Chapter 2: End-to-End Machine Learning Project

- Look at the big picture
    - Frame the problem: supervised or not, classification or regression (estimates off by more than 20%)
    - Select a performance measure:
        - RMSE, *L2*, Euclidean norm
        - MAE, *L1*, Manhattan norm
        - Norm index: higher focuses more on large values
            - RMSE more sensitive to outliers than MAE
    - Check the assumptions
- Get the data
    - Create the workspace
    - Download the data
        - Automating this can be helpful if need to re-run or run on multiple machines
    - Take a quick look at the structure
        - Missing data?
        - Data type?
        - df['col_x'].value_counts(), df.describe()
    - Create a test set: prevent data snooping
        - Stratified sampling: same proportions in sample as in population
- Discover and visualize data to gain insights
    - Visualize labels
    - Look for correlations: *useful correlation examples*
    - Experiment with attribute combinations
- Prepare the data for machine learning algorithms
    - General
        - Will want to write functions for data transformations so that you will build up a library of functions to deploy on new data
    - Data cleaning
        - Deal with missing features
    - Handling text and categorical attributes
        - One-hot encoding
        - Problems when attribute has many possible categories
    - Custom transformers
    - Feature scaling
        - Min-max scaling (aka normalization): bounded [0,1]
        - Standardization: unbounded → issue for some neural networks
    - Transformation pipelines
        - Pipeline: syntax
        - Column transformer
- Select and train model
    - Training and evaluating Training set
    - Better evaluation using cross-validation

    - Cross validation can help tell whether model is over/underfitting data before look at test set
    - Fixes to poor fit
        - Underfitting
            - More powerful model
            - Better features
            - Reduce constraints (only possible if already regularized)
        - Overfitting
            - Simplify model
            - Add constraints (e.g., regularization)
            - More data
- Fine-tune your model
    - General: efficient ways to play with hyperparameters
    - Grid search
    - Randomized search
    - Ensemble methods
    - Analyze best models and errors
    - Evaluate your system on test set
- Present your solution
    - What you learned, what worked, what didn't, assumptions, limitations
    - Even model doesn't perform much better, may be better to deploy to free up traditional forecasters' time
- Launch, monitor, and maintain your system
    - Ways to deploy
        - Save with joblib
        - Dedicated web service
        - Deploy on Google Cloud AI platform
    - Monitor for decay
        - Data and information change over time
        - Cats and dogs don't change, but images of them do!
        - Ways to automate monitoring
            - Collect fresh data regularly and label
            - Write script to train model and fine-tune hyperparameters at a regular frequency
            - Write script to evaluate new and old model on updated test set
        - May need to monitor new input data for quality6