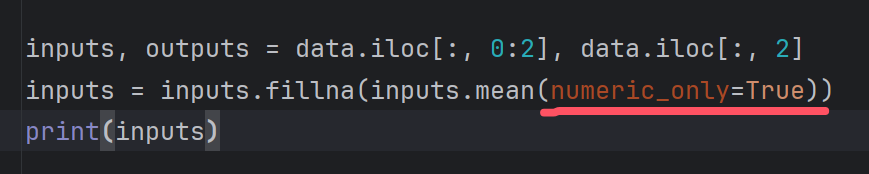
数据处理

1. 使用pandas包中的read\_csv函数可以读取CSV中的原始数据集。
2. 使用DataFrame类下的iloc函数可以对数据集进行切片，先是行再是列。
3. 使用fillna(x.mean())可以将mean中的数据缺省值以列平均值填充。此处因为pandas版本太高，运行之后报错，卡了一会（见4）。还是有条警告：

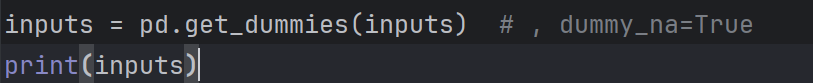
FutureWarning: The default value of numeric\_only in DataFrame.mean is deprecated. In a future version, it will default to False. In addition, specifying 'numeric\_only=None' is deprecated. Select only valid columns or specify the value of numeric\_only to silence this warning.

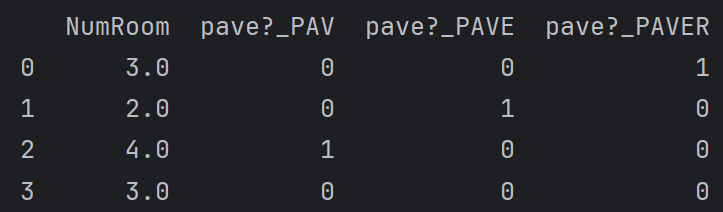
经查询，是因为没有显式设置numeric\_only参数导致的，当numeric\_only参数的值为True时，只会对**数值型列**计算均值，当numeric\_only参数的值为False时，这个操作会对**所有列**计算均值。

1. 事实上，刚刚所说的版本太高并不是问题，问题是我没有设置numeric\_only参数的值，当我更新版本后，在mean()中加入numeric\_only=True程序可以正常运行。总结出一个经验：**遇到报错可以降低版本看看。**



1. pandas可以将类别值和离散值分离成一个独立的列，使用get\_dummies可以将非数值类型分别成一个独立的列。原数据及运行结果如图所示。如果想要显示N/A缺省值的列，可以设置参数dummy\_na=True

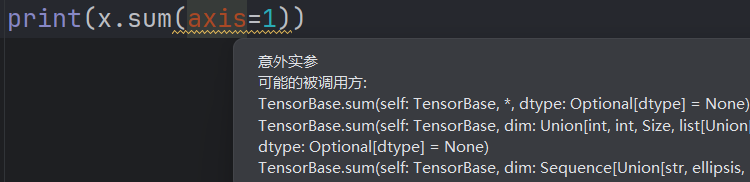




1. 将数值类型转化为张量格式。使用上次学习的tensor可以对数值进行操作，但是我们需要先把数值转化为张量类型。

线性代数

1. 向量与矩阵在pytorch框架中用tensor表示，如上次所学。值得注意的是：尽管单个向量的默认方向是列向量，但在表示表格数据集的矩阵中，将每个数据样本作为矩阵中的行向量更为常见。**向量是一阶张量，矩阵是二阶张量。**
2. y = x 并不能是传统意义上的“把x赋值给y”，而是“y是x的引用”。要想做到“把x赋值给y”，需要使用y = x.clone()。
3. 设x, y都是矩阵，x \* y并不是矩阵相乘，而是两个矩阵的Hadamard积。要想得到矩阵叉乘的结果，应该使用torch.mm(x, y)，想要得到矩阵点乘的结果，使用torch.dot(x, y)
4. 可以使用x.sum()函数计算张量x内所有元素的和。使用x.sum(axis=0)可以求得每一列的和。使用x.sum(axis=1)可以求得每一行的和。本质上是一种降维。轴0和轴1分别是行和列。但是我发现加上**axis=**会警告，**改为dim=之后**程序功能不变，并且没有警告。据此我推测书上可能使用得不太规范，Tensor类求和应该使用dim参数。另外，x.mean()可以求平均值，同样也有axis参数。使用keepdims=True可以保持维数不变。

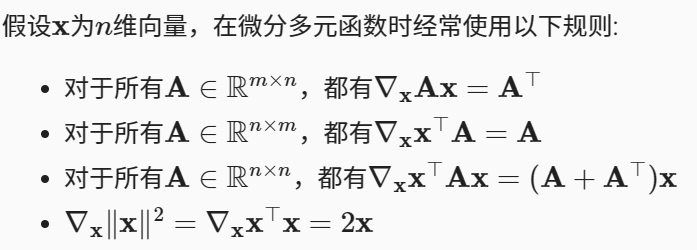


1. 使用torch.mv(A, x)可以表示**Ax**这个矩阵与向量的积。其中A的轴1元素个数应当与x的轴0元素个数相同。经试验，不同会导致报错并退出程序。
2. 学习了范数的概念，知道了Lp范数、L1范数和L2范数和计算方式。直接用norm()函数计算出的是L2范数。
3. 有趣的细节：
   1. 对于任意张量x，len(x)代表的是x的轴0的长度（元素个数）。
   2. 向量进行Hadamard积运算或进行除法时，如果维数不够会自动广播。

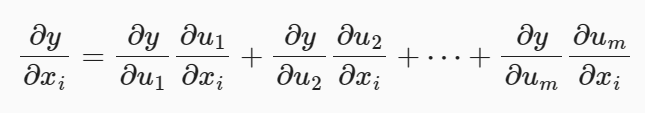
微积分

1. 使用matplotlib在jupyter notebook中绘制函数图像。看了半天没看懂这段代码怎么把图像画出来的。听学长说这个**数据可视化**是一块很大的内容，有空再学，目前不是重点。
2. 了解了偏导数的定义。计算某一个值x\_i时，可以把其他变量看作常数。
3. 梯度：函数f(x)相对于x的梯度是一个包含n个偏导数的向量。

<https://www.bilibili.com/video/BV1a541127cX/>通过该视频了解了梯度。



1. 链式求导法则：就是复合函数求导。然后多个复合函数内同意个变量x\_i可以通过直接相加得到导数。



自动微分

1. 自动微分可以加快求导，还能使系统能够随后**反向传播**梯度。 这里，反向传播意味着跟踪整个计算图，填充关于每个参数的偏导数。
2. 当requires\_grad\_=True时，Pytorch会在反向传播时计算并存储每个张量的梯度。

未来

1. 学到这里，我认为我需要对多元函数、偏导数、梯度更进一步了解，要不然进行不下去。所以我打算先学习一下数学知识。<https://www.bilibili.com/video/BV1UB4y1b7ac>。
2. 学完数学知识之后，学习自动微分、概率。