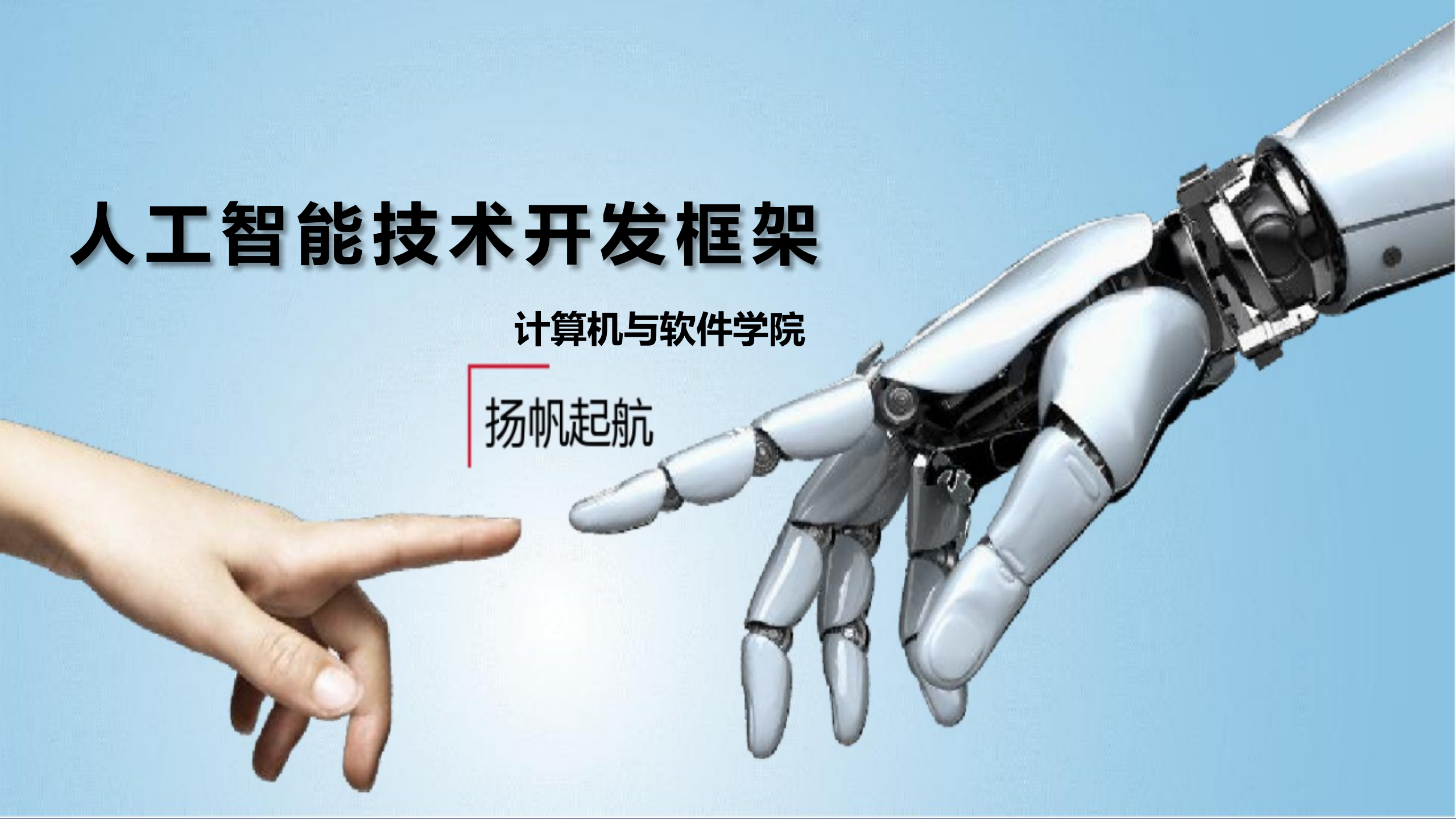
人工智能技术开发框架

计算机与软件学院

扬帆起航

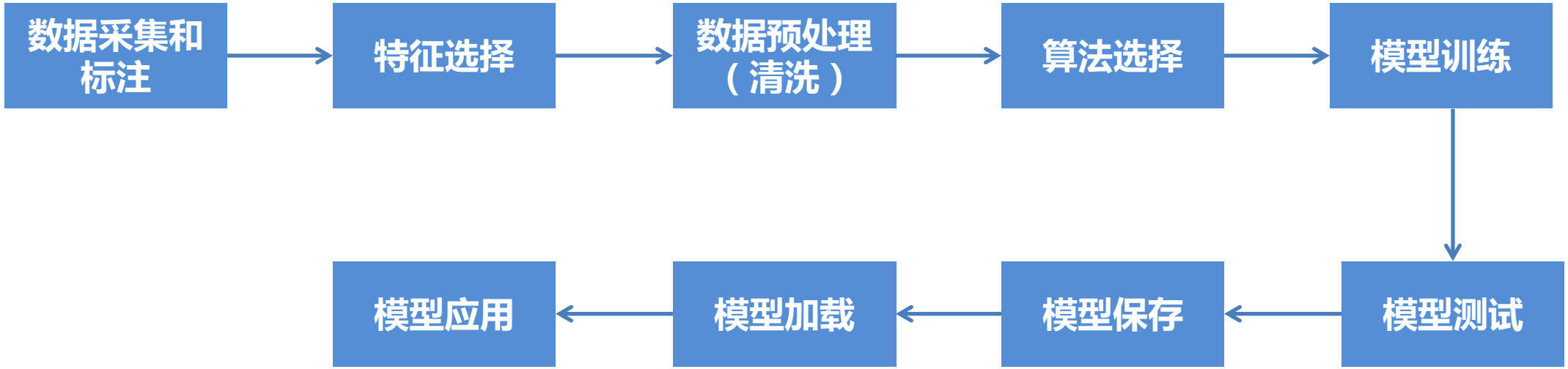


第四章 机器学习框架Scikit-Learn

机器学习框架sklearn

