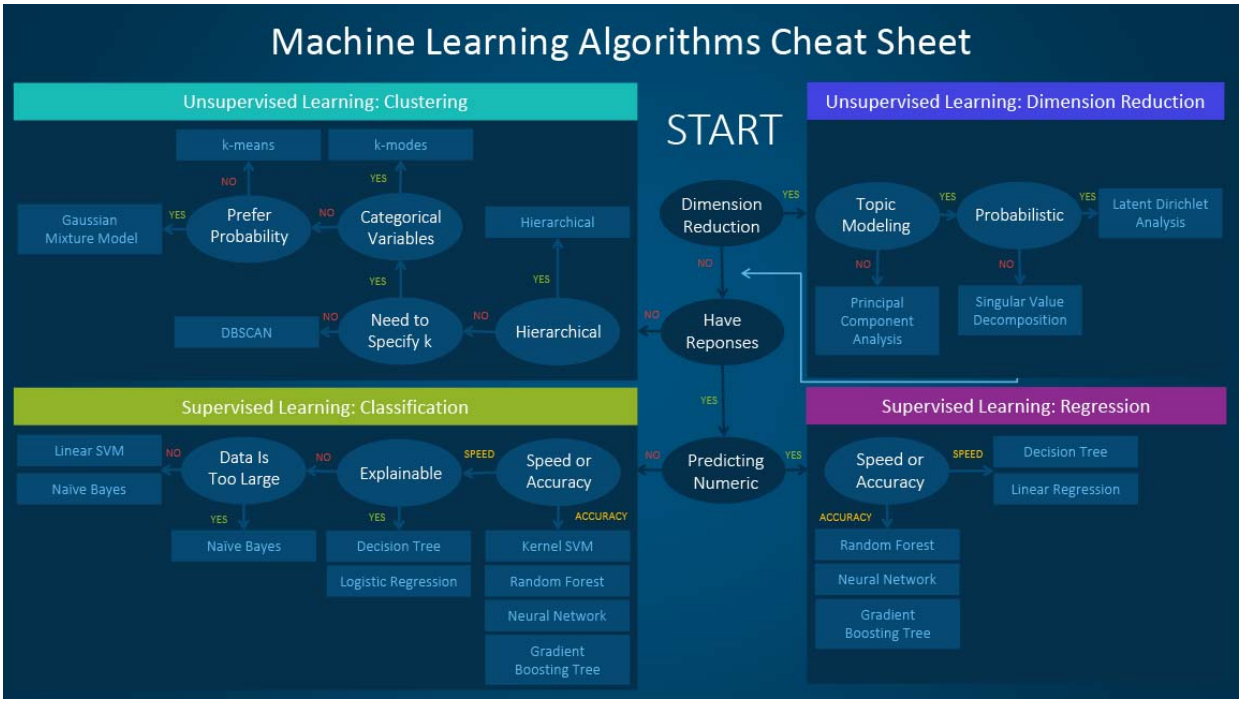
在实际项目中，如何选择合适的机器学习模型？

在这个文章中，我们主要面向初学者或中级数据分析师，他们对识别和应用机器学习算法都非常感兴趣，但是初学者在面对各种机器学习算法时，都会遇到一个问题是“在实际项目中，我到底应该使用哪种算法呢？”。这个问题的答案取决于许多的因素，其中包括：

1. 数据的维度大小，数据的质量和数据的特征属性；
2. 你可以利用的计算资源；
3. 你所在的项目组对该项目的时间预计；
4. 你手上的数据能应用在哪些项目中；

即使是一位经验丰富的数据科学家，在没有对数据尝试很多种不同的算法之前，他也不能确定哪一种算法在数据上面有更好的表现。但是，我们并不主张这种方式，一个一个算法去试验。我们希望自己有一点先验知识，可以指导我们去如何选择算法模型，帮助我们少走一点弯路。

