

# 机器学习如何入门

## 引言

可能你对这个名字叫“**机器学习**”的家伙不是特别的了解，但是相信用过iPhone的同学都知道iPhone的语音助手Siri，它能帮你打电话，查看天气等等；相信大家尤其是美女童鞋都用过美颜相机，它能自动化的给我们拍出更漂亮的照片；逛京东淘宝的时候，细心的童鞋应该也会发现它们会有一个栏目“猜你喜欢”；最近异军突起的新闻客户端软件今日头条，它们就是会根据分析你的日常喜好给每个人推荐不同的新闻……没错，这些功能背后的核心就是今天要介绍的主题：**机器学习**。

## 什么是机器学习

对于这个问题的解释，说实话我很有压力，因为在分享篇文章之前就有朋友告诉我，这个百度上一搜一大片，还需要你讲吗？但是，我觉得并非如此。正如同一千个读者眼里有一千个林黛玉一样，我解释的当然是我个人自从读研到工作这么多年对机器学习的学习到应用过程的独特见解。

首先我们看下图了解一下机器学习在AI（Artificial Intelligence 人工智能）领域的地位。在图中，我们可以看到，机器学习是人工智能的一个子领域。而现在火的不要不要的**深度学习**其实是机器学习的一个子分支。



机器学习在人工智能中的地位

那么到底什么才是真正的机器学习呢？在这里我将对比我和学术界大神的解释：

- **大神的解释**

机器学习研究的是计算机怎样模拟人类的学习行为，以获取新的知识或技能，并重新组织已有的知识结构使之不断改善自身。简单一点说，就是计算机从数据中学习出规律和模式，以应用在新数据上做预测的任务。

- **我的解释**

传统的机器学习主要做的事情就是利用统计学的基本观点，利用要学习的问题的历史样本数据的分布对总体样本分布进行估计。分析数据大致特性建立数学分布模型，并利用最优化的知识对模型的参数进行调优学习，使得最终的学习模型能够对已知样本进行很好的模拟与估计。最终利用学习好的模型对未知标签的样本进行预测和估计的过程。

但是越说越觉得机器学习有距离感，云里雾里高深莫测，我们不是专家，但说起算有一些从业经验，做过一些项目在实际数据上应用机器学习。这一篇就我们的经验和各位同仁的分享，总结一些对于初学者入门有帮助的方法和对进阶有用的资料。

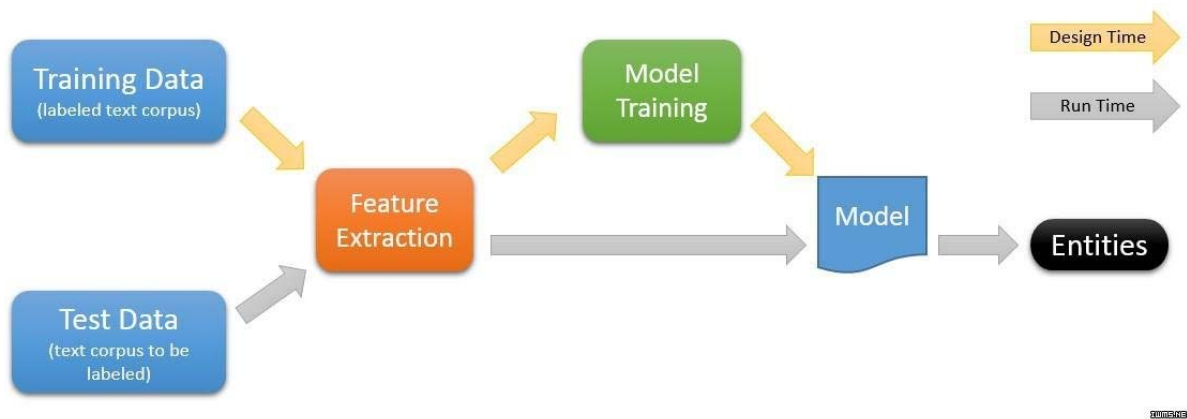
## 机器学习的基本问题

对于机器学习中的基本问题，我们将从以下几个角度进行讲解：机器学习的特点；机器学习的对象；机器学习的分类；机器学习的要素；模型的评估与选择。

### 机器学习的特点

机器学习主要特点如下：

1. 机器学习以数据为研究对象，是数据驱动的科学；
2. 机器学习的目的是对数据进行预测与分析；
3. 机器学习以模型方法为中心，利用统计学习的方法构建模型并且利用模型对未知数据进行预测和分析；
4. 统计学习是概率论、统计学、信息论、计算理论、最优化理论以及计算机科学等多领域的交叉学科，并且逐渐形成自己独立的理论体系和方法论。



机器学习的一般训练过程

## 机器学习的对象

机器学习研究的对象是多维向量空间的数据。它从各种不同类型的**数据**（数字，文本，图像，音频，视频）出发，提取数据的特征，抽象出数据的模型，发现数据中的知识，又回到数据的分析与预测中去。

## 机器学习的分类

对于机器学习的分类，绝大多数人只简单的分为有监督学习和无监督学习这两类。严格意义上来讲应该分为四大类：**有监督学习**、**无监督学习**、**半监督学习**、**强化学习**。下面对这四种学习做一下简要的介绍：

