Hands on Machine Learning

Chater 2

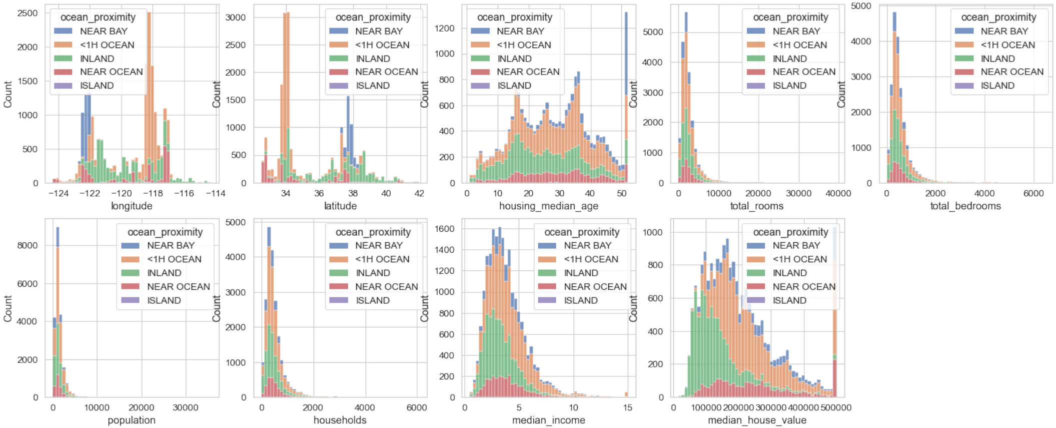
Benedikt Zönnchen

February 22, 2022

Problem

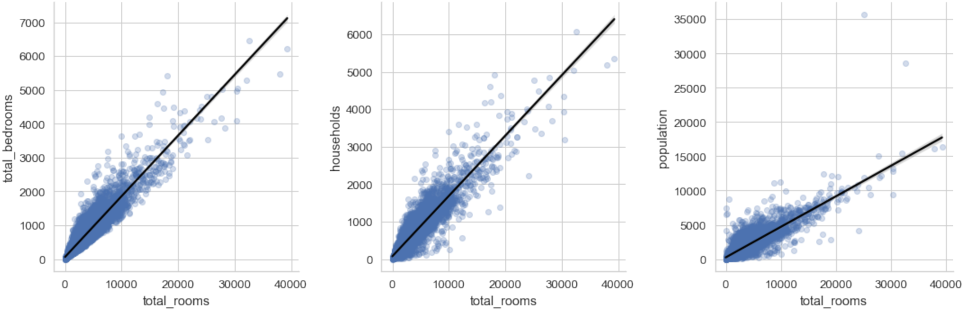
Given 8 attributes (longitude, latitude, house age, rooms, bedrooms, population, households, income), we want to predict the (mean) *house value*.

Data inspection



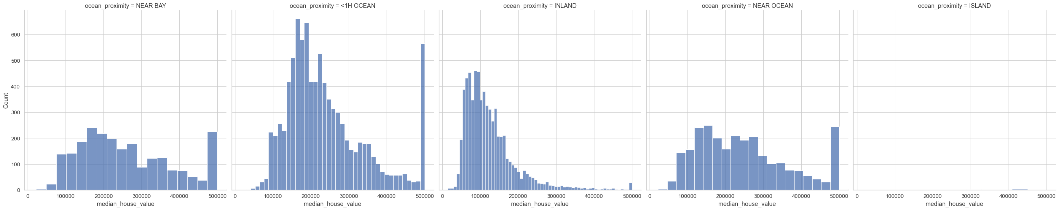
Data inspection

Strong correlation between *rooms*, *bedrooms* *households* and *population*:



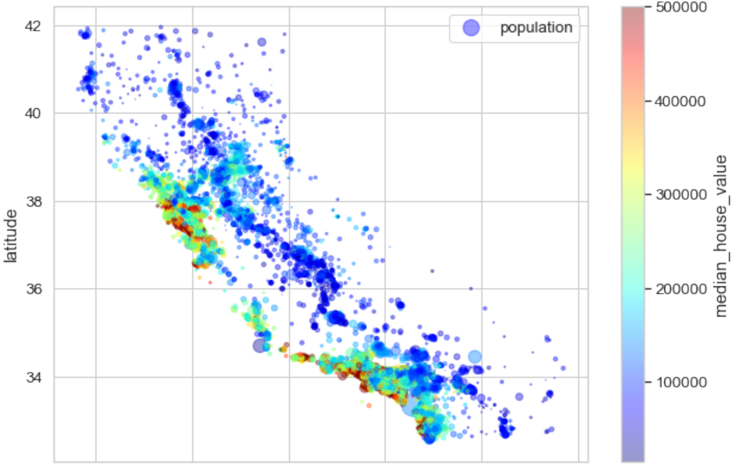
Data inspection

How does the *house value* change with *ocean proximity* (discrete category)?



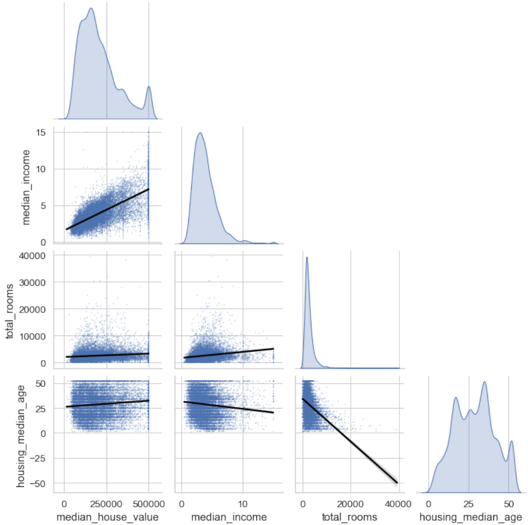
- *near bay* and *near ocean* look similar
- *inland* house values are more lean towards the lower end
- there are some very expensive houses hidden in the data

Data inspection



Data inspection

- *income* and *house value* are correlated
- everything else is not really correlated with the *house value*



Machine learning pipeline

Preparation:

1. Preparation
2. download the data
3. load the data
4. inspect the data
5. add *income category* to split the data effectively
6. split the data into *training set* and *test set* (0.8/0.2)
7. split the *training set* into *label* and *remaining data*
8. **data cleaning**: deal with missing data (fill, drop row, or drop attribute)
9. convert categories into numbers
10. add combined attributes
11. **feature scaling**: *normalization* or *standardization*

Machine learning pipeline

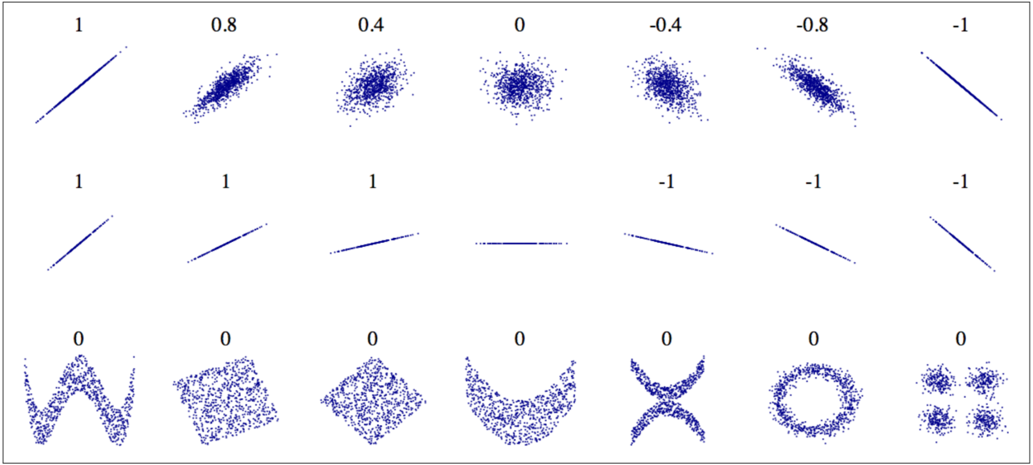
Model creation:

1. select and train a (or multiple) model(s) (**use cross-validation**)
2. select your shortlist of promising models
3. fine-tune your model:
   - hyperparameter optimization: *random walk*, *grid search* (**use cross-validation**)
   - ensemble methods: combine your best models
   - feature manipulation: drop non-influential features
4. evaluate your result on the *test set*

Build your system

Pitfalls/remarks

Correlation captures only **linear** relations:



Pitfalls/remarks

Be careful if you use your *test set*:

- if you have not a lot of data, group your data before splitting
- use cross-validation
- adapting the model after evaluation (using the *test set*) leads to *overfitting*

Terms

Random variable (informal)

A random variable is a measurable **function** from a probability space (set of possible outcomes) Ω into a measurement space E.

