给你因为你告诉过我好吧 Okay.大家早上好大家都还好吗？因为天气越来越冷了我记得第一周的时候 大家都在抱怨天气不好大家都还好吗？别这样好的别担心苏格兰通常不会太冷所以整个冬天的气温都在零度左右但也不会达到零下几度，所以如果你觉得已经很冷了，也不用太担心，尤其是早上。那就裹得暖和些吧。我看到你已经穿着外套上课了。我希望这不是个好兆头。你觉得这里很冷还是还好？因为时间还早，所以你觉得很舒服好吧很好，很好我原本计划这周做客座讲师 你们可能在教案上看到了不幸的是，上周我收到了客座讲师的邮件，说他今天不能来，所以他问我能不能改期，这就是为什么我们要把两场讲座对调一下。因此，这原本是第七周的讲座。所以这周我们会讲集群。然后到了第七周，我们就不讲集群了，而是请嘉宾讲课。所以这只是个交换下周是你的阅读周所以下周我们不会有任何讲座或计算机实验室。所以下周将是你们补习的机会，你们可以读读书，做做评估，或者完成你们的作业。所以，下周请不要来听课，因为我不会在这里，我想没有我的课堂就不会那么有趣了。我的意思是，我想你可以进来，但没有人会在那里，所以这真的没有意义。让我把门关上吧我这么说是因为我们要把东西调换一下你会看到，我们这周做的计算机实验室将更侧重于数据分割和重采样步骤。然后，我们将在第七周进行聚类计算机实验室，届时我们的特邀嘉宾将发表演讲，因为无论如何，我们都不会在实验室中涉及任何新材料。所以这是个很好的要求。在这个时间做聚类实验再好不过了。所以，如果你看了计算机实验室的笔记，也不用担心。这只涉及到本次讲座的一半内容。好了，在上周的最后一讲中，我们谈到了回归模型的评估。希望你们都还记得。我们讨论了简单的多重逻辑回归，以及如何对其进行评估。我们在矩阵中查看了不同类型的误差。我们还查看了 C 曲线。我们还简短地谈到了正则化。这就是使用套索或回归的概念，它允许你在拟合模型的同时进行变量选择，如果你有很多合适的变量，这就非常方便了。然后，我们还开始讨论数据分割、重采样，特别是交叉验证。我们还想谈谈类不平衡的处理。由于上周我对时间管理不当，我们没能讲到这一点。所以，我保证这周我们会做的。所以，我们将从回顾上周有点仓促的内容开始。也就是说，我们将再次简要谈谈数据分割和交叉验证。我们还将讨论类不平衡处理，这在许多应用中都非常重要，尤其是在信用评分中。因此，在很多情况下，你只有少数几个特定类别的案例。因此，如果只是在这种不平衡的不均衡数据上训练预测模型，我们将无法学会对两个类别进行均衡预测。因此，在这种情况下，模型可能只预测最常见的类别，因为在大多数情况下，这才是正确的预测。但这也意味着，该模型并不适合用来平均预测两个类别。因此，在这种情况下，我们会使用几种不同的方法对数据进行超采样或低采样，使其分布更加均匀。今天的重头戏是无监督机器学习。耶！我最喜欢的话题。我们将讨论聚类算法。我们将重点讨论分区聚类和分层聚类。我将简要介绍几种你可能感兴趣的其他方法，如果你对此感兴趣，还可以进一步阅读。因此，我还会推荐几本书和推荐读物。本周只介绍威滕和詹姆斯。你会注意到，我们的好朋友库恩和约翰逊在他们的书中没有提到聚类，这很公平。你会发现，与其他方法相比，聚类是一种比较特殊的方法，尤其是在预测建模方面。这就是为什么在你接触到的所有预测分析书籍中都没有提到聚类的原因。究其原因，大多数预测分析书籍更关注有监督的回归和分类方法。好的。让我们从上周关于数据分割的这张非常重要的幻灯片开始。大家应该还记得，过去我们总是基本上在整个数据集上训练我们的模型。因此，我们有整个数据集。我们查看了所有可用的变量。我们有某种输出值，一个类标签或一个数字，诸如此类。然后，我们尝试用整个数据建立一个模型，然后预测整个结果。现在我们要引入的想法是，我们不再使用整个数据集来建立模型，而是将数据分成两部分。一部分是训练数据。这就是我们要建立模型的数据。另一部分是测试数据。这部分数据是用来测试模型的。我们这样做的原因是，如果你在整个数据集上拟合和调整模型，那么有时你的模型会变得非常习惯于数据。因此，它只能预测通过数据集学习到的特定情况。但你真正好奇的是，你的模型在处理未来可能出现的新数据时表现如何？因此，我们想看看我们的模型从训练数据中学到的规则泛化到未见数据（即我们的测试数据）的能力如何。现在，根据你的模型有多复杂，你可能会进行三步拆分，先在训练数据上训练模型，然后在验证数据上进行实际的参数优化和调整，这可能是一个漫长的过程。最后，你还要将测试数据作为一个保留样本，然后在此基础上进行测试。因此，在这种情况下，我提到的重要一点是，你的测试数据应该是不可见的。我的意思是，你应该尽量避免测试数据中的任何信息泄漏到你的训练过程中，因为一旦发生这种情况，你就无法真正评估你的模型在未见数据上的表现，因为它并不是未见的。它只是隐约可见，或者部分信息已经泄露或溢出。因此，测试数据必须完全分开保存。举例来说，这意味着在大多数情况下，我们首先要拆分数据，然后分别对这两半数据进行预处理。这意味着，例如，如果你要进行任何转换或类不平衡处理，我们稍后会讲到。所有这些步骤都应分别针对训练数据和测试数据进行。这样才能真正确保你在两者之间选择的任何参数或任何因素都不会外溢。现在，通过 k 倍交叉验证，我们将这一点发挥到了极致。因此，在这种情况下，我们不是分成两组。我们是将训练数据和测试数据分成两组。然后，我们再将训练数据进一步拆分。因此，我们基本上对训练数据集做了同样的事情。测试数据保存在安全的地方，我们不会去碰它。你把训练数据拆分成 K 个小集，K 是一个参数，由你自己选择。这主要取决于你的计算能力，因为 K 越多，成本就越高。但你要把数据分成大小大致相同的 k 组，然后拟合模型。所以，你要在所有这些减去一个的基础上训练你的模型。举例来说，如果你把数据分成五份，然后在其中四份上训练模型，在第五份上测试拟合模型，然后做同样的事情，把它放回去，重新开始整个过程。你再取四组数据，在第五组数据上进行测试，重复这个过程，直到每一部分数据都有机会进入数据拟合的训练和测试部分。然后重复这一过程，并报告每一轮的评估指标。例如，如果 k 等于 5，你就会进行五次分割。那么你就应该得出五个准确度指标。然后你报告的最终结果将是所有这五个指标的平均值。图片看起来是这样的。你有整个数据集。你把它分成三个部分。你用其中三分之二的数据建立模型，然后对最后三分之一的数据进行预测，然后第二轮你再次用三分之二的数据建立模型，对其中一个弃权的数据进行预测，第三轮用三分之二的数据和一个弃权的数据进行预测。现在，上周有人问我，我们到底用测试数据做了什么，因为你提到不应该在训练部分使用测试数据，但这就是在做这件事。但请记住，我们在最后和最开始都有一些没有使用的测试数据。因此，这并不是整个培训和测试的一部分。我的想法是，你在 K 折样本中产生的这些评估措施。这是一种临时结果。因此，我认为这是一种正在进行的中期准确性评估措施。在变体中，你仍然有完全保留的测试集。因此，你仍然可以测试实际上的模型。拟合模型，我们在最后保留样本，并报告真正的保留结果。话虽如此，但由于模型每一轮都是全新建立的，因此你所得出的这些临时测量结果，仍然是该轮模型中未见过的数据。因此，它仍然是一个可以报告的有效指标。所以你仍然可以报告。这不是训练。它不是一种样本内结果。它仍然是交叉验证那一轮的样本外结果。