1. 使用SVM,朴素贝叶斯，KNN, 使用Sklearn自带digits数据集训练识别手写体

代码： plot\_digits\_classification.py

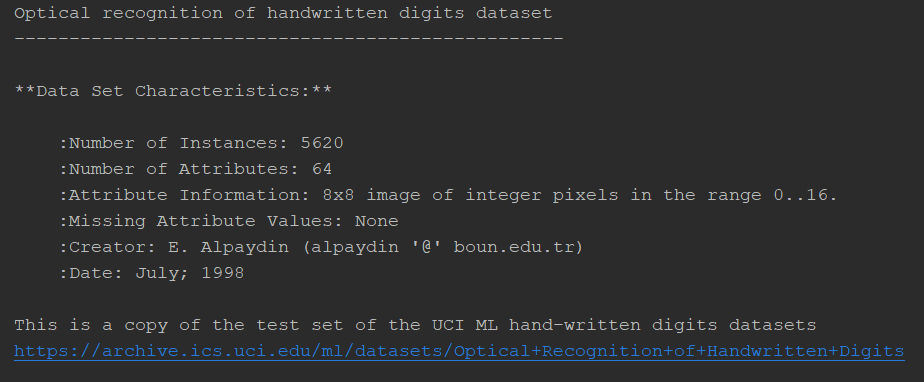
按照以下步骤，阅读并理解代码，然后运行看下效果。

1. digits数据集的介绍

手写数字数据集

# The digits dataset  
digits = datasets.load\_digits()  
print(digits.DESCR)

输出结果：

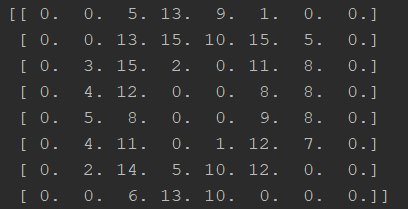


该数据集是1797张8\*8像素大小的灰度图，采用分类器进行手写体识别时，都是将每张图像看成64维的特征向量。

显示第一张图片

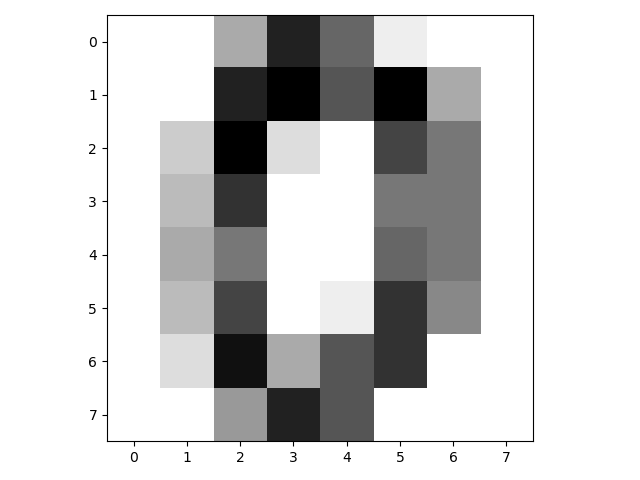


输出：



显示数字灰度图

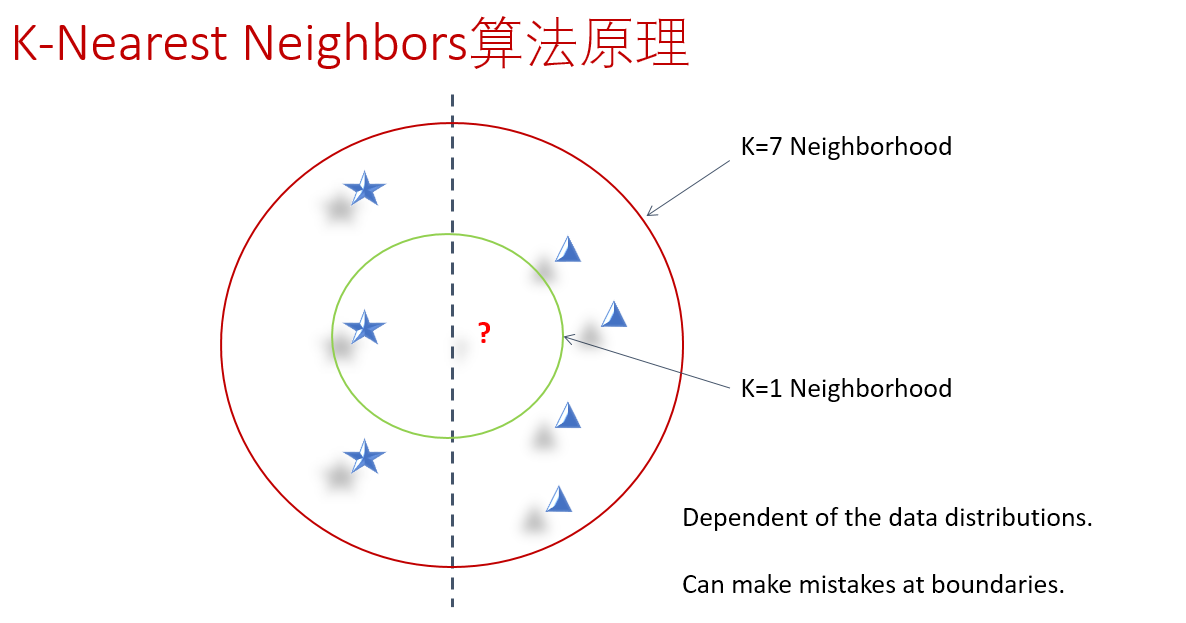
import matplotlib.pyplot as plt  
plt.imshow(digits.images[0], cmap=plt.cm.gray\_r, interpolation='nearest')



