

Intro to Machine Learning

MasseyHacks 2017

Deep Blue

- AI computer program built to play Chess
- Programmed for one purpose only



Deep Blue at the computer history museum

AlphaGo

- Machine learning algorithm that plays the board game “Go”
- Excelled in the Go world championship in 2016
- Same algorithm could be taught to play Atari games



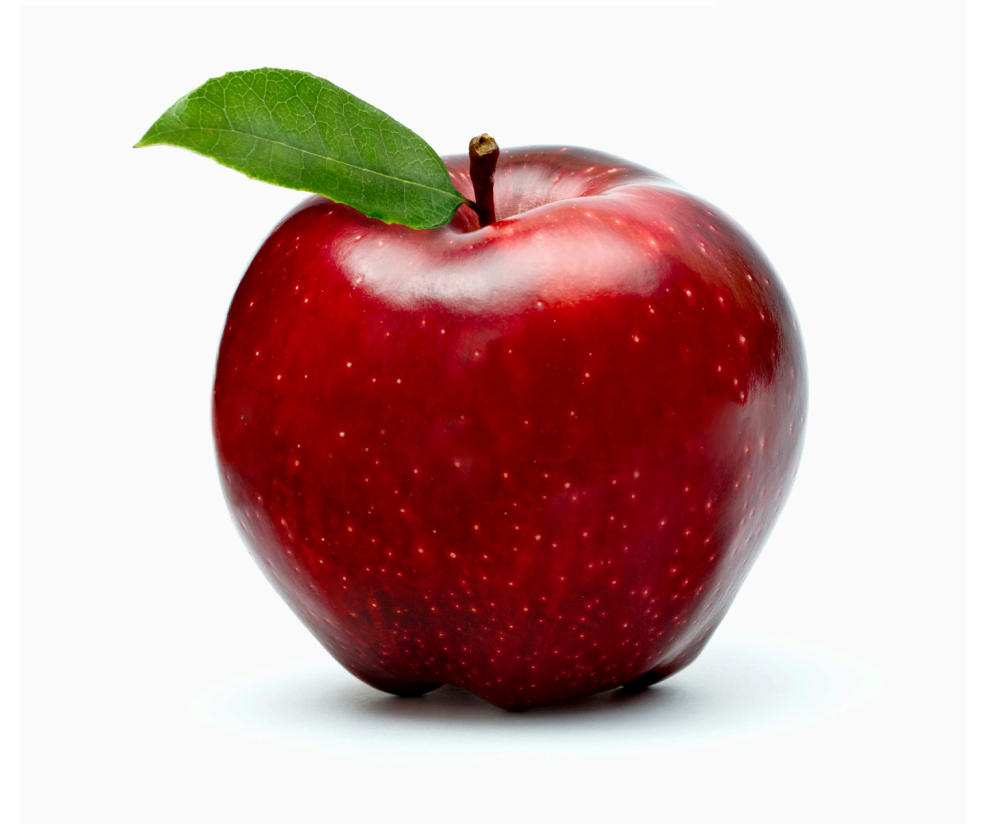
AlphaGo playing Go against Lee Sedol - a world champion

Machine Learning:

Machine learning is the subfield of computer science that gives "computers the ability to learn without being explicitly programmed."

The Problem

- How would you write a program to tell the difference between an apple and an orange?



Lots of code

- Too specialized
- Lots of shaky, hard to solve problems to solve
- Not re-usable for other problems

```
def findColors(image):  
    # lots of manual checks
```

```
def findEdges(image):  
    # lots of manual checks
```

```
def findShapes(image):  
    # lots of manual checks
```

```
def determineFruit(image):  
    # lots of manual checks
```

```
def handleProbability(image):  
    # lots of manual checks
```

Classifiers

- One algorithm that can be applied to many problems
- Uses **supervised learning** to use example data to predict the classification of new data



```
import sklearn
```


Training Data

- We need **descriptions** for a fruit, and a **label** to match.
- **Weight** and **texture** are *features*.
- Good features that effectively discriminate between your data types will make a very accurate classifier

| Weight | Texture | Label |
|--------|---------|--------|
| 140g | Smooth | Apple |
| 130g | Smooth | Apple |
| 150g | Bumpy | Orange |
| 170g | Bumpy | Orange |

```
import sklearn
```

```
features = [(140, "smooth"),  
            (130, "smooth"),  
            (150, "bumpy"),  
            (170, "bumpy")]
```

```
labels = ["apple", "apple", "orange", "orange"]
```

```
import sklearn
```

```
textureSmooth = 0  
textureBumpy = 1
```

```
labelApple = 0  
labelOrange = 1
```

```
features = [(140, textureSmooth),  
            (130, textureSmooth),  
            (150, textureBumpy),  
            (170, textureBumpy)]
```

