

A decorative graphic on the left side of the slide consists of a network of thin, dark blue lines. These lines branch out from the left edge, forming a circuit-like pattern. At the end of many of these lines are small, empty circles, resembling nodes or components in a network diagram. The lines and nodes are arranged in a way that suggests a flow or connection, typical of a circuit board or a data network.

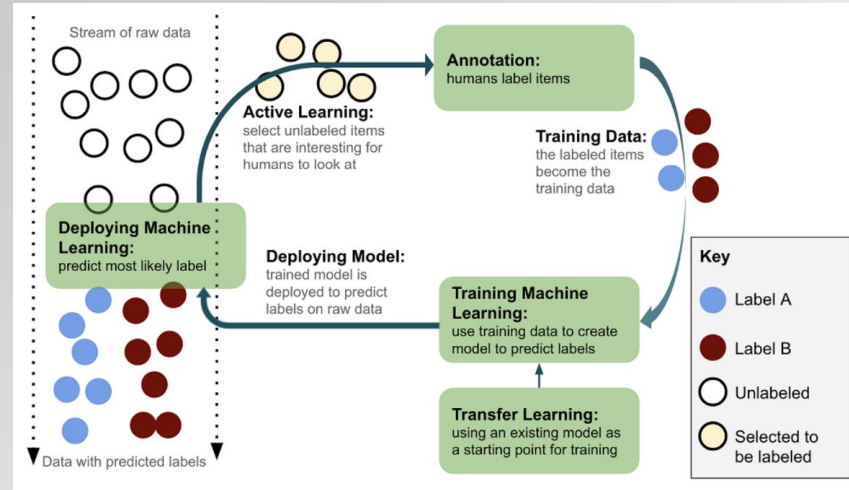
CSC 462: Machine Learning

7.8 Active Learning

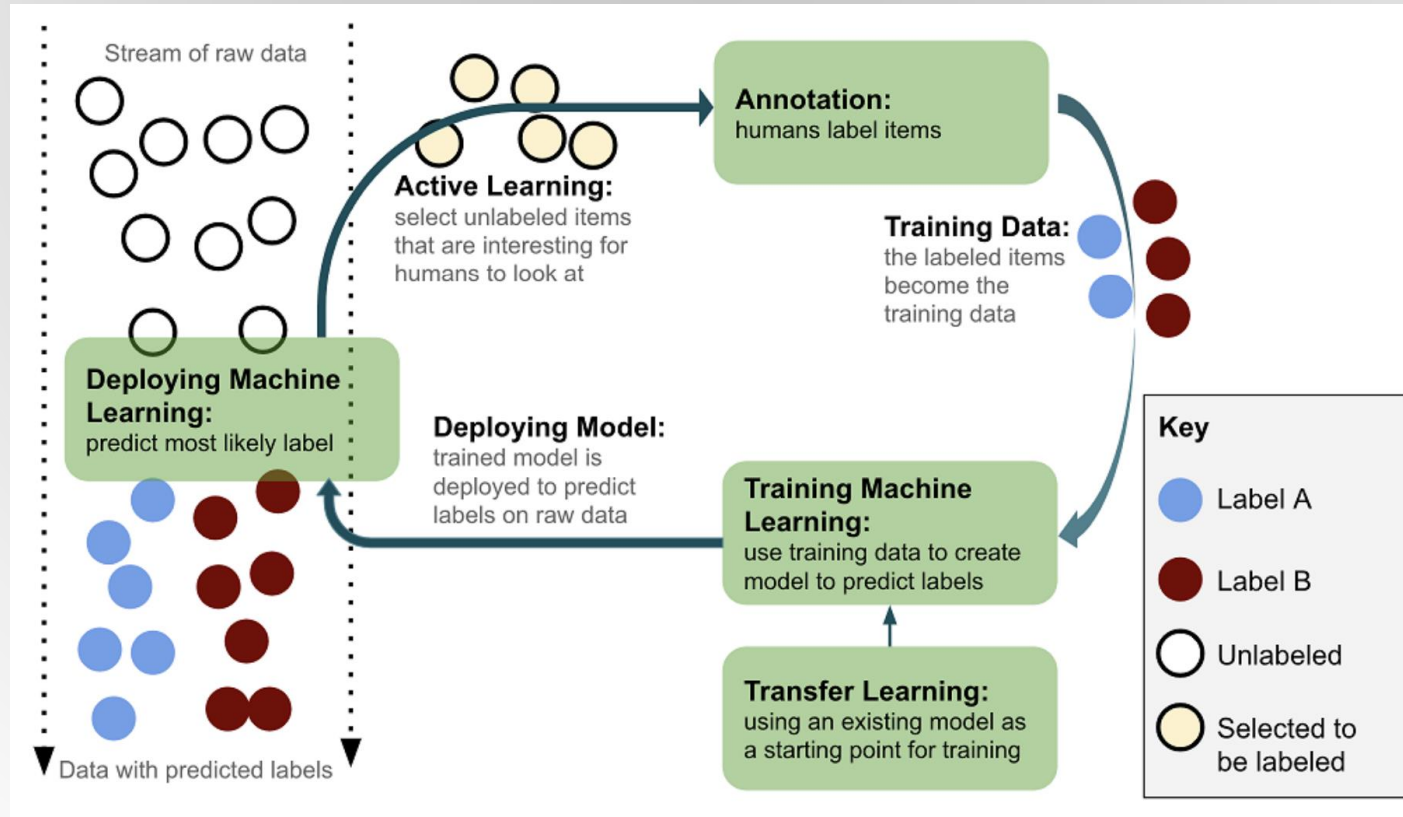
Dr. Sultan Alfarhood

7.8 Active Learning

- Usually applied when obtaining labeled examples is **costly**
- The idea is that we start the learning with relatively **few labeled examples**, and a large number of unlabeled ones, and then add labels only to those examples that contribute the most to the model quality
- Strategies
 1. Data density and uncertainty based
 2. Support vector-based



Active Learning



Data density and uncertainty based active learning

1. **Apply** the ML model (trained using the existing labeled examples) to each of the remaining unlabeled examples

- Or to some random sample of them to save the computing time

2. For each unlabeled example x , the following **importance score** is computed:

$$\text{density}(x) \cdot \text{uncertainty}(x)$$

- Density reflects how many examples surround x in its close neighborhood
- Uncertainty reflects how uncertain the prediction of the model f is for x

3. Pick the one with the highest importance score and ask the expert to **annotate it**

4. Add the new annotated example to the training set, **rebuild** the model and continue the process until some stopping criterion is satisfied

- A stopping criterion can be chosen in advance (the maximum number of requests to the expert based on the available **budget**)
- Or depend on **how well** our model performs according to some metric.

Support vector-based active learning

1. **Build** an SVM model using the labeled data
 2. Ask our expert to **annotate** the unlabeled example that lies the **closest** to the **hyperplane** that separates the two classes
- The idea is that if the example lies closest to the hyperplane, then it is the least certain and would contribute the most to