### • 有监督学习

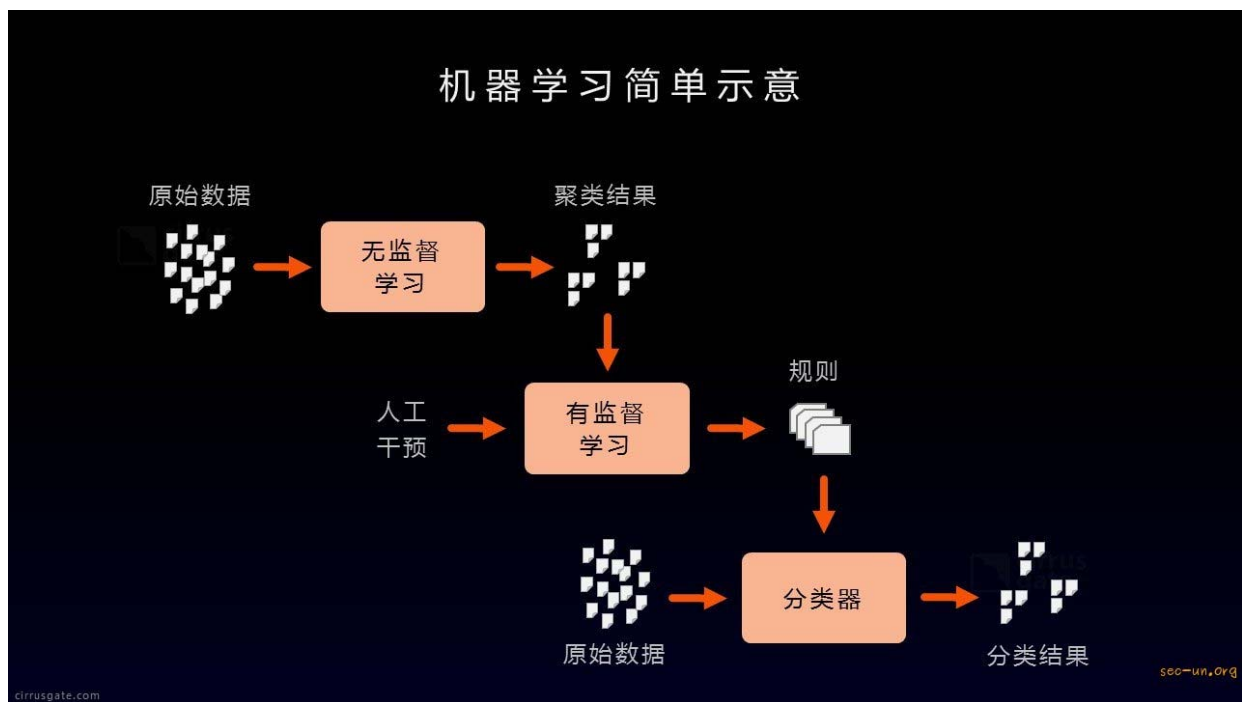
有监督学习是指进行训练的数据包含两部分信息：特征向量 + 类别标签。也就是说，他们在训练的时候每一个数据向量所属的类别是事先知道的。在设计学习算法的时候，学习调整参数的过程会根据类标进行调整，类似于学习的过程中被监督了一样，而不是漫无目标地去学习，故此得名。

### • 无监督学习

相对于有监督而言，无监督方法的训练数据没有类标，只有特征向量。甚至很多时候我们都不知道总共的类别有多少个。因此，无监督学习就不叫做分类，而往往叫做聚类。就是采用一定的算法，把特征性质相近的样本聚在一起成为一类。

### • 半监督学习

半监督学习是一种结合有监督学习和无监督学习的一种学习方式。它是近年来研究的热点，原因是在真正的模型建立的过程中，往往有类标的数据很少，而绝大多数的数据样本是没有确定类标的。这时候，我们无法直接应用有监督的学习方法进行模型的训练，因为有监督学习算法在有类标数据很少的情况下学习的效果往往很差。但是，我们也不能直接利用无监督学习的方式进行学习，因为这样，我们就没有充分的利用那些已给出的类标的有用信息。



典型半监督训练过程

## • 强化学习

所谓强化学习就是智能系统从环境到行为映射的学习，以使奖励信号(强化信号)函数值最大，强化学习不同于连接主义学习中的监督学习，主要表现在教师信号上，强化学习中由环境提供的强化信号是对产生动作的好坏作一种评价(通常为标量信号)，而不是告诉强化学习系统RLS(reinforcement learning system)如何去产生正确的动作。由于外部环境提供的信息很少，RLS必须靠自身的经历进行学习。通过这种方式，RLS在行动-评价的环境中获得知识，改进行动方案以适应环境。

