Part 1

遗传算法和机器学习有什么关系?

遗传算法（Genetic Algorithms, GAs）和机器学习都是计算机科学和人工智能的重要子领域，它们之间存在
联和交叉。以下是遗传算法和机器学习之间的关系：

1. **优化方法**：遗传算法是一种启发式搜索和全局优化方法，受自然选择过程的启发。在机器学习中，尤其
   习的背景下，我们经常需要优化复杂的损失函数。遗传算法可以作为这些优化问题的解决方法。
2. **特征选择**：在机器学习中，特征选择是一个关键步骤，它决定了模型使用哪些特征进行训练。遗传算法
   于自动选择最优的特征组合，从而提高模型的表现。
3. **参数调整**：在机器学习和深度学习中，参数调整是一个重要但计算密集的过程。遗传算法可以被用作一
   搜索最佳的参数组合。
4. **神经网络的结构搜索**：在深度学习中，确定一个神经网络的最佳结构是一个开放的问题。遗传算法可以
   合适的网络结构，如层数、每层的神经元数量等。
5. **机器学习中的组合问题**：某些机器学习问题，如旅行商问题、时间表问题等，是组合优化问题。遗传算
   用于这类问题的解决。
6. **混合方法**：遗传算法有时会与其他机器学习方法结合，形成所谓的“混合方法”。例如，神经网络的权重

以通过遗传算法进行，然后再使用传统的优化算法（如梯度下降）进行微调。

7. **强化学习**：在强化学习中，遗传算法可以被用于策略搜索，特别是在策略空间很大或非常复杂的情况下。

总的来说，遗传算法为机器学习提供了一种全局搜索和优化的策略，可以与其他机器学习技术相结合，从而在题上获得更好的性能。然而，与其他优化技术一样，遗传算法不总是最佳选择，其适用性取决于特定的应用和

课程信息

Week 3 quiz 30 min

内容

Problems can be classified in different ways:

- Black box model
- Search problems
- Optimization vs constraint satisfaction
- NP problems

Black box model

1. Optimisation: Model and desired output is known, task is to **find inputs**
2. Modelling: We have corresponding sets of inputs & outputs and **seek model** that delivers correct ou every known input
3. Simulation: We have a given model and **wish to know the outputs** that arise under different input co

Optimisation/modelling problems search through huge space of possibilities
Search space: collection of all objects of interest including the desired solution

search problems, which define search spaces
problem-solvers, which tell how to move through search spaces

Objective function: a way of assigning a value to a possible solution that reflects its quality on scale
Constraint: binary evaluation telling whether a given requirement holds or not

Optimisation example 1: university timetabling

“Black box”模型在优化背景下的解释往往与其在机器学习中的含义略有不同。在优化背景下，一个“black bo模型是指我们可以评估其输出（或“响应”）但不知道或不使用其内部工作原理的函数。

以大学课程排程为例，这是一个经典的优化问题，有以下特点：

1. **极大的搜索空间**：考虑到所有可能的课程、教师、教室、时间段的组合，排程的可能性是巨大的。
2. **课表必须是好的**：这意味着生成的时间表必须满足各种质量标准。
3. **“好”是由多个竞争标准定义的**：例如，可能希望最小化教师的空闲时间，最大化教室的使用率，或确保学没有时间冲突。
4. **课表必须是可行的**：这意味着不能为一个教师、一个教室排定重叠的课程，每个课程只能安排一次，等等

5. **绝大多数的搜索空间是不可行的**：由于有许多约束条件，所以绝大多数的可能排程都是不可行的。

在这种背景下，将排程问题视为一个“black box”优化问题意味着我们可以不必关心所有可能的排程细节或约
形式，而是使用一种方法（例如遗传算法）来搜索这个大规模的空间，寻找一个满足所有条件的最佳解决方
方法的好处是它的通用性和灵活性，因为它可以应用于各种排程问题，而无需为每一个问题重新设计算法。

Optimisation example 2: satellite structure

在讨论“Black box”模型的背景下，考虑一个与卫星结构优化相关的例子。

优化实例2：卫星结构

1. **为NASA优化的卫星设计以最大化振动隔离**：在这种场景下，目的是设计一个卫星结构，它可以最大化地
离外部的振动，这对于确保卫星内部的仪器和组件的稳定性和持久性至关重要。

2. **进化：设计结构**：这里的“进化”意味着我们不是静态地创建一个卫星设计，而是使用一种方法（例如遗传
其他进化算法）来迭代地改进和修改设计。每个新的设计或“个体”都是基于前一代的设计进行修改和变异

3. **适应度：振动抗性**：在这个优化问题中，每个设计的“适应度”或好坏是基于它的振动抗性来评估的。一个
高振动抗性的设计将被视为更“适应”其环境，并且更有可能在进化过程中被保留和繁殖。

4. **进化“创造性”**：这是指使用进化算法可以产生一些出人意料、独特且有效的设计。与传统的设计方法相比
算法往往能够在大型搜索空间中找到新的、有创意的解决方案。

在这个背景下，当我们把这个卫星结构优化问题看作一个“Black box”问题时，我们的主要关注点是输入（设
和输出（振动抗性）之间的关系，而不是内部的具体机械或物理原理。我们使用进化算法来探索这个设计空间
最佳的卫星结构，而不需要深入了解每个设计的具体工作机制。这种方法允许我们在广阔的设计空间中找到高
创意的解决方案。

Optimisation example 3: 8 queens problem

优化实例3：8皇后问题

8皇后问题是一个经典的计算问题，涉及到如何在一个8x8的棋盘上放置8个皇后，使得没有任何两个皇后在同
同一列或同一对角线上。以下是这个问题的概述以及其与“Black box”模型的关系：

1. **问题描述**：在一个8x8的棋盘上，放置8个皇后，确保它们彼此之间不会互相攻击。皇后是棋盘上唯一可
一行、列和对角线上的所有其他方格的棋子。

2. **冲突定义**：如果两个皇后处于同一行、列或对角线上，那么它们处于冲突状态。

3. **可扩展性**：这个问题可以推广到n皇后问题，即在一个nxn的棋盘上放置n个皇后，遵循相同的规则。

在一个“Black box”优化背景下，我们可以将这个问题看作是在搜索一个解的空间中，寻找一个没有任何冲突
置方法。具体来说：

- **输入**：8个皇后在棋盘上的位置。
- **输出**：冲突的数量（我们希望这个数字为0）。

我们可能不关心搜索空间的内部结构或具体的搜索方法。我们只关心输入（皇后的位置）和输出（冲突数）之
系。

为了解决这个问题，可以使用各种算法，包括回溯搜索、模拟退火、遗传算法等。在使用像遗传算法这样的“

box"优化技术时，我们会定义一个适应度函数（例如，冲突的数量）来评估每个解的质量，并通过迭代的方
解，直到找到一个没有冲突的解为止。

NP problems

- Problem size: dimensionality of the problem at hand and number of different values for the problem
- Running-time: number of operations the algorithm takes to terminate
  - Worst-case as a function of problem size
  - Polynomial, super-polynomial, exponential
- Problem reduction: transforming current problem into another via mapping

Class

- Class P: 这类问题可以在多项式时间内由一个确定性算法解决
- Class NP: 这类问题有一个特点，那就是给定一个解决方案，我们可以在多项式时间内验证该解决方案是
- Class NP-complete: 如果一个问题属于NP类，且所有NP类的问题都可以在多项式时间内约简到这个问
  这个问题就是NP完全的
- Class NP-hard: 如果一个问题至少与NP完全问题一样难，那么它就是NP-hard。注意，这不需要解决方
  式时间内被验证

General scheme of EAs

这一页PPT描述了进化算法（Evolutionary Algorithm, EA）的基本框架。以下是对此框架的详细解释：

进化算法的一般框架：

1. **初始化 (Initialization)**：
   - 在算法的开始，会随机生成一个解的初始种群。这些初始的解也被称为"个体"。
2. **选择父本 (Parent selection)**：
   - 从当前种群中选择一些个体作为"父本"进行交叉和突变。选择的过程通常基于个体的适应度（也就是
     "健康"或"质量"）。
3. **重组 (Recombination 或 Crossover)**：
   - 这是模拟生物的交配过程。从父本中选择的两个个体会交换它们的部分信息来创建新的个体或"后代
4. **突变 (Mutation)**：
   - 随机修改个体的某些部分以增加种群的多样性。突变的概率通常设定得很低，因为过多的突变可能
     的解。
5. **生存者选择 (Survivor selection)**：
   - 在这个阶段，从当前的个体和新产生的后代中选择哪些个体将进入下一代。这通常基于适应度，使
     的解更有可能被选中。
6. **终止 (Termination)**：
   - 这决定了算法何时停止。这可以是固定数量的代数、达到某个适应度阈值、没有明显的进步或其他
     条件。

