**Alert 2:**

Run program every *1 minute* since data for each device is coming every one minute and calculate the current status of each device and update it in csv file every one minute.

**Data Cleaning:**

* **Consumption value as 0:** if consumption\_value is zero, then drop that row since consumption value can’t be zero and also my score increases from **0.90856** to **0.98898**. Hence dropping zero consumption value is good choice.
* **Consumption value as empty:** if consumption\_value is empty, then drop that row since consumption value can’t be empty.
* **Consumption value is too large:**

Consumption\_value > mean + 3 \* standard\_deviation

In above case also drop that row since value is too large. It is outliers.

New Files created to refrainfrom **precomputation on the whole data set:**

**current\_alert\_2\_status.csv:** This file store the current/ live alert status of every device and is updated every 1 minute.

|  |  |  |
| --- | --- | --- |
| house\_id | household\_id | alert\_status |
| 0 | 0 | 0 |
| 2 | 5 | 0 |
| 32 | 6 | 0 |
| 32 | 5 | 0 |
| 32 | 4 | 0 |
| 3 | 0 | 0 |
| 32 | 3 | 0 |
| 32 | 7 | 0 |
| 32 | 2 | 0 |
| 3 | 1 | 0 |
| 31 | 0 | 0 |
| 30 | 0 | 0 |

**alert\_2\_last\_hour\_record.csv:**

Stores consumption\_values from all devices in last one hour.

|  |  |
| --- | --- |
| timestamp | consumption\_value |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

Variable to store mean of all the values as ***‘Mean’***

Variable to store standard deviation of the values as ***‘SD’***

If new consumption\_values comes at timestamp *1377987540* then add them and will remove all values that entered before *1377987540-60\*60=1377983940(one hour)* in past and calculate new mean and standard deviation.

**Imputation Method Used:**

If consumption\_value > mean + standard\_deviation then status of device is 1 otherwise 0.

Now, we have to *update mean and standard deviation:*

**Mold**: mean up to last record

**Sold:** standard deviation up to last record

**Cold:** count up to last record

Similarly, **Mnew, Snew, Cnew** are updated values after new consumption value **Xnew**.

Similiarly, we update *update mean and standard deviation* for values which outdated (entered before one hour) as left the**alert\_2\_last\_hour\_record.csv:**

And these values in alert\_1\_previous\_record.csv every *1 minute.*