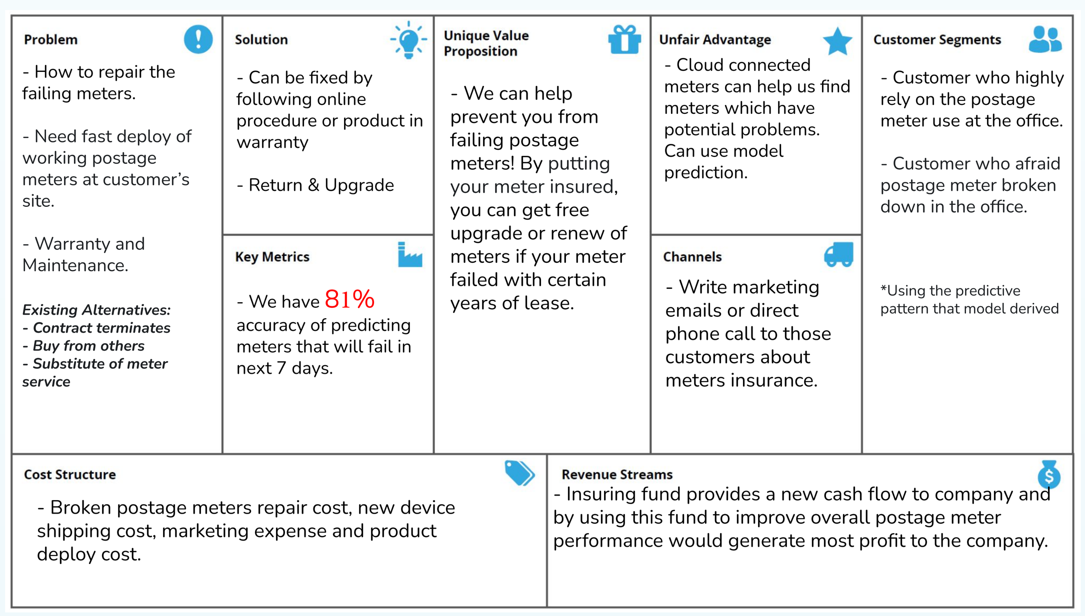
Baruch Data Science Challenge 2021
Meters Failure Prediction Project

Group 9:
Tianhao Wu, Jingnan Qi, Kaiyuan Song

