Final Term Project (FTP)

Dr. Reza Jafari
DATS 6450: Time Series Modeling & Analysis

Maeshal Hijazi
MH - 12/16/2021

Table of Contents

Abstract	5
Introduction	5
Description	6
Stationarity	9
Time series Decomposition	13
Holt-Winters Method	15
Feature Selection	15
Base Models	16
Multiple Linear Regression	17
ARMA/ARIMA/SARIMA Model	20
LM Algorithm	24
Diagnostic Analysis	26
Forecast Function	27
h-step Ahead Predictions	28
Conclusion	29

List of Figures

Figure 1 - Number of missing data points in the chosen dataset 7
Figure 2 - Dependent variable plot versus time for the chosen dataset 8
Figure 3 - ACF plot for the dependent variable 8
Figure 4 - PACF plot for the dependent variable 9
Figure 5 - Heatmap of the correlation matrix of the available features 9
Figure 6 - Rolling mean of the dependent variable 10
Figure 7 - Rolling variance of the dependent variable 11
Figure 8 - ADF test result for the dependent variable 11
Figure 9 - KPSS test result for the dependent variable 11
Figure 10 - Rolling mean of the dependent variable after differencing 12
Figure 11 - Rolling variance of the dependent variable after differencing 12
Figure 12 - ADF test result for the dependent variable after differencing 12
Figure 13 - KPSS test result for the dependent variable after differencing 13
Figure 14 - STL decomposition for the dependent variable 13
Figure 15 - Detrended dependent variable 14
Figure 16 - Seasonally adjusted dependent variable 14
Figure 17 - Strength of trend for dependent variable 14
Figure 18 - Strength of seasonality for dependent variable 14
Figure 19 - Holt's Winter forecast on the test set 15
Figure 20 - Singular values for all features 15
Figure 21 - ROC of variance done through PCA 16
Figure 22 - Singular values for the remaining feature 16
Figure 23 - Base models' forecast 17
Figure 24 - Model summary of the final 18
Figure 25 - One-step ahead prediction for the regression model 19
Figure 26 - ACF of the residual for the regression model 19
Figure 27 - Regression model prediction on the test set 20
Figure 28 - GPAC table for the dependent variable after 4 differencing 21
Figure 29 - ACF/PACF for the dependent variable after 4 differencing 22
Figure 30 - Multiplicative model # 1 22
Figure 31 - Multiplicative model # 2 22
Figure 32 - ACF of the residual for model # 1 22
Figure 33 - ACF of the residual for model # 2 23
Figure 34 -One-step ahead prediction for model 1 for the first 250 samples 23
Figure 35 -One-step ahead prediction for model 1 for the last 250 samples 24
Figure 36 - Confidence interval for each estimated parameter in the non-seasonal submodel 26
Figure 37 - Confidence interval for each estimated parameter in the first seasonal submodel (24 seasonality) 26
Figure 38 - Confidence interval for each estimated parameter in the second seasonal submodel (168 seasonality) 26

Figure 39 - Zeros and poles for the non-seasonal submodel 27
Figure 40 - Zeros and poles for the first seasonal submodel (24 seasonality) 27
Figure 41 - Zeros and poles for the second seasonal submodel (168 seasonality) 27
Figure 42 - Forecast function 28
Figure 43 - h-step forecast for the multiplicative model 28

List of Tables

Table I - MSE for base models 16
Table II - Calculated metrics for the final regression model 18
Table III - Residual information 19
Table IV - Estimated parameters for the non-seasonal submodel 24
Table V - Estimated parameters for the seasonal submodel (24 seasonality) 25
Table VI - Estimated parameters for the seasonal submodel (168 seasonality) 25

Abstract

In this project students had to study, analyze, and model a time series of their own. It must be a non-classified dataset with more than 500 samples. The dataset used in this project was the hourly load data in kW for Electric Reliability Council of Texas (ERCOT) for 8 weather zones from January 1st, 2020 till September 31st, 2021. Students had to apply and implement what was taught in class on the picked dataset and try to build the most possible accurate model. This model was used to forecast up to the length of the test set used in the project.

Introduction

The dataset picked for this final project was an hourly dataset for the load demand of 8 different weather zones in Texas. These zones were as follows: coast, east, feast, north, ncent, south, scent, and the west. These zones were added together to form the column for the total demand which was named as ERCOT. The west zone was assumed to be the dependent variable (output) and the remaining zones with the total were assumed as the independent variables (inputs).

As an introductory, students checked if the dataset did not have any missing values. If it had, students followed certain time series techniques to fill them. Then, students plotted the dependent variable versus time and calculated the ACF and PACF of it. Furthermore, students also plotted the correlation matrix in a heatmap for all features including the output. After that students splitted the dataset into the first 80% for training and the remaining for testing. These steps are available in the *Description* subsection.

After describing the dataset, students in this final project checked the stationarity of the dependent variable by plotting the rolling mean and variance and calculating the ADF and KPSS statistical tests. If the output was not stationary, then students stationarized it using seasonal or nonseasonal differencing depending on the type of the dataset chosen. These steps are available in the *Stationarity* subsection. After that, students decomposed the time series dataset and approximated the trend and seasonality by assigning the period of the model. Furthermore, students in that subsection plotted the detrended dataset and the seasonally adjusted dataset and calculated the strength of trend and seasonality. These strengths indicated how trended and/or seasonal was the dataset chosen. These steps are available in the *Time series Decomposition* subsection.

In the next subsection, students used the Holt's Winter method to find the best fit for the given training data and to test it on the testing data. These steps are available in the *Holt-Winters method* subsection. After that, students selected the most important features of the independent variable to pass them the multiple feature linear regression model. The selection process was done using Variance Inflation Factor (VIF) and Principal Component Analysis (PCA). In addition, students checked if collinearity existed between the independent variables or not by calculating the singular values and conditional number. All these steps and detailed information are available in the *Feature Selection* subsection.

In order for students to compare their best model, they calculated some base models for the given dataset. These models are: average method, naive method, average method, and simple and exponential smoothing method. These models are available in the *Base Models* subsection. Other than the Holt's Winter method, students implemented a multiple feature linear regression model using the selected features. For the best developed regression model, students performed one-step ahead prediction and compared it to the train set. Using the one-step ahead prediction, students calculated the residual which was then used to calculate the Q value. Students used that residual to plot the ACF and calculate the mean and variance for it. Furthermore, students checked the t-test and f-test results of the best developed regression model and calculated the Akaike information criterion (AIC), Bayesian information criterion (BIC), root mean squared error (RMSE), R^2, and adjusted R^2. All these calculations and plots are available in the *Multiple Linear Regression* subsection.

With regression and Holt's Winter models, students built at least an ARMA, ARIMA, or SARIMA model in order to compare to the previous two models. The orders for the chosen model were estimated/predicted using the GPAC table of the differenced dependent variable and ACF and PACF plots for it. All these plots and estimated orders are discussed and indicated in the *ARMA/ARIMA/SARIMA Model* subsection. The parameters of the picked model were estimated using the Levenberg Marquardt (LM) algorithm and printed in the *LM Algorithm* subsection. In that subsection, students printed the estimated parameters with the confidence interval for each and the standard deviation for the whole estimate.

To see if the ARMA, ARIMA, or SARIMA model was correctly describing the dependent variable, students calculated χ^2 test which is a whiteness test that indicates if the residual is white or not. If the residual was white, students can conclude the model developed was accurate. In addition, students checked for zero and pole cancellation after parameter estimation. All these calculations and plots are available in the *Diagnostic Analysis* subsection.

After the developed ARMA, ARIMA, or SARIMA model passed the diagnostic tests, students wrote a forecast function in order to forecast h-step ahead predictions. The forecasted predictions are compared to the true test dataset. The forecast function is available in the *Forecast Function* subsection while the h-step ahead prediction results are available in *h-step Ahead Predictions* subsection.

