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

2.1 Numpy Demo

NumPy Demo

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

Numpy Demo

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

2.2 Matplotlib Demo

Matplotlib Demo

Matplotlib is a Python package for plotting

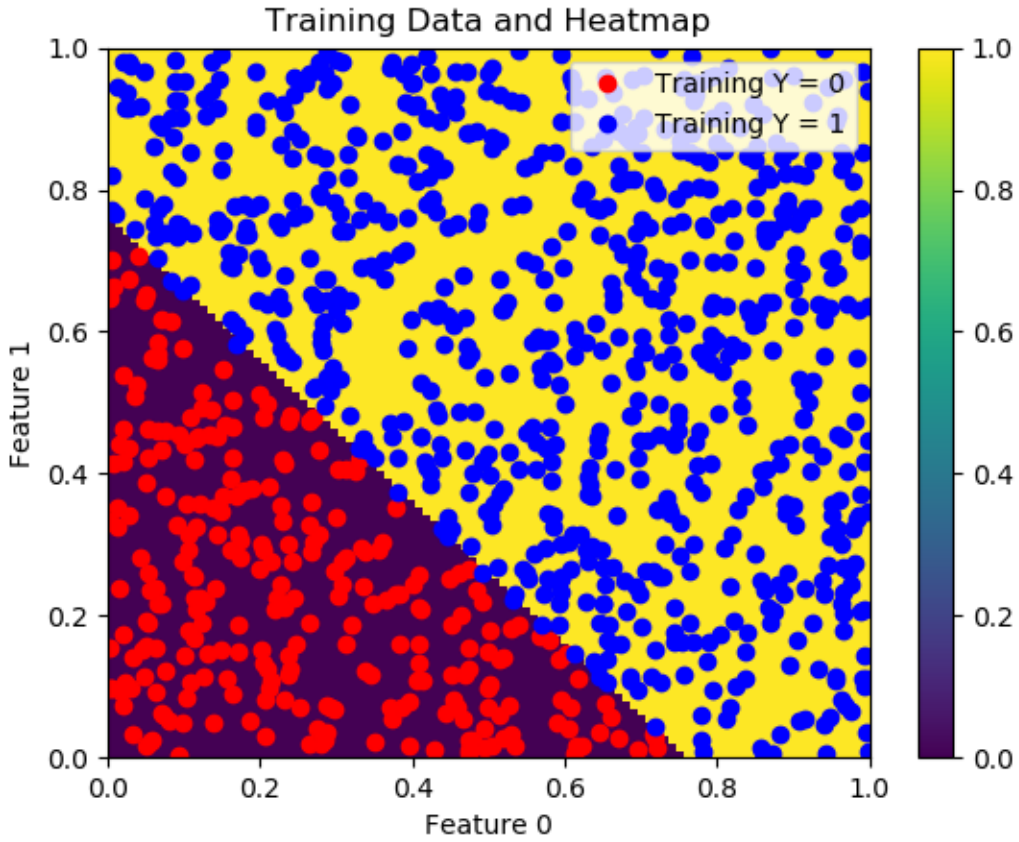
- See following site for details: https://matplotlib.org

Matplotlib Basics: Commands and Functions

Operation	matplotlib functions
Plotting	plot
Multiple plots and adding legends	subplot(), legend()
Scatter plots	scatter
Plotting training data for classification	

Matplotlib Heatmap

- Goal: produce plot showing “training data” and Machine Learning prediction
- Training data: red and blue points
- Predicted Results: results predicted by model (purple is predicted 0 and yellow is predicted 1)



Plotting Logistic Regression Training Data

Generate training data: points in (X0,X1) plane and labels (0 or 1)

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

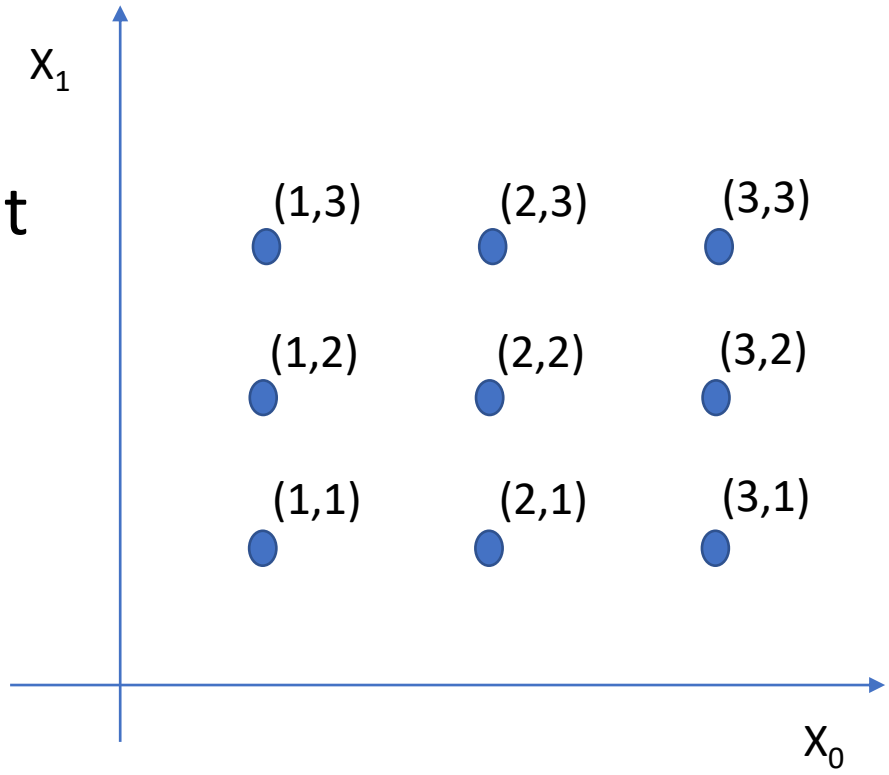
Plotting Heatmap of Results

Assume that training algorithm has been performed with training data

1. Create grid of points similar to that on right
2. Points: (1,1), (2,1), (3,1), (1,2), (2,2), (3,2), (1,3), (2,3), (3,3)
3. Feature Matrix:

$$X = \begin{bmatrix} 1 & 2 & 3 & 1 & 2 & 3 & 1 & 2 & 3 \\ 1 & 1 & 1 & 2 & 2 & 2 & 3 & 3 & 3 \end{bmatrix}$$

4. Apply prediction algorithm with feature matrix representing grid of points to produce 0 or 1 label for each point
5. Use pcolormesh function in matplotlib.pyplot applied to grid and labels to generate heatmap



Matplotlib Heatmap Demo

- IntroML/Examples/Chapter2/MatplotlibBasicsDemo.ipynb
  - Jupyter Notebook showing basic matplotlib plotting functions
- IntroML/Examples/Chapter2/MatplotlibHeatmapDemo.ipynb
  - Jupyter Notebook showing how to produce plot of training data and heatmap

2.3 Pandas Demo

Pandas Demo

Pandas is a Python package containing data structures and analysis tools

- 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	pandas.read_csv
List items in data frame	pandas.head(), pandas.tail()
Extract column from data frame and remove column from data frame	pandas.drop()
Extract values from data frame	values

Pandas Demo

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

2.4 unittest Demo

unittest Demo

unittest is a package that is part of the Python release

- Package allows set up of unit tests
- 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