

CHAPTER 2

End-to-End Machine Learning Project

In this chapter, you will go through an example project end to end, pretending to be a recently hired data scientist in a real estate company.¹ Here are the main steps you will go through:

1. Look at the big picture.
2. Get the data.
3. Discover and visualize the data to gain insights.
4. Prepare the data for Machine Learning algorithms.
5. Select a model and train it.
6. Fine-tune your model.
7. Present your solution.
8. Launch, monitor, and maintain your system.

Working with Real Data

When you are learning about Machine Learning it is best to actually experiment with real-world data, not just artificial datasets. Fortunately, there are thousands of open datasets to choose from, ranging across all sorts of domains. Here are a few places you can look to get data:

- Popular open data repositories:

¹ The example project is completely fictitious; the goal is just to illustrate the main steps of a Machine Learning project, not to learn anything about the real estate business.

Download from finelybook www.finelybook.com

- [UC Irvine Machine Learning Repository](#)
- [Kaggle datasets](#)
- [Amazon's AWS datasets](#)
- Meta portals (they list open data repositories):
 - <http://dataportals.org/>
 - <http://opendatamonitor.eu/>
 - <http://quandl.com/>
- Other pages listing many popular open data repositories:
 - [Wikipedia's list of Machine Learning datasets](#)
 - [Quora.com question](#)
 - [Datasets subreddit](#)

In this chapter we chose the California Housing Prices dataset from the StatLib repository² (see [Figure 2-1](#)). This dataset was based on data from the 1990 California census. It is not exactly recent (you could still afford a nice house in the Bay Area at the time), but it has many qualities for learning, so we will pretend it is recent data. We also added a categorical attribute and removed a few features for teaching purposes.

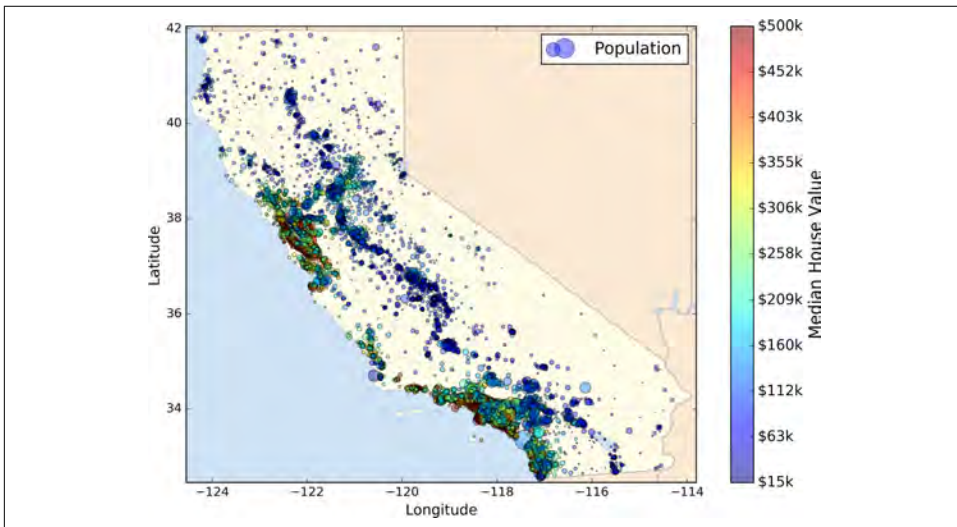


Figure 2-1. California housing prices

² The original dataset appeared in R. Kelley Pace and Ronald Barry, “Sparse Spatial Autoregressions,” *Statistics & Probability Letters* 33, no. 3 (1997): 291–297.

Look at the Big Picture

Welcome to Machine Learning Housing Corporation! The first task you are asked to perform is to build a model of housing prices in California using the California census data. This data has metrics such as the population, median income, median housing price, and so on for each block group in California. Block groups are the smallest geographical unit for which the US Census Bureau publishes sample data (a block group typically has a population of 600 to 3,000 people). We will just call them “districts” for short.

Your model should learn from this data and be able to predict the median housing price in any district, given all the other metrics.



Since you are a well-organized data scientist, the first thing you do is to pull out your Machine Learning project checklist. You can start with the one in [Appendix B](#); it should work reasonably well for most Machine Learning projects but make sure to adapt it to your needs. In this chapter we will go through many checklist items, but we will also skip a few, either because they are self-explanatory or because they will be discussed in later chapters.

Frame the Problem

The first question to ask your boss is what exactly is the business objective; building a model is probably not the end goal. How does the company expect to use and benefit from this model? This is important because it will determine how you frame the problem, what algorithms you will select, what performance measure you will use to evaluate your model, and how much effort you should spend tweaking it.

Your boss answers that your model’s output (a prediction of a district’s median housing price) will be fed to another Machine Learning system (see [Figure 2-2](#)), along with many other *signals*.³ 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.

³ A piece of information fed to a Machine Learning system is often called a *signal* in reference to Shannon’s information theory: you want a high signal/noise ratio.