整个进化算法的循环通常如下：**初始化 -> 选择父本 -> 重组 -> 突变 -> 生存者选择 -> 判断是否终止**。如果
终止条件，算法返回并重新选择父本，然后继续该循环。

总之，这个PPT页面提供了一个高级的、结构化的进化算法的概览，描述了从初始种群的创建到算法的终止的
段。

ML Part 1. Introduction to Machine Learning

内容涵盖以下主题：

1. **机器学习简介**：包括对机器学习的定义和与人类学习的比较。
2. **人类学习类型**：探讨人类如何通过专家指导、从专家知识中指导和自我学习的过程。
3. **机器学习的类型**：介绍了三种主要的机器学习类型，包括监督学习、非监督学习和强化学习。
4. **监督学习和非监督学习**：详细介绍了这两种学习类型，包括它们的特点、应用和相关问题。
5. **机器学习应用**：讲述了机器学习在银行、保险和医疗等多个领域的实际应用。
6. **机器学习的工具和语言**：讨论了在机器学习中广泛使用的一些编程语言和工具，如Python、R和MATLAI
7. **机器学习的问题**：探讨了机器学习应用中可能遇到的问题，包括隐私问题和伦理考虑。

ML Part 2. Data Preparation for Machine Learning

《机器学习中的数据准备》PPT详细介绍了机器学习中数据准备的各个方面。以下是详细内容的概要：

1. 引言

- 介绍了人类学习与机器学习类型（监督学习、非监督学习、强化学习）的关系。
- 举例说明了机器学习在不同领域（如银行、保险、医疗）的应用。

2. 机器学习活动

- 描述了机器学习项目的初始步骤，强调了数据的理解和预处理的重要性。
- 介绍了数据类型的理解、数据探索、识别数据问题、数据预处理等步骤。

3. 数据类型

- 讲述了机器学习中的基本数据类型：定性数据（名义数据和顺序数据）和定量数据（区间数据和比率数

4. 探索数据结构

- 讨论了如何探索数据集中的数值型和分类型数据。
- 强调了数据字典在理解数据结构中的重要性。

5. 数据质量和修正

- 探讨了数据质量对机器学习成功的重要性，以及如何处理数据质量问题。

6. 数据预处理

- 介绍了数据预处理技术，如特征缩放、归一化和标准化。

7. 数据维度缩减和特征选择

- 讨论了降低数据维度和选择合适特征的方法和重要性。

8. 总结

- 概述了数据集的基本信息、数据类型、数据探索、数据质量和预处理等关键点。

这份PPT为机器学习初学者提供了一个全面的数据处理指南，确保他们能有效地开始他们的机器学习项目。

ML Part 3. Bayesian Decision Theory

这份PPT是关于机器学习的讲座笔记，主要集中在贝叶斯决策理论（Bayesian Decision Theory）及其应用。PPT的主要内容概述：

1. **贝叶斯定理（Bayes Theorem）**：
    - 介绍了贝叶斯定理的基础，并以海鲈鱼和三文鱼分类的例子来阐释如何应用贝叶斯定理进行决策。
2. **贝叶斯决策规则（Bayes Decision Rule）**：
    - 解释了如何基于后验概率来做决策。
    - 探讨了如何在有多个特征和多个类别的情况下应用贝叶斯决策规则。
3. **正态（高斯）密度函数（Normal/Gaussian Density Function）**：
    - 讲解了单变量和多变量正态分布。
    - 详细讨论了如何估计参数，包括使用极大似然估计法。
4. **高斯混合模型（Gaussian Mixture Models, GMM）**：
    - 介绍了GMM的基础概念，如何使用GMM来处理多峰分布的数据。
    - 解释了参数估计问题，特别是使用期望最大化（Expectation-Maximization, EM）算法进行参数估计
5. **朴素贝叶斯（Naïve Bayes）**：
    - 详细讨论了朴素贝叶斯分类器的概念和类型，包括高斯型、伯努利型和多项式型朴素贝叶斯。
    - 举例说明了如何应用朴素贝叶斯进行分类，包括文本分类。

整体而言，这份PPT深入探讨了贝叶斯决策理论及其在模式识别和机器学习中的应用。通过实例和详细解释，更好地理解这些概念和技术。

ML Part 4. Linear Discriminant Analysis (LDA)

这个PPT主要介绍了线性判别分析（Linear Discriminant Analysis，简称LDA），这是一种常用于模式识别和方法，特别是在机器学习和数据挖掘领域。以下是对PPT内容的详细介绍：

1. **线性判别函数的设计**：PPT从一个两类分类问题开始，介绍如何设计线性判别函数（即一条线或一个超平分隔两个类别的样本。这涉及权重向量和偏置或阈值权重的定义。
2. **决策规则**：介绍了基于线性判别函数的决策规则，包括如何决定属于哪个类别以及如何确定决策边界。
3. **几何解释**：提供了线性判别分析的几何解释，解释了权重向量如何与决策超平面正交，以及如何将特征空为两个半空间。

4. **判别函数的距离度量**：介绍了判别函数如何度量样本点到决策超平面的距离，并对这个距离的正负进行了
5. **投影和类间分离**：讨论了如何通过投影将样本分离到超平面上，并强调了使得投影点尽可能分离的重要性
6. **Fisher线性判别**：介绍了Fisher线性判别的概念，包括如何最大化类间距离与类内距离的比率。
7. **散布矩阵和特征值问题**：解释了计算散布矩阵的方法，并将问题转化为特征值问题。
8. **多类判别分析**：扩展了LDA到多类情况，介绍了如何从d维空间投影到d-1维空间。
9. **案例分析**：最后，通过实际的数据集案例来展示线性判别分析的应用。

ML Part 5. Support Vector Machines

这个PPT详细介绍了支持向量机（Support Vector Machines，SVM）的概念和原理。以下是主要内容的概要

1. **动机（Motivation）**：开始时，文档介绍了线性分类的目标，即找到一个超平面来分离两个类别的数据。
2. **问题表述（Problem Formulation）**：这部分详细说明了超平面的数学方程，并讨论了将数据点与超平面大化的概念。
3. **支持向量机的目标（Goal of SVM）**：这里强调了SVM的目标是找到一个特定的超平面，该超平面可以最间的分离边界（margin of separation）。
4. **判别函数（Discriminant Function）**：讨论了如何使用代数方法测量数据点与超平面的距离。
5. **支持向量（Support Vectors）**：介绍了支持向量的概念，这些是决定分类面的关键数据点。
6. **最优超平面（Optimal Hyperplane）**：讨论了如何确定最优超平面，以及如何通过最大化分类边界来最小向量的欧几里得范数。
7. **原始问题和对偶问题（Primal and Dual Problem）**：分析了SVM中的优化问题，包括拉格朗日乘数法在应用。
8. **线性可分与非线性可分模式（Linearly Separable and Non-separable Patterns）**：通过例子展示了在线和非线性可分情况下如何构建分类器。
9. **核支持向量机（Kernel Support Vector Machines）**：简要介绍了核SVM的概念，这是处理线性不可分数法。

ML Part 6. Classification Trees

这份讲义《ML-LectureNote6》详细讲述了机器学习领域中分类树（Classification Trees）的相关知识。以下容的详细解释：

1. **分类树简介（第2-3页）：**
    - 讲义从将分类树与其他分类方法（如线性判别分析LDA和支持向量机SVM）进行比较开始，强调在没数据（没有定量价值的命名或标记数据）的情况下，分类树的使用优势。传统基于距离度量的分类器情况下可能表现不佳。

2. **分类树基础（第2-4页）：**
    - 描述了分类树作为类似流程图的树状结构，每个节点代表基于数据特征或属性的问题。决策过程遵
      是/否或真/假问题，最终导致带有类标签的叶节点。
3. **分类树的优点（第3页）：**
    - 讨论了分类树的可解释性，说明了它们如何允许轻松理解和追踪测试数据的决策过程。
4. **构建分类树（第4-8页）：**
    - 这一部分涵盖了构建分类树的过程，包括确定每个节点的分裂数量、选择节点属性、定义叶节点以
      以简化。还涉及了如何处理不纯净的叶节点。
5. **节点不纯度度量（第10-15页）：**
    - 介绍了不同的不纯度度量方法，如熵不纯度、方差不纯度和基尼不纯度。讲义解释了这些度量如何
      每个节点的属性测试，并实现数据纯净度。
6. **节点选择和分裂准则（第16-17页，19-21页）：**
    - 讨论了如何为节点选择属性以及节点分裂的准则，包括二分法和多重分裂。讲义探讨了不纯度减少
      率的概念，以优化树结构。
7. **停止准则和修剪（第33-36页）：**
    - 这部分讲述了何时停止树的生长以避免过拟合，以及修剪如何有助于在不显著增加不纯度的情况下
      构。
8. **标记叶节点（第37页）：**
    - 描述了树构建的最后一步，即根据大多数样本对每个叶节点进行类别标记。
9. **不同类型的分类树和随机森林（第54-63页）：**
    - 讲义以讨论不同类型的分类树（如CART、ID3、C4.5）和随机森林的概念结束。详细阐述了随机森
      动机及其特点，如多样性、对维数灾难的免疫、可并行化和稳定性。

