



wepon的专栏

- 目录视图
- 摘要视图
- RSS 订阅

个人资料

wepon_

关注发私信

访问：692034次

积分：5183

等级：BLOG 6

排名：第5110名

原创：72篇

转载：3篇

译文：0篇

评论：370条

About

个人网站：<http://wepon.me/>

github：<https://github.com/wepe>

知乎：<https://www.zhihu.com/people/wepon-huang>

很久没上CSDN，有问题欢迎邮件交流：masterwepon@163.com

文章搜索

文章分类

Machine Learning (31)

Kaggle (2)

scikit-learn使用手册 (1)

python (16)

数据库 (1)

openCV (3)

Leetcode (28)

ACM题解析 (1)

品味经典书籍 (1)

杂谈 (0)

文章存档

2016年02月 (1)

程序员，为什么写不好一份简历？[征文 | 你会为 AI 转型么？](#)[福利 | 免费参加 2017 OpenStack Days China](#)

大数据竞赛平台——Kaggle 入门

标签：[Kaggle](#) [机器学习](#) [数据挖掘](#) [python](#)

2014-12-14 21:3479765人阅读评论(34)收藏举报

分类：

[Kaggle \(1 \)](#) [python \(15 \)](#) [Machine Learning \(30 \)](#)

版权声明：本文为博主原创文章，未经博主允许不得转载。

目录(?)

[+]

大数据竞赛平台——Kaggle 入门篇

这篇文章适合那些刚接触Kaggle、想尽快熟悉Kaggle并且独立完成一个竞赛项目的网友，对于已经在Kaggle上参赛过的网友来说，大可不必耗费时间阅读本文。本文分为两部分介绍Kaggle，第一部分简单介绍Kaggle，第二部分将展示解决一个竞赛项目的全过程。如有错误，请指正！

1、Kaggle简介

Kaggle是一个数据分析的竞赛平台，网址：<https://www.kaggle.com/>

企业或者研究者可以将数据、问题描述、期望的指标发布到Kaggle上，以竞赛的形式向广大的数据科学家征集解决方案，类似于[KDD-CUP](#)（国际知识发现和数据挖掘竞赛）。Kaggle上的参赛者将数据下载下来，分析数据，然后运用机器学习、数据挖掘等知识，建立算法模型，解决问题得出结果，最后将结果提交，如果提交的结果符合指标要求并且在参赛者中排名第一，将获得比赛丰厚的奖金。更多内容可以参阅：[大数据众包平台](#)

下面我以图文的形式介绍Kaggle：

[进入Kaggle网站：](#)

2015年09月	(1)
2015年08月	(1)
2015年05月	(3)
2015年04月	(3)
展开	

阅读排行	
大数据竞赛平台——Kaggle 入...	(79657)
交叉熵代价函数	(54503)
DeepLearning tutorial (4) C...	(38530)
DeepLearning tutorial (6) 易...	(38494)
DeepLearning tutorial (5) C...	(36594)
正则化方法：L1和L2 regulariz...	(24591)
【机器学习算法实现】主成分...	(21406)
DeepLearning tutorial (1) S...	(20412)
scikit-learn中PCA的使用方法	(17973)
机器学习算法中如何选取超参...	(16929)

评论排行	
DeepLearning tutorial (6) 易...	(76)
DeepLearning tutorial (5) C...	(59)
大数据竞赛平台——Kaggle 入...	(34)
DeepLearning tutorial (7) 深...	(33)
DeepLearning tutorial (4) C...	(21)
正则化方法：L1和L2 regulariz...	(15)
Kaggle入门——使用scikit-lear...	(14)
交叉熵代价函数	(10)
【机器学习算法实现】logistic...	(8)
DeepLearning tutorial (3) M...	(7)

推荐文章	
* CSDN日报20170714——《从创业到再就业，浅述对程序员职业生涯的看法》	
* Android 逆向 锁屏密码算法解析以及破解方案	
* 从单一WAR到多活，记述一个创业公司的架构演变	
* 我的AI转型之路与AI之我见（非985211的奋斗路程与视角）	
* AI大行其道，你准备好了吗？—谨送给徘徊于转行AI的程序员	
* AI转型中的思考和洞见	

