ad was below a minimum price threshold, or does not win the auction, so the ad is not shown at all). More importantly, we get no information about what outcome would have resulted from recommending any of the other items. This would be like training a classifier by picking one class $\hat{y}$ for each training example $\boldsymbol{x}$ (typically the class with the highest probability according to the model) and then only getting as feedback whether this was the correct class or not. Clearly, each example conveys less information than in the supervised case where the true label y is directly accessible, so more examples are necessary. Worse, if we are not careful, we could end up with a system that continues picking the wrong decisions even as more and more data is collected, because the correct decision initially had a very low probability: until the learner picks that correct decision, it does not learn about the correct decision. This is similar to the situation in reinforcement learning where only the reward for the selected action is observed. In general, reinforcement learning can involve a sequence of many actions and many rewards. The bandits scenario is a special case of reinforcement learning, in which the learner takes only a single action and receives a single reward. The bandit problem is easier in the sense that the learner knows which reward is associated with which action. In the general reinforcement learning scenario, a high reward or a low reward might have been caused by a recent action or by an action in the distant past. The term **contextual** bandits refers to the case where the action is taken in the context of some input variable that can inform the decision. For example, we at least know the user identity, and we want to pick an item. The mapping from context to action is also called a **policy**. The feedback loop between the learner and the data distribution (which now depends on the actions of the learner) is a central research issue in the reinforcement learning and bandits literature.

Reinforcement learning requires choosing a tradeoff between **exploration** and **exploitation**. Exploitation refers to taking actions that come from the current, best version of the learned policy—actions that we know will achieve a high reward. Exploration refers to taking actions specifically in order to obtain more training data. If we know that given context $\boldsymbol{x}$, action a gives us a reward of 1, we do not know whether that is the best possible reward. We may want to exploit our current policy and continue taking action a in order to be relatively sure of obtaining a reward of 1. However, we may also want to explore by trying action a'. We do not know what will happen if we try action a'. We hope to get a reward of 2, but we run the risk of getting a reward of 0. Either way, we at least gain some knowledge.

Exploration can be implemented in many ways, ranging from occasionally taking random actions intended to cover the entire space of possible actions, to model-based approaches that compute a choice of action based on its expected reward and the model's amount of uncertainty about that reward.