ML Part 7. Performance Evaluation for Classifiers

总体介绍

这个PPT是关于机器学习领域内分类器的性能评估。下面是详细的介绍：

1. **概述与评估程序**：首先介绍了在监督学习中如何训练分类模型，并强调了模型评估的重要性。随后，详
   几种评估程序，包括：
    - **保留法（Holdout method）**：将数据集分为训练集和测试集，用来评估分类器的性能。
    - **重复保留法（Repeated holdout method）**：是保留法的一种变体，通过多次随机分割数据来提高
      定性。
    - **K折交叉验证（K-fold cross-validation）**：将数据集分为K个互不重叠的部分，每部分轮流作为测

余作为训练集。
    - **留一法交叉验证（Leave-one-out cross-validation, LOOCV）**：每次留下一个样本作为测试集，其
    训练集，适用于样本量较小的情况。
    - **重复的K折交叉验证（Repeated k-fold cross-validation）**：为了降低评估的变异性，多次进行K折
    并计算平均性能。

2. **性能评估指标**：接着介绍了多种用于评估分类器性能的指标，包括：
    - **准确率（Accuracy）**：正确分类的样本比例。
    - **错误率（Error rate）**：错误分类的样本比例。
    - **灵敏度（Sensitivity）和特异性（Specificity）**：分别衡量正类和负类的正确分类比例。
    - **精确度（Precision）和召回率（Recall）**：分别衡量正类预测的准确性和完整性。
    - **F分数（F-score）**：结合了精确度和召回率，是两者的调和平均。
    - **接收者操作特征曲线（ROC curve）和曲线下面积（AUC）**：用于评估分类模型在不同阈值下的性
    是在不平衡数据集上。

7.1 Evaluation Procedure

7.1.1 Holdout method

保留法的核心思想是将数据集分成两个互斥的部分：训练集和测试集。这两部分数据在模型训练和评估过程中
演不同的角色：

1. **训练集（Training Set）**：用来训练模型。在这个数据集上，分类器学习识别各种模式和特征。
2. **测试集（Test Set）**：用来评估模型。这部分数据在训练阶段对模型来说是不可见的，用来测试模型在未
上的表现。

操作步骤如下：

- **数据分割**：首先，数据集被随机分割成两个部分。通常，大约70%的数据用于训练集，剩下的30%用作
但这个比例可以根据具体情况调整。
- **模型训练**：然后，使用训练集来训练分类器。在这一阶段，模型尝试从数据中学习和提取特征，建立分类
- **模型评估**：训练完成后，使用测试集来评估模型的性能。模型的预测结果将与测试集中的实际结果进行比
评估模型的准确性、泛化能力等。

保留法的优点是简单易行，计算成本相对较低。但它也有缺点，比如结果可能受到数据分割方式的影响，有时
法准确反映模型的真实性能。此外，在数据量较小的情况下，可能会导致模型的训练不充分或评估结果的偏差
此，在实际应用中，需要谨慎选择数据分割比例，并且可能需要结合其他评估方法来获得更全面的性能评估。

stratified random sampling

7.1.2 Repeated holdout method

在机器学习中的7.1.2节中介绍了“重复保留法（Repeated Holdout Method）”，这是对基本保留法的一个扩展
是具体介绍：

重复保留法的核心思想与基本保留法相同，即将数据集分成两个部分：训练集和测试集。然而，与基本保留法
是，重复保留法会多次随机分割数据集，并重复训练和测试过程，以提高评估结果的稳定性和可靠性。

具体步骤如下：

1. **多次分割**：将数据集多次随机分割成训练集和测试集。每次分割都可能产生不同的训练集和测试集组合。
2. **重复训练和测试**：对于每次分割产生的训练集和测试集，分别训练模型并在测试集上评估其性能。这意味
   会在不同的数据子集上多次训练和测试。
3. **平均性能评估**：计算所有重复实验中的性能指标（如准确率、召回率等）的平均值，以获得对模型性能的
   估计。

重复保留法的优点包括：

- **减少偏差**：通过多次随机分割和测试，可以减少由于特定数据分割导致的偏差。
- **提高评估的可靠性**：多次重复实验可以提供更稳定的性能评估结果。

然而，这种方法的缺点是计算成本更高，因为需要多次训练和测试模型。此外，尽管重复保留法可以提供更稳
果，但它仍然受限于原始数据集的大小和质量。

在实际应用中，选择重复次数时需要考虑到计算资源和时间成本。通常，重复次数越多，评估结果越稳定，但
算成本也越高。因此，需要在准确性和计算成本之间找到一个平衡点。

7.1.3 K-fold cross-validation method

K折交叉验证的主要目的是通过重复使用不同的训练集和测试集组合来评估模型的性能，以提高评估的准确性
性。这种方法特别适用于数据量不足时的模型评估。

具体步骤如下：

1. **数据分割**：首先，将整个数据集分割成K个大小相等（或近似相等）的子集。K通常是一个用户定义的数
   5或10。
2. **重复训练和测试**：对于每一个子集，将其作为测试集，而将剩余的K-1个子集合并作为训练集。这样，每
   会轮流作为测试集一次。
3. **性能评估**：对每一次的训练和测试，都会得到一个性能评估指标（如准确率）。最后，计算这K次测试的
   的平均值，作为模型整体性能的估计。

K折交叉验证的优点包括：

- **减少偏差**：每个数据点都会被用作测试集一次，从而减少了评估结果对特定数据分割方式的依赖。
- **提高评估稳定性**：多次训练和测试提供了更全面的性能评估。

但是，K折交叉验证也有其缺点：

- **计算成本高**：需要进行K次训练和测试，尤其是当K值较大或模型复杂时，计算成本会显著增加。
- **数据分割的影响**：如果数据集本身有偏差或不平衡，可能会影响评估结果的准确性。

在实际应用中，选择K的值通常需要在偏差减少和计算成本之间做出权衡。K值较小可能导致评估结果的偏差
K值较大则会增加计算成本。常用的K值为5或10，但最终选择应根据具体情况和资源限制来定。

7.1.4 Leave-one-out cross-validation

留一法交叉验证是在数据集较小时常用的一种模型评估方法。它的核心思想是对每个数据点进行单独的测试，次测试都是在几乎全部的数据上训练得到的模型。

具体操作步骤如下：

1. **数据处理**：如果数据集共有N个数据点，那么在留一法交叉验证中，每次会留下一个数据点作为测试集，N-1个数据点作为训练集。
2. **重复训练和测试**：对于数据集中的每一个数据点，都重复这个过程。也就是说，每个数据点都会轮流作为测试集，而其余的N-1个点作为训练集。
3. **性能评估**：对于每一次的训练和测试，都会得到一个性能评估指标（如准确率）。最后，计算所有这些性能指标的平均值，作为模型整体性能的估计。

留一法交叉验证的优点包括：

- **高准确性**：因为每次测试都是在几乎全部的数据上训练的模型，所以可以得到相对较高的评估准确性。
- **减少偏差**：每个数据点都有机会作为测试集，可以减少评估结果对特定数据分割方式的依赖。

然而，这种方法也有明显的缺点：

- **计算成本极高**：对于较大的数据集，需要进行N次训练和测试，计算量巨大。
- **可能过拟合**：在某些情况下，因为每次都使用几乎所有的数据来训练模型，可能导致模型对特定数据点过拟合。

留一法交叉验证通常适用于数据量较小的情况，因为在大数据集上，它的计算成本可能过于高昂。同时，它可相对较准确的性能评估，特别是在数据点之间差异性较大时。

7.1.5 Repeated k-fold cross-validation

在机器学习领域的7.1.5节中，介绍了“重复的K折交叉验证（Repeated K-fold Cross-Validation）”，这是一种折交叉验证和重复随机抽样的方法。以下是其具体介绍：

重复的K折交叉验证旨在通过多次执行K折交叉验证来提高模型评估的稳定性和可靠性。这种方法特别适合于对数据分割特别敏感的模型。

具体操作步骤如下：

1. **多次执行K折交叉验证**：首先确定K的值（常见的值如5或10），然后不止一次地执行K折交叉验证。每一包含将数据随机分为K个子集，并进行K次的训练和测试。
2. **随机分割数据集**：在每次执行K折交叉验证前，数据集会被重新随机分割。这意味着每次的K折交叉验证同的数据分割上进行的。
3. **计算平均性能指标**：对于每次的K折交叉验证，都会得到一组性能评估指标（如准确率）。然后对所有重交叉验证的结果求平均，以获得最终的性能评估。

