



Data Challenge 2021

Team 22

Christine Lin

Lei Gao

Hojin Lee

Katherine Jimenez

The Problem:

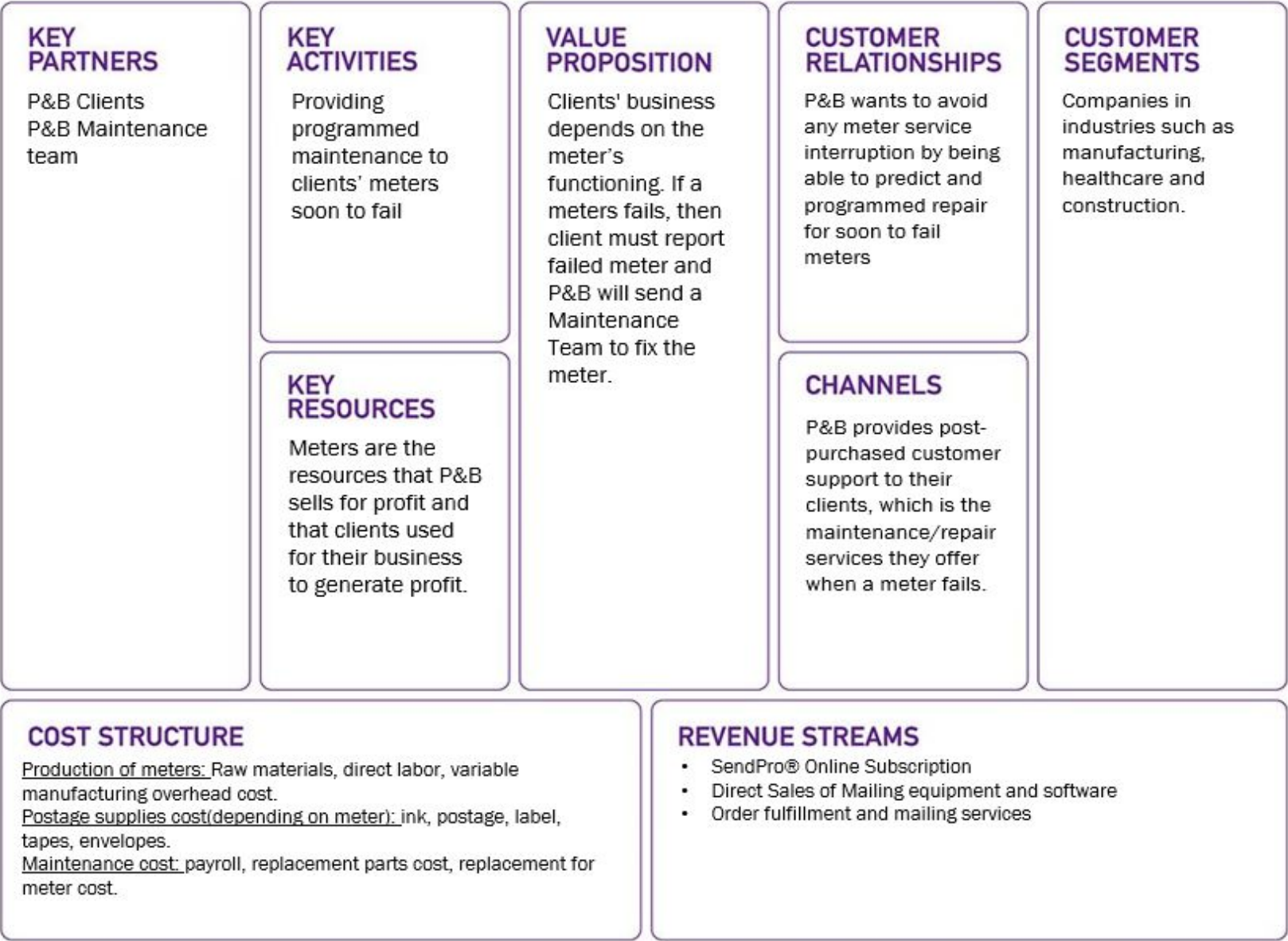
- Pitney Bowes sells meters to their clients which are vital to their business.
- The meters tend to fail from time to time and when they do, they disrupt the business of their clients.
- Once a meter fails, client must contact P&B to report the failed meter, so maintenance services team assist with fixing the meter.

The Goal:

To build a model that can predict which meter will fail within the next 7 days. So, P&B can schedule the maintenance for the meter and prevent meter from suddenly disrupting client's business.



