Prepared by
Md Kamal Hossain

Google cloud architecture to operationalize ML mode

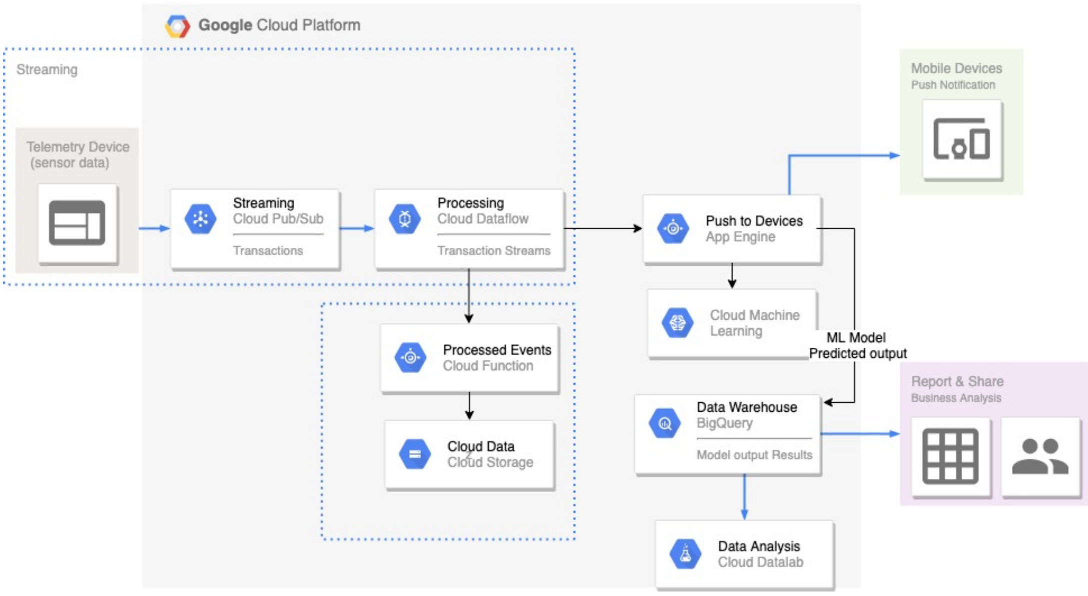


Figure: Cloud architecture for Predictive maintenance ML model

Reasons to consume cloud services as shown in the architecture diagram:

1. **Cloud Pub/Sub:** streaming telemetry data will be push to cloud Pub/Sub. Telemetry is producer here.
2. **Cloud Dataflow:** From Pub/Sub telemetry will be consume by Dataflow and processed/featured engineered the data to be used and served with ML model.
3. **Cloud Function & Cloud Storage:** The consumed telemetry events will be processed and then stored in cloud storage for post analysis and future use case development.
4. **App Engine:** The pre-processed data from cloud dataflow will be passed to app engine for the following actions –
   a. call trained ML model to predict the probability of device failure with new processed data from cloud dataflow
   b. if likelihood of a device failure is high, send a push notification.
   c. Store the model prediction output along with other data in BigQuery.
5. **BigQuery:** Data warehouse and visualization, automated report generation and post analysis.
6. **Cloud Machine Learning Engine:** The trained model is deployed in cloud ML engine to serve the prediction in a serverless and scale-able manner.
7. **Cloud Datalab:** The future training/retraining of ML model and the deployment of the model in cloud ML engine is done using cloud Datalab.

Reduce maintenance cost through predictive techniques – Assignment

Problem statement:

3D technologies, a ASEAN based company provides telemetry devices to its customer. These devices transmit daily aggregated telemetry attributes back to the company Customer Services department. To reduce cost, head of customer services want to do predictive maintenance or time-based preventive maintenance for cost savings.

To predict the devices failure, a machine learning (ML) based predictive model will be developed using the aggregated telemetry data. This ML model will predict the probability of telemetry device.

