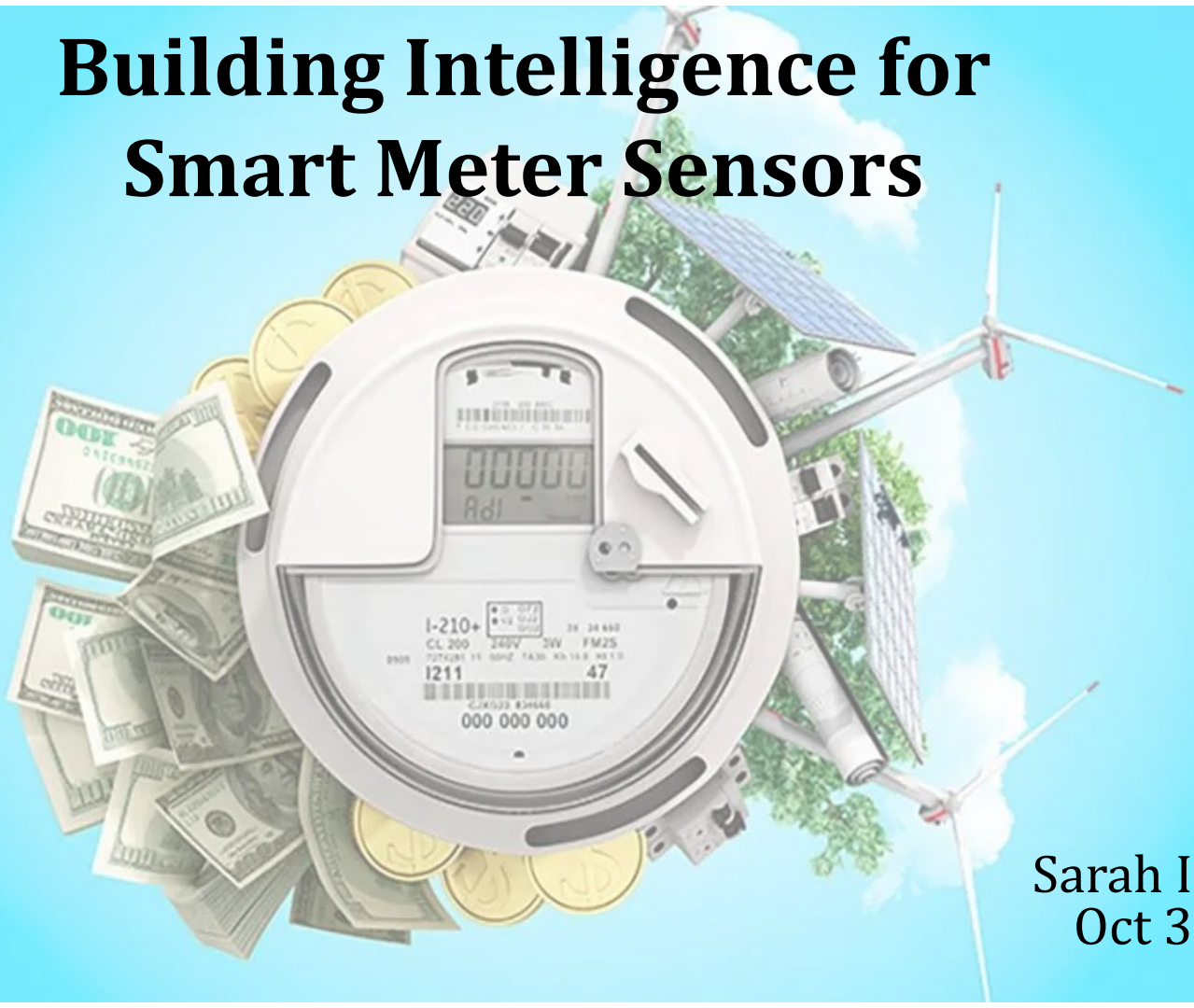
Building Intelligence for Smart Meter Sensors

