Week 2: Regression w/ multiple input variables

MULTIPLE LINEAR REGRESSION

Probably the most widely used ML alg.

For house price prediction, consider multiple input variables as a vector $\vec{x}$:

$$\begin{bmatrix} x_1 \\ x_2 \\ \dots \\ x_n \end{bmatrix} \begin{matrix} = \text{sq. ft.} \\ = \text{bedrooms} \\ \\ = \text{house age} \end{matrix}$$

with, n features

Model: $\boxed{f_{\vec{w},b}(\vec{x}) = \vec{w} \cdot \vec{x} + b}$

where $\vec{w} = [w_1, \dots, w_n]$ and $b \in \mathbb{R}$ are parameters

Note: in NumPy with arrays x, w you can just do {np.dot(w,x)} which may use hardware vectorization. Works w/ many ops!

↳ So in grad. descent you can write

```
f = np.dot(w,x) + b
w = w - 0.1 * d
```

} both vectorized!

np might do all this vectorized in a single step

Gradient descent for multiple lin. reg.

Params: $\vec{w} = [w_1, \dots, w_n]$

Model: $f_{\vec{w},b}(\vec{x}) = \vec{w} \cdot \vec{x} + b$

Cost fn: $J(\vec{w}, b)$

Grad Descent: Repeatedly:

for $j = 1$ to n: $w_j = w_j - \alpha \frac{1}{m} \sum_{i}^{m} (f(\vec{x}_i) - y_i) x_{ij}$

and: $b = b - \alpha \frac{1}{m} \sum_{i}^{m} (f(x_i) - y_i)$

until convergence.