1-Summarizing:
ML is the ability to achieve tasks depending on the previous experiences and data without get the explicit code.

You should use the highest percentage of data to train your model, and the model is trained and predicts the answers.

We use accuracy to measure performance of model and typically used to classification tasks.

Why ML? in simplicity some subjects like email spam is difficult to write it in traditional way or code and algorithms but in contrast using ML makes much easier and shorter than other ways.

Data mining is the ability to search for many data to enhance performance of ML.

We use ML when 1-complex problems 2-large number of rules 3-to constantly update our mode.

Examples of ML applied: Analyzing image, detected tumor in brain scan, summarizing long document automatically using NLP or Transform is working better.

According to what I understood that for predicting numbers such as market we use regression like linear regression, polynomial regression and others regression models.

Types of ML sys.:

1-Supervised: I give data training with desired solution.

Note: target word used in regression, label word used in classification, features called predictors or attributes.

2-Unsupervised: I give data training without desired solution and model clustering them according to its information.

Using unsupervised learning to visualization.

feature extraction is merge two data column with strong correlations between them to simplify the data goal.

3-semi-supervised:gives data training with fifty-fifty labels like Google

photo 4-self supervised: build fully label data set from fully unlabeled

data set Transferring knowledge is most popular using in deep neural

networks.

5-Reinforcement learning: it's a method learn that agent gets rewards on each right behavior, and this called policy.

Batch: it's training data incrementally and been up to date because performance over time decay and it's done offline. In addition, we train data after each week for slowly decay performance such as detect face of cat and dog and for each data, we train data to market as sample

Online learning is constantly updated or training data online cause high decay of performance over time. In addition, we have out-of-core learning used to load part of data runs as training step and repeat the process until run all the data.

Learning rate if it's high the system forget old data and store the latest data but if low rate it's learned going to be slowly and also be less sensitive for noises for new data

To prevent this challenge, you need to monitor the system and input data

Instance-based is learning by heart and predicts depending on similarity of trained data it's not bad but also not the best.

Model-based: trains from examples and then predict depending on

them. Model selection: consists of model and fully specifying it's

architecture.

In summary: you studied the data, need to select data, trained data with searched the model parameters to reduce cost function(how model is bad) in last made predicted for new data (this called inference).

Facing two main challenges here:

Bad data:

a-insufficient quantity of training data: for build a model you need a large amount of data and if it's for speech or image recognition may need a millions of data.

b-Nonrepresentative training data: when you build a model you must include most cases

to recognize ,and you must focus on two things avoid small data (sampling noise) and avoid very large data if it's nonrepresentative about a method is flawed, this called sampling bias.

Sampling bias: it's about getting incorrect data or not data we looking for.

Nonresponse bias: it's fall under sampling bias and it's about giving wrong data and hence the accuracy of model going to be less.

c-Poor-Quality data

we are talking about error or missing value in our data that can be handled by discard them or fix it manually and if there are 5% of data are missing you can ignore this attribute or fill it by mean or median ,or you try to train one a model with it and one model without it.

D-Irrelevant features: when building a model, you must need to use relevant features or a good features and this process called feature engineering and we do this by following three steps first

Features selection: select useful features to train.

Features extraction: combining existing features to produce a more useful

one Creating new features by gathering new data

E-Overfitting: Model do very well in training set but at all does not generalize

well. Causes of Overfitting:

1-selecting a wrong model such as linear model with less parameters rather than polynomial model.

2-gathering more training data

3-noises in the training data (fix data error or removes outliers).

Regularization: is constrained a model to make it simpler and reduce the risk and there are two parameters theta 0 for height ,theta 1 for slope these two freedom of a model

We can control a regularization by hyperparameter and it's a parameter of learning algorithm not the model and put regularization hyperparameter to a very large value you will get slope to zero and this will result not overfit but will be less to find a good solution.

F-Underfitting: accrued when model is simple like linear, and the reality is complex we can solve this problem by following these steps:

Choosing a more powerful model (With more parameters).

Feed better features.

Reduce constraint of regularization hyperparameters.

Testing and Validation:

Now to ensure that our model does well we split our data for two categories training set for training and test set for testing , and calculate error rate or called generalization error (or out-of-sample error),notice if your model gets low error rate in training and gets high in
testing so we have here Overfitting and typically we split 80% for training and 20% for testing but in some cases like if we have one million of data we can put test set just 1%.

Hyperparameter Tuning and Model Selection:

To choose a most useful model we need to follow these processes:

Split your data into training set and test set and then spilt your training set for new reduced train set and validation set and might called development set ,now training all models you have to get a better one with lowest error rate after than the winner model is retrained with main training set (reduced-training set + dev set) ,finally we test our model on test set.

We add dev set as extra because testing only in test set many times will cause problem in generalization error.

Data mismatch:

Now you can get thousands of data but not at all valid to be represented, so to evaluate your data you need to split training data to train set and dev set(there is no duplicates) after if you found result of dev set disappointing so there are two reasons 1-overfitting 2-mismatch

to know what is the exact reason we do the following

as we split training set to(reduced-training set + training-dev) as we split test set

to(reduced-test set + test-dev) ,now after doing that we train our model on reduced-training set and evaluate them by training-dev if error rate high the problem is overfitting but if the error rate appropriate next to test-dev if error rate high the problem is mismatch to solve this problem use preprocessing the data and finally test your data on reduced-test set .

In total there is no best model between other models and to specify which one is best for your model is done by evaluating.

2-What is ML:

It’s a world consisting of many methods that make a very good model because where are many problems are complex and can’t code due to not constant and the branch of ML many such as supervised learning(regression and classification),unsupervised, semi supervised, self-supervised and reinforcement learning and there are too details about how to enhance my model and what I should avoid and so on. In addition, the mechanism of ML working is from getting data and predicting answers depending on them.

3-application on ML:

Frankly, my friends and I build a reinforcement model for robot to solve maze map