Python程序设计

陈远祥

chenyxmail@gmail.com

北京邮电大学 电子工程学院

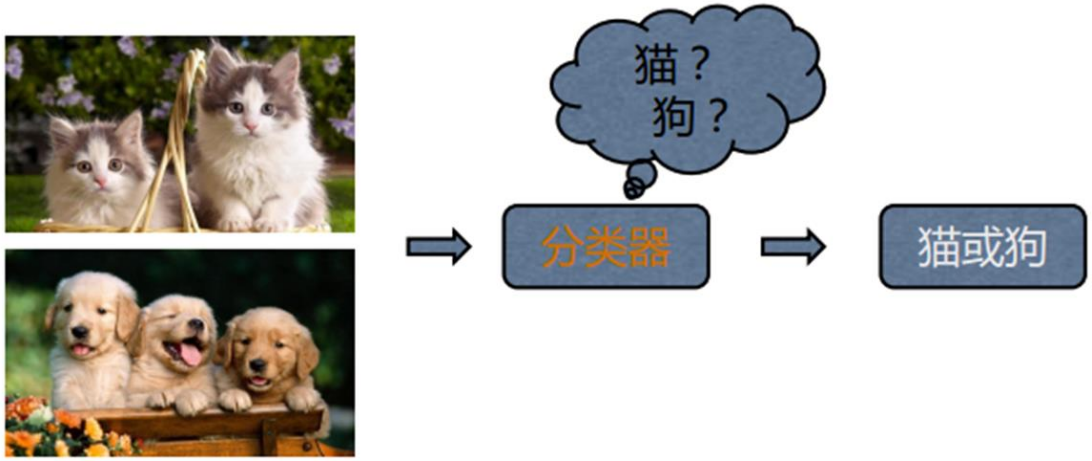
机器学习

- 机器学习分类
  - ✓ 监督学习（supervised learning）
  - ✓ 无监督学习（unsupervised learning ）
  - ✓ 强化学习（reinforcement learning，增强学习）
  - ✓ 半监督学习（semi-supervised learning ）

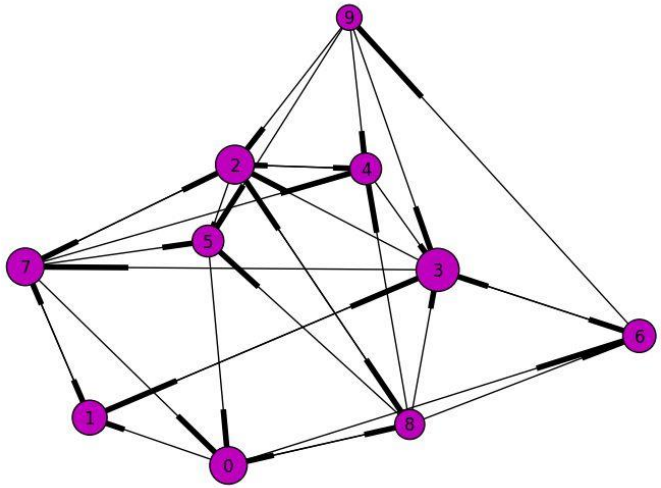
监督学习

监督学习目标

- ✓ 利用一组带有标签的数据，学习从输入到输出的映射，然后将这种映射关系应用到未知数据上，达到分类或回归的目的
- ✓ 分类：当输出是离散的，学习任务为分类任务
- ✓ 回归：当输出是连续的，学习任务为回归任务



监督学习-分类



分类

- 分类学习
  - ✓ 输入：一组有标签的训练数据(也称观察和评估)，标签表明了这些数据（观察）的所署类别
  - ✓ 输出：分类模型根据这些训练数据，训练自己的模型参数，学习出一个适合这组数据的分类器，当有新数据（非训练数据）需要进行类别判断，就可以将这组新数据作为输入送给学好的分类器进行判断

分类

- 分类学习
  - ✓ 训练集(training set)：顾名思义用来训练模型的已标注数据，用来建立模型，发现规律
  - ✓ 测试集(testing set)：也是已标注数据，通常做法是将标注隐藏，输送给训练好的模型，通过结果与真实标注进行对比，评估模型的学习能力

分类

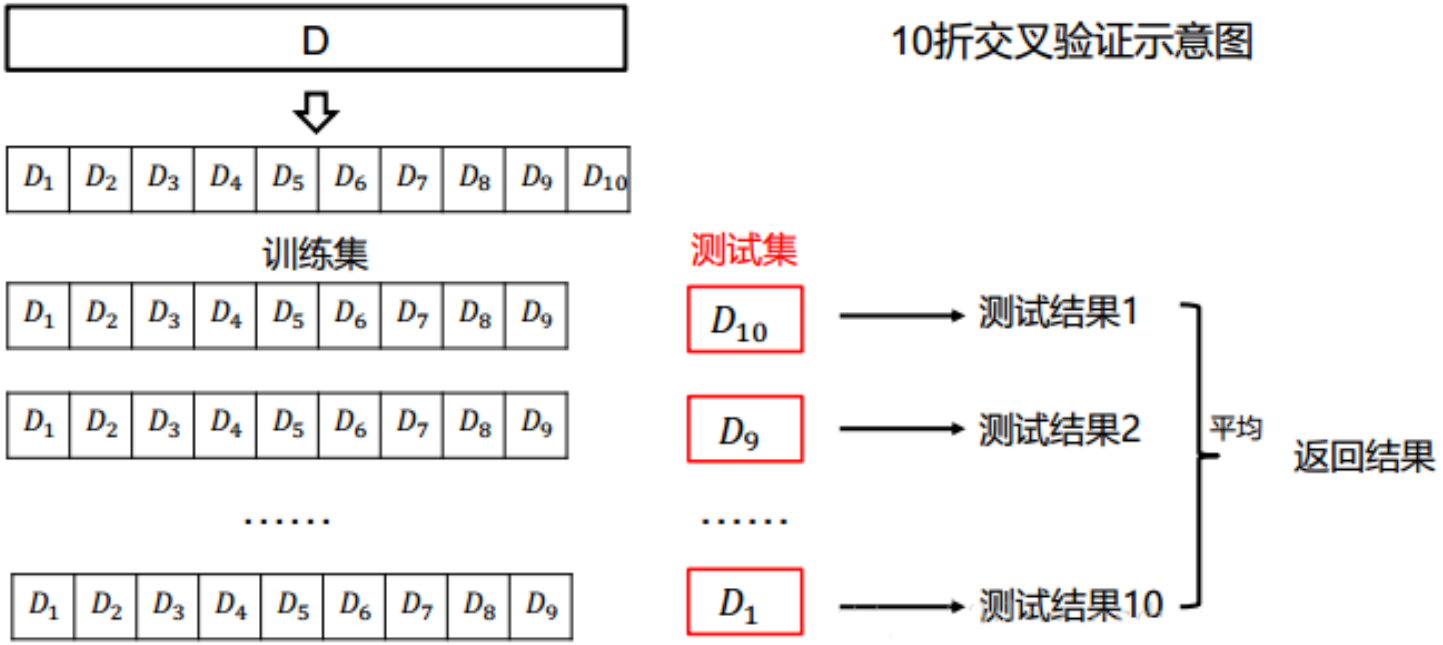
- 训练集/测试集的划分方法
  - ✓ 根据已有标注数据，随机选出一部分数据（70%）作为训练数据，余下的作为测试数据，此外还有交叉验证法，自助法用来评估分类模型

