机器学习总结

一、 机器学习的定义

机器学习是一种程序，他可以让系统在未经过人为主动编程的情况下，具有经验（数据）中自动学习并且自我改进的能力。机器学习始于对数据的观察，我们给出的示例、经验数据和指导，以便于让计算机根据我们的提供的示例查找数据并做出最好的决策。机器学习主要的目的是允许计算机在没有人工干预或帮助的情况下自动学习，并做出相应的操作调整。

二、 机器学习的应用场景（常用）

（1）虚拟个人助理
（2）导航打车
（3）视频监控
（4）社交媒体

三、机器学习相关知识

1.机器学习的算法主要有两种：监督学习和无监督学习，除此之外还有半监督学习。

（1）监督式学习

通过现有训练数据集进行建模，再用模型对新的数据样本进行分类或者回归分析的机器学习算法，在监督学习中，训练集一般包含样本特征变量以及分类标签，机器使用不同的算法得出数据并通过这些数据推断出分类的方法，然后用于新的样本。

（2）无监督学习

在没有训练集的情况下，对没有标签的数据进行分析并建立合适的建模，以便给出问题解决的方法。

2.分类和回归

分类和回归都是监督式学习分类预测样本属于哪一类，而回归预测样本目标字段的数值。

3.数据集和特征

数据集是预测系统的原材料，用于训练机器学习模型的历史数据。数据集，由若干条数据组成，而每条数据又包含若干个特征。

特征是描述数据集中每个样本的属性，有的时候也成为“字段”。

4.特征工程

特征工程是创建预测模型之前的过程，在这个过程中我们将对数据的特征进行分析、清理和结构化。此过程是最重要且成本最高的部分之一，目的是消除那些不利于进行预测的特征并适当地组织他们，以防止模型收到无用信息，防止这些信息误导模型并最终导致模型的准确率下降的情况发生，简而言之，特征工程可以理解成将数据特征准备好。以便我们能够训练出更准确的模型的过程。

5.过拟合和拟合

当数据中存在大量噪声被机器学习算法捕快时就会发生过拟合。简单来说，当模型或算法对训练数据拟合的“太好”，就会发生过拟合。过拟合通常是模型过于复杂的结果，可以通过整合多个模型并在测数据上使用交叉检验比较其预测准确率，根据实际情况降低模型的复杂度来防止过拟合。

当机器学习算法无法捕获数据的潜在趋势时，就会发生欠拟合。通俗来说，当模型或算法无法很好地拟合数据时，就会发生欠拟合。

6.sklearn 的基本操作

Sklearn 是一个常见的第三方开源机器学习库，他对机器学习算法进行了封装，包含分类、回归，聚类、数据降维。Sklearn 内置了一些标准的数据集，用于分类对象的数据集，识别手写数据集。

首先从 sklearn 中导入 datasets 模块

Import sklearn import dataset

载入手写数字识别数据集，载入数据集键名并运行代码

a=datasets.load_digits()

digits.keys

三、 线性模型

1. 线性模型并非特指某一种算法，而是代表了一类算法。在机器学习的范畴中常用的线性模型包括线性回归、岭回归、、套索回归、逻辑回归和线性支持向量机。

线性模型的基本公式：y^=w[0]*x[0]+w[1]*x[1]+…+w[n]+b

在公式中，x[0]~x[n]代表数据集中每个样本的特征值，n 表示数据集样本都中的每个有 n 个特征，w 和 b 分别代表模型计算出来每个特征的权重和偏重，而 y-hat 是模型计算出来的预测值。

假设某个数据集中的样本都只有一个特征。如下：^y=w*x+b。

2. 线性回归模型的使用

基于一个简单的训练集，用最简单的线性模型，也就是线性回归（linear Regression）模型。

首先要导入线性回归模型，创建一个回归器参数默认，它相对的样本数据赋值给对象名（x,y），利用（x,y）去训练线性回归模型。为了使线性回归更在直观的展现在用户面前使用 matpotlib 图像可视化函数，利用（x,y）将数据以散点图形式展现出来。