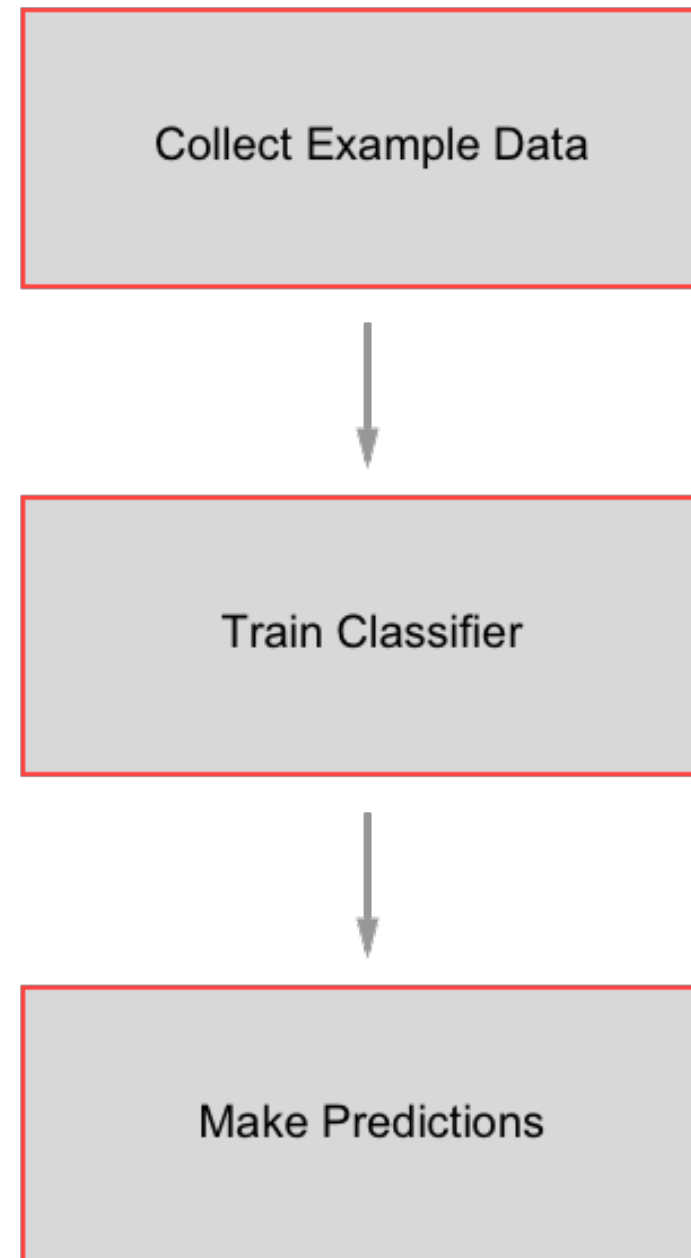
```
labels = [labelApple, labelApple, labelOrange, labelOrange]
```

Types of Classifiers

| |
|-------------------------------|
| Decision Tree |
| SVM |
| Bayesian |
| Neural Network |
| K Nearest Neighbor |
| QLearning |
| Genetic Algorithm |
| Markov Decision Processes |
| Convolutional Neural Networks |
| etc. |

Decision Trees

- One of the most basic types of classifiers
- Easy to visualize




```
from sklearn import tree
```

```
textureSmooth = 0  
textureBumpy = 1
```

```
labelApple = 0  
labelOrange = 1
```

```
features = [(140, textureSmooth),  
            (130, textureSmooth),  
            (150, textureBumpy),  
            (170, textureBumpy)]
```

```
labels = [labelApple, labelApple, labelOrange,  
          labelOrange]
```

```
classifier = tree.DecisionTreeClassifier()  
classifier = classifier.fit(features, labels)
```

**What fruit would this be
classified as?**

(160 , textureBumpy)

```
from sklearn import tree
```

```
textureSmooth = 0  
textureBumpy = 1
```

```
labelApple = 0  
labelOrange = 1
```

```
features = [(140, textureSmooth),  
            (130, textureSmooth),  
            (150, textureBumpy),  
            (170, textureBumpy)]
```

```
labels = [labelApple, labelApple, labelOrange, labelOrange]
```

```
classifier = tree.DecisionTreeClassifier()  
classifier = classifier.fit(features, labels)
```

```
print(classifier.predict([(160, textureBumpy)]))
```



[1]