Predicting Electricity Usage: Deliverable 2

Data Extraction, Processing, Initial Modeling, Model Comparison, Error Reports

Kevin Taylor | Kelly Du | Nathaniel Ho

Agenda

Introduction

Data

Variables

Pre-Modeling

Modeling

Results




Introduction

Problem

What will energy consumption look like after 2014?

Objective

Predict energy consumption after 2014 using data from 2011-2014

From:

Raw uncleaned dataset 'LD2011_2014.txt'

To:

Trained models with performance analysis

Value Creation

Evaluate the accuracy of a forecasting model on the data

Predict post-2014 energy consumption

Clean raw dataset so it is usable for other forecasting models

Data

Data Extraction

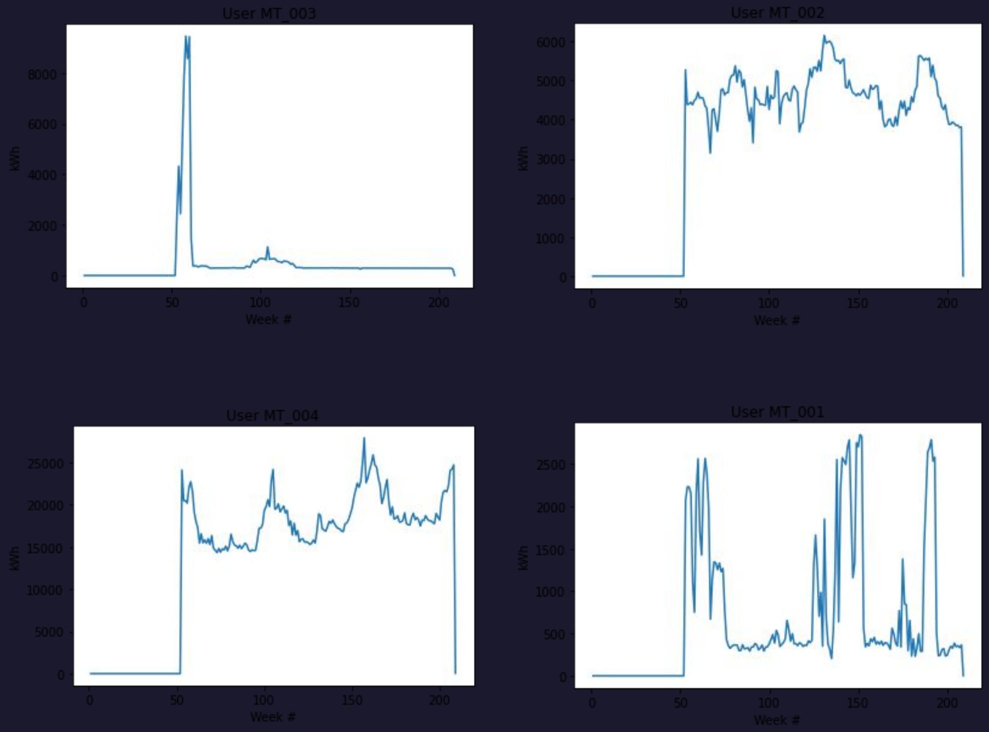
- Source: https://archive.ics.uci.edu/ml/datasets/ElectricityLoadDiagrams20112014#
    - Dataset is in a .zip file; extraction yields a .txt file
    - Text file delimited by ";"
    - Decimal values encoded as 0,00
    - Values in kW per 15 minutes - to convert to kWh, values must be divided by 4
    - Missing values ie) accounts created during the timeframe are encoded with zeros
    - Biannual time change results in either one hour of zeros or aggregation, depending on the season

- Shape of (140256, 370), or 140,256 rows and 371 columns (one for Datetime, 370 accounts)
- Head of raw data:

	Datetime	MT_001	MT_002	MT_003	MT_004	MT_005	MT_006	MT_007	MT_008	MT_009	...	MT_365	MT_366	MT_367	MT_368	MT_369	MT_370
0	2011-01-01 00:15:00	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0
1	2011-01-01 00:30:00	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0
2	2011-01-01 00:45:00	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0
3	2011-01-01 01:00:00	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0
4	2011-01-01 01:15:00	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0

Data Processing

- Data Extraction Process
  - Read entire dataset delimited by “;” into pandas DataFrame object
- Data Diagnostics
  - Dataset is clean (i.e. no duplicates, missing values, etc.)
  - Time change assumed not to affect overall trend (occurs every year)
- Modeling Preprocessing
  - Transform data into long pivot form
  - Divide all values by 4 to arrive at kWh, a unit of energy
  - Aggregate the dataset by account ID, year and week






Variables

Variables Overview

Target Variables	*Predictive Variables*

• Variable that is predicted in the forecasting model.
• Weekly usage of electricity in kWh is our "y."
• Denoted in DataFrame as *value*
	• Variables that predict target variable
	• Organized into direct or derived variables
	  ○ Direct variable: Directly from dataset
	  ○ Derived variable: Created by manipulating direct variables
	• All variables are direct

Predictive Variables Overview

- Initial model fit (SARIMA) does not use any exogenous variables
- SARIMA components (Autoregressive, Moving Average) use target variables at different lags, and weighted average forecast errors, as predictor variables, as well as a seasonal component.
- SARIMAX uses exogenous variables
    - ie) Holiday Week indicator, weather, additional seasonality component ie) quarterly, monthly
- The FB Prophet models trained did not use exogenous variables either