Business Understanding-Lean Canvas



+

•

Problem Analysis

PITNEY & BOWES DATA CHALLENGE 2021

Problem and Goal

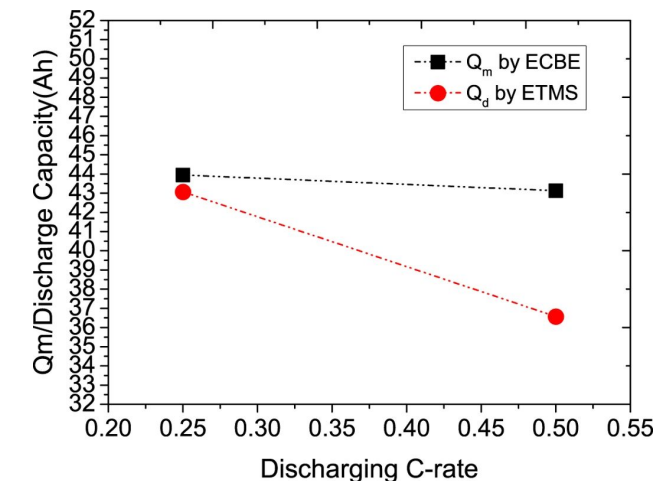
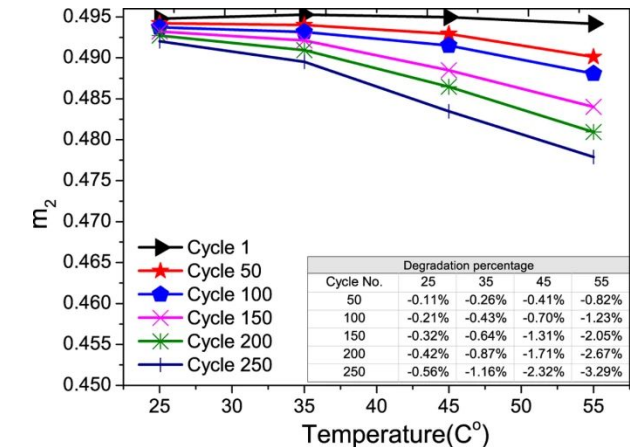
- The key is to determine what contributes to the failure of meters in 7 days.

Initial Confusion on Data Browsing

- We found that **(23%) of meters fail**
Should we change the battery providers?

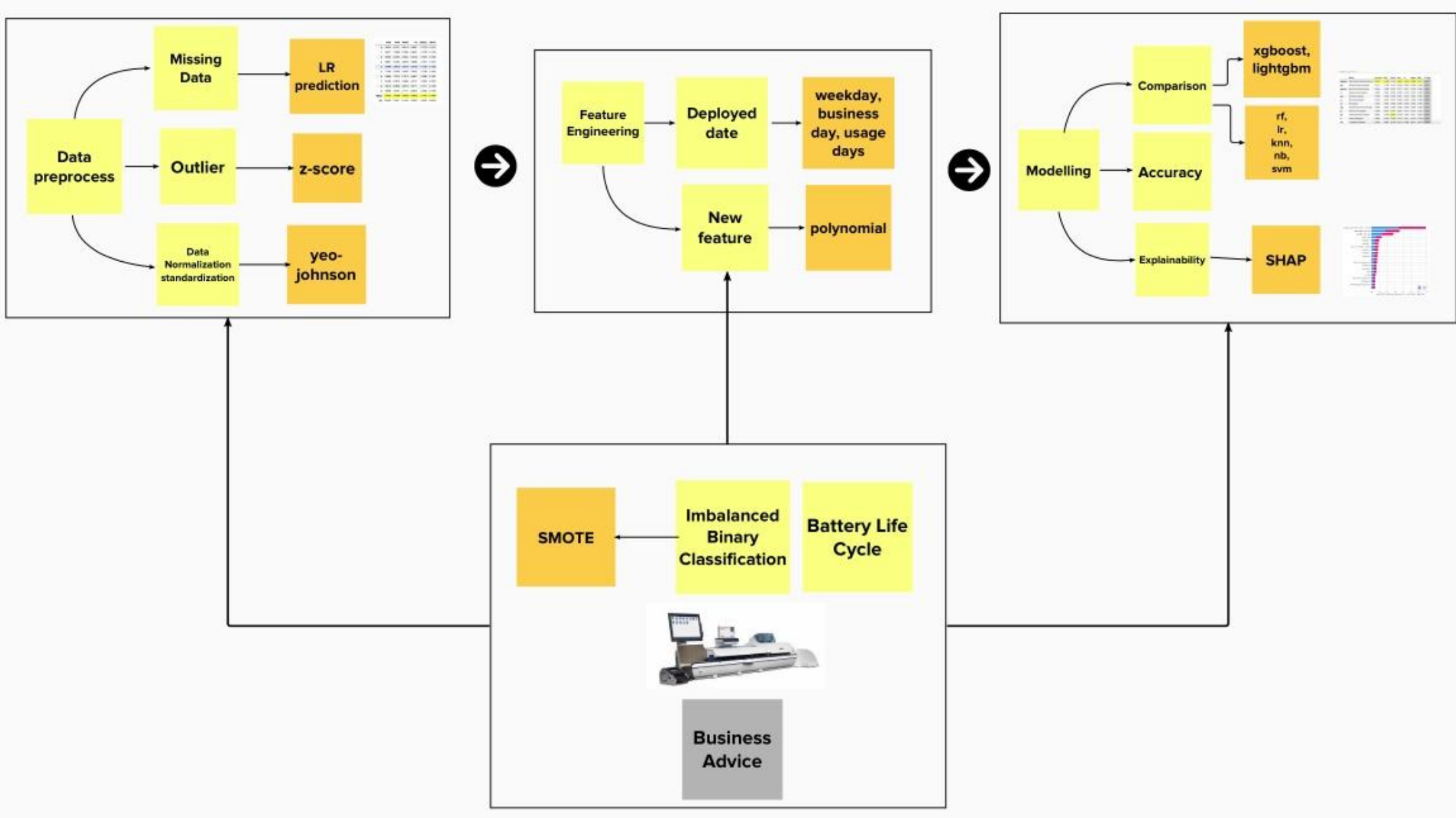
Initial Business Assumption based on Research

- This is a battery lifetime prediction problem. Our team did some research and found that **temperature and voltage** are two major factors to battery life.
- From a business perspective, **deployed date and product segmentation** which can be inferred from “deviceid” or so, may give us some clues on the prediction.
- Apart from **accuracy**, predictive maintenance focus more on **precision or F1 score** to decrease False Positive based on the assumption that visiting cost is low.



