作业 26：PCA 实现高维数据可视化

要求：

已知鸢尾花数据是 4 维的，共三类样本。使用 PCA 实现对鸢尾花数据进行降维，实现在二维平面上的可视化。

萼片长度	萼片宽度	花瓣长度	花瓣宽度	类别
5.1	3.5	1.4	0.2	Iris-setosa
4.9	3	1.4	0.2	Iris-setosa
4.7	3.2	1.3	0.2	Iris-setosa
4.6	3.1	1.5	0.2	Iris-setosa
5	3.6	1.4	0.2	Iris-setosa
5.4	3.9	1.7	0.4	Iris-setosa
4.6	3.4	1.4	0.3	Iris-setosa
5	3.4	1.5	0.2	Iris-setosa
4.4	2.9	1.4	0.2	Iris-setosa
4.9	3.1	1.5	0.1	Iris-setosa
5.4	3.7	1.5	0.2	Iris-setosa
4.8	3.4	1.6	0.2	Iris-setosa
4.8	3	1.4	0.1	Iris-setosa
4.3	3	1.1	0.1	Iris-setosa
5.8	4	1.2	0.2	Iris-setosa

图. 鸢尾花数据

提示：

1. 建立工程，导入 sklearn 相关工具包：

```
import matplotlib.pyplot as plt
from sklearn.decomposition import PCA
from sklearn.datasets import load_iris
```

2. 加载数据并进行降维：

```
data = load_iris()
#以字典形式加载鸢尾花数据集
y = data.target #使用 y 表示数据集中的标签
X = data.data #使用 X 表示数据集中的属性数据
pca = PCA(n_components=2)
reduced_X = pca.fit_transform(X)
```

3. 按类别对降维后的数据进行保存
4. 降维后数据点的可视化

作业 27：降维之 NMF

要求：

已知 Olivetti 人脸数据共 400 个，每个数据是 64*64 大小。由于 NMF 分解得到的 W 矩阵相当于从原始矩阵中提取的特征，那么就可以使用 NMF 对 400 个人脸数据进行特征提取。

通过设置 k 的大小，设置提取的特征的数目。在本实验中设置 k=6，随后将提取的特征以图像的形式展示出来。

提示：

1.建立工程，导入 sklearn 相关工具包：

```
import matplotlib.pyplot as plt
from sklearn import decomposition
from sklearn.datasets import fetch_olivetti_faces
from numpy.random import RandomState
```

2. 设置基本参数并加载数据：

```
n_row, n_col = 2, 3
n_components = n_row * n_col
image_shape = (64, 64)
dataset=fetch_olivetti_faces(shuffle=True,random_state=RandomState(
0))
faces = datasets.data
```

3.1 设置图像的展示方式

3.2 创建特征提取的对象 NMF，使用 PCA 作为对比：

4. 降维后数据点的可视化

作业 28：图像分割

要求：

利用 K-means 聚类算法对图像像素点颜色进行聚类实现简单的图像分割输出：同一聚类中的点使用相同颜色标记，不同聚类颜色不同

提示：

数据：本实例中的数据可以是任意大小的图片，为了使效果更佳直观，可以采用区分度比较明显的图片。

使用算法：Kmeans

实现步骤：

1. 建立工程并导入 sklearn 包
    - 创建 Kmeans.py 文件
    - 导入 sklearn 相关包
        - import numpy as np
        - import PIL.Image as image

- from sklearn.cluster import KMeans

2. 加载图片并进行预处理

加载训练数据

```
def loadData(filePath):
    f = open(filePath,'rb')   #以二进制形式打开文件
    data = []
    img = image.open(f)       #以列表形式返回图片像素值
    m,n = img.size            #获得图片的大小
    for i in range(m):        #将每个像素点RGB颜色处理到0-1
        for j in range(n):    #范围内并存放进data
            x,y,z = img.getpixel((i,j))
            data.append([x/256.0,y/256.0,z/256.0])
    f.close()
    return np.mat(data),m,n   #以矩阵形式返回data，以及图片大小

imgData,row,col = loadData('kmeans/bull.jpg') #加载数据
```