## 机器学习的要素

简单地说，机器学习的三要素就是：模型、策略和算法。

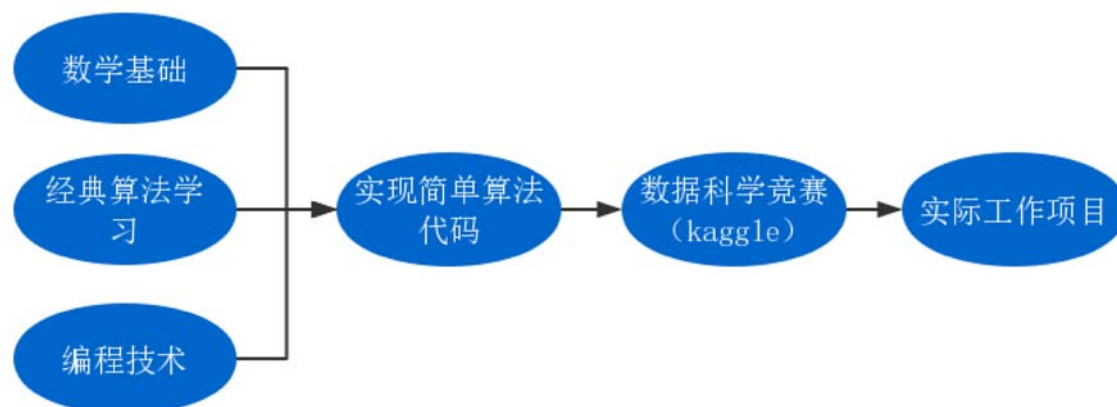
- **模型** 其实就是机器学习训练的过程中所要学习的条件概率分布或者决策函数。
- **策略** 就是使用一种什么样的评价度量模型训练过程中的学习好坏的方法，同时根据这个方法去实施的调整模型的参数，以期望训练的模型将来对未知的数据具有最好的预测准确度。
- **算法** 算法是指模型的具体计算方法。它基于训练数据集，根据学习策略，从假设空间中选择最优模型，最后考虑用什么样的计算方法去求解这个最优模型。

## 入门方法与学习路径

总的来说，机器学习的学习门槛还是蛮高的。当然，也得看你的目标是什么了。如果你的目标只是为了用机器学习的一些算法解决一些简单的分类回归问题，那么其实也不

难。但是，如果你的目标是成为机器学习科学家，提出并改进一些算法的新的应用场景或者是算法的执行性能的话，那么你的学习难度和学习周期必定是很艰辛和漫长的。

本文对所有的读者的假设是前者，因此我们也制定了与机器学习科学家不一致的学习道路。大致的学习过程如下图所示：



机器学习的入门过程

对于上图，之所以最左边写了『数学基础』『经典算法学习』『编程技术』三个并行的部分，是因为机器学习是一个将数学、算法理论和工程实践紧密结合的领域，需要扎实的理论基础帮助引导数据分析与模型调优，同时也需要精湛的工程开发能力去高效化地训练和部署模型和服务。

在互联网领域从事机器学习的人基本上属于以下两种背景：其中绝大部分是程序员出身，这类童鞋工程经验相对会多一些；另一部分是学数学统计领域的，这部分童鞋理论基础相对扎实一些。因此对比上图，这二类童鞋入门机器学习，所欠缺和需要加强的部分是不一样的。

下面就从三个基本技能讲起。

## 数学基础

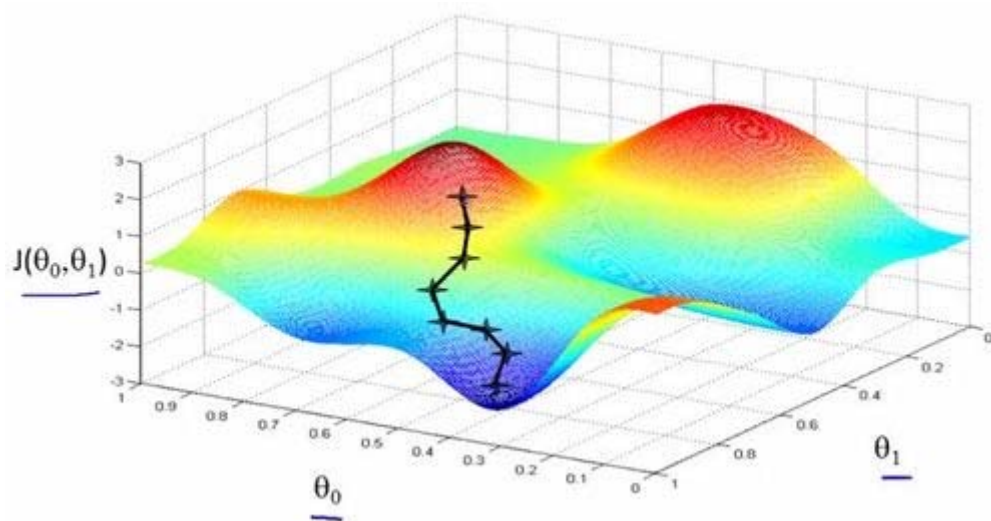
曾经有无数的满怀激情，誓要在机器学习领域有一番作为的同学，在看到公式的一刻突然就觉得自己狗带了。是的，机器学习之所以门槛高并且显得高大上的主要原因就是数学。每一个算法，要在训练集上最大程度拟合同时又保证泛化能力，需要不断分析结果和数据，调优参数，这需要我们对数据分布和模型底层的数学原理有一定的理解。所幸的是如果只是想合理应用机器学习，而不是做相关方向高精尖的研究，所需要的数学知识读完本科的理工科童鞋还是能很容易的把这些数学知识学明白的。

