# Forecast Autocorrections: Step Description

Forecast Autocorrections step is aimed to adjust output forecast values in order to prevent provided most inadequate results to downstream systems.

## Code Realization Requirements

The code should be created in a form of a SAS macro set on SAS VIYA platform.

z

# Input Data

The initial data for the forecast flag definition are listed below. These tables should be present in the system before this step for example in STG or in DDS area.

• IN\_PRODUCT (see data requirements)

• IN\_LOCATION (see data requirements)

## ACC\_AGG\_HYBRID\_FORECAST\_<tgt\_type>

Hybrid Forecast values that is disaggregated to sku/location/customer/distribution channel level and disaccumulated to daily granularity.

|  |  |
| --- | --- |
| **ACC\_AGG\_HYBRID\_FORECAST\_<tgt\_type>** | |
| Column Name | Description |
| **PRODUCT\_LVL\_ID<m>** | Level m =**OUT\_PRODUCT\_LVL**of product hierarchy |
| **LOCATION\_LVL\_ID<n>** | Level n =**OUT \_LOCATION\_LVL** of location hierarchy |
| **CUSOMER\_LVL\_ID<k>** | Level k =**OUT \_CUSTOMER\_LVL** of customer hierarchy |
| **DISTR\_CHANNEL\_LVL\_ID<l>** | Level l =**OUT \_DISTR\_LVL** of distribution channel hierarchy |
| **PERIOD\_DT** | Period of sales (OUT\_TIME\_LVL) |
| **SEGMENT\_NAME** | Name of segment that was linked to a pair product/location within VF Project (can be missing) |
| **VF\_FORECAST\_VALUE** | VF forecast value |
| **DEMAND\_TYPE** | ‘promo’ or ‘regular’ – type ML model is used to forecast |
| **ASSORTMENT\_TYPE** | new, or old |
| **ML\_FORECAST\_VALUE** | ML forecasted value |
| **HYBRID\_FORECAST\_VALUE** | HYBRID forecast value |
| **ENSEMBLE\_FORECAST\_VALUE** | Ensemble Forecast Value |
| **FORECAST\_SOURCE** | Forecast Source |

## DEMAND\_RESTORED\_<tgt\_type>

Demand information regarding the past till last known day of the history.

|  |  |
| --- | --- |
| **DM.DEMAND\_RESTORED**\_<tgt\_type> | |
| Column Name | Description |
| **PRODUCT\_ID** | Product ID (the lowest level of the product hierarchy) |
| **LOCATION\_ID** | Location ID |
| **CUSTOMER\_ID** | Customer ID |
| **DISTR\_CHANNEL\_ID** | Distribution Channel ID |
| **PERIOD\_DT** | Date of sales (calendar day) |
| **TGT\_QTY\_R** | Restored demand |
| **PROMO\_FLG** | 1|promo event was active  0|no promo event |
| **TGT\_QTY** | Total sales in units per day (w/o returns) |
| **STOCK\_QTY** | Stock qty (BOP) |
| **PROMO\_FLG** | 1|promo event was active  0|no promo event |
| **DEFICIT\_FLG1** | 1|primary deficit occurred  0|no primary deficit |
| **DEFICIT\_FLG2** | 1|secondary deficit occurred  0|no secondary deficit |

## FORECAST\_FLAG

Product/location lifecycle history containing the following fields is used as an input:

|  |  |
| --- | --- |
| **DM.FORECAST\_FLAG** | |
| Column Name | Description |
| **PRODUCT\_ID** | Product ID (the lowest level of the product hierarchy) |
| **LOCATION\_ID** | Location ID |
| **CUSTOMER\_ID** | Customer ID |
| **DISTR\_CHANNEL\_ID** | Distribution Channel ID |
| **PERIOD\_START\_DT** | Period start date |
| **PERIOD\_END\_DT** | Period end date |
| **STATUS** | One of the following status: active, blocked, out-of-sale |

## PRE\_ABT\_<tgt\_type>

General table for ML and VF ABTs preparation, it includes all quintuple PRODUCT|LOCATION|CUSTOMER|DISTR\_CHANNEL|date regarding the past and the forecast period. Description of this table is provided in the Pre ABT Definition Algorithm doc.

