# ICS 504: Machine Learning
# Lecture 2
# Supervised Learning
# Linear Regression I

Dr. Caroline Sabty
caroline.sabty@giu-uni.de
Faculty of Informatics and Computer Science
German International University in Cairo

# Acknowledgment

The course and the slides are based on the slides of Dr. Seif Eldawlatly and based on the course created by Prof. Jose Portilla

# Additional Resources

- ISLR - Introduction to Statistical Learning
  - Freely available book that gives a fantastic overview of many of the ML algorithms we discuss in the course.
  - Quick note, it’s code is for R users, but the math behind algorithms is the same regardless of programming language used in development.

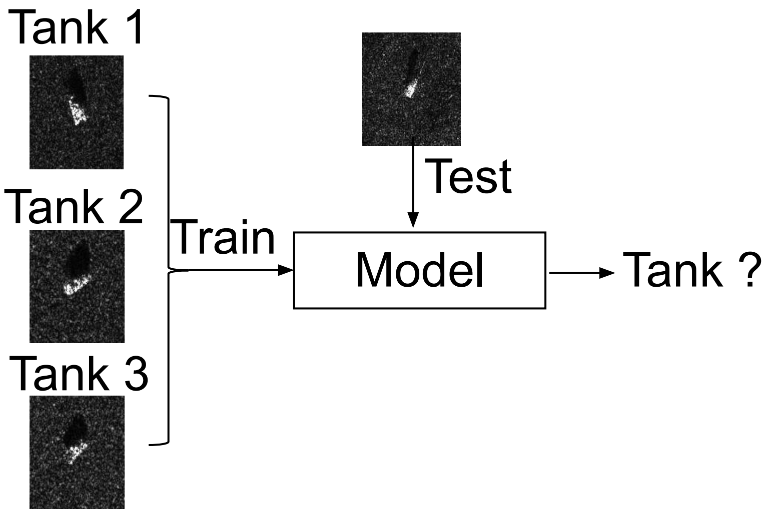
# Machine Learning

Real World → Collect & Store Data → Clean & Organize Data → Exploratory Data Analysis → Machine Learning Models

**Supervised Learning:**
*Predict an Outcome*

# Supervised Learning

- **Supervised Learning**
  - Requires historical labeled data:
    - Historical
  - Known results and data from the past
    - Labeled
  - The desired output is known
  - Two main label types:
    - Categorical Value to Predict
      - Classification Task
    - Continuous Value to Predict
      - Regression Task



# Supervised Machine Learning Process

- Start with collecting and organizing a data set based on history: Historical labeled data on previously sold houses.

| Area $m^2$ | Bedrooms | Bathrooms | Price |
|---|---|---|---|
| 200 | 3 | 2 | \$500,000 |
| 190 | 2 | 1 | \$450,000 |
| 230 | 3 | 3 | \$650,000 |
| 180 | 1 | 1 | \$400,000 |
| 210 | 2 | 2 | \$550,000 |

- If a new house comes on the market with a known Area, Bedrooms, and Bathrooms: *Predict what price should it sell at.*

- Data Product:
  - Input house features
  - Output predicted selling price

# Supervised Machine Learning Process

**Data** → **X: Features**
**y: Label**

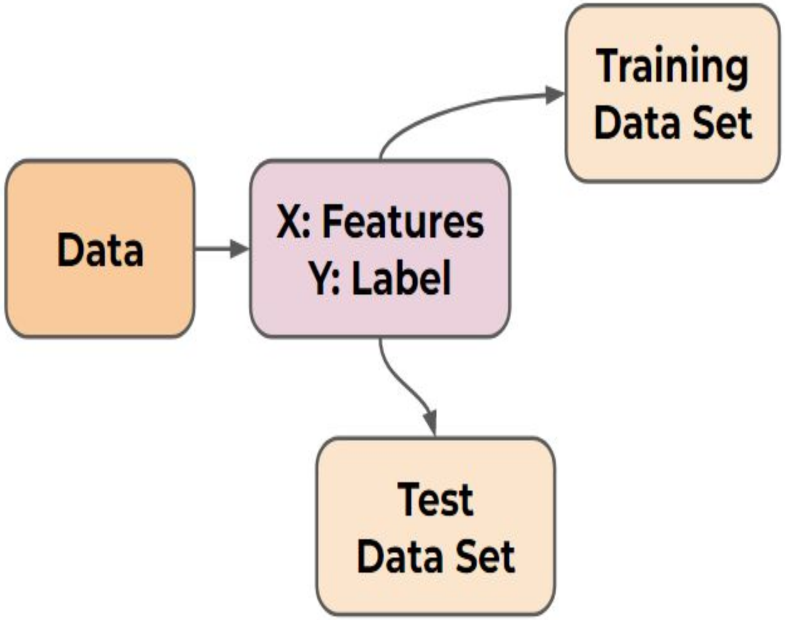
| Area m$^2$ | Bedrooms | Bathrooms | Price |
|---|---|---|---|
| 200 | 3 | 2 | $500,000 |
| 190 | 2 | 1 | $450,000 |
| 230 | 3 | 3 | $650,000 |
| 180 | 1 | 1 | $400,000 |
| 210 | 2 | 2 | $550,000 |

Features → are the known characteristics or components in the data that we are predicting the labels from

Label → What do we want to predict

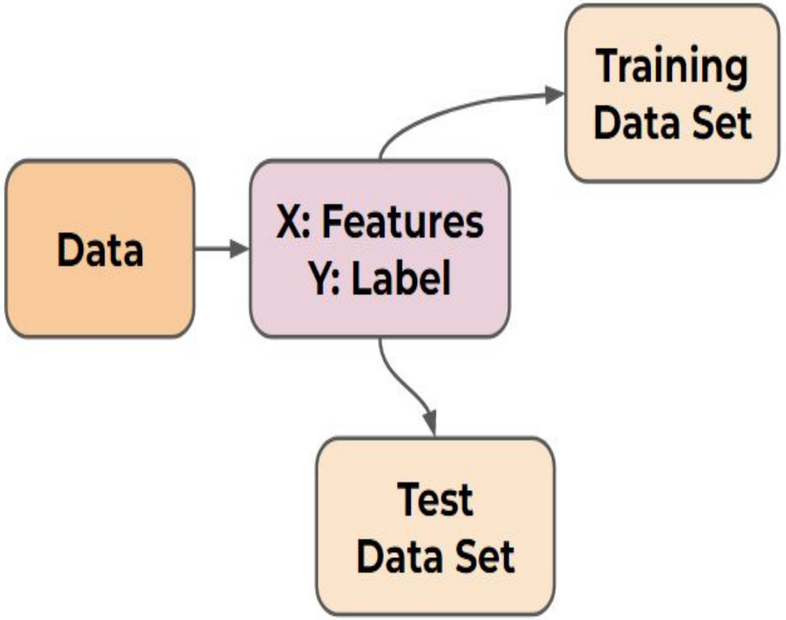
# Supervised Machine Learning Process

- Split data into training set and test set
- Why perform this split? How to split?



# Supervised Machine Learning Process

- Split data into training set and test set
- Why perform this split? How to split?
- How would you judge a human real estate agent's performance?
- Ask the person to take a look at historical data...
- Then give her the features of a house and ask her to predict a selling price.

