附录B　机器学习项目清单

该清单可以帮助你完成你的机器学习项目。主要有8步：

1.架构问题，关注蓝图。

2.获取数据。

3.研究数据以获取灵感。

4.准备数据以更好地将低层模型暴露给机器学习算法。

5.研究各种不同的模型，并列出最好的模型。

6.微调模型，并将其组合为更好的解决方案。

7.提出解决方案。

8.启动、监视、维护系统。

当然，为了满足需求，你可以随时调整这个清单。

架构问题，关注蓝图

1.用商业术语定义目标。

2.方案如何使用？

3.目前的解决方案/办法是什么？

4.应该如何架构问题（有监督/无监督，在线/离线，等等）？

5.如何测量性能？

6.性能指标是否与业务目标一致？

7.每个业务目标需要的最低性能是什么？

8.有没有一些相似的问题？能重用一些经验和工具吗？

9.有没有相关有经验的人？

10.如何手动解决此问题？

11.列出目前为止你（或其他人）的假设。

12.如果可能的话，验证假设。

获取数据

注意：尽可能的自动化，以便获取最新数据。

1.列出需要的数据及其体量。

2.查找并记录获取数据的途径。

3.检查需要的空间。

4.检查法律义务，必要时获取授权。

5.获取访问权限。

6.创建工作空间（确保具有足够的存储空间）。

7.获取数据。

8.将数据转换为可操作的格式（不改变数据本身）。

9.确保删除或保护敏感信息（例如，匿名）。

10.检查数据的类型和大小（时间序列、样本、地点等）。

11.采样一个测试数据集，放在一边，永远不要用它（没有数据窥视！）。

研究数据

注意：试着从这些步骤的领域专家那里获取灵感。

1.创建数据的副本用于研究（如果需要，可以将其抽样为可管理的大小）。

2.创建一个Jupyter笔记本来记录数据研究。

3.研究每个属性及其特征：

·名字。

·类型（分类、整型/浮点型、有界/无界、文本、结构等）。

·缺失值的百分比。

·噪音和噪音类型（随机、异常、舍入误差等）。

·可能有用的任务？

·分布类型（高斯、统一、对数等）。

4.对于有监督的学习任务，确认目标属性。

5.可视化数据。

6.研究属性之间的相关性。

7.研究如何手动解决问题。

8.确定希望使用转换。

9.确定可能有用的额外数据（回到之前的“获取数据”部分）。

10.记录学习到的东西。

准备数据

注意：

·在数据的副本上工作（保持原始数据集不变）。

·编写适用于所有数据转换的函数，原因有五个：

·可以很容易地准备下一次得到新数据时的数据。

·可以在未来的项目中使用这些转换。

·清理和准备测试数据集。

·一旦解决方案失效，用来清理和准备新数据实例。

·可以轻松地将你的准备选择作为超参数。

1.数据清理：

·修复或删除异常值（可选）。

·填充缺失值（例如，使用零、平均数、中位数等）或删除该行（或列）。

2.特征选择（可选）：

·删除不能为任务提供任何有用信息的属性。

3.在适当情况下，处理特征：

·离散连续特征。

·分解特征（如，分类、日期/时间等）。

·添加期望的特征转换（如，log（x）、sqrt（x）、x^2等）。

·聚合特征称为期望的新特征。

列出期望的模型

注意：

·如果数据很大，可能需要采样为较小的训练集，以便于在合理的时间内训练不同的模型（注意，这会对诸如大型神经集或随机森林等复杂模型造成不利影响）。

·再次，尽可能地自动化这些步骤。

1.使用标准参数，从不同类别（例如，线性、朴素贝叶斯、SVM、随机森林、神经网络等）中训练需求快速的不成熟的模型。

2.测量并比较它们的性能。

·对于每个模型，使用N倍交叉验证并计算N次折叠的性能测试的均值和标准差。

3.分析每个算法最重要的变量。

4.分析模型产生的错误类型。

1.数据清理：

·修复或删除异常值（可选）。

·填充缺失值（例如，使用零、平均数、中位数等）或删除该行（或列）。

2.特征选择（可选）：

·删除不能为任务提供任何有用信息的属性。

3.在适当情况下，处理特征：

·离散连续特征。

·分解特征（如，分类、日期/时间等）。

·添加期望的特征转换（如，log（x）、sqrt（x）、x^2等）。

·聚合特征称为期望的新特征。

列出期望的模型

注意：

·如果数据很大，可能需要采样为较小的训练集，以便于在合理的时间内训练不同的模型（注意，这会对诸如大型神经集或随机森林等复杂模型造成不利影响）。

·再次，尽可能地自动化这些步骤。

1.使用标准参数，从不同类别（例如，线性、朴素贝叶斯、SVM、随机森林、神经网络等）中训练需求快速的不成熟的模型。

2.测量并比较它们的性能。

·对于每个模型，使用N倍交叉验证并计算N次折叠的性能测试的均值和标准差。

3.分析每个算法最重要的变量。

4.分析模型产生的错误类型。

·人类用什么样的数据避免这些错误？

5.快速进行特征选择和处理。

6.对前面五步进行一两次快速迭代。

7.列出前三到五个最有希望的模型，倾向于选择有不同错误类型的模型。

微调系统

注意：

·你将希望为这一步使用尽可能多的数据，特别是在微调结束时。

·永远尽可能地自动化。

1.使用交叉验证微调超参数。

·把数据转换选择当作超参数，尤其是不确定时（例如，应该用零或者平均值填充缺失值吗？或者直接删除它？）。

·除非需要研究的超参数值很少，否则更喜欢在网格搜索上随机搜索。如果训练很长，你可能更喜欢贝叶斯优化方法（例如，如Jasper Snoek、Hugo Larochelle和Ryan Adams所述，使用高斯过程进行先验（https://goo.gl/PEFfGr））。[1]

2.尝试组合方法。组合多个好模型往往比单独运行效果好。

3.一旦你对最终模型有信心，在测试集上测量它的性能以估计泛化误差。

 测量泛化误差后，不要调整模型：只需要开始过度拟合测试集。

展示解决方案

1.文档化你所做的工作。

2.创建完美的演示。

·首先确保突出蓝图。

3.解释为什么你的解决方案达到了业务目标。

4.不要忘记展示你发现的一些有趣的地方。

·描述什么可以工作，什么不行。

·列出你的假设和系统的局限。

5.确保你的关键发现被完美展示或易于记忆的陈述。

启动

1.准备好生产环境的解决方案（插入生产数据输入，写单元测试等）。

2.编写监控代码，定期检查系统的性能，出问题时及时报警。

·同样需要考虑缓慢退化：随着数据的增加，模型往往会“腐烂”。

·测量性能可能需要人工流水线（例如，众包服务）。

·同时监控输入质量（例如，发送随机值的故障传感器，或其他团队的输出过时）。这对在线学习系统尤为重要。

3.定期对新数据重新建模（尽可能自动化）。

[1] “Practical Bayesian Optimization of Machine Learning Algorithms”，J.Snoek、H.Larochelle和R.Adams（2012）。