# CS 480

## *Introduction to Artificial Intelligence*

**November 11th, 2021**

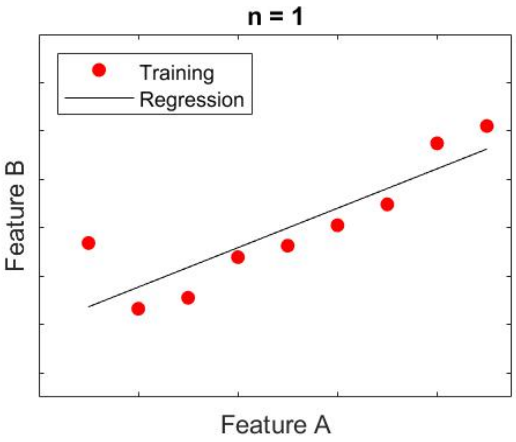
# Announcements / Reminders

- **Quiz #02: due on Sunday (11/14/21) at 11:00 PM CST**
- **Programming Assignment #02:**
  - **weekend**
- **Written Assignment #03:**
  - **weekend**
- **Final Exam:**
  - **Thursday, December 2nd, 2021 (during lecture time)**

# Plan for Today

- **Casual Introduction to Machine Learning**

# Univariate Linear Regression




**Real function:** $y = w_1 * x + w_0$

**Hypothesis / model:** $h_{\boldsymbol{w}}(x) = w_1 * x + w_0$

**where** $\boldsymbol{w} = \langle w_0, w_1 \rangle$ **are coefficients / weights.**

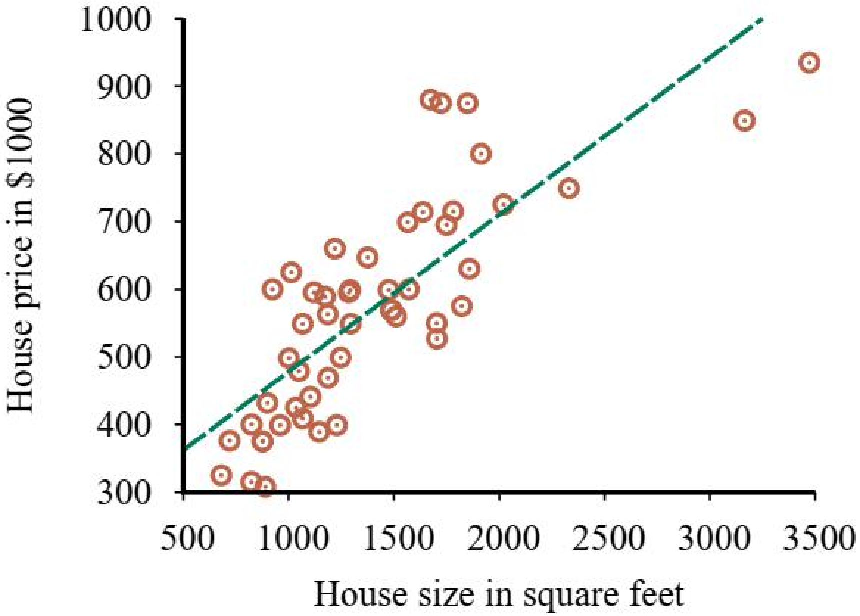
**Squared-error loss function:**

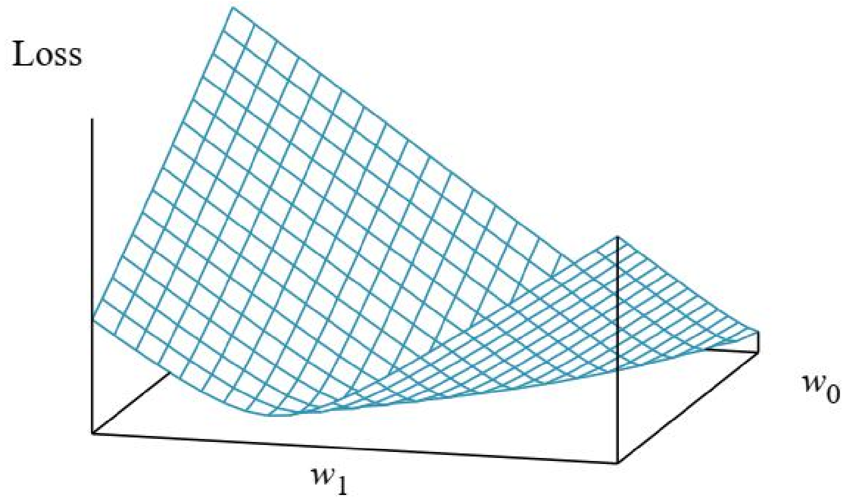
$$Loss(h_{\boldsymbol{w}}) = \sum_{j=1}^{N} \left(y_j - h_{\boldsymbol{w}}(x_j)\right)^2 = \sum_{j=1}^{N} \left(y_j - (w_1 * x_j + w_0)\right)^2$$

**We want to find** $\boldsymbol{w}^* = \underset{\boldsymbol{w}}{argmin}\, Loss(h_{\boldsymbol{w}})$:

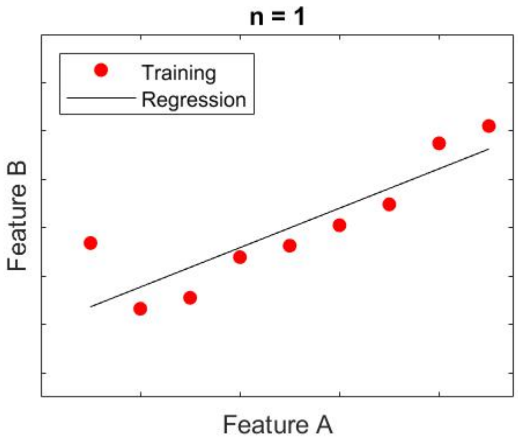
**solve** $\frac{\partial Loss(h_{\boldsymbol{w}})}{\partial w_0} = 0\,,\ \frac{\partial Loss(h_{\boldsymbol{w}})}{\partial w_1} = 0$

# Weight / Parameter Space



# Gradient Descent




**Givne a squared-error loss function:**

$$Loss(h_{\boldsymbol{w}}) = \sum_{j=1}^{N} \left(y_j - h_{\boldsymbol{w}}(x_j)\right)^2 = \sum_{j=1}^{N} \left(y_j - (w_1 * x_j + w_0)\right)^2$$

**We want to find** $\boldsymbol{w}^* = \underset{\boldsymbol{w}}{argmin}\, Loss(h_{\boldsymbol{w}})$:

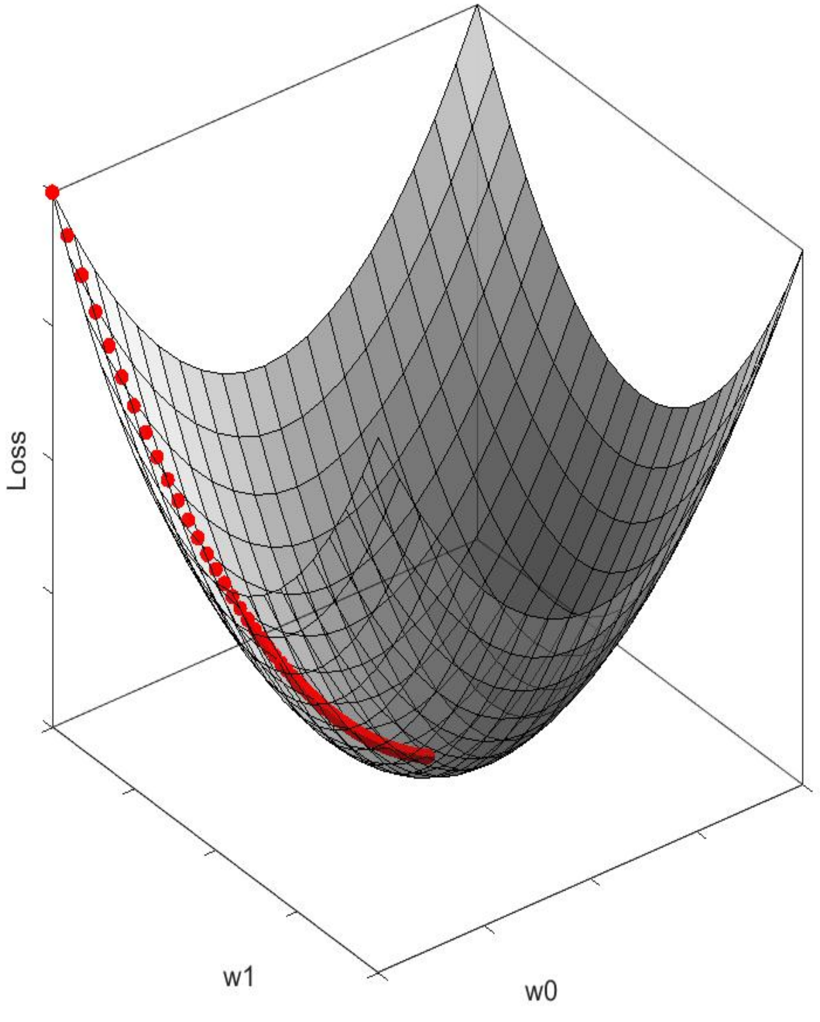
**solve** $\frac{\partial Loss(h_{\boldsymbol{w}})}{\partial w_0} = 0$ , $\frac{\partial Loss(h_{\boldsymbol{w}})}{\partial w_1} = 0$

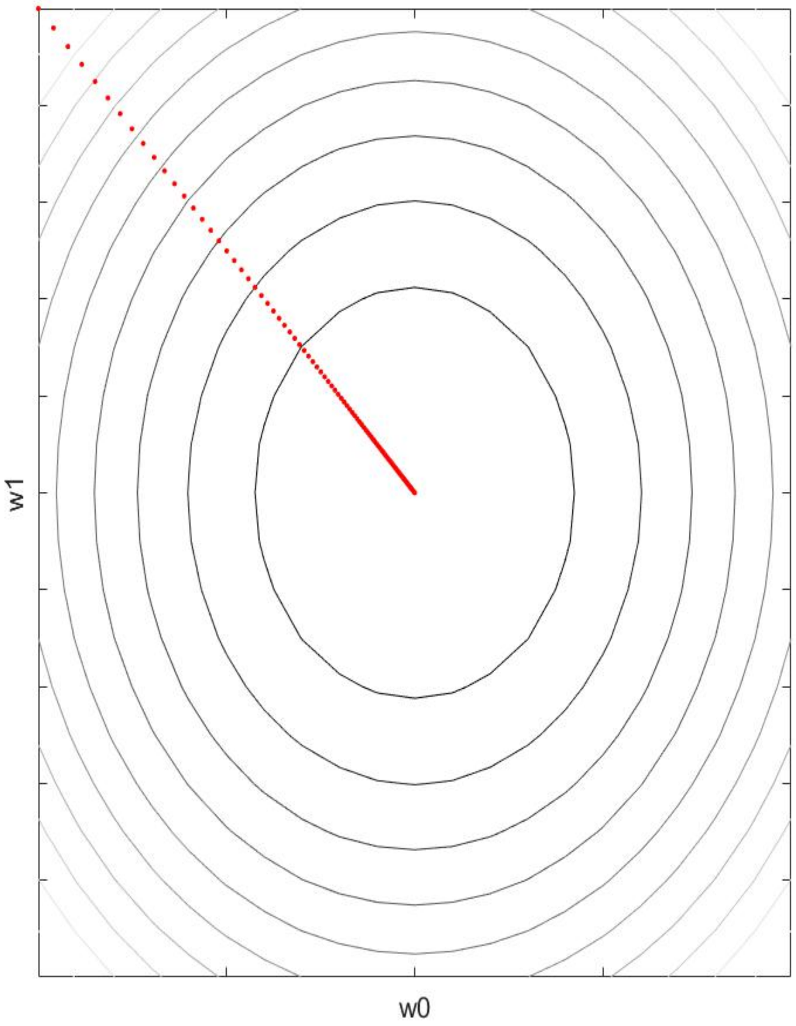
**Sometimes these will be hard (or impossible) to solve. Gradient descent technique can be instead:**

$\mathbf{w} \leftarrow$ **any point in the parameter space**
**while not** converged **do**
**for each** $\mathrm{w_i}$ **in** $\mathbf{w}$ **do** $w_i \leftarrow w_i - \alpha * \frac{\partial Loss(\boldsymbol{w})}{\partial\, w_i}$