Design Thinking

PITNEY & BOWES DATA CHALLENGE 2021



Exploratory Analysis

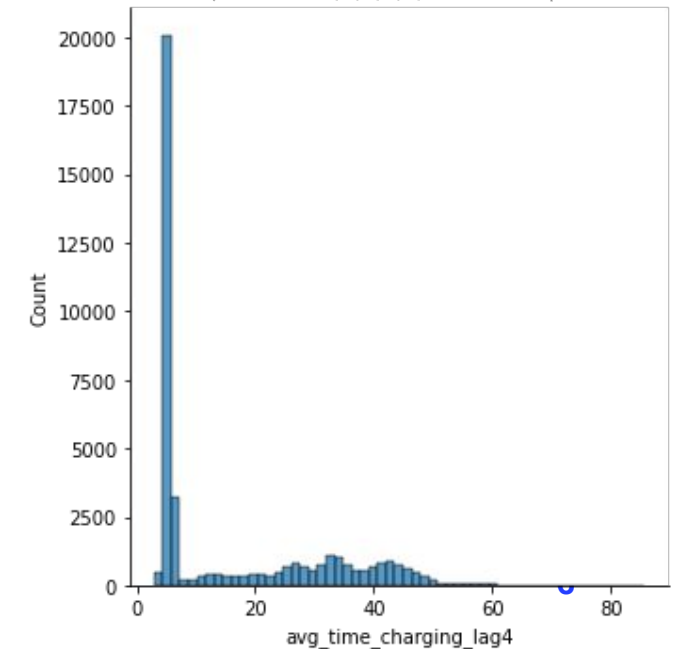
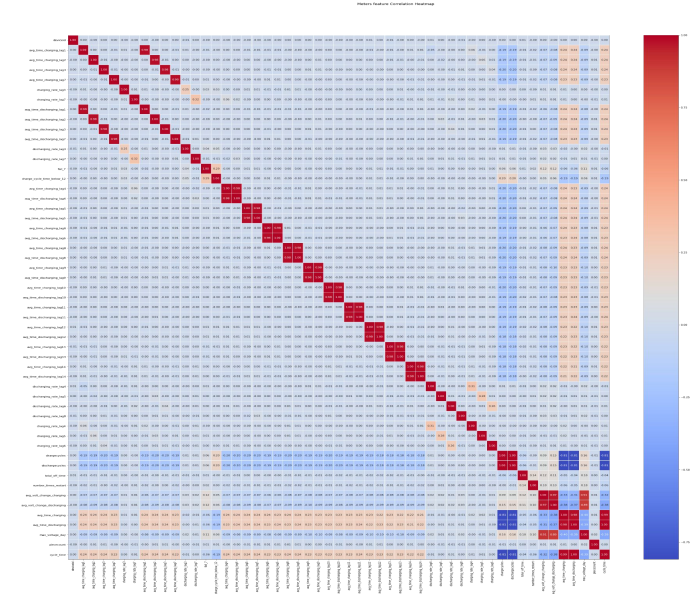
After uploading the train dataset, we identified the following:

- Some data are not Gaussian distribution, so **normalization/standardization** process is needed.
- **Outliers** exists within dataset, however, we run the models with and without outliers and this did not provide an improvement for our model.
- Avg_time_dis/charging_lag for more than 10 days account for the majority of **missing values**.
- Some features are **correlated** to feature fail_7.
- Failed meters instances only represent 23% of the data provided for fail_7. This means that the dataset is **imbalance**.

Data preprocess and model strategy:

F1 score is important for imbalance data, and a biased model will be performed without oversampling or undersampling.

PITNEY & BOWES DATA CHALLENGE 2021



DATA PREPROCESSING

Normalization & Standardization

MinMaxScaler and Zscore Normalization changes the values of dataset to a common scale without affecting the difference in the range of the values.

