

Python и его применение в анализе данных

Панов Максим

Московский Физико-Технический Институт
Факультет Управления и Прикладной Математики

24.03.2010 г.

Содержание презентации

- ▶ Что такое Python?

Содержание презентации

- ▶ Что такое Python?
- ▶ Математические библиотеки Python

Содержание презентации

- ▶ Что такое Python?
- ▶ Математические библиотеки Python
- ▶ Однослойный перцептрон

Содержание презентации

- ▶ Что такое Python?
- ▶ Математические библиотеки Python
- ▶ Однослойный перцептрон
- ▶ Выводы

Содержание презентации

- ▶ Что такое Python?
- ▶ Математические библиотеки Python
- ▶ Однослойный перцептрон
- ▶ Выводы
- ▶ То, что не вошло в этот обзор

Что такое Python?

Python - высокоуровневый язык программирования общего назначения

Основные акценты на:

- ▶ производительность разработчика
- ▶ читаемость кода

Дзэн Питона

Красивое лучше, чем уродливое.

Дзэн Питона

Красивое лучше, чем уродливое.
Явное лучше, чем неявное.

Дзэн Питона

Красивое лучше, чем уродливое.

Явное лучше, чем неявное.

Простое лучше, чем сложное.

Дзэн Питона

Красивое лучше, чем уродливое.

Явное лучше, чем неявное.

Простое лучше, чем сложное.

Сложное лучше, чем запутанное.

Дзэн Питона

Красивое лучше, чем уродливое.

Явное лучше, чем неявное.

Простое лучше, чем сложное.

Сложное лучше, чем запутанное.

Плоское лучше, чем вложенное.

Дзэн Питона

Красивое лучше, чем уродливое.

Явное лучше, чем неявное.

Простое лучше, чем сложное.

Сложное лучше, чем запутанное.

Плоское лучше, чем вложенное.

Разреженное лучше, чем плотное.

Дзэн Питона

Красивое лучше, чем уродливое.

Явное лучше, чем неявное.

Простое лучше, чем сложное.

Сложное лучше, чем запутанное.

Плоское лучше, чем вложенное.

Разреженное лучше, чем плотное.

Читабельность имеет значение.

Дээн Питона

Красивое лучше, чем уродливое.

Явное лучше, чем неявное.

Простое лучше, чем сложное.

Сложное лучше, чем запутанное.

Плоское лучше, чем вложенное.

Разреженное лучше, чем плотное.

Читабельность имеет значение.

Особые случаи не настолько особые, чтобы нарушать правила.

Дзэн Питона

Красивое лучше, чем уродливое.

Явное лучше, чем неявное.

Простое лучше, чем сложное.

Сложное лучше, чем запутанное.

Плоское лучше, чем вложенное.

Разреженное лучше, чем плотное.

Читабельность имеет значение.

Особые случаи не настолько особые, чтобы нарушать правила.

Хотя практичность побеждает стремление к чистоте.

Дзэн Питона

Красивое лучше, чем уродливое.

Явное лучше, чем неявное.

Простое лучше, чем сложное.

Сложное лучше, чем запутанное.

Плоское лучше, чем вложенное.

Разреженное лучше, чем плотное.

Читабельность имеет значение.

Особые случаи не настолько особые, чтобы нарушать правила.

Хотя практичность побеждает стремление к чистоте.

Ошибки никогда не должны замалчиваться.

Дзэн Питона

Красивое лучше, чем уродливое.

Явное лучше, чем неявное.

Простое лучше, чем сложное.

Сложное лучше, чем запутанное.

Плоское лучше, чем вложенное.

Разреженное лучше, чем плотное.

Читабельность имеет значение.

Особые случаи не настолько особые, чтобы нарушать правила.

Хотя практичность побеждает стремление к чистоте.

Ошибки никогда не должны замалчиваться.

Если не замалчиваются явно.

Дзэн Питона

Если видишь двусмысленность, отбрось искушение угадать.

Дзэн Питона

Если видишь двусмысленность, отбрось искушение угадать.
Должен существовать один — и, желательно, только один —
очевидный способ сделать это.

Дзэн Питона

Если видишь двусмысленность, отбрось искушение угадать.
Должен существовать один — и, желательно, только один —
очевидный способ сделать это.
Хотя он поначалу может быть и не очевиден, если вы не
голландец.

Дзэн Питона

Если видишь двусмысленность, отбрось искушение угадать.
Должен существовать один — и, желательно, только один —
очевидный способ сделать это.
Хотя он поначалу может быть и не очевиден, если вы не
голландец.
Сейчас лучше, чем никогда.

Дзэн Питона

Если видишь двусмысленность, отбрось искушение угадать.
Должен существовать один — и, желательно, только один —
очевидный способ сделать это.

Хотя он поначалу может быть и не очевиден, если вы не
голландец.

Сейчас лучше, чем никогда.

Хотя никогда зачастую лучше, чем прямо сейчас.

Дзэн Питона

Если видишь двусмысленность, отбрось искушение угадать.
Должен существовать один — и, желательно, только один —
очевидный способ сделать это.

Хотя он поначалу может быть и не очевиден, если вы не
голландец.

Сейчас лучше, чем никогда.

Хотя никогда зачастую лучше, чем прямо сейчас.

Если реализацию сложно объяснить — идея плоха.

Дзэн Питона

Если видишь двусмысленность, отбрось искушение угадать.

Должен существовать один — и, желательно, только один — очевидный способ сделать это.

Хотя он поначалу может быть и не очевиден, если вы не голландец.

