# 学生实验报告（参考模板）

|  |  |  |  |
| --- | --- | --- | --- |
| 学号 | 2200400826 | 学院 | 计算机科学与技术学院 |
| 姓名 | 许文鑫 | 专业 | 人工智能 |

## 实验名称

SVM分类模型

## 实验目的

1. 掌握样本处理过程，原始数据的训练集、测试集切分方式
2. 理解集体学习算法的原理和处理流程
3. 进行模型训练、测试及结果分析的能力

## 实验原理

（1）pandas的数据分析方法库和数据模型

（2）sklearn工具包进行数据集选择、切分和参数优化

（3）GridSearchCV 进行参数调优

（4）SVM模型原理

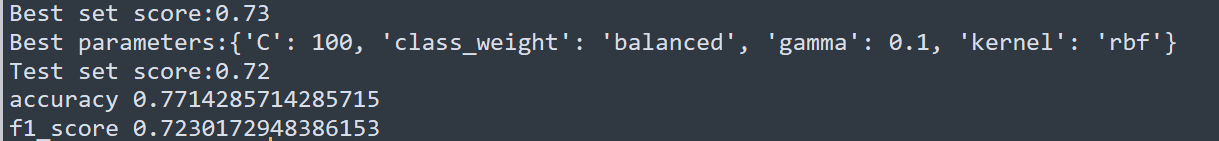
## 实验条件与环境

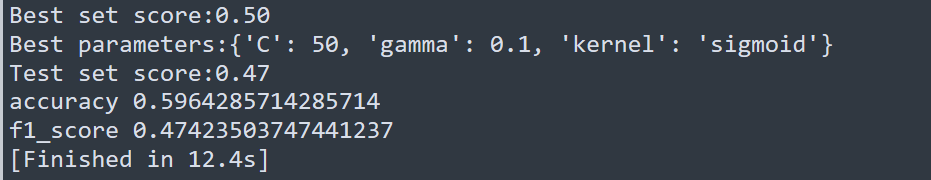
|  |  |  |  |
| --- | --- | --- | --- |
| 要求 | 名称 | 版本要求 | 备注 |
| 编程语言 | Python | 3.6以上 |  |
| 开发环境 | 阿里云DSW平台 |  |  |
| 第三方工具包/库/插件 | Numpy | 1.15.0及以上 |  |
| 第三方工具包/库/插件 | Pandas | 0.25及以上 |  |
| 第三方工具包/库/插件 | Matplotlib | 3.0.0及以上 |  |
| 第三方工具包/库/插件 | Seaborn | 0.11及以上 |  |
| 第三方工具包/库/插件 | Sklearn | 1.19.0及以上 |  |
| 其他工具 | 无 |  |  |
| 硬件环境 | 台式机、笔记本均可 | 无要求 |  |

