这一段预测速度的程序，是用实际测得的目标物体的运动距离、给的激励电流、目标物体的质量三个参数来预测到达终点的速度。实际上是gridsearch函数部分承担了大部分工作。我选了（）和（）两个函数、将xx和xx作为参数进行网格搜索。特征选择的部分运用了一些技巧，动了一些脑筋。本来最初选的表征电流参数的值是电流的最大值，但后来修改为电流对时间的积分值，电流上升下降过程对速度的影响也能够得到描述。这就好像说推荐算法，我们预测某一个客户的喜好，我只看他买的东西不够全面，更有效的做法是看他浏览的商品是什么。所以我认为往往选特征比选参数或算法要更值得研究。

智能车项目

用的是增强学习目前最常用的一种算法，增强学习有两类。举例子来说，一类用于解决自动驾驶通过奖励或惩罚汽车采取的动作训练汽车产生正确的策略，我的输入是一个事件模型，起点终点时间限制，路径也是有限的几种，通过simulator也就是模拟器把这个模型运行起来自动产生一系列transition，传递给使用Q-learning算法的学习器通过迭代的办法学习最优策略。另一类是棋牌博弈类的，最初的输入量使用transition，比如每次给出一个落子位置，建模器modeler再根据这个牌面和规则建模，模型传递给规划器，规划器学习产生最优的策略。特点是状态空间极大。

该项目包含了一个学习器agent、模拟器simulator、环境模型中生成了道路、路上来往车辆

图像分类 深度学习

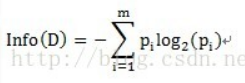
刚刚学完CNN的理论部分，才安装完TensorFlow、申请了亚马逊云计算服务。CNN在提取图像特征上比较突出的特点是，通过feature map和局部感受野的方法大大减少了所要学习的参数。1000\*1000大小的图像，10\*10的卷积核，那么一个隐层神经元的感受野大小就是10\*10，且只与这100个像素单元连接，这样一种卷积得到的一层作为一个Feature Map认为提取了图像的一种特征。实际上这个feature map里只有100个权值参数因为所有神经元的卷积核是一样的，权值共享。这样通过增加feature map增加特征的提取，10个feature map则有1000个需要训练的权值参数。（考虑每个神经元的偏置部分。所以卷积核的权值个数需要加1 ）

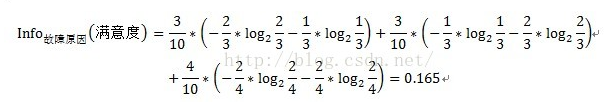
S2层每个单元的4个输入相加，乘以一个可训练参数，再加上一个可训练偏置。结果通过sigmoid函数计算。可训练系数和偏置控制着sigmoid函数的非线性程度。如果系数比较小，那么运算近似于线性运算，亚采样相当于模糊图像。

* 一种卷积核，一种移动步长 → 确定一种映射方式，提取到一种特征
* 下采样层也叫池化层，其具体操作与卷基层的操作基本相同，只不过下采样的卷积核为只取对应位置的最大值、平均值等（最大池化、平均池化），并且不经过反向传播的修改。可看作是模糊滤波器，起到二次特征提取的作用。隐层与隐层之间空间分辨率递减，而每层所含的平面数递增，这样可用于检测更多的特征信息。

房价预测：决策树

如何选取分裂点：选择信息增益最大的，直观解释就是对目标变量影响最大的因素信息增益越大。





先用numpy pandas 和 ipython的display函数看了一下数据的大致范围，最大最小均值四分位数等等，还可以用seaborn做简单的可视化；用3σ方法检测异常数据并删除，然后分割数据集80%为训练集20%为测试集。使用sklearn的decisiontree函数，最大深度设了4个并生成学习曲线，得到该参数的最佳取值。用r2score决定系数评估模型预测效果。

使用了kfold函数产生一个K折交叉验证，并使用网格搜索gridsearch函数将决策树的maxdepth参数从1-10 全部运行一遍，并返回grid.cv\_result。该部分就相当于网格搜索法和k折交叉验证共同作用把我们感兴趣的参数在已有数据集下全部尝试运行了一遍，返回最优参数。

薪资预测

data = pd.read\_csv("census.csv")

display(data.head(n=2))

data.shape

data.drop('income', axis = 1)

倾斜的数据（skewed data）

比如统计居民收入，假设一种分布是10%的人口占据了90%的财富值，那么如果不对这种倾斜进行处理，我们提取到的数据的特征就不是样本整体的影响，而是会放大这10%样本的特征其余90%则会相对被削弱，也就是说穷人数据反映出的特征会被淹没掉，即便样本数量多但占数值量过小可能会被作为0值而体现不出任何特征。所以归一化处理倾斜数据的意义就在于减小极大极小值对数据的负面影响，常用的方法就是log转换。在数据上面施加一个缩放并不会改变数据分布的形式。

# 对于倾斜的数据使用Log转换

skewed = ['capital-gain', 'capital-loss']

features\_raw[skewed] = data[skewed].apply(lambda x: np.log(x + 1))

把非数字的特征转换成数字量：“独热编码”的方法

比如样本的某项特征是“资产种类”，该特征包含三种取值，地产、公司、股票，拆解成三个特征取值为0 1

这个项目中所用的模型评价标准是f-score，这个值是一个关于查全率和查准率的函数，包含一个β系数调节更侧重哪一个参数。

Boosting：

在我所假设的模式下，我的预测结果中的样本分布情况与真实样本分布的不同，两者间的偏差我们定义为误差。

我给个假设是有股票资产的全部收入高于50k，结果有60%的样本符合这一假设，那么误差就为40%，这样的一个假设做为一个弱学习器，