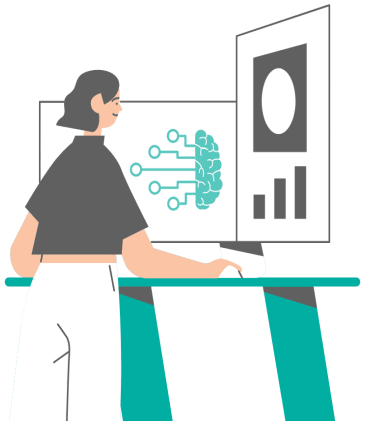
Importance of Domain Knowledge for Machine Learning?

If you incorporate domain knowledge into your architecture and your model, it can make it a lot easier to explain the results, both to yourself and to an outside viewer. Every bit of domain knowledge can serve as a stepping stone through the black box of a machine learning model.



Developing machine learning models involves a lot of steps. Whether you're working with labeled or unlabeled data, you might think numbers are just numbers, and it doesn't matter what each of the features of a dataset signifies when it comes to spitting out insights with the potential for true impact. It's true that there are tons of great machine learning libraries out there like scikit-learn which make it straightforward to gather up some data and plop them into a cookie-cutter model. Pretty quickly, you might start to think there's no problem you can't solve with machine learning.

Frankly, that's a beginner's mindset. You are not yet aware of everything you don't know. Datasets given in machine learning courses or the free ones you find online have often already been groomed and are convenient to use when applying machine learning models, but once you take your skills and knowledge out of the play-pen and into the real world, you'll face some additional challenges.

Lots of people believe that domain knowledge, or additional knowledge regarding the industry or area the data pertains to, is superfluous. And it's kind of true. Do you NEED domain knowledge in the area you're developing the model? No. You can still produce fairly accurate models without it. Theoretically, deep and machine learning are black-box approaches. This means you can put labeled data into a model without deep knowledge of the area and without even looking at the data very closely.

But, if you go down this route though, you'll have to deal with the consequences. This is a very inefficient way to train classifiers, and in order to properly function, you'll require massive amounts of labeled datasets and a lot of computational power in order to produce accurate models.

If you incorporate domain knowledge into your architecture and your model, it can make it a lot easier to explain the results, both to yourself and to an outside viewer. Every bit of domain knowledge can serve as a stepping stone through the black box of a machine learning model.

It›s very easy to think that domain knowledge isn't required because for lots of visible datasets like COCO, the limited domain knowledge that is required is part of being a seeing human. Even more complex data sets that contain cancer cells are similarly obvious to the human eye, despite a lack of expert-level knowledge. You can do basic evaluation of similarity or differences between cells without any specific medical knowledge.

Natural language processing (NLP) and computer vision are prime examples of areas where it's easy to think that domain knowledge is entirely unnecessary, but more so because they are such normal tasks for us, we may not even notice how we're applying our domain knowledge.

If you start working in areas like outlier detection, which isn't such an everyday human task, the importance of domain knowledge quickly becomes apparent.

Domain Knowledge for Data Pre-Processing

Let's dig into how domain knowledge can be leveraged in the data pre-processing step of the machine learning model development cycle.

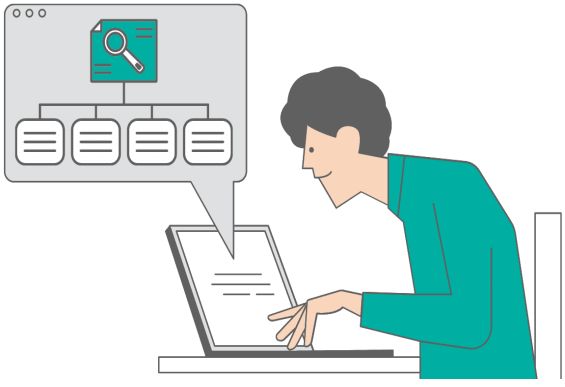
In a dataset, not every data point has the same value. If you collect ١٠٠ new samples which are identical, they don't help the model learn any additional information. They might actually focus the model in a specific direction which isn't important.

If you're looking at ١٠٠ pictures of umbrellas, and you know this model is supposed to classify all kinds of accessories, it's clear that your sample dataset isn't representative of the whole population. Without domain knowledge, it can be very challenging to know which data points add value or if they are already represented in the data set.

If you're working in an area that doesn't so easily lend itself to your existing general knowledge, you can build in biases through the training data which can hurt the accuracy and robustness of your model.

Another way in which domain knowledge can pack a punch in the data pre-processing step is in determining feature importance. If you have a good feel for the importance of each feature, you can develop better strategies to process the data accordingly. It's really important to understand what the actual features are in order to do so. This has a big influence on how you handle the features going forward.

