上海理工大学光电信息与计算机工程学院

**《数据挖掘》实验报告**

****

**专　　业 智能科学与技术**

**姓 名　　 高浩琦**

**学　 号 2035060413**

**年　　级 2020级**

**指导教师 孙占全**

**成 绩：**

**教师签字：**

# 实验二 KNN、Bayes和SVM分类

1. 实验目的和要求

1、掌握kNN分类器的基本原理，利用kNN分类器实现wine.csv数据分类，40%做测试；

2、在SVM的分类算法理论学习基础上，通过实验了解SVM分类的应用；

1. 掌握用Python使用SVM算法对数字图像识别digit数据集进行分类，测试分类准确率，并画出学习曲线；

4、学会使用Python训练和测试Bayes分类器，利用Bayes分类器对mashroom.csv数据进行分类，并对结果进行分A析，画ROC曲线。

二、实验算法

KNN

from sklearn.datasets import load\_wine

from sklearn.neighbors import KNeighborsClassifier

from sklearn.model\_selection import train\_test\_split

import pandas as pd

import matplotlib.pyplot as plt

data = load\_wine()

data\_x = data.data

data\_y = data.target

xtrain,xtest,ytrain,ytest = train\_test\_split(data\_x,data\_y,test\_size = 0.4 ,random\_state = 1 )

# print(xtest.shape)

score = []

k = [3,5,7]

i = 0

for x in k:

    knn = KNeighborsClassifier(n\_neighbors = x)

    clf = knn.fit(xtrain,ytrain)

    score.append(clf.score(xtest,ytest))

    i = i + 1

print("k = 3:",score[0])

print("k = 5:",score[1])

print("k = 7",score[2])

fig = plt.figure(alpha = 0.5)

plt.bar(k,score,color = 'b')

plt.xlabel = "k"

plt.ylabel = "ppp"

# plt.legend = "ss"

# fig.align\_ylabels = "p"

plt.show()

BAYES

from sklearn.preprocessing import LabelEncoder

import pandas as pd

import matplotlib.pyplot as plt

from sklearn.model\_selection import train\_test\_split

from sklearn.preprocessing import StandardScaler

from sklearn.naive\_bayes import GaussianNB

from sklearn.naive\_bayes import BernoulliNB

data = pd.read\_csv("mushroom.csv",header= None)

## 字母转化为数字 ##

labelincoder = LabelEncoder()

for col in data.columns:

    data[col] = labelincoder.fit\_transform(data[col])

    # print(col)

## 提取 类标签和变量属性

y = data[0]

x = data.drop(0, axis = 1)

xtrain,xtest,ytrain,ytest = train\_test\_split(x,y,test\_size = 0.3 ,random\_state = 1 )

## 标准化

ss\_x = StandardScaler()

xtrain = ss\_x.fit\_transform(xtrain)

xtest = ss\_x.fit\_transform(xtest)

## 计算模型

clf = GaussianNB()

model\_tree = clf.fit(xtrain,ytrain)

score = []

score.append(model\_tree.score(xtest,ytest))

# clf = MultinomialNB()

# model\_tree = clf.fit(xtrain,ytrain)

# score = model\_tree.score(xtest,ytest)

# print(score)

clf = BernoulliNB()

model\_tree = clf.fit(xtrain,ytrain)

score.append(model\_tree.score(xtest,ytest))

print("GaussianNB:",score[0])

print("BernoulliNB:",score[0])

SVM

import matplotlib.pylab as plt

import pandas as pd

from sklearn.datasets import load\_digits

digits = load\_digits()

y = digits.target

# %%

df = pd.DataFrame(data = digits.data)

df.head()

# %%

from sklearn.model\_selection import KFold

kf = KFold(n\_splits= 6, shuffle= True)

# %%

df.loc[1:6]

# %%

# for xtrain,xtest in kf.split():

from sklearn.svm import SVC

from sklearn.preprocessing import StandardScaler

score = []

for train\_index, test\_index in kf.split(df):

