Machine Learning Workshop

Yueng De La Hoz, Ph.D.
Machine learning ⊆ artificial intelligence

**ARTIFICIAL INTELLIGENCE**
Design an intelligent agent that perceives its environment and makes decisions to maximize chances of achieving its goal.
Subfields: vision, robotics, machine learning, natural language processing, planning, ...

**MACHINE LEARNING**
Gives “computers the ability to learn without being explicitly programmed” (Arthur Samuel, 1959)

**SUPERVISED LEARNING**
Classification, regression

**UNSUPERVISED LEARNING**
Clustering, dimensionality reduction, recommendation

**REINFORCEMENT LEARNING**
Reward maximization
Machine learning: the problem setting

In general, a learning problem considers a set of $n$ samples of data and then tries to predict properties of unknown data.

We can separate learning problems in a few large categories:

- **Supervised Learning.** Making predictions using data.
- **Unsupervised Learning.** Extract structure from data. How to best represent data.
Supervised Learning

We know what the goal is. The data comes with additional attributes that we want to predict.

- **Classification**: samples belong to two or more classes and we want to learn from already labeled data how to predict the class of unlabeled data.
Supervised Learning

- **Regression**: if the desired output consists of one or more continuous variables, then the task is called regression.

You’re looking for a number
Unsupervised Learning

The training data consists of a set of input vectors $x$ without any corresponding target values. The goal in such problems may be to discover groups of similar examples within the data, where it is called clustering, or to determine the distribution of data within the input space, known as density estimation, or to project the data from a high-dimensional space down to two or three dimensions for the purpose of visualization.
Unsupervised Learning

Clustering

Dimensionality Reduction
Question

1. If I want to determine survival on the titanic.
   a. Supervised
   b. Unsupervised

2. If I want to know how much money I can make as a data scientist based on my grades, gender, and many more attributes:
   a. Supervised
   b. Unsupervised

3. If I want to know how people are grouped together naturally in a specific population.
   a. Supervised
   b. Unsupervised
Learning Model
The Dataset

<table>
<thead>
<tr>
<th>Number of new Recipients</th>
<th>Email Length (K)</th>
<th>Country (IP)</th>
<th>Customer Type</th>
<th>Email Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>2</td>
<td>Germany</td>
<td>Gold</td>
<td>Ham</td>
</tr>
<tr>
<td>1</td>
<td>4</td>
<td>Germany</td>
<td>Silver</td>
<td>Ham</td>
</tr>
<tr>
<td>5</td>
<td>2</td>
<td>Nigeria</td>
<td>Bronze</td>
<td>Spam</td>
</tr>
<tr>
<td>2</td>
<td>4</td>
<td>Russia</td>
<td>Bronze</td>
<td>Spam</td>
</tr>
<tr>
<td>3</td>
<td>4</td>
<td>Germany</td>
<td>Bronze</td>
<td>Ham</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>USA</td>
<td>Silver</td>
<td>Ham</td>
</tr>
<tr>
<td>4</td>
<td>2</td>
<td>USA</td>
<td>Silver</td>
<td>Spam</td>
</tr>
</tbody>
</table>
Linear regression is one of the most basic ways we can model relationships. Our model here can be described as $y=mx+b$, where $m$ is the slope (to change the steepness), $b$ is the bias (to move the line up and down the graph), $x$ is the explanatory variable, and $y$ is the output. For example, let’s say that the x-axis below is study_time while the y_axis is test_score.
Cost

By tweaking \( m \) and \( b \), we can create a line that will best describe the relationship. How do we know we’re close? By using a thing called a **cost function**. It literally tells us the cost. A high cost value means it’s expensive—our approximation is far from describing the real relationship. On the other hand, a low cost value means it’s cheap—our approximation is close to describing the relationship.

\[
\text{MSE} = \frac{1}{n} \sum_{i=1}^{n} (y_i - \hat{y}_i)^2
\]
Gradient Descent
def linear_regression(X, y, m_current=0, b_current=0, epochs=1000, learning_rate=0.0001):
    N = float(len(y))
    for i in range(epochs):
        y_current = (m_current * X) + b_current
        cost = sum([data**2 for data in (y-y_current)]) / N
        m_gradient = -(2/N) * sum(X * (y - y_current))
        b_gradient = -(2/N) * sum(y - y_current)
        m_current = m_current - (learning_rate * m_gradient)
        b_current = b_current - (learning_rate * b_gradient)
    return m_current, b_current, cost
Python

- Easy to understand Language.
- Flexible.
- A great community behind it.
- A vast library for Artificial Intelligence.
- Career Opportunities (Average salary is $94,053)
Installation of libraries

- Installation of python3 and pip with homebrew (mac only)
- Virtual Environment
- Pip tool
  - sklearn