Business Understanding

Lean Canvas



Data Understanding

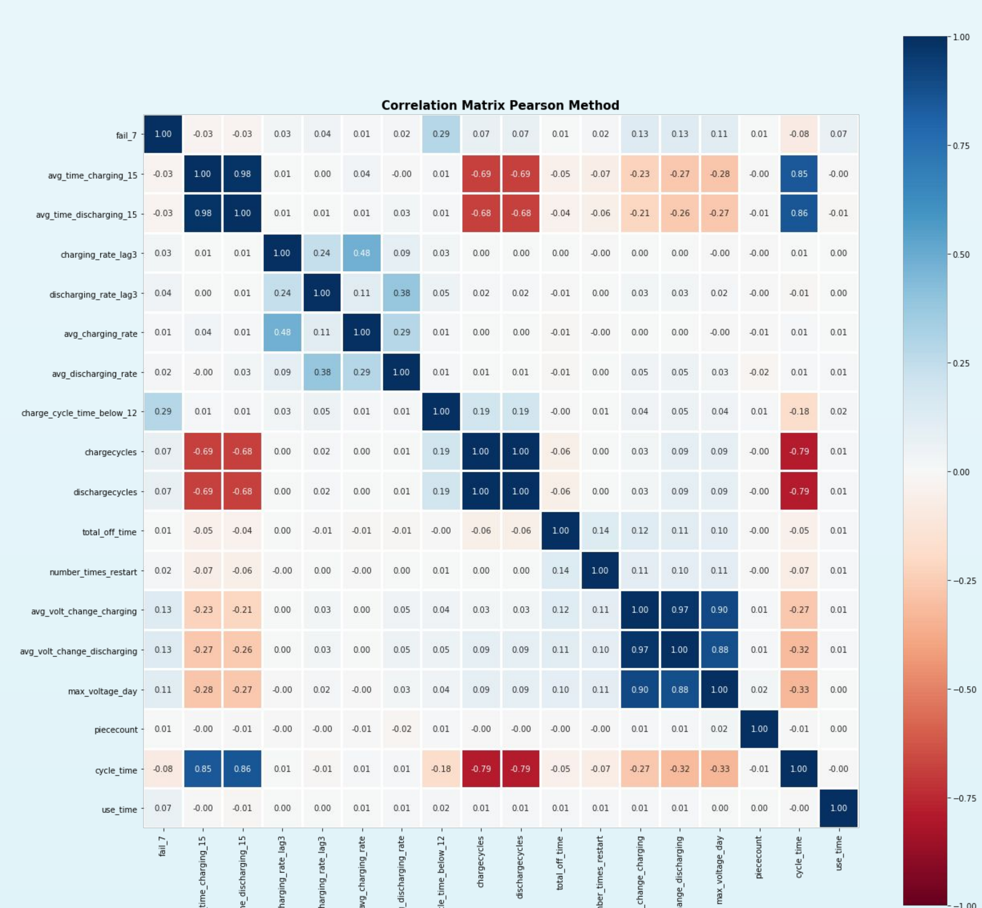
train.csv - 40500 meters
test.csv - 4500 meters