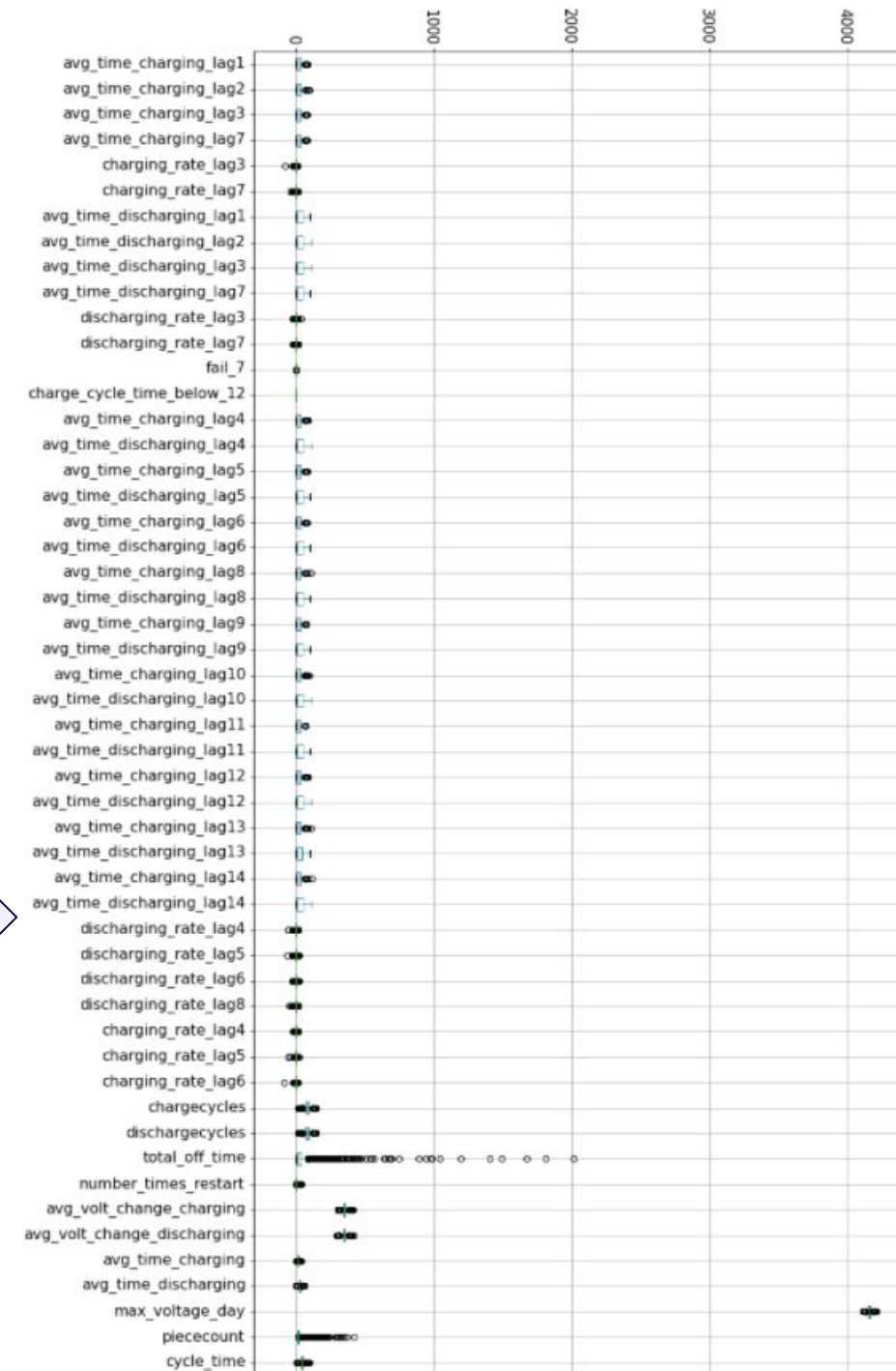
yeo-johnson Standardization for long tailed data.

Outliers

We removed the outliers to see the effect they will cause in the metrics of the models. It turned out that removing outliers **does not provide any significant improvement to the models' performance** metrics (accuracy, precision, recall and F1 score).

But we decided to keep the outliers because:

- 1) lack of domain knowledge
- 2) valuable info for deep exploration



DATA PREPROCESSING

Missing Values

Train dataset has 33,713 missing values.

To estimate missing values, we used the following methods:

- 1) Mean
- 2) Median
- 3) Previous values by deployed date: **based on assumption that meters deployed at same day has similar parameters.**
- 4) **Linear Regression Imputer : based on some missing values has correlations with other variables.**
- 5) KNN imputer

We found that the best estimator for the missing values from model performance view is:

→ Linear Regression Imputer

Linear Regression Imputer metrics

	MAE	MSE	RMSE	R2	RMSLE	MAPE
0	1.8444	8.5837	2.9298	0.9671	0.2282	0.1673
1	1.8443	6.4036	2.5305	0.9679	0.2237	0.1799
2	1.9321	8.9616	2.9936	0.9587	0.2089	0.1681
3	1.7459	7.5283	2.7438	0.9648	0.2266	0.1760
4	1.8020	6.5387	2.5571	0.9713	0.1894	0.1564
5	2.0164	9.5724	3.0939	0.9642	0.2712	0.2111
6	1.6548	5.6747	2.3822	0.9746	0.2131	0.1743
7	1.7145	6.0255	2.4547	0.9730	0.2293	0.1777
8	1.8488	6.8815	2.6233	0.9693	0.1868	0.1554
9	1.8274	6.4130	2.5324	0.9745	0.2016	0.1722
Mean	1.8230	7.2583	2.6841	0.9686	0.2179	0.1738

DATA PREPROCESSING

To handle **imbalanced** data we tried:

- 1) SMOTE
- 2) SMOTHEENN
- 3) TomekLinks
- 4) EditedNearestNeighbors
- 5) SMOTETomek

We got the best result with technique **SMOTE**.

At the same time, SMOTE is practical to handle imbalance data in our models.

Comparison on Random Forest	Accuracy	F1
No SMOTE	80%	31%

Comparison on Random Forest	Accuracy	F1
SMOTE	75%	45%
SMOTHEENN	63%	48%
TomekLinks	80%	37%
EditedNearestNeighbors	73%	49%
SMOTETomek	75%	45%

FEATURING ENGINEERING-TECHNICAL METHODS

- PCA -

We applied **Principal Component Analysis (PCA)** which can improve the performance of the classifier.

We used these two PCA methods:

- **Linear PCA:** which reduces the dimensionality in the data when there is a linear pattern in the decision boundary that separates the two classes, Fail (1) and Non-Fail (0).
- **RVF Kernel PCA:** which reduces the dimensionality in the data when there is a non-linear pattern in the decision boundary that separates the two classes, Fail (1) and Non-Fail (0).

Note: applying PCA (both linear and Kernel) the result we got did not prove improvement to our model's performance.

- T Test -

To have more insight in the pattern/relationship within the data, we applied the **t-Test**. By taking a sample from both classes Fail (1) and non-Fail (0).

