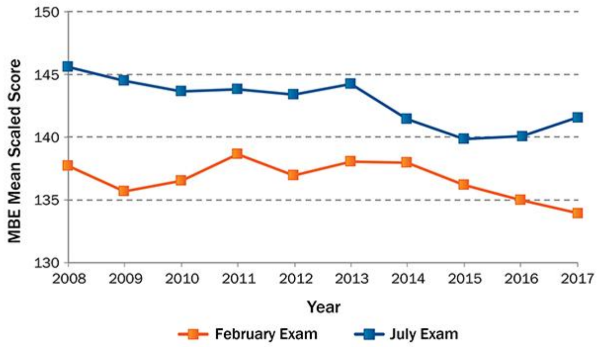
Statistics

- A branch of mathematics that deals with collecting, analyzing, interpreting, presenting, and organizing data

- We use statistics to help understand trends in large quantities of data

- Statistics is based on probability, but applied to large quantities of related data (often called a **population**)

- We often ask the question, if there were (infinitely) many events, what would our data look like?



Probability Distributions

- Imagine you are drawing a number at random from 1 to 100. What numbers do you think would be drawn most frequently?

Probability Distributions

- Imagine you are drawing a number at random from 1 to 100. What numbers do you think would be drawn most frequently?
- Would numbers near 50 get drawn more often?

Probability Distributions

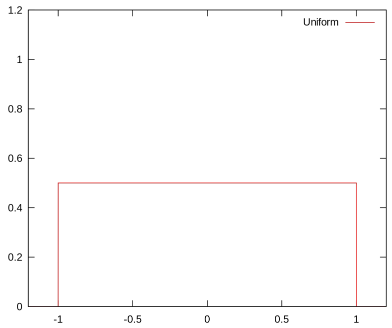
- Imagine you are drawing a number at random from 1 to 100. What numbers do you think would be drawn most frequently?
- Would numbers near 50 get drawn more often?
- Do you expect that it is equally likely for any number to be drawn?

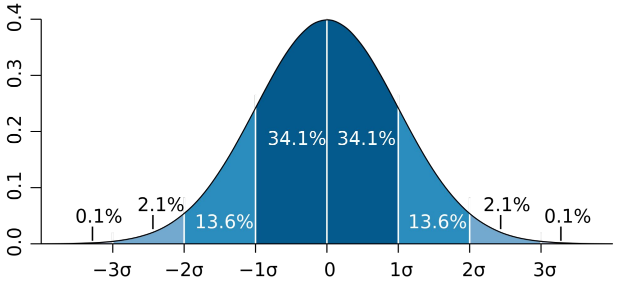
Probability Distributions

- Imagine you are drawing a number at random from 1 to 100. What numbers do you think would be drawn most frequently?
- Would numbers near 50 get drawn more often?
- Do you expect that it is equally likely for any number to be drawn?
- The answer: It depends!

Probability Distributions

- A **probability distribution** describes how likely certain events in your population will occur

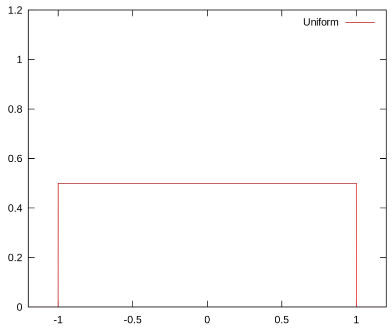


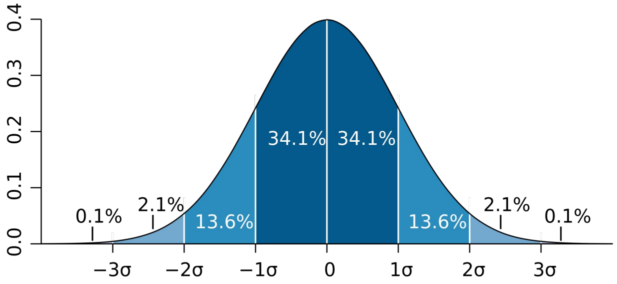


Probability Distributions

- A **probability distribution** describes how likely certain events in your population will occur

- We use distributions to present our data after we have observed lots of examples:
  - Rolled a die thousands of time
  - Gathered weather report data over a decade

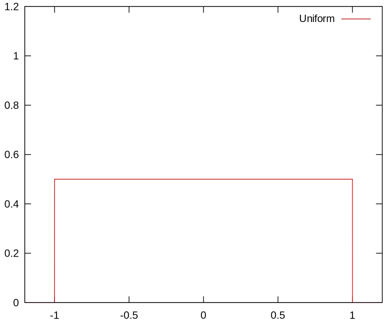




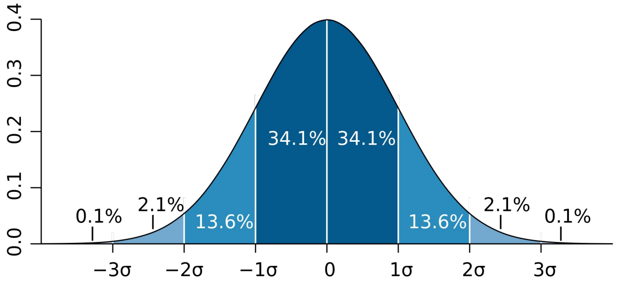
Probability Distributions

- A **probability distribution** describes how likely certain events in your population will occur

- We use distributions to present our data after we have observed lots of examples:
  - Rolled a die thousands of times
  - Gathered weather report data over a decade



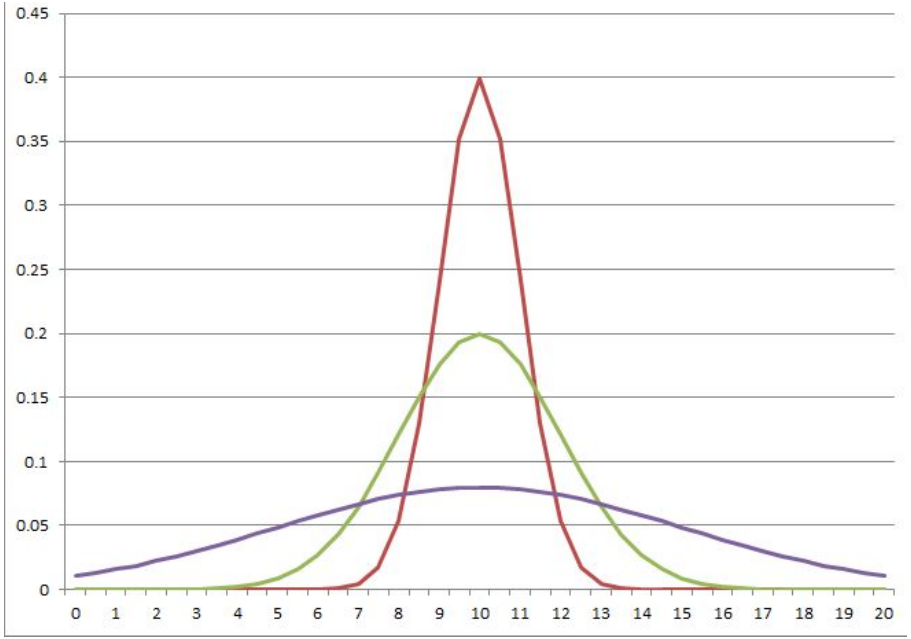
Uniform



Normal

Variance

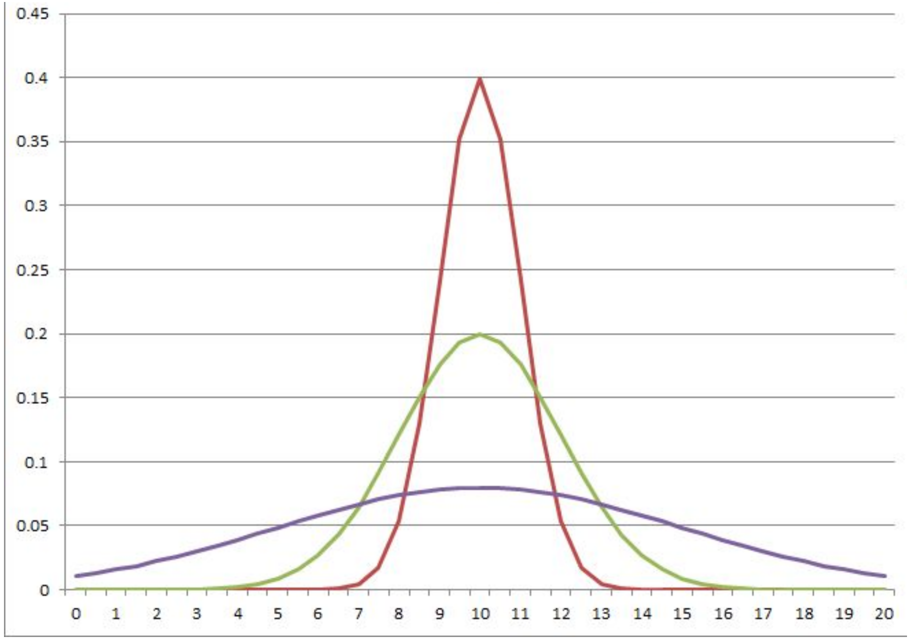
- Does knowing the mean of a distribution tell us everything we want to know about it?



