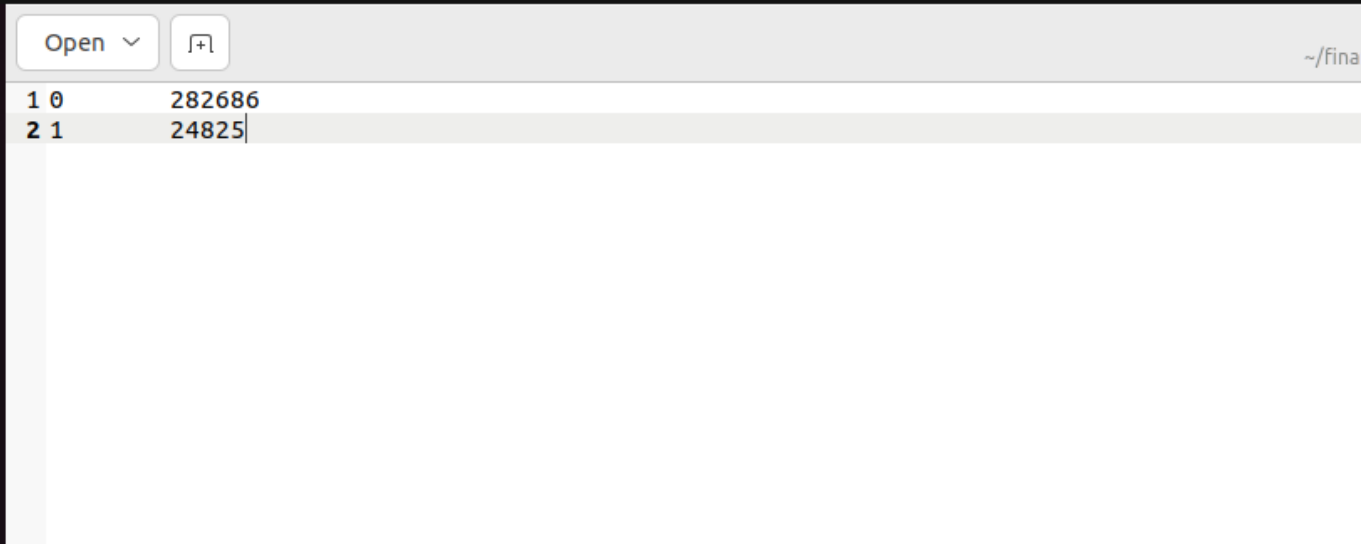


Task1

思路： 用TextInputformat对输入进行标准化，然后Task1.map的任务就是输出键值对<Target,1>,Task1.reduce的任务就是对输入的键值对的值进行求和即可。

结果截图：

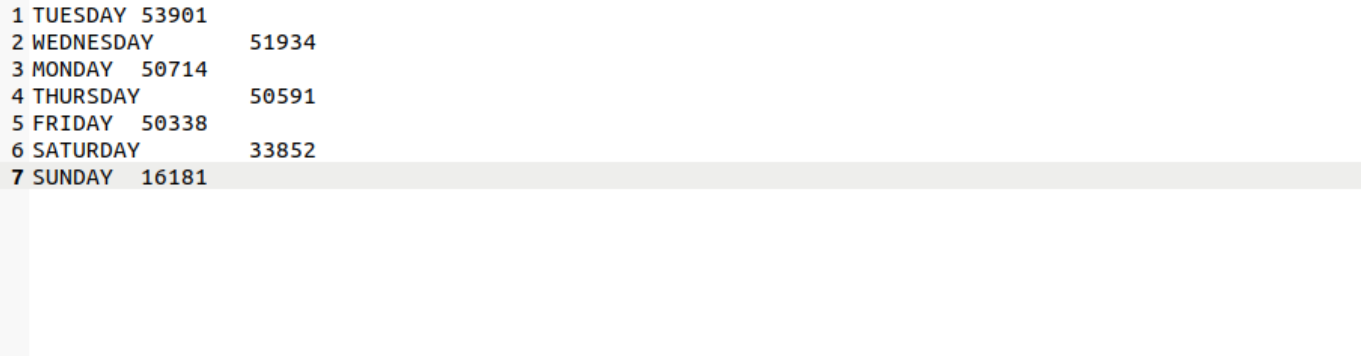


Task2

思路：

大致过程与Task1相同， Task2.map输出的键值对是<WEEKDAY_APPR_PROCESS_START,1>,Task2.reduce对拿到的键值对进行求和，并利用 Collections.reverseOrder()将元素逆序排列，最后在cleanup中输出排序好的结果。