It turned out that the null hypothesis was rejected.
 $m_1 \neq m_2$

Note: this means that the pattern between the dataset are not strong by chance but by other elements that we have yet to identify and measure.



- Polynomial-Features

- We also tried several polynomial feature methods.
- **We found that the polynomial_degree = 3 is the best one.**

FEATURING ENGINEERING-BUSINESS METHODS



CREATING NEW VARIABLES

There are 2 variables in the dataset that by themselves do not provide meaningful information to our model:

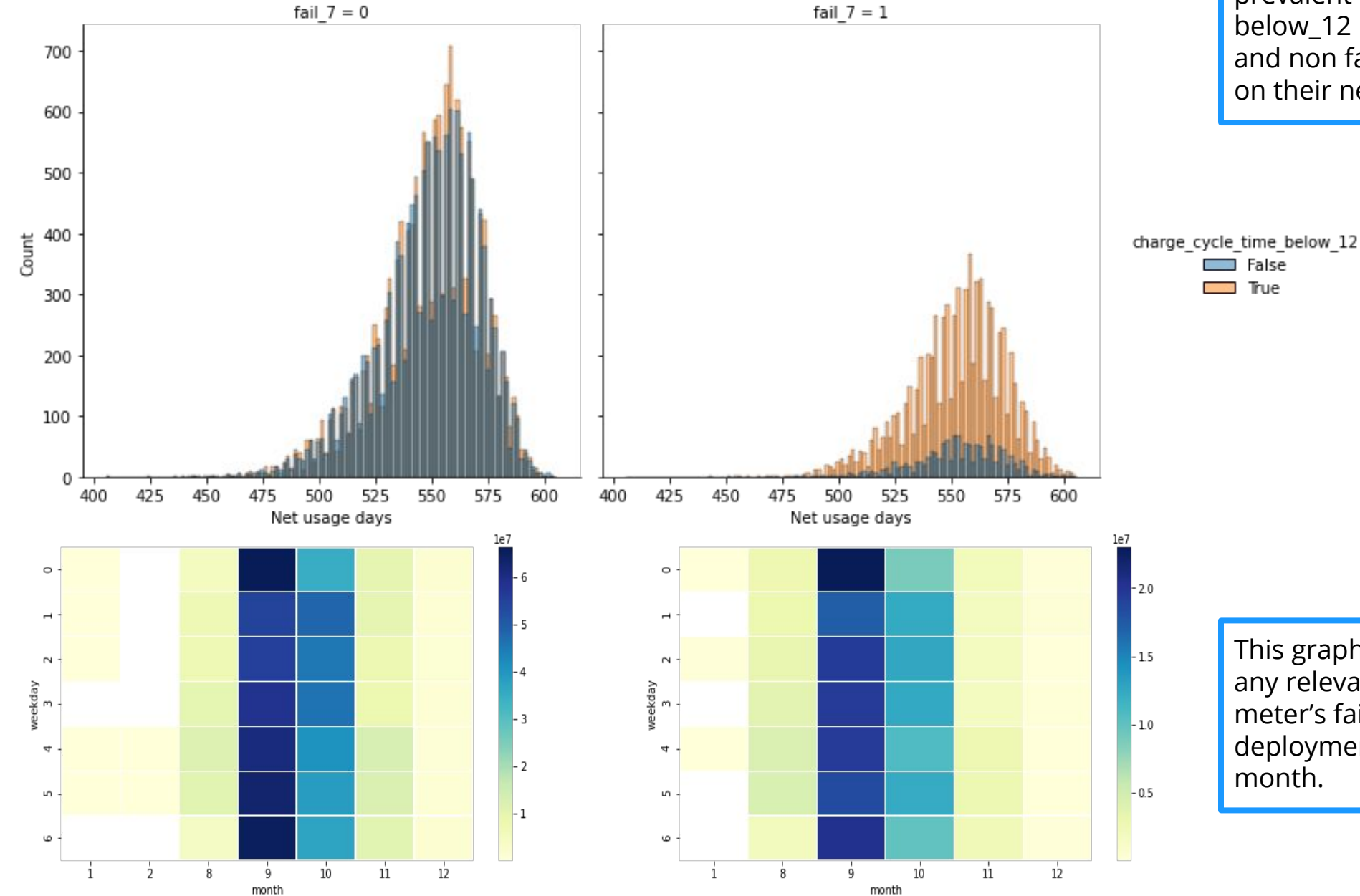
- **LastRecord**
- **DateDeployed**

By using LastRecord and DateDeployed created new features such as:

- **Net usage days**
- **is_business_day**
- **Weekday**
- **Month**

DateDeployed can imply more information from business view. For example, the product production date may be similar or deployed on non-business day may have some urgent requirement or so.