分类

- 训练集/测试集的划分方法
  - ✓ 交叉验证法先将数据集D划分为k个大小相似的互斥子集，每个自己都尽可能保持数据分布的一致性，即从D中通过分层采样得到。然后，每次用k-1个子集的并集作为训练集，余下的那个子集作为测试集；这样就可获得k组训练/测试集，从而可进行k次训练和测试，最终返回的是这个k个测试结果的均值

分类

- 训练集/测试集的划分方法
  - ✓ 通常把交叉验证法称为“k者交叉验证”，k最常用的取值是10，此时称为10折交叉验证



分类

分类学习-评价标准

- ✓ 精确率：是针对我们预测结果而言的，（以二分类为例）它表示的是预测为正的样本中有多少是真正的正样本。那么预测为正就有两种可能了，一种就是把正类预测为正类(TP)，另一种就是把负类预测为正类(FP)，也就是：P=TP/(TP+FP)

分类

- 分类学习-评价标准
  - ✓ 召回率：是针对我们原来的样本而言的，它表示的是样本中的正例有多少被预测正确。那也有两种可能，一种是把原来的正类预测成正类(TP)，另一种就是把原来的正类预测为负类(TN)，也就是：P=TP/(TP+TN)

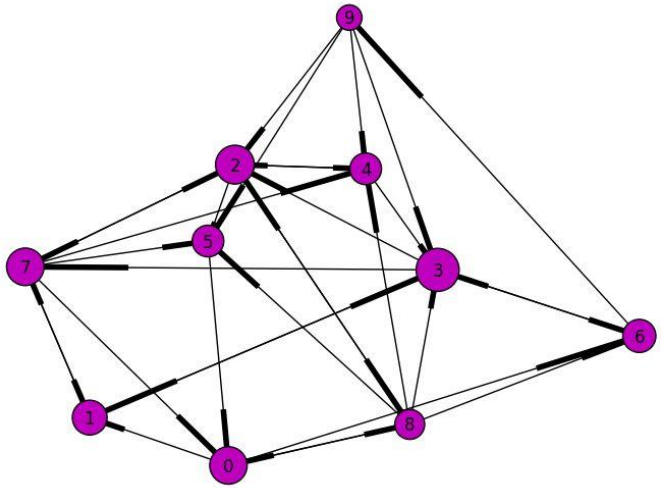
分类

- 分类学习-评价标准
  - ✓ 举例：假设我们手上有60个正样本，40个负样本，我们要找出所有的正样本，分类算法查找出50个，其中只有40个是真正的正样本
  - ✓ TP：将正类预测为正类数40；TN：将正类预测为负类数20；FP：将负类预测为正类数10；FN：将负类预测为负类数30
  - ✓ 准确率（accuracy）=预测对的/所有=(TP+FN)/(TP+FN+FP+TN) =(40+30)/100=70%
  - ✓ 精确率（precision）=TP/(TP+FP)=40/(50)=80%
  - ✓ 召回率（recall）=TP/(TP+TN)=40/60=66.7%

分类

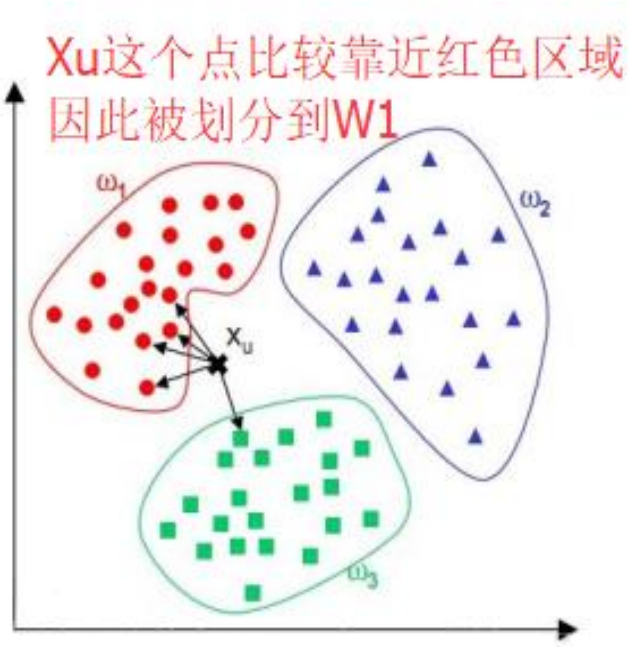
- 分类学习应用
  - ✓ 金融：贷款是否批准进行评估
  - ✓ 医疗诊断：判断一个肿瘤是恶性还是良性
  - ✓ 欺诈检测：判断一笔银行的交易是否涉嫌欺诈
  - ✓ 网页分类：判断网页的所属类别，财经或者是娱乐？