我们还简要讨论了时间序列数据。这又回到了我们如何分割数据的问题上。可以随机分割。在大多数情况下，你会随机进行 k 次折叠，但在某些情况下，确保在构建样本时保留时间序列结构是有意义的。所以我提到，时间序列数据很特殊，因为观测数据是相互依赖的。因此，它们不是独立的观测值，因为每个观测值都依赖于前一个或多个观测值。因此，如果你对数据进行拆分，随机取样并没有实际意义。相反，您应该对时间序列进行分块，这样才能保留数据的时间结构。同样，如果你有空间数据，在很多情况下，选择特定的地点并将它们放在一起，而不是在所有地方随机取样，也是有意义的。因为这样可以保留依赖结构。没错。所以也有一种选择 K 的方法。他们尊重你样本中的群体。所以，如果你回想一下我们在讲分层抽样时的数据分析原理讲座，这就变得很有趣了。因此，如果模型在数据的不同子群中表现同样出色，这对你来说很重要，那么你就会想要进行分层抽样来形成 K 组。在这种情况下，有许多可供选择的重采样技术，因为我们整天都得做事。因此，我们发明了我们认为更好的新技术和新方法。例如，其中之一就是撇除交叉验证（leave one out cross-validation）。这是 k 折叠交叉验证的一种极端情况，k 等于样本的数量。因此，你实际上会创建。例如，如果你有 1000 条记录，你会做 1000 k 次交叉验证，即训练一个 999 预测一个，训练一个 999 预测一个。然后重复进行。你可以想象，如果你有一个庞大的数据集，这样做的成本有多高。但你也可以想象，如果你的数据集较小，这样做的准确性有多高。因此，我们有自举法、重复训练测试法、分割法，所有这些方法都可以改变交叉验证中的抽样过程。举例来说，自举法就是用替换法进行抽样。因此，你实际上是为 k 折交叉验证取样。然后再把所有记录放回去，从中再随机抽样。因此，记录可以重复出现在每个折叠的测试或训练部分。所有这些在教科书中都有很好的总结。如果你想了解更多，可以看看。就这样在计算机实验室中，我们还将进行K折交叉验证。好的这是一个简短的参数调整过程的图片，这也回到了 "测试数据从何而来 "的问题上。测试数据从何而来？通常情况下，我们首先要定义一组调整参数值。这可以说是最困难的步骤之一，因为你怎么知道从哪里开始调整过程才是合理的假设。而且，这还会影响你的调整效率，以及你实际获得最优解的程度。最常见的方法是查阅文献，看看类似的研究，看看他们发现了哪些参数，然后使用这些起始参数，因为它们已经是相对较好的估计值了。如果没有类似的研究，你就必须凭直觉和一些专家知识来确定合理的直径。在很多情况下，你也会选择一个小号、一个中号、一个大号，然后检查它们的性能如何。因此，它们会给你一个可能寻找最佳参数范围的想法，但这有点像巫术，所以非常模糊，你并不真正确定你在做什么。因此，我认为这实际上是最困难的步骤之一。所以，我们接下来要做的就是，例如，我们对每一组我们认为可能拟合得很好的参数进行拟合。我们会对数据进行采样，拟合模型，然后预测我们的保留率。这就是我们所做的主要重复过程。例如，通过 K 折交叉验证，我们反复拟合并检查参数在特定模型中的表现。最后，你会得到某种性能曲线。举例来说，这就是你通过 k 折交叉验证获得的不同准确度。然后，你就可以据此确定最终的最佳调整参数。我把它们放在引号里，是因为我们将在明天的讲座中简要讨论，寻找最佳参数可能非常棘手。在很多情况下，你真正能做到的只是找到局部最优，而不是全局最优。因此，举例来说，我们使用的很多优化参数，比如梯度下降，都很容易只在参数空间中找到局部最优。我们将在有关神经网络的讲座中更多地讨论这个问题，在那里它们也变得非常重要。因此，我们希望最终能得到一组合理的参数，并以此在实际训练集上重新拟合模型。在这种情况下，他们建议使用整个训练集，而不是所有的 k 个折叠。然后，我们可以用它在最后一轮中预测我们的测试集，这样我们就能得到该特定模型的最终预测值和最终准确率值。因此，这是唯一一次真正使用测试数据的机会。解决不平衡问题。正如我提到的，这一点非常重要，它主要是类预测中的一个问题。因此，样本中类别的频率会影响模型在不同类别上的训练效果。因此，如果你有一个或多个这样的类，而且它们的比例非常低，因此在你的训练数据中非常罕见，那么这就会影响你的模型预测或预测这些低出现率类的能力。因此，如果遇到这种情况，一定要非常小心。实际上，在市场营销的很多应用中，它们都非常常见。例如，当我们试图预测客户流失率时，就经常会遇到这种情况。因此，在很多情况下，我们的客户，我们的客户样本中，有很多客户一直在使用，而很少有客户流失。但实际上，流失的客户正是我们感兴趣的客户。因此，我们试图预测哪些客户可能会离开公司，转而投奔竞争对手。但如果我们没有大量这些数据点，就很难训练信用评分模型。同样，你也会注意到，比如在你的小组作业中，拖欠工资的人比其他人更少。因此，训练你的模型来预测这种情况是相当困难的。因此，如果你已经知道存在类不平衡问题，那么你显然可以通过选择数据的方式来避免这种情况的发生。因此，如果你知道期刊很少见，那么你就会尝试更频繁地对期刊进行调查，从而获得更多期刊数据。阶级失衡不会一开始就发生。这并不总是可能的。在某些情况下，这些人根本就不存在，所以你不可能问他们更多的问题。在某些情况下，你只有在完成收集工作后才能真正了解问题所在。因此，你无法回到数据收集阶段。在这种情况下，你就需要对数据进行 "下取样 "或 "上取样"。上采样基本上是增加更多的数据点，特别是增加更多稀有类别的数据点。而通过下采样，你会减少更频繁出现的类别。这样做的目的只是为了在两者之间取得平衡。因此，选择哪种方法其实并不重要，只要能达到以下目的即可。最终，你的类会相对均衡。它们不一定要 100%均衡，但也要相对均衡，这样才能给模型一个很好的机会，让它能够均衡地学习这两种知识。现在，选择哪种方法还取决于你已有的数据集大小。如果你有一个非常大的数据集，那么创建更多的数据点可能会导致计算成本过高；如果你已经有一个小数据集，那么通过降低采样率来使用它可能也不是很聪明，因为这样只会降低你训练模型的效果。下面是几个例子，说明这看起来像什么。你可以看到最左边是我们的原始数据。你可以看到红色和蓝色的两个类别。这就是第一类和第二类。在原始数据中，你可以看到蓝色的点比红色的点要少一些。在这个空间中，他们是一个非常独特的群体。因此，他们是我们关注的群体。我们希望能够预测第二类。现在有两种方法可以做到这一点。我们可以降低红色点的采样率。你可以在这里看到。现在，我们的总体点数减少了。红色和蓝色的点数大致相同。或者我们可以对蓝色点进行高采样。这就是下一张图片，你可以看到我们的红点数量和原来一样，但现在我们有了更多的蓝点。它们在这里有点重叠。这就是它们颜色变深的原因。还有一种混合方法，两者兼而有之。因此，我们既要上采样，也要下采样。我们在中间做了一些调整。其中一种可能的超采样方法比较流行。它的缩写是 "Smote"，即 "合成少数群体过采样技术"。你也可以直接叫它 "Smote"，因为听起来有点像 "指环王 "里的 "龙烟"，就像 "Smote"。我就是这么想象的。这是一种数据采样程序，属于混合方法的一种。因此，它可以根据你的能力和需要，进行向上采样和向下采样。