We haven't really taken the time to take notes on any of the prior lectures (2) as they literally were strictly introductory. However now we begin with real shit. **Our lecturer; Jörg Simon Wicker**. Week before we dwelled on the different between "Machine Learning" and "Data Mining".

Supervised Learning

Branch of supervised learning (effectively taking features of examples and corresponding labels, train on them, and compare results with actuals). So train as model on some correctly classified data; taxonomy as follows.

- Example (row)
- Features (input columns)
- Class Label (output)

Some notation

Feature matrix X, entry x_{ij} (row, column), d (dimensions), n (size), y is the classification matrix, **label vector**, (with size $n \times 1$). (features = attributes).

Given an example x_i, we want our model (trained on X), to predict $\hat{y_i}$. Training error is some derivative of $\sum_i^n \hat{y_i} \neq y_i$ (or just for some i).

Decision Trees Introduction

Idea of baseline (called "mode" I think here) enters the chat. It's exactly what you think it is. We keep a mental of a tree, but think of just a bunch of "if-else" blocks too. *The issue is, how can we surface a decision tree given some data.*

Stumps

The "simple trees" that specifically have a single conditional node - two classifying leaf nodes. (see slide 15 for some pictures...)

We could build the rule via trying out many rules and rank on **classification accuracy**, the candidate rules are pitted against each other, the one with the highest classification accuracy is selected as the condition of this decision stump (**this is across all features**).

Trees

Most common decision tree learning algorithm in practice uses *greedy recursive splitting*. Okay so the notes here are not going to be fully expressive, look at lecture recording 3rd March, around 48min. Effectively we really do just recursively assemble via the "Stump" building algo, stitch together a tree. It's not too difficult.

To what depth ("over fitting")? Do we remove features on each split (no)?

Hypothesis Space

Learning can be defined as searching for the best hypothesis for all observed data. Contextualised to decision trees, your space is made up of all possible variants of decision trees that "work" for the observed data.

If the space is small enough, then it's safe to iterate through all, but *imagine if the space is just too large to brute force.*