Description

In this subsection, students described the dependent variable and independent variables of the picked dataset.

a. In this subpart, students checked if the dataset was missing any data points. Figure 1 shows that the dataset did not have any missing data point therefore, no need to impute anything.

b. In this subpart, students plotted the dependent variable versus time which is shown in Figure 2. From that plot, clearly the dependent variable is trended.

c. In this subpart, students plotted the ACF and PACF plots of the dependent variable as shown in Figures 3 and 4. It can be seen that there is a high seasonality (seasonality of 24) in the ACF plot shown in Figure 3.

d. In this subpart, students plotted a seaborn heatmap for the Pearson's correlation coefficients which is shown in Figure 5. Clearly, all features are highly correlated because if the load is high in a certain weather zone, it is expected to be high in neighboring locations. Furthermore, since ERCOT is the total of all the loads in the 8 different regions, it has to be very correlated with other features.

```
Description_Part a:
HourEnding    0
COAST         0
EAST          0
FWEST         0
NORTH         0
NCENT         0
SOUTH         0
SCENT         0
WEST          0
ERCOT         0
dtype: int64

The number of missing data is 0
```

Figure 1 - Number of missing data points in the chosen dataset

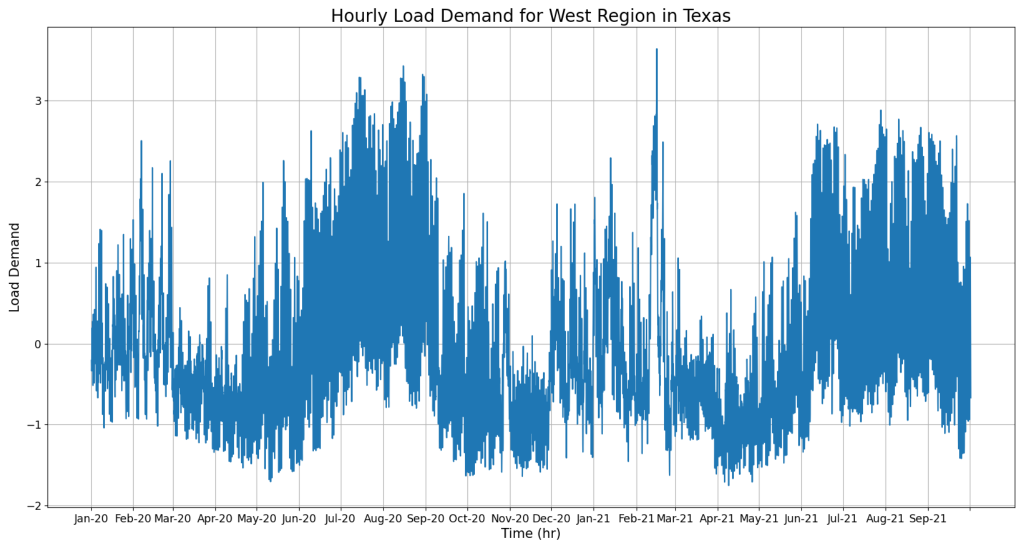

Figure 2 - Dependent variable plot versus time for the chosen dataset

e. In this subpart, students splitted the first 80% of the dataset for training and the last 20% for testing.

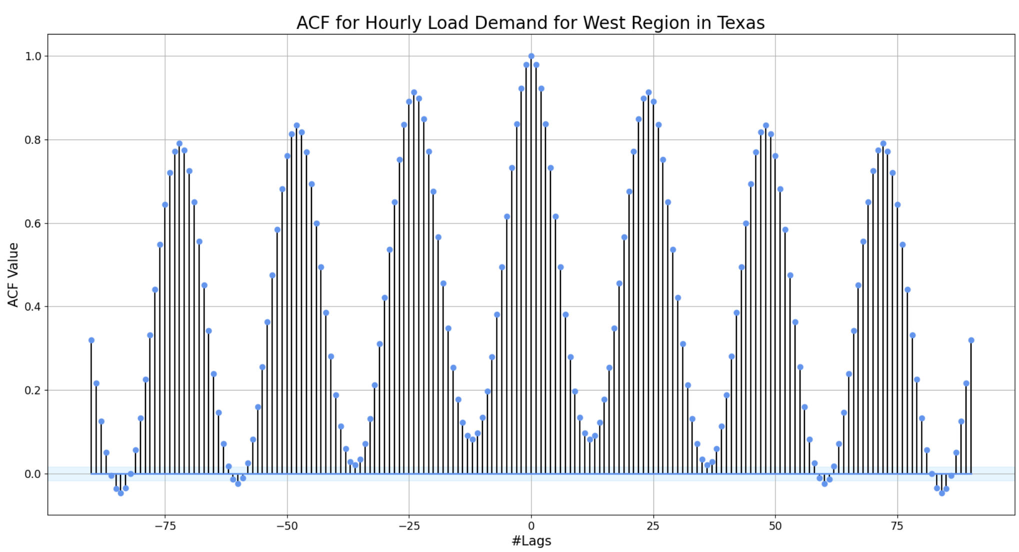

Figure 3 - ACF plot for the dependent variable

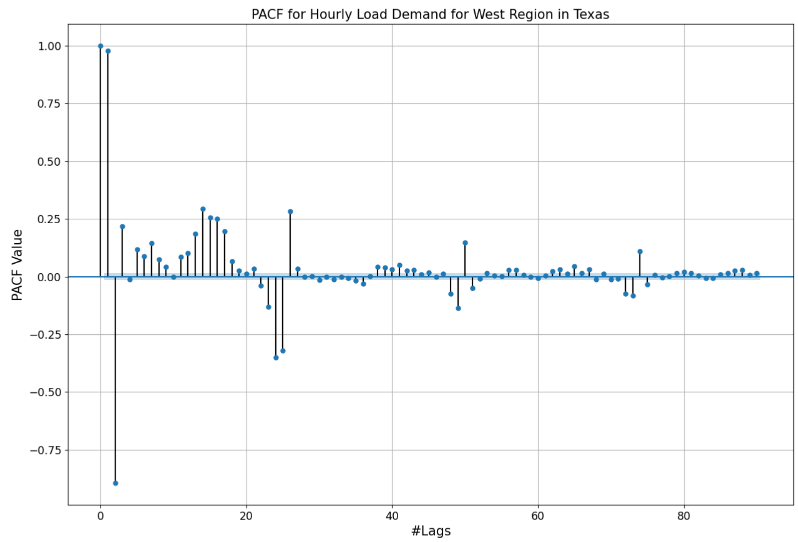

Figure 4 - PACF plot for the dependent variable

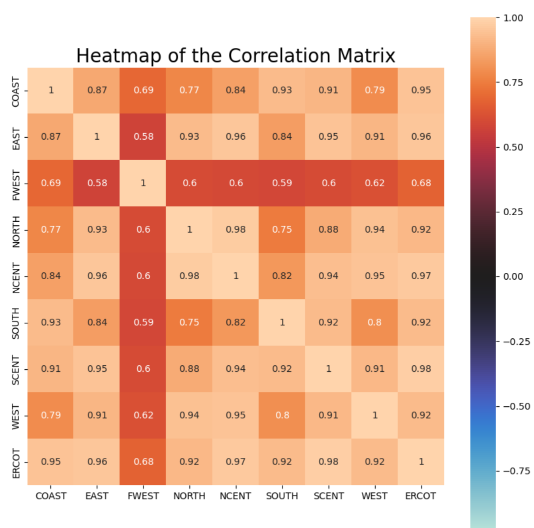

Figure 5 - Heatmap of the correlation matrix of the available features

Stationarity

Students in this subsection checked if the dependent variable was stationary or not by plotting the rolling mean variance for it and by calculating the ADF and KPSS statistical tests. The rolling mean and variance plots are shown in Figures 6 and 7, respectively. Both figures

indicate that the rolling mean and variance are not stabilizing (not converging) which means that the dependent variable is
nonstationary. Furthermore, the results of the ADF and KPSS statistical tests on the dependent variable are shown in Figures 8 and 9, respectively. Even though the p-value for the ADF statistical test is zero, the plots of the rolling mean and variance and the p-value of the KPSS statistical test indicate that the dependent variable is nonstationary.

To stationarize the dependent variable, students applied seasonal or nonseasonal differencing. Since the dataset for this project was highly seasonal, students applied a 24 hr seasonal differencing (24 lags). Then, students replotted the rolling mean and variance of the differenced dependent variable as shown in Figures 10 and 11, respectively. In addition, students recalculated the ADF and KPSS statistical tests for the differenced dependent variable. The results for both tests are shown in Figures 12 and 13, respectively. Since both plots were converging, then the dependent variable became stationary after differencing. Furthermore, the p-value for the ADF test is below the threshold while for the KPSS is higher than the threshold indicating as well that the dependent variable became stationary after differencing.