基本所有常见机器学习算法需要的数学基础，都集中在微积分、线性代数和概率与统计当中。下面我们先过一过知识重点，文章的后部分会介绍一些帮助学习和巩固这些知识的资料。

### 微积分

微分的计算及其几何、物理含义，是机器学习中大多数算法的求解过程的核心。比如算法中运用到梯度下降法、牛顿法等。如果对其几何意义有充分的理解，就能理解“梯度下降是用平面来逼近局部，牛顿法是用曲面逼近局部”，能够更好地理解运用这样的方法。

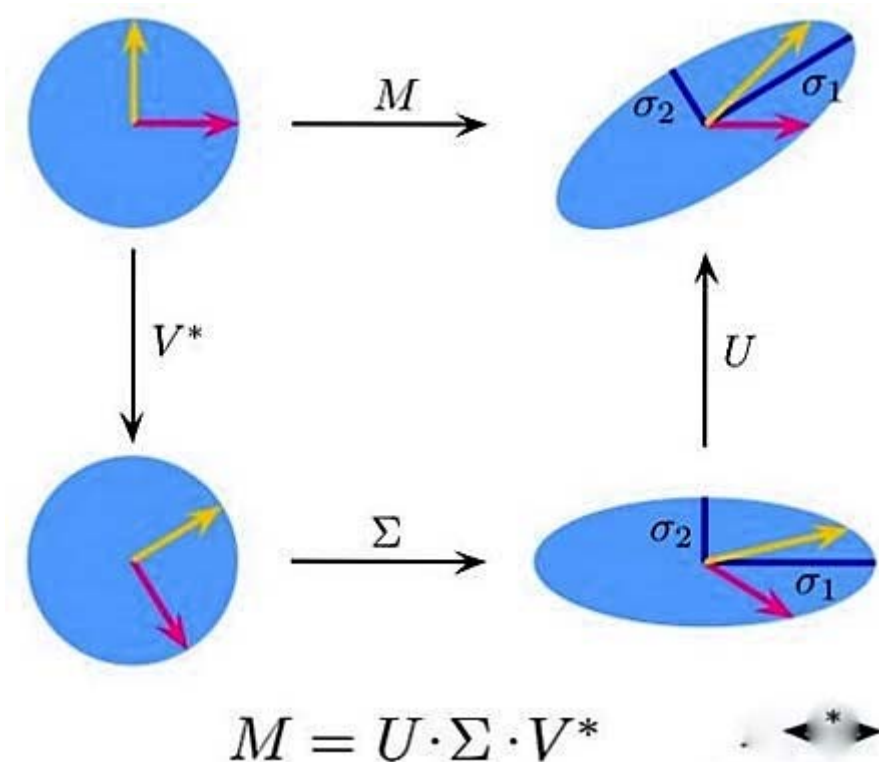
凸优化和条件最优化的相关知识在算法中的应用随处可见，如果能有系统的学习将使得你对算法的认识达到一个新高度。



梯度下降法示意图

## 线性代数

大多数机器学习的算法要应用起来，依赖于高效的计算，这种场景下，程序员童鞋们习惯的多层for循环通常就行不通了，而大多数的循环操作可转化成矩阵之间的乘法运算，这就和线性代数有莫大的关系了。向量的内积运算更是随处可见。矩阵乘法与分解在机器学习的主成分分析（PCA）和奇异值分解（SVD）等部分呈现刷屏状地出现。

