2023

APPLIANCES ENERGY PREDICTION

DATS6313 – TIME SERIES MODELING & ANALYSIS FINAL TERM PROJECT

ERIKA PHAM

TABLE OF CONTENT

TABLE OF FIGURES AND TABLES

ABSTRACT

This project explores the prediction of energy usage by household appliances using time series modeling and analysis. Utilizing the UCI Machine Learning Repository's Appliance Energy Prediction dataset, this project examines energy consumption data from a low-energy building, including variables like temperature, humidity, and weather conditions. Various models, including Holt-Winters, multiple linear regression, and ARIMA, were implemented and tested for effectiveness. Average prediction model, one of the base model, performed the best; which suggests much room for improvement, including reassessment of which features to include.

1 - Introduction.

The goal of this project is to fulfill the course's term project requirements as well as a cumulative "exam" on everything we have learned this semester. All tools and techniques are utilized and showcased within this report.

Specifically, we will be building a model to forecast the amount of energy that household appliances use, according to different temperature, humidity and weather conditions. The report will go through each step of time series modeling process, from data processing, to model selection, to forecasting and model performance assessment.

2- Description of the dataset

Overview

The dataset chosen for this project is Appliance Energy Prediction from UCI Machine Learning library (https://archive.ics.uci.edu/dataset/374/appliances+energy+prediction).

This is experimental data of appliances energy use in a low energy building. The data set is sampled every 10 min for about 4.5 months, starting 2016-01-11 at 17:00:00. Weather from the nearest airport weather station (Chievres Airport, Belgium) was downloaded from a public data set merged together with the experimental data sets using the date and time column. Two random variables have been included in the data set for testing the regression models and to filter out non predictive attributes (parameters).

There are 19735 entries. The forecast variable is "Appliances", which refers to energy use in Wh. The dataset is sampled at regular ten-minute intervals, ensuring uniformity in the time series data and making it well-suited for time series modeling. Table 1 includes the data's 27 continuous numerical features.

Table 1: List of Independent Variables

Index	Variable Name	Variable Description	Unit of Measurement
1	lights	Energy use of light fixtures in the house	Wh
2	T1	Temperature in kitchen area	Celsius
3	RH_1	Humidity in kitchen area	%
4	T2	Temperature in living room area	Celsius
5	RH_2	Humidity in living room area	%
6	T3	Temperature in laundry room area	Celsius
7	RH_3	Humidity in laundry room area	%
8	T4	Temperature in office room	Celsius
9	RH_4	Humidity in office room	%
10	T5	Temperature in bathroom	Celsius
11	RH_5	Humidity in bathroom	%
12	T6	Temperature outside the building (north side)	Celsius
13	RH_6	Humidity outside the building (north side)	%
14	T7	Temperature in ironing room	Celsius
15	RH_7	Humidity in ironing room	%
16	T8	Temperature in teenager room 2	Celsius
17	RH_8	Humidity in teenager room 2	%
18	T9	Temperature in parents room	Celsius
19	RH_9	Humidity in parents room	%
20	To	Temperature outside (from Chievres weather station)	Celsius
21	Press_mm_hg	Pressure (from Chievres weather station)	mm Hg
22	RH_out	Humidity outside (from Chievres weather station)	%
23	Wind speed	Wind speed (from Chievres weather station)	m/s
24	Visibility	Visibility (from Chievres weather station)	km
25	Tdewpoint	Dewpoint temperature (from Chievres weather station)	Celsius
26	rv1	Random variable 1	nondimensional
27	rv2	Random variable 2	nondimensional

It also includes “date”, which is the timestamp for each entry in the “year-month-day hour:minute:second” format.

Preprocessing

Data was cleaned for missing observations (there were none). The 'date' column was passed through Pandas 'to_datetime' to make sure it is in uniform datetime format. No dummy-encoding or up/down sampling needed for this data set.