机器学习算法表



上面的机器学习算法表可以帮助我们如何去选择一个合适的机器学习算法，对于我们特定的项目问题。这篇文章，我们主要来讲讲如何去使用这个表格。

因为这个表格是为初学者所设定的，所以我们在讨论这些算法的时候，会做一些简化的假设工作。

这里所推荐的机器学习算法是由几位数据科学家，机器学习专家和算法开发人员所共同反馈总结的。随着后续的发展，我们会收集更加全的算法来更新这张表。

如何使用这种算法表

其实阅读这种算法表非常简单，我们可以采取 如果你想要进行 ，那么你可以使用 这种模式来读取。比如：

- 如果你想要进行降维操作，那么你可以使用主成分分析方法（**PCA**）；
- 如果你想要快速进行手写数字预测，那么你可以使用决策树或者逻辑回归；
- 如果你想要进行数据分层操作，那么你可以使用分层聚类。

有时候，我们可能会有很多的条件需要去匹配算法，但有时候可能我们连一条总结的规则都没有，以至于不能去利用这个算法表。其实，这是很正常的，因为这个算法表是我们凭借工程师的经验总结处理的，因此有一些规则并不是很准确。我和几个好朋友一起讨论过这个问题，我们一直觉得寻找最好的算法的唯一路径可能就是去尝遍所有的算法。但是这种方法非常“蠢”。

机器学习类型

这部分我们会介绍一些最流行的机器学习模型类型。如果你对这些类别比较熟悉，那么对你以后去选择机器学习模型是非常有利的。

监督学习

监督学习算法是基于一组标记数据进行预测的。比如，历史销售数据可以来预测未来的销售价格。应用监督学习算法，我们需要一个包含标签的训练数据集。我们可以使用这个训练数据集去训练我们的模型，从而得到一个从输入数据到输出期望数据之间的映射函数。这个模型的推断作用是从一个数据集中学习出一种模式，可以让这个模型适应新的数据，也就是说去预测一些没有看到过的数据。

- 分类：当数据被用于预测一个分类时，监督学习算法也可以称为是一种分类算法。比如，我们的一张图片可以被分类标记为狗或者猫。如果我们的分类标签只有两个类别，那么我们也把这个分类称之为二分类问题。当我们需要分类的东西超过两个类别的时候，这个模型就是一个多分类模型了。
- 回归：当我们预测的值是一个连续值时，这个问题就变成了一个回归问题。
- 预测：这是根据过去和现在的一些历史数据，来预测将来的数据。最常用的一个领域就是趋势分析。比如，我们可以根现在和过去几年的销售额来预测下一年的销售额。

半监督学习

监督学习带来的最大挑战是标注数据，这是一项非常耗时的工程而且非常昂贵。那么如果标签的数量有限，我们应该怎么办呢？我们可以使用一些非标记的数据来加强监督学习。由于在这种情况下我们的机器学习算法不是完全的监督学习，所有我们把该算法称之为半监督学习算法。在半监督学习中，我们可以使用未标记的数据和一小部分的标记数据来训练我们的模型，从而来提高我们模型的准确性。

无监督学习

在使用无监督学习的时候，我们所使用的数据都是不用进行标记的。我们的算法模型会自动的去发现数据内在的一些模式，比如聚类结构，层次结构，稀疏树和图等等。

- 聚类：将一组数据进行分组，使得一个组里面的数据跟别的组里面的数据是有一定的区别，也就是说每一个组即使一个聚类。这种方法经常被用来做数据切分，也就是把一个大的数据集先切割成几个小的数据集，而每一个小的数据集都是一个高度相似的数据集。这样可以帮助分析者从中更好的找到数据之间的内部结构。

- 降维：减少数据变量中的维度。在很多的应用中，原始数据都是非常高维度的特征，但是这些维度中很多的特征都是多余的，或者说跟任务的没有相关性。降低维度可以帮助我们更好的而发现真实数据之间潜在的内部关系。

强化学习

强化学习是根据环境对智能体（agent）的反馈来分析和优化智能体的行为。智能体根据不同的场景会去尝试不同的动作，然后分析不同动作所会带来什么的回报，选取其中最大回报作为所采取的最终动作。反复试错和奖励机制是强化学习和别的算法最不同的地方。

那么如何选择这些类别的算法呢？

当我们去选择一个算法的时候，总是会考虑到很多的方面，比如：模型准确率，训练时间，可扩展性等等。这其中，最重要的可能就是准确率，但是对于初学者而言，可能最重要的是他们的熟悉程度。如果他们对一个模型很熟悉，那么第一个尝试的往往就是这个模型。