因此，如果你使用了这种方法，就会发现你可以指定要做多少向上采样，多少向下采样。非常重要的是，你必须指定用于推算新病例的邻居数量。因此，向下取样很简单。基本上就是向上删除几个病例。抽样则比较麻烦，因为你要为稀有类别创建人工案例，但这些案例对该类别仍具有代表性和准确性。因此，我们通常会使用 k 和 n k 近邻之类的方法，即使用现有案例，然后创建与之相近或相似的新案例。因此，在这种情况下，例如，我们这里的新合成数据就是随机选择的数据点及其邻近点的预测结果的随机组合。这意味着你有了蓝色的点。因此，你知道这些蓝点通常具有什么样的参数。它们位于哪个空间。然后，你可以使用这些预测因子或蓝色案例的参数组合，创建更多的预测因子或参数。然后再创建几个与之相似且相邻的点。我们称其为 "邻居"，因为在散点图中，它们是相邻的点，因为它们非常接近。是的，Smote 显然也可以从多数类中的多数点中向下抽样。它是通过随机抽样的方式实现的。因此，我们只需随机选取几个点，基本上就可以删除它们了。好了，这就是我们对数据分割、抽样和不平衡处理的快速了解。对这部分还有问题吗？有的。合成新案例。就像增加数据点一样。没错预测不就是这样吗？基本上是这样没错我们将把凯恩作为技术之一。所以K和这里是用来创建这些新数据图的。所以，我们会在几堂课中讲到这是预测技术之一。是的。我们试图用数据来研究预测模型。我们正在预测数据。研究模型有什么意义？研究模型有什么意义所以我的意思是，我们基本上是在为部分数据建立模型。所以我们只针对蓝色部分建立模型。但如果你不在同一时间、同一地点、同一情况下为红色数据建立模型，那你就没有意义了。是啊。所以，这就是为什么先预测模型的一部分，然后再预测整个数据集。是啊，我想如果我们使用上部的技术。对 然后然后呢？是的，很明显。所以假设我们的数据是呃，这不是那么聪明。是啊，这是个好例子。所以你已经可以看到这里的数据点有一点重叠。所以，很多数据点并不是新数据点。它们只是在预测模型的相同空间中创建的几乎相同的数据点。这其实并不重要。所以问题就在于此。你本身并没有创造出新的信息。你只是在复制已有的信息，但对模型来说，这并不重要，因为它仍然把这些案例当作真实案例。这样一来，你就不会引入太多偏见，因为你并没有制造虚假信息。你只是在重现现有的、你知道是真实的信息。但你是对的。如果你的班级非常非常小，这就是个大问题。因为在这种情况下，你并不知道哪些人应该是离群值。例如，如果你只有五条记录，你不知道哪些是真实的，哪些是离群值。因此，你可能会不小心对离群类过度采样，这可能会使你的数据产生偏差。是啊。所以，正如你提到的，使用近邻法，有可能会出错。哦，是的。有没有什么办法能把误差降到最低呢？其实没有，因为你不知道数据实际应该在哪里。所以你唯一能做的就是利用正当理由、批判性思维和专家知识来思考，这是一个可能出现的现实数据集吗？但除了研究更多可能有或可能没有的数据外，你能做的其实并不多。还有一个问题是，近邻是否更适合对数据进行聚类。因为你看的是最近的点。所以，但它对预测也同样有效，因为这是同一件事，对吗？对所以数据点要弄清楚嗯，数据应该在哪里？基本上是的没错所以，这就是为什么我们谈论K和N，我认为在下一讲，这应该是第八周，因为转变。你说得没错，基本上就是这样。它通常用于分类目的，所以这是它的主要用途。我们之所以在这里使用它，是因为它在创建未知数方面非常强大，在现有数据的基础上，应该创建更多的数据。因此，我们在这里创建的这些额外的蓝色点，只是基于我们已知的邻居，我们会发现 K 和 N 之间存在一个很大的问题，实际上这就是所谓的 k。因此，这是另一种研究者选择的参数，你要把它添加到你的模型中。你所做的每一个选择，你所选择的每一个参数，都会给你的模型带来一点主观性。因此，根据你选择的参数，在创建新数据点时，你会考虑多少邻域也会影响你的模型？所以，这是个有点棘手的问题，但有时这也是唯一能让你真正获得这种情况下所需数据集的方法。是的，这是关于交叉验证的 k 值。那么我们如何选择不同的 k 值呢？没错所以又是英国。总是很棘手好吧，你应该做多少个折叠？折叠次数越多，就越精确。但计算成本也会更高。因此，根据你的数据集的大小，你可能只做三、五、十次。如果你的数据集较小，你可能会做 50 次、20 次、70 次。因此，并没有一个可以选择的规则。基本上，这取决于你的数据集，也取决于你的电脑有多强大。因此，如果你愿意，可以在实验室里进行测试，因为你可以选择三个折叠，看看准确率，然后做 20 个，看看准确率，然后测试你的电脑是否崩溃，你是否能让整个大楼的服务器瘫痪，我不知道。这样你就可以测试了。这其实不是一个固定的规则。这取决于你的应用类型。如果你的数据变化很大。因此，如果你的数据有很多细微差别，你会希望使用更多的数据，因为这样就能捕捉到所有的细微差别。但如果你的数据集很大，就得少用，因为太贵了。好的。获取。让我们来看看聚类。到目前为止，在这门课程中，我们真正讨论的都是有监督的方法。你应该还记得，在第一讲中，我们讲到了有监督学习和无监督学习的区别。因此，有监督的技术和我们迄今为止所学到的都是使用某种训练集。我们建立并调整一个模型。我们估算参数。例如你的击球手和回归模型。然后使用调整后的模型来预测某种结果、类别或数值。所以，你有某种输入数据。你有某种输出数据。然后使用算法来学习映射函数。例如，通过逻辑或线性回归等线性组合，X 与 Y 之间是如何连接的。如今，当我们谈论无监督技术时，它们不会使用单独的训练数据和测试数据。原因是我们没有输出 Y，我们也没有试图预测的任何类别或值。我们只有一个数据集。你可以把它想象成一个 X。我们正在寻找其中的结构。这是一个模糊的术语。我是一个非常视觉化的人，所以我总是从视觉上思考这个问题。所以我总是在想我的这种数据散点图。我对这些数据块、数据块和数据结构很感兴趣。数据在哪里聚集？哪里的数据更分散？散点图中的漏洞在哪里？数据在整个空间中是如何分布的。所以我们要寻找模式。这就是为什么无监督学习是机器学习领域中的模式识别。因此，我们要寻找数据中出现的模式，而不是真正试图预测任何类型的标签。所以你可能会问，好吧，这和预测建模有什么关系？因为我们在试图预测什么，不是吗？但有时，你实际上想要预测的是这种结构。所以，你试图了解群体，因为从长远来看，这能告诉你一些关于数据的信息，而这正是预测的真正意义所在。你还记得我们在讨论什么样的预测吗？它是利用信息对未来做出决策。因此，我们可以通过聚类来做到这一点。这里有几个不同的例子。我们主要用它来识别相似的数据点。例如，相似数据点就是数据中的肿块。因此，在客户细分中，聚类通常用于识别在某种程度上相似的客户群体。因此，公司今后应该以类似的方式对待他们。例如，你还可以用它来预测未来的行为。例如，如果你有客户细分，你就有不同的客户群体。而且，你知道，其中一个群体流失的几率很高，那么这就已经在预测他们未来的行为了。因此，你已经知道客户流失是什么样的。因为正是这个群体和这个群体的参数选择似乎决定了未来的流失行为。你也可以将其用于空间数据。例如，我在自己的研究中经常尝试在一个国家的不同地区找出在健康或财务状况、购买行为、购买行为等方面表现类似的亚人群。我这样做的目的是为一个国家的这些地区制定政策建议，因为我可以告诉政策制定者，比如这些地区未来可能会比其他地区遭遇更多的经济困难，或者其他人用它来绘制受洪水或野火等事件影响的地点地图。