结果截图



Task3

选择的模型

朴素贝叶斯

数据预处理

- 首先查看数据是否有缺失值：

```
OBS_30_CNT_SOCIAL_CIRCLE      1021
DEF_30_CNT_SOCIAL_CIRCLE      1021
OBS_60_CNT_SOCIAL_CIRCLE      1021
DEF_60_CNT_SOCIAL_CIRCLE      1021
DAYS_LAST_PHONE_CHANGE        1
```

因此，我们需要先处理缺失值：考虑到OBS_30_CNT_SOCIAL_CIRCLE等的含义，我采用了平均值填充NAN的方法。

- 删除高度相关的特征变量

因为我采用的模型是朴素贝叶斯，其假设便是各个特征之间没有任何关系，是相互独立的，因此要在数据预处理阶段删除高相关性的特征变量。根据其相关系数矩阵和特征变量的描述,最终决定删除：

```
"FLAG_EMP_PHONE", "FLAG_WORK_PHONE", "FLAG_PHONE", "REG_REGION_NOT_LIVE_REGION", "REG_
REGION_NOT_WORK_REGION", "LIVE_REGION_NOT_WORK_REGION", "OBS_30_CNT_SOCIAL_CIRCLE", "
OBS_60_CNT_SOCIAL_CIRCLE", "DEF_60_CNT_SOCIAL_CIRCLE", "DAYS_BIRTH", "NAME_INCOME_TYP
E"
```

这11个变量。

- 特征离散化

因为该数据集有一些连续的数值特征，而朴素贝叶斯当面对具有连续值特征的数据时，其性能可能不是最优的。因此我决定将这些特征离散化处理。具体来说，这些连续的特征值

有："AMT_CREDIT","AMT_INCOME_TOTAL","REGION_POPULATION_RELATIVE"。然后根据其四分位数进行离散：在MIN~25分位数，映射到1，25~50分位数，映射到2；50~75分位数，映射到3；75分位数~MAX，映射到4。

数据集的划分

我选择的随机划分，训练集和测试集大小为4：1

搭建模型

Task3Conf

该类主要获取Task3数据集的相关配置信息，通过读取提前写好的Task3.conf的数据集配置文件，获得：类的数量，每一种类的名称；特征的数量，特征的名称,并将其存入相关属性中。

Task3Train

该类根据训练集进行训练：首先在初始化时，通过Task3Conf类获取到数据集信息；然后在map任务中，我们输出两种键值对：第一种是<Target,1>;第二种是<Target#AttributeName#value,1>，也就是标签值#属性名#属性值。各Map节点输出的局部频度数据FYi和FxYi。而在reduce任务中，我们的任务将map节点输出的局部频度数据FYi和FxYij整合成全局的频度数据FYi和FxYi，具体做法就是简单地加和即可。

Task3TrainResult

该类通过读取Task3Train的输出文件，获取到全局的FYi和FxYij频度数据，保存到一个字典里，为测试做好铺垫。

Task3Test

该类实现测试功能，同时也计算模型的性能：accuracy。在初始化时，通过Task3Conf和Task3TrainResult类得到数据集信息和训练结果。在map任务中，对于输入每个测试数据，计算其最大可能属于哪个类别，所需的频度数据通过Task3TrainResult来获得；同时判断其预测值和真实值是否相同。因此map输出两种键值对：<Sample.ID,PredictTarget>和<"Accuracy","1#TrueOrFlase">。在reducer任务中，对于预测结果的键值对，我们直接输出即可；我们需要处理的是<"Accuracy","1#\$TrueOrFlase">这类键值对：通过"#"将其值拆开，然后对1求和，代表着总的样本数Total;对True求和，代表着分对的样本数，然后二者相除即可。

需要注意的是，由于在map的计算过程中牵扯到连乘，因此相应涉及到的变量其类型最好设置为BigInteger，否则极易溢出

Task3Driver

驱动类，进行任务的配置。总共要进行两个Job，第一个Job进行训练，第二个Job进行测试。通过Configuration()实例记录相关文件位置信息。

结果截图：

