Hands on ML

Machine Learning is about making machines get better at some task by learning from data, instead of having to explicitly code rules , There are many different types of ML systems: supervised or not, batch or online, instance-based or model-based.

`Chapter 1:`

⇒ **ML** : is the science (and art) of programming computers so they can learn from data.

Spam filter is one example of ML model :

- We can't write a spam filter with explicit code it takes too much time and efforts
- a spam filter based on machine learning techniques automatically learns which words and phrases are good predictors of spam by detecting unusually frequent patterns of words in the spam examples compared The program is much shorter, easier to maintain, and most likely more accurate.
- what spam filter model does:

  1) study the problem

  2) Train ML model

  3) inspect solutions

  4) better solutions

  5) repeat if indeed

Applications for ML :

- Detecting tumors in brain scans
- Automatically classifying news articles

- Summarizing long documents automatically
- Creating a chatbot or a personal assistant
- Making your app react to voice commands
- Detecting credit card fraud
- Building an intelligent bot for a game

Training supervision :

1) supervised ML :

- the training set you feed to the algorithm includes the desired solutions, called labels
- A typical supervised learning task is classification. The spam filter is a good example
  of this: it is trained with many example emails along with their class (spam or ham),
  and it must learn how to classify new emails.
- Another typical task is to predict a target numeric value, such as the price of a
  car, given a set of features (mileage, age, brand, etc.). This sort of task is called

  regression
- **Note :** that some regression models can be used for classification as well, and vice
  versa. For example, logistic regression is commonly used for classification

2) Unsupervised ML :

- the training data is unlabeled , The system tries to learn without a teacher.
- A related task is dimensionality reduction, in which the goal is to simplify the data
  without losing too much information.

- **anomaly detection** for example, detecting unusual credit card transactions to prevent fraud
- another common unsupervised task is association rule learning, in which the
  goal is to dig into large amounts of data and discover interesting relations between
  attributes.

3) Semi-Supervised ML :

- Some algorithms can deal with data that's partially labeled.
- Google Photos, are good examples of this

4) Self-Supervised ML :

- involves actually generating a fully labeled dataset from a fully unlabeled one
- repair damaged images or to erase unwanted objects from pictures.

5) Reinforcement Learning:

- The learning system, called an agent in this context, can observe the environment, select and perform actions, and get rewards in return
- DeepMind's AlphaGo program is also a good example of reinforcement learning

▼ Batch vs Online Learning :

- whether or not the system can learn incrementally from a stream of incoming data

1) Batch learning :

- The system is incapable of learning incrementally: it must be trained
  using all the available data.
- model's performance tends to decay slowly over time, simply because the world continues to evolve while the model remains unchanged.(rot or data drift)

- training using the full set of data can take many hours, so you would typically train a new system only every 24 hours or even just weekly

2) Online learning :

- you train the system incrementally by feeding it data instances sequentially, either individually or in small groups called mini-batches. Each learning step is fast and cheap, so the system can learn about new data on the fly, as it arrives
- if bad data is fed to the system, the system's performance will decline, possibly quickly
- if there is no memory to take new data this called out of core
- learning rate definition appears

▼ Instance-based vs time-based :

1) Instance-based Learning:

- the system learns the examples by heart, then generalizes to new cases by using a similarity measure to compare them to the learned examples

2) Model-Based learning:

- build a model of these examples and then use that model to make predictions

▼ Challenges in ML :

Data :

1) Insufficient Quantity of Training Data : Even for very simple problems you typically
need thousands of examples, and for complex problems such as image or speech recognition you may need millions of examples

2) Nonrepresentative Training Data :It is crucial to use a training set that is representative of the cases you want to generalize to. This is often harder than it sounds: if the sample is too small, you will have sampling noise (i.e., nonrepresentative data as a result of chance), but even very large samples can be nonrepresentative if the sampling method is flawed. This is called sampling bias.

3) Poor-Quality Data : outliers and the missing values

4) Irrelevant Features : (solving it using feature engineering)

- Feature selection (selecting the most useful features to train on among existing features)
- Feature extraction (combining existing features to produce a more useful one as we saw earlier, dimensionality reduction algorithms can help)
- Creating new features by gathering new data

Algorithms :

1) **Overfitting the Training Data** : Overgeneralizing is something that we humans do all too often, and unfortunately machines can fall into the same trap if we are not careful

Note : Overfitting happens when the model is too complex relative to the amount and noisiness of the training data or the data is too small

- Here are possible solutions:
  - Simplify the model by selecting one with fewer parameters (e.g., a linear model rather than a high-degree polynomial model), by reducing the number of attributes in the training data, or by constraining the model.
  - Gather more training data.
  - Reduce the noise in the training data (e.g., fix data errors and remove outliers).
- To reduce the complexity we use regularization :

The amount of regularization to apply during learning can be controlled by a hyper-parameter(lambda)

2) **Underfitting** : it occurs when your model is too simple to learn the underlying structure of the data.

- Here are possible solutions:
    - Select a more powerful model, with more parameters.
    - Feed better features to the learning algorithm (feature engineering).
    - Reduce the constraints on the model (e.g., reduce the regularization
hyperparameter).

▼ Test and Validate :

- A better option is to split your data into two sets: the training set and the test set.
- The error rate on new cases is called the generalization error (or out-of-sample error), and by evaluating your model on the test set, you get an estimate of this error. This value tells you how well your model will perform on instances it has never seen before.

▼

Applications for ML :

- Detecting tumors in brain scans
- Automatically classifying news articles
- Summarizing long documents automatically
- Creating a chatbot or a personal assistant
- Making your app react to voice commands
- Detecting credit card fraud
- Building an intelligent bot for a game