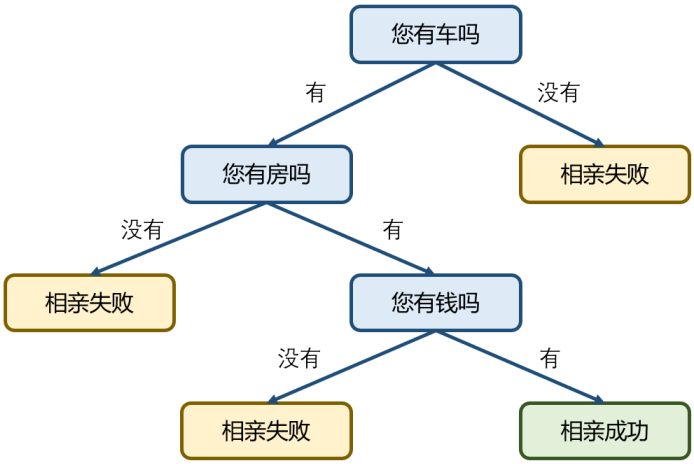
实验四 机器学习方法

一． 决策树方法

1. 简介

决策树是一种常见的分类模型，在金融风控、医疗辅助诊断等诸多行业具有较为广泛的应用。决策树的核心思想是基于树结构对数据进行划分，这种思想是人类处理问题时的本能方法。例如在婚恋市场中，女方通常会先询问男方是否有房产，如果有房产再了解是否有车产，如果有车产再看是否有稳定工作……最后得出是否要深入了解的判断。



2. 优缺点

优点：

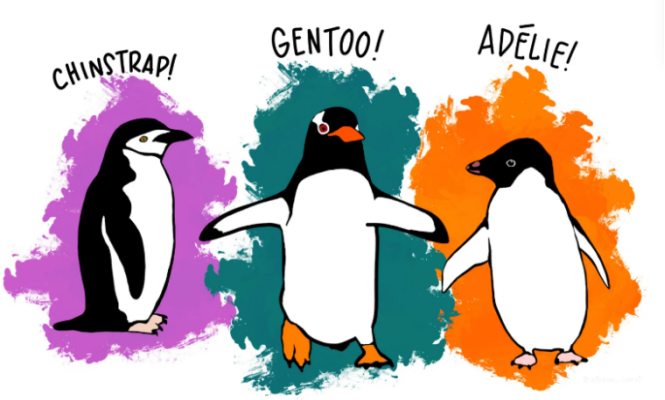
1. 具有很好的解释性，模型可以生成可以理解的规则；
2. 可以发现特征的重要程度；
3. 模型的计算复杂度较低。

缺点：

1. 模型容易过拟合，需要采用减枝技术处理；
2. 预测能力有限，无法达到其他强监督模型效果。
3. 方差高，数据分布的轻微改变很容易造成树结构完全不同。

3. 任务目标：用决策树算法搭建企鹅分类模型

本项目数据来源为帕尔默企鹅数据：由 Kristen Gorman 博士和南极洲 LTER 的帕尔默科考站共同创建，包含 344 只企鹅的数据。



特征	数据类型	说明
species	离散值	标签信息，值为 Adelie|Chinstrap|Gentoo 之一
island	离散值	岛屿，值为 Torgersen|Biscoe|Dream 之一
culmen_length_mm	连续值	喙的长度（mm）
culmen_depth_mm	连续值	喙的高度（mm）
flipper_length_mm	连续值	脚蹼长度（mm）
body_mass_g	连续值	体重（克）
sex	离散值	性别，值为 MALE（雄）| FEMALE（雌） 之一

我们希望通过给出一个企鹅的所有特征，通过构建决策树模型，来预测该企鹅的类型。我们一般通过测试集结果的精度来判断我们效果的好坏。

数据集下载地址：下载链接，其中的 penguins_raw.csv 文件为数据集。

注：

1. **本项目可用任意语言（C，Java，Python）实现，这里采用的是 Python，后面的项目分析部分也是基于 Python 写的。**
2. **需要提交的内容要包含：完整的运行代码 + 可视化数据截图 + 模型准确率（accuracy）截图**

