

Machine Learning: Introduction to Linear Regression, Logistic Regression, and Neural Networks

Chapter 2 Python Demos

Python Demos

Section	Title	Description
2.1	NumPy Demo	This section has a demo of the principal numpy functions to be used in the course
2.2	Matplotlib Demo Basic	This section has a demo of the basic matplotlib plotting functions
2.3	Matplotlib Demo Heatmap	This section has a demo of how to create a scatter plot and heatmap used to visualize machine learning data and results for classification
2.4	Pandas Demo	This section has a demo of how to use the basic pandas commands to read data from file and perform basic manipulations
2.5	unittest Demo	This section has a demo of how to set up unit tests
2.6	sklearn Demo	This section has a demo of how to convert messages into a “feature matrix”

2.1 NumPy Demo

NumPy Demo

NumPy (Numpy/numpy) is a Python package for scientific computing

- Key object is multi-dimensional numpy array
- numpy functions manipulate these arrays
 - Can perform standard matrix and vector operations
 - Can perform operations on entire array without explicit looping
- Course codes use numpy array as fundamental building block
- See following site for details: <https://numpy.org/>

Key Numpy Commands and Functions

Operation	numpy functions
Array creation	<code>numpy.array()</code>
Array indexing	
Component-wise operations: addition, multiplication, scalar multiplication	<code>+, *, *</code> with scalar
Functions	<code>numpy.exp()</code> , <code>numpy.absolute()</code> , <code>numpy.square()</code>
Concatenation, shaping, removal, addition of axes	<code>numpy.concatenate()</code> , <code>numpy.reshape()</code> , <code>numpy.squeeze()</code> , <code>numpy.expand_dims()</code>
Sum entries of array	<code>numpy.sum()</code>
Array of zeros, array of ones	<code>numpy.zeros()</code> , <code>numpy.ones()</code>
Array of random numbers: setting seed, from uniform distribution, from normal distribution	<code>numpy.random.seed()</code> , <code>numpy.random.rand()</code> , <code>numpy.random.randn()</code>

Numpy Demo

- `IntroML/Examples/Chapter2/NumpyDemo.ipynb`
 - Jupyter Notebook showing key numpy functions to be used in this course

2.2 Matplotlib Basic Demo

Matplotlib Basic Demo

Matplotlib is a Python package for plotting

- See following site for details: <https://matplotlib.org>
- This section has a demo of the basic plotting commands

Matplotlib Basics: Commands and Functions

Operation	matplotlib functions
Basic plotting using lists and numpy arrays and adding legends	plot(), legend()
Multiple plots	subplot()
Object oriented approach	
Scatter plots	plot(), scatter()
Histograms	hist()
Bar chart	bar()

Matplotlib Heatmap Demo

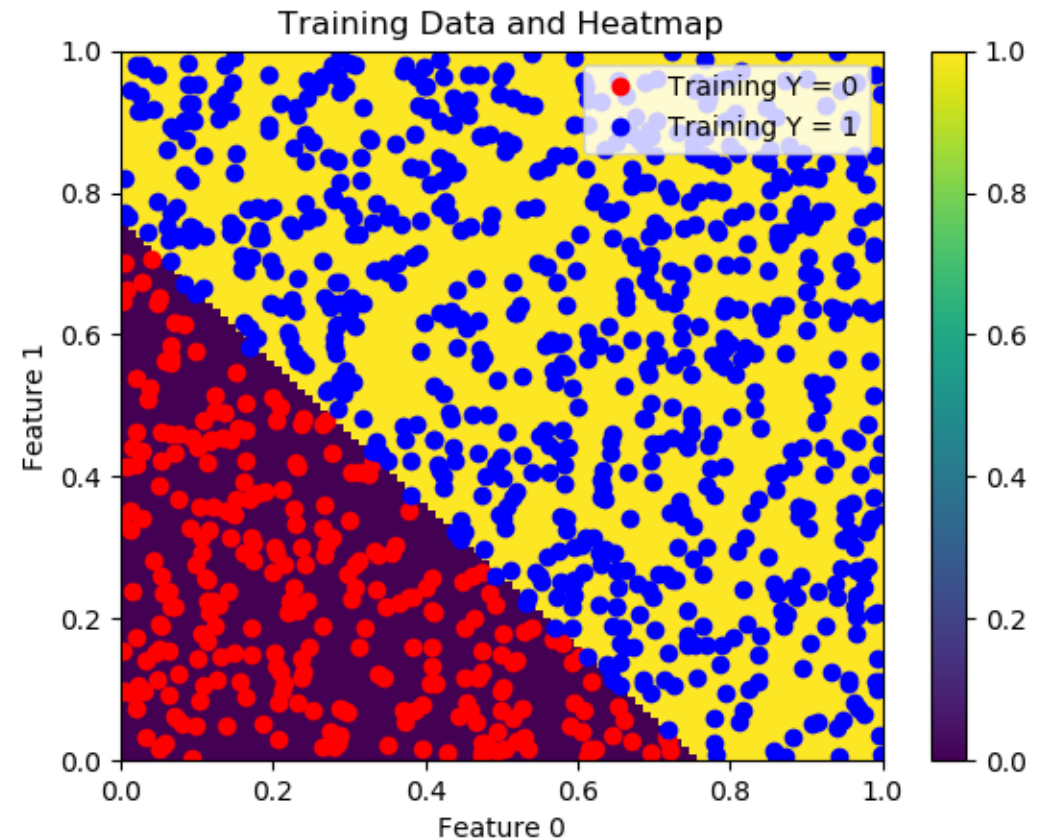
- [IntroML/Examples/Chapter2/MatplotlibBasicsDemo.ipynb](#)
 - Jupyter Notebook showing basic matplotlib plotting functions

