物流线路分布统计与规划

为了更好的帮助您理解我做的工作，特编写本文档。

我和罗如瑜同学组队参加大学生大数据分析与挖掘竞赛，最初想使用比赛数据集完成本课程实践课题。我们一路走到决赛，11 月 23 日的课上我们向老师询问后老师同意我们每个人以自己在比赛中的工作作为实践课题内容。我做的工作是利用地址数据和运单数据分析线路分布。

我们十月份做的时候 hive、pig、spark 这些工具都还没学，我这部分主要是基于 python 和 MySQL 数据库完成。Python 通过 pymysql 库与 MySQL 数据库通信，从数据库中查询数据，查询结果经过 python 处理存储为本地 csv 文件或者 pandas DataFrame 的内容。为了和课程所学框架相配合，我对项目部分内容使用 Hive 和 HBase 进行复现。

一、数据处理

前面提到，比赛中我是用 MySQL 数据库做的，使用 Hive 进行局部复现。

创建数据库、创建表格和加载数据的代码。使用 HDFS 中存储的文件作为外部数据源向表格导入数据。

```
beeline -u jdbc:hive2://course-cluster-namenode-0:10000/default -n stu2017210017 -p 210017;
#创建数据库
create database logistics;
use logistics;
#创建地址数据表
create table address_A(`ID` string, `yundanID` string, type string, province string, city string, district string, `gpsX` double, `gpsY` double, `mainNode` int, sequence int) row format delimited fields terminated by ',';
#创建运单数据表
create table waybill_A(`ID` string, `yundanBianhao` string, `pickupTime` string, `arrivalTime` string, fang double, ton double, `truckID` string) row format delimited fields terminated by ',';
#加载地址数据到表格
load data inpath '/user/stu2017210017/data/address_A.txt' overwrite into table address_A;
#加载运单数据到表格
load data inpath '/user/stu2017210017/data/waybill_A.txt' overwrite into table waybill_A;
```

从运单数据表查询出客户所有运单 ID.

```
SELECT DISTINCT ID FROM waybill_A;
```

查询结果如图 1.

之后利用上一步查询已经得到的运单 ID 集合，将每一个运单 ID 替换下面语句的{}处，查询每个运单采用的路线。运单 247816 的结果如图 2.

```
SELECT type, province, city FROM address_A WHERE yundanID = {};
```