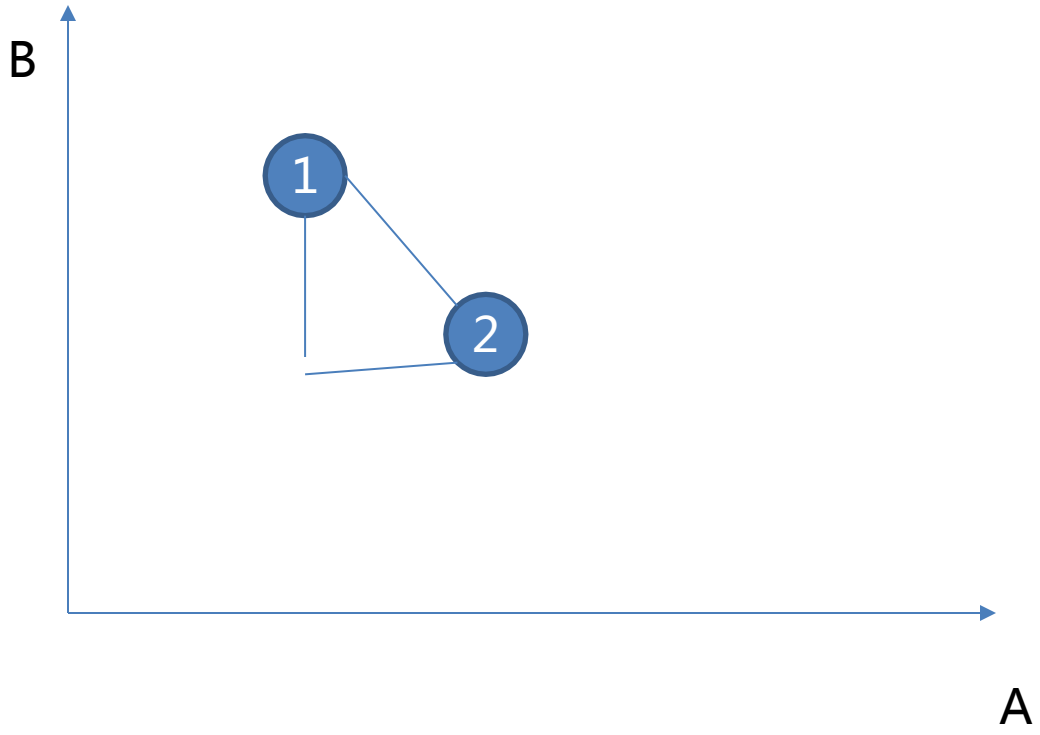
- sklearn是一个Python第三方提供的非常强力的机器学习库，它包含了从数据预处理到训练模型的各个方面。在实战使用scikit-learn中可以极大的节省我们编写代码的时间以及减少我们的代码量，使我们有更多的精力去分析数据分布，调整模型和修改超参