Target variable: fail_7 (1) failed (0) not failed

Total features: 54
- 51 numerical
- 1 bool
- 2 date

Data Quality Check

Timeliness	√	The information is available at last record date 4/1/2021.
Completeness	!	Missing data exists. 4 lag columns have more than 10% missing values. Need to be cleaned.
Uniqueness	√	No duplication.
Validity	√	Each column conforms to its format.
Consistency	!	Outlier detected (6.44%). Need to be removed.
Accuracy	√	The information represents the reality of the situation.



Top Features correlated with fail_7:

charge_cycle_time_below_12	0.29
avg_volt_change_charging	0.13
avg_volt_change_discharging	0.13
max_voltage_day	0.11
chargecycles	0.07
dischargecycles	0.07
use_time	0.07
cycle_time	-0.08

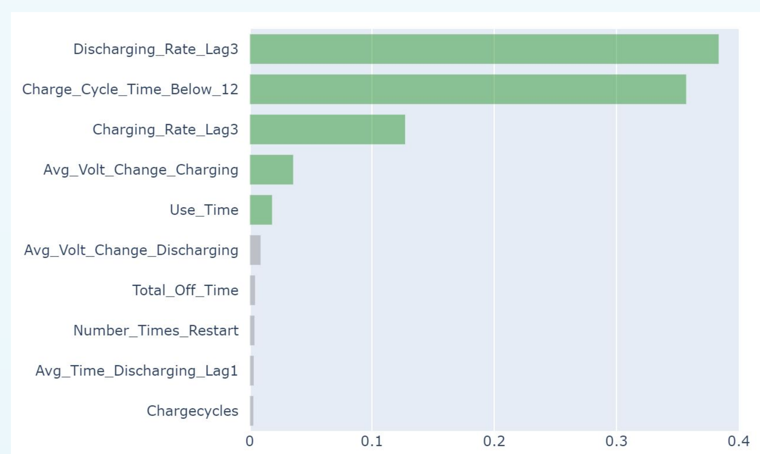
Data Preparation

Data Dictionary

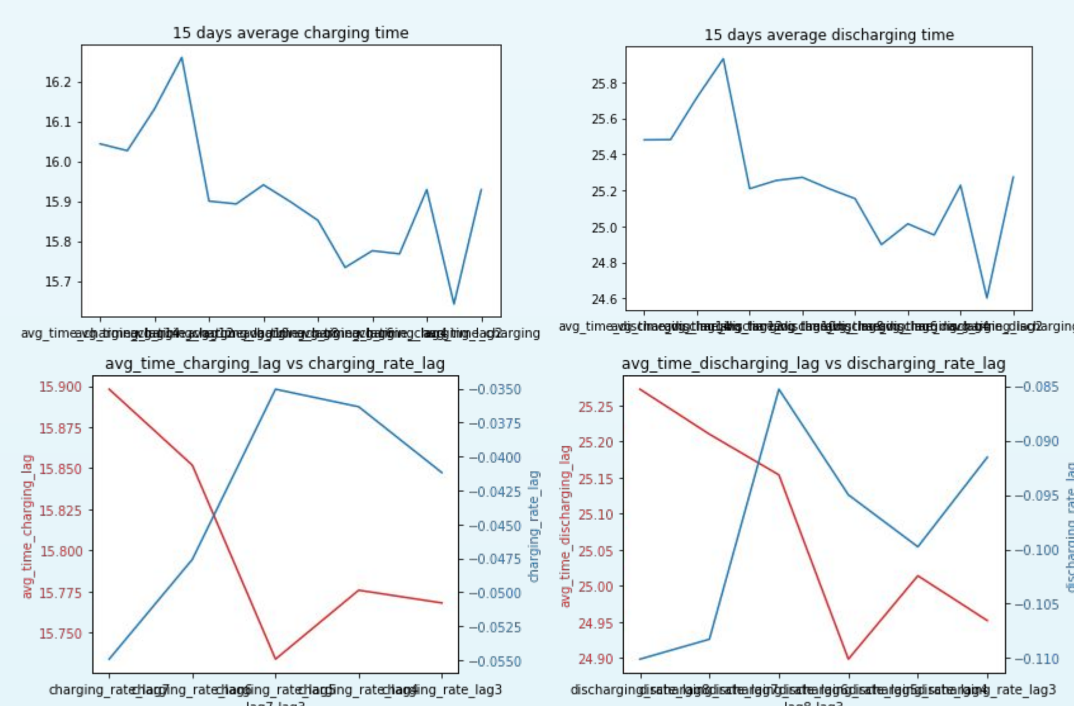
- **Get rid of unwanted columns**: deviceid, LastRecord (all 4/1/2021), Date Deployed
- **Scale Features**: normalize X variables for better performance in KNN and other ML algos
- **Handle Missing Values**: impute utilizing k-Nearest Neighbors from other X variables
- **Remove outliers**: detect anomalies using *isolation forest* (2609 records/6.44%)
- **Derived attributes**:
  - use_time = LastRecord - Date Deployed
  - avg_time_(/dis)charging_15 = average of 15 days of average (/dis)charging time
  - avg_(/dis)charging_rate = average of 7 days of (/dis)charging rate

Feature Importance (from GBM)

- **Top 10 Relative Feature Importance**: decrease weighted impurity/increase information gain
- Top 5 dominate
- Why **lag 3** is more important than other days? ->"Monday Effect"