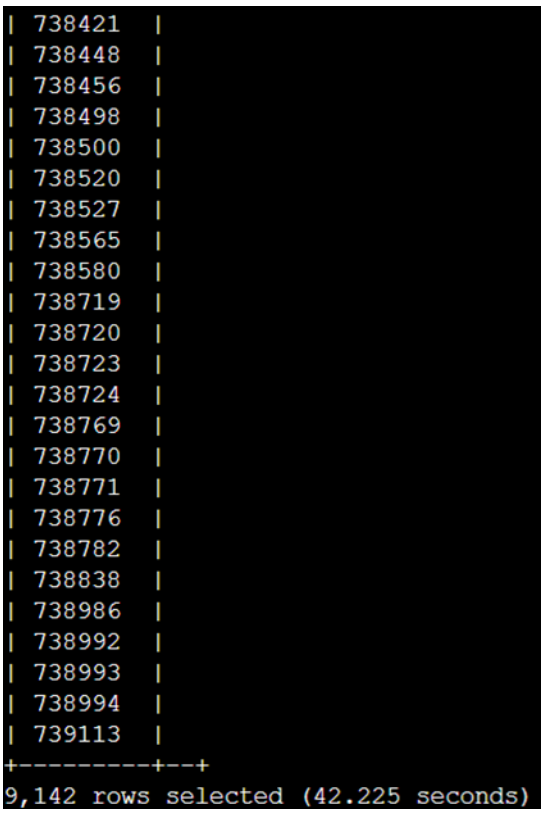

图1　　客户 A 运单数据所有运单 ID 集合

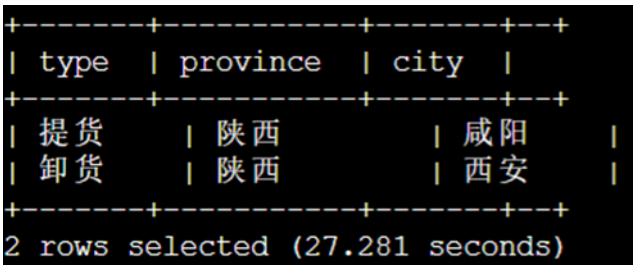

图2　　运单 247816 途经地点和操作

在项目代码中对应 findPositionSequence.py 文件的 databaseQuery 函数（23-46 行）。代码中定义 selectYundanID 变量对应查运单 ID 的代码，selectSequence 变量对应查每个运单途径城市的行为和顺序的代码。之后对查询结果的形式进行处理（查询结果是嵌套元组，组合成 336 行数据帧，每行有 2 列，第 1 列是运单 ID，第 2 列是运单途经节点的顺序和名称，结果存到 track_A.csv 中，如图 3。

Track_B.csv 是客户 B 的数据表，如图 4。在 hive 上我存储客户 B 的数据，创建对应数据表可以得到相似查询结果。在这里我无法使用 python 与 pyhive 结合的方式复现全部操作，原因有 1.因为查询和处理时间太长。在本地 mysql 数据

库每次查询只需要百分之一到十分之一秒量级下需要处理 2 分钟，hive 一次查询需要 20-40 秒，不知道批量查询会不会更快，但是处理时间还是非常可怕的（如果集群工作稳定，需要 2 小时以上）。2.需要在集群上装 pyhive 包，我没有管理员权限，无法装包。

1	2
247816,	"(('提货', '陕西,咸阳'), ('卸货', '陕西,西安'))"
247822,	"(('提货', '陕西,咸阳'), ('卸货', '陕西,西安'))"
247824,	"(('提货', '陕西,咸阳'), ('卸货', '陕西,西安'))"
247884,	"(('提货', '陕西,咸阳'), ('卸货', '宁夏,银川'))"
248004,	"(('提货', '陕西,咸阳'), ('卸货', '陕西,西安'))"
248005,	"(('提货', '陕西,咸阳'), ('卸货', '陕西,西安'))"
248610,	"(('提货', '陕西,咸阳'), ('卸货', '陕西,西安'))"
248611,	"(('提货', '陕西,咸阳'), ('卸货', '陕西,西安'))"
248612,	"(('提货', '陕西,咸阳'), ('卸货', '陕西,西安'))"
252821,	"(('提货', '陕西,咸阳'), ('卸货', '陕西,咸阳'))"
253678,	"(('提货', '陕西,咸阳'), ('卸货', '重庆,重庆'))"
254110,	"(('提货', '陕西,咸阳'), ('卸货', '重庆,重庆'))"
254111,	"(('提货', '陕西,咸阳'), ('卸货', '重庆,重庆'))"
254285,	"(('提货', '陕西,咸阳'), ('卸货', '陕西,汉中'))"
254294,	"(('提货', '陕西,咸阳'), ('卸货', '重庆,重庆'))"
254315,	"(('提货', '陕西,咸阳'), ('卸货', '重庆,重庆'))"
254316,	"(('提货', '陕西,咸阳'), ('卸货', '重庆,重庆'))"
254479,	"(('提货', '陕西,咸阳'), ('卸货', '陕西,西安'))"
254495,	"(('提货', '陕西,咸阳'), ('卸货', '陕西,西安'))"
254528,	"(('提货', '陕西,咸阳'), ('卸货', '陕西,西安'))"
254533,	"(('提货', '陕西,咸阳'), ('卸货', '陕西,西安'))"
254584,	"(('提货', '陕西,咸阳'), ('卸货', '陕西,汉中'))"
254680,	"(('提货', '陕西,咸阳'), ('卸货', '陕西,西安'))"
254737,	"(('提货', '陕西,咸阳'), ('卸货', '陕西,西安'))"
254836,	"(('提货', '陕西,咸阳'), ('卸货', '陕西,汉中'))"
255021	"(('提货', '陕西,咸阳'), ('卸货', '陕西,西安'))"