Figure 1. Plot of Dependent Variable versus Time

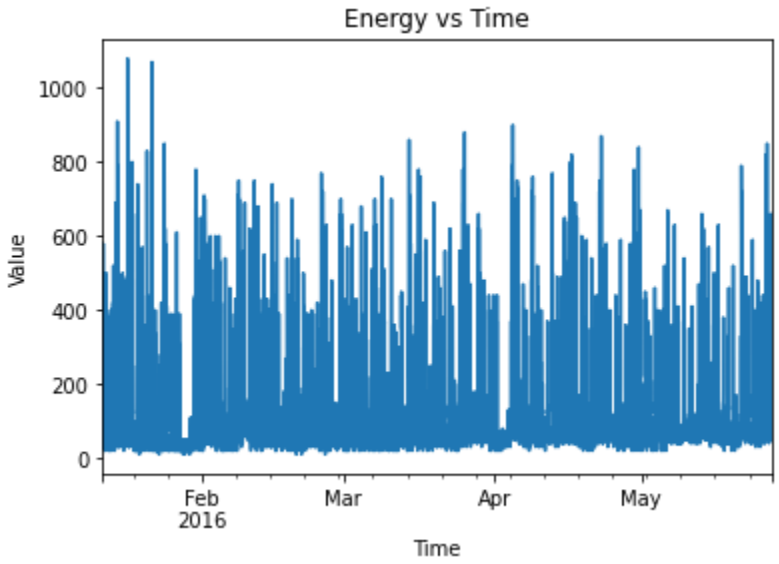
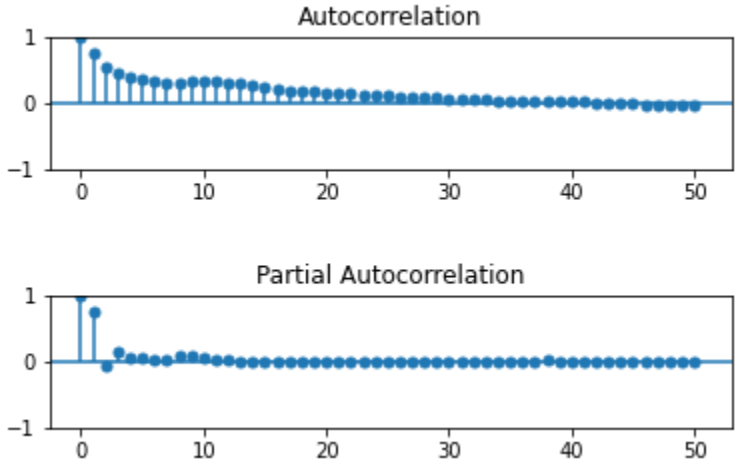


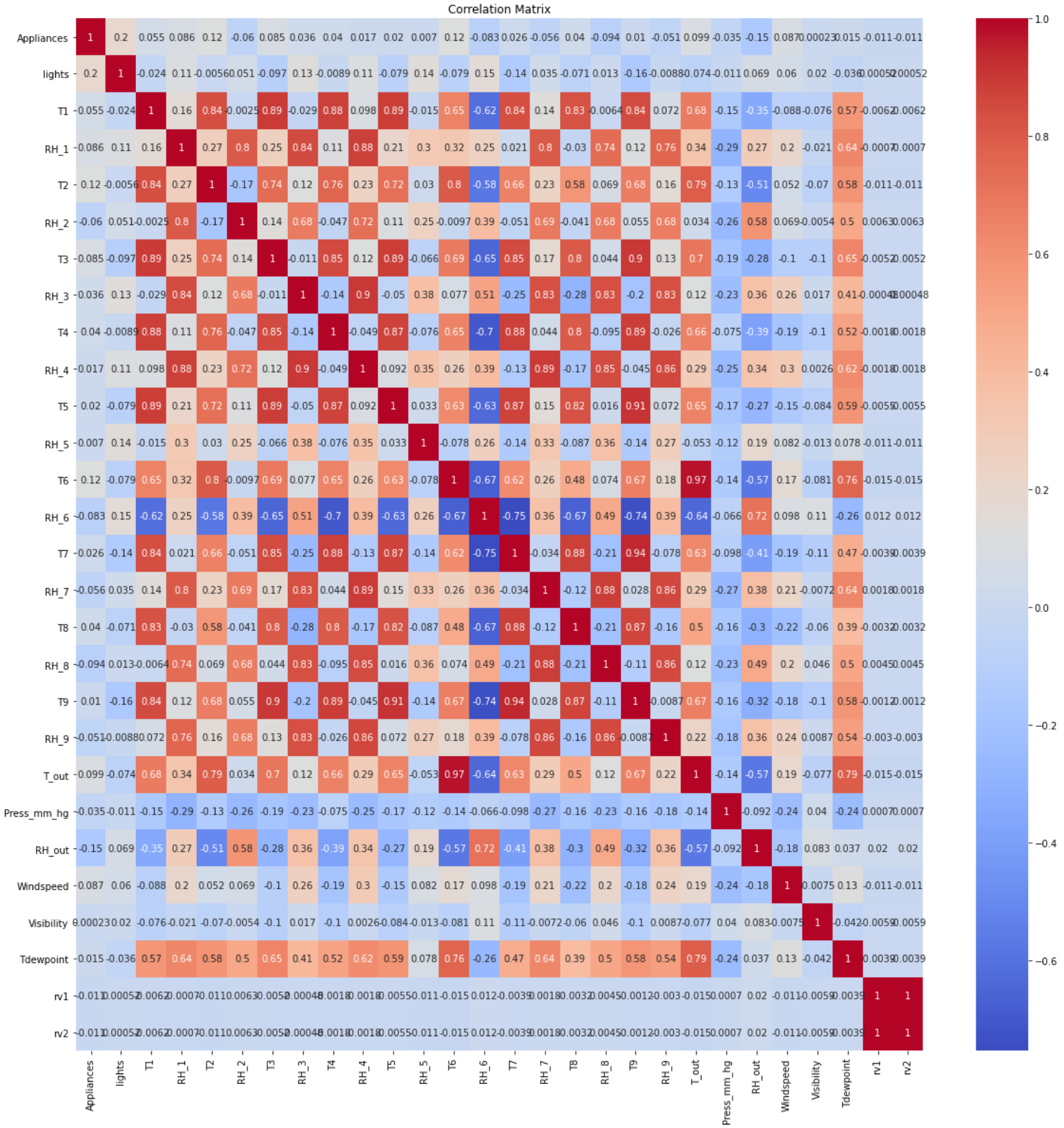
Figure 1 shows the energy consumed throughout the entire measured period of time. It seems that energy usage varies greatly, with no clear trend or seasonal pattern.