|  |  |
| --- | --- |
| **PRE\_ABT** | |
| Column Name | Description |
| **PRODUCT\_ID** | Product ID (the lowest level of the product hierarchy) |
| **LOCATION\_ID** | Location ID |
| **CUSTOMER\_ID** | Customer ID |
| **DISTR\_CHANNEL\_ID** | Distribution Channel ID |
| **PERIOD\_DT** | Date of sales (calendar day) |
| **TGT\_QTY\_R** | Restored demand |
| **TGT\_QTY** | Total sales in units per day (w/o returns) |
| **STOCK\_QTY** | Stock qty (BOP) |
| **DEFICIT\_FLG1** | 1|primary deficit occurred  0|no primary deficit |
| **DEFICIT\_FLG2** | 1|secondary deficit occurred  0|no secondary deficit |
| **PROMO\_FLG** | 1|promo event was active  0|no promo event |
| **NUM\_AUTORIZATION** | 1| if there is a demand info for product/location/day  0| otherwise |
| **PRICE\_PROMO** | Promo price for promo periods and regular price for non-promo periods |
| **PROMO\_TYPE1\_FLG** | 1|promo event of type 1 was active  0|no promo event of type 1 |
| **…** |  |
| **PROMO\_TYPEn\_FLG** | 1|promo event of type n was active  0|no promo event of type n |
| **PRICE\_ACT** | actual price |
| **PRICE\_REG** | regular price |

## CONFIG\_PARAMETERS

The following config parameters are used within the step.

|  |  |
| --- | --- |
| **CONFIG.CONFIG\_PARAMETERS** | |
| Column Name | Description |
| **IB\_ADJ\_FORECAST\_LIST** | Which output forecast tables must be used for alert calculation, default value = ‘ACC\_AGG\_HYBRID\_FORECAST’ |
|  |  |
|  |  |
| **IB\_NPF\_MAX\_HIST\_DEPTH** | Maximal number of observation if the past for new assortment, default value = 28 |
| **IB\_ADJ2\_MIN\_OBSERV\_NUM** | Minimal number of observations in the past to provide autocorrection, default value = 7 |
| **IB\_ADJ2\_BASE\_PAST\_PERIOD** | Shows numbers of days in the past that are used to calculate default forecast value, default value = 56 |
| **IB\_ADJ3\_BASE\_PAST\_PERIOD** | Shows numbers of days in the past that are used to calculate forecast confidence interval boundaries, default value = 56 |
| **IB\_ADJ3\_SEASONL\_CALC\_LVL** | Hierarchy level at which seasonality coefficient is calculated, default=  (PRODUCT\_LVL\_ID<m> (m=**VF\_PRODUCT\_LVL<=OUT\_PRODUCT\_LVL**)  LOCATION\_LVL\_ID<n> ( n=**VF\_LOCATION\_LVL<=OUT\_LOCATION\_LVL**) CUSTOMER\_LVL\_ID<k>  (k = **VF\_CUSTOMER\_LVL<=OUT\_CUSTOMER\_LVL**) DISTR\_CHANNEL\_LVL\_ID<l> (l = **VF\_DISTR\_CHANNEL\_LVL<=OUT\_CHANNEL\_LVL**) week) |
| **IB\_ADJ3\_USE\_SEAS\_COEF\_FLG** | Flag that indicates whether seasonal adjustment of boundaries is needed, default = 1 |
| **IB\_ADJ3\_MIN\_OBSERV\_NUM** | Minimal number of observations in the past to provide autocorrection, default value = 28,  This value is always >=5 |
| **IB\_ADJ3\_CORRECTION\_METHOD** | Approach how to replace outlier forecast value 2 values possible=’mean’ or ‘bound’, default = ‘mean’, |
| **IBN\_FF\_ACTIVE\_STATUS\_LIST** | List of statuses in FORECAST\_FLAG table that relates to active life period, default = (‘active’) |

### CONFIG-FILE TARGET VARIABLES

Target variables config file:

|  |  |  |
| --- | --- | --- |
| **CONFIG.TGT\_VAR\_CONFIG** | | |
| **Column Name** | **Description** | **Example** |
| **tgt\_type** | One of 3 types of the target variable:   * SELLIN – means CPG sales to its customer, * SELLOUT – means CPG’s customer sales to their clients, * POS – means sales in the point of sales, can be relevant for both Retailer and CPG | POS |
| **tgt\_qty\_table** | Name of the table which contains quantity information, one of 3 variants is possibe: IN\_SELL\_IN, IN\_SELL\_OUT, IN\_SALES | IN\_SALES |
| **value\_src** | Name of the target variable from the source table. It should be quantity of sales.  Feasible values: INVOICE\_QTY, SALES\_QTY, SHIPMENT\_QTY, ORDER\_QTY. | SALES\_QTY |
| **act\_flag** | Activity flag, whether this target variable is needed to be forecasted. Feasible values: 0 or 1 | 1 |
| **dr\_scen** | Demand restoration scenario for target variable:   * Scenario 0 (parameter = 0). Demand restoration isn’t needed. * Scenario 1 (parameter = 1). Retail scenario to a greater extent. Demand restoration is performed based on stock data. Demand extending/prolongation for short seasonal products isn’t performed. * Scenario 2 (parameter = 2). CPG scenario to a greater extent. Only demand extending/prolongation for short seasonal products is performed. * Scenario 3 (parameter = 3). Scenario 1 and Scenario 2 are performed sequentially. | 3 |
| **link\_with\_stock** | Flag, whether this target variable is linked with provided stock data. Feasible values: 0 or 1 | 1 |
| **link\_with\_promo** | Flag, whether this target variable is linked with provided promo data. Feasible values: 0 or 1 | 0 |
| **link\_with\_price** | Flag, whether this target variable is linked with provided price data. Feasible values: 0 or 1 | 1 |
| **vf\_product\_lvl** | Aggregation level for ML ABT by product hierarchy, default value is 8, which means PRODUCT\_ID | 7 |
| **vf\_location\_lvl** | Aggregation level for ML ABT by product hierarchy, default value is 6, which means LOCATION\_ID | 1 |
| **vf\_customer\_lvl** | Aggregation level for ML ABT by product hierarchy, default value is 6, which means CUSTOMER\_ID | 5 |
| **vf\_distr\_channel\_lvl** | Aggregation level for ML ABT by product hierarchy, default value is 3, which means DISTR\_CHANNEL\_ID | 1 |
| **vf\_time\_lvl** | Accumulation level for ML ABT by time hierarchy, default value is WEEK.2, which means weeks began from Monday | MONTH |
| **ml\_product\_lvl** | Aggregation level for ML ABT by product hierarchy, default value is 8, which means PRODUCT\_ID | 7 |
| **ml\_location\_lvl** | Aggregation level for ML ABT by product hierarchy, default value is 6, which means LOCATION\_ID | 5 |
| **ml\_customer\_lvl** | Aggregation level for ML ABT by product hierarchy, default value is 6, which means CUSTOMER\_ID | 4 |
| **ml\_distr\_channel\_lvl** | Aggregation level for ML ABT by product hierarchy, default value is 3, which means DISTR\_CHANNEL\_ID | 1 |
| **ml\_time\_lvl** | Accumulation level for ML ABT by time hierarchy, default value is WEEK.2, which means weeks began from Monday | WEEK.2 |
| **out\_product\_lvl** | Aggregation level for ML ABT by product hierarchy, default value is 8, which means PRODUCT\_ID | 8 |
| **out\_location\_lvl** | Aggregation level for ML ABT by product hierarchy, default value is 6, which means LOCATION\_ID | 6 |
| **out\_customer\_lvl** | Aggregation level for OUT tables by product hierarchy, default value is 6, which means CUSTOMER\_ID | 6 |
| **out\_distr\_channel\_lvl** | Aggregation level for OUT tables by product hierarchy, default value is 2, which means DISTR\_CHANNEL\_ID | 2 |
| **out\_time\_lvl** | Accumulation level for OUT Tables by time hierarchy, default value is WEEK.2, which means weeks began from Monday | WEEK.2 |

### ARLEY CRITERION CRITICAL VALUES

Table of statistical values the Arley criterion coefficient, selected according to the sample size *n* and the significance level α (probability of error of the 1st kind, the probability of removing a given value *x* from the sample as extreme, when it is not actually extreme);

The values of the coefficients of the Arley criterion (*za*) depending on the number of observations and significance (α) is presented below.:

|  |  |  |  |
| --- | --- | --- | --- |
| **CNT\_OBSERVATIONS\_LBOUND** | **CNT\_OBSERVATIONS\_UBOUND** | **K\_ARLEY\_001** | **K\_ARLEY\_005** |
| 0 | 3 | 1.60 | 1.50 |
| 4 | 5 | 2.15 | 1.85 |
| 6 | 7 | 2.30 | 1.90 |
| 8 | 9 | 2.40 | 1.92 |
| 10 | 11 | 2.44 | 1.92 |
| 12 | 13 | 2.46 | 1.92 |
| 14 | 15 | 2.46 | 1.92 |
| 16 | 999999 | 2.46 | 1.92 |

## Autocorrection parameters

## Autocorrection parameters

Section is not used for the first release

These settings define whether a specific type of the adjustment should be applied, whether a seasonal coefficient for adjustment type # 3 should be applied, and the replacement algorithm (i.e. the average value or the value of the interval boundary).

