

Final Assignment

Forecasting – June 2021

I. Executive summary

As the world becomes more digitalized, forecasting techniques and data insights ought to be more greatly utilized and examined so organization may make informed and educated estimates that predict the future levels, trends, and seasonality based on historical data. As a result, this report is designed to explore a variety of statistical analysis and forecasting methods to help the stakeholders to better navigate and determine the unknown parameters, performance measurements, and the rationale behind the decisions.

II. Introduction

This report uses M3 data from the Makridakis forecasting competitions, which contains is a list of 3003 time series data with different time intervals (yearly, quarterly, monthly, etc) and categories (Micro, Macro, Demographic, Finance, Industrial, etc). On top of it, the report consists of two parts: manual modeling and batch modeling.

In the manual modeling section, we examine the quarterly employment data of Finland from 1978 to 1996, explore the characteristics and patterns of such dataset including trends, levels, seasonality, and error-proneness, conduct model validation tests, measure forecasting accuracy, and ultimately justify what's the best forecasting model for this specific dataset among regression, exponential smoothing, or autoregressive integrated moving average models (ARIMA).

In the batch modeling section, 70 sets of quarterly time series data are examined altogether. After giving a short description of the datasets, this section aims to produce forecasts using automatic exponential smoothing model, automatic ARIMA model, and multiple aggregation prediction algorithm selection models, evaluate and audit the models using a defined model selection strategy, and eventually evaluate a combination strategy consisting the above-mentioned selection methods using some numbers of benchmark forecasting models to justify the efficacy of such combined method strategy.

III. Manual Modeling

Data exploration and observations:

First let's examine the quarterly employment data of Finland from 1978 to 1996, which consists of a total of 62 quarterly data points/periods. Data before 1993 is of in-sample as training data, whereas data from 1993/Q4 to 1995/Q3 is of out-of-sample for future validation and performance measurements. By looking at the visualization of data distribution in Figure 1, we can see that the trend peaked just below 6,000 in 1982, and dipped into 5,000 level till 1990, and declined more drastically in 1990s.

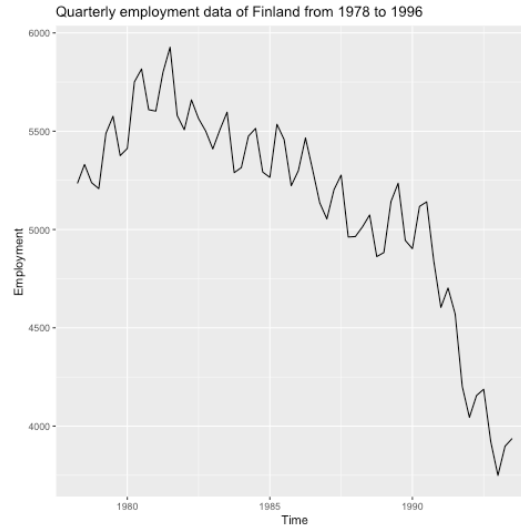


Figure 1. Quarterly employment data of Finland

Just to add more details to the background, upon brief research, the declining employment rate is due to “unavoidable developments in the international economy, such as the collapse of exports to the Soviet Union, the fall in the terms of trade, and the rise in the interest rates in Europe after the German unification. Domestic economic policies also contributed to the adverse development. Real interest rose to close to 15% and real asset prices fell creating problems first to the highly indebted private sector firms, and, eventually, to the banking sector. Due to adverse macroeconomic shocks, job destruction was rapid during the first years in the 1990’s. The inflow to unemployment rose by 60% compared to the prerecession level. The recession in the 1990’s was also associated with a rapid re-structuring of the economy”¹. Any model will need to take all these observations into account in order to effectively forecast the employment rate of Finland in the future. To extract more insights statistically, let us examine the time plot, the autoregression, and the components of the series.

First let’s look at the time plot. From 1980 to 1990, the employment data has some level changes and a clear downtrend with consistent seasonality, or a relatively consistent rise and fall. However, the pattern changes after 1990 as the downtrend becomes more drastic. This suggests possibly a multiplicative seasonality here: as the level decreases, so does the seasonality effect. During the time period, from measures of central tendency table (Table 1), the employment data peaked at 5,927 and bottomed at 3,749 in Quarter 3 of 1991 and Quarter 1 of 1993 respectively.