Figure 2. ACF & PACF Plot of the Data



We see a tails-off/cuts-off pattern, which suggests this could be an Auto Regressive (AR) model.

Figure 3. Correlation Matrix



I will note all coefficients that have a correlation coefficient >= 0.9. RH_3 (humidity in laundry room) is negatively correlated with RH_4 (humidity in office room). For indoors temperature, T3 (laundry room), T5 (bathroom), T7 (ironing room) are all negatively correlated with T9 (parents' room). For outdoors temperature, T6 (outside the building, north side) is extremely highly correlated with T_out (Temperature outside), with a coefficient of 0.97. Visibility has an extremely small positive correlation with energy consumption; very close to 0. The random variables rv1 and rv2 have no significant correlation to any variables.

The data was then split into train/test set, with a ratio of 80/20.

```
Training set size: 15788
Test set size: 3947
```

3- Stationarity Check:

A crucial step in performing Time Series modeling is making sure the data is stationary. That includes doing ACF/PACF analysis and using statistical tests like ADF and KPSS.

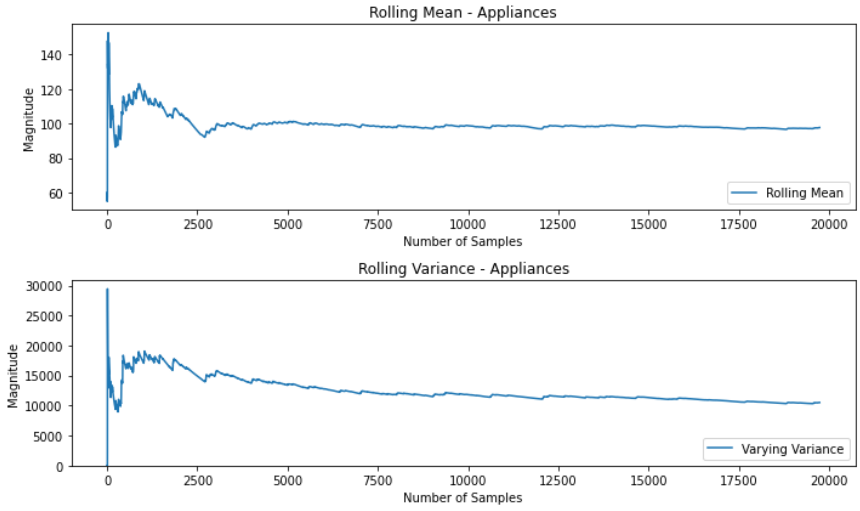


Figure 4. Plot of Rolling Mean and Variance of Data

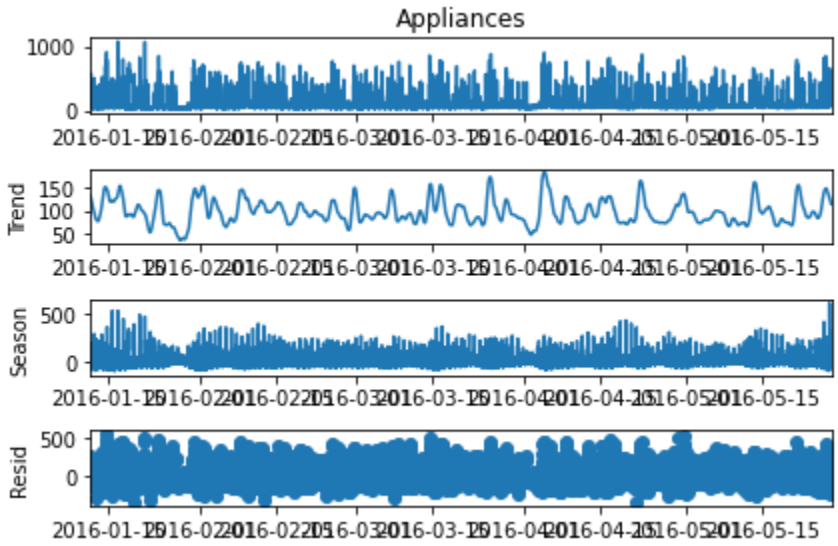
```
ADF Statistic: -21.616378
p-value: 0.000000
Critical Values:
        1%: -3.431
        5%: -2.862
        10%: -2.567
None
Results of KPSS Test:
Test Statistic           0.036599
p-value                  0.100000
Lags Used               74.000000
Critical Value (10%)     0.347000
Critical Value (5%)      0.463000
Critical Value (2.5%)    0.574000
Critical Value (1%)      0.739000
```

Figure 5. ADF and KPSS Test Results

The plot of rolling mean and variance for the dependent variable is stable once all samples are included. The p-value for ADF-test is 0.00 < 0.05, and KPSS is 0.1 > 0.05; so we accept that the data is stationary. No transformation is needed.

