

机器学习与模式识别

作业题目：SVM、贝叶斯

学 院： 计算机科学与技术学院

专 业： 软件工程

班 级： 软工171

姓 名： 吴政文

学 号： 1777000147

指导教师： 赵兴成

2020年6月26日

# 3.SVM、贝叶斯（实验报告本）

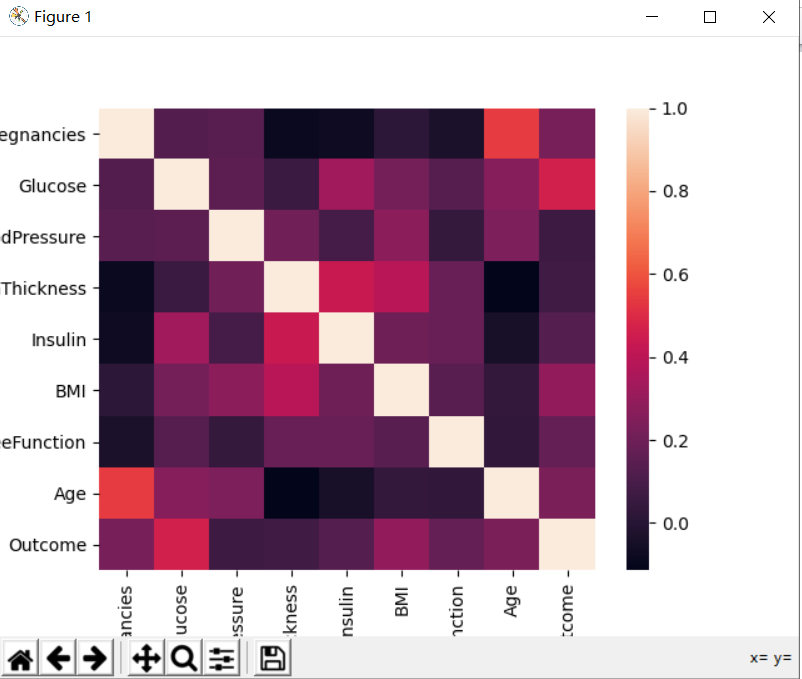
## 贝叶斯（糖尿病病情预测）

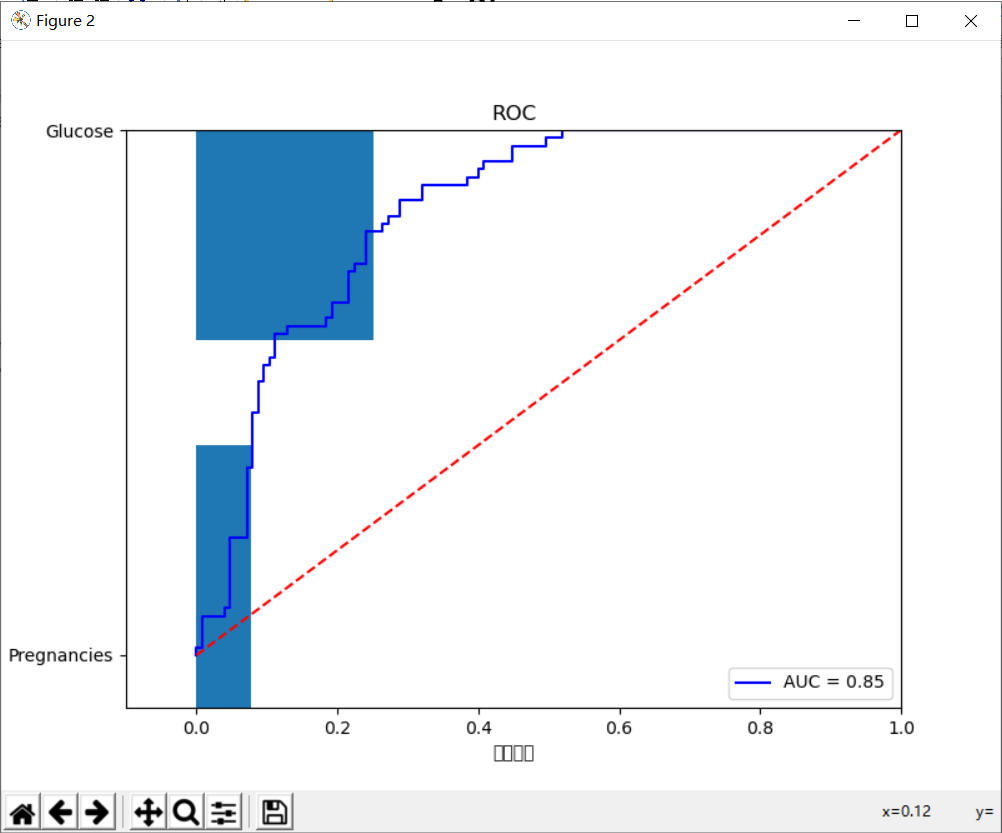
代码：

import pandas as pd  
import numpy as np  
import matplotlib.pyplot as plt  
import seaborn as sns  
from sklearn.model\_selection import train\_test\_split  
from sklearn.ensemble import RandomForestClassifier  
from sklearn.naive\_bayes import GaussianNB  
  
diabetes = pd.read\_csv("diabetes.csv")  
  