These settings will be applied in the subsequent process to every case respectively.

|  |  |
| --- | --- |
| **Column Name** | **Description** |
| **PRODUCT\_ID** | Product ID (the lowest level of the product hierarchy) |
| **STORE\_LOCATION\_ID** | Store location ID |
| **TBD in next releases** |  |
|  |  |
|  |  |

## INITIAL\_GLOBAL parameters

All parameters are listed in initial\_global file.

|  |  |
| --- | --- |
| **INITIAL\_GLOBAL parameters init** | |
| Column Name | Description |
| **IB\_HIST\_END\_DT** | Last known date (i.e. sales and stock information is known) |
| **IB\_FCST\_HORIZON** | Forecast Horizon |

## Other Dependencies

Denote: **OUT\_LVLS** means (out\_product\_lvl, out\_location\_lvl, out\_customer\_lvl, out\_distr\_channel\_lvl)

**OUT\_DIM\_LVLS** = (product\_lvl\_id<out\_product\_lvl>, location\_lvl\_id<out\_location\_lvl>, customer\_lvl\_id<out\_customer\_lvl>, distr\_channel\_lvl\_id<out\_ distr\_channel\_lvl>)

# Algorithm Definition

## Type 1: delete forecast for inactive period and mark forecast with missing valuesfor in-active period of assortment lifecycle

A case when the time series that should not be used in forecasting process, has non-zero forecast for the forecast period. The forecast should be fully cleared from these values.

**Inputs:** I1 = DISACC\_DISAGG\_HYBRID\_FORECAST\_<tgt\_type>, I2 = FORECAST\_FLAG

**Transformation algorithm:**

1. For each PRODUCT\_ID|STORE\_LOCATION\_ID|CUSTOMER\_ID|DISTR\_CHANNEL\_ID only dates PERIOD\_DT that are related to FORECAST\_FLAG activity period must be present in output table, other dates must be filtered out
   1. Aggregate FORECAST\_FLAG

on out\_ \*\*product\_lvl-s/ out\_location\_lvl/ out\_customer\_lvl/ out\_distr\_channel\_lvl /level {out\_lvls} and and special out\_time\_lvl :

SELECT OUT\_LVLS,

PERIOD\_START\_DT,

PERIOD\_END\_DT,

STATUS

FROM FORECAST\_FLAG

LEFT JOIN PRODUCT on product\_id

LEFT JOIN LOCATION on location\_id

LEFT JOIN CUSTOMER on customer\_id

LEFT JOIN DISTR\_CHANNEL on distr\_channel\_id

WHERE STATUS in IB\_FF\_ACTIVE\_STATUS\_LIST

GROUP BY OUT\_LVLS, intnx(out\_time\_lvl, 2.PERIOD\_START\_DT, 0, ‘b’), intnx(out\_time\_lvl, 2.PERIOD\_END\_DT, 0, ‘b’)

* 1. I1 full join[[1]](#footnote-2) a.I2 a. onon PRODUCT\_ID and LOCATION\_ID and CUSTOMER and DISTR\_CHANNEL out lvls
  2. AND