4- Time Series Decomposition

Figure 6. STL Decomposition Plot and Strength Measurement




```
Strength of trend for the raw data is 17.526%
Strength of seasonality for the raw data is 49.983%
```

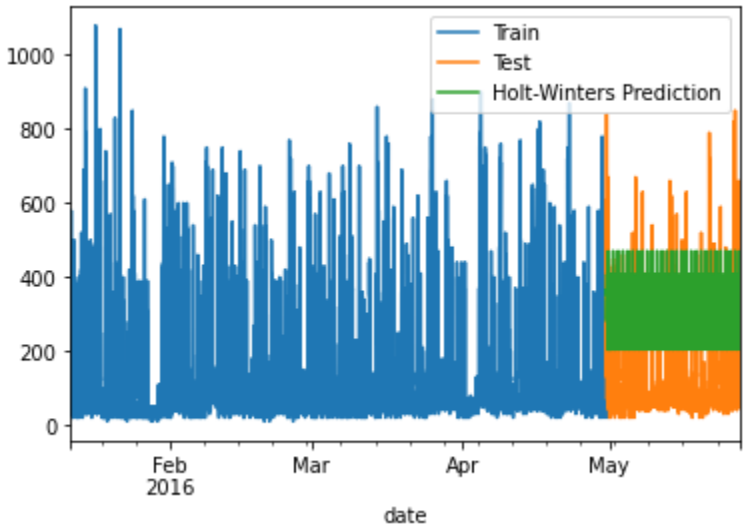
After decomposition using the STL method, we can see that there are not very strong trends or seasonal patterns throughout the data. It is likely that we do not need to account for seasonality when modeling.

5- Holt-Winters Method

The Holt-Winters method, also known as the Triple Exponential Smoothing method, is a time series forecasting technique used for data with seasonal patterns. It extends the Exponential Smoothing approach by adding seasonality components to the model.

For this project, I used additive methods. The seasonal period is set to the sampling periods, 144, as there are 144 10-minute intervals per day.

Figure 7. Holt-Winters Method Prediction



The RMSE for Holt-Winters is 204.71, which we will see that it performed worse than base models. Its poor performance is understandable as the data shows no strong trends or seasonality.

6- Feature Selection

PCA, SVD, Condition Number

After standardizing the data, I applied PCA, SVD decomposition and calculated the condition number.

Figure 8 shows the graph for the cumulative explained variance by each principal component. We see that at around 20 features, the ratio of explained variance is very close to 1, meaning we do not need to include all variables. The graph of singular values in Figure 9 also shows that we could remove some variables as it shows a sharp dip at the end (could be due to the 2 random variables).

Figure 8. PCA Cumulative Explained Variance

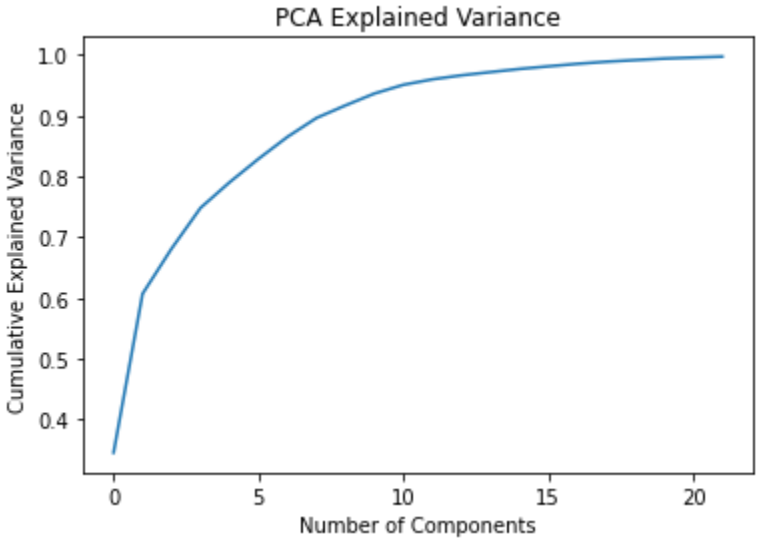
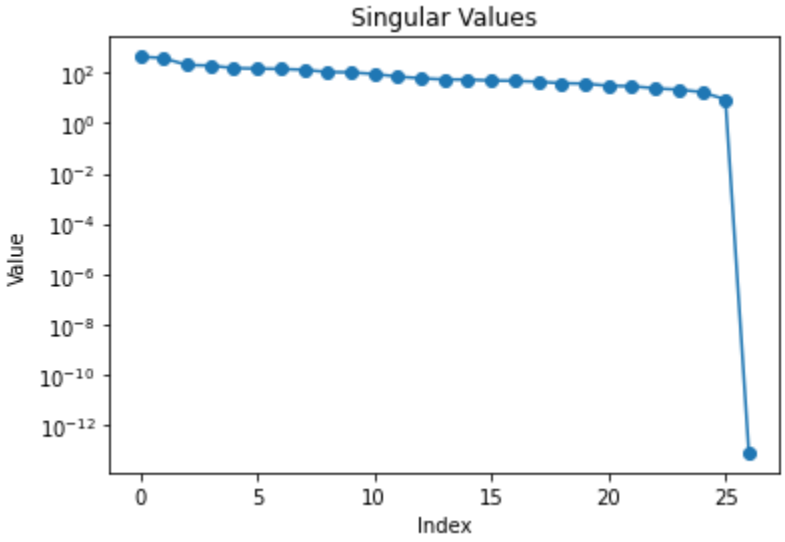
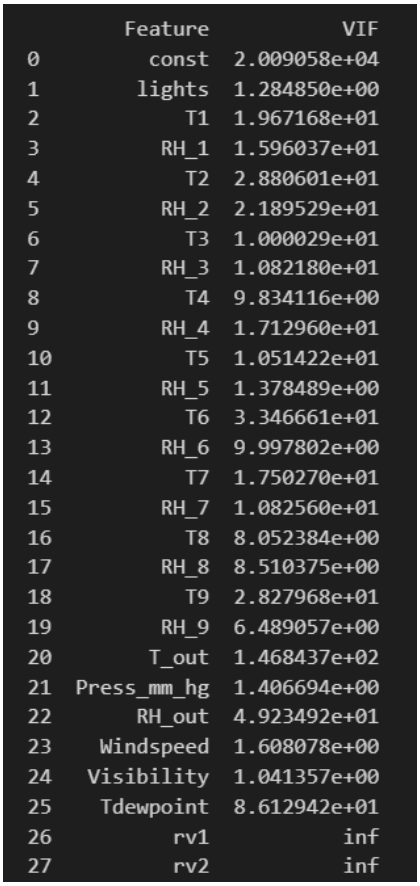


