机器学习建模流程：

**明确问题**

首先，我们需要预览这个项目。项目的目的是什么，以房价预测为例，数据为StatLib的加州房产数据，那么目的就是预测街区的房产价格中位数。

**划定问题及分析**

要知道商业目的是什么，毕竟建立模型不是最终目的。比如说，目的是输出一个价格传给另一套系统来判断是否值得投资以及利润是多少。  
要知道现在的解决方案效果怎么样，比如会给出一个现解决方案的误差率是alpha。  
现在我们可以进一步研究问题，明确这个问题是监督/非监督，还是强化模型？是分类/回归，还是聚类等其他。要使用批量学习还是线上学习？  
分析，我们有房价的值，所以是一个监督问题；我们最终是要预测得到房价中位数，因此是一个回归问题，而且是一个多变量预测回归，因为有很多影响参数；另外，没有连续的数据流入，没有特别需求需要对数据变动作出快速适应。数据量不大可以放到内存中，因此批量学习就可以。【如果数据量很大，你可以要么在多个服务器上对批量学习做拆分（使用 MapReduce 技术，后面会看到），或是使用线上学习】

**选择性能指标**

在这里我们需要选择一个评价指标，回归问题的典型指标是均方根误差RMSE，它表征的是系统预测误差的标准差。  
另外，也可以使用差平方绝对误差。

**核实假设**

再一次核实之前的分析是否准确，需要联系下游的处理进行检查。

1、获取数据：def fetch\_housing\_data()

2、加载数据：def load\_housing\_data()

3、数据分析理解：借助pandas，numpy，matplotlib。

绘制数据的散点图，观察属性之间的关系；

Input：分析属性

Output：标签

理解数据特征，选择算法模型

分析属性之间的关系，简化数据属性，进行某些属性的合并，从而选择最相关的属性来训练模型。

分割数据集

查看数据：

查看数据结果，预览数据

import pandas as pd

data=pd.read\_csv('路径')

data.head()

data.info()#返回特征的数量及类型

data.describe()#返回数量、均值、标准差、最值等信息

通过直方图查看数据分布

%matplotlib inline # only in a Jupyter notebook

import matplotlib.pyplot as plt

data.hist(bins=50, figsize=(20,15))

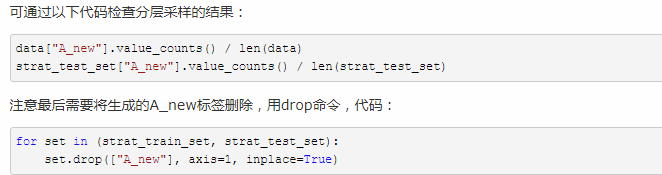
plt.show()#在jupyter中可以不加这条语句

在查看数据之前，最好先创建一下测试机，以免查看数据后的思维定式影响测试集选择。

在查看数据前，最好先创建一下测试集，以免查看数据后因为思维定势影响测试集的选择。  
一种方法是可以随机选择测试集，比如随机选择20%的数据作为测试集，但是这样当数据集更新时，测试集会变化，我们可以使用随机数处理。



另外，随机取数有可能丢失掉关键特征的分布。比如，有一个特征A对最终标签的贡献很大（两者之间相关性很强），  
那么我们也应该在测试集中保证A的分布符合原数据集的分布趋势。这时可以使用分层采样。



数据可视化及数据探索，在分好数据集后，对训练集数据进行观察，查找关联，进行属性组合实验，可以有效的选择属性。

有时候仅仅使用原有的特征数据的效果并不好，这是可以考虑一下将一些特征组合产生新的特征，比如将人数/家庭，得到每户的人数这样一个特征。如：

housing['population\_per\_household']=housing['population']/housing['households']

然后可以重新观察相关性。

4、数据的预处理

**Data Cleaning**

**Missing feature**： 1、get rid of the corresponding district(弃用该样本)

2、get rid of whole attribute(弃用整个属性)

3、set the value to some value(zero, the mean, the median)

对于第三种：可以使用SimpleImputer对象来替换缺失数据

**Handling Text and Categorical Attributes：**将文本，categories转换为数值

**Custom Transforms:** 属性转换，将某些属性合并为一个属性，从而减少属性

**Feature Scaling：**There are two common ways to get all attributes to have the same scale: min-max scaling （*normalization*） and standardization.

min-max scaling (*normalization*)：首先减去最小值，然后再除以最大值与最小值之差。归一化，在0~1区间

Standardization: 首先减去均值，然后除以方差。它范围是不确定的，但是在神经网络中要求在0~1范围。其优点是对异常值不敏感，而scale: min-max scaling对异常值就比较敏感。

**Transformation Pipelines:** 数据转化要经历许多步骤，有一定的顺序。

缺失数据处理，属性组合，feature scaling

Select and Train a model

根据数据属性与标签的关联选择训练模型

Training and Evaluating on the Training Set

要注意的是：在非常确定所选择的模型有效之前，一般不希望使用test集来，验证模型，从而避免了解测试集后针对测试集修改模型，造成模型有一定的数据集偏好。这算是：先保证Ein很小，在保证Ein和Eout之差的绝对值很小。

因此一般讲训练集就分为两个部分，一部分训练，一部分用于初步的模型验证，从而在训练集中就行Ein很小以及Ein和Eout之差的绝对值很小，这两个要求。

Better Evaluating Using Cross-Validation

使用交叉验证法：将训练集分为很小的训练集和验证集

现有方便的方法：使用sklearn 中的crass-validation feature.

