Intro to Machine Learning

Colin Mullaney

What IS machine learning?

“Machine learning is the process of using various algorithms or *models* to identify **trends** in data”

“Use known examples to make **generalizations** about unknown examples”

“*Teaching* a machine to find **relationships** and **patterns** in data“

“Using existing data to accomplish some goal”

Supervised vs Unsupervised Learning

Supervised:

Known outcome (“label”)

Regression (continuous/numeric outcome)

Classification (binary/categorical outcome)

Unsupervised:

No defined/known outcome

Try to learn a “hidden structure” from the dataset

Clustering (making groups from similar data points)

General Terms:

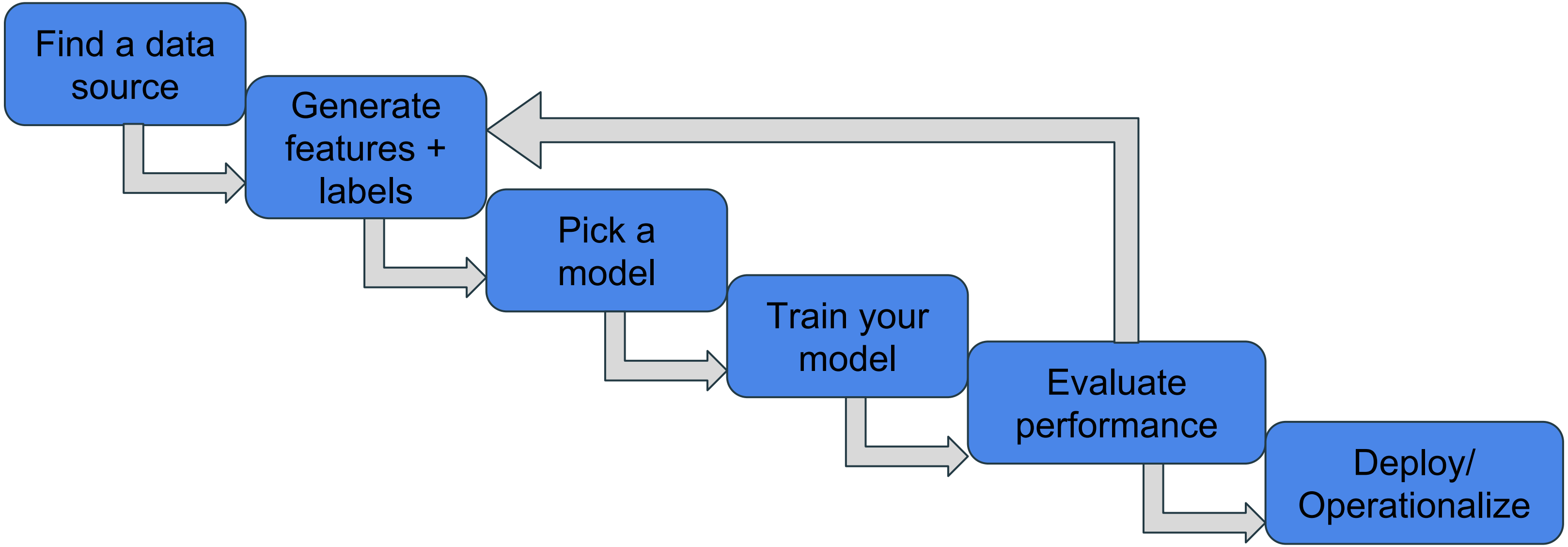
Model: a function from some input (X) to some output (Y)

Features: the input data (X) to your model

Labels: the output (Y) that you are trying to predict

Train/Fit: process of determining the function from X => Y

General M.L. Workflow:



Step 0: Have a data source

- A great place to start searching is Kaggle.com
- They have a ton of example datasets, all with the intention of being used for machine learning
- Want a clear outcome variable to predict (Y)
- Want additional input data to use for prediction (X)
- Be careful of null values!

Step 1: Generate features

- For most machine learning models, your input data (X) needs to be some numerical form
- Certain models can accept Boolean values, or even categories
- In general, it is best to convert your input data to numerical values whenever possible (so it is easier to switch back and forth between models)

Types of Input Features

- Continuous/Numeric
- Binary: 1/0, True/False
- Categorical: A, B, C...
- Ordinal: Categories where order matters
  (age groups, high/med/low risk, etc)
- Dummy variables: turning categories into multiple binary features

“Dummy” Variables

- Turning a categorical variable into multiple numeric variables
- Each category option is transformed into a binary variable

id	State
1	MA
2	CT
3	MA
4	RI

id	State_MA	State_CT	State_RI
1	1	0	0
2	0	1	0
3	1	0	0
4	0	0	1

Type of Prediction Problems (labels)

- Continuous/Numeric (regression)
- Categorical (classification)
    - Binary (2 outcomes)
    - Multi-class (3+ outcomes)
    - Multi-label (any combination of multiple outcomes)

Step 2: Split up your data

- For training a supervised machine learning model, you must split up your available data into a **training** set and a **testing** (or evaluation) set
- You need to have a “held-out” dataset that your model does not train on, in order to evaluate your model without bias
- If you train and test on the same set of data, your model’s performance will be inflated/misleading
- Typical splits are: 80/20, 75/25, ….etc