3. 加载 Kmeans 聚类算法
   - 加载 Kmeans 聚类算法
     - km = KMeans(n_clusters=3)
     - 其中 n_clusters 属性指定了聚类中心的个数为 3

4. 对像素点进行聚类并输出
   - 依据聚类中心，对属于同一聚类的点使用同样的颜色进行标记

作业 29：人体运动状态信息评级

要求：

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。

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。

提示：

1. 模块导入：

导入 numpy 库和 pandas 库。从 sklearn 库中导入预处理模块 Imputer。导入自动生成训练集和测试集的模块 train_test_split。导入预测结果评估模块 classification_report

接下来，从 sklearn 库中依次导入三个分类器模块：K 近邻分类器

KNeighborsClassifier、决策树分类器 DecisionTreeClassifier 和高斯朴素贝叶斯函数 GaussianNB。

2. 数据导入函数

编写数据导入函数，设置传入两个参数，分别是特征文件的列表 feature_paths 和标签文件的列表 label_paths。

定义 feature 数组变量，列数量和特征维度一致为 41；定义空的标签变量，列数量与标签维度一致为 1。

使用 pandas 库的 read_table 函数读取一个特征文件的内容，其中指定分隔符为逗号、缺失值为问号且文件不包含表头行。

使用 Imputer 函数，通过设定 strategy 参数为‘mean’，使用平均值对缺失数据进行补全。fit()函数用于训练预处理器，transform()函数用于生成预处理结果。

将预处理后的数据加入 feature，依次遍历完所有特征文件

遵循与处理特征文件相同的思想，我们首先使用 pandas 库的 read_table 函数读取一个标签文件的内容，其中指定分隔符为逗号且文件不包含表头行。

由于标签文件没有缺失值，所以直接将读取到的新数据加入 label 集合，依次遍历完所有标签文件，得到标签集合 label。

最后函数将特征集合 feature 与标签集合 label 返回。

3.1 主函数-数据准备

设置数据路径 feature_paths 和 label_paths。

使用 python 的分片方法，将数据路径中的前 4 个值作为训练集，并作为参数传入 load_dataset()函数中，得到训练集合的特征 x_train，训练集的标签 y_train。

将最后一个值对应的数据作为测试集，送入 load_dataset()函数中，得到测试集合的特征 x_test，测试集的标签 y_test。

使用 train_test_split()函数，通过设置测试集比例 test_size 为 0，将数据随机打乱，便于后续分类器的初始化和训练。

3.2 创建主函数

创建 k 近邻分类器、决策树分类器、贝叶斯分类器，并在测试集上进行预测。

4. 分类结果分析

使用 classification_report 函数对分类结果，从精确率 precision、召回率 recall、f1 值 f1-score 和支持度 support 四个维度进行衡量。分别对三个分类器的分类结果进行输出。

