ML - Session 3
E2E ML

Main steps you will go through:

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

Working with Real Data

- Work with real-world data, not just artificial datasets!

- Popular open data repositories:
  - UC Irvine Machine Learning Repository
  - Kaggle
  - Amazon Datasets

Working with Real Data

- Open datasets:
    - http://dataportals.org/
    - http://opendatamonitor.eu/
    - http://quandl.com/

Example problem

- A dataset based on data from the 1990 California census.
- Categorical attributes
- Few features removed for teaching purposes.

Taken from: Hands on Machine Learning 2E

Goal

- To build a model of housing prices in California using the California census data.

Frame the Problem

- What exactly is the business objective? Building a model is rarely the ultimate goal.
- What is the benefit?

Human problem

Your system will be fed to another ML system that will determine whether it is worth investing in a given area or not.

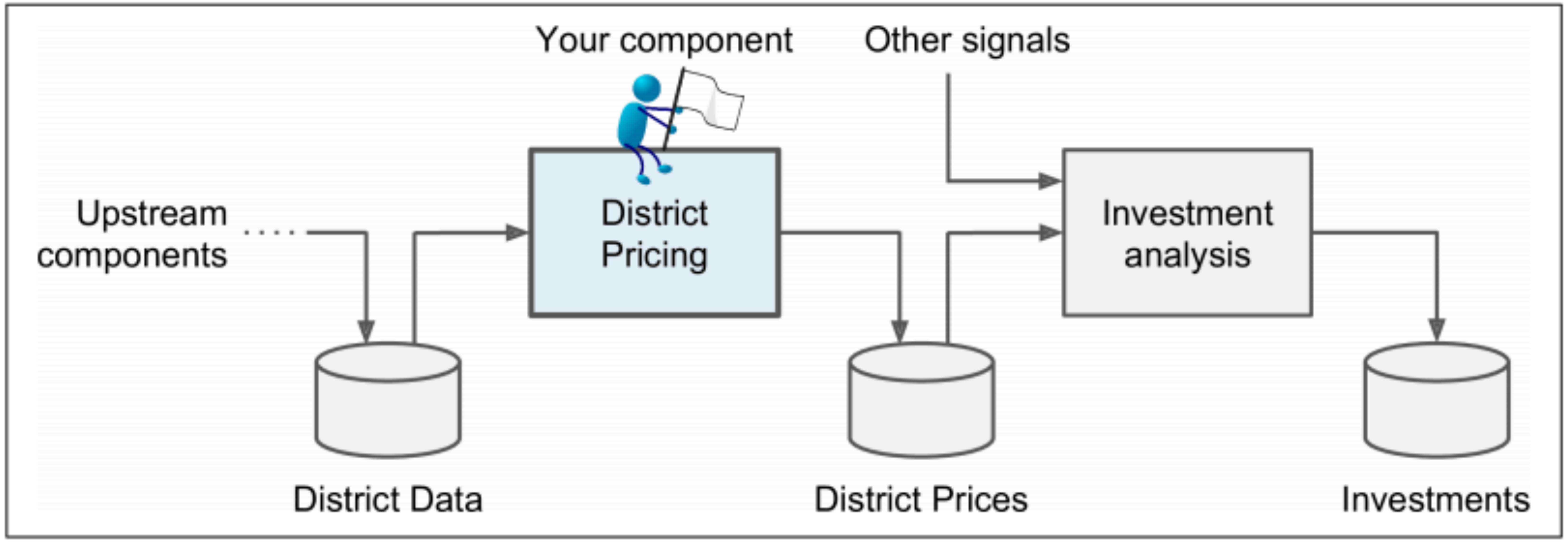


Figure 2-2. A Machine Learning pipeline for real estate investments

Current solution

What does the current solution look like (if any)?

- Reference performance.
- Insights on how to solve the problem.

Let's start

Frame the problem:

Is it supervised, unsupervised, or Reinforcement Learning?

Is it a classification task, a regression task, or something else?

Should you use batch learning or online learning techniques?

Note

Don’t follow the concept of univariate or multivariate provided by ‘Hands-on Machine Learning’

Univariate: Only one input variable is used to estimate the target.

