FinTech @ IU Python Session #5 – Plotting & Intro to ML

Prepared by

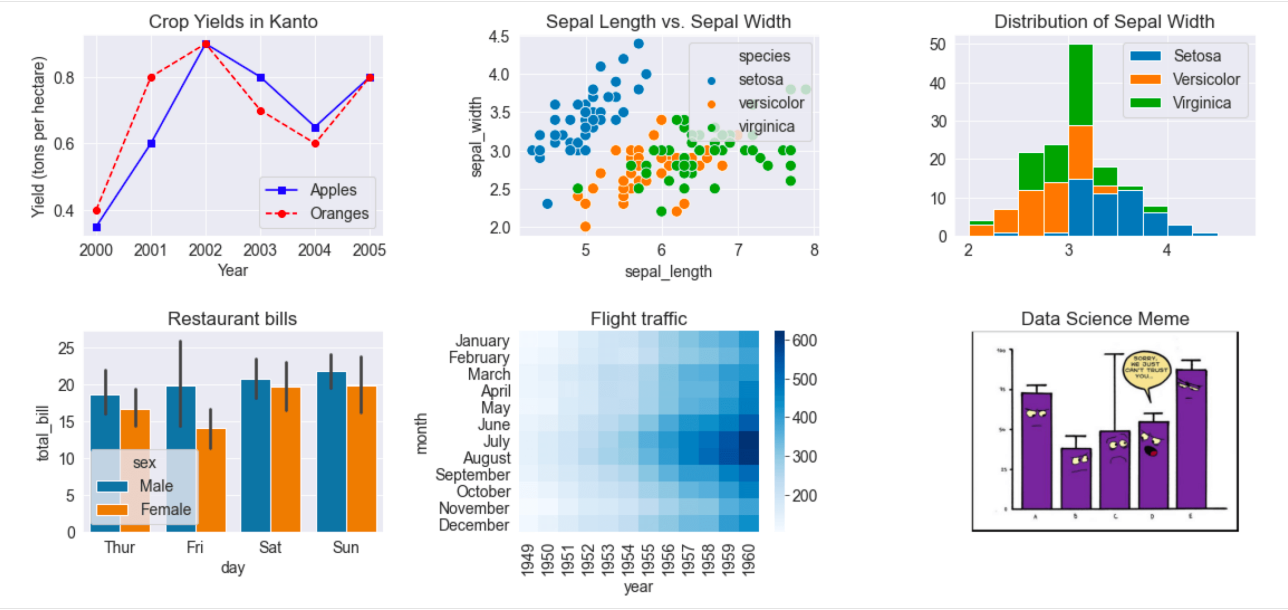
Gabriel Shores
Director of Technology – FinTech @ IU
https://www.linkedin.com/in/gabriel-shores-379b81291/

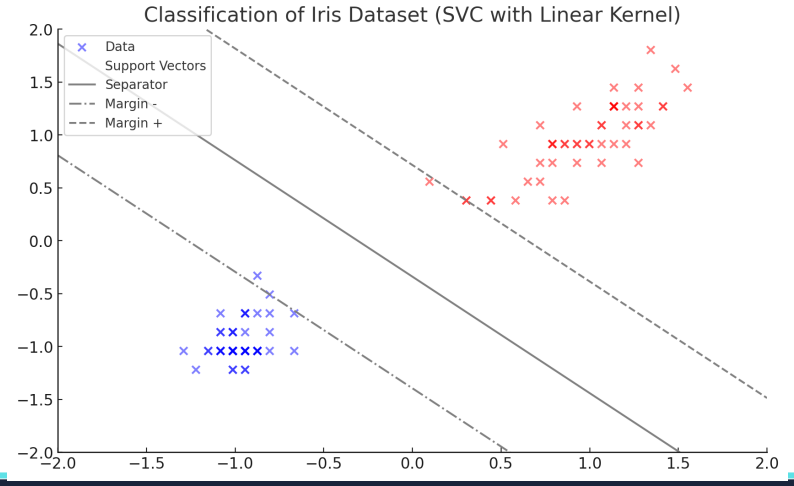
Cameron Nelson
Fund Equities Lead– FinTech @ IU
https://www.linkedin.com/in/cameron-j-nelson/