4. 项目分析

要利用 python 进行数据分析，首先要配置好相关的环境，如果想一步到位，推荐大家安装 Anaconda，这是一个用于深度学习和数据分析的库，里面已经配置好所有的环境并下载好了需要的包，安装教程请访问连接：如何安装 anaconda3，

此处不在进行相关赘述。

想进一步将 anaconda3 和 pycharm 绑定的，请访问连接：如何连接 anaconda 部署到 pycharm 中；如果不想配置的，可以直接打开 anaconda3，其中有一个 jupyter notebook，打开之后新建一个文件，也可以直接编写：

第一步：导入包和相关数据

```
##   基础函数库
import numpy as np
import pandas as pd
import graphviz

## 绘图函数库
import matplotlib.pyplot as plt
import seaborn as sns

## 我们利用 Pandas 自带的 read_csv 函数读取并转化为 DataFrame 格式，这里要注意修改一下你自己文件的位置，要不然导入不进来
data = pd.read_csv('./penguins_raw.csv')

## 为了方便我们仅选取四个简单的特征，有兴趣的同学可以研究下其他特征的含义以及使用方法
data = data[['Species','Culmen Length (mm)','Culmen Depth (mm)',
                    'Flipper Length (mm)','Body Mass (g)']]
```

第二步：简单查看一下数据的情况

```
## 利用.info()查看数据的整体信息
data.info()

## 进行简单的数据查看，我们可以利用 .head() 头部.tail()尾部
data.head()
data = data.fillna(-1)
data.tail()

## 其对应的类别标签为'Adelie Penguin', 'Gentoo penguin', 'Chinstrap penguin'三种不同企鹅的类别。
data['Species'].unique()

'''为了方便我们将标签转化为数字
        'Adelie Penguin (Pygoscelis adeliae)'          ------0
        'Gentoo penguin (Pygoscelis papua)'             ------1
        'Chinstrap penguin (Pygoscelis antarctica)     ------2 '''
```

```
def trans(x):
    if x == data['Species'].unique()[0]:
        return 0
    if x == data['Species'].unique()[1]:
        return 1
    if x == data['Species'].unique()[2]:
        return 2

data['Species'] = data['Species'].apply(trans)
```

第三步：利用决策树进行计算

```
## 为了正确评估模型性能，将数据划分为训练集和测试集，并在训练集上训练模型，在测试集上验证模型性能。
from sklearn.model_selection import train_test_split

## 选择其类别为 0 和 1 的样本 （不包括类别为 2 的样本）
data_target_part = data[data['Species'].isin([0,1])][['Species']]
data_features_part = data[data['Species'].isin([0,1])][['Culmen Length (mm)','Culmen Depth (mm)','Flipper Length (mm)','Body Mass (g)']]

## 测试集大小为 20%， 80%/20%分（请补全缺失代码）
**********请补全这句缺失的代码 **********
x_train, x_test, y_train, y_test =
**********

## 从 sklearn 中导入决策树模型
from sklearn.tree import DecisionTreeClassifier
from sklearn import tree


## 定义 决策树模型
clf = DecisionTreeClassifier(criterion='entropy')
# 在训练集上训练决策树模型（请补全缺失代码）

**********请补全这句缺失的代码 **********
**********
```

```
## 可视化
import graphviz
dot_data = tree.export_graphviz(clf, out_file=None)
graph = graphviz.Source(dot_data)
graph.render("penguins")
```

