In-house ML Model for Inverter's 'Trip' Prediction & Forecasting for GMR solar plant

Ishan Jain

March 4, 2021

Table of Content

Section 1 - Fault detection and classification

Problem Statement

to classify the alarm_raised into Trip/Fault and No-trip/No-fault using binary classification.

Assumptions

1. Trips in each inverter are independent of the other inverters.
2. Trips are identified and defined in the models with the following condition: data['alarm_class'] = np.where(((data['Time'] >= "07:00") & (data['Time'] <= "17:00") & (data['active_power']=0) & (data['irradiance']>100) & (data['alarm_name'].notnull())) , 'yes', 'no')
3. All the alarms raised which are not considered into 'Trips' are considered as 'no-Trip' in binary classification.

Methodology

Following models are attempted to classify and detect the faults in the historic data from Nov 2020 to Jan 2021:

#	**Models**	**Performance**
1	Logistics Regression	● Overall Accuracy : 88.75 % ● Sensitivity/Recall : 50.2 % ● Specificity : 95.67 % ● Precision : 67.5 %
2	Random Forest Classifier	● Overall Accuracy : 98.83 % ● Sensitivity/Recall : 97.75 % ● Specificity : 99.02 % ● Precision : 94.7 %
3	XGBoost	● Overall Accuracy : 93.53 % ● Sensitivity/Recall : 66.74 % ● Specificity : 98.34 % ● Precision : 87.8 %
4	AdaBoost	● Overall Accuracy : 92.36 % ● Sensitivity/Recall : 63.81 % ● Specificity : 97.48 % ● Precision : 81.95 %
5	LightGBM	● Overall Accuracy : 94.19 % ● Sensitivity/Recall : 69.97 % ● Specificity : 98.53 % ● Precision : 89.5 %

6	Gradient Boosting Classifier	● Overall Accuracy : 93.68 % ● Sensitivity/Recall : 67.36 % ● Specificity : 98.4 % ● Precision : 88.27 %
7	LSTM	● Overall Accuracy : 88 % ● Sensitivity/Recall : 90 % ● Precision : 88 % ● F1 score : 89 %

Results

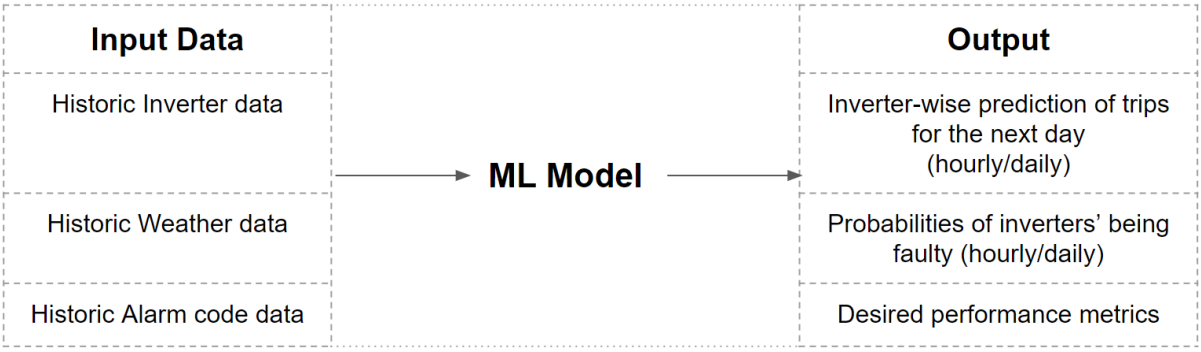
Following five models are excellent in detecting and classifying trips for the given data with a good performance matrix as described above.

1. XGBoost
2. Random Forest
3. Light GBM
4. AdaBoost Classifier
5. Bagging Classifier

Section 2 - Fault forecasting

Problem Statement

To develop an ML model for predicting trips (alarms where inverters are getting shut-down, i.e. active power = 0) in each inverter of the plants for the next day.



Assumptions

1. Trips in each inverter are independent of the other inverters.
2. Trips are identified and defined in the models with the following condition:
   data['alarm_class'] = np.where(((data['Time'] >= "07:00") & (data['Time'] <= "17:00") & (data['active_power']<=10) & (data['irradiance']>10) & (data['alarm_name'].notnull())) , 'yes', 'no')
3. All the alarms raised which are not considered into 'Trips' are considered as 'no-Trip' in binary classification.

Performance Matrix

Key performance parameters are calculated from the confusion metrics (as described below) and they are - True Positives, True Negatives, False Positives, False Negatives.

- **True Positives (TP)** - These are the correctly predicted positive values which means that the value of the actual class is yes and the value of the predicted class is also yes.
- **True Negatives (TN)** - These are the correctly predicted negative values which means that the value of the actual class is no and the value of predicted class is also no.
- **False Positives (FP)** – When actual class is no and predicted class is yes.
- **False Negatives (FN)** – When actual class is yes but predicted class in no.

		Predicted	
		Negative	**Positive**
Actual	**Negative**	True Negative	False Positive
	Positive	False Negative	True Positive

False positives and false negatives, these values occur when your actual class contradicts with the predicted class.

Based -on the confusion matrix, Accuracy, Precision, Recall, and F-1 Score are calculated and they are defined as:

- **Accuracy** - Accuracy is the most intuitive performance measure and it is simply a ratio of correctly predicted observation to the total observations. One may think that, if we have high accuracy then our model is best. Yes, accuracy is a great measure but only when we have symmetric datasets where values of false positive and false negatives are almost the same. Therefore, we have to look at other parameters to evaluate the performance of the model.

- **Precision** - Precision is the ratio of correctly predicted positive observations to the total predicted positive observations.

- **Recall** (Sensitivity) - Recall is the ratio of correctly predicted positive observations to all observations in actual class - yes.

$$\text{Precision} = \frac{True\ Positive}{True\ Positive + False\ Positive}$$

$$\text{Recall} = \frac{True\ Positive}{True\ Positive + False\ Negative}$$

- **F1 score** - F1 Score is the weighted average of Precision and Recall. Therefore, this score takes both false positives and false negatives into account. Intuitively it is not as easy to understand as accuracy, but F1 is usually more useful than accuracy, especially if we have an uneven class distribution.

$$F1 = 2 \times \frac{Precision * Recall}{Precision + Recall}$$

Methodology

The following methods have been attempted to solve above mentioned problem statement:

ID	Methods	Description
1	Method 1: Survival analysis	Statistical approach using KMF() to predict the probability of failure for each inverter.
2	Method 2: Inverter-wise, LSTM model, binary classification no feature engineering	LSTM-based binary classification is used with 60 lookback days with oversampling techniques. This approach considered only one selected inverter. Here, X_train and Y_train had different data-frequency.
3	Method 3: Inverter-wise, LSTM model, binary classification with feature engineering	LSTM-based binary classification is used with 60 lookback days with oversampling techniques. This approach considered only one selected inverter. Here, X_train and Y_train have the same data-frequency (30-min or 60-min).
4	Method 4: LSTM, multi- classification with feature engineering	LSTM-based multi-classification is used with 60 lookback days with oversampling techniques. This approach considered only one selected inverter. Here, X_train and Y_train have the same data-frequency (60-min).
5	Method 5: Plant-wise, XGBoost, binary classification, resampling of 60 min with no lookback	Post data-frequency of 60 min and feature engineering, XGBoost based binary classification is used with no lookback days.
6	Method 6: Plant-wise, XGBoost, binary classification, resampling of 24 hours with no lookback	Post data-frequency of 24 hours and feature engineering, XGBoost based binary classification is used with no lookback days.
7	Method 7: Plant-wise, XGBoost, binary classification, resampling of 24 hours with 60 days lookback	Post data-frequency of 24 hours and feature engineering, XGBoost based binary classification is used with 60 days of lookback.
8	Method 8: Plant-wise, LightGBM, binary classification, resampling of 24 hours with 60 days lookback	Post data-frequency of 24 hours and feature engineering, LightGBM based binary classification is used with 60 days of lookback.
9	Method 9: Plant-wise, AdaBoost, binary classification, resampling of 24 hours with 60 days lookback	Post data-frequency of 24 hours and feature engineering, Ensemble techniques based binary classification is used with 60 days of lookback.

10	Method 10: Plant-wise, Voting Classifier, binary classification, resampling of 24 hours with 60 days lookback	Post data-frequency of 24 hours and feature engineering, Voting Classifier based binary classification is used with 60 days of lookback.
11	Method 11: Plant-wise, Random Forest Classifier, binary classification, resampling of 24 hours with 60 days lookback	Post data-frequency of 24 hours and feature engineering, Random-Forest Classifier based binary classification is used with 60 days of lookback.
12	Method 12: Plant-wise, AdaBoost Classifier, multi-classification, resampling of 24 hours with 60 days lookback	Post data-frequency of 24 hours and feature engineering, AdaBoost Classifier based multi-classification is used with 60 days of lookback.
13	Method 13: Plant-wise, LightGBM Classifier, multi-classification, resampling of 24 hours with 60 days lookback	Post data-frequency of 24 hours and feature engineering, LightGBM Classifier based multi-classification is used with 60 days of lookback.
14	Method 14: Plant-wise, XGBoost Classifier, multi-classification, resampling of 24 hours with 60 days lookback	Post data-frequency of 24 hours and feature engineering, XGBoost Classifier based multi-classification is used with 60 days of lookback.
15	Method 15: Selected-inverter, AdaBoost Classifier, multi-classification, resampling of 24 hours with 60 days lookback	Selected only 35 inverters with ‘trips’ in the data. Post data-frequency of 24 hours and feature engineering, AdaBoost Classifier based multi-classification is used with 60 days of lookback.
16	Method 16: MLP Classifier, binary-classification, resampling of 24 hours with 60 days lookback	Post data-frequency of 24 hours and feature engineering, MLP Classifier based multi-classification is used with 60 days of lookback.
17	Method 17: SGD Classifier, binary-classification, resampling of 24 hours with 60 days lookback	Post data-frequency of 24 hours and feature engineering, SGD Classifier based multi-classification is used with 60 days of lookback.
18	Method 18: Gradient Boosting Classifier, binary-classification, resampling of 24 hours with 60 days lookback	Post data-frequency of 24 hours and feature engineering, Gradient Boosting Classifier based multi-classification is used with 60 days of lookback.

Method 1: Survival Analysis

Steps	**Description**
Data\ Engineering	Merge all the files for the 24 hours * 365 data (5-min frequency). Use all the raw features from device_data and sun_data. Keep Categorical features from fault_data. ● Fill null_values and missing values. ● Remove duplicates. ● Remove outliers.
Trips Selection Logic	Consider the data from 7am to 5pm (5-min frequency). Use following fault selection logic to create a 'Target': data['alarm_class'] = np.where(((data['Time'] >= "07:00") & (data['Time'] <= "17:00") & (data['active_power']=0) & (data['irradiance']>50) & (data['alarm_name'].notnull())) , 'yes', 'no')
X_train and Y_train	Select the data only in between 7am to 5pm. Resample the whole data for 10 hours -> i.e. one device will have one row per 10 hours (selected window). Create 'Duration' and 'Event Observed' columns.
Model	Create a For loop for inverters: - Select data for each inverter in a dataframe. - Create a duration (delta time) column for each inverter. It will be used as Duration/time in the function - Select 'Target'. - Use from lifelines import KaplanMeierFitter() - Use the model and predict the next day's failure probability.
Results	Predictions (for probability of failure) are not good as most of the predictions are indicating no faults. This will need further improvements.