我们还可以将其用于时间序列，例如，我们可以研究不同股票的时间序列是如何表现类似的。因此，我们正在创建这些群组，这些特定种群的参考群组。然后，我们就可以用这种方法来预测某些股票的行为，因为它们与该参考组中的其他股票类似。因此，我们也可以用这种方法进行预测。让我们来谈谈聚类分析，因为这是无监督学习中的一个大话题。因此，它经常被视为无监督机器学习的同义词。这并不特别准确。还有其他一些无监督学习方法。严格来说，我们即将讨论的主成分分析也属于无监督机器学习。但大多数人说的无监督，指的是聚类。这是一个大问题。因此，我们今天将介绍两大类聚类分析算法。它们是划分方法。因此，我们将向你介绍 K 均值和 K 媒介。我们还将介绍分层聚类。这些都是分割和聚类方法。由于时间关系，我们今天不会介绍其他方法。这次我不会讲得太远。因此，我们不会谈论基于密度的方法、基于图论的方法和基于概率的方法。所有这些都非常令人兴奋。所以，如果你想跟我谈这些，请说吧，因为我对这些非常感兴趣。尤其是基于密度的方法非常有趣，因为它们再次使用了邻域的概念，这与我们之前提到的 k 和 n 想法非常相似。因此，它们也非常有趣。它们还有很多优点，因为它们能够识别离群值。因此，你会看到其他方法会看到今天的 K 表示层次方法，那么在做聚类时，它们确实能够识别离群值。例如，它们必须在进行预处理 DBscan 时被捕获。这种基于密度的方法能够在聚类时自动识别和标记异常值，非常方便。因此，它可以减少你的预处理时间。好的，我已经多次提到相似性、亲近度、邻居等概念。距离和相似性是聚类的关键。如果我们想识别点群、时间序列或区域，类似的任何东西。哪些是相似的。我们必须定义相似在我们的语境中是什么意思。因此，有些人使用相似性。有些人用距离。在这里，它们基本上是同义词。至于使用哪一个，则取决于你的应用环境。我通常会说相似性，除非我说的是空间上下文，那我就会说距离。因此，我们将讨论数字数据，特别是连续数字数据的相似性度量。你通常会遇到两大类。我希望大家都熟悉欧氏距离，因为这是你能想到的最基本的距离度量。举个例子，如果我们计算这里的两个向量之间的距离，你可以把它们想象成两列。你可以把它们想象成数据的两列，也可以把它们想象成数据的两行，这取决于你如何切换。它们基本上是两个观测值。比如两个人。你要根据描述这两个人的不同因素来确定他们的相似度。举例来说，如果我们有他们的收入，或者我们有他们的其他数值，他们是养老基金或类似的东西。因此，我们有几个数值来描述不同的人。我们想知道他们有多相似。基本上，我们要做的就是通过欧氏距离来创建这个不相似系数。基本上，我们只需要看看这些值中的每一个，它们的相似程度有多高？因此，我们用成员 x I 的精灵值减去成员 x j 的精灵值。我们将所有这些不同的因子相加，然后取其平方根。这就是这两个人之间的欧氏距离。这种测量方法还有其他非常相似的改良方法，比如曼哈顿距离，它不是取平方的平方根，而是取这两个人之间的绝对距离或绝对相似度，并根据不同的因素将其相加。这样，我们就有了所有不同的因素。现在，显然你不会只遇到数字数据。如果所有数据都是数值数据，那就太简单了。你可能还会遇到二进制数据。在这种情况下衡量差异的一种方法是看其中有多少因素是相同的，有多少因素对任何两个人来说都是不相同的。然后基本上计算一下，这两个人的类别中有多少重叠？二进制数据非常重要，因为请记住，如果你回想一下我们的虚拟编码理念，这是你的主要数据类型之一。如果你有分类数据，这在社会科学中非常常见。因此，如果你有所有的二进制数据，这就是你的异质性测量。因此，如果两个人的数据都是分类数据，那么你就可以计算出他们之间的所有重叠。这就是 N11，一个特定因素同时存在的数量。再除以 N11 加上 w 倍。这些基本上都是不相似的。因此，两个人之间的因素不相同或不重叠。这个 W 是一个系数，你可以将其设置为你所选择的任何值。这是一种加权系数。因此，这些非重叠非同时因素的重要性有多大。你想给它们多大的权重。共同权重就是一个。在这种情况下，我们称之为系数。所以，这也是你时不时会遇到的一个词。现在，我提到了分类数据。其实还有另一种方法可以处理这种数据。这就是匹配系数。它与我们刚才看到的距离非常相似。因此，我们仍然有这样一个值：如果他们在这个特定特征中不匹配，那么这个值就是 0；如果他们在那个特征中匹配，那么这个值就是 w。这就是我们没有做的细微改变。我们不计算 0 和 1。然后将它们相加，我们实际上是立即给每个非重叠的匹配特征赋予一个权重。一般来说，权重是 1，但也可以设置为任何权重。对于分类数据来说，这也是一种非常类似的测量方法。你会发现，有时我使用最大的 s，有时我使用较大的 d。例如，我用 d 表示汉明距离，用 d 表示距离或相似度。你会注意到，我是用 1 减去后面的某个因子或某个项。如果我去掉一个减号，只看后面的，那就是相似度。那么它们有多相似呢？如果我们把相似度减去 1，就得到了它们的不相似度。所以，我之所以有时用不相似度，有时用相似度，是因为教科书中对它们的定义是这样的。因此，匹配标准是专门用来测量相似度的。对于匹配标准之类的东西，其实并没有一个正式的定义。所以，我必须把它定义为不相似性。但如果你对这种相似性感兴趣，你可以用匹配标准减去一个。这就是你的不相似度。因此，你通常会遇到的不仅仅是分类数据或数字数据。你会遇到混合数据。在社会科学领域，这种情况极为常见。你会有一些你做过的调查。你会有收入之类的数字数据。你会有一些分类数据，比如教育程度。你必须找到这些人之间的差异，同时考虑到所有这些不同的维度。有不同的方法可以做到这一点。我过去的做法是，先专门为每个因素类别分别创建异质性度量或矩阵，然后再将它们组合起来。因此，有一些方法可以克服这个问题，将它们组合起来，然后使用组合矩阵进行聚类。但你也可以使用已经为混合数据创建的不相似度量。如果你读过任何关于聚类的文章，他们的名字就会跳出来。我不知道他们为职业生涯做了什么。真是不可思议。他们写了我写过和读过的所有关于聚类的书和论文。所以我的整个论文基本上都是关于这两个人的。真是不可思议。所以你会一次又一次地看到他们。他们以伽罗瓦距离为基础，而你可能对伽罗瓦距离并不陌生，他们在书中提出了以下观点。这和我们已经见过的东西很相似。因此，我们在这里有了相似度量。这是 L 特征中记录 I 和记录 J 的 s。这里还有一个系数，一个二进制系数 delta，用于定义该测量值是否缺失。这样就可以一次性汇总所有测量值。因此，其中每一个测量值是否为缺失的相似性，取决于该特定组合的数据是否可用。因此，他们为离散变量或二元变量创建了两种不同的相似性计算方法。在这里，他们基本上是在做一个简单的匹配系数。因此，要么它们是匹配的，那么系数就是 1；要么它们是不匹配的，那么系数就是 0。那么就是 0。对于连续变量和顺序变量，他们在这里提出了这种测量方法，即查看两个值的绝对差值，然后除以该变量值的范围。因此，这有点类似于你之前看到的曼哈顿距离。但它是按变量的范围缩放的。然后将每个人的数据相加，再除以系数的个数。记住，如果缺失，delta 值为 0；如果没有缺失，delta 值为 1。这样就得到了混合数据的总体相似度。