3. 回归岭原理

L2 正则化参数：在实际过程中，如果模型的验证集中的准确率低于模型训练集中的准确率，就说明回归模型出现了过拟合问题，因此回归岭可以解决线性回归不容易通过控制模型的复杂度来降低过拟合。首先导入回归岭（from sklearn.linear_moldel import ridge），使用回归岭对数据进行拟合(fit())，并用数据可视化(matplotlib)展现出来。可以展现出与线，性模型很相似。也可以用正则化参数 alpha 进行调整，可有效避免过拟合。较高的 alpha 值代表算法对模型的限制更加严格，alpha 值越大，系数的极差越小，模型也就不容易出现过拟合现象。

4. 套索回归

套索工具可以利用 L1 正则化方法使某些特征的权重等于 0，使得模型具有自动选择特征的能力。常见的 L1 正则化线性模型的套索回归用（Lasso）导入套索回归（from sklearn.linear_model import Lasso）使用套索回归拟合数据，并返回模型参数。可以看出返回了 alpha 参数，我们分别通过训练集和测试集取出 alpha 参数来得出准确率（lasso.score(x,y)可以看出准确率很低。套索回归的讲多个特征权重降到了 0 出现了欠拟合。我们需要套索回归调整降低 alpha 参数，提高验证集和训练集的准确率。

也可以通过多次调整 alpha 参数取不同的值时套索回归模型给各个特征分配的权重如（alpha=0.001、0.1、1）来选择数量的对比。

使用不同的正则化方式的算法对比，可以得出如果数据集中样本特征较少，且每个特征都比较重要，则应该选择使用 L2 正则化的线性模型比如回归岭，反之，如果数据集中样本特征较多，且只有一部分比较重要，则应该使用 L1 正则化地线性模型如套索回归。

5. 了解逻辑回归和支持向量机

在线性模型中，还有一些用来分类任务的算法模型逻辑回归（Logistic Regression）和支持向量机（linrea SVM）在逻辑回归和支持向量机中，也有一个正则化参数调节 C，在回归岭和支持向量机中 C!=alpha,而等于 Cˆ-1。

训练逻辑回归并预测：导入逻辑回归（from sklearn.linear_model import Logistic Regression）创建分类器(clf= Logistic Regression)然后定义数据集中的（x,y）,在对 x 中的数据进行删除行列调整（axis=1(列)，0(行)），在把删除的列（行）赋值给 y 进行预测得出样本特征的结果。

训练线性支持向量机模型并预测：导入线性支持向量机（from sklearn.svm import Linear SVC）,然后再导入分类器（svc=Linear SVC()拟合数据预测数据，可以得出样本对象的特征趋势。

四、 决策树和随机森林

决策树算法：通过对样本特征进行一系列“是”或“否”的判断，进而做出决策。举例“送礼物”决策过程中，思维结构呈现树状型结构，不同的礼物被当做树的叶子，也就是数据集样本标签。而相对的任务性别，喜好特征为样本特征。

首先通过导入 pandas 库和 sklean 库的 tree 模块,用字典方式创建数据集，转化为数据框，再定义（x,y）分别作为样本特征并创建分类器（clf=tree.DecisionTreeClassifier()）,然后进行预测结果。可以帮助我们进行选择礼物的决策。根据数字（0 或 1）代表样本特征和礼物样本匹配,通过决策树是或否的判断进行最终礼物选择。

决策树中的 max_depth 参数用来设置决策树的最大层数，层数越大。模型越复杂，反之层数越小，模型越简单，而对于超高维度的数据的数据集，效率会降低。因此，有时我们需要对最大层数进行限制，也就是限制提问“是”或“否”的个数。导入 sklearn 模块中导入用于制作分类的包 make_blobs(from sklearn.datasets import make_blobs),利用 make_blobs 生成数据，分别赋值给(x,y),然后把样本数据（n_samples）,分类数量（centers），样本特征数量（n_features）分别设置一个值。通过 x.shape 查看样本数据形态。也可以通过样本数据可视化工具对散点(scatter)图片的大小位置(figure=(figsize())),样本特征,颜色色调区分标签（camp（c））,散点边缘色（edgecolors）进行图片装饰。