Method 2: LSTM model, binary classification, no feature engineering

Steps	**Description**
Data Engineering	Merge all the files for the 24 hours * 365 data (5-min frequency). Use all the raw features from device_data and sun_data. Keep Categorical features from fault_data. ● Fill null_values and missing values. ● Remove duplicates. ● Remove outliers.
Trips Selection Logic	Consider the data from 7am to 5pm (5-min frequency). Use following fault selection logic to create a 'Target': data['alarm_class'] = np.where(((data['Time'] >= "07:00") & (data['Time'] <= "17:00") & (data['active_power']=0) & (data['irradiance']>50) & (data['alarm_name'].notnull())) , 'yes', 'no')
X_train and Y_train	Select the data only in between 7am to 5pm. X_train has data frequency of 15 min Y_train has data frequency of 24 hour
System	Inverter - focused
Model	LSTM model with 5 hidden layers, dropout, and sigmoid activation function
Results	Training accuracy and F1 score had lower values therefore, the prediction results' are not-satisfactory for this method.

Method 3: LSTM model, binary classification with feature engineering

Steps	**Description**
Data Engineering	Merge all the files for the 24 hours * 365 data (5-min frequency). Use all the raw features from device_data and sun_data. Keep Categorical features from fault_data. ● Fill null_values and missing values. ● Remove duplicates. ● Remove outliers.
Trips Selection Logic	Consider the data from 7am to 5pm (5-min frequency). Use following fault selection logic to create a 'Target': data['alarm_class'] = np.where(((data['Time'] >= "07:00") & (data['Time'] <= "17:00") & (data['active_power']=0) & (data['irradiance']>50) & (data['alarm_name'].isnull())) , 'yes', 'no')
X_train and Y_train	Select the data only in between 7am to 5pm. ● Input Data Frequency = 60 mins. ● Output Data Frequency (Labels) = 24 hrs (1 label/ per day). ● Look back = 60 days (Only labels) ● Sampling technique = SMOTE (samping_strategy=0.6, k_neighbors=2). ● Feature Engineering = Yes, New feature addition
System	Inverter - focused
Model	LSTM model with 5 hidden layers, dropout, and sigmoid activation function
Results	Training accuracy and F1 score had lower values. Overall, results are not acceptable.

Method 4: LSTM, multi- classification with feature engineering

Steps	**Description**
Data Engineering	Merge all the files for the 24 hours * 365 data (5-min frequency). Use all the raw features from device_data and sun_data. Keep Categorical features from fault_data. ● Fill null_values and missing values. ● Remove duplicates. ● Remove outliers.
Trips Selection Logic	Consider the data from 7am to 5pm (5-min frequency). Use following fault selection logic to create a 'Target': data['alarm_class'] = np.where(((data['Time'] >= "07:00") & (data['Time'] <= "17:00") & (data['active_power']=0) & (data['irradiance']>50) & (data['alarm_name'].notnull())) , 'yes', 'no')
X_train and Y_train	Select the data only in between 7am to 5pm.
System	Inverter - focused (used Inverter 3 only)
Model	LSTM model with 5 hidden layers, dropout, and sigmoid activation function
Results	Training accuracy and F1 score had lower values therefore, the prediction results are not acceptable.

Method 5: XGBoost, binary classification, resampling of 60 min with no lookback

Steps	**Description**
Data Engineering	Merge all the files for the 24 hours * 365 data (5-min frequency). Use all the raw features from device_data and sun_data. Keep Categorical features from fault_data. ● Fill null_values and missing values. ● Remove duplicates. ● Remove outliers.
Trips Selection Logic	Consider the data from 7am to 5pm (5-min frequency). Use following fault selection logic to create a 'Target': data['alarm_class'] = np.where(((data['Time'] >= "07:00") & (data['Time'] <= "17:00") & (data['active_power']=0) & (data['irradiance']>50) & (data['alarm_name'].notnull())) , 'yes', 'no')
X_train and Y_train	Select the data only in between 7am to 5pm. X_train has data frequency of 15 min Y_train has data frequency of 24 hour
System	Inverter-wise prediction for 45 inverters
Model	XGBoost with SMOTETomek
Results	False Positives are higher. Monthly metrics: ● Accuracy: 96% ● Precision: 3% ● Recall: 22% ● F-1 Score: 5%

For all the 45 inverters in the GMR plant in the month of January 2021, total number of TPs are 15, total FNs are 54, FPs are 489, and TNs 14,292. It means that the model has predicted about 504 'Trips' for actual 'Trips' of 69 out of which total of 15 'Trips' were True Positives, a total of 54 'Trips' were False Negatives, and a total of 489 'Trips' were False Positives.

id	date	tn	fp	fn	tp	Accuracy	Precision	Recall	F-1Score	Actual_model_trips	Predicted_model_trips	Diff
0	01-01-2021	490	0	5	0	99%	#DIV/0!	0%	#DIV/0!	5	0	5
1	02-01-2021	481	10	4	0	97%	0%	0%	#DIV/0!	4	10	-6
2	03-01-2021	490	2	2	1	99%	33%	33%	33%	3	3	0
3	04-01-2021	486	0	9	0	98%	#DIV/0!	0%	#DIV/0!	9	0	9
4	05-01-2021	490	3	0	2	99%	40%	100%	57%	2	5	-3
5	06-01-2021	494	1	0	0	100%	0%	#DIV/0!	#DIV/0!	0	1	-1
6	07-01-2021	494	0	1	0	100%	#DIV/0!	0%	#DIV/0!	1	0	1
7	08-01-2021	485	9	0	1	98%	10%	100%	18%	1	10	-9
8	09-01-2021	482	9	3	1	98%	10%	25%	14%	4	10	-6
9	10-01-2021	483	11	0	1	98%	8%	100%	15%	1	12	-11
10	11-01-2021	490	2	3	0	99%	0%	0%	#DIV/0!	3	2	1
11	12-01-2021	483	11	0	1	98%	8%	100%	15%	1	12	-11
12	13-01-2021	478	9	7	1	97%	10%	13%	11%	8	10	-2
13	14-01-2021	494	0	1	0	100%	#DIV/0!	0%	#DIV/0!	1	0	1
14	15-01-2021	484	10	1	0	98%	0%	0%	#DIV/0!	1	10	-9
15	16-01-2021	485	10	0	0	98%	0%	#DIV/0!	#DIV/0!	0	10	-10
16	17-01-2021	464	30	1	0	94%	0%	0%	#DIV/0!	1	30	-29
17	18-01-2021	400	90	5	0	81%	0%	0%	#DIV/0!	5	90	-85
18	19-01-2021	490	4	1	0	99%	0%	0%	#DIV/0!	1	4	-3
19	20-01-2021	351	144	0	0	71%	0%	#DIV/0!	#DIV/0!	0	144	-144
20	21-01-2021	485	9	1	0	98%	0%	0%	#DIV/0!	1	9	-8
21	22-01-2021	484	10	1	0	98%	0%	0%	#DIV/0!	1	10	-9
22	23-01-2021	426	69	0	0	86%	0%	#DIV/0!	#DIV/0!	0	69	-69
23	24-01-2021	484	9	2	0	98%	0%	0%	#DIV/0!	2	9	-7
24	25-01-2021	493	0	2	0	100%	#DIV/0!	0%	#DIV/0!	2	0	2
25	26-01-2021	484	9	0	2	98%	18%	100%	31%	2	11	-9
26	27-01-2021	493	0	2	0	100%	#DIV/0!	0%	#DIV/0!	2	0	2
27	28-01-2021	482	9	2	2	98%	18%	50%	27%	4	11	-7
28	29-01-2021	482	10	0	3	98%	23%	100%	38%	3	13	-10
29	30-01-2021	485	9	1	0	98%	0%	0%	#DIV/0!	1	9	-8
	MONTHLY	14292	489	54	15	96%	3%	22%	5%	69	504	-435

Method 6: XGBoost, binary classification, resampling of 24 hour with no lookback

Steps	**Description**
Data Engineering	Merge all the files for the 24 hours * 365 data (5-min frequency). Use all the raw features from device_data and sun_data. Keep Categorical features from fault_data. ● Fill null_values and missing values. ● Remove duplicates. ● Remove outliers.
Trips Selection Logic	Consider the data from 7am to 5pm (5-min frequency). Use following fault selection logic to create a 'Target': data['alarm_class'] = np.where(((data['Time'] >= "07:00") & (data['Time'] <= "17:00") & (data['active_power']=0) & (data['irradiance']>50) & (data['alarm_name'].notnull())) , 'yes', 'no')
X_train and Y_train	Select the data only in between 7am to 5pm.
System	Inverter-wise prediction for 45 inverters
Model	XGBoost with SMOTETomek
Results	False positives are lesser than Model 5. Monthly metrics: ● Accuracy: 93% ● Precision: 24% ● Recall: 46% ● F-1 Score: 31%

For all the 45 inverters in the GMR plant in the month of January 2021, total number of TPs are 23, total FNs are 27, FPs are 74, and TNs 1226. It means that the model has predicted about 97 'Trips' for actual 'Trips' of 50 out of which total of 23 'Trips' were True Positives,
a total of 27 'Trips' were False Negatives, and a total of 74 'Trips' were False Positives.

Id	date	tn	fp	fn	tp	Accuracy	Precision	Recall	F-1Score	Actual_model_trips	Predicted_model_trips	Diff
0	01-01-2021	44	0	1	0	98%	#DIV/0!	0%	#DIV/0!	1	0	1
1	02-01-2021	42	0	2	1	96%	100%	33%	50%	3	1	2
2	03-01-2021	40	3	0	2	93%	40%	100%	57%	2	5	-3
3	04-01-2021	37	0	6	2	87%	100%	25%	40%	8	2	6
4	05-01-2021	36	8	0	1	82%	11%	100%	20%	1	9	-8
5	06-01-2021	42	3	0	0	93%	0%	#DIV/0!	#DIV/0!	0	3	-3
6	07-01-2021	44	0	1	0	98%	#DIV/0!	0%	#DIV/0!	1	0	1
7	08-01-2021	44	0	0	1	100%	100%	100%	100%	1	1	0
8	09-01-2021	40	1	3	1	91%	50%	25%	33%	4	2	2
9	10-01-2021	40	4	0	1	91%	20%	100%	33%	1	5	-4
10	11-01-2021	43	0	1	1	98%	100%	50%	67%	2	1	1
11	12-01-2021	43	1	0	1	98%	50%	100%	67%	1	2	-1
12	13-01-2021	36	2	5	2	84%	50%	29%	36%	7	4	3
13	14-01-2021	37	7	1	0	82%	0%	0%	#DIV/0!	1	7	-6
14	15-01-2021	44	0	0	1	100%	100%	100%	100%	1	1	0
15	16-01-2021	41	4	0	0	91%	0%	#DIV/0!	#DIV/0!	0	4	-4
16	17-01-2021	37	7	0	1	84%	13%	100%	22%	1	8	-7
17	18-01-2021	38	2	5	0	84%	0%	0%	#DIV/0!	5	2	3
18	19-01-2021	37	7	0	1	84%	13%	100%	22%	1	8	-7
19	20-01-2021	41	4	0	0	91%	0%	#DIV/0!	#DIV/0!	0	4	-4
20	21-01-2021	43	1	1	0	96%	0%	0%	#DIV/0!	1	1	0
21	22-01-2021	42	2	1	0	93%	0%	0%	#DIV/0!	1	2	-1
22	23-01-2021	40	5	0	0	89%	0%	#DIV/0!	#DIV/0!	0	5	-5
23	24-01-2021	43	1	0	1	98%	50%	100%	67%	1	2	-1
24	25-01-2021	40	4	0	1	91%	20%	100%	33%	1	5	-4
25	26-01-2021	43	1	0	1	98%	50%	100%	67%	1	2	-1
26	27-01-2021	41	3	0	1	93%	25%	100%	40%	1	4	-3
27	28-01-2021	41	3	0	1	93%	25%	100%	40%	1	4	-3
28	29-01-2021	43	1	0	1	98%	50%	100%	67%	1	2	-1
29	30-01-2021	44	0	0	1	100%	100%	100%	100%	1	1	0
	MONTHLY	1226	74	27	23	93%	24%	46%	31%	50	97	-47