Variance

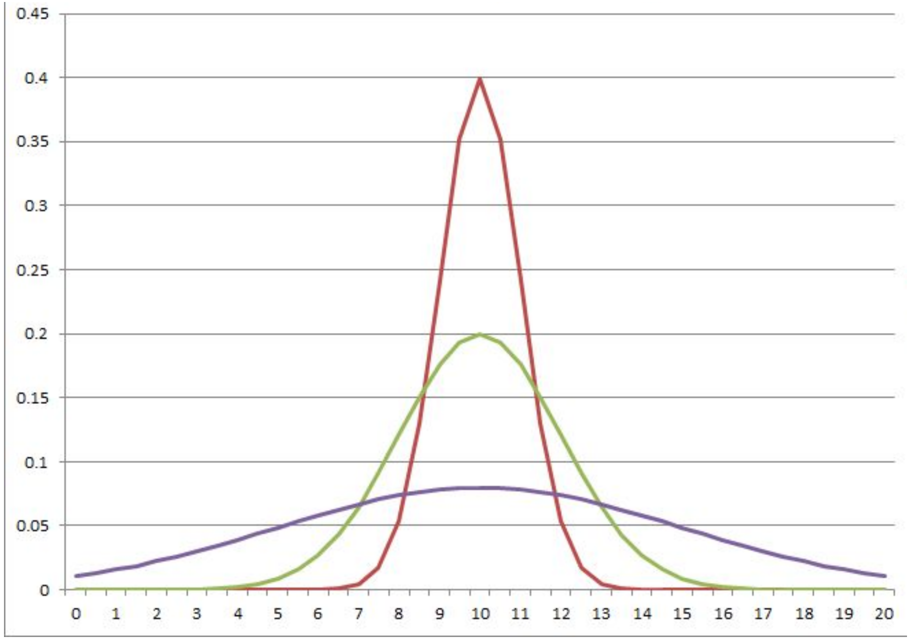
- Does knowing the mean of a distribution tell us everything we want to know about it?

- We want some measure of how spread out a distribution is, or how far its values are from the mean



Variance

- Does knowing the mean of a distribution tell us everything we want to know about it?
- We want some measure of how spread out a distribution is, or how far its values are from the mean
- For different distributions, we use different equations to compute variance



Computing Variance

- One approach to computing variance when you have a small sample population:

Computing Variance

- One approach to computing variance when you have a small sample population:

1. Compute the mean ($\bar{x}$) of your sample

$\bar{x}$

Var(X) =

Computing Variance

- One approach to computing variance when you have a small sample population:

1. Compute the mean ($\bar{x}$) of your sample
2. Calculate the difference between each element in your sample and the mean

$$x - \bar{x}$$

$$\mathrm{Var}(X) =$$

Computing Variance

- One approach to computing variance when you have a small sample population:

1. Compute the mean ($\bar{x}$) of your sample
2. Calculate the difference between each element in your sample and the mean
3. Square all of these differences

$$Var(X) = \quad (x - \bar{x})^2$$

Computing Variance

- One approach to computing variance when you have a small sample population:

1. Compute the mean ($\bar{x}$) of your sample
2. Calculate the difference between each element in your sample and the mean
3. Square all of these differences
4. Add all of the squared differences together

$$\mathrm{Var}(X) = \sum (x - \bar{x})^2$$

Computing Variance

- One approach to computing variance when you have a small sample population:

1. Compute the mean ($\bar{x}$) of your sample
2. Calculate the difference between each element in your sample and the mean
3. Square all of these differences
4. Add all of the squared differences together
5. Divide by the number of elements in your sample

$$\mathrm{Var}(X) = \frac{\sum (x - \bar{x})^2}{n}$$

Standard Deviation

- Because we square our values when we find the variance, this means that our variance numbers will be quite large

- **Standard deviation** (often written as σ(x)) is a measure of how spread out our data is, which uses the original unit of measure

- To compute the standard deviation, we take the square root of the variance

$$\mathrm{Var}(X) = \frac{\sum (x - \bar{x})^2}{n} \qquad \sigma(x) = \sqrt{\mathrm{Var}(X)}$$

Variance Exercise

- Given this set of test scores, compute the variance:

  {85, 86, 90, 95, 100, 98, 76, 66, 50, 99}

Variance Exercise

- Given this set of test scores, compute the variance:

  {85, 86, 90, 95, 100, 98, 76, 66, 50, 99}

- First, let's compute the mean ($\bar{x}$)

Variance Exercise

- Given this set of test scores, compute the variance:

  {85, 86, 90, 95, 100, 98, 76, 66, 50, 99}

- First, let's compute the mean ($\bar{x}$)

  (85 + 86 + 90 + 95 + 100 + 98 + 76 + 66 + 50 + 99) / 10 = 84.5

Variance Exercise

- Given this set of test scores, compute the variance:

  {85, 86, 90, 95, 100, 98, 76, 66, 50, 99}

- Next, compute the difference between all of the scores and the mean ($\bar{x} = 84.5$)

Variance Exercise

- Given this set of test scores, compute the variance:

  $\{85, 86, 90, 95, 100, 98, 76, 66, 50, 99\}$

- Next, compute the difference between all of the scores and the mean ($\bar{x} = 84.5$)

x	85	86	90	95	100	98	76	66	50	99
$x - \bar{x}$	0.5	1.5	4.5	10.5	15.5	13.5	-8.5	-18.5	-34.5	14.5
$(x - \bar{x})^2$										

Variance Exercise

- Given this set of test scores, compute the variance:

  {85, 86, 90, 95, 100, 98, 76, 66, 50, 99}

- Now compute the square of the differences

x	85	86	90	95	100	98	76	66	50	99
$x - \bar{x}$	0.5	1.5	4.5	10.5	15.5	13.5	-8.5	-18.5	-34.5	14.5
$(x - \bar{x})^2$										

Variance Exercise

- Given this set of test scores, compute the variance:

  {85, 86, 90, 95, 100, 98, 76, 66, 50, 99}

- Now compute the square of the differences

x	85	86	90	95	100	98	76	66	50	99
$x - \bar{x}$	0.5	1.5	4.5	10.5	15.5	13.5	-8.5	-18.5	-34.5	14.5
$(x - \bar{x})^2$	0.25	2.25	20.25	110.25	240.25	182.25	72.25	342.25	1190.25	210.25

Variance Exercise

- Given this set of test scores, compute the variance:

  {85, 86, 90, 95, 100, 98, 76, 66, 50, 99}

- Now compute the sum of the square of the differences

x	85	86	90	95	100	98	76	66	50	99
$x - \bar{x}$	0.5	1.5	4.5	10.5	15.5	13.5	-8.5	-18.5	-34.5	14.5
$(x - \bar{x})^2$	0.25	2.25	20.25	110.25	240.25	182.25	72.25	342.25	1190.25	210.25

$$\sum (x - \bar{x})^2$$

Variance Exercise

- Given this set of test scores, compute the variance:

  {85, 86, 90, 95, 100, 98, 76, 66, 50, 99}

- Now compute the sum of the square of the differences

x	85	86	90	95	100	98	76	66	50	99
$x - \bar{x}$	0.5	1.5	4.5	10.5	15.5	13.5	-8.5	-18.5	-34.5	14.5
$(x - \bar{x})^2$	0.25	2.25	20.25	110.25	240.25	182.25	72.25	342.25	1190.25	210.25

$$\sum (x - \bar{x})^2 = 2370.5$$

Variance Exercise

- Given this set of test scores, compute the variance:

  {85, 86, 90, 95, 100, 98, 76, 66, 50, 99}

- Now divide by the number of elements (n = 10)

$$\sum (x - \bar{x})^2 = 2370.5$$

Variance Exercise

- Given this set of test scores, compute the variance:

  {85, 86, 90, 95, 100, 98, 76, 66, 50, 99}

- Now divide by the number of elements (n = 10)

$$\sum (x - \bar{x})^2 = 2370.5$$

$$Var(X) = \frac{\sum (x - \bar{x})^2}{n} = 237.05$$

Variance Exercise

- Given this set of test scores, compute the variance:

  {85, 86, 90, 95, 100, 98, 76, 66, 50, 99}

- Now divide by the number of elements (n = 10)

$$\sum (x - \bar{x})^2 = 2370.5$$

$$\sigma(x) = 15.4$$

$$\mathrm{Var}(X) = \frac{\sum (x - \bar{x})^2}{n} = 237.05$$

Variance Exercise

- Given this set of test scores, compute the variance:

  {85, 86, 90, 95, 100, 98, 76, 66, 50, 99}

- Now divide by the number of elements (n = 10)

$$\sum (x - \bar{x})^2 = 2370.5$$

$$\mathrm{Var}(X) = \frac{\sum (x - \bar{x})^2}{n} = 237.05$$

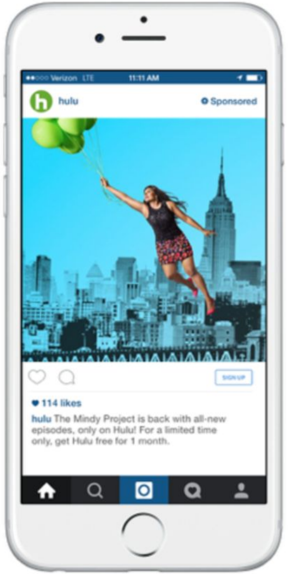
$\sigma(x) = 15.4$

Since $\bar{x} = 84.5$, notice $\bar{x} + \sigma(x)$ roughly gets us to 100

Linear Regression

Let’s go back to our example…

How many likes will my Instagram post get?

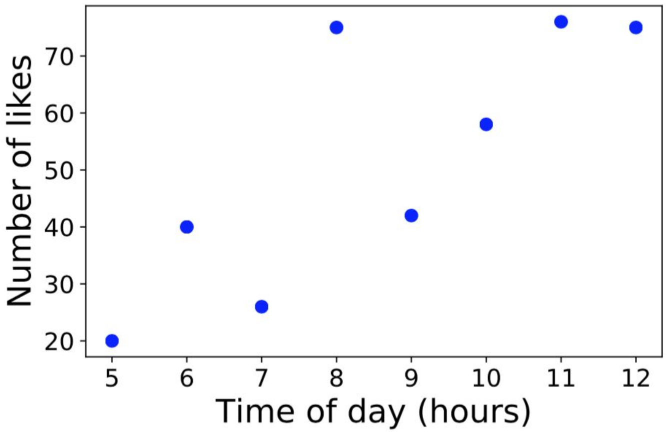


- What is the label?
- What are some features we might collect?

What is the best time to post on Instagram?

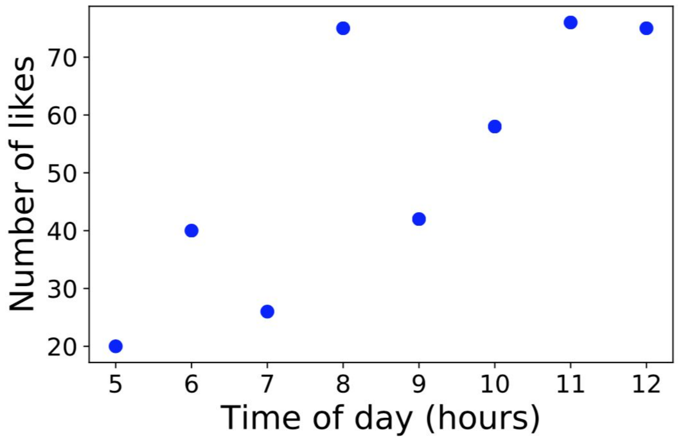
Example Data

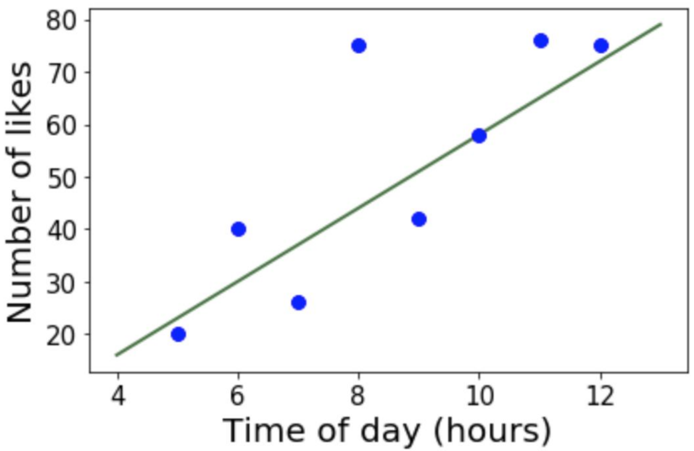
Time of day (hours)	Number of Likes
5	20
6	40
7	26
8	75
9	42
10	58
11	76
12	75



Regression

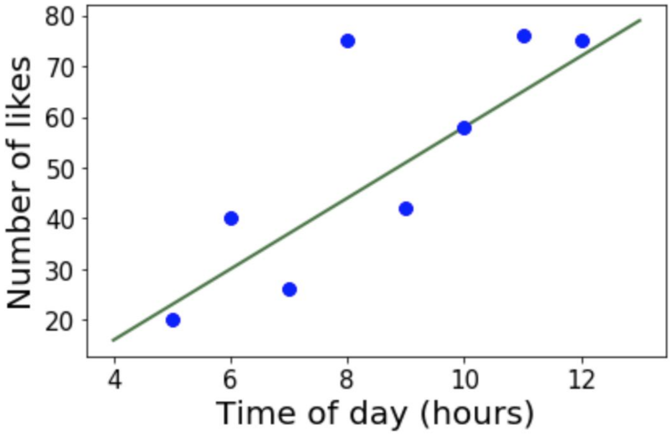
- A common analysis process used in statistics
- Regression looks for a relationship between the input and output values in order to try to predict output values for unseen inputs





Linear Regression

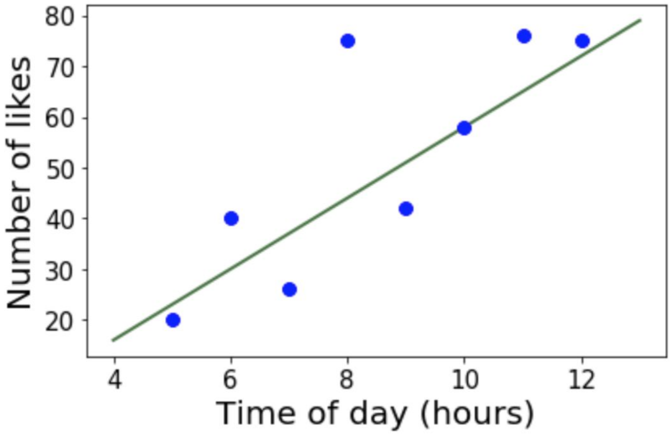
- Suppose we guess that the relationship between our input and outputs is **linear**



Linear Regression

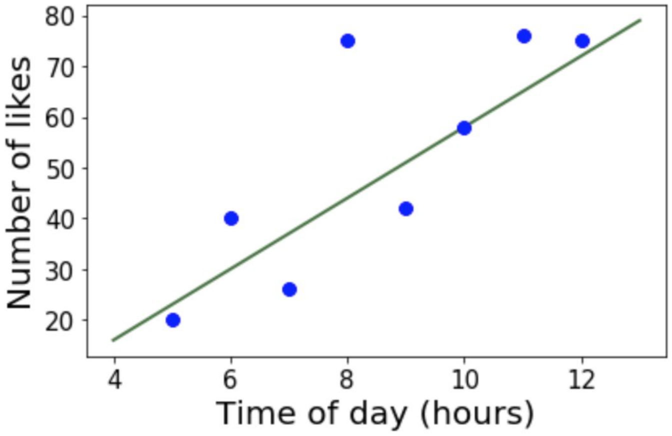
- Suppose we guess that the relationship between our input and outputs is **linear**

- In other words, we are guessing that we can describe this relationship with a line



