1. In the first part of the homework, it is told to prepare a regression model based on the data. The data is about traffic volume of a certain area and is divided into two as train and test data. The information before 2018-01-01 is considered train data. And the after is test which ends at 2018-09-30. The task is forecasting the 24 hours after the data. Since the forecast horizon is 24 hours, the usage of lagged variables are beneficial. On the other hand, independent variables can also be predicted for the target time. But some of the variables are not numerical. Which makes them nearly impossible to forecast with the present knowledge. This assumption makes the lagged variables used in the next steps. The data contains date, some weather information, holiday information and traffic volume. The information about weather has both numerical and verbal variables. The numerical ones are number of clouds and temperature. The first model goals to have an explanation of traffic through these two data.

Model 1:

Call:

lm(formula = traffic\_volume ~ lagtemp24 + lagcloud24, data = noholimess)

Coefficients:

(Intercept) lagtemp24 lagcloud24

-1607.789 16.981 1.941

As it can be seen both temperature and number of clouds have positive effect on the next day’s traffic. The intercept is low as the temperature is given in Kelvin. And the coldest weather in the data is about 240K.

Model 2:

This model adds weather descriptions given in the data to use weather data more efficiently.

Call:

lm(formula = traffic\_volume ~ lagtemp24 + lagcloud24 + lagwthrd24,

data = noholimess)

Coefficients:

(Intercept) lagtemp24

-1270.960 15.770

lagcloud24 lagwthrd24drizzle

4.114 -192.508

lagwthrd24few clouds lagwthrd24fog

305.677 -563.295

lagwthrd24freezing rain lagwthrd24haze

-1882.720 22.649

lagwthrd24heavy intensity drizzle lagwthrd24heavy intensity rain

-549.527 -281.097

lagwthrd24heavy snow lagwthrd24light intensity drizzle

-221.833 -227.581

lagwthrd24light intensity shower rain lagwthrd24light rain

-919.709 -157.439

lagwthrd24light rain and snow lagwthrd24light shower snow

101.394 1514.699

lagwthrd24light snow lagwthrd24mist

-179.621 -383.422

lagwthrd24moderate rain lagwthrd24overcast clouds

-301.970 -257.026

lagwthrd24proximity shower rain lagwthrd24proximity thunderstorm

-201.103 -166.606

lagwthrd24proximity thunderstorm with drizzle lagwthrd24proximity thunderstorm with rain

454.011 -162.678

lagwthrd24scattered clouds lagwthrd24shower drizzle

379.318 -2798.038

lagwthrd24shower snow lagwthrd24sky is clear

771.698 -63.772

lagwthrd24Sky is Clear lagwthrd24sleet

33.178 -2328.306

lagwthrd24smoke lagwthrd24snow

-391.329 -105.777

lagwthrd24SQUALLS lagwthrd24thunderstorm

-1269.076 -384.163

lagwthrd24thunderstorm with drizzle lagwthrd24thunderstorm with heavy rain

-2990.083 209.824

lagwthrd24thunderstorm with light drizzle lagwthrd24thunderstorm with light rain

-1384.347 -93.376

lagwthrd24thunderstorm with rain lagwthrd24very heavy rain

-740.332 51.840

Even though the weather description is one column. There are many different “values” it can take. And since there are no ordinal relationship between all. The model treats them as different dummy variables. The effect of each weather event can be observed further above. The effect of temperature stays same while the clouds’ effect slightly increases.

Model 3:

This model wishes to use seasonality in a way. The numbers are being used to represent time. But in reality there is no relationship as “*morning is bigger than noon*”. Which means the usage of time in a standard way can be dull. If the date is taken as class: date, the model sees them as increasing numbers. Which gives us trend in a way. However when the day, month and hour are taken as factors; it gives the relationship of them with the traffic.

Call:

lm(formula = traffic\_volume ~ day + month + hour, data = noholimess)

Coefficients:

(Intercept) day2 day3 day4 day5 day6 day7

613.815 -16.875 94.084 -80.150 -29.930 41.085 120.736

day8 day9 day10 day11 day12 day13 day14

116.967 65.011 11.111 -49.106 24.406 80.865 152.330

day15 day16 day17 day18 day19 day20 day21

93.646 -5.375 81.831 83.858 84.192 88.849 142.576

day22 day23 day24 day25 day26 day27 day28

36.541 -106.560 -121.933 -101.778 -91.686 -50.623 2.237

day29 day30 day31 month2 month3 month4 month5

-2.933 -44.769 93.455 174.072 288.504 334.516 284.870

month6 month7 month8 month9 month10 month11 month12

349.194 142.123 339.319 286.750 296.520 128.138 -69.919

hour1 hour2 hour3 hour4 hour5 hour6 hour7

-320.254 -448.688 -470.807 -143.092 1225.439 3309.145 3887.134

hour8 hour9 hour10 hour11 hour12 hour13 hour14

3756.719 3543.527 3350.917 3635.060 3888.197 3889.920 4112.796

hour15 hour16 hour17 hour18 hour19 hour20 hour21

4407.030 4827.269 4486.255 3434.907 2435.532 1992.365 1826.856

hour22 hour23

1362.999 626.649

The effect of “time” can be seen in the table above. The table contains one less than each periodical classification (12 months but only 11 in the table) as it classifies one of them as default.