当给定一个数据集的时候，我们首先想到的应该是如何快速的得到一个结果，也就是我们常说的 demo 算法。在这个过程中，我们首先关心的并不是算法结果的好坏，而是一整个算法在数据上面运行的流程。初学者更加倾向于去选择一些容易实现的算法，并且可以快速得到结果。这样的工作节奏是非常好的，一旦你获得了一些结果并且熟悉了数据，你可能就会愿意花更多的时候去使用更加复杂的算法来理解这些数据，从而获得更好的结果。

即使我们到了这个阶段，最好的算法可能也不是那个获得最高准确率的算法，因为对于一个算法我们需要仔细的去调整参数和长时间训练才能得到一个算法模型的最佳性能。而上面我们只是去简单的运行了一下模型，得到一个结果而已。

选择算法时的注意事项

正确率

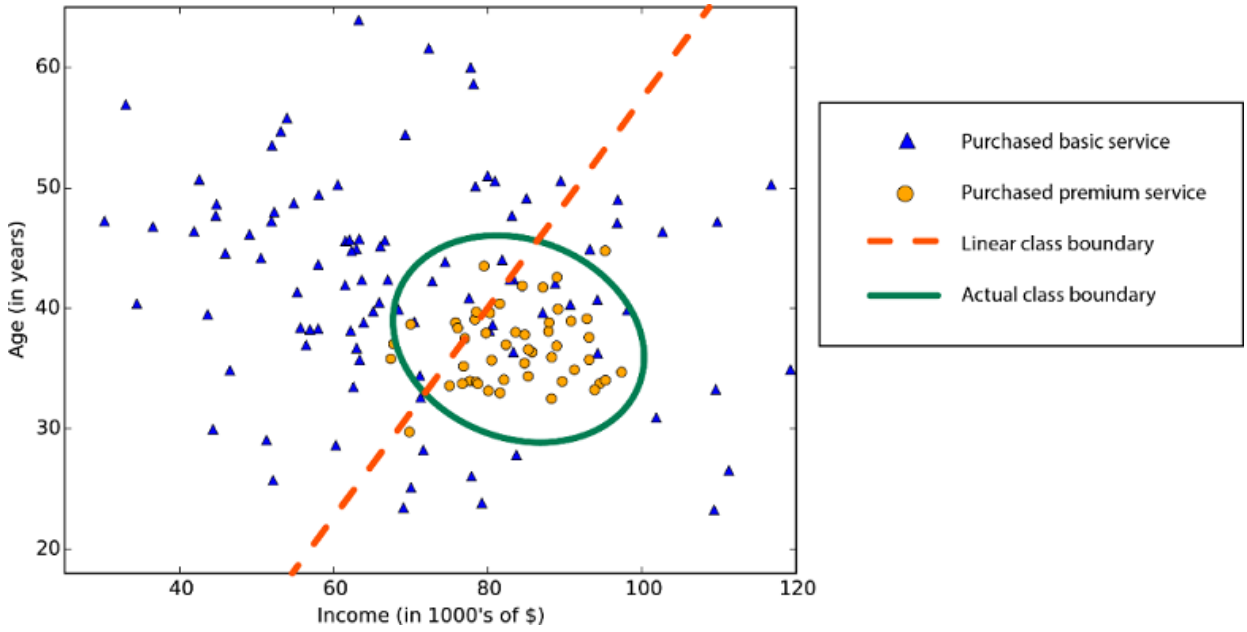
获得最准确的答案可能不总是最必要的。有时一个近似答案也是足够了，当然这取决于你想要如何去使用你自己的算法模型。如果是这种情况，你可以采用一个近似的方法来缩短你构建模型的时间。这种近似的处理方式还有另一个优点，就是可以帮助我们一定程度上面避免过拟合。