Method 7: XGBoost, binary classification, resampling of 24 hour with 60 days lookback

Steps	**Description**
Data Engineering	Merge all the files for the 24 hours * 365 data (5-min frequency). Use all the raw features from device_data and sun_data. Keep Categorical features from fault_data. ● Fill null_values and missing values. ● Remove duplicates. ● Remove outliers.
Trips Selection Logic	Consider the data from 7am to 5pm (5-min frequency). Use following fault selection logic to create a 'Target': data['alarm_class'] = np.where(((data['Time'] >= "07:00") & (data['Time'] <= "17:00") & (data['active_power']=0) & (data['irradiance']>50) & (data['alarm_name'].notnull())) , 'yes', 'no')
X_train and Y_train	Select the data only in between 7am to 5pm. 60-days of lookback (labels) is considered in X_train.
System	Inverter-wise prediction for 45 inverters
Model	XGBoost XGBClassifier(is_unbalance = True)
Results	True Positives are lesser. Monthly metrics: ● Accuracy: 96% ● Precision: 40% ● Recall: 20% ● F-1 Score: 27%

For all the 45 inverters in the GMR plant in the month of January 2021, total number of TPs are 25, total FNs are 25, FPs are 44, and TNs 1256. It means that the model has predicted about 25 'Trips' for actual 'Trips' of 50 out of which total of 10 'Trips' were True Positives,
a total of 40 'Trips' were False Negatives, and a total of 15 'Trips' were False Positives.

ID	date	tn	fp	fn	tp	Accuracy	Precision	Recall	F-1Score	Actual_model_trips	Predicted_model_trips	Diff
0	01-01-2021	44	0	1	0	98%	#DIV/0!	0%	#DIV/0!	1	0	1
1	02-01-2021	42	0	2	1	96%	100%	33%	50%	3	1	2
2	03-01-2021	43	0	1	1	98%	100%	50%	67%	2	1	1
3	04-01-2021	37	0	8	0	82%	#DIV/0!	0%	#DIV/0!	8	0	8
4	05-01-2021	37	7	0	1	84%	13%	100%	22%	1	8	-7
5	06-01-2021	44	1	0	0	98%	0%	#DIV/0!	#DIV/0!	0	1	-1
6	07-01-2021	44	0	1	0	98%	#DIV/0!	0%	#DIV/0!	1	0	1
7	08-01-2021	44	0	1	0	98%	#DIV/0!	0%	#DIV/0!	1	0	1
8	09-01-2021	41	0	4	0	91%	#DIV/0!	0%	#DIV/0!	4	0	4
9	10-01-2021	43	1	0	1	98%	50%	100%	67%	1	2	-1
10	11-01-2021	43	0	2	0	96%	#DIV/0!	0%	#DIV/0!	2	0	2
11	12-01-2021	43	1	1	0	96%	0%	0%	#DIV/0!	1	1	0
12	13-01-2021	37	1	7	0	82%	0%	0%	#DIV/0!	7	1	6
13	14-01-2021	44	0	1	0	98%	#DIV/0!	0%	#DIV/0!	1	0	1
14	15-01-2021	44	0	0	1	100%	100%	100%	100%	1	1	0
15	16-01-2021	45	0	0	0	100%	#DIV/0!	#DIV/0!	#DIV/0!	0	0	0
16	17-01-2021	44	0	1	0	98%	#DIV/0!	0%	#DIV/0!	1	0	1
17	18-01-2021	39	1	5	0	87%	0%	0%	#DIV/0!	5	1	4
18	19-01-2021	43	1	1	0	96%	0%	0%	#DIV/0!	1	1	0
19	20-01-2021	43	2	0	0	96%	0%	#DIV/0!	#DIV/0!	0	2	-2
20	21-01-2021	44	0	1	0	98%	#DIV/0!	0%	#DIV/0!	1	0	1
21	22-01-2021	44	0	1	0	98%	#DIV/0!	0%	#DIV/0!	1	0	1
22	23-01-2021	45	0	0	0	100%	#DIV/0!	#DIV/0!	#DIV/0!	0	0	0
23	24-01-2021	44	0	1	0	98%	#DIV/0!	0%	#DIV/0!	1	0	1
24	25-01-2021	44	0	1	0	98%	#DIV/0!	0%	#DIV/0!	1	0	1
25	26-01-2021	44	0	0	1	100%	100%	100%	100%	1	1	0
26	27-01-2021	44	0	0	1	100%	100%	100%	100%	1	1	0
27	28-01-2021	44	0	0	1	100%	100%	100%	100%	1	1	0
28	29-01-2021	44	0	0	1	100%	100%	100%	100%	1	1	0
29	30-01-2021	44	0	0	1	100%	100%	100%	100%	1	1	0
	MONTHLY	1285	15	40	10	96%	40%	20%	27%	50	25	25

Method 8: LightGBM, binary classification, resampling of 24 hour with 60 days lookback

Steps	**Description**
Data Engineering	Merge all the files for the 24 hours * 365 data (5-min frequency). Use all the raw features from device_data and sun_data. Keep Categorical features from fault_data. ● Fill null_values and missing values. ● Remove duplicates. ● Remove outliers.
Trips Selection Logic	Consider the data from 7am to 5pm (5-min frequency). Use following fault selection logic to create a 'Target': data['alarm_class'] = np.where(((data['Time'] >= "07:00") & (data['Time'] <= "17:00") & (data['active_power']=0) & (data['irradiance']>50) & (data['alarm_name'].notnull())) , 'yes', 'no')
X_train and Y_train	Select the data only in between 7am to 5pm. 60-days of lookback (labels) is considered in X_train.
System	Inverter-wise prediction for 45 inverters
Model	XGBClassifier(is_unbalance = True)
Results	Monthly metrics: ● Accuracy: 95% ● Precision: 36% ● Recall: 50% ● F-1 Score: 42%

For all the 45 inverters in the GMR plant in the month of January 2021, total number of TPs are 25, total FNs are 25, FPs are 44, and TNs 1256. It means that the model has predicted about 69 'Trips' for actual 'Trips' of 50 out of which total of 25 'Trips' were True Positives,
a total of 25 'Trips' were False Negatives, and a total of 44 'Trips' were False Positives.

Id	date	tn	fp	fn	tp	Accuracy	Precision	Recall	F-1Score	Actual_model_trips	Predicted_model_trips	Diff
0	01-01-2021	44	0	0	1	100%	100%	100%	100%	1	1	0
1	02-01-2021	42	0	0	3	100%	100%	100%	100%	3	3	0
2	03-01-2021	43	0	0	2	100%	100%	100%	100%	2	2	0
3	04-01-2021	37	0	6	2	87%	100%	25%	40%	8	2	6
4	05-01-2021	32	12	0	1	73%	8%	100%	14%	1	13	-12
5	06-01-2021	43	2	0	0	96%	0%	#DIV/0!	#DIV/0!	0	2	-2
6	07-01-2021	43	1	0	1	98%	50%	100%	67%	1	2	-1
7	08-01-2021	44	0	1	0	98%	#DIV/0!	0%	#DIV/0!	1	0	1
8	09-01-2021	41	0	3	1	93%	100%	25%	40%	4	1	3
9	10-01-2021	43	1	1	0	96%	0%	0%	#DIV/0!	1	1	0
10	11-01-2021	43	0	0	2	100%	100%	100%	100%	2	2	0
11	12-01-2021	43	1	0	1	98%	50%	100%	67%	1	2	-1
12	13-01-2021	37	1	7	0	82%	0%	0%	#DIV/0!	7	1	6
13	14-01-2021	40	4	0	1	91%	20%	100%	33%	1	5	-4
14	15-01-2021	43	1	0	1	98%	50%	100%	67%	1	2	-1
15	16-01-2021	42	3	0	0	93%	0%	#DIV/0!	#DIV/0!	0	3	-3
16	17-01-2021	44	0	0	1	100%	100%	100%	100%	1	1	0
17	18-01-2021	39	1	4	1	89%	50%	20%	29%	5	2	3
18	19-01-2021	41	3	0	1	93%	25%	100%	40%	1	4	-3
19	20-01-2021	43	2	0	0	96%	0%	#DIV/0!	#DIV/0!	0	2	-2
20	21-01-2021	42	2	1	0	93%	0%	0%	#DIV/0!	1	2	-1
21	22-01-2021	41	3	1	0	91%	0%	0%	#DIV/0!	1	3	-2
22	23-01-2021	44	1	0	0	98%	0%	#DIV/0!	#DIV/0!	0	1	-1
23	24-01-2021	43	1	1	0	96%	0%	0%	#DIV/0!	1	1	0
24	25-01-2021	43	1	0	1	98%	50%	100%	67%	1	2	-1
25	26-01-2021	40	4	0	1	91%	20%	100%	33%	1	5	-4
26	27-01-2021	44	0	0	1	100%	100%	100%	100%	1	1	0
27	28-01-2021	44	0	0	1	100%	100%	100%	100%	1	1	0
28	29-01-2021	44	0	0	1	100%	100%	100%	100%	1	1	0
29	30-01-2021	44	0	0	1	100%	100%	100%	100%	1	1	0
	MONTHLY	1256	44	25	25	95%	36%	50%	42%	50	69	-19

Method 9: AdaBoost, binary classification, resampling of 24 hour with 60 days lookback

Steps	**Description**
Data Engineering	Merge all the files for the 24 hours * 365 data (5-min frequency). Use all the raw features from device_data and sun_data. Keep Categorical features from fault_data. ● Fill null_values and missing values. ● Remove duplicates. ● Remove outliers.
Trips Selection Logic	Consider the data from 7am to 5pm (5-min frequency). Use following fault selection logic to create a 'Target': data['alarm_class'] = np.where(((data['Time'] >= "07:00") & (data['Time'] <= "17:00") & (data['active_power']=0) & (data['irradiance']>50) & (data['alarm_name'].notnull())) , 'yes', 'no')
X_train and Y_train	Select the data only in between 7am to 5pm. 60-days of lookback (labels) is considered in X_train.
System	Inverter-wise prediction for 45 inverters
Model	AdaBoostClassifier with SMOTETomek
Results	Monthly metrics: ● Accuracy: 91% ● Precision: 22% ● Recall: 52% ● F-1 Score: 31%

For all the 45 inverters in the GMR plant in the month of January 2021, total number of TPs are 26, total FNs are 24, FPs are 91, and TNs 1209. It means that the model has predicted about 117 'Trips' for actual 'Trips' of 50 out of which total of 26 'Trips' were True Positives, a total of 24 'Trips' were False Negatives, and a total of 91 'Trips' were False Positives.