重复的K折交叉验证的优点包括：

- **提高评估的稳定性**：由于多次重复执行，结果的稳定性和可靠性得到提高。

- **减少评估的偏差**：多次随机分割数据可以减少因特定数据分割导致的偏差。

但是，这种方法的缺点也很明显：

- **计算成本高**：需要执行多次K折交叉验证，每次都包含K次的训练和测试，因此计算成本相当高。
- **时间成本大**：特别是在数据量大或模型复杂的情况下，所需时间可能会很长。

在实际应用中，选择重复次数和K的值需要根据数据集的大小、模型的复杂性以及可用的计算资源来决定。通下，重复次数可以设置为5、10等，但需要在提高评估稳定性和减少计算成本之间找到平衡。重复的K折交叉于那些对数据分割方式敏感或者需要更稳定评估结果的模型评估场景。

7.2 Performance Evaluation Metrics

在性能评估指标中，TP（True Positive）、FP（False Positive）、TN（True Negative）和FN（False Nega基础概念，用于表示分类器在不同情况下的预测结果。以下是使用这些术语表示的各个指标的公式：

1. **准确率（Accuracy）**：
   $\text{Accuracy} = \frac{TP+TN}{TP+FP+TN+FN}$ 表示所有分类正确的样本占总样本的比例。

2. **错误率（Error Rate）**：
   $\text{Error Rate} = \frac{FP+FN}{TP+FP+TN+FN} = 1 - \text{Accuracy}$ 表示所有分类错误的样本占总样本的比例。

3. **灵敏度（Sensitivity）或召回率（Recall）**：
   $\text{Sensitivity (Recall)} = \frac{TP}{TP+FN}$ 表示所有实际为正类的样本中被正确识别为正类的比例。

4. **特异性（Specificity）**：
   $\text{Specificity} = \frac{TN}{TN+FP}$ 表示所有实际为负类的样本中被正确识别为负类的比例。

5. **精确度（Precision）**：
   $\text{Precision} = \frac{TP}{TP+FP}$ 表示所有预测为正类的样本中实际为正类的比例。

6. **F分数（F-score）**：
   - 特别地，F1分数是精确度和召回率的调和平均：$\text{F1 Score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$ 表示精确度和召衡点。

这些指标是评估分类器性能的常用工具，能够提供关于分类器在不同方面的性能信息。选择哪些指标进行评价具体的应用场景和需求。例如，在疾病筛查中，可能更重视灵敏度（减少漏诊），而在垃圾邮件检测中，可能精确度（减少误判）。

7.2.7 Receiver operating characteristic (ROC) curve

在机器学习的性能评估指标中，7.2.7节介绍的"接收者操作特征曲线（Receiver Operating Characteristic, R Curve）"是一种用于评估分类器性能的重要工具。以下是对ROC曲线的具体解释：

定义和意义

- **ROC曲线**是一种图形化的评估方法，用于展示在不同阈值下分类器性能的变化。
- 它通过绘制真阳性率（True Positive Rate, TPR）和假阳性率（False Positive Rate, FPR）来展示分类器能的分类阈值上的性能。

计算方法

- **真阳性率（TPR）**，也称为灵敏度或召回率，定义为：$\mathrm{TPR} = \frac{TP}{TP+FN}$
- **假阳性率（FPR）**，定义为：$\mathrm{FPR} = \frac{FP}{FP+TN}$

曲线绘制

- 在ROC曲线上，每一个点代表了一个特定阈值下的（FPR, TPR）对。
- 横轴表示假阳性率（FPR），纵轴表示真阳性率（TPR）。
- 曲线开始于(0, 0)点（表示没有任何阳性被分类器识别），结束于(1, 1)点（表示所有阳性都被分类器识别

评估标准

- ROC曲线越靠近左上角，表示分类器的性能越好。
- 一个完美的分类器的ROC曲线会经过左上角（即FPR为0，TPR为1），而一个随机猜测的分类器则会产生
  对角线（从左下角到右上角）的ROC曲线。

曲线下面积（AUC）

- **曲线下面积（Area Under the Curve, AUC）**是评估分类器性能的另一个重要指标。
- AUC值越大，表明分类器的性能越好。
- AUC为1表示完美分类器，AUC为0.5表示随机猜测。

应用场景

- ROC曲线特别适用于评估不平衡数据集（即正负样本数量差异大）的分类器性能。
- 它提供了一种评估分类器在处理易混淆的样本时性能的方式。

总之，ROC曲线是一种强大的工具，可以帮助研究者和实践者在不同的分类阈值下理解和比较分类器的性能。
析ROC曲线，可以选择最适合特定应用的分类阈值。

ML Part 8. Regression

8.1 Simple and multiple linear regression models

1. **简单线性回归**:
   - 简单线性回归建立在统计学中拟合一条直线的概念上。例如，使用工作经验（自变量）来预测工资
     量）。
   - 此模型使用单个预测变量（自变量）和线性函数来预测目标变量（因变量），表示为 $Y = \beta_0 + \beta$
     其中，Y 是目标（因变量），X 是预测（自变量），β_0 是截距，β_1 是斜率，而 ϵ 是残差（误差）
2. **多元线性回归**:
   - 在多元线性回归中，使用多于一个的独立变量来预测因变量。
   - 这个模型的一般形式为 $Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \ldots + \beta_n X_n + \epsilon$。其中，$X_1, X_2, \ldots, X_n$ 是

量，$\beta_0, \beta_1, \ldots, \beta_n$ 是系数或参数。

这一部分提供了对简单线性回归和多元线性回归基本概念的详细解释，包括它们的数学表示和用途。这对于理
使用这些模型来分析和预测数据中的线性关系至关重要。

8.2 Ordinary least square estimation

1. **最小二乘法的基本概念**:
   - 假设有 n 个训练数据对（标记数据），形式为 (x_i, y_i)。普通最小二乘法旨在找到一组系数，使得这
     能最好地拟合这些数据点。
2. **线性回归模型**:
   - 线性回归模型可以表示为 $y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \ldots + \beta_n x_n + \epsilon$，其中 β 是待估计的系数，x
     量，ϵ 是误差项。
3. **损失函数**:
   - 损失函数定义为 $L(\beta) = \sum_{i=1}^{n} (y_i - \hat{y}_i)^2$，其中 $\hat{y}_i$ 是使用模型预测的 y 值。最小二乘法的目标是
     β，使得损失函数 $L(\beta)$ 最小。
4. **最小化损失函数**:
   - 损失函数 $L(\beta)$ 关于系数 β 的最小化是通过求解 $\frac{\partial L(\beta)}{\partial \beta} = 0$ 来实现的，这将导致一个最优系数解。
5. **普通最小二乘估计的结果**:
   - 最终，普通最小二乘法给出了系数的估计值，这些估计值在统计上被认为可以最佳地代表数据中的线
     系。
6. **例子与应用**:
   - PPT中还展示了一个实际数据集的例子来说明如何应用普通最小二乘法来估计线性回归模型的参数。

普通最小二乘估计是线性回归分析中的一个基本和重要的概念，它提供了一种简单而有效的方法来估计线性关
系数。通过最小化实际观测值和预测值之间的差的平方和，可以得到对数据最佳拟合的线性模型。

8.3 Ridge regression

岭回归的背景和需求：

1. **普通最小二乘估计的局限性**:
   - 在有很多预测变量的情况下，普通最小二乘估计可能存在问题，例如参数估计可能不唯一，或者模型
     适合测试数据。
   - 当设计矩阵（X）是奇异的或接近奇异的，普通最小二乘估计会遇到计算问题。
2. **过拟合问题**:
   - 普通最小二乘估计可能过度拟合训练数据，导致模型泛化能力差。

岭回归的核心思想：

1. **引入正则化项**:
   - 岭回归通过在损失函数中添加一个正则化项（惩罚项）来解决这些问题。这个正则化项是参数的L2
     方和）。
2. **损失函数的变化**:
   - 岭回归的损失函数变为 $L(\beta) = \sum_{i=1}^{n}(y_i - \hat{y}_i)^2 + \lambda \sum_{j=1}^{p} \beta_j^2$，其中 λ 是正则化系数，控制惩罚
     度。

岭回归的效果：

1. **参数收缩和稳定性**:
   - 通过正则化项，岭回归减少了模型参数的复杂度，防止了过拟合，提高了模型的泛化能力。
   - 参数的值会向零收缩，但不会完全等于零（与Lasso回归不同）。
2. **选择正则化系数 λ**:
   - 选择合适的 λ 对模型性能至关重要。λ 值的选择通常通过交叉验证来确定。
3. **适用场景**:
   - 岭回归特别适合处理预测变量之间存在多重共线性的情况，或者当预测变量的数量超过样本数量的情