Linear Regression

- Suppose we guess that the relationship between our input and outputs is **linear**
- In other words, we are guessing that we can describe this relationship with a line
- Recall the equation for describing a line:

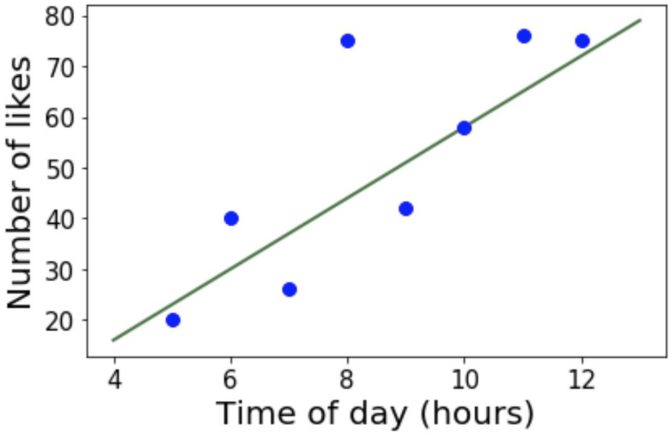


Linear Regression

- Suppose we guess that the relationship between our input and outputs is **linear**

- In other words, we are guessing that we can describe this relationship with a line

- Recall the equation for describing a line:
  y = mx + b

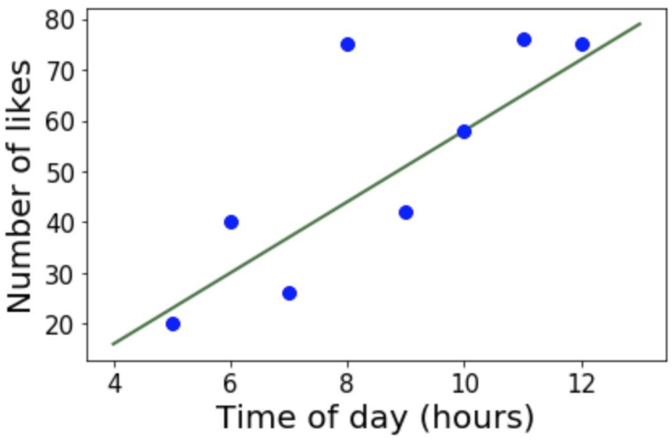


Linear Regression

- Suppose we guess that the relationship between our input and outputs is **linear**

- In other words, we are guessing that we can describe this relationship with a line

- Recall the equation for describing a line:
  y = mx + b



Where:
- y is the output
- x is the input
- m is the slope
- b is the y-intercept

Parameters for Linear Regression

- Recall that for k-Nearest Neighbors, we were able to pick one parameter: k

- For linear regression, we have two parameters (also called **weights**) that we want to learn

Parameters for Linear Regression

- Recall that for k-Nearest Neighbors, we were able to pick one parameter: k

- For linear regression, we have two parameters (also called **weights**) that we want to learn

- Looking at the equation of the line: $y = mx + b$

  m and b are the parameters that change what line we draw

Parameters for Linear Regression

- Recall that for k-Nearest Neighbors, we were able to pick one parameter: k

- For linear regression, we have two parameters (also called **weights**) that we want to learn

- Looking at the equation of the line: $y = mx + b$

  m and b are the parameters that change what line we draw

- In machine learning, we often rewrite the weights as w_1 and w_2

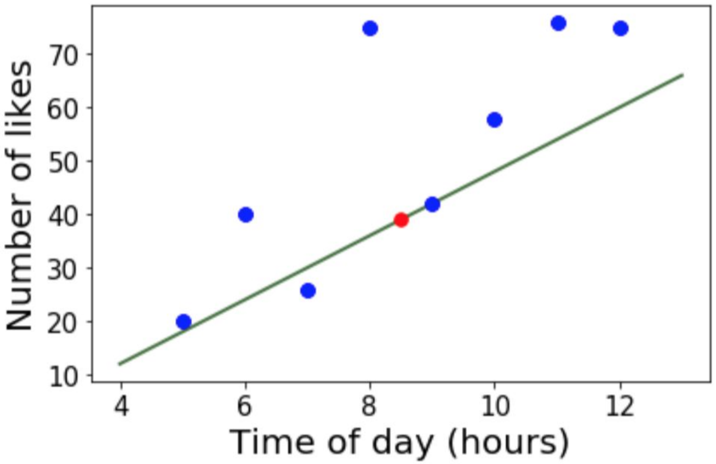
  $y = w_1 x + w_2$

Predicting a new label

- Suppose the formula for the line is:

  $\hat{y} = 6x - 12$

- What is the predicted number of likes for a post at 8:30am?



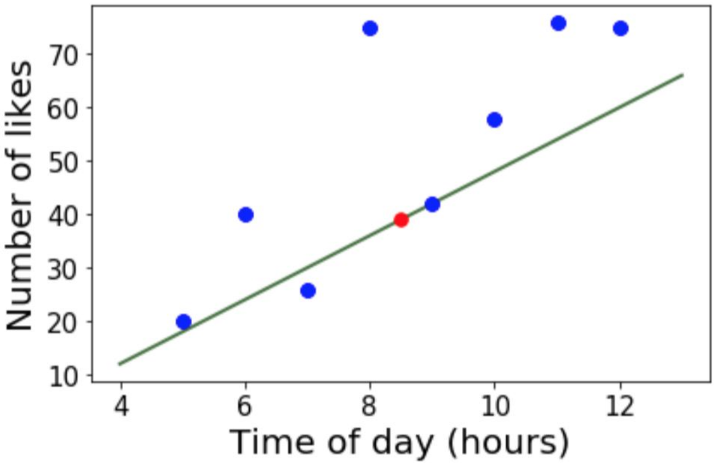
Predicting a new label

- Suppose the formula for the line is:

  $\hat{y} = 6x - 12$

- What is the predicted number of likes for a post at 8:30am?

- Substitute 8.5 into the formula for x



Predicting a new label

- Suppose the formula for the line is:

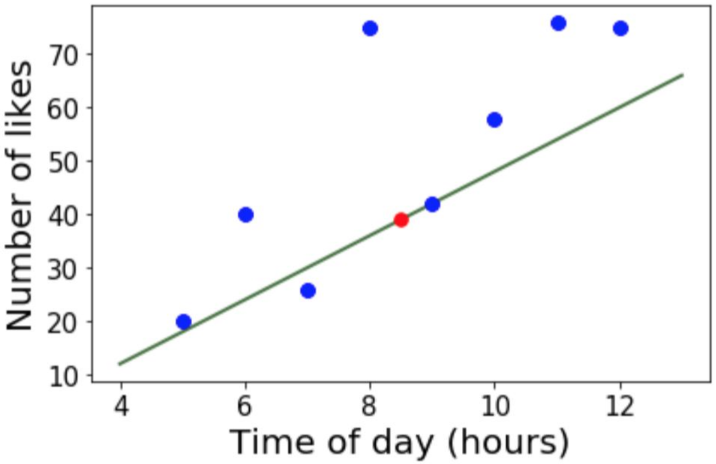
  $\hat{y} = 6x - 12$

- What is the predicted number of likes for a post at 8:30am?

