**Clean Dataset Documentation**

**Description:**

The original dataset contains a lot of noise due to the notifications feature as well as a lot of redundancy of data. The redundancy of the data, added to the fact that the data was sampled more frequently in some periods than in the others (use of the monster logger in some periods) cause the model to converge into the most sampled period.

To limit this problem, and to have a more compact and meaningful representation of the data, we decided to build a non-redundant version of the dataset. The idea is to do the following: First, we filter the noisy notifications from the data and keep only the new added notification in each record. Filtering means that we remove the useless notifications (non displayed notifications or duplicate notifications) with keeping the other features that were provided to us in the record (ex: information about place, battery,…). Second, we take each separately and represent its realization and it´s time only if its realization differs from the previous one. For example, if we have 10 samples successive at times t1,..,t10 indicating that user u is in place x1 and the 11th sample at time t11 indicating that u is in place x2, we represent only the fact that u was in x1 from t1 to t10 and in u2 from t11.

This is general idea, but when this representation is obvious for certain features as the time, it is less obvious for some others. For example, concerning the Bleutooth devices detected, what do we represent and when? When a new device is detected? do we represent all the devices again or only the new one?

In this document, we answer to those questions and consider feature by feature and explain the way how we decided to represent it.

An important thing to keep in mind is that we want to keep all the information in the new dataset, that is to say that we can reconstruct again the original dataset with all the attributes and values starting from the new one.

**General rules:**

Before discussing individual updates, we discuss some rules that apply to all the features.

* The data is in a json format and has the following structure:
  + The name of the feature as a first key. For example app\_launch, or location.
  + Then for each features, the keys are be the dates and times of start and end of the different realizations.
  + Finally for each date and time key, there is the corresponding realization (or event as value)
* All the times are represented into the local time zone of the user followed by the time in milliseconds to be able to differentiate two events that occurred in the same second.
  + The time date time key has the following format:

year-month-day hour:minute:second , time\_in\_millis -> year-month-day hour:minute:second , time\_in\_millis

Where the first entry corresponds to the first time when we recorded this entry and the second to the last time when we recorded it.

* In each realization, a feature event indicates what event was in the origin generating the record of this value. Event can take one of the following values:
  + Notification
  + Launcher\_on
  + Launcher\_off
  + Screen\_on
  + Screen\_off
  + Ducplicated\_notification
* As we said, we want to be able to reconstruct the original dataset from the clean one. For that reason, each realization contain the attribute seq that is a unique identifier for the record from where comes this realization. In this way we can reconstruct the whole record from the realizations. Note that as our realizations represent the concatenation of a lot of ones, we represent the different sequences numbers as an array.
* For the new realizations, we may want to know what attributes has changed since the last one. For that reason, all the features contain a new attribute called changed that contains the names of the attributes that has changed since the last realization.
* When a feature is missing in some record, it means that it is unknown. For that reason the end date of the current realization can be setted up (to the time of the previous realization) and the next time the feature is seen, a new realization starts.
* For the features that are represented as single arrays (not arrays), there is an attribute “changed” that contains the attributes that have changed since the last realization

**Feature update:**

Now it is time to discuss the individual features update

* **activityRecognitionResult:** this feature be divided in two features; one that contains the main attributes and the other that contains the extra attributes.
  + **activityRecognitionResult\_main:** is added as a new entry only when at least one the attributes change from the last record. For the onces that has not changed they will be missing in the changed feature. contains the following attributes
    - **activity**
    - **motion**
    - **posture**
  + **activityRecognitionResult\_other:** is added as a new entry only when at least one the attributes change from the last record. For the onces that has not changed they will be missing in the changed feature. contains the following attributes
    - **carrying**
    - **screen**
    - **usb**
* **androidActivityRecognitionResult:** this feature also be divided in two features; one that contains information about the main activity (the most probable one) and the other that contains information about the other activities (less probable ones)
  + **androidActivityRecognitionResult\_main:** is added as a new entry only when the attribute activity changes from the last record. Confidence attribute is an array that contains the successive confidences recorded of the current activity
    - **activity**
    - **confidence**
  + **androidActivityRecognitionResult\_other:** is added as a new entry only when at least one the attributes change from the last record. For the ones that has not changed they will be missing in the changed feature. contains the following attributes
    - **activities**
    - **confidences**
* **appLaunch:** an application launch is a punctual event and not an event that occurs during a certain interval of time. For that reason, the key only represent a date and not an interval. It be in the form : year-month-day hour:minute:second , time\_in\_millis.

Besides for that particular event, all the app launches are represented event if a user launches the same app two successive times. The attribute is:

* + - **app name**
* **battery:**  devided in two features
  + **battery\_main:** is added as a new entry only when at least one the attributes change from the last record. For the ones that has not changed they will be missing in the changed feature. contains the following attributes
    - **health**
    - **status**
    - **plugged**
    - **present**
  + **battery\_other:** is added as a new entry only when at least one the attributes change from the last record. For the ones that has not changed they contains as a value st\_value. contains the following attributes
    - **level**
    - **scale**
    - **technology**
    - **temperature**
    - **voltage**
* **bluetooth:** As the Bluetooth is represented as an array where each entry represents a detected device, a new realization is added if one of the attributes concerning one detected device change or if one device is added or one is removed.
* **event:** an event is a punctual event and not an event that occurs during a certain interval of time. For that reason, the key only represent a date and not an interval. It be in the form : year-month-day hour:minute:second , time\_in\_millis.

Moreover all the events be represented. All the attributes be represented.

* **notifications:** a notification is a punctual event and not an event that occurs during a certain interval of time. For that reason, the key only represent a date and not an interval. It be in the form : year-month-day hour:minute:second , time\_in\_millis.

Moreover all the notifications be represented (even if we have successively the same notification). All the attributes be represented.

* **headsetPlug:** It contains the following attributes. A change of any of them causes a new realization to be written:
  + **microphone**
  + **name**
  + **state**
* **launcherLayouts[]:** we decide to make the exception and to not represent this feature because it is not meaningful to us
* **location:** as long as the latitude and the longitude are different from zero, we consider this entry as valid. Moreover as long as the latitude and the longitude have the same values we consider that the user stays in the same place and represent those entries in the same realization (no matter how much is the accuracy). The accuracies are represented as an array where each accuracy represents the accuracy of one sampled location. The same for the other entries.
* **networkInfo:** Most of the arguments are represented and a change of one of them result in writing a new realization. The arguments that do not cause the creation of a new realization are represented as an array an are the following:
  + **detailState**
  + **extraInfo**
  + **subtype**
  + **subTypeName**
* **Predictors:**  also be ignored
* **sensor:** Represented as a punctual event. This means that each new value is written as a new realization.
* **settingInfo:** same as Bluetooth.
* **telephony:** As for location the attributes that count to write a new realization are the following
  + **cdmaCellLocBaseStationId**
  + **cdmaCellLocBaseStationLat**
  + **cdmaCellLocBaseStationLng**
  + **networkRoming**

For the others, they will be represented as an array

* **wifiAps:** as Bluetooth with considering only the following attributes
  + **bssId**
  + **ssid**
  + **capabilities**
  + **frequency**
  + **level**
* **wifiConnectedAp:** a realization be added if one of the following attributes change
  + **bssid**
  + **ssid**

For the others, the values are simply be represented as an array