/\*use only dates inside lifecycle period\*/

I1.PERIOD\_DT>= I2a.PERIOD\_START\_DT

AND

I1.PERIOD\_DT<= I2.PERIOD\_ENDSTART\_DT

AND

/\*use only lifecycles inside forecasting period \*/

I2.PERIOD\_END\_DT>

AND

I2.PERIOD\_START\_DT<=+

AND

/\*use only lifecycle periods with appropriate status\*/

UPPERCASE(I2.STATUS) in

* 1. Add a column FLG\_APPLY\_CORR1

* 1. Add a column FLG\_APPLY\_CORR2

**Output**: As a result of this step, a table of the following structure is constructed, T1

|  |  |
| --- | --- |
| Column Name | Description |
| **PRODUCT\_LVL\_ID<m>PRODUCT\_ID** | Level m =**OUT\_PRODUCT\_LVL**of product hierarchy Product ID (the lowest level of the product hierarchy) |
| **LOCATION\_LVL\_ID<n>STORE\_LOCATION\_ID** | Level n =**OUT \_LOCATION\_LVL** of location hierarchyLocation ID |
| **CUSOMER\_LVL\_ID<k>** | Level k =**OUT \_CUSTOMER\_LVL** of customer hierarchy |
| **DISTR\_CHANNEL\_LVL\_ID<l>** | Level l =**OUT \_DISTR\_LVL** of distribution channel hierarchy |
| **PERIOD\_DT** | PeriodDate of sales (calendar dayOUT\_TIME\_LVL) |
| **SEGMENT\_NAME** | Time Series VF Segment Name |
| **VF\_FORECAST\_VALUE** | VF forecast value |
| **DEMAND\_TYPE** | ‘promo’ or ‘regular’ – type ML model is used to forecast |
| **ASSORTMENT\_TYPE** | new, or old |
| **ML\_FORECAST\_VALUE** | ML forecasted value |
| **HYBRID\_FORECAST\_VALUE** | HYBRID forecast value |
| **ENSEMBLE\_FORECAST\_VALUE** | Ensemble Forecast Value |
| **FORECAST\_SOURCE** | Forecast Source |
| **FLG\_APPLY\_CORR1** | Flag “Apply correction type # 1” |
| **FLG\_APPLY\_CORR2** | Flag “Apply correction type # 2” |

## Type 2: replace missing forecast value for a regular assortment

A case when the time series that should be forecasted, have missing forecast values for the forecast period.

**Inputs:** T1, DEMAND\_RESTORED\_DEMAND\_<tgt\_type>, I3=PREP \_FOR\_ABT\_<tgt\_type>

**Transformation algorithm:**

1. Select only rows from T1 where **FLG\_APPLY\_CORR2 = 1**
2. Aggregate DEMAND\_RESTORED\_<tgt\_type> on **out lvls** and out\_time\_lvl and period\_dt

SELECT **OUT\_DIM\_LVLS,** intnx(out\_time\_lvl, 2.PERIOD\_DT, 0, ‘b’)

,case when sum(1-PROMO\_FLG)=0 then missing else sum(TGT\_QTY\_R\*(1-PROMO\_FLG))/sum(1-PROMO\_FLG)\*count(\*) end as **S\_NP\_TGT\_QTY\_R**

,case when sum(PROMO\_FLG)=0 then missing else sum(TGT\_QTY\_R\*(PROMO\_FLG))/sum(PROMO\_FLG)\*count(\*) end as **S\_P\_TGT\_QTY\_R**

FROM DEMAND\_RESTORED

**GROUP BY** intnx(out\_time\_lvl, 2.PERIOD\_DT, 0, ‘b’)**, OUT\_DIM\_LVLS**

1. Calculate simple forecast and statistics values based on historical values from 21.
   1. 11. left join 2 on lvls and for eachRESTOED\_DEMAND on PRODUCT\_ID, LOCATION\_ID, PERIOD\_DT

SELECT **OUT\_DIM\_LVLS,**

MAX(PERIOD\_DT)

FROM 2

/\*number of observations in the past, regular\*/ **For each quadruplepair PRODUCT \_LOCATION CUSTOMER DISTR\_CHANNEL in 2**

SELECT **OUT\_DIM\_LVLS,**

/\*number of observations in the past\*/

=

COUNT(2.**PERIOD\_DT) as OBS\_NUMBER**

CASE WHEN COUNT(2.PERIOD\_TD)> **IB\_ADJ2\_MIN\_OBSERV\_NUM**

THEN AVERAGE(2.S\_NP\_TGT\_QTY\_R)

ELSE missing

END as **,**

/\*promo demand forecast is an average of promo demand in the past\*/

COALESCE(AVERAGE(S\_P\_TGT\_QTY\_R),

/\*workaround for the case when there is no promo periods in the past\*/

) as

= MAX(**PERIOD\_DT**)

FROM 2 1.