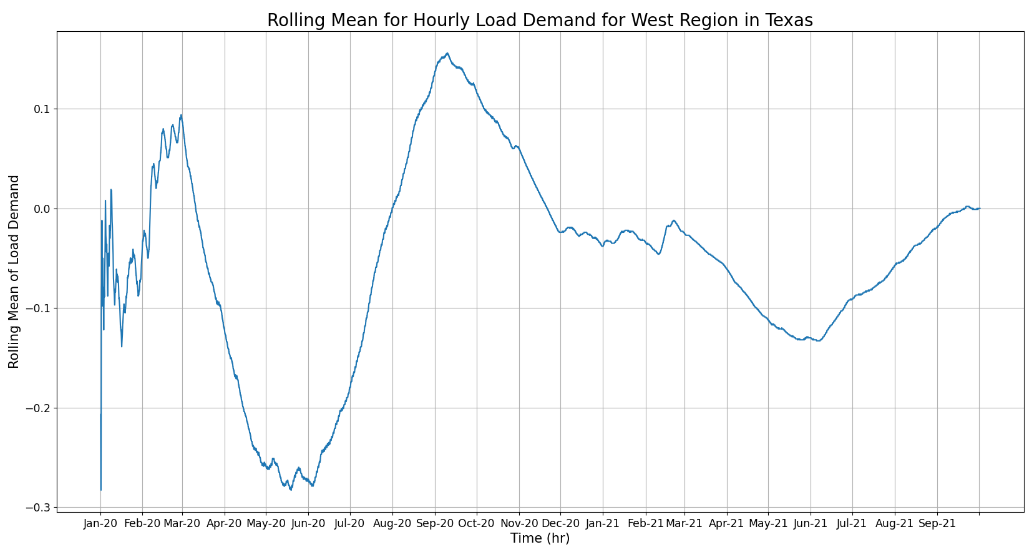

Figure 6 - Rolling mean of the dependent variable

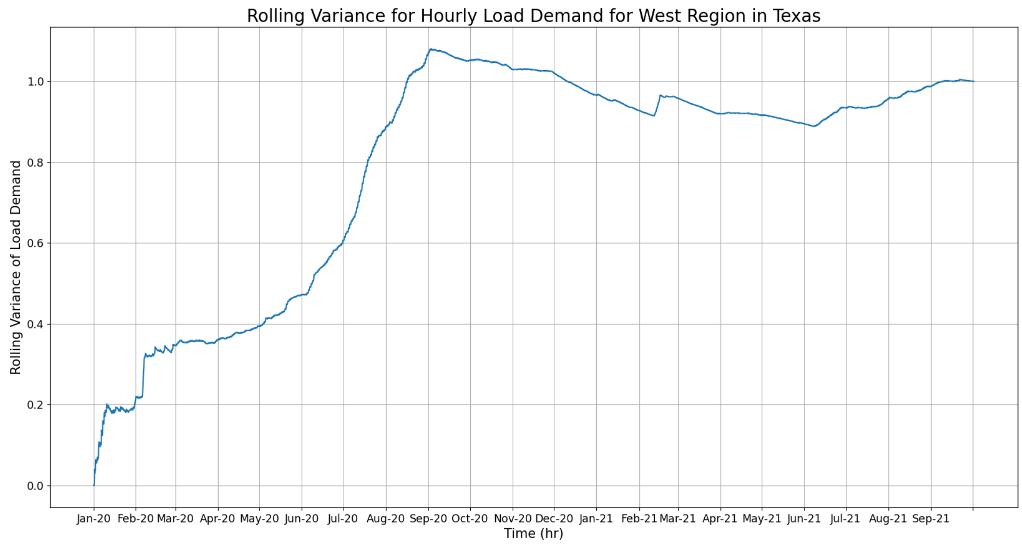

Figure 7 - Rolling variance of the dependent variable

```
ADF Statistic: -6.069750
p-value: 0.000000
Critical Values:
    1%: -3.431
    5%: -2.862
    10%: -2.567
```

Figure 8 - ADF test result for the dependent variable

```
Test Statistic              1.114002
p-value                     0.010000
Lags Used                  63.000000
Critical Value (10%)        0.347000
Critical Value (5%)         0.463000
Critical Value (2.5%)       0.574000
Critical Value (1%)         0.739000
dtype: float64
```

Figure 9 - KPSS test result for the dependent variable

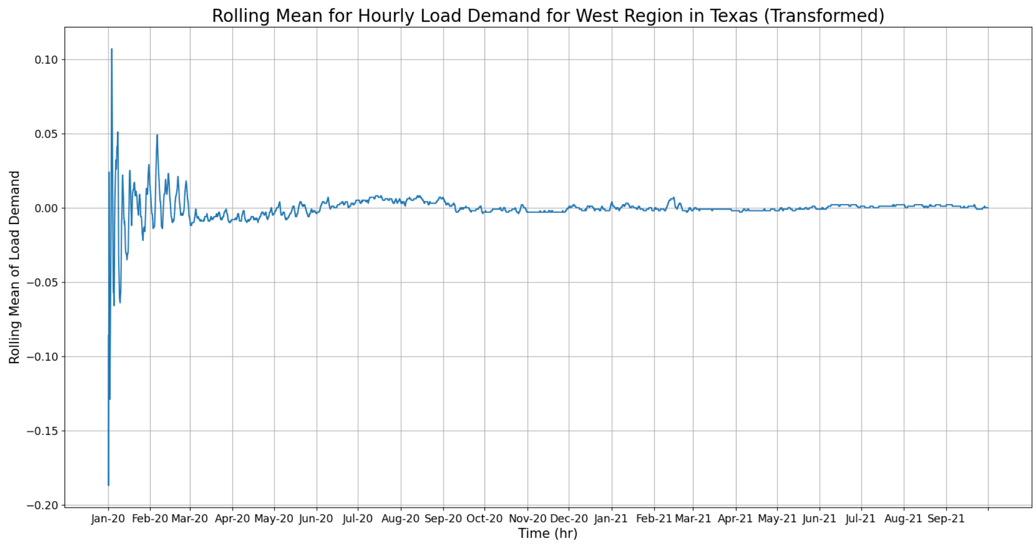

Figure 10 - Rolling mean of the dependent variable after differencing

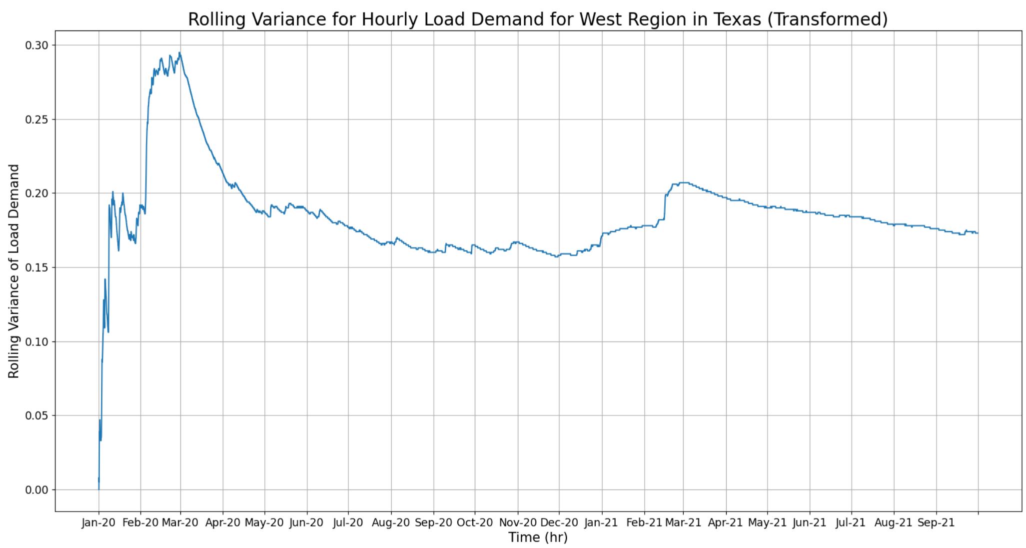

Figure 11 - Rolling variance of the dependent variable after differencing

```
ADF Statistic: -21.398354
p-value: 0.000000
Critical Values:
    1%: -3.431
    5%: -2.862
    10%: -2.567
```

Figure 12 - ADF test result for the dependent variable after differencing

```
Results of KPSS Test:
Test Statistic            0.007686
p-value                   0.100000
Lags Used                68.000000
Critical Value (10%)      0.347000
Critical Value (5%)       0.463000
Critical Value (2.5%)     0.574000
Critical Value (1%)       0.739000
```

Figure 13 - KPSS test result for the dependent variable after differencing

Time series Decomposition

In this subsection, students decomposed the time series dependent variable into trend, seasonality, and residual as shown in Figure 14. Clearly, the dependent variable is highly trended and highly seasonally as shown in the figure. Students also approximated the trend and seasonality of the dependent variable by plotting the detrended and seasonally adjusted datasets as shown in Figures 15 and 16, respectively. A multicative model was assumed because the rolling mean and variance were not constant. Furthermore, students calculated the strength for both seasonality and trend as shown in Figures in Figures 17 and 18, respectively. Clearly, the dependent variable is highly seasonal and trended.