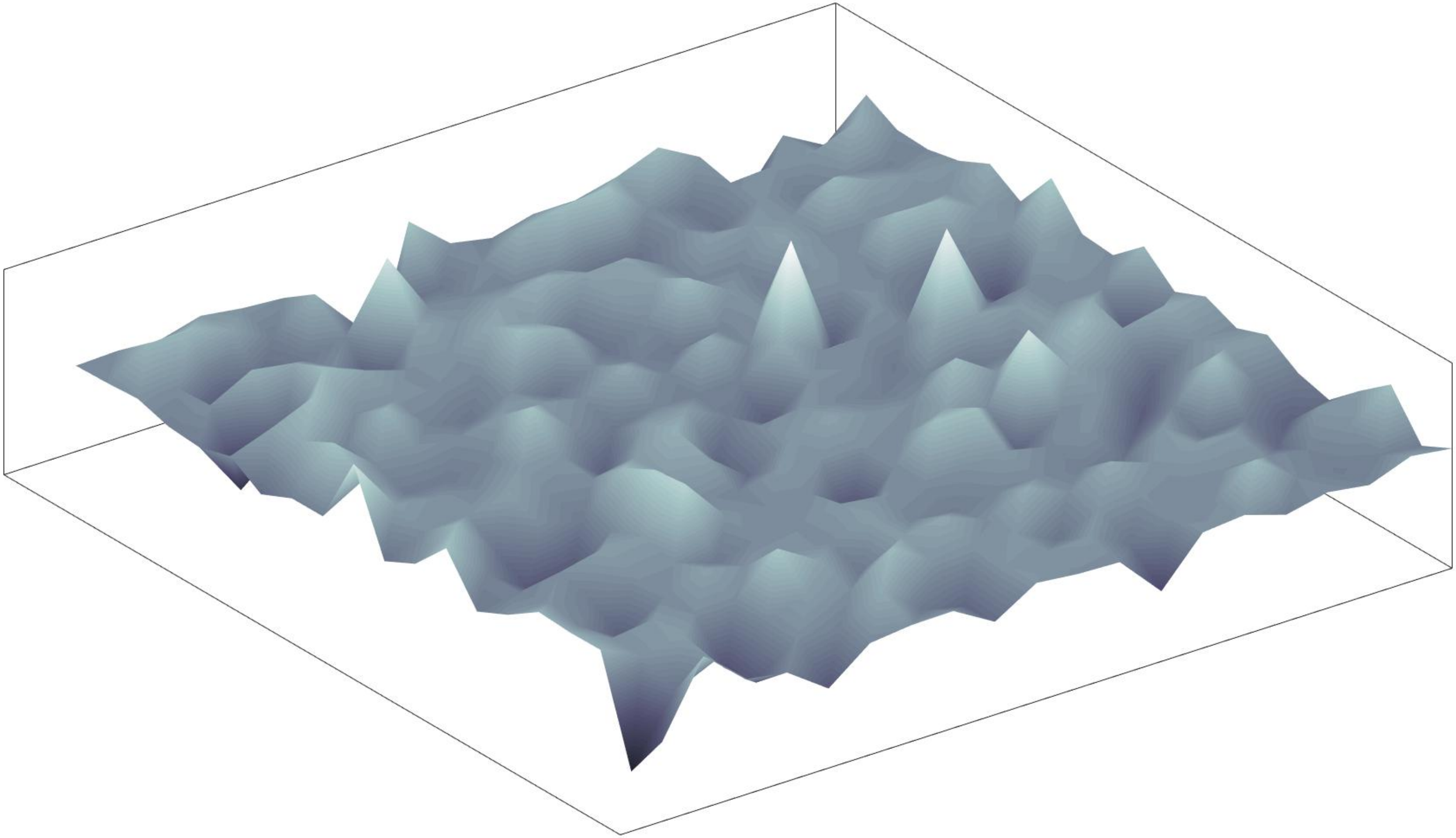
$\alpha$ **- step size / learning rate**

# Gradient Descent

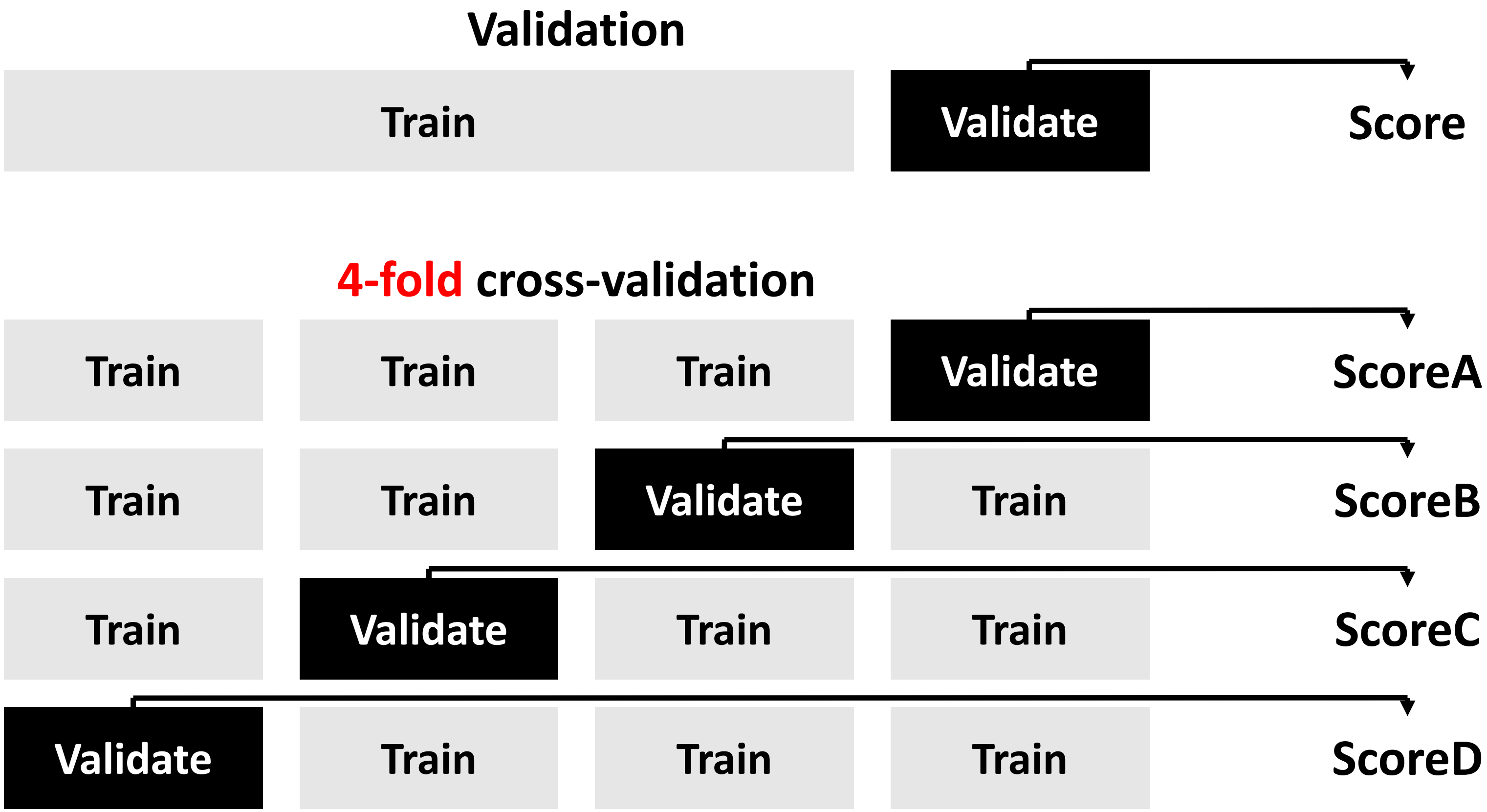


# Challenging Parameter Space



# K-Fold Cross-Validation




$$\text{Score} = \frac{\text{ScoreA} + \text{ScoreB} + \text{ScoreC} + \text{ScoreD}}{4}$$

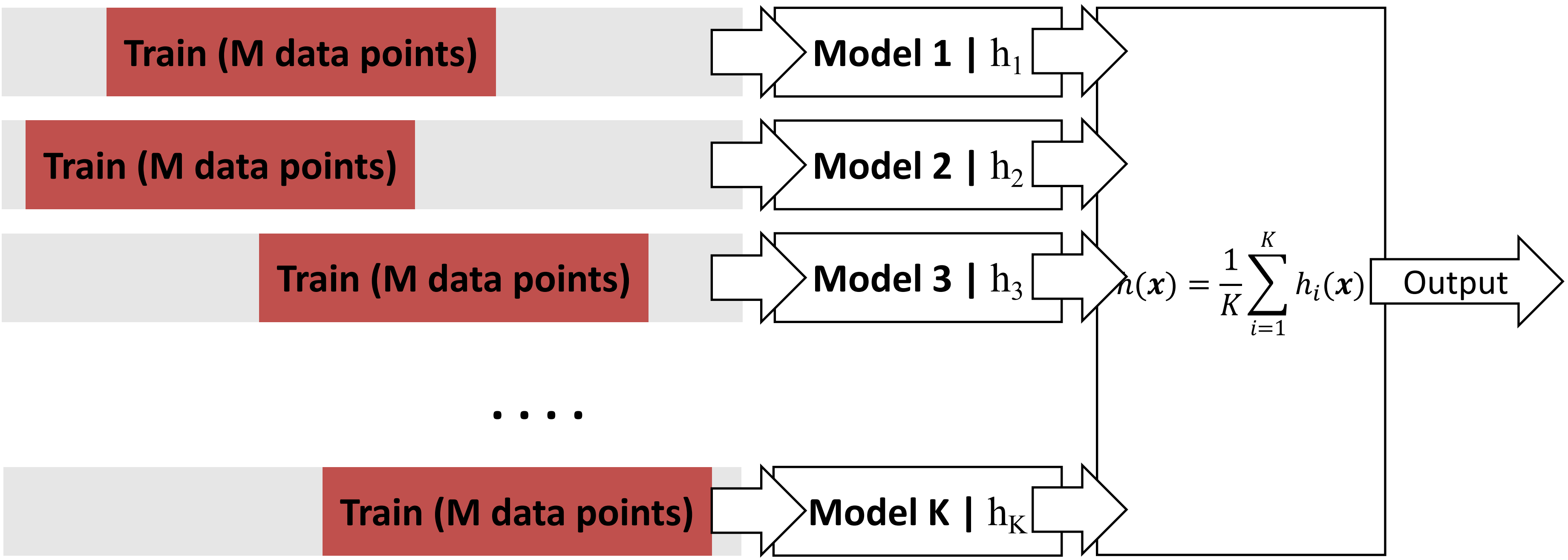
# Ensemble Learning

**In ensemble learning we are creating a collection (an ensemble) of hypotheses (models) $h_1, h_2, \ldots, h_N$ and combine their predictions by averaging, voting, or another level of machine learning. Indvidual hypotheses (models) are based models and their combination is the ensemble model.**

- **Bagging**
- **Boosting**
- **Random Trees**
- **etc.**

# Bagging: Regression

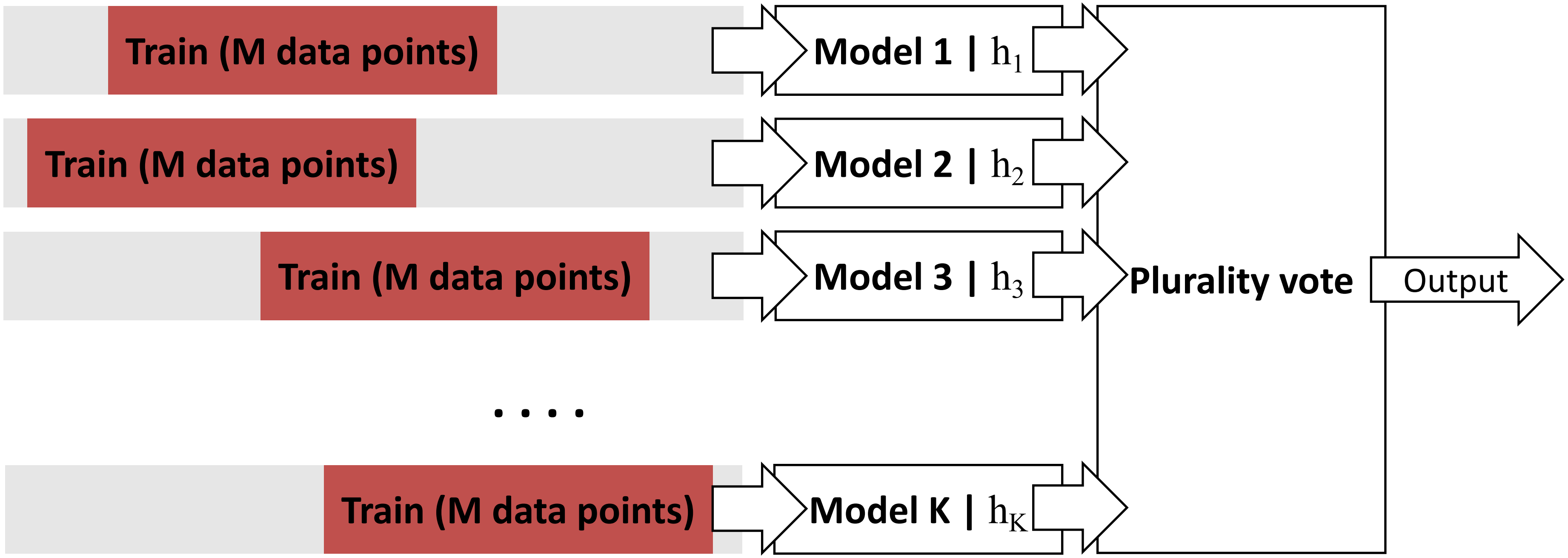
**In bagging we generate $K$ training sets by sampling with replacement from the original training set.**




