**河海大学**

《机器学习》课程实验报告

姓名： 陈督

学号： 171807020003

专业：计算机技术(非全)

联系方式：18351960995

实验报告

**摘要**

本实验采用鸢尾花(Iris)作为数据集，进行机器学习。

1. **实验环境**

Windows 10

Anaconda 1.7.2

Python 3.6.0

PyCharm 2017.3.4 (Professional Edition)

Visual Studio Code 1.28.2

1. **实验内容**

2.1 提取数据

读取iris.csv数据，根据“species”即按照不同花的类别进行分组，在iris.csv数据集中，共有三种类别的鸢尾花，分别为：

setosa, versicolour, virginica

分别求该三类花中花萼、花瓣的长、宽度各属性数据的平均值：

import pandas as pd

from matplotlib import pyplot as plt

iris\_data = pd.read\_csv("iris.csv")

grouped\_data = iris\_data.groupby("species")

group\_mean = grouped\_data.mean()

group\_mean.plot(kind="bar")

plt.legend(loc="upper center", bbox\_to\_anchor=(0.5, 1.2), ncol=2)

plt.show()

如图1条形图所示，可以看出三类不同的花种类中的属性特点：

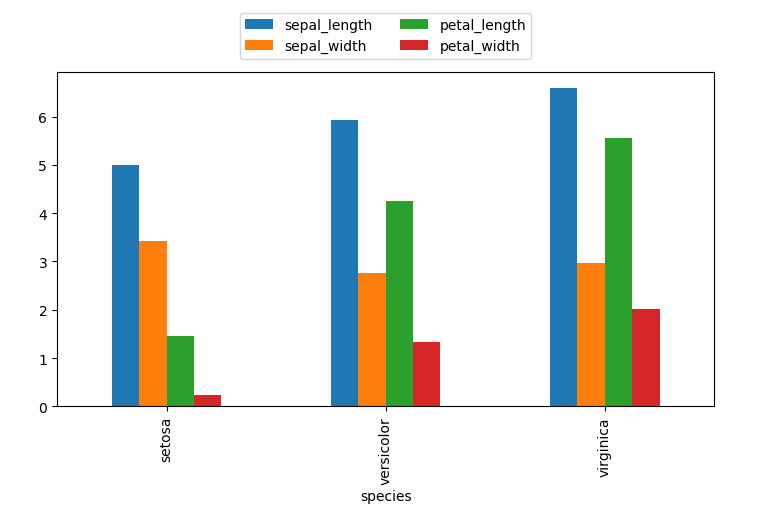
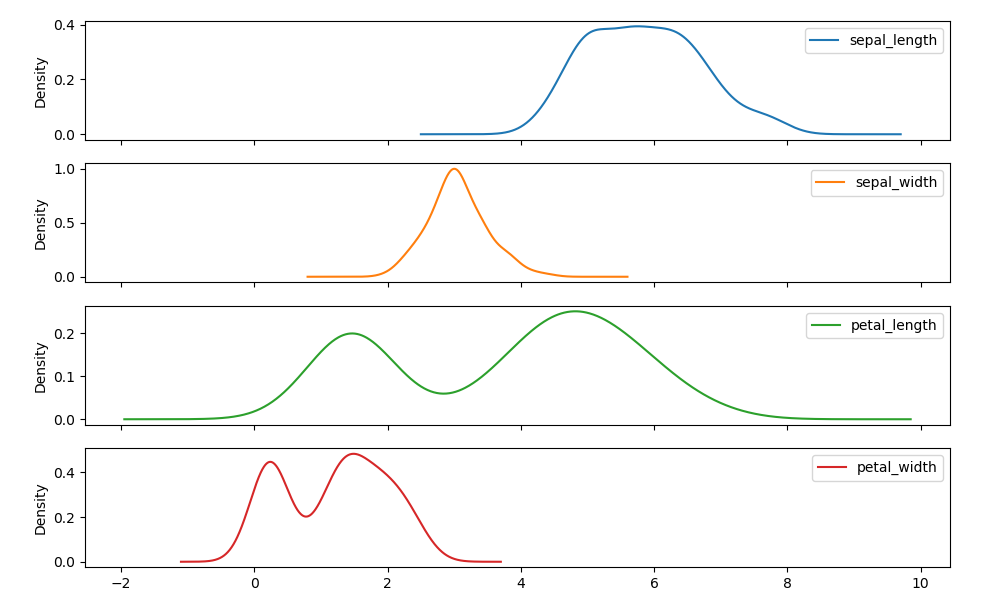


