Machine Learning minus the Math: Münchner Volkshochschule Workshop Summary, Part 1

This brief, itemized note contains a highlight of the topics discussed in the MVHS workshop (Grundlagen des Machine Learnings und Erstellen von AI/KI-Apps mit Python). The purpose of this workshop was to provide an intuitive understanding of the key machine learning concepts. This is part 1 of 4, from the June 27th, 2019 session.

- **AI** can be defined as simulation of intellectual activity. In this way, a simple calculator is an AI device.
- **Turing test** is a famous test of AI. If a chatbot can convince me that it is human, then it is intelligent. Machine learning is a subfield of AI. Deep learning is a field of machine learning.
- The term "**Machine**" means code or computer program.
- **Machine learning** is basically the process of a piece of code finding out how to do something on its own, instead of you telling it exactly step-by-step how to do it.
- Machine learning is essentially the same thing as pattern recognition, statistical modeling, data mining, predictive analysis, knowledge discovery, etc. Machine learning is overtaking most fields of AI, such as computer vision, language understanding and robotics. ML is also becoming a prevalent tool for making use of big data.
- Machine learning can be divided into 3 sub-fields: *unsupervised learning, supervised learning and reinforcement learning*.

1. **Unsupervised learning** is essentially about simplifying data or dimensionality reduction (i.e. instead of describing the data with 10 numbers, describe it with 2). Unsupervised learning can be used to simplify the data, which can help with many other things such as storage, data retrieval, data visualization and ensuing supervised learning. Fake videos are generated using unsupervised learning.

    Manifold learning is the process of finding the surface on which the data points lie in space. Think of taking a wrinkled letter and flattening it to be able to read it.

Well-known examples of unsupervised learning algorithms include:

- **Principal Component Analysis (PCA):** a method for repositioning the coordinate system to see the data better. When I walk to the front of the TV to watch it (instead of looking at it from the side), I have done PCA.
- **K-means**: a clustering method. It allows data points switch to the cluster closest to them until there is a good division between data points. Think about separating a mix of red beans and white beans on a tray.

2. **Supervised learning**: SL is about predictions. Given a bunch of numbers (called **features**) about some objects (such as animals or houses or text), train a model which predicts something about the object (such as the name of the animal, the price of the house or the grammatical mistake in the text). This something about the object is called its **label**.

- When features are extracted from the data with hand (for example, when I decide to use the weight to differentiate cats from horses), the process is called **Feature Engineering**.
- Labels are also called **Target** or **Response**.
- When the labels are continuous numbers (such as the price of a house), then the problem is called **Regression**.
- When the labels are distinct things (such as animal names, disease state, etc.) then the problem is called **Classification**. In this course we will mainly focus on classification.
- **Training** is the process of adjusting the model **parameters** until the model can predict accurately.
- **Training set** is a part of available data used to train the model. **Test set** is a part of data used to test the prediction accuracy.
- Test set should ideally been used only once: to test the final algorithm. So, how do we know the algorithm is learning anything useful *while* we are train it? We cut out a part of the training set and don't use it for training. Instead, we keep checking our accuracy use it while we are training with the training data set. This is called the **validation** set. When you are cooking dinner for a big party (the big test), and your partner keep checking the food taste, that is validation.
- **Overfitting** happens when the model has good prediction accuracy on the training set but bad accuracy on the test set. It happens when the model has learned useless

and accidental (i.e. not essential) information about the data.

- **Underfitting**, also called bias happens when the model is too simple to learn important things about the data. So, it has bad accuracy on both training and test data sets.
- Example: imagine you want a model that differentiates wolves from dogs (given their pictures). Now it so happens that all your dog pictures are from white dogs and all the wolf pictures are from black wolves. If your model learns that the dogs are the white ones and the wolves are the black ones, then it has overfitted. This is because color is not an essential feature in this case. You need a wiser model. On the other hand, if whatever you do, your model just cannot see the difference between a dog and a wolf, then it is underfitting. You need a smarter model.

3. Reinforcement learning: if unsupervised learning is about finding patterns (e.g. grouping similar animals together) and supervised learning is about making predictions (what makes a cat, a cat and not a dog), then reinforcement learning is about learning a method (also called a policy) to win (or achieve some final goal). Winning chess can be achieved by many consequent moves, none of which is per se good or bad. Only the final reward counts.