11/08/2024

Graphs

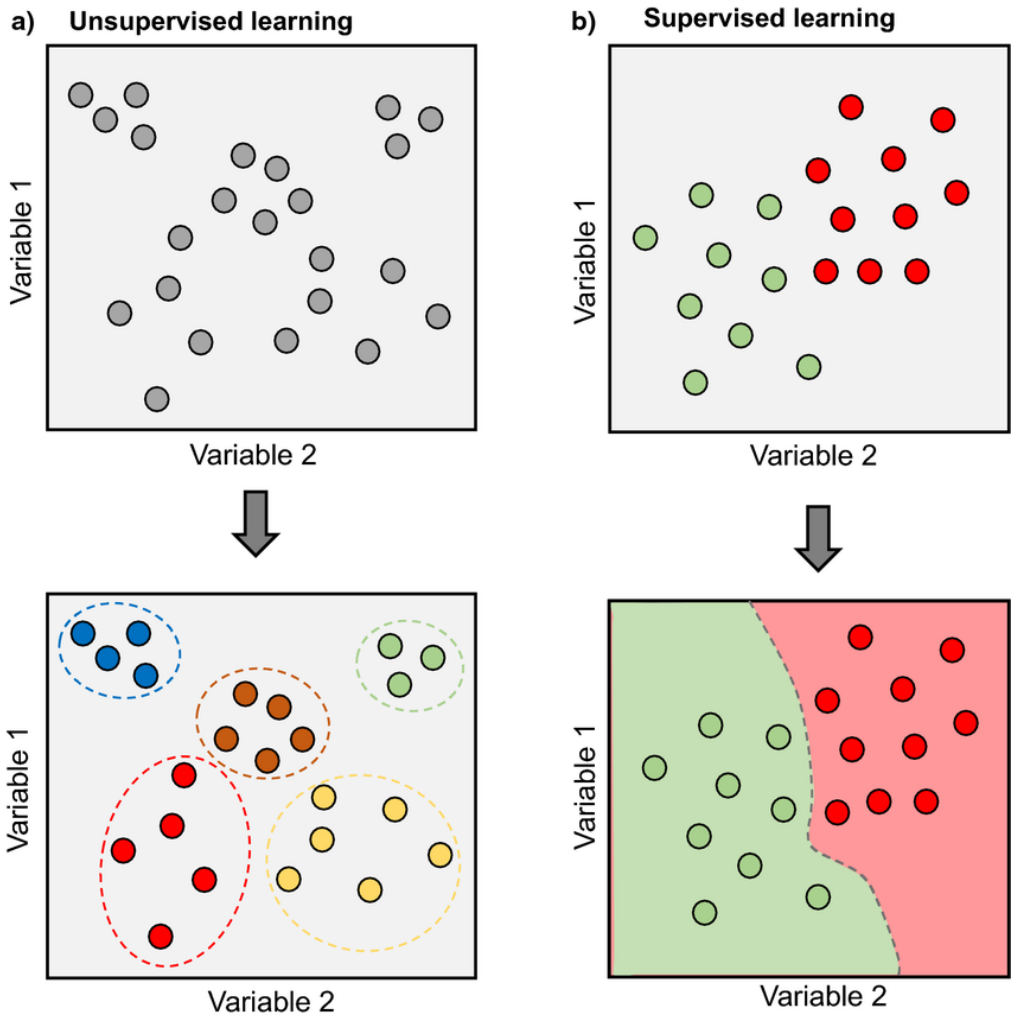
- Analyzing data is often easier through visualizations
- The matplotlib library allows users to make different types of graphs
- 3 parts to a graph
- Figures are the backgrounds which everything lies upon
- Axes contain the different axes, labels, titles, and gridlines
- Plots/subplots are the visual representations of data and includes line charts, bar charts, and scatter plots