Min.	1 st Qu.	Median	Mean	3 rd Qu.	Max.
3,749	4,949	5,252	5,127	5,498	5,927

Table 1. Measure of central tendency of employment data

Next, to examine if past behaviors or phenomenon have impact on future activities, we can take a look at the autoregressive model of the dataset to determine if there are some correlations between values in a time series and the values that precede and succeed them. In figure 2, positive linear patterns are shown, which suggest positive autocorrelations are present. In other words, preceding employment data plays a strong role in the succeeding employment during the period.

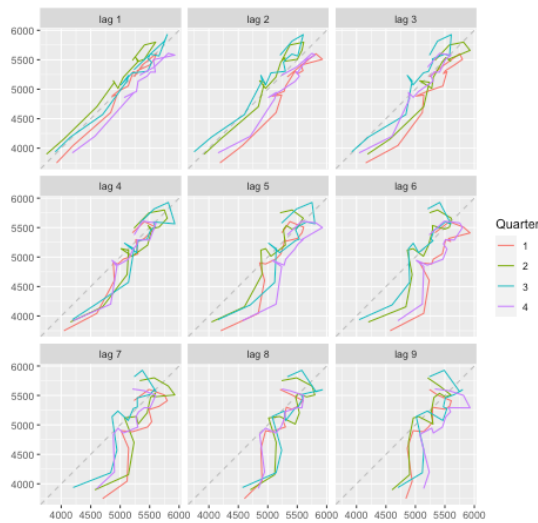


Figure 2. Lagged scatterplots for quarterly employment data

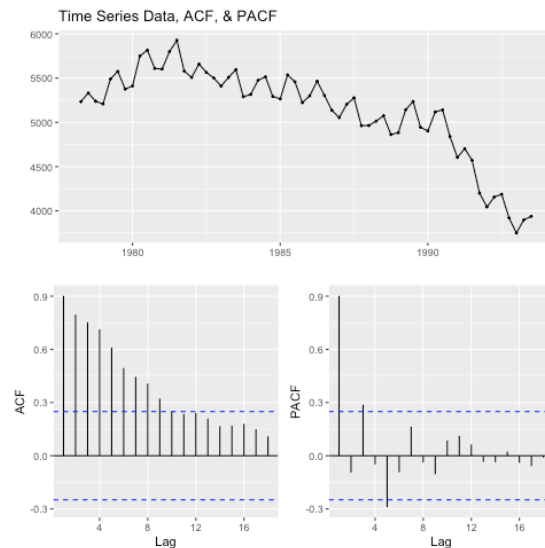


Figure 3. TS data, ACF, & PACF plo

To probe more into the autoregressive nature of the time series data, let's look at figure 3. By first examine the autocorrelation function (ACF) plot, we can see a slowly decaying of autocorrelation coefficients, which suggests that means future values of the time series are correlated, or heavily affected by past value, suggesting the time series is autoregressive and non-stationary. The partial ACF (PACF) plot also suggests something similar, as we see spikes at lag 3 and lag 5 beyond the standard error lines, indicative of certain degrees of association between lag 0 and lag 3 and 5. In short, to ensure the mean of the time series can be stabilized, further data transformation such as differencing or log transformation should be taken into consideration when fitting forecasting models.

Time Series Components:

Before we move on to finding an adequate forecast model, let us look at the component of the time series. To briefly recap, decomposition of time series may be described as the tool to remove noise(E), isolate overall trend (T), and identify seasonal characteristics(S). The formula may be written as $Y_t = T_t + S_t + E_t$, or $Y_t = T_t \times S_t \times E_t$ if it's a multiplicative model.

As observed visually from time series data, there's an overall downtrend with fluctuating seasonality, so we may assume this is a multiplicative model. To further identified the trend component, we may identify using the moving average. And as you can see from Figure 4, by using a moving average of length 4(because the dataset is a quarterly data) and length 8 for benchmarking, the results of these moving averages yield the overall trend of the data with higher degrees of smoothing applied on the data.

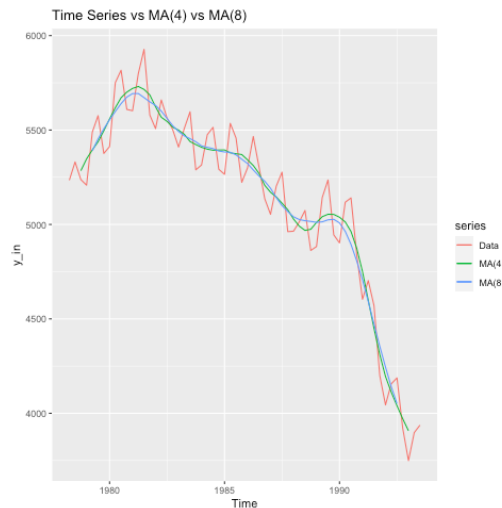


Figure 4. TS vs MA(4) vs MA(8)