For 10 days in the month, the model has recorded a daily F-1 score above 50% with daily Accuracy above 90%. Additionally, For 10 days in the month, the model has recorded a daily F-1 score in between 15% - 40% with daily Accuracy inthe range 73% - 93%.

For 16 days in the month, the model has recorded a daily Recall score above 50% with daily Precision in the range of 8% to 100%. Additionally, For 4 days in the month, the model has recorded a daily Recall score in between 14% to 33% with daily Precision in the range of 17% to 100%.

ID	date	tn	fp	fn	tp	Accuracy	Precision	Recall	F-1Score	Actual_model_trips	Predicted_model_trips	Diff
0	01-01-2021	44	0	1	0	98%	#DIV/0!	0%	#DIV/0!	1	0	1
1	02-01-2021	42	0	2	1	96%	100%	33%	50%	3	1	2
2	03-01-2021	39	4	0	2	91%	33%	100%	50%	2	6	-4
3	04-01-2021	35	2	3	5	89%	71%	63%	67%	8	7	1
4	05-01-2021	33	11	0	1	76%	8%	100%	15%	1	12	-11
5	06-01-2021	41	4	0	0	91%	0%	#DIV/0!	#DIV/0!	0	4	-4
6	07-01-2021	40	4	1	0	89%	0%	0%	#DIV/0!	1	4	-3
7	08-01-2021	43	1	1	0	96%	0%	0%	#DIV/0!	1	1	0
8	09-01-2021	41	0	3	1	93%	100%	25%	40%	4	1	3
9	10-01-2021	35	9	0	1	80%	10%	100%	18%	1	10	-9
10	11-01-2021	40	3	0	2	93%	40%	100%	57%	2	5	-3
11	12-01-2021	43	1	1	0	96%	0%	0%	#DIV/0!	1	1	0
12	13-01-2021	36	2	6	1	82%	33%	14%	20%	7	3	4
13	14-01-2021	39	5	0	1	89%	17%	100%	29%	1	6	-5
14	15-01-2021	43	1	0	1	98%	50%	100%	67%	1	2	-1
15	16-01-2021	42	3	0	0	93%	0%	#DIV/0!	#DIV/0!	0	3	-3
16	17-01-2021	37	7	0	1	84%	13%	100%	22%	1	8	-7
17	18-01-2021	35	5	4	1	80%	17%	20%	18%	5	6	-1
18	19-01-2021	42	2	1	0	93%	0%	0%	#DIV/0!	1	2	-1
19	20-01-2021	43	2	0	0	96%	0%	#DIV/0!	#DIV/0!	0	2	-2
20	21-01-2021	41	3	0	1	93%	25%	100%	40%	1	4	-3
21	22-01-2021	40	4	1	0	89%	0%	0%	#DIV/0!	1	4	-3
22	23-01-2021	41	4	0	0	91%	0%	#DIV/0!	#DIV/0!	0	4	-4
23	24-01-2021	37	7	0	1	84%	13%	100%	22%	1	8	-7
24	25-01-2021	44	0	0	1	100%	100%	100%	100%	1	1	0
25	26-01-2021	42	2	0	1	96%	33%	100%	50%	1	3	-2
26	27-01-2021	41	3	0	1	93%	25%	100%	40%	1	4	-3
27	28-01-2021	44	0	0	1	100%	100%	100%	100%	1	1	0
28	29-01-2021	44	0	0	1	100%	100%	100%	100%	1	1	0
29	30-01-2021	42	2	0	1	96%	33%	100%	50%	1	3	-2
	MONTHLY	1209	91	24	26	91%	22%	52%	31%	50	117	-67

Method 10: Voting Classifier, binary classification, resampling of 24 hour with 60 days lookback

Steps	**Description**
Data Engineering	Merge all the files for the 24 hours * 365 data (5-min frequency). Use all the raw features from device_data and sun_data. Keep Categorical features from fault_data. ● Fill null_values and missing values. ● Remove duplicates. ● Remove outliers.
Trips Selection Logic	Consider the data from 7am to 5pm (5-min frequency). Use following fault selection logic to create a 'Target': data['alarm_class'] = np.where(((data['Time'] >= "07:00") & (data['Time'] <= "17:00") & (data['active_power']=0) & (data['irradiance']>50) & (data['alarm_name'].notnull())) , 'yes', 'no')
X_train and Y_train	Select the data only in between 7am to 5pm. 60-days of lookback (labels) is considered in X_train.
System	Inverter-wise prediction for 45 inverters
Model	Voting Classifier with SMOTETomek
Results	True Positives are lesser and False Positives are relatively higher. Monthly metrics: ● Accuracy: 96% ● Precision: 54% ● Recall: 30% ● F-1 Score: 38%

For all the 45 inverters in the GMR plant in the month of January 2021, total number of TPs are 15, total FNs are 35, FPs are 13, and TNs 1287. It means that the model has predicted about 28 'Trips' for actual 'Trips' of 50 out of which total of 15 'Trips' were True Positives,
a total of 35 'Trips' were False Negatives, and a total of 13 'Trips' were False Positives.

id	date	tn	fp	fn	tp	Accuracy	Precision	Recall	F-1Score	Actual_model_trips	Predicted_model_trips	Diff
0	01-01-2021	44	0	1	0	98%	#DIV/0!	0%	#DIV/0!	1	0	1
1	02-01-2021	42	0	2	1	96%	100%	33%	50%	3	1	2
2	03-01-2021	43	0	0	2	100%	100%	100%	100%	2	2	0
3	04-01-2021	37	0	7	1	84%	100%	13%	22%	8	1	7
4	05-01-2021	38	6	0	1	87%	14%	100%	25%	1	7	-6
5	06-01-2021	44	1	0	0	98%	0%	#DIV/0!	#DIV/0!	0	1	-1
6	07-01-2021	44	0	1	0	98%	#DIV/0!	0%	#DIV/0!	1	0	1
7	08-01-2021	44	0	1	0	98%	#DIV/0!	0%	#DIV/0!	1	0	1
8	09-01-2021	41	0	4	0	91%	#DIV/0!	0%	#DIV/0!	4	0	4
9	10-01-2021	43	1	1	0	96%	0%	0%	#DIV/0!	1	1	0
10	11-01-2021	43	0	1	1	98%	100%	50%	67%	2	1	1
11	12-01-2021	43	1	0	1	98%	50%	100%	67%	1	2	-1
12	13-01-2021	37	1	7	0	82%	0%	0%	#DIV/0!	7	1	6
13	14-01-2021	43	1	1	0	96%	0%	0%	#DIV/0!	1	1	0
14	15-01-2021	44	0	0	1	100%	100%	100%	100%	1	1	0
15	16-01-2021	45	0	0	0	100%	#DIV/0!	#DIV/0!	#DIV/0!	0	0	0
16	17-01-2021	44	0	1	0	98%	#DIV/0!	0%	#DIV/0!	1	0	1
17	18-01-2021	40	0	4	1	91%	100%	20%	33%	5	1	4
18	19-01-2021	43	1	1	0	96%	0%	0%	#DIV/0!	1	1	0
19	20-01-2021	44	1	0	0	98%	0%	#DIV/0!	#DIV/0!	0	1	-1
20	21-01-2021	44	0	1	0	98%	#DIV/0!	0%	#DIV/0!	1	0	1
21	22-01-2021	44	0	1	0	98%	#DIV/0!	0%	#DIV/0!	1	0	1
22	23-01-2021	45	0	0	0	100%	#DIV/0!	#DIV/0!	#DIV/0!	0	0	0
23	24-01-2021	44	0	1	0	98%	#DIV/0!	0%	#DIV/0!	1	0	1
24	25-01-2021	44	0	0	1	100%	100%	100%	100%	1	1	0
25	26-01-2021	44	0	0	1	100%	100%	100%	100%	1	1	0
26	27-01-2021	44	0	0	1	100%	100%	100%	100%	1	1	0
27	28-01-2021	44	0	0	1	100%	100%	100%	100%	1	1	0
28	29-01-2021	44	0	0	1	100%	100%	100%	100%	1	1	0
29	30-01-2021	44	0	0	1	100%	100%	100%	100%	1	1	0
	MONTHLY	1287	13	35	15	96%	54%	30%	38%	50	28	22

Method 11: Random Forest Classifier, binary classification, resampling of 24 hour with 60 days lookback

Steps	**Description**
Data Engineering	Merge all the files for the 24 hours * 365 data (5-min frequency). Use all the raw features from device_data and sun_data. Keep Categorical features from fault_data. ● Fill null_values and missing values. ● Remove duplicates. ● Remove outliers.
Trips Selection Logic	Consider the data from 7am to 5pm (5-min frequency). Use following fault selection logic to create a 'Target': data['alarm_class'] = np.where(((data['Time'] >= "07:00") & (data['Time'] <= "17:00") & (data['active_power']=0) & (data['irradiance']>50) & (data['alarm_name'].notnull())) , 'yes', 'no')
X_train and Y_train	Select the data only in between 7am to 5pm. 60-days of lookback (labels) is considered in X_train.
System	Inverter-wise prediction for 45 inverters
Model	Random Forest Classifier with SMOTETomek
Results	True Positives are lesser and False Positives are relatively higher. Monthly metrics: ● Accuracy: 96% ● Precision: 48% ● Recall: 42% ● F-1 Score: 45%

For all the 45 inverters in the GMR plant in the month of January 2021, total number of TPs are 21, total FNs are 29, FPs are 23, and TNs 1277. It means that the model has predicted about 44 'Trips' for actual 'Trips' of 50 out of which total of 21 'Trips' were True Positives,
a total of 29 'Trips' were False Negatives, and a total of 23 'Trips' were False Positives.

