Introduction

• The world is filled with data! Some examples:*  
  Healthcare  Speech Recognition  
  Search Engines  Manufacturing  Transportation  
  Financial Services  Image Processing  Retail  
  Spam Filters  Music recommendations  

• Machine Learning helps us understand and make decisions based on these data

* See [https://www.kaggle.com/datasets](https://www.kaggle.com/datasets) for a collection of datasets.
What is Machine Learning?

- The science and application of algorithms that help us make sense of (usually large) data
- “Machine learning is the science of getting computers to act without being explicitly programmed” by Prof. Andrew Ng
- Using data to answer questions
  - Training
  - Prediction

What does ML do?

- Reduce time programming
  - Custom programs with lots of rules vs. general program that you train and optimize.
    E.g., how would you write a program to check spelling errors?
- Customize and scale products
  - Expand that program with lots of rules (possibly months of programming) vs. train your program with the new data.
    E.g., how would you scale your spelling checker program to multiple languages?
- Solve problems that are perceived as unprogrammable
  - Face and speech recognition, for instance.
Explicit vs. General Rules

- Traditional Programs
  
  if email contains “Free Money”
  
  then is-spam True;

- ML Programs (learn from data)
  
  try to classify some emails;
  
  change self to reduce errors;
  
  repeat;

Different types of Machine Learning

- Supervised learning
  
  - Labeled data
  
  - Direct feedback
  
  - Predict outcome/future

- Unsupervised learning
  
  - No labels/targets
  
  - No feedback
  
  - Find hidden structure in data

- Reinforcement learning
  
  - Decision process
  
  - Reward system
  
  - Learn series of actions
Terminology

• Label
  – The parameter that we are predicting.
  – The $y$ variable in basic linear regression.

• Features
  – The input variables describing our data.
  – The $\{x_1, x_2, x_3, \ldots, x_n\}$ in basic linear regression

• Model
  – Define how the relationship between label and features.

Supervised Learning

Source: Python Machine Learning, By Sebastian Raschka, Vahid Mirjalili
Classification

- The goal is to predict the category of new label instances.
- Examples:
  - Classify new emails as Spam or Not Spam (Binary).
  - Handwritten letters (Multiclass).

```
Not Spam  Spam  New data
```

```
Training data
```

```
x_1  x_2
```

Linear Regression

- Another type of classification is regression analysis, where the goal is to predict continuous outcomes.
- For example, given the characteristics of a house, predict its price.

See Zillow Zestimate (https://www.zillow.com/zestimate/)

```
\text{The slope of the curve (weight) is } y = wx + b.
\text{The arrow represents loss.}
```

```
b \text{ is called the bias}
```
Computing the Loss

• The loss is a penalty for bad prediction
• The goal is to find a set of weights and biases that have low loss
• **Mean Square Error (MSE)** is the average squared loss per sample over the entire dataset

\[
MSE = \frac{1}{N} \sum_{(x,y) \in D} (y - \text{prediction}(x))^2
\]

where D is the data set

Adding features to our model

• What if we were to add features to our house price prediction model?
  ✓ Square footage
    – Lot size
    – Number of bedrooms
    – Number of bathrooms
    – Year built
Adding features to our model

<table>
<thead>
<tr>
<th></th>
<th>Sq. Ft</th>
<th>Lot</th>
<th>Beds</th>
<th>Bath</th>
<th>Year</th>
<th>Price</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>3,500</td>
<td>10,001</td>
<td>6</td>
<td>4</td>
<td>1890</td>
<td>1,299,000</td>
</tr>
<tr>
<td>2</td>
<td>2,295</td>
<td>5,227</td>
<td>5</td>
<td>3</td>
<td>1900</td>
<td>800,000</td>
</tr>
<tr>
<td>3</td>
<td>2,484</td>
<td>4,791</td>
<td>5</td>
<td>2</td>
<td>1910</td>
<td>1,250,000</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>N</td>
<td>1,190</td>
<td>4.050</td>
<td>3</td>
<td>1.5</td>
<td>1920</td>
<td>645,000</td>
</tr>
</tbody>
</table>