训练时间

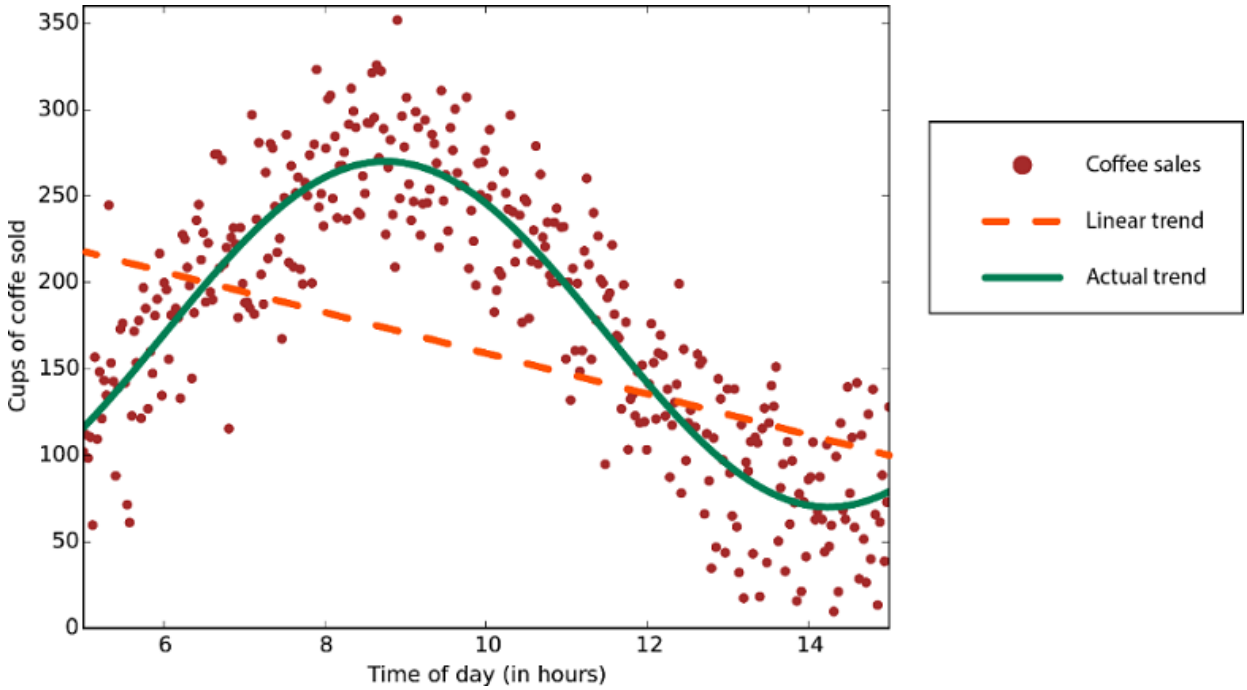
训练模型所需要的时间在不同算法之间是变化很大的，有些算法可能几分钟就可以训练完成，有些算法可能需要几个小时才能训练完成。训练时间往往与模型准确率是密切相关的，简单的说，可能训练时间越长，模型的准确率就越高。另外，有些算法可能对数值离散点数据更加敏感，而有些可能对连续数据更加敏感。如果我们的数据集非常大，而且时间非常紧，那么根据模型的训练时间来选择算法是一条非常好的路径。

线性

很多的机器学习算法是可以利用线性模型来解决的。线性分类算法假设数据是可以利用一条直线来进行分裂的。线性回归模型假设数据遵循一条直线划分，这些假设对于一些数据分析并不是一个很坏的假设，但是在某些方面，这些假设可能就会降低很多的准确率。



对于一些非线性边界 —— 依赖于线性分类模型就会降低很多的精度了。



有些数据可能无法简单的判断数据是线性的还是非线性的，但是在实际项目中很多的数据都会有一种非线性趋势，这也是我们使用线性回归方法产生比较大的误差的一个原因。尽管线性模型存在很多的不好方面，但是他往往是最简单的算法，我们可以进行快速开发和试错。