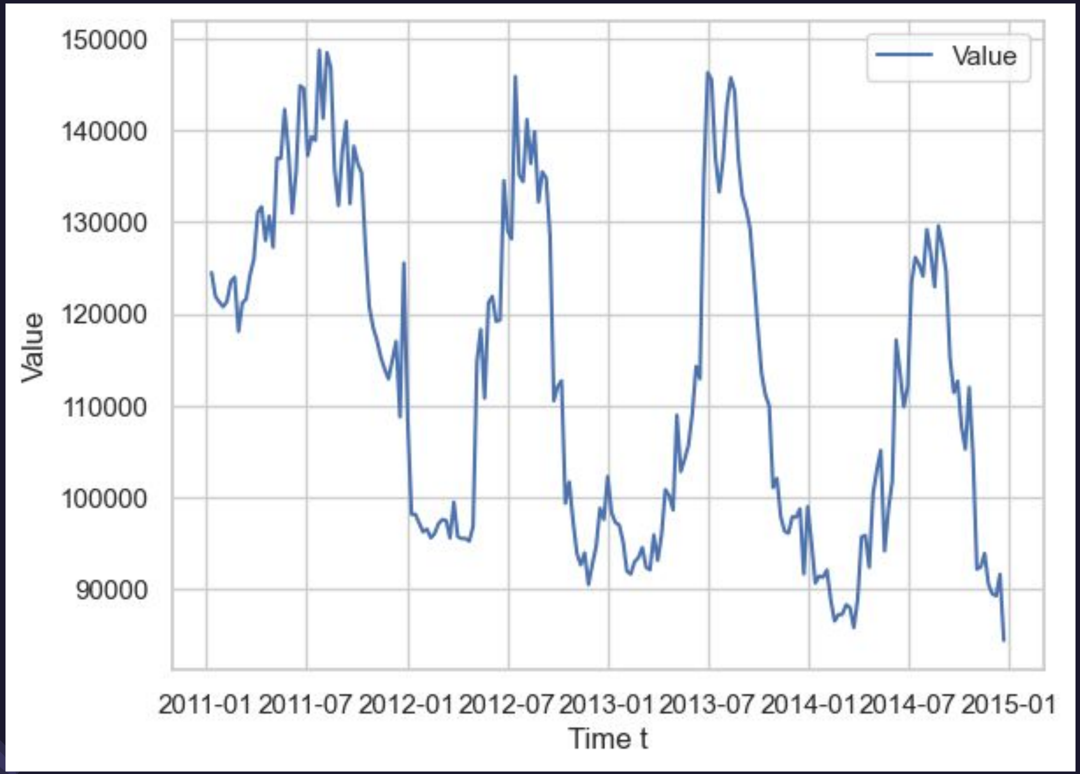

Pre-Modeling

Pre-Modeling

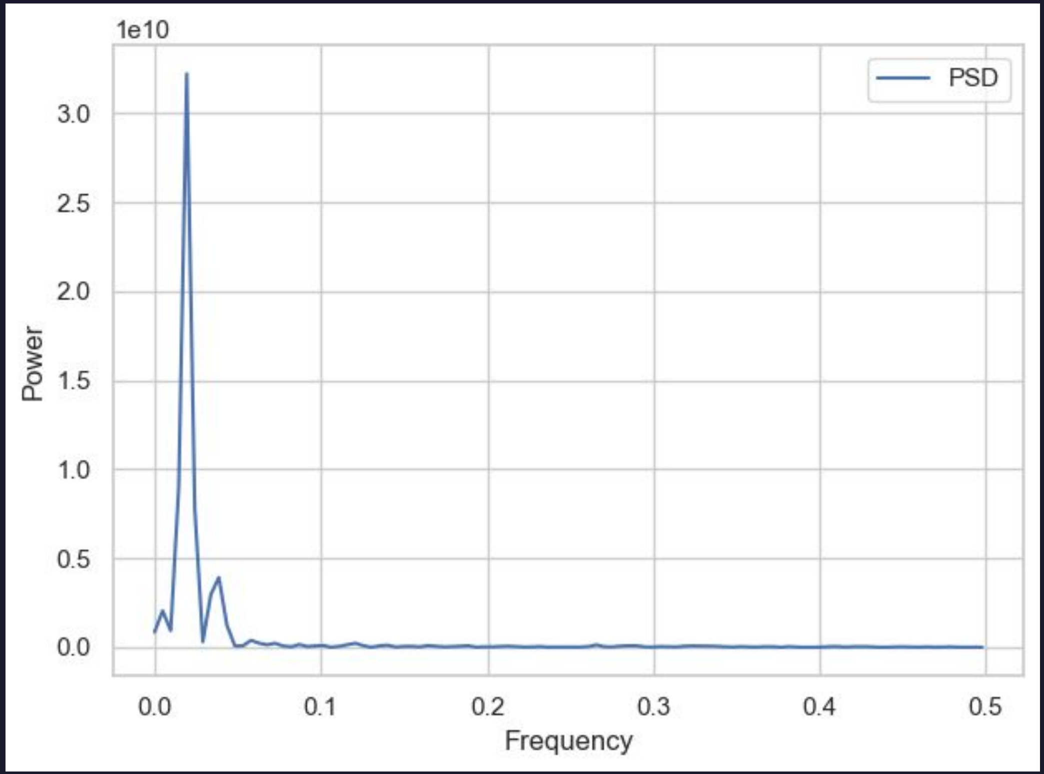
- Exploratory Data Analysis
  - Identify trend and seasonality in the data for initial modeling
  - Perform this exercise on all accounts aggregated, then apply the chosen model to each individual account
- Pre-Modeling Utility Functions
  - Create utility functions for train-validation-test split, MAPE, walk-forward validation, etc.

