UNC CHARLOTTE

The WILLIAM STATES LEE COLLEGE *of* ENGINEERING

Introduction to ML
Lecture 5: Python for ML

Hamed Tabkhi
Department of Electrical and Computer Engineering,
University of North Carolina Charlotte (UNCC)
htabkhiv@uncc.edu

UNC CHARLOTTE

Anaconda

- A Python distribution made for large-scale data processing, predictive analytics, and scientific computing.
- Anaconda comes with NumPy, SciPy, matplotlib, pandas, IPython, Jupyter Notebook, and scikit-learn.
- Available on Mac OS, Windows, and Linux, it is a very convenient solution and is the one we suggest for people without an existing installation of the scientific Python packages.
- Anaconda now also includes the commercial Intel MKL library for free. Using MKL (which is done automatically when Anaconda is installed) can give significant speed improvements for many algorithms in scikit-learn.

- https://www.anaconda.com/

NumPy

- NumPy is one of the fundamental packages for scientific computing in Python.
- It contains functionality for multidimensional arrays, high-level mathematical functions such as linear algebra operations and the Fourier transform, and pseudorandom number generators.
- **Any data you're using will have to be converted to a NumPy array.**
- In scikit-learn, the NumPy array is the fundamental data structure.
- scikit-learn takes in data in the form of NumPy arrays.
- The core functionality of NumPy is the ndarray class, a multidimensional (*n*-dimensional) array.
- All elements of the array must be of the same type.

NumPy

In[1]:

```
import numpy as np

x = np.array([[1, 2, 3], [4, 5, 6]])
print("x:\n{}".format(x))
```

Out[1]:

```
x:
[[1 2 3]
 [4 5 6]]
```

matplotlib

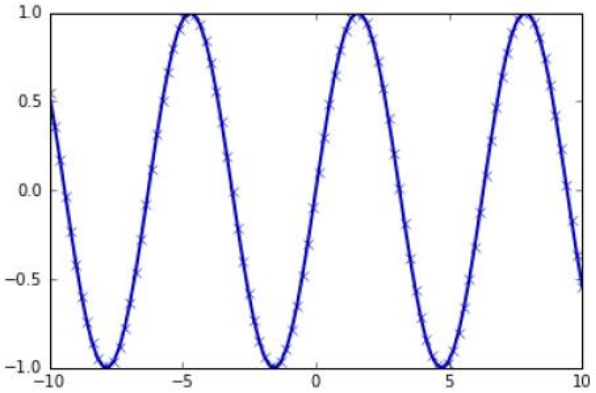
- matplotlib is the primary scientific plotting library in Python.
- It provides functions for making publication-quality visualizations such as line charts, histograms, scatter plots, and so on.
- Visualizing your data and different aspects of your analysis can give you important insights, and we will be using matplotlib for all our visualizations.
- When working inside the Jupyter Notebook, you can show figures directly in the browser by using the %matplotlib notebook and %matplotlib inline commands.

matplotlib

In[5]:

```
%matplotlib inline
import matplotlib.pyplot as plt

# Generate a sequence of numbers from -10 to 10 with 100 steps in between
x = np.linspace(-10, 10, 100)
# Create a second array using sine
y = np.sin(x)
# The plot function makes a line chart of one array against another
plt.plot(x, y, marker="x")
```



UNC CHARLOTTE

pandas

- pandas is a Python library for data wrangling and analysis.
- It is built around a data structure called the DataFrame that is modeled after the R DataFrame.
- pandas provides a great range of methods to modify and operate on tables; similar to an Excel spreadsheet.
- In contrast to NumPy, which requires that all entries in an array be of the same type, pandas allows each column to have a separate type (for example, integers, dates, floating-point numbers, and strings).
- In particular, it allows SQL-like queries and joins of tables.
- Another valuable tool provided by pandas is its ability to ingest from a great variety of file formats and databases, like SQL, Excel files, and comma-separated values (CSV) files.

pandas

In[6]:

```
import pandas as pd
from IPython.display import display

# create a simple dataset of people
data = {'Name': ["John", "Anna", "Peter", "Linda"],
        'Location' : ["New York", "Paris", "Berlin", "London"],
        'Age' : [24, 13, 53, 33]
       }

data_pandas = pd.DataFrame(data)
# IPython.display allows "pretty printing" of dataframes
# in the Jupyter notebook
display(data_pandas)
```

	Age	Location	Name
0	24	New York	John
1	13	Paris	Anna
2	53	Berlin	Peter
3	33	London	Linda

In[7]:

```
# Select all rows that have an age column greater than 30
display(data_pandas[data_pandas.Age > 30])
```

This produces the following result:

	Age	Location	Name
2	53	Berlin	Peter
3	33	London	Linda

SciPy

- SciPy is a collection of functions for scientific computing in Python.
- It provides, among other functionality, advanced linear algebra routines, mathematical function optimization, signal processing, special mathematical functions, and statistical distributions.
- scikit-learn draws from SciPy's collection of functions for implementing its algorithms.
- The most important part of SciPy for us is scipy.sparse: this provides *sparse matrices*, which are another representation that is used for data in scikitlearn.
- Sparse matrices are used whenever we want to store a 2D array that contains mostly zeros

SciPy

In[2]:

```
from scipy import sparse

# Create a 2D NumPy array with a diagonal of ones, and zeros everywhere else
eye = np.eye(4)
print("NumPy array:\n{}".format(eye))
```

Out[2]:

```
NumPy array:
[[ 1.  0.  0.  0.]
 [ 0.  1.  0.  0.]
 [ 0.  0.  1.  0.]
 [ 0.  0.  0.  1.]]
```

In[3]:

```
# Convert the NumPy array to a SciPy sparse matrix in CSR format
# Only the nonzero entries are stored
sparse_matrix = sparse.csr_matrix(eye)
print("\nSciPy sparse CSR matrix:\n{}".format(sparse_matrix))
```

Out[3]:

```
SciPy sparse CSR matrix:
  (0, 0)    1.0
  (1, 1)    1.0
  (2, 2)    1.0
  (3, 3)    1.0
```

SciPy

In[4]:

```
data = np.ones(4)
row_indices = np.arange(4)
col_indices = np.arange(4)
eye_coo = sparse.coo_matrix((data, (row_indices, col_indi
print("COO representation:\n{}".format(eye_coo))
```

Out[4]:

```
COO representation:
  (0, 0)    1.0
  (1, 1)    1.0
  (2, 2)    1.0
  (3, 3)    1.0
```

More details on SciPy sparse matrices can be found in the SciI

scikit-learn

- scikit-learn is an open=source project, meaning that it is free to use and distribute, and anyone can easily obtain the source code to see what is going on behind the scenes.
- The scikit-learn project is constantly being developed and improved, and it has a very active user community.
- It contains a number of state-of-the-art machine learning algorithms, as well as comprehensive documentation about each algorithm.
- scikit-learn is a very popular tool, and the most prominent Python library for machine learning.
- It is widely used in industry and academia, and a wealth of tutorials and code snippets are available online.
- scikit-learn works well with a number of other scientific Python tools
- The online documentation is very thorough, and this book will provide you with all the prerequisites in machine learning to understand it in detail.

- https://scikit-learn.org/stable/supervised_learning.html#supervised-learning

- https://scikit-learn.org/stable/user_guide.html