如果用 1 减去数据的相似度，就得到了这两条记录 x x I 和 x j 的不相似度。你不必记住所有的公式。如果你现在感到不知所措，那么重要的是你要知道这些公式都存在。因此，在决定如何创建异质性矩阵之前，首先检查一下你有什么样的数据是非常重要的。我经常看到的一个常见错误是，人们直接把数据扔进 K-means，但你不能直接把数据扔进 K-means，因为 K-means专门使用欧氏距离，而欧氏距离只对连续变量有意义。所以，如果你有二进制数据，你可以把它扔进 K-means。你这没有任何意义所以说，针对不同类型的数据，采用不同的相似度测量方法非常重要。现在，我们还有一些规则要讲，但我觉得我们现在应该休息十分钟，下半场再讲，因为我知道你们有点累了。我需要喝杯水，休息一下，10分钟后再见吧好的，安静大家安静我们回来了还有很多事情要谈我还想借此机会提一下，你们的一位同事好心提醒了我一件事。你们可以为此感谢他。因为我突然又对我一直在做的一个项目感到兴奋。有人问我，聚类与逻辑回归等其他方法之间是否存在联系？怎样才能结合起来？这两个概念之间有什么联系？实际上，我已经将聚类与逻辑回归结合起来使用，我正在建立一个混合模型，首先使用聚类将数据分成若干组，然后为每一组建立单独的模型，使用逻辑回归预测特定结果。同样，你也可以使用不同的变量，你会发现这些细分中的每一个变量对于不同的模型构建都更为重要。例如，您可能会发现，如果您试图将客户流失行为作为二元结果进行预测，那么您可以首先将数据划分为不同的客户群，然后使用对客户及其行为重要的因素为这些客户群建立单独的逻辑回归模型，而不是在所有数据上画一张白纸。因此，有一些方法可以将这些模型结合到混合方法中，这非常有趣。因此，聚类不仅可以作为一种独立的工具，作为一种数据发现、模式识别的探索性工具，还可以作为一种发现信息的方法，这些信息可以在第二步输入预测模型。例如，你可以将分组标签作为模型的一部分。这可能也非常有趣。好的。快速绕行让我们来谈谈讲座中的相似性测量方法。通常，我们会假设我们的相似性度量满足几个条件。我在前面已经提到过其中一个条件，即相似度可以计算为 1 减去不相似度。因此，它们是一种互补关系。我们通常还会假设两个点 I 和 j 之间的相似度是对称的。这意味着从 I 到 j 的距离与从 j 到 I 的距离相同。我们还通常假设这些相似度和距离为正值，或至少大于或等于我们所有点的零值。在某些情况下，我们还需要满足更多的条件，如果这些条件也满足的话。除上述条件外，我们还会将我们的不相似度量称为不相似度量。因此，在这种情况下，举例来说，我们会有三角形不等式。如果我们有三个点，那么 I 和 j 之间的距离小于或等于从 A 到 k 的距离加上从 k 到 j 的距离。所以，如果你回想一下，如果你在本科阶段学过图论，这正是一个概念，如果你经过一个顶点，如果你经过连接顶点，如果你把这些连接的边相加，你就可以计算出从顶点一到顶点三的距离。是的，很明显，如果 I 和 j 在同一点上，那么两者之间的距离应该为零。我的意思是，这也说得通。如果他们是同一个人，那么他们之间就不存在差异。因此，作为一名研究人员，我对 "相似性 "这个概念非常感兴趣。因此，我在攻读博士学位时。这其实也是我反复问自己的问题之一，因为我一直在使用现有的异质性测量方法，你只需输入数据，就能得到某个数字，然后你就把它当作事实接受了。因此，你只需接受 A 和 B 之间的相似度相差 15 个点，而不会真正去思考 15.3 意味着什么？那么，这到底意味着什么呢？是否有一种方法可以解释这种差异呢？还有很重要的一点，这种差异是否会根据我们的环境而改变？我参与的一个项目使用不同的数据源来描述距离，即从世界不同地点到爱丁堡这座城市的感知距离。我们之所以提出这样的问题，是因为我们对科维德事件后旅游业的恢复很感兴趣。因此，我们想知道哪些国家的游客最有可能提前回到爱丁堡这个旅游胜地，这对公司来说非常重要，因为他们应该向这些地区投放广告，因为这些地区的人们希望尽快访问爱丁堡。因此，我们研究了这种距离的三个维度，其中一个相对简单，即游客的兴趣。我们通过查看谷歌趋势的长期数据来衡量这一点。另一方面，我们对游客的经济承受能力很感兴趣。因此，我们关注的是经济指标，即人们在这些国家的经济状况如何。他们有多愿意花更多的钱去休闲，比如去爱丁堡旅游？第三是旅行时间，以飞往爱丁堡的航班时间和航班连接来衡量。因此，如果你用一种特殊的方式来考虑距离问题，那就是一种非常不同的方式。与 200 年前相比，思考这些问题的方式也不同了。200 年前，如果你要从爱丁堡去伦敦，那就意味着要坐很长时间的长途汽车。非常不舒服，也不安全。你不会偶尔这样做。因为时间太长了。如今，你可以开车去，也可以坐火车去，还可以坐飞机去。所有这些不同的旅行方式都决定了你的旅行时间。因此，它决定了你在时间上的旅行距离。但它也取决于你到那个地方的感知距离。举个例子，你可以坐火车从爱丁堡到伦敦，大概需要五个小时，也可以坐飞机，可能需要一个小时。这看起来已经是很大的差别了。相差 40 个小时，对吧？但再想想，你必须提前两个小时到机场办理行李托运手续。这就是三个小时。你可能还得去机场。这可能又要花半个小时左右。然后，你可能还得去旅行。如果你必须走出飞机，你可能会坐在墙上。失败。伦敦希思罗机场简直就是噩梦。所以你还要在飞机上再坐半个小时，等待行李处理人员取走你的行李。等着等着，你的行李不见了，你就坐在了伦敦希思罗机场，而这里不是伦敦市区。因此，你必须从伦敦希思罗机场前往伦敦市中心。突然间，坐五个小时的火车就到了伦敦。休斯顿似乎不再那么漫长了，不是吗？因此，我们感知到的距离可能与以公里为单位的实际距离大相径庭。这就是我在论文中一直在思考的问题。我现在仍然对这个概念很感兴趣。因此，差异取决于环境。而差异应该由上下文决定。对于空间数据来说，如果距离很大，你可能需要考虑地球的曲率。突然之间，距离就不是欧几里得了。它不是一条直线。它可能遵循飞行路线，可能遵循汽车路线，也可能遵循地球曲率。这取决于你测量的是哪种距离。对于时间序列数据来说，这一点也非常重要。我刚才提到了我们的旅游项目。我们对谷歌趋势随时间推移测量的兴趣很感兴趣。因此，我们研究了这些类型的时间序列，并比较了不同国家的兴趣时间序列。那么，与从美国来爱丁堡旅游的兴趣相比，爱丁堡和比利时的兴趣随着时间的推移会发生怎样的变化？因此，具体到时间序列数据，有不同的异地城市衡量标准。因此，我们之前谈到的所有这些测量方法主要都是针对数据的，例如使用数字或分类数据来描述个人或地点。它并不真正考虑数据随时间变化的形状。因此，如果你有时间序列，你显然可以查看每个时间步长，然后计算每个时间值之间的欧氏距离。但这并没有真正考虑到时间系统中可能存在的偏移，可能存在某种扭曲的峰值，例如，比利时出现了一个峰值，两周后法国也出现了同样的峰值。如果你只是分别查看每个时间点，就无法捕捉到这种情况。实际上，你必须观察时间序列的形状，看看它们之间的距离有多远，相差有多大。举例来说，有一种方法叫做 "大流行时间扭曲"，这是一种专门针对时间序列的异同度测量方法，它考虑的是时间序列的形状，而不仅仅是每个时间点的数值。因此，空间数据和时间序列数据。我再说一遍。同样，它们都属于空间数据类型。这就是为什么处理起来比较棘手。观测数据之间存在依赖关系。而且，我们对这些数据还做了很多假设。