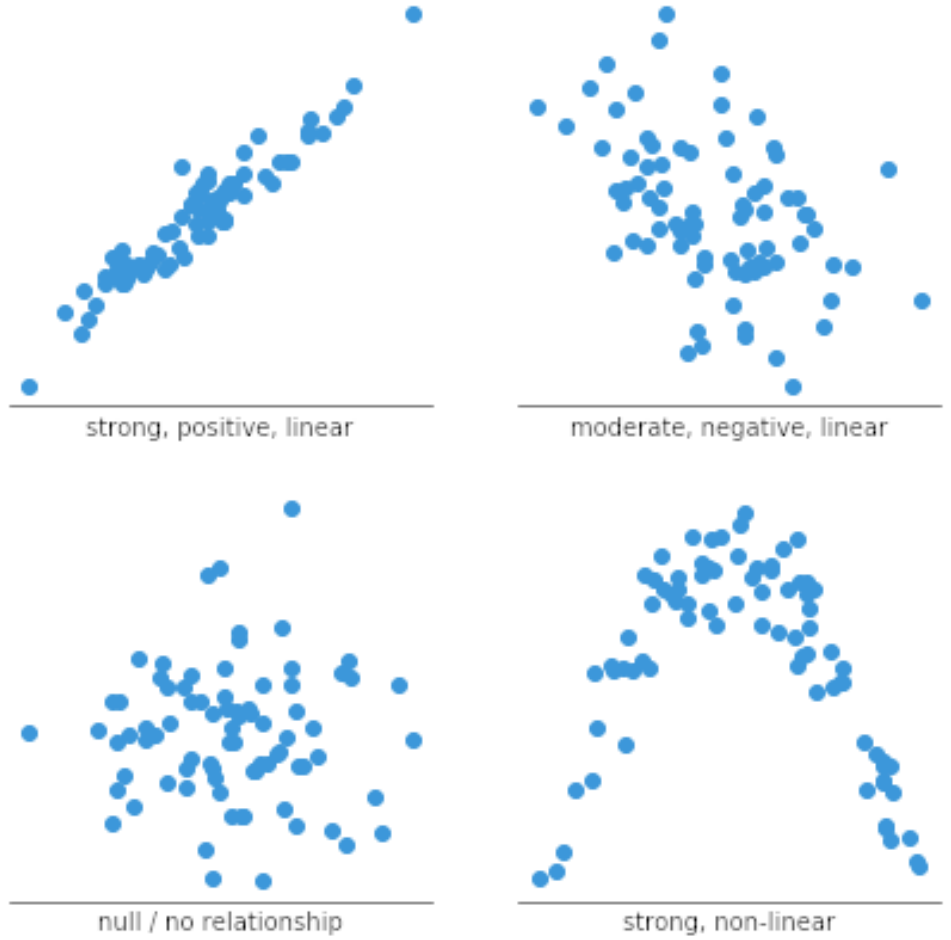
Machine Learning

- Machine learning involves learning from data and making predictions or decisions without explicitly programming it
- Supervised learning – Model is trained on labeled data (Predicting prices, image classification, spam detection)
- Unsupervised learning – Determine labels / group data (Customer segmentation, detecting unusual transactions)
- Reinforcement learning – Force models to learn through positive or negative feedback (Game AI, recommendation models)