Model 4:

This model uses the information of traffic from 24 hours ago.

Call:

lm(formula = traffic\_volume ~ lag24, data = noholimess)

Coefficients:

(Intercept) lag24

2063.6156 0.3668

Model 5 and 6 try to improve results by mixing components of different models.

Model 5:

Call:

lm(formula = traffic\_volume ~ lag24 + day + hour + month, data = noholimess)

Coefficients:

(Intercept) lag24 day2 day3 day4 day5 day6

507.73072 0.05085 -10.80460 101.14542 -76.01137 -15.75251 52.75100

day7 day8 day9 day10 day11 day12 day13

126.88956 115.55934 66.74709 22.22321 -41.64441 37.45611 83.96732

day14 day15 day16 day17 day18 day19 day20

160.40240 97.00110 1.52334 86.05084 87.84437 92.80964 96.44735

day21 day22 day23 day24 day25 day26 day27

148.18481 30.26516 -96.54175 -101.60616 -89.91585 -77.55018 -34.19227

day28 day29 day30 day31 hour1 hour2 hour3

12.79438 5.63531 -31.98320 108.57968 -313.57689 -446.46371 -474.86378

hour4 hour5 hour6 hour7 hour8 hour9 hour10

-163.29186 1177.19764 3220.93370 3790.93570 3659.16177 3454.96010 3265.56084

hour11 hour12 hour13 hour14 hour15 hour16 hour17

3548.33918 3798.61883 3798.89996 4019.21552 4310.38751 4729.47992 4396.82859

hour18 hour19 hour20 hour21 hour22 hour23 month2

3372.11640 2393.91147 1957.89382 1800.31982 1342.13473 617.33179 165.67935

month3 month4 month5 month6 month7 month8 month9

273.73631 317.04044 268.95807 330.22494 134.33094 321.63232 271.28752

month10 month11 month12

277.61944 121.03333 -68.90508

Model 6:

Call:

lm(formula = traffic\_volume ~ lag24 + day + hour + month + lagtemp24,

data = noholimess)

Coefficients:

(Intercept) lag24 day2 day3 day4 day5 day6

402.89015 0.05063 -11.12560 101.34315 -76.28565 -16.18748 52.43695

day7 day8 day9 day10 day11 day12 day13

126.35227 115.47026 66.53412 21.86472 -42.21311 37.02850 83.72805

day14 day15 day16 day17 day18 day19 day20

160.24949 96.66556 0.91531 85.16665 87.06976 92.35889 95.63549

day21 day22 day23 day24 day25 day26 day27

147.02394 29.76319 -97.10934 -101.91577 -90.54854 -77.86055 -34.26918

day28 day29 day30 day31 hour1 hour2 hour3

12.80690 5.34340 -32.14158 107.87393 -313.44009 -446.26328 -474.39821

hour4 hour5 hour6 hour7 hour8 hour9 hour10

-162.79414 1177.82795 3221.64985 3791.61675 3659.67175 3455.12263 3265.44886

hour11 hour12 hour13 hour14 hour15 hour16 hour17

3548.01671 3798.06533 3798.26178 4018.41706 4309.60765 4728.73521 4396.15838

hour18 hour19 hour20 hour21 hour22 hour23 month2

3371.44197 2393.36191 1957.50825 1800.07084 1342.05492 617.31957 165.49109

month3 month4 month5 month6 month7 month8 month9

270.83101 311.24581 260.28606 319.09566 122.30897 310.35690 260.70126

month10 month11 month12 lagtemp24

270.42719 116.71605 -69.89811 0.39996

The next part is testing the models with the test data. The most common one is comparing the mean squared errors of each model.

> rmse1

[1] 1700.945

> rmse2

[1] 1652.621

> rmse3

[1] 653.5972

> rmse4

[1] 1536.518

> rmse5

[1] 648.3279

> rmse6

[1] 648.2822

As it can be seen the best model is 6. With model 5 coming a close second. In these results it can be said that the main effect is coming from “seasonality” of days hours and months.