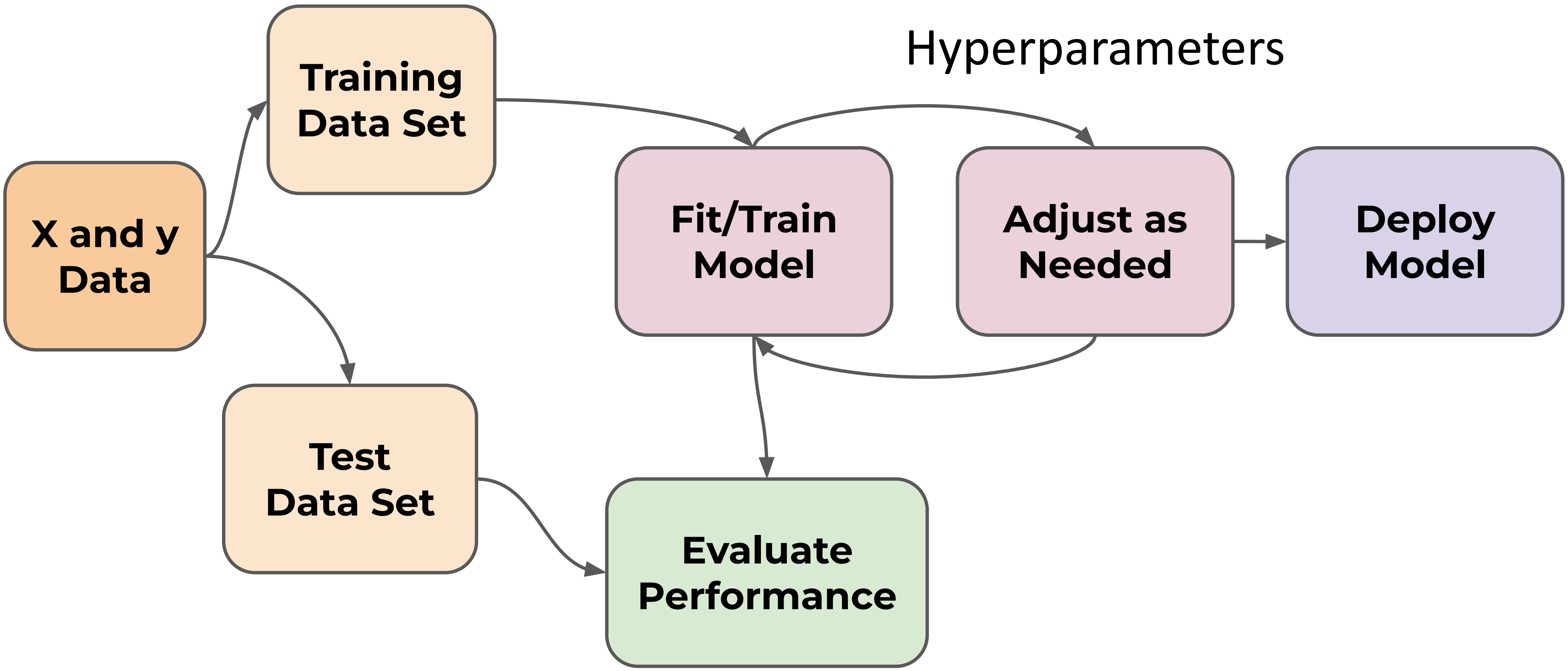


# Supervised Machine Learning Process

- Notice how we have 4 components

| | Area $m^2$ | Bedrooms | Bathrooms | Price | |
|---|---|---|---|---|---|
| X TRAIN | 200 | 3 | 2 | $500,000 | Y TRAIN |
| | 190 | 2 | 1 | $450,000 | |
| | 230 | 3 | 3 | $650,000 | |
| X TEST | 180 | 1 | 1 | $400,000 | Y TEST |
| | 210 | 2 | 2 | $550,000 | |

# Full and Simplified Process



# Linear Regression

# Resources

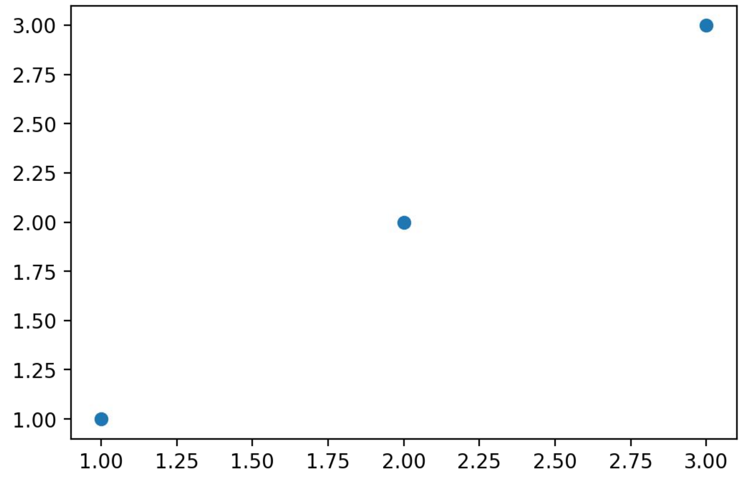
- Relevant Reading in ISLR
    - Section 3 : Linear Regression
        - 3.1 Simple Linear Regression

# Linear Regression

- The first machine learning algorithm we will explore is also one of the oldest
- **Linear Regression:** allows us to build a relationship between multiple features to estimate a target output.
- **Simple linear regression** is used to estimate the relationship between two quantitative variables. You can use simple linear regression when you want to know:
    - How strong the relationship is between two variables (e.g. the relationship between rainfall and soil erosion).
    - The value of the dependent variable at a certain value of the independent variable.
- This will include understanding:
    - Linear Relationships
    - Ordinary Least Squares
    - Cost Functions
    - Gradient Descent
    - Vectorization

# Linear Regression

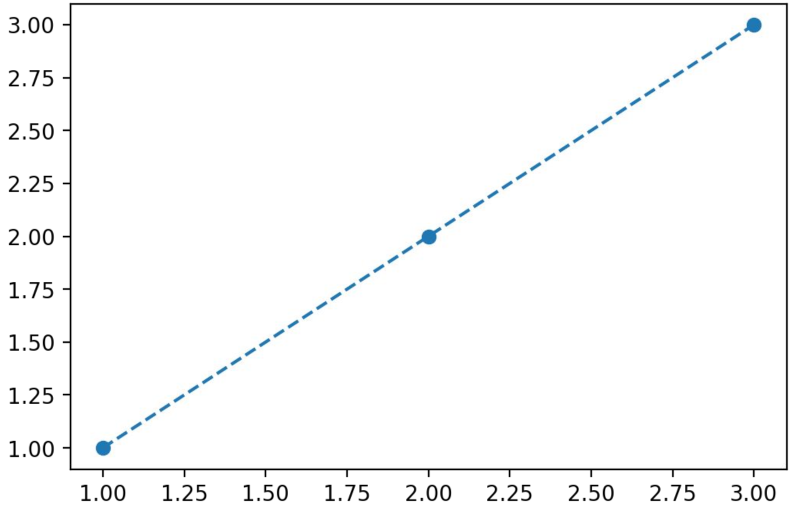
- Put simply, a linear relationship implies some constant straight line relationship.
- The simplest possible being y = x.




- Here we see x = [1,2,3] and y = [1,2,3]

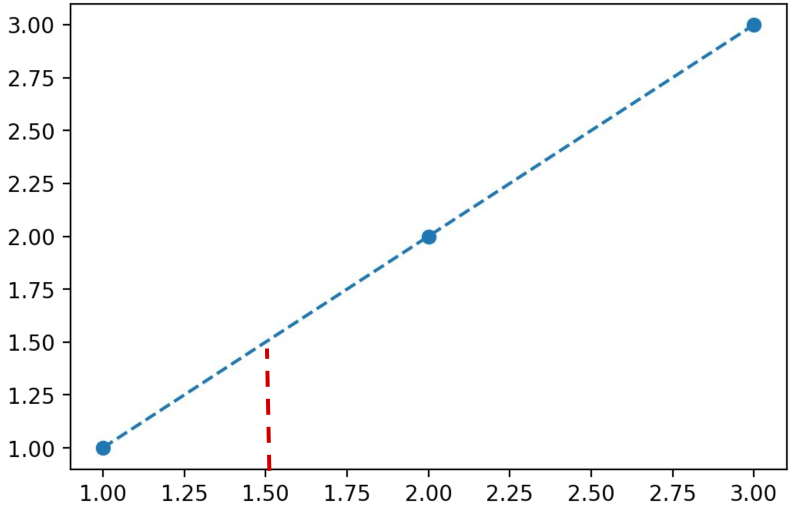
# Linear Regression

- We could then (based on the three real data points) build out the relationship y=x as our “fitted” line.