    # print("TRAIN:", train\_index.size, "\nTEST:", test\_index.size)

    X\_train, X\_test = df.loc[train\_index], df.loc[test\_index]

    Y\_train, Y\_test = y[train\_index], y[test\_index]

    # ss\_x = StandardScaler()

    # X\_train = ss\_x.fit\_transform(X\_train)

    # X\_test = ss\_x.fit\_transform(X\_test)

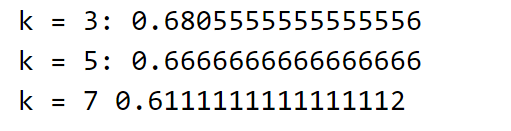
    svm = SVC()

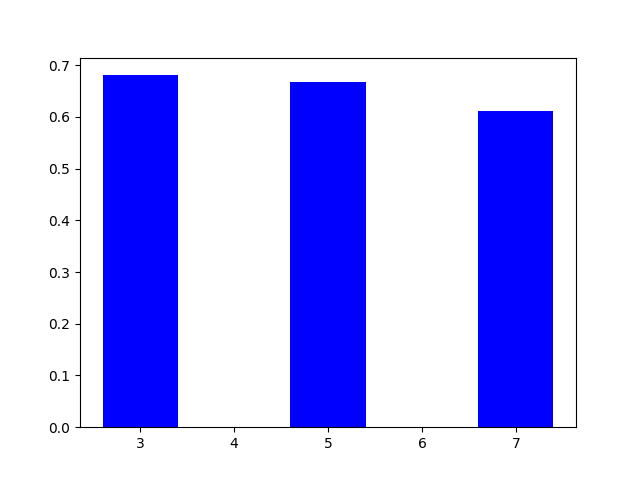
    dlf = svm.fit(X\_train,Y\_train)

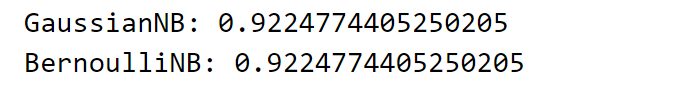
    score.append(dlf.score(X\_test,Y\_test))

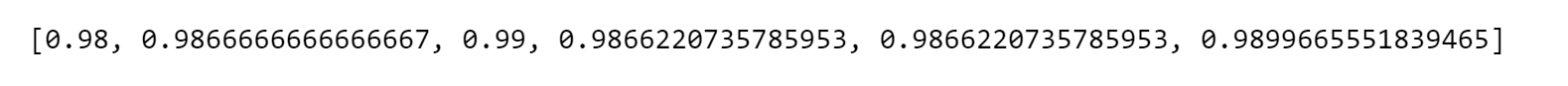
print(score)

三、实验过程记录

1. KNN



2. Bayes

3. SVM

四、结果与分析

1. 由结果可知，KNN算法对于wine数据来说并不是一个很好的拟合算法，在k = 3,5,7 拟合概率均在0.6上下，由此可见KNN算法对训练数据依赖度特别大，对训练数据的容错性太差。如果训练数据集中，有一两个数据是错误的，刚刚好又在需要分类的数值的旁边，这样就会直接导致预测的数据的不准确。所以在预测该类型数据的时候我们要尽量避免使用这种算法。

2. 贝叶斯分类器的分类原理是通过某对象的先验概率，从几个利用贝叶斯公式计算出其后验概率，即该对象属于某一类的概率，选择具有最大后验概率的类作为该对象所属的类。也就是说，贝叶斯分类器是最小错误率意义上的优化。目前研究较多的贝叶斯分类器主要有四种，分别是：Naive Bayes、TAN、BAN和GBN。实验中采用了GBN和BAN分类器，结果均超过了90%，非常不错。

3．SVM分类思想很简单，就是将样本与决策面的间隔最大化，实验中采用了K-折交叉验证，结果产生了很好的拟合优度，分类效果较好。

五、自评

通过三次的实验课后，基本明白了数据挖掘对数据的处理与相关模型的拟合，掌握了python-sklearn库的使用，收获满满。