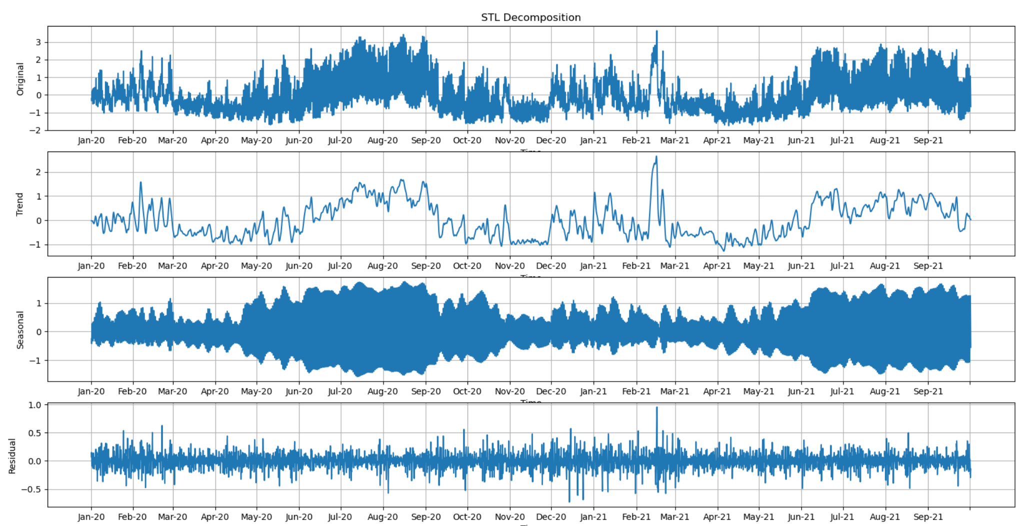

Figure 14 - STL decomposition for the dependent variable

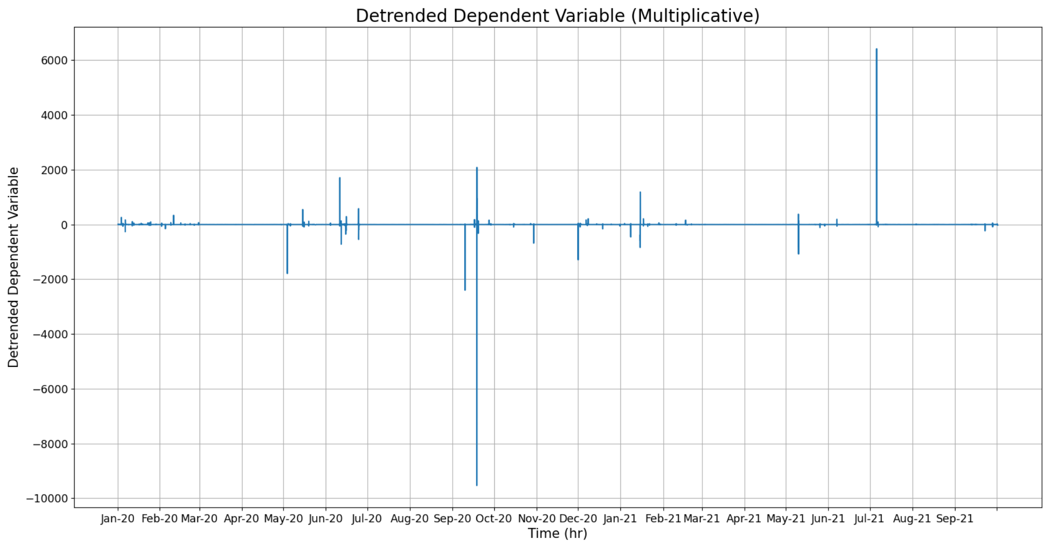

Figure 15 - Detrended dependent variable

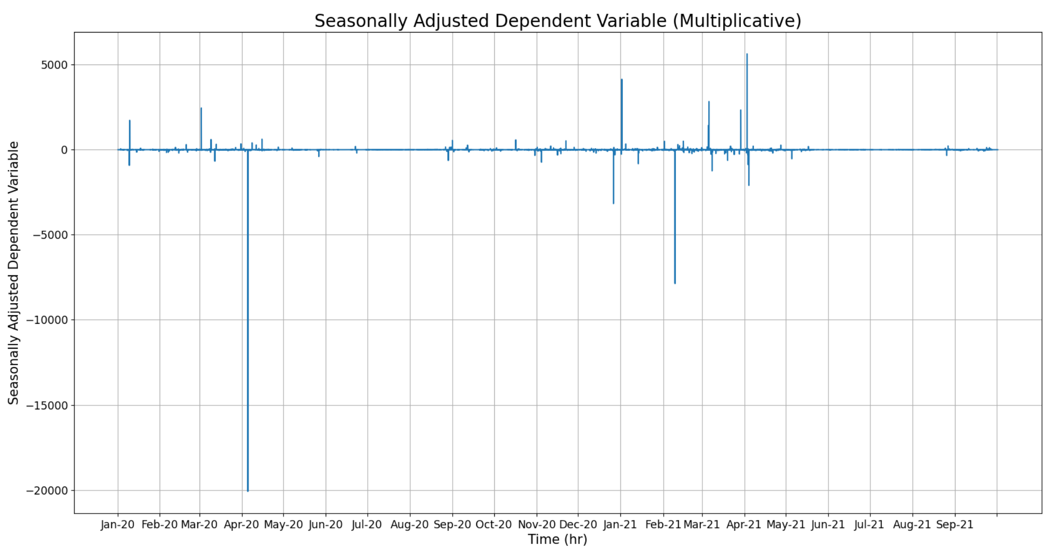

Figure 16 - Seasonally adjusted dependent variable

The strength of trend was 0.972 (highly trended)

Figure 17 - Strength of trend for dependent variable

The strength of seasonality was 0.968 (highly seasonal)

Figure 18 - Strength of seasonality for dependent variable

Holt-Winters method

In this subsection, students built a Holt's Winter model to find the best fit that fits the dependent variable. The best model was tested on the test set. The forecast of the test set is shown in Figure 19. Clearly, the Holt's Winter method did not perform well. The MSE for this method is 0.41.

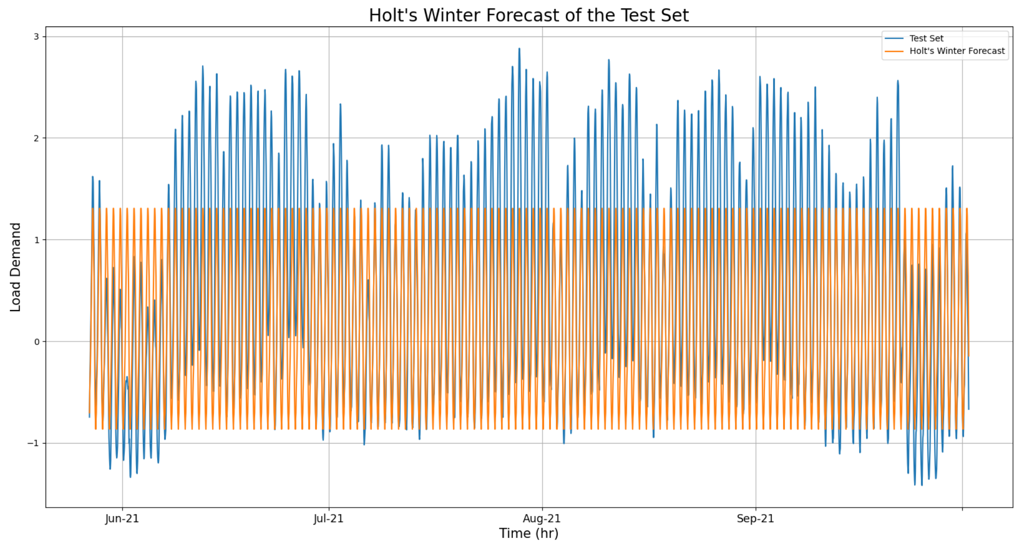

Figure 19 - Holt's Winter forecast on the test set

Feature selection

In this subsection, students performed a feature selection process and detected multicollinearity between features in order to pick the minimum features required to build a multiple feature linear regression model. Students checked if collinearity existed between the features by calculating the singular values and the conditional number. The singular values for all dependent variables are shown in Figure 20. Furthermore, the conditional number for all features was 291.096. Since some of the singular values are close to zero and the conditional number is greater than 100 there exists moderate collinearity between features and thus some features have to be removed.

[105509.189, 8829.17, 5701.358, 1117.664, 867.054, 466.862, 187.456, 1.245]

Figure 20 - Singular values for all features

To determine which features to keep and which to remove, students performed PCA. Based on the PCA results shown in Figure 21, only 3 features were needed. These 3 features define 98% of the variance. To pick these 3 features, the student performed VIF. The feature with the highest VIF score was dropped and then VIF for the remaining features was calculated. The process was repeated until three features remained. The constant feature was removed because the mean for each variable in the dataset was subtracted from it. The 3 remaining features were fwest, north, and south. The singular value numbers for the remaining features are shown in Figure 22. In addition, the conditional number for the remaining features was 4.302. These results show that there is no collinearity anymore after removing some of the features.

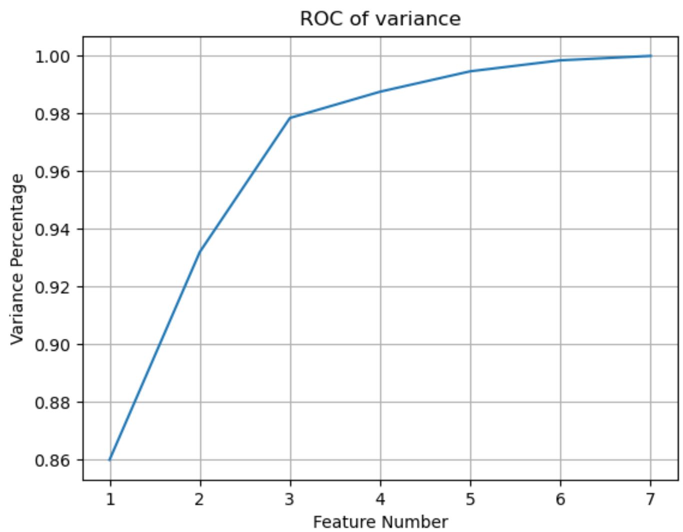

Figure 21 - ROC of variance done through PCA

[105509.1891, 8829.17, 5701.3583]

Figure 22 - Singular values for the remaining feature

Base Models

In order to test built forecast models, they have to be compared against some base models. These base models are known as average model, naive model, drift model, and simple and exponential smoothing model. Students used these models to perform an h-step prediction as shown in Figure 23 and compared the mean squared error (MSE) for them against any built model such as SARIMA model or Holt's Winter model. The MSE for all base models are summarized in Table I. Based on the MSE value shown in the Table, the average method performed the best for the chosen dependent variable.

