DL PREDICTOR SYSTEM TESTCASES AND IMPLEMENTATION

July 16, 2020

Categories

- Testing Data Preparation

- Testcases Design Rules

- Testcases Designed.

- Testcases Implementation

Test Preparation: Data Pre-processing

- Prepare predictions from ES
  - A data sample is taken as the right one.
  - Uckey (native,b6le0s4qo8,3G,g_m,6,CPC,57)
  - Uckey's attributes (7 attrs):
    - m, si, t, g, a, pm, r
  - Predictions of days (e.g. 10 days)
  - 4 price categories * (0 – 24) hours

- Prepare fact data from Hive tables
  - Uckey's transformation to attrs:
  - Uckey.split(",") => df.m, df.si, df.t, df.g, df.a, df.pm, df.r
  - Columns: Uckey, count_array, hour, day, bucket_id
  - Some data samples are taken as the capture shows.

- ES Prediction's transformation Sample

{'a': u'5', 'g': u'g_f', 'm': u'magazinelock', 'daily_count': 16138.536705628001,
 'si': u'01', 'r': u'74', 'uckey': u'magazinelock,01,WIFI,g_f,5,CPM,74', 't': u'WIFI', 'day':
'2020-01-30', 'pm': u'CPM'}

- Fact data's transformation sample:

[Row(uckey=u'magazinelock,01,3G,g_m,4,CPM,', bucket_id=260, count_array=[u'1:5'],
hour=17, day=u'2020-01-17', houly_count=5, m=u'magazinelock', si=u'01', t=u'3G',
g=u'g_m', a=u'4', pm=u'CPM', r=u'40')] # Fact data sample.

```
+-------------------------+---------+-----------+----+----------+
|uckey                    |bucket_id|count_array|hour|day       |
+-------------------------+---------+-----------+----+----------+
|magazinelock,01,2G,,,CPM,|476      |[1:14]     |5   |2020-01-19|
|magazinelock,01,2G,,,CPM,|476      |[1:2]      |0   |2020-01-22|
|magazinelock,01,2G,,,CPM,10|113    |[1:1]      |2   |2020-01-18|
```

Testcases Design Rules

- **Rule 1:** Date range based. Starting date and ending date are required.

- **Rule 2:** Attributes conditions based. 7 attributes are considered: m, si, t, g, a, pm, r

- **Rule 3:** Focus more on system level testing with comparisons of total counts.

- **Rule 4:** Focus more on gender and age.

- **Rule 5:** Try to cover more testcases with combinations of the attributes.