print(diabetes.head())  
print(diabetes.info())  
print(diabetes.describe())  
print(diabetes.Outcome.value\_counts())  
# 2.数据分析  
# %matplotlib inline # ipynb调用画板的申明，在py环境中直接删除即可  
print(sns.heatmap(diabetes.corr()))  
# 3.数据集划分  
X\_train, X\_test, y\_train, y\_test = train\_test\_split(  
 diabetes.loc[:, diabetes.columns != 'Outcome'],  
 diabetes['Outcome'],  
 stratify=diabetes['Outcome'],  
 random\_state=66  
)  
print(X\_train)  
print(y\_train.head())  
# 4.特征工程  
rf = RandomForestClassifier(n\_estimators=100, random\_state=0)  
print(rf.fit(X\_train, y\_train))  
  
print("Accuracy on training set: {:.3f}".format(rf.score(X\_train, y\_train)))  
print("Accuracy on training set: {:.3f}".format(rf.score(X\_test, y\_test)))  
diabetes\_features = [x for i, x in enumerate(diabetes.columns) if i != 8] # \*\*\*  
  
  
def plot\_feature\_importances\_diabetes(model):  
 plt.figure(figsize=(8, 6))  
 n\_features = 8  
 plt.barh(range(n\_features), model.feature\_importances\_, align='center')  
 plt.yticks(np.arange(n\_features), diabetes\_features)  
 plt.xlabel("Feature importance")  
 plt.ylabel("Feature")  
 plt.ylim(-1, n\_features)  
  
  
plot\_feature\_importances\_diabetes(rf)  
# 5.模型引入  
NBmodel = GaussianNB()  
NBmodel.fit(X\_train, y\_train)  
  
# 模型评价  
NBmodel.score(X\_train,y\_train)  
from sklearn.metrics import roc\_curve, auc  
y\_pred\_score = NBmodel.predict\_proba(X\_test)  
y\_pred\_score  
fpr, tpr, thresholds = roc\_curve(y\_test, y\_pred\_score[:, 1])  
roc\_auc = auc(fpr,tpr)  
print(roc\_auc)  
  
plt.title('ROC')  
plt.plot(fpr, tpr, 'b', label='AUC = %0.2f' % roc\_auc)  
plt.legend(loc='lower right')  
plt.plot([0, 1], [0, 1], 'r--')  
plt.xlim([-0.1, 1.0])  
plt.ylim([-0.1, 1.0])  
plt.ylabel('正样本率')  
plt.xlabel('负样本率')  
plt.show()

结果截图：







## 贝叶斯（鸢尾花分类）

代码：

rom sklearn.model\_selection import train\_test\_split  
from sklearn.datasets import load\_iris  
import graphviz  
from sklearn import tree  
from sklearn.tree import DecisionTreeClassifier  
from sklearn.metrics import accuracy\_score  
  
# 1.鸢尾花数据读取、数据观察和初步分析  
iris = load\_iris()  
feature = iris.feature\_names  
print(feature)  
  
target = iris.target\_names  
print(target)  
  
data = iris.data[:3]  
print(data)  
  
target\_ = iris.target[:3]  
print(target\_)  
# 观察样本大小  
print(len(iris.data))  
print(len(iris.target))  
  
# 2.鸢尾花数据分析  
print("平均花萼长度%.2fcm" % iris.data[:,0].mean())  
print("平均花萼宽度%.2fcm" % iris.data[:,1].mean())  
print("平均花瓣长度%.2fcm" % iris.data[:,2].mean())  
print("平均花瓣宽度%.2fcm" % iris.data[:,3].mean())  
# 3.鸢尾花数据分析结果可视化  
# 省略  
# 4.鸢尾花数据集划分  
X\_train, X\_test, y\_train, y\_test = train\_test\_split(iris.data,  
 iris.target,  
 test\_size=0.2,  
 random\_state=0)  
# print(X\_train, X\_test, y\_train, y\_test)  
# 5.引入决策树模型  
# 将决策树模型的目标函数设定为信息增益  
clf = DecisionTreeClassifier(criterion="entropy")  
clf = clf.fit(X\_train, y\_train)  
print(clf)  
# 将决策树模型的目标函数设定为基尼指数，即默认参数  
clf2 = DecisionTreeClassifier()  
clf2 = clf.fit(X\_train, y\_train)  
  
# 6.模型评价准确率  
iris\_predicted1 = clf.predict(X\_test)  
iris\_predicted2 = clf2.predict(X\_test)  
# 我们从机器学习的评价库中调用准确率指标  
print('当目标函数为信息增益时，模型准确率为：{:.2f}'.format(accuracy\_score(y\_test, iris\_predicted1)))  
print('当目标函数为基尼指数时，模型准确率为：{:.2f}'.format(accuracy\_score(y\_test, iris\_predicted2)))  
  
# 7.划分结果可视化  
dot\_data = tree.export\_graphviz(clf2,  
 out\_file=None,  
 feature\_names=iris.feature\_names,  
 class\_names=iris.target\_names,  
 filled=True, rounded=True,  
 special\_characters=True)  
graph2 = graphviz.Source(dot\_data)  
# 显示画出的决策图生成pdf文件  
graph2.view()  
# 重命名  
graph2.render('鸢尾花分类', view=True)

结果截图：