2.随机森林算法

随机森林算法是我们从样本中中抽取一部分来训练一棵决策树，在抽取另一部分样本训练第二课决策树，以此类推，最后把若干个决策树预测的结果取得平均值,这样就避免了单棵决策树过拟合的问题。这种整合多个模型的方法称为“集合方法”。训练多棵决策树模型，并使用他们决策结果的平均值作为最终预测的结果。因此被称为“随机森林”。导入随机森林算法（from sklearn.ensemble import RandomForestClassifier）,用 make_blobs 生成数据集并将特征赋值给（x,y）然后创建随机森林分类器，拟合样本数据（forset= RandomForestClassifier().fit(x,y)）,返回构造模型参数返回（forset）可以得出随机森林中几个参数（1）bootstrap 参数：有放回的随机森林，“自展法”

在多个样本中，可以保证我们训练每颗决策树模型都不同。(2) max_featrues 参数经过 bootstrap 参数抽样以后，随机森林的每课决策树会在样本中选择不同个数的样本特征。(3) n_estimators 参数控制的是随机森林中决策的数量，决策树的数量越多，通常随机森林所得到的结果准确率也越高，但是训练时间也更长。反之数量越少，训练时间越短，准确率也会越低。

决策树生成的模型对于不同的分界线都是直线，随机森林边界线就曲折了很多。这样看起来随机森林比决策树生成的模型复杂得多。在使用 make_blobs 生成数据集中，通过指定 n_featurns 让样本只有两个特征导致的，如果数据集中样本特征更多，那么随机森林生成的模型会有可能比决策树的模型更简单。

五、 支持向量机

1.基本原理

支持向量机是一种新的算法，支持向量机的一种线性支持向量机（Linear SVM）它是一种使用线性内核的用于分类的支持向量机，我们重点研究对象是内核化的向量机，他是线性支持向量机的扩展，模型不再是像线性模型的直线或是超平面。因此可以解决非线性特征的数据样本分类或者回归问题。

生成一个非线性特征的数据集，导入相关库(from sklearn.datasets import make_blobs)，结合图像可视化（import matplotlib.pyplot as plt）和 numpy 库（import numpy as np）,然后生成数据集（x,y）对 y 进行整除 2，把标签从四个变成两个，最后用散点图方式呈现出来。可以得出线性模型的散点分布图像很难用一条直线分割出来无论是直线还是超平面，能够看出该数据是线性不可分。

2.将数据投射到高维空间

通过对样本增加特征，所以我们对其中一个样本特征作为第三个特征，让数据集特征数量增加到 3，这样可以让二维投射到三维。为了绘制三维图形，需要导入 matplotlib 中绘制三维绘图工具（from mpl_toolkits.mplot3D import Axes3D,axes3d）

然后创建绘图，把第二个特征的平方作为第三个特征（[x,x[:,:1:]**2]）添加到原始数据。这样就能创建三维图像，数据也从二维转为三维。我们可以轻松的从数据样本中进行分隔开。

2.支持向量机核函数和 gamma 参数

在支持向量机中，有两种常用的核函数可以将数据特征投射到高维，一种是多项式内核（polynomial kernel），另一种是径向基内核(Radial basis Funtion,RBF)。所以对原始样本数据之间的距离进行重新计算公式：kRBF (x1, x2) = exp (-γ ‖x1 - x2‖2) 这里 x1 和 x2 是数据点，‖x1 - x2‖ 表示欧氏距离，γ（gamma）是控制高斯核宽度的函数。

导入支持向量机（from sklearn import svm）生成一个用于试验的数据（x, y =make_blobs(n_samples=50,centers=2,random_state=20)

训练一个支持向量机模型，内核选择 RBF clf = svm.SVC(kernel = ‘rbf’, gamma= ‘auto’)然后拟合数据（x,y）,通过图像可视化相关配置，可以得出对于支持向量机来说，使用 RBF 内核的支持向量机模型对于支持向量机来说，并非每一个样本都对模型的结果有重要的影响，而是处在虚线(称为“决定边界”)上的样本才对模型的分类起到至关重要的作用。而这些处于决定边界上的样本，被称为“支持向量”（support vectors)。前面提到的 gamma 参数，控制的就是 RBF 内核圈入样本的数量，这里设置为 auto，支持向量机便会根据数据集的标签，自动选择 gamma 参数的大小，以便将尽可能多的样本放入正确的分类。如果手动调节