Table I - MSE for base models

Type	MSE
Average	1.495
Naive	2.927
Drift	3.151
SES	2.759

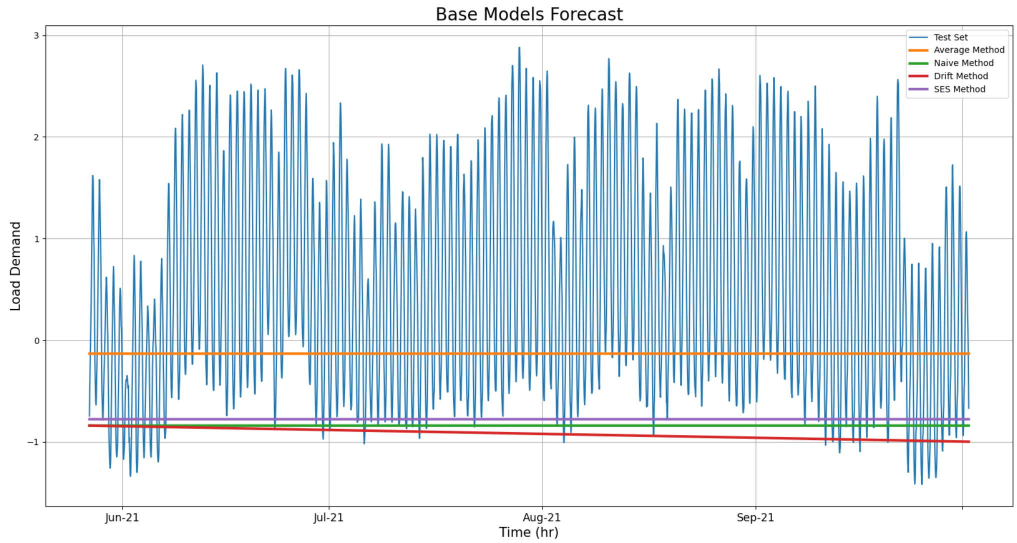

Figure 23 - Base models' forecast

Multiple Linear Regression

Using the selected features, students in this subsection built multiple features linear regression model. The model summary of the final regression model is shown in Figure 24. Clearly, all features are statistically important since the p-values are all zeros. Thus, the model passes the t-test. Furthermore, the F-test probability is very close to zero, thus the model passed the F-test. The model error metrics are shown in Table II. This table includes AIC, BIC, RMSE, R^2, and adjusted R^2.

In order to determine if the regression model is an accurate model, a whiteness test was performed. To do the whiteness test, students had to first calculate the one-step ahead prediction as shown in Figure 25 and subtract it from the train test set.

```
                            OLS Regression Results
=======================================================================================
Dep. Variable:                   WEST   R-squared (uncentered):                   0.912
Model:                            OLS   Adj. R-squared (uncentered):              0.912
Method:                 Least Squares   F-statistic:                          4.221e+04
Date:                Tue, 07 Dec 2021   Prob (F-statistic):                        0.00
Time:                        22:40:16   Log-Likelihood:                         -1976.1
No. Observations:               12268   AIC:                                      3958.
Df Residuals:                   12265   BIC:                                      3980.
Df Model:                           3
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
FWEST          0.1061      0.004     29.307      0.000       0.099       0.113
NORTH          0.7983      0.004    204.607      0.000       0.791       0.806
SOUTH          0.1796      0.004     48.017      0.000       0.172       0.187
==============================================================================
Omnibus:                      436.540   Durbin-Watson:                   0.045
Prob(Omnibus):                  0.000   Jarque-Bera (JB):             1012.483
Skew:                           0.201   Prob(JB):                    1.39e-220
Kurtosis:                       4.349   Cond. No.                         2.48
==============================================================================
```

Figure 24 - Model summary of the final

The difference is called the residual. If the ACF of the residual was close to zero, then the residual is white and the model is accurate. The ACF of the residual is shown in Figure 26. Clearly, the ACF is not white, thus we can conclude that the model is not accurate. Furthermore, the Q value, the mean, and the variance of the residual are shown in Table III. The model was tested on the test set and the results are shown in Figure 27. The MSE of the regression model on the test set was 0.145. This is smaller than the average method MSE value.

Table II - Calculated metrics for the final regression model

Metric	**Value**
AIC	3958.238
BIC	3980.482
RMSE	0.380
R^2	0.912
Adjusted R^2	0.912

