Introduction

1. What is the definition of ML?

- Machine learning for a program who is performing a given task T and having performance measure P with Experience E is said to be learning if the performance measure P of the program on the task T improves with the Experience E.

2. What is a classifier?

- Classification for a system is that which inputs a vector of discrete or continuous values and produces a discrete output. The inputs are the set of attributes (x) on which a labeled output(y) is produced, thus every instance is in a form of a key value pair of (x,y).

For example: Whether a student got an internship or not is based on the cumulative GPA he/she scored in his previous semesters and the number of years of work experience he/she has. So here the inputs, the cumulative GPA and the number of years are discrete and also the output produced is binary in nature as it is just yes or no.

Learning

1. What are the 3 components of a learning system, according to the author? Explain them briefly.

- The Three components of learning system are:

Representation
Evaluation
Optimization

Representation: It is the format or a formal language in which the input or training data is feed to the classifier so that it is compatible with the computer system, thus can be classified appropriately. The representation of input data is equally important to the choosing the set of outputs or classifiers produced. This group of formatted or represented input data is called as the hypothesis space. Decision trees, Graphical models, Neural networks are some formats for representation.

Evaluation: An evaluation function [f(x)] is the decision making function or an assessment function that analyzes the input data and classifies it into the appropriate class. The evaluation function or the objective function is used to differentiate a good classifier from a bad one. The Accuracy, Error rate, Precision, divergence etc of the program are dependent on the evaluation function.

Optimization: Optimization technique is key to the efficiency of the learner as it is the method that distinguishes between the classifiers by assigning them scores. One with the best score is the most optimal classifier. Some common optimizing techniques or algorithms are Greedy search, branch and bound, Gradient descent, Conjugate gradient, Linear programming, etc.

3. Algorithm 1 presents a decision tree learner that determines whether to split a decision tree node and how to split it. It depends on information gain between attributes and the predicted value. Do a quick search on information gain and write down its definition and equation below.

- To define Information gain we need to define entropy first.

Entropy is which that characterizes the (im) purity of an arbitrary collection of examples. Generally, if the target attribute can take on c different values, then the entropy of S relative to this c-wise classification is defined as C

Entropy(S) $=\sum_{i=1}^{C} -pi\ log_2 pi$

Where pi is the proportion of S belonging to class i.

Given an entropy, information gain, is simply the expected reduction in entropy
Caused by partitioning the examples according to this attribute.

Gain(S, A) of an attribute A, relative to a collection of examples S, is defined as

$$\mathrm{Gain(S, A)} = \mathrm{Entropy(S)} - \sum_{v \in Values(A)} \frac{|\mathrm{S_v}|}{|\mathrm{S}|} \mathrm{Entropy(S_v)}$$

Generalization

1. Why is generalization more important than just getting a good result on training data i.e. the data that was used to train the classifier?

- Since we cannot completely rely on a training data set of examples. It is considered that Generalization is to go beyond the training data set. Performing well on training data set is like just memorizing the examples and it gives an illusion of success. Hence generalization is important as it exposes the classifier to a new set of unknown data and thus that for the classifier becomes more of random guessing instead of training data.

2. What is cross-validation? What are its advantages?

- The classifier can get corrupt by test data in various number of ways. A common example is lot of tuning of the classifier by the test data. Thus this impurity or contamination can be validated by cross validation. In cross validation the training data in divided into a number of subsets, holding out each one while training on the rest, testing each learned classifier on the examples it did not see, and averaging the results.
The advantage of cross validation is that each subset becomes the train and test data sets also the averaging reduces the contamination.

3. How is generalization different from other optimization problems?

- Generalization is different from other optimization problem as generalization does not have the exact objective function to optimize. Also in generalization the training data is used as a replacement to the test error which can be dangerous. On the bright side, the evaluation function is an approximation to the actual goal function, thus there is no need of fully optimizing it.

Data alone is not enough

1. Try to understand how a function involving 100 Boolean variables would lead to a total 2100 different possible examples (no need to write anything down, just try to understand). If you have a scenario where the function involves 10 Boolean variables, how many possible examples (called instance space) can there be? If you see 100 examples, what percentage of the instance space have you seen?