图3　　客户 A 所有运单路线键值对

1	2
247788,	"(('提货', '河北,邢台'), ('卸货', '安徽,合肥'))"
247790,	"(('提货', '河北,廊坊'), ('卸货', '山东,菏泽'))"
247804,	"(('提货', '浙江,宁波'), ('卸货', '浙江,衢州'))"
247806,	"(('提货', '浙江,宁波'), ('卸货', '浙江,衢州'))"
247807,	"(('提货', '浙江,宁波'), ('卸货', '浙江,宁波'))"
247817,	"(('提货', '安徽,马鞍山'), ('卸货', '安徽,马鞍山'))"
247818,	"(('提货', '安徽,马鞍山'), ('卸货', '安徽,合肥'))"
247820,	"(('提货', '安徽,马鞍山'), ('卸货', '安徽,合肥'))"
247837,	"(('提货', '安徽,马鞍山'), ('卸货', '江苏,常州'))"
247841,	"(('提货', '安徽,芜湖'), ('卸货', '湖北,武汉'))"
247843,	"(('提货', '安徽,芜湖'), ('卸货', '湖北,武汉'))"
247864,	"(('提货', '安徽,芜湖'), ('卸货', '安徽,安庆'))"
247885,	"(('提货', '安徽,合肥'), ('卸货', '江苏,南京'))"
247919,	"(('提货', '安徽,合肥'), ('卸货', '江苏,南京'))"
248008,	"(('提货', '浙江,宁波'), ('卸货', '浙江,宁波'))"
248012,	"(('提货', '浙江,宁波'), ('卸货', '河南,新乡'))"
248013,	"(('提货', '浙江,宁波'), ('卸货', '河南,新乡'))"
248014,	"(('提货', '河北,邢台'), ('卸货', '河北,邢台'))"
248015,	"(('提货', '河北,邢台'), ('卸货', '河北,邢台'))"
248060,	"(('提货', '江苏,苏州'), ('卸货', '江苏,扬州'))"
248068,	"(('提货', '安徽,芜湖'), ('卸货', '安徽,合肥'))"
248072,	"(('提货', '安徽,马鞍山'), ('卸货', '江苏,常州'))"
248075,	"(('提货', '安徽,马鞍山'), ('卸货', '江苏,镇江'))"
248108,	"(('提货', '陕西,宝鸡'), ('卸货', '陕西,宝鸡'))"
248125,	"(('提货', '安徽,芜湖'), ('卸货', '安徽,芜湖'))"
248152	"(('提货', '安徽,芜湖'), ('卸货', '安徽,芜湖'))"

图4　　客户 B 所有运单路线键值对

得到数据帧后，每一行是以运单 ID 为 key，路线为 value 的键值对。之后以路线为 key，1 为 value，groupByKey 操作得到每条路线对应的运单数。我没有使用 spark/scala 的 groupByKey 算子，使用的是 pandas.Series.value_counts()方法。对应 findPositionSequence.py 的 trackDistributionStatistics 函数的 86-88 行。处理得到历史线路分布，如图 5 图 6。客户 B 的数据集能得到类似结果图 7 图 8。

Index	track	freq	hot
0	陕西,咸阳(提货)->陕西,西安(卸货)	3514	0.983413
1	陕西,咸阳(提货)->河南,郑州(卸货)	856	0.935296
2	陕西,咸阳(提货)->陕西,咸阳(卸货)	581	0.907549
3	陕西,咸阳(提货)->新疆,乌鲁木齐(卸货)	499	0.893995
4	陕西,咸阳(提货)->甘肃,兰州(卸货)	367	0.861247
5	陕西,咸阳(提货)->陕西,渭南(卸货)	292	0.831707
6	陕西,咸阳(提货)->宁夏,银川(卸货)	217	0.786188
7	陕西,咸阳(提货)->陕西,榆林(卸货)	188	0.761215
8	陕西,咸阳(提货)->河南,洛阳(卸货)	185	0.758294
9	陕西,咸阳(提货)->陕西,宝鸡(卸货)	178	0.751194
10	陕西,咸阳(提货)->青海,西宁(卸货)	175	0.748021
11	陕西,咸阳(提货)->甘肃,平凉(卸货)	175	0.748021