分类学习算法举例



分类学习算法举例

- K近邻分类器(KNN)
  - ✓ KNN：通过计算待分类数据点，与已有数据集中的所有数据点的距离。取距离最小的前K个点，根据“少数服从多数“的原则，将这个数据点划分为出现次数最多的那个类别



分类学习算法举例

- sklearn中的K近邻分类器
  - ✓ 在sklearn库中，可以使用sklearn.neighbors.KNeighborsClassifier创建一个K近邻分类器，主要参数有：
  - ✓ n_neighbors：用于指定分类器中K的大小(默认值为5)
  - ✓ weights：设置选中的K个点对分类结果影响的权重（默认值为平均权重“uniform”，可以选择“distance”代表越近的点权重越高，或者传入自己编写的以距离为参数的权重计算函数）
  - ✓ algorithm：设置用于计算临近点的方法（选项中有ball_tree、 kd_tree和brute，分别代表不同的寻找邻居的优化算法，默认值为auto，根据训练数据自动选择）

分类学习算法举例

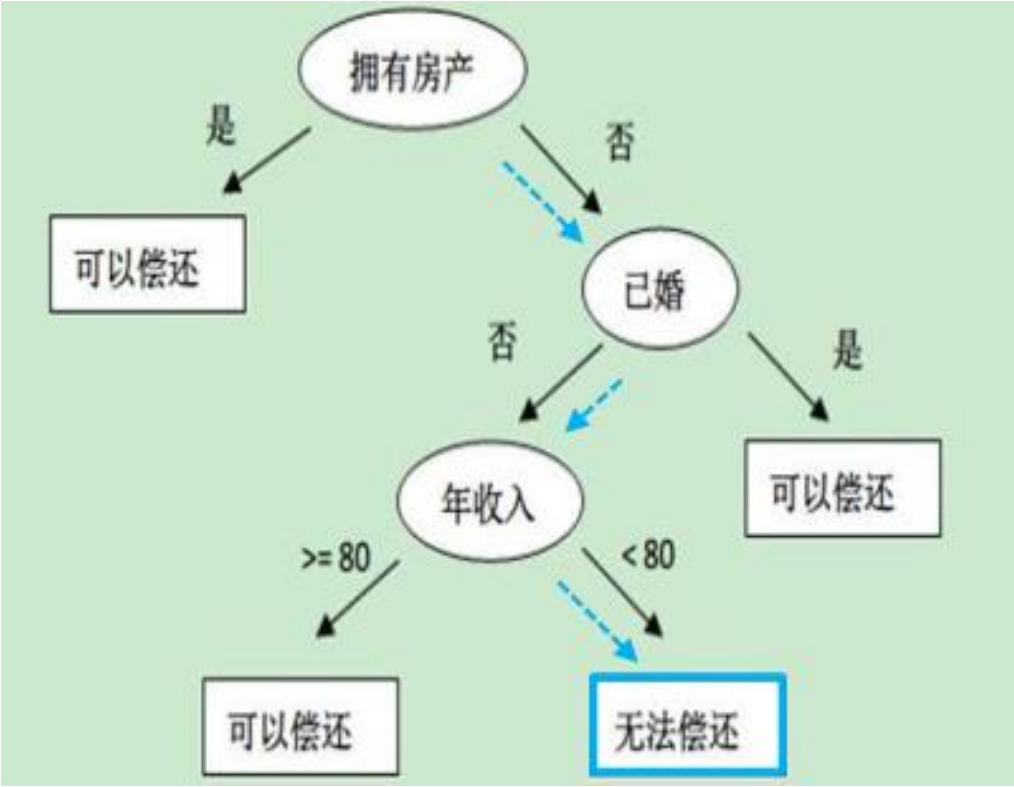
- sklearn中的K近邻分类器
  - ✓ #knn-pre

分类学习算法举例

- 决策树
  - ✓ 决策树是一种树形结构的分类器，通过顺序询问分类点的属性决定分类点最终的类别。通常根据特征的信息增益或其他指标，构建一颗决策树。在分类时，只需要按照决策树中的结点依次进行判断，即可得到样本所属类别

分类学习算法举例

- 决策树
  - ✓ 举例：信用卡偿还能力决策树



分类学习算法举例

- sklearn中的决策树
  - ✓ 在sklearn库中，可以使用sklearn.tree.DecisionTreeClassifier创建一个决策树用于分类，其主要参数有：
  - ✓ criterion：用于选择属性的准则，可以传入“gini”代表基尼系数，或者“entropy”代表信息增益
  - ✓ max_features：表示在决策树结点进行分裂时，从多少个特征中选择最优特征。可以设定固定数目、百分比或其他标准。它的默认值是使用所有特征个数

分类学习算法举例

- 决策树
  - ✓ #decisiontree.py

分类学习算法举例

- 朴素贝叶斯
  - ✓ 朴素贝叶斯分类器是一个以贝叶斯定理为基础的多分类的分类器
  - ✓ 对于给定数据，首先基于特征的条件独立性假设，学习输入输出的联合概率分布，然后基于此模型，对给定的输入x，利用贝叶斯定理求出后验概率最大的输出y

$$p(A|B) = \frac{p(B|A) \cdot p(A)}{p(B)}$$

分类学习算法举例

- sklearn中的朴素贝叶斯
  - ✓ 在sklearn库中，实现了三个朴素贝叶斯分类器，如下表所示

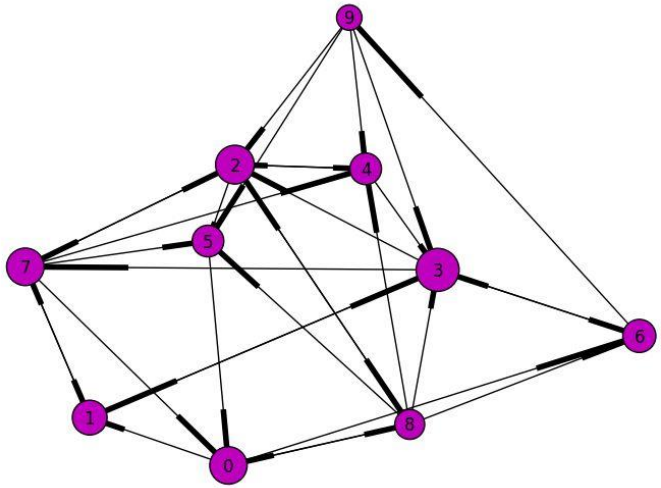
分类器	描述
naive_bayes.GussianNB	高斯朴素贝叶斯
naive_bayes.MultinomialNB	针对多项式模型的朴素贝叶斯分类器
naive_bayes.BernoulliNB	针对多元伯努利模型的朴素贝叶斯分类器

