作业答案

1. 使用思维导图总结逻辑回归的内容

略

2. 理解分类评估方法并进行详细的描述
    - 准确率：所有样本中预测正确的样本比例
    - 精确率：预测为正例样本中真正为样本的比例，查准率
    - 召回率：真实为正例的样本中预测为正样本的比例，查全率
    - f1-score: 精确率和召回率的组合
    - ROC曲线和AUC指标：
        - ROC是以FPR和TPR绘制的模型评估的曲线
        - AUC是ROC的曲线下面积，取值在0-1之间，一般是大于0.5

3.动手实现癌症分类和电信用户流失案例(数据处理，特征工程，CV。。。。)

```
#癌症分类
import numpy as np
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression
from sklearn.preprocessing import StandardScaler
from sklearn.metrics import accuracy_score

data = pd.read_csv('data/breast-cancer-wisconsin.csv')
data.info()
data = data.replace(to_replace='?',value=np.nan)
data=data.dropna()

x = data.iloc[:,1:-1]
y=data['Class']


x_train,x_test,y_train,y_test =
train_test_split(x,y,test_size=0.2,random_state=22)

transfer = StandardScaler()
x_train = transfer.fit_transform(x_train)
x_test = transfer.transform(x_test)

model = LogisticRegression()#逻辑回归

model.fit(x_train,y_train)

my_pred = model.predict(x_test)
print(my_pred)

print('预测结果-->',accuracy_score(y_test, my_pred))
#结果:预测结果--> 0.9854014598540146
```

```python
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression
from sklearn.preprocessing import StandardScaler
from sklearn.metrics import accuracy_score
from sklearn.metrics import roc_auc_score

data = pd.read_csv('data/churn.csv')

data = pd.get_dummies(data)
data.drop(['Churn_No', 'gender_Male'],axis=1,inplace=True)
data.rename(columns={'Churn_Yes':'flag'},inplace=True)

x = data[['Contract_Month', 'internet_other', 'PaymentElectronic']]
y = data['flag']

x_train,x_test,y_train,y_test =
train_test_split(x,y,test_size=0.2,random_state=22)

transfer = StandardScaler()
x_train = transfer.fit_transform(x_train)
x_test = transfer.transform(x_test)

model = LogisticRegression()#逻辑回归

model.fit(x_train,y_train)

my_pred = model.predict(x_test)
print(my_pred)

print('预测结果-->',accuracy_score(y_test, my_pred))
my_score = model.score(x_test, y_test)
print('my_score-->',my_score)

# 计算AUC值
my_roc_auc_score = roc_auc_score(y_test, my_pred)
print('my_roc_auc_score-->', my_roc_auc_score)
```