Fine-Tune my model

One way to do that would be to fiddle with hyperparameters manually, until you find a great combination of hyperparameters.手动调参

Instead you should get Scikit-Learn’s GridSearchCV to search for you. 借助scikit-learn

1. GridSearchCV
2. RandomSearchCV

Ensemble Methods

分析最好的模型和他们的误差

# Look at the big picture

## 1.1 Frame the problem

明确问题，分析需求。搞清楚研究的问题是不是机器学习可以解决的问题，是的话，属于哪种机器学习类型？适合采用哪种机器学习方法？

## 1.2 Select a performance measure

选择模型评价方式，均方误差（MSE），均方根误差（RMSE）等等

## 1.3 Check the assumptions

列出和验证假设，因为这些假设是简化模型的基础，同时也是保证模型与实际情况相吻合。

# Get data

## 2.1 Create workspace

就是搭建编程环境，从而方便程序实现

## 2.2 download data

方式很多，爬虫，下载，传感器收集等等。

爬虫：利用urllib。

## 2.3 take a quick look of the data structure

使用pandas可以方便的加载数据，并查看。

使用matplotlib可以方便的图形化数据，如创建每个attribute的条形统计图

## 2.4 Create a test set

创建测试集，要在清洗数据之前，创建出测试集，从而避免知道测试集后，做出某些针对测试集的设定。

使用numpy的random可以方便的划分数据集，

划分测试集要注意一下几个问题：

1. 保证每次运行时，划分的数据集一样（通过numpy的random.seed()可以实现）
2. 当有数据更新时，保证原来的数据划分不变，而新数据按照划分要求划分（通过hashlib，借助identifier可以实现）
3. 原始数据集中的数据是具有按照一定比例的分级或者分层特性，划分后，每个自己也要具有类似的比例分层或者分级（通过sklearn.model\_selection中的 StratifiedShuffleSplit可以实现）

# Discover and visualize the data to gain the insight

## 3.1 Visualizing data

通过pandas库中的对象的plot绘制数据的散点图

## 3.2 Looking for Correlations

绘制属性的关联图，发现属性之间的关系。应用corr()方法

## 3.3 Experimenting with attribute combinations

进行属性组合尝试，从而简化数据集的属性，如房间数、房屋数，这两个就可以组合为单位房屋的房间数。

# Prepare the data for machine learning

注意构造函数，来处理数据，从而提高复用性，以及逻辑清晰度

## 4.1 data cleaning

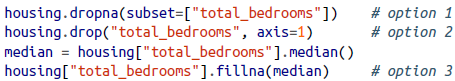
数据清洗，主要是处理缺失数据。

有三种处理思路：

1）Get rid of the corresponding districts.

2）Get rid of the whole attribute.

3）Set the values to some value (zero, the mean, the median, etc.).



利用pandas库中的对象的方法可以很方便的处理。

通过SimpleImputer对象来处理，但是只能处理数值数据的缺失

from sklearn.impute import SimpleImputer

imputer = SimpleImputer(strategy="median")

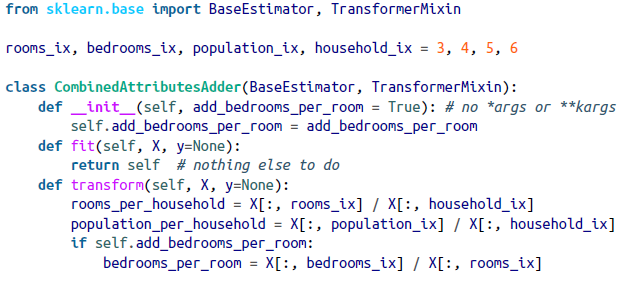
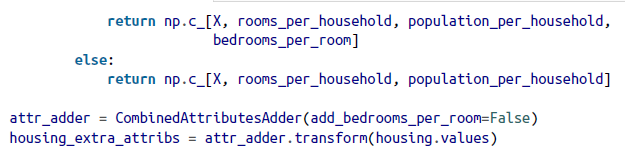
## 4.2 Handling Text and Categorical Attributes

将文本属性，转化为数值属性，从而便与处理。

利用LabelEncoder可以很方便的将文本标签转化为数值标签。

## 4.3 Custom transformer

定制属性：将数据的属性进行操作，组合属性，从而简化数据属性，或者提高属性的影响。

since Scikit-Learn relies on duck typing (not inheritance), all you need is to create a class and implement three methods: fit() (returning self), transform(), and fit\_transform(). You can get the last one for free by simply adding TransformerMixin as a base class. Also, if you add BaseEstimator as a base class (and avoid \*args and \*\*kargs in your constructor) you will get two extra methods (get\_params() and set\_params()) that will be useful for automatic hyperparameter tuning.  