LEFT JOIN 3.a on **OUT\_DIM\_LVLS**

WHERE **PERIOD\_DT**>= MAX(2.**PERIOD\_DT**)- **IB\_ADJ2\_BASE\_PAST\_PERIOD**

LEFT JOIN 2 on **OUT\_DIM\_LVLS**

FROM 2DEMAND\_RESTOREDGROUP BY **OUT\_DIM\_LVLS**

HAVING 11.**PERIOD\_DT**>= MAX(2.**PERIOD\_DT**)- **IB\_ADJ2\_BASE\_PAST\_PERIOD**

/\*regular demand forecast is an average of regular demand in the past\*/

/\*regular demand forecast is an average of regular demand in the past\*/ =

CASE WHEN COUNT(**PERIOD\_DT)**> **IB\_ADJ2\_MIN\_OBSERV\_NUM**

THEN AVERAGE(S\_NP\_TGT\_QTY\_R) AVERAGE(**SALES\_QTY\_R**)

ELSE missing

END/\*promo demand forecast is an average of promo demand in the past\*/

FROM DEMAND\_RESTORED2

WHERE **PROMO\_FLG** = 0

* 1. ANDWHERE **PERIOD\_DT**>=

**MAX**(DEMAND\_RESTORED**.PERIOD\_DT**)- **IB\_ALERT2ADJ2\_BASE\_PAST\_PERIOD\**

/\*promo demand forecast is an average of promo demand in the past\*/

=

COALESCE(AVERAGE(S\_P\_TGT\_QTY\_R**SALES\_QTY\_R**),

/\*workaround for the case when there is no promo periods in the past\*/

)

FROM DEMAND\_RESTORED2

WHERE **PROMO\_FLG** = 1

AND **PERIOD\_DT**>=

MAX(DEMAND\_RESTORED**.PERIOD\_DT**)-**IB\_AALERT2DJ2\_BASE\_PAST\_PERIOD**

1. I3 PRE\_ABT\_<tgt\_type> aggregate on **PRODUCT and LOCATION and CUSTOMER and DISTR\_CHANNEL out lvlsOUT\_DIM\_LVLS** and intnx(out\_time\_lvl, 2.PERIOD\_START\_DT, 0, ‘b’) where PERIOD\_DT>IB\_HIST\_END\_DT:
2. Provide default value for autocorrected forecast

T1 left join 3. oOn **OUT\_DIM\_LVLS**

PRODUCT\_ID and LOCATION\_ID and CUSTOMER and DISTR\_CHANNEL out lvls left join I3=PREP\_FOR\_ABT4. on **OUT\_DIM\_LVLS**PRODUCT\_ID and LOCATION\_ID and CUSTOMER and DISTR\_CHANNEL out lvls and PERIOD\_DT

and provide new valuesvalue for those rows in T1 where

* 3. **BF\_AUTOCOR\_FORECAST\_VALUE = HYBRID\_FORECAST\_VALUE**
  4. **Calculate**

**Output**: As a result of this step, a table of the following structure is constructed, T2

|  |  |
| --- | --- |
| Column Name | Description |
| <The same structure as T1> |  |
| **ENSEMBLE\_FORECAST\_VALUE** | Ensemble Forecast Value |
| **FORECAST\_SOURCE** | Forecast Source |
| **BF\_AUTOCOR\_FORECAST\_VALUE** | Hybrid forecast value before autocorrections |

## Type 3: outlier detection and correction

A case when the time series that should have been forecasted, has the forecast with abnormally low or high values (i.e. below or above the acceptable borders).

**Conditions:**

1. The PRODUCT|STORE\_LOCATION|CUSTOMER|DISTR\_CHANNEL combination is active according to forecast flag AND
2. The date belongs to the forecast period from **IB\_HIST\_END\_DT**+1 till **IB\_HIST\_END\_DT** + **IB\_HIST\_END\_DT** AND
3. Demand forecast is an outlier i.e. falls into the 1 and 3 intervals according to the Arley criterion (see the description below), AND
4. There is enough demand history for Arley criterion boundaries calculation AND
5. No promo of any type is planned for the demand correction day
6. **IB\_ADJ3\_SEASONL\_CALC\_LVL** is always lower (more wide) than out lvls.