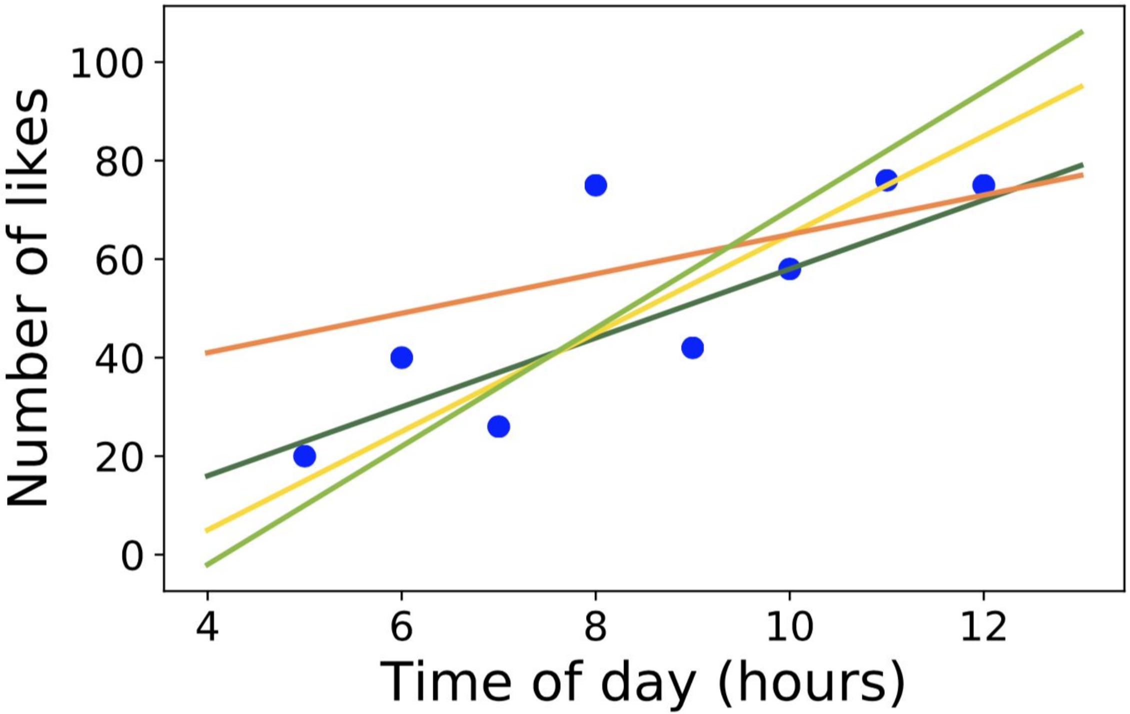
- Substitute 8.5 into the formula for x

- Our predicted output (or label), $\hat{\mathbf{y}}$, is

  $\hat{y} = 6 * (8.5) - 12 = 39$

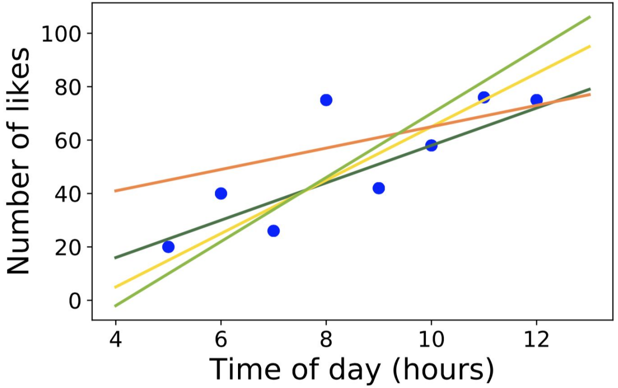


Which line is best?



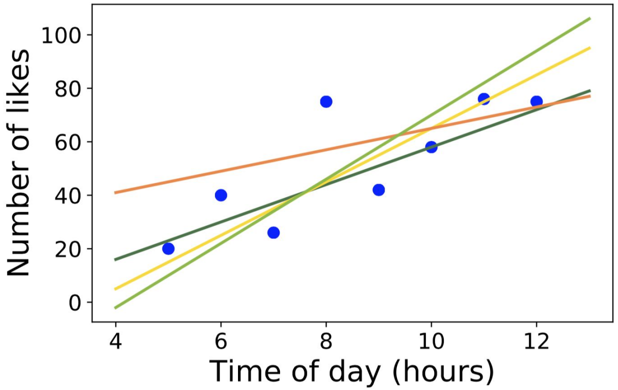
Which line is best?

- Ideal case: The line that goes through all of the points



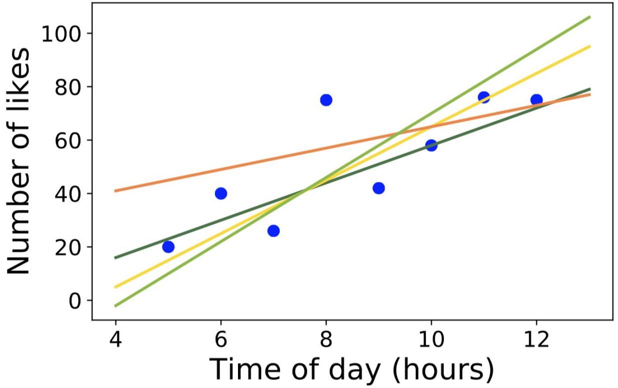
Which line is best?

- Ideal case: The line that goes through all of the points
- Reality: It's often not possible to go through every single point perfectly



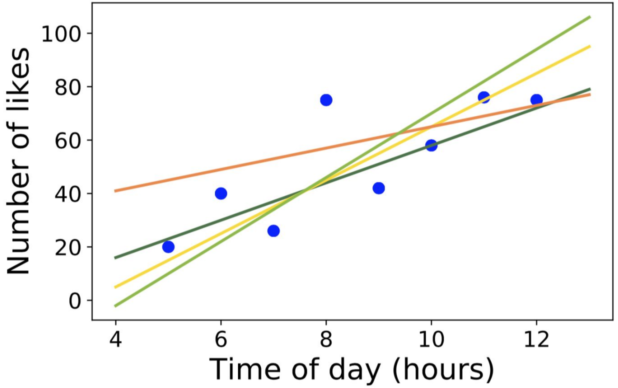
Which line is best?

- Ideal case: The line that goes through all of the points
- Reality: It's often not possible to go through every single point perfectly
- We want to pick the line that goes through all of our points as closely as possible



Which line is best?

- Ideal case: The line that goes through all of the points
- Reality: It's often not possible to go through every single point perfectly
- We want to pick the line that goes through all of our points as closely as possible
- This is called the line of **best fit**

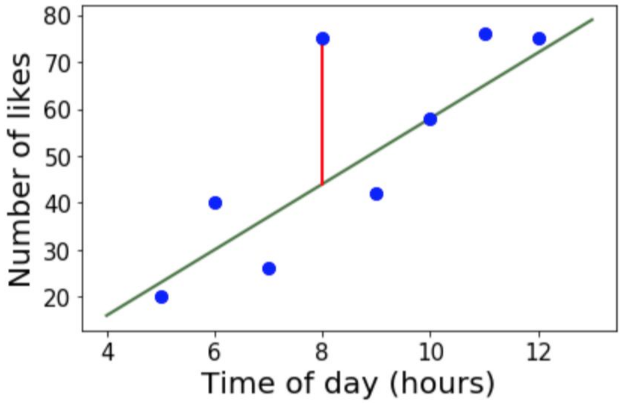


Measuring error

- How can we tell how good our line is?

Measuring error

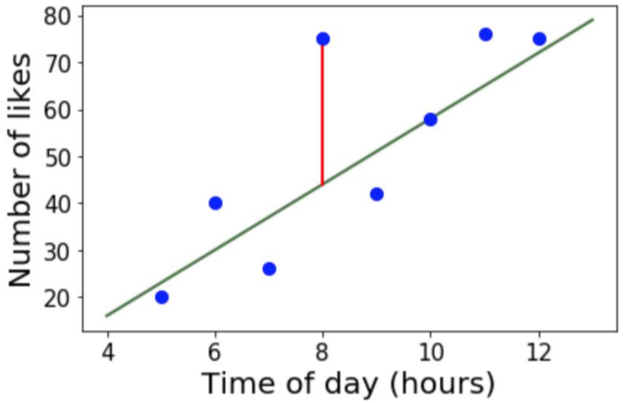
- How can we tell how good our line is?
- One way is to measure the difference between our predicted label (ŷ) and the actual label (y) for each input
- This is the same as taking the distance from the predicted label to the actual label
- We call this difference the **error**



Measuring error

- The difference between the predicted label and the actual label, for one example:

Error = ŷ - y



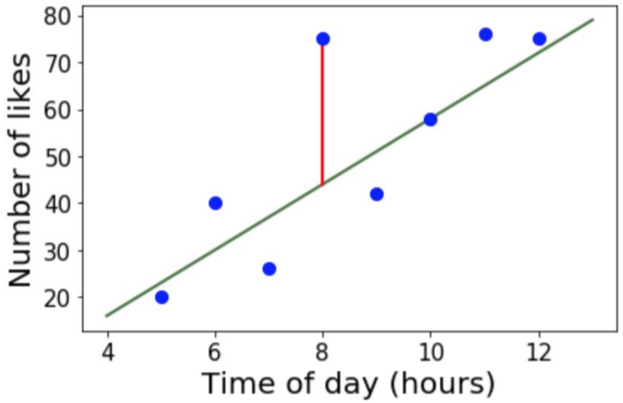
Measuring error

- The difference between the predicted label and the actual label, for one example:

  Error = $\hat{y}$ - y

- We can rewrite $\hat{y}$ using our line equation:

  Error = $(w_1x + w_2) - y$



Measuring error

- The difference between the predicted label and the actual label, for one example:

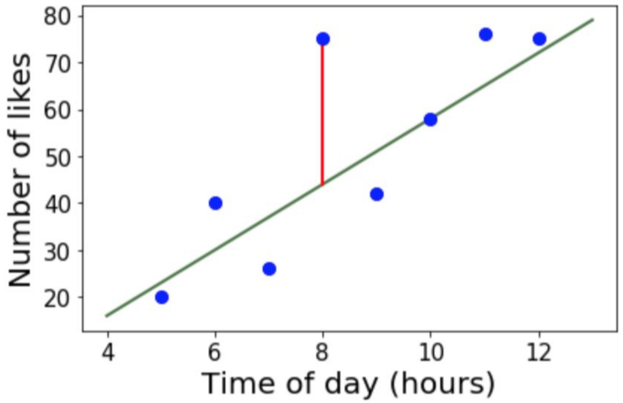
  Error = $\hat{y}$ - y

- We can rewrite $\hat{y}$ using our line equation:

  Error = $(w_1x + w_2) - y$

- Error should never be negative -- we can square the equation to make this true

  Error = $(w_1x + w_2 - y)^2$



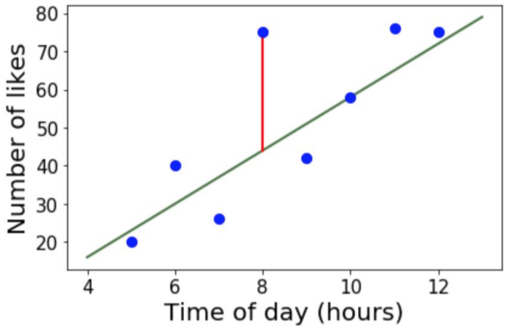
Measuring error

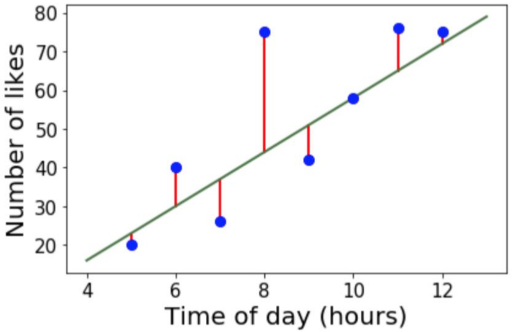
$$\text{Error} = (w_1x + w_2 - y)^2$$

- We call this the **squared error**
- But this is error only for a single point -- we want to sum all of the errors for all of the points

$$\text{Error} = \Sigma\,(w_1x + w_2 - y)^2$$

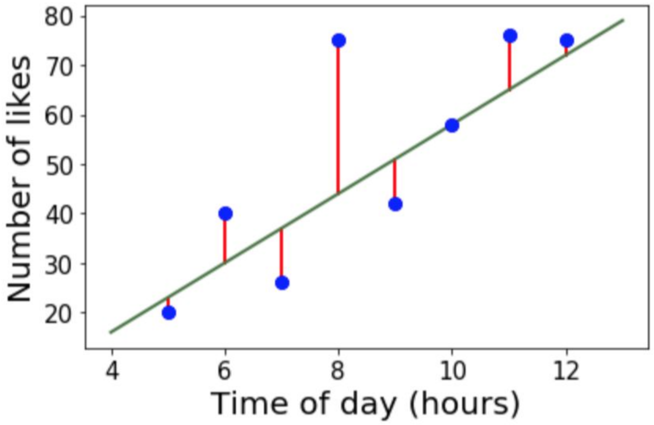
- This is called the **sum of squared errors**



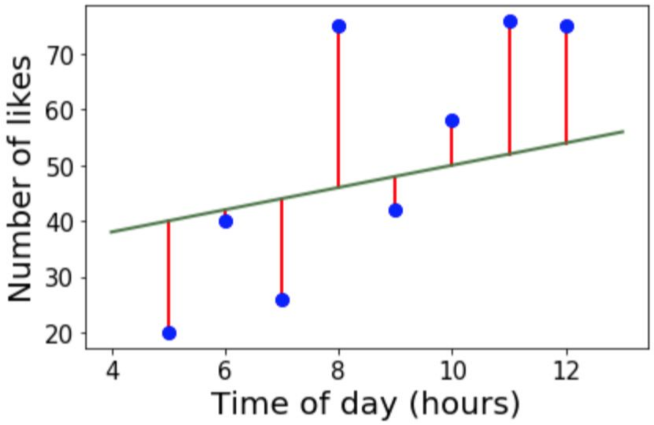


Which line is better?

- Using sum of squared errors, which line do we think is better now?



(a)



(b)

Loss functions

- So far we've defined error as the function $\Sigma\,(w_1x + w_2 - y)^2$

Loss functions

- So far we've defined error as the function $\Sigma\ (w_1x + w_2 - y)^2$
- This is often called a **loss function** (or cost function)

Loss functions

- So far we've defined error as the function $\Sigma\,(w_1x + w_2 - y)^2$
- This is often called a **loss function** (or cost function)
- There are also other loss functions we can use!

Mean squared error: $1/N * \Sigma\,(w_1x + w_2 - y)^2$

Absolute value: $\Sigma\,|\,w_1x + w_2 - y\,|$

Loss functions

- So far we've defined error as the function $\Sigma\,(w_1x + w_2 - y)^2$
- This is often called a **loss function** (or cost function)
- There are also other loss functions we can use!

  Mean squared error: $1/N * \Sigma\,(w_1x + w_2 - y)^2$

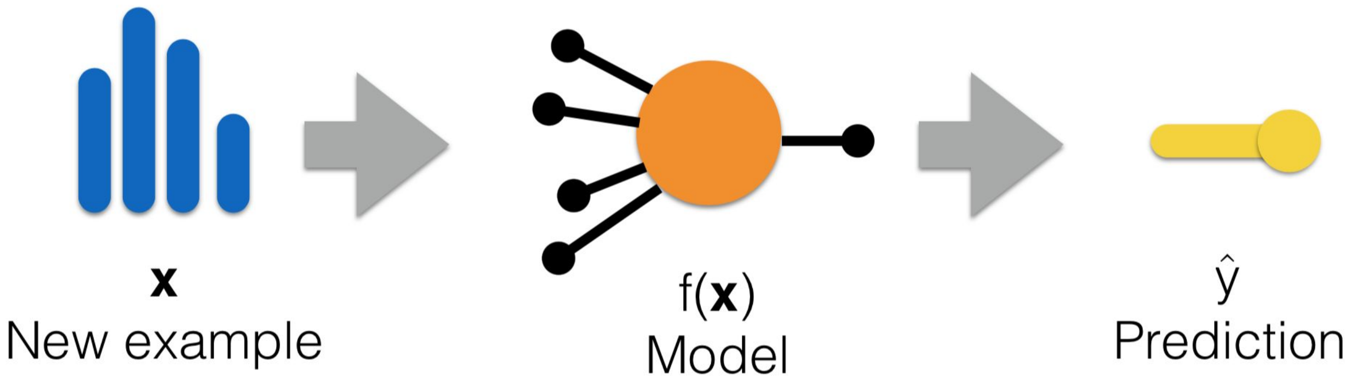
  Absolute value: $\Sigma\,|\,w_1x + w_2 - y\,|$

- There's are many choices of loss functions we can use, but we won't cover them

Training a Linear Regression Model

Linear Regression Model

- Recall that a machine learning model is a function that takes examples and return labels
- While linear regression is an algorithm, the model is the specific line that we use