  - ✓ 区别在于假设某一特征的所有属于某个类别的观测值符合特定分布，如，分类问题的特征包括人的身高，身高符合高斯分布，这类问题适合高斯朴素贝叶斯

分类学习算法举例

- 朴素贝叶斯
  - ✓ #gaussian_nb.py

分类-人体运动状态预测



人体运动状态预测

- 背景介绍
  - ✓ 可穿戴式设备的流行，让我们可以更便利地使用传感器获取人体的各项数据，甚至生理数据
  - ✓ 当传感器采集到大量数据后，我们就可以通过对数据进行分析和建模，通过各项特征的数值进行用户状态的判断，根据用户所处的状态提供给用户更加精准、便利的服务

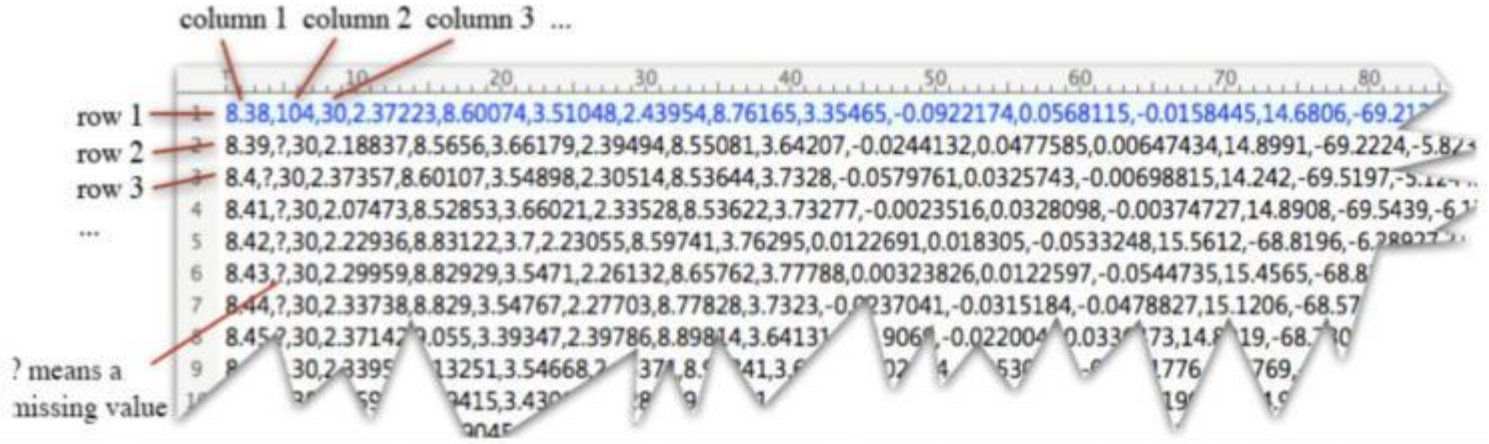
人体运动状态预测

- 数据介绍
  - ✓ 我们现在收集了来自 A,B,C,D,E 5位用户的可穿戴设备上的传感器数据，每位用户的数据集包含一个特征文件（a.feature）和一个标签文件（a.label）

人体运动状态预测

数据介绍

- ✓ 特征文件中每一行对应一个时刻的所有传感器数值，标签文件中每行记录了和特征文件中对应时刻的标记过的用户姿态，两个文件的行数相同，相同行之间互相对应
- ✓ ?号表示缺失值，特征文件共包含41列特征



1	2	3-15	16-28	29-41
时间戳	心率	传感器1	传感器2	传感器3

人体运动状态预测

- 数据介绍-feature
  - ✓ 在传感器1对应的13列数据特征中，包含：1项温度数据、3项一型三轴加速度数据、3项二型三轴加速度数据、3项三轴陀螺仪数据和3项三轴磁场数据
  - ✓ 人体的温度数据可以反映当前活动的剧烈程度，一般在静止状态时，体温趋于稳定在36.5度上下；当温度高于37度时，可能是进行短时间的剧烈运动，比如跑步和骑行

3	4-6	7-9	10-12	13-15
温度	一型三轴加速度	二型三轴加速度	三轴陀螺仪	三轴磁场

人体运动状态预测

数据介绍-feature

- 在数据中有两个型号的加速度传感器，可以通过互相印证的方式，保证数据的完整性和准确性。通过加速度传感器对应的三个数值，可以知道空间中x、y、z三个轴上对应的加速度，而空间上的加速度和用户的姿态有密切的关系，比如用户向上起跳时，z轴上的加速度会激增。

1	2	3-15	16-28	29-41
时间戳	心率	传感器1	传感器2	传感器3

3	4-6	7-9	10-12	13-15
温度	一型三轴加速度	二型三轴加速度	三轴陀螺仪	三轴磁场

人体运动状态预测

- 数据介绍-feature
  - ✓ 陀螺仪是角运动检测的常用仪器，可以判断出用户佩戴传感器时的身体角度是水平、倾斜还是垂直。直观地，通过这些数值都是推断姿态的重要指标

1	2	3-15	16-28	29-41
时间戳	心率	传感器1	传感器2	传感器3

3	4-6	7-9	10-12	13-15
温度	一型三轴加速度	二型三轴加速度	三轴陀螺仪	三轴磁场

人体运动状态预测

数据介绍-feature

- 磁场传感器可以检测用户周围的磁场强度和数值大小，这些数据可以帮助我们理解用户所处的环境。比如在一个办公场所，用户座位附近的磁场是大体上固定的，当磁场发生改变时，我们可以推断用户的位置和场景发生了变化