图5　　客户 A 历史线路分布

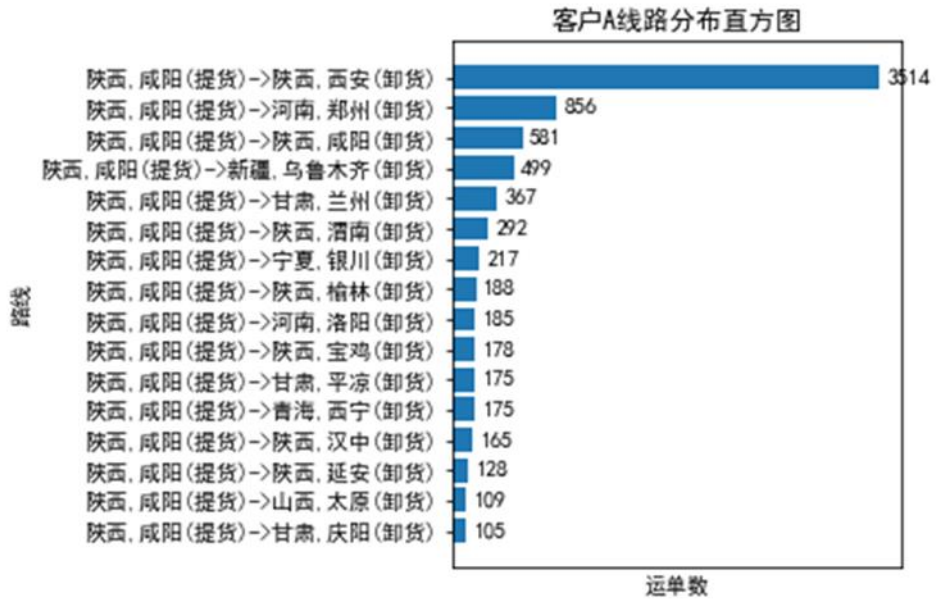

图6　　客户 A 线路分布直方图

Index	track	freq	hot
0	陕西,咸阳(提货)->陕西,西安(卸货)	3514	0.983413
1	陕西,咸阳(提货)->河南,郑州(卸货)	856	0.935296
2	陕西,咸阳(提货)->陕西,咸阳(卸货)	581	0.907549
3	陕西,咸阳(提货)->新疆,乌鲁木齐(卸货)	499	0.893995
4	陕西,咸阳(提货)->甘肃,兰州(卸货)	367	0.861247
5	陕西,咸阳(提货)->陕西,渭南(卸货)	292	0.831707
6	陕西,咸阳(提货)->宁夏,银川(卸货)	217	0.786188
7	陕西,咸阳(提货)->陕西,榆林(卸货)	188	0.761215
8	陕西,咸阳(提货)->河南,洛阳(卸货)	185	0.758294
9	陕西,咸阳(提货)->陕西,宝鸡(卸货)	178	0.751194
10	陕西,咸阳(提货)->青海,西宁(卸货)	175	0.748021
11	陕西,咸阳(提货)->甘肃,平凉(卸货)	175	0.748021