最新评论	
DeepLearning tutorial (5) CNN卷积神... 草谷乐	: 博主,你好,请问from theano.tensor.signal.pool import down...
OpenCV人脸检测（C++代码） jiachen0212	: 赞！
DeepLearning tutorial (6) 易用的深度学... dittoya	: ValueError: 'The specified si ze contains a dim...
DeepLearning tutorial (6) 易用的深度学... dittoya	: ValueError: ‘The specified size co ntains a dimensi...
DeepLearning tutorial (1) Softmax回归... gg112324d	: @kobecsb:sigmoid二分类soft max多分类
朴素贝叶斯理论推导与三种常见模型 南宫娜娜	: 看了一下您的代码 def _calculat

Active Competitions			
Featured		Helping Santa's Helpers Jingle bells, Santa tells ...	24 days 206 teams \$20,000
		Click-Through Rate Prediction Predict whether a mobile ad will be clicked	57 days 843 teams \$15,000
		BCI Challenge @ NER 2015 A spell on you if you cannot detect errors!	2 months 122 teams \$1,000

这是当前正在火热进行的有奖比赛，有冠军杯形状的是“Featured”，汇集数据科学高手去参赛。下面那个灰色的有试剂瓶形状的是“Research Point”。这两个类别的比赛是有奖竞赛，难度自然不小，作为入门者，应比赛：

101		Data Science London + Scikit-learn Scikit-learn is an open-source machine learning library for Python. Give it a try here!	17 days 149 teams Knowledge
		When bag of words meets bags of popcorn Use Google's Word2Vec for movie reviews	12 months 30 teams Knowledge
		Digit Recognizer Classify handwritten digits using the famous MNIST data	12 months 495 teams Knowledge
		Titanic: Machine Learning from Disaster Predict survival on the Titanic (with tutorials in Excel, Python, R, and an introduction to Random Forests)	12 months 2124 teams
		Facial Keypoints Detection Detect the location of keypoints on face images	12 months 38 teams Knowledge
		First steps with Julia Identify characters from Google Street View Pictures + tutorial with Julia.	12 months 32 teams Knowledge
Playground		Sentiment Analysis on Movie Reviews Classify the sentiment of sentences from the Rotten Tomatoes dataset	2 months 627 teams Knowledge
		Finding Elo Predict a chess player's FIDE Elo rating from one game	3 months 88 teams Knowledge
		Billion Word Imputation Find and impute missing words in the billion word corpus	4 months 59 teams Knowledge
		Forest Cover Type Prediction Use cartographic variables to classify forest categories	4 months 925 teams Knowledge
		Bike Sharing Demand Forecast use of a city bikeshare system	5 months 1591 teams Knowledge
		Random Acts of Pizza Predicting altruism through free pizza	5 months 285 teams
		Poker Rule Induction Determining the rules of a hand of five-card stud	5 months 19 teams

左图的比赛是“101”，右图的是“Playground”，都是练习赛，适合入门。入门Kaggle最好的方法就是独立完成101和playground这两个级别的竞赛项目。本文的第二部分将选101中的“Digit Recognition”作为讲解。

[点击进入赛题“Digit Recognition”](#)：

关闭

e_feature_prob(self,feature...

朴素贝叶斯理论推导与三种常见模型

南宫娜娜：你好，这里2.1.1 举例中计算条件概率的时候，介绍2.1的描述中：“Nyk是类别为yk的样本个数，...

【机器学习算法实现】主成分分析(PCA)—...

Bruin0：@funny_QZQ:你好，我想问一下，pca降维之后的数据还能知道代表的什么意思吗？

【机器学习算法实现】主成分分析(PCA)—...

Bruin0：pca降维之后，降维的数据代表原来数据的什么含义

DeepLearning tutorial（6）易用的深度学...

iHunter001：按照博主的程序试着跑了一下，出现了错误TypeError: Layer can receive a t...

9665407401
3134727121
1742351244

Knowledge • 496 teams

Digit Recognizer

Wed 25 Jul 2012

Thu 31 Dec 2015 (12 months to go)

Dashboard

Home

Data

Make a submission

Information

Description

Evaluation

Rules

Tutorial

Forum

Leaderboard

Visualization

My Team

GitHub

My Submissions

Competition Details » Get the Data » Make a submission

Classify handwritten digits using the famous MNIST data

http://blog.csdn.net/u012162613

This competition is the first in a series of tutorial competitions designed to introduce people to Machine Learning.