Pre-Modeling
Trend / Seasonality

Average Energy Use (kWh) - All Accounts

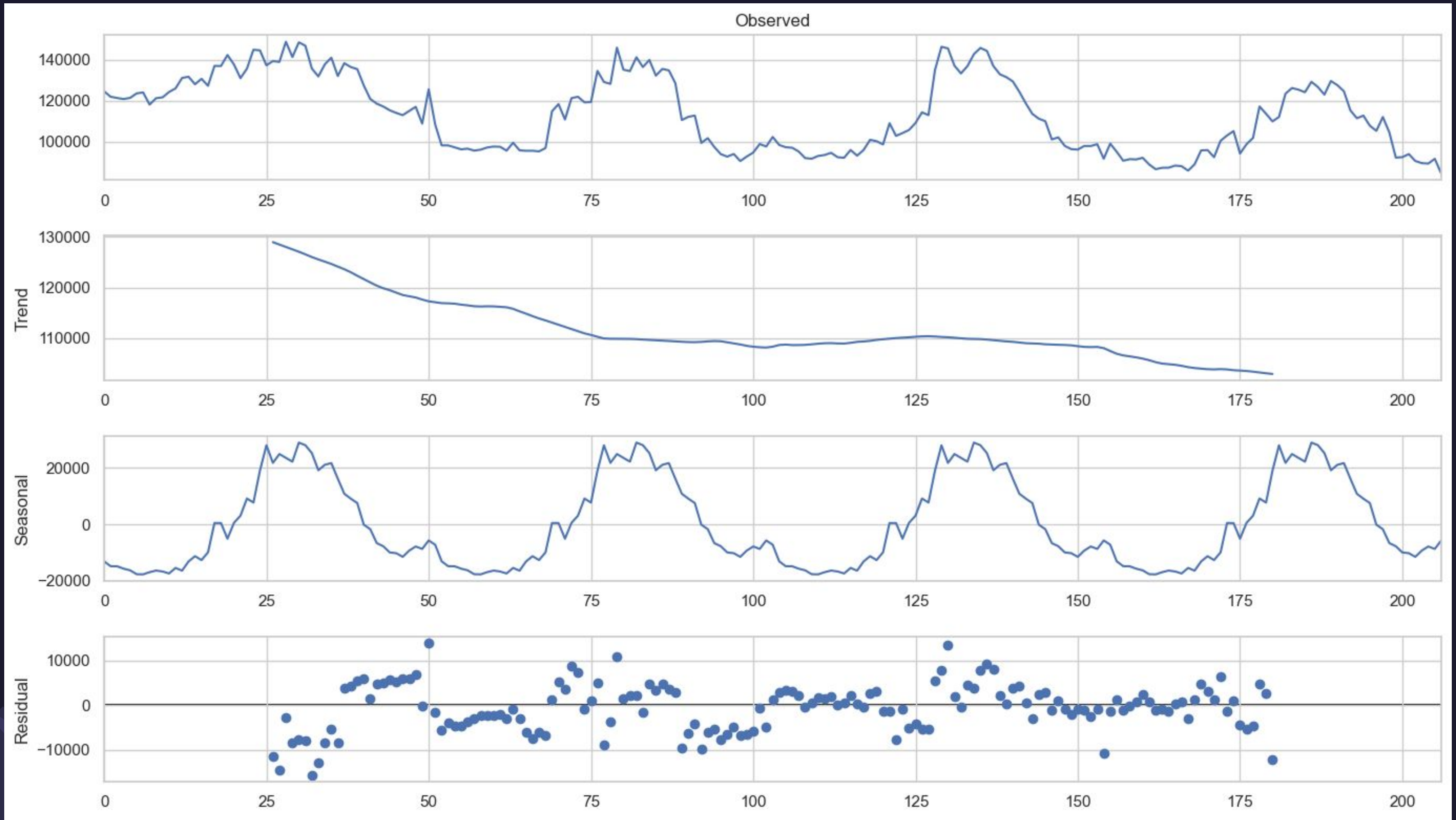


PSD of Average Energy Use



Time series plot and PSD indicate a first-order downward trend and yearly seasonality
Maximum power corresponds to a period of 51.75

Pre-Modeling
Seasonal Decomposition



- First order-trend indicates first-order differencing / integration component likely fits best on aggregated dataset
- Seasonal component appears to have a yearly period
- Initial choice for model parameters: SARIMA(0, 1, 1)(0, 1, 1)$_{52}$
- Hyperparameter tuning will be done




Modeling

Modeling

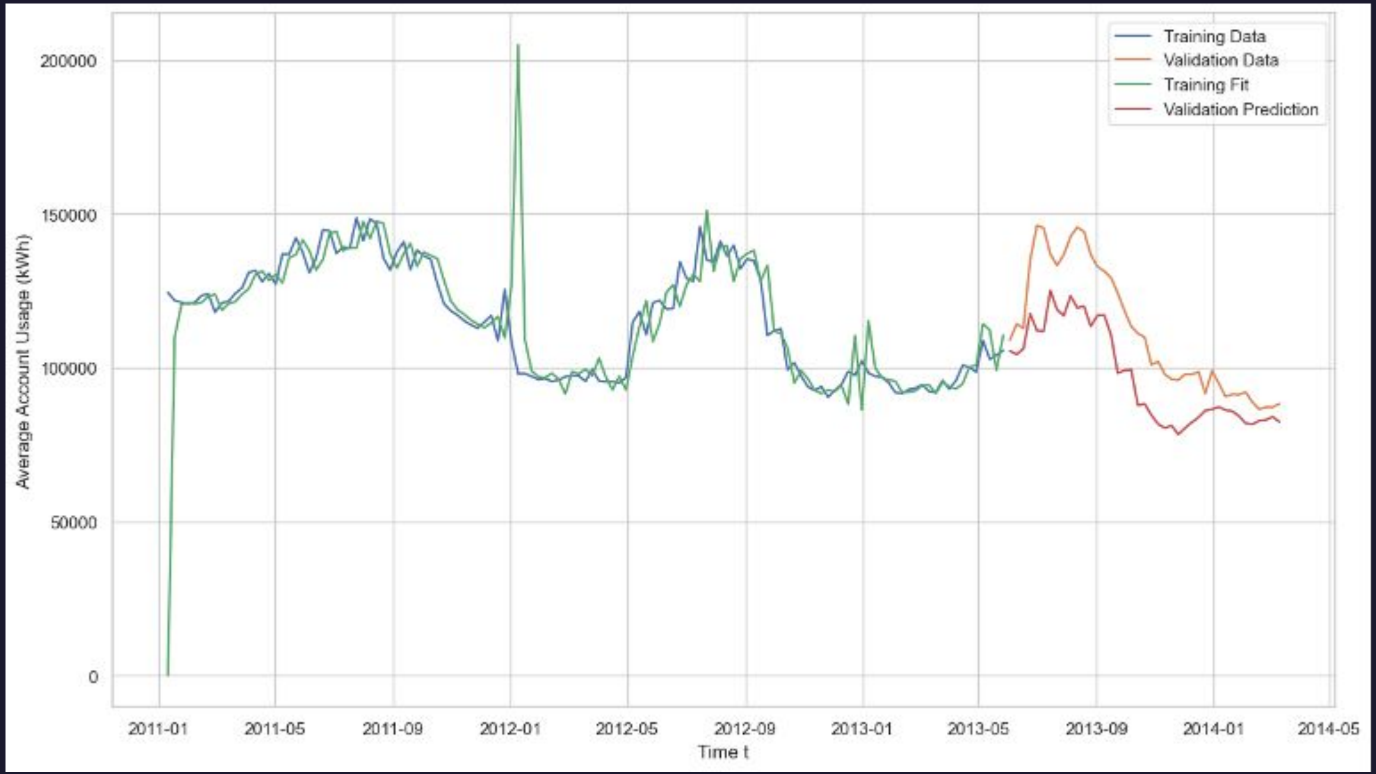
- Algorithmic Solution Design
  - Fit an initial model with SARIMA, Facebook Prophet
  - Evaluate MAPE on complete dataset and individual accounts
- Evaluation Metric
  - Mean Absolute Percentage Error (MAPE)
- Algorithmic Solution Finalization
  - Compare Predictions vs. Actual data to test the SARIMA model, Facebook Prophet, MAPE calculations by account
  - Report MAPE by 3 test regions for each model, compare the errors with box plots.
- Train-Validation-Test Split
  - Training Set: First 60% of Timeframe
  - Validation Set: Middle 20% of Timeframe
  - Test Set: Last 20% of Timeframe
  - Test set was divided into three equally sized regions