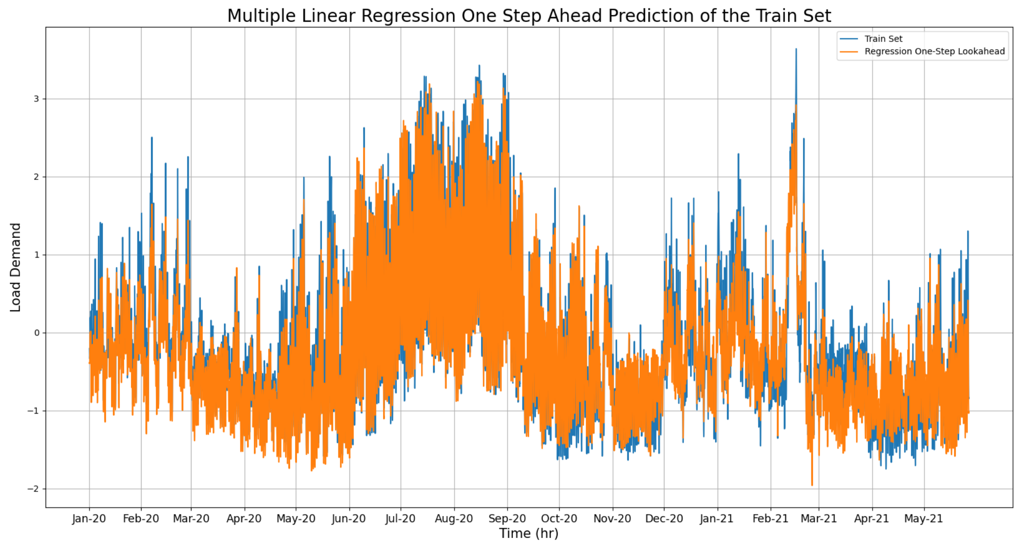

Figure 25 - One-step ahead prediction for the regression model

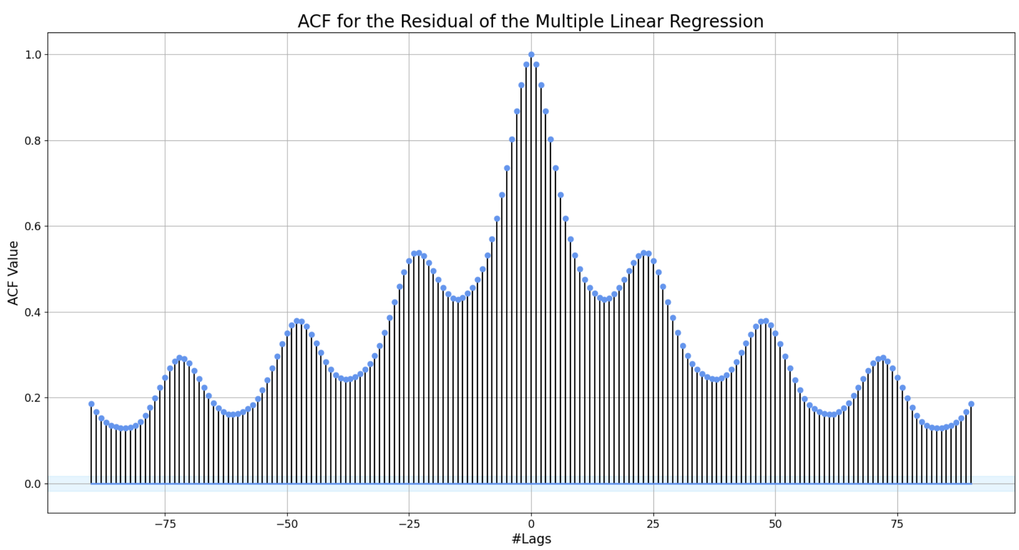

Figure 26 - ACF of the residual for the regression model

Table III - Residual information

Q	205507.824
Mean for error	0.063
Variance for error	0.077

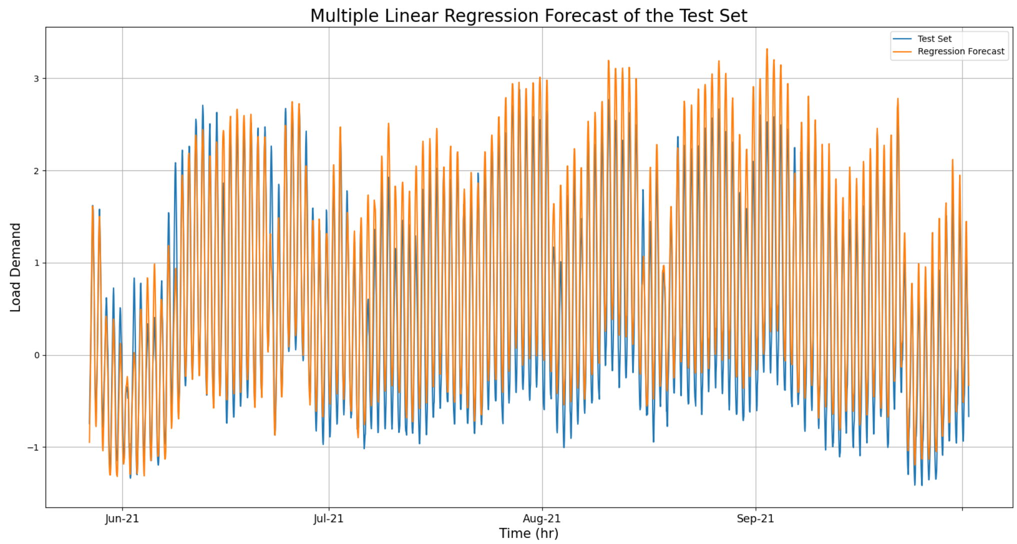

Figure 27 - Regression model prediction on the test set

ARMA/ARIMA/SARIMA Model

In this subsection, students had to identify patterns in the GPAC table and ACF and PACF plots to develop an ARMA, ARIMA, or a SARIMA model depending on the type of the time series dataset chosen. The GPAC table of the dependent variable after being differenced twice non-seasonally, one time seasonally with a seasonality of 24, and one time seasonally with a seasonality of 168 is shown in Figure 28. Furthermore, the ACF and PACF plot of the dependent variable after the 4 differencing is shown in Figure 29. Two orders were estimated using the GPAC table for the non-seasonal part which were: (n_a, n_b) equals (1, 0) and (n_a, n_b) equals (24, 0).

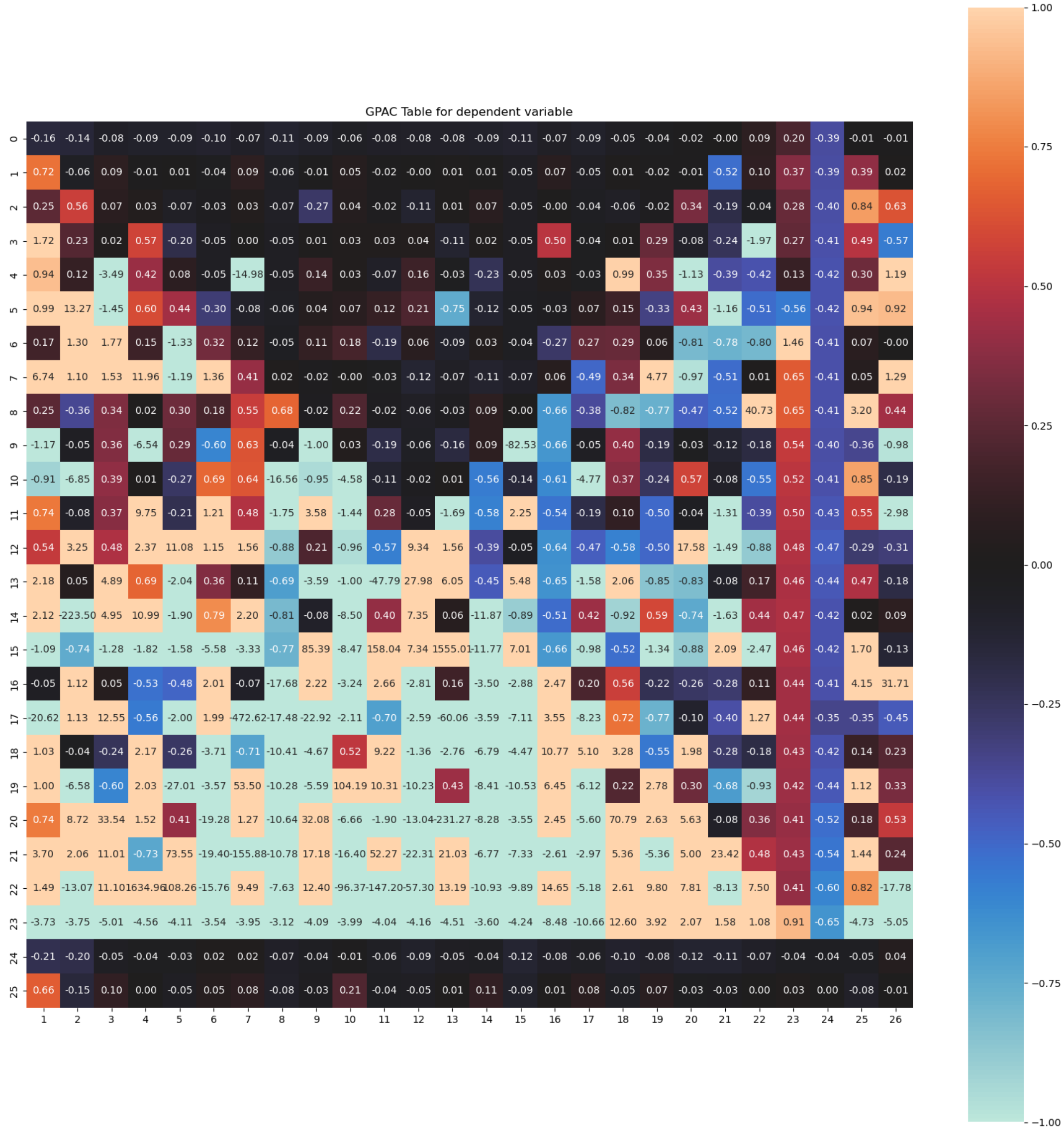

Figure 28 - GPAC table for the dependent variable after 4 differencing

For the seasonal orders, clearly there is a cutoff in the ACF plot after lag 24, 124, and 168. In addition, the PACF plot is tailing off every 24 lags which is due to the MA process. Therefore, the estimated orders for the seasonal part after many trials and errors were $(n_a, n_b)_{24}$ equals (0, 1) and $(n_a, n_b)_{168}$ equals (1, 2). Since there exists two seasonalities, a multiplicative model was needed because the built-in SARIMA package in Python is only capable of doing one seasonality. Therefore, two multiplicative models were compared. Model 1 is shown in Figure 30 and model 2 is shown in Figure 31.

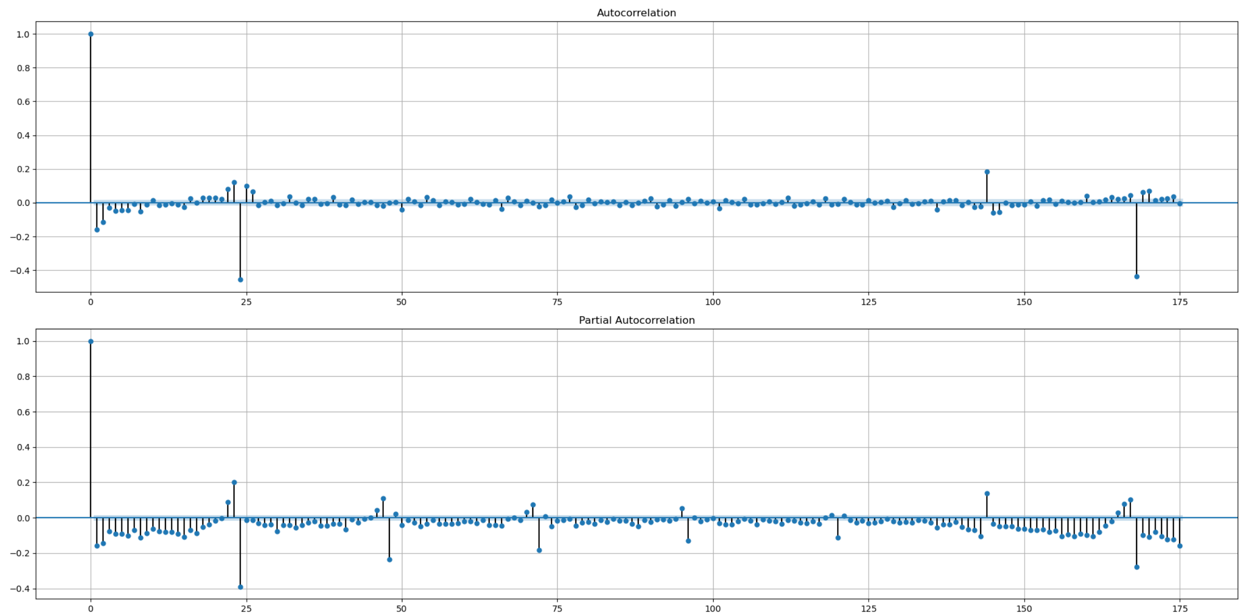

Figure 29 - ACF/PACF for the dependent variable after 4 differencing

$$(24, 2, 0) \times (0, 1, 1)_{24} \times (1, 1, 2,)_{168}$$

Figure 30 - Multiplicative model # 1

$$(1, 2, 0) \times (0, 1, 1)_{24} \times (1, 1, 2,)_{168}$$

Figure 31 - Multiplicative model # 2

Students calculated the one-step ahead prediction for both models and then calculated the residuals. Then, students plotted the ACFs for the residuals and were diagnosed. Figures 32 and 33 show the ACFs of the residuals for model 1 and 2, respectively. Based on the plots, clearly model 1 performed better than model 2 and thus model 1 was concluded to be better. The one-step ahead prediction used for calculating the residuals for model 1 is shown in Figures 34 and 35. Figure 34 was for the first 250 samples while Figure 35 was for the last 250 samples.