Testcases Designed

- Queries:
- Date range: 2020-01-30 – 2020-02-08, 10 days.
- [
- None,
- {"g": "g_m"},
- {"g": "g_f"},
- {"a": "4"},
- {"a": "6"},
- {"g": "g_m", "a": "4"},
- {"g": "g_f", "a": "2"},
- {"g": "g_f", "a": "6"},
- {"r": "40"}, #beijing
- {"r": "49"}, #shenyang
- {"r": "80"}, #cities group
- {"m": "native"},
- {"pm": "CPC"},
- {"a": "6", "g": "g_f", "m": "native"},
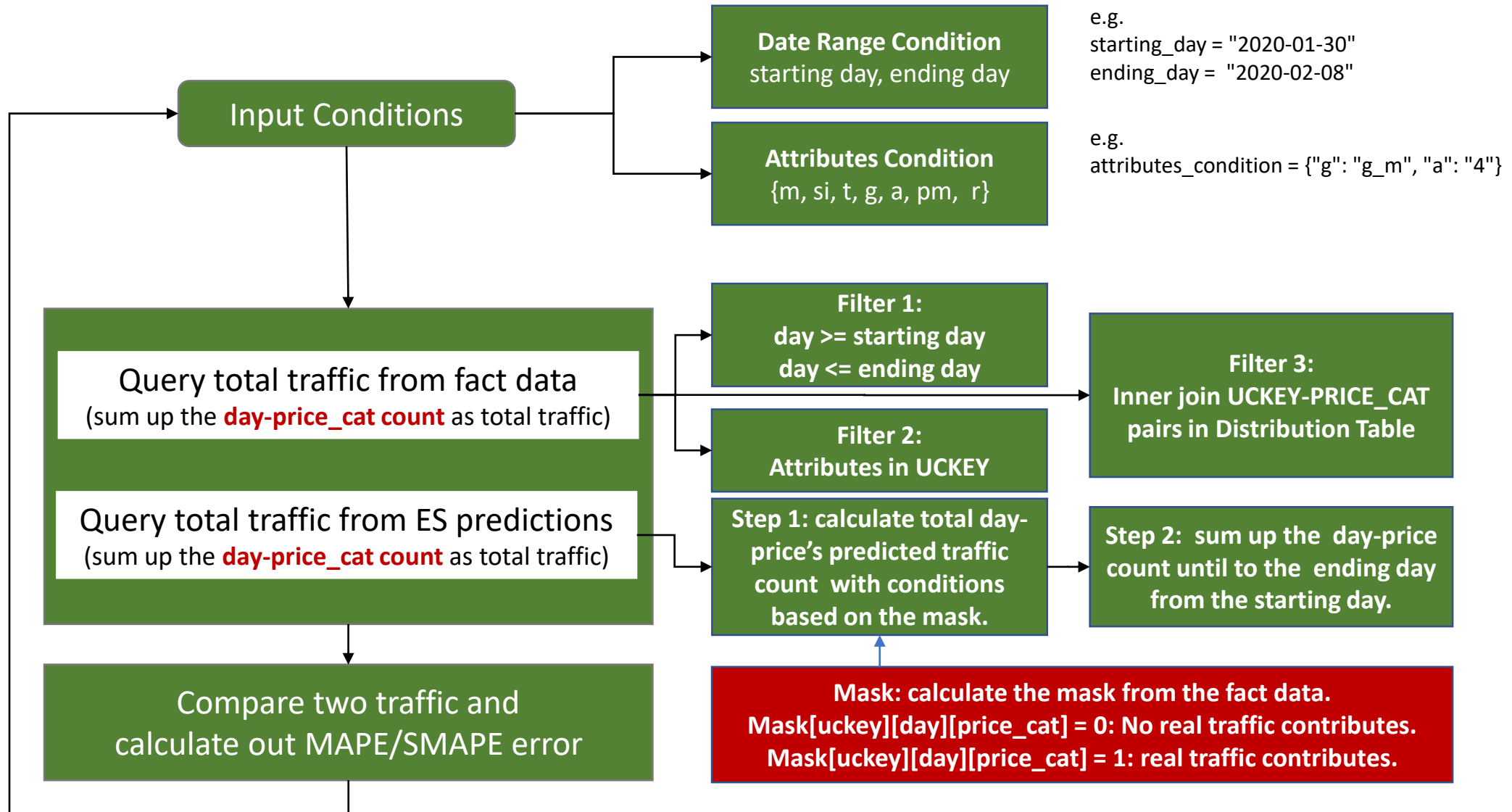- {"a": "6", "g": "g_f", "r": "40", "m":"native"}
-]

Testcases Implementation Design 1: Total traffic queries based.