gamma 参数，将其值设置得越小，则 RBF 内核会将更多的样本“圈”进来，这时的模型就更简单（决定边界更平滑）；反之,gamma 值设置得越大，则 RBF 内核圈入的样本越少，模型也就更复杂。
在 sklearn 中如果不制定核函数则会默认情况下会使用 RBF 内核。对于 RBF 内核的支持向量机来说，gamma 参数直接决定了模型的复杂度，
创建 3 个 RBF 内核的支持向量机模型，gamma 参数分别取 0.5、5 和 50models = (svm.SVC(kernel=rbf, gamma=0.5),svm.SVC(kernel=rbf,gamma=5),svm.svc(kernel=rbf, gamma=50))从这里看出支持向量机的 gamma 参数逐步越大，从 0.5 到 5，再到 50。当 gamma 值为 0.5 时，RBF 内核的半径最大，决定边界也最平滑；当 gamma 值达到 5 时，模型的 RBF 内核半径变小了很多，决定边界也变得相对复杂；而当 gamma 参数增大到 50 时，模型 RBF 内核半径变得更小，决定边界也似乎在试图更紧密地“包裹”住样本，将更多的点放入正确的分类。也就是说，越大的 gamma 参数值对应的支持向量机模型越复杂，也越容易出现过拟合的现象。从经验来说，一般会设置 gamma 值为特征数量的倒数，如样本有 2 个特征，则 gamma 值取 0.5。
3.支持向量机的 C 参数
这个参数用来调节模型的复杂程度从对三个 C 参数对模型的影响创建 3 个 RBF 内核的支持向量机模型，C 参数分别取 0.01、1 和 100 models = (svm.SVC(kernel=rbf, C=0.01),svm.SVC(kernel=rbf,C=1),svm.SVC(kernel=rbf, C=100))从 C 参数分别是 0.01,1,100 对比可以看出参数越小，模型越平滑，但这时的模型倾向于欠拟合，C 参数越大，模型越复杂，会把训练集中更多的样本归入正确的分类。而这时的模型倾向于过拟合，所以根据不同的情况，可以通过 C 参数太调节控制模型的复杂度和准确率。
导入数据集拆分工具
(from sklearn.model_selection import train_test_split)
导入支持向量机回归模型（from sklearn.svm import SVR）
#因为“Sex”这一列是字符串，所以用 get_dummies 进行以下处理 data = pd.get_dummies(abalone).通过对数据集 get_dummies 处理后原字符串类型的样本特征转变为整型类型。
因为分类器往往默认数据数据是连续的、有序的。但是,直接数字并不是有序的,而是随机分配的。为了解决上述问题,其中一种可能的解决方法是采用独热编码。之所以使用 One-Hot 编码,是因为在很多机器学习任务中,特征并不总是连续值,也有可能是离散值（如上表中的数据）。将这些数据用数字来表示，执行的效率会高很多。-

六、 朴素贝叶斯

(1) 朴素贝叶斯算法是基于贝叶斯定理,在假设“所有特征都相互独立”的情况下，根据类先验概率和修正因子对类后验概率进行估计的一种算法。
(2) 朴素贝叶斯算法的特点是原理简单、易于实现、分类过程中效率高、时间花费少。
(3) 常见的朴素贝叶斯算法有 3 种。
①伯努利朴素贝叶斯，它适合特征符合伯努利分布的数据集。
②高斯朴素贝叶斯，它适合特征大致符合高斯分布（或可以转化为高斯分布）的数据集，同时在样本数量较大的数据集中表现相对更好。多项式朴素贝叶斯，它适合特征符合多项式分布（或可以转化为多项式分布）的数

据集，且在小样本的数据集中表现也不差.