The probability $\mathbf{P}$ that the random variable X takes on a value in $S \subseteq E$ is noted as

$$\mathbf{P}(X \in S) = \mathbf{P}(\{\omega \in \Omega : X(\omega) \in S\})$$

The probability $\mathbf{P}$ that the random variable X takes on a value in $x \in E$ is noted as

$$\mathbf{P}(X = x) = \mathbf{P}(\{\omega \in \Omega : X(\omega) = x\})$$

Terms

Expected value (finite)

Let X be a *random variables* with a **finite** list of values $x_1, \dots, x_k$ than the *expected value* μ_X of X is defined by

$$\mu_X = \mathbf{E}[X] = \sum_{i=1}^{k} x_i \cdot \mathbf{P}(X = x_i) \tag{1}$$

The expected value is a weighted average of the x_i values.

Terms

Standard deviation

Let X be a *random variables* than the *standard deviation* σ_X of X is defined by

$$\sigma_X = \sqrt{\mathbf{E}[(X - \mu_X)^2]} = \sqrt{\mathbf{E}[(X - \mathbf{E}[X])^2]} \tag{2}$$

The standard deviation is the square root of the *variance*.

Terms

Correlation

Let X, Y be two random variables with *expected values* μ_X, μ_Y and *standard deviations* σ_X, σ_Y than the correlation coefficient is defined by

$$\rho_{X,Y} = \mathrm{corr}(X, Y) = \frac{\mathrm{cov}(X, Y)}{\sigma_X \cdot \sigma_Y} = \frac{\mathbf{E}[(X - \mu_X) \cdot (Y - \mu_Y)]}{\sigma_X \cdot \sigma_Y}. \tag{3}$$

where $\mathbf{E}[X]$ is the expected value of X, i.e., $\mu_X = \mathbf{E}[X]$ and $\mu_Y = \mathbf{E}[Y]$.

Terms

Cross-validation

Let T be our data points with $|T| = N$. Let $k < N$ than we k subsets:

$$T_1, \dots, T_k \subset T$$

Than we *train* and *validate* k models by using

$$\{T_1, \dots, T_k\} \setminus \{T_i\}$$

as *training set* and

$$T_i$$

as *test set* for $i \in \{1, \dots, k\}$.

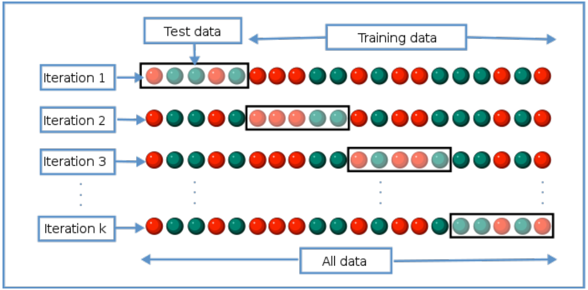
Terms

k-fold cross-validation

Use a partition, that is,

$$\bigcup_{i=1}^{k} T_i = T \text{ and } i \neq j \Rightarrow T_i \cap T_j = \emptyset$$

holds.



Terms

Normalization (scaling)

Shift and rescale (**linear transformation**) values such that they all lie in $[0;1]$:

$$f(x) = \frac{x - x_{\text{min}}}{x_{\text{max}} - x_{\text{min}}} \tag{4}$$

Many algorithms expect *normalized* values.

Terms

Standardization (scaling)

Shift by the mean value and scale by the standard deviation (**linear transformation**):

$$y = f(x) = \frac{x - \mu_X}{\sigma_X} \tag{5}$$

The resulting distribution has unit variance and is centered around zero. The scaling is less affected by *outliers*.