机器学习流程



数据预处理（清洗）

数据预处理工具：Numpy，Pandas



数据标准化

- 数据正规化：sklearn.preprocessing.Normalizer(norm='l2')
  - 正则化的过程是将每个样本缩放到单位范数(每个样本的范数为1)，对每个样本计算其p-范数，然后对该样本中每个元素除以该范数，这样处理的结果是使得每个处理后样本的p-范数(L1-norm,L2-norm)等于1。
- 数据归一化（零均值）：sklearn.preprocessing.StandardScaler()
  - 将数据按其属性(按列进行)减去其均值，然后除以其方差。最后得到的结果是，对每个属性/每列来说所有数据都聚集在0附近，方差值为1。
- 最大最小标准化：sklearn.preprocessing.MinMaxScaler(feature_range=(0, 1))
  - 将属性缩放到一个指定的最大值和最小值(通常是1-0)之间。

机器学习方法

- 有监督学习
- 无监督学习
- 半监督学习

有监督学习

- 回归
- 分类

很多算法既可以用于分类，也可以用于回归，如：决策树、支持向量机、随机森林、Adaboost等

回归

- 线性回归
- 多项式变换

线性回归

- 线性回归：sklearn.linear_model.LinearRegression()
    - 没什么特别参数，正常使用默认参数

多项式变换

- 背景：有时从已知数据里挖掘出更多的特征不是件容易的事情，此时可以用纯数学的方法，增加多项式特征。比如，原来的输入特征只有问X1,X2，优化后可以增加特征，变成X1,X2, X1^2，X2^2。这样也可以增加模型复杂度，从而改善欠拟合的问题。多项式特征变换后再线性回归，本质上是增加特征数量，构造新的特征以解决欠拟合问题
- 多项式变换：sklearn.preprocessing.PolynomialFeatures(degree=2)
    - 主要参数是degree，体现了变换的复杂性

示例代码：

```
quadratic = PolynomialFeatures(degree=2)
X_train = quadratic.fit_transform(X_train)
X_test = quadratic.transform(X_test)
```

分类

- 逻辑回归：sklearn.linear_model.LogisticRegression()
- 决策树（随机森林的特例）：sklearn.tree.DecisionTreeClassifier()
- 朴素贝叶斯：sklearn.naive_bayes()
- K近邻（KNN）：sklearn.neighbors.KNeighborsClassifier(n_neighbors=k)
- 支持向量机（SVM）：sklearn.svm.SVC()。核函数的本质：线性不可分情况下，将数据进行非线性映射到高维空间，高维特征空间中变成线性可分
- 集成学习：（1）随机森林：sklearn.ensemble.RandomForestClassifier()；（2）AdaBoost：sklearn.ensemble.AdaBoostClassifier()

逻辑回归

- sklearn.linear_model.LogisticRegression(penalty='l2’，C=1.0)
- 主要参数：

  Penalty:｛‘l1’，‘l2’，‘elasticnet’，‘none’｝, default=’l2’。正则化方式。

  C: float, default=1.0。正则化强度的倒数；必须是正浮点数。与支持向量机一样，较小的值指定更强的正则化。

决策树

- skleam.tree.DecisionTreeClassifier()
- 主要参数：
  - criterion：{“gini”，“entropy”}, default=”gini”。特征选择算法。一种是基于信息熵，另外一种是基于基尼不纯度。有研究表明，这两种算法的差异性不大，对模型的准确性没有太大的影响。相对而言，信息熵运算效率会低一些，因为它有对数运算
  - splitter：{“best”，“random”}, default=”best”。创建决策树分支的选项，一种是选择最优的分支创建原则，另外一种是从排名靠前的特征中，随机选择一个特征来创建分支，这个方法和正则项的效果类似，可以避免过拟合问题
  - max depth：int, default=None。指定决策树的最大深度。通过指定该参数，用来解决模型过拟合问题
  - min_samples_split：int or float, default=2。这个参数指定能创建分支的数据集的大小，默认是2。如果一个节点的数据样本个数小于这个数值，则不再创建分支。这也是一种前剪枝的方法
  - min_samples_leaf：int or float, default=1。创建分支后的节点样本数量必须大于等于这个数值，否则不再创建分支。这也是一种前剪枝的方法
  - max_leaf_nodes：int, default=None。除了限制最小的样本节点个数，该参数可以限制最大的样本节点个数
  - min_impurity_split：float, default=0。可以使用该参数来指定信息增益的阈值。决策树在创建分支时，信息增益必须大于这个阈值，否则不创建分支

