ML_Introduction

May 29, 2019

```
In [1]: %matplotlib inline
        import matplotlib
        import seaborn as sns
        sns.set()
        matplotlib.rcParams['figure.dpi'] = 144
```

1 Introduction to Machine Learning

Machine Learning is a broad collection of tools used to find patterns in existing data and to make predictions about future data.

For example, we can use ML algorithms to approximate an unknown function or to group data into clusters.

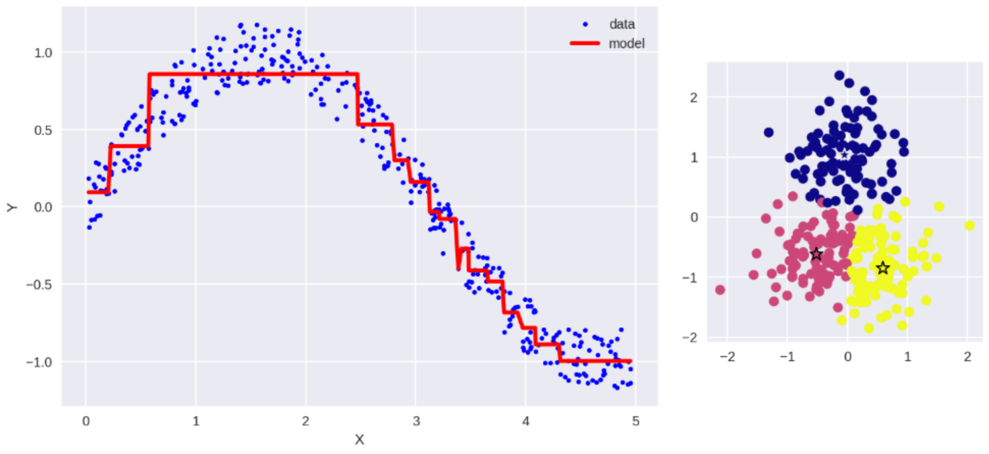
Machine Learning is an area of active research and development and has produced many impressive applications:

- Voice and image recognition
- Translation services
- Financial models
- Spam detection
- Self-driving cars

However, Machine Learning is not the answer to every problem, and it has limitations:

- ML models are only as good as the data they train on. If something changes between training and application, or if we're missing important features to start with, then we won't get good results.
- Human intelligence is required to choose an appropriate algorithm (which sometimes includes choosing model parameters) and to evaluate performance. We can run most ML algorithms and get a result with relatively little effort, but it's up to us to make sure the result is meaningful and useful.

- ML algorithms often prioritize performance over explicability. We may train a model that can make good predictions but not be able to explain where those predictions are coming from.



ex

1.1 Statistics vs Machine Learning

The exact border between Statistics and Machine Learning is poorly defined, and many statistical methods (e.g. Linear Regression) are considered to be part of the ML toolkit. However, we can identify unique features which are characteristic of each.

Statistical models are more likely to depend on distributional assumptions about data (parametric analysis) and more likely to offer theoretical or closed form solutions to relevant optimization problems. By contrast, Machine Learning models generally emphasize flexibility over theory, and often use incremental optimization algorithms like gradient descent.

1.2 Data as a Matrix

In order to do Machine Learning, we need to represent our data mathematically. Our convention will be to represent each data set as a matrix (NumPy array) where:

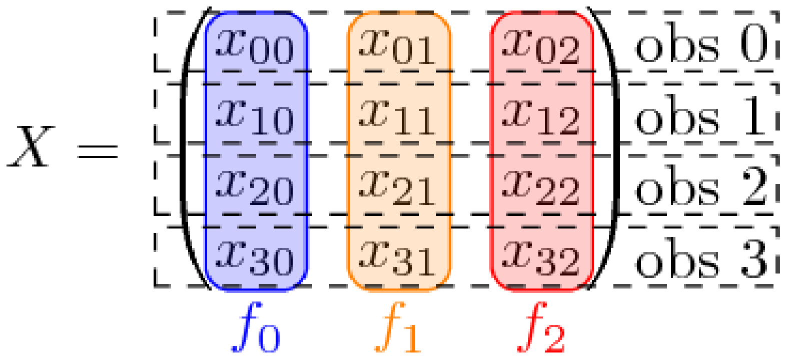
- **rows** correspond the different observations or data points
- **columns** correspond to different *features* (data attributes)

For example, if we have a (very small) data set which has three features f_0, f_1, f_2 and four observations, we can represent this as a 4x3 matrix.

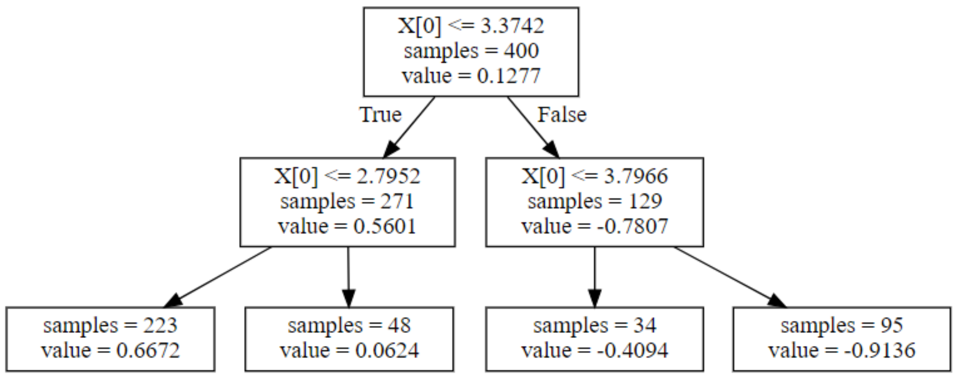
1.3 Models as Functions

Machine Learning **models** are functions which take observations as inputs and return some outputs. Frequently these outputs are predictions, but they might be cluster labels or something else, depending on the application.

For example, a linear model might predict a target variable using the formula $f(x) = 2x + 3$. However, the rule used by a model does not need to have an explicit mathematical representation. In general, it can by any algorithm or procedure. For example, a decision tree model works by dividing data into groups (by iteratively comparing feature values to predetermined split values) and then making a different constant prediction for each group.



feature_matrix



tree_example