Figure 9. Singular Values



VIF Analysis

Figure 10. VIF Table



	Feature	VIF
0	const	2.009058e+04
1	lights	1.284850e+00
2	T1	1.967168e+01
3	RH_1	1.596037e+01
4	T2	2.880601e+01
5	RH_2	2.189529e+01
6	T3	1.000029e+01
7	RH_3	1.082180e+01
8	T4	9.834116e+00
9	RH_4	1.712960e+01
10	T5	1.051422e+01
11	RH_5	1.378489e+00
12	T6	3.346661e+01
13	RH_6	9.997802e+00
14	T7	1.750270e+01
15	RH_7	1.082560e+01
16	T8	8.052384e+00
17	RH_8	8.510375e+00
18	T9	2.827968e+01
19	RH_9	6.489057e+00
20	T_out	1.468437e+02
21	Press_mm_hg	1.406694e+00
22	RH_out	4.923492e+01
23	Windspeed	1.608078e+00
24	Visibility	1.041357e+00
25	Tdewpoint	8.612942e+01
26	rv1	inf
27	rv2	inf

VIF > 10 (high multicollinearity): T1, RH_1, T2, RH_2, T3, RH_3, RH_4, T5, T6, T7, T9, T_out, RH_out, and Tdewpoint. Rv1 and rv2 have infinite VIF, which suggests near-perfect correlation with other variables. All of these are worth considering to remove when reducing dimensionality.

Backward Stepwise Selection

Using alpha = 0.05, each feature with p-value greater than or equal to alpha is removed. 19 features were kept, with 'T5', 'RH_4', 'T1', 'rv2', 'rv1', 'Press_mm_hg', 'T7', 'RH_5' eliminated. The list of selected features is: 'lights', 'RH_1', 'T2', 'RH_2', 'T3', 'RH_3', 'T4', 'T6', 'RH_6', 'RH_7', 'T8', 'RH_8', 'T9', 'RH_9', 'T_out', 'RH_out', 'Windspeed', 'Visibility', 'Tdewpoint'. We will use this list of features going forward with modeling.

7- Base Models

We will utilize Naïve, Average, Drift, Simple Smoothing and Exponential Smoothing for h-step predictions. Table 2 Shows the models' performance as measured by RMSE.

Table 2. Base Models' RMSE

Model	RMSE
Naïve	223.05
Average	91.05
Drift	250.96
Simple Smoothing	104.14
Exponential Smoothing	262.90

Average model performed the best, with the lowest RMSE by far of 91.05. Exponential Smoothing was the worst-performing.

8- Multiple Linear Regression

The features included were chosen through backwards stepwise selection. Table 3 shows the OLS results. Figure 8 shows the predictions against the test set.