朴素贝叶斯

朴素贝叶斯算法在自然语言领域有广泛的应用

- 伯努利分布：skleam.naive_bayes.BemoulliNB()
- 多项式分布：skleam.naive_bayes.MultinomialNB()
- 高斯分布：skleam.naive_bayes.GaussianNB()

伯努利分布

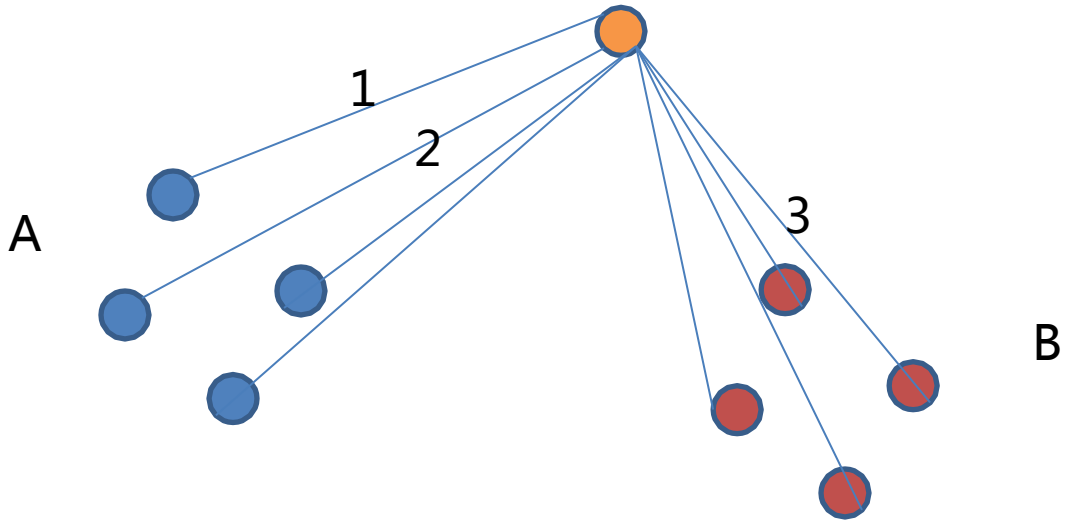
- 抛一枚硬币，要么出现正面，要么出现反面（假设硬币不会立起来〉。假如出现正面的概率是p，则出现反面的概率就是1-p。符合这种规律的概率分布，称为伯努利分布(Bernoulli Distribution ）
- 更一般的情况，即不止两种可能性时，假设每种可能性是Pi， 则满足所有的概率Pi之和为1 的概率分布，称为类别分布（ Categorical Distribution ）。例如投掷一个骰子， 则会出现6 种可能性，所有的可能性加起来的概率为1
- 满足伯努利分布的分类问题可以使用skleam.naive_bayes.BemoulliNB()，通常使用默认参数即可

多项式分布

- 多项式分布是指满足类别分布的实验， 连续做n 次后，每种类别出现的特定次数组合的概率分布情况
- 投6 次骰子，每个点数都出现一次，可以是（ 1, 2, 3, 4, 5, 6 ），也可以是（ 1, 3, 2, 4, 5, 6 ），那么出现（1, 2, 3, 4, 5, 6 ）这样特定顺序组合的概率是满足多项式分布的
- 看一个例子，同时投掷6个质地均匀的骰子，出现（1, 2, 3, 4, 5, 6 ）这种组合的概率是多少？可以把这个问题转换成连续6 次投掷质地均匀的骰子，每个类别都出现一次的概率，这是个典型的多项式分布问题
- 实际应用中，涉及大量的自然语言处理的手段，包括中文分词技术、文档分类、词的数学表示等，满足多项式分布的特点，可以使用skleam.naive_bayes.MultinomialNB()，通常使用默认参数即可