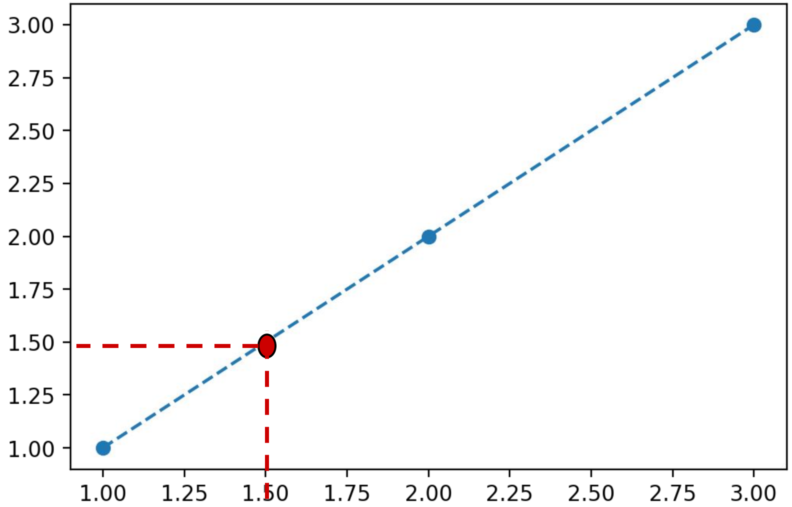
# Linear Regression

- This implies for some new x value I can predict its related y

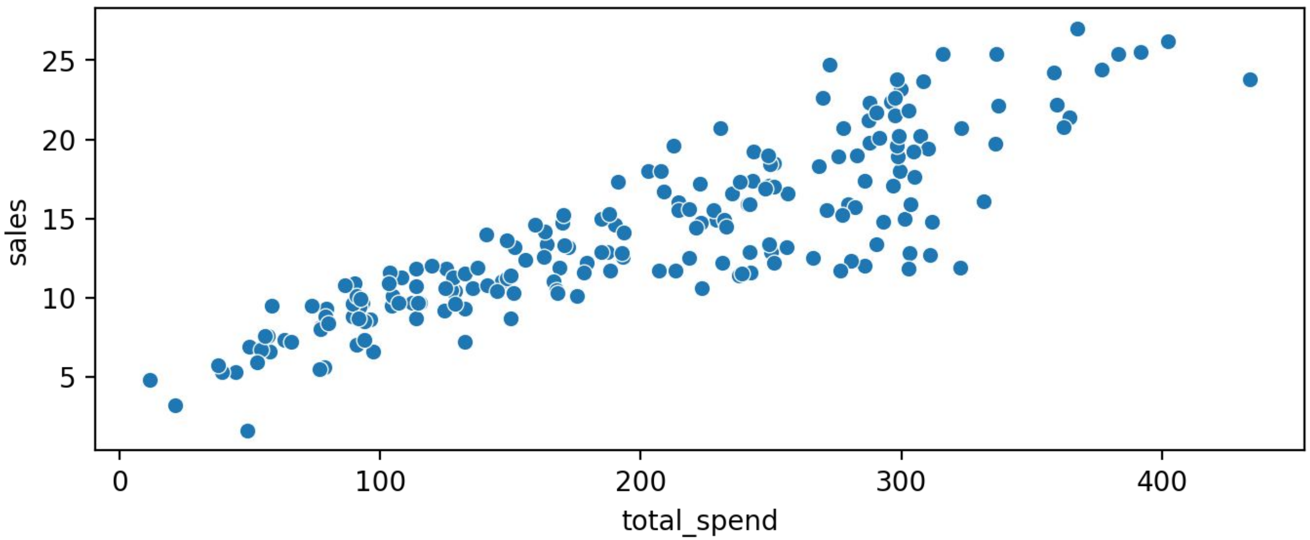


# Linear Regression

- This implies for some new x value I can predict its related y

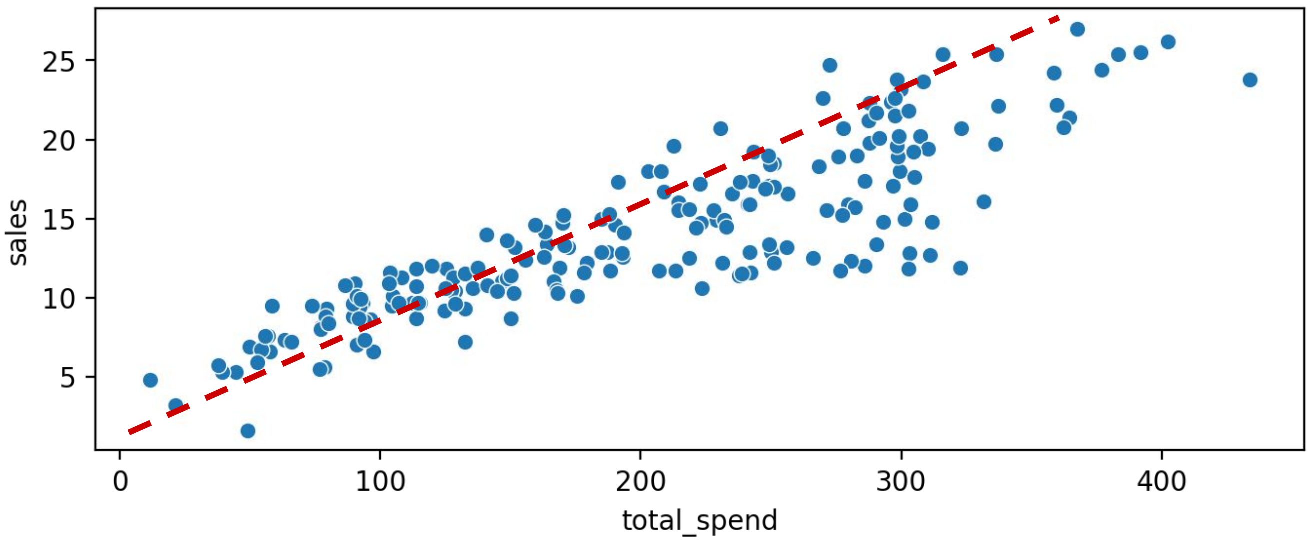


- But what happens with real data? Where do we draw this line?



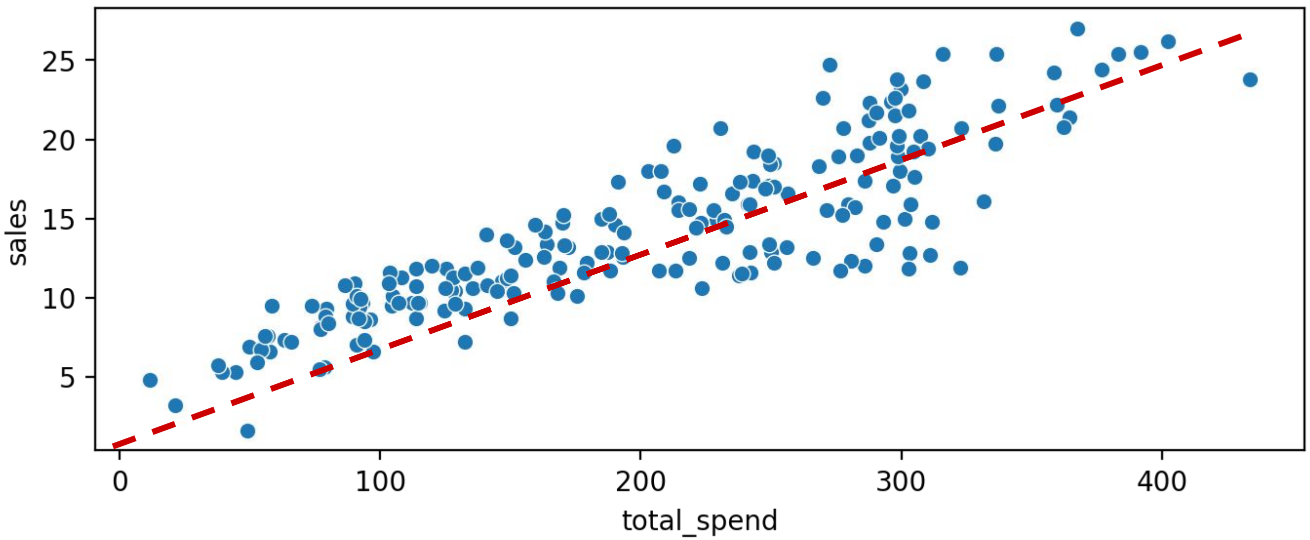
# Linear Regression

- But what happens with real data? Where do we draw this line?



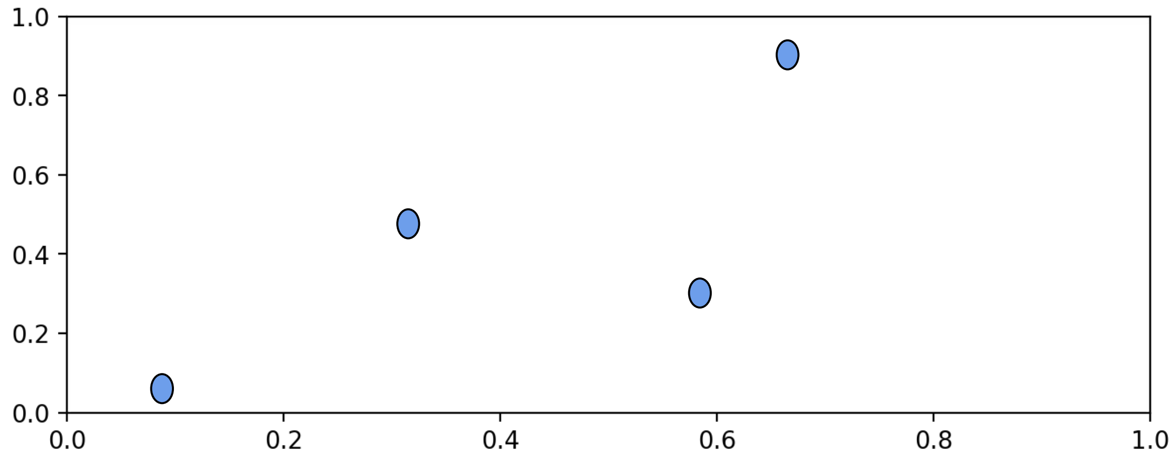
# Linear Regression

- But what happens with real data? Where do we draw this line?