模型参数

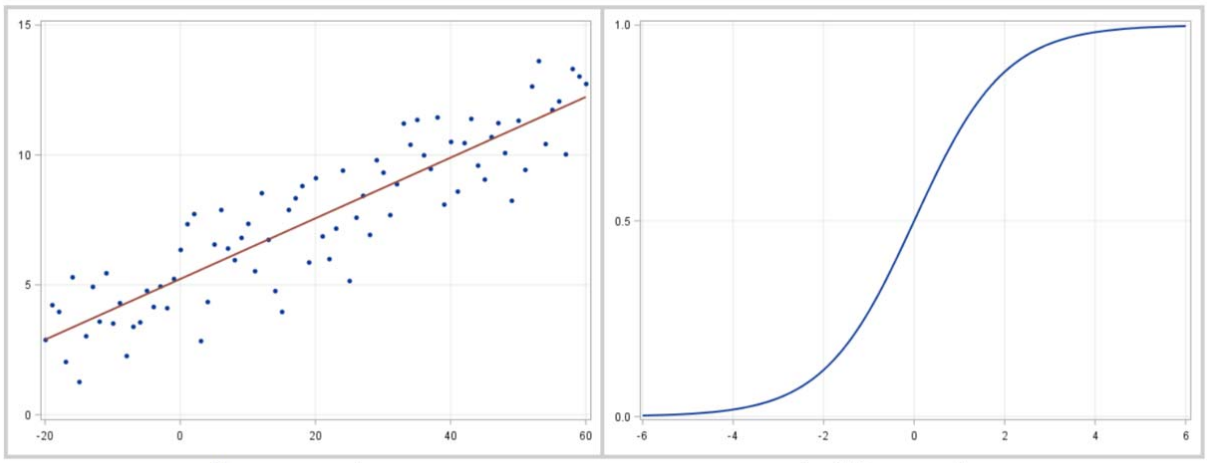
参数是机器学习模型中最重要的部分。比如，模型的迭代次数，模型的规模大小等等都会影响到最后我们需要得到的结果，对算法的训练时间和准确性都是非常敏感的。通常，具有大量参数的算法都需要我们更多的实验和调参来找到一个最好的参数组合。

当然大型的参数组合也是具有很多好处的，比如算法的灵活性会更加的强大。通常，我们可以得到一个更加好的模型结果。

个别算法的精准使用

对于个别算法，我们需要认真仔细的研究它的“脾气”，知道这些算法的输入数据特征是什么，算法具体描述是什么，他们是如何工作的额，以及他们的输出结果是代表什么含义。接下来，我们来学习几个例子。

线性回归和逻辑回归



线性回归是利用数理统计中回归分析，来确定两种或两种以上变量间相互依赖的定量关系的一种统计分析方法。预测值 y 与变量 X 之间的关系是：

$$y = \beta^T X + \epsilon$$

其中训练数据集是：

$$\{x_i, y_i\}_{i=1}^N$$

参数向量 β 是我们需要模型学习的。

如果因变量不是连续的，而是离散分类的，那么线性回归就需要被转换成逻辑回归。逻辑回归是一种非常简单，但是非常强大的分类算法。因此，当我们讨论二分类问题时，可以把等式写成：

$$\{y_i \in (-1, 1)\}_{i=1}^N$$

在逻辑回归中，我们使用不同的假设估计来区分属于类别 “1” 的概率和属于类别 “-1” 的概率。具体的说，我们尝试学习的函数是：

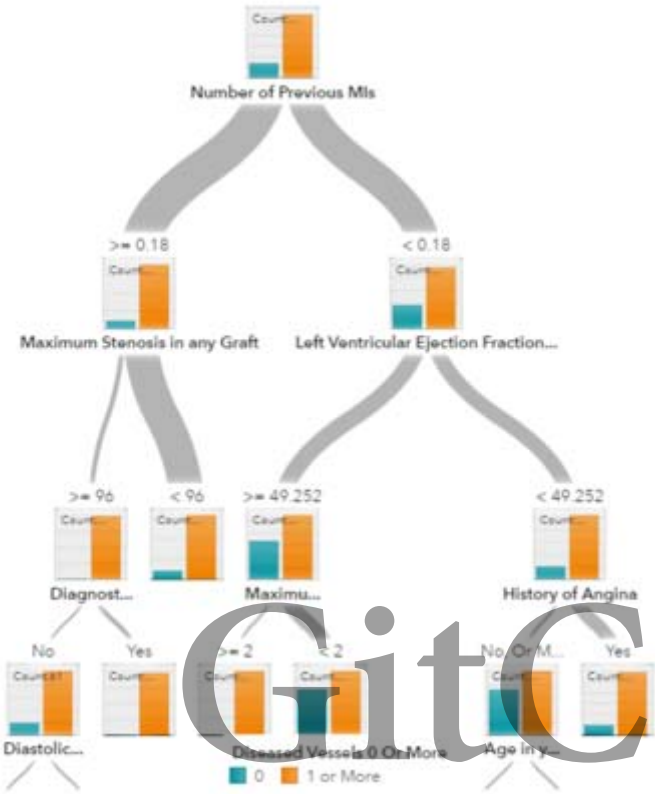
$$p(y_i = 1|x_i) = \sigma(\beta^T x_i)$$