The goal in this competition is to take an image of a handwritten single digit, and determine what that digit is. As the competition progresses, we will release tutorials which explain different machine learning algorithms and help you to get started.

The data for this competition were taken from the MNIST dataset. The MN ("Modified National Institute of Standards and Technology") dataset is a cl the Machine Learning community that has been extensively studied. Mor the dataset, including Machine Learning algorithms that have been tried o

这是一个识别数字0～9的练习赛，“Competition Details”是这个比赛的
参赛者需要解决的问题。“Get the Data”是数据下载，参赛者用这些数
据训练自己的模型，得出结果，数据一般都是以csv格式给出：

9665407401
3134727121
1742351244

Knowledge • 496 teams

Digit Recognizer

Wed 25 Jul 2012

Thu 31 Dec 2015 (12 months to go)

Dashboard

Home

Data

Make a submission

Information

Description

Evaluation

Rules

Tutorial

Forum

Leaderboard

Visualization

My Team

GitHub

My Submissions

Competition Details » Get the Data » Make a submission

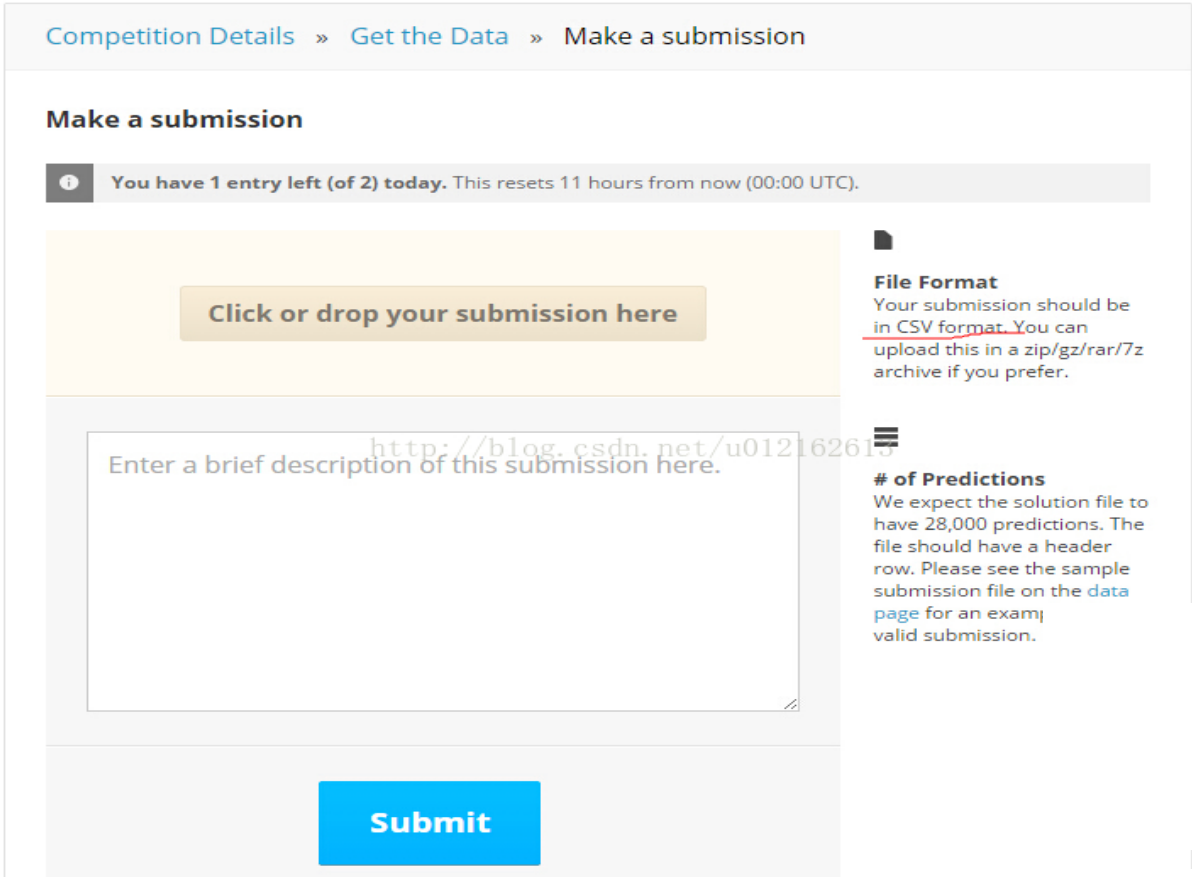
Data Files

File Name	Available Formats
train	.csv (73.22 mb)
test	.csv (48.75 mb)
knn_benchmark	.R (316 b)
knn_benchmark	.csv (235.26 kb)
rf_benchmark	.R (381 b)
rf_benchmark	.csv (235.26 kb)

其中，train.csv就是训练样本，test.csv就是测试样本，由于这个是训练赛，所以还
提供了两种解决方案，knn_benchmark.R和rf_benchmark.R，前者是用R语。言写
的knn算法程序，后者是用R语言写的随机森林算法程序，它们的结果分别是
knn_benchmark.csv和rf_benchmark.csv。关于csv格式文件，我前一篇文章有详
述：【Python】csv模块的使用。

关闭

得出结果后，接下来就是提交结果“Make a submission”：



要求提交的文件是csv格式的，假如你将结果保存在result.csv，那么点“drop submission here”，选中result.csv文件上传即可，系统将测试你提交的结果的准确率，然后排名。

