**Alert Engine Design Document**

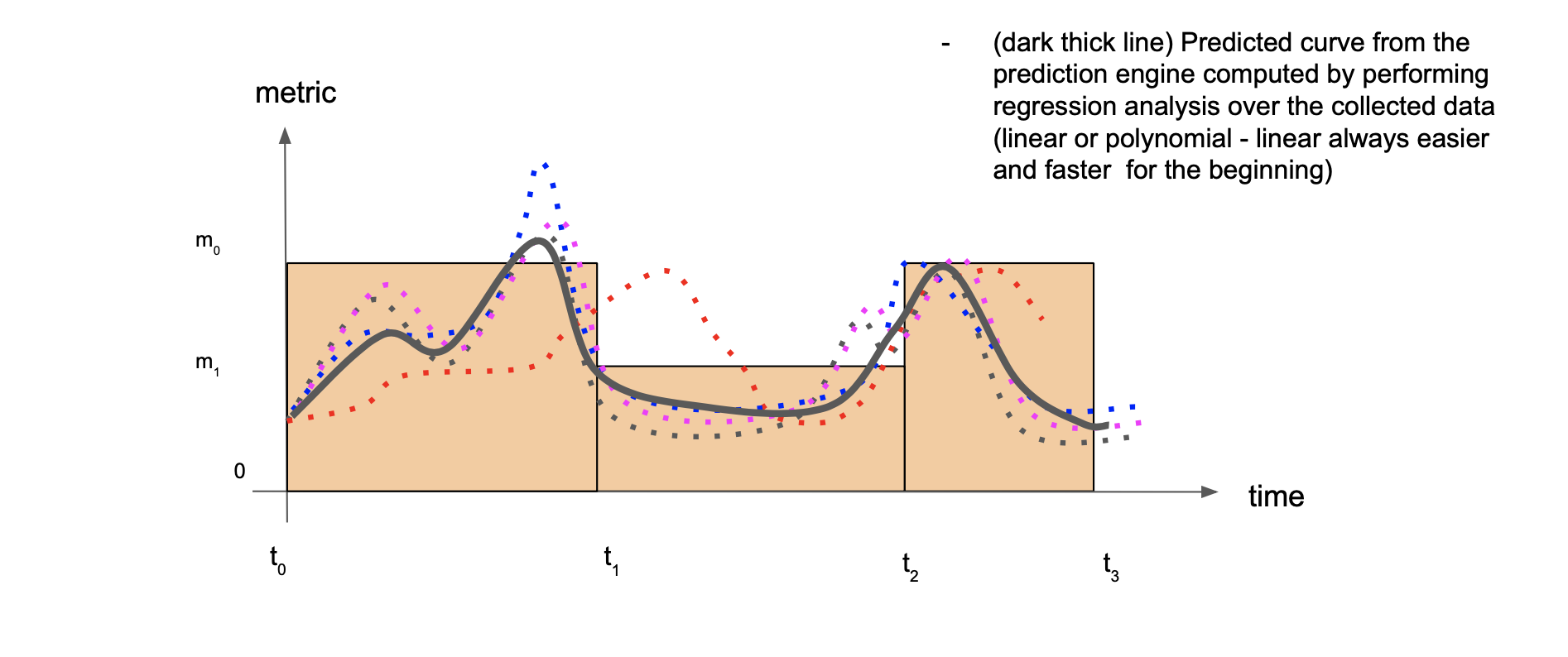
Created by: Mohit, Date: 15/11/2020

(Based on the conf call with JF and Sai.)

|  |  |  |
| --- | --- | --- |
| **Target Release** | - | **Background**  Suppose that the user does not want to be bothered with the monitoring of the changing ambience inside the office or kitchen or hotel, as she/he wants to focus on work. The aim of the alert engine is to handle the problem of indoor environment monitoring for the user efficiently by continuously monitoring the indoor air and thermal quality and sending alerts/warnings in case of the air quality and thermal indexes go below or above imputed or recommended thresholds. |
| Document Owner | Mohit |
| Engineer/Developer | Mohit |
| QA | Mohit, Jean-Francois, Sai |
|  |  |
|  |  |