To build the ML model, I have aggregated the data to predict the probability of telemetry device failure within 07 days. The details about the data pre-processing and techniques are describe in Feature Engineering section.

Exploratory data analysis:

For the exploratory analysis, the following machine and tool has been used –
1. Google bigquery
2. Jupyter notebook (python 3.7)

Sample data:

	date	device	failure	attribute1	attribute2	attribute3	attribute4	attribute5	attribute6	attribute7	attribute8	attribute9
0	2015-01-01	S1F01085	0	215630672	56	0	52	6	407438	0	0	7
1	2015-01-01	S1F0166B	0	61370680	0	3	0	6	403174	0	0	0
2	2015-01-01	S1F01E6Y	0	173295968	0	0	0	12	237394	0	0	0
3	2015-01-01	S1F01JE0	0	79694024	0	0	0	6	410186	0	0	0
4	2015-01-01	S1F01R2B	0	135970480	0	0	0	15	313173	0	0	3

Figure 1: Sample data

```
Distinct value count:

date : 304
device : 1169
failure : 2
attribute1 : 123877
attribute2 : 558
attribute3 : 47
attribute4 : 115
attribute5 : 60
attribute6 : 44838
attribute7 : 28
attribute8 : 28
attribute9 : 65

failure record count
0    124388
1       106
```

The telemetry data are spread-out over 304 days for 1169 devices. Out of these 1169 devices, only 106 devices were failed during this period.

I want to observe the correlation among the recorded attributes. From the plot, I can say none of the attributes have strong correlation with label attributes which is “**failure**”.

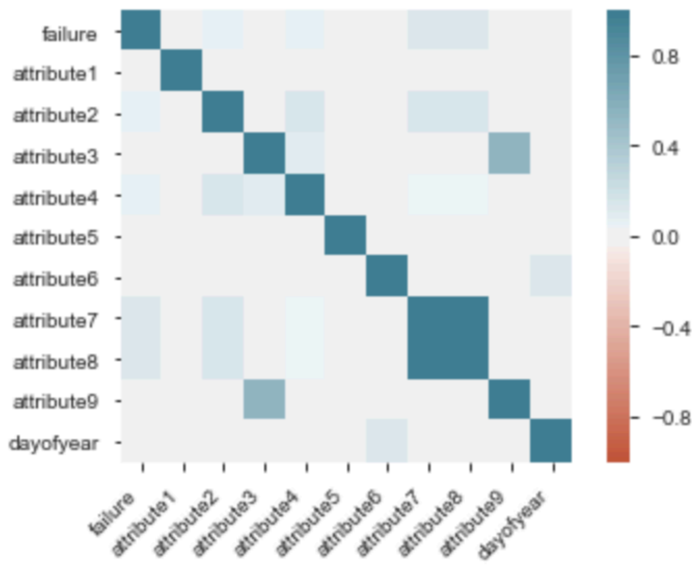


Figure 2: Correlation among attributes and label variables

Some of the interesting facts about the dataset:

- each device failed max 01 times.
- minimum records for a device is 5.
- maximum records for a device is 299.
- attribute1, attribute2 and attribute 6 seems alarm code to me (basically, it was guess). Hence, I have consider those as categorical features.
- Rest of the attributes are considered as numerical features.
- attribute2, attribute 4, attribute 7 and attribute 8 has some correlation with failure.
- there are not much difference one or two days before the device failure as observed.

	date	device	failure	attribute1	attribute2	attribute3	attribute4	attribute5	attribute6	attribute7	attribute8	attribute9	pd_timestamp
11443	2015-01-13	S1F023H2	0	177478384	0	0	1	19	507971	16	16	3	2015-01-13
12198	2015-01-14	S1F023H2	0	203744280	0	0	1	19	509250	16	16	3	2015-01-14
12914	2015-01-15	S1F023H2	0	222474632	0	0	1	19	510519	16	16	3	2015-01-15
13629	2015-01-16	S1F023H2	0	243825496	0	0	1	19	511783	16	16	3	2015-01-16
14344	2015-01-17	S1F023H2	0	20761856	0	0	1	19	513110	16	16	3	2015-01-17
15059	2015-01-18	S1F023H2	0	41291000	0	0	1	19	513722	16	16	3	2015-01-18
15773	2015-01-19	S1F023H2	1	64499464	0	0	1	19	514661	16	16	3	2015-01-19