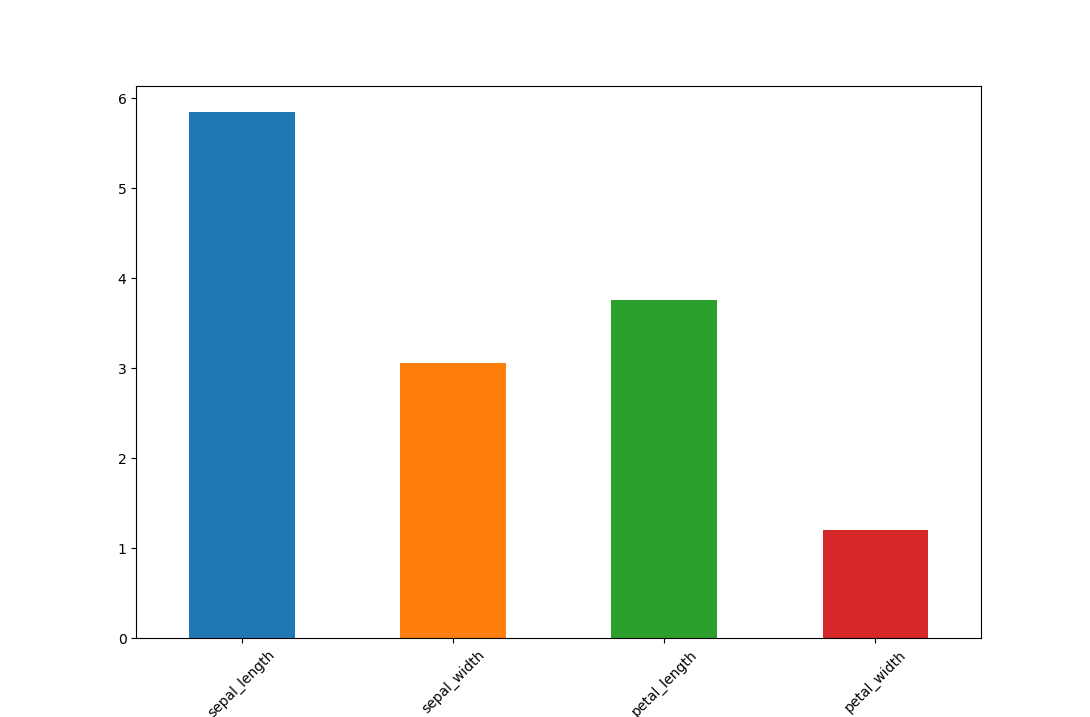
图1.三类鸢尾花属性特征平均值

高斯分布密度图

iris\_data.plot(kind="kde",subplots=True,figsize=(10,6))

plt.show()



四条属性特征的平均值，条状图：

1. **实验分析**

* 交叉验证训练数据集：

import pandas as pd

import pandas as pd

from sklearn import model\_selection

from sklearn.neighbors import KNeighborsClassifier

x = pd.read\_csv("iris.csv")

y = x.pop("species")

x\_train, x\_test, y\_train, y\_test = model\_selection.train\_test\_split(x.values, y.values, test\_size=0.1)

scores = model\_selection.cross\_val\_score(KNeighborsClassifier(3), x, y, cv=5)

mean\_score = scores.mean()

print("mean score: "+str(mean\_score))

输出：

mean score: 0.9666666666666668

* KNN训练数据集：

import pandas as pd

from sklearn import model\_selection

from sklearn.neighbors import KNeighborsClassifier

x = pd.read\_csv("iris.csv")

y = x.pop("species")

x\_train, x\_test, y\_train, y\_test = model\_selection.train\_test\_split(x.values, y.values, test\_size=0.1)

knn = KNeighborsClassifier(3).fit(x\_train, y\_train)

for y\_pred, y\_true in zip(knn.predict(x\_test), y\_test):

print(y\_pred, y\_true)

print(knn.score(x\_test, y\_test))

输出：

virginica virginica

setosa setosa

versicolor versicolor

virginica virginica

versicolor versicolor

versicolor versicolor

setosa setosa

versicolor versicolor

versicolor versicolor

versicolor versicolor

virginica versicolor

setosa setosa

versicolor versicolor

setosa setosa

versicolor versicolor

Knn score:0.9333333333333333

验证精度0.93

* 逻辑斯蒂回归 训练数据集：

import pandas as pd

from sklearn import model\_selection

from sklearn.linear\_model import LogisticRegression

x = pd.read\_csv("iris.csv")

y = x.pop("species")

x\_train, x\_test, y\_train, y\_test = model\_selection.train\_test\_split(x.values, y.values, test\_size=0.1)

lr = LogisticRegression(multi\_class="multinomial", solver="lbfgs").fit(x\_train, y\_train)

print(lr.predict\_proba(x\_test))