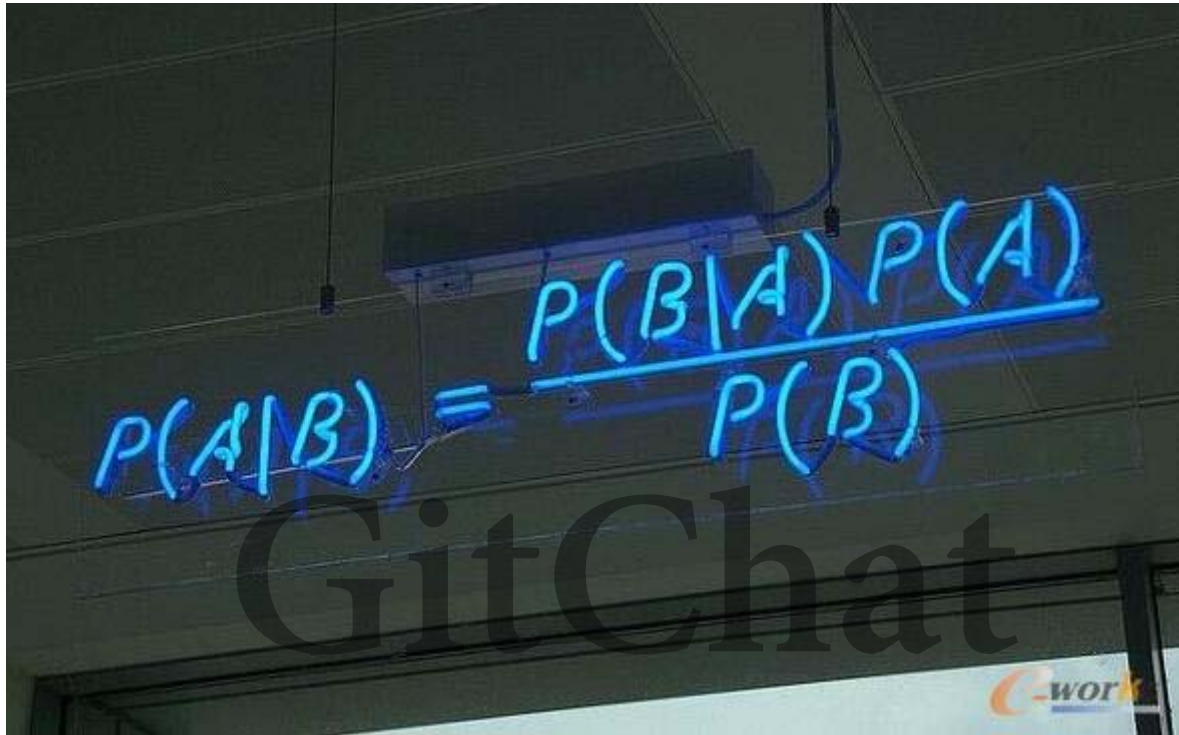




## 概率与统计

从广义来说，机器学习在做的很多事情，和统计层面数据分析和发掘隐藏的模式，是非常类似的。

极大似然思想、贝叶斯模型 是理论基础，朴素贝叶斯(*Naive Bayes*)、语言模型(*Ngram*)、隐马尔科夫(*HMM*)、隐变量混合概率模型是他们的高级形态。常见分布如高斯分布是混合高斯模型(*GMM*)等的基础。



朴素贝叶斯算法的基本原理

## 经典算法学习

绝大多数平常的应用中，经典的机器学习算法就能够解决其中绝大多数的问题。因此，对机器学习经典算法的学习和掌握是相当有必要的。

接下来我们会分门别类的介绍一下：

- **分类算法：** 逻辑回归（LR），朴素贝叶斯（Naive Bayes），支持向量机（SVM），随机森林（Random Forest），AdaBoost，GDBT，KNN，决策树.....
- **回归算法：** 线性回归（Linear Regression），多项式回归（Polynomial Regression），逐步回归（Stepwise Regression），岭回归（Ridge Regression），套索回归（Lasso Regression）
- **聚类算法：** K均值（K-Means），谱聚类、DBSCAN聚类、模糊聚类、GMM聚类、层次聚

- **降维算法**：PCA（主成分分析）、SVD（奇异值分解）
- **推荐算法**：协同过滤算法

在这里，我还是希望解释一下 **算法** 这个概念在不同的地方出现的意义给广大的读者带来的疑惑。本文介绍的机器学习算法和我们程序员所说的“数据结构与算法分析”里的算法略有不同。前者更关注结果数据的召回率、精确度、准确性等方面，后者更关注执行过程的时间复杂度、空间复杂度等方面。当然，实际机器学习问题中，对效率和资源占用的考量是不可或缺的。