Table 3. OLS results

OLS Regression Results

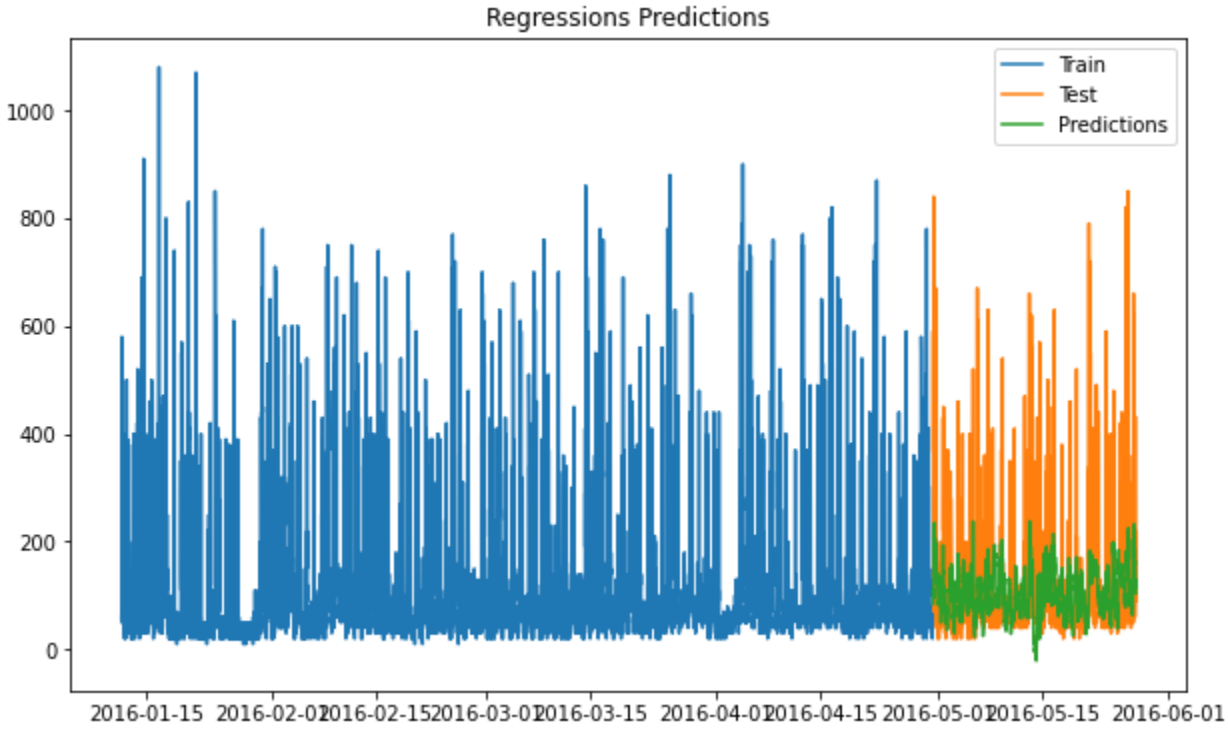
Dep. Variable:	Appliances	R-squared:	0.172
Model:	OLS	Adj. R-squared:	0.171
Method:	Least Squares	F-statistic:	171.9
Date:	Mon, 18 Dec 2023	Prob (F-statistic):	0.00
Time:	23:12:04	Log-Likelihood:	-94422.
No. Observations:	15788	AIC:	1.889e+05
Df Residuals:	15768	BIC:	1.890e+05
Df Model:	19		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
const	108.6740	56.341	1.929	0.054	-1.760	219.108
lights	2.0670	0.103	20.092	0.000	1.865	2.269
RH_1	13.7721	0.653	21.100	0.000	12.493	15.052
T2	-19.7939	1.292	-15.324	0.000	-22.326	-17.262

RH_2	-14.4672	0.759	-19.063	0.000	-15.955	-12.980
T3	25.9798	1.086	23.920	0.000	23.851	28.109
RH_3	8.2948	0.816	10.159	0.000	6.694	9.895
T4	-2.6232	0.988	-2.655	0.008	-4.560	-0.687
T6	8.0874	0.782	10.345	0.000	6.555	9.620
RH_6	0.2965	0.079	3.753	0.000	0.142	0.451
RH_7	-0.9051	0.431	-2.101	0.036	-1.750	-0.061
T8	8.0768	0.908	8.894	0.000	6.297	9.857
RH_8	-6.1427	0.388	-15.832	0.000	-6.903	-5.382
T9	-13.5957	1.685	-8.069	0.000	-16.898	-10.293
RH_9	-0.9845	0.440	-2.237	0.025	-1.847	-0.122
T_out	-9.1916	2.733	-3.363	0.001	-14.548	-3.835
RH_out	-0.5930	0.515	-1.151	0.250	-1.603	0.417
Windspeed	0.9848	0.378	2.603	0.009	0.243	1.726
Visibility	0.1921	0.063	3.061	0.002	0.069	0.315
Tdewpoint	3.5533	2.695	1.318	0.187	-1.730	8.836

Omnibus:	11050.936	Durbin-Watson:	0.622
Prob(Omnibus):	0.000	Jarque-Bera (JB):	160744.270
Skew:	3.284	Prob(JB):	0.00
Kurtosis:	17.185	Cond. No.	1.16e+04

Figure 11. OLS Predictions



Without cross-validation, the R-squared is 0.17 and RMSE is 85.92. After 5-fold time series cross-validation, the average r-squared is 0.07 and RMSE is 94.3. It is one of the better performing models in this project.

Analyzing residuals below, we see that the ACF/PACF plots do not show the pattern of white noise (figure 12). Q-value comparison as well as variance and mean of residual also support that the residuals for the model are not white noise. This model needs adjusting in order to perform better, perhaps choosing a different set of features.