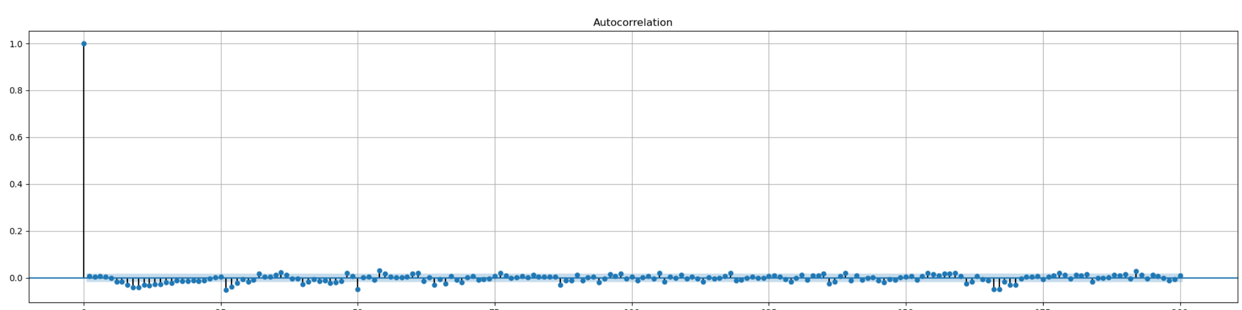

Figure 32 - ACF of the residual for model # 1

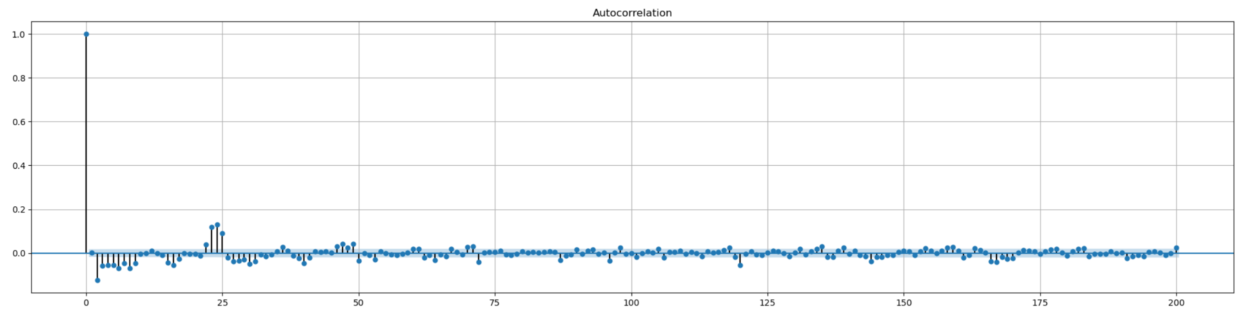

Figure 33 - ACF of the residual for model # 2

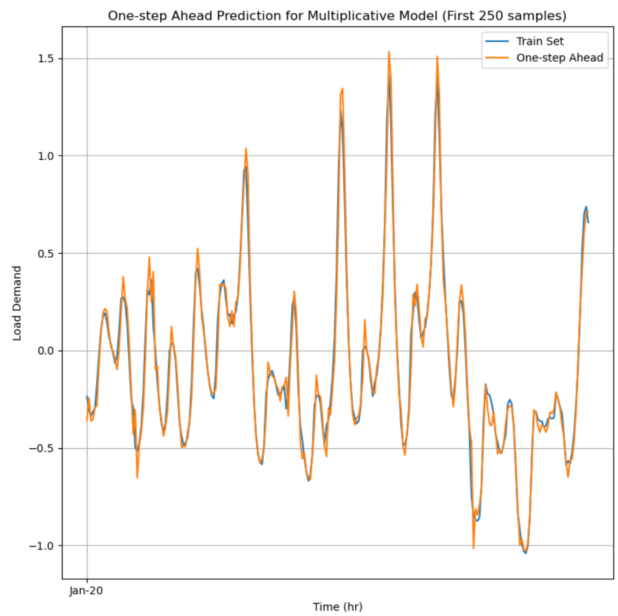

Figure 34 -One-step ahead prediction for model 1 for the first 250 samples

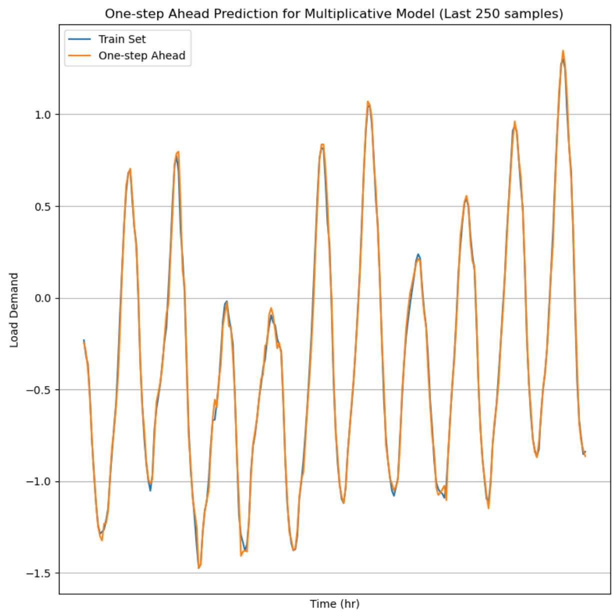

Figure 35 -One-step ahead prediction for model 1 for the last 250 samples

LM Algorithm

The parameters for the multiplicative model were estimated using the LM algorithm in this subsection. Table IV, Table V, and Table VI show the estimated parameters for each submodel. Furthermore, the confidence intervals for each estimated parameter for each submodel are shown in Figures 36, 37, and 38, respectively. All figures show that all parameters are statistically significant.

Table IV - Estimated parameters for the non-seasonal submodel

a_1	0.2682
a_2	0.3366
a_3	0.2992
a_4	0.3209
a_5	0.3301
a_6	0.3509
a_7	0.3428
a_8	0.3695
a_9	0.3504
a_{10}	0.3236
a_{11}	0.3303

a_{12}	0.3097
a_{13}	0.3152
a_{14}	0.3076
a_{15}	0.3272
a_{16}	0.3231
a_{17}	0.3047
a_{18}	0.2666
a_{19}	0.2530
a_{20}	0.2369
a_{21}	0.2246
a_{22}	0.1731
a_{23}	0.0800
a_{24}	-0.0303

Table V - Estimated parameters for the seasonal submodel (24 seasonality)

b_1	-0.9100

Table VI - Estimated parameters for the seasonal submodel (168 seasonality)

a_1	-0.7034
b_1	-1.4672
b_2	0.4754

```
0.2520281343152856 < a1 < 0.28443120411621414
0.31989460917160173 < a2 < 0.35330948854848354
0.281825267357961 < a3 < 0.31649881394357193
0.30328985829169175 < a4 < 0.33855178206011727
0.31217495812016594 < a5 < 0.34812395453399936
0.33261075735481976 < a6 < 0.3692413979568337
0.3240576775173712 < a7 < 0.36145991027798113
0.35057226641433864 < a8 < 0.3883427676701527
0.3312886659313191 < a9 < 0.3694704871835919
0.3044084580237512 < a10 < 0.3428331615960258
0.3110079370673442 < a11 < 0.3495159942834495
0.29035220621225133 < a12 < 0.3290469602018006
0.2958044344368811 < a13 < 0.33451434386780454
0.2883594605734628 < a14 < 0.3269403940181131
0.3079288709186377 < a15 < 0.3464286685869162
0.30393807651707133 < a16 < 0.34222921519045274
0.2857792843008181 < a17 < 0.32364733015442926
0.24775708432127 < a18 < 0.28536371471242455
0.23450925140446774 < a19 < 0.27145485734047897
0.21869389148928206 < a20 < 0.25502093288243427
0.20678613611735142 < a21 < 0.242458706100376
0.15550353813404122 < a22 < 0.19079116234294657
0.06304076708068007 < a23 < 0.09686389988682134
-0.04770639297575098 < a24 < -0.012987938364844055
```