**Bagging tends to reduce variance and helps with smaller data sets.**

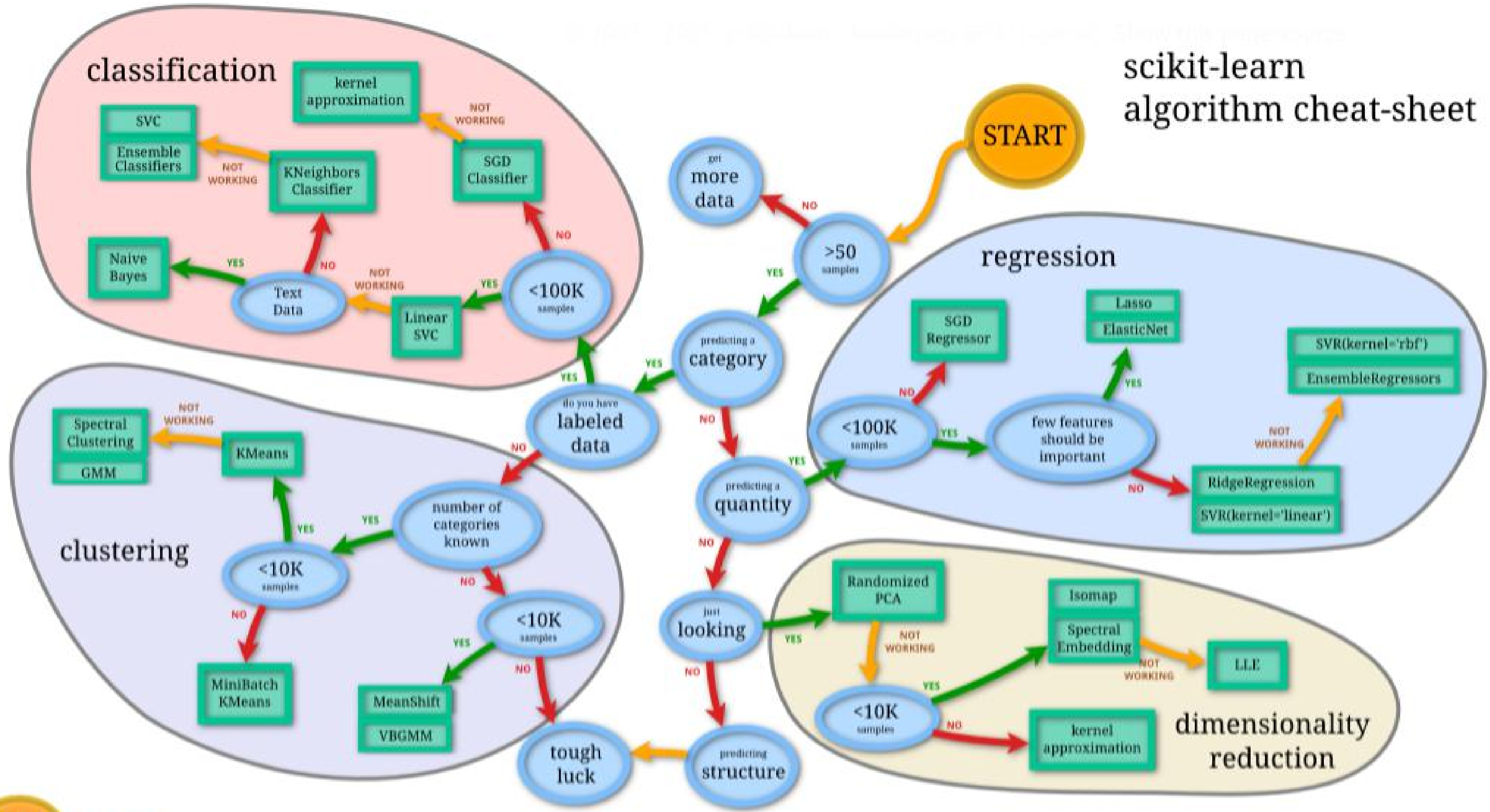
# Bagging: Classification

**In bagging we generate $K$ training sets by sampling with replacement from the original training set.**




**Bagging tends to reduce variance and helps with smaller data sets.**

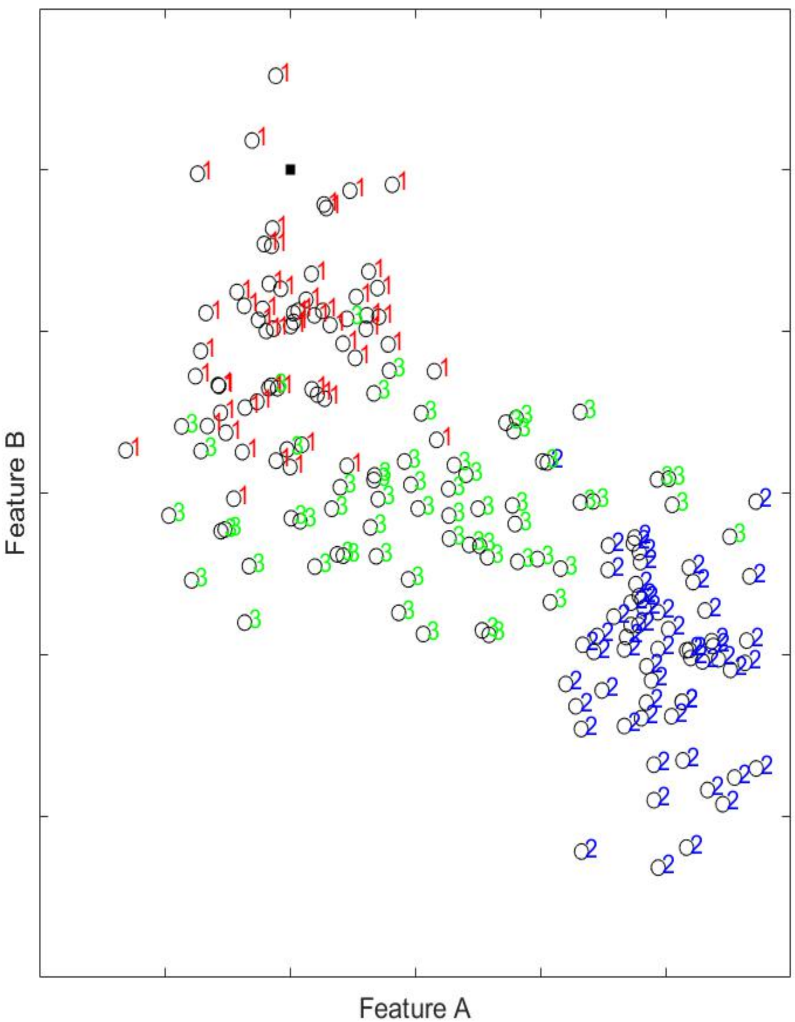
# scikit-learn Algorithm Cheat Sheet

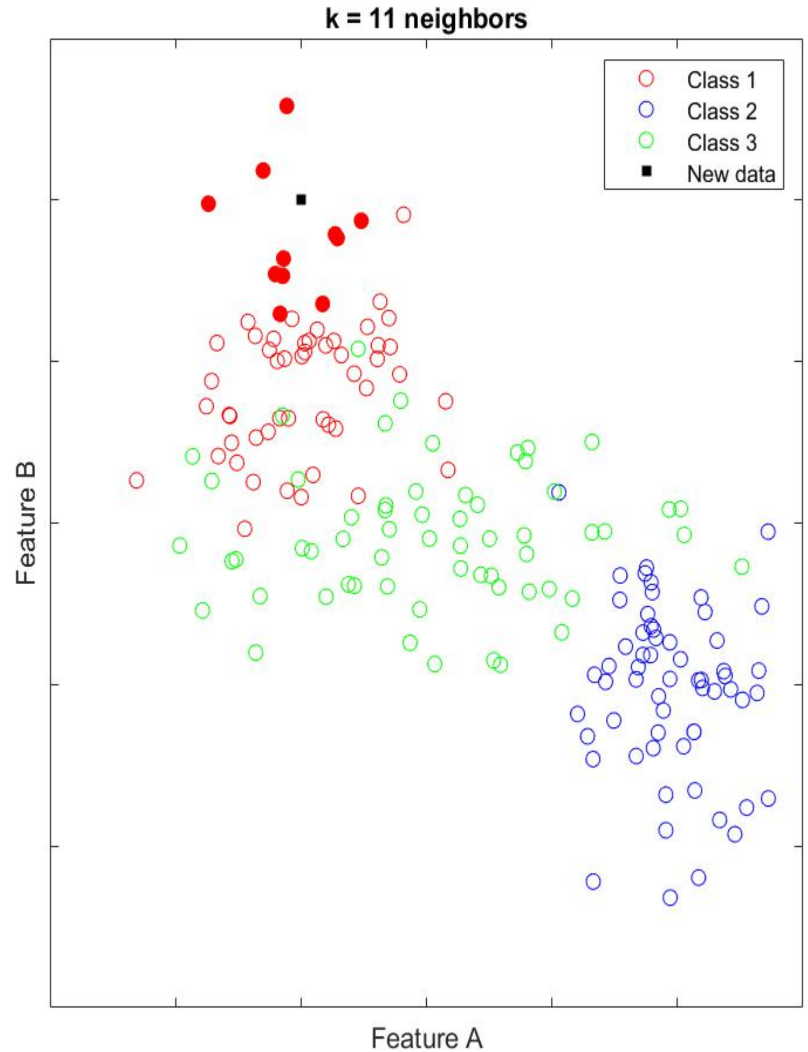




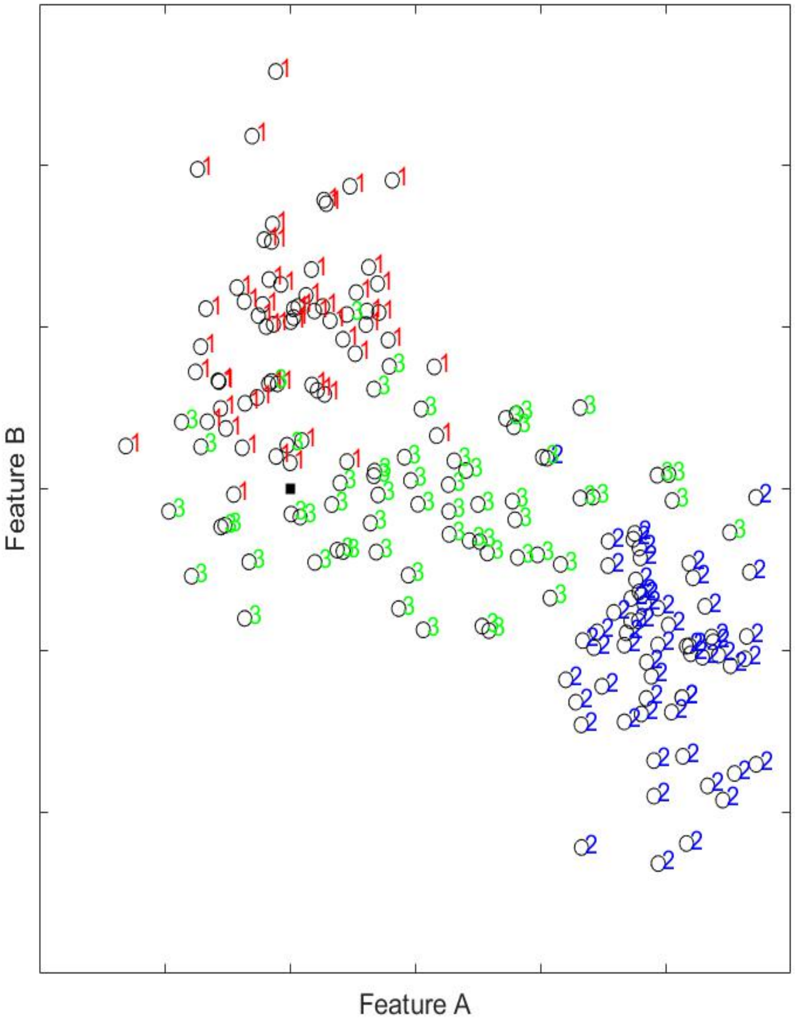
*Source: https://scikit-learn.org/stable/tutorial/machine_learning_map/index.html*