**Inputs:** T2, 4.2.2, DEMAND\_RESTORED\_DEMAND\_<tgt\_type>, I3=PREP\_FOR\_ABT

**Transformation algorithm:**

**Correction:**

1. T2 left join RESTOED\_DEMAND on PRODUCT\_ID, LOCATION\_ID, PERIOD\_DT
2. For each PRODUCT|LOCATION|CUSTOMER|DISTR\_CHANNEL|PERIOD\_DT from T2in 2 calculate 1) number of observation **OBS\_NUM2OBS\_NUM**, 2) average value **Mean** and 3) **STD** value based on S\_NP\_TGT\_QTY\_R SALES\_QTY\_R column based on DEMAND\_RESTORED4.2.2 value
   1. **OBS\_NUM2** = COUNT(**PERIOD\_DT)**,

**Mean**=CASE WHEN **OBS\_NUM** > **IB\_NPF\_MAX\_HIST\_DEPTH** /\*condition b\*/

THEN AVERAGE(S\_NP\_TGT\_QTY\_R **SALES\_QTY\_R**)

ELSE missing

END,

**STD =** STANDARD(S\_NP\_TGT\_QTY\_R **SALES\_QTY\_R**)

FROM DEMAND\_RESTORED4.2.2

WHERE **PROMO\_FLG** = 0 /\*condition e\*/

/\*condition c\*/

AND **PERIOD\_DT**>=

**MAX(DEMAND\_RESTORED.PERIOD\_DT)IB\_HIST\_END**-**IB\_ALERT3ADJ32\_BASE\_PAST\_PERIOD**

Where

* 1. Calculate Arley Criteria boundaries according to the formula below:

Note: if Mean is missing then UPPER and LOWER are also missingsBy default (if otherwise is not set in the parameterization table) **IB\_ADJ3\_SIGN\_LVL**

* 1. T2 left join 2 on PRODUCT\_ID, LOCATION\_ID, CUSTOMER and DISTR\_CHANNEL out lvls

1. Calculate seasonality correction coefficient on **IB\_ADJ3\_SEASONL\_CALC\_LVL**
   1. /\*Mean forecast value for each PRODUCT|LOCATION| CUSTOMER|DISTR\_CHANNEL|Date from the IB\_ADJ3\_SEASONL\_CALC\_LVL level **and keep the original level of granularity PRODUCT\_ID, LOCATION\_ID,** **CUSTOMER\_ID, DISTR\_CHANNEL\_ID[[2]](#footnote-3)**\*/

**Mean\_Forecast\_Value\_Agg** = AVG(**HYBRID\_FORECAST\_VALUE**)

FROM T2. Left join PRODUCT and LOCATION and CUSTOMER and DISTR\_CHANNEL hierarchies on {PRODUCT\_ID and LOCATION\_IDout lvls} (distinct rows)

WHERE **DEMAND\_TYPE** = ‘regular’

GROUP BY **IB\_ADJ3\_SEASONL\_CALC\_LVL**, T2.**PERIOD\_DT**

automatically remerge

/\*Mean historical demand value for each PRODUCT|LOCATION|CUSTOMER|DISTR\_CHANNEL from the IB\_ADJ3\_SEASONL\_CALC\_LVL level **and keep the original level of granularity PRODUCT\_ID, LOCATION\_ID, CUSTOMER\_ID, DISTR\_CHANNEL\_ID** \*/

* 1. **Mean\_Demand\_Value\_Agg =** AVG(S\_NP\_TGT\_QTY\_RSALES\_QTY\_R)

FROM 4.2.2DEMAND\_RESTORED. Left join PRODUCT and LOCATION and CUSTOMER and DISTR\_CHANNEL hierarchies on {out lvls} (distinct rows)**PRODUCT\_ID** and **LOCATION\_ID**

WHERE **PROMO\_FLG**= 0

GROUP BY **IB\_ADJ3\_SEASONL\_CALC\_LVL**

* 1. c. LEFT JOIN ON **PRODUCT\_ID, LOCATION\_ID, out\_lvls and PERIOD\_D****T**

LEFT JOIN b. on **PRODUCT\_ID, LOCATION\_IDout\_lvls,**

and calculate adjustment coefficient

– standard deviation of the time series x around mean, defined according to the formulas below:

The values of the lower and upper bounds are adjusted for seasonality of the PRODUCT\_ID – STORE\_LOCATION\_ID:

Note: if UPPER(LOWER) is missing then UPPER and LOWER are also missing

Where:

X – primary coefficient;

FCt – average forecast on day *t* for SKU among all stores and days with the promo flag = “0”;

– average sales (average of time series x).