朴素贝叶斯是一种监督式算法，在给定分类标签的情况，假设样本特征之间需要相互，这种需要朴素贝叶斯算法，贝叶斯的创始人用数学概率和统计学早期发展有重大影响，原理就是用原本的样本数据进行预测想要得到和好的结果，需要等到预测结果调整在最好的状态概率最高的时候，才是得出最好的结果。

使用贝叶斯定理来计算公式有 P(X|Y)=P(Y)*(P(X|Y)/P(X))

1. 训练伯努利朴素贝叶斯模型
2. 导入伯努利朴素贝叶斯（from sklearn.naive_bayes import BernoulliNB），然后定义样本特征（x,y），创建伯努利朴素贝叶斯分类器（clf= BernoulliNB()），拟合得出结果可以看出使用少量的样本训练伯努利朴素贝叶斯模型其准确率会较高。

   2.不同的朴素贝叶斯变体

   （1）高斯贝叶斯用于特征类型为连续数值类型的样本。假设样本特征符合高斯分布（正态分布），并进行概率计算，所以高斯贝叶斯在样本特征符合高斯分布的数据集中表现会比较好。

   导入高斯贝叶斯（from sklearn.naive_bayes import GaussianNB）,创建高斯贝叶斯分类器并进行预测（clf= GaussianNB().fit(x,y)）得出结果看出使用高斯贝叶斯算法训练的模型其准确率很高。

七、 K 近邻算法

K 最近邻算法是最简单的机器学习算法之一，也是一个理论上比较成熟的、运用基于样本估计的最大后验概率规则的判别方法。其思路是在特征空间中，如果一个样本附近的 k 个最近（特征空间中最邻近的 k 个）样本的大多数属于某一个类别，则该样本也属于这个类别。即给定一个训练集，对丁新输入实例，在训练集中找到与该实例最邻近的 k 个实例，这 k 个实例中的多数属于某个类，就把该输入实例分类到这个类中。K 最近邻算法在实际使用中会有很多问题，如它对规模超大的数据集拟合的时间较长，对高维数据集拟合欠佳，以及对于稀疏数据集束手无策等，因此在当前的各种常见的应用场景中，K 最近邻算法的使用并不多见。

导入 K 近邻算法（from sklearn.neighbors import KNeighborsClassifier）通过数据分类，进行可视化工具展现出两类数据可以看出 K 近邻算法基于数据创造了一个模型分类，这样如果有数据就能自动分类到对应的分类中。

K 近邻算法处理多元分类：首先生成多元分类，可以通过修改 make_bolbs 的 centers 参数，增加数据数量，同时增加样本数量（n_samples），利用散点图可视化代码进行可视化，提高而分类难度。

也可以通过算法拟合数据可以有利于大部分数据点放置于正确的分类中。

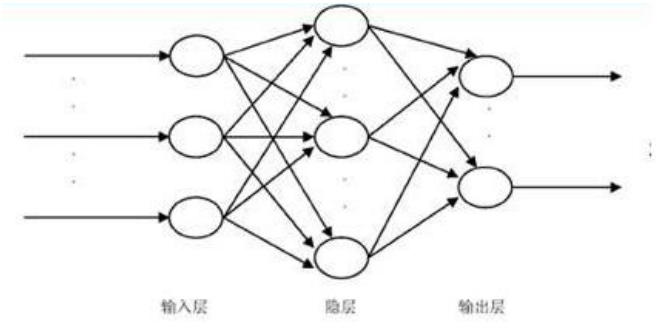
K 近邻算法可以使用回归分析模型对 KNN 使用默认参数，n_neightbors=5,reg 是 KNN 类创建的一个对象。使用 reg 对象对数据集进行拟合(reg.fit(x,y)使用建好的模型对新样本进行预测，先验证准确率，可以看出模型对验证集的吻合度越高，模型得分越高。

八、 神经网络

神经网络原理：

神经网络是一种算法结构，它让机器能够进行自我学习，常见的例子如语音命令、音乐创作和图像识别等。典型的神经网络由大量互连的人造神经元组成，它们按顺序堆叠在一起，以“层”的形式形成数百万个连接。一般情况下，“层”之间仅通过输入和输出与它们之间的神经元层互连（这与人类大脑中的神经元有很大

的不同，它们的互连是全方位的)。这种分层的神经网络是当今机器学习的主要方式之一，通过传递大量的标签数据，可以帮助模型学习如何解读数据(在某些方面甚至能够比人类做得还好)。多层神经网络（多层感知机）实际就是单层网络的扩展,其算法在过程里添加了隐藏层(hidden layer)，然后在隐藏层重复进行上述加权求和计算,最后再把隐藏层所计算的结果用来生成最终结果。