图7　客户 B 历史线路分布

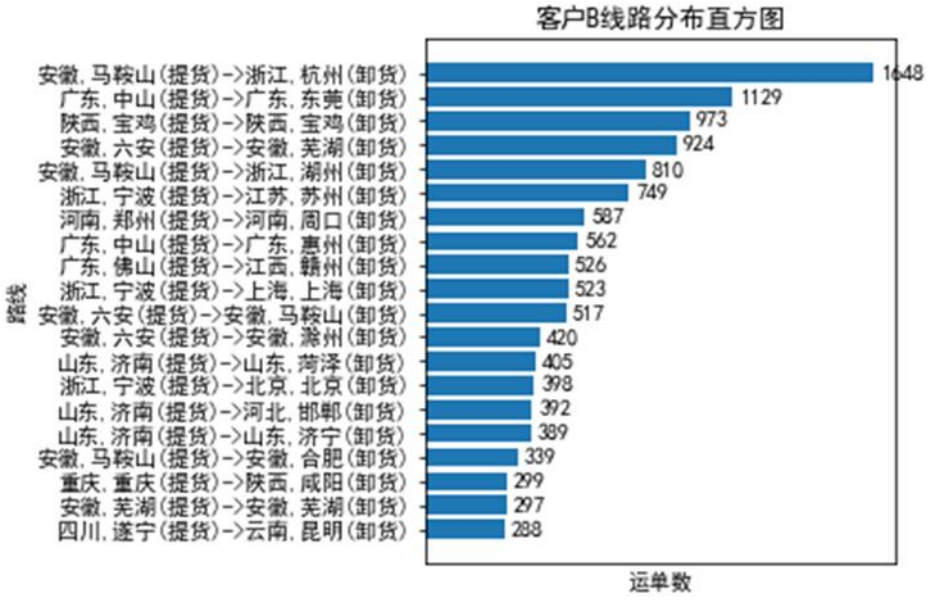

图8　客户 B 线路分布直方图

二、模型求解

为了根据订单需求向司机推荐路线，我构建模型。

(一)指标定义

1. 热门度

热门度（hot）根据客户的历史线路分布求解。在求出每条历史路线对应的运单数之后，我们得到一张表格，如表 1。设表格第 2 列运单数构成向量

$$waybill = [t_1, t_2, \ldots, t_N] \tag{1}$$

式中 N 代表客户历史线路总数。热门度 hot 定义式如下

$$ttt = \ln(waybill + 1) \tag{2}$$

$$hot = sigmoid(ttt - 0.5 * \max(ttt)) \tag{3}$$

式中 sigmoid 函数的表达式是

$$sigmoid(\mathrm{x}) = \frac{1}{1 + e^{-x}} \tag{4}$$

首先定义向量 ttt 是考虑路线对应的运单数相差极大，比较热门的路线可以有超过一千个运单采用，而很多路线对应运单数小于 10，甚至如果求出的最短路线在历史线路分布中不存在，对应运单数是 0，所以定义 ttt 是向量 waybill 加 1 取对数。之后根据 sigmoid 函数的性质，为了使热门度取值分布在[0, 1]之间，将 sigmoid 函数的输入调整到范围[-0.5*max(ttt), 0.5*max(ttt)]，这样热门路线对应的热门度很高，达到 0.95 以上；冷门路线热门度很低，小于 0.05.

模型考虑该指标的原因与反馈率有关，历史上较多司机选择的路线可能具有更加著名，路况更好或者更安全的特点，模型在推荐路线时要考虑司机们的选择的合理性。

表1　客户历史线路分布示例

线路	运单数
A	3514
B	188
C	367
…	…

2. 路线长度度量

路线长度度量(short)的定义式如下。

$$\mathrm{short(X)} = \begin{cases} sigmoid(-2\dfrac{distance - \mu 1}{\sigma 1}), \text{如果备选路线多于一条} \\ 1, \text{如果备选路线只有} X \text{这一条} \end{cases} \tag{5}$$

式中 distance 表示如果备选路线多于 1 条，使用最短路线算法求出每条路线的长度构成的向量$\mathrm{distance} = [d_A, d_B, d_C, \ldots]$。μ1表示 distance 的平均值，σ1表示 distance 的标准差。这样做是为了把 distance 向量标准化（normalization）。之后作为 Sigmoid 函数的自变量时之所以取相反数是因为让长度短的路线得分更高。乘 2 使函数值分布更离散。如果备选路线只有一条，那么这条路线就是最短的，short 指标得分为 1.

这样定义路线长度度量，可以让指标得分在[0, 1]之间分布。

模型考虑该指标的原因是减少路线长度，提高运输效率。

3. 成本度量

定义重量长度积的概念，因为物流行业的运输成本与货物重量和运输路程之积正相关，所以考虑到成本，规划线路时要使重量长度积尽量小。

成本度量的定义式如下

$$cost(\mathrm{X}) = \begin{cases} sigmoid\left(-2\dfrac{weightDistance - \mu_2}{\sigma_2}\right), \\ \text{如果备选路线数大于 1} \\ 1, \text{ 如果备选路线只有 X 这一条} \end{cases} \tag{6}$$

