# Chapter 2: End-to-End Machine Learning Project

Pull out your Machine Learning project checklist:

## 1 - Look at the Big Picture

### Frame the Problem

The first question to ask your boss is what precisely the business objective is. Building a model is probably not the end goal. Knowing the purpose is important because it will determine how you frame the problem, which algorithms you will select, which performance measure you will use to evaluate your model, and how much effort you will spend tweaking it.

Your boss answers that your model's output will be fed to another Machine Learning system along with many other signals. This downstream system will determine whether it is worth investing in a given area or not. Getting this right is critical, as it directly affects revenue.

The next question to ask your boss is what the current solution looks like. The current situation will often give you a reference for performance, as well as insights on how to solve the problem.

With all this information, you are now ready to start designing your system. First, determine what kind of training supervision the model will need.

### Select a Performance Measure

Your next step is to select a performance measure.

### Check the Assumptions

Lastly, it is good practice to list and verify the assumptions that have been made so far (by you or others); this can help you catch serious issues early on.

Great! You're all set, the lights are green, and you can start coding now!

# 2 - Get the Data

It’s time to get your hands dirty. Don’t hesitate to pick up your laptop and walk through the code examples.

## Download the Data

In typical environments, your data would be available in a relational database or some other common data store and spread across multiple tables/documents/files. To access it, you would first need to get your credentials
and access authorizations and familiarize yourself with the data schema.

Rather than manually downloading and decompressing the data, it’s usually preferable to write a function that does it for you. This is useful in particular if the data changes regularly: you can write a small script that uses the function to fetch the latest data (or you can set up a scheduled job to do that automatically at regular intervals). Automating the process of fetching the data is also useful if you need to install the dataset on multiple machines.

## Take a Quick Look at the Data Structure

to have a better understanding of the kind of data you are dealing with.

## Create a Test Set

It may sound strange to voluntarily set aside part of the data at this stage. After all, you have only taken a glance at the data, and surely you should learn a whole lot more about it before you decide what algorithms to use.

Creating a test set is theoretically simple: pick some instances randomly, typically 20% of the dataset (or less if your dataset is very large), and set them aside.

# 3 - Discover and Visualize the Data to Gain Insights

So far you have only taken a glance at the data to get a general understanding of the kind of data you are manipulating. Now the goal is to go into a little more depth.

First, make sure you have put the test set aside and you are only exploring the training set. Also, if the training set is very large, you may want to sample an exploration set, to make manipulations easy and fast during the exploration phase.

## Visualizing Geographical Data

Our brains are very good at spotting patterns in pictures.

## Looking for Correlations

The correlation coefficient ranges from –1 to 1. When it is close to 1, it means that there is a strong positive correlation. When the coefficient is close to –1, it means that there is a strong negative correlation.

### WARNING:

The correlation coefficient only measures linear correlations. It may completely miss out on nonlinear relationships.

## Experimenting with Attribute Combinations

One last thing you may want to do before preparing the data for the Machine Learning algorithms is to try out various attribute combinations.

This round of exploration does not have to be thorough; the point is to start on the right foot and quickly gain insights that will help you get a first reasonably good prototype. But this is an iterative process: once you get a prototype up and running, you can analyze its output to gain more insights and come back to this exploration step.

# 3 - Prepare the Data for Machine Learning Algorithms

It’s time to prepare the data for your Machine Learning algorithms. Instead of doing this manually, you should write functions for this purpose.

But first, let’s revert to a clean training set. Let’s also separate the predictors and the labels, since we don’t necessarily want to apply the same transformations to the predictors and the target values.

## Data Cleaning

## Handling Text and Categorical Attributes

## Feature Scaling and Transformation

### Custom Transformers

## Transformation Pipelines

# 4 - Select and Train a Model

At last! You framed the problem, you got the data and explored it, you sampled a training set and a test set, and you wrote a preprocessing pipeline to automatically clean up and prepare your data for Machine Learning algorithms. You are now ready to select and train a Machine Learning model.

## Training and Evaluating the Training Set

## Better Evaluation Using Cross-Validation

# 5 - Fine-Tune Your Model

## Grid Search

## Randomized Search

## Ensemble Methods

## Analyze the Best Models and Their Errors

## Evaluate Your System on the Test Set