输出：

[[9.70533456e-01 2.94662513e-02 2.92728063e-07]

[9.78989257e-03 9.74624390e-01 1.55857172e-02]

[9.69292708e-01 3.07071685e-02 1.23156410e-07]

[1.20298053e-03 7.05650909e-01 2.93146110e-01]

[6.66863373e-04 4.04833308e-01 5.94499829e-01]

[9.80416610e-01 1.95833446e-02 4.49615031e-08]

[4.05023923e-03 6.90482182e-01 3.05467579e-01]

[9.87885928e-01 1.21140642e-02 7.35428893e-09]

[1.55757335e-06 3.44820698e-02 9.65516373e-01]

[2.17728246e-05 1.48699498e-01 8.51278729e-01]

[1.63208805e-04 1.64518095e-01 8.35318696e-01]

[9.24693785e-03 9.26279873e-01 6.44731892e-02]

[5.89182143e-03 9.13945965e-01 8.01622137e-02]

[9.85734470e-01 1.42654852e-02 4.49619993e-08]

[2.50520233e-02 9.56135088e-01 1.88128888e-02]]

图上为预测的数据和分组的结果。

* 朴素贝叶斯训练数据集：

import pandas as pd

from sklearn import model\_selection

from sklearn.naive\_bayes import GaussianNB

x = pd.read\_csv("iris.csv")

y = x.pop("species")

x\_train, x\_test, y\_train, y\_test = model\_selection.train\_test\_split(x.values, y.values, test\_size=0.1)

gnb = GaussianNB().fit(x\_train, y\_train)

print(gnb.predict\_proba(x\_test))

print(gnb.class\_prior\_)

输出：

[[1.25857863e-135 7.80527634e-002 9.21947237e-001]

[6.14762838e-097 9.93511611e-001 6.48838893e-003]

[2.51589156e-041 9.99999376e-001 6.24248983e-007]

[5.50645457e-192 1.19465265e-007 9.99999881e-001]

[1.31104544e-129 1.68556832e-001 8.31443168e-001]

[1.56742488e-059 9.99997731e-001 2.26912150e-006]

[2.49428820e-184 1.30940118e-007 9.99999869e-001]

[1.00000000e+000 6.37900597e-015 1.77962092e-022]

[9.71285190e-178 8.81342899e-004 9.99118657e-001]

[1.01734984e-076 9.99894744e-001 1.05256199e-004]

[3.15299852e-233 2.57404568e-011 1.00000000e+000]

[1.36753683e-201 5.12488039e-007 9.99999488e-001]

[4.20900054e-086 9.96905261e-001 3.09473932e-003]

[1.00000000e+000 3.34079292e-019 1.32417298e-026]

[8.59613656e-123 9.72943637e-001 2.70563635e-002]]

[0.35555556 0.32592593 0.31851852]

图上为预测的数据和分组的结果。

1. **实验总结**

通过sklearn的交叉验证、KNN、逻辑蒂斯回归这三种方式分别训练数据集并进行对比。

import pandas as pd

from sklearn import model\_selection

from sklearn.naive\_bayes import GaussianNB

from sklearn.neighbors import KNeighborsClassifier

from sklearn.linear\_model import LogisticRegression

x = pd.read\_csv("iris.csv")

y = x.pop("species")

x\_train, x\_test, y\_train, y\_test = model\_selection.train\_test\_split(x.values, y.values, test\_size=0.1)

models = {

"knn": KNeighborsClassifier(6),

"gnb": GaussianNB(),

"lr": LogisticRegression(multi\_class="multinomial", solver="lbfgs")

}

for name, model in models.items():

score = model\_selection.cross\_val\_score(model, x, y, cv=5).mean()

print(name, score)

输出：

knn 0.9800000000000001

gnb 0.9533333333333334

lr 0.9733333333333334

即KNN算法的验证精度为0.98，朴素贝叶斯算法的验证精度为0.95，逻辑斯蒂回归算法的验证精度为0.97。可以看出KNN算法的验证精度更高。

**附件：**

压缩包为python代码。