Id	date	tn	fp	fn	tp	Accuracy	Precision	Recall	F-1Score	Actual_model_trips	Predicted_model_trips	Diff
0	01-01-2021	44	0	0	1	100%	100%	100%	100%	1	1	0
1	02-01-2021	42	0	0	3	100%	100%	100%	100%	3	3	0
2	03-01-2021	43	0	0	2	100%	100%	100%	100%	2	2	0
3	04-01-2021	37	0	8	0	82%	#DIV/0!	0%	#DIV/0!	8	0	8
4	05-01-2021	39	5	0	1	89%	17%	100%	29%	1	6	-5
5	06-01-2021	44	1	0	0	98%	0%	#DIV/0!	#DIV/0!	0	1	-1
6	07-01-2021	44	0	1	0	98%	#DIV/0!	0%	#DIV/0!	1	0	1
7	08-01-2021	44	0	1	0	98%	#DIV/0!	0%	#DIV/0!	1	0	1
8	09-01-2021	41	0	3	1	93%	100%	25%	40%	4	1	3
9	10-01-2021	40	4	0	1	91%	20%	100%	33%	1	5	-4
10	11-01-2021	43	0	0	2	100%	100%	100%	100%	2	2	0
11	12-01-2021	43	1	0	1	98%	50%	100%	67%	1	2	-1
12	13-01-2021	37	1	7	0	82%	0%	0%	#DIV/0!	7	1	6
13	14-01-2021	43	1	0	1	98%	50%	100%	67%	1	2	-1
14	15-01-2021	43	1	0	1	98%	50%	100%	67%	1	2	-1
15	16-01-2021	44	1	0	0	98%	0%	#DIV/0!	#DIV/0!	0	1	-1
16	17-01-2021	44	0	1	0	98%	#DIV/0!	0%	#DIV/0!	1	0	1
17	18-01-2021	39	1	4	1	89%	50%	20%	29%	5	2	3
18	19-01-2021	43	1	0	1	98%	50%	100%	67%	1	2	-1
19	20-01-2021	43	2	0	0	96%	0%	#DIV/0!	#DIV/0!	0	2	-2
20	21-01-2021	43	1	1	0	96%	0%	0%	#DIV/0!	1	1	0
21	22-01-2021	43	1	1	0	96%	0%	0%	#DIV/0!	1	1	0
22	23-01-2021	45	0	0	0	100%	#DIV/0!	#DIV/0!	#DIV/0!	0	0	0
23	24-01-2021	44	0	1	0	98%	#DIV/0!	0%	#DIV/0!	1	0	1
24	25-01-2021	44	0	1	0	98%	#DIV/0!	0%	#DIV/0!	1	0	1
25	26-01-2021	44	0	0	1	100%	100%	100%	100%	1	1	0
26	27-01-2021	43	1	0	1	98%	50%	100%	67%	1	2	-1
27	28-01-2021	44	0	0	1	100%	100%	100%	100%	1	1	0
28	29-01-2021	44	0	0	1	100%	100%	100%	100%	1	1	0
29	30-01-2021	43	1	0	1	98%	50%	100%	67%	1	2	-1
	MONTHLY	1277	23	29	21	96%	48%	42%	45%	50	44	6

Method 12: AdaBoost, multi-classification, resampling of 24 hour with 60 days lookback

Steps	**Description**
Data Engineering	Merge all the files for the 24 hours * 365 data (5-min frequency). Use all the raw features from device_data and sun_data. Keep Categorical features from fault_data. ● Fill null_values and missing values. ● Remove duplicates. ● Remove outliers.
Trips Selection Logic	Consider the data from 7am to 5pm (5-min frequency). Use following fault selection logic to create ‘Target’ = ‘Trip’: data['alarm_class'] = np.where(((data['Time'] >= "07:00") & (data['Time'] <= "17:00") & (data['active_power']=0) & (data['irradiance']>50) & (data['alarm_name'].notnull())) , 'yes', 'no') Use remaining ‘alarm_name’ for ‘Target’ = ‘Warning’ Use remaining rows for ‘Target’ = ‘no-fault’
X_train and Y_train	Select the data only in between 7am to 5pm. 60-days of lookback (labels) is considered in X_train.
System	Inverter-wise prediction for 45 inverters
Model	AdaBoostClassifier with SMOTETomek
Results	Results are in 3*3 matrix. Results are not comparable with results of binary-classification methods, therefore, this model is not used further in the discussion.

Method 13: LightGBM, multi-classification, resampling of 24 hour with 60 days lookback

Steps	**Description**
Data Engineering	Merge all the files for the 24 hours * 365 data (5-min frequency). Use all the raw features from device_data and sun_data. Keep Categorical features from fault_data. ● Fill null_values and missing values. ● Remove duplicates. ● Remove outliers.
Trips Selection Logic	Consider the data from 7am to 5pm (5-min frequency). Use following fault selection logic to create ‘Target’ = ‘Trip’: data['alarm_class'] = np.where(((data['Time'] >= "07:00") & (data['Time'] <= "17:00") & (data['active_power']=0) & (data['irradiance']>50) & (data['alarm_name'].notnull())) , 'yes', 'no') Use remaining ‘alarm_name’ for ‘Target’ = ‘Warning’ Use remaining rows for ‘Target’ = ‘no-fault’
X_train and Y_train	Select the data only in between 7am to 5pm. 60-days of lookback (labels) is considered in X_train.
System	Inverter-wise prediction for 45 inverters
Model	LightGBM Classifier with SMOTETomek
Results	Results are in 3*3 matrix. Results are not comparable with results of binary-classification methods, therefore, this model is not used further in the discussion.

Method 14: XGBoost, multi-classification, resampling of 24 hour with 60 days lookback

Steps	**Description**
Data Engineering	Merge all the files for the 24 hours * 365 data (5-min frequency). Use all the raw features from device_data and sun_data. Keep Categorical features from fault_data. ● Fill null_values and missing values. ● Remove duplicates. ● Remove outliers.
Trips Selection Logic	Consider the data from 7am to 5pm (5-min frequency). Use following fault selection logic to create 'Target' = 'Trip': data['alarm_class'] = np.where(((data['Time'] >= "07:00") & (data['Time'] <= "17:00") & (data['active_power']=0) & (data['irradiance']>50) & (data['alarm_name'].notnull())) , 'yes', 'no') Use remaining 'alarm_name' for 'Target' = 'Warning' Use remaining rows for 'Target' = 'no-fault'
X_train and Y_train	Select the data only in between 7am to 5pm. 60-days of lookback (labels) is considered in X_train.
System	Inverter-wise prediction for 45 inverters
Model	XGBoost Classifier with SMOTETomek
Results	Results are in 3*3 matrix. Results are not comparable with results of binary-classification methods, therefore, this model is not used further in the discussion.

Method 15: Selected-inverter, AdaBoost Classifier, multi-classification, resampling of 24 hours with 60 days lookback

Steps	**Description**
Data Engineering	Merge all the files for the 24 hours * 365 data (5-min frequency). Use all the raw features from device_data and sun_data. Keep Categorical features from fault_data. ● Fill null_values and missing values. ● Remove duplicates. ● Remove outliers.
Trips Selection Logic	Consider the data from 7am to 5pm (5-min frequency). Use following fault selection logic to create 'Target' = 'Trip': data['alarm_class'] = np.where(((data['Time'] >= "07:00") & (data['Time'] <= "17:00") & (data['active_power']=0) & (data['irradiance']>50) & (data['alarm_name'].notnull())) , 'yes', 'no') Use remaining rows for 'Target' = 'no-fault'
X_train and Y_train	Select the data only in between 7am to 5pm. Select the data for the 35 inverters with 'trips' in 2020. 60-days of lookback (labels) is considered in X_train.
System	Inverter-wise prediction for **35** inverters which had 'trips' in the training data.
Model	AdaBoostClassifier with SMOTETomek
Results	Monthly metrics (wrt 35 inverters): ● Accuracy: 91% ● Precision: 28% ● Recall: 54% ● F-1 Score: 36%

For all the 35 inverters in the GMR plant in the month of January 2021, total number of TPs are 27, total FNs are 23, FPs are 71, and TNs 929. It means that the model has predicted about 98 'Trips' for actual 'Trips' of 50 out of which total of 27 'Trips' were True Positives,
a total of 23 'Trips' were False Negatives, and a total of 71 'Trips' were False Positives.

For 12 days in the month, the model has recorded a daily F-1 score above 50% with daily Accuracy above 89%. Additionally, For 9 days in the month, the model has recorded a daily F-1 score in between 14% - 40% with daily Accuracy inthe range 69% - 91%.

For 18 days in the month, the model has recorded a daily Recall score above 50% with daily Precision in the range of 8% to 100%. Additionally, For 3 days in the month, the model has recorded a daily Recall score in between 20% to 35% with daily Precision in the range of 20% to 100%.

ID	date	tn	fp	fn	tp	Accuracy	Precision	Recall	F-1Score	Actual_model_trips	Predicted_model_trips	Diff
0	01-01-2021	34	0	1	0	97%	#DIV/0!	0%	#DIV/0!	1	0	1
1	02-01-2021	32	0	2	1	94%	100%	33%	50%	3	1	2
2	03-01-2021	31	2	0	2	94%	50%	100%	67%	2	4	-2
3	04-01-2021	26	1	3	5	89%	83%	63%	71%	8	6	2
4	05-01-2021	22	12	0	1	66%	8%	100%	14%	1	13	-12
5	06-01-2021	34	1	0	0	97%	0%	#DIV/0!	#DIV/0!	0	1	-1
6	07-01-2021	32	2	1	0	91%	0%	0%	#DIV/0!	1	2	-1
7	08-01-2021	34	0	1	0	97%	#DIV/0!	0%	#DIV/0!	1	0	1
8	09-01-2021	31	0	3	1	91%	100%	25%	40%	4	1	3
9	10-01-2021	27	7	0	1	80%	13%	100%	22%	1	8	-7
10	11-01-2021	31	2	0	2	94%	50%	100%	67%	2	4	-2
11	12-01-2021	33	1	0	1	97%	50%	100%	67%	1	2	-1
12	13-01-2021	27	1	7	0	77%	0%	0%	#DIV/0!	7	1	6
13	14-01-2021	29	5	0	1	86%	17%	100%	29%	1	6	-5
14	15-01-2021	34	0	0	1	100%	100%	100%	100%	1	1	0
15	16-01-2021	32	3	0	0	91%	0%	#DIV/0!	#DIV/0!	0	3	-3
16	17-01-2021	32	2	0	1	94%	33%	100%	50%	1	3	-2
17	18-01-2021	26	4	4	1	77%	20%	20%	20%	5	5	0
18	19-01-2021	27	7	0	1	80%	13%	100%	22%	1	8	-7
19	20-01-2021	35	0	0	0	100%	#DIV/0!	#DIV/0!	#DIV/0!	0	0	0
20	21-01-2021	29	5	0	1	86%	17%	100%	29%	1	6	-5
21	22-01-2021	32	2	1	0	91%	0%	0%	#DIV/0!	1	2	-1
22	23-01-2021	32	3	0	0	91%	0%	#DIV/0!	#DIV/0!	0	3	-3
23	24-01-2021	30	4	0	1	89%	20%	100%	33%	1	5	-4
24	25-01-2021	33	1	0	1	97%	50%	100%	67%	1	2	-1
25	26-01-2021	31	3	0	1	91%	25%	100%	40%	1	4	-3
26	27-01-2021	32	2	0	1	94%	33%	100%	50%	1	3	-2
27	28-01-2021	34	0	0	1	100%	100%	100%	100%	1	1	0
28	29-01-2021	34	0	0	1	100%	100%	100%	100%	1	1	0
29	30-01-2021	33	1	0	1	97%	50%	100%	67%	1	2	-1
	MONTHLY	929	71	23	27	91%	28%	54%	36%	50	98	-48

Method 16: MLP Classifier, binary-classification, resampling of 24 hours with 60 days lookback

Steps	**Description**
Data Engineering	Merge all the files for the 24 hours * 365 data (5-min frequency). Use all the raw features from device_data and sun_data. Keep Categorical features from fault_data. ● Fill null_values and missing values. ● Remove duplicates. ● Remove outliers.
Trips Selection Logic	Consider the data from 7am to 5pm (5-min frequency). Use following fault selection logic to create 'Target' = 'Trip': data['alarm_class'] = np.where(((data['Time'] >= "07:00") & (data['Time'] <= "17:00") & (data['active_power']=0) & (data['irradiance']>50) & (data['alarm_name'].notnull())) , 'yes', 'no') Use remaining rows for 'Target' = 'no-fault'
X_train and Y_train	Select the data only in between 7am to 5pm. Select the data for the 35 inverters with 'trips' in 2020. 60-days of lookback (labels) is considered in X_train.
System	Inverter-wise prediction for 45 inverters
Model	MLP classifier with Smote
Results	Monthly metrics: ● Accuracy: 66% ● Precision: 8% ● Recall: 82% ● F-1 Score: 15%

For all the 45 inverters in the GMR plant in the month of January 2021, total number of TPs are 41, total FNs are 9, FPs are 448, and TNs 852. It means that the model has predicted about 489 'Trips' for actual 'Trips' of 50 out of which total of 41 'Trips' were True Positives,
a total of 9 'Trips' were False Negatives, and a total of 448 'Trips' were False Positives.