New Features Insight

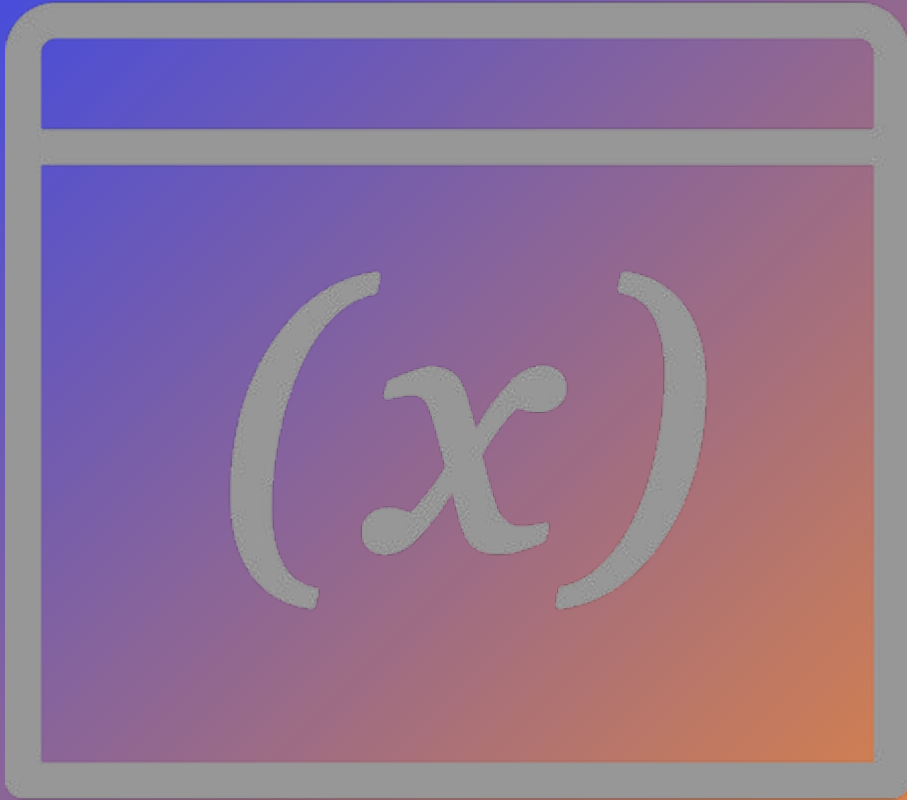


This graph shows how prevalent charge_cycle_time below_12 is among both failing and non failing meters, based on their net usage days

This graph shows if there is any relevance regarding the meter's failure by it's deployment weekday and month.

Dropping insignificant variables

- After creating Net Usage Days variable, we dropped variables:
 - **LastRecord**
 - **DateDeployed**
- We also dropped variable **deviceid**, which serves as an identifier for each meter (row)



BUILDING, TRAINING & COMPARING ML MODELS

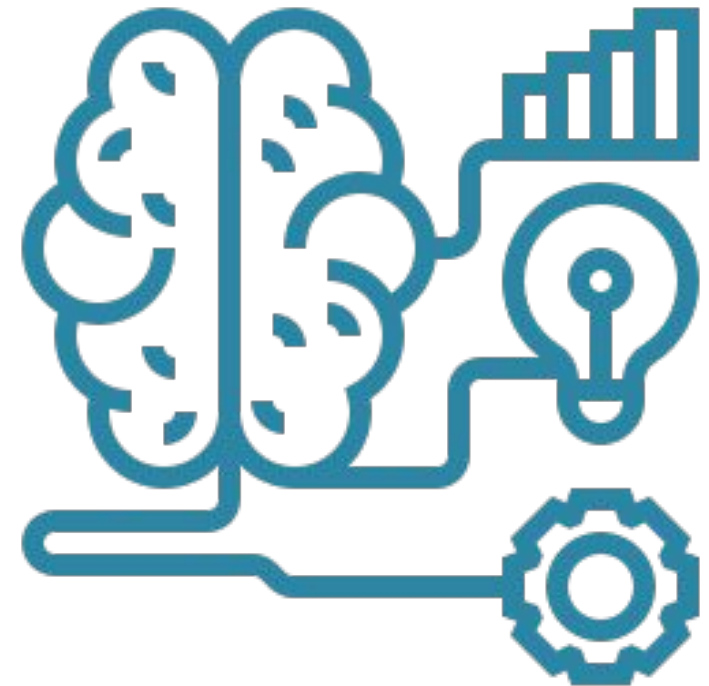
Models & Platforms/Libraries

Models:


1. Light Gradient Boosting Machine
2. Random Forest Classifier
3. Gradient Boosting Classifier
4. Ada Boost Classifier
5. Extra Trees Classifier
6. Decision Tree Classifier
7. SVM Linear Kernel
8. Logistic Regression
9. Linear Discriminant Analysis
10. Ridge Classifier
11. K Neighbors Classifier
12. Naïve Bayes
13. Quadratic Discriminant Analysis
14. H2OStackedEnsemble

Model platforms/libraries:


- Pycaret
- H2O
- sklearn
- fast.ai



*We used **cross validation** method to make our model more stable and robust.