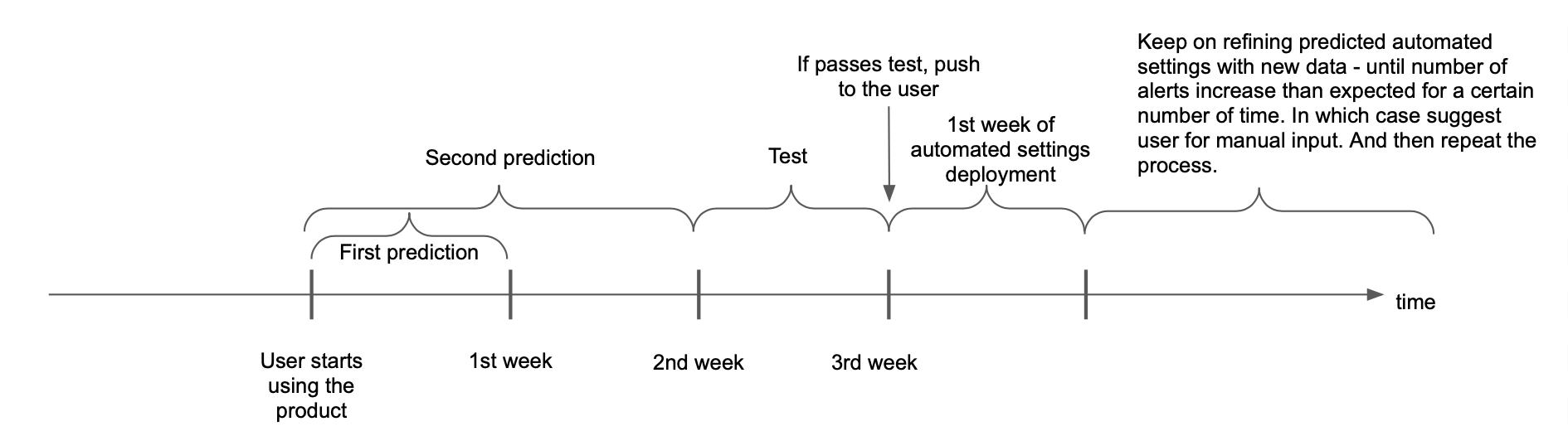
**Alert engine operational requirements:**

1. **Base Case - Manual Alerting System**
   1. In the beginning, the user inputs the 5 index operational ranges, i.e., the upper and lower cut-off values for each index desired by her/him. The 5 indices are: indoor air quality index, wellness index, health index, threat index and thermal comfort.
      1. Note: the 8 base metrics relating to the pollutants are measured by the SafeAir device and are **not** configured by the user. The default operating ranges of these metrics will be provided by the SafeAir researchers ( e.g.:, the default operational range of CO, CO2 etc.).
      2. The user will **not** specify the time for which the operational ranges are valid during a day.
   2. Offset times for the 5 indexes are provided by default.
      1. Offset time for an index is the duration (in minutes or hours) that represents the tolerable limit (in time) to generate an alert. If the readings continue to be out of operational range for longer than the offset duration the SafeAir device generates an alert.
      2. Edge Case: Accidental jumps of the metric readings to back in the range while otherwise it is out of range, and vice versa. (This should be handled by simply ignoring such accidental changes, but within a limit. Prediction version (next subsection) of this would inherently handle that.)
   3. If the metric values stay out of the operational range for the offset time, an alert is generated with the index values and the corresponding metric values that are causing the index to be out of range.
      1. This alert is then mapped to a contextual and potentially actionable message for the user.
   4. A threshold number of alerts over a threshold duration are used to detect if the user behaviour has changed. If the number of alerts being generated are beyond the threshold number of alerts for as long as the threshold duration, the app asks the user to update the index ranges manually - suggesting a pattern change in the user behaviour and to adapt to the new environment. This is the system’s decision making procedure to decide that calibration of the operational indices might be needed as the user behaviour has changed. (This is how the system detects the change in the environment and calls for user intervention for the new settings.)
      1. The threshold values for the number of alerts and the duration will be provided by default. These are different from offsets for an alert.
      2. The user could be prompted to infer new input values by suggestions about the out of range values of the indices. E.g.: ‘In the last days, we noticed that the thermal index has increased to 35 degree (average of out of range values) celsius from the previously set expected value of 28 degree celsius. Would you like to set it as the new normal?’
   5. The system consistently performs the steps ‘a’ to ‘d’ forever or until switched off.
      1. Note: the manual alert system is only concerned with the (user inputed) operational index ranges. It has no sense of pattern recognition, i.e., what is happening within those ranges.
2. **Pattern change prediction within metric readings using a predictive engine**
   1. Assumptions:
      1. Base case, i.e., manually configured alert engine is up and running. The user has provided operational index ranges for the indices. The alerts are generated as in the base case. (Note: Alert mechanism only involves checking whether the index value lies within the operational range or not and is independent of the prediction engine.)