Results

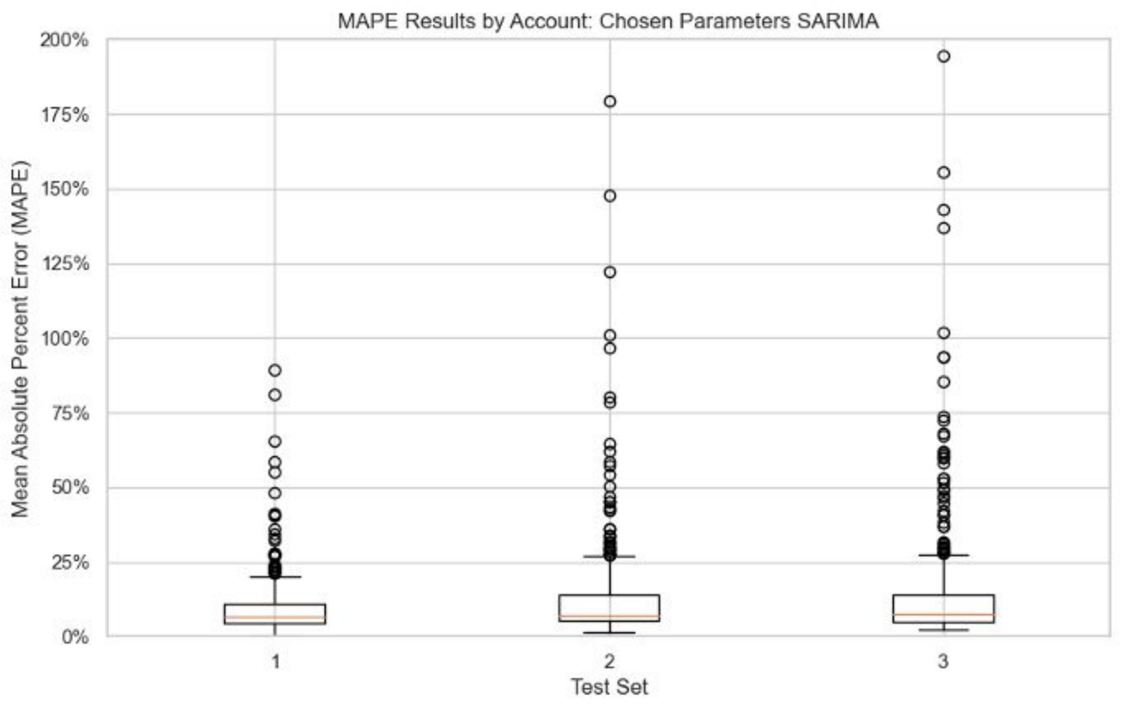
Results: SARIMA(0, 1, 1)(0, 1, 1)$_{52}$

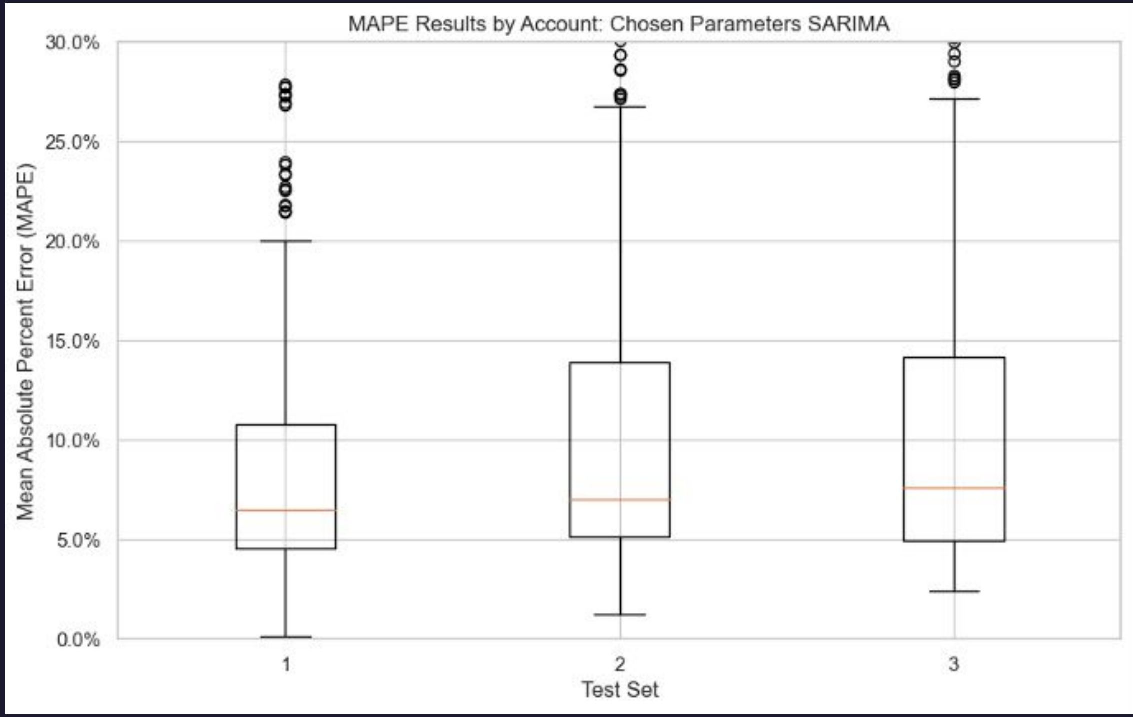
- Train-Validation-Test Split
  - Training Set: First 60% of Timeframe
  - Validation Set: Next 20% of Timeframe
  - Test Set: Last 20% of Timeframe
- Validation Set MAPE =12.8%