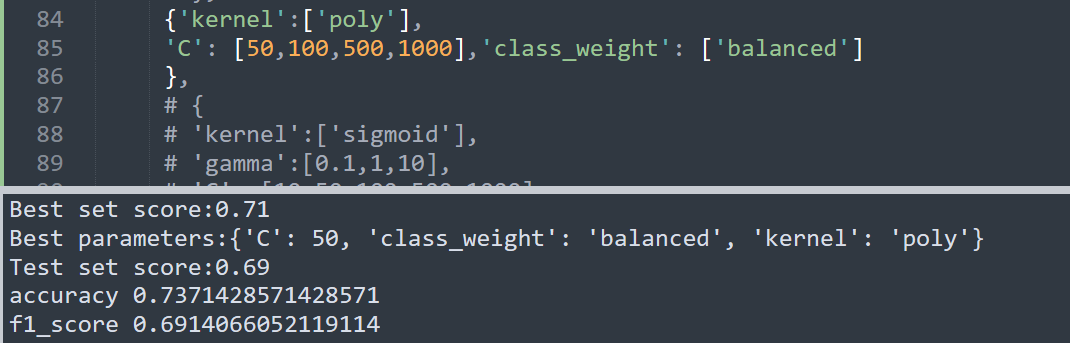
## 实验步骤及操作

|  |  |  |  |
| --- | --- | --- | --- |
| 序号 | 步骤名称 | 步骤描述 | 代码 |
| 1 | 读取数据，特征抽取 | 载入训练数据，数据预处理、数据清洗、把方向转换为弧度制，增加他们的sin,cos值 | import os  import numpy as np  import pandas as pd  import datetime  from sklearn.model\_selection import \*  from sklearn.svm import \*  from sklearn.metrics import \*  path = './hy\_round1\_testA\_20200102'  total = pd.DataFrame()  for dirname, \_, filenames in os.walk(path):  for filename in filenames:  if filename == '7000.csv':  part = pd.read\_csv(f'./hy\_round1\_testA\_20200102/{filename}')  total = part  else:  print(filename)  part = pd.read\_csv(f'./hy\_round1\_testA\_20200102/{filename}')  total = pd.concat([total, part], axis=0, ignore\_index=True)  total.to\_feather("test.feather")  train = pd.read\_feather(‘train.feather’)  def feature\_engineer(df, test=False):      df['speed'] = df['speed']      df['ori'] = df['ori'] / 180.0 \* np.pi      df['speed\_sin'] = df['speed'] \* np.sin(df['ori'])      df['speed\_cos'] = df['speed'] \* np.cos(df['ori'])        if test:          df = df.groupby(['id']).agg({'x': ['std', 'min', 'max', 'mean'],                                       'y': ['std', 'min', 'max', 'mean'],                                       'speed\_sin': ['std', 'min', 'max', 'mean'],                                       'speed\_cos': ['std', 'min', 'max', 'mean'],                                       'speed': ['std', 'min', 'max', 'mean'],                                       'ori': ['std', 'min', 'max', 'mean']}).reset\_index()          df.columns = ['id',                        'x\_std', 'x\_min', 'x\_max', 'x\_mean',                        'y\_std', 'y\_min', 'y\_max', 'y\_mean',                        'speed\_sin\_std', 'speed\_sin\_min', 'speed\_sin\_max', 'speed\_sin\_mean',                        'speed\_cos\_std', 'speed\_cos\_min', 'speed\_cos\_max', 'speed\_cos\_mean',                        'speed\_std', 'speed\_min', 'speed\_max', 'speed\_mean',                        'ori\_std', 'ori\_min', 'ori\_max', 'ori\_mean']        else:          df = df.groupby(['id', 'type']).agg({'x': ['std', 'min', 'max', 'mean'],                                               'y': ['std', 'min', 'max', 'mean'],                                               'speed\_sin': ['std', 'min', 'max', 'mean'],                                               'speed\_cos': ['std', 'min', 'max', 'mean'],                                               'speed': ['std', 'min', 'max', 'mean'],                                               'ori': ['std', 'min', 'max', 'mean']}).reset\_index()          df.columns = ['id', 'type',                        'x\_std', 'x\_min', 'x\_max', 'x\_mean',                        'y\_std', 'y\_min', 'y\_max', 'y\_mean',                        'speed\_sin\_std', 'speed\_sin\_min', 'speed\_sin\_max', 'speed\_sin\_mean',                        'speed\_cos\_std', 'speed\_cos\_min', 'speed\_cos\_max', 'speed\_cos\_mean',                        'speed\_std', 'speed\_min', 'speed\_max', 'speed\_mean',                        'ori\_std', 'ori\_min', 'ori\_max', 'ori\_mean']            return df  train = feature\_engineer(train) |
| 2 | 数据特征选择 | 选择出比较重要的特征作为分类依据 | important = ['x\_std', 'x\_min', 'x\_max', 'x\_mean',  'y\_std', 'y\_min', 'y\_max', 'y\_mean',  'speed\_sin\_std', 'speed\_sin\_max', 'speed\_sin\_mean',  'speed\_cos\_std', 'speed\_cos\_max', 'speed\_cos\_mean',  'speed\_std', 'speed\_max', 'speed\_mean',  'ori\_std', 'ori\_max', 'ori\_mean']  num = ['id',  'x\_std', 'x\_min', 'x\_max', 'x\_mean',  'y\_std', 'y\_min', 'y\_max', 'y\_mean',  'speed\_sin\_std', 'speed\_sin\_min', 'speed\_sin\_max', 'speed\_sin\_mean',  'speed\_cos\_std', 'speed\_cos\_min', 'speed\_cos\_max', 'speed\_cos\_mean',  'speed\_std', 'speed\_min', 'speed\_max', 'speed\_mean',  'ori\_std', 'ori\_min', 'ori\_max', 'ori\_mean']  X = pd.read\_feather('train.feather')  X.columns = ['id', 'x', 'y', 'speed', 'ori', 'time', 'type']  type\_dict = {'围网':0, '拖网':1, '刺网':2}  X.type = X.type.map(type\_dict)  X = feature\_engineer(X)  scaler = StandardScaler()  X[important] = scaler.fit\_transform(X[important])  Y = X['type']  X = X[important] |
| 3 | 模型训练 | 训练集切分，参数配置，模型训练、调优 | X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, Y, test\_size=0.2, random\_state=0)  # 将参数写成字典下形式  parameters = [  # {'kernel': ['rbf'], #核函数  # 'C': [10,50,100,500,1000], #惩罚因子  # 'gamma': [0.1, 1, 10], #径向基函数的系数  # 'class\_weight': ['balanced'] #样本均衡度  # },  {'kernel':['linear'],  'C':[10,50,100,500,1000], 'class\_weight': ['balanced']  },  # {'kernel':['poly'],  # 'C': [50,100,500,1000],'class\_weight': ['balanced']  # },  # {  # 'kernel':['sigmoid'],  # 'gamma':[0.1,1,10],  # 'C': [10,50,100,500,1000],  # }  ]  # 参数调优  clf = GridSearchCV(estimator=SVC(), param\_grid=parameters, cv=5, n\_jobs=5, scoring='f1\_macro')  clf.fit(X\_train, y\_train)  print("Best set score:{:.2f}".format(clf.best\_score\_))  print("Best parameters:{}".format(clf.best\_params\_))  print("Test set score:{:.2f}".format(clf.score(X\_test, y\_test))) |
| 3 | 模型测试 | 进行模型测试，分析测试结果，尝试不同的核函数及其他参数对分类结果的影响 | predicted = clf.predict(X\_test) # 模型预测  accuracy = accuracy\_score(y\_test, predicted)  print("accuracy", accuracy)  print("f1\_score",f1\_score(y\_test,predicted,labels=[0,1,2],average='macro')) |

## 实验结果及分析讨论







多个核函数对比可知，径向基核函数的准确率最高，分类效果最好。

## 收获与体会

改变了特征向量准确率显著上升。

在课堂上理解了SVM基本原理，适用范围后，在实验课上实现了SVM，并在此过程中，又加深了传统机器学习算法的训练、测试处理流程，进一步掌握样本处理、数据预处理过程，提高了利用sklearn工具包进行SVM训练、测试及结果分析的能力。加深了我对SVM的理解。

## 备注及其他

无