Python Data Products
Course 2: Design thinking and predictive pipelines

Lecture: Regression in Python
In this lecture we will...
• Explore how to express linear regression equations in terms of Python data structures
• Work through a (simple) real-world regression example
• Compare a "manual" implementation of linear regression to a library function
Example – Air quality prediction

We'll look at the problem of predicting **air quality**, using an index called pm2.5, measured in Beijing.

- This is a "simpler" dataset than some of the others we've been working with, as the relevant features are all real-valued.
- It's also useful in our following lecture (on time-series prediction), since the data is in the form of a time series.
Example – UCI Dataset Repository

Beijing PM2.5 Data Data Set

$\textbf{Download}$ $\textbf{Data Folder}$ $\textbf{Data Set Description}$

**Abstract:** This hourly data set contains the PM2.5 data of US Embassy in Beijing. Meanwhile, meteorological data from Beijing Capital International Airport are also included.

<table>
<thead>
<tr>
<th>Data Set Characteristics:</th>
<th>Multivariate, Time-Series</th>
<th>Number of Instances:</th>
<th>43824</th>
<th>Area:</th>
<th>Physical</th>
</tr>
</thead>
<tbody>
<tr>
<td>Attribute Characteristics:</td>
<td>Integer, Real</td>
<td>Number of Attributes:</td>
<td>13</td>
<td>Date Donated</td>
<td>2017-01-19</td>
</tr>
<tr>
<td>Associated Tasks:</td>
<td>Regression</td>
<td>Missing Values?</td>
<td>Yes</td>
<td>Number of Web Hits:</td>
<td>70637</td>
</tr>
</tbody>
</table>

**Source:**
Song Xi Chen, sax@pam.pku.edu.cn, Guanghua School of Management, Center for Statistical Science, Peking University.

**Data Set Information:**
The data’s time period is between Jan 1st, 2010 to Dec 31st, 2014. Missing data are denoted as `?`.

[https://archive.ics.uci.edu/ml/datasets/Beijing+PM2.5+Data](https://archive.ics.uci.edu/ml/datasets/Beijing+PM2.5+Data)
What are we trying to predict?

E.g. pm2.5 vs. Temperature:

\[ \text{pm2.5} = \theta_0 + \theta_1 \times \text{temp} \]
Code: Reading the file

```python
In [2]:
  :path = "datasets/PRSA_data_2010.1.1-2014.12.31.csv"
   f = open(path, 'r')

In [3]:
   dataset = []
   header = f.readline().strip().split(',')
   for line in f:
       line = line.split(',')
       dataset.append(line)

In [4]:
   N = len(dataset)

Out[4]: 43824

In [5]:
   header

Out[5]: ['No', 'year', 'month', 'day', 'hour', 'pm2.5', 'DEWP', 'TEMP', 'PRES', 'cbwd', 'lvs',
         'Label that we want to predict', 'Feature we want to use for prediction']
```
Code: Extracting features and labels

```python
In [6]: header.index('pm2.5')
Out[6]: 5

In [7]: header.index('TEMP')
Out[7]: 7

In [8]: y = [float(d[5]) for d in dataset]

---------------------------------------------------------------------------
ValueError                       Traceback (most recent call last)
<ipython-input-8-caefd9ca5537> in <module>()
----> 1 y = [float(d[5]) for d in dataset]

<ipython-input-8-caefd9ca5537> in <listcomp>(.0)
----> 1 y = [float(d[5]) for d in dataset]

ValueError: could not convert string to float: 'NA'

In [9]: dataset = [d for d in dataset if d[5] != 'NA']
```
Code: Let's try again...

In [10]: y = [float(d[5]) for d in dataset]

In [11]: def feature(datum):
   ...:     feat = [1, float(datum[7])]
   ...:     return feat

In [12]: X = [feature(d) for d in dataset]

In [13]: y[:10]

Out[13]: [129.0, 148.0, 159.0, 181.0, 138.0, 109.0, 105.0, 124.0, 120.0, 132.0]

In [14]: X[:10]

Out[14]: [[1, -4.0],
         [1, -4.0],
         [1, -5.0],
         [1, -5.0],
         [1, -5.0],
         [1, -6.0],
         [1, -6.0],
         [1, -5.0],
         [1, -5.0],
         [1, -5.0]]
Reminder: Constant feature

Why did we implement our feature function like this?

```
In [11]: def feature(datum):
    feat = [1, float(datum[7])]
    return feat
```
Code: Finding the parameters

In [15]: \[\text{theta, residuals, rank, s = numpy.linalg.lstsq}(X, y)\]

In [16]: \[\text{theta}\]

Out[16]: \[\text{array([107.10183392, -0.68447989])}\]

\[\text{pm2.5 = 107.1 - 0.68 * temp}\]
pm2.5 = 3263.7
   - 3.109 \times \text{temp}
   - 3.065 \times \text{pressure}
   - 0.460 \times \text{wind speed}
Code: Doing the same thing manually

```
In [20]: theta
Out[20]: array([[ 3.26373064e+03, -3.10933772e+00, -3.06517728e+00, -4.60017221e-01]])

In [21]: X = numpy.matrix(X)
y = numpy.matrix(y)
numpy.linalg.inv(X.T * X) * X.T * y.T
Out[21]: matrix([[ 3.26373064e+03],
                 [-3.10933772e+00],
                 [-3.06517728e+00],
                 [-4.60017221e-01]])
```
Summary of concepts

• Demonstrated how to perform simple linear regression in Python
• Performed linear regression on an "air quality" example from the UCI Machine Learning Repository
• Introduced the numpy "least squares" function for linear regression

On your own...

• Try extending the code provided here to use different features and feature combinations