作业答案

1.说明常见的距离度量方法

- 欧式距离法
  - 误差平方求和开根号（开平方）
  - $d_{12}=\sqrt{(x_1-x_2)^2+(y_1-y_2)^2}$
- 曼哈顿距离法
  - 误差绝对值求和
  - $$d_{12}=|x_1-x_2|+|y_1-y_2|$$
- 切比雪夫距离
  - 误差绝对值求最大值
  - $d_{12}=\max(|x_1-x_2|,|y_1-y_2|)$
- 闵可夫斯基距离、

$$d_{12}=\sqrt[p]{\sum_{k=1}^{n}|x_{1k}-x_{2k}|^p}$$

  - 不是一种新的距离的度量方式。
  - 是对多个距离度量公式的概括性的表述
  - 对应维度数据相减p次方求和后开p次方
  - 其中p是一个变参数：

    当 p=1 时，就是曼哈顿距离；

    当 p=2 时，就是欧氏距离；

    当 p→∞ 时，就是切比雪夫距离

    根据 p 的不同，闵氏距离可表示某一类种的距离

2.说明特征预处理的方法

- 归一化：

$$X'=\frac{x-min}{max-min}$$

$$X''=X'*(mx-mi)+mi$$

  - 对原始数据进行变换到【mi,mx】(默认为[0,1])之间
  - 受异常值影响，一般不常用
- 标准化：

$$X'=\frac{x-\text{mean}}{\sigma}$$

  - 将原始数据转换为均值为0标准差为1的标准正态分布的数据
  - 不易受异常值影响，常用

3.编写KNN代码实现鸢尾花分类案例

```
# 导入工具包
from sklearn.datasets import load_iris
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn.neighbors import KNeighborsClassifier
from sklearn.metrics import accuracy_score
import seaborn as sns
import matplotlib.pyplot as plt
import pandas as pd

# 加载数据
iris_data = load_iris()
# print(iris_data)
print(iris_data.feature_names)
iris_df = pd.DataFrame(iris_data['data'], columns=iris_data.feature_names)
iris_df['label'] = iris_data.target
sns.lmplot(x='sepal length (cm)', y='sepal width (cm)', data=iris_df,
hue='label')
plt.show()

# 数据集划分
x_train, x_test, y_train, y_text = train_test_split(iris_data.data,
iris_data.target, test_size=0.3, random_state=22)
print(len(iris_data.data))
print(len(x_train))

# 标准化
process = StandardScaler()
x_train = process.fit_transform(x_train)
x_test = process.transform(x_test)

# 模型训练
model = KNeighborsClassifier(n_neighbors=3)
model.fit(x_train, y_train)

x = [[5.1, 3.5, 1.4, 0.2]]
x = process.transform(x)

# 模型预测
y_predict = model.predict(x_test)
print(model.predict_proba(x))

# 模型评估
acc = accuracy_score(y_text, y_predict)
print(acc)
```

4.编写KNN代码实现手写数字识别（特征预处理，交叉验证网格搜索）

```
# 导入工具包
import matplotlib.pyplot as plt
```

```python
import pandas as pd
from sklearn.model_selection import train_test_split, GridSearchCV
from sklearn.neighbors import KNeighborsClassifier
from sklearn.preprocessing import  MinMaxScaler


# 获取数据
data = pd.read_csv('手写数字识别.csv')
x = data.iloc[:, 1:]
y = data.iloc[:, 0]
# 特征归一化【注意】
transform = MinMaxScaler()
x = transform.fit_transform(x)
# x = x / 255.

# 数据集划分
x_train, x_test, y_train, y_test = train_test_split(x, y, test_size=0.2,
stratify=y, random_state=22)
# 模型实例化
model = KNeighborsClassifier(n_neighbors=1)
# 网格搜索交叉验证
model = GridSearchCV(estimator=model, param_grid={'n_neighbors': [3, 5, 7,
9, 10, 11]}, cv=4)
model.fit(x_train, y_train)
print(model.best_estimator_)
# 模型训练

model = model.best_estimator_
model.predict(x_test)
# 模型预测
img = plt.imread('demo.png')
img=img.reshape(1,-1) / 255.
print(model.predict(img))
# 模型评估
print(model.score(x_test, y_test))
```