岭回归是一种重要的线性回归技术，它通过引入正则化项来解决普通最小二乘法在面对特定数据集时可能出现
题。通过调整正则化系数 λ，可以有效控制模型的复杂度和预测能力。

8.4 Lasso regression

Lasso回归的核心概念：

1. **L1正则化**:
   - Lasso回归是Least Absolute Shrinkage and Selection Operator（最小绝对收缩和选择算子）的缩写
     过在损失函数中添加L1正则化项来实现。这个正则化项是模型参数的L1范数（绝对值之和）。
2. **损失函数的变化**:
   - Lasso回归的损失函数为 $L(\beta) = \sum_{i=1}^{n}(y_i - \hat{y}_i)^2 + \lambda \sum_{j=1}^{p} |\beta_j|$，其中 λ 是正则化系数。

Lasso回归的效果和特点：

1. **变量选择和稀疏模型**:
   - 与岭回归不同，Lasso回归能够将某些系数完全压缩至零。这意味着模型可以自动进行变量选择，从
     简洁（稀疏）的模型。
2. **参数收缩**:

- Lasso回归通过压缩系数值来减少模型复杂度，从而防止过拟合，并提高模型的泛化能力。

3. **选择正则化系数 λ**:
   - 正确选择 λ 对模型性能至关重要。λ 的值通常通过交叉验证来确定。
4. **没有解析解，需使用数值方法**:
   - 与岭回归不同，Lasso回归通常没有解析解，需要使用数值优化方法来求解，如迭代收缩阈值算法快速迭代收缩阈值算法（FISTA）。

Lasso回归是一种重要的回归技术，特别适用于处理具有大量特征的数据集，其中许多特征可能是无关紧要的正则化，Lasso有助于生成更精简且易于解释的模型，同时保持良好的预测性能。

8.5 Model Evaluation

1. **均方误差（Mean Squared Error, MSE）**:
   - MSE 是预测值与实际值差异的平方的平均值。它的计算公式为 $MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2$，其中际值，$\hat{y}_i$ 是预测值，n 是样本数。MSE 越小，模型的性能越好。
2. **均方根误差（Root Mean Squared Error, RMSE）**:
   - RMSE 是 MSE 的平方根，表示实际值与预测值之差的标准差。它的计算公式为 $RMSE = \sqrt{MS}$ RMSE 越小，模型的性能越好。
3. **平均绝对误差（Mean Absolute Error, MAE）**:
   - MAE 是预测值与实际值之差的绝对值的平均数。它的计算公式为 $MAE = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i|$。与 RMSE 相比，MAE 对异常值的敏感性较小。
4. **决定系数（R-squared）**:
   - R-squared 衡量模型解释的数据变异性的程度。它的值在 0 到 1 之间，值越高，表明模型对数据的好。计算公式为 $R^2 = 1 - \frac{SS_{res}}{SS_{tot}}$，其中 SS_{res} 是残差平方和，SS_{tot} 是总平方和。
5. **调整后的决定系数（Adjusted R-squared）**:
   - 考虑到 R-squared 值会随着模型中变量数量的增加而增加，即使这些变量对模型的预测能力没有贡引入了调整后的决定系数。它在原有 R-squared 的基础上进行了调整，考虑了模型中变量的数量。

这些评估指标对于判断模型的性能和选择最适合的模型非常重要。MSE、RMSE 和 MAE 通常用于评估模型的性，而 R-squared 和 Adjusted R-squared 则用于评估模型对数据变异性的解释能力。在实际应用中，选择合估指标取决于具体的问题和模型的目标。

ML Part 9. Feature Selection for Dimensionality Redu and Classification

9.1 Introduction

1. **先前的分类器设计**：在之前的部分，分类器的设计使用了所有可用的特征。这部分讨论的是当这些特征数
时，可能出现的问题。

2. **维度诅咒**：所谓的“维度诅咒”是指在处理高维数据时遇到的一系列问题。这里的“维度”指的是数据集中存
数量。由于高维数据而导致的机器学习模型训练困难，被称为“维度诅咒”。维度诅咒的主要问题包括数据
和距离集中。

3. **减少特征数量的原因**：有多个原因促使我们减少特征数量。最明显的原因是计算复杂性。另一个重要原因
器的泛化能力。通常，训练样本数量与自由分类器参数数量的比率越高，分类器的泛化性能越好。

4. **大量特征的影响**：大量的特征直接转化为分类器参数的增加（例如，在神经网络中的突触权重，在线性分
的权重）。因此，对于有限且通常有限的训练样本来说，尽可能减少特征数量符合我们设计具有良好泛化
分类器的愿望。

5. **特征选择或减少的重要性**：这部分的主要任务可以总结为：如何在减少特征数量的同时，尽可能保留它们
区分信息？这个问题被称为特征选择或减少。如果选择了具有较小区分能力的特征，随后设计的分类器性
佳。相反，如果选择了信息丰富的特征，则分类器将具有良好的性能。

6. **特征选择与特征提取的区别**：文献中对特征选择和特征提取的术语有些混淆。简而言之，特征选择是从原
中选择一部分特征，以减少模型复杂性、提高模型的计算效率并改善泛化。而特征提取是从原始特征集中
衍生信息，以创建新的特征子空间。主要目标是在保留大部分相关信息的同时压缩数据。常用的主成分分
（PCA）是一种特征提取方法。

9.2 The Peaking Phenomenon

1. **为了设计泛化性能良好的分类器**：提到为了设计具有良好泛化性能的分类器，训练样本的数量必须相对于
量（即特征空间的维度）足够大。以线性分类器为例，未知参数的数量是特征数量加一。为了对这些参数
好的估计，样本数量必须大于特征数量加一。特征数量越大，估计越好，因为我们可以过滤掉噪声的影响
最小化异常值的影响。

2. **峰值现象的实际影响**：在实践中，对于有限数量的训练样本，增加特征数量会带来初步的性能提升，但超
临界值后，特征数量的进一步增加会导致错误概率增加。这种现象被称为“峰值现象”。峰值现象的一般趋
图所示：随着特征数量的增加，错误的概率先降低，达到一个临界值后又开始增加。

3. **特征数量与训练数据的大小**：因此，在实践中，对于少量的训练数据，应使用少量的特征。如果有大量的
据可用，则可以使用更多的特征以获得更好的性能。

这一部分的核心内容是，虽然增加特征可以提高分类器的性能，但超过某个点后，额外的特征会反而降低性能
现象在机器学习和模式识别中非常重要，因为它影响着特征选择和模型设计的策略。

9.3 Individual Feature Evaluation

1. **特征选择的第一步**：特征选择的第一步是单独查看每个特征，并测试它们对于当前问题的个体区分能力。
独查看特征远非最优，但这种程序有助于识别和丢弃“坏”的特征。这将减轻后续基于集合的特征评估和选
必要的计算负担。

2. **两种个体特征评估方法**：本部分介绍了两种用于个体特征评估的方法，这两种方法基于两种不同的“好”特
释，包括类别可分性和特征的相关性。

3. **费舍尔比率（Fisher's Ratio）**：讨论了一个好的特征应该在不同类别中取不同的值。可以使用费舍尔比一个特征的类别可分性。费舍尔比率的定义是：设两类样本的均值和标准差分别为μ1、μ2和σ1、σ2，则率被定义为：$(\mu1 - \mu2)^2 / (\sigma1^2 + \sigma2^2)$。费舍尔比率的分子实际上是类间差异，而分母是类内散布。费舍尔大，特征的区分性就越强。通常用于连续变量。

4. **互信息（Mutual Information）**：一个好的特征应该包含大量信息以决定类标签的值。可以基于特征和类间的互信息来评估特征的相关性。互信息（MI）是两个变量之间相互依赖的量度。更具体地，它量化了通过另一个变量获得的关于一个变量的信息量。互信息通常用于离散特征。

这一部分的核心内容是，通过对特征的单独评估，我们可以识别和丢弃对分类任务帮助不大的特征，这是特征选择过程的重要一步。通过这种方法，我们可以减少后续特征选择中的计算负担，并确保选择的特征集具有较高的区分能力。

9.4 Feature subset selection

1. **特征子集选择问题定义**：这个问题定义为：给定一个完整的特征集，选择一个产生最佳分类性能的子集。由于维数值现象，应选择并使用所有可用特征的子集作为模式分类模型的输入。这一部分讨论了，尽管个体特征评估对于识别有效特征，但不能简单地选择排名最高的特征形成特征子集，因为这些特征之间可能存在强烈的相关性，导致冗余。

2. **特征子集选择算法的两个主要组成部分**：
   - 搜索算法：用于生成候选特征子集。
   - 评价标准：用于评估由搜索算法生成的候选特征子集的优劣。

3. **搜索算法**：
   - 穷尽搜索方法：遍历所有可能的特征子集组合。
   - 顺序向前选择：从空集开始，逐步添加特征。
   - 顺序向后消除：从完整特征集开始，逐步删除特征。

