The Machine Learning Landscape

What is machine learning?

- Machine learning is a field of programming that allows programs to learn from data using an algorithm or a model.
- A machine learning model uses a training set where each individual example is called a training instance.
- The priority is to increase accuracy and performance.

Types of Machine Learning

Training Supervision

Type	Definition	Applications
Supervised	The training set that the algorithm uses includes the desired solutions called labels or targets.	● Classification, like a spam filter ● May use regression models, like logistical regression
Unsupervised	The data is unlabeled, so it does not contain the desired solutions.	● Clustering, which detects similarities within groups, is used in blogs and social media apps ● Visualization algorithms to output 2D or 3D images ● Dimensionality reduction to simplify data without losing too much information ● Anomaly detection to detect credit card fraud, for example ● Novelty detection to detect new instances that look different from all instances in the training set ● Association rule learning to find interesting relations between attributes in large amounts of data
Semi-supervised learning	The data is partially labeled. This is useful for programs that do not	● Photo-hosting services, like Google Photos ● May be used in conjunction with

	need to classify every piece of data.	other algorithms, such as clustering
Self-supervised learning	The process of generating a fully labeled dataset from a fully unlabeled one.	● Repairing damaged images or erasing unwanted parts in an image
Reinforcement learning	The learning system, aka agent, can observe the environment, perform actions, and get rewards or penalties for rewards. The agent learns the best strategy or policy to get the most rewards over time.	● Robots, like DeepMind’s AlphaGo program

Batch vs Online Learning

Batch	Online
● The system cannot learn incrementally and must be trained using all the available data. ● Once the system is launched, it cannot learn anymore and simply applies what it was trained to do (offline learning.) ● The model’s performance decays over time as the world evolves, a phenomenon called model rot or data drift. ● The model needs to be trained from scratch using the new and old data to update the system. ● Often, batch learning requires significant computing resources and time.	● The system learns incrementally from data instances sequentially or in mini-batches. ● The algorithm is trained on each part of the data until it finishes using all the data. ● It is helpful for programs that need to learn on the fly, like detecting changes in the stock market, or programs with limited resources, like mobile applications. ● The learning rate of the program determines how fast it adapts to new data. ● Prone to bugs or users trying to cheat the game.

Instance-Based vs Model-Based Learning

Instance	Model
● The system learns from the training instances by heart and generalizes to new instances based on a measure of similarity. ● For example, a spam filter can mark emails as spam by comparing each email to the learned example based on the number of words they have in common.	● A model learns from the training set and makes predictions. ● Model selection is the process of choosing a specific model for a dataset. ● Programmers use a utility function to measure how good the model is or a cost function to measure how bad it performs.

Problems with the Data

Data Quality

- A sufficient amount of data is needed to create a good learning algorithm.
- If the data is bad, then even a good algorithm has little use.
- The data needs to represent the new cases the algorithm will generalize to.
  - If the sample is too small, there will be sampling noise
  - If the sample is too large, there will be sampling bias
- A dataset filled with errors, outliers, or noise will lead to the system performing poorly.

Feature Engineering

- If the data has too many irrelevant features, then the results of the system will be irrelevant as well.
- Feature engineering involves:
  - Feature selection: choosing the most relevant features to train the system
  - Feature extraction: combining existing features to create new features
  - Creating new features by collecting new data

Overfitting the Data

- The model performs well on the training set but does not generalize well.
- Overfitting can occur when the training set is too noisy or too small.
- Possible solutions
  - Simplifying the model
  - Collecting more data
  - Fixing errors or removing outliers
- Regularization is the process of constraining a model to simplify it and reducing the risk of overfitting.

- A hyperparameter controls the amount of regularization.

Underfitting the Data

- Underfitting occurs when the model is too simple to learn any interesting patterns.
- Possible solutions:
    - Selecting a more complex model
    - Choosing more relevant features
    - Reducing the constraints on the model by reducing the regularization hyperparameter

Training and Validation

- A common method of testing a model is to split the data into two sets – the training set and the test set – commonly using an 80/20 split.
- To validate the model, we can use holdout validation, where part of the training data is held out (this set is called the validation set) to evaluate several possible models and choose the best one.
- Going further, we can use cross-validation by using several small validation sets and evaluating each model on each validation set.