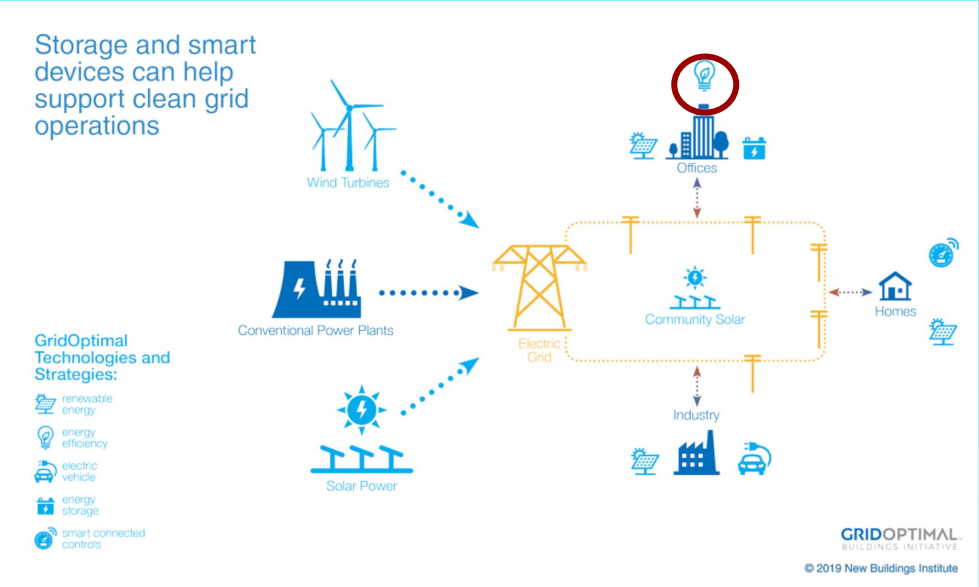


Sarah I. Daniels
Oct 3rd, 2022

The Problem

- 30% of energy consumed by buildings is wasted

- New Generation IoT
  → Smart Meter Transition



Objective: Develop an algorithm for anomaly detection

- *minimizes false positive rate*
- not sacrificing false negative rate

$$F_1 = 2 \cdot \frac{precision \cdot recall}{precision + recall}$$

Data

Labeled by: Large-scale Energy Anomaly Detection (LEAD) Team

Original Source: American Society of Heating, Refrigerating and Air-Conditioning Engineers

Building Features (N=200)

- Primary Use of Building
- Floor Count
- Square Feet
- Meter Reading
- Meter Reading Daily std dev
- Year Built

Time Features

- Month*
- Weekday*
- Hour*
- Holiday (y/n)

Weather Features

- Sea level pressure
- Dew temperature
- Air temperature
- Wind direction
- Wind speed
- Cloud coverage
- Precipitation Depth (1-hour)
- Site ID

*Modified to cyclic periods using

$$x_{sin} = \sin\left(\frac{2*\pi*x}{\max(x)}\right)$$

$$x_{cos} = \cos\left(\frac{2*\pi*x}{\max(x)}\right)$$

Temperature Lags

- Air temp max lag
- Air temp std lag

Stacking Classifiers**

- Meter reading x ID
- Meter reading x hour
- Meter reading x weekday
- Meter reading x month
- Meter reading x primary use
- Meter reading x site ID

**See appendix

Methods

Cleaning features (e.g., remove irrelevant fields, impute meter readings)
Exploratory Data Analysis (e.g., pair plots, correlation matrices)

STEP 1

STEP 2

Baseline Models (i.e., KNN, Logistic Regression)
Imbalance corrections (Logistic Regression)

Compare Model Performance (i.e, KNN, Random Forest, Decision trees)
Tuning hyperparameters (XGBoost)

STEP 3

STEP 4

Model Selection and Feature Importance

Python Tools: sklearn, matplotlib/plotly, seaborn

Baseline Models

	Precision	Recall	F1	Accuracy
KNN	0.70	0.36	0.47	0.98
Logistic Regression	0.0	0.0	0.0	0.97