另外，除了“Competition Details”、“Get the Data”、“Make a submission”，侧边栏的“Home”、“Information”、“Forum”等，也提供了关于竞赛的一些相关信息，包括排名、规则、辅导.....

【以上是第一部分，暂且写这么多，有补充的以后再更】

2、竞赛项目解题全过程

(1) 知识准备

首先，想解决上面的题目，还是需要一点ML算法的基础的，另外就是要会用编程语言和相应的第三方库来实现算法，常用的有：

Python以及对应的库numpy、scipy、scikit-learn（实现了ML的一些算法，可以直接用）、theano（DeepLearning的算法包）。

R语言、weka

如果用到深度学习的算法，cuda、caffe也可以用

总之，使用什么编程语言、什么平台、什么第三方库都无所谓，无论用什么方法，Kaggle只需要你线上提交结果，线下你如何实现算法是没有限制的。

关闭

Ok，下面讲解题过程，以“Digit Recognition”为例，数字识别这个问题我之前写过两篇文章，分别用kNN算法和Logistic算法去实现，有完整的代码，有兴趣可以阅读：[kNN算法实现数字识别](#)、[Logistic回归实现数字识别](#)




(2) Digit Recognition解题过程

下面我将采用kNN算法来解决Kaggle上的这道Digit Recognition训练题。上面提到，我之前用kNN算法实现过，这里我将直接copy之前的算法的核心代码，核心代码是关于kNN算法的主体实现，我不再赘述，我把重点放在处理数据上。

以下工程基于Python、numpy

- 获取数据

从“Get the Data”下载以下三个csv文件：

 knn_benchmark.csv	2014/12/7 13:38	Microsoft Excel ...	2
 test.csv	2014/12/7 14:13	Microsoft Excel ...	49,9
 train.csv	2014/12/7 14:40	Microsoft Excel ...	74,9

- 分析train.csv数据

train.csv是训练样本集，大小42001*785，第一行是文字描述，所以数据大小是42000*785，其中第一列的每一个数字是它对应行的label，可以将第一列单独取出来，得到42000*1的向量trainLabel，剩下的就是42000*784的特征向量集trainData，所以从**train.csv**可以获取两个矩阵**trainLabel**、**trainData**。

下面给出代码，另外关于如何从csv文件中读取数据，参阅：[csv模块的使用](#)

```
[python]
01. def loadTrainData():
02.     l=[]
03.     with open('train.csv') as file:
04.         lines=csv.reader(file)
05.         for line in lines:
06.             l.append(line) #42001*785
07.     l.remove(l[0])
08.     l=array(l)
09.     label=l[:,0]
10.     data=l[:,1:]
11.     return nomalizing(toInt(data)),toInt(label)
```