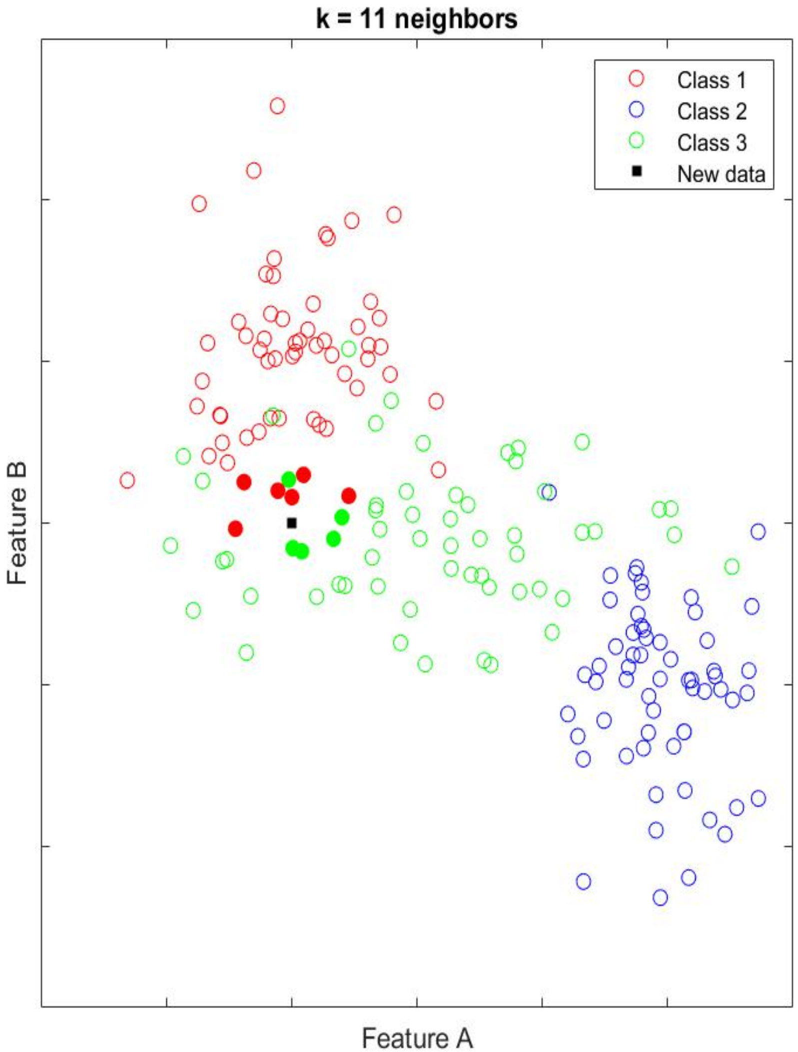
# k = 11 Nearest Neighbors



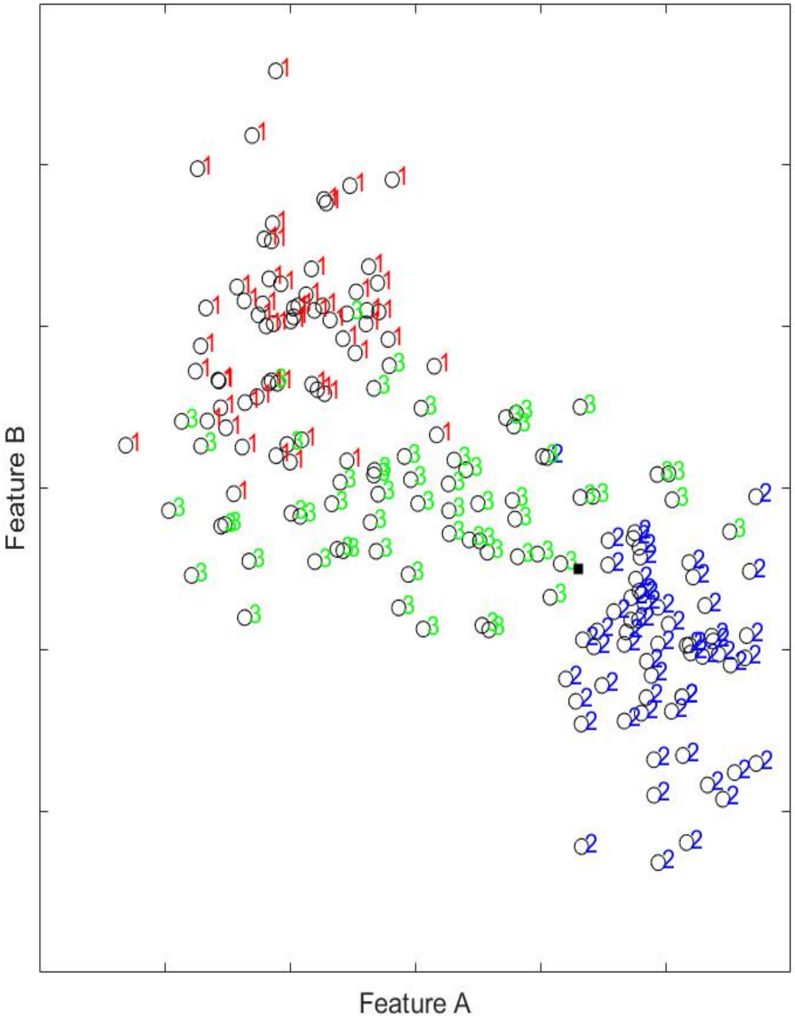



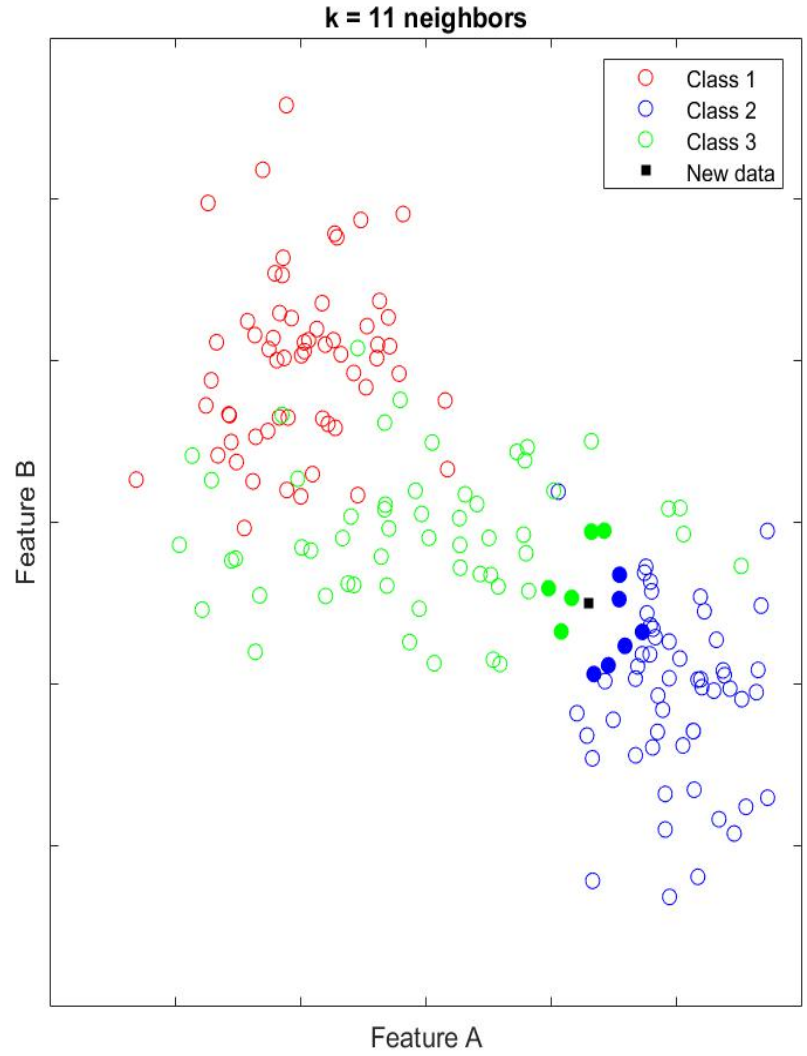
# k = 11 Nearest Neighbors



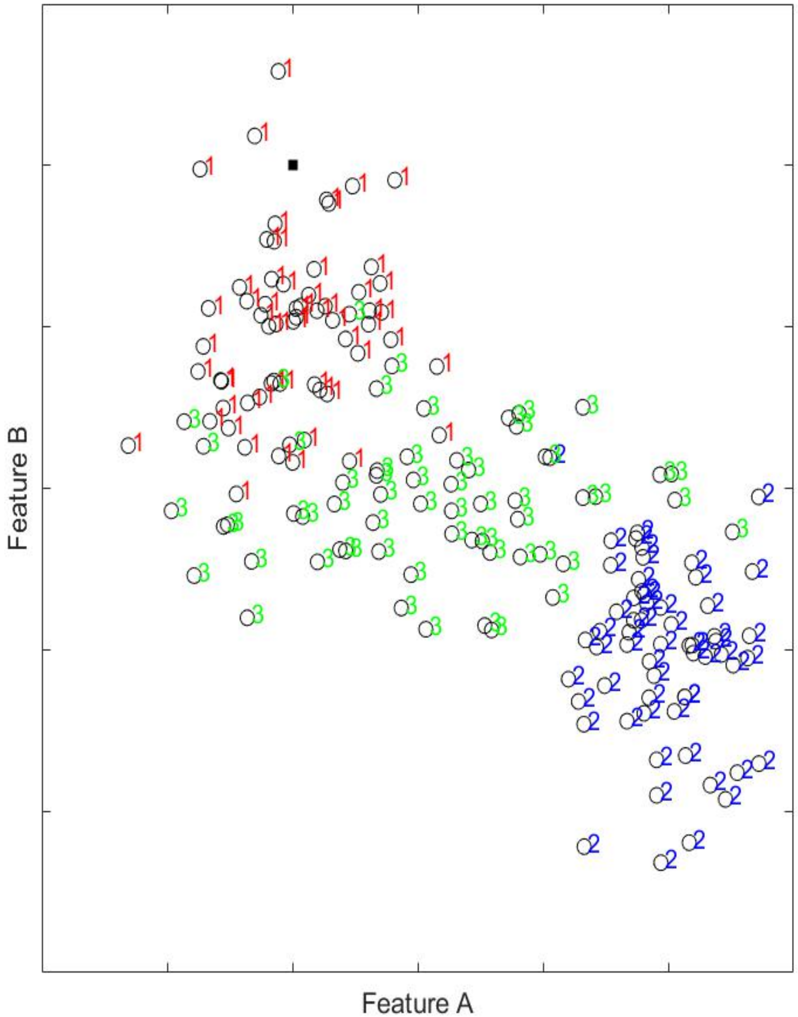



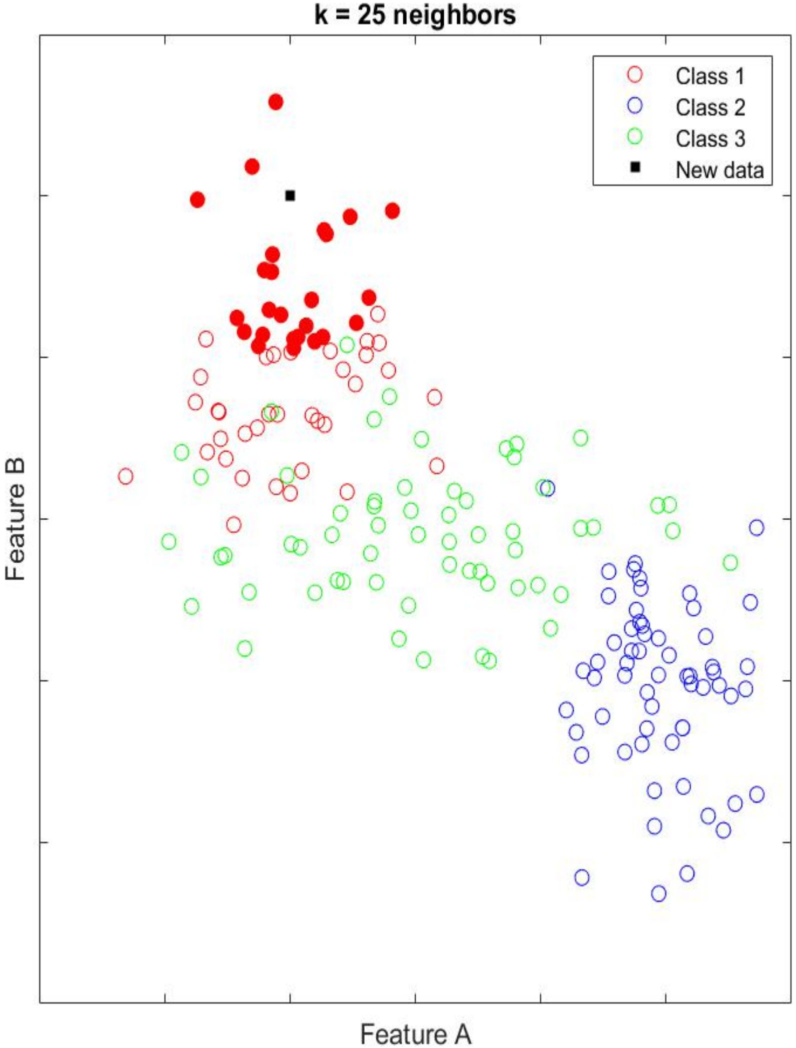
# k = 11 Nearest Neighbors



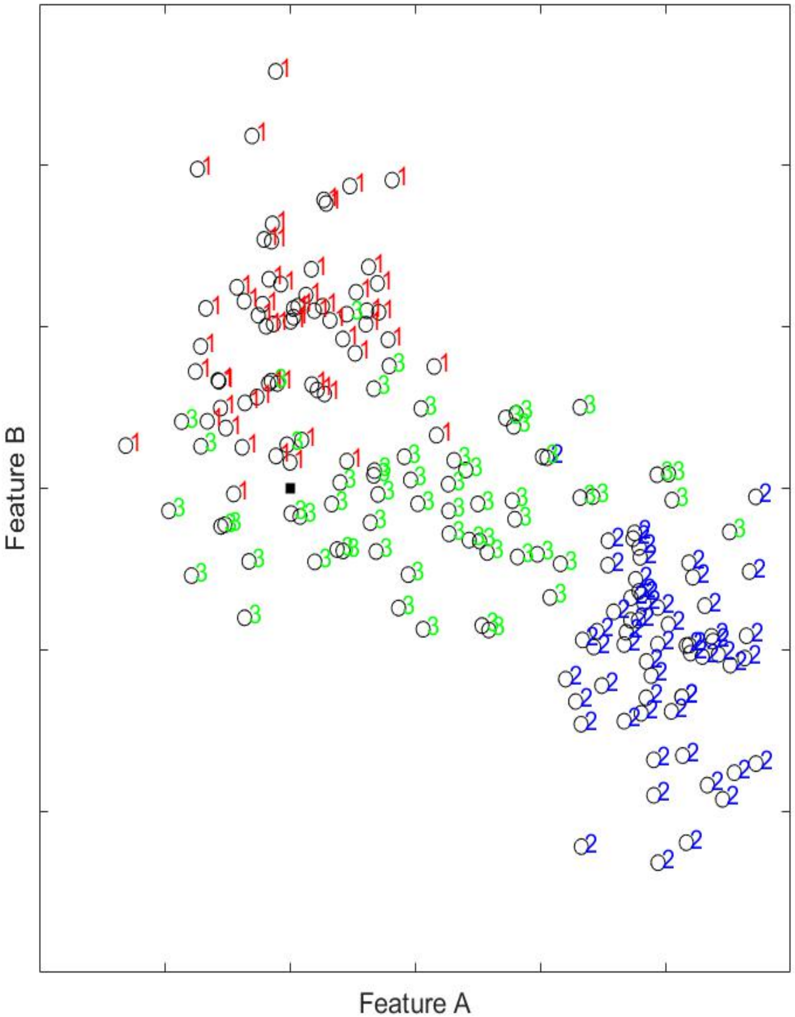



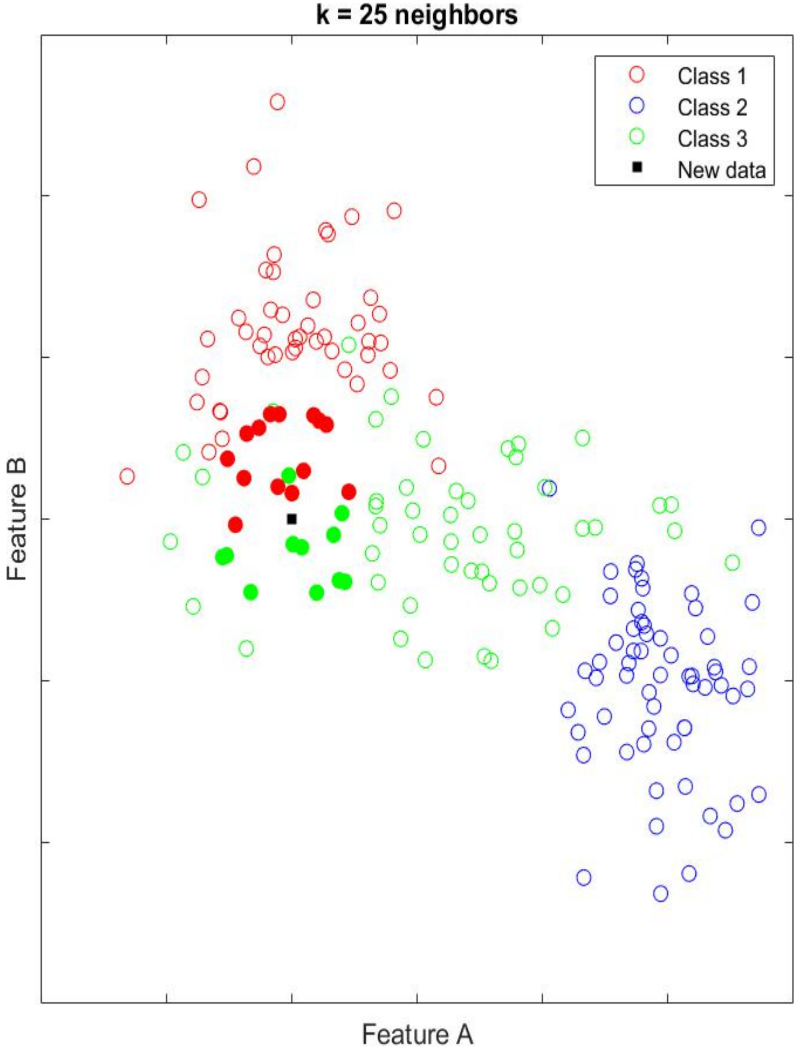
# k = 25 Nearest Neighbors



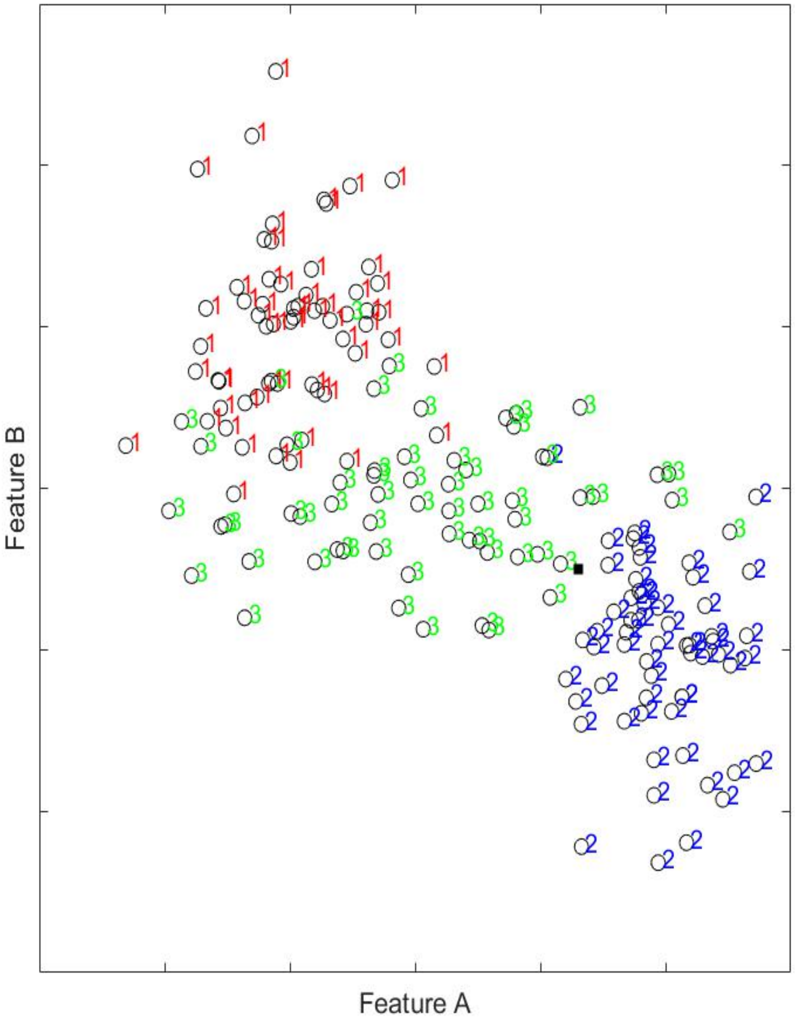



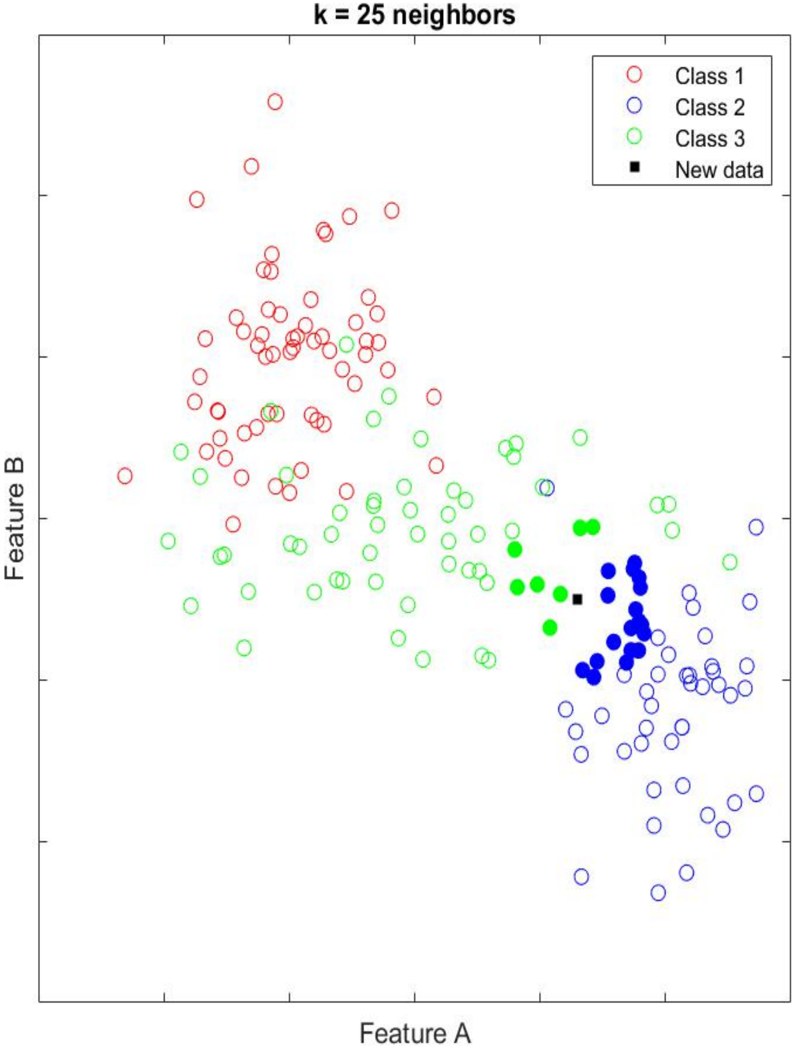