Classifying House Prices

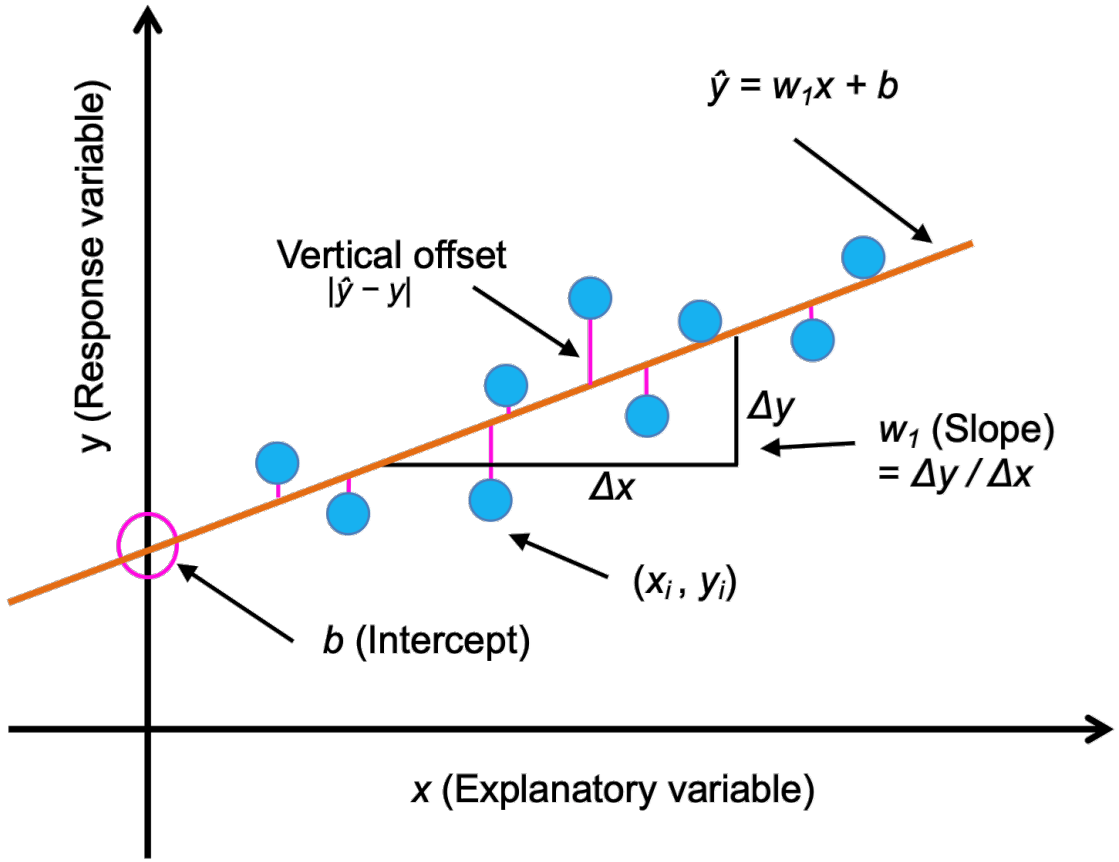
- Features – The measurable characteristics or variables provided in the dataset used to predict an outcome
- Explanatory Variables – The variables that classify and categorize data
- Let's say we want to predict the price of a house given some number of rooms, location, and other such factors along with the associated prices
- Let's try to select one of these features and make predictions with it



Classifying House Prices

- Looking at our scatterplots, one way to predict data is with a line in the form $\hat{y} = w_1x + b$
- For more features we can use $\hat{y} = w_1x_1 + w_2x_2 + \ldots + w_nx_n + b$
- To measure how close our model, the line of best fit, is, we need to calculate how far away each point is from it
- We use what is called the cost function, in this case the Mean Squared Error

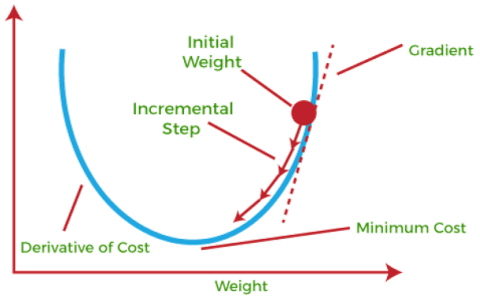
$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y^{(i)} - \hat{y}^{(i)})^2$$

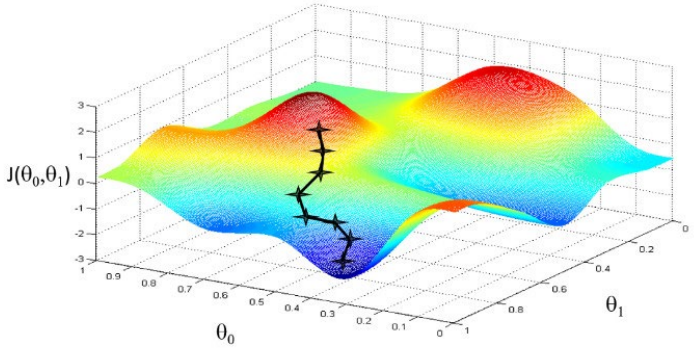


Gradient Descent

- Let's try to get the cost function to its minimum value
- For each model, lets calculate the cost function and which direction we can go to decrease the cost function, opposite of the direction it increases the most
- Adjust the weights and do it again until we hit the point which produces the least cost
- The gradient is the derivative in calculus
- Learning rate adjusts how much we want to move at each increment