Results: SARIMA(0, 1, 1)(0, 1, 1)$_{52}$

- Test set was divided into three equal regions
- Median MAPE on Test Set:
- Region 1: 6.5%; Region 2: 7.0%; Region 3: 7.6%

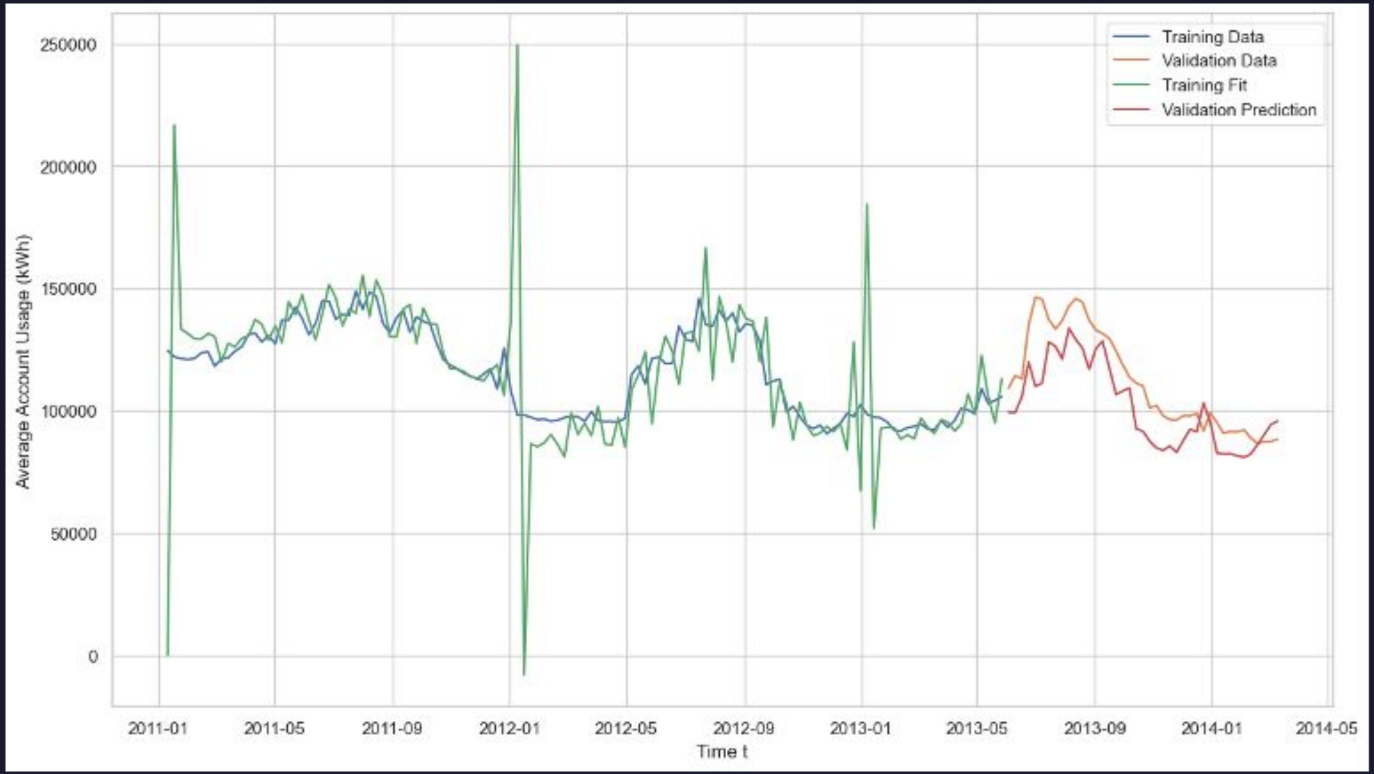




Results: Hyperparameter-Tuned SARIMA

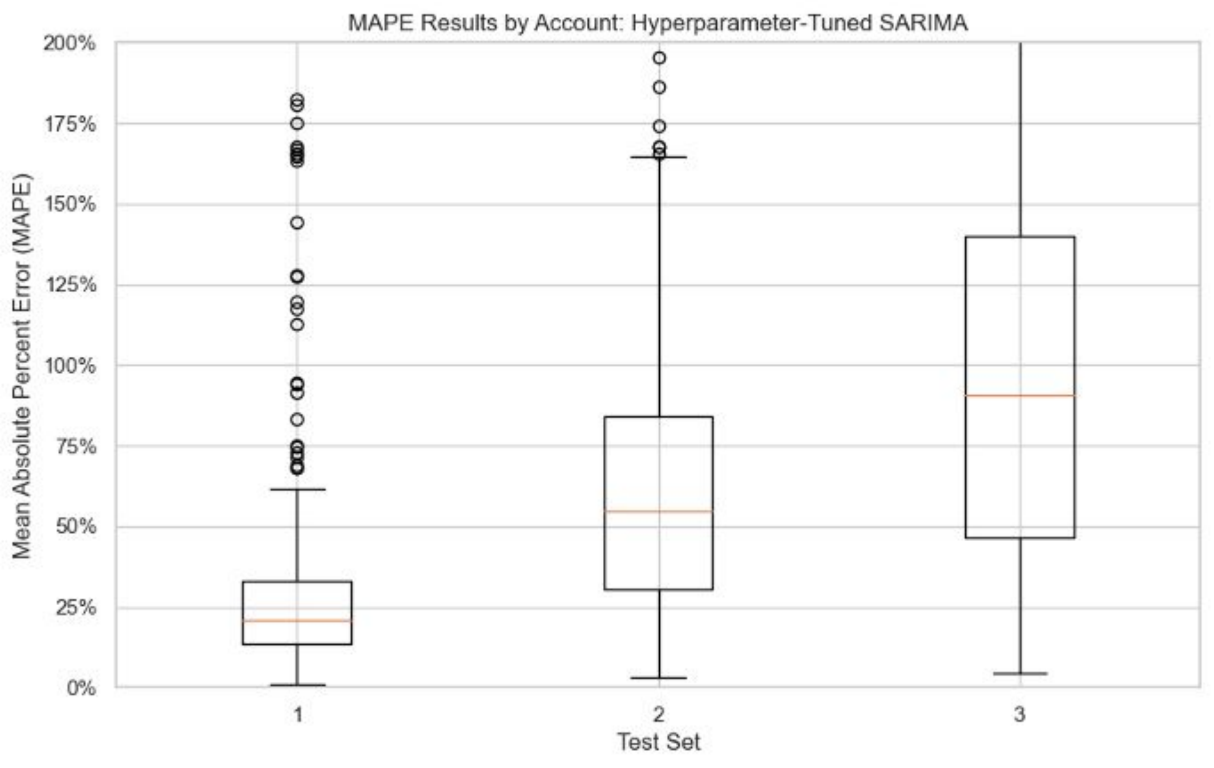
- Train-Validation-Test Split
  - Training Set: First 60% of Timeframe
  - Validation Set: Next 20% of Timeframe
  - Test Set: Last 20% of Timeframe
- Optimized Validation Set