• We can represent the data as a matrix, where the \( i^{th} \) house can be written as

\[
x^{(i)} = \begin{bmatrix} x_1^{(i)} & x_2^{(i)} & x_3^{(i)} & x_4^{(i)} & x_5^{(i)} \end{bmatrix}
\]

• and each feature can be represented as a column vector

\[
x_j = \begin{bmatrix} x_j^{(1)} \\ x_j^{(2)} \\ \vdots \\ x_j^{(N)} \end{bmatrix}
\]
Adding features to our model

- And the target (price of the house) can also be represented as a column vector

\[ y = \begin{bmatrix} y^{(1)} \\ y^{(2)} \\ \vdots \\ y^{(N)} \end{bmatrix} \]

- You can imagine that with millions of houses and even more features our dataset can grow large
- Thus, the computation time of our model increases

Dimensionality Reduction

- In some cases, features might be highly correlated, and therefore redundant
- The number of features can be compressed into a smaller dimensional subspace, reducing storage and increasing performance of the model
- In some cases, dimensionality reduction can lead to better predictive performance if the dataset contains irrelevant features (or noise). In this case, we say that the dataset has a low signal-to-noise ratio
• If we only have one dataset, we can divide it into a training and test set

Training Set  Test Set

• A good practice is to randomize the split between training and test data

• Never train on the test set

Training and Test sets

• A training set is used to train and optimize our model
• The larger the training set, the more our model will be able to learn

• A test set is used to evaluate our model
• The test set should be large enough to yield statistically meaningful predictions
Reducing Loss

- Since our loss function is convex, we can use gradient descent to find the weights to minimize our cost function.
- The gradient descent is the derivative of the loss function with respect to the model parameters (weights).
- The size of the step is determined by the learning rate.
Optimizing computation

- In large data sets, calculating the gradient descent on billions of data points can be very computationally intensive.
- Computing the gradient descent in a small fraction of the data set produces similar results. This is called **Stochastic Gradient Descent**.
- An intermediate solution, is computing the gradient descent in a small batch of data. This approach is called **Mini-Batch Stochastic Gradient Descent**.

Demo 1
Machine Learning Libraries in Python

- **scikit-learn**
  - A collection of efficient tools for Machine Learning.

- **Seaborn**
  - Data visualization library built on Matplotlib.
  - [https://seaborn.pydata.org/](https://seaborn.pydata.org/)

- **TensorFlow**
  - An open-source Machine Learning framework developed by Google.
  - [https://www.tensorflow.org/](https://www.tensorflow.org/)

- **Pandas**
  - A data analysis library
  - [https://pandas.pydata.org/](https://pandas.pydata.org/)

- **SciPy**
  - An open source library for scientific computing
  - [https://www.scipy.org/](https://www.scipy.org/)

- **PyTorch**
  - Machine Learning library developed by Facebook
  - [https://pytorch.org/](https://pytorch.org/)

Managing Your Environment

- Anaconda is a open-source Python distribution for data science and machine learning.
- Anaconda comes with its own version of a virtual environment (the other being Virtualenv)
- To install Anaconda, head over to [https://conda.io/docs/user-guide/install/index.html#regular-installation](https://conda.io/docs/user-guide/install/index.html#regular-installation) and follow the instructions according to your operating system.
- Test your installation with $ conda list
Managing Your Environment

• Create a Python 3 virtual environment using

```bash
# to create a new environment
conda create -n myenv python=3.6

# to activate and do work
source activate myenv

# to deactivate when done
source deactivate
```

• And manage any packages using

`$ conda install <package name>`

More: [https://conda.io/docs/_downloads/conda-cheatsheet.pdf](https://conda.io/docs/_downloads/conda-cheatsheet.pdf)

Demo – scikit-learn
Lecture Resources

- https://developers.google.com/machine-learning/crash-course/
- Python Machine Learning, By Sebastian Raschka, Vahid Mirjalili