## 编程技术

### 技术选择

编程技术无非是语言和开发环境了。在此，对初入门学习机器学习的小白童鞋来说，我的个人建议是：**Python + PyCharm**。如下图所示是他们的Logo。



Python 与 PyCharm 软件示意图

语言和工具选择好了，对于小白来说，我们当然使用成熟的机器学习库。那么对于python机器学习来说，毫无疑问我们选择的是scikit-learn。

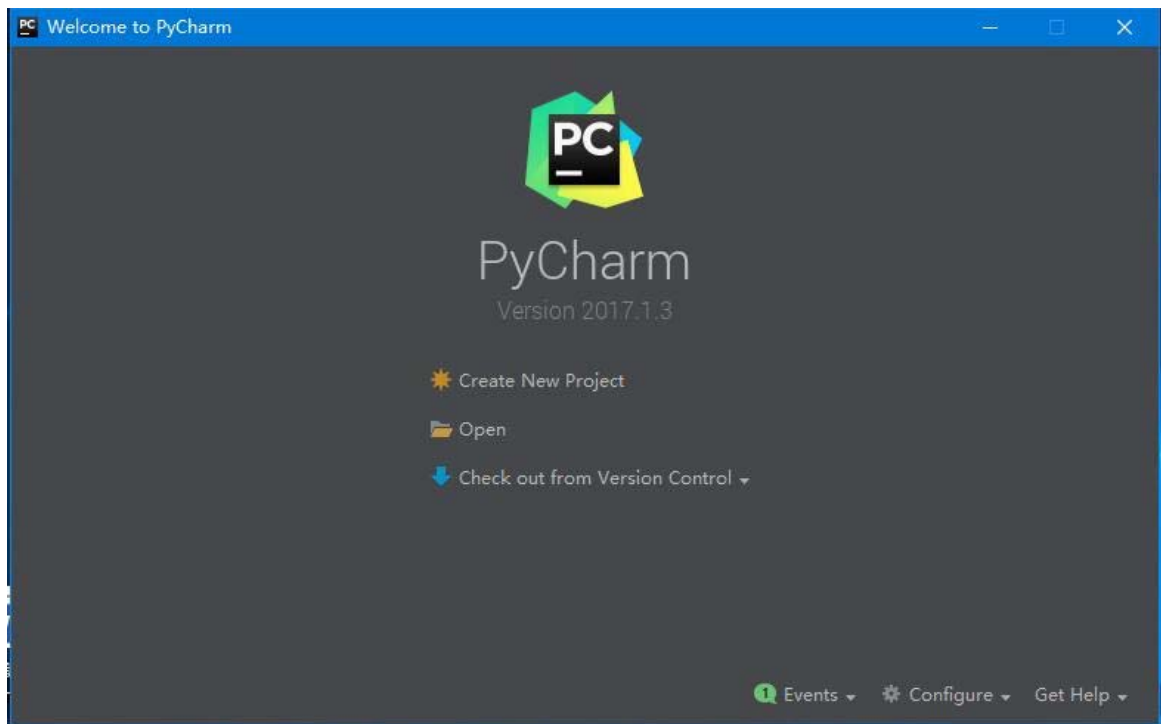
### 软件安装

关于在Windows下安装python和scikit-learn的方法步骤，请参阅我的CSDN博客[Windows下安装Scikit-Learn](#)。对于PyCharm的下载，请点击[PyCharm官网](#)去下载，当然windows下软件的安装不用解释，傻瓜式的点击**下一步**就行了。