The prediction of Model 6 is:

1 2 3 4 5 6 7 8 9 10

820.5729 490.3269 355.9554 333.3714 663.1858 2022.6897 4108.4324 4723.0917 4632.9185 4446.3093

11 12 13 14 15 16 17 18 19 20

4276.0798 4564.7582 4806.5312 4806.7277 5023.0776 5314.2682 5732.6498 5392.5914 4358.2721 3359.4366

21 22 23 24

2884.7252 2695.7826 2201.6124 1451.7755

1. The second task is building an arima model upon train data and by that model, forecasting the same day as task 1. To build this model the trend and the seasonality of train data should be taken out. To do that decompose function can be used. After decomposing data, the arima models can be used to forecast.

> arimamodel1=arima(dtrdse,order = c(0,0,1))

> arimamodel1

Call:

arima(x = dtrdse, order = c(0, 0, 1))

Coefficients:

ma1 intercept

0.8321 0.2731

s.e. 0.0020 10.1470

sigma^2 estimated as 1234683: log likelihood = -339409.5, aic = 678825

> arimamodel2=arima(dtrdse,order = c(0,1,0))

> arimamodel2

Call:

arima(x = dtrdse, order = c(0, 1, 0))

sigma^2 estimated as 612873: log likelihood = -325304.3, aic = 650610.7

> arimamodel3=arima(dtrdse,order = c(1,0,0))

> arimamodel3

Call:

arima(x = dtrdse, order = c(1, 0, 0))

Coefficients:

ar1 intercept

0.9083 0.2711

s.e. 0.0021 41.5665

sigma^2 estimated as 584795: log likelihood = -324369.5, aic = 648744.9

> arimamodel4=arima(dtrdse,order = c(1,0,1))

> arimamodel4

Call:

arima(x = dtrdse, order = c(1, 0, 1))

Coefficients:

ar1 ma1 intercept

0.8548 0.4074 0.2695

s.e. 0.0027 0.0041 33.5861

sigma^2 estimated as 483494: log likelihood = -320541.2, aic = 641090.5

> arimamodel5=arima(dtrdse,order = c(5,0,1))

> arimamodel5

Call:

arima(x = dtrdse, order = c(5, 0, 1))

Coefficients:

ar1 ar2 ar3 ar4 ar5 ma1 intercept

2.0653 -1.4125 0.2609 0.1226 -0.0876 -0.7782 0.0098

s.e. 0.0097 0.0157 0.0140 0.0116 0.0053 0.0085 14.3224

sigma^2 estimated as 440667: log likelihood = -318674.7, aic = 637365.4

> arimamodel6=arima(dtrdse,order = c(5,0,1),xreg = noholimess$lag24)

> arimamodel6

Call:

arima(x = dtrdse, order = c(5, 0, 1), xreg = noholimess$lag24)

Coefficients:

ar1 ar2 ar3 ar4 ar5 ma1 intercept noholimess$lag24

2.0620 -1.3927 0.2449 0.1167 -0.0778 -0.7933 -427.1638 0.1310

s.e. 0.0108 0.0167 0.0140 0.0116 0.0054 0.0098 20.9389 0.0047

sigma^2 estimated as 432031: log likelihood = -318086.6, aic = 638191.2

The best model is Model 5 according to AIC value. Since the best model is decided, now it can be used for predicting the same day in task 1. Since the model is arima the test data is added to train data to make the dates same (01-10-2018).

Time Series:

Start = c(6, 4405)

End = c(6, 4428)

Frequency = 8760

[1] 750.4010 964.0571 711.6752 996.2565 1187.9932 1955.4801 3182.8126 4407.9499 5377.1635

[10] 5202.4909 4978.1436 4955.2149 4974.3441 4734.8931 4415.6877 3962.0947 3634.9960 3257.4184

[19] 2535.0103 2221.1249 2373.7222 3117.1428 3846.5639 3635.3797

1. The third and the final task asks for an ensembling strategy. Which requires all models to be calculated. And taking the average of them for each hour of the day.

ARIMA predictions:

> finalarima1

Time Series:

Start = c(6, 4405)

End = c(6, 4428)

Frequency = 8760

[1] 2331.819 3465.671 2844.527 2759.173 2551.420 2891.249 3691.439 4520.805 5147.564 4695.511 4263.637

[12] 4103.351 4052.263 3803.775 3528.300 3161.042 2951.811 2712.439 2137.921 1972.004 2264.464 3133.113

[23] 3968.374 3840.660

> finalarima2

Time Series:

Start = c(6, 4405)

End = c(6, 4428)

Frequency = 8760

[1] 509.697021 330.969555 -290.174726 -375.528082 -583.281667 -243.452386 556.738014

[8] 1386.103584 2012.862614 1560.809201 1128.935148 968.649463 917.561244 669.074132

[15] 393.598527 26.340422 -182.890753 -422.262192 -996.780628 -1162.697454 -870.237785

[22] -1.588299 833.673128 705.959087

> finalarima3

Time Series:

Start = c(6, 4405)

End = c(6, 4428)

Frequency = 8760

[1] 783.3229 854.3366 461.1338 583.8251 565.9561 1079.0950 2037.4670 3011.2066 3769.7373

[10] 3437.9533 3115.8504 3055.7541 3096.1098 2931.0845 2731.7853 2434.0543 2288.2812 2106.8287

[19] 1585.1735 1467.5054 1804.0023 2712.8450 3584.7912 3490.5598

> finalarima4

Time Series:

Start = c(6, 4405)

End = c(6, 4428)

Frequency = 8760

[1] 855.6662 1063.0730 774.5979 975.8516 1015.0206 1567.5831 2551.0511 3538.3172 4301.1135

[10] 3966.2613 3635.3604 3562.0670 3585.9258 3402.0084 3182.1621 2862.8306 2694.8901 2491.0921

[19] 1947.2215 1807.7089 2122.9169 3011.1648 3863.3111 3750.1440

> finalarima5

Time Series:

Start = c(6, 4405)

End = c(6, 4428)

Frequency = 8760

[1] 750.4010 964.0571 711.6752 996.2565 1187.9932 1955.4801 3182.8126 4407.9499 5377.1635

[10] 5202.4909 4978.1436 4955.2149 4974.3441 4734.8931 4415.6877 3962.0947 3634.9960 3257.4184

[19] 2535.0103 2221.1249 2373.7222 3117.1428 3846.5639 3635.3797

Linear Regression predictions

> value1

1 2 3 4 5 6 7 8 9 10 11

3322.918 3323.087 3319.691 3321.050 3324.616 3324.446 3326.484 3332.257 3316.042 3358.748 3338.287

12 13 14 15 16 17 18 19 20 21 22

3373.183 3380.824 3380.824 3357.816 3357.816 3366.986 3373.948 3363.929 3351.193 3368.598 3368.088

23 24

3357.220 3357.729

> value2

1 2 3 4 5 6 7 8 9 10 11

3258.895 3259.053 3255.899 3257.161 3260.472 3260.315 3262.207 3267.569 3474.875 3292.170 3495.534

12 13 14 15 16 17 18 19 20 21 22

3405.161 3412.257 3342.115 3312.566 3286.088 3364.746 3528.651 3361.907 3507.519 3301.316 3391.262

23 24

3290.750 3291.224

> value3

1 2 3 4 5 6 7 8 9 10

910.3354 590.0812 461.6471 439.5281 767.2433 2135.7740 4219.4809 4797.4696 4667.0546 4453.8624

11 12 13 14 15 16 17 18 19 20

4261.2527 4545.3956 4798.5326 4800.2552 5023.1317 5317.3651 5737.6041 5396.5906 4345.2424 3345.8675

21 22 23 24

2902.7005 2737.1914 2273.3345 1536.9842

> value4

1 2 3 4 5 6 7 8 9 10 11

2318.930 2197.142 2186.504 2226.489 2357.814 2494.642 2798.011 3120.823 3421.257 3548.914 3688.310

12 13 14 15 16 17 18 19 20 21 22

3731.596 3670.335 3670.335 3641.723 3641.723 3634.753 3579.361 3511.498 3363.298 3083.773 2855.604

23 24

2595.521 2413.572

> value5

1 2 3 4 5 6 7 8 9 10

820.7442 490.2840 355.9224 333.0654 662.8429 2022.3007 4108.0927 4722.8458 4632.7209 4446.2163

11 12 13 14 15 16 17 18 19 20

4276.1414 4564.9204 4806.7076 4806.9887 5023.3377 5314.5097 5732.6359 5392.3056 4358.1855 3359.4358

21 22 23 24

2884.6677 2695.4628 2201.2225 1451.1962

> value6

1 2 3 4 5 6 7 8 9 10

820.5729 490.3269 355.9554 333.3714 663.1858 2022.6897 4108.4324 4723.0917 4632.9185 4446.3093

11 12 13 14 15 16 17 18 19 20

4276.0798 4564.7582 4806.5312 4806.7277 5023.0776 5314.2682 5732.6498 5392.5914 4358.2721 3359.4366

21 22 23 24

2884.7252 2695.7826 2201.6124 1451.7755

The average is:

Time Series:

Start = c(6, 4405)

End = c(6, 4428)

Frequency = 8760

1 2 3 4 5 6 7 8 9 10 11

1516.664 1548.007 1312.489 1350.022 1433.935 2046.375 3076.565 3711.676 4068.483 3855.386 3677.957

12 13 14 15 16 17 18 19 20 21 22

3711.823 3772.854 3668.007 3603.017 3516.194 3541.496 3346.269 2773.416 2417.491 2374.604 2701.461

23 24

2910.579 2629.562