作业 30：上证指数涨跌预测实例

要求：

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。

提示：

数据为中核科技 1997 年到 2017 年的股票数据部分截图，红框部分为选取的特征值。

日期	股票代码	名称	收盘价	最高价	最低价	开盘价	前收盘	涨跌额	涨跌幅	换手率	成交量	成交金额	总市值	流通市值
2017/1/20	'000777	中核科技	21.17	21.29	20.9	20.9	20.86	0.31	1.4861	1.0687	4097505	86664725.78	8116950444	8116950444
2017/1/19	'000777	中核科技	20.86	21.14	20.82	21.12	21.12	-0.26	-1.2311	1.0455	4008703	83926679.28	7998090990	7998090990
2017/1/18	'000777	中核科技	21.12	21.44	21.09	21.4	21.37	-0.25	-1.1699	0.922	3535002	75292556.6	8097779564	8097779564
2017/1/17	'000777	中核科技	21.37	21.49	20.75	21.17	21.15	0.22	1.0402	1.3459	5160269	109652595.5	8193633962	8193633962
2017/1/16	'000777	中核科技	21.15	22.5	20.28	22.5	22.53	-1.38	-6.1252	3.1691	12150966	261947917.1	8109282092	8109282092
2017/1/13	'000777	中核科技	22.53	22.88	22.43	22.71	22.85	-0.32	-1.4004	1.8603	7132550	161394780.8	8638398370	8638398370
2017/1/12	'000777	中核科技	22.85	23.53	22.75	23.41	23.51	-0.66	-2.8073	2.817	10800996	249876234.2	8761092000	8761092000
2017/1/11	'000777	中核科技	23.51	23.71	23.06	23.22	23.25	0.26	1.1183	4.0062	15360483	360093755.2	9014147611	9014147611
2017/1/10	'000777	中核科技	23.25	23.59	23.23	23.4	23.57	-0.32	-1.3577	2.713	10402149	243289916.6	8914459037	8914459037
2017/1/9	'000777	中核科技	23.57	23.7	22.72	22.96	23	0.57	2.4783	5.3134	20372449	475747935.6	9037152667	9037152667
2017/1/6	'000777	中核科技	23	23.19	22.82	22.95	22.87	0.13	0.5684	3.0819	11816610	271885545.4	8818604639	8818604639
2017/1/5	'000777	中核科技	22.87	22.93	22.56	22.75	22.75	0.12	0.5275	2.6699	10236812	233103957.5	8768760352	8768760352
2017/1/4	'000777	中核科技	22.75	22.81	22.54	22.65	22.6	0.15	0.6637	1.5802	6058882	137503830.2	8722750241	8722750241
2017/1/3	'000777	中核科技	22.6	22.68	22.36	22.49	22.38	0.22	0.983	1.3948	5348100	120728947.2	8665237602	8665237602
2016/12/30	'000777	中核科技	22.38	22.63	22.31	22.49	22.58	-0.2	-0.8857	1.322	5068828	113686645.3	8580885731	8580885731
2016/12/29	'000777	中核科技	22.58	22.7	22.36	22.41	22.43	0.15	0.6687	1.2307	4718858	106240524.4	8657569250	8657569250
2016/12/28	'000777	中核科技	22.43	22.72	22.42	22.63	22.58	-0.15	-0.6643	1.4301	5483427	123681991.5	8600056611	8600056611
2016/12/27	'000777	中核科技	22.58	22.93	22.56	22.92	22.91	-0.33	-1.4404	1.5646	5998804	136263536.3	8657569250	8657569250
2016/12/26	'000777	中核科技	22.91	22.96	22.38	22.7	22.89	0.02	0.0874	2.1045	8068925	182955263.2	8784097056	8784097056
2016/12/23	'000777	中核科技	22.89	23.25	22.64	22.95	23.11	-0.22	-0.952	2.38	9125180	208889546.6	8776428704	8776428704
2016/12/22	'000777	中核科技	23.11	23.55	22.75	22.82	22.82	0.29	1.2708	3.7389	14335433	333074476.8	8860780574	8860780574
2016/12/21	'000777	中核科技	22.82	22.96	22.58	22.59	22.53	0.29	1.2872	2.2115	8479447	193133942.4	8749589472	8749589472
2016/12/20	'000777	中核科技	22.53	22.67	22.41	22.67	22.69	-0.16	-0.7052	1.329	5095772	114711037.5	8638398370	8638398370
2016/12/19	'000777	中核科技	22.69	22.77	22.51	22.67	22.63	0.06	0.2651	1.4709	5639790	127588225.1	8699745185	8699745185
2016/12/16	'000777	中核科技	22.63	22.88	22.58	22.73	22.71	-0.08	-0.3523	1.9302	7400685	168016411.1	8676740130	8676740130

使用算法： SVM

实现步骤：

1. 建立工程，导入 sklearn 相关包

```
import pandas as pd
import numpy as np
from sklearn import svm
from sklearn import cross_validation
```