4. **评价标准**：
   - 基于分类器的分类性能。
   - 基于类别可分性的度量，如马氏距离或散布矩阵。

5. **特征子集选择算法的分类**：
   - 过滤方法（Filter Method）：不涉及分类器，仅基于数据的内在特性。
   - 包装方法（Wrapper Method）：包括分类器在循环中，使用分类性能来评估特征子集。
   - 嵌入方法（Embedded Method）：结合过滤和包装方法的优点，通过内置的特征选择方法实现，例如Lasso（最小绝对收缩和选择算子）。

这部分内容的核心是，特征子集选择对于提高模型的性能和减少计算复杂性至关重要。它涉及到多种不同的方法，以找到最优的特征子集，既减少冗余又保持最大的相关性。

Embedded Method

嵌入式方法（Embedded Method）在特征子集选择中结合了过滤方法（Filter Method）和包装方法（Wrapper Method）的特质。这种方法的特点是通过内置的特征选择机制来选择特征子集，同时考虑到特征选择的准确

的计算效率。以下是对嵌入式方法的更具体解释：

1. **结合过滤和包装方法的特点**：嵌入式方法融合了过滤方法的高效计算和基于数据内在特性的特征选择，以方法基于模型性能的特征选择策略。它们直接在学习算法内部进行特征选择，这意味着特征选择和模型训时进行的。

2. **由特定算法实现**：嵌入式方法通常由特定的算法实现，这些算法在学习过程中自动执行特征选择。与过滤包装方法不同，嵌入式方法不需要一个独立的步骤来进行特征选择。它们通常更加高效，因为特征选择和练是一体化的过程。

3. **Lasso算法作为典型例子**：Lasso（最小绝对收缩和选择算子）是嵌入式方法中最流行的一个例子。Lass入正则化项（L1惩罚项）来收缩系数（参数）向零。这个过程不仅有助于防止过拟合，而且能实现特征选某个特征的系数变为零时，该特征就被模型排除在外。这种方法在回归分析中尤其常用，但也可以应用于型的机器学习模型。

4. **对特征数量的惩罚机制**：Lasso等嵌入式方法通过对模型中特征数量的惩罚，自动选择重要的特征。这意不仅优化模型的性能，而且还通过减少模型复杂度来提高计算效率。

综上所述，嵌入式方法在特征选择过程中实现了计算效率和模型性能的平衡。通过这种方法，算法能够自动选型最有用的特征，从而优化模型的整体性能。

ML Part 10. Clustering Analysis

1. 聚类分析简介
    - **无监督学习**：与有监督学习（使用标记数据）不同，无监督学习处理未标记的数据样本。
    - **应用领域**：例如市场细分、搜索结果分组等。
2. 聚类分析基础
    - **相似性度量**：通过计算样本间的距离来度量相似性，例如闵可夫斯基距离（Minkowski Distance）和得距离（Euclidean Distance）。
    - **聚类算法类型**：包括基于质心的聚类（如K均值聚类）、层次聚类、基于分布的聚类和基于密度的聚
3. 基于质心的聚类
    - **K均值聚类**：最流行的聚类方法之一，旨在将数据划分为K个簇。
    - **聚类准则**：使用“群内平方和”（Within Cluster Sum of Squares, WCSS）作为优化准则。
4. 层次聚类
    - **算法类型**：包括自底向上的聚合算法和自顶向下的分裂算法。
    - **特点**：不需要预先指定簇的数量，适合于不同规模和密度的数据集。
5. 基于分布的聚类
    - **高斯混合模型**（Gaussian Mixture Model, GMM）：将数据建模为多个高斯分布的混合。
    - **参数估计**：使用期望最大化（Expectation-Maximization, EM）算法来估计模型参数。
6. 基于密度的聚类

- **DBSCAN**（Density-Based Spatial Clustering of Applications with Noise）：识别并连接高密度区簇。
- **优点与挑战**：鲁棒性高，能处理噪声数据，但在处理不同密度的数据集时可能面临挑战。

10.1 Introduction

第10.1节“介绍”部分的内容是关于聚类分析的。在这一节中，首先指出了之前的学习假设是基于标记数据的，标记数据的过程称为监督学习。然后，介绍了非监督学习，它使用未标记的样本。非监督学习不需要标记训练的监督，因此非常适合于发现数据中的模式和分组。

非监督学习主要包括两个主题：聚类分析和关联分析。本课程重点关注聚类分析，它已经广泛应用于市场细分结果分组等领域。聚类分析大致上是指，通过聚类或数据点分组来描述数据，这些数据点在内部具有较强的相正式的聚类过程使用一个标准函数，并寻找优化该函数的分组。

10.2 Basics of Clustering Analysis

1. **聚类定义和目标**：聚类大致上是指，通过聚类或数据点分组来描述数据，其中这些数据点内部具有较强的性。形式化的聚类过程使用一个标准函数，并寻找优化该函数的分组。
2. **相似性度量**：聚类问题的描述是寻找数据集中的自然分组，需要定义两个问题：一是什么是自然分组，二聚类/组中的样本相互之间比其他聚类中的样本更相似的意义。最常见的相似性度量是样本之间的距离，聚类中的样本距离相对较小。
3. **距离度量**：常见的距离度量包括欧几里得距离、曼哈顿距离、闵可夫斯基距离等。还可以使用其他度量方马哈拉诺比斯距离和余弦相似度。
4. **聚类算法类型**：
   - **基于质心的聚类**（划分方法）：例如k均值聚类算法，将数据划分为非层次性的组。
   - **层次聚类**：不需要预先指定聚类数量，通过创建树形结构（树状图）来划分数据集。
   - **基于分布的聚类**：根据数据属于特定分布的概率进行划分，如高斯混合模型（GMM）。
   - **基于密度的聚类**：连接高密度区域的数据点形成聚类，如DBSCAN算法。

这些方法在不同的应用场景中有不同的优势和局限性，选择合适的方法取决于数据特性和分析目标。

10.3 Centroids-based Clustering

1. **基于质心的聚类概念**：基于质心的聚类是一种简单但有效的聚类算法。其核心思想是，一个聚类可以通过心向量来表征，且距离这个中心向量较近的数据点被分配到相应的聚类中。
2. **k-均值聚类算法**：k-均值是最受欢迎的基于质心的聚类方法之一。其基本思想是将n个样本划分为k个互不子集，每个子集代表一个聚类，子集内的样本比不同子集间的样本更相似。
3. **内部聚类和平方和准则（WCSS）**：k-均值聚类使用内部聚类和平方和（WCSS）作为准则。这个准则的是：对于给定的聚类，其均值向量是聚类中样本的最佳代表，因为它最小化了误差向量的平方和。
4. **k-均值聚类算法过程**：算法包含几个步骤：
   - **初始化**：设置聚类数k，并随机生成k个点或质心作为初始聚类中心。

- **分配样本**：将每个数据点分配给最近的质心，形成k个预定义的聚类。
- **计算新质心**：计算每个聚类的新质心。
- **迭代**：重复分配样本和计算新质心的步骤，直到质心和样本的分配不再改变。

5. **选择k值的方法**：选择k值的一个常用方法是肘部法则。该方法通过计算不同k值下的WCSS值，并在WCS的曲线上找到一个明显的弯曲点（肘部）来确定最佳的聚类数。

这些概念和方法为理解和应用基于质心的聚类提供了基础。特别是k-均值聚类，由于其简单性和有效性，在实中非常流行。

10.4 Hierarchical Clustering

第10.4节"层次聚类"介绍了层次聚类方法。层次聚类是一种常用的数据结构总结方法，可以通过创建一个嵌套分集来表示，这通常以树形图（树状图）的形式展现。

以下是这部分的主要内容：

1. **层次聚类算法**：层次聚类算法可以分为两种：
   - **聚合算法（自底向上方法）**：以每个样本作为一个单独的簇，逐渐合并成对的簇，直到所有的簇合并包含所有样本的簇。
   - **分裂算法（自顶向下方法）**：从包含所有样本的单一簇开始，逐渐分裂簇，直到每个簇只包含一个样
2. **层次聚类的优势**：与k-均值聚类算法不同，层次聚类不需要事先指定簇的数量。
3. **聚合算法的实现**：
   - 初始时，将每个样本视为一个单独的簇。
   - 根据样本之间的相似性或距离，将最接近的两个簇合并。
   - 重复上述合并步骤，直到所有样本都被合并到一个簇中。
4. **相似性度量**：在合并过程中，可以通过不同的方法来度量两个簇之间的相似性，例如：
   - **单链接（最近邻居）**：两个簇之间的距离定义为簇内所有样本对的最小距离。
   - **完全链接（最远邻居）**：两个簇之间的距离定义为簇内所有样本对的最大距离。
   - **中心链接**：两个簇之间的距离定义为簇的质心之间的距离。
   - **平均链接**：计算每对数据之间的距离，将这些距离加总后除以数据对的总数，得到两个簇之间的平均
5. **树状图（Dendrogram）**：层次聚类过程可以通过树状图来表示，展示了样本之间的层次关系。