(some uckeys have no real traffic on a specific predicted day will return non-zero predictions which will be counted as predicted traffic.)
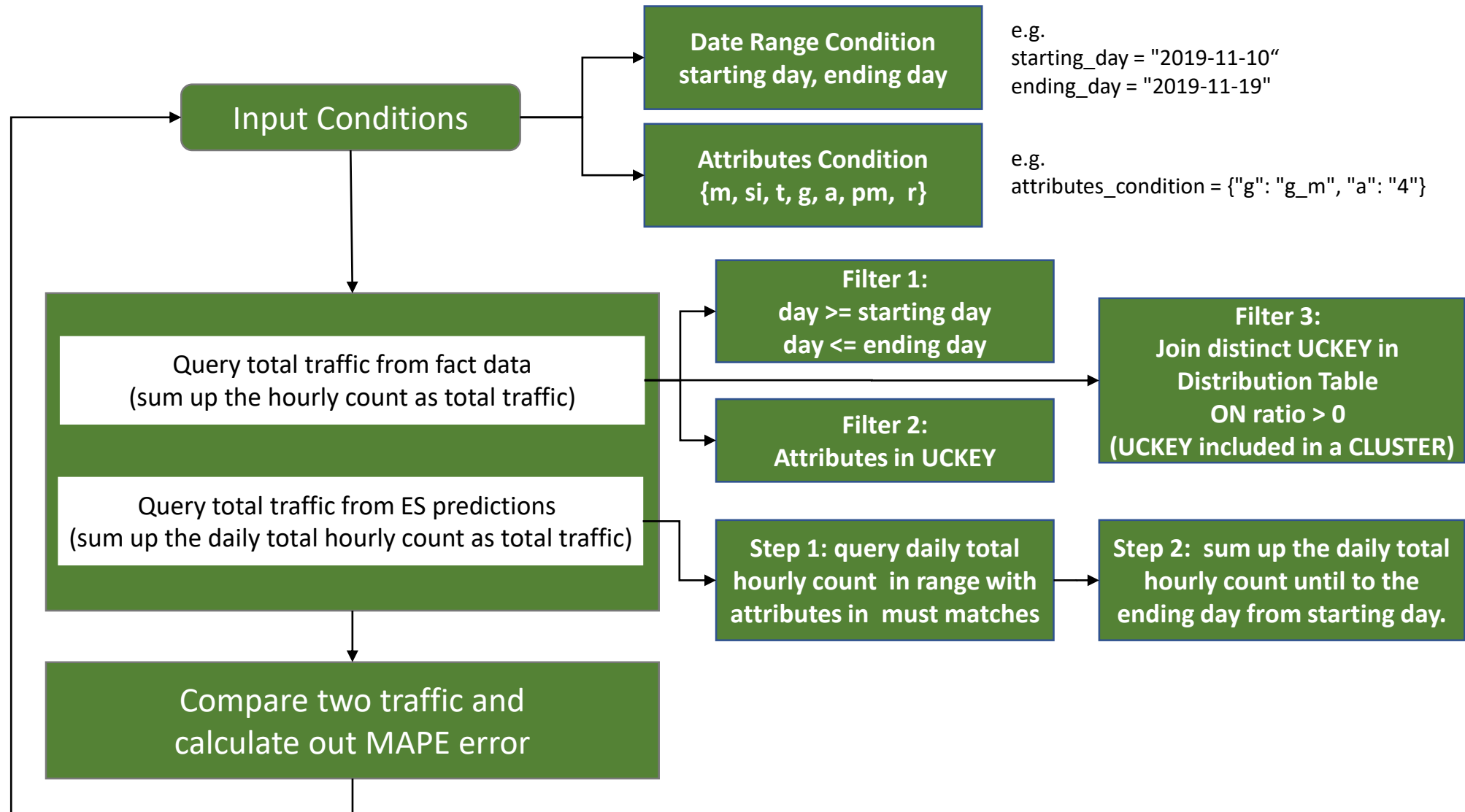
Testcases Implementation Design 2: Mask of Uckey's daily price_cat's traffic based.
(some uckeys have no real traffic on a specific predicted day will return non-zero predictions which will not be counted as predicted traffic.)

Input Conditions

Date Range Condition
starting day, ending day

e.g.
starting_day = "2020-01-30"
ending_day = "2020-02-08"

Attributes Condition
{m, si, t, g, a, pm, r}

e.g.
attributes_condition = {"g": "g_m", "a": "4"}

Query total traffic from fact data
(sum up the **day-price_cat count** as total traffic)

Filter 1:
day >= starting day
day <= ending day

Filter 3:
Inner join UCKEY-PRICE_CAT pairs in Distribution Table

Filter 2:
Attributes in UCKEY

Query total traffic from ES predictions
(sum up the **day-price_cat count** as total traffic)

Step 1: calculate total day-price's predicted traffic count with conditions based on the mask.

Step 2: sum up the day-price count until to the ending day from the starting day.

Compare two traffic and calculate out MAPE/SMAPE error

Mask: calculate the mask from the fact data.
Mask[uckey][day][price_cat] = 0: No real traffic contributes.
Mask[uckey][day][price_cat] = 1: real traffic contributes.