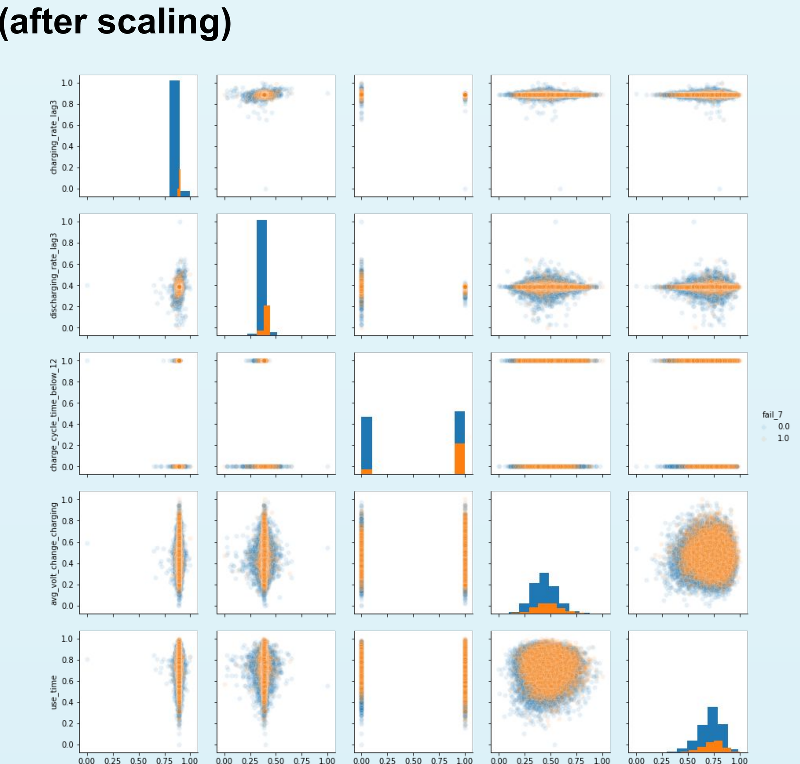


	importance
discharging_rate_lag3	0.384039
charge_cycle_time_below_12	0.356454
charging_rate_lag3	0.127867
avg_volt_change_charging	0.035832
use_time	0.018514
avg_volt_change_discharging	0.009257
total_off_time	0.004567
number_times_restart	0.004348
avg_time_discharging_lag1	0.003751
chargecycles	0.003381



- We can see that charging/discharging time and charging/discharging rate have **negative relationship**
- Especially for lag3 (Monday), the average charging/discharging time spent is relatively higher, while the charging/discharging rate is relatively lower

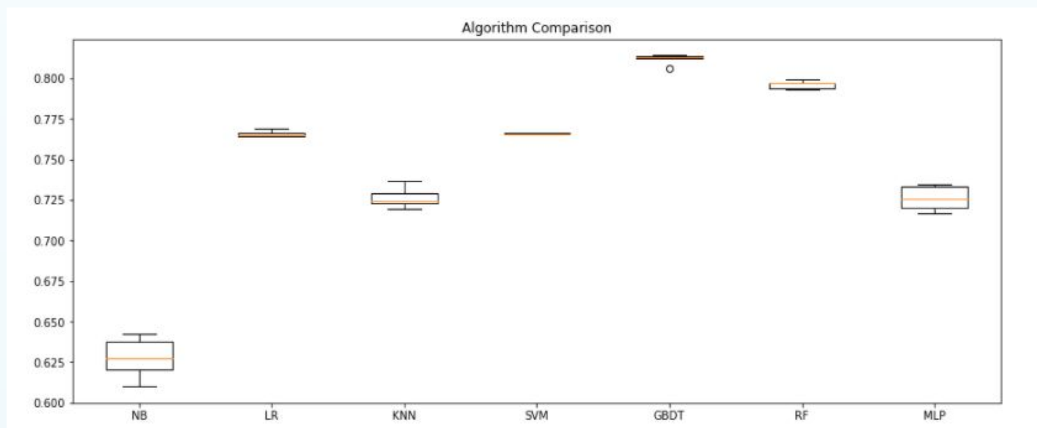
Pairplot of the 5 most important features with target variable (after scaling)



- Blue: 0 (not failed)
- Yellow: 1 (failed)

Modeling

Model Selections: Gradient Boosting Decision Tree performs the best (Cross-validation accuracy)



Naive Bayes	0.627466	0.011617
Logistic Regression	0.765586	0.001852
K-Nearest Neighbors	0.726622	0.005926
Support Vector Machine	0.765822	0.000015
Gradient Boosting Decision Tree	0.811818	0.002791
Random Forest	0.795891	0.002208
Multilayer perceptron	0.726123	0.007150

- SVM takes too long to run and does not scale well in large samples datasets
- Random Forest generally underperforms by GBDT in all scenarios
- Even after parameter tuning, Logistic regression and Neural Network as shown above does not perform well enough
- **GBDT is the one to go!**