$$p(y_i = -1|x_i) = 1 - \sigma(\beta^T x_i)$$

其中，

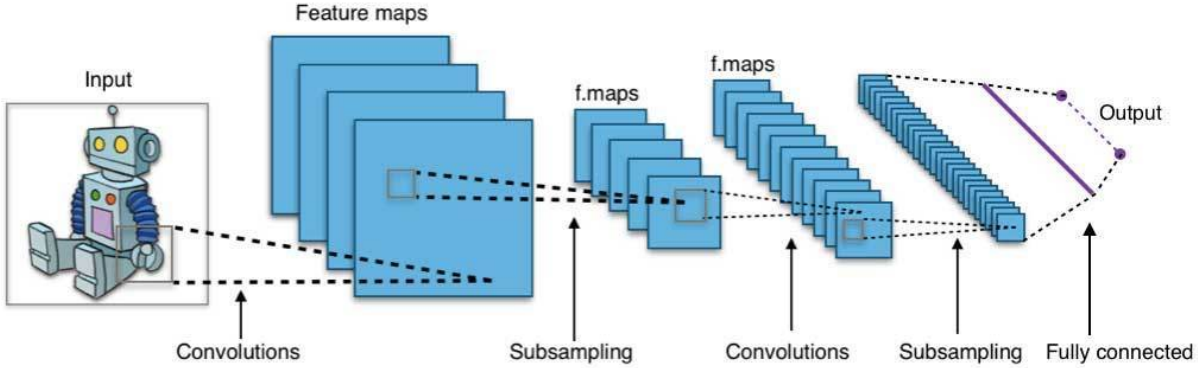
$$\sigma(x) = \frac{1}{1+exp(-x)}$$

决策树和集成树

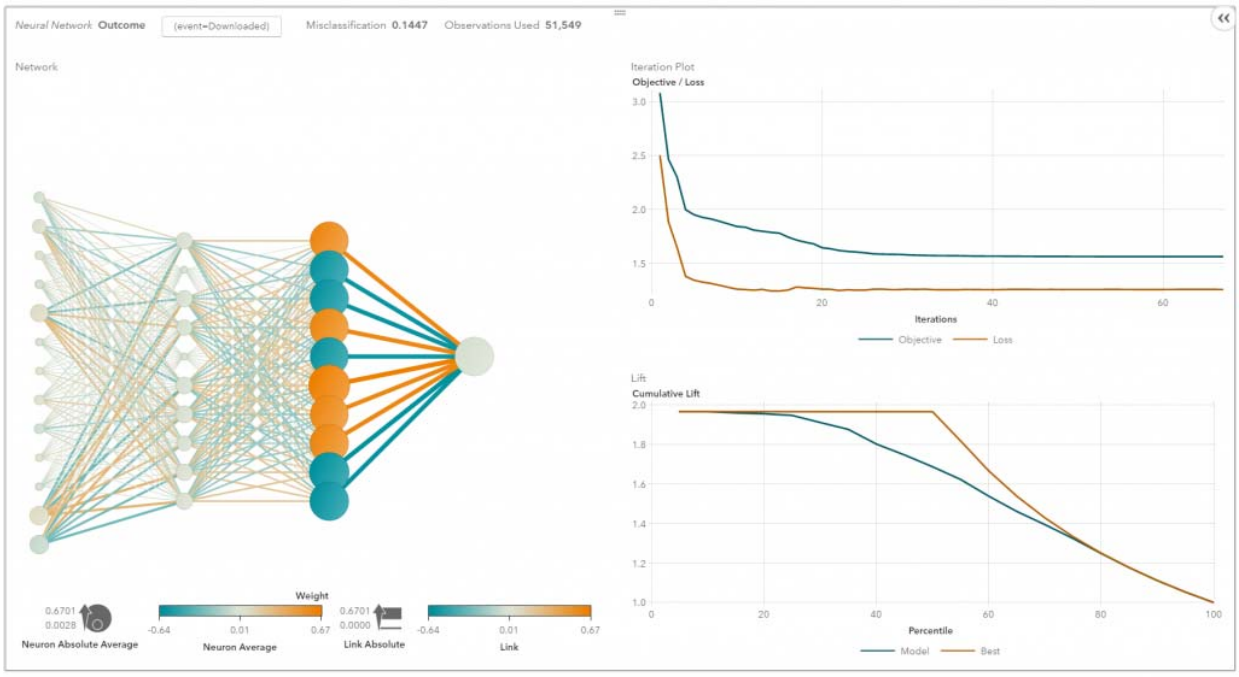


决策树，随机森林和梯度提升都是基于决策树实现的算法。决策树有很多种，但是所有的变种都只做一件事 —— 将特征标签细分到特定相同的区域里面。决策树是非常容易理解的，而且非常容易实现。然而，当我们把树的深度做的很深的时候，模型就非常容易过拟合。这时候，采用随机森林和梯度提升算法可以获得良好的性能，这两种模型也是目前比较流行的方式。

神经网络和深度学习



神经网络是在20世纪80年代中期由于其并行和分布式的处理能力而兴起的。近年来，由于卷积神经网络，循环神经网络和一些无监督学习算法的兴起，图形处理单元（GPU）和大规模并行处理（MPP）等越来越强大的计算能力，使得神经网络再次得到了复兴。



换句话说，以前的浅层神经网络已经演变成了深层神经网络。深度神经网络在监督学习中取得了非常好的表现，比如语音识别和图像分类领域都获得了比人类好的正确率。在无监督领域，比如特征提取，深度学习也取得了很好的效果。

一般情况下，一个神经网络主要由三方面组成：输入层，隐藏层和输出层。训练数据定义了输入层和输出层的维度大小。当我们的输出层是一些分类标签的时候，那么那么我们整个模型所处理的就是一个分类问题。当输出层是一个连续变量的时候，那么我们的整个模型所处理的就是一个回归问题。当我们的输出层和输入层相同时，那么我们的这个模型所处理的可能是提取数据内部的特征。中间的隐藏层大小决定了整个模型的复杂性和建模能力。

总结

至此，我们已经学习了几个算法的精准使用。在我们实际的项目中，我们需要做到对自己所熟悉的个别算法灵活使用。具体的算法表，可以查看下面这个：

Algorithm	Accuracy	Training time	Linearity	Parameters	Notes
Two-class classification					
logistic regression		●	●	5	
decision forest	●	○		6	

Algorithm	Accuracy	Training time	Linearity	Parameters	Notes
decision jungle	●	○		6	Low memory footprint
boosted decision tree	●	○		6	Large memory footprint