因此，回到你的数据分析项目原则，例如，有人问我如何将其用作时间序列数据。我告诉你，从技术上讲，是的，我真的不在乎你使用什么类型的数据，因为我在乎的是你选择数据的方式是否能引起你的兴趣并能回答问题。话虽如此，但使用时间序列数据时一定要小心，因为它与你可能从调查中获得的数据不同，你要看的是一项横截面调查，对大卫-邓恩（David Dunn）的观察结果，一次大约有一千人。就是这样。因此，调查之一与长期收集的时间序列截然不同。现在，队列数据、纵向数据等所有这些数据类型显然都有重叠之处。因此，在处理不同数据类型时要小心谨慎。现在我们来看看几种不同的聚类算法。现在，我已经用这些额外的步骤和注意事项把你 "骗 "死了。毕竟这并不容易，不是吗？最常用的聚类算法之一是 K-means。K-means 的原理相对简单。你把数据分成 K 组。这又是我们的 K。所以我们必须选择另一个参数。然后将其作为一种分区算法，这基本上意味着它可以分区。它将你的数据空间划分为若干组。因此，有一种模糊方法，你不会将这些组完全切割，而是允许成员度，而不是决定第一组、第二组、第三组，你可能会说，哦，这个人 50%是第一组，20%是第二组，30%是第三组。这就是模糊聚类的成员度。不过，让我们把注意力集中在切割分区部分本身。康明斯公司的运作方式非常有趣，因为我们是随机开始的。我们只需拥有数据集，选择 k 个不同的观测点，然后就有了我们的聚类中心点，也就是我们每个组的中心点。然后，我们使用欧氏距离将每个观测值分配到最接近的群组或中心点。这就是距离测量的原理。然后，我们取新的中间点，即刚刚分配给它的所有观测值的新平均值，重新计算中心点。然后重复第二和第三步。因此，我们会将每个观测值分配到最新的新中心点。这可能有变化，也可能没有变化。然后我们重新计算新的中心点作为新的中间点。重复这个过程，直到中心点不再移动。这时我们就可以说，我们的解决方案已经稳定，我们最终得到了我们的聚类，我们的 k 个聚类。现在听起来有点奇怪，因为这里面有很多随机分配之类的东西。所以，如果我把它画出来，看起来会是这样的。所以，我们会有某种数据空间，我们会有点在这里的某个地方。让我们在这里和这里各画一点。所以，我试图创建一个空间，让你看到自然形成的集群。好的，我希望你能看到这里可能有四个集群和三个点。我不知道我在做什么。所以 K 表示的意思基本上就是设置，比方说，由于某种原因，我们知道是四个点。那么，莱恩，来吧，让我画画。这里画一个，这里画一个。这里一个然后就是这里K -means会把这些点分配到最接近的组里你可以看到我随机设置比如说，这些点就会被分配到这一组。这是最近的这个可能也是最接近的。所以它们会开始寻找所有这些点的距离。然后这些可能就属于这里了。等等等等。因此，我们现在开始将每一组、每一个点分配到其最近的点上。你可以看到，这可能并不是一个完美的解决方案。因此，这里有几个点我们认为可能属于这里。但它们被分割开了，所以这里会有一些奇怪的移动。K-means 的好处在于，随着时间的推移，这些点会移动，这取决于你选择的初始聚类有多好。所以，随着时间的推移。随着时间的推移，如果我们有这里和这里，让我们试着复制一下。如果我们现在计算例如，我们知道左上角的这些点被分配到了同一个聚类的中心点。因此，这些群组现在被分配到群组标签一。然后我们计算所有被分配到同一中心点的点的平均值。因此，如果我们计算平均值，新的中心点可能会向上偏移一点。所以它现在可能在这里。原来在这里。因为我们在计算平均值，所以它现在会向上稍微偏左一点。它的位置是所有这些点的平均值。它的位置是所有这些点的平均值。所以我们不会把它画得太靠右。就说它现在在这里。然后我们重复这个过程。你已经可以看到，这个中心点的绘制现在变得容易多了。所有这些点都有可能被分配到中心点上。因此，我们已经在这里找到了一些解决方案。我们对所有这四个点进行同样的处理。因此，我们会计算每个中心点的新位置，将其作为最初分配给这些点的平均值。通过反复计算，我们会得出某种解决方案，希望每一组都是自己的聚类。它们的中心点位于中间。我说的是希望。原因是随机选择初始中心点可能是 K 均值法的弊端。它的计算成本很低，但也意味着这取决于这些初始点到底是在哪里选择的，或者是在哪里设置的，在哪里可能会找到一个非最优解。例如，如果你如果你选择的点的方式导致 K-means最终倾向于分裂出非常大的群组，举个例子。因此，我们有时会有一些非常大的聚类，而这里可能会有一些小的聚类。现在，我们希望看到的最优解就是最上面的那个是一个大的群组。然而，在很多情况下，K-means 往往会将它们分割开来。因此，它倾向于寻找这些小的球形聚类。因此，它会将其分成两个不同的组。根据你的应用环境，这可能是好事，也可能不是。因此，在某些情况下，我们希望在不同的组中找到小的集群。而在其他情况下，如果真的存在较大的群组，我们实际上也希望能用不同的方法来定义这些群组。因此，如果这一切看起来有点复杂，K-means 可能会有点棘手。我建议你去看看。基本上，在 YouTube 上输入 K-means 算法，就会出现很多精美的 GIF 小视频，这些视频展示了这些点是如何移动的，以及它是如何收敛到一个解决方案的，希望这个解决方案是最优的。但由于初始点的随机选择，可能并不是最优的。所以，回到 "好吧"，因为这才是真正决定我们能找到什么样的解决方案的因素。早些时候，我谈到了 K-means 希望分裂的这种解决方案。如果你选择的 K 值是 4，它肯定会分割。如果你选择的 K 值是 3，那么它可能会分割，也可能不会，这取决于初始中心点。例如，它可能会尝试将这一半与底部合并，将这一半与底部合并。我见过用 k 来求解的奇怪方法。那么，我们该如何选择最优的 k 值呢？早些时候？我刚刚告诉你了很明显在某些情况下，就是这么简单。在某些情况下，你只需将数据绘制成散点图，然后说，嘿，这很明显是为了大多数情况下，比如99%的情况下，这是不可能发生的，因为你的数据不会那么容易划分，也不会那么漂亮，尤其是你的数据不会是二维的。一旦数据超过两个维度，可视化就会变得非常棘手。因此，我们有几种不同的方法来选择 k，其中最常用的有三种：肘部标准、剪影测量和差距统计。肘部标准法是一种非常主观的基于情节的方法。它基本上是将解释变异作为一种方法或聚类数量的函数来看待，然后选择它认为的最佳聚类数量，即变异改善不大的聚类数量。所以，你还记得我们之前提到的 "解释变异 "的概念吧。当我们讨论线性回归和逻辑回归时，它又回来了。因此，这种方法非常主观。但在某些情况下，k 的选择是非常明确和简单的。我之前说到过这样的四幅图。一目了然。所以，如果你有这样的东西，肘部标准完全可以选择，你可以想知道这看起来像什么，为什么我们称之为肘部方法。原因就在于这幅图。如果这里有 k 的数量，例如，有解释变异，我们就来解释变异。有不同的方法。不同的方法。可以是 Y 轴上的解释变异，也可以是集群内部的方差和。那么，聚类的多样性或规模有多大，有多奇怪呢？我们来看看解释变异。然后是肘图威尔就像这样它会一直上升 一直上升 一直上升然后在某个点会停下来然后相对变直并不完美这样才完美不会像那样，但相对完美。然后你会选择K 等于三，因为三之后，你的解释变异就不会有更多的改善了。我们称其为肘部，因为它是曲线的肘部。