Feature Engineering

- initial: ***0.8158***
- Only use the 5 dominant features: ***0.8165***
  - discharging_rate_lag3
  - charge_cycle_time_below_12
  - changing_rate_lag3
  - avg_volt_change_changing
  - use_time
- Add interaction & polynomial features of 5: ***0.8132***
  - Interaction terms: use_time * charge_cycle_time_below_12
  - Polynomial terms: avg_volt_change_chaging ^ 2
- Add average of charging rate in the past :***0.8126***
- Create other features w/ physical meaning: ***<0.8158***
  - charging_energy = avg_charging_rate * avg_charging_time
  - chaging_power = max_voltage_day ^ 2

Conclusion: use original features

Decision Trees & GBDT

- Decision Trees:
  - Partition data into subsets containing similar values
  - DT finds out the most informative splits about target variables
- GBDT:
  - Combine multiple Decision Trees together in a serial manner
  - Each tree is very shallow, and corrects mistakes from previous ones

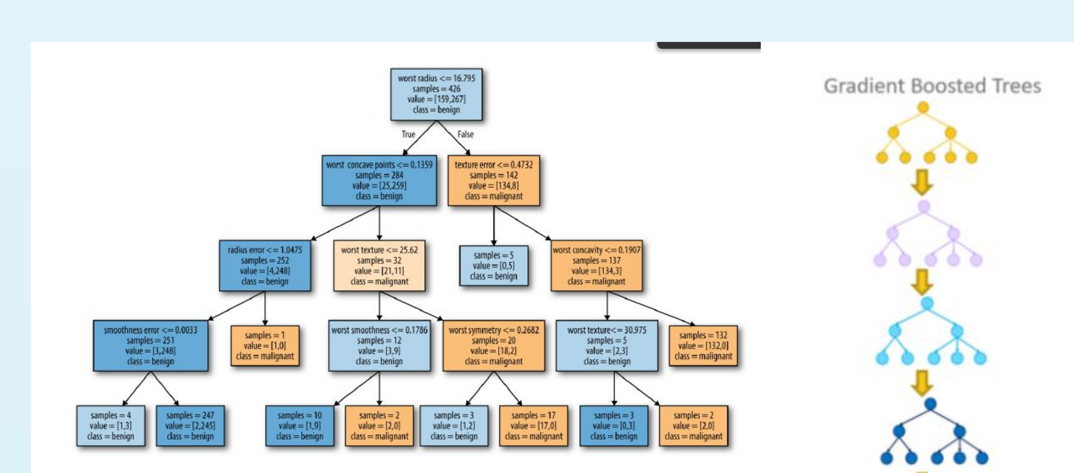
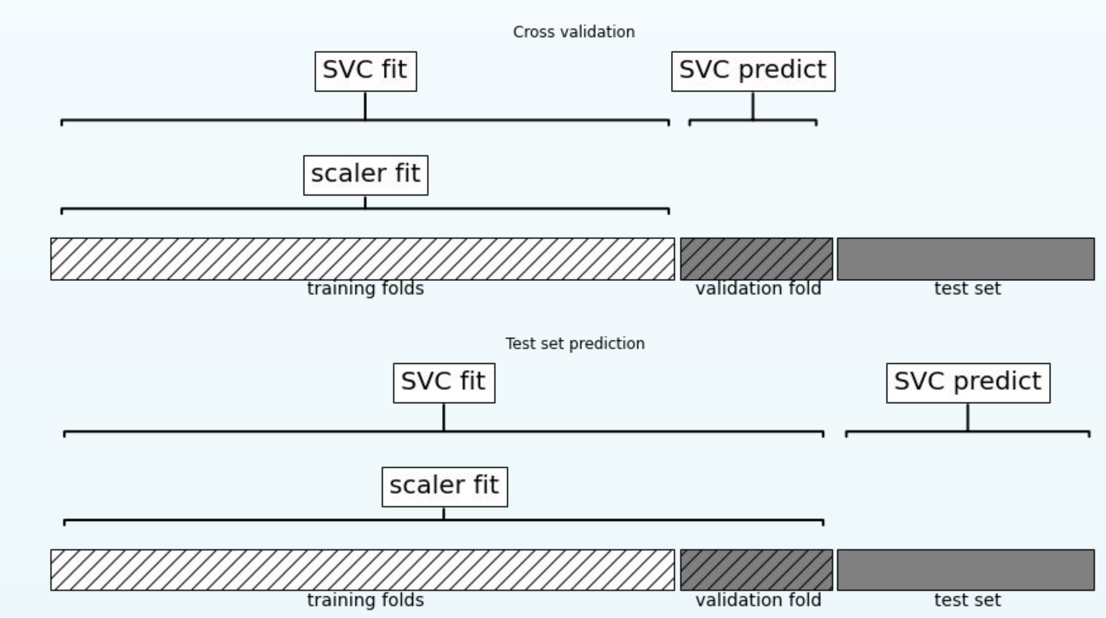