式中 weightDistance 代表重量长度积。计算方法是，以客户 A 运单需求：咸阳提货，依次在西安卸货 20 吨、西宁卸货 40 吨、安康卸货 60 吨为例。

这样货车在咸阳共提货 120 吨，载 120 吨货物走咸阳到西安的 20.094 公里，卸货 20 吨后，载 100 吨货物走西安到西宁的 854.134 公里，卸货 40 吨后，载 60 吨货物走西宁到安康的 942.513 公里，所以路线 1 的重量长度积为

$$\begin{aligned} &weightDistance(1) \\ &= 120 \times 20.094 + 100 \times 854.134 + 60 \times 942.513 \\ &= 144375.46\text{（吨・公里）} \end{aligned} \tag{7}$$

按照同样的方法计算其它备选路线的重量长度积，放到一个向量里，计算向量的均值μ_2和标准差σ_2，代入公式（7）得到所有路线的成本度量。

模型考虑该指标的原因是让重量长度积尽量小，成本尽量低。

下面将三项指标加权求和，得到线路推荐度度量值。

4. 线路推荐度度量值

公式

$$\mathrm{score(X)} = a\ hot(X) + b\ \mathrm{short}(X) + c\ \mathrm{cost}(X) \tag{8}$$

式中a，b和c是模型的三个参数，其中a是路线热门度的权重，模型中设置为 0.2；b是路线长度度量的比重，模型中设置为 0.3；c表示路线成本度量的权重，模型中设置为 0.5.三个参数满足关系$a + b + c = 1$。

如果最短路线算法求出的路线不在历史线路分布中，对应历史运单数为 0，$hot(X) = sigmoid(-0.5 * max(ttt))$。从公式可以看出，路线的总分是 1 分。模型在推荐路线时综合考虑路线热门度、长度和成本因素，同时推荐得分最高的几条路线供司机选择。

(二)推荐流程

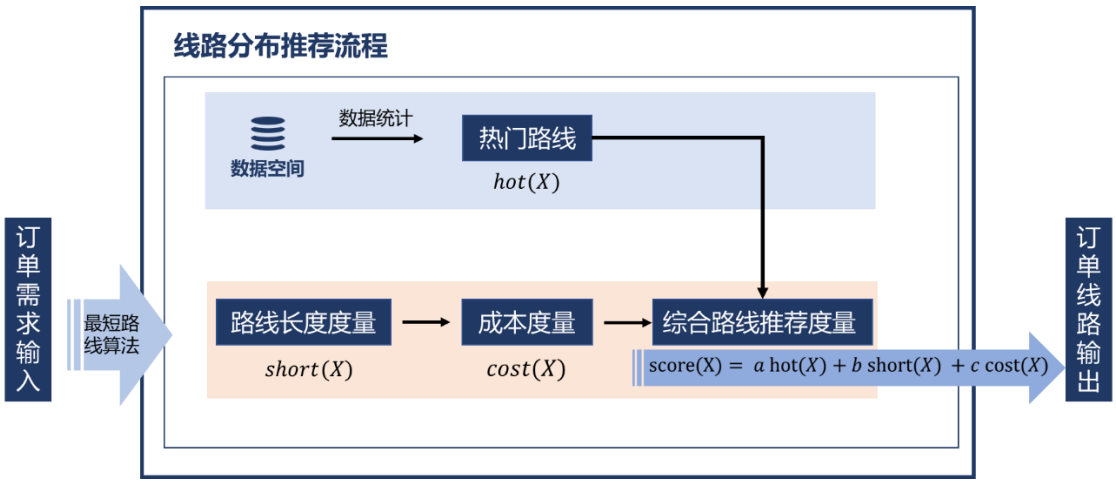

图9　线路分布推荐流程图

推荐流程是首先统计历史线路分布，计算线路热门度。之后使用最短路线算法根据输入的订单需求规划路线，计算长度度量和成本度量。最后对三项指标加权求和计算推荐度量值，向司机推荐度量值最高的几条路线。