1	2	3-15	16-28	29-41
时间戳	心率	传感器1	传感器2	传感器3

3	4-6	7-9	10-12	13-15
温度	一型三轴加速度	二型三轴加速度	三轴陀螺仪	三轴磁场

人体运动状态预测

- 数据介绍-label
  - ✓ 标签文件内容每一行代表与特征文件中对应行的用户姿态类别。总共有0-24共25种身体姿态，如，无活动状态，坐态、跑态等。标签文件作为训练集的标准参考准则，可以进行特征的监督学习

人体运动状态预测

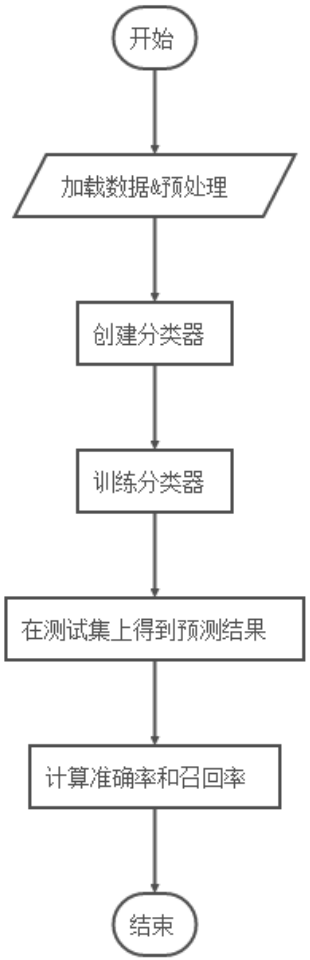
- 任务介绍
  - ✓ 假设现在出现了一个新用户，但我们只有传感器采集的数据，那么该如何得到这个新用户的姿态呢？
  - ✓ 又或者对同一用户如果传感器采集了新的数据，怎么样根据新的数据判断当前用户处于什么样的姿态呢？

人体运动状态预测

- 算法流程
  - ✓ 需要从特征文件和标签文件中将所有数据加载到内存中，由于存在缺失值，此步骤还需要进行简单的数据预处理
  - ✓ 创建对应的分类器，并使用训练数据进行训练
  - ✓ 利用测试集预测，通过使用真实值和预测值的比对，计算模型整体的准确率和召回率来评测模型

人体运动状态预测

- 算法流程



✓ #state_predict.py

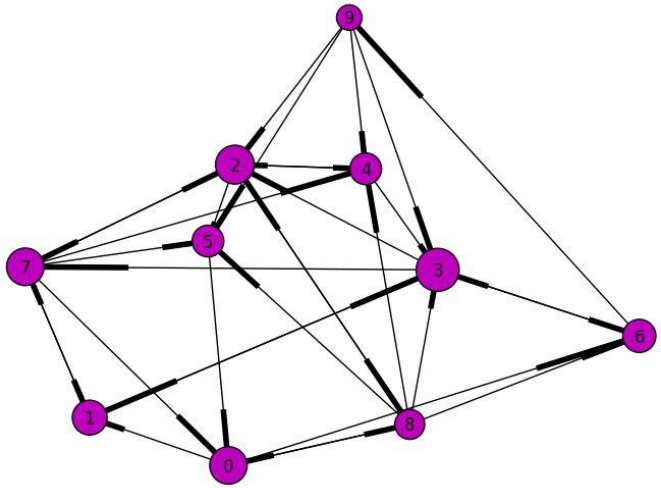
人体运动状态预测

- 结果对比：

模型	K近邻	决策树	贝叶斯
准确度	0.69	0.66	0.74
召回率	0.69	0.66	0.68
F1值	0.68	0.62	0.67

- 结论：
  - ✓ 从准确度的角度衡量，贝叶斯分类器的效果最好
  - ✓ 从召回率和F1值的角度衡量，k近邻效果最好
  - ✓ 贝叶斯分类器和k近邻的效果好于决策树

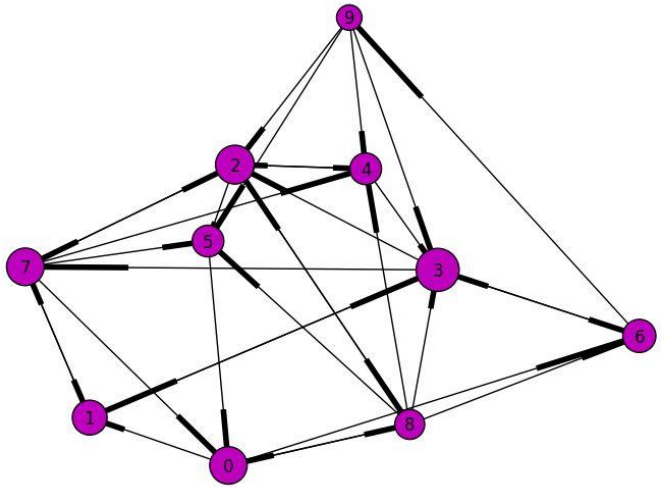
分类-上证指数涨跌预测



上证指数涨跌预测

- 数据介绍：
  - ✓ 网易财经上获得的上证指数的历史数据
  - ✓ 实验目的：根据给出当前时间前150天的历史数据，预测当天上证指数的涨跌
  - ✓ 技术路线：sklearn.svm.SVC
  - ✓ 数据实例：中国石油股票数据
  - ✓ #stock_pre

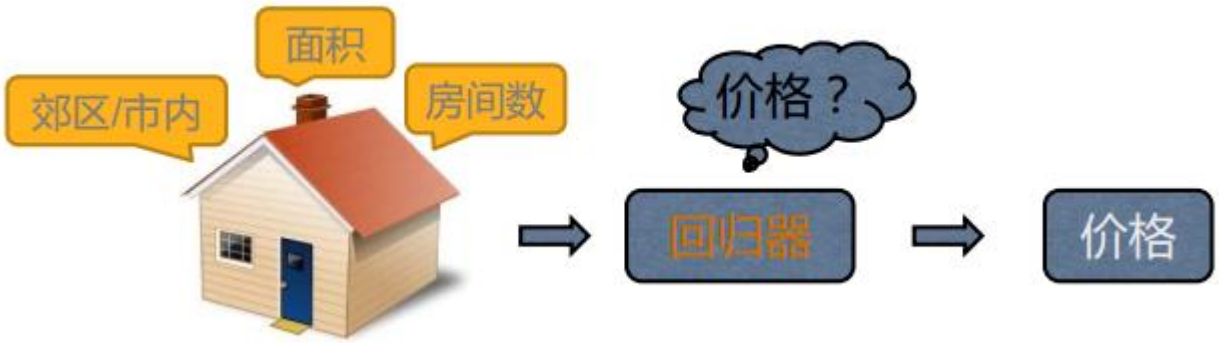
监督学习-回归分析



回归分析

回归分析

- ✓ 回归：统计学分析数据的方法，目的在于了解两个或多个变数间是否相关、研究其相关方向与强度，并建立数学模型以便观察特定变数来预测研究者感兴趣的变数
- ✓ 回归分析可以帮助人们了解在自变量变化时因变量的变化量。一般来说，通过回归分析我们可以由给出的自变量估计因变量的条件期望