高斯分布

- 高斯分布（Gaussian Distribution）也称为正态分布（Normal Distribution），是自然界最常见的一种概率密度函数。人的身高满足高斯分布，特别高和特别矮的人出现的相对概率都比较低。人的智商也符合高斯分布，特别聪明的天才和特别笨的人出现的相对概率都比较低
- 满足高斯分布的分类问题可以使用skleam.naive_bayes.GaussianNB()，通常使用默认参数即可



K近邻（KNN）分类器

- k近邻算法的核心思想是未标记样本的类别，由距离其最近的k个邻居投票来决定。假设，有一个已经标记的数据集，即已经知道了数据集中每个样本所属的类别。此时，有一个未标记的数据样本，任务是预测出这个数据样本所属的类别。k近邻算法的原理是：计算待标记的数据样本和数据集中每个样本的距离，取距离最近的k个样本。待标记的数据样本所属的类别，就由这k个距离最近的样本投票产生
- skleam.neighbors.KNeighborsClassifier(n_neighbors=5)
  - 主要参数是k即n_neighbors。参数选择需要根据数据来决定。k值越大，模型的偏差越大，对噪声数据越不敏感，当k值很大时，可能造成模型欠拟合。k值越小，模型的方差就会越大，当k值太小，就会造成模型过拟合

支持向量机（SVM）

- sklearn.svm.SVC(C=1.0, kernel='rbf', degree=3)
- 主要参数：
  - C：float, default=1.0。惩罚参数
  - kernel：kernel{‘linear’，‘poly’，‘rbf’，‘sigmoid’，‘precomputed’}, default=’rbf’。常用的核函数是高斯径向基核函数rbf、多项式核函数poly、 sigmoid 核函数。核函数的本质：线性不可分情况下，将数据进行非线性映射到高维空间，高维特征空间中变成线性可分
  - Degree：int, default=3。当使用poly核函数时该参数有效，对于其他核函数则无效

集成学习

集成算法(Ensemble）是一种元算法(Meta-algorithm），它利用统计学采样原理，训练出成百上千个不同的算法模型。当需要预测一个新样本时，使用这些模型分别对这个样本进行预测，然后采用少数服从多数原则， 决定新样本的类别。集成算法可以有效地解决过拟合问题。

- 随机森林：RandomForestClassifier()
- Adaboost：AdaBoostClassifier()

随机森林

- sklearn.ensemble.RandomForestClassifier()，以决策树作为基分类器（学习器）。
- 主要参数：
  - n_estimators：int, default=100。森林中树的数量
  - criterion：{“gini”，“entropy”}, default=”gini”。特征选择算法。一种是基于信息熵，另外一种是基于基尼不纯度。有研究表明，这两种算法的差异性不大，对模型的准确性没有太大的影响。相对而言，信息熵运算效率会低一些，因为它有对数运算
  - max depth：int, default=None。指定决策树的最大深度。通过指定该参数，用来解决模型过拟合问题
  - min_samples_split：int or float, default=2。这个参数指定能创建分支的数据集的大小，默认是2。如果一个节点的数据样本个数小于这个数值，则不再创建分支。这也是一种前剪枝的方法
  - min_samples_leaf：int or float, default=1。创建分支后的节点样本数量必须大于等于这个数值，否则不再创建分支。这也是一种前剪枝的方法
  - max_leaf_nodes：int, default=None。除了限制最小的样本节点个数，该参数可以限制最大的样本节点个数
  - min_impurity_split：float, default=None。可以使用该参数来指定信息增益的阀值。决策树在创建分支时，信息增益必须大于这个阈值，否则不创建分支

Adaboost

- sklearn.ensemble.AdaBoostClassifier() ，默认以决策树作为基分类器（学习器）。
- 主要参数：
  - base_estimator：object, default=None。指定基分类器的类型，默认以决策树作为基分类器。
  - n_estimators：int, default=50。终止boosting算法的最大分类器数量。如果达到该数量则学习过程将提前停止
  - learning_rate：float, default=1。在每次提升迭代中应用于每个分类器的权重。较高的学习率会增加每个分类器的贡献。在learning_rate和n_estimators之间有一个折衷
  - algorithm：{‘SAMME’，‘SAMME.R’}, default=’SAMME.R’。‘SAMME.R’和‘SAMME’是两种算法

无监督学习

- 聚类
- 降维