Training a model

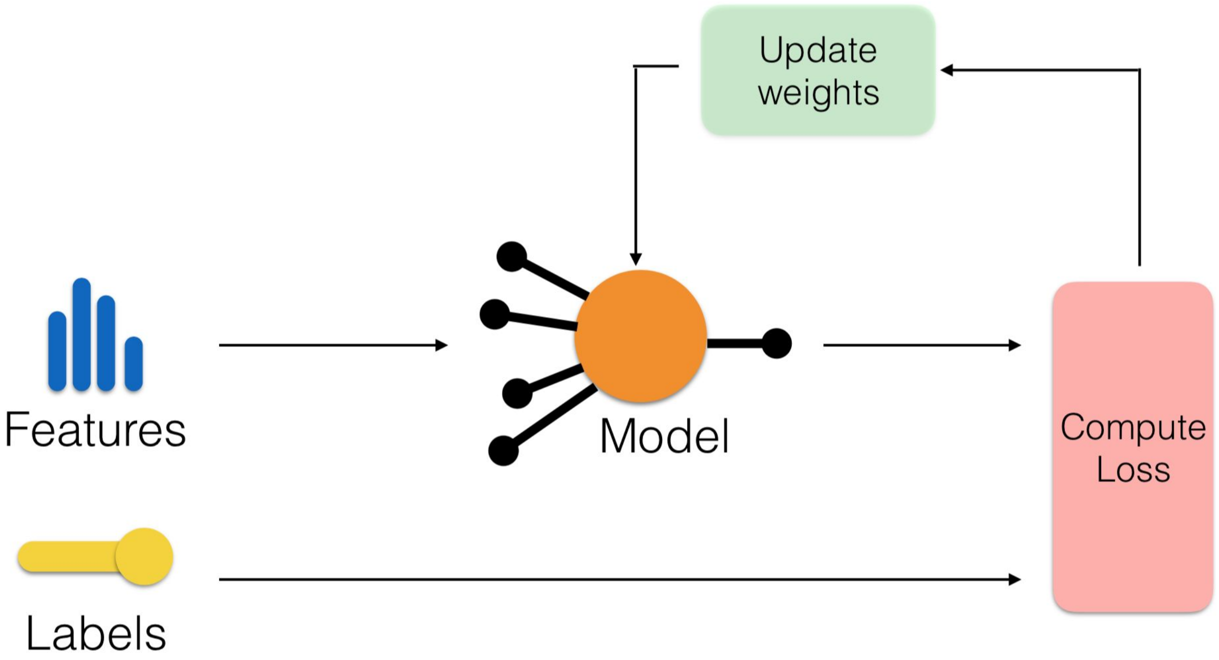
- **Training** a model means learning good values for our parameters (or weights) that minimize loss

- For our linear models, this means learning a good w_1 and w_2

- This is the heart of machine learning: using procedures to improve our models without humans changing the parameters ourselves

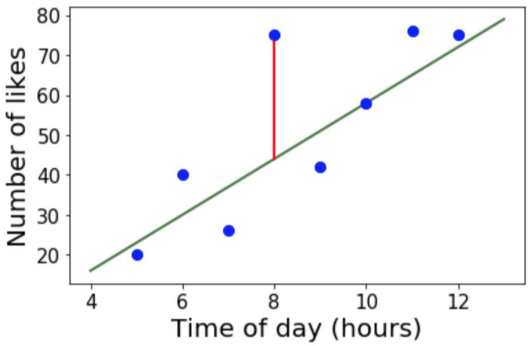
How do we train the model?

We use an **iterative** approach



Updating the weights

- If our prediction ($\hat{y}$) is too low, what should we do?
- We want the predicted output or label to be larger, so we want to increase the value of our weights
- But by how much should we increase w_1 and w_2?



Here: $w_1x + w_2 < y$

Updating the weights

- Recall that we've defined error, or loss, as:

  $(w_1 x + w_2 - y)^2$

  for a single example with the actual output, y

Updating the weights

- Recall that we've defined error, or loss, as:

  $(w_1x + w_2 - y)^2$

  for a single example with the actual output, y

- We can take this error and use it to tell us how much to update our weights

  $w_1 \text{ (new)} = w_1 \text{ (old)} - \Sigma (\alpha * \text{error} * x)$
  $w_2 \text{ (new)} = w_2 \text{ (old)} - \Sigma (\alpha * \text{error})$

Updating the weights

- Recall that we've defined error, or loss, as:

  $(w_1 x + w_2 - y)^2$

  for a single example with the actual output, y

- We can take this error and use it to tell us how much to update our weights

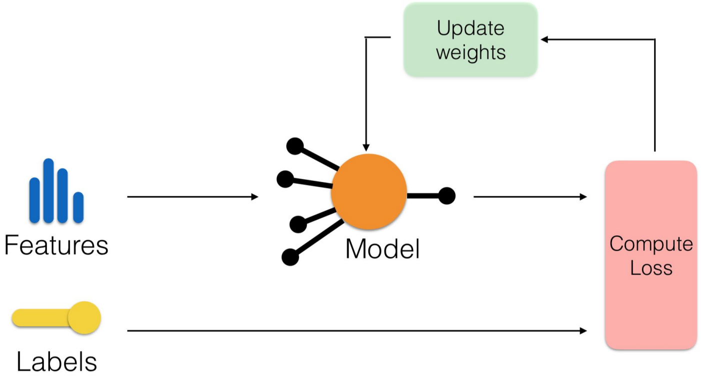
  $w_1 \text{ (new)} = w_1 \text{ (old)} - \Sigma (\alpha * \text{error} * x)$
  $w_2 \text{ (new)} = w_2 \text{ (old)} - \Sigma (\alpha * \text{error})$

- α is known as the **learning rate**, which is a constant that we get to pick, typically a small value in the range [0, 1]

Iterate!

- We repeat this process over and over, each time updating our weights so that our predictions get better

- This process of updating the weights, computing loss, and repeating is called **gradient descent**



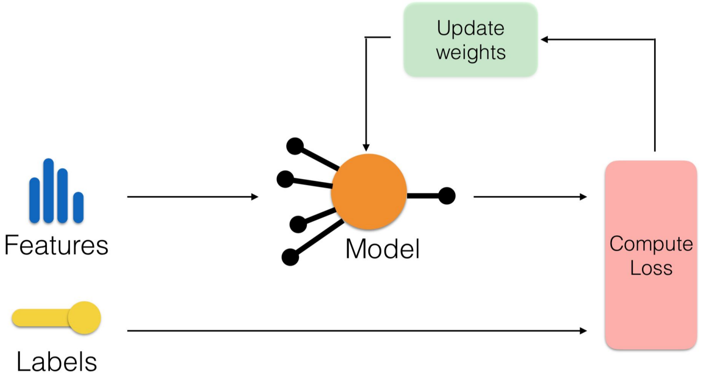
Iterate!

- We repeat this process over and over, each time updating our weights so that our predictions get better

- This process of updating the weights, computing loss, and repeating is called **gradient descent**

- Updating the weights based on the error of one example at a time is called **stochastic gradient descent**

- This is often faster and more effective than computing total loss for *all* examples each time



Directly computing

- For simple cases we can directly compute w_1 and w_2

$$w_1 = \bar{y} - w_2\bar{x}$$

$$w_2 = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sum_{i=1}^{n}(x_i - \bar{x})^2} = \frac{\mathrm{Cov}(x,y)}{\mathrm{Var}(x)}$$

Directly computing

- For simple cases we can directly compute w_1 and w_2

$$w_1 = \bar{y} - w_2\bar{x}$$
$$w_2 = \frac{\sum_{i=1}^{n}(x_i-\bar{x})(y_i-\bar{y})}{\sum_{i=1}^{n}(x_i-\bar{x})^2} = \frac{\text{Cov}(x,y)}{\text{Var}(x)}$$

- We can expand this more generally for any $\hat{\beta} = [w_1, w_2, \ldots, w_k]^T$

$$\hat{\beta} = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{y}$$

Adding more features

- How many features have we used so far to predict # of likes?

Adding more features

- How many features have we used so far to predict # of likes?

- What if we added more features, such as the number of followers you have?

4,000+ likes with 113,000 followers!

Adding more features

- How many features have we used so far to predict # of likes?
- What if we added more features, such as the number of followers you have?
- For each feature we add, we will want to add a weight as well

4,000+ likes with 113,000 followers!

Adding more features

- How many features have we used so far to predict # of likes?
- What if we added more features, such as the number of followers you have?
- For each feature we add, we will want to add a weight as well
- For two features, our model would be:

$$\hat{y} = w_1x_1 + w_2x_2 + w_3$$

4,000+ likes with 113,000 followers!