1. 简要了解几种分类器原理以及sklearn中如何调用分类器
   1. KNN分类器

工作原理

* 存在一个样本数据集合，也称作训练样本集，并且样本集中每个数据都存在标签，即我们知道样本集中每个数据与所属分类的对应关系。
* 输入没有标签的新数据后，将新数据的每个特征与样本集中数据对应的特征进行比较，然后算法提取样本集中特征最相似数据（最近邻）的分类标签。
* 一般来说，只选择样本数据集中前K个最相似的数据。K一般不大于20，最后，选择k个中出现次数最多的分类，作为新数据的分类



KNN算法一般流程：

* 收集数据：可以使用任何方法
* 准备数据：距离计算所需要的数值，最后是结构化的数据格式。
* 分析数据：可以使用任何方法
* 训练算法： （此步骤kNN）中不适用
* 测试算法：计算错误率
* 使用算法：首先需要输入样本数据，然后运行k-近邻算法判定输入数据分别属于哪个分类，最后应用对计算出的分类执行后续的处理。
  1. SVM分类器

支持向量机（Support Vector Machine，SVM）是一种二类分类模型，其求解目标在于确定一个分类的超平面，以最大化特征空间上的间隔。

# 

# 

# 

# 

* 1. 朴素贝叶斯分类器

# 朴素贝叶斯分类器的构造基础是贝叶斯理论。抽象一些说，朴素贝叶斯分类器会单独考虑每一维度特征被分类的条件概率，进而综合这些概率并对其所在的特征向量做出分类预测。因此，这个模型的基本数学假设是：各个维度上的特征被分类的条件概率之间是相互独立的。如果采用概率模型来表述.则定义x=[x1,x2,...,xn]为某一n维特征向量，y∈{c1,c2,...,ck}为该特征向量x所有k种可能的类别，记P(y=ci I x)为特征向量x属于类别ci的概率，我们的目标是寻找所有y∈{c1,c2,...,ck}中P( y | x)最大的，即argmax P ( y |x )；并且考虑到P(x)对于同一样本都是相同的，因此可以忽略不计。

# 

# 

# 

# 

# 

* 1. 认识多分类问题

二元分类器在两个类别中区分，多类别分类器（也称为多项分类器）可以区分两个以上的类别。 手写体识别是多分类问题，一些算法（比如随机森林分类器或者朴素贝叶斯分类器）可以直接处理多类分类问题。其他一些算法（比如 SVM 分类器或者线性分类器）则是严格的二分类器。然后，有许多策略可以让你用二分类器去执行多类分类。

* “一对所有”（OvA）策略：训练10个二分类器，每一个对应一个数字（探测器 0，探测器 1，探测器 2，以此类推）。然后当你想对某张图片进行分类的时候，让每一个分类器对这个图片进行分类，选出决策分数最高的那个分类器。
* “一对一”（OvO）策略：一个分类器用来处理数字 0 和数字 1，一个用来处理数字 0 和数字 2，一个用来处理数字 1 和 2，以此类推。如果有 N 个类。你需要训练N\*(N-1)/2个分类器。

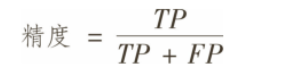
一些算法（比如 SVM 分类器）在训练集的大小上很难扩展，所以对于这些算法，OvO 是比较好的，因为它可以在小的数据集上面可以更多地训练，较之于巨大的数据集而言。但是，对于大部分的二分类器来说，OvA 是更好的选择。Scikit-Learn 可以探测出你想使用一个二分类器去完成多分类的任务，它会自动地执行 OvA（除了 SVM 分类器，它使用 OvO）。