- For 10 Boolean variables the total number of possible instance will be 210 = 1024 instances.
Percentage of instances seen after 100 instances = (100/1024)*100 = 9.765%

2. What is the "no free lunch" theorem in machine learning? You can do a Google search if the paper isn't clear enough.

- Every learner must embody some knowledge or assumptions beyond the data it's given in order to generalize beyond it. "No free lunch" theorems means no learner can beat random guessing over all possible functions to be learned. According to the theorem, Learning cannot be achieved just by seeing the training data.

3. What general assumptions allow us to carry out the machine learning process? What is the meaning of induction?

- The general assumptions for carrying out machine learning are:
1. Smoothness: The function we are leaning so as to produce the desired output from the input features is not too steep.
2. Similar examples have similar labels: Similar instances are mapped to the same labels.
3. Limited dependencies: The instances given are independent of each other
4. Limited complexity: The model just isn't confusing or convoluted and can be represented fairly simply.

Induction is a technique is more powerful than deduction which uses comparatively less input knowledge to produce more useful results.

5. How is learning like farming?

- Learning is compared to farming is a very simple manner. The role of seeds in farming is mapped to knowledge in learning and the nutrients to data. Thus like farming, as nature does most of the work of growing crops similarly in learning the objective function does majority of work by classifying the inputs into the appropriate output labels.

Overfitting

1. What is overfitting? How does it lead to a wrong idea that you have done a really good job on training dataset?

- Overfitting in machine learning is when a model performs too well on the training data set i.e. that the model has learned the noise and the random fluctuations as concepts. The problem is that these new concepts do not apply to the new data set or the test data set and thus have a negative impact on the classifier's ability to generalize.
-
For example: When your learner outputs a classifier that is 100% accurate on the training data but only 50% accurate on test data, when in fact it could have output one that is 75% accurate on both, it has over fit and giving a wrong idea that you have done a good job on training data set.

2. What is meant by bias and variance? You don't have to be really precise in defining them, just get the idea.

- Bias is a learner's tendency to consistently learn the same wrong thing. e.g.: When the classes are not separated by a hyperplane, so bias is the difference between the predicted value and the true target.
Variance is the tendency to learn random things irrespective of the real signal. E.g.: There can be multiple decision trees for the same phenomenon or instance.

3. What are some of the things that can help combat overfitting?

- Solutions to overfitting are:
1. Cross validation - By using it to choose the best size of decision tree to learn, as too many parameters can make itself start to overfit.
2. Regulation term – To keep the classifier in check a regulation term is added to the evaluation function. Thereby this wraps the classifier with more structure and leaves very little space for overfitting.
3. Statistical significance test – It used to identify how different is the new distribution of the class structure different from the previous.

Intuition fails in high dimensions

1. Why do algorithms that work well in lower dimensions fail at higher dimensions?

- There are two main reasons for failure:
    1. Generalizing becomes exponentially harder as the dimensionality (number of features) of the examples grows as a fixed sized training set covers a diminished fraction or a part of the input space. So when the input set has around 100 dimensions, the training set will have a trillion example but the classifier will cover only a fraction (about 10-18) of the input space thus making it very hard for the machine to learn.

    2. Similarity-based reasoning on which the machine learning depends on breaks down at high dimensions. This means that when there are just 2 relevant attributes the classification is not at all complex but when another 98 unimportant attributes or noise is added it contaminates the classifier forcing it to produce undesired outcomes.

2. What is meant by "blessing of non-uniformity"?

- In high dimensions it's hard to understand what is happening. This in turn makes it difficult to design a good classifier. Naively, one might think that gathering more features never hurts, since at worst they provide no new information about the class. But in fact their benefits may be outweighed by the curse of dimensionality.
Fortunately, there is an effect that partly counteracts the curse, which might be called the "blessing of non-uniformity."
In most applications examples are not spread uniformly throughout the instance space, but are concentrated on or near a lower-dimensional manifold.

Theoretical guarantees

1. What has been one of the major developments in the recent decades about results of induction?

- One of the major developments of recent decades has been the fact we can have guarantees on the results of induction, particularly if we're willing to settle for probabilistic guarantees.

Feature engineering

1. What is the most important factor that determines whether a machine learning project succeeds?

- The most important factor is the features used. If you have many independent features that each correlate well with the class, learning is easy. On the other hand, if the class is a very complex function of the features, you may not be able to learn it. Often, the raw data is not in a form that is conducive to learning, but you can construct features from it that are. This is typically where most of the effort in a machine learning project goes and it for no doubt is the most important feature.