Figure 3: Screenshot of a failed device data for last 07 days

Feature Engineering:

The data was aggregated to capture the behaviour and pattern of the failed devices in last 07 days.

- The failed devices data was available until those are failed. Hence, the last record of failed device is when the failure happened.
- The devices those did not fail and those fail, have some difference in reading for last 07 days records. There is a pattern as shown in screenshot.

	date	device	failure	attribute1	attribute2	attribute3	attribute4	attribute5	attribute6	attribute7	attribute8	attribute9
70518	2015-04-14	Z1F148T1	0	141054368	0	0	0	8	198466	96	96	0
70922	2015-04-15	Z1F148T1	0	152826840	0	0	0	8	198466	96	96	0
71326	2015-04-16	Z1F148T1	0	171971496	0	0	0	8	198466	96	96	0
71689	2015-04-17	Z1F148T1	0	183377408	0	0	0	8	198466	96	96	0
72049	2015-04-18	Z1F148T1	0	201716856	0	0	9	8	198466	80	80	0
72409	2015-04-19	Z1F148T1	0	40510240	0	0	9	8	198466	80	80	0
72769	2015-04-20	Z1F148T1	1	134062824	0	0	9	8	198466	80	80	0

Figure 4: Last 07 days record of a failed device

	date	device	failure	attribute1	attribute2	attribute3	attribute4	attribute5	attribute6	attribute7	attribute8	attribute9
32151	2015-02-11	S1F01E6Y	0	57781112	0	0	0	12	259450	0	0	0
32863	2015-02-12	S1F01E6Y	0	182622184	0	0	0	12	259460	0	0	0
33575	2015-02-13	S1F01E6Y	0	79408632	0	0	0	12	259463	0	0	0
34287	2015-02-14	S1F01E6Y	0	83214688	0	0	0	12	259468	0	0	0
34999	2015-02-15	S1F01E6Y	0	227370128	0	0	0	12	259479	0	0	0
35710	2015-02-16	S1F01E6Y	0	182876688	0	0	0	12	259486	0	0	0
36421	2015-02-17	S1F01E6Y	0	147350000	0	0	0	12	259491	0	0	0

Figure 5: Last 07 days record of a good device

- Before a device goes off, there is a spike in some of the attributes value, which is a good indicator towards a device failure as shown in screenshot.