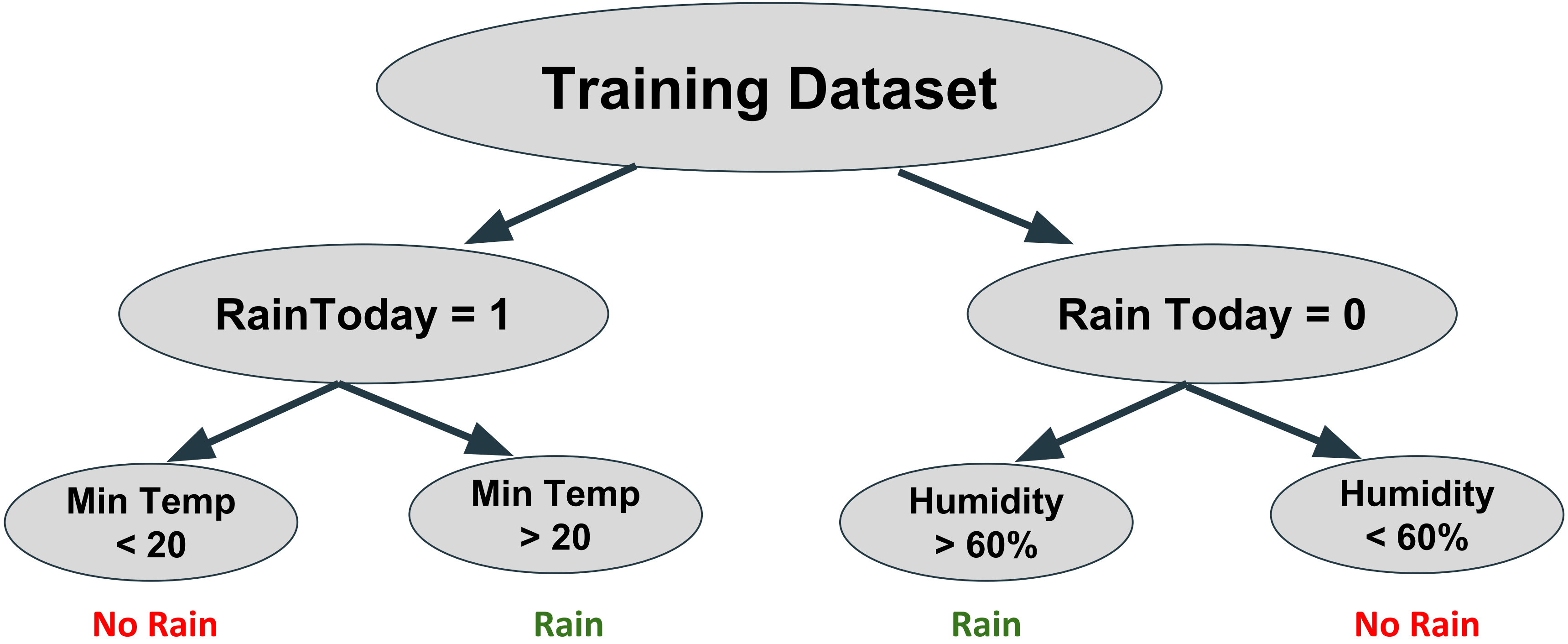
Step 3: Pick a model and hyperparameters

- As we found out earlier, a model is really just a **function** from input (X) to output (Y)
- Each model has a set of parameters that it “learns” when it is trained:
  - Linear regression: the **weights** applied to each feature
  - Decision tree: the **features** and **thresholds** used to split up your data
- **Hyperparameters:** set by the user (with default values). These hyperparameters will impact the training process and performance, but they are not “learned” by the model.

Common Model Names to Know:

- Linear Regression (continuous)
- Logistic Regression (binary category)
- Decision Tree [Classifier/Regressor] (categorical/continuous)
- Random Forest [Classifier/Regressor] (categorical/continuous)
- Support Vector Machine (categorical/continuous)
- Neural Network (categorical/continuous)

Decision Tree



Grid Search

- Method for determining the best combination of hyperparameters for a model
- For each hyperparameter, define multiple values to use
- For each combination of hyperparameters, train and evaluate your model

	p1		
p2	(1, 1e3)	(10, 1e3)	(100, 1e3)
	(1, 1e4)	(10, 1e4)	(100, 1e4)
	(1, 1e5)	(10, 1e5)	(100, 1e5)

Step 4: Train your model

- Using your **training data set** and outcome labels, “fit” your model
- The model will use the specified hyperparameters, and “learn” the best function to map from input (X) data to output labels (Y)
- Each model usually has its own training process, and function that it uses to optimize (some quantitative way to say how *good* the predictions are)

Cross Validation

- Train your model multiple times, on different subsets of data, and calculate its performance each time
- Get a mean and standard deviation of performance (rather than just a single number)
- This allows you to see how variable your model's performance is based on the data that is used to train it

Step 5: Evaluate your model

- Once you’ve trained a model, you want to know how predictive it is
- Need some sort of quantitative measure to compare performance across models
- If you decide to swap out models, edit features, or tweak hyperparameters, you want to know how your performance changes
- If you are going to use your model in “production” you want to know how well it should do

Common Metrics (binary classification)

- **Accuracy**: percentage of correct predictions
  - (TP + TN) / (P + N)
- **Precision**: percentage of positive class predictions that are correct
  - TP / (TP + FP)
- **Recall**: percentage of positive labels that are predicted as positive
  - TP / (TP + FN)

Actual Label

Predicted Label		+	-
	+	TP	FP
	-	FN	TN
		P	N

Next Steps

- Read through these slides again
- Watch some videos on the different types of ML models
- Try this workshop again with another dataset!
- Read through the pandas and sklearn documentation