neural network	●			9	Additional customization is possible
averaged perceptron	○	○	●	4	
support vector machine		○	●	5	Good for large feature sets
locally deep support vector machine	○			8	Good for large feature sets
Bayes' point machine		○	●	3	
Multi-class classification					
logistic regression	●	●	●	5	
decision forest	●	○		6	
decision jungle	●	○		6	Low memory footprint
neural network	●			9	Additional customization is possible
one-v-all	-	-	-	-	See properties of the two-class method selected
Regression					
linear		●	●	4	
Bayesian linear		○	●	2	
decision forest	●	○		6	
boosted decision tree	●	○		5	Large memory footprint

Algorithm	Accuracy	Training time	Linearity	Parameters	Notes
fast forest quantile	●	○		9	Distributions rather than point predictions
neural network	●			9	Additional customization is possible
Poisson			●	5	Technically log-linear. For predicting counts
ordinal				0	For predicting rank-ordering
Anomaly detection					
support vector machine	○	○		2	Especially good for large feature sets
PCA-based anomaly detection		○	●	3	
K-means		○	●	4	A clustering algorithm

图中标记解释：● - 表示拥有卓越的精确度，快速训练时间和线性度；○ - 表示良好的准确性和适中的训练时间。参数字段中的数值越大表示模型需要的参数数量越多。