回归分析

- 回归分析应用
  - ✓ 回归方法适合对一些带有时序信息的数据进行预测或者趋势拟合，常用在金融及其他涉及时间序列分析的领域：
  - ✓ 股票趋势预测
  - ✓ 交通流量预测
  - ✓ 房价预测
  - ✓ 。。。

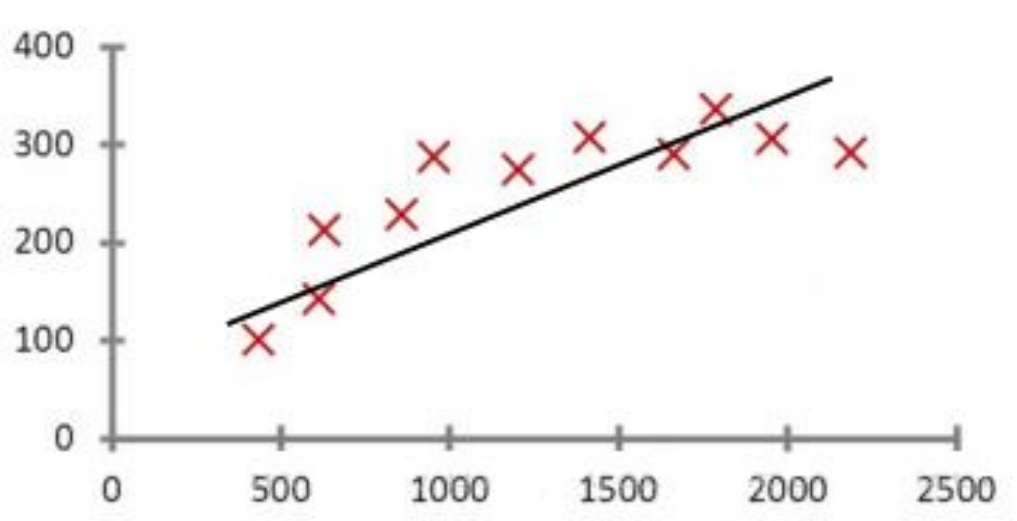
回归分析

- sklearn中的回归算法
  - ✓ Sklearn提供的回归函数主要被封装在两个子模块中，分别是sklearn.linear_model和sklearn.preprocessing
  - ✓ sklearn.linear_modlel封装的是一些线性函数，线性回归函数包括有：普通线性回归函数（LinearRegression），岭回归（Ridge），Lasso（Lasso）
  - ✓ 非线性回归函数，如多项式回归（PolynomialFeatures）则通过sklearn.preprocessing子模块进行调用

回归分析

线性回归

- 线性回归(Linear Regression)是利用数理统计中回归分析，来确定两种或两种以上变量间相互依赖的定量关系的一种统计分析方法，其使用形如y=wTx+b的线性模型拟合数据输入和输出之间的映射关系的



回归分析

- 线性回归
  - ✓ 线性回归利用称为线性回归方程的最小平方函数对一个或多个自变量和因变量之间关系进行建模。这种函数是一个或多个称为回归系数的模型参数的线性组合
  - ✓ 只有一个自变量的情况称为简单回归，大于一个自变量情况的叫做多元回归

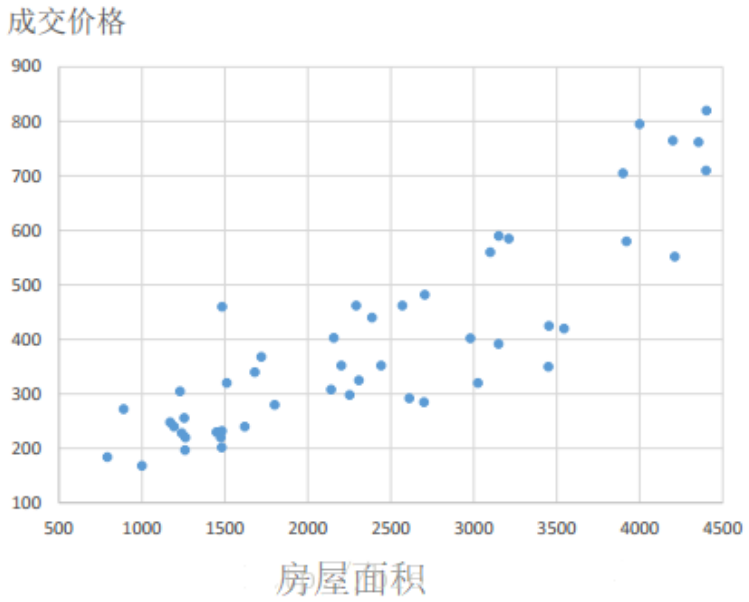
回归分析

线性回归-实际用途

- ✓ 利用数据拟合模型进行预测:如果目标是预测或者映射，线性回归可以用来对观测数据集的y和X的值拟合出一个预测模型。当完成这样一个模型以后，对于一个新增的X值，在没有给定与它相配对的y的情况下，可以用这个拟合过的模型预测出一个y值
- ✓ 相关性分析去除冗余:给定一个变量y和一些变量X1-Xn，这些变量有可能与y相关，线性回归分析可以用来量化y与Xi之间相关性的强度，评估出与y不相关的Xi，并识别出哪些Xi的子集包含了关于y的冗余信息

