Lecture 2 - Supervised Learning

Supervised Learning

> **Supervised Learning**
> Learning a mapping from input data to desired output values given labeled training data.

Supervised Learning performs 2 different types of tasks:

1. Classification
2. Regression

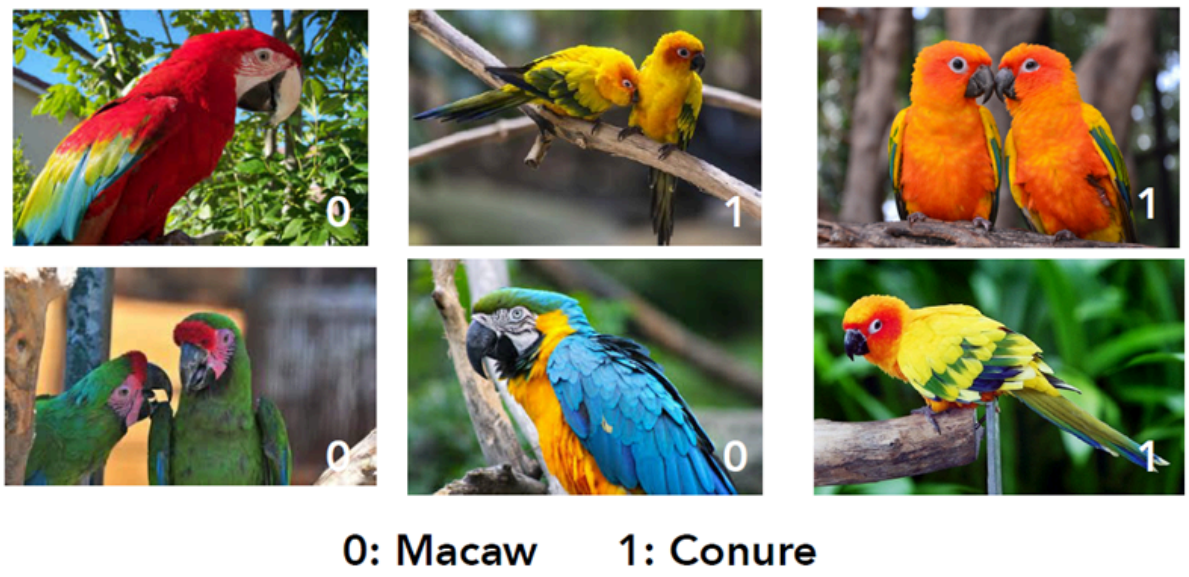
Classification

Classification is a form of predictive modeling approach to characterize the relationship between some collection of observational input data and a set of categorical labels.

Suppose we have training images from two classes, class 0 is macaw and class 1 is conure, and we would like to train a classifier to assign a label to incoming test images whether they belong to class class 0 or class 1.

In [1]:
```
from IPython.display import Image
Image('figures/classification.png', width=800)
```

Out[1]:



This is a **classification** example. Each data point was classified into a **discrete class** (either conure or macaw).

Classifiers can further be sub-categorized as **discriminative** or **generative** classifiers.

- A **discriminative** approach for classification is one in which we partition the feature space into regions for each class. Then, when we have a test point, we evaluate in which region it landed on and classify it accordingly.

- A **generative** approach for classification is one in which we estimate the parameters for distributions that generate the data for each class using Bayesian principles. When we have a test point, we can compute the posterior probability of that point belonging to each class and assign the point to the class with the highest posterior probability.

Regression

Regression is a form of *predictive modeling* approach to characterize the relationship between some collection of observational input data and a continuous desired response.

- A linear regression model is a linear combination of input values.

Consider the example below:

- The goal is to *train* a model that takes in the silhouette images with their correspondent labels (age of the person in the silhouette) and learn a linear relationship between images and age.

In [2]:
```
Image('figures/regression.png', width=800)
```

Out[2]:

- After the model is trained, the **goal** is to be able to *predict* the desired output value of any *new* unlabeled test data.

Applications of regression include: electric/solar power forecast, stock market, inventory investment, etc.

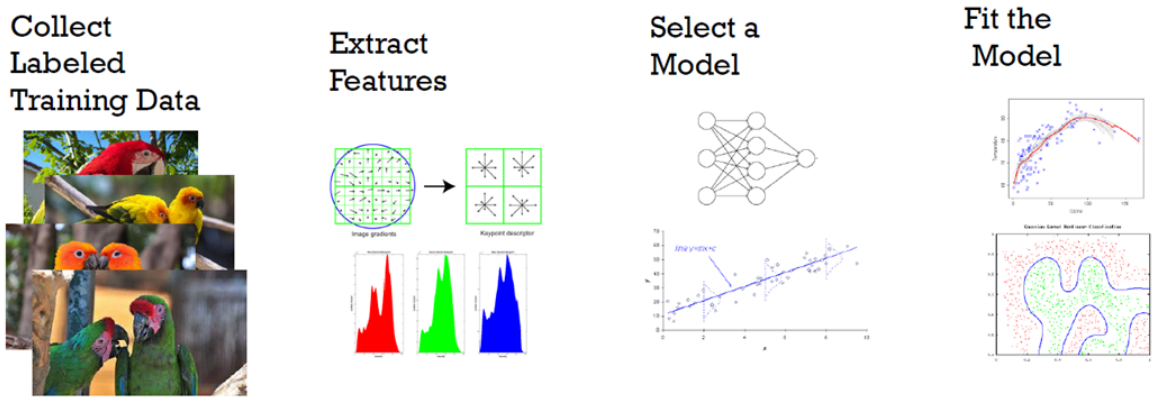
Typical Flowchart for Supervised Learning