2. In a ML project, which is more time consuming – feature engineering or the actual learning process? Explain how ML is an iterative process?

- Feature engineering is more time consuming than actual learning process as feature engineering requires to gather data integrate it, clean it and pre-process it, and how much trial and error can go into feature design. Also feature engineering is more difficult and domain specific while learners can be largely general purpose.
Machine learning is not a one-shot process of building a data set and running a learner, but an iterative process of running the learner, analyzing the results, modifying the data and/or the learner, and repeating.

3. What, according to the author, is one of the holy grails of ML?

More and more automation of the feature engineering process is the holy grails of Machine learning.

More data beats a cleverer algorithm

1. If your ML solution is not performing well, what are two things that you can do? Which one is a better option?

- There are two main choices:
- Design a better learning algorithm.
- Gather more data (more examples, and possibly more raw features, subject to the curse of dimensionality).

Gathering more data is a better option as it is the quickest path to success also as a rule of thumb, a dumb algorithm with lots and lots of data beats a clever one with modest amounts of it.

2. What are the 3 limited resources in ML computations? What is the bottleneck today? What is one of the solutions?

- The three main limited resources are:
- Time.
- Memory.
- Training data.

Time resource is the bottleneck in today's time as the is tons and tons of data going in everyday and not enough time to process it so majority of it goes wasted.
One of the solution is to come up with fast ways to learn complex classifiers and in recent times there has been a remarkable development in this direction.

3. A surprising fact mentioned by the author is that all representations (types of learners) essentially "all do the same". Can you explain? Which learners should you try first?

- The reason for using cleverer algorithms has a smaller payoff than you might expect is that, to a first approximation hence they all do the same. This is surprising when you consider representations as different as, say, sets of rules and neural networks. All learners essentially work by grouping nearby examples into the same class; the key difference is in the meaning of "nearby." With non-uniformly distributed data, learners can produce widely different frontiers while still making the same predictions in the regions that matter. This also helps explain why powerful learners can be unstable but still accurate.
Thus we should try the simplest learners first. More sophisticated learners are seductive, but they are usually harder to use, because they have more knobs you need to turn to get good results, and because their internals are more opaque.

4. The author divides learners into two types based on their representation size. Write a brief summary.

- Learners can be divided into two major types:
- Those whose representation has a fixed size, like linear classifiers.
- Those whose representation can grow with the data, like decision trees.
-

Fixed-size learners can only take advantage of so much data. Variable-size learners can in principle learn any function given sufficient data, but in practice they may not, because of limitations of the algorithm (e.g., greedy search falls into local optima) or computational cost. Also, because of the curse of dimensionality, no existing amount of data may be enough. For these reasons, clever algorithms—those that make the most of the data and computing resources available—often pay off in the end, provided you're willing to put in the effort.

Learn many models, not just one

1. Is it better to have variation of a single model or a combination of different models, known as ensemble or stacking? Explain briefly.

- Systematic empirical comparisons showed that the best learner varies from application to application, and systems containing many different learners started to appear. Effort now went into trying many variations of many learners, and still selecting just the best one. But then researchers noticed that, if instead of selecting the best variation found, we combine many variations, the results are better—often much better—and at little extra effort for the user.

Simplicity does not imply accuracy

1. Read the last paragraph and explain why it makes sense to prefer simpler algorithms and hypotheses.

- It make sense to prefer the simpler hypotheses because simpler classifiers are easy to understand, debug, and update as compared to complex classifiers also as it is said simplicity is a virtue in its own right, not because of a hypothetical connection with accuracy.

Correlation does not imply causation

1. It has been established that correlation between independent variables and predicted variables does not imply causation, still correlation is used by many researchers. Explain briefly the reason.

Correlation is used by many researchers even though it does not imply causation because correlation between independent input variables and the predicted output variable guides us or gives a good idea of which attributes or variables have the most impact on the predicted output and filtering out the variables which barely contribute or are termed as noise. Machine learning is usually applied to observational data, where the predictive variables are not under the control of the learner, as opposed to experimental data, where they are. Correlation is a sign of a potential causal connection, and we can use it as a guide to further investigation. Thus removing the attributes that may contaminate the output results in a more concentrated and a highly close approximation to the true objective function. Thus using correlated variables is recommended.
For example: If we find that beer and diapers are often bought together at the supermarket, then perhaps putting beer next to the diaper section will increase sales.