From November 2011 through February 2014, the City of London gathered data on household data usage through smart meters. Homes were categorized based on income: Affluent, Comfortable and Adversity. The data was collected on an hourly basis and summarized by day. The data is available at <https://www.kaggle.com/jeanmidev/smart-meters-in-london> along with weather data for the same time period in London.

Part of the rationale for distributing the smart meters is to reduce overall consumption. If people understand how and when they are spending, they will be more aware of how their choices impact their overall energy bill, and will be able to conserve energy through actions such as purchasing energy efficient appliances and using less energy overall. By training a model that includes temperature impact on energy usage over the selected period, the model can serve as a baseline. The impact of energy saving can be calculated by taking the difference between what is forecast for a period using this model and what happened. Since the unit is kilowatts, I can take the average price per kilowatt and multiply that the series of differences. I can take the sum of that series to calculate the overall savings.

To create the forecasts, I compared a few different techniques. To begin I created a new data frame grouped by date, calculating the average consumption per household for the total population as well as each of the income groups. I took the raw weather data and calculated the average temperature per day. I joined the tables on day and plotted each of the income groups along with the total population and the average temperature. From the plots, it was clear that each group was distinct; unsurprisingly the amount of energy used increased along with the income bracket. The plots showed seasonal cycles with high energy usage in winter and low energy usage in summer. For the small amount of missing data, I used forward filling with the rationale that energy usage on a given day would be similar to the prior day.

I tested for stationarity and found that none of the time series are considered stationary without differencing. Since the data showed an obvious yearly trend, I decided to difference the data by 365 days, and re-tested stationarity. After differencing, all but one of the series were considered stationary at an alpha of 0.05. The one series that was not, was stationary at an alpha of 0.10 so I decided to move forward with the analysis.

Using the differenced data, I decided to compare a naïve model, which simply calculated a forecast based on a one-year shift to a Seasonal Autoregressive Integrated Moving Averages with Exogenous Variables (SARIMAX) model, to a Seasonal Autoregressive Integrated Moving Averages (SARIMA) with the seasonal component but without the exogenous variable, and to an Autoregressive Integrated Moving Averages (ARIMAX) model with no seasonal component and with the exogenous variable. The SARIMAX and SARIMA models required parameter tuning for the values of p, the number of lags in the autoregression, d, the number of differences taken, and q, the number of elements in calculating a moving average, as well as P, D and Q, the same types of parameters for the seasonal aspect. The ARIMA model required tuning for p, d and q. To tune these parameters, I ran a series of grid searches. I ended up running grid searches for SARIMAX, SARIMA and ARIMAX for the total population as well as each income group.

I split the data by taking 70% of the data in the original series as the train set and the remaining 30% as the test set. For each group I fit the models. Since I trained and forecast on the difference from the 365 days prior, I had to un-difference the data by adding the forecast difference to data point 365 days prior. Then I calculated the Mean Average Error (MAE) for the naïve, the SARIMAX, the SARIMA, and the ARIMA forecasts.

In all cases, in terms of MAE, the SARIMAX outperformed the naïve forecast. In two instances, the ARIMA forecast was just slightly better than SARIMAX. In two other instances, it far underperformed SARIMAX. In all cases the SARIMA model underperformed SARIMAX, and in most cases it was close to the naïve forecast. This suggests that overall, the SARIMAX with both the seasonal component, and incorporating the exogenous variable of temperature, is the best choice.

Finally, to estimate cost savings, I created a function that takes several arguments: the average cost of kilowatt, along with time series of usage and temperatures before and after a usage change implementation (such as new appliances or an effort to conserve). The function trains a model on the time series before implementation. This creates a baseline for the expected amount energy spent given the set of average daily temperatures. For the period after implementation the function subtracts the actual from the predicted for each day in the period, multiplies each difference by the average price of kilowatts, and then sums the total difference, which equals the total cost savings. I created a simulated dataset showing energy savings to see how much money would be saved given a 25% reduction in energy used. With an average cost of 0.15/kilowatt, over the course of one year, an affluent user could save an estimated 182.42, a comfortable user could save an estimated 58.61 and an adversity user could save 81.02.

The hope is that this information can empower consumers to understand their usage habits, will inspire change, saving money and decreasing greenhouse gas emissions. Sharing visualizations with actual monetary value can help drive that story and enable the city to meet its goals.