The usual flow (but not always) for supervised learning is:

- **Training stage**
  1. Collect labeled training data - often the most time-consuming and expensive task. This constitutes the **input space**.
  2. Extract features - extract *useful* features from the input (or observational) data such that they have discriminatory information in successfully mapping the desired output. This constitutes the **feature space**.
  3. Select a model - relationship between input data and desired output.
  4. Fit the model - change model parameters (**Learning Algorithm**) in order to meet some **Objective Function**.

In [3]:
```
Image('figures/training.png', width=800)
```

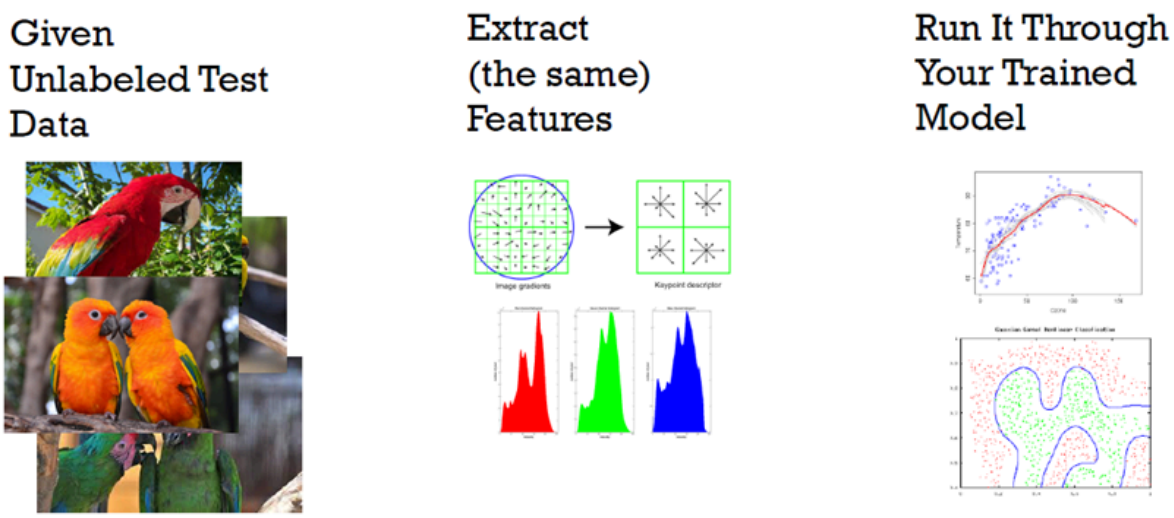
Out[3]:



- **Testing:**
  1. Given unlabeled test data
  2. Extract (the same) features
  3. Run the unlabeled data through the trained model

In [4]:
```
Image('figures/testing.png', width=800)
```

Out[4]:



Components of a Supervised ML System

Let's open the *virtual whiteboard* to describe the steps of "fitting a model".

From the whiteboard notes, we saw the flowchart for Supervised Learning.

(See whiteboard notes) The system has a **feedback** loop which will make this approach completely **autonomous** without user intervention.

- But yet we have fully control of each component's design choice.

Challenges

Some of the challenges of supervised learning include:

- How do you know if you have *representative* training data?
- How do you know if you extracted *good* features?
- How do you know if you selected the *right* model?
- How do you know if you selected the *right* objective function?
- How do you know if you trained the model *well*?

Many of these challenges are alleviated (not solved entirely, but helped significantly) with *LOTS AND LOTS* of **data** and a good **experimental design**.

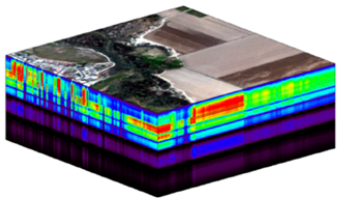
In [5]:
```
Image('figures/PHDcomics.png', width=700)
```

Out[5]:

But, sometimes, obtaining labeled training data is hard, expensive, time consuming and, in some cases, infeasible.

In [6]:
```
Image('figures/NEON.png')
```

Out[6]:



From NEON
neonscience.org