Domain Knowledge for Choosing the Right Model

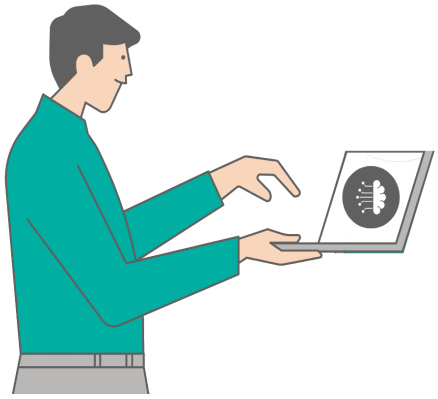


There are many different machine learning models, and some may be more fitting than others given many factors. Is the data labeled or unlabeled? How much data do you have? What kind of data types are the features? Are the data types of the

features homogeneous? Is your target output a continuous value or a classification? Choosing the right model is important, but it's very rare to be able to apply your selected model directly without making adjustments to it. Random forests, for example, can handle heterogeneous data types right out of the box.

Selecting the right model requires in-depth machine learning knowledge, but there are lots of resources out there to help you make your selection if you're not quite a machine learning expert. I've gathered my top three from machine learning cheat sheets from Towards Data Science, datacamp, and Microsoft.

Domain Knowledge for Adjusting Model and Architecture



Domain knowledge allows you to better adjust the model to fit the situation. Mathematical optimizations only go so far, and often to get big jumps in improvement, it's crucial to have considerable domain knowledge in the area.

A significant way to apply your domain knowledge to drive an improvement in the accuracy and robustness of your model is to incorporate the domain knowledge into the architecture of the model you are developing.

As I mentioned before, natural language processing is one of the areas of machine learning which makes it clear how domain knowledge can be helpful. Let's talk about word embeddings and attention to showcase how speaking a human language is a big help, but thinking like a linguist can really elevate the performance of a natural language processing model.

Domain Knowledge for Natural Language Processing

Domain knowledge has been applied to all applications of machine learning. Small adjustments have been made over the last few decades to better apply machine learning models in many areas. Domain knowledge has definitely been applied to the models used in natural language processing. Let's walk through a few examples of how these developments came about.

Word Embeddings

If you think about numbers and words, there's a pretty big difference in how we think about them. If you have the heights of everyone in a group, you can easily spit out some stats regarding the median, outliers, etc, all based on height. If everyone in the group gave you a word to represent how they are feeling today, how would you convert that to any kind of meaningful aggregate?

You should consider what you can do to create a digital representation of a word. Should you just use letters? Does that make sense? As a person who speaks the language, we immediately have the meaning behind the word. We do not store words by their letters. Think about a tree. Did you picture a tree or did your mind go to t-r-e-e? Storing the representation of a word as the letters doesn't really bring us any advantage when it comes to understanding meaning or significance.

Word embeddings are "a type of word representation that allows words with similar meaning to have a similar [numerical] representation". The numerical representations are learned using unsupervised

learning models.

These numerical representations are vectors that represent how a word is used. This numerical representation goes so far as to allow you to use the Euclidean distance between two-word representations to quantify how similar two words are used in the training text.

The vectors for "Adidas" and "Nike" would likely be quite similar. Exactly what each field of the vector represents is certainly unclear as they're developed using unsupervised learning, but it makes sense that a word that represents a similar concept has a similar representation as far as the model understands.

Check out our post "Supervised vs Unsupervised Learning" if you want to know what supervised and unsupervised learning actually are and the algorithms that use these learning approaches.

Attention

Attention is a very valuable and useful concept. Attention has made its way into natural language processing and image recognition in the world of machine learning models for a very good reason.

Natural language processing for translation has been a deep learning model since the early ۲۰۰۰s. Around ۲۰۱۳, long short-term memory (LSTM) debuted in the field and dominated for a few years. An LTSM model reads the sentence, creates a hidden representation, and then uses the hidden representation to generate the output sentence.

As humans, if we translate, we don't just read the sentence and spit out the translation. We tend to look at the whole sentence again and again, or we'll focus on certain parts when we want to revisit the context of the target word.

For example the word "read". Is it representing present or past tense? Which other parts of the sentence do you focus on to determine that? Do you need information from surrounding sentences? "I read books every year" means one thing on its own, but if I were to say, "I did many things before I went to university. I read books every year. I ate dinner with my parents every day," the sentence takes on a different meaning, and "read" represents an action that took place in the present in the first example and in the past for the second.

Attention builds these relationships between different words in the sentence or sentences. For each word we want to translate, it highlights different words in the original sentence according to the importance of these associated words with the target word. Someone who isn't an experienced or professional translator might just translate word-for-word, and while the general information would still be conveyed, accounting for the importance of associated words when translating the target word will produce a much more accurate translation. Attention allows us to build that professional translation pattern into the architecture of the deep learning model.

Why Domain Knowledge Is Crucial for Machine Learning

Without domain knowledge, you can check all the boxes of producing an acceptable model which spits out some numbers. With domain knowledge, you'll know what data is best to use to train and test your model. You will also realize how you can tailor the model you use to better represent the data set and the problem you're trying to tackle, and how to make the best use of the insights your model produces.

Machine learning is a toolbox. If you pull out an electrical saw, you'll probably manage to cut some wood, but you probably won't be able to construct a bunch of cabinets without the expert knowledge of a carpenter. Domain knowledge will allow you to take the impact of your machine learning skills to a much higher level of significance.