从右至左，分别是输入层，隐藏层，输入层每层都有一个神经元，也就是节点。输出层如果是多个节点则模型常用于解决分类问题，箭头连线都有一个权重值。导入 MLP 神经网络（from sklearn.neural_metwork import MLPClassifier）利用 pandas 库，导入数据及拆分函数（from sklearn.model_selection import train_test_split）生成数据集，将数据集拆分为训练集和验证集。

激活函数：是将非线性特性引入神经网络的十分重要的环节，通过输入层某个节点加权求和后，还被作用一个函数就是激活函数。常用的激活函数有六个：identity 函数输出等于输入、Sigmoid 函数任意值输出范围 0-1、tanh 函数可转化为-1-1 的值、relu 分段函数、cos 余弦函数、sinc 函数信息处理。

九、 聚类

聚类算法就是把相似的东西放到一起,，通常不需要使用训练数据进行学习，被称作是无监督学习。算法通过猜测那些数据应该放在一起，并且给不同的堆里设置数字标签。

聚类算法的种类：聚类算法的种类比较多，有基于划分的方法(partitioning methods)、基于层次的方法（hierarchical methods)、基于密度的方法（density-based methods)、基于网格的方法(grid-based methods)、基于模型的方法(model-based methods)、基于模糊的方法等

在各种聚类算法中，K 均值算法可以说是最简单的算法之一，但是简单不代表不好用。在大数据应用实践中，K 均值算法一般是在聚类中用得最多的算法之一。K 均值算法(k-means clustering algorithm)是最普及的划分聚类算法之一，该算法是一种迭代求解的算法，它接受一个未标记的数据集，并将每个数据样本聚集到其最近距离均值的类中。

K 均值算法是典型的基于距离的聚类算法。它采用距离作为相似度的评价指标，也就是两个样本的距离越近，其相似度就越大。K 均值算法认为聚类簇是由距离靠近的样本组成的，因此最终目标就是得到紧凑且独立的聚类簇。该算法只能处理数值型数据。

利用 k 均值算法建立模型进行聚类分析：首先导入相关包（KMearn）工具（from sklearn.cluster import KMearn），numpy 包，要求聚类数量相关参数（n_cluster=）拟合数据（KMearn.fit()）,最后进行数据可视化相关操作。得出 KMearn 聚类和分类效果相似。

BSCAN；基于密度的有噪声的应用空间聚类。就是把所有的密度可达对象组成了一个簇。算法通过对特征空间密度检测，密度较大地方会成为一类，较小的成为

一类，这样形成一个分界线，是的不需要指定 n_clusters 参数。
算法将数据点分为以下 3 类。
核心点，在半径 eps 内含有超过指定数目 MinPts 的点。
边界点，点在半径 eps 内的数量小于指定数目 MinPts，但是落在核心点的邻域内。
噪声点，既不是核心点也不是边界点的点。其工作原理和步骤如下。(1)寻找核心点并形成聚类簇：在一个数据集中，任意两个样本点是“密度直达”或“密度可达”的关系，那么这两个样本点归为同一簇类。
(2)随机选择（1）中形成的聚类簇，遍历检查其内部的所有点是否为核心点，完成聚类簇的合并。
(3)遍历所有聚类簇，完成整个数据集的聚类。

十、 PCA 降维、特征提取

降维可视化：通过 PCA、LDA 或 SVD 矩阵分解，将高纬数据转换为 2 维，便于可视化查看数据分布，了解数据特性。pandasmatplotlib。如果数据纬度过高需要降维，可以用 T-sne 可以对数据降维和可视化 Pandas、Matplotlib 和 seaborn 库的一些常用可视化方法，并对客户离网数据集进行了可视化分析和 t-SNE 降维。
作为一个非监督学习的降维方法，它只需要特征值分解，就可以对数据进行压缩，去噪。因此在实际场景应用很广泛。为了克服 PCA 的一些缺点，出现了很多 PCA 的变种，比如第六节的为解决非线性降维的 KPCA，还有解决内存限制的增量 PCA 方法 Incremental PCA，以及解决稀疏数据降维的 PCA 方法 Sparse PCA 等。
PCA 降维
导入 PCA
（From sklearn.decomposition import PCA），设置主成分（pca=PCA(_component8=3)），拟合数据（pca.fit(X)），pca.transfroms(x)转换数据，最后将其可视化操作可得出原本二维的原始数据变成了一维。