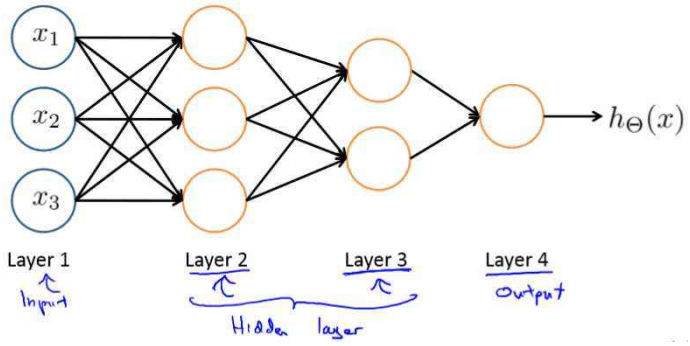
```
## 在训练集和测试集上分布利用训练好的模型进行预测
train_predict = clf.predict(x_train)
test_predict = clf.predict(x_test)
from sklearn import metrics

## 利用 accuracy（准确度）【预测正确的样本数目占总预测样本数目的比例】评估模型效果
print('The accuracy of the Logistic Regression is:',metrics.accuracy_score(y_train,train_predict))
print('The accuracy of the Logistic Regression is:',metrics.accuracy_score(y_test,test_predict))

## 查看混淆矩阵 (预测值和真实值的各类情况统计矩阵)
confusion_matrix_result = metrics.confusion_matrix(test_predict,y_test)
print('The confusion matrix result:\n',confusion_matrix_result)

# 利用热力图对于结果进行可视化
plt.figure(figsize=(8, 6))
sns.heatmap(confusion_matrix_result, annot=True, cmap='Blues')
plt.xlabel('Predicted labels')
plt.ylabel('True labels')
plt.show()
```

二． 神经网络方法

1. 简介

BP（Back Propagation）网络是 1986 年由 Rumelhart 和 McCelland 为首的科学家小组提出，是一种按误差逆传播算法训练的多层前馈网络，是目前应用最广

泛的神经网络模型之一。BP 网络能学习和存贮大量的输入-输出模式映射关系，而无需事前揭示描述这种映射关系的数学方程。它的学习规则是使用最速下降法，通过反向传播来不断调整网络的权值和阈值，使网络的误差平方和最小。BP 神经网络模型拓扑结构包括输入层（input）、隐层(hide layer)和输出层(output layer)。在模拟过程中收集系统所产生的误差，通过误差反传，然后调整权值大小，通过该不断迭代更新，最后使得模型趋于整体最优化（这是一个循环，我们在训练神经网络的时候是要不断的去重复这个过程的）。



2. 优缺点

优点：

1. 非线性映射能力：BP 神经网络实质上实现了一个从输入到输出的映射功能，数学理论证明三层的神经网络就能够以任意精度逼近任何非线性连续函数。这使得其特别适合于求解内部机制复杂的问题，即 BP 神经网络具有较强的非线性映射能力；
2. 自学习和自适应能力：BP 神经网络在训练时，能够通过学习自动提取输入、输出数据间的“合理规则”，并自适应地将学习内容记忆于网络的权值中。即 BP 神经网络具有高度自学习和自适应的能力；
3. 泛化能力：所谓泛化能力是指在设计模式分类器时，即要考虑网络在保证对所需分类对象进行正确分类，还要关心网络在经过训练后，能否对未见过的模式或有噪声污染的模式，进行正确的分类。也即 BP 神经网络具有将学习成果应用于新知识的能力。具有很好的解释性，模型可以生成可以理解的规则。