MODEL EVALUATION

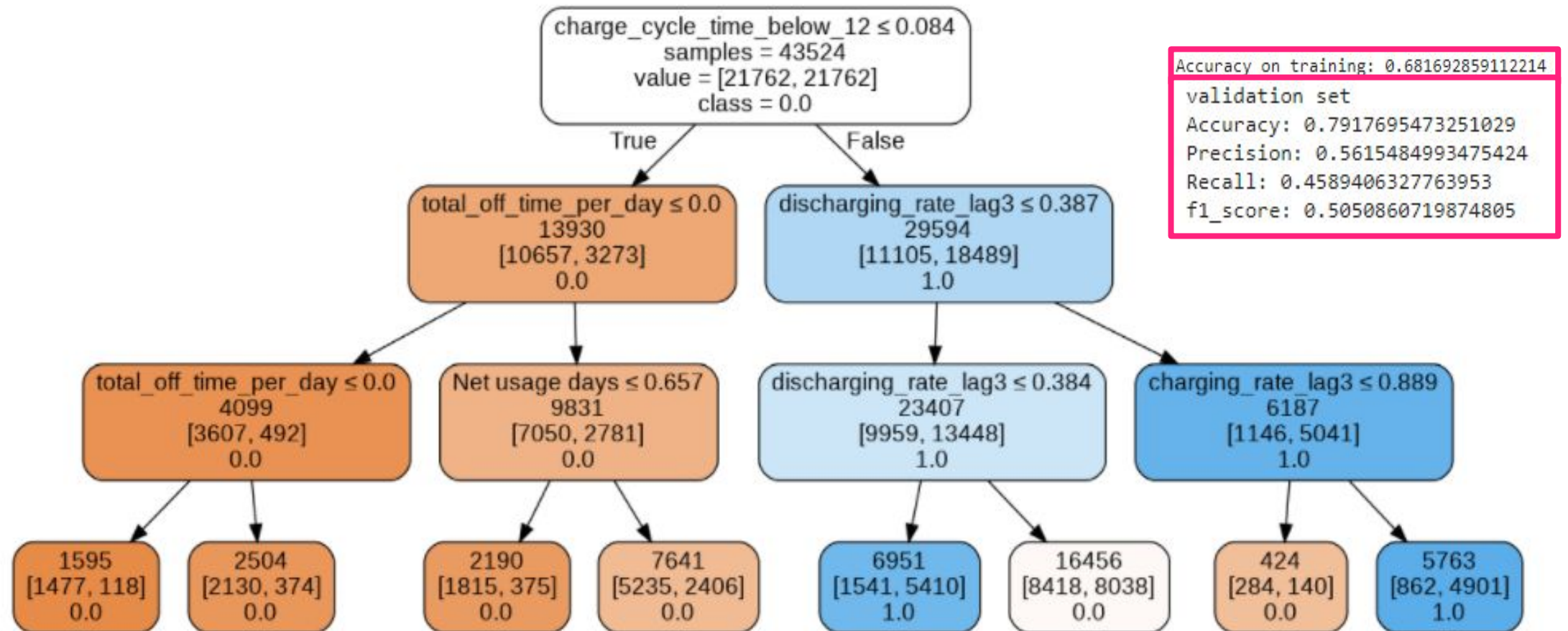


Pycaret Platform Performance

	Model	Accuracy	AUC	Recall	Prec.	F1	Kappa	MCC	TT (Sec)
lightgbm	Light Gradient Boosting Machine	0.8187	0.7851	0.4198	0.6688	0.5156	0.4111	0.4281	1.2160
gbc	Gradient Boosting Classifier	0.8126	0.7801	0.4156	0.6434	0.5048	0.3957	0.4102	7.8630
rf	Random Forest Classifier	0.8106	0.7689	0.3650	0.6599	0.4698	0.3660	0.3899	3.0120
xgboost	Extreme Gradient Boosting	0.8069	0.7643	0.3990	0.6258	0.4871	0.3753	0.3898	6.0140
ada	Ada Boost Classifier	0.7986	0.7653	0.4106	0.5903	0.4838	0.3637	0.3732	1.8420
et	Extra Trees Classifier	0.7705	0.7273	0.1358	0.5047	0.2137	0.1287	0.1673	1.8120
dt	Decision Tree Classifier	0.7114	0.6128	0.4302	0.3856	0.4066	0.2167	0.2173	0.7620
lr	Logistic Regression	0.6284	0.7236	0.7732	0.3576	0.4890	0.2546	0.3017	1.3820
ridge	Ridge Classifier	0.6170	0.0000	0.7930	0.3522	0.4878	0.2485	0.3010	0.3170
lda	Linear Discriminant Analysis	0.6168	0.7235	0.7926	0.3521	0.4876	0.2482	0.3006	0.4730
svm	SVM - Linear Kernel	0.5837	0.0000	0.8375	0.3370	0.4806	0.2271	0.2937	0.5390
knn	K Neighbors Classifier	0.4652	0.5443	0.6528	0.2480	0.3595	0.0394	0.0534	3.1830
nb	Naive Bayes	0.3775	0.6636	0.9049	0.2584	0.4012	0.0678	0.1351	0.4260
qda	Quadratic Discriminant Analysis	0.2299	0.4999	0.9998	0.2299	0.3738	-0.0000	-0.0024	0.5570

Decision Tree Performance

- We tried depth 1 to 10 and found that depth 3 is the best.
- We ran the Decision Tree in depth 3 after SMOTE and found the following.



H2O Stacked Ensemble

- Using H2O ensemble, we built and stacked 3 different models:
 - Generalized linear modeling
 - Gradient boosting machine
 - Random forest