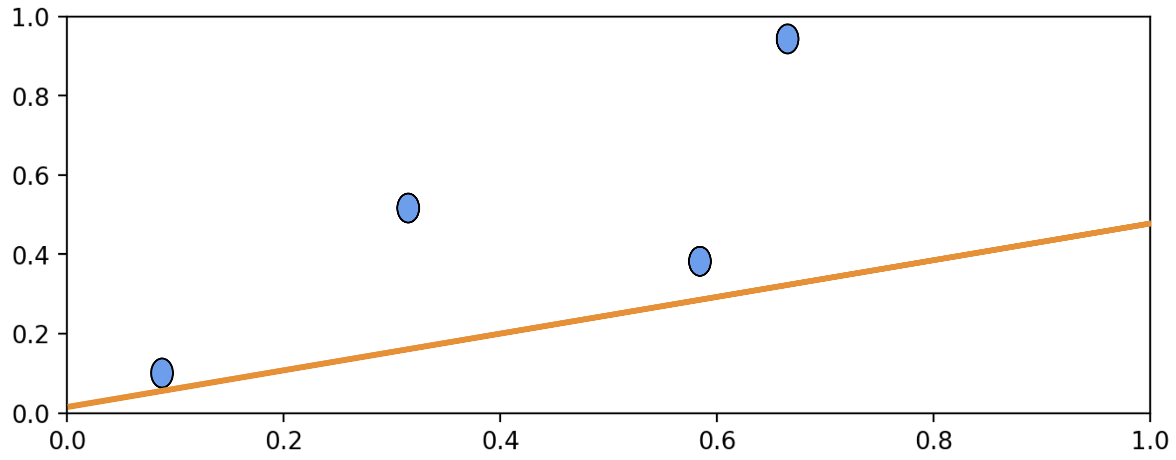
# Linear Regression

- Fundamentally, we understand we want to minimize the overall distance from the points to the line.



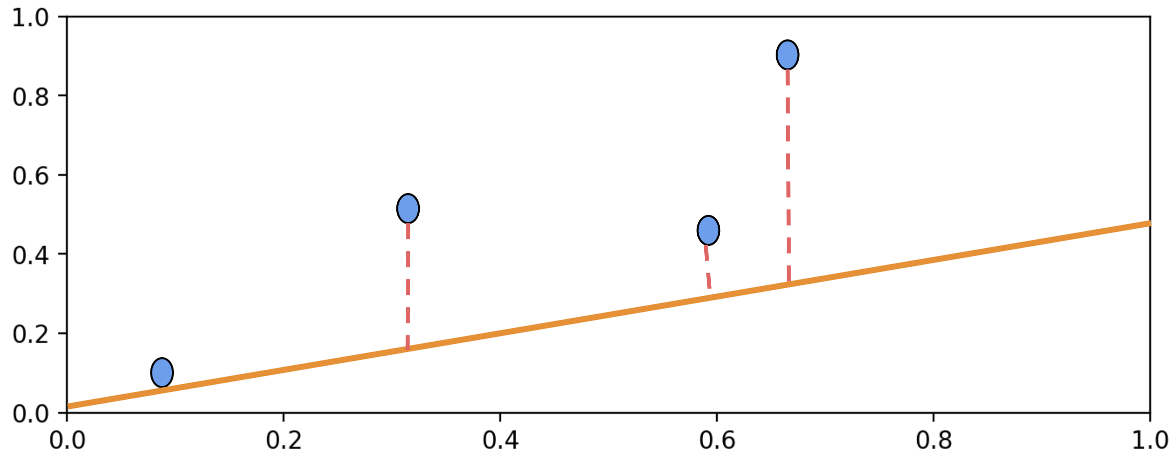
# Linear Regression

- Fundamentally, we understand we want to minimize the overall distance from the points to the line.



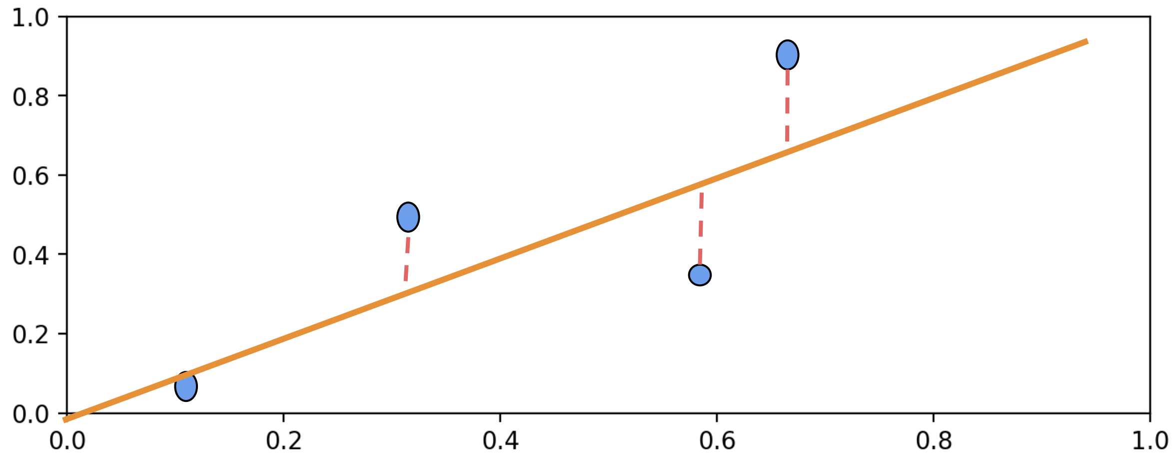
# Linear Regression

- We also know we can measure this error from the real data points to the line, known as the **residual error** $\rightarrow$ we want to minimize it



# Linear Regression

- Some lines will clearly be better fits than others.
- We can also see the residuals can be both positive and negative.

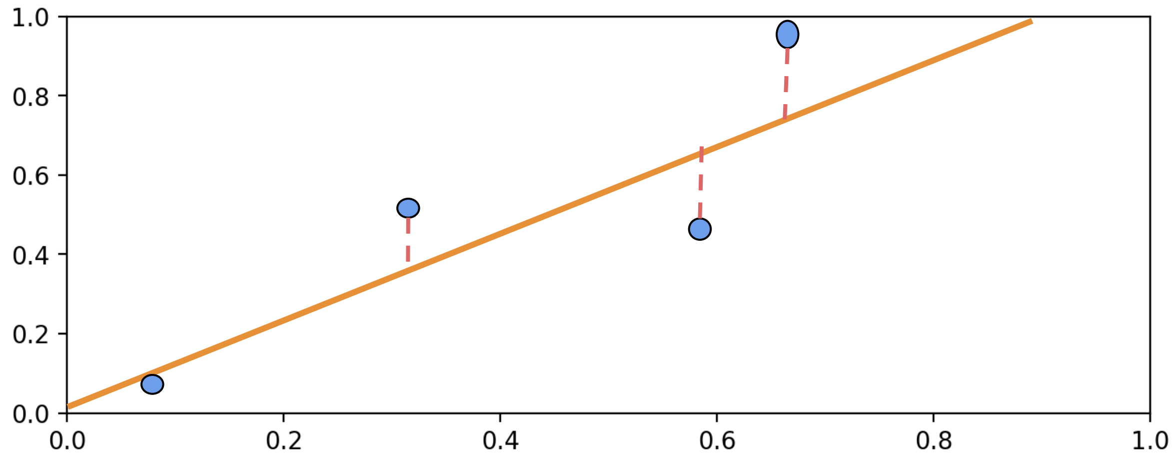


# Ordinary Least Squares

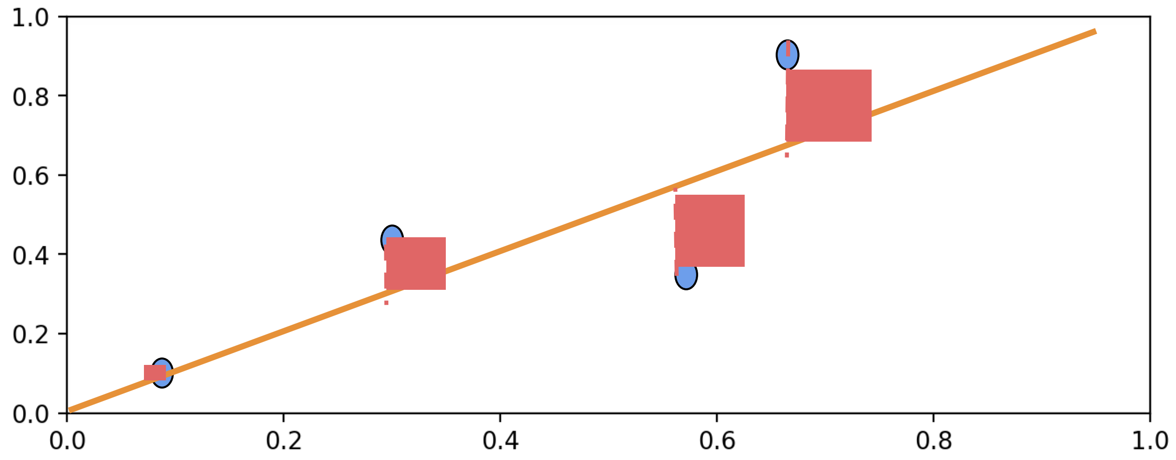
- **Ordinary Least Squares (OLS):**
  - It is a common technique for estimating coefficients of linear regression equations.
  - Works by minimizing the sum of the squares of the differences between the observed dependent variable (values of the variable being observed) in the given dataset and those predicted by the linear function.