这里还有两个函数需要说明一下，toInt()函数，是将字符串转换为整数，因为从csv文件读取出来的，是字符串类型的，比如‘253’，而我们接下来运算需要的是整数类型的，因此要转换，int(‘253’)=253。toInt()函数如下：

```
[python]
01. def toInt(array):
02.     array=mat(array)
03.     m,n=shape(array)
04.     newArray=zeros((m,n))
05.     for i in xrange(m):
06.         for j in xrange(n):
07.             newArray[i,j]=int(array[i,j])
08.     return newArray
```

关闭

nomalizing()函数做的工作是归一化，因为train.csv里面提供的表示图像的数据是0~255的，为了简化运算，我们可以将其转化为二值图像，因此将所有非0的数字，即1~255都归一化为1。nomalizing()函数如下：

```
[python]
01. def nomalizing(array):
02.     m,n=shape(array)
03.     for i in xrange(m):
04.         for j in xrange(n):
05.             if array[i,j]!=0:
06.                 array[i,j]=1
07.     return array
```

• 分析test.csv数据

test.csv里的数据大小是28001*784，第一行是文字描述，因此实际的测试数据样本是28000*784，与train.csv不同，没有label，28000*784即28000个测试样本，我们要做的工作就是为这28000个测试样本找出正确的label。所以从test.csv我们可以得到测试样本集testData，代码如下：

```
[python]
01. def loadTestData():
02.     l=[]
03.     with open('test.csv') as file:
04.         lines=csv.reader(file)
05.         for line in lines:
06.             l.append(line)
07.         #28001*784
08.     l.remove(l[0])
09.     data=array(l)
10.     return nomalizing(toInt(data))
```

• 分析knn_benchmark.csv

前面已经提到，由于digit recognition是训练赛，所以这个文件是官方给出的参考结果，本来可以不理这个文件的，但是我下面为了对比自己的训练结果，所以也把knn_benchmark.csv这个文件读取出来，这个文件里的数据是28001*2，第一行是文字说明，可以去掉，第一列表示图片序号1~28000，第二列是图片对应的数字。从knn_benchmark.csv可以得到28000*1

关闭

码：

```
[python]
01. def loadTestResult():
02.     l=[]
03.     with open('knn_benchmark.csv') as file:
04.         lines=csv.reader(file)
05.         for line in lines:
06.             l.append(line)
07.         #28001*2
08.     l.remove(l[0])
09.     label=array(l)
```

```
10.         return toInt(label[:,1])
```

到这里，数据分析和处理已经完成，我们获得的矩阵有：trainData、trainLabel、testData、testResult

• 算法设计

这里我们采用kNN算法来分类，核心代码：

[python]

```
01. def classify(inX, dataSet, labels, k):
02.     inX=mat(inX)
03.     dataSet=mat(dataSet)
04.     labels=mat(labels)
05.     dataSetSize = dataSet.shape[0]
06.     diffMat = tile(inX, (dataSetSize,1)) - dataSet
07.     sqDiffMat = array(diffMat)**2
08.     sqDistances = sqDiffMat.sum(axis=1)
09.     distances = sqDistances**0.5
10.     sortedDistIndicies = distances.argsort()
11.     classCount={}
12.     for i in range(k):
13.         voteIlabel = labels[0,sortedDistIndicies[i]]
14.         classCount[voteIlabel] = classCount.get(voteIlabel,0) + 1
15.     sortedClassCount = sorted(classCount.iteritems(), key=operator.itemgetter(1), r
16.     return sortedClassCount[0][0]
```

关于这个函数，参考：[kNN算法实现数字识别](#)

简单说明一下，inX就是输入的单个样本，是一个特征向量。dataSet是训练样本，对应上面的trainData，labels对应trainLabel，k是knn算法选定的k，一般选择0~20之间的数字。这个函数将返回inX的label，即图片inX对应的数字。对于测试集里28000个样本，调用28000次这个函数即可。

• 保存结果

kaggle上要求提交的文件格式是csv，上面我们得到了28000个测试样本的label，必须将其保存成csv格式文件才可以提交，关于csv，参考：[【Python】csv模块的使用](#)。

代码:

[python]

```
01. def saveResult(result):
02.     with open('result.csv','wb') as myFile:
03.         myWriter=csv.writer(myFile)
04.         for i in result:
05.             tmp=[]
06.             tmp.append(i)
07.             myWriter.writerow(tmp)
```

关闭

• 综合各函数

上面各个函数已经做完了所有需要做的工作，现在需要写一个函数将它们组合起来解决digit recognition这个题目。我们写一个handwritingClassTest函数，运行这个函数，就可以得到训练结果result.csv。

```
[python]
01. def handwritingClassTest():
02.     trainData,trainLabel=loadTrainData()
03.     testData=loadTestData()
04.     testLabel=loadTestResult()
05.     m,n=shape(testData)
06.     errorCount=0
07.     resultList=[]
08.     for i in range(m):
09.         classifierResult = classify(testData[i], trainData, trainLabel)
10.         resultList.append(classifierResult)
11.         print "the classifier came back with: %d, the real answer : %d" % (classifierResult, testLabel[0,i])
12.         if (classifierResult != testLabel[0,i]): errorCount += 1.0
13.     print "\nthe total number of errors is: %d" % errorCount
14.     print "\nthe total error rate is: %f" % (errorCount/float(m))
15.     saveResult(resultList)
```

运行这个函数，可以得到result.csv文件：

	A	B
1	2	2
2	0	0
3	9	9
4	9	9
5	3	3
6	7	7
7	0	0
8	3	3
9	0	0
10	3	3
11	5	5
12	7	7
13	4	4
14	0	0
15	4	4
16	5	5
17	3	3
18	1	1
19	9	9
20	0	0
21	9	9
22	1	1
23	1	1
24	5	5
25	7	7
26	4	4
27	2	2

关闭

2 0 9 9 3 7 0 3.....就是每个图片对应的数字。与参考结果knn_benchmark.csv比较一下：


```
Python 1
the classifier came back with: 2, the real answer is: 2
the classifier came back with: 2, the real answer is: 2
the classifier came back with: 9, the real answer is: 9
the classifier came back with: 6, the real answer is: 6
the classifier came back with: 7, the real answer is: 7
the classifier came back with: 6, the real answer is: 6
the classifier came back with: 1, the real answer is: 1
the classifier came back with: 9, the real answer is: 9
the classifier came back with: 7, the real answer is: 7
the classifier came back with: 9, the real answer is: 9
the classifier came back with: 7, the real answer is: 7
the classifier came back with: 3, the real answer is: 3
the classifier came back with: 9, the real answer is: 9
the classifier came back with: 2, the real answer is: 2

the total number of errors is: 1004
the total error rate is: 0.035857
```

28000个样本中有1004个与kknns_benchmark.csv中的不一样。错3.5%，这个效果并不好，原因是我并未将所有训练样本都拿来训练，间，我只取一半的训练样本来训练，即上面的结果对应的代码是：

```
[python]
01. classifierResult = classify(testData[i], trainData[0:20000], trainLabel[0:20000], 5
```

训练一半的样本，程序跑了将近70分钟（在个人PC上）。

• 提交结果

将result.csv整理成kknns_benchmark.csv那种格式，即加入第一行文字说明，加入第一列的图片序号，然后make a submission，结果准确率96.5%：

314	30	Steve Shank	0.96557	2	Fri, 05 Dec 2014 18:34:04 (-0.8h)
315	30	raito	0.96557	4	Sun, 07 Dec 2014 03:49:52 (-11.6h)
316	30	wepon	0.96557	3	Sun, 14 Dec 2014 14:34:44 (-7.4d)
317	new	chiwei	0.96557	1	Tue, 09 Dec 2014 07:06:53
318	new	Stan Valchek	0.96557	1	Thu, 11 Dec 2014 18:54:04

下载工程代码：[github地址](#)

【完】

关闭

顶

78

踩

1

- 上一篇 【Python】 csv模块的使用
- 下一篇 【leetcode 桶排序】 Maximum Gap

相关文章推荐

• Kaggle竞赛入门教程之Kaggle简介（新手向）

• 【机器学习算法实现】kNN算法_手写识别——基...

• 【机器学习算法实现】logistic回归_基于Python和...

