常用代码

作者: 刘坤鑫

头文件

```
import numpy as np
import pandas as pd
import  matplotlib.pyplot as plt
from matplotlib.colors import ListedColormap
%matplotlib  inline

import sklearn.datasets as datasets
from sklearn.externals import joblib
from sklearn.model_selection import train_test_split
from sklearn.model_selection import GridSearchCV
from sklearn.neighbors import KNeighborsClassifier
from sklearn.neighbors import KNeighborsRegressor
from sklearn.linear_model import LinearRegression
from sklearn.linear_model import Ridge
from sklearn.linear_model import Lasso
```

读取文件

```
data=pd.read_csv(path)
x_train=np.load(path+"\\"+"x_train.npy")
x_train=pd.read_csv(path+"xtrain.csv",index_col=False)
data=pd.read_table(path,header=None)
```

数据操作

```
data.shape
data.index
data.columns
X=data[['age','education','marital_status','sex' ,'hours_per_week']]
X["education"].unique()
X_train=X.iloc[:-1000] #训练数据
np.random.shuffle(inds) #打乱，洗牌
x_train,x_test,y_train,y_test= train_test_split(
    x,y,train_size=0.8, random_state=1) #数据自动切割
```

存储

```
joblib.dump(knn,
            r"C:\Users\Tsinghua-yincheng\Desktop\SZday92\ren.m")
mynewknn=joblib.load(
    r"C:\Users\Tsinghua-yincheng\Desktop\SZday92\ren.m")

plt.imsave(r"C:\Users\Tsinghua-yincheng\Desktop\SZday92\1.bmp",zero1) #数组保存为
图片
```

解决绘图中文乱码

```
import matplotlib as mpl
mpl.rcParams['font.sans-serif'] = ['simHei']
mpl.rcParams['axes.unicode_minus'] = False#中文乱码
```

网格搜索

```
lasso_model= GridSearchCV(model,
                          param_grid={"alpha":alpha_scan},
                          cv=10) #自动匹配最佳alpha
lasso_model.fit(x_train,y_train) #训练数据
lasso_model.best_params_  #寻找最佳alpha
```

PCA降维

```
from sklearn.svm  import SVC  #分类
from  sklearn.decomposition  import PCA #降维
svc=SVC()
pca=PCA(n_components=150,svd_solver="randomized",whiten=True).fit(x_train) #降维
x_train_pca=pca.transform(x_train) #训练的数据转化
x_test_pca=pca.transform(x_test) #测试的数据转化
svc.fit(x_train_pca,y_train) #训练
svc.score(x_test_pca,y_test)#评分
```

字符串转换为数字

```
for i in data[['school', 'sex','address', 'famsize', 'Pstatus',
       'Mjob', 'Fjob', 'reason', 'guardian',
        'schoolsup', 'famsup', 'paid', 'activities', 'nursery',
       'higher', 'internet', 'romantic', 'passed']]:
    stu=data[i].unique() #不重复
    def transform(i): #每个字符串匹配一个数字
        return np.argwhere(stu==i)[0,0]
    data[i]=data[i].map(transform) #所有特征，数字化
```

数据预处理

```
#归一操作 ，数字转化0-1之间
for i in data.columns:
    sum=data[i].sum()  #总和
    data[i]=data[i]/sum #比例
print(data)

#将目标值转为0,1 代表通过或者不通过
for i in data['passed']:
    if i :
        data['passed']=i

#将data中的部分列复制提取为X
X=data[[ 'Pstatus', 'Medu', 'Fedu',
       'Mjob', 'Fjob', 'reason', 'guardian', 'traveltime', 'studytime',
       'failures', 'schoolsup', 'famsup', 'paid', 'activities', 'nursery',
       'higher', 'internet', 'romantic', 'famrel', 'freetime', 'goout', 'Dalc',
       'Walc', 'health','passed']].copy() #x深复制
#去掉偏离波动大的值
X=X.drop(X[(np.abs(X-X.mean()) > (2*X.std())).any(axis=1)].index) #去掉极端数据
print(X)
y=X['passed'].copy()
print(y)
print(X.shape,y.shape)
```

交叉验证

```
from  sklearn.cross_validation import cross_val_score  #交叉验证
knn=KNeighborsClassifier (n_neighbors=5)
cross_val_score(knn,x,y ,cv=5,scoring="accuracy") #交叉验证

for  k   in  k_range:
    knn_tmp=KNeighborsClassifier (n_neighbors=k)
    scores=cross_val_score(knn_tmp,x,y ,cv=10,scoring="accuracy") #交叉验证
    k_score.append((k,scores.mean()))
k_range
```