# Linear Regression

- We can visualize squared error to minimize:



- We can visualize squared error to minimize:



# Linear Regression

- Having a squared error will help us simplify our calculations later on when setting up a derivative for the purpose of minimization
- Let's continue exploring OLS by converting a real data set into mathematical notation, then working to solve a linear relationship between features and a variable

# Algorithm Theory - OLS Equations

# Linear Regression OLS Theory

- We know the equation of a simple straight line:
  - **$y = mx + b$**
    - m is slope
    - b is intercept with y-axis (determines the distance of the line directly above or below the origin

- We can see for **$y=mx+b$** there is only room for one possible feature x.
- OLS will allow us to directly solve for the slope **m** and intercept **b**.
- We will later see we'll need tools like gradient descent to scale this to multiple features.

# What's Next ?

- Let's explore how we could translate a real data set into mathematical notation for linear regression.
- Then we'll solve a simple case of one feature to explore OLS in action.
- Afterwards we'll focus on gradient descent for real world data set situations.

# Linear Regression

- Linear Regression allows us to build a relationship between multiple **features** to estimate a **target output**.

| Area $m^2$ | Bedrooms | Bathrooms | Price |
|---|---|---|---|
| 200 | 3 | 2 | $500,000 |
| 190 | 2 | 1 | $450,000 |
| 230 | 3 | 3 | $650,000 |
| 180 | 1 | 1 | $400,000 |
| 210 | 2 | 2 | $550,000 |

- Translate the dataset into a generalized form for linear regression

# Linear Regression

- We can translate this data into generalized mathematical notation: Matrix **X** containing multiple features and vector **y** contains some labels that we try to predict

**X** **y**

| Area $m^2$ | Bedrooms | Bathrooms | Price |
|---|---|---|---|
| 200 | 3 | 2 | $500,000 |
| 190 | 2 | 1 | $450,000 |
| 230 | 3 | 3 | $650,000 |
| 180 | 1 | 1 | $400,000 |
| 210 | 2 | 2 | $550,000 |

# Linear Regression

- We can translate this data into generalized mathematical notation…

| **X** | | | **y** |
|---|---|---|---|
| $\mathbf{x_1}$ | $\mathbf{x_2}$ | $\mathbf{x_3}$ | $\mathbf{y}$ |
| $x^1{}_1$ | $x^1{}_1$ | $x^1{}_1$ | $y_1$ |
| $x^2{}_1$ | $x^2{}_1$ | $x^2{}_1$ | $y_2$ |
| $x^3{}_1$ | $x^3{}_1$ | $x^3{}_1$ | $y_3$ |
| $x^4{}_1$ | $x^4{}_1$ | $x^4{}_1$ | $y_4$ |
| $x^5{}_1$ | $x^5{}_1$ | $x^5{}_1$ | $y_5$ |

- Now let’s build out a linear relationship between the features X and label y.

# Linear Regression

- Now let’s build out a linear relationship between the features X and label y.

| X | | | y |
|---|---|---|---|
| $x_1$ | $x_2$ | $x_3$ | $y$ |

# Linear Regression

- Reformat for **y = x** equation

| **y** | **X** | | |
|---|---|---|---|
| $\mathbf{y}$ | $\mathbf{x_1}$ | $\mathbf{x_2}$ | $\mathbf{x_3}$ |

# Linear Regression

- Each feature should have some Beta coefficient associated with it.

| y | X | | |
|---|---|---|---|
| y | $x_1$ | $x_2$ | $x_3$ |

$$\hat{y} = \beta_0 x_0 + \cdots + \beta_n x_n$$

# Linear Regression

- This is the same as the common notation for a simple line: **y=mx+b**

| y | X | | |
|---|---|---|---|
| y | $x_1$ | $x_2$ | $x_3$ |

$$\hat{y} = \beta_0 x_0 + \cdots + \beta_n x_n$$

# Linear Regression

- Each feature should have some Beta coefficient associated with it.

| y | X | | |
|---|---|---|---|
| y | $x_1$ | $x_2$ | $x_3$ |

$$\hat{y} = \beta_0 x_0 + \cdots + \beta_n x_n$$

the predictions for the y value

linear combination of beta coefficients for n number of features

# Linear Regression

- This is stating there is some Beta coefficient for each feature to minimize error.

| y | X | | |
|---|---|---|---|
| $y$ | $x_1$ | $x_2$ | $x_3$ |

$$\hat{y} = \beta_0 x_0 + \cdots + \beta_n x_n$$

# Linear Regression

- We can also express this equation as a sum:

| y | X | | |
|---|---|---|---|
| $y$ | $x_1$ | $x_2$ | $x_3$ |

$$\hat{y} = \beta_0 x_0 + \cdots + \beta_n x_n$$

$$\hat{y} = \sum_{i=0}^{n} \beta_i x_i$$

# Linear Regression

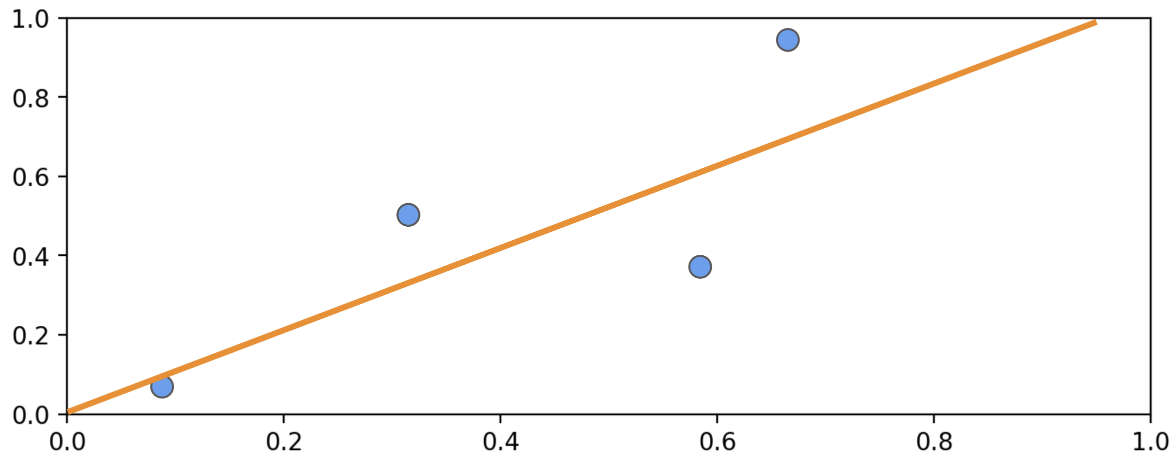
- Note the y hat symbol displays a prediction. There is usually no set of Betas to create a perfect fit to y!