(Figure: metric in the figure is any index. m0 and m1 are the index operational ranges manually input from the user. The four dotted lines are the SafeAir device readings over the 4 days and the thick red line is the current prediction (i.e., after 4 days of readings) by the predictive engine with the estimate that the next day’s readings will not be too far away from the predicted thick line.)

* 1. Background process: The predictive engine monitors the data of the past readings and generates the pattern for the indices for the whole 24 hour period. (See picture above for reference for a single index.) The prediction engine uses regression analysis to achieve the following:
     1. Use prediction to suggest if the operational index range set by the user is consistent with the user behaviour. E.g.:
        1. The user behaviour causes the device readings to be with too much within the manually set operational ranges of the index. The prediction engine will suggest a tighter operational range set to have close to reality alerting possible.
        2. The user behaviour causes a few alerts every day during specific time periods, i.e., the readings are out of the operational range. The prediction engine will suggest corrected operational range by studying such patterns for a while.
        3. If there is a sudden change in user behaviour causing a large number of alerts, i.e., more than the threshold value of number of alerts and for the threshold duration - (manual engine will prompt the user with the resetting of the operational ranges for the indices) - the prediction engine will then repeat step 1 and 2. Note: Once there is enough data over several weeks and months about the user behaviour then certain seasonal changes in the user behaviour (that cause a lot of alerts) could be detected. Naturally, this will need much longer study of the user behaviour.
     2. Note that the prediction engine is oblivious to the user and works in the background.
        1. The mechanism to decide if there is a need to update the operational index ranges by the user is still the same as for the base case, i.e., the deviation from the threshold number of alerts and for a threshold number of times - which signals a potential pattern change. Or if the user behaviour causes the device readings to be well within the operational ranges set by her/him.
     3. The difference to the base case (manual input) is that the prediction engine computes and suggests the new operational index ranges. This is not fully done in the manual case without the user’s input. The alerting process remains the same.
     4. Offset values
  2. The prediction engine continues to monitor the readings and make suggestions to the user to update the operational ranges whenever necessary to align the user behaviour (reality) with the alerting mechanism. The process repeats itself forever or until switched off.



* + 1. Note that whenever the prediction engine cannot help, the system by default falls back to the manual alerting mechanism i.e., the base case.

Data Needed.

1. **Advanced Recommendations - (To be decided in detail later with consultation from Sina, JF, Sai ...)**
   1. Example: detect the change of rate of indexes and generate recommendations about the future so that the user could be signaled to take appropriate actions.

**Feature requirements (to be updated after confirmation of the operational requirements)**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **#** | **Title** | **Feature Description** | **Priority** | **Notes** |
| 1 | Air quality Index Classifier | The air quality index (AQI) classifier monitors the basic metrics being updated from the SafeWir device and computes the  AQI index value.  It then classifies the value into the following categories:   1. Good 2. Moderate 3. Unhealthy 4. Hazardous   (This is applicable to index 1-5 below.) | Must Have | Resource: Sina’s Slides - <https://drive.google.com/file/d/1-XypO_Dp6N25YeitRtk9HTj0Qn17vaCp/view?usp=sharing>   * Need input from JF and Sina. * Need to communicate with backend engineers who will integrate it into the app |
| 2 | Thermal Comfort Classifier | The thermal comfort index (TCI) classifier monitors the basic metrics being updated from the SafeWir device and computes the  AQI index value. It then classifies the value into the following categories:   1. Cold and dry 2. Cold 3. Cold and humid 4. Humid 5. Comfort Zone 6. Hot and Dry 7. Hot 8. Hot and Humid 9. Dry | Must Have |
| 3 | Indoor Environment Classifier | Computes the indoor environment index and classifies into:   1. Good 2. Moderate 3. Unhealthy 4. Hazardous | Must Have |
| 4 | Wellbeing Index classifier | Computes the wellbeing index and classifies into:   1. Good 2. Moderate 3. Unhealthy 4. Hazardous | Must have |
| 5 | Health Index classifier | Computes the health index and classifies into:   1. Good 2. Moderate 3. Unhealthy 4. Hazardous | Must have |
| 6 | Alert-Message Mapping | Maps the alerts generated from the output of the classifiers to an to messages (containing relevant statistics) and any potential actionable steps. | Must have | * Need to discuss and design with Sina |