Approaching a Data Science Problem in the Wild

While we have conveniently applied several machine learning algorithms on meticulously-collected datasets, actually tackling data science problems in the real world is more complicated. Here we discuss the different questions one must be aware of when approaching a data science problem. In the projects so far, we have repeated the same process described below — just never mentioned it explicitly! We abstract a lot of details from what we have learned and give you a rundown of the process.

What is the data we are dealing with?

The type of data you are dealing with, like natural text and images, will greatly affect what algorithms are most appropriate for the situation. Since different types of data lend themselves to different machine learning tasks uniquely, with practice you will find particular algorithms well-suited for particular data types.

What kind of information do you want to extract from the data?

What outputs are you expecting from your program? Are you trying to learn a function predicts categorical labels $\boldsymbol{y}$ from data $\mathbf{X}$ or continuous values? Or are you trying to learn a pattern implicit in the input data $\mathbf{X}$? Often we are looking for a function that takes in $\mathbf{X}$ (input space) and outputs $\boldsymbol{y}$ (output space). We train on a training set of and assume the function will generalize on new unseen data.

How do we evaluate our model?

We need a metric that measures how well the model does. This is typically some sort of loss function that the training function optimizes and/or readily interpretable metrics like accuracy. Experimenting with different loss functions can lead to interesting changes in the performance of a machine learning algorithm.

Splitting the data into training/validation/test sets

In the real world, we typically have a limited dataset we can work with to build models. The data available needs to be used both to train and test the models. Thus, we need to split the dataset into a **training set** and a **test set**; we usually split the training set into multiple **validation sets** for cross-validation as well.

The training set is used to learn the model's parameters that optimize the chosen metric, typically a loss function. The testing set is used to evaluate the model *in the end*, and must be completely separate from the training process to ensure a fair test. The validation sets thus help evaluate the trained models and iterate through different versions.

Models almost always have a set of hyperparameters as well, which are tuned based on the performance on the validation sets. Once the model is optimized with the optimal hyperparameter setting, it can be compared against other machine learning methods on the test set. Finally, the best model can be trained on all available data if it will be used for prediction, regression, or other tasks in "production".

Processing the data for input

As we have discussed before, raw data come in different types: floating point scientific measurements, natural text, images etc. Since machine learning models operate on vectors of numbers, each data point must be converted into a vector-based representation with a fixed number of features across all data points.

Raw datasets are also often "dirty" and need to be "cleaned". For instance, one might try to remove misclassified or extremely noisy examples from the dataset. The next step is to "preprocess" the data before operated on with algorithms. For instance, one might change natural text to lowercase, remove punctuation, remove common stop-words like 'a', 'the', 'and' in English. For images, one might crop all images to a standard size, limit RGB values to not be too extreme, and so on.

Selecting the model

It is the data scientist's responsibility to look at the problem, decide what assumptions about the data can be made, and leverage said choices to pick the best model for a given problem. It is good practice to start with simple models such as linear models and neighbor-based approaches, then move onto complex models like CARTs.

Training and Evaluating

Finally, we train and evaluate on the data and pick the models that generalize best on unseen data (test set)!