$$\hat{y} = \sum_{i=0}^{n} \beta_i x_i$$

# Linear Regression

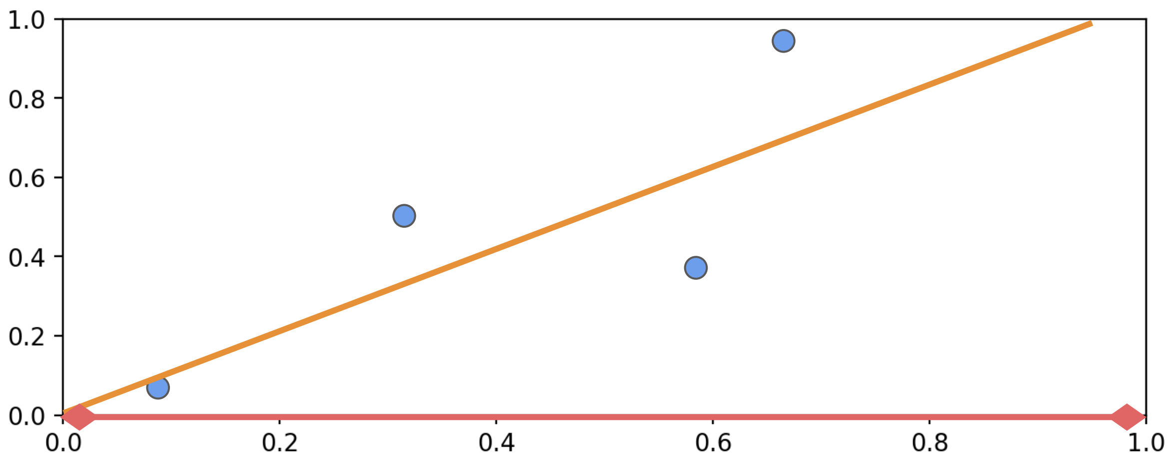
- Line equation:

$$\hat{y} = \sum_{i=0}^{n} \beta_i x_i$$



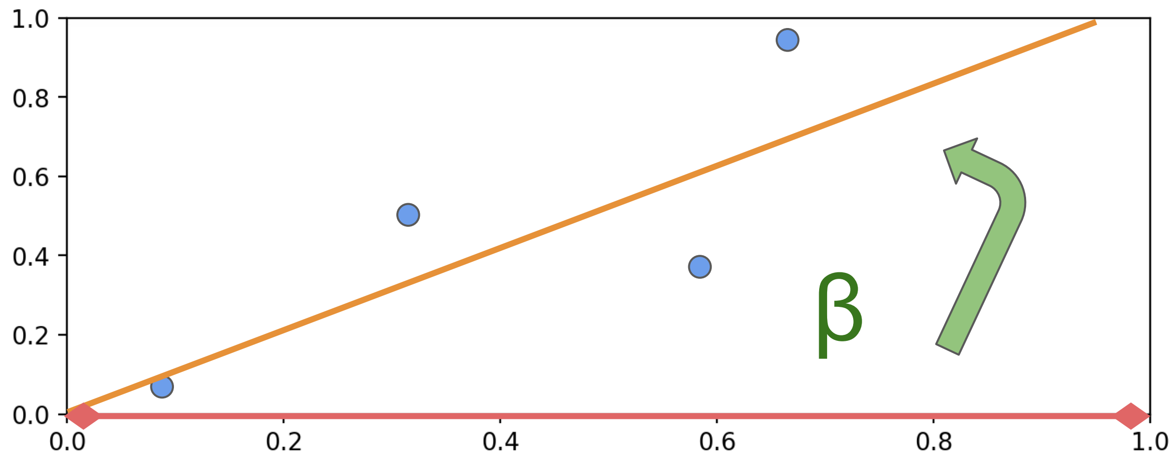
# Linear Regression

$$\hat{y} = \sum_{i=0}^{n} \beta_i \boxed{x_i}$$



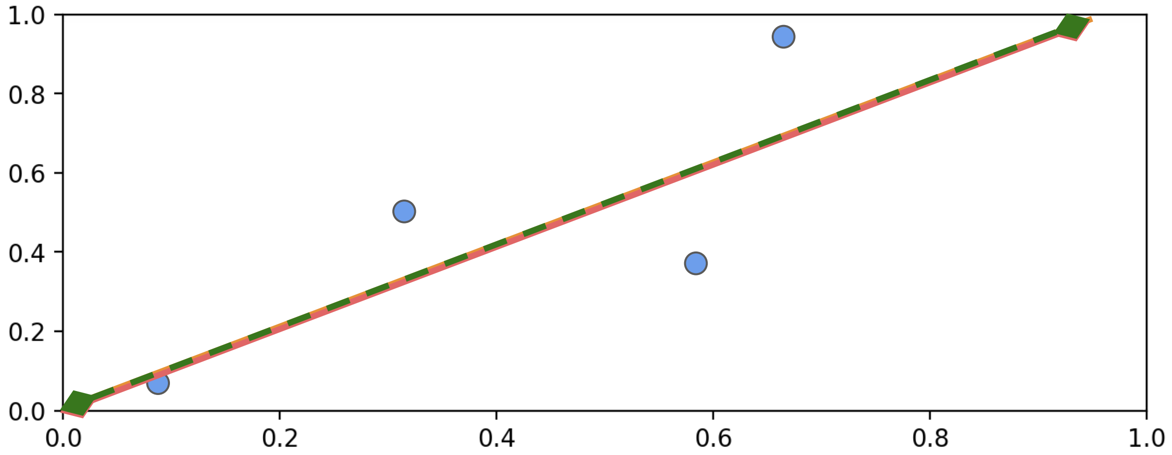
# Linear Regression

$$\hat{y} = \sum_{i=0}^{n} \boxed{\beta_i} \boxed{x_i}$$



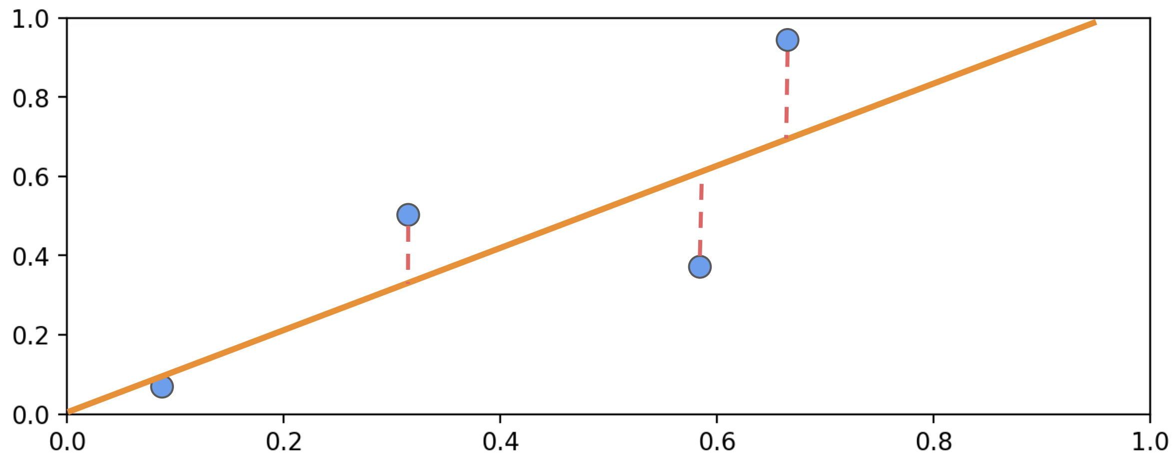
# Linear Regression

$$\hat{y} = \sum_{i=0}^{n} \boxed{\beta_i} \boxed{x_i}$$



# Linear Regression

$$\boxed{\hat{y}} = \sum_{i=0}^{n} \beta_i x_i$$



# Linear Regression

- For simple problems with one X feature we can easily solve for Betas values with an analytical solution.
- Let's quickly solve a simple example problem, then later we will see that for multiple features we will need gradient descent.

# Linear Regression

- Recall the equation of the line follows the form **$y = mx + b$** where
  - $m$ is the **slope** of the line
  - $b$ is where the line crosses the y-axis when $x=0$ ( b is **y-intercept**)


$m>0$ $m<0$ $m=0$

# Linear Regression

- In a linear regression, where we try to formulate the relationship between variables, y=mx + b becomes

$$\boxed{\hat{y} = b_0 + b_1 x}$$

- Our goal is to predict a value of the dependent variable (y) based on the value of an independent variable (x)

# Linear Regression

$$\boxed{\hat{y} = b_0 + b_1 x}$$

- How do we derive *b1* and *b0*

measures the strength of the linear relationship between two variables

$$\boxed{b_1 = \rho_{x,y} \frac{\sigma_y}{\sigma_x}}$$

$\rho_{x,y}$ Pearson Correlation Coefficient

$\sigma_i$ Standard deviation of *i*

# Linear Regression

$$\hat{y} = b_0 + b_1 x$$

- How do we derive *b1* and *b0*

measures the strength of the linear relationship between two variables

$$b_1 = \rho_{x,y} \frac{\sigma_y}{\sigma_x}$$

$\rho_{x,y}$ Pearson Correlation Coefficient

$\sigma_i$ Standard deviation of *i*

the degree of dispersion or the scatter of the data points relative to its mean and is calculated as the square root of the variance.