Figure 36 - Confidence interval for each estimated parameter in the non-seasonal submodel

```
-0.9181861576138861 < b1 < -0.9018362723175946
```

Figure 37 - Confidence interval for each estimated parameter in the first seasonal submodel (24 seasonality)

```
-0.743389998047835 < a1 < -0.6635042072909675
-1.5149186171314006 < b1 < -1.4195778310919993
0.4292611381664491 < b2 < 0.5214762632451988
```

Figure 38 - Confidence interval for each estimated parameter in the second seasonal submodel (168 seasonality)

Diagnostic Analysis

Even though the estimated parameters for the multiplicative model were statistically important and that the ACF of the residual was close to zero, the multiplicative model did not pass the χ^2 test (whiteness test). The calculated Q for the multiplicative model was 1393.53 while the Q critical was 831.10. Furthermore, the zeros and poles for each submodel are shown in Figures 39, 40, and 41, respectively. All figures indicate that there is no zero and pole cancellation and thus there was no need for model reduction. Since the multiplicative was very high dimensional, the covariance matrix was very big and would not fit in the report, yet is available in the Python script. In addition, the variance for the residual was 0.0028 while for the forecast error was 39.55. The high variance of forecast error is because the model did not pass the white test. Since the model did not pass the whiteness test, the built multiplicative model was biased. Since the dataset was highly seasonal, only multiplicative models would model the dataset.

```
Pole # 1 is: (0.9480595020957164+0.26084524240361995j)
Pole # 2 is: (0.9480595020957164-0.26084524240361995j)
Pole # 3 is: (0.8364762901574361+0.4833477948873404j)
Pole # 4 is: (0.8364762901574361-0.4833477948873404j)
Pole # 5 is: (0.6692333070508046+0.6537261050711233j)
Pole # 6 is: (0.6692333070508046-0.6537261050711233j)
Pole # 7 is: (0.46966272086063804+0.8145969629811042j)
Pole # 8 is: (0.46966272086063804-0.8145969629811042j)
Pole # 9 is: (0.22675182109960645+0.8989896593314592j)
Pole # 10 is: (0.22675182109960645-0.8989896593314592j)
Pole # 11 is: (-0.013880020194532994+0.9046468377782134j)
Pole # 12 is: (-0.013880020194532994-0.9046468377782134j)
Pole # 13 is: (-0.2438602255361679+0.8713840624421302j)
Pole # 14 is: (-0.2438602255361679-0.8713840624421302j)
Pole # 15 is: (-0.8897213735356633+0j)
Pole # 16 is: (-0.8444576031716209+0.2351333845031392j)
Pole # 17 is: (-0.8444576031716209-0.2351333845031392j)
Pole # 18 is: (-0.7680485070389499+0.4410129666765609j)
Pole # 19 is: (-0.7680485070389499-0.4410129666765609j)
Pole # 20 is: (-0.6298315998006419+0.6176518393719428j)
Pole # 21 is: (-0.6298315998006419-0.6176518393719428j)
Pole # 22 is: (-0.4524242170622575+0.7711543693841031j)
Pole # 23 is: (-0.4524242170622575-0.7711543693841031j)
Pole # 24 is: (0.2261287673998535 8+0j)
```

Figure 39 - Zeros and poles for the non-seasonal submodel

```
Zero # 1 is: 0.9100112149657403
```

Figure 40 - Zeros and poles for the first seasonal submodel (24 seasonality)

```
Zero # 1 is: 0.98429448741742
Zero # 2 is: 0.48295373669427993
Pole # 1 is: 0.7034471026694012
```

Figure 41 - Zeros and poles for the second seasonal submodel (168 seasonality)

Forecast function

In order to test the multiplicative model built, students wrote a forecast function. The forecast function is shown in Figure 42.

```python
# Initialize the h-step forecast
y_hat_t_h = []

# Initialize the h-step forecast parts (MA and AR)
y_AR_h = []
y_MA_h = []

# Loop through the length of the test set
for h in range(1, len(y_test)):
    # Initialize the yt for each h
    yt = 0

    # Loop through the AR parameters
    for param_ar_index in range(1, len(AR_all.c)):
        # Check if the parameter was non-zero
        if AR_all.c[param_ar_index] != 0:
            # Check if the index of the parameter was less than h
            if param_ar_index < h:
                # Add to yt the multiplication of the parameter by previous y_hat
                yt += AR_all.c[param_ar_index] * y_hat_t_h[h - param_ar_index - 1]
            else:
                # Add to yt the multiplication of the parameter by the part of y_train
                yt += AR_all.c[param_ar_index] * np.array(y_train)[h - param_ar_index - 1]

    # Multiplu by -1 because will be taken to the other side and append
    y_AR_h.append(-1 * yt)

    # Initialize the et for each h
    et = 0

    # Check if h was more than the length of the MA parameters
    if h > len(MA_all.c):
        # Append zero
        y_MA_h.append(0)
    else:
        # loop from h to the last parameter of MA :
        for param_ma_index in range(h, len(MA_all.c)):
            # Check if the parameter was non-zero
            if MA_all.c[param_ma_index] != 0:
                # Update et
                et += MA_all.c[param_ma_index] * (np.array(y_train)[h - param_ma_index - 1] - y_one[h - param_ma_index - 2])

        # append et to MA_h
        y_MA_h.append(et)

    # Add y_AR_h and y_MA_h
    y_hat_t_h.append(y_AR_h[h - 1] + y_MA_h[h - 1])
```

Figure 42 - Forecast function

h-step Ahead Predictions

In this subsection, students performed h-step ahead prediction of their best ARMA, ARIMA, SARIMA model using the forecast function shown in Figure 42. The h-step forecast is shown in Figure 43. After certain steps, the forecast starts to exponentially decrease which is because all the MA terms will become zero. The MSE for the multiplicative model was 67 which is very high compared to the base models and other models built in this project.

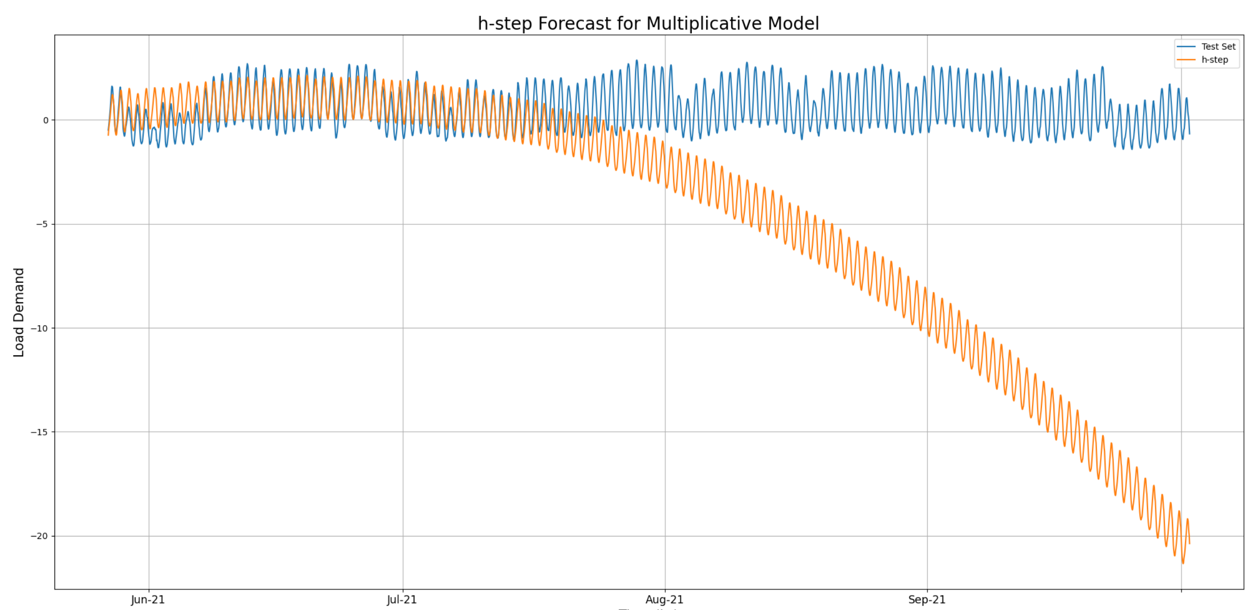

Figure 43 - h-step forecast for the multiplicative model

Conclusion

In conclusion, the main goal of this project was to imply, diagnose, and analyze what students learned through the semester. Students picked a dataset of their own and had to model it using some of the famous techniques learned in this class. The models used here were base models, Holt's Winter model, multiple feature linear regression model, and ARMA, ARIMA, or SARIMA model. The best model for the chosen dataset was the multiple feature regression model with an MSE of 0.145. Some of the challenges faced in this project were building a multiplicative model with high dimensions, waiting so much for the parameter estimation using LM algorithm, and estimating the orders of the non-seasonal and seasonal submodels. Furthermore, the data seems to be highly nonlinear, thus it was very difficult to build a multiplicative model that would pass the white test.