• Kaggle 首战拿银总结 | 入门指导 (长文、干货)

• 从0到1走进 Kaggle

• Kaggle如何入门？

• Kaggle入门

• kaggle入门（python数据处理）

• 滴，学生卡！--Kaggle入门

• 参加kaggle竞赛是怎样一种体验？



猜你在找

C语言及程序设计（讲师：贺利坚）

Python爬虫工程师培养课程全套（讲师

Python全栈开发入门与实战课（讲师：李杰）

2017软考网络规划设计师视频套餐（讲师

2017软考软件设计师视频套餐（讲师：任铎）

2017软考-信息系统项目管理师视频套餐（讲师：江峰）

软考(高级)项目经理实战营（讲师：张传波）

微信公众平台开发套餐（讲师：刘运强）

深度学习原理+实战+算法+主流框架套餐（讲师：唐宇迪）

2017系统集成项目管理工程师通关套餐（讲师：徐朋）

查看评论

okcing

22楼 2017-03-29 11:19发表

十分感谢您的文章！

haotian933

21楼 2017-03-10 13:46发表

我顶回一个，弥补你一半的罪过

tomperfect

20楼 2017-03-09 10:59发表

博主，十分抱歉，一不小心点到了踩，本来想再点一下能撤销，结果第二下变成了踩两次。。F5刷新之后就改不回来了。。。顶两下，十分对不住！

haotian933

Re: 2017-03-10 13:47发表

回复tomperfect：我顶回一个，弥补你一半的罪过

tomperfect

Re: 2017-03-14 08:44发表

回复haotian933：多谢！

rgfreedom

19楼 2017-03-08 21:51发表

真棒！这是我见过写的最好的Kaggle入门博客。

屋卡

18楼 2017-02-28 10:53发表

好厉害，今天找到了个好网站

tengfei461807914

17楼 2016-11-05 12:13发表

哇

TaoHuang2013

16楼 2016-10-27 02:18发表

关闭



不错，想知道你对kaggle的评价



猫猫与橙子

15楼 2016-10-02 20:10发表

你好！请问你可以下载kaggle比赛上的数据吗？我想要一份人脸表情数据，但是网络链接不过去，不知道是怎么回事？博主能不能给我传一份（1014360228@qq.com），谢谢



桂花菜籽

Re: 2016-12-07 09:51发表

回复qq_22764813：你好，我也苦于无法下载数据文件，能否也发我一份。邮箱：915866177@qq.com,谢谢。



猫猫与橙子

Re: 2017-01-03

回复GuiHuaCaiZi：不好意思，我也没方法下载！



qiaoduoxi7834

14楼 2016-09-20 23:18发表

代码是Py3还是Py2啊？



fengxuezhiling

13楼 2016-08-21 15:57发表

博主kaggle怎么注册不了



THEONE10211024

12楼 2016-08-10 15:26发表

博主你好，我感觉在个人电脑上训练太耗时，有什么比较高效的办法没？



sherlockzoom

11楼 2016-03-19 00:53发表

数据读取，可以使用pandas。



BoT-Win

10楼 2016-02-27 22:10发表

请问博主为什么我在Get the Data里只能看到train.csv和test.csv呢，并没有knn_benchmark.csv



琳小白

Re: 2016-03-06 20:41发表

回复ss123687400：我也是



AC_XXZ

9楼 2015-12-09 19:57发表

您好，怎么提交啊，一直不对
将result.csv整理成kkn_benchmark.csv那种格式，即加入第一行文字说明
我是这样子的。。。
能配个图说明下吗？



oxuzhenyi

Re: 2017-02-21 23:20发表

回复u013445530：knn_benchmark.csv要下载，见github的连接https://github.com/clytwynec/digit_recognition/blob/master/data/knn_benchmark.csv