1. 分类性能考核指标

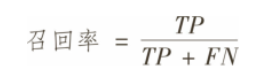
3.1 二分类性能指标

二分类性能指标主要有以下这些，以下指标涉及到TP, FP, TN, FN，查阅了解 TP, FP, TN, FN的定义，以及以下性能评估指标的意义。

（1）精度

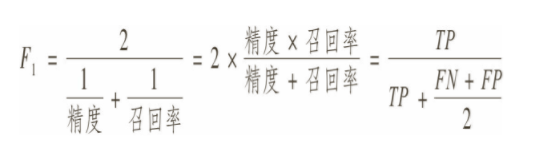


（2）召回率



（3）F1-Measure

# F1 分数是精度和召回率的谐波平均值。谐波平均值会给予较低的值更高的权重。因此，只有当召回率和精度都很高时，分类器才能得到较高的F1 分数



3.2多分类性能指标

请查阅了解如何在二分类评估指标基础上，如何评价多类别分类器的性能，有哪些性能指标可用。

（二）文本分类

模仿手写数字识别代码，使用SVM, KNN，朴素贝叶斯对新闻数据集进行文本分类，比较几个分类器的性能；python源码文件请命名为 news\_classfication\_学号最后三位数字.py,

如news\_classification\_134.py , 另外再提交一份word文档，根据我黄色加亮部分的要求，根据你调研的结果，答复我的问题要求，写到文档里，一起[发到我1055733857@qq.com](mailto:发到我1055733857@qq.com)邮箱，注意都不要传到群里。 以下是我给的代码片段，供大家参考。

2.1 载入20分类的新闻数据集

了解新闻数据集： 20newsgroups数据集是用于文本分类、文本挖据和信息检索研究的国际标准数据集之一。数据集收集了大约20,000左右的新闻组文档，均匀分为20个不同主题的新闻组集合；

了解以下代码的含义，查询sklearn文档，了解所调用方法以及方法参数

from sklearn.datasets import [fetch\_20newsgroups](https://scikit-learn.org/stable/modules/generated/sklearn.datasets.fetch_20newsgroups.html#sklearn.datasets.fetch_20newsgroups)

# sample\_cate 指定需要下载哪几个主题类别的新闻数据

sample\_cate = ['alt.atheism', 'soc.religion.christian', 'comp.graphics', 'sci.med', 'rec.sport.baseball']  
# 需要从网络上下载，受连接外网速度限制, 可能要耐心等待几分钟时间

newsgroups\_train = fetch\_20newsgroups(subset='train', categories=sample\_cate, shuffle=True)

# 以上得到训练集，以下代码得到测试集  
newsgroups\_test = fetch\_20newsgroups(subset='test', categories=sample\_cate, shuffle=True)

了解有监督学习中训练集、测试集、验证集的概念和各自作用

2.2 对文本数据进行向量化

查阅资料，了解one-hot词集模型(SOW,Set of Words)、词袋模型(BOW, Bag of Words)、TF-IDF(Term Frequency-Inverse Document Frequency) 、词频（TF）、逆文档频率(IDF)等概念；

了解sklearn中词袋模型、TF-IDF对应的向量化类, 了解如何向量化，调用参数意义等；

from sklearn.feature\_extraction.text import CountVectorizer  
from sklearn.feature\_extraction.text import TfidfVectorizer

词袋+词频, 对训练集、测试集都要进行向量化

count\_vectorizer = CountVectorizer(stop\_words='english')  
cv\_news\_train\_vector = count\_vectorizer.fit\_transform(newsgroups\_train.data)  
print("news\_train\_vector.shape:", cv\_news\_train\_vector.shape)  
cv\_news\_test\_vector = count\_vectorizer.transform(newsgroups\_test.data)  
print("news\_test\_vector.shape :", cv\_news\_test\_vector.shape)

词袋+TF-IDF

tfidf\_vectorizer = TfidfVectorizer(stop\_words='english')  
tfidf\_news\_train\_vector = tfidf\_vectorizer.fit\_transform(newsgroups\_train.data)  
print("news\_train\_vector.shape:", tfidf\_news\_train\_vector.shape)  
tfidf\_news\_test\_vector = tfidf\_vectorizer.transform(newsgroups\_test.data)  
print("news\_test\_vector.shape :", tfidf\_news\_test\_vector.shape)

2.3 初始化分类器

查阅资料，了解朴素贝叶斯的三个常用模型，高斯模型、多项式模型、伯努利模型的区别；

nb\_clf = MultinomialNB() #初始化多项式模型贝叶斯分类器

2.4 在训练集上训练分类器

nb\_clf.fit(cv\_news\_train\_vector, newsgroups\_train.target)

2.5 在测试集上测试分类器

cv\_nb\_predict = nb\_clf.predict(cv\_news\_test\_vector)

2.6 评估测试结果

from sklearn import metrics

了解metrics的作用以及classification\_report方法

print("Classification report for classifier %s:\n%s\n"  
 % (nb\_clf, metrics.classification\_report(newsgroups\_test.target, cv\_nb\_predict, target\_names=newsgroups\_test.target\_names)))

2.7 实现分类代码

参考第一个例子的代码，并按照上述提示，实现贝叶斯分类，之后添加SVM, KNN两个分类器，分别比较三个分类器采用词袋+词频(CounterVectorizer)、词袋+TF-IDF向量(TfidfVectorizer)时的分类性能。