Using Pandas to Get Familiar With Your Data

Her makine öğrenimi projesinin ilk adımı, verilerle aşina olmaktır. Bunun için **Pandas** kütüphanesini kullanacaksınız. Pandas, veri bilimcilerin verileri keşfetmek ve üzerinde oynamak için kullandığı birincil araçtır. Çoğu kişi kodlarında pandas'ı kısaca **pd** olarak kullanır. Bunu şu komutla yaparız:

import pandas as pd

**Pandas** kütüphanesinin en önemli bölümü **DataFrame**'dir. Bir DataFrame, **tablo** olarak düşünebileceğiniz türden verileri tutar. Bu, Excel'deki bir sayfaya veya bir SQL veritabanındaki bir tabloya benzer.

Pandas, bu tür verilerle yapmak isteyeceğiniz çoğu şey için güçlü yöntemlere sahiptir.

Örnek olarak, **Avustralya'nın Melbourne şehrindeki ev fiyatları hakkındaki verilere** bakacağız. Uygulamalı alıştırmalarda, aynı süreçleri, Iowa'daki ev fiyatlarını içeren yeni bir veri setine uygulayacaksınız.

Örnek (Melbourne) verileri ../input/melbourne-housing-snapshot/melb\_data.csv dosya yolunda bulunur.

Verileri aşağıdaki komutlarla yükler ve keşfederiz:

*# save filepath to variable for easier access*

melbourne\_file\_path = '../input/melbourne-housing-snapshot/melb\_data.csv'

*# read the data and store data in DataFrame titled melbourne\_data*

melbourne\_data = pd.read\_csv(melbourne\_file\_path)

*# print a summary of the data in Melbourne data*

melbourne\_data.describe()

| Rooms | Price | Distance | Postcode | Bedroom2 | Bathroom | Car | Landsize | BuildingArea | YearBuilt | Lattitude | Longtitude | Propertycount |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| count | 13580.000000 | 1.358000e+04 | 13580.000000 | 13580.000000 | 13580.000000 | 13580.000000 | 13518.000000 | 13580.000000 | 7130.000000 | 8205.000000 | 13580.000000 | 13580.000000 | 13580.000000 |
| mean | 2.937997 | 1.075684e+06 | 10.137776 | 3105.301915 | 2.914728 | 1.534242 | 1.610075 | 558.416127 | 151.967650 | 1964.684217 | -37.809203 | 144.995216 | 7454.417378 |
| std | 0.955748 | 6.393107e+05 | 5.868725 | 90.676964 | 0.965921 | 0.691712 | 0.962634 | 3990.669241 | 541.014538 | 37.273762 | 0.079260 | 0.103916 | 4378.581772 |
| min | 1.000000 | 8.500000e+04 | 0.000000 | 3000.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 1196.000000 | -38.182550 | 144.431810 | 249.000000 |
| 25% | 2.000000 | 6.500000e+05 | 6.100000 | 3044.000000 | 2.000000 | 1.000000 | 1.000000 | 177.000000 | 93.000000 | 1940.000000 | -37.856822 | 144.929600 | 4380.000000 |
| 50% | 3.000000 | 9.030000e+05 | 9.200000 | 3084.000000 | 3.000000 | 1.000000 | 2.000000 | 440.000000 | 126.000000 | 1970.000000 | -37.802355 | 145.000100 | 6555.000000 |
| 75% | 3.000000 | 1.330000e+06 | 13.000000 | 3148.000000 | 3.000000 | 2.000000 | 2.000000 | 651.000000 | 174.000000 | 1999.000000 | -37.756400 | 145.058305 | 10331.000000 |
| max | 10.000000 | 9.000000e+06 | 48.100000 | 3977.000000 | 20.000000 | 8.000000 | 10.000000 | 433014.000000 | 44515.000000 | 2018.000000 | -37.408530 | 145.526350 | 21650.000000 |

# Interpreting Data Description

Sonuçlar, orijinal veri setinizdeki her sütun için 8 sayı gösterir.

* **İlk sayı, "count" (sayım)**, eksik olmayan değere sahip kaç satır olduğunu gösterir. Eksik değerler birçok nedenden dolayı ortaya çıkar. Örneğin, 1 yatak odalı bir ev incelenirken 2. yatak odasının boyutu toplanmaz. Eksik veriler konusuna daha sonra geri döneceğiz.
* **İkinci değer "mean" (ortalama)**, yani aritmetik ortalamadır.
* Onun altında **"std" (standart sapma)**, değerlerin sayısal olarak ne kadar dağınık olduğunu ölçer.
* **"min" (minimum), "25%", "50%", "75%" ve "max" (maksimum) değerlerini yorumlamak için**, her sütunu en düşükten en yükseğe doğru sıraladığınızı hayal edin.
  + İlk (en küçük) değer **minimumdur**.
  + Listenin dörtte birine kadar gittiğinizde, değerlerin %25'inden büyük ve %75'inden küçük olan bir sayı bulursunuz. Bu, **%25 değeridir** ("25. yüzdelik dilim" olarak telaffuz edilir).
  + %50 ve %75'lik yüzdelik dilimler benzer şekilde tanımlanır.
  + **Maksimum** ise en büyük sayıdır.