举例，客户 A 的需求是在咸阳提货，西安卸货 20 吨、西宁卸货 40 吨、安康卸货 60 吨。第一步模型已经统计客户 A 的历史线路分布，直方图如图 6，从前面的分析我们知道客户 A 对咸阳到西安路线的需求最大，这条路线的热门度最高。第二步从第一步求出的历史线路分布中筛选出符合需求的路线 1。1 个运单采用这条路线，近似长度是 1817 千米，重量长度积是 144375，推导过程见公式（7）。

表2　路线 1 信息

路线编号	路线	对应运单数	近似长度	路线热门度	重量长度积
1	陕西,咸阳(提货)->陕西,西安(卸货)->青海,西宁(卸货)->陕西,安康(卸货)	1	1817 km	0.0326	144375

第三步算法推荐重量长度积最小的三条路线，如图 10。可以看到路线 2、3、

4 的重量长度积均小于路线 1。

路线 2：陕西咸阳（提货）->陕西西安（卸货）->陕西安康（卸货）->青海西宁（卸货）
近似长度：1144 km，重量长度积：58201.5
路线 3：陕西咸阳（提货）->陕西安康（卸货）->陕西西安（卸货）->青海西宁（卸货）
近似长度：1236 km，重量长度积：69138.1
路线 4：陕西咸阳(提货)->陕西安康(卸货)->青海西宁(卸货)->陕西西安(卸货)
近似长度：1998 km，重量长度积：97752.4

图10　路线 234 信息

第四步对四条路线按照三项指标评分，结果如表 3 所示。看到路线 23 长度较小，长度度量值较大；路线 234 重量长度积较小，成本度量值较大。推荐路线时综合考虑热门度，长度和成本，所以路线 23 总分较高。模型最后推荐这两条路线供司机选择。

表3　路线评分结果

路线	热门度	长度度量	成本度量	总分	总分排名
1	0.0326	0.2192	0.0628	0.1037	4
2	0.0166	0.8718	0.8552	0.6925	1
3	0.0166	0.8145	0.7699	0.6326	2
4	0.0166	0.1065	0.4305	0.2505	3

三、结果存储

我在 hbase 中创建 analysis_result 表，定义三个列族：raw_info,存储路线原始信息，即路线的运单数、路线路由、长度、重量长度积；score,存储路线评分情况，包括热门度、长度度量和成本度量三项指标的得分和总分；rank，存储路线总分排名和推荐指数。表格内容如表 4。

表4　向 HBase 存储的表格内容

Row	Raw_info				score				rank	
	Waybills 运单数	Distance 长度	Track 路由	weightDistance 重量长度积	Hot 热门度	Short 长度度量	Cost 成本度量	Total_score 总分	Realrank 排名	Star 推荐指数
Route1	1	1817	Xianyang, Shaanxi(Load)->Xi\'an, Shaanxi(Unload)->Xining, Qinghai(Unload)->Ankang, Shaanxi(Unload)	144375	0.0326	0.2192	0.0628	0.1037	4	1
Route2	0	1144	Xianyang, Shaanxi(Load)->Xi\'an, Shaanxi(Unload)->Ankang, Shaanxi(Unload)->Xining, Qinghai(Unload)	58201.5	0.0166	0.8718	0.8552	0.6925	1	4
Route3	0	1236	Xianyang, Shaanxi(Load)->Ankang, Shaanxi(Unload)->Xi\'an, Shaanxi(Unload)->Xining, Qinghai(Unload)	69138.1	0.0166	0.8145	0.7699	0.6326	2	4
Route4	0	1998	Xianyang, Shaanxi(Load)->Ankang, Shaanxi(Unload)->Xining, Qinghai(Unload)->Xi\'an, Shaanxi(Unload)	97752.4	0.0166	0.1065	0.4305	0.2505	3	2

存储数据的代码写在 hbasecommands.txt 文件，使用命令 hbase shell hbasecommands.txt 批量插入数据和查询存储内容。

查看存储结果。

（1） 查看总分最高的路线 2 的得分情况。

get 'analysis_result', 'route2', {COLUMN=>'score'}

```
COLUMN                                   CELL
 score:cost                              timestamp=1606743687965, value=0.8552
 score:hot                               timestamp=1606743687863, value=0.0166
 score:short                             timestamp=1606743687917, value=0.8718
 score:total_score                       timestamp=1606743688000, value=0.6925
4 row(s) in 0.0910 seconds
```

