Forecasting Project

Process Documentation

Deliverable 1

Kevin Taylor, Kelly Du, Nathaniel Ho

# Contents

[Introduction 1](#_Toc127296505)

[Data 2](#_Toc127296506)

[Data Extraction 2](#_Toc127296507)

[Data Processing 3](#_Toc127296508)

[Variables 4](#_Toc127296509)

[Pre-Modeling 5](#_Toc127296510)

[Trend/Seasonality 5](#_Toc127296511)

[Seasonal Decomposition 7](#_Toc127296512)

[Modeling 8](#_Toc127296513)

[Results 8](#_Toc127296514)

[All Accounts Combined 9](#_Toc127296515)

[Individual Accounts 9](#_Toc127296516)

[Team Information 11](#_Toc127296517)

# Figures

[Figure 1: Project Workflow 1](#_Toc127296536)

[Figure 2: Initial DataFrame Head 2](#_Toc127296537)

[Figure 3: Four Accounts’ Electricity Usage 3](#_Toc127296538)

[Figure 4: MAPE Calculation 5](#_Toc127296539)

[Figure 5: Train-Test Split 5](#_Toc127296540)

[Figure 6: Average Energy Use (kWh) - All Accounts 6](#_Toc127296541)

[Figure 7: PSD of Average Energy Use 6](#_Toc127296542)

[Figure 8: Weekly Average Use - Seasonal Decomposition 7](#_Toc127296543)

[Figure 9: All Accounts Combined Model Fit / Prediction 9](#_Toc127296544)

[Figure 10: MAPE Distribution - Individual Accounts 9](#_Toc127296545)

[Figure 11: MAPE Distribution – Best and Worst Case 10](#_Toc127296546)

[Figure 12: Team Information 11](#_Toc127296547)

# Introduction

There are many different algorithms can be developed for timeseries such as traditional Extreme Learning Machine (ELM) as well as other commonly used machine learning methods like Recurrent Neural Network (RNN), Linear Regression (LR), k-Smooth Regression (KSR), k-Nearest Neighbor Regression (kNNR), Gaussian Process Regression (GPR), and Generalized Regression Neural Network (GRNN). In this engagement, we are using SARIMAX from the *statsmodels* library to forecast future values in a timeseries as a first-pass model.

The purpose of this document is to provide a detailed technical overview of:

1. Data
2. Variables
3. Pre-Modeling
4. Modeling
5. Results

Diagram

Description automatically generated

Figure : Project Workflow

The goal of this project is to develop an accurate model for forecasting energy consumption. To do so, the initial process consisted of reading and pre-processing the data, identifying potential variables, and fitting and testing an initial model. For Deliverable 1, a SARIMA model was fit to the data for each user account and was tested.

A timeseries model can create value by predicting energy consumption outside the study period. To develop this model, cleaning the raw dataset was necessary, as was evaluating the accuracy of an initial forecasting model on the data.

# Data

The data was gathered from the archive under the title “LD2011\_2014.txt,” a text file delimited by semicolon. Decimal values were encoded with a comma ie) “0,00.” Values in the dataset were listed in kW power over 15-minute intervals. To convert values to kWh, the values were divided by 4. Missing values, including those for accounts created during the timeframe, are encoded in the data with zeros. The biannual time change results in either one hour of zero readings, or two hours aggregated, depending on the season (March time change vs. October time change).

The python *pandas* package was used to read the data delimited by “;” in to a DataFrame object, followed by some data preprocessing.

A screenshot of a computer

Description automatically generated with medium confidence

Figure : Initial DataFrame Head

## Data Extraction

Data extraction was over the entire dataset. The time period in the data is 2011-2014, but some accounts were created during the timeframe – these values are encoded as zeros. After initially processing the datetimes of the data, we are left with a dataset with the shape of (140256, 371), or 140,256 rows and 371 columns. There was one column to record the date/time, and all other columns were readings for 370 accounts.

Each row corresponds to a date and time electricity usage was recorded, while each column corresponds to the reading for that account.

**Data Diagnostics**

We checked the quality of the data, including the number of records, duplicates, and missing values. There were no duplicates or missing values to report. It is worth noting that, due to daylight savings time, there are values in March from 1:00am to 2:00am that are zeros, while in October there are values that are averaged between 1:00am to 2:00am. Since this action occurs periodically, we aggregated the data without special treatment for these records**.**

## Data Processing

To pre-process our data into a workable form, we first transformed the data into long pivot form. This helped transform the data so that we could group rows according to year and week. From there, we divided the values for each week by 4; this is because the data collected was in 15-minute intervals. Therefore, to transform a kilowatt power in 15-minute intervals to kilowatt-hours (kWh), dividing by 4 was necessary. We grouped the data by year, week, and account, then summed the values. The first four accounts are visualized below, created from the pre-processed DataFrame. These accounts were not registered for electricity until the year 2012, so there are zero values for the first year.

|  |  |
| --- | --- |
|  |  |
|  |  |

Figure : Four Accounts’ Electricity Usage

# Variables

Weekly usage of electricity is our target variable. It is denoted as the variable *value* within our model. For an individual account, this represents the account’s total usage in kWh for a particular week. To perform initial EDA we also researched the data’s trend/seasonality on the dataset aggregated over all accounts, by mean weekly usage. The group gained the following understanding about model variables:

|  |  |
| --- | --- |
| **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 |

The initial model fit for this deliverable, a SARIMA model, does not have any exogenous (predictive) variables. Rather, SARIMA components use target variables at different lags and weighted average forecast errors as predictor variables, as well as a seasonal component. In future iterations of this deliverable, additional models will use exogenous or derived variables to improve the model.