Сейчас лучше, чем никогда.

Хотя никогда зачастую лучше, чем прямо сейчас.

Если реализацию сложно объяснить — идея плоха.

Если реализацию легко объяснить — идея, возможно, хороша.

Дзэн Питона

Если видишь двусмысленность, отбрось искушение угадать.

Должен существовать один — и, желательно, только один — очевидный способ сделать это.

Хотя он поначалу может быть и не очевиден, если вы не голландец.

Сейчас лучше, чем никогда.

Хотя никогда зачастую лучше, чем прямо сейчас.

Если реализацию сложно объяснить — идея плоха.

Если реализацию легко объяснить — идея, возможно, хороша.

Пространства имён, черт возьми, — отличная штука! Будем делать их побольше!

Математические библиотеки Python

- ▶ NumPy
- ▶ SciPy
- ▶ Matplotlib

NumPy

- ▶ NumPy - это расширение языка Python, добавляющее поддержку многомерных массивов и матриц, вместе с большой библиотекой математических функций для операций с этими массивами.

SciPy

- ▶ SciPy - это открытая библиотека высококачественных научных инструментов для языка программирования Python. SciPy содержит модули для оптимизации, интегрирования, специальных функций, обработки сигналов, обработки изображений, генетических алгоритмов, решения обыкновенных дифференциальных уравнений.

Matplotlib

- ▶ Matplotlib - это библиотека языка программирования Python и его расширения NumPy, предназначенная для визуализации данных.

Импорт библиотек

```
from numpy import *  
from pylab import plot, show
```

Загрузка данных

```
features = loadtxt('iris.txt', delimiter = ',')  
features = mat(features)  
features = delete(features, [0, 1], axis = 1)  
types = concatenate((zeros((50, 1)), ones((50, 1)))), axis = 0)
```


Подготовка данных

```
def OneLayerPerceptronTrain(features, types, accuracy):
```

Подготовка данных

```
def OneLayerPerceptronTrain(features, types, accuracy):  
    nFeatures = features.shape[1]  
    nObjects = features.shape[0]
```

Подготовка данных

```
def OneLayerPerceptronTrain(features, types, accuracy):  
    nFeatures = features.shape[1]  
    nObjects = features.shape[0]  
  
    weights = - ones(nFeatures + 1) / 2 / nFeatures +  
              random.rand(1, nFeatures + 1) / nFeatures
```

Подготовка данных

```
def OneLayerPerceptronTrain(features, types, accuracy):  
    nFeatures = features.shape[1]  
    nObjects = features.shape[0]  
  
    weights = - ones(nFeatures + 1) / 2 / nFeatures +  
              random.rand(1, nFeatures + 1) / nFeatures  
  
    meanFeatures = mean(features, axis = 0)  
    features -= meanFeatures  
    features = concatenate((features, -ones((nObjects, 1))), axis=1)
```

Главный цикл

```
while (abs(empiricalRisk - supplEmpiricalRisk)) > accuracy):
```

Главный цикл

```
while (abs(empiricalRisk - supplEmpiricalRisk)) > accuracy):  
    i = int(floor(random.rand(1, 1) * nObjects))  
    lossFunc[i] = power(sigmoidFunc[i] - types[i], 2)
```

Главный цикл

```
while (abs(empiricalRisk - supplEmpiricalRisk)) > accuracy):  
    i = int(floor(random.rand(1, 1) * nObjects))  
    lossFunc[i] = power(sigmoidFunc[i] - types[i], 2)  
    weights = weights - 2 * rate * (sigmoidFunc[i] types[i]) *  
    (1 - sigmoidFunc[i]) * sigmoidFunc[i] * features[i, :]
```

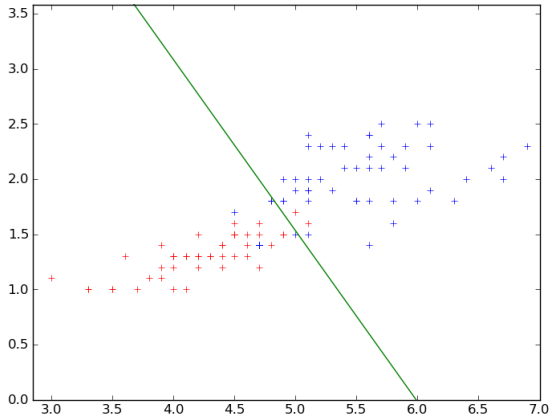
Главный цикл

```
while (abs(empiricalRisk - supplEmpiricalRisk)) > accuracy):  
    i = int(floor(random.rand(1, 1) * nObjects))  
    lossFunc[i] = power(sigmoidFunc[i] - types[i], 2)  
    weights = weights - 2 * rate * (sigmoidFunc[i] - types[i]) *  
    (1 - sigmoidFunc[i]) * sigmoidFunc[i] * features[i, :]  
    empiricalRisk = (1-delta)*empiricalRisk + delta*lossFunc[i]  
return weights.T
```


Графика

```
plot(features[where(typ==0),0], features[where(typ==0),1], 'r+')  
plot(features[where(typ==1),0], features[where(typ==1),1], 'b+')  
plot(discriminantLine[0], discriminantLine[1], 'g');
```

Графика



Python + NumPy + ScyPi + Matplotlib

=

мощная альтернатива MatLab

- ▶ RPy
- ▶ SymPy
- ▶ Pythonica
- ▶ Orange
- ▶ Sage

Спасибо за внимание!
Вопросы приветствуются!