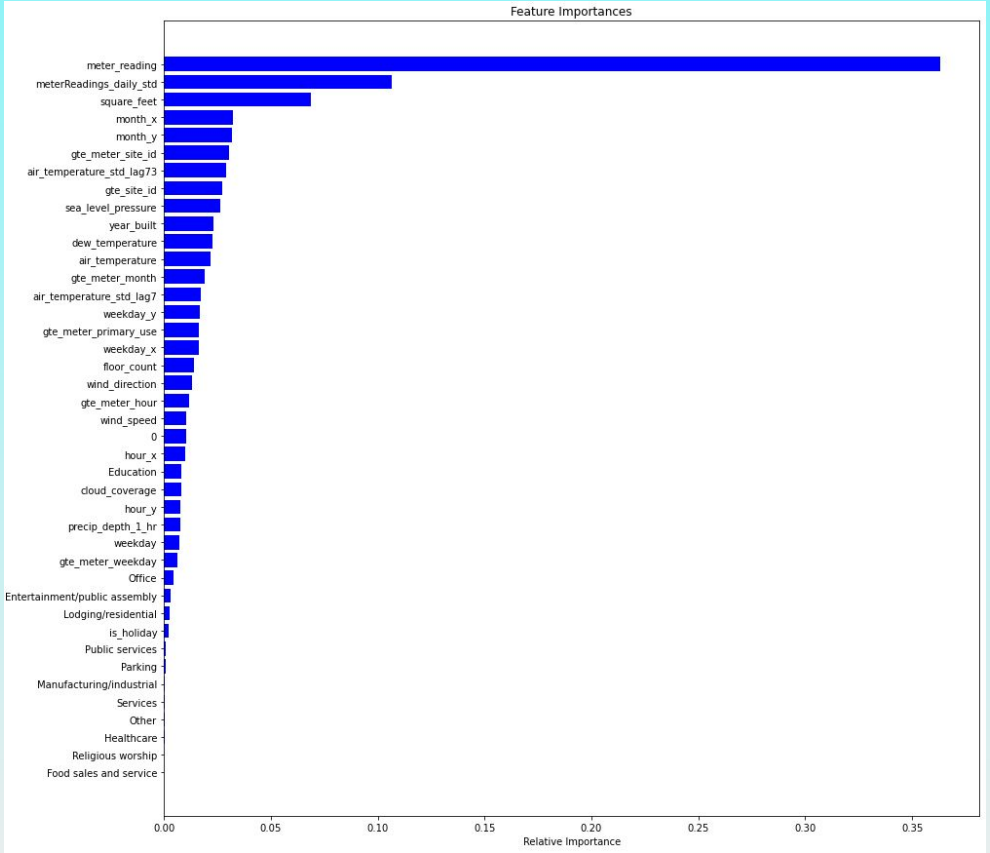
Model Comparison

	Precision	Recall	F1	Accuracy
KNN (Baseline)	0.70	0.36	0.47	0.98
Logistic Regression (Baseline)	0.0	0.0	0.0	0.97
Logistic Regression (w/weight balancing and threshold reduced)	0.03	0.76	0.58	0.37
Random Forest	0.93	0.70	0.80	0.99
XGBoost	0.89	0.62	0.73	0.99
XGBoost (hyperparameter tuning w/CV)	0.97	0.59	0.73	0.99

Best Performance Model - Random Forest

Top 5 Features

- Meter Reading (Energy consumption in kWh)
- Daily Std of Meter Reading
- Square Feet
- Month (sin/cos features)
- Stacked Feature
  (meter reading x weather site ID)



Future Direction

- Reassess Precision vs. F1 Score w/client using cost analysis
- Expand dataset to all 400 buildings w/labeled anomaly detection
- Develop faster run pipelines w/cloud computing to optimize Decision Trees
- Investigate other optimal stacking (or voting) ensembling methods
- Consider unsupervised learning models to increase recall

Conclusion

- Random Forest performs best in balancing precision/recall (F1 score), maintaining great precision
- Correct class recognition among predicted anomalies 93% of the time
- Features of greatest influence include variations of meter readings, building attribute, and time (month)

Appendix | Target Encoding Features

Feature Stacking:

1) Groups the data by category
2) Learns probabilities of target and predicts estimated values for each category
3) Include these predictions as additional features in main model

Bayesian Approach

$$f(\theta \mid y) = \frac{f(y \mid \theta)\, f(\theta)}{f(y)}.$$

Use prior distribution to calculate posterior distribution

$$\mu_{post} = \frac{\tau_{prior}\mu_{prior} + n\tau\mu_{mle}}{\tau_{prior} + n\tau}$$

Use prior mean to calculate posterior mean

$$\mu_{post} = \alpha\mu_{prior} + (1-\alpha)\mu_{mle}.$$

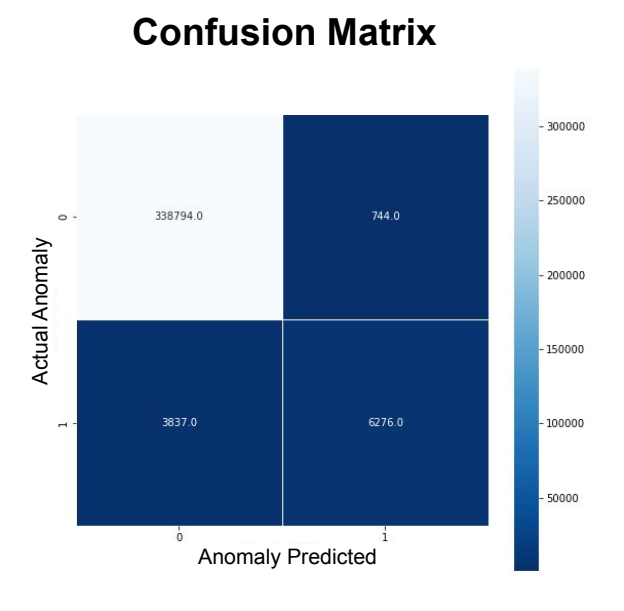
Use prior mean and mle mean to calculate posterior mean given α is:

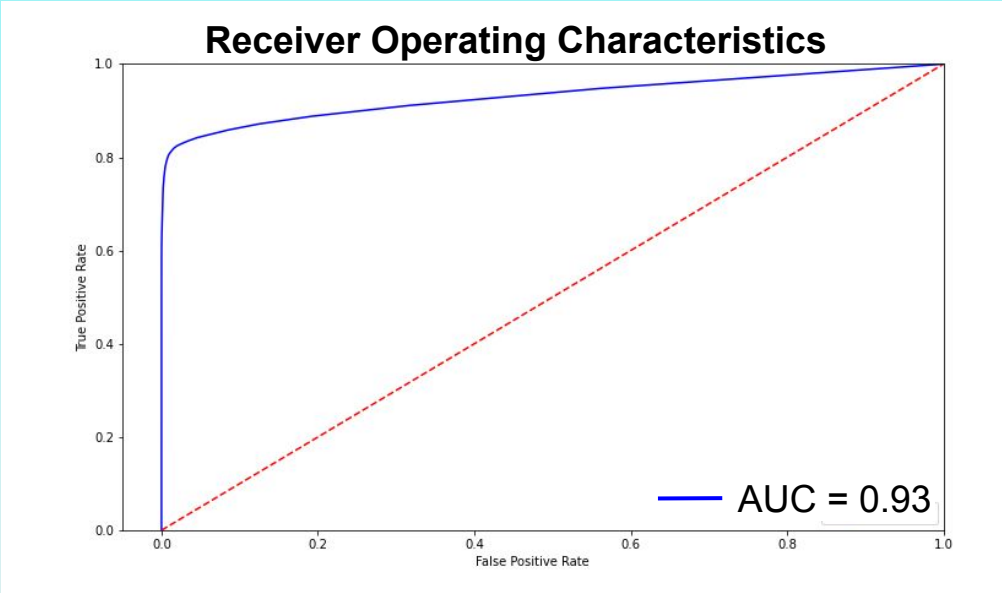
$$\alpha = \frac{\tau_{prior}}{\tau_{prior} + n\tau}$$

μ_{prior} =mean log(meter_reading)
μ_{post} =mean log(meter_reading |target)
"updated belief about data"
$\tau = 1/\sigma^2_{mle}$ = precision of maximum likelihood distribution of (target | meter_reading)
$\tau_{prior} = 1/\sigma^2_{prior}$ = precision of the prior distribution (regularization term)

https://www.kaggle.com/code/mmotoki/hierarchical-bayesian-target-encoding/notebook

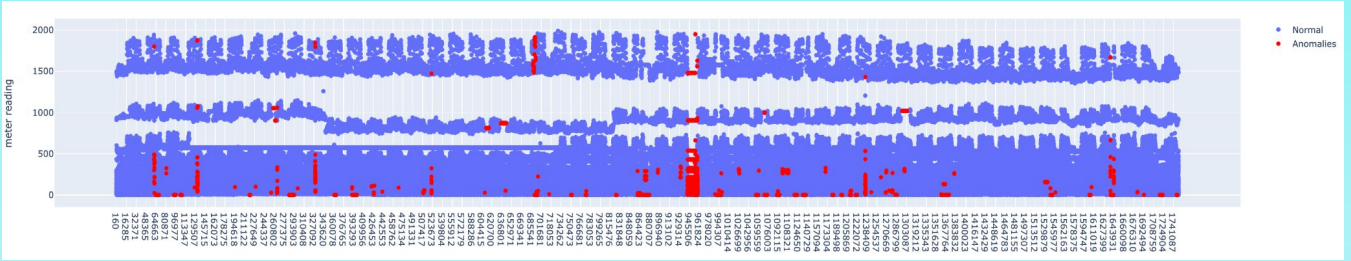
Best Performance Model -Random Forest





Anomaly Dependency on Meter Readings

Predicted anomalies (XGB)



True anomalies