这就是改变方向的原因。现在这个非常漂亮。可能看起来不是这样的。它可能会更加平滑，这就增加了选择的难度。这也很主观。所以有些人可能会说，好吧，好吧。但实际情况是，这样的改进有多大。这并没有多大改善。这真的值得吗？难道我们不应该使用案例二吗？你可以这么做，因为这是主观的。所以在这种情况下，没有人阻止你选择 K2。在某些情况下，这可能是个正确的选择，因为我不知道你的应用案例，使用案例二可能更有意义，因为你有两个广告活动要运行。所以，你需要两组人，你不需要 V，因为你不知道该如何处理第三组人。因此，在这种情况下，你会选择 k 为 2，尽管肘部图可能会有不同的说法。我们刚才说过，肘部曲线图非常主观，因为我们只是在寻找曲线的这种变化。我们还可以使用更客观的测量方法。严格来说，剪影测量是对聚类解决方案质量的一种测量。因此，这是你想在事后看看你的解决方案到底有多好的东西。不过，你也可以通过将其绘制为聚类数量 k 的函数来判断哪一个聚类的结果最好。因此，它基本上可以有多相似。数据点与同一聚类中的点有多相似，与其他聚类中的点又有多不相似。这就是聚类的意义所在。你想找到与同组其他点尽可能相似的点，以及与其他组中的点尽可能不同的点。因此，相似或不同又是由你的不相似度量来定义的。例如欧氏距离，这取决于你的数据。这是一种更客观的测量方法。但它的计算难度更大。因此，这就涉及到更多的步骤，比如，我只是绘制出这个曲线，然后在曲线中寻找这个小点，然后我就很高兴了。因此，我们突然需要做更多的计算。具体来说，我们想知道聚类内部的相似度。我们称之为簇内相似度或簇内相似度。例如，在这种情况下，我们有两个点 I 和 j，然后是同一个聚类 C，我们有聚类 C 的大小，即该聚类中点的数量。我们想知道同一聚类中每个点之间的相似度。因此，我们要计算相似度，可以是欧氏距离，也可以是任何一种测量方法。然后，我们将同一聚类中所有点的距离相加。这就是聚类内的平方和。也可以这么叫。然后除以聚类中的点数减一再减一。因为我们总是在计算每个点的平方和。比如这里的 I，然后做同样的计算。事情是这样的对于点 I 而言，不过是在聚类之间。那么，点 I 与它所在簇之外的所有点的距离是多少。因此，再次计算所有的距离，将所有的距离相加，我们称之为簇间相似度或英特尔簇间相似度。不要与簇内相似度混淆。这就是为什么我更喜欢用 "群内相似度"，因为我的发音有时会把这两个词都吞掉。计算方法是一样的。我们想知道从 A 点到 j 点的所有距离。我们用这个距离除以另一个聚类中的点数。最重要的是，我们要寻找最小值。为什么要寻找最小值呢？因为我们只对邻近的聚类感兴趣。我们并不太关心很远的其他聚类。我们想知道的是，在我们的数据子空间中，我们与相邻群组的区别有多大？那么，我们与邻近集群的区别有多大？AI 数据点的剪影计算方法是：AI 点在聚类间的距离减去 AI 点在自己聚类内的距离，然后除以这两个项的最大值。这就是剪影测量或剪影系数。给你又叫考夫曼了他们又出现了，就是所有这些剪影测量值的最大值。所以你要计算每个数据点的剪影 然后取其最大值 最大平均值嗯这就是整个数据空间的总体剪影系数。因此，我总能看到这在计算上是比较昂贵的。数据量越大，计算成本就越高，因为你必须对每个数据点都进行计算。但这是一个非常稳健的测量方法，因为它基本上考察的是聚类之间的区别和分离程度。聚类分离得越好，你的解决方案就越好。因此，剪影系数是一个单一的值，你可以用它来选择特定的 k 值，即计算你能想象到的每个可能的 k 值的剪影系数，然后对它们进行比较。基本上，你要寻找的是集群内部和集群之间的不相似性。正如我所提到的，K 会提供每种可能的解决方案。现在你可能会想，好吧，这意味着我需要一个范围。这个范围是多少？实际上没人能告诉你，因为你必须自己去尝试。这取决于你的数据。因此，并没有规定你应该检查多少个聚类，因为你的数据越多样，你能找到的可能聚类就越多。好的经验法则是，以你的应用案例实际能够处理的数量为准。如果你有一家公司，我想对他们的客户进行聚类聚类，你告诉他们我划分了 364 个不同的客户群，他们会让你滚出去再做一次，因为他们无法处理 364 个不同的客户群。营销部门会把手举到空中。我们不喜欢这样。所以他们同时想要一个更小的数字。如果你有两个小数字，你就不能真正捕捉到模式。因此，你必须在两者之间找到一个切合实际的数字，这个数字既要大到足以捕捉到一切，又要小到足以处理解决方案。因此，我们的第三个衡量标准就是差距统计量。差距统计量非常相似。它与轮廓统计非常相似。它仍然关注的是聚类内部。也就是聚类内部的差异性。但现在我们将其与提议的基线分布进行比较。因此，我们要寻找的是聚类解决方案与假想空间之间的差距，在假想空间中，你的数据不会聚类，因为它在整个空间中是均匀的。因此，举例来说，如果你有一个二维的想法，你有一个非常好的、独特的数据点分组，即每四个角，那么你可以将它与同样的数据在该空间中均匀分布的情况进行比较。聚类解法与我们这种均匀分布这些点的解法之间的差别有多大？那么，我们的聚类解法与随机数据的差距有多大？我们真实数据上的两小时聚类解，它描述了这些聚类与均匀分布相比有多大的差异。好的，这就是我们用来选择的三个主要指标。好的。我们来谈谈 K 媒体。因为我之前抱怨过人们只用 K 平均值，但你不能这么做。所以，K 媒体就是这样的东西。它是我之前提到的那个问题的解决方案。K-means 使用的是欧氏距离。欧氏距离只适用于数字数据。因此，如果你遇到混合数据的情况，或者你有分类数据、二进制数据和数值数据等所有数据，你就可以使用 K meteoroids。你已经知道 K meteoroids 是谁发明的了，因为是同一批人。还是我们的好朋友加利福尼亚，他们发明了 K meteoroids 算法。潘所以它和 K-means 算法的原理基本相同。你仍然需要提前选择 K。你还是要根据数据点的相似度来划分数据空间。但在这种情况下，你定义相似性的方式发生了变化。例如，你可以使用 Jaccard 相似性指数。你应该记得，这是之前用于二进制数据的指数。你也可以使用混合数据类型。我们之前谈到的所有这些相似性度量方法，都可以用在 k 陨石上。我自己就使用 k 陨石。当我在进行测量时，我之前谈到的分析有关于客户对旅游业兴趣的时间序列数据。我有关于距离的飞行数据。因此，我有这些不同的数据类型、不同的异质性概念。因此，我将它们合并为一个异质性矩阵。由于没有使用欧氏距离，我无法使用 K 均值法。所以我选择了 k 陨石，因为它允许我自行定义我的异同矩阵，并将其作为输入输入到算法中。因此，如果你选择了 K-陨石，你就可以看到，你不必只定义数据。你只需给它一个自己定义的不同度量的相似性矩阵，比如我们之前讨论过的那些。因此，我之前已经谈到了聚类评估，例如，使用剪影测量，这就是你的一种评估指标。但我们也谈到了聚类的真正目的，即我们希望找到最相似数据点的聚类。因此，我们希望将聚类内部的不相似度降到最低。因此，我们要尽量减少集群内的不相似度，即各数据点在各自集群内的不相似度。我们要将它们与其他数据点分开，从而最大限度地提高聚类间的相似度。还有其他需要注意的事项，我已经在整个讲座中讲过了。这些结果对我们的上下文到底有多大用处？我们的结果或方法的计算成本有多高？聚类并不能真正预测结果。