# k = 25 Nearest Neighbors



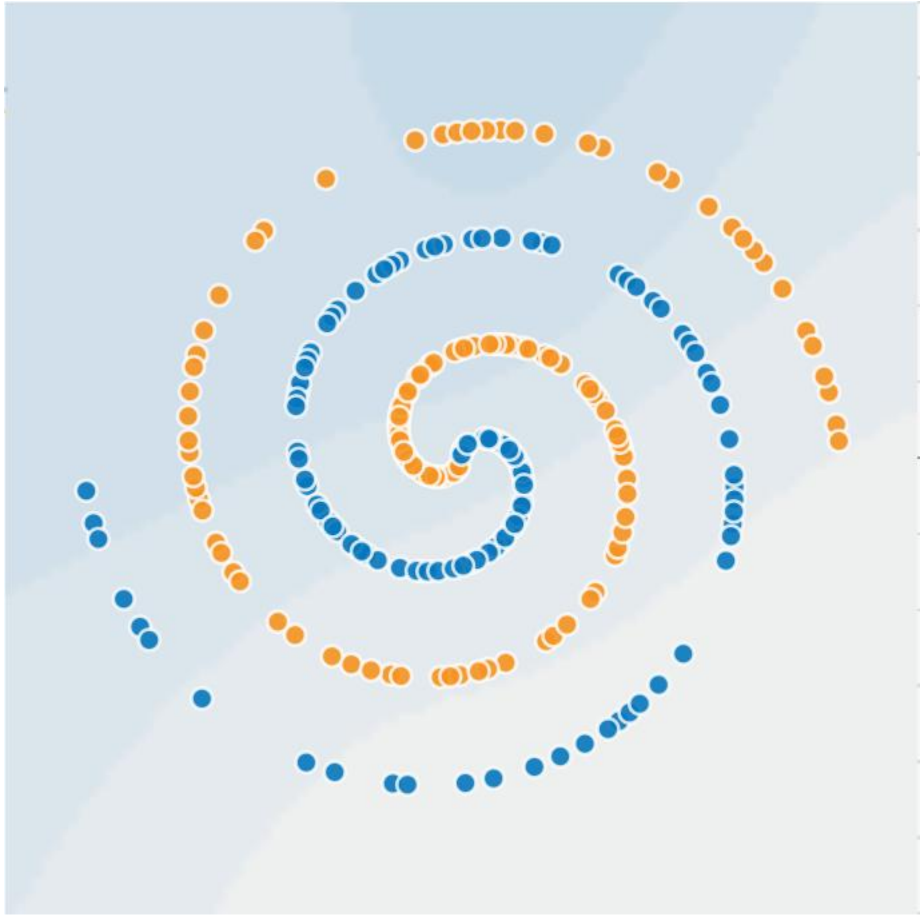



# k = 25 Nearest Neighbors

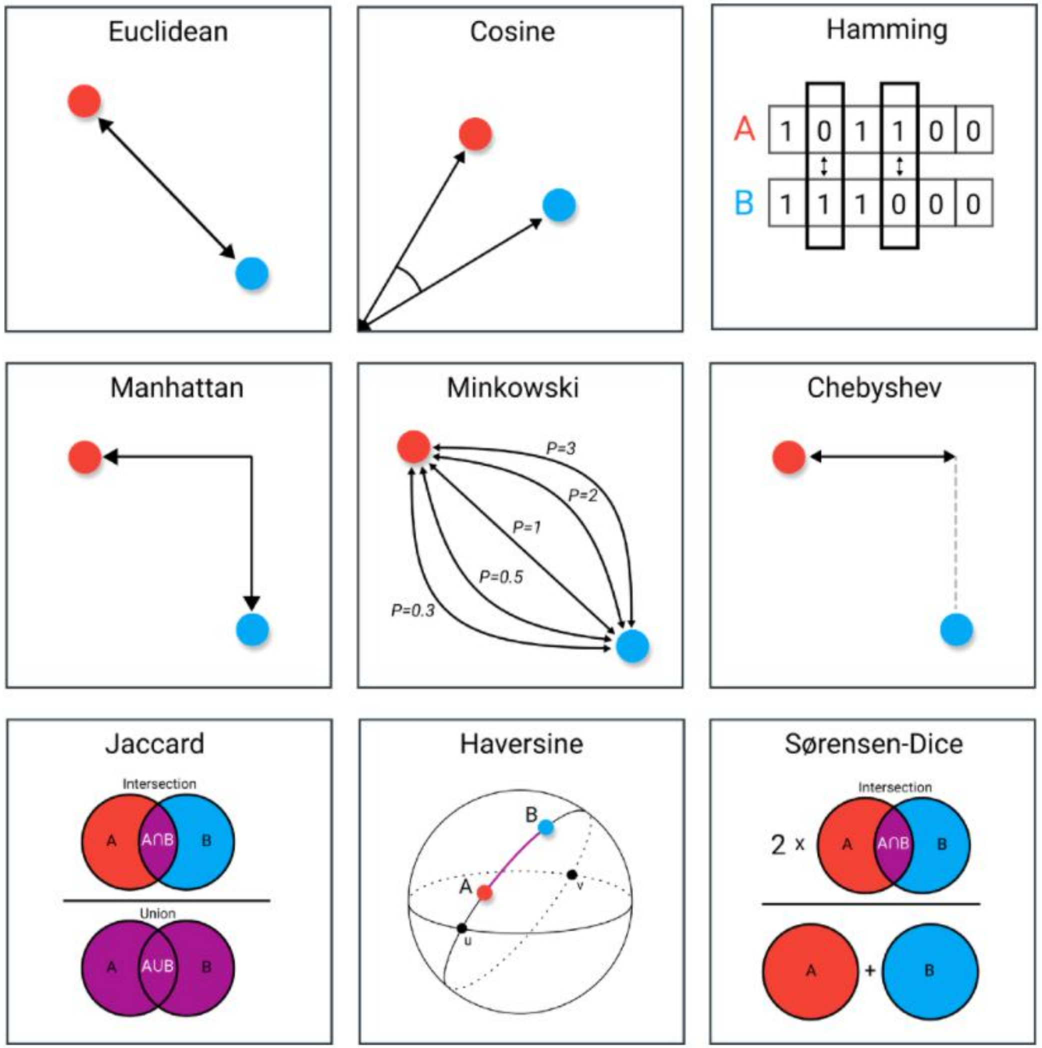


# How Would kNN Do Here?



# Distance Measures




*Source: https://towardsdatascience.com/9-distance-measures-in-data-science-918109d069fa*

# Practical ML: Feature Engineering

- **One-hot encoding**

  **red = [1, 0, 0]**

  **yellow = [0, 1, 0]**

  **green = [0, 0, 1]**

- **Binning / Bucketing**
- **Normalization**
- **Dealing with missing data / features**