Adding more features

- With our new model:

$$\hat{y} = w_1x_1 + w_2x_2 + w_3$$

- We also will want some new update rules:

$$w_1\,(\text{new}) = w_1\,(\text{old}) - \alpha * \text{error} * x_1$$
$$w_2\,(\text{new}) = w_2\,(\text{old}) - \alpha * \text{error} * x_2$$
$$w_3\,(\text{new}) = w_3\,(\text{old}) - \alpha * \text{error}$$

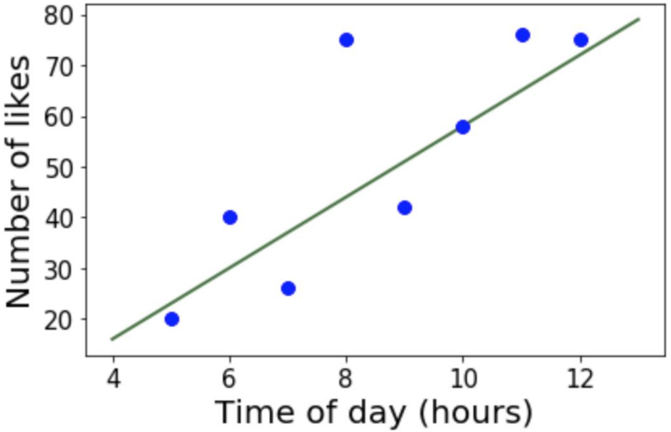
- What might this look like if we had more features?

4,000+ likes with 113,000 followers!

Logistic Regression

Non-Linear Functions of Features

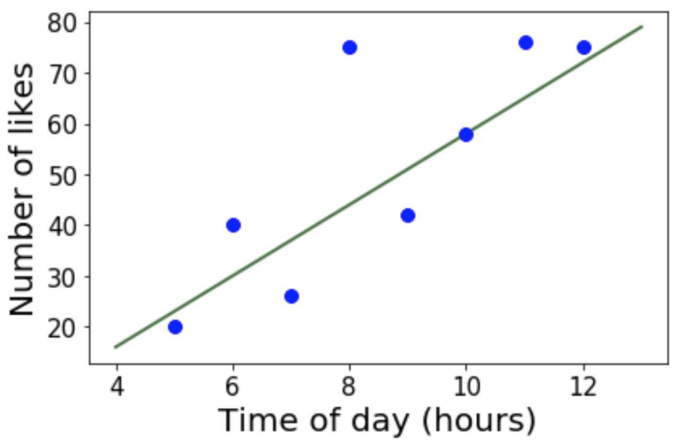
- So far, we've only seen examples of **linear combinations** of features



Non-Linear Functions of Features

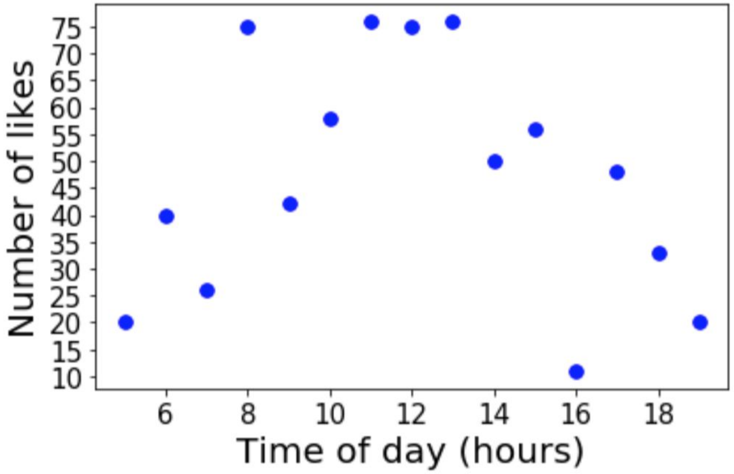
- So far, we've only seen examples of **linear combinations** of features

- Recall we assumed that we could find a *line* of best fit



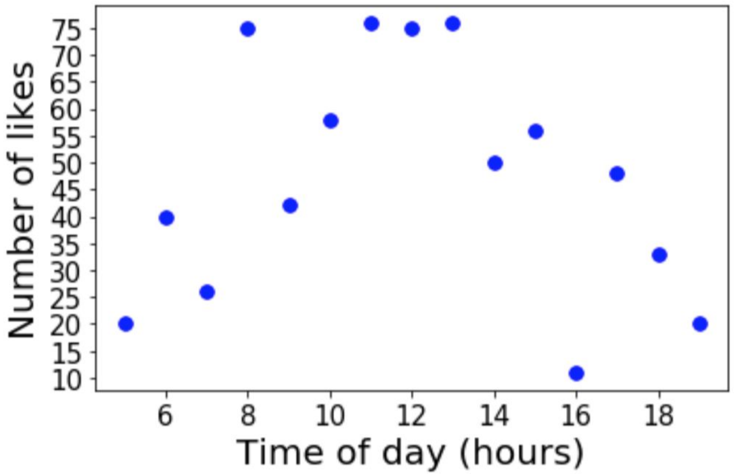
Non-Linear Functions of Features

- So far, we've only seen examples of **linear combinations** of features
- Recall we assumed that we could find a *line* of best fit
- But what if our data is messier and a line doesn't work?



Non-Linear Functions of Features

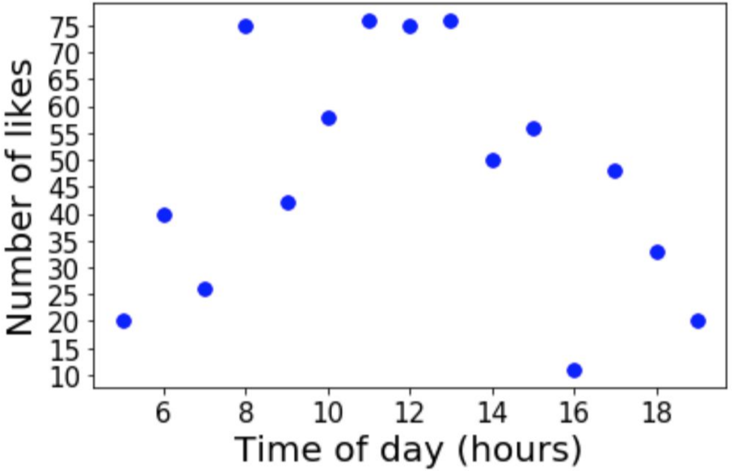
- So far, we've only seen examples of **linear combinations** of features
- Recall we assumed that we could find a *line* of best fit
- But what if our data is messier and a line doesn't work?
- We might try **nonlinear** functions



Non-Linear Functions of Features

- So far, we've only seen examples of **linear combinations** of features
- Recall we assumed that we could find a *line* of best fit
- But what if our data is messier and a line doesn't work?
- We might try **nonlinear** functions

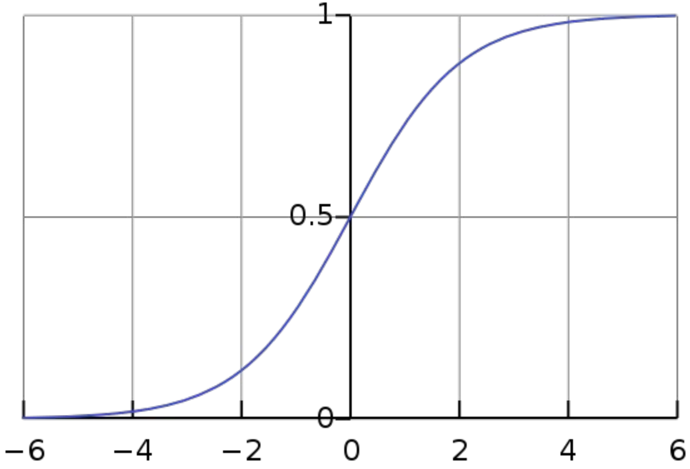
  sin(x)
  log(x)
  arctan(x)
  1/x



Logistic Regression

- One very popular approach is to use **logistic regression**
- This is based on the logistic function

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



- The above is a simplified version of the logistic function known as a **sigmoid** function

Logistic Regression

- In logistic regression, we multiply our input by the weights first and then plug this value into the logistic (or sigmoid) function before getting our output

$$\hat{y} = \frac{1}{1 + e^{-(w1x + w2)}}$$

- Recall that the portion within the parentheses above is what we previously used for linear regression

Logistic Regression

- In logistic regression, we multiply our input by the weights first and then plug this value into the logistic (or sigmoid) function before getting our output

$$\hat{y} = \frac{1}{1 + e^{-(w1x + w2)}}$$

- Recall that the portion within the parentheses above is what we previously used for linear regression

- By using the sigmoid function, the relationship between our inputs and our output is not linear and we can model more interesting behavior in our data set

Loss Function for Logistic Regression

- Because the logistic function is more complicated than the line, the loss function we need to use is more complicated as well
- We will also need different update rules to improve our weights for logistic regression
- We won’t cover the loss function and update rules for logistic regression here!
- The takeaway: different functions will require different loss functions and update rules

Concepts Learned

- Regression
- Linear Regression
- Loss Function
- Gradient Descent
- Logistic Regression