For 7 days in the month, the model has recorded a daily F-1 score above 50% with daily Accuracy above 70%. Additionally, For 17 days in the month, the model has recorded a daily F-1 score in between 4% - 40% with daily Accuracy inthe range 2% - 93%.

For 23 days in the month, the model has recorded a daily Recall score above 50% with daily Precision in the range of 2% to 100%. Additionally, For 1 day in the month, the model has recorded a daily Recall score of 25% with daily Precision of 100%.

Id	date	tn	fp	fn	tp	Accuracy	Precision	Recall	F-1Score	Actual_model_trips	Predicted_model_trips	Diff
0	01-01-2021	43	1	1	0	96%	0%	0%	#DIV/0!	1	1	0
1	02-01-2021	36	6	0	3	87%	33%	100%	50%	3	9	-6
2	03-01-2021	4	39	0	2	13%	5%	100%	9%	2	41	-39
3	04-01-2021	34	3	3	5	87%	63%	63%	63%	8	8	0
4	05-01-2021	9	35	0	1	22%	3%	100%	5%	1	36	-35
5	06-01-2021	42	3	0	0	93%	0%	#DIV/0!	#DIV/0!	0	3	-3
6	07-01-2021	37	7	0	1	84%	13%	100%	22%	1	8	-7
7	08-01-2021	43	1	0	1	98%	50%	100%	67%	1	2	-1
8	09-01-2021	41	0	3	1	93%	100%	25%	40%	4	1	3
9	10-01-2021	28	16	0	1	64%	6%	100%	11%	1	17	-16
10	11-01-2021	31	12	0	2	73%	14%	100%	25%	2	14	-12
11	12-01-2021	41	3	0	1	93%	25%	100%	40%	1	4	-3
12	13-01-2021	25	13	0	7	71%	35%	100%	52%	7	20	-13
13	14-01-2021	2	42	0	1	7%	2%	100%	5%	1	43	-42
14	15-01-2021	36	8	0	1	82%	11%	100%	20%	1	9	-8
15	16-01-2021	16	29	0	0	36%	0%	#DIV/0!	#DIV/0!	0	29	-29
16	17-01-2021	41	3	0	1	93%	25%	100%	40%	1	4	-3
17	18-01-2021	28	12	1	4	71%	25%	80%	38%	5	16	-11
18	19-01-2021	0	44	0	1	2%	2%	100%	4%	1	45	-44
19	20-01-2021	32	13	0	0	71%	0%	#DIV/0!	#DIV/0!	0	13	-13
20	21-01-2021	1	43	0	1	4%	2%	100%	4%	1	44	-43
21	22-01-2021	34	10	1	0	76%	0%	0%	#DIV/0!	1	10	-9
22	23-01-2021	42	3	0	0	93%	0%	#DIV/0!	#DIV/0!	0	3	-3
23	24-01-2021	21	23	0	1	49%	4%	100%	8%	1	24	-23
24	25-01-2021	10	34	0	1	24%	3%	100%	6%	1	35	-34
25	26-01-2021	43	1	0	1	98%	50%	100%	67%	1	2	-1
26	27-01-2021	44	0	0	1	100%	100%	100%	100%	1	1	0
27	28-01-2021	44	0	0	1	100%	100%	100%	100%	1	1	0
28	29-01-2021	20	24	0	1	47%	4%	100%	8%	1	25	-24
29	30-01-2021	24	20	0	1	56%	5%	100%	9%	1	21	-20
	MONTHLY	852	448	9	41	66%	8%	82%	15%	50	489	-439

Method 17: SGD Classifier, binary-classification, resampling of 24 hours with 60 days lookback

Steps	**Description**
Data Engineering	Merge all the files for the 24 hours * 365 data (5-min frequency). Use all the raw features from device_data and sun_data. Keep Categorical features from fault_data. ● Fill null_values and missing values. ● Remove duplicates. ● Remove outliers.
Trips Selection Logic	Consider the data from 7am to 5pm (5-min frequency). Use following fault selection logic to create 'Target' = 'Trip': data['alarm_class'] = np.where(((data['Time'] >= "07:00") & (data['Time'] <= "17:00") & (data['active_power']=0) & (data['irradiance']>50) & (data['alarm_name'].notnull())) , 'yes', 'no') Use remaining rows for 'Target' = 'no-fault'
X_train and Y_train	Select the data only in between 7am to 5pm. Select the data for the 35 inverters with 'trips' in 2020. 60-days of lookback (labels) is considered in X_train.
System	Inverter-wise prediction for 45 inverters
Model	SGD Classifier with Smote
Results	Monthly metrics: ● Accuracy: 57% ● Precision: 4% ● Recall: 42% ● F-1 Score: 7%

For all the 45 inverters in the GMR plant in the month of January 2021, total number of TPs are 21, total FNs are 29, FPs are 552, and TNs 748. It means that the model has predicted about 573 'Trips' for actual 'Trips' of 50 out of which total of 21 'Trips' were True Positives, a total of 29 'Trips' were False Negatives, and a total of 552 'Trips' were False Positives.

Id	date	tn	fp	fn	tp	Accuracy	Precision	Recall	F-1Score	Actual_model_trips	Predicted_model_trips	Diff
0	2021-01-01	0	44	0	1	2%	2%	100%	4%	1	45	-44
1	2021-01-02	38	4	3	0	84%	0%	0%	#DIV/0!	3	4	-1
2	2021-01-03	3	40	0	2	11%	5%	100%	9%	2	42	-40
3	2021-01-04	0	37	0	8	18%	18%	100%	30%	8	45	-37
4	2021-01-05	40	4	0	1	91%	20%	100%	33%	1	5	-4
5	2021-01-06	35	10	0	0	78%	0%	#DIV/0!	#DIV/0!	0	10	-10
6	2021-01-07	3	41	0	1	9%	2%	100%	5%	1	42	-41
7	2021-01-08	42	2	1	0	93%	0%	0%	#DIV/0!	1	2	-1
8	2021-01-09	38	3	4	0	84%	0%	0%	#DIV/0!	4	3	1
9	2021-01-10	17	27	1	0	38%	0%	0%	#DIV/0!	1	27	-26
10	2021-01-11	43	0	2	0	96%	#DIV/0!	0%	#DIV/0!	2	0	2
11	2021-01-12	27	17	1	0	60%	0%	0%	#DIV/0!	1	17	-16
12	2021-01-13	38	0	7	0	84%	#DIV/0!	0%	#DIV/0!	7	0	7
13	2021-01-14	0	44	0	1	2%	2%	100%	4%	1	45	-44
14	2021-01-15	42	2	1	0	93%	0%	0%	#DIV/0!	1	2	-1
15	2021-01-16	1	44	0	0	2%	0%	#DIV/0!	#DIV/0!	0	44	-44
16	2021-01-17	0	44	0	1	2%	2%	100%	4%	1	45	-44
17	2021-01-18	25	15	4	1	58%	6%	20%	10%	5	16	-11
18	2021-01-19	44	0	1	0	98%	#DIV/0!	0%	#DIV/0!	1	0	1
19	2021-01-20	43	2	0	0	96%	0%	#DIV/0!	#DIV/0!	0	2	-2
20	2021-01-21	44	0	1	0	98%	#DIV/0!	0%	#DIV/0!	1	0	1
21	2021-01-22	8	36	0	1	20%	3%	100%	5%	1	37	-36
22	2021-01-23	45	0	0	0	100%	#DIV/0!	#DIV/0!	#DIV/0!	0	0	0
23	2021-01-24	0	44	0	1	2%	2%	100%	4%	1	45	-44
24	2021-01-25	0	44	0	1	2%	2%	100%	4%	1	45	-44
25	2021-01-26	29	15	0	1	67%	6%	100%	12%	1	16	-15
26	2021-01-27	44	0	1	0	98%	#DIV/0!	0%	#DIV/0!	1	0	1
27	2021-01-28	44	0	1	0	98%	#DIV/0!	0%	#DIV/0!	1	0	1
28	2021-01-29	11	33	0	1	27%	3%	100%	6%	1	34	-33
29	2021-01-30	44	0	1	0	98%	#DIV/0!	0%	#DIV/0!	1	0	1
	MONTHLY	748	552	29	21	57%	4%	42%	7%	50	573	-523

Method 18: Gradient Boosting Classifier, binary-classification, resampling of 24 hours with 60 days lookback

Steps	**Description**
Data Engineering	Merge all the files for the 24 hours * 365 data (5-min frequency). Use all the raw features from device_data and sun_data. Keep Categorical features from fault_data. ● Fill null_values and missing values. ● Remove duplicates. ● Remove outliers.
Trips Selection Logic	Consider the data from 7am to 5pm (5-min frequency). Use following fault selection logic to create 'Target' = 'Trip': data['alarm_class'] = np.where(((data['Time'] >= "07:00") & (data['Time'] <= "17:00") & (data['active_power']=0) & (data['irradiance']>50) & (data['alarm_name'].notnull())) , 'yes', 'no') Use remaining rows for 'Target' = 'no-fault'
X_train and Y_train	Select the data only in between 7am to 5pm. Select the data for the 35 inverters with 'trips' in 2020. 60-days of lookback (labels) is considered in X_train.
System	Inverter-wise prediction for 45 inverters
Model	Gradient Boosting Classifier with Smote
Results	Monthly metrics: ● Accuracy: 95% ● Precision: 33% ● Recall: 46% ● F-1 Score: 39%

For all the 45 inverters in the GMR plant in the month of January 2021, total number of TPs are 41, total FNs are 9, FPs are 448, and TNs 852. It means that the model has predicted about 489 'Trips' for actual 'Trips' of 50 out of which total of 41 'Trips' were True Positives,
a total of 9 'Trips' were False Negatives, and a total of 448 'Trips' were False Positives.