  MAPE =10.4%



Results: Hyperparameter-Tuned SARIMA

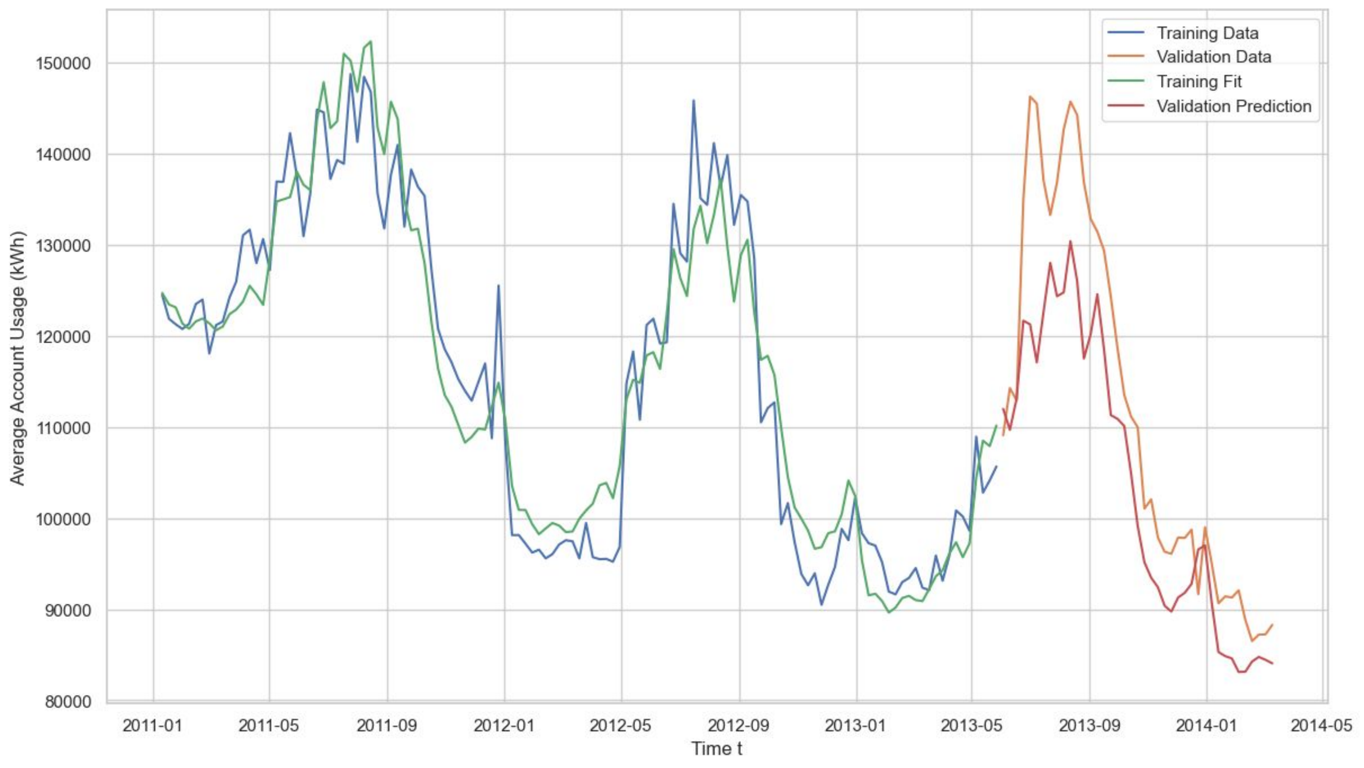
- Test set was divided into three equal regions
- Model: SARIMA(1, 2, 1)(1, 2, 1)$_{52}$
- Median MAPE on Test Set:
- Region 1: 21.1%
- Region 2: 54.8%
- Region 3: 90.6%