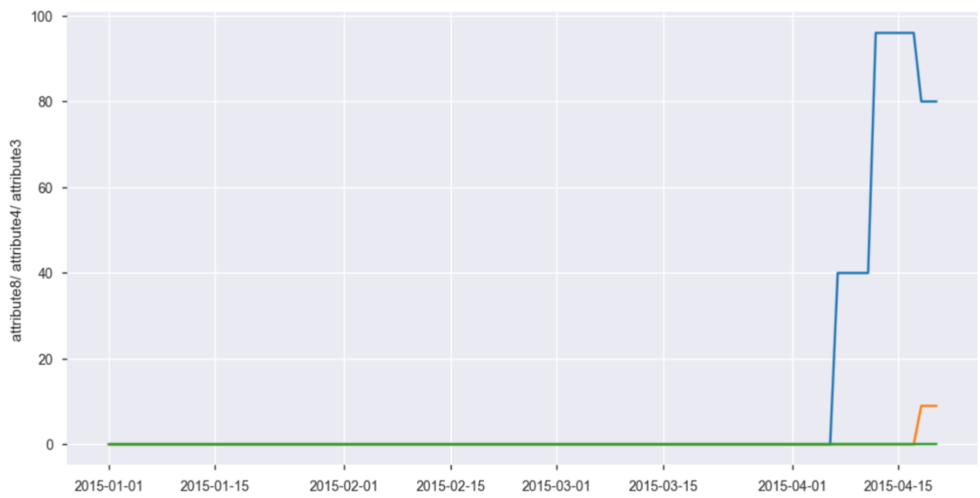
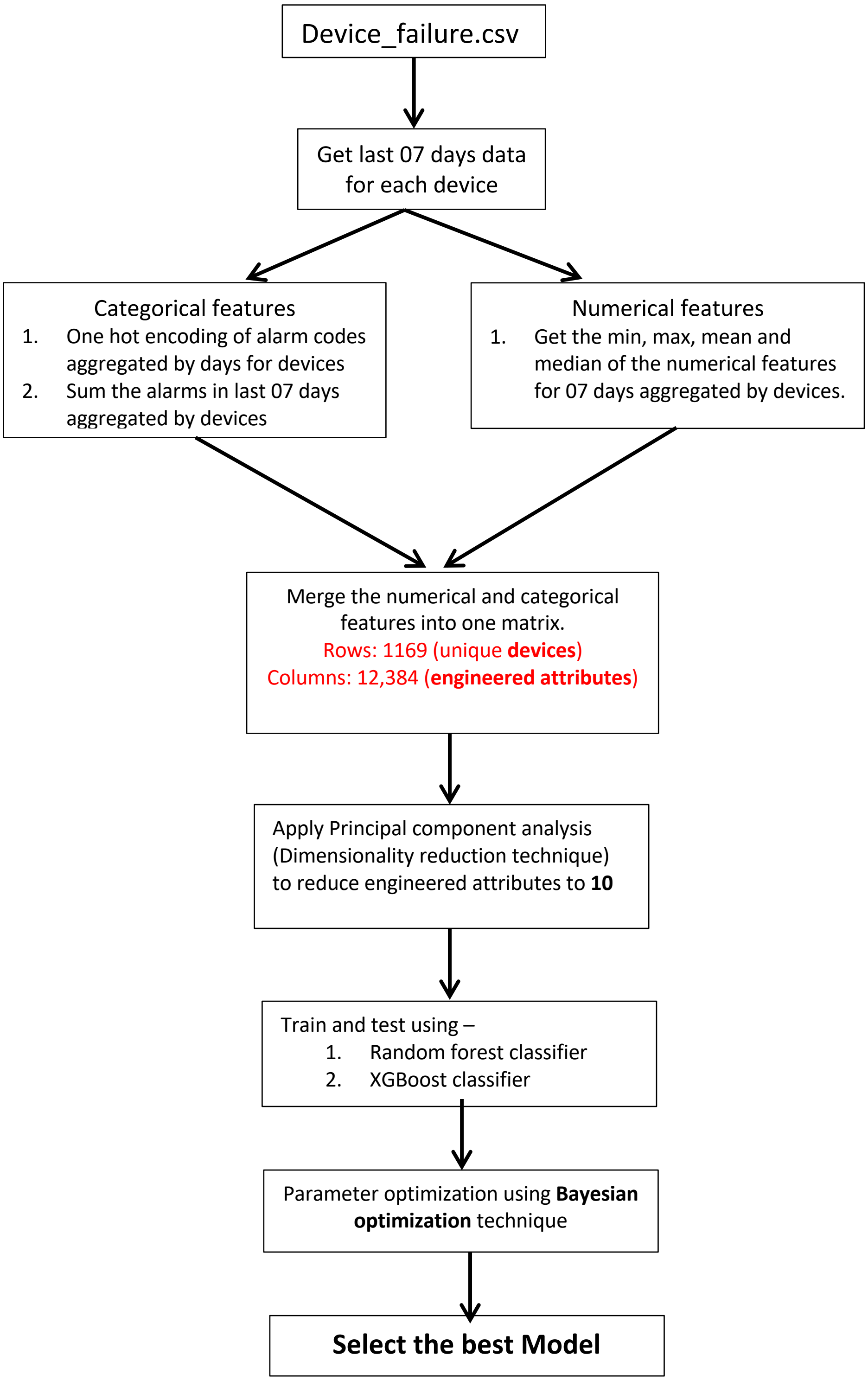