Id	date	tn	fp	fn	tp	Accuracy	Precision	Recall	F-1Score	Actual_model_trips	Predicted_model_trips	Diff
0	01-01-2021	44	0	1	0	98%	#DIV/0!	0%	#DIV/0!	1	0	1
1	02-01-2021	42	0	2	1	96%	100%	33%	50%	3	1	2
2	03-01-2021	42	1	1	1	96%	50%	50%	50%	2	2	0
3	04-01-2021	36	1	5	3	87%	75%	38%	50%	8	4	4
4	05-01-2021	35	9	0	1	80%	10%	100%	18%	1	10	-9
5	06-01-2021	43	2	0	0	96%	0%	#DIV/0!	#DIV/0!	0	2	-2
6	07-01-2021	44	0	1	0	98%	#DIV/0!	0%	#DIV/0!	1	0	1
7	08-01-2021	44	0	0	1	100%	100%	100%	100%	1	1	0
8	09-01-2021	41	0	3	1	93%	100%	25%	40%	4	1	3
9	10-01-2021	37	7	0	1	84%	13%	100%	22%	1	8	-7
10	11-01-2021	42	1	1	1	96%	50%	50%	50%	2	2	0
11	12-01-2021	43	1	0	1	98%	50%	100%	67%	1	2	-1
12	13-01-2021	37	1	7	0	82%	0%	0%	#DIV/0!	7	1	6
13	14-01-2021	42	2	0	1	96%	33%	100%	50%	1	3	-2
14	15-01-2021	43	1	0	1	98%	50%	100%	67%	1	2	-1
15	16-01-2021	42	3	0	0	93%	0%	#DIV/0!	#DIV/0!	0	3	-3
16	17-01-2021	44	0	0	1	100%	100%	100%	100%	1	1	0
17	18-01-2021	39	1	4	1	89%	50%	20%	29%	5	2	3
18	19-01-2021	40	4	0	1	91%	20%	100%	33%	1	5	-4
19	20-01-2021	43	2	0	0	96%	0%	#DIV/0!	#DIV/0!	0	2	-2
20	21-01-2021	43	1	1	0	96%	0%	0%	#DIV/0!	1	1	0
21	22-01-2021	41	3	1	0	91%	0%	0%	#DIV/0!	1	3	-2
22	23-01-2021	42	3	0	0	93%	0%	#DIV/0!	#DIV/0!	0	3	-3
23	24-01-2021	43	1	0	1	98%	50%	100%	67%	1	2	-1
24	25-01-2021	44	0	0	1	100%	100%	100%	100%	1	1	0
25	26-01-2021	43	1	0	1	98%	50%	100%	67%	1	2	-1
26	27-01-2021	43	1	0	1	98%	50%	100%	67%	1	2	-1
27	28-01-2021	44	0	0	1	100%	100%	100%	100%	1	1	0
28	29-01-2021	44	0	0	1	100%	100%	100%	100%	1	1	0
29	30-01-2021	44	0	0	1	100%	100%	100%	100%	1	1	0
	MONTHLY	1254	46	27	23	95%	33%	46%	39%	50	69	-19

Discussion

Comparison

Model	TN	FP	FN	TP	Accuracy	Precision	Recall	F-1 Score
Model 7	1285	15	40	10	96%	40%	20%	27%
Model 8	1256	44	25	25	95%	36%	50%	42%
Model 9	1209	91	24	26	91%	22%	52%	31%
Model 10	1287	13	35	15	96%	54%	30%	38%
Model 11	1277	23	29	21	96%	48%	42%	45%
Model 15 (sp)	929	71	23	27	91%	28%	54%	36%
Model 16	852	448	9	41	66%	8%	82%	15%
Model 17	748	552	29	21	57%	4%	42%	7%
Model 18	1254	46	27	23	95%	33%	46%	39%

Above mentioned nine models are prepared based-on the binary-classification models. We have not compared Model 1, Model 2, Model 3, Model 4 as they didn't produce adequate results. Furthermore, Model 12, Model 13, and Model 14 are multi-classification and therefore, not included here.

Except Model 15, all the other models include all the 45 inverters in the training set. Only model 15 includes 35 inverters which had non-zero 'trips' in the training set.

- Model 7: 80% of the actual 'trips' are not being detected.
- Model 8: 50% of the actual 'trips' are not being detected.
- Model 9: only 22% of the work-order generated are for the actual 'trips'. Remaining 78% were false-positives. About ~48% of the actual 'trips' are not being detected.
- Model 10: only ~54% of the work-order generated are for the actual 'trips'. Remaining ~46% were false-positives. Approx 70% of the actual 'trips' are not being detected.
- Model 11: only 48% of the work-order generated are for the actual 'trips'. Remaining are were false-positives. Approx 58% of the actual 'trips' are not being detected.
- Model 15: only 30% of the work-order generated are for the actual 'trips'. Remaining are false-positives. Approx 46% of the actual 'trips' are not being detected.
- Model 16: only 8% of the work-order generated are for the actual 'trips'. Remaining are false-positives.
- Model 17: only ~3.7% of the work-order generated are for the actual 'trips'. Remaining are false-positives.
- Model 18: only ~34% of the work-order generated are for the actual 'trips'.

Selection

Model	TN	FP	FN	TP	Accuracy	Precision	Recall	F-1 Score
Model 9	1209	91	24	26	91%	22%	52%	31%
Model 10	1287	13	35	15	96%	**54%**	30%	38%
Model 11	1277	23	29	21	96%	**48%**	42%	45%
Model 15 (sp)	929	71	23	27	91%	28%	54%	36%

Above mentioned four models (Model 9, Model 10, Model 11, Model 15) are selected based on the business requirements.

Model 9, Model 10 and Model 11 include all the 45 inverters in the training set. Model 15 includes only 35 inverters which had non-zero ‘trips’ in the training set.

- Model 9: predicted 26 ‘trips’ correctly (missed 24 ‘trips’) for total of 117 forecasted ‘trips’
- Model 10: predicted 15 ‘trips’ correctly (missed 35 ‘trips’) for total of 26 forecasted ‘trips’
- Model 11: predicted 21 ‘trips’ correctly (missed 29 ‘trips’) for total of 44 forecasted ‘trips’
- Model 15: predicted 27 ‘trips’ correctly (missed 23 ‘trips’) for total of 98 forecasted ‘trips’

Results of Model 9

During data validation over 30-days of data with Method 9, following results were observed. Only on Feb 04, Feb 05, Feb 13, Feb 18 results were not desirable, however, most days results were acceptable.

Effect of Sampling Techniques

To resolve the class imbalance issues in the classification problem, we have used following sampling techniques:

1. Oversampling
    a. SMOTE
        i. Synthetic Minority Over-sampling Technique (SMOTE) is a technique that generates new observations by interpolating between observations in the original dataset.
    b. ADYSYN
        i. Adaptive Synthetic (ADASYN) sampling is the same as SMOTE, however, the number of samples generated for a given xi is proportional to the number of nearby samples which do not belong to the same class as xi. Thus, ADASYN tends to focus solely on outliers when generating new synthetic training examples.
2. Undersampling
    a. Near Miss

        i. NearMiss is an under-sampling technique. Instead of resampling the Minority class, using a distance, this will make the majority class equal to the minority class.
    b. Tomek Link
        i. Tomek links are pairs of very close instances but of opposite classes. Removing the instances of the majority class of each pair increases the space between the two classes, facilitating the classification process. Tomek's link exists if the two samples are the nearest neighbors of each other.
    c. Cluster Centroid
        i. The demarcation of instances as important and unimportant is done by using the concept of Clustering on Feature-Space Geometry.
3. Hybrid
    a. SMOTEENN
        i. Over-sampling using SMOTE and cleaning using ENN.
        ii. Combine over- and under-sampling using SMOTE and Edited Nearest Neighbours.
    b. SMOTETomek
        i. Over-sampling using SMOTE and cleaning using Tomek links.
        ii. Combine over- and under-sampling using SMOTE and Tomek links.

For the below mentioned analysis with sampling (over- or under- sampling), parameters in the models are kept constant to: base_estimator=None, n_estimators=50, learning_rate=1, algorithm='SAMME.R', random_state=42

Sampling Techniques	**TPs**	**FNs**	**FPs**	**TNs**
SMOTE	27	23	73	1227
ADYSYN	27	23	84	1216
Near Miss	43	7	506	794
Tomek Link	20	30	23	1277
Cluster centroid	49	1	1179	121
SMOTEENN	23	27	88	1212
SMOTETomek	27	23	86	1214

Effect of the model's parameter - Learning rate

Learning rate is a parameter, denoted by α(alpha), is used to tune how accurately a model converges on a result. For example, if the gradient magnitude is 1.5 and the learning rate is 0.01, then the algorithm will pick the next point 0.015 away from the previous point.

For the below mentioned analysis on ‘learning rate’, parameters in the models are kept constant to: base_estimator=None, n_estimators=100, algorithm='SAMME', random_state=42.

Learning Rate (With SMOTE)	**TPs**	**FNs**	**FPs**	**TNs**
1.5	24	26	51	1249
1	31	19	89	1211
0.1	42	8	262	1038
0.01	50	0	367	933
0.001	50	0	367	933

Effect of the model’s parameter - ‘Algorithm’

SAMME algorithm (Stagewise Additive Modeling) uses a discrete AdaBoost algorithm. SAMME.R is a real boosting algorithm where ’base_estimator’ must support the calculation of class probabilities. SAMME.R uses the probability estimates to update the additive model, while SAMME uses the classifications only.

- The SAMME.R algorithm typically converges faster than SAMME, achieving a lower test error with fewer boosting iterations.
- Discrete SAMME AdaBoost adapts based on errors in predicted class labels whereas real SAMME.R uses the predicted class probabilities.

Algorithm (w/ SMOTE)	**TPs**	**FNs**	**FPs**	**TNs**
SAMME (base_estimator=None, n_estimators=100, learning_rate=1, algorithm='SAMME', random_state=42)	31	19	89	1211
SAMME.R (base_estimator=None, n_estimators=100, learning_rate=1, algorithm='SAMME.R', random_state=42)	20	30	55	1245
SAMME.R (base_estimator=DecisionTreeClassifier(max_depth= 1), n_estimators=100, learning_rate=1, algorithm='SAMME.R', random_state=42)	18	32	18	1282

Above mentioned analysis indicates that use of ‘SAMME’ is better than ‘SAMME.R’

Results of Model 10

Model 10 includes all the 45 inverters in the training set.

- predicted 15 ‘trips’ correctly (missed 35 ‘trips’) for total of 26 forecasted ‘trips’
- only ~54% of the work-order generated are for the actual ‘trips’. Remaining ~46% were false-positives. Approx 70% of the actual ‘trips’ are not being detected.

Results of Model 11

Model 11 includes all the 45 inverters in the training set.

- predicted 21 ‘trips’ correctly (missed 29 ‘trips’) for total of 44 forecasted ‘trips’
- only **48% of the work-order generated** are for the actual ‘trips’. Remaining are were false-positives. Approx 58% of the actual ‘trips’ are not being detected.

Results of Model 15

Model 15 includes only 35 inverters which had non-zero ‘trips’ in the training set.

- predicted 27 ‘trips’ correctly (missed 23 ‘trips’) for total of 98 forecasted ‘trips’
- only 30% of the work-order generated are for the actual ‘trips’. Remaining are false-positives. Approx 46% of the actual ‘trips’ are not being detected.