Results: FB Prophet - Chosen Parameters

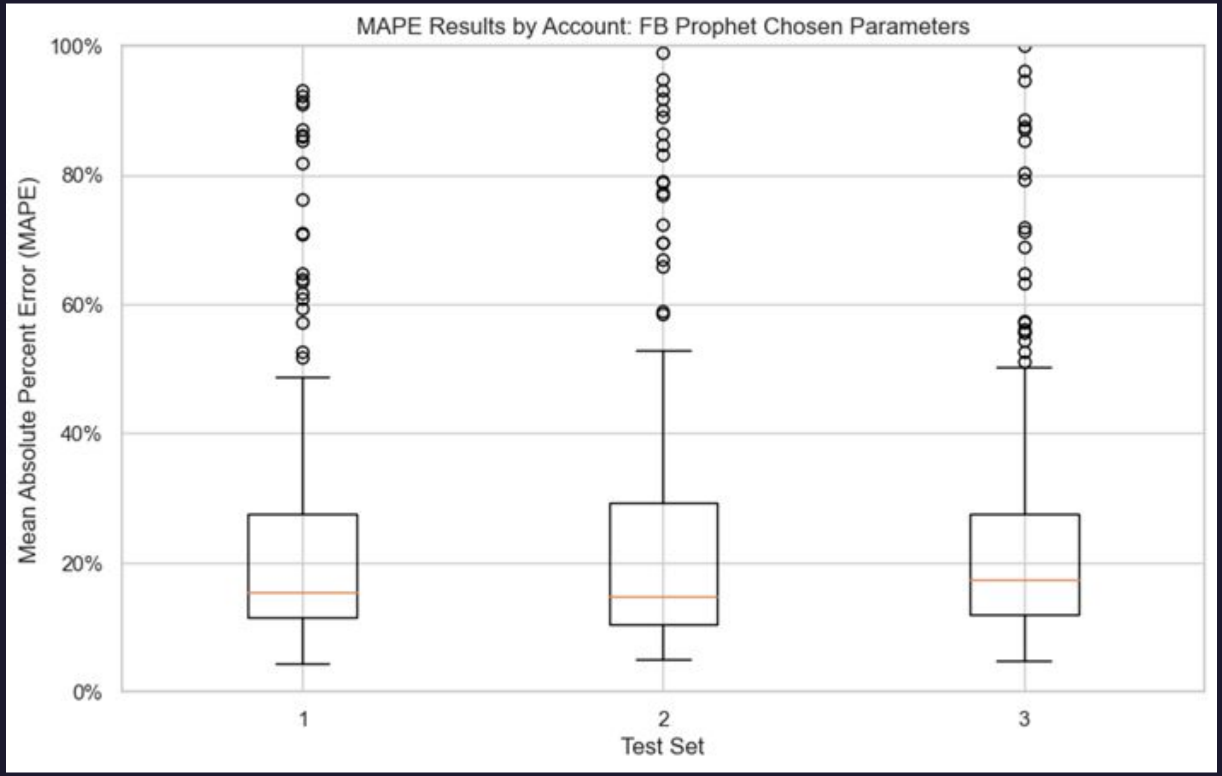
Model: Facebook Prophet
(period=365.25,
fourier_order=12,
monthly=False)

- Train-Validation-Test Split
  - Training Set: First 60% of Timeframe
  - Validation Set: Next 20% of Timeframe
  - Test Set: Last 20% of Timeframe
- Validation Set MAPE = 18.9%



Results: FB Prophet - Chosen Parameters

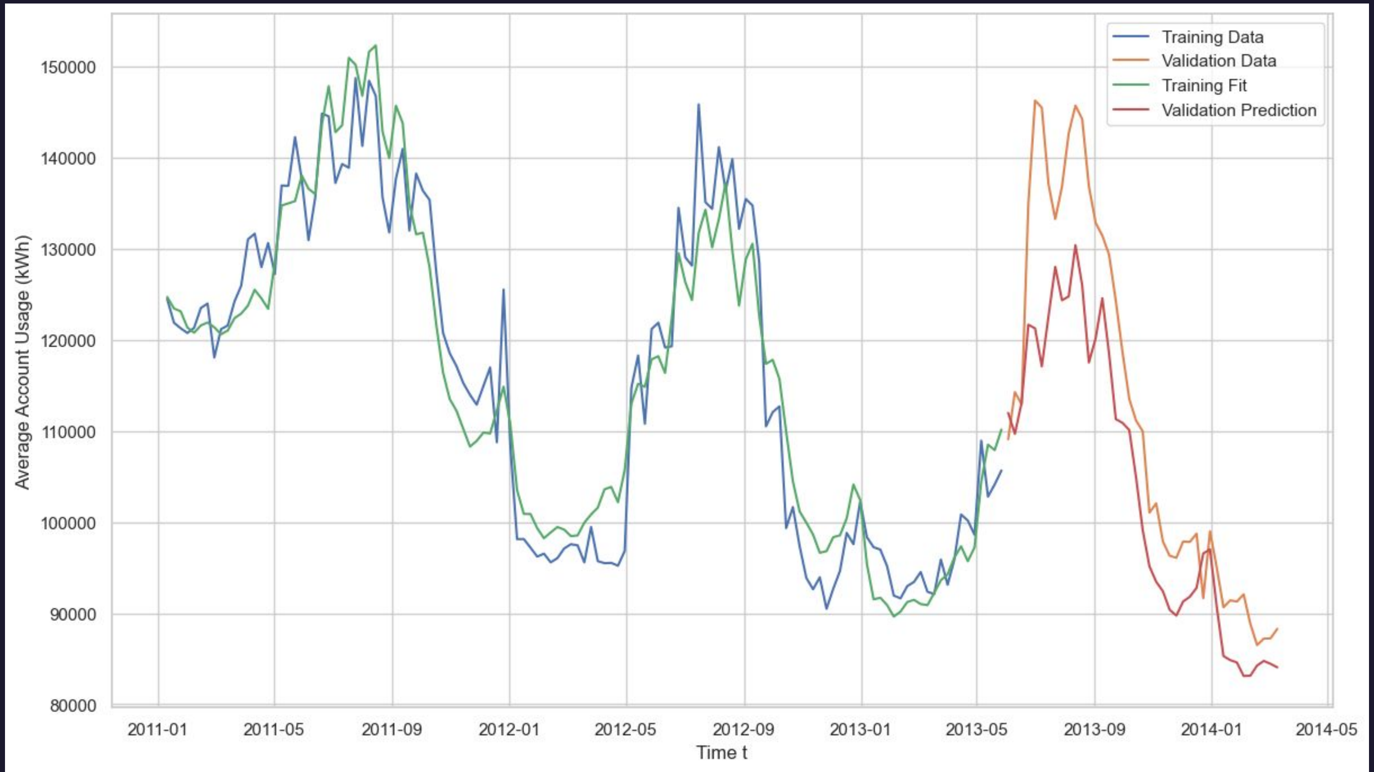
- Test set was divided into three equal regions
- Median MAPE on Test Set:
- Region 1: 15.4%
- Region 2: 14.8%
- Region 3: 17.4%