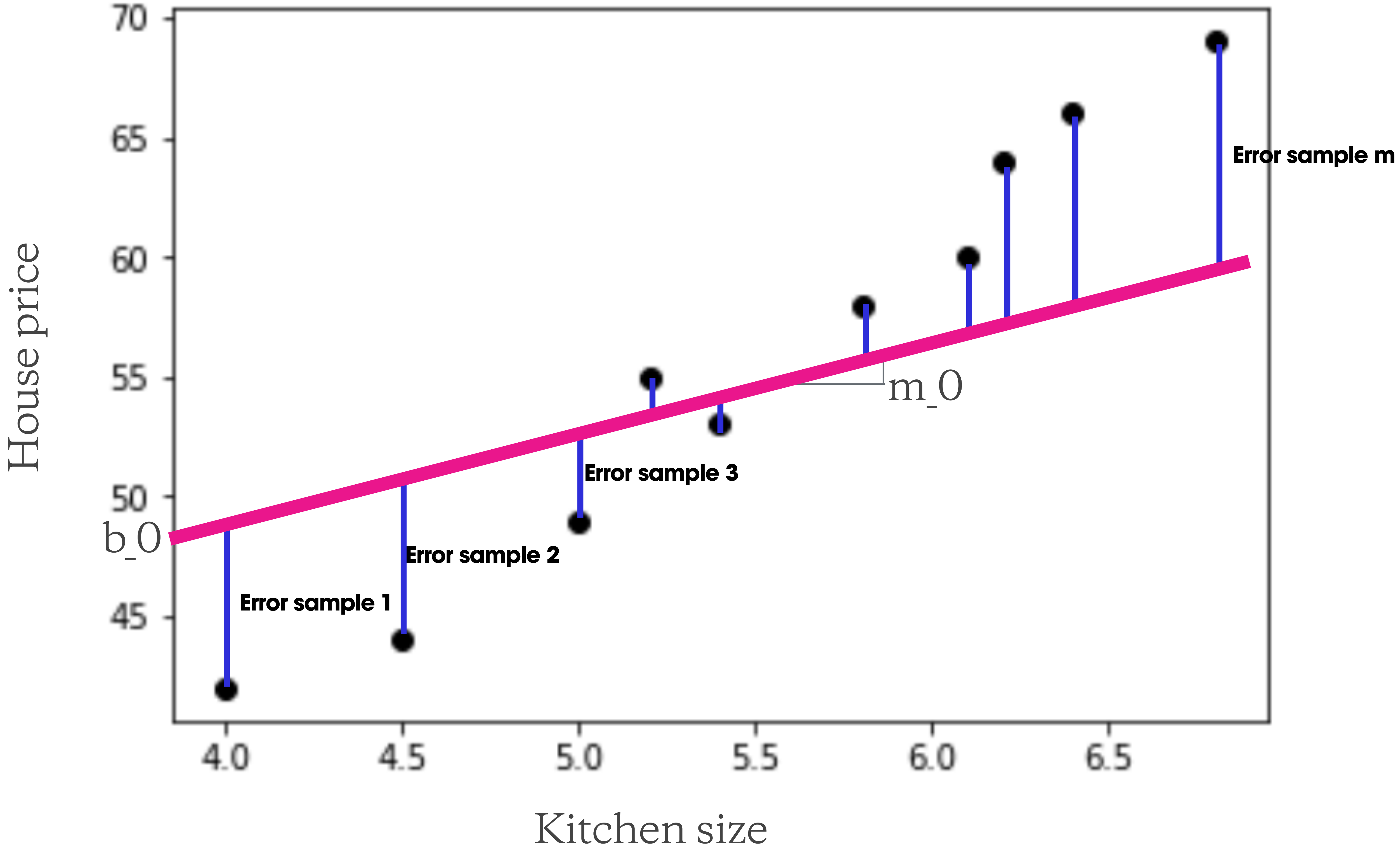
Multivariate: There are more than one input variables used to estimate the target.

Select a Performance Measure

A typical performance measure for regression problems is the Root Mean Square Error (RMSE). It gives an idea of how much error the system typically makes in its predictions, with a higher weight for large errors.

$$\mathrm{RMSE}(\mathbf{X}, h) = \sqrt{\frac{1}{m} \sum_{i=1}^{m} \left(h\left(\mathbf{x}^{(i)}\right) - y^{(i)}\right)^2}$$

Select a Performance Measure



$$\mathrm{RMSE}(\mathbf{X}, h) = \sqrt{\frac{1}{m} \sum_{i=1}^{m} \left(h\left(\mathbf{x}^{(i)}\right) - y^{(i)} \right)^2}$$

Output = m_0 * kitchenSize + b_0

Select a Performance Measure

In some contexts you may prefer to use another function.

$$\mathrm{MAE}(\mathbf{X}, h) = \frac{1}{m} \sum_{i=1}^{m} \left| h\left(\mathbf{x}^{(i)}\right) - y^{(i)} \right|$$

Both the RMSE and the MAE are ways to measure the distance between two vectors: the vector of predictions and the vector of target values.

Select a Performance Measure

Computing the root of a sum of squares (RMSE) corresponds to the Euclidean norm: it is the notion of distance you are familiar with. It is also called the $\ell 2$ norm

Computing the sum of absolutes (MAE) corresponds to the $\ell 1$ norm. It is sometimes called the Manhattan norm because it measures the distance between two points in a city if you can only travel along orthogonal city blocks.

Check the Assumptions

What if the downstream system actually converts the prices into categories (e.g., “cheap,” “medium,” or “expensive”) and then uses those categories instead of the prices themselves?