2.3 Matplotlib Heatmap Demo

Matplotlib Heatmap

- Goal: produce plot showing “training data” and Machine Learning prediction
- Training data: red and blue points
- Machine Learning Prediction: heatmap (purple region is predicted 0 and yellow region is predicted 1)

Throughout this course we will produce plots like this to visualize effectiveness of machine learning algorithms for classification



Plotting Training Data

Generate training data: points in (X_0, X_1) plane and labels (0 or 1)

1. Identify indices for label = 0 – plot corresponding points red
2. Identify indices for label = 1 – plot corresponding points blue

Plotting Heatmap of Results

1. Create grid of points similar to that on right: (0.25,0.25), (0.5,0.25), (0.75,0.25), (0.25,0.5), (0.5,0.5), (0.75,0.5), (0.25,0.75), (0.5,0.75), (0.75,0.75)

2. Want to assign 0 or 1 to each point to create heatmap

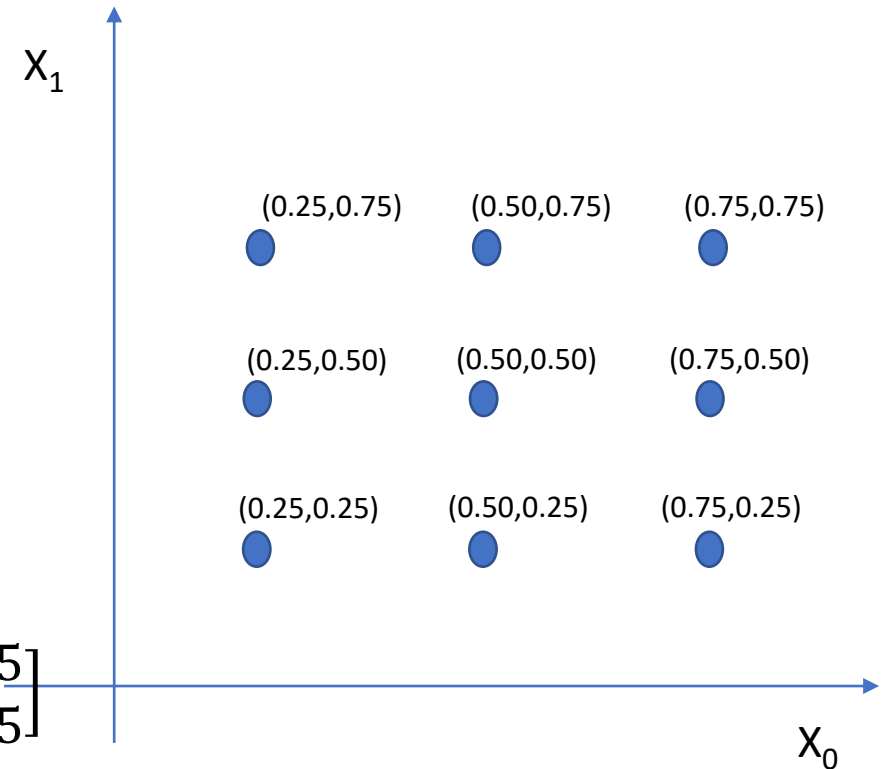
3. Use meshgrid to create 2d grids for X0 and X1 points