# Reinforcement Learning (RL)

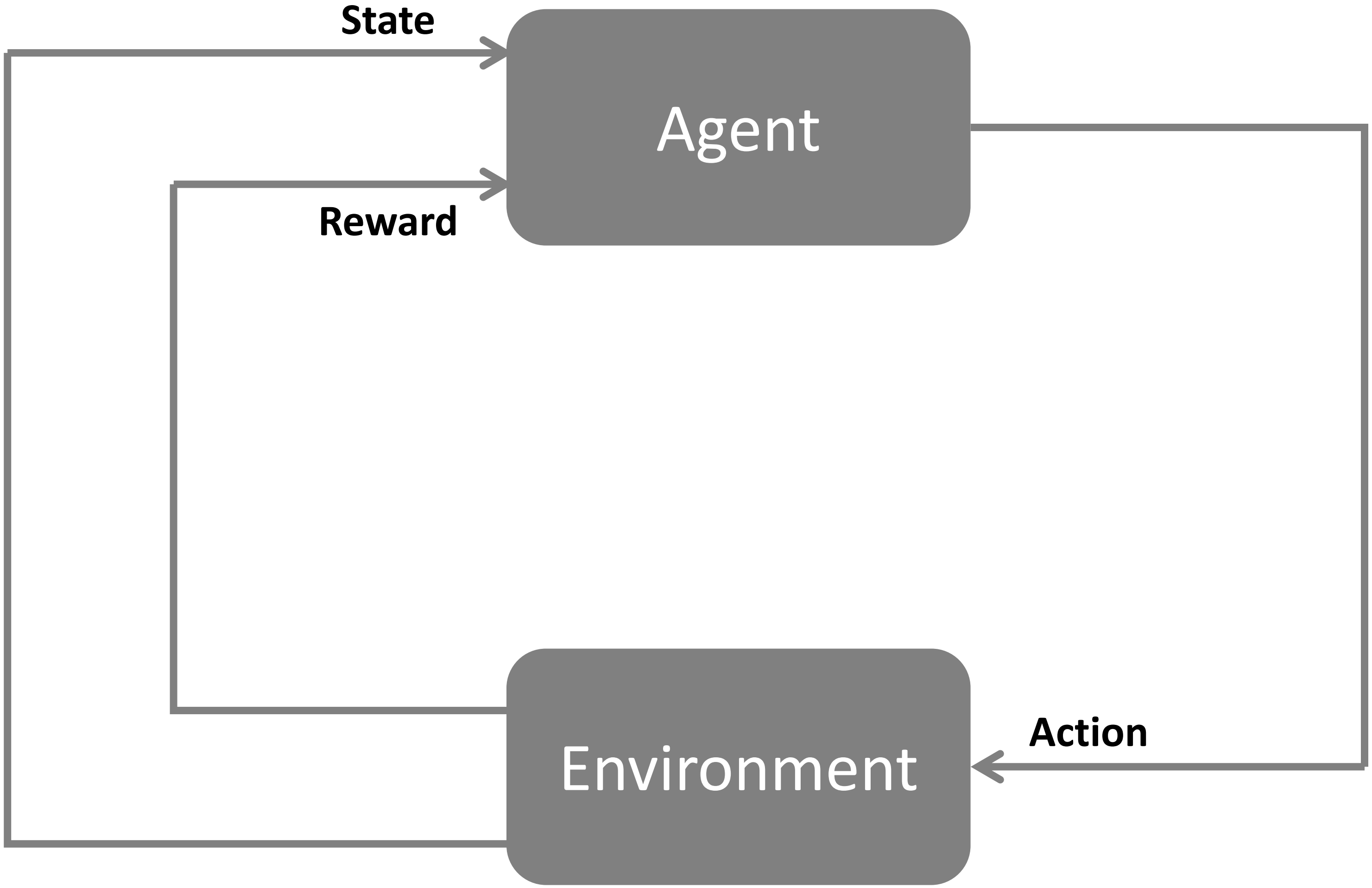
# What is Reinforcement Learning?

**Idea:**

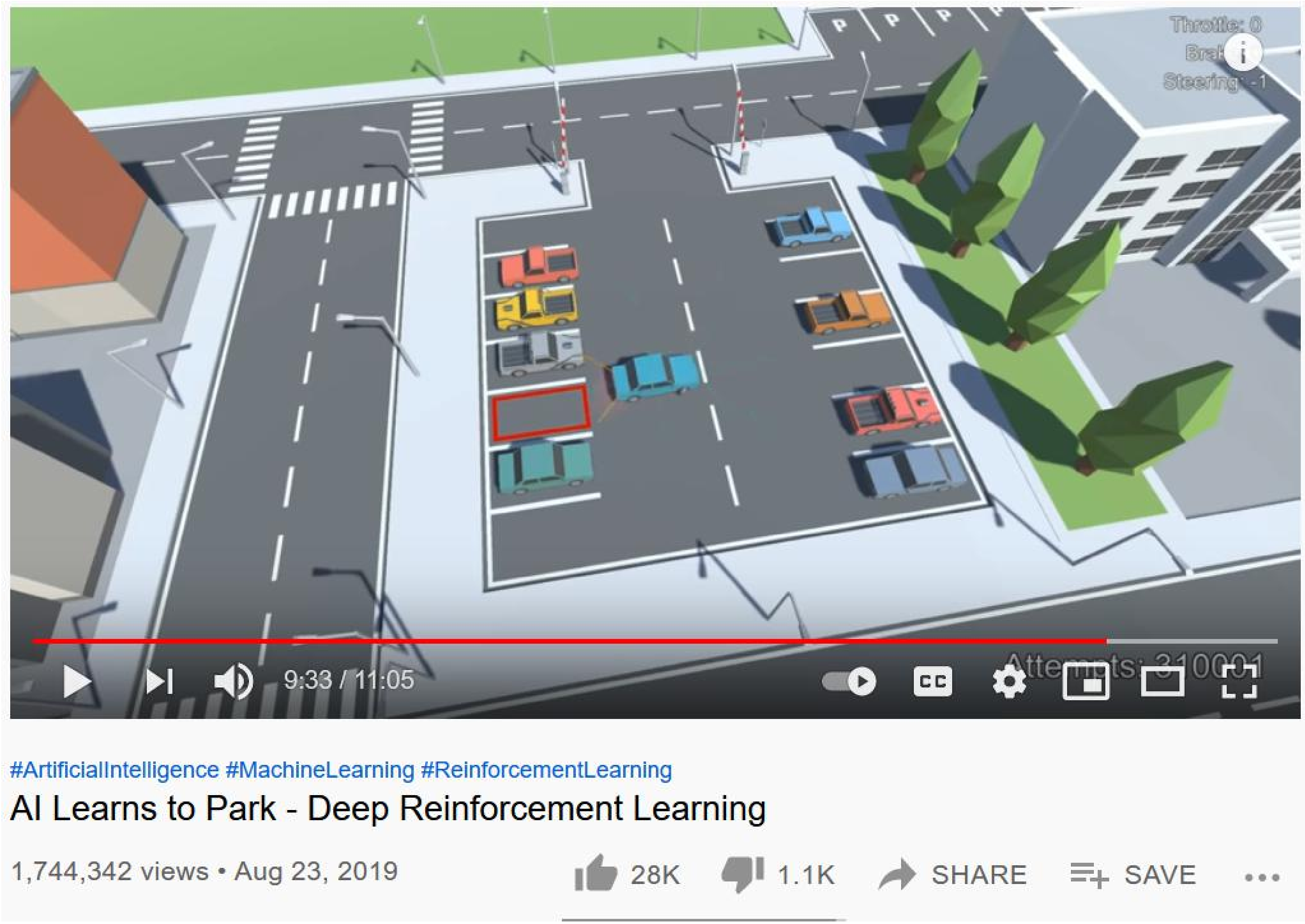
Reinforcement learning is inspired by behavioral psychology. It is **based on a rewarding / punishing an algorithm**.

Rewards and punishments are based on algorithm’s action within its environment.