缺点：

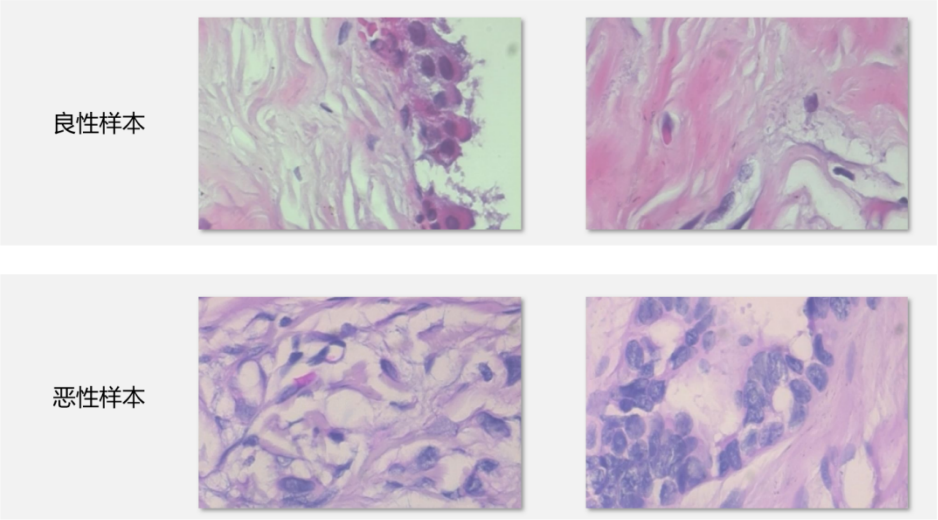
1. 局部极小化问题：从数学角度看，传统的 BP 神经网络为一种局部搜索的优化方法，它要解决的是一个复杂非线性化问题，网络的权值是通过沿局部改善的方向逐渐进行调整的，这样会使算法陷入局部极值，权值收敛到局部极小点，从而导致网络训练失败。加上 BP 神经网络对初始网络权重非常敏感，以不同的权重初始化网络，其往往会收敛于不同的局部极小，这也是每次训练得到不同结果的根本原因。
2. BP 神经网络算法的收敛速度慢：由于 BP 神经网络算法本质上为梯度下降法，它所要优化的目标函数是非常复杂的，因此，必然会出现“锯齿形现象”，这使得 BP 算法低效；又由于优化的目标函数很复杂，它必然会在神经元输出接近 0 或 1 的情况下，出现一些平坦区，在这些区域内，权值误差改变很小，使训练过程几乎停顿；BP 神经网络模型中，为了使网络执行 BP 算法，不能使用传统的一维搜索法求每次迭代的步长，而必须把步长的更新规则预先赋予网络，这种方法也会引起算法低效。以上种种，导致了 BP 神经网络算法收敛速度慢的现象。
3. BP 神经网络结构选择不一：BP 神经网络结构的选择至今尚无一种统一而完整的理论指导，一般只能由经验选定。网络结构选择过大，训练中效率不高，可能出现过拟合现象，造成网络性能低，容错性下降，若选择过小，则又会造成网络可能不收敛。而网络的结构直接影响网络的逼近能力及推广性质。因此，应用中如何选择合适的网络结构是一个重要的问题。

3. 任务目标：用 BP 神经网络搭建乳腺癌预测模型

乳腺癌是全球第二常见的女性癌症。2012 年，它占所有新癌症病例的 12%，占所有女性癌症病例的 25%。当乳腺细胞生长失控时，乳腺癌就开始了。这些细胞通常形成一个肿瘤，通常可以在 x 光片上直接看到或感觉到有一个肿块。如果癌细胞能生长到周围组织或扩散到身体的其他地方，那么这个肿瘤就是恶性的。

我们希望构建一个算法，通过查看活检图像自动识别患者是否患有乳腺癌。算法要保证精确，因为人的生命安全是第一的。

本项目的数据集可以直接从 sklearn 机器学习包中导入，同时也可以自行下载：下载链接



注：

1. **本项目可用任意语言（C，Java，Python）实现，这里采用的是 Python，后面的项目分析部分也是基于 Python 写的。**
2. **需要提交的内容要包含：完整的运行代码 + 可视化数据截图 + 模型准确率（accuracy）截图**

4. 问题分析

第一步：导入需要的相关包和数据库

```
# 导入乳腺癌数据集
from sklearn.datasets import load_breast_cancer
# 导入 BP 模型
from sklearn.neural_network import MLPClassifier
# 导入训练集分割方法
from sklearn.model_selection import train_test_split
# 导入预测指标计算函数和混淆矩阵计算函数
from sklearn.metrics import classification_report, confusion_matrix
# 导入绘图包
import seaborn as sns
import matplotlib
from mpl_toolkits.mplot3d import Axes3D

# 导入乳腺癌数据集
cancer = load_breast_cancer()
```