（2） 查看路线 3 的全部信息。

get 'analysis_result', 'route3'

```
COLUMN                                   CELL
 rank:Realrank                           timestamp=1606743688635, value=2
 rank:star                               timestamp=1606743688672, value=4
 raw_info:distance                       timestamp=1606743688276, value=1236
 raw_info:track                          timestamp=1606743688193, value=Xianyang, Shaanxi(Load)->Ankang, Shaanxi(Unload)->Xi'an, Shaanxi(Unload)->Xining, Qinghai(Unload
                                         )
 raw_info:waybills                       timestamp=1606743688227, value=0
 raw_info:weightDistance                 timestamp=1606743688347, value=69138.1
 score:cost                              timestamp=1606743688510, value=0.7699
 score:hot                               timestamp=1606743688416, value=0.0166
 score:short                             timestamp=1606743688471, value=0.8145
 score:total_score                       timestamp=1606743688554, value=0.6326
10 row(s) in 0.1640 seconds
```

（3） 扫描全表。

scan 'analysis_result'

```
ROW                                      COLUMN+CELL
 route1                                  column=rank:Realrank, timestamp=1606743687567, value=4
 route1                                  column=rank:star, timestamp=1606743687607, value=1
 route1                                  column=raw_info:distance, timestamp=1606743687225, value=1817
 route1                                  column=raw_info:track, timestamp=1606743687092, value=Xianyang, Shaanxi(Load)->Xi'an, Shaanxi(Unload)->Xining, Qinghai(Unload)-
                                         >Ankang, Shaanxi(Unload)
 route1                                  column=raw_info:waybills, timestamp=1606743687186, value=1
 route1                                  column=raw_info:weightDistance, timestamp=1606743687250, value=144375
 route1                                  column=score:cost, timestamp=1606743687452, value=0.0628
 route1                                  column=score:hot, timestamp=1606743687281, value=0.0326
 route1                                  column=score:short, timestamp=1606743687410, value=0.2192
 route1                                  column=score:total_score, timestamp=1606743687490, value=0.1037
 route2                                  column=rank:Realrank, timestamp=1606743688033, value=1
 route2                                  column=rank:star, timestamp=1606743688156, value=4
 route2                                  column=raw_info:distance, timestamp=1606743687759, value=1144
 route2                                  column=raw_info:track, timestamp=1606743687652, value=Xianyang, Shaanxi(Load)->Xi'an, Shaanxi(Unload)->Ankang, Shaanxi(Unload)-
                                         >Xining, Qinghai(Unload)
 route2                                  column=raw_info:waybills, timestamp=1606743687697, value=0
 route2                                  column=raw_info:weightDistance, timestamp=1606743687805, value=58201.5
 route2                                  column=score:cost, timestamp=1606743687965, value=0.8552
 route2                                  column=score:hot, timestamp=1606743687863, value=0.0166
 route2                                  column=score:short, timestamp=1606743687917, value=0.8718
 route2                                  column=score:total_score, timestamp=1606743688000, value=0.6925
 route3                                  column=rank:Realrank, timestamp=1606743688635, value=2
 route3                                  column=rank:star, timestamp=1606743688672, value=4
 route3                                  column=raw_info:distance, timestamp=1606743688276, value=1236
 route3                                  column=raw_info:track, timestamp=1606743688193, value=Xianyang, Shaanxi(Load)->Ankang, Shaanxi(Unload)->Xi'an, Shaanxi(Unload)-
                                         >Xining, Qinghai(Unload)
 route3                                  column=raw_info:waybills, timestamp=1606743688227, value=0
 route3                                  column=raw_info:weightDistance, timestamp=1606743688347, value=69138.1
 route3                                  column=score:cost, timestamp=1606743688510, value=0.7699
 route3                                  column=score:hot, timestamp=1606743688416, value=0.0166
 route3                                  column=score:short, timestamp=1606743688471, value=0.8145
 route3                                  column=score:total_score, timestamp=1606743688554, value=0.6326
```