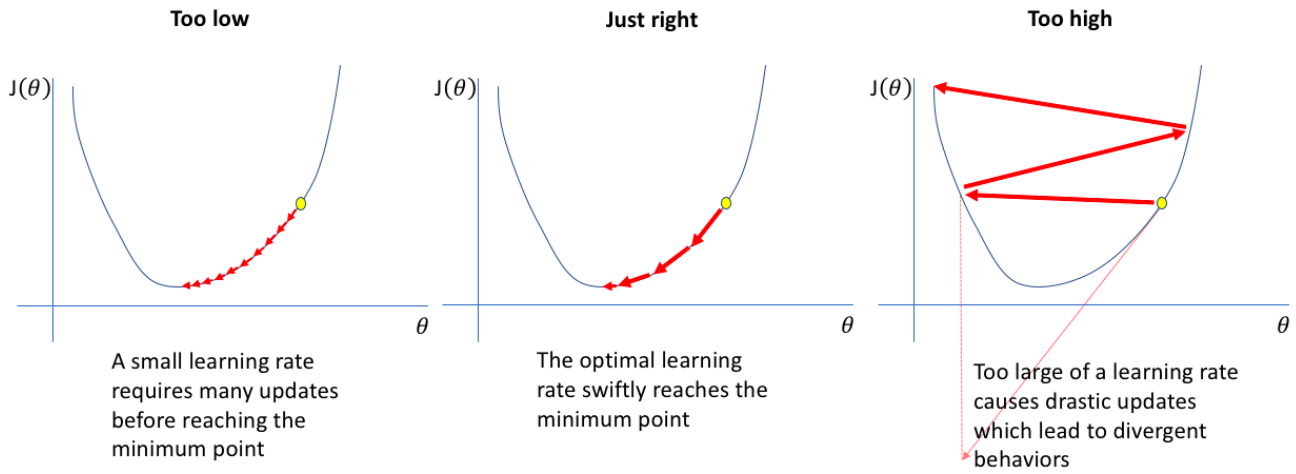
Understanding Gradient Descent





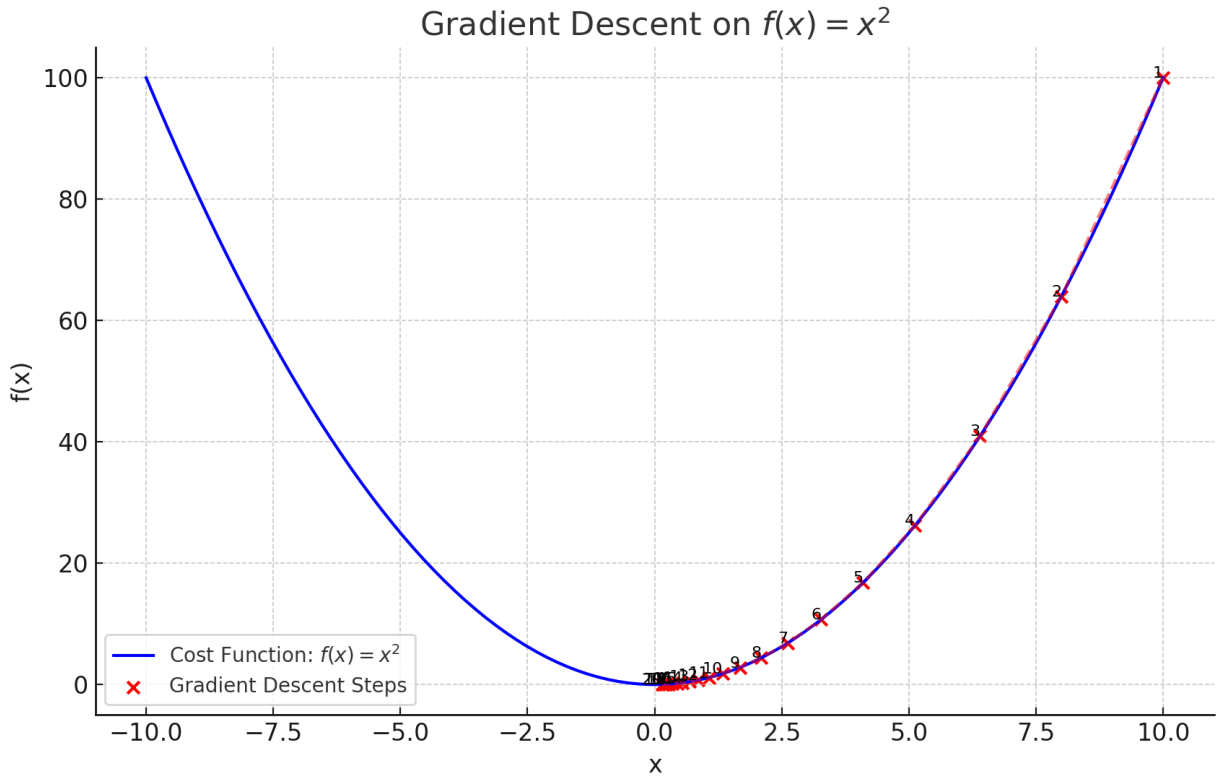
$$f(x) = x^2 - 4x + 3$$

x = x - learning_rate * grad



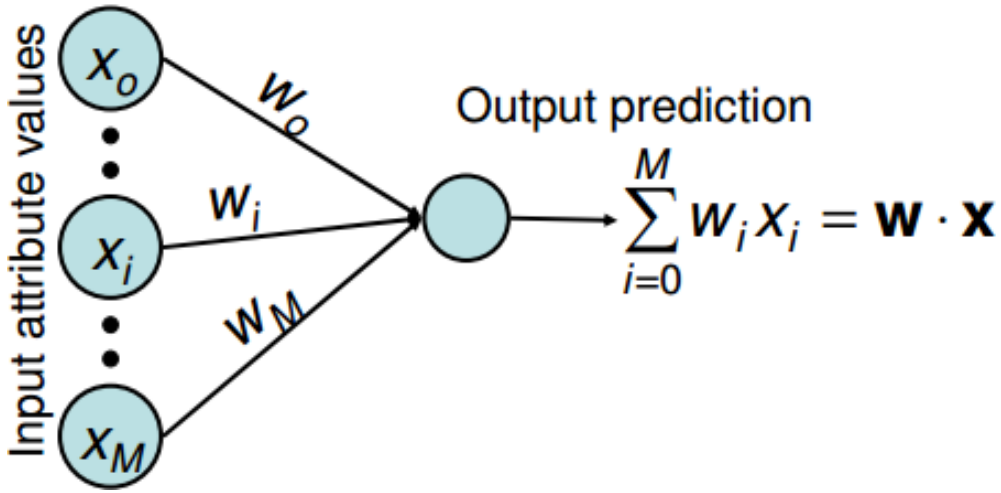
Gradient Descent

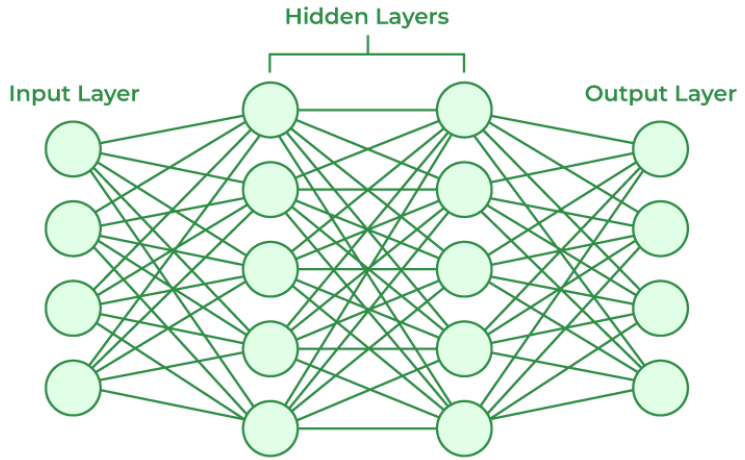
```python
# Define the cost function
def cost_function(x):
    return x ** 2

# Define the derivative (gradient) of the cost function
def gradient(x):
    return 2 * x

# Gradient descent algorithm
def gradient_descent(starting_x, learning_rate, num_iterations):
    x = starting_x  # Start at an initial value of x
    for i in range(num_iterations):
        grad = gradient(x)  # Calculate the gradient at the current x
        x = x - learning_rate * grad  # Move in opposite direction of the gradient
    return x

# Parameters
starting_x = 10  # Starting point
learning_rate = 0.1  # Step size
num_iterations = 20  # Number of steps

# Run gradient descent
gradient_descent(starting_x, learning_rate, num_iterations)# 0
```



Applications

- While linear regression is useful, it doesn't classify data, and can't fit more complex data
- However, the ideas used in linear regression are present in almost all ML algorithms, with neural networks being especially similar





The End!

- Thank you!