# RL: Agents and Environments

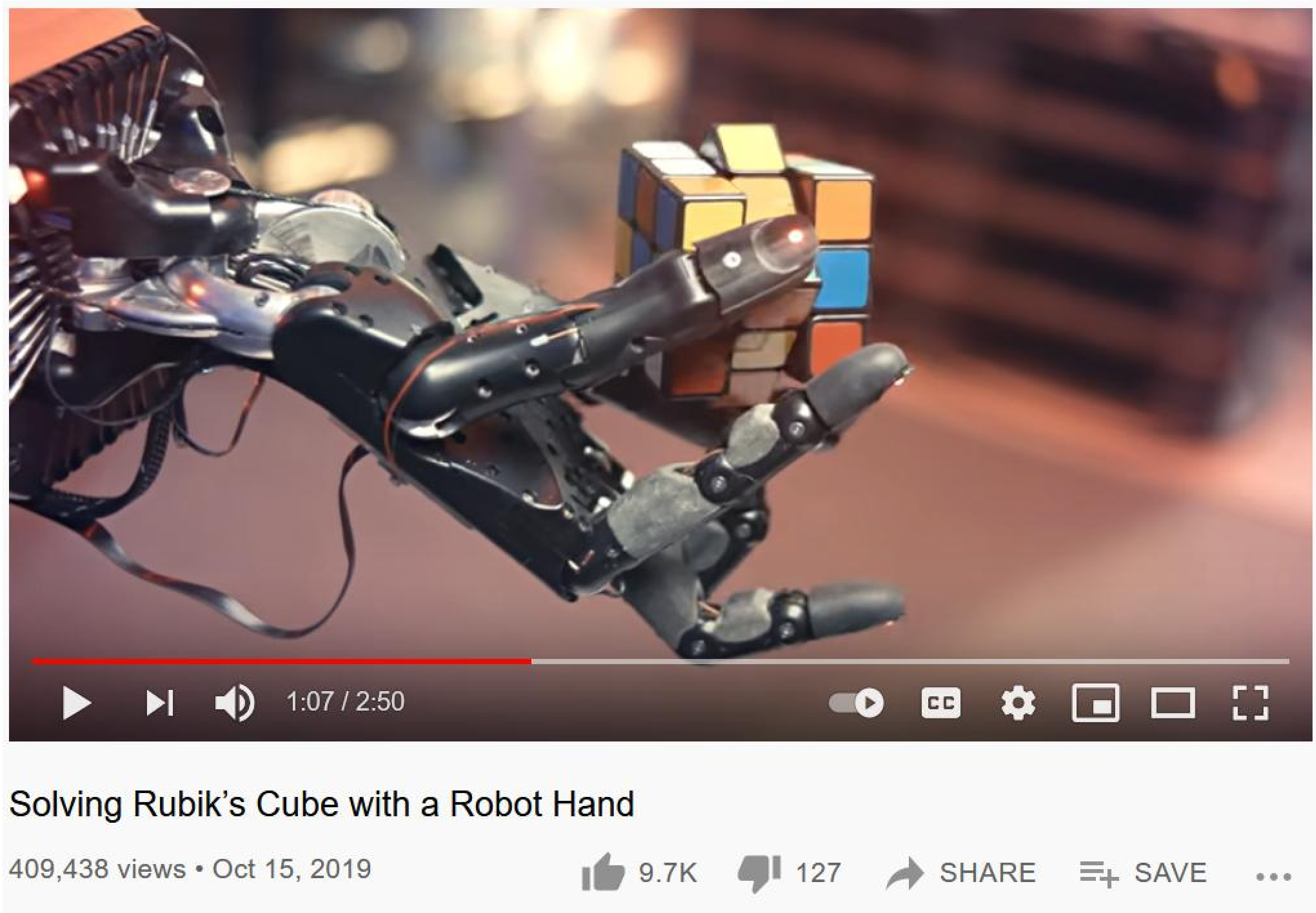


# Reinforcement Learning in Action




*Source: https://www.youtube.com/watch?v=VMp6pq6_QjI*

# Reinforcement Learning in Action

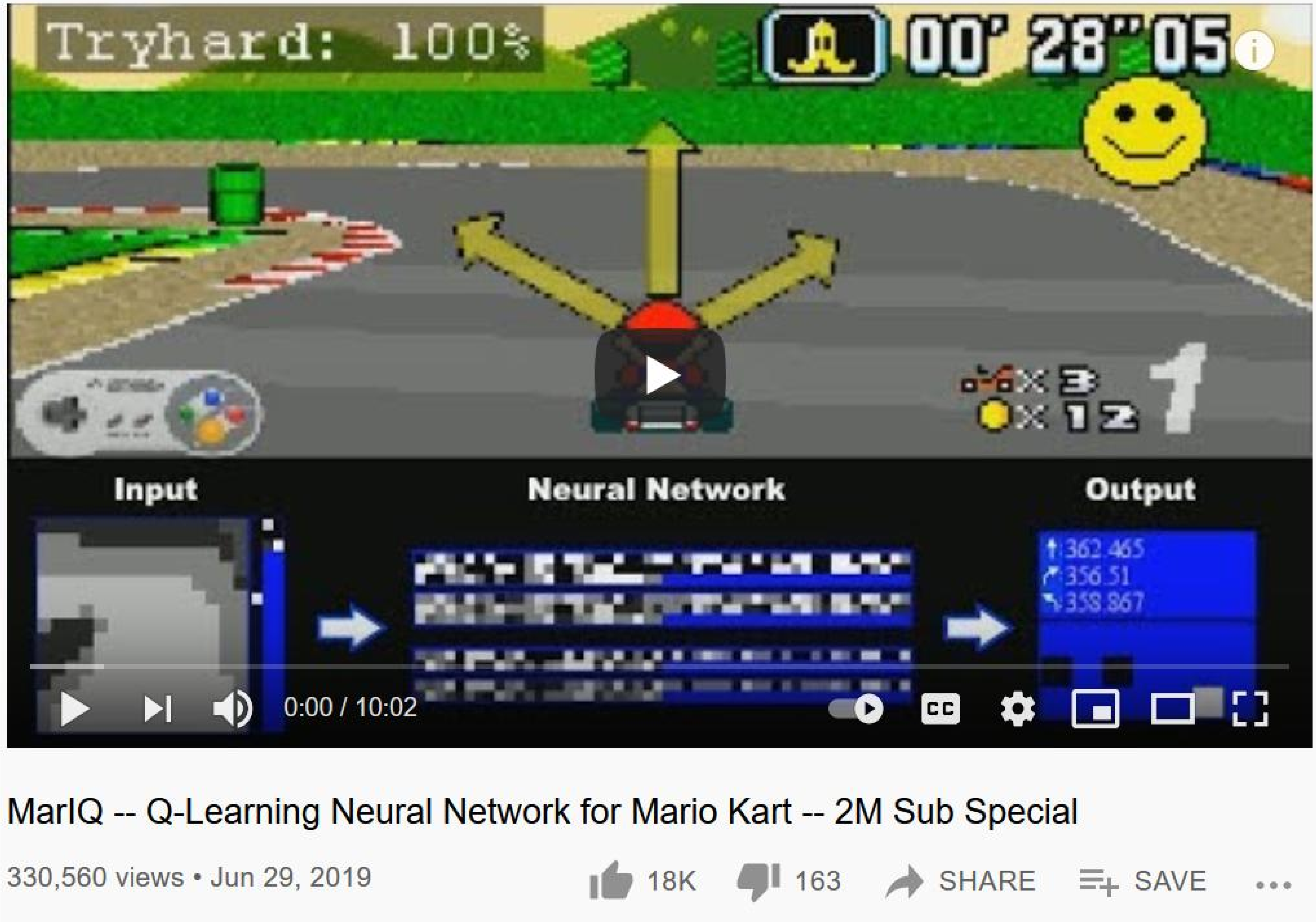



*Source: https://www.youtube.com/watch?v=x4O8pojMF0w*

# Reinforcement Learning in Action

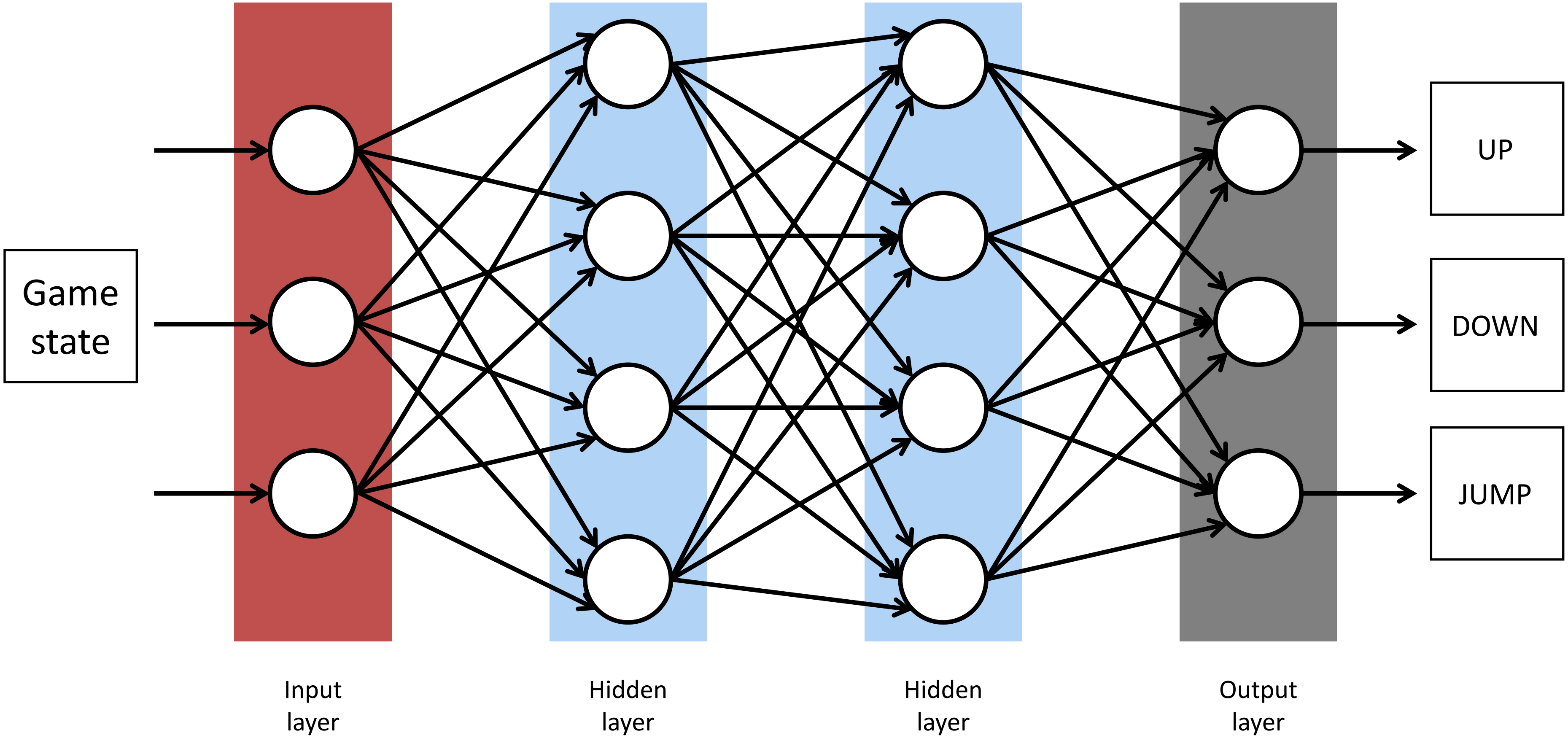

*Source: https://www.youtube.com/watch?v=kopoLzvh5jY*

# Reinforcement Learning in Action



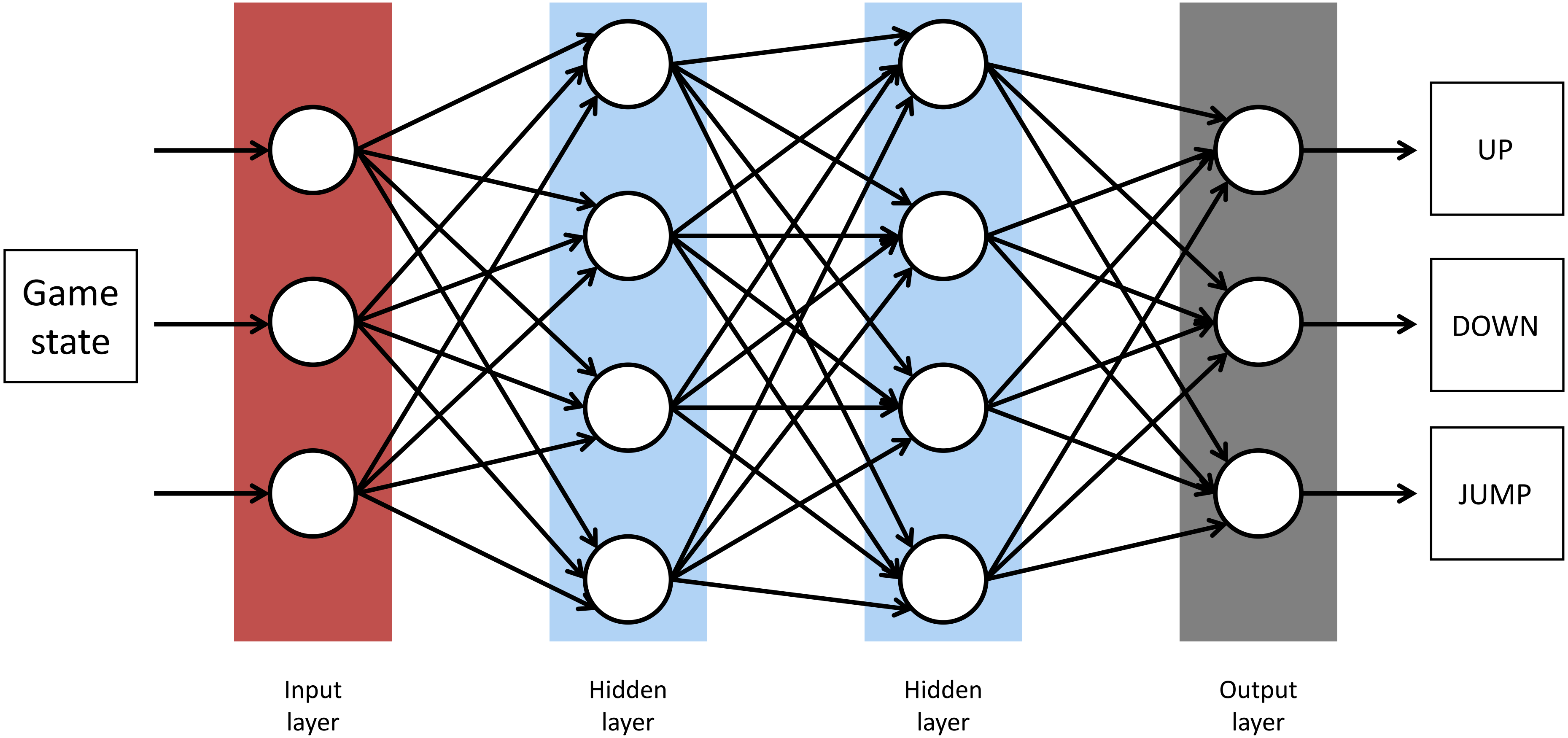

*Source: https://www.youtube.com/watch?v=Tnu4O_xEmVk*

# ANN for Simple Game Playing



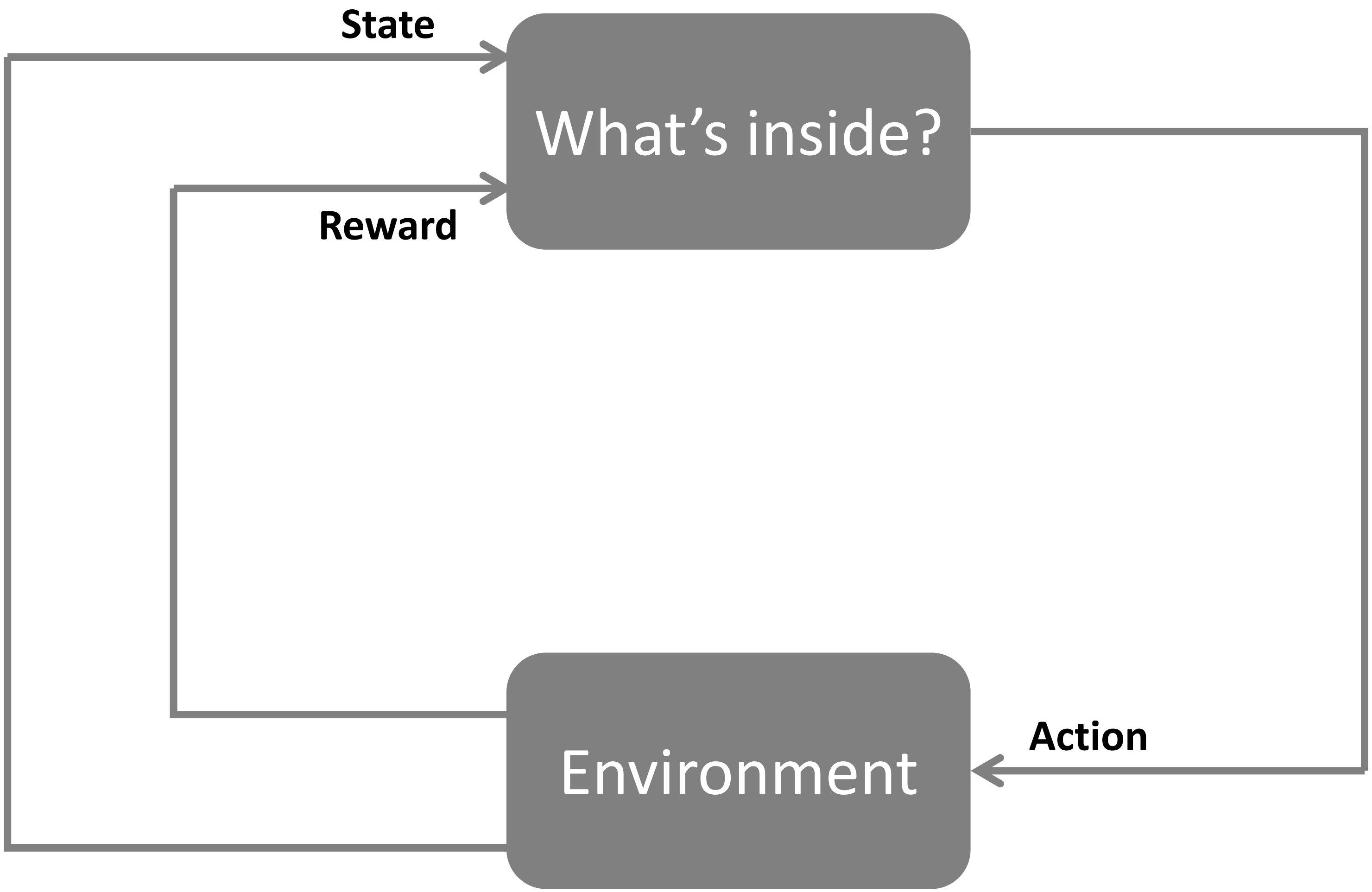
# ANN for Simple Game Playing

Current game is an input. Decisions (UP/DOWN/JUMP) are rewarded/punished.