Figure 6: Changes of attributes value before its failure

- Finally, there are not much distinct difference in attributes value for last couple of days. Hence, the predict ML model was developed to predict whether the device will fail in 07 days or not.

- End to end Data pre-processing pipeline:



Result and Analysis:

The objective of this predictive ML model is the reduce the maintenance cost by increasing the frequency of scheduled check if necessary. Hence, in our case overall accuracy is important but a balanced ratio of precision and recall is more important then only high accuracy. In this case,

$$\text{Recall} = \frac{\text{true positive}}{\text{true positive} + \text{false negative}}$$

$$= \frac{\text{correctly predicted device failure}}{\text{correctly predicted device failure} + \text{wrongly predicted device will not fail}}$$

$$\text{Precision} = \frac{\text{true positive}}{\text{true positive} + \text{false positive}}$$

$$= \frac{\text{correctly predicted device failure}}{\text{correctly predicted device failure} + \text{wrongly predicted device will fail}}$$

Hence, in this case if the recall or precision is low, technician will end up to visit the site wrongly or device will fail without knowing its upfront which will eventually cost more. However, in cases where we want to find an optimal blend of precision and recall; we can combine the two metrics using what is called the F1 score.
The F1 score is the harmonic mean of precision and recall taking both metrics into account in the following equation:

$$F1\ score = 2 * \frac{precision * recall}{precision + recall}$$

Hence, during our model building process, we want to get optimal value pf precision and recall and maximize F1 score to select best model.

```
   1. classifier:  RandomForestClassifier
* dataset is unbalanced, hence class height of {1:10,0:1} has been used


Accuracy: 0.8974358974358975
F1 score: 0.35714285714285715
Recall: 0.3448275862068966
Precision: 0.37037037037037035

 classification report:
               precision    recall  f1-score   support

           0       0.94      0.95      0.94       322
           1       0.37      0.34      0.36        29

    accuracy                           0.90       351
   macro avg       0.66      0.65      0.65       351
weighted avg       0.89      0.90      0.90       351


 confussion matrix:
 [[305  17]
 [ 19  10]]
```

2. classifier: XGBClassifier

```
Accuracy: 0.9401709401709402
F1 score: 0.631578947368421
Recall: 0.6
Precision: 0.6666666666666666

 classification report:
               precision    recall  f1-score   support

           0       0.96      0.97      0.97       321
           1       0.67      0.60      0.63        30

    accuracy                           0.94       351
   macro avg       0.81      0.79      0.80       351
weighted avg       0.94      0.94      0.94       351


 confusion matrix:
 [[312   9]
 [ 12  18]]
```

XGboost classifier shows better performance compare to random forest classifier to predict whether a device will fail within 07 days or not. We have tried Bayesian Optimization technique to find tune Hyper parameters. Based on our initial investigation, default parameters doing better. Hence, the final ML model is XGBoost classifier.

Conclusion:

Total number of records are only 1169 and dataset was unbalanced. Random forest and xgboost are very good in classification. Hence, these two algorithms are used to the predictive model. Moreover, it was featured engineered/ fine tuned to obtained a balance precision and recall, and optimal F1 score. The XGboost ML model has achieve 94% accuracy and F1 score of 0.58 which is considered a good model.

The model now can predict whether a device will fail or not within 07 days approximately with 94% confidence. Hence, this predictive model will definitely help to optimize maintenance schedule and reduce the cost by proving predictive maintenance.