总体来说，层次聚类是一种强大的工具，能够揭示数据的层次结构，并且不需要预先指定簇的数量，使其在某下比基于质心的方法更有优势。

10.5 Distribution-based Clustering

第10.5节"基于分布的聚类"探讨了以数据分布为基础的聚类方法，特别是高斯混合模型（Gaussian Mixture M GMM）。以下是这部分内容的要点：

1. **基于分布的聚类概念**：这种聚类方法假设数据是由不同的概率分布生成的，最常见的是高斯分布。基于分

类的核心思想是模拟数据集作为几个高斯分布的混合。

2. **高斯混合模型（GMM）**：GMM是一种特别的分布式聚类方法，它假设数据是由多个高斯分布组成的混合
个高斯分布代表一个聚类，其参数包括均值（代表聚类中心）和协方差（表示聚类的形状和尺寸）。

3. **参数估计**：GMM的参数可以通过期望最大化（Expectation-Maximization, EM）算法估计。EM算法交替
步骤：在期望步骤（E-step）中，算法估计每个样本属于每个聚类的概率；在最大化步骤（M-step）中
新每个聚类的参数。

4. **聚类分配**：一旦估计了GMM的参数，每个数据点可以被分配到最有可能生成它的高斯分布所代表的聚类

5. **GMM应用**：GMM不仅用于聚类分析，还广泛用于监督学习中的类条件概率密度函数建模。

6. **优点和局限性**：GMM聚类的一个主要优点是它不仅可以找出聚类中心，还可以评估聚类的形状和大小。
而，GMM假设数据分布是高斯分布，这在某些数据集中可能不是合理的假设。

总体而言，基于分布的聚类提供了一种强大的工具来识别和模拟复杂的数据集中的潜在结构。高斯混合模型因
性和强大的建模能力而成为这一方法的一个突出例子。

10.6 Density-based Clustering

第10.6节"基于密度的聚类"详细介绍了基于数据点密度的聚类方法，特别关注了DBSCAN（Density-Based S
Clustering of Applications with Noise）算法。以下是这部分的主要内容：

1. **基于密度的聚类概念**：这种聚类方法侧重于识别密集区域，并将其作为簇，而将稀疏区域的数据点视为噪
常值。这与k均值和层次聚类方法不同，后者更适合于识别球形或凸形簇。

2. **DBSCAN算法原理**：

   - DBSCAN算法基于两个主要概念：核心点和ε-邻域。核心点是指在其ε-邻域内有足够多（至少MinP
居的点。
   - 如果一个点在核心点的ε-邻域内，或者可以通过一系列核心点达到，那么这个点就被认为是可达的。
   - 算法从任意未被访问的点开始，探索该点的ε-邻域，如果它是一个核心点，则开始形成一个新簇。然
地将直接密度可达的所有点添加到该簇中。

3. **DBSCAN的参数**：

   - **MinPts**：确定核心点所需的最小邻居数。
   - **ε（Epsilon）**：用于定义点的邻域的距离阈值。

4. **DBSCAN的优点**：

   - 能够识别任意形状的簇。
   - 对噪声和异常值具有良好的鲁棒性。
   - 不需要预先指定簇的数量。

5. **DBSCAN的局限性**：

   - 对于不同密度的簇可能难以处理。
   - ε和MinPts的选择对结果有显著影响。

- 在高维数据中选择合适的参数可能较为困难。

总体来说，DBSCAN是一种有效的密度聚类方法，特别适用于数据集中簇的形状和大小各异，或者数据中包含异常值的情况。

ML Part 11. Clustering Evaluation

总体介绍

这个PPT是关于聚类评估（Clustering Evaluation）的详细介绍，涉及多种不同的评估指标。以下是每个幻灯详细解释：

1. **11.1 引言**
   - 聚类评估是指确定生成的聚类有多好的任务。
   - 假设有N个样本被聚类算法划分为K个不相交的子集（即聚类），每个子集包含一定数量的样本。
   - 常用的聚类评估指标包括：轮廓系数（Silhouette coefficient）、邓恩指数（Dunn index）、戴维斯指数（Davies-Bouldin index）、卡林斯基-哈拉巴斯指数（Calinski-Harabasz index）、兰德指数index）。
   - 后续将详细介绍每个评估指标。
2. **11.2 轮廓系数**
   - 轮廓系数是一种衡量每个样本在其分配的聚类中适应程度的指标，它结合了以下两方面的信息：
     a. 内聚度：样本与同一聚类中其他样本的接近程度。
     b. 分离度：样本与其他聚类中样本的远离程度。
   - 轮廓系数的计算方法详述，并提到其值域为-1到1。
   - 轮廓系数较高表示样本聚类效果好，具有清晰的聚类分离和紧密的聚类内聚。
   - 轮廓系数较低则可能表示聚类准确度较低，聚类之间可能有重叠或样本分配不佳。
3. **11.3 邓恩指数**
   - 邓恩指数旨在量化聚类的紧凑性和方差。
   - 一个聚类如果样本间方差小，则认为是紧凑的。
   - 两个聚类如果彼此距离较远，则认为分离良好。
   - 邓恩指数的定义和计算方法详述，其值越高表示聚类越紧凑且分离良好。
4. **11.4 戴维斯-波尔丁指数**
   - 戴维斯-波尔丁指数基于聚类内部分散和聚类间分离来评估聚类的好坏。
   - 对于每个聚类，首先计算其质心（聚类中心）。
   - 聚类内部分散（即聚类内离散度）和聚类间的分离（即聚类间距离）的定义和计算方法详述。
   - 戴维斯-波尔丁指数越小表示聚类定义得越好。
5. **11.5 卡林斯基-哈拉巴斯指数**
   - 卡林斯基-哈拉巴斯指数是衡量样本与其所在聚类（内聚）与其他聚类（分离）相似性的指标。

- 它首先估算聚类内样本与聚类质心的距离来计算内聚度。
- 分离度是基于聚类质心与全局质心的距离来估算的。
- 卡林斯基-哈拉巴斯指数越高，意味着聚类越密集且分离良好。

6. **11.6 兰德指数**
   - 兰德指数是通过

比较两种聚类方法下样本对分配情况的相似度来计算的。

- 它统计所有样本对在两种不同聚类方法下分配在同一聚类或不同聚类的情况。
- 兰德指数的计算方法详述，其值在0到1之间。
- 0表示两种聚类方法在任何数据对的聚类上都不一致，而1表示完全一致。

每个指标都通过算法公式和相关概念进行了详细说明，以及如何应用这些指标来评估聚类的效果。此外，PP了使用这些指标的具体例子和图表。

11.1 Introduction

在PPT的第1节“11.1 引言”中，内容主要介绍了聚类评估的基本概念和重要性。具体内容如下：

- **聚类评估的定义：** 聚类评估指的是确定生成的聚类有多好的任务。这是机器学习中一个重要的步骤，用类算法的有效性和效率。
- **聚类的基本假设：** 假设有N个样本被一个聚类算法划分成K个不相交的子集（即聚类）。每个子集包含一样本，这些样本在某种度量下彼此相似。
- **聚类评估指标：** 幻灯片列举了一些常用的聚类评估指标，包括：
  i. 轮廓系数（Silhouette Coefficient）
  ii. 邓恩指数（Dunn Index）
  iii. 戴维斯-波尔丁指数（Davies-Bouldin Index）
  iv. 卡林斯基-哈拉巴斯指数（Calinski-Harabasz Index）
  v. 兰德指数（Rand Index）
- **指标介绍：** 幻灯片接着提到，接下来的部分将详细介绍每个评估指标。这意味着在随后的幻灯片中，每会被详细解释，包括其定义、计算方法以及在聚类评估中的应用。

这一部分为整个聚类评估讲座的基础，为听众提供了关于聚类评估的基本理解和后续内容的预览。

11.2 Silhouette coefficient

在PPT的第2节“11.2 Silhouette Coefficient”中，内容主要介绍了轮廓系数的概念和重要性。具体内容如下：

- **轮廓系数的定义：** 轮廓系数是一种衡量每个样本在其分配的聚类中适应程度的指标。它结合了以下两方息：
  i. **内聚度（Cohesion）**：衡量一个样本与同一聚类中其他样本的接近程度。
  ii. **分离度（Separation）**：衡量一个样本与其他聚类中样本的远离程度。