In this case, getting the price perfectly right is not important at all; your system just needs to get the category right.

Get the Data

Create the Workspace.

Verify python and pip version

```
python3 -m pip --version
```

Creating an Isolated Environment (strongly recommended)

- Install virtualenv

Isolated Environment

Virtualenv installation:

```
python3 -m pip install --user -U virtualenv
```

Create environment:

```
python -m virtualenv -p python3 $ENV_PATH
```

Isolated Environment

Activate your environment

```
source $ENV_PATH/bin/activate
```

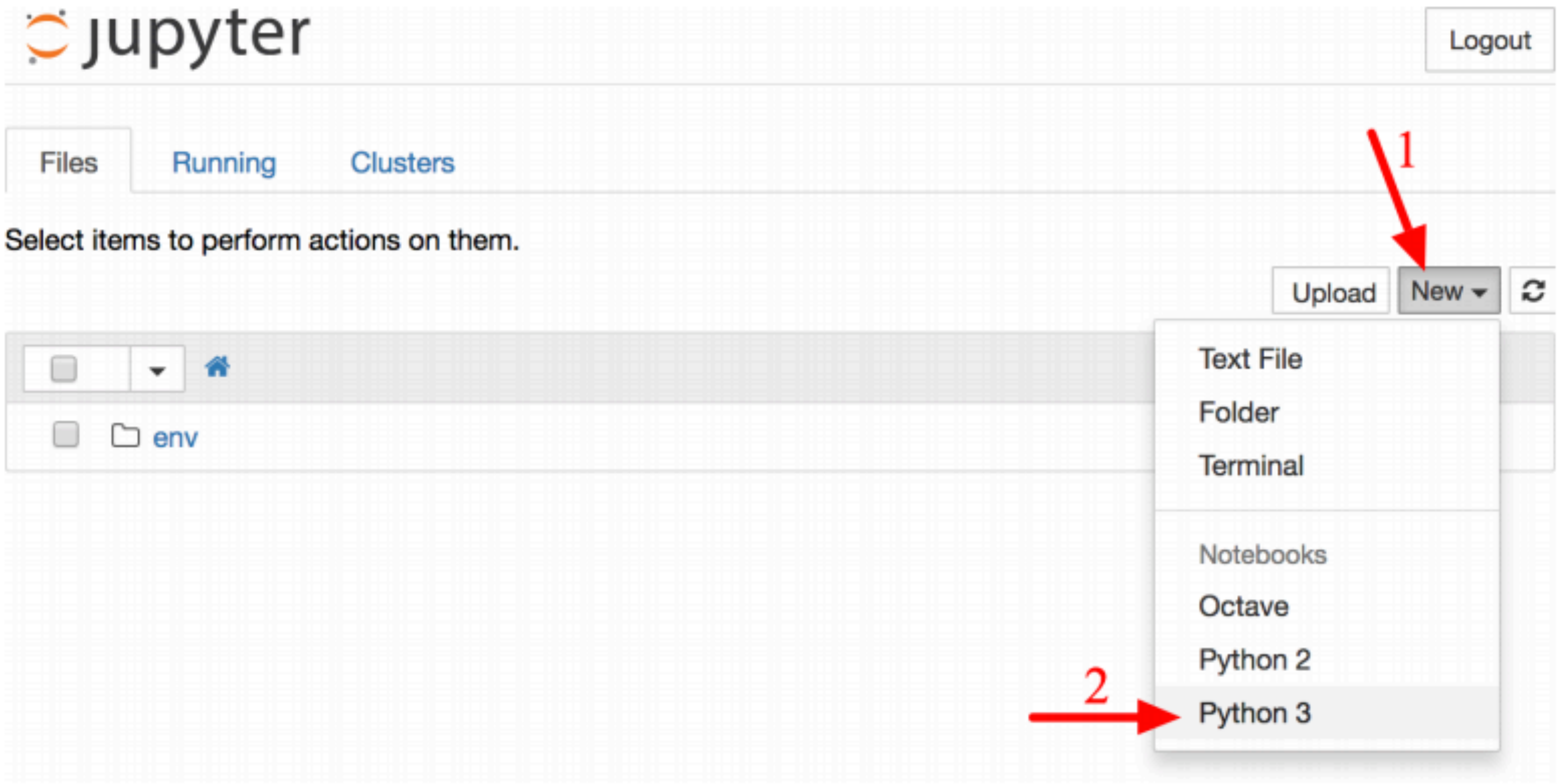
Now you can install all the required modules and their dependencies

```
pip install jupyter matplotlib numpy pandas scipy
scikit-learn
```

Isolated Environment

Open Jupiter Notebook:

```
jupyter notebook
```



Select items to perform actions on them.

Download the Data

Link:

https://github.com/ageron/handson-ml2

Datasets -> Housing -> Housing.tgz