starting_day	ending_day	attributes_condition	factdata_traffic	predicted_traffic	mape_error	smape_error
1/30/2020	2/8/2020	None	2,074,430,999	2,535,608,646	0.22	0.2
1/30/2020	2/8/2020	{'g': 'g_m'}	1,104,508,445	1,450,558,908	0.31	0.27
1/30/2020	2/8/2020	{'g': 'g_f'}	399,345,309	529,468,219	0.32	0.28
1/30/2020	2/8/2020	{'a': '4'}	422,124,618	549,151,476	0.30	0.26
1/30/2020	2/8/2020	{'a': '6'}	200,182,919	234,781,873	0.17	0.16
1/30/2020	2/8/2020	{'a': '4', 'g': 'g_m'}	307,306,502	399,646,723	0.3	0.26
1/30/2020	2/8/2020	{'a': '4', 'g': 'g_f'}	113,549,883	149,257,579	0.31	0.27
1/30/2020	2/8/2020	{'a': '6', 'g': 'g_f'}	34,329,212	42,753,793	0.25	0.22
1/30/2020	2/8/2020	{'r': '40'}	22,866,136	33,711,642	0.47	0.38
1/30/2020	2/8/2020	{'r': '49'}	13,753,700	19,911,000	0.45	0.37
1/30/2020	2/8/2020	{'r': '80'}	163,768,651	189,864,162	0.16	0.15
1/30/2020	2/8/2020	{'m': 'native'}	1,478,198,677	1,366,274,677	0.08	0.08
1/30/2020	2/8/2020	{'pm': 'CPC'}	1,292,735,873	1,208,712,654	0.06	0.07
1/30/2020	2/8/2020	{'a': '6', 'm': 'native', 'g': 'g_f'}	23,197,262	18,206,967	0.22	0.24
1/30/2020	2/8/2020	{'a': '6', 'r': '40', 'm': 'native', 'g': 'g_f'}	230,585	201,232	0.13	0.14
				AVG	**0.24**	**0.22**
				AVG of {a, g}	**0.29**	**0.25**

Testcases Implementation

Table 1: Testing result on system level
- Use attributes based system queries.
- Not uckeys related.
- Included 0 daily traffic's predictions.

Table 2: Testing result on uckeys level
- Use traffic count of uckey's daily-price_cat.
- Uckeys related with mask calculated.
- Excluded 0 daily traffic's predictions.

Conclusion:
- Excluded 0 traffic' prediction has lower error.
- The avg MAPE error is promoted to 8%.
- The avg MAPE error on (gender, age) is 6%.

starting_day	ending_day	attributes_condition	factdata_uckeys_used_count	es_uckeys_used_count	factdata_traffic	predicted_traffic	mape_error	smape_error
1/30/2020	2/8/2020	None	91389	91389	2074430999	2006835420	0.03	0.03
1/30/2020	2/8/2020	{'g': 'g_m'}	44174	44174	1104508445	1113559788	0.01	0.01
1/30/2020	2/8/2020	{'g': 'g_f'}	36798	36798	399345309	380166765	0.05	0.05
1/30/2020	2/8/2020	{'a': '4'}	15632	15632	422124618	408187969	0.03	0.03
1/30/2020	2/8/2020	{'a': '6'}	12334	12334	200182919	181568009	0.09	0.1
1/30/20020	2/8/2020	{'a': '4', 'g': 'g_m'}	8142	8142	307306502	302580485	0.02	0.02
1/30/2020	2/8/2020	{'a': '4', 'g': 'g_f'}	6763	6763	113549883	105398093	0.07	0.07
1/30/2020	2/8/2020	{'a': '6', 'g': 'g_f'}	5239	5239	34329212	30152069	0.12	0.13
1/30/2020	2/8/2020	{'r': '40'}	1164	1164	22866136	24753777	0.08	0.08
1/30/2020	2/8/2020	{'r': '49'}	1256	1256	13753700	14737937	0.07	0.07
1/30/2020	2/8/2020	{'r': '80'}	1747	1747	163768651	153487664	0.06	0.06
1/30/760020	2/8/2020	{'m': 'native'}	59626	59626	1478198677	1329760046	0.1	0.11
1/30/026020	2/8/2020	{'pm': 'CPC'}	36781	36781	1292735873	1201026355	0.07	0.07
1/30/2020	2/8/2020	{'a': '6', 'm': 'native', 'g': 'g_f'}	3314	3314	23197262	17472622	0.25	0.28
1/30/2020	2/8/2020	{'a': '6', 'r': '40', 'm': 'native', 'g': 'g_f'}	48	48	230585	189145	0.18	0.2
						AVG	**0.08**	**0.09**
						AVG of {a, g}	**0.06**	**0.06**