Figure 12. ACF/PACF Plot of OLS Residuals

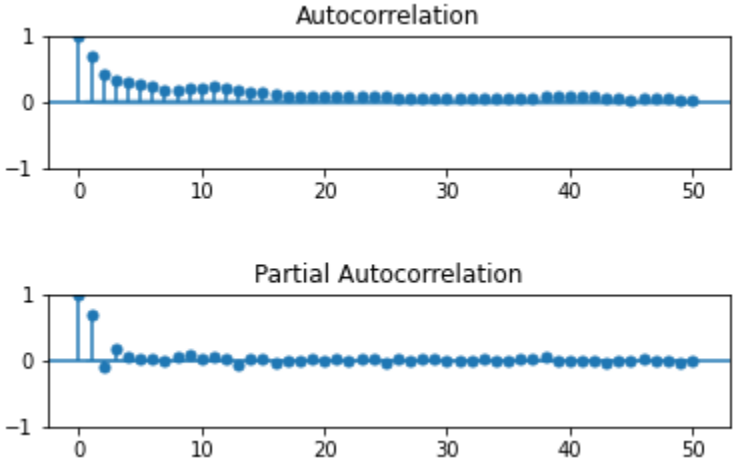


Figure 13. OLS Residuals Q-Value, Variance and Mean

```
Q =  72.42362346881491
Q* =  43.77297182574219
The residual is NOT white
Variance of Residuals: 7358.844410692727, Mean of Residuals: -4.896044324404505
```

9- ARMA/ARIMA/SARIMA/Multiplicative Model

Preliminary Order Determination

Figure 14 shows a GPAC table that highlights a column of constant at k=1 and j=0. Other less clear patterns include k = 8 and j = 10. Figure 15 shows the ACF/PACF plot with tails-off/cuts-off (at lag 1) pattern. This highly indicates an AR(1) model and will be the chosen model for experiment going forward.

Figure 14. GPAC Table

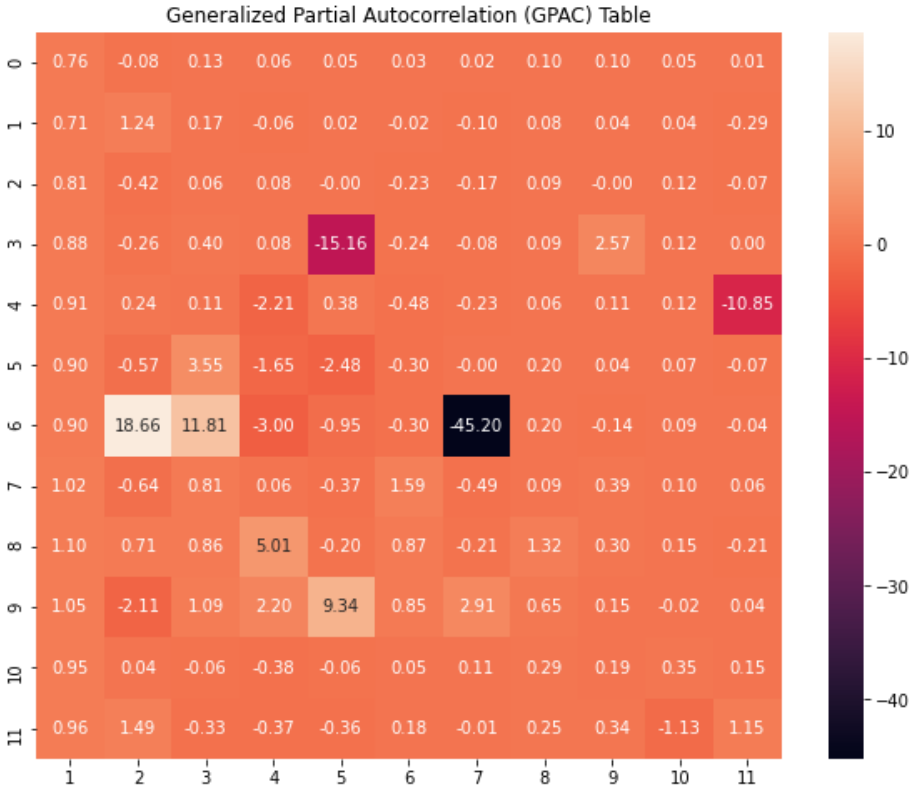
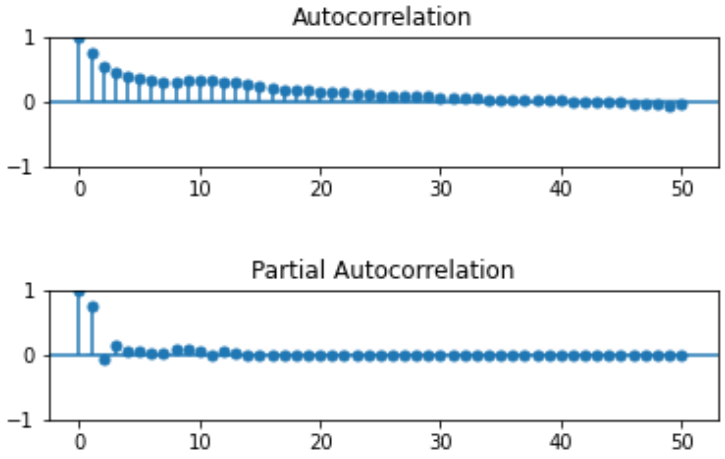


Figure 15. ACF/PACF for ARMA(1, 0) Residuals



Parameter Estimation

Since we want an AR(1) model with no seasonal components, I fit an ARIMA(1, 0, 0) model in sm.tsa.ARIMA() to find the coefficients for the parameters. Table 4 shows the results, which makes the model ARIMA(1,0,0): y(t) = 98.0245 + 0.7569y(t-1) + e(t). All coefficients are significant with very small p-values.