# Linear Regression

$$\hat{y} = b_0 + b_1 x$$

- How do we derive *b1* and *b0*

measures the strength of the linear relationship between two variables

$$b_1 = \rho_{x,y} \frac{\sigma_y}{\sigma_x}$$

$\rho_{x,y}$ Pearson Correlation Coefficient
$\sigma_i$ Standard deviation of *i*

the degree of dispersion or the scatter of the data points relative to its mean and is calculated as the square root of the variance.

$$= \frac{\sum (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum (x_i - \bar{x})^2 \sum (y_i - \bar{y})^2}} \cdot \frac{\sqrt{\frac{\sum (y - \bar{y})^2}{n}}}{\sqrt{\frac{\sum (x - \bar{x})^2}{n}}} = \frac{\sum (x - \bar{x})(y - \bar{y})}{\sum (x - \bar{x})^2}$$

# Linear Regression

$$\boxed{\hat{y} = b_0 + b_1 x}$$

- How do we derive *b1* and *b0*

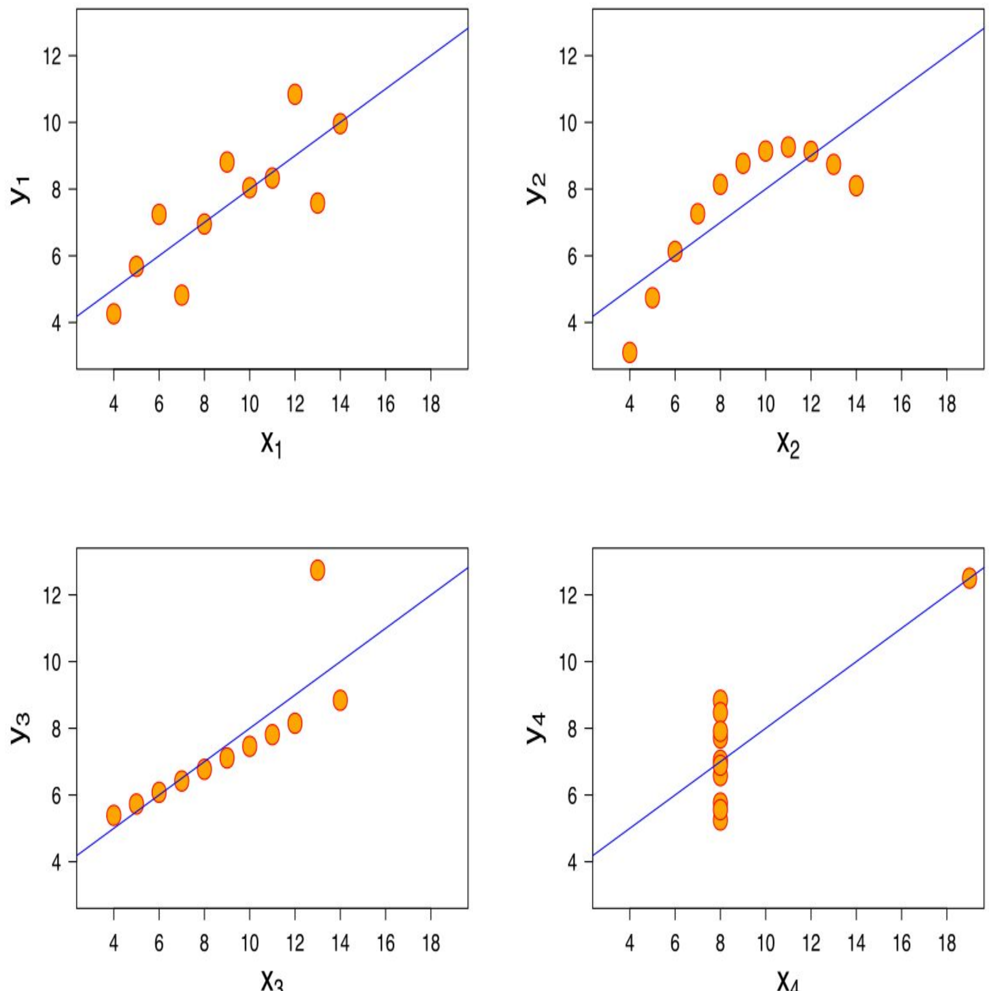
$$\boxed{b_1 = \frac{\sum(x - \bar{x})(y - \bar{y})}{\sum(x - \bar{x})^2}}$$

$$\boxed{b_0 = \bar{y} - b_1 \bar{x}}$$

# Limitations of Linear Regression

- Anscombe's Quartet shows the pitfalls of relying on pure calculations

- Each graph results in the same calculated regression line



# Linear Regression Example

- A factory manager wants to find the relationship between the number of operational hours of the plant in a week and weekly productivity
- Here the **independent variable** x is hours of operation, and the **dependent variable** y is production volume.

# Linear Regression Example

- A factory manager wants to find the relationship between the number of operational hours of the plant in a week and weekly productivity
- Here the **independent variable** x is hours of operation, and the **dependent variable** y is production volume.
- We want to build a linear relation between hours of operations and production volume

# Linear Regression Example

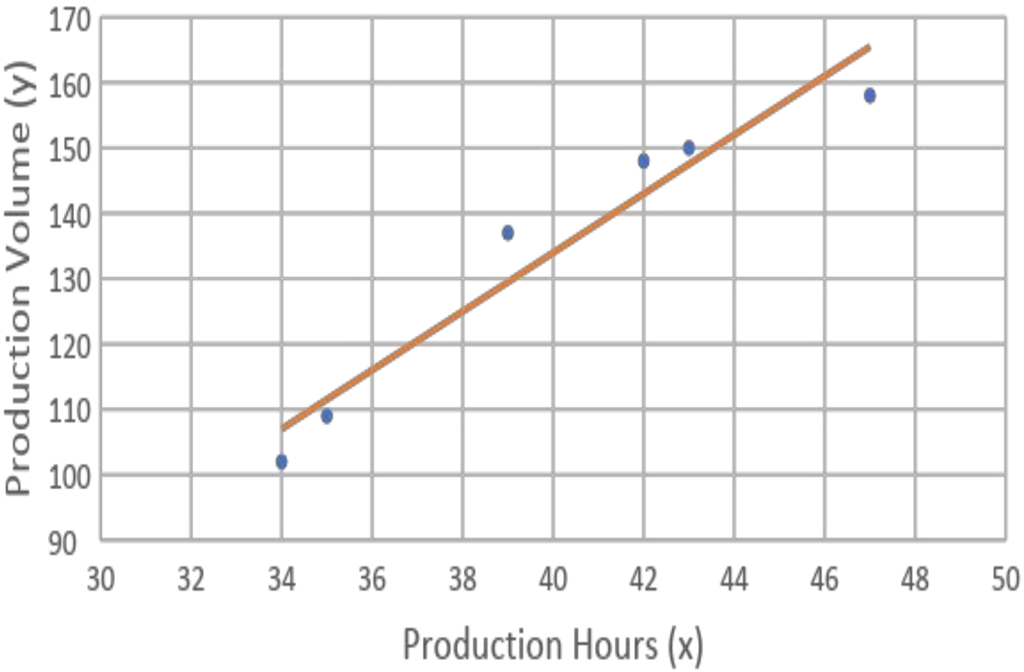
- The manager develops the following table

| Production Hours(x) | Production Volume(y) |
|---|---|
| 34 | 102 |
| 35 | 109 |
| 39 | 137 |
| 42 | 148 |
| 43 | 150 |
| 47 | 158 |

# Linear Regression Example

- Plot the data

| Production Hours(x) | Production Volume(y) |
|---|---|
| 34 | 102 |
| 35 | 109 |
| 39 | 137 |
| 42 | 148 |
| 43 | 150 |
| 47 | 158 |




Is there a linear pattern? Can we plot a best fit line

# Linear Regression Example

- Run Calculations:

| Production Hours(x) | Production Volume(y) |
|---|---|
| 34 | 102 |
| 35 | 109 |
| 39 | 137 |
| 42 | 148 |
| 43 | 150 |
| 47 | 158 |

$$\boxed{\hat{y} = b_0 + b_1 x}$$

$$b_1 = \frac{\sum(x - \bar{x})(y - \bar{y})}{\sum(x - \bar{x})^2}$$

$$b_0 = \bar{y} - b_1\bar{x}$$

# Linear Regression Example

| | | Production Hours(x) | Production Volume(y) | $(x-\bar{x})$ | $(y-\bar{y})$ | $(x-\bar{x})\,(y-\bar{y})$ | $(x-\bar{x})^2$ |
|---|---|---|---|---|---|---|---|
| 2 | | 34 | 102 | -6 | -32 | 192 | 36 |
| 3 | | 35 | 109 | -5 | -25 | 125 | 25 |
| 4 | | 39 | 137 | -1 | 3 | -3 | 1 |
| 5 | | 42 | 148 | 2 | 14 | 28 | 4 |
| 6 | | 43 | 150 | 3 | 16 | 48 | 9 |
| 7 | | 47 | 158 | 7 | 24 | 168 | 49 |
| 8 | $\bar{x}, \bar{y}$ | 40 | 134 | | Sum = | 558 | 124 |