- **轮廓系数的计算：**
  - 对于特定样本，轮廓系数是通过计算内聚度和分离度的差值与这两者中较大值的比率来定义的。
  - 内聚度是指样本与同一聚类中其他样本之间的平均距离。
  - 分离度是指样本与属于其他聚类的所有其他样本之间的最小平均距离。
- **轮廓系数的范围和含义：**
  - 轮廓系数的值域在-1到1之间。
  - 较高的轮廓系数表示样本聚类效果好，具有清晰的聚类分离和紧密的聚类内聚。
  - 相反，较低的轮廓系数可能表明聚类准确度较低，聚类之间可能有重叠或样本分配不佳。
- **实例应用：**
  - 幻灯片中还包括了一个实例，展示了如何在具体数据集上计算轮廓系数，并根据轮廓系数的结果来评
    数量的聚类对数据集的影响。

这一部分详细介绍了轮廓系数的计算方法和其在聚类评估中的应用，使听众能够理解如何利用这一指标来判断
法的有效性。

11.3 Dunn Index

在PPT的第3节“11.3 Dunn Index”中，详细介绍了邓恩指数（Dunn Index）的概念和计算方法。以下是具体内

- **邓恩指数的目的：** 邓恩指数旨在量化聚类的紧凑性和方差。它是一种用于评估聚类效果的指标。
- **紧凑性的定义：**
  - 一个聚类如果样本间方差小，则认为是紧凑的。
  - 聚类的紧凑性可以通过最大距离（最大方差）来量化，用公式表达为 $\delta = \max_{i \neq j} d(i, j)$，其中 $d($
    同一聚类中两个样本 i 和 j 之间的距离。
- **分离性的定义：**
  - 两个聚类如果彼此距离较远，则认为分离良好。
  - 聚类的分离性可以通过最小距离（最小方差）来量化，用公式表达为 $\Delta = \min_{i \neq j} d(i, j)$，其中 d
    示不同聚类中两个样本 i 和 j 之间的距离。
- **邓恩指数的计算：**
  - 邓恩指数是通过比较聚类的紧凑性和分离性来计算的。
  - 具体来说，邓恩指数定义为聚类分离度（最小距离）与聚类内方差（最大距离）的比率，计算公式为
    $\text{Dunn Index} = \frac{\Delta}{\delta}$。
- **邓恩指数的含义：**
  - 较高的邓恩指数表明聚类更加紧凑且分离良好，即聚类内部差异小，而不同聚类之间差异大。
  - 较低的邓恩指数表明聚类较不紧凑或分离不明显。
- **示例应用：**

- 幻灯片还包含了一个示例，展示了如何在具体数据集上计算邓恩指数，并根据邓恩指数的结果来评
  量的聚类对数据集的影响。

邓恩指数通过量化聚类的紧凑性和分离性，为评估聚类效果提供了一个有力的工具。通过这个指数，我们可
同聚类算法和聚类数量对于数据集的聚类效果的影响。

11.4 Davies-Bouldin index

在PPT的第4节“11.4 Davies-Bouldin Index”中，详细介绍了戴维斯-波尔丁指数（Davies-Bouldin Index）的
算方法。以下是具体内容：

- **戴维斯-波尔丁指数的目的：** 此指数也用于评估聚类的好坏，它基于聚类内部的离散度（分散）和聚类间
  计算。
- **计算过程：**
  - **聚类中心（质心）的计算：** 对于每个聚类，首先计算其质心（聚类中心）。
  - **聚类内部分散（离散度）：** 定义为聚类内所有样本与质心之间的平均距离。
  - **聚类间的分离：** 定义为不同聚类的质心之间的距离。
- **相似度的定义：**
  - 对于任意两个聚类，其相似度定义为其内部分散（离散度）之和除以它们的质心间距离。
- **戴维斯-波尔丁指数的定义：**
  - 每个聚类的相似度由其与最相似（即最接近）的其他聚类确定。
  - 戴维斯-波尔丁指数是所有聚类相似度的平均值。
  - 此指数的最优值是最小值，即较小的戴维斯-波尔丁指数表示聚类定义得更好，因为聚类内部紧密而
    分离明显。
- **示例应用：**
  - 幻灯片包含了一个示例，展示了如何在具体数据集上计算戴维斯-波尔丁指数，并根据这个指数的结
    不同数量的聚类对数据集的影响。

戴维斯-波尔丁指数通过衡量聚类内部的一致性和聚类间的分离度，为评估聚类效果提供了一种有用的方法。
指数，我们可以判断不同的聚类算法和聚类数量对于数据集的聚类效果的影响。

11.5 Calinski-Harabasz Index

在PPT的第5节“11.5 Calinski-Harabasz Index”中，详细介绍了卡林斯基-哈拉巴斯指数（Calinski-Harabasz
概念和计算方法。以下是具体内容：

- **卡林斯基-哈拉巴斯指数的目的：**
  - 此指数用于衡量样本与其所在聚类（内聚）与其他聚类（分离）的相似性。
  - 它是一个评估聚类效果的重要指标，关注于聚类的密集程度和聚类之间的分离程度。
- **内聚（Cohesion）的估算：**

    - 内聚是基于样本在聚类内与聚类质心之间的距离来估算的。
    - 对于每个聚类，计算聚类内所有样本与其质心之间的距离平方和。

- **分离（Separation）的估算：**
    - 分离是基于聚类质心与全局质心之间的距离来估算的。
    - 全局质心是根据所有样本的平均位置计算得出的。

- **卡林斯基-哈拉巴斯指数的计算：**
    - 卡林斯基-哈拉巴斯指数是聚类间分离度和聚类内部的内聚度之比的总和。
    - 具体的计算公式涉及到聚类数量、样本总数以及上述内聚和分离的估算值。

- **指数的含义：**
    - 较高的卡林斯基-哈拉巴斯指数表示聚类效果好，意味着聚类内部紧密而聚类之间分离明显。
    - 通常，更高的值指示更好的聚类效果。

- **示例应用：**
    - 幻灯片还包括了一个示例，展示了如何在具体数据集上计算卡林斯基-哈拉巴斯指数，并根据这个指来评估不同数量的聚类对数据集的影响。

卡林斯基-哈拉巴斯指数通过评估聚类的内聚和分离程度，提供了一种评价聚类效果的有力工具，特别是在确类数量时特别有用。通过这个指数，我们可以判断不同的聚类算法和聚类数量对于数据集的聚类效果的影响。

11.6 Rand Index

在PPT的第6节“11.6 Rand Index”中，详细介绍了兰德指数（Rand Index）的概念和计算方法。以下是具体内

- **兰德指数的目的：**
    - 兰德指数是一种常用的度量方法，用于计算两种聚类方法之间的相似度。
    - 它通过考虑所有样本对，并统计在两种不同聚类方法下被分配到相同或不同聚类的样本对数量。

- **兰德指数的计算：**
    - 兰德指数的公式为 $RI = \frac{a+b}{\binom{n}{2}}$，其中：
        - a 是在两种聚类方法中都被分配到同一聚类的样本对数。
        - b 是在两种聚类方法中都被分配到不同聚类的样本对数。
        - $\binom{n}{2}$ 是所有可能的样本对数（n个样本中任选两个的组合数）。

- **兰德指数的值域和含义：**
    - 兰德指数的值在0到1之间。
    - 值为0表示两种聚类方法在任何数据对的聚类上都不一致。
    - 值为1表示两种聚类方法在每个数据对的聚类上完全一致。

- **示例应用：**

- 幻灯片可能还包括了一个示例，展示了如何在具体数据集上计算兰德指数，并根据这个指数的结果来
  同聚类方法之间的一致性。

兰德指数是一个有效的工具，用于比较和评估不同聚类方法的相似度，特别是在没有标签数据可用于监督学习
下。通过这个指数，我们可以衡量不同聚类方法在划分数据集时的一致性程度。

期末复习

注意点

遗传算法
问题的分类，3个问题
特征提取、取子集的方式
fisher lda的公式
k-fold和别的fold的过程
合并的树怎么画，single linkage和别的方式
聚类方式
连续数字的遗传算法和编码的遗传算法
几个NP问题的区别
各种crossover的方式
贝叶斯分类 p3 23
fisher lda
评价分类器的指标
多种feature selection方法
聚类方法，场景
DBSCAN聚类的过程

各种crossover方法

怎么算逆矩阵

计算一个矩阵的逆矩阵通常涉及几个步骤，特别是对于较大的矩阵。以下是一个标准的方法，适用于任何可逆
阵：

1. **检查矩阵是否可逆**：
   - 确保矩阵是方阵（即行数和列数相同）。
   - 计算矩阵的行列式。如果行列式为0，则矩阵不可逆。
2. **创建增广矩阵**：
   - 在原矩阵的右侧增加一个同等大小的单位矩阵，形成一个增广矩阵。例如，对于一个2x2矩阵 A，增
     $[A|I]$，其中 I 是单位矩阵。
3. **使用高斯-约当消元法**：
   - 通过行操作（如行交换、行倍乘、行相加）将矩阵的左半部分（原矩阵 A）转换为单位矩阵。