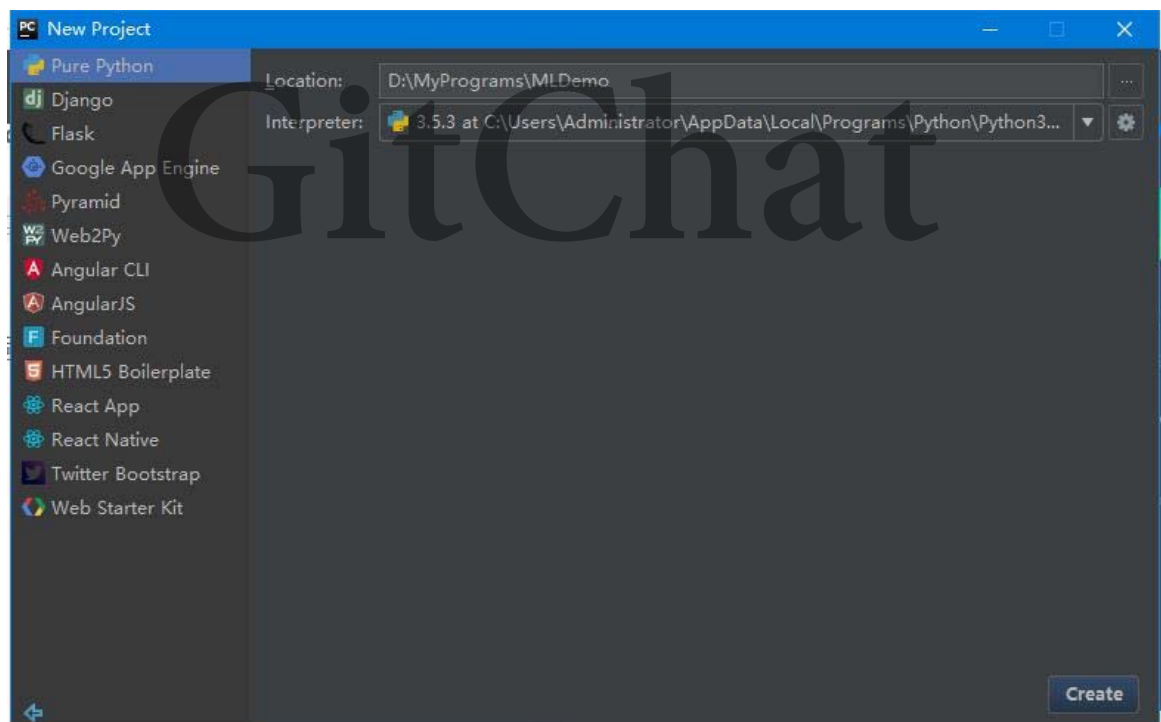
### 实战操作

- 创建项目
  - 打开 **PyCharm**，点击 **Create New Project**

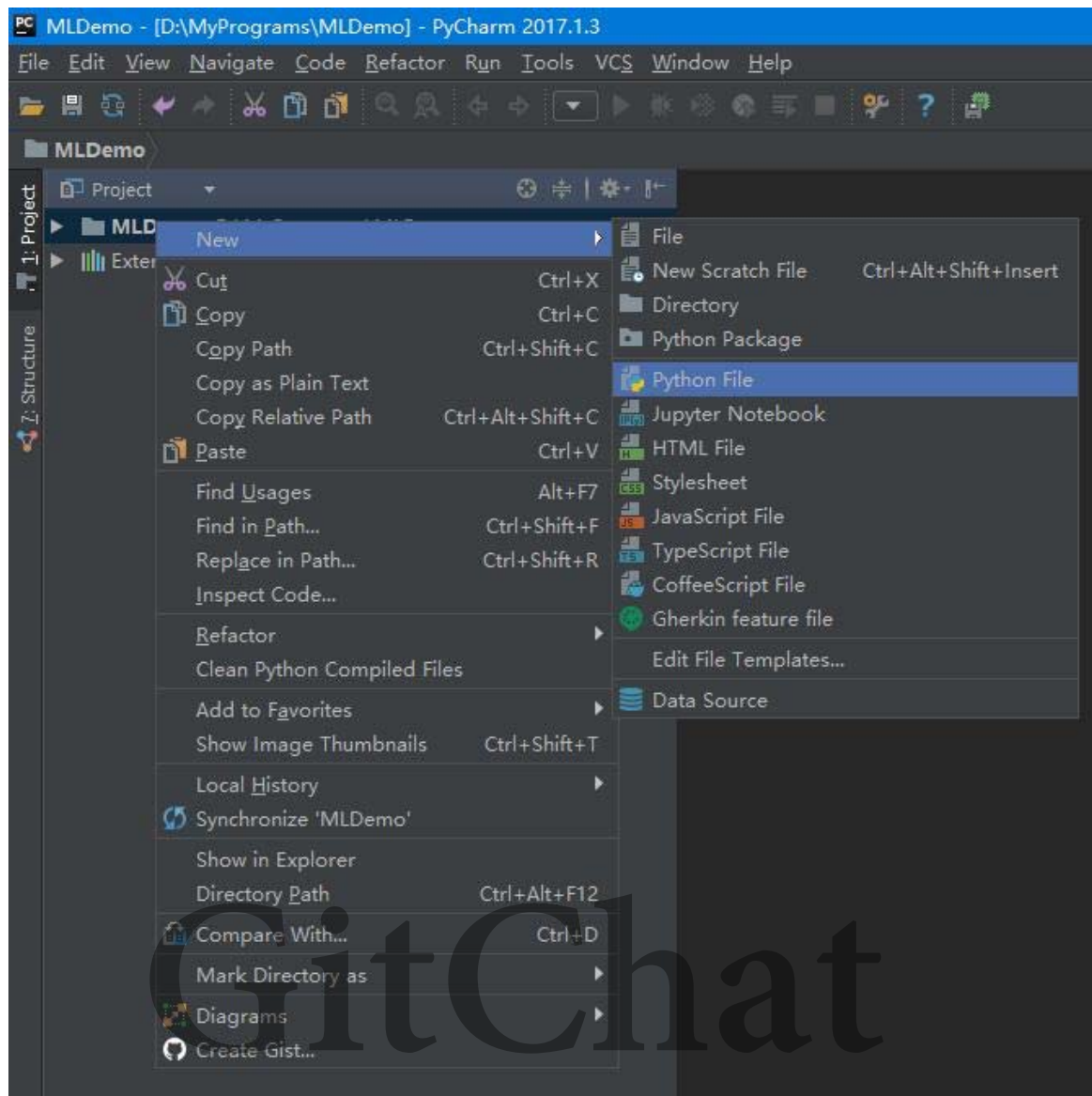




- 接下来选择Pure Python，并选择程序目录同时设置项目名称为 MLDemo，点击右下角的 **Create**。



- 在生成的项目MLDemo 上右击，依次选择 **New** → **Python**，命名 **MLDemo**

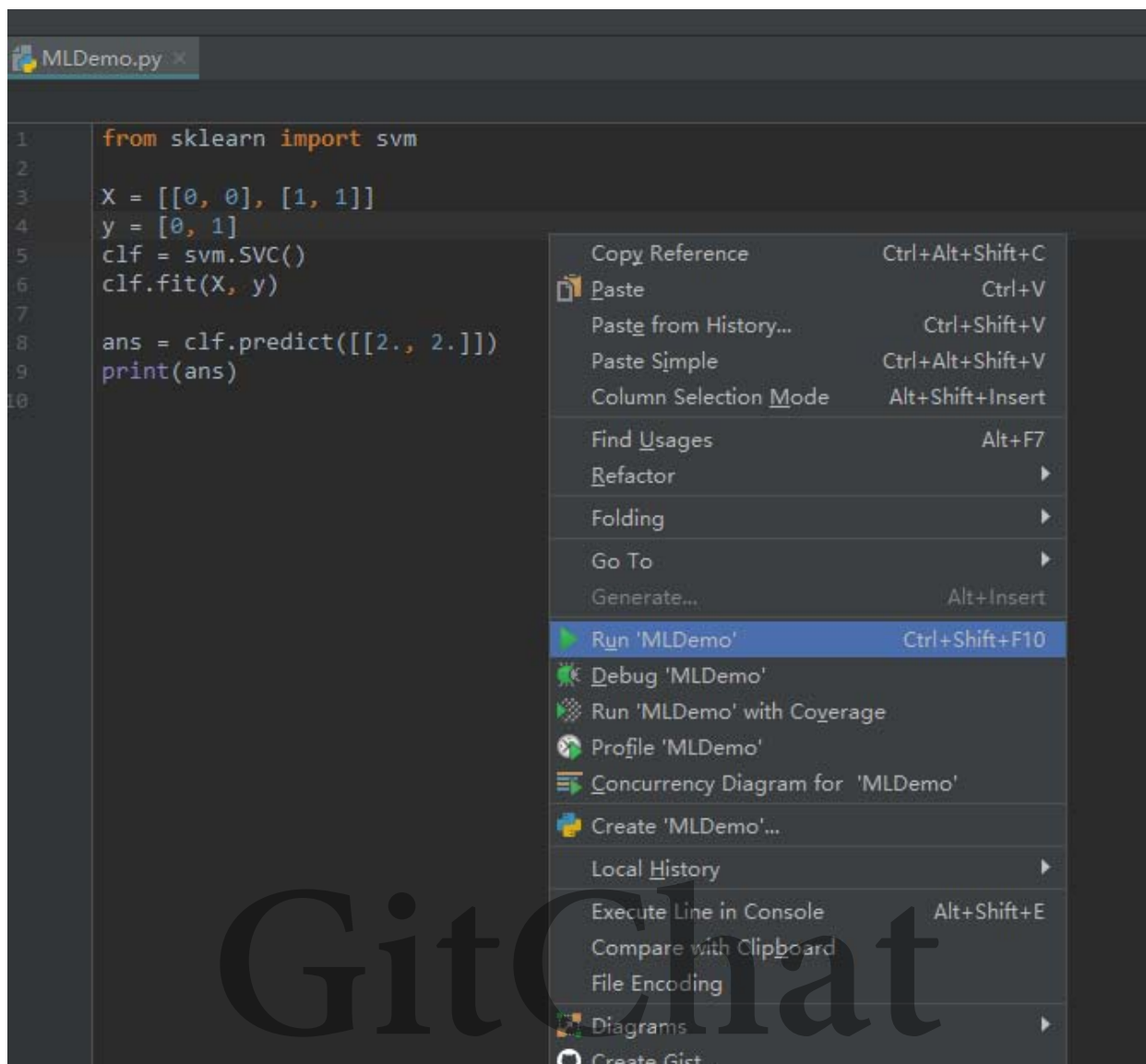


- 编写如下代码，然后右击代码区，点击 **Run MLDemo**

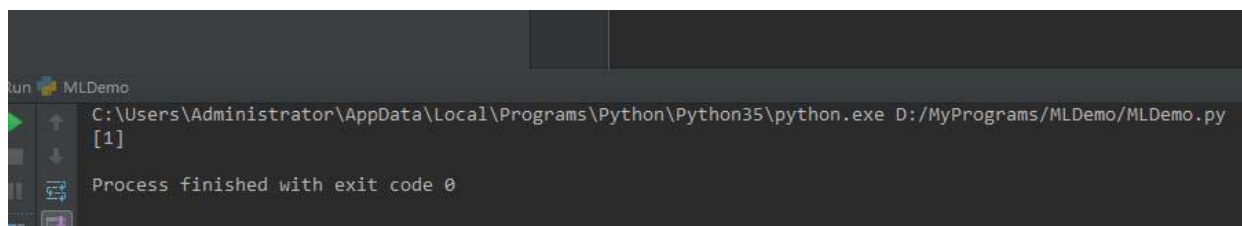
```
from sklearn import svm

X = [[0, 0], [1, 1]]
y = [0, 1]
clf = svm.SVC()
clf.fit(X, y)

ans = clf.predict([[2., 2.]])
print(ans)
```



- 对运行结果进行解释



源程序调用了sklearn包的svm类，用于后续程序的分类器是训练：

1.  $X = [[0, 0], [1, 1]]$  是定义两个训练样本的特征向量；
2.  $y = [0, 1]$  是与  $X$  中特征向量对应的类标签；
3.  $clf = svm.SVC()$  是定义用于分类的svm分类器；
4.  $clf.fit(X, y)$  是对以  $x$  和  $y$  为特征向量和类标的样本数据进行有监督训练；