Figure 2-27. Visualization of the decision tree built on the Breast Cancer dataset

Evaluation

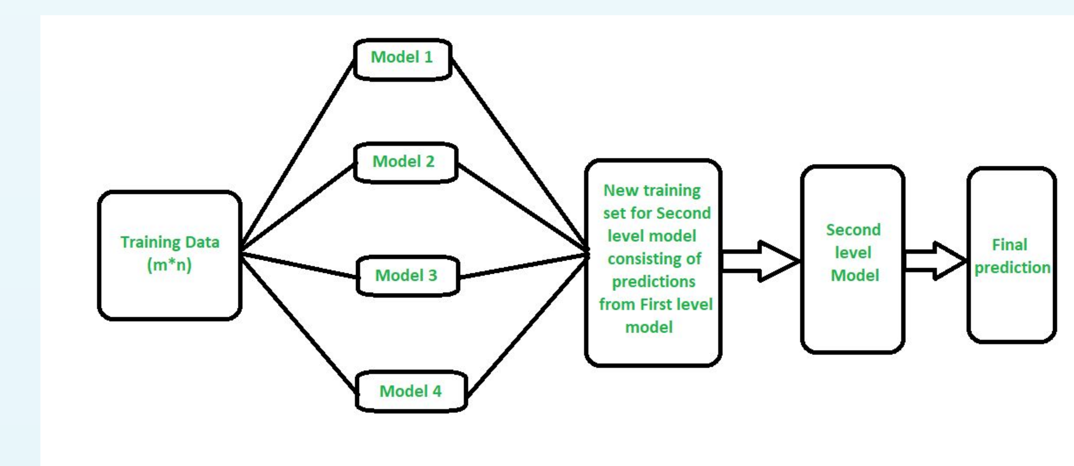
Data Leakage & Pipeline

- When doing imputation before splitting folds in cross-validation, info from validation fold will leak to training folds, it will generate overly optimistic result, and affect selection of suboptimal parameters
- **SOLUTION**: use pipeline, to do dataset splitting before any preprocessing



Ensembling

- *Stacking* GBDT, Logistic Regression, and Neural Network together, to increase performance (0.816)
- Stacking: use multiple model outputs as features for 2nd level model to make final prediction



Business Conclusions

- Discharging rate 3 days ago & whether Charge cycle time<12 on the last recorded day are the most crucial features
- Average voltage change in charging and total days of use are also important
- By focus on monitoring these 5 core features & use GBDT to generate "dangerous" partitions to predict failure

Deployment

Design Thinking

What could be done if we know these postage meters are going to fail in next 7 days?

Postage meters can be fixed by following online procedure.

Determine fail conditions, send feedback to engineering team. Make product **fail safe** and ready to be repaired as soon as possible.

Company side

Have service team ready for all the postage meters have more than **442** days of use time.

Analysis the data from machine and customer side and send replacement of the problem part.

Customer side

For small postage meters, send new postage meters for customers in return of the broken (potential broken) one.

For big postage meters, deploy a service team to customer's office and do meter examination.