因此，评估结果非常棘手，因为这是一种探索性的解决方案。你不能说这个方案是最优的，因为你不知道聚类的成员。这是你创造出来的概念，所以你无法真正测试你对每个数据点的分类是否与分类情况下的方法一样，因为你不知道真正的类标签，你创造的类标签基本上是白费的。因此，评估聚类解决方案是一门学问。这非常棘手。通常情况下，我们要看聚类的分离程度，以及它们对我们的解决方案有多大用处。好了，让我们用十分钟左右的时间来讨论分层聚类。这与我之前说的 K-means 是不同的方法。K-means 是一种划分方法。也就是说，我们将空间划分为若干组。就是这样。这是一种单向的方法。因此，我们对每个成员进行划分。现在每个数据点都有一个群组成员。就这样就是这样。我的聚类工作方式有点不同。我们并不是只划分一次，而是随着时间的推移，创建一个可能的聚类解决方案的层次结构。这是一个迭代的过程，我们不断前进，分割更多或合并更多。因此，我们要做的不是分割，而是决定要创建多少个聚类。所以，我们要决定在哪里砍树，而不是划分，也不是选择你的 K，你会明白这到底意味着什么。我想这样做。因为这是一种迭代工作。分层过程有两种方法可以做到这一点，一种是自上而下的过程，一种是自下而上的过程。决策树也是如此。因此，我们现在采用的是自上而下的流程，与之非常相似。这也叫分层聚类。我认为最流行的算法是戴安娜算法。因此，在这种情况下，所有的观察结果都从一组开始。所以是所有的点。想想你的二维空间。每个人都在同一个群组里。第一簇或零簇，随你怎么做。然后随着时间的推移，我们会把这个空间分成越来越多、越来越小的群组。所以我们做了一次切割。我们有两个簇，再做一次切割。现在我们有了三个簇，再做一次切割。我们有四个有了我们继续每一步都把空间越分越多，直到最后每个数据点都有自己的簇。所以，我们从每个人都在一个簇里，一直到每个人都在自己的簇里。自下而上的过程与此完全相反。每个数据点都在自己的聚类中，然后我们开始一步步合并那些最接近的数据点。因此，我们先合并两个，然后再合并一个，再合并一个。我们开始构建一个解决方案，直到最后所有数据点都在同一个聚类中。所以，过程是一样的，只是翻转了一下。我认为最流行的聚类算法是 Agnes。因此，在这一领域，我们有戴安娜和艾格尼丝两种算法。所以，当我谈到这个切割的想法时，我们的意思是我们在左上方得到这种树，我们称之为树枝图。Addenbrooke 描述了随着时间的推移，解决方案是如何合并的。因此，你在这里看到的每一片叶子都是我们九个观测点中的一个。在右侧的二维数据空间中，你也可以看到这九个观测点的样子。随着我们在树上移动，树叶也在不断融合。一开始，这九个点中的每一片叶子都是它们自己的聚类。但后来我们发现，嘿，5 和 7，1 和 6。它们真的很接近。所以我们可以看到这些点 1和6 5和7它们非常接近。我们合并它们。我们在第一步就把它们合并了，这样我们就能看到一个聚类了。然后我们可以看到这个 5-7 聚类非常接近 0.8。我们在右边也能看到。5-7 聚类与 8 聚类很接近。所以我们也开始合并这个聚类。下一步，我们可以看到我们的 1-6 聚类接近于 4。所以我们合并它，然后它又接近 3。继续合并。你可以看到 0.9 是一个离群值。对它离我们最远。所以 0.9 是最后一个被合并到最终解决方案中的。所以我们会随着时间的推移合并这些数据。融合的时间越早，观测结果就越接近。所以你可以看到5号、7号、6号和1号。它们是最接近的所以它们被合并了最早的九号离其他人最远这就是为什么我们把它们合并到最近的位置所以，""的概念。我们提到过这是聚类的一个关键概念。我们将在这里再次讨论这个问题，因为我们必须确定这意味着什么。我们什么时候合并分组？我们什么时候决定要合并？如果我们有多个点，或者在聚类中有多个点，那就更是如此了。举例来说，我们把 5 和 7 合为一组。我们该如何判断 8 是否与它们相近呢？这有不同的思考方法。你可以计算最大的聚类间最大相似度，也可以计算最小的聚类间最小相似度。你可以求最小聚类间相似度、平均值或中心点。那么这意味着什么呢？基本上，你可以看一下，如果这里有第八个聚类，这里有第五个第七个聚类。它们之间最接近的距离是多少。那就是 5 和 8。它们之间最远的距离是多少？7和8。中等距离是多少？那就是八5和7之间的中间点是什么？所以，你可以看到，如果我看最大距离，我们将有完全的联系。最小聚类肯定有最小距离，我们有单一联系、平均值，这将是一种平均联系或中心点，这是中心点之间的差异，平均值和中心点非常相似。我们之所以将二者区分开来，只是因为我们计算平均类间相似度的方法不同。这就是整个聚类之间的相似度的平均值与两个中心点之间的相似度的比较。所以这是一个总和，我们将所有的距离相加，然后计算出平均值。这只是两个中心点之间的差异。在很多情况下，它们几乎是一样的。但这取决于数据在聚类中的分布情况，以及它们是否正好是平均值。因此，你也会看到，取决于你选择哪一个，这将影响所形成的聚类的形状。因此，在聚类差异最大的情况下，我们在决定是否要将它们连接起来时，实际上是在寻找可能的最远点。因此，这些聚类通常会形成非常明显的小聚类，彼此相距甚远。集群间的最小相似性更容易实现。因此，合并的可能性要大得多，因为如果有两个点彼此接近，那就足够了，即使同一组中的其他点距离很远。因此，在这种情况下，我们很容易就能建立起非常细长、非常大的聚类。然后，平均法、中心点法在两者之间取得了平衡。所以它们是最平衡的选择。这是一种保守的选择。比较随和的选择，然后是介于两者之间的平均值。是的。最后，我们来谈谈如何处理血管造影。因为这是一棵树，并不是只有一种解决方案。它是一组嵌套的可能解决方案，你可能想选择这些解决方案。因此，在某些情况下，你可以选择保留整棵树。在某些情况下，你所关心的只是描述数据中的结构和模式。在这种情况下，你就可以保留整棵树，并将其作为结果呈现出来，因为它描述的是数据中每个点的结构和相似性。因此，如果你的目标是探索，你可以保留整棵树。在某些情况下，你希望在某一点上把树作为一个点来切割。因此，我们在这里看到的就是这种图形。你可能会在某一点上选择它。这是你想要接受的最大相似度。举例来说，如果你选择 1.0，你就会在这里剪切。然后，你会将 2、9 和 3 保留在各自独立的聚类中，只接受 164 和 5、7、8 这些点的合并。因此，这也是一种剪枝的方法。到了一定程度，你就会决定，好吧，这就是我想在解决方案中使用相似度的极限。所以通常情况下，当合并或分割对相似度没有进一步的显著影响时，我们就想砍掉。因此，这又回到了这个思考过程，或者说是一种剪影测量，我们在这里看的是 "好吧"。什么时候我们的相似度会稍微均衡一点。这就是我们要切割的解决方案。这就是我们最终接受的聚类解决方案。不过，具体在哪里切分，还是有点主观。因此，聚类本身就是一种非常主观的方法，因为它没有对错之分。它是探索性的。因此，我们要探索数据，从中找出可能的模式。我们来讨论一下。也许下次我们可以简单讨论一下聚类，因为我觉得今天的时间安排得非常好。我在一开始就提到下周是阅读周。因此，计算机实验室将没有讲座。利用这段时间抓紧时间阅读。本周我们将进行计算机实验，特别是交叉验证。如果你们还有其他关于聚类的问题，我还会在这里待十分钟左右，除非有人想把我赶出教室。好的，明天见。