5.  $ans = clf.predict([[2., 2.]])$  是预测  $[2., 2.]$  这个点属于哪一类，并将预测结果返回给ans；

6. `print(ans)` 输出返回结果，我们知道该点属于1类，也就是与 [1, 1]同属一类，结果正确；

## 你是否真的准备好了？

说完了机器学习的入门过程，我得给大家泼点冷水。虽然说目前AI真的很火热，就在刚刚，我写累了休息看新闻的时候就有新闻推送给我：商汤科技B轮融资4.5亿美元。这场革命是机遇，但是它真的适合你吗？我可以很肯定的说，并不是所有人都适合转行AI。

**下面是的总结，想转行的人可以自我对照：**

1. 如果你天生感觉学习数学很吃力，并且代码能力很一般的人。我可以很负责人的告诉你，转行AI，学习机器学习算法将会是你人生的灾难。对于这类 **猿友** 你一定不能转行AI；
2. 如果你数学一般，但是编程能力非常好，你曾经有着用**代码改变世界的雄心**。对于这一类 **猿友**，我觉得你转行也行，但是你一定要走应用化的AI道路。因为数学是你的天花板，你注定成不了 **Hinton** 那样的学术大牛；
3. 如果你数学很好，但是编程薄弱。恭喜你，你具备了转行AI的先天优势。对于这类 **猿友**，我觉得你可以转行AI，但是你得努力把编程水平提上来。
4. 如果你数学很牛，曾经与**菲尔兹奖**擦肩而过，曾经给Apache顶级项目贡献N万行核心代码。恭喜你，AI领域需要的就是你，你就是未来的**Hinton**，**吴恩达**.....

---

参考文献：

1. 李航. 统计学习方法[M]. 清华大学出版社, 2012.
2. 周志华. 机器学习 := Machine learning[M]. 清华大学出版社, 2016.
3. [Windows下安装Scikit-Learn](#)
4. [Scikit-learn实战之SVM分类](#)