回归分析

- 房价与房屋尺寸关系的线性拟合
  - ✓ 背景：我们可以根据已知的房屋成交价和房屋的尺寸进行线性回归，继而可以对已知房屋尺寸，而未知房屋成交价格的实例进行成交价格的预测
  - ✓ 目标：对房屋成交信息建立回归方程，并依据回归方程对房屋价格进行预测



回归分析

- 房价与房屋尺寸关系的线性拟合
  - ✓ 技术路线：sklearn.linear_model.LinearRegression
  - ✓ 调用sklearn.linear_model.LinearRegression()所需参数：
  - ✓ fit_intercept ：布尔型参数，表示是否计算该模型截距。可选参数
  - ✓ normalize ：布尔型参数，若为True，则X在回归前进行归一化。可选参数。默认值为False
  - ✓ copy_X ：布尔型参数，若为True，则X将被复制；否则将被覆盖。 可选参数。默认值为True
  - ✓ n_jobs ：整型参数，表示用于计算的作业数量；若为-1，则用所有的CPU。可选参数。默认值为1

回归分析

- 房价与房屋尺寸关系的线性拟合
  - ✓ #pricepre_line

回归分析

- 非线性拟合：多项式回归
  - ✓ 多项式回归(Polynomial Regression)是研究一个因变量与一个或多个自变量间多项式的回归分析方法
  - ✓ 如果自变量只有一个时，称为一元多项式回归；如果自变量有多个时，称为多元多项式回归

$$\hat{y} = b_0 + b_1 x + b_2 x^2 + \cdots + b_m x^m$$

$$\hat{y} = b_0 + b_1 x_1 + b_2 x_2 + b_3 x_1^2 + b_4 x_2^2 + b_5 x_1 x_2$$

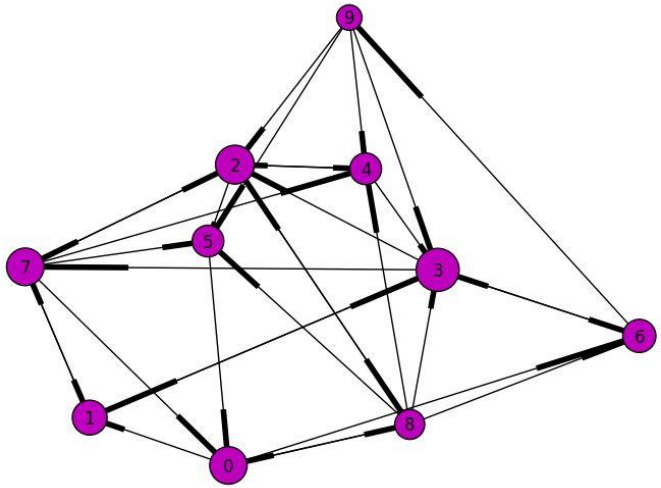
回归分析

- 什么时候用多项式回归
  - ✓ 在一元回归分析中，如果依变量y与自变量x的关系为非线性的，但是又找不到适当的函数曲线来拟合，则可以采用一元多项式回归
  - ✓ 多项式回归的最大优点就是可以通过增加x的高次项对实测点进行逼近，直至满意为止
  - ✓ 事实上，多项式回归可以处理相当一类非线性问题，它在回归分析中占有重要的地位，因为任一函数都可以分段用多项式来逼近

回归分析

- 多项式拟合-技术路线：
  - ✓ sklearn.preprocessing.PolynomialFeatures
  - ✓ sklearn中的多项式回归实际上是先将变量X处理成多项式特征，然后使用线性模型学习多项式特征的参数，以达到多项式回归的目的
  - ✓ 例如：X = [x1, x2]
  - ✓ 1.使用PolynomialFeatures构造X的二次多项式特征X_Poly：X_Poly=[x1, x2, x1x2, $x1^2$, $x2^2$]
  - ✓ 2.使用linear_model学习X_Poly和y之间的映射关系，即参数：y=[w1x1 + w2x2 + w3x1x2 + w4$x1^2$ + w5$x2^2$]

手写数字识别实例分析



手写数字识别实例分析

- 图像识别：
  - ✓ 图像识别（Image Recognition）是指利用计算机对图像进行处理、分析和理解，以识别各种不同模式的目标和对像的技术
  - ✓ 图像识别的发展经历了三个阶段：文字识别、数字图像处理与识别、物体识别。机器学习领域一般将此类识别问题转化为分类问题

手写数字识别实例分析

- 手写识别：
  - ✓ 手写识别是常见的图像识别任务。计算机通过手写体图片来识别出图片中的字，与印刷字体不同的是，不同人的手写体风格迥异，大小不一，造成了计算机对手写识别任务的一些困难
  - ✓ 数字手写体识别由于其有限的类别（0~9共10个数字，因此也是一个多分类任务）成为了相对简单的手写识别任务。 DBRHD和MNIST是常用的两个数字手写识别数据集

手写数字识别实例分析

- 关于MNIST
  - ✓ MNIST 的 下 载 链 接 ： http://yann.lecun.com/exdb/mnist/
  - ✓ MNIST是一个包含数字0~9的手写体图片数据集，图片已归一化为以手写数字为中心的28*28规格的图片（每一个图片由28*28个像素点组成，每个像素点的值区间为0~255，0表示白色，255表示黑色。）

手写数字识别实例分析

- 关于MNIST
  - ✓ MNIST由训练集与测试集两个部分组成，各部分规模如下：
  - ✓ 训练集：60,000个手写体图片及对应标签
  - ✓ 测试集：10,000个手写体图片及对应标签

手写数字识别实例分析

- MNIST数据结构
  - ✓ 在MNIST数据集中的每张图片由28x28个像素点构成，每个像素点用一个灰度值表示。在MNIST，将28x28的像素展开为一个一维的行向量，这些行向量就是图片数组里的行(每行784个值,或者说每行就是代表了一张图片)
  - ✓ #data_read.py