Thus, for every PRODUCT\_ID and STORE\_LOCATION\_ID combination, three intervals are defined:

1. Calculate adjusted forecast values for each PRODUCT|LOCATION|CUSTOMER|DISTR\_CHANNEL|DT in c. based on boundaries:

**Output**: As a result of this step, a table ADJ\_DISACC\_DISAGG\_HYBRID\_FORECAST\_<tgt\_type> is created (details see 22.1).

1. Depending on the interval the forecast belongs to, the following actions are performed:

|  |  |
| --- | --- |
| 1. **Interval** | 1. **Apply demand correction** |
| 1. 1 | 1. Yes |
| 1. 2 | 1. No |
| 1. 3 | 1. Yes |

1. The forecast is adjusted according to the described Type # 2 algorithm for checking and correction, depending on the parameter value in the setup table, the forecast is corrected to:
2. The corresponding interval boundary (upper or lower), OR
3. The average.
4. By default, the correction to the average is applied.

## Output from the Algorithm

Final output table represents all the corrected forecasts merged together and it should look like as follows :

|  |  |
| --- | --- |
| **DM.FCSTADJ\_DISACC\_DISAGG\_HYBRID\_FORECAST \_FULL\_CONSOLID\_C** | |
| Column Name | Description |
| **PRODUCT\_LVL\_ID<m>PRODUCT\_ID** | Level m =**OUT\_PRODUCT\_LVL**of product hierarchy Product ID (the lowest level of the product hierarchy) |
| **LOCATION\_LVL\_ID<n>STORE\_LOCATION\_ID** | Level n =**OUT \_LOCATION\_LVL** of location hierarchyLocation ID |
| **CUSOMER\_LVL\_ID<k>** | Level k =**OUT \_CUSTOMER\_LVL** of customer hierarchy |
| **DISTR\_CHANNEL\_LVL\_ID<l>** | Level l =**OUT \_DISTR\_LVL** of distribution channel hierarchy |
| **PERIOD\_DT** | Period start date |
| **SEGMENT\_NAME** | Name of TS segment from VF |
| **VF\_FORECAST\_VALUE** | VF forecast value |
| **DEMAND\_TYPE** | ‘promo’ or ‘regular’ – type ML model is used to forecast |
| **ASSORTMENT\_TYPE** | new, or old |
| **ML\_FORECAST\_VALUE** | ML forecasted value |
| **HYBRID\_FORECAST\_VALUE** | HYBRID forecast value |
| **FINAL\_FORECAST\_VALUE** | FINAL forecast value |
| **ENSEMBLE\_FORECAST\_VALUE** | Ensemble Forecast Value |
| **FORECAST\_SOURCE** | Forecast Source |
| **BF\_AUTOCOR\_FORECAST\_VALUE** | Hybrid forecast value before autocorrections |
| **FLG\_APPLY\_CORR1** | Flag “Apply correction type # 1” |
| **FLG\_APPLY\_CORR2** | Flag “Apply correction type # 2” |
| **FLG\_APPLY\_CORR3** | Flag “Apply correction type # 3” |

|  |  |
| --- | --- |
| **FLG\_USE\_SEAS\_COEFF** | **Flag “Apply seasonal coefficient”** |
| **FLG\_USE\_BORDER\_VAL** | Flag “Apply interval boundary for adjustment 3 as target value” |

## Output from the Algorithm

1. Table with all types of forecast

|  |  |
| --- | --- |
| **HYBRID\_FORECAST** | |
| Column Name | Description |
| 1. **PRODUCT\_ID** | 1. Product ID (the lowest level of the product hierarchy) |
| 1. **STORE\_LOCATION\_ID** | 1. Location ID |
| 1. **PERIOD\_DT** | 1. Date of sales (calendar day) |
| 1. **SEGMENT\_NAME** |  |
| 1. **VF\_FORECAST\_VALUE** | 1. VF forecast value |
| 1. **DEMAND\_TYPE** | 1. ‘promo’ or ‘regular’ – type ML model is used to forecast |
| 1. **ASSORTMENT\_TYPE** | 1. new, or old |
| 1. **ML\_FORECAST\_VALUE** | 1. ML forecasted value |
| 1. **HYBRID\_FORECAST\_VALUE** | 1. HYBRID forecast value |

1. Full join results will be needed for FLA\_APPLY CORR2 [↑](#footnote-ref-2)
2. It is assumed that the aggregated results are automatically remerged level back onto all of the detail rows. [↑](#footnote-ref-3)