Results: Hyperparameter-Tuned FB Prophet

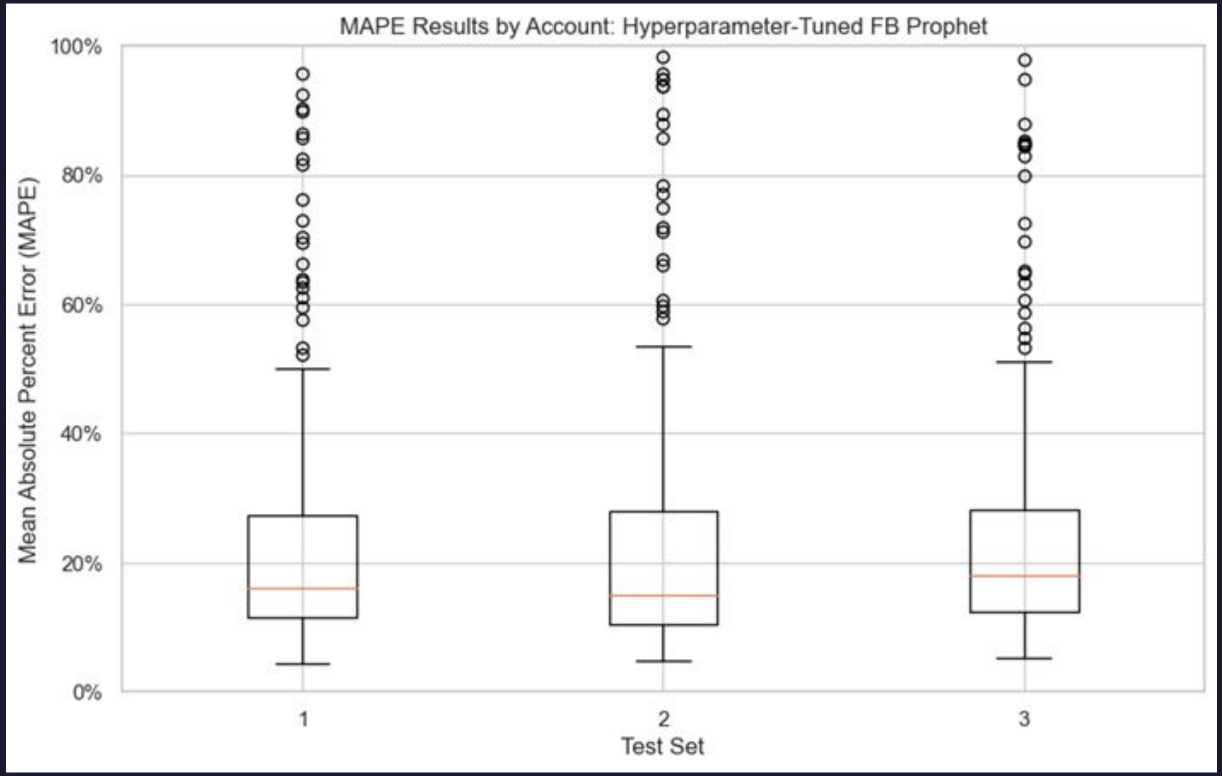
Model: Tuned FB Prophet

- Train-Validation-Test Split
  - Training Set: First 60% of Timeframe
  - Validation Set: Next 20% of Timeframe
  - Test Set: Last 20% of Timeframe
- Validation Set MAPE = 18.79%



Results: Hyperparameter-Tuned FB Prophet

- Test set was divided into three equal regions
- Model: Tuned Facebook Prophet
- Median MAPE on Test Set:
- Region 1: 16.02%
- Region 2: 14.91%
- Region 3: 18.11%



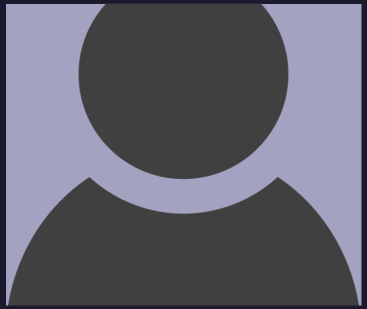
Results Summary

Model	Aggregated Data Cross-Validation MAPE	Test Set 1 MAPE	Test Set 2 MAPE	Test Set 3 MAPE
SARIMA $(0, 1, 1)x(0, 1, 1)_{52}$	12.8%	6.5%	7.0%	7.6%
SARIMA $(1, 2, 1)x(1, 2, 1)_{52}$	10.4%	21.1%	54.8%	90.6%
FB Prophet: Chosen Parameters	18.9%	15.4%	14.8%	17.4%
FB Prophet: Hyperparameters Tuned by Account	N/A	16.0%	14.9%	18.1%

Future Ideas

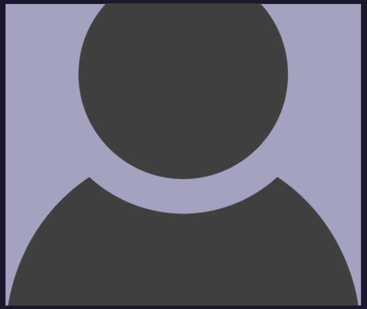
- Incorporation of Exogenous Variables to Transform SARIMA -> SARIMAX:
  - Since the SARIMA model performed the best on the test set, continuing with this model would be wise. Creation of time-based features such as holiday indicators, or other exogenous variables such as average monthly temperature, could help improve model performance.
- Other Models:
  - Trying other models on the data, including neural networks, could improve the performance of the model on the test set.

Team



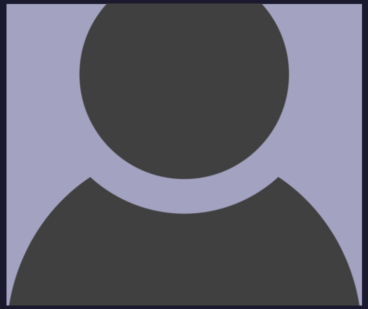
Kevin Taylor

MS Data Science Student



Nathaniel Ho

MS Data Science Student



Kelly Du

MS Data Science Student