手写数字识别实例分析

- 手写识别研究现况
  - ✓ 已有许多模型在MNIST数据集上进行了实验，有些模型对数据集进行了偏斜矫正，甚至在数据集上进行了人为的扭曲、偏移、缩放及失真等操作以获取更加多样性的样本，使得模型更具有泛化性
- 常用于数字手写体的分类器：
  - ✓ 线性分类器
  - ✓ K最近邻分类器
  - ✓ Boosted Stumps
  - ✓ 非线性分类器
  - ✓ SVM
  - ✓ 多层感知器
  - ✓ 卷积神经网络

手写数字识别实例分析

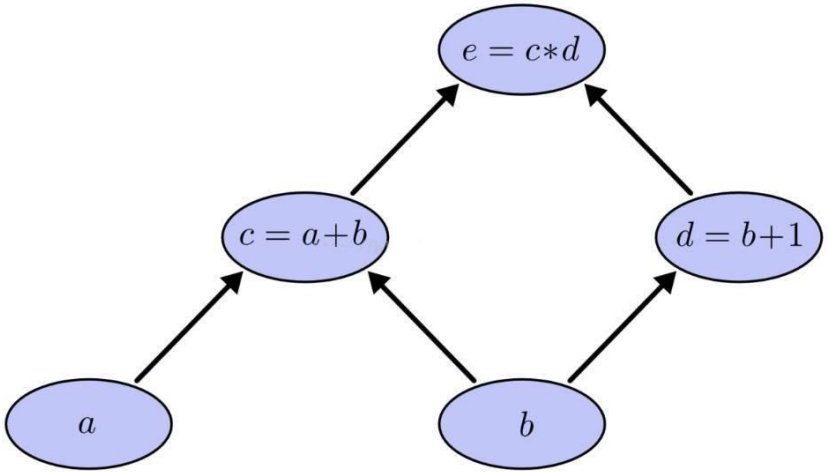
- tensorflow是什么？
  - ✓ tensorflow是google开源的机器学习工具，在2015年11月其实现正式开源，开源协议Apache 2.0
- why tensorflow?
  - ✓ Tensorflow拥有易用的python接口，而且可以部署在一台或多台cpu,gpu上，兼容多个平台，包括但不限于安卓/windows/linux等等平台上，而且拥有tensorboard这种可视化工具，可以使用 checkpoint进行实验管理，得益于图计算，它可以进行自动微分计算，拥有庞大的社区，而且很多优秀的项目已经使用tensorflow进行开发了

手写数字识别实例分析

- **Tensorflow基础**

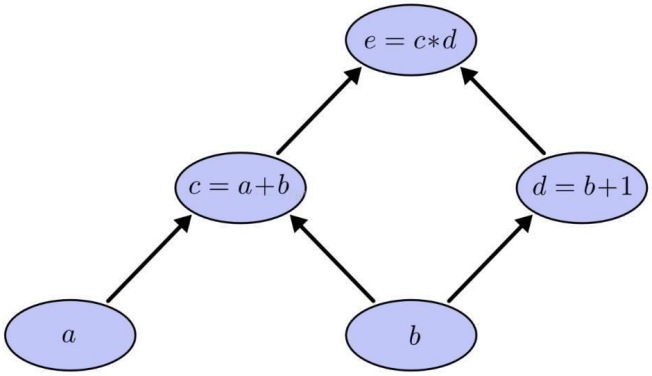
  1）计算图

  Tensorflow是基于计算图的框架，假设我们有这样一个需要计算的表达式。该表达式包括了两个加法与一个乘法，为了更好讲述引入中间变量c与d。由此该表达式可以表示为：



手写数字识别实例分析

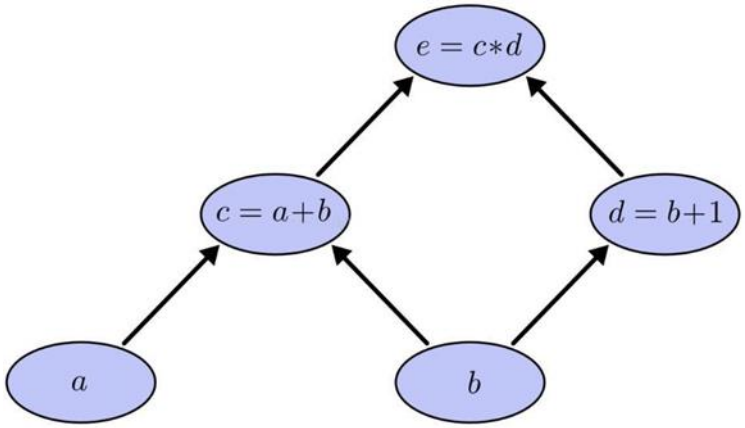
- Tensorflow基础



当需要计算e时就需要计算c与d，而计算c就需要计算a与b，计算d需要计算b。这样就形成了依赖关系。这种有向无环图就叫做计算图，因为对于图中的每一个节点其微分都很容易得出，因此应用链式法则求得一个复杂的表达式的导数就成为可能，所以它会应用在类似tensorflow这种需要应用反向传播算法的框架中

手写数字识别实例分析

- graph, session, operation, tensor四个概念
  - ✓ Graph：一张有边与点的图，其表示了需要进行计算的任务；
  - ✓ Tensor：类型化的多维数组，图的边；
  - ✓ Operation:执行计算的单元，图的节点；
  - ✓ Session:称之为会话的上下文，用于执行图



手写数字识别实例分析

- graph, session, operation, tensor
  - ✓ Graph仅仅定义了所有operation与tensor流向，没有进行任何计算。而session根据graph的定义分配资源，计算operation，得出结果
  - ✓ Operation可以是加减乘除等数学运算，也可以是各种各样的优化算法。每个operation都会有零个或多个输入，零个或多个输出
  - ✓ tensor就是其输入与输出，其可以表示一维二维多维向量或者常量。而且除了Variables指向的tensor外所有的tensor在流入下一个节点后都不再保存

手写数字识别实例分析

- 用tensorflow进行graph构建，分为五部分：
  - ✓ 为输入X与输出y定义placeholder
  - ✓ 定义权重W
  - ✓ 定义模型结构
  - ✓ 定义损失函数
  - ✓ 定义优化算法

手写数字识别实例分析

- TensorFlow笔记之MNIST例程详解
  - ✓ #tensor

谢谢