PCA 算法的主要优点有：
1）仅仅需要以方差衡量信息量，不受数据集以外的因素影响。
2）各主成分之间正交，可消除原始数据成分间的相互影响的因素。
3）计算方法简单，主要运算是特征值分解，易于实现。
PCA 算法的主要缺点有：
1)主成分各个特征维度的含义具有一定的模糊性，不如原始样本特征的解释性强。
2)方差小的非主成分也可能含有对样本差异的重要信息，因降维丢弃可能对后续数据处理有影响。
创建一个 PCA 实例，令主成分数量为 100，并使用 whiten 参数开启数据白化功能前面需要拆分函数对数据图像灰度值和数据分类标签设置好。
使用该 PCA 实例拟合训练集 pca=PCA(n components-100,whiten-True,random_state-0).fit(Xft＃对训练集与验证集的特征进行转换 X_ftrain pca=pca.transform(X.ftrain) X_ftest_pca=pca.transform(X_ftest)
查看数据白化后的样本特征 print("进行数据白化后的样本特征：（}”.format(X_ftrain_pca.shape))。对于两个人物图像的抓取特征需要

进行单独像素颜色的对应位置进行比较，很难捕捉到人物特征，PCA 有一种数据白化功能，将 PCA 提取的主成分重新缩放和原始比例相同的特征，实现特征提取。

HSNE 的中心思想是:找到数据的二维表示,并尽可能保持数据点之间的距离。t-SNE 首先对每个数据点进行随机二维表示，然后尝试使原始特征空间中距离较近的点更近，使原始特征空间中距离较远的点更远。t-SNE 更强调距离较近的点，换句话说，它试图保存指示哪些点彼此相邻的信息。

```
#导入 t-SNE 模块
from sklearn.manifold import TSNE
#创建一个 t-SNE 模型
tsne = TSNE(random_state=42)
#使用 t-SNE 拟合并转换原始数据
digits_tsne = tsne.fit_transform(digits.data)
```

十一、模型选择、优化及评估

模型选择(model selection)有两层含义：一是在假设空间上训练得到的模型可能不止一个，需要从中进行选择，也就是我们上一篇笔记所介绍的归纳偏好的具体实现（在实践中往往是选择同等效果下复杂度较小的模型）；二是对于一个具体问题，我们可能希望尝试不同方法，于是就有了不同的模型，在这些模型训练结束后，我们需要决定使用哪一个，但这种模型选择往往需要结合模型评估方法，因为对于某种归纳偏好，不同方法下的不同模型的实现各不相同，只能根据在测试集上的最终表现效果来选择。

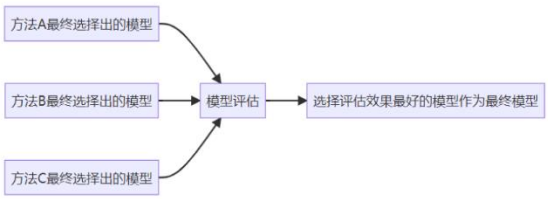
交叉验证

在样本量足够大的理想情况下，应当把数据集分割为三部分：训练集(training set)、验证集(validation set)和测试集(testing set)，分别用于模型训练、模型选择和模型评估，用于评估模型泛化能力的测试集只出现在最后的模型评估环节。常用的交叉验证法有三种：简单交叉验证/留出法(hold-out)：将数据集按一定比例随机分为两部分：训练集和测试集，分别在其上训练和测试所有备选模型，选出测试结果最好的，这相当于用模型评估代替了模型选择，直接砍掉验证集来增加其余两个集合的样本量。

使用 K-折交叉验证法评估模型

这里使用 scikit-learn 库中的鸢尾花数据集进行 K-折交叉验证法

导入鸢尾花数据集




```
from sklearn.datasets import load_iris 导入交叉验证工具
from sklearn.model_selection import cross_val_score 导入用于分类的支持向量机模型 from sklearn.svm import svc
```

K 折交叉验证(k-fold cross validation)：将数据集随机划分 K 为个大小相同或基本相同的子集，分别把(K-1)每一个子集作为测试集，其余(K-1)个子集作为训练集，就得到了 K 组不同的训练、测试集，在这 K 组训练、测试集上训练并测试每一种模型，选择平均测试误差最小的模型；有时为了避免单次随机划分的特殊性，还会进行多次随机划分，将多个交叉验证的结果再进行一次平均；

