数据分析训练题

## 问题描述

(实体识别)

现有一个Amazon的众多商品的数据记录文件(Amazon.csv)，同时有Google对众多商品的数据库记录文件(Google.csv)。如何将两者的数据匹配起来？

## 问题解决

1. 分割文本，得到有效Tokens

**"token" 其实就是指的一个不区分大小写单词，或者说是以空格等等的符号分割的一个个字符串。"tokens"则是token的列表。一个token允许在tokens中存在多次（在tokens中存在多次意味着该单词在原文中出现了多次）。**在读入csv文件之后，我们应分别为Amazon和Google的每条数据求它的tokens。

**同时像"is"、"of"这样的token，我们对今后的token分析没有贡献。**读入stopwords.txt以删除这些token。

二.求TF-IDF

求TF：对于**一个**tokens中的一个token，

**TF(token)=该token在tokens中的出现次数/tokens中的总token数**

在Python中，可以用一个字典来存放一条数据的所有tf值

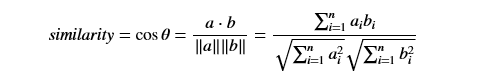
求IDF：对于**所有**tokens中的一个token，

**IDF(token)=tokens(token*列表*)*的个数*/出现过该token的tokens(token*列表*)*的个数***

求TF-IDF:TF-IDF(token)=TF(token)\*IDF(token)

三.求余弦相似度

把每一个token理解成一个维度，那么对于一条Amazon的数据和一条Google的数据，我们就可以计算二者TF-IDF的余弦值。可以把余弦值看作一种相似度。余弦值越大就意味着二者越相似。



四.与最优匹配比较，求出最佳阈值

Amazon\_Google\_perfectMapping.csv记录着最优的匹配数据

## 优化算法

如果要求a,b的相似度，根据公式可知，即使我们还没确定b是哪一个，我们依然可以算出a的norm(即向量的模)，由此我们可以预先求出每条数据的norm。

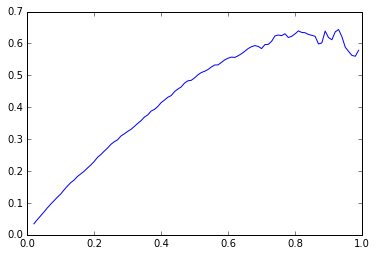
对于一个在Amazon数据集中的a，我们总需要遍历一遍Google数据集中的每条数据，那么我们可以考虑保存一个Amazon数据集的逆向索引以供使用。在之前，一个ID对应着一个tokens，一个tokens中含有众多token。逆向索引就是将一个token对应着多个ID的字典。

预处理出逆向索引与norms这两个字典，可以加速相似度的计算。

## 参考结果

最大精确率： 0.643835616438

最大精确率下的阈值：0.93



横坐标为阈值，纵坐标为精确率。