# 训练词向量流程：

## 1.data\_read\_csv.py :

将最开始的输入数据转换成约定的格式，如果原始文件格式满足要求，则不需要运行此py文件。

## 2.data\_load.py:

读取上一步的输出文件，将上一步的输出文件每一行存成一维列表，整个文件就存储成了二维列表

## 3.dic\_build\_bookedid:

将第二步的二维列表转换成一维的列表，也就是把二维列表中所有的元素按顺序放到一个一维列表中，并转换成对应的索引

## 4.model\_build\_bookid.py

构建tensorflow 神经网络模型并进行训练，对于feed 进神经网络的数据就是调用batch\_generate\_bookid 来生成的

## 5.batch\_generate\_bookid:

为神经网络生成batch数据，

最终训练好的房源向量存储到了"./data/word2vec.txt"中， "./data/dic\_word\_index" 是一个词典{key:房源ID，value：index}

"./data/dic\_index\_word"是一个词典{key:index,value:房源ID}这是我们最终需要的三个结果

# .py文件具体介绍：

## 1. data\_read\_csv.py :

输入：原始文件路径

功能：整理原始文件的格式，每一行代表一个会话（session），用“|”分割出用户浏览的房源ID（lu\_id）和最终的预定房源ID（booked\_lu\_id） （预定的房源由“ ”分隔 ）

输出：经过整理后的文件路径

## 2.data\_load.py:

输入：经过整理后的文件路径

功能：读取文件的一行，转成一维列表，列表的最后一个元素存booked\_id(如果不存在存储成-1)

输出：一个二维列表

## 3.dic\_build\_bookid.py:

输入：第二步的输出——二维列表，代码中的vocabulary\_size 是我们最终需要的房源ID个数，是一个 超参数。使用时只需要调整这个参数

功能：将输入的房源映射成一个索引index

输出：也就是把二维列表中所有房源id 放到一个一维列表中，并存成索引

## 4.model\_build\_bookid.py:

输入：vocabulary\_size = 200000 所有不同房源ID的个数

batch\_size = 129 #要与你的skipWindows相匹配

embedding\_size = 128 # 生成向量维度.

skip\_window = 2 # 左右窗口.

num\_skips = 2 # 同一个keyword产生label的次数.实际上还要再加一个book-id

num\_sampled = 64 # 负样本抽样数.

建议以上参数只调整vocabulary\_size 其他参数不动

功能：训练训练房源向量

输出：最终的房源向量存储到"./data/wored2vec.txt"中，形状是（vocabulary\_size,房源向量的维度）

## 5.batch\_generate\_bookid.py:

输入：只有几个相对的路径，建议不要修改

功能：为神经网络生成batch\_data

输出：返回batch\_data