留一交叉验证(leave-one-out cross validation)：K 折交叉验证的特例，将 K 取为样本量 N，也即把每个样本单独作为测试集，其余样本作为训练集。这种方法的计算量较大，一般仅用于数据稀少的情况。

当样本量实在过小时，可以考虑采用有放回抽样，抽取次数和原数据集样本量相等，形

成一个新的样本作为训练集，而将未被抽到过的样本全体作为测试集，这种在样本量较小的情况下构造样本的想法来源于统计学中的自助法(bootstrapping)。

十二、数据预处理，特征选择

数据预处理需要根据数据本身的特性进行，有不同的格式和不同的要求，有缺失值的要填，有无效数据的要剔，有冗余维的要选，这些步骤都和数据本身的特性紧密相关。数据预处理大致分为三个步骤：数据的准备、数据的转换、数据的输出。数据处理是系统工程的基本环节，也是提高算法准确度的有效手段。因此，为了提高算法模型的准确度，在机器学习中也要根据算法的特征和数据的特征对数据进行转换。这里将利用 scikit-learn 来转换数据，以便我们将处理后的数据应用到算法中，这样也可以提高算法模型的准确度。

主要介绍以下几种数据转换方法：调整数据尺度（RescaleData）、正态化数据（StandardizeData）、标准化数据(NormalizeData)、二值数据(BinarizeData)。

特征选择：基于模型的特征选择使用有监督的机器学习算法来判断每个特征的重要性，并且只保留最重要的特征。用于特征选择的模型不一定要与最终用于预测的模型相同。特征选择模型需要为每个特征提供一些重要的度量，如线性模型中的 coefficients，或者决策树算法中的 feature_importances_。与单变量统计选择不同的是，基于模型的特征选择同时考虑样本的所有特征，因此其可以发现特征之间的交互性。

在 scikit-learn 中，可以使用 SelectFromModel 来进行基于模型的特征选择。导入基于模型的特征选择工具（from sklearn. feature_selection import SelectFromModel）导入决策树（from sklearn.tree import DecisionTreeRegressor）

十三、文本处理数据

基于计数向量器的文本特征提取

数据特征大致可以分为两类:一类是表示数值的连续特征，另一类是表示样本分类的类型特征。在自然语言处理领域，我们会遇到第三类数据一文本数据。

CountVectorizer 中实现的，它是一个变换器（transformer)。

导入计数向量器 CountVectorizer

```
from sklearn.feature_extraction.text import CountVectorizer
vect = CountVectorizer()
```

为了对汉语进行正确的向量化，首先要做的事情是进行分词处理。本章将使用“结巴分词”来对汉语进行分词处理。

导入结巴分词(import jieba)

数据属性的两种类型的特征：连续特征与分类特征，前者用于描述数量，后者是固定列表中的元素

文本通常只是数据集中的字符串，但并非所有的字符串特征都应该被当作文本来处理。字符串特征有时可以表示分类变量。

使用计数向量器提取文本特征，对汉语文本数据进行分词处理，建立词包模型将文本特征变成数组。

通过对机器学习的学习和了解，感觉非常有意思和很神奇，非常智能化和理性化，虽然有很多的算法的深层含义不太理解，但是对各种算法模型有了大概思维框架，懂的了利用数据样本处理计算模型的过程，利用数据，训练数据，预测得出结果的方式。 学习机器学习的总体目的便是考虑运用什么样的模型对数据

进行学习，使得得到的模型能够更好的对数据进行预测，提高学习效率。方法有很多，模型也很多。①得到一个有限的训练数据集合②确定包含所有可能的模型的假设空间，即学习模型的集合③确定模型选择的准则，即学习的策略④实现求解最优模型的算法，即学习的算法⑤通过学习方法选择最优模型⑥利用学习的最优模型对新数据进行预测分析机器学习方法的三要素为：模型（模型的假设空间）、策略（模型选择的准则）、算法（模型学习的算法）对于学习模型的好坏也有这一套评判方法，比如 0-1 损失函数、平方损失函数、对数损失函数、绝对损失函数等。

。