1：奇异值分解SVD的原理是怎样的，都有哪些应用场景

假设我们的矩阵A是一个m×n的矩阵，那么我们定义矩阵A的SVD为：A=UΣVT

其中U是一个m×m的矩阵，Σ是一个m×n的矩阵，除了主对角线上的元素以外全为0，主对角线上的每个元素都称为奇异值，V是一个n×n的矩阵。U和V都是酉矩阵，即满足UTU=I,VTV=I。V可以通过求解ATA的特征向量而得，U可以通过求解AAT的特征向量而得，Σ可以通过求解ATA或AAT的特征值开方得到

SVD可以用于PCA降维，来做数据压缩和去噪。也可以用于推荐算法，将用户和喜好对应的矩阵做特征分解，进而得到隐含的用户需求来做推荐。

2：funkSVD, BiasSVD，SVD++算法之间的区别是怎样的

funkSVD就是将矩阵M分解成PTQ，将矩阵分解转化成一个最优化问题，有时还加个正则项，利用梯度下降法求解， BiasSVD在 funkSVD的基础上加入用户偏差矩阵和商品偏差矩阵作修正，SVD++在 BiasSVD的基础上进行了改进，考虑用户的隐式反馈，加入了用户个性化的内容

3：矩阵分解算法在推荐系统中有哪些应用场景，存在哪些不足

对于推荐系统来说存在两大场景即评分预测（rating prediction）与Top-N推荐（item recommendation，item ranking）。评分预测场景主要用于评价网站，比如用户给自己看过的电影评多少分（MovieLens），或者用户给自己看过的书籍评价多少分（Douban）。矩阵分解技术主要应用于该场景。对于Top-N推荐，也可以在评分的基础上向用户推荐可能的高评分的item给客户。不足之处在于对数据稀疏和冷启动问题较为难办。

矩阵分解最大的不足之处在于只能捕捉线性关系，而捕捉不了隐含的非线性关系

4：item流行度在推荐系统中有怎样的应用

可以用在冷启动时，对新用户作基于流行度的推荐，向新用户推荐流行度高的商品，对于老用户，可以考虑高流行度对商品推荐的降权影响，在对用户相似度的计算时加上对流行度的修正，挖掘长尾，作更个性化的推荐

5：推荐系统的召回阶段都有哪些策略

召回阶段主要是从全量的商品库中得到用户可能感兴趣的一小部分候选集，基于协同过滤与基于内容的推荐算法都是可以的，因为是粗筛，所以算法的效率越快越好