2. 数据加载&&数据预处理

```
data=pd.read_csv('stock/000777.csv',encoding='gbk',parse_dates=[0],
index_col=0)
data.sort_index(0,ascending=True,inplace=True)
dayfeature=150
featurenum=5*dayfeature
x=np.zeros((data.shape[0]-dayfeature,featurenum+1))
y=np.zeros((data.shape[0]-dayfeature))
```

3. 创建 SVM 并进行交叉验证

作业 31：线性回归+房价与房屋尺寸关系的线性拟合

要求：

背景：与房价密切相关的除了单位的房价，还有房屋的尺寸。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。

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。

提示：

使用算法：线性回归
步骤：

1. 建立工程并导入 sklearn 包
   创建 house.py 文件
   导入 sklearn 相关包
   - import matplotlib.pyplot as plt
   - from sklearn import linear_model
2. 加载训练数据，建立回归方程

```
datasets_X = []
datasets_Y = []
fr = open('prices.txt','r')
lines = fr.readlines()
for line in lines:
    items = line.strip().split(',')
    datasets_X.append(int(items[0]))
    datasets_Y.append(int(items[1]))
length = len(datasets_X)
datasets_X = np.array(datasets_X).reshape([length,1])
datasets_Y = np.array(datasets_Y)
```

建立datasets_X和datasets_Y用来存储数据中的房屋尺寸和房屋成交价格。

打开数据集所在文件prices.txt，读取数据。

一次读取整个文件。

```
datasets_X = []
datasets_Y = []
fr = open('prices.txt','r')
lines = fr.readlines()
for line in lines:
    items = line.strip().split(',')
    datasets_X.append(int(items[0]))
    datasets_Y.append(int(items[1]))
length = len(datasets_X)
datasets_X = np.array(datasets_X).reshape([length,1])
datasets_Y = np.array(datasets_Y)
```

逐行进行操作，循环遍历所有数据

去除数据文件中的逗号

将读取的数据转换为int型，并分别写入datasets_X和datasets_Y。

……

3. 可视化处理

作业 32：多项式回归+房价与房屋尺寸的非线性拟合

要求：

背景：我们在前面已经根据已知的房屋成交价和房屋的尺寸进行了线性回归，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了成交价格的预测，但是在实际的应用中这样的拟合往往不够好，因此我们在

此对该数据集进行多项式回归。

对房屋成交信息建立多项式回归方程，并依据回归方程对房屋价格进行预测。

提示：

使用算法：线性回归

步骤：

1. 建立工程并导入 sklearn 包

创建 house.py 文件

导入 sklearn 相关包

- import matplotlib.pyplot as plt
- import numpy as np
- from sklearn import linear_model
- from sklearn.preprocessing import PolynomialFeatures

这里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。

2. 加载训练数据，建立回归方程

```
datasets_X = []
datasets_Y = []
fr = open('prices.txt','r')
lines = fr.readlines()
for line in lines:
    items = line.strip().split(',')
    datasets_X.append(int(items[0]))
    datasets_Y.append(int(items[1]))
length = len(datasets_X)
datasets_X = np.array(datasets_X).reshape([length,1])
datasets_Y = np.array(datasets_Y)
```

建立datasets_X和datasets_Y用来存储数据中的房屋尺寸和房屋成交价格。

打开数据集所在文件prices.txt，读取数据。

一次读取整个文件。

```
datasets_X = []
datasets_Y = []
fr = open('prices.txt','r')
lines = fr.readlines()
for line in lines:
    items = line.strip().split(',')
    datasets_X.append(int(items[0]))
    datasets_Y.append(int(items[1]))
length = len(datasets_X)
datasets_X = np.array(datasets_X).reshape([length,1])
datasets_Y = np.array(datasets_Y)
```

逐行进行操作，循环遍历所有数据

去除数据文件中的逗号

将读取的数据转换为int型，并分别写入datasets_X和datasets_Y。

……

3. 可视化处理