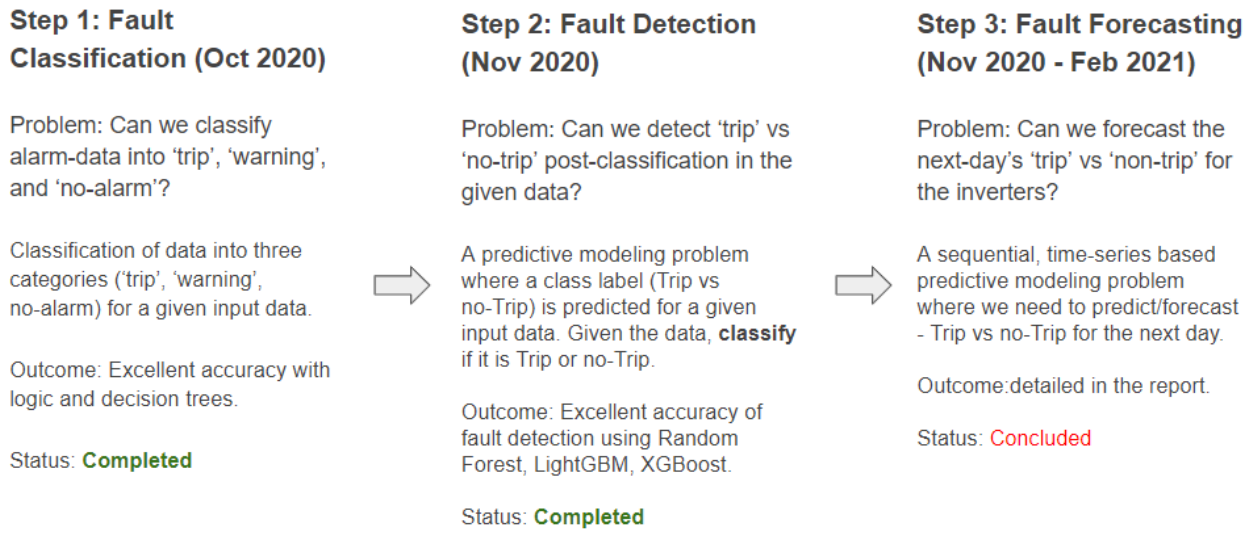
Section 3 - Summary

Project Steps

The project has been divided into three steps as depicted in the below mentioned figure:

- fault classification,
- fault detection, and
- fault forecasting.

Section 1 of this report covers both fault detection and fault classification and Section 2 discusses the fault forecasting in detail.



Time Spent

For the project - fault forecasting total time spent has been ~5.5 months among these four stages:

- Data stage: data analysis and engineering stage,
- Model stage: ML modeling stage,
- Validation stage: validation of the model, and
- Iterative stage: iteration on the data engineering

1. Data stage	2. Modeling stage	3. Validation stage	4. Iterative stage
Data Access for training (GMR, Mansa-1, Porbander)	Select one base-model (LSTM, GRU, XGBoost)	Inverter-wise prediction of trips (hourly/daily)	Increase the training data size
Pre-processing	Train the model	Probabilities of inverters' being faulty (hourly/daily)	Improvements in the data structure & alarms' selection logic
Trip's selection logic	Validate and test the model	Performance metrics	Modifications in model
1 months	2 months	1.5 months	1.5 months

Details on monthly activities

Month	Activities	Description
July - August 2020	• Data Science roadmap • Solar industry domain knowledge • Data Analysis • Moonshot project finalization	Prepared list of the 40 data science projects and prioritized on moonshot projects. Conducted data analysis for different plants with multiple discussions with customer-facing team.
September 2020	• GPT-3 presentation • Inverter OEMs database • Strategic planning for the first moonshot project - Fault Forecasting • Freelancer hiring contract	Created a database for inverter OEMs for alarm codes and alarms messages. GPT-3 presentation Freelancer hiring and strategic planning for the fault prediction/forecast.
October 2020	• Alarm profile and fault selection logic • Fault classification and prediction for GMR Plant	Understanding alarm profile and trips for the trips classification from warnings. Variation on different fault selection logic.

		Python models for fault detection and classification.
November 2020	• Fault classification and prediction for Porbander Plant • Fault Forecasting on GMR with LSTM	Successful completion of fault detection and classification. Post-analysis, started LSTM model for fault forecasting and completed the first version of the model.
December 2020	• Data structure issues • Improvement for solving imbalance problem • Fault forecasting validation • Project - irradiance forecasting with ClimaCell and GMR data • Data science interviews	Multiple times in this month data structure was changed leading to difficulty in timely completion of performance validation. Data Science hiring Additional project for predicting irradiance values for the next day based-on the clima-cell and GMR data.
January 2021	• Model validation on Nov - Dec data • LSTM Model update for improving performance • Solution for data predictability • Intern hiring	Conducted model validation based-on the Nov/Dec data post-fixing the data structure issues. Updated LSTM model and focused on one of the key issues on the data predictability.
February 2021	• Model changes and overhaul for improving TPs • Survival analysis for fault forecasting • Multi-classification for the fault forecasting • Non-LSTM models for fault forecasting	Re-development of the python models for fixing up the data predictability and imbalance issue. Implemented survival analysis and multi-classification, however, it failed in making good-prediction (need more efforts). Successfully solve the problems related to imbalance and data predictability.
March 2021	• Commercialization of the model • Current fault forecasting model	Focus is on commercialization.

	validation ● Project - irradiance forecasting with ClimaCell and GMR data	

Insights and Learnings

1. Data Structure
    a. Changes in data structure lead to significant data engineering and modification in code. If the plant-wise data-structure is different, the model might require working from scratch. Therefore, it is recommended to keep a standard format in the database.

    b. In the last 4 months, we have worked on 3 different types of data-structure where data shape, data volume, and initial data engineering were redone.

    c. Currently, we have been importing data using API and column sequence is unstructured.

2. Data Predictability
    a. Active power becomes zero when there is a trip in the inverter. However, it is not a leading indicator for the occurrence of the ‘Trips’.

    b. Since raw data is not a strong predictor of the ‘Trips’, additional new features - time-based, categorical alarm names, difference of the previous features (delta features), and lagged target variables using lookback are used.

    c. It is observed that 5-min data frequency is having a poor predictability of ‘Target’. Additionally, 30-min, 1-hour resampling is resulting in similar results as **24-hour** (i.e. daily) resampling of the data.

3. Inverter Behavior
    a. Out of 45 inverters at the plant only 35 inverters had ‘Trip’ as per the selected logic in the year 2020.

4. ‘Trip’ Selection Logic
    a. We have considered 3 different types of logic for identifying the ‘Trips’ where inverters will shut off, however logic based-on ‘active power’ = 0 is finally used and selected.

5. Class Imbalance Problem

a. Our data is highly imbalanced in terms of how many actual ‘Trips’ and ‘no-faults’ are observed. Therefore, it leads to the class imbalance problem, where more weightage is given to the majority class.

b. In classification algorithms, the class imbalance problem is solved using over- or under-sampling techniques and/or class weights adjustments.

c. We have found that **SMOTE** (one of the oversampling techniques) performed relatively well.

d. Undersampling or hybrid techniques were not better in comparison to over sampling.

6. Time Duration & Data Frequency
    a. We have found that the best time-duration is 7am to 5pm wrt full 24-hours.

    b. Data frequency is given to be 5-min, however 1-hour to 1-day data frequency is used and finally 1-day frequency has been finalized.

7. Feature Engineering
    a. New features based-on the time, categorical alarm names, statistical features on inverter, weather data are being used and useful for improving data’s predictability.

8. Alarms Distribution

Time-travel to 2020

What would I do differently if I could go back 6-month in time?

#	Category	What will I do differently?
1	Inverter OEMs database	In Sept 2020, we started building Inverter’s OEMs database to be used in our models to make it OEM agnostic, however, that turned out to be fruitless exercise.
2	Data Structure and API access	From Sept to Dec 2020, we have used manual download of data from InfluxDB, however use of the current API is relatively faster and efficient. We should have facilitated API access sooner.
3	Model Selection	LSTM vs Classification In the past (esp during the work of the freelancer), we have kept the model to LSTM. In our current approaches, we have varied to other classification techniques for forecasting. It seems for the resampled data and to solve imbalance data problem approaches based-on the classification stands the best?

Current Gaps

Based-on the table below, one of the key gaps is that current models are not plug-and-play.

#	Defined questions	Results (have we solved the question?)
1	Are the models producing output in the right format?	Yes. They are producing next day's predictions in the desired format.
2	Are the models site (location) independent?	Yes. As long as the data structure and range of the features are the same. Data engineering might be additional for each site.
3	Are the models inverter-OEM's agnostic?	Yes. As long as the data structure. Data engineering might be additional.
4	Are the models error-codes independent?	Yes. The methods/models are only considering the sum or count of error codes irrespective of the actual error codes or alarm names. However, data engineering might be additional.
5	Is the current model plug-and-play?	No. However, it can be made plug-and-play post-data engineering at each site.
6	Is the model producing accuracy, precision, recall, and F-1 score above 85% during **testing and validation**?	Partially, for some days only. Accuracy is mostly above 90% however, F-1 score takes toll.
7	Is the model giving reproducible results of performance metrics?	Yes.

Recommendation

Based-on the results obtained, we have to rephrase the problem. Initially our focus has been to maximize the True Positives (TPs) and minimize the False Negatives (FNs), however, we need to retarget getting better accuracy of work-order generation (i.e. Precision).

Also, we can use ensemble of the models:

Data => Model A + Model B + Model C => Select the final results.

Improvement & Direction

Here are the list of potential improvements:

#	Category	Description
1	Survival analysis	Improve the model 1 - survival analysis technique
2	Multi-classification models	Improve the multi-classification techniques (Model 12, Model 13, and Model 14)
3	Training data size (Volume)	We have used 2020 data in the training for these results. With the increase in the size of the training data performance of the model may also improve.
4	Inverters' OEM	Currently the model has been trained for SMA inverters. For the next steps, other OEMs should be explored and trained to build the portfolio.
5	Location data	Based-on the location of plants we can consider the plants on easten, western, southern, and northern India and develop the model to understand and improve the geo-spatial variations.
6	Time-zone and day-time saving	Currently we have considered 7am to 5pm duration, however, this might not be accurate for a different time-zone or location. We should make a logic for considering sun-rise/sun-set or time-zone related variation.
7	Physical configuration	What changes we need to make into the model for bifacial solar panels? How will the model change, if any?
8	String and grid-side data	How to incorporate string or grid-side data? Can it be useful or improve the model's output?
9	'Plug & Play' model	How to make the model plug-and-play?

Commercialization Plan

In order to commercialize the current model, we have to consider following one of the following goals:

1. **Better accuracy of work-order generated** - equivalent to Precision (recommended)

   OR

2. **Better accuracy of 'trips' forecasted** - equivalent to Recall

   OR

3. Optimum accuracy of work-order generated (Precision) and accuracy of 'trips' forecasted (Recall) - equivalent to F-1 score.

Steps for commercialization:

#	Steps	Description
1	Daily internal validation on GMR	Daily forecast and data analysis post forecasting on GMR data to build confidence internally.
2	Pilot @ GMR client	Trial at the client site.
3.	Model fitting for the other plants	Internal model-fitting for the other locations and OEMs
4	Pilot @ other clients	Trial at the client site.