Table 4. ARIMA(1, 0, 0) Results

SARIMAX Results

Dep. Variable:	Appliances	No. Observations:	15788
Model:	ARIMA(1, 0, 0)	Log Likelihood	-89194.473
Date:	Mon, 18 Dec 2023	AIC	178394.945
Time:	23:12:09	BIC	178417.946
Sample:	01-11-2016 - 04-30-2016	HQIC	178402.557
Covariance Type:	opg		

	coef	std err	z	P>|z|	[0.025	0.975]
const	98.0245	4.268	22.965	0.000	89.659	106.390
ar.L1	0.7569	0.004	214.905	0.000	0.750	0.764
sigma2	4727.4683	22.548	209.661	0.000	4683.275	4771.662

Ljung-Box (L1) (Q):	53.75	Jarque-Bera (JB):	361080.24
Prob(Q):	0.00	Prob(JB):	0.00
Heteroskedasticity (H):	0.84	Skew:	3.18
Prob(H) (two-sided):	0.00	Kurtosis:	25.55

Forecast Function – h-step Prediction

I developed a forecast function that can handle any ARIMA/SARIMA model, assuming that we have already determined the model order; using SARIMAX package. It takes historical/previous data, the model order and seasonal order, and steps/periods and returns the forecasted values. It can do 1-step or h-step predictions. This was used to calculate the forecasted error for ARIMA(1, 0, 0).

```
def forecast_function(history, order, seasonal_order, forecast_periods):
    """
    """
    # Fit the SARIMA model
    model = sm.tsa.SARIMAX(history, order=order, seasonal_order=seasonal_order,
                           enforce_stationarity=False, enforce_invertibility=False)
    model_fit = model.fit(disp=False)

    # Make forecast
    forecast = model_fit.forecast(steps=forecast_periods)

    return forecast
```

Residual Analysis

a. Whiteness Chi-square test:

```
Q =  1.594454497885215
Q* =  42.55696780429269
The residual is white
```

b. Display the estimated variance of the error and the estimated covariance of the estimated parameters.

```
Variance of the errors: 4727.560362364504
Covariance matrix of the parameters:
            const     ar.L1      sigma2
const   18.219319 -0.010906  -70.696950
ar.L1   -0.010906  0.000012    0.037842
sigma2 -70.696950  0.037842  508.418753
```

c. Is the derived model biased or this is an unbiased estimator?

```
Mean of the residuals: 0.0096823923052894 06
```

It is close to 0, meaning that the derived model is unbiased.

d. Check the variance of the residual errors versus the variance of the forecast errors.

```
Variance of the errors: 4727.560362364504
Forecast Variance:  8104.445983130075
```

This suggests that the model does not generalize well as it performs much worse on unseen data.

e. Model simplification: Perform zero-pole cancellation operation and display the final coefficient confidence interval.

We see that it has the same root as the current model. The confidence interval suggests all parameters are significant since there is no 0 in between.

```
Final coefficient(s):  -0.7569
```

```
Confidence interval:
                0           1
const      89.658522  106.390380
ar.L1       0.749990    0.763796
sigma2   4683.274723 4771.661819
```

10- Final Model Selection

Table 5. All Models' Performance

	RMSE	% Improvement	R-squared
Naive	223.05	0.00	NaN
Average	91.05	59.18	NaN
Drift	250.96	-12.51	NaN
Simple Smoothing	104.14	53.31	NaN
Exponential Smoothing	262.90	-17.87	NaN
Holt-Winters	204.71	8.22	NaN
Regressions	94.38	57.69	0.07
ARMA	131.51	41.04	NaN

This table shows all models' performance as measured by RMSE, with the baseline model being Naïve. Average performed the best, with 91.05 RMSE and is 59.2% better than Naïve. Regression also performed similarly, with RMSE of 94.4. According to this table, we would go with one of the base models, Average, as no other models has performed better.

11- Summary and Conclusion

As shown above, the best performing model was Average, which is one of the base models. Since it is not complete and fully bug-free, I did not include – but a LSTM was built, and shown the best performance, with a much lower RMSE than the rest If there was more time, I would have done the following:

1. Redo feature selection
2. Test other ARIMA model orders
3. Finish a LSTM model
4. Try 2-3 other models, such as K-NN, SVM or Random Forests

Lei Xiang et al (2020) in the Journal of Physics: Conference Series documented their work on the same data set, and they achieved an RMSE of 2.48 for LSTM with a different set of features included. Given time, I would like to replicate what they do in the paper to see if I could achieve the same results.

12 – Appendix

The project submission includes “Final Project README.txt”, “Final_Project_EP_TS.py” (main file) and “tools.py (toolbox)”. A separate .txt Appendix of the main code will also be included.

13 – References

(1) Xiang, L., et al. (2020). [Title of the Paper]. Journal of Physics: Conference Series, 1453(012064). https://iopscience.iop.org/article/10.1088/1742-6596/1453/1/012064
(2) Candanedo,Luis. (2017). Appliances energy prediction. UCI Machine Learning Repository. https://doi.org/10.24432/C5VC8G.