4. Reshape into 1d grids and create feature matrix

$$X = \begin{bmatrix} 0.25 & 0.5 & 0.75 & 0.25 & 0.5 & 0.75 & 0.25 & 0.5 & 0.75 \\ 0.25 & 0.25 & 0.25 & 0.5 & 0.5 & 0.5 & 0.75 & 0.75 & 0.75 \end{bmatrix}$$

5. Apply prediction algorithm to feature matrix to determine 0 or 1 label for each point

6. Reshape prediction into 3x3 grid and use pcolormesh function in matplotlib.pyplot to generate heatmap



Matplotlib Heatmap Demo

- `IntroML/Examples/Chapter2/MatplotlibHeatmapDemo.ipynb`
 - Jupyter Notebook showing how to produce plot of training data and heatmap

2.4 Pandas Demo

Pandas Demo

Pandas is a Python package containing data structures and analysis tools

- Will use pandas functions to read data from csv file and process data for case studies
- Key structure is data frame
- See following site for details: <https://pandas.pydata.org/>

Key Pandas Commands and Functions

Operation	pandas functions
Read data from csv file and put into data frame	<code>pandas.read_csv</code>
List items in data frame	<code>pandas.head()</code> , <code>pandas.tail()</code>
Extract column from data frame and remove column from data frame	<code>pandas.drop()</code>
Extract values from data frame	values attribute

Pandas Demo

- `IntroML/Examples/Chapter2/PandasDemo.ipynb`
 - Jupyter Notebook demo showing key matplotlib functions used in course

2.5 unittest Demo

unittest Demo

unittest is a package that is part of the Python release

- Package allows set up of unit tests
- Will set up unit tests to check derivative calculation functionality of course machine learning framework
- See following site for details:

<https://docs.python.org/3/library/unittest.html>

Unit Test Functionality

```
In [1]: import unittest

class Test(unittest.TestCase):
    def test1(self):
        x = 7
        y = 8
        z1 = (x+y)*(x+y)
        z2 = x*x + 2*x*y + y*y
        error = abs(z1-z2)
        self.assertLessEqual(error, 1e-7)

if __name__ == "__main__":
    #this is command in python when running in command window
    #unittest.main()
    # this is command in the jupyter notebook
    unittest.main(argv=['first-arg-is-ignored'], exit=False)
```

.

Ran 1 test in 0.016s

OK

- Use functionality in unittest package
- Documentation at <https://docs.python.org/3.7/index.html>
- Create a class derived from unittest.TestCase
- Individual unit tests are set up as methods of the class
- Test should have “assert” command which determines pass or fail
- Use unittest.main to run tests
- Will get OK if test passes

unittest Demo

- `IntroML/Examples/Chapter2/unittestDemo.ipynb`
 - Jupyter Notebook demo showing how to set up a unittest

2.6 sklearn Demo

sklearn Demo

- sklearn is shortened version of name of package scikit-learn
- Will use text processing functionality in CountVectorizer in sklearn for spam classification case study
- See following site for details: <https://scikit-learn.org/stable/index.html>

CountVectorizer - Example

- 3 Messages: "Call me soon", "CALL to win", "Pick me up soon"
- CountVectorizer determines unique words (My, my, MY are the same)
- CountVectorizer creates matrix (# words x # messages) with number of times each word appears in a message

Vocabulary:

call
me
pick
soon
to
up
win

Feature Matrix:

1	1	0
1	0	1
0	0	1
1	0	1
0	1	0
0	0	1
0	1	0

Most Common Words in Vocabulary

- Sum Feature Matrix in col direction to get count of words for all messages

Vocabulary:

call
me
pick
soon
to
up
win

Feature Matrix:

$$\begin{bmatrix} 1 & 1 & 0 \\ 1 & 0 & 1 \\ 0 & 0 & 1 \\ 1 & 0 & 1 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix}$$

Count:

$$\begin{bmatrix} 2 \\ 2 \\ 1 \\ 2 \\ 1 \\ 1 \\ 1 \end{bmatrix}$$

- Can use numpy argsort function to get indices of most common words:
 - In case of ties, go in order of index
 - Indices of 4 most common words: 0, 1, 3, 2

sklearn

- IntroML/Examples/Chapter2/sklearnDemo.ipynb
 - Jupyter Notebook demo use of CountVectorizer for text processing