$$\boxed{\hat{y} = b_0 + b_1 x}$$

$$b_1 = \frac{\sum(x-\bar{x})(y-\bar{y})}{\sum(x-\bar{x})^2}$$

$$b_0 = \bar{y} - b_1\bar{x}$$

# Linear Regression Example

| | | Production Hours(x) | Production Volume(y) | $(x-\bar{x})$ | $(y-\bar{y})$ | $(x-\bar{x})(y-\bar{y})$ | $(x-\bar{x})^2$ |
|---|---|---|---|---|---|---|---|
| 2 | | 34 | 102 | -6 | -32 | 192 | 36 |
| 3 | | 35 | 109 | -5 | -25 | 125 | 25 |
| 4 | | 39 | 137 | -1 | 3 | -3 | 1 |
| 5 | | 42 | 148 | 2 | 14 | 28 | 4 |
| 6 | | 43 | 150 | 3 | 16 | 48 | 9 |
| 7 | | 47 | 158 | 7 | 24 | 168 | 49 |
| 8 | $\bar{x},\bar{y}$ | 40 | 134 | | Sum = | 558 | 124 |

$$\boxed{\hat{y} = b_0 + b_1 x}$$

$$b_1 = \frac{\sum(x-\bar{x})(y-\bar{y})}{\sum(x-\bar{x})^2}$$

$$b_0 = \bar{y} - b_1\bar{x}$$

$$b_1 = \frac{558}{124} = 4.5$$

$$b_0 = 134 - (4.5 \times 40) = -46$$

$$\hat{y} = -46 + 4.5x$$

# Linear Regression Example

- Based on this formula, if the manager wants to produce 125 units per week, the plant should run for

| Production Hours(x) | Production Volume(y) |
| --- | --- |
| 34 | 102 |
| 35 | 109 |
| 39 | 137 |
| 42 | 148 |
| 43 | 150 |
| 47 | 158 |

$$125 = -46 + 4.5x$$
$$x = 38\ hours\ per\ week$$

# Linear Regression

- As we expand to more than a single feature however, an analytical solution quickly becomes unscalable.
- Instead of OLS we shift focus on **minimizing** a **cost** function with **gradient descent**.

# Linear Regression

- We can use **gradient descent** to solve a **cost function** to calculate Beta values!
- We'll work on developing a cost function to minimize

$$\hat{y} = \sum_{i=0}^{n} \beta_i x_i$$

# Algorithm Theory - Cost Function

# What we know so far

- Linear Relationships
  - **y = mx+b**
- OLS
  - Solve simple linear regression ($b_0$ and $b_1$)
- Not scalable for multiple features
- Translating real data to Matrix Notation
- Generalized formula for Beta coefficients

$$\hat{y} = \sum_{i=0}^{n} \beta_i x_i$$

- Recall we are **searching for Beta values** for a best-fit line.
- The equation below simply defines our line, but how to choose beta coefficients?

$$\hat{y} = \sum_{i=0}^{n} \beta_i x_i$$

- We've decided to define a "best-fit" as **minimizing the squared error**$\longrightarrow$ let's define our cost/loss function or actual error we want to minimize and how can we relate it back to the beta coefficients

$$\hat{y} = \sum_{i=0}^{n} \beta_i x_i$$

# Residual Error

- The residual error (error between our prediction and true value) for some row j is: $y^j - \hat{y}^j$

# Residual Error

- The residual error (error between our prediction and true value) for some row j is: $y^j - \hat{y}^j$
- Squared Error for some row j is then: $\left(y^j - \hat{y}^j\right)^2$

# Residual Error

- The residual error (error between our prediction and true value) for some row j is: $$y^j - \hat{y}^j$$
- Squared Error for some row j is then: $$\left(y^j - \hat{y}^j\right)^2$$
- Sum of squared errors for **m** rows is then: $$\sum_{j=1}^{m}\left(y^j - \hat{y}^j\right)^2$$

# Residual Error

- The residual error (error between our prediction and true value) for some row j is: $$y^j - \hat{y}^j$$

- Squared Error for some row j is then: $$\left(y^j - \hat{y}^j\right)^2$$

- Sum of squared errors for **m** rows is then: $$\sum_{j=1}^{m}\left(y^j - \hat{y}^j\right)^2$$

- Average squared error for m rows is then: $$\frac{1}{m}\sum_{j=1}^{m}\left(y^j - \hat{y}^j\right)^2$$

- Exactly what we need for a **cost function**!

$$\frac{1}{m}\sum_{j=1}^{m}\left(y^j - \hat{y}^j\right)^2$$

- Begin by defining a cost function **J**.

$$J(\boldsymbol{\beta})$$

- A cost function is defined by some measure of error.
- This means we wish to minimize the cost function$\longrightarrow$ choose the values of beta that will minimize the error

- Our cost function can be defined by the squared error:

$$J(\boldsymbol{\beta}) = \frac{1}{2m} \sum_{j=1}^{m} \left( y^j - \hat{y}^j \right)^2$$

- Note lowercase *j* is the specific data row.

$$J(\boldsymbol{\beta}) = \frac{1}{2m} \sum_{j=1}^{m} \left( y^j - \hat{y}^j \right)^2$$

- Want to minimize cost for set of Betas.

$$\boxed{J(\boldsymbol{\beta})} = \frac{1}{2m}\sum_{j=1}^{m}\left(y^j - \hat{y}^j\right)^2$$

- Error between real y and predicted ŷ

$$J(\beta) = \frac{1}{2m}\sum_{j=1}^{m}\left(\boxed{y^j - \hat{y}^j}\right)^2$$

- Squaring corrects for negative and positive errors.

$$J(\beta) = \frac{1}{2m} \boxed{\sum_{j=1}^{m}} \left(y^j - \hat{y}^j\right)^{\boxed{2}}$$

- Summing error for m rows.

$$J(\beta) = \frac{1}{2m} \sum_{j=1}^{\boxed{m}} \left(y^j - \hat{y}^j\right)^2$$

- Summing error for m rows.

$$J(\beta) = \frac{1}{2m} \boxed{\sum_{j=1}^{m}} \left(y^j - \hat{y}^j\right)^2$$

- Divide by m to get mean

$$J(\boldsymbol{\beta}) = \boxed{\frac{1}{2m}} \sum_{j=1}^{m} \left( y^j - \hat{y}^j \right)^2$$

- Additional ½ is for convenience for derivative.
- Recall: when we want to minimize we take the derivative and set it equal to 0

$$J(\boldsymbol{\beta}) = \frac{1}{\boxed{2}m} \sum_{j=1}^{m} \left( y^j - \hat{y}^j \right)^{\boxed{2}}$$

- What is ŷ ?

$$J(\beta) = \frac{1}{2m}\sum_{j=1}^{m}\left(y^j - \boxed{\hat{y}}^j\right)^2$$

- It will be a function of Betas and Features!

$$J(\beta) = \frac{1}{2m}\sum_{j=1}^{m}\left(y^j - \hat{y}^j\right)^2$$

$$= \frac{1}{2m}\sum_{j=1}^{m}\left(y^j - \sum_{i=0}^{n}\beta_i x_i^j\right)^2$$

- Recall from calculus to minimize a function we can take its derivative and set it equal to zero.

$$\frac{\partial J}{\partial \beta_k}(\boldsymbol{\beta}) = \frac{\partial}{\partial \beta_k}\left(\frac{1}{2m}\sum_{j=1}^{m}\left(y^j - \sum_{i=0}^{n}\beta_i x_i^j\right)^2\right)$$
$$= \frac{1}{m}\sum_{j=1}^{m}\left(y^j - \sum_{i=0}^{n}\beta_i x_i^j\right)(-x_k^j)$$

- Recall from calculus to minimize a function we can take its derivative and set it equal to zero.

$$\begin{aligned}\boxed{\frac{\partial J}{\partial \beta_k}}(\boldsymbol{\beta}) &= \boxed{\frac{\partial}{\partial \beta_k}}\left(\frac{1}{2m}\sum_{j=1}^{m}\left(y^j - \sum_{i=0}^{n}\beta_i x_i^j\right)^2\right) \\ &= \frac{1}{m}\sum_{j=1}^{m}\left(y^j - \sum_{i=0}^{n}\beta_i x_i^j\right)(-x_k^j)\end{aligned}$$

- Recall from calculus to minimize a function we can take its derivative and set it equal to zero.

$$\frac{\partial J}{\partial \beta_k}(\boldsymbol{\beta}) = \frac{\partial}{\partial \beta_k}\left(\frac{1}{\boxed{2}m}\sum_{j=1}^{m}\left(y^j - \sum_{i=0}^{n}\beta_i x_i^j\right)^{\boxed{2}}\right)$$

$$= \frac{1}{m}\sum_{j=1}^{m}\left(y^j - \sum_{i=0}^{n}\beta_i x_i^j\right)(-x_k^j)$$

- Unfortunately, it is not scalable to try to get an analytical solution to minimize this cost function.
- In the next lecture we will learn to use **gradient descent** to minimize this **cost function**.