第二步：查看数据形态，并分割训练集和测试集

```
# 分割数据为训练集和测试集
cancer_data = cancer['data']
print('cancer_data 数据维度为：',cancer_data.shape)
cancer_target = cancer['target']
print('cancer_target 标签维度为：',cancer_target.shape)
cancer_names = cancer['feature_names']
cancer_desc = cancer['DESCR']

#分为训练集与测试集（）
cancer_data_train,cancer_data_test=train_test_split(cancer_data,test_size=0.2,rando
m_state=42)
cancer_target_train,cancer_target_test=train_test_split(cancer_target,test_size=0.2,ra
ndom_state=42)
```

第三步：搭建神经网络

```
# 建立 BP 模型，采用 Adam 优化器，relu 非线性映射函数
----------请补全相应的代码----------
BP =
# 进行模型训练
----------请补全相应的代码----------
```

第四步：对模型进行预测，可视化观测结果

```
# 进行模型预测
predict_train_labels = BP.predict(cancer_data_train)

# 可视化真实数据
fig = plt.figure()
ax = Axes3D(fig, rect=[0, 0, 1, 1], elev=20, azim=20)
ax.scatter(cancer_data_train[:, 0], cancer_data_train[:, 1], cancer_data_train[:, 2],
marker='o', c=cancer_target_train)
plt.title('True Label Map')
plt.show()

# 可视化预测数据
fig = plt.figure()
ax = Axes3D(fig, rect=[0, 0, 1, 1], elev=20, azim=20)
ax.scatter(cancer_data_train[:, 0], cancer_data_train[:, 1], cancer_data_train[:, 2],
marker='o', c=predict_train_labels)
plt.title('Cancer with BP Model')
plt.show()
```

第五步：实验结果

```
# 显示预测分数
print("预测准确率: {:.4f}".format(BP.score(cancer_data_test, cancer_target_test)))
predict_test_labels = BP.predict(cancer_data_test)

# 进行预测结果指标统计 统计每一类别的预测准确率、召回率、F1 分数
print(classification_report(cancer_target_test, predict_test_labels))
```

5. 实验提交要求

将实验 1-2 问题分析，实验运行情况按照实验报告样例编写并且保存到以”学号+姓名.doc/docx”形式的文件中，例如”202022408001+张三.doc/docx”；于 11 月 5 日之前交给袁雪婵同学即可。实验报告样例如下：

封面（提交时删除）

《人工智能》实验报告

实验名称__知识表征方法_（每次实验自行更改）

学　　号_____

姓　　名_____

日期

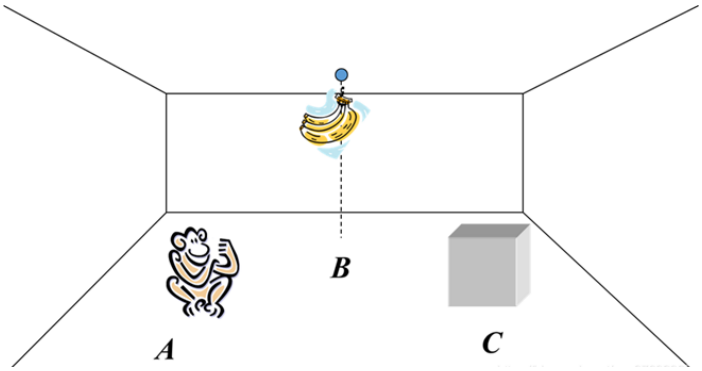
内页：（提交时删除）

实验一 知识表示方法

一、猴子摘香蕉问题

1.实验内容：

房内有一个猴子，一个箱子，天花板上挂了一串香蕉，其位置如图 1 所示，猴子为了拿到香蕉，它必须把箱子搬到香蕉下面，然后再爬到箱子上。请定义必要的谓词，列出问题的初始化状态（即下图所示状态），目标状态（猴子拿到了香蕉，站在箱子上，箱子位于位置 b）。



2.实验思路：

3.程序清单：需加适当注释

4.运行结果说明：

二、……

…

文字用小 4 号或 4 号；程序和注释用 5 号，程序不能截图，需可再运行。