# Pre-Modeling

Pre-modeling was performed in two main workflows:

* Exploratory Data Analysis
  + Trend and seasonality in the data are identified for initial modeling. This exercise is performed on all accounts aggregated by week, and the takeaways and model assumptions are then applied to each individual account.
* Pre-Model Utility Functions
  + Utility functions are created for a train-test split, mean absolute percentage error (MAPE) calculation and walk-forward validation, important functions that can and will be applied to future models as well.

A picture containing text, clock

Description automatically generated

Figure : MAPE Calculation

Chart

Description automatically generated

Figure : Train-Test Split

## Trend/Seasonality

Using the utility functions created to assist the process, the time series and power spectral decomposition (PSD) indicated a first-order trend and a yearly seasonality. In fact, the maximum power in the PSD corresponded to a period of 51.75 weeks, rounded to 52 weeks or one year.

Chart

Description automatically generated

Figure : Average Energy Use (kWh) - All Accounts

Chart

Description automatically generated

Figure : PSD of Average Energy Use

## Seasonal Decomposition

The process of time series decomposition breaks down a series into its constituent parts: level, trend, seasonality, and noise. This decomposition provides a clear and comprehensive model for analyzing and understanding time series, which is essential for effective time series analysis and forecasting. In the following plot we think of the components as combining additively: y(t) = Level + Trend + Seasonality + Noise with period of 52.

Performing and visualizing a seasonal decomposition revealed two important points. First, the first-order trend indicates first-order differencing would be required in a model. Second, the seasonal component appears to have a yearly period. For initial model parameters, a SARIMA(0, 1, 1)x(0,1, 1)52 model was chosen.

Chart, line chart

Description automatically generated

Figure : Weekly Average Use - Seasonal Decomposition

# Modeling

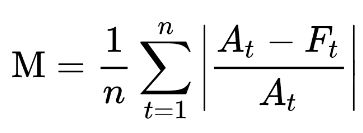
SARIMAX is an extension of ARIMA models that considers seasonality. In a SARIMAX model, the hyperparameters that need to be adjusted are the parameters used in regular ARIMA models (p,d,q) as well as additional parameters for seasonality (P,D,Q,s). Thus, we have a total of 7 parameters that need to be optimized.

According to the SARIMA formula (p,d,q)(P,D,Q,s), the parameters for these models are defined as follows:

|  |  |
| --- | --- |
| ***Variable*** | **Explanation** |
| *p and seasonal P* | Represents the number of autoregressive terms (the number of lags in the stationary series). |
| *d and seasonal D* | Represents the number of differences required to make the series stationary. |
| *q and seasonal Q* | Represents the number of moving average terms (the number of lags in the forecast errors). |
| *s* | Represents the length of the seasonal cycle in the data. |

**Evaluation Metric**

The Mean Absolute Percentage Error (MAPE) is a commonly used metric to evaluate the performance of a SARIMA model. This metric is calculated by taking the absolute difference between each predicted data point and its corresponding test point, dividing that value by the test point, and then averaging all absolute percentage differences to give the MAPE.



# Results

To evaluate the model effectiveness, we fit a SARIMA model to the data aggregated over all accounts and calculated the MAPE between the predicted values and the actual data. The test set was created from the latter 20% of the data and the training set from the first 80% of the data. After fitting the model to the aggregated data, we performed the same modeling for each account. For each account, the test set was created from the latter 20% of the data and the training set from the first 80% of the data.

## All Accounts Combined

Fitting a SARIMA(0, 1, 1)x(0,1, 1)52 resulted in a test MAPE of **4.5%**.

Chart, line chart

Description automatically generated

Figure : All Accounts Combined Model Fit / Prediction

## Individual Accounts

Performing the same process at the individual account level resulted in many different models, with many different MAPE values. A histogram was developed for the 370 individual accounts’ MAPE calculated on the test set (last 20% of the data).

Chart, histogram

Description automatically generated

Figure : MAPE Distribution - Individual Accounts

Test set MAPE varied greatly between the individual accounts. The SARIMA model performed very well for some accounts and poorly for others.

|  |
| --- |
| Account MT\_146 MAPE = 2.5%  Account MT\_146 MAPE = 2.5% |
| Account MT\_127 MAPE = 1700% - Skewed due to actual values close to zero, but not equal to zero |

Figure : MAPE Distribution – Best and Worst Case

The Total MAPE can be used to aggregate and quantify error for each account, defined as:

Using this MAPE formula, the total test set MAPE was **15.6%**. However, this value is likely skewed by actual values that are very close to zero, resulting in large prediction error percentages. If we instead use the median in stead of the mean, median MAPE was **6.5%**.

# Team Information

|  |  |  |
| --- | --- | --- |
|  |  |  |
| **Kevin Taylor** | **Nathaniel Ho** | **Kelly Du** |
| MS Data Science Student | MS Data Science Student | MS Data Science Student |

Figure 12: Team Information