## 4.4 Feature scaling

特征缩放主要归为两种方法：

min-max scaling (*normalization*)：首先减去最小值，然后再除以最大值与最小值之差。归一化，在0~1区间。（MinMaxScaler ）

Standardization: 首先减去均值，然后除以方差。它范围是不确定的，但是在神经网络中要求在0~1范围。其优点是对异常值不敏感，而scale: min-max scaling对异常值就比较敏感。（StandardScaler）

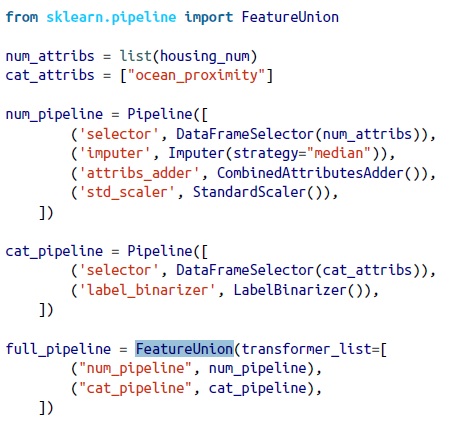
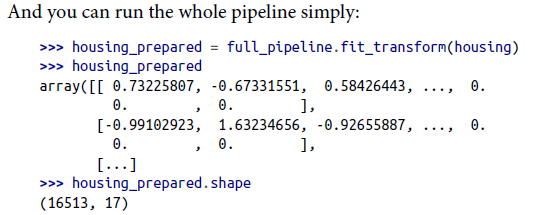


## 4.5 Transformation pipeline

数据转换管道：

通过创建转换管道，预处理数据，完成对数据的清洗（缺失数据处理），文本标签转换，定制属性，特征缩放等

单独的一个pipeline只能单独处理数值属性或者单独处理文本属性（即文本转化为数值），通过FeatureUnion()将两类pipeline组合起来。

# 5. Select model and train it

在分析了问题，获取到了数据，并初步分析探索了数据，创建了训练集和测试集，并且编写了一个pipeline（数据转换管道）来预处理数据。那么现在就准好了机器学习算法所需要的数据了

## 5.1 Training and evaluating on the training set

直接从sklearn库中导入选择的算法模型来训练。

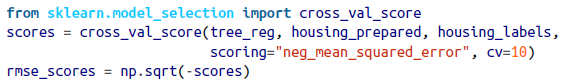
如；



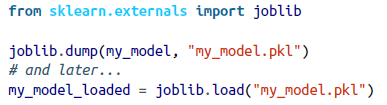
通过，可以分析均方误差，以及均方根误差。

## 5.2 Better Evaluation Using Cross-Validation

使用交叉验证的方法来提高训练精度，即将训练集分为很多小的子训练集，随机选择这些子集，一部分用于训练，一部分用于validation。



通过如下的代码可以存储训练好的模型，以及加载模型。



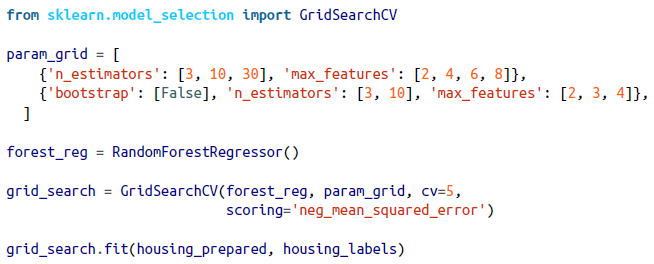
# 6. Fine-tune model

模型参数调节，通过调节模型参数可以优化模型。

主要有两种方式： 1）手动调整参数；2）调用Scikit-Learn's 中的GridSearchCV或者RandomizedSearchCV自动调整。

## 6.1 Grid search

告诉GridSearchCV 哪些超参数以及对应参数尝试哪些值，它就会自动组合评估这些参数及其值。



## 6.2 Randomized Search

在每次迭代中对每个超参数随机选择某个值，并对超参数进行随机组合。这种方式：能够搜索更多的参数组合，为了协调最优超参数的搜索，可以通过控制迭代次数来控制计算代价。

## 6.3 Ensemble Methods

组合方法：Another way to fine-tune your system is to try to combine the models that perform best.

通过组合多个模型来提高系统的精度（分类，预测）

## 6.4 Analyze the Best Models and Their Errors

反映各个属性的对预测精度的相对重要性

# 7. Present solution

## 7.1 Evaluate Your System on the Test Set

在对训练模型微调后，已经获得一个相对较好的模型，因此可以考虑在测试集上来验证模型了。

有时在训练集上通过交叉验证法的到的模型误差很小，但是在测试集上误差却很大，当这种情况出现时，不要去修改超参数来是模型在测试集上有很好的表现，这样做只是自欺欺人而已。

# 8. Launch, monitor, and maitain systerm

要注意监控系统的现场表现，在失效时及时警报。这对及时捕获突然的崩溃和性能退化是非常重要的。