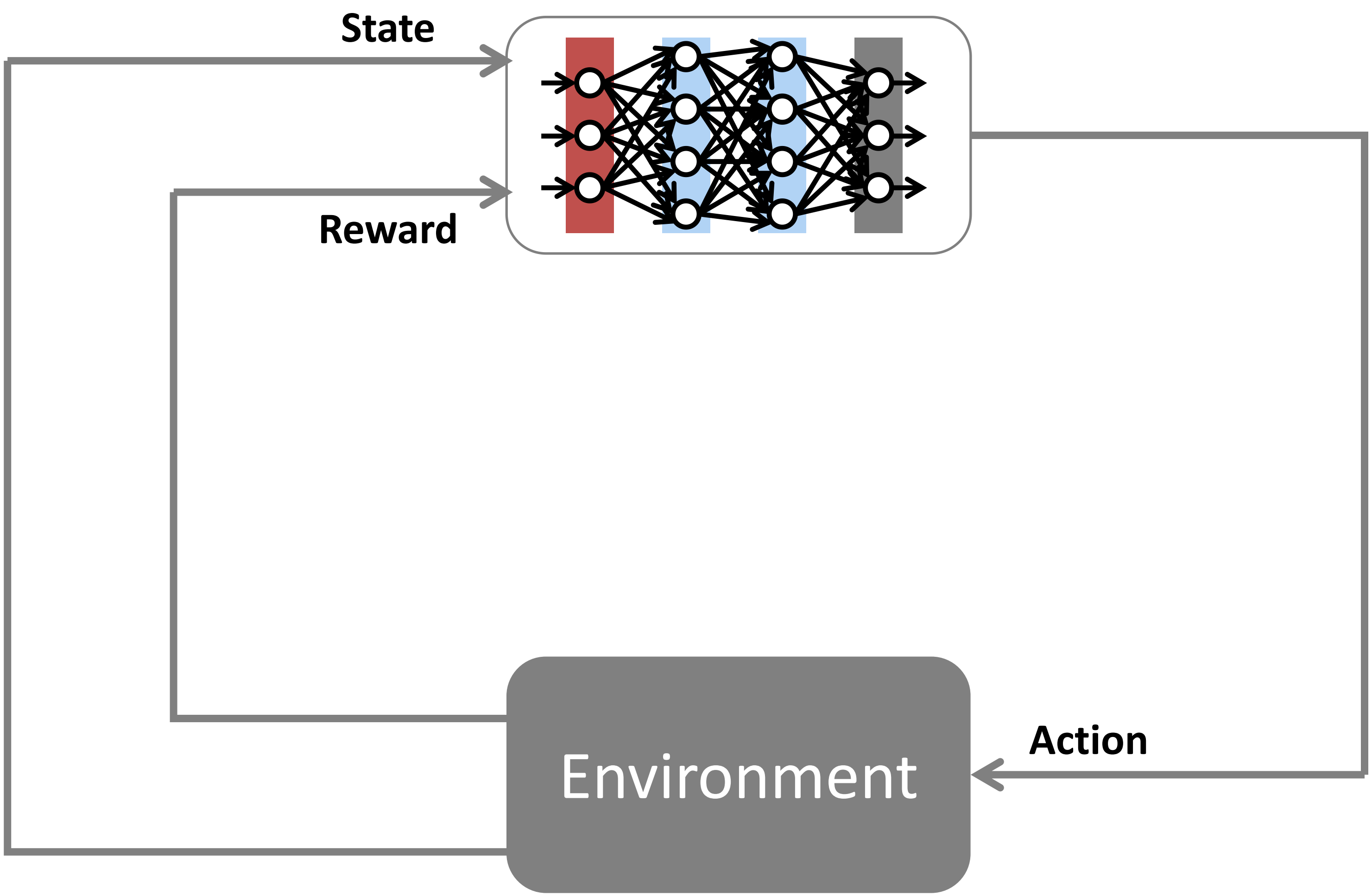



**Correct all the weights** using Reinforcement Learning.

# RL: Agents and Environments

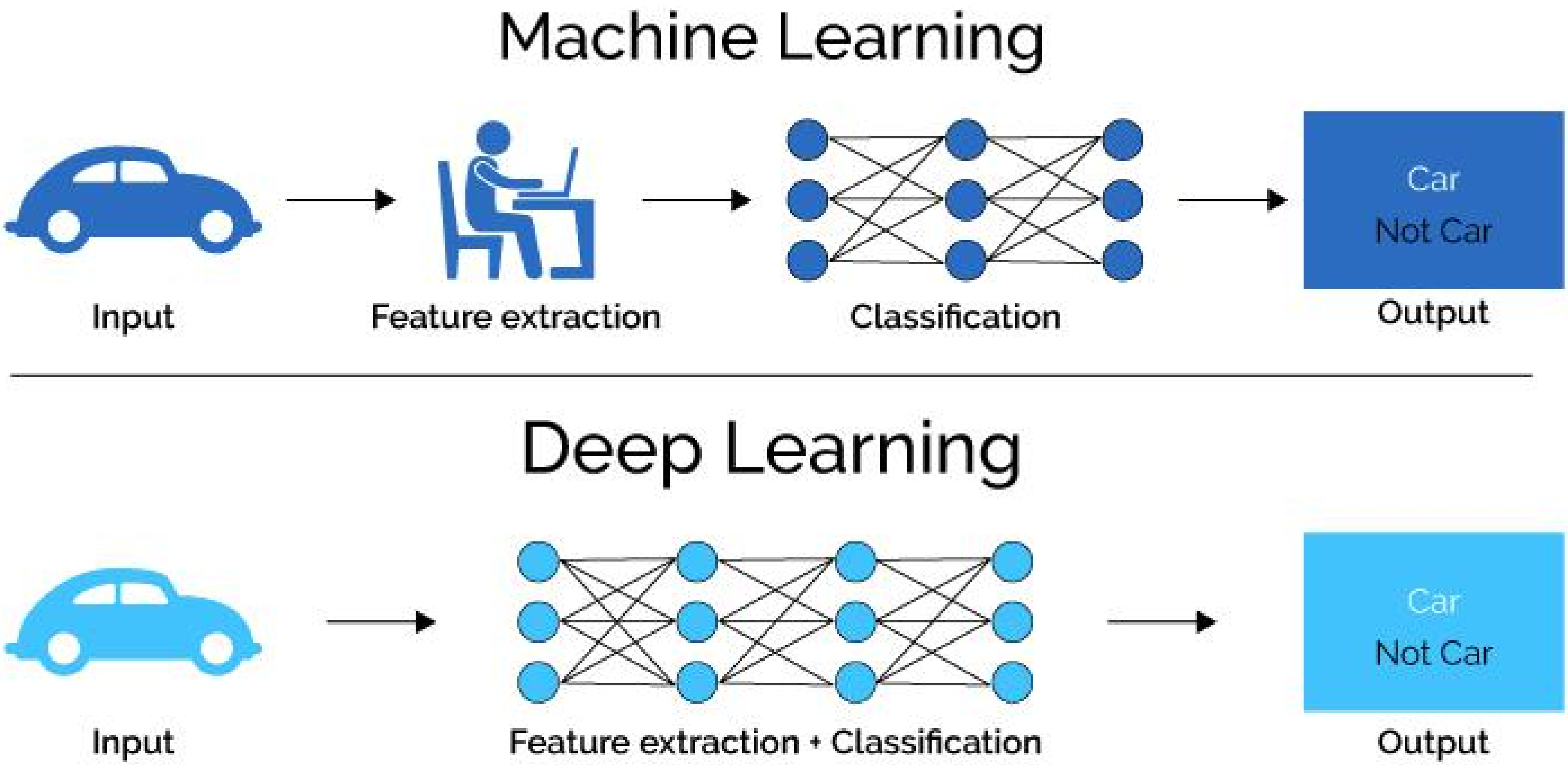


# RL: Agents and Environments



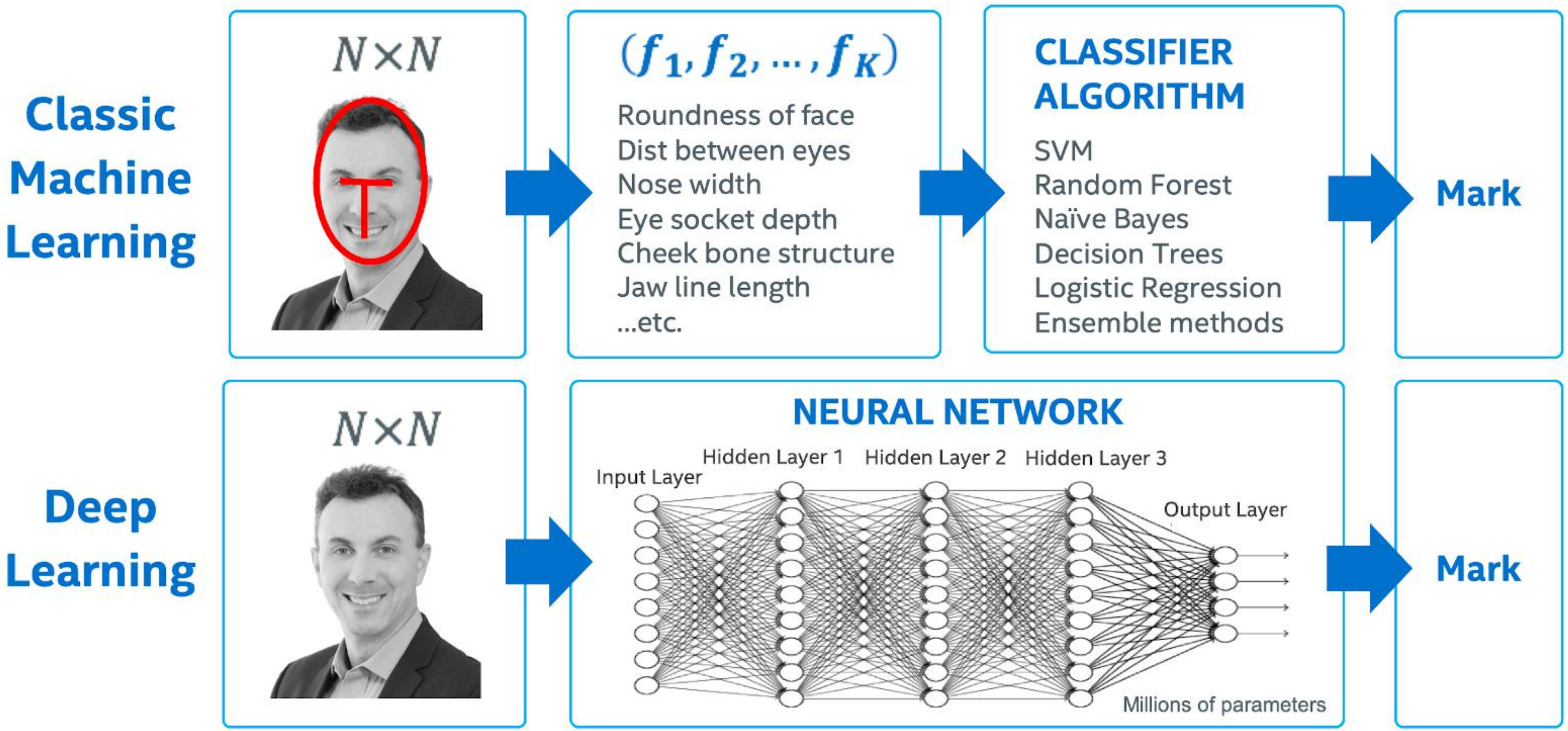
# Deep Learning

# Machine Learning vs. Deep Learning



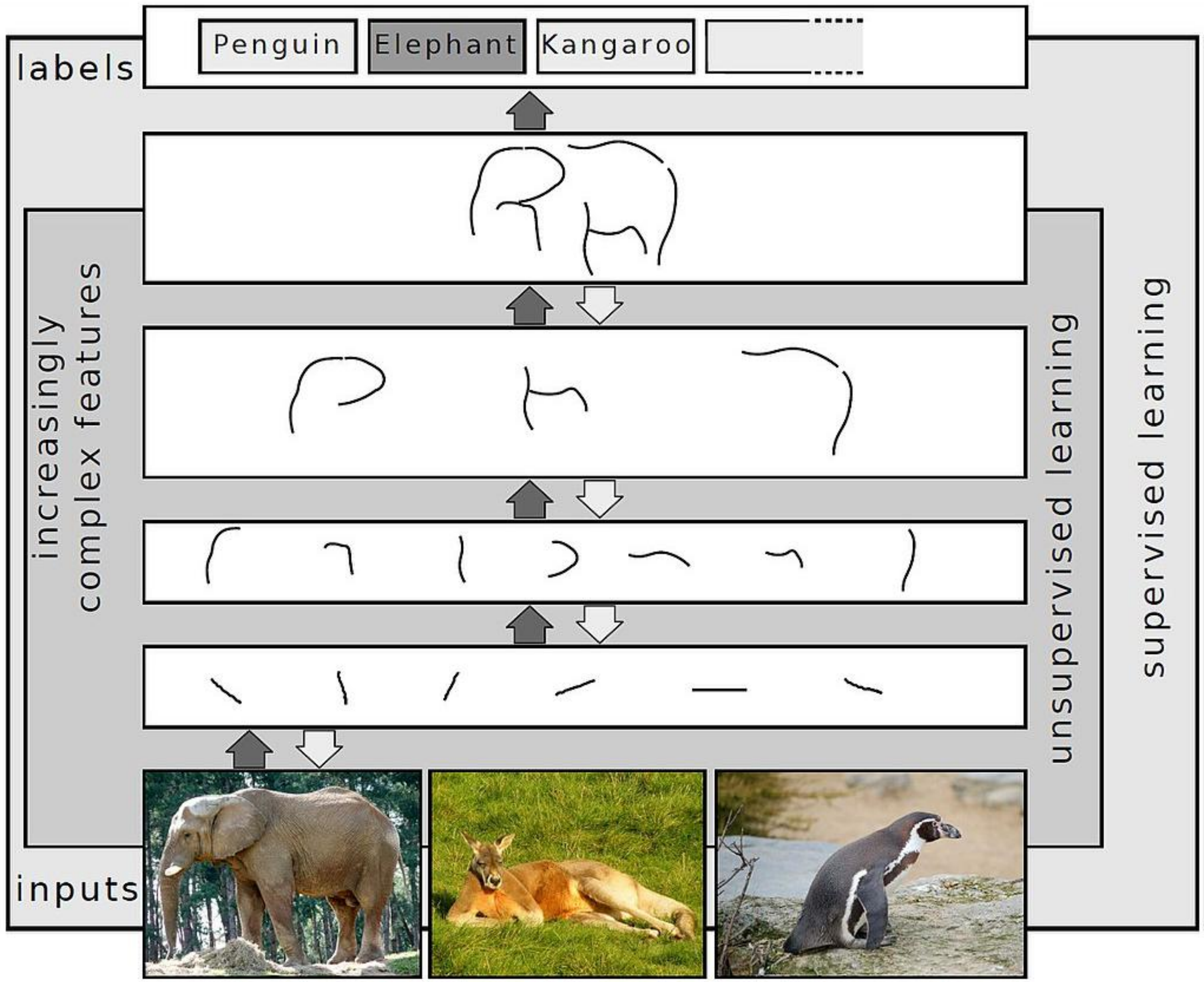

*Source: https://www.quora.com/What-is-the-difference-between-deep-learning-and-usual-machine-learning*

# Machine Learning vs. Deep Learning




*Source: https://www.intel.com/content/www/us/en/artificial-intelligence/posts/difference-between-ai-machine-learning-deep-learning.html*

# Deep Learning: Feature Extraction




*Source: https://en.wikipedia.org/wiki/Deep_learning*

# Exercise: Object Recognition

**https://braneshop.com.au/object-detection-in-the-browser.html**

**(you can try it on your smartphone)**

# Exercise: Image Colorizer

**https://deepai.org/machine-learning-model/colorizer**

# Exercise: Deep Learning

**https://www.handwriting-generator.com/**