子辰曦

8楼 2015-07-22 15:26发表

博主，您好，我运行了一下你的读文件代码，好像光把train.csv 文件读进内存就要花 186秒吧。请问，读文件真的这么慢吗？

sinat_32229987

Re: 2015-10-22 20:51发表

关闭

	回复u012675539：我的也要130秒。因为为：使用了Numpy的数组，依然循环每个元素。如果通过使用numpy内置函数，可以缩短至13秒	
	dataalking 很清晰细致的介绍，打算去kaggle瞅瞅。	7楼 2015-05-19 23:30发表
	yl11525 选用小样本量是否应该这么写： <code>classify(testData[i], trainData[0:5000], trainLabel.transpose()[0:5000])</code>	6楼 2015-04-09 20:48发表
	emily_menrath 你好,好开心看到这个博客,我特意注册了个账号,为了能在这发言,哈哈. 我大概几个月前就在kaggle注册了,然后断断续续的尝试了一些项目,不过感觉你好像比我注册的早哦. 哈哈...你平时用什么用的比较多啊? 是R 还是python,我一般都是R. 不过最近感觉,想练练python, R解决big data 问题太棘手了,经常给我转个1个小时 出结果,我都能抄盘菜了.. ..呵呵,我最近在做telematic那个, 大家交个朋友把,要是以后有时间一起组个team 玩玩	5楼 2014-12-18
	wepon_ 回复emily_menrath：你好！Kaggle我也是最近才开始玩的，我现在大部分精力放在学习算法上，然后上Kaggle练练手，严格来说也是Kaggle新手～～也有大神说，解决Kaggle上的问题最重要不在于机器学习算法，在于数据处理....Anyway，我觉得算法还是很重要的～～我用python比较多，没用过R，上手python很快的，有时间我也学学R。我现在周围都没人玩Kaggle，我们有机会确实可以组个team	Re: 2014-12-18 11:59发表
	打湿井盖 回复u012162613：可不可以带上我，超级感兴趣，我也是主要研究算法，研究了一年多的SRC，用matlab，因为DL太火，观望了一年多压力山大，最近弄python 所以来到这，收获很大，wepon能否交个朋友，我也是研究CV,ML	Re: 2015-05-13 11:45发表
	lipeng1993 python你用的是哪个软件	4楼 2014-12-17 19:25发表
	wepon_ 回复fohho：哪个软件都行吧，在linux上的话我直接用系统自带的，再装numpy、scikit-learn等库，就可以了。你要是嫌麻烦直接安装pythonxy，什么包都一次性装好了	Re: 2014-12-17 19:28发表
	lipeng1993 这些程序需要在什么软件上运行？	3楼 2014-12-17 19:24发表
	wepon_ 补充：trainLabel其实是list，里面只有一个元素，这个元素也是list，即trainLabel=[[2,0,9,0.....]]。故trainLabel[0]就表示[2,0,9,0.....]，trainLabel[0,0:20000]就表示[2,0,9,0.....]的前20000个	2楼 2014-12-17 01:04发表
	wepon_ 文章最后提到，我只用前20000个训练样本，传入的label应该改为trainLabel[0,0:20000]，因为trainLabel是行向量，写成trainLabel[0:200	1楼 2014-12-17 00:04发表

关闭

00]的话其实还是传入了1*420000的向量到classify()函数，侥幸的是classify()函数不会因此而出错。但严谨一点比较好。【结论：哪些是行向量、哪些是列向量，要分清，因为这些算法都是跟矩阵打交道，稍不注意就出错，还错得很隐秘】



开拓者V587

Re: 2015-07-28 17:34发表

回复u012162613：请问，label为何变为一个行向量了，在loadTrainData()函数中，label = trainData[:,0]，这明显是一个列向量啊。难道是python自动把列向量转为行向量了？初学python和机器学习，还忘指教。



s2392735818

Re: 2015-09-18 21:06发表

回复u012191966：博主在git上托管的代码已经把这一块更正了。

发表评论

用户 名： haijunz

评论内容：



提交

* 以上用户言论只代表其个人观点，不代表CSDN网站的观点或立场

公司简介 | 招贤纳士 | 广告服务 | 联系方式 | 版权声明 | 法律顾问 | 问题报告 | 合作伙伴 | 论坛反馈

网站客服 杂志客服 微博客服 webmaster@csdn.net 400-660-0108 | 北京创新乐知信息技术有限公司 版权所有 | 江苏知之为计算机有限公司 |

江苏乐知网络技术有限公司

京 ICP 证 09002463 号 | Copyright © 1999-2017, CSDN.NET, All Rights Reserved



关闭