Best Base-learner Test AUC: 0.757926485713008

Ensemble Test AUC: 0.7695380762867237

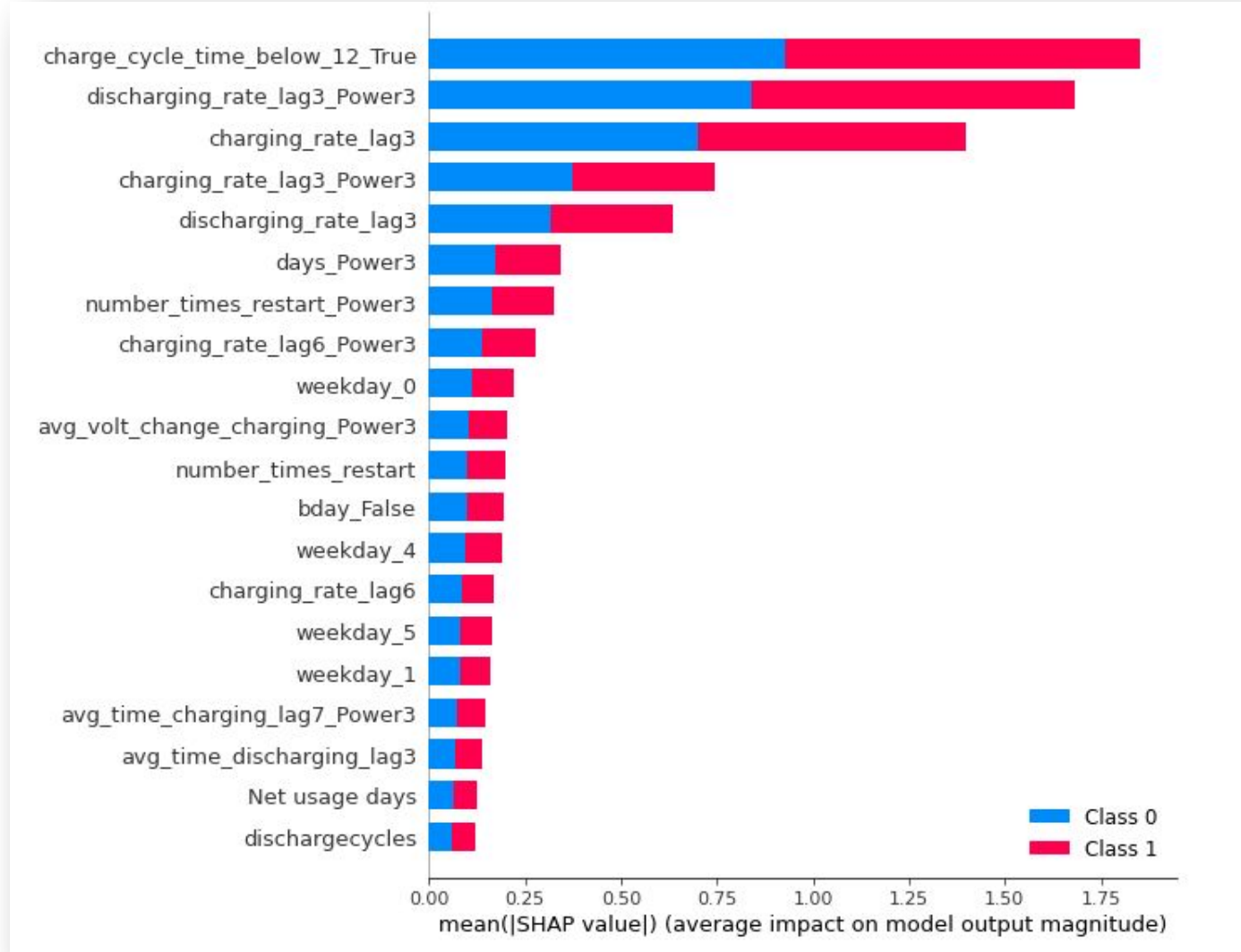
Best Base-learner Test Accuracy: [[0.9999574248198284, 0.9407407407407408]]

Ensemble Test Accuracy: [[0.8475964528597448, 0.9686419753086419]]

Best Base-learner Test f1: [[0.9872285842783786, 0.231473771856786]]

Ensemble Test f1: [[0.6710299964704983, 0.2234848484848485]]

Explainability: SHAP Graph



From technical perspective,

As shown in the feature importance diagram using SHAP, we conclude that the top 5 features indicating a likelihood of a failed meter are the followings:

1. **charged cycle time below 12**
2. **discharging rate lag3 power3**
3. **charging rate lag3**
4. **charging rate lag3 power3**
5. **discharging rate lag3**

Therefore, charged cycle time below 12 is the most important factor that affects the failure of a meter. Additionally, charging and discharged rate for the previous 3 days have significant impact as well.

These top features are consistent with decision tree model.

Model Comparison

Models	Accuracy	F1 score	Explainability	Runtime
LightGBM	81%	51%	Low but SHAP	Low
Decision Tree	79%	50%	High	Low
H2O	97%	22%	Low	Medium

We trained multiple models. Some models have higher accuracy but lower F1, some models have lower accuracy but higher F1. For the underfitting/overfitting consideration, we take **LightGBM** as our final model. And **Decision Tree** performed not too bad and it's more easy to understand from business perspective and real world deployment.

Business Conclusion

Based on our study of the data, the model's performance(SHAP graph on Lightgbm model and decision tree), we conclude that business wise:

- Pitney Bowes can estimate which meters will fail within 7 days by using the features below. By doing so, P&B will be able to identify the meters that will fail, provide scheduled maintenance to meter fails and avoid the customer's business to be interrupted.

Top technical features:

1. Charge Cycle time below 12
2. Discharging Rate by day 3
3. Charging Rate by day 3

Top business features:

1. Number days in use
2. Number of times that meter restarts

From business view, charge cycle time and rate are two major features. The latest 3 days data is still important to predict failure meter.

Recommendations

- We believe that our model can be improved by adding additional features which were not covered in the dataset. For example, **Temperature** is also a critical feature that could add valuable information to the model.
- We can customize the models based on business requirement with **Threshold adjustment**. For example, decrease false positive but increase false negative.
- Because 23% of the meters in train set fail and we discovered that the charging cycle time below is the top 1 feature, we understand that these are key factors of the battery used by meters. We recommend P&B to consider **try other batteries with the meters** to test for improvement in fail_7 and the 5 top features found.



THANK YOU