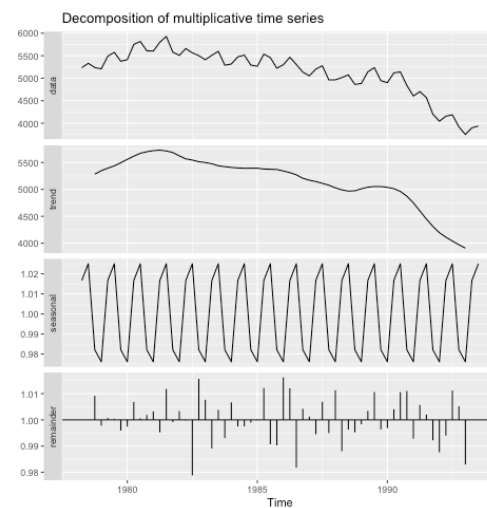


Figure 5. Decomposition of dataset

To observe the remaining two components, seasonality and noise, we may see figure 5 where all components are presented. As suggested, the seasonality seems to be constant over time. This is however due to the assumption behind classical decomposition where there's deterministic seasonality, or the seasonal components do not change overtime. Whereas for the noise component, there's no conspicuous trend or pattern, so we cannot conclude that the noise components are completely random. More arrangements will likely need to be done for future forecast model fittings.

To bring in another concept when discussing time series components, a time series is said to be non-stationary if the mean value of time-series, the variance, and the seasonality fluctuates. In the case of our observed time series, the above conditions seem to apply. In other words, the time series is said to be non-stationary.

Model Selections and Fitting:

In order to determine the best efficacy and the performance of the models are produced within and across different forecasting methods, **Mean Absolute Percentage Error (MAPE)** will be studied against the out-of-sample (future) data to determine which method produces the best accuracies and the smallest errors percentage wise.

1. Exponential Smoothing Methods

Four exponential smoothing models, ETS(error, trend, seasonality), are chosen among the exponential smoothing families, and here are their differences and justifications.

- **Holt's Exponential Smoothing with linear trend, ETS(A,A,N):** The method allows the smoothing of level and trend components, thus the forecast function will be trending instead of flat, which is justifiable considering the overall downtrend of the time series data being discovered.
- **Damped Exponential Smoothing, ETS(A,Ad,N):** Similar to the previous method, but a damping parameter ϕ is applied on the trend component to prevent overfitting of data.
- **Holt-Winters with additive seasonality, ETS(A,A,A):** The combination of the first method and seasonal exponential smoothing method where it is capable of handle seasonality component. Since previously we are uncertain whether seasonality is additive or multiplicative, we will consider seasonality component to be additive for this method, and verify the performance.
- **Holt-Winter's Exponential Smoothing with Damped Trend, ETS(M,Ad,M):** Similar to Holt-Winters with additive seasonality, except the seasonality and the error components are changed to multiplicative.

As computed, table 2 contains the models, information of the damping parameters, information criteria (AIC), and MAPE metrics of the four chose methods. The results seem to suggest that, although the information criteria shows that Holt-Winter's Exponential Smoothing with Damped Trend has better model quality, Damped Exponential Smoothing method produces the best accuracy as it produces the smallest MAPE value against the out-of-sample data.

Models	Damping	Damping Parameter ϕ	AIC	MAPE (%)
ETS(A,A,N)	-	N/A	902	9.55
ETS(A,A,N)	V	0.814	902	4.39
ETS(A,A,A)	-	N/A	846	10.5
ETS(M,A,M)	V	0.924	834	6.64

Table 2. Exponential Smoothing Methods Comparison Table

2. ARIMA Methods

One ARIMA model is being chosen upon manual ARIMA modelling. Firstly, Ljung-Box portmanteau test is conducted, suggesting further time series differencing is required due to the time series being autocorrelated ($p\text{-value} = 3.472e-13$). After one first differencing and one seasonal differencing using the Augmented Dickey-Fuller(ADF) unit root tests, we've finally reached the appropriate numbers of differencing with a significantly small $p\text{-value}$ suggesting the differenced time series data is finally stationary (See Figure 6). On top of it, the significant spike at lag 4 in the differenced time series' ACF plot (Figure 7) suggests a seasonal MA(1) component. In other words, the computations above suggests $ARIMA(0,1,0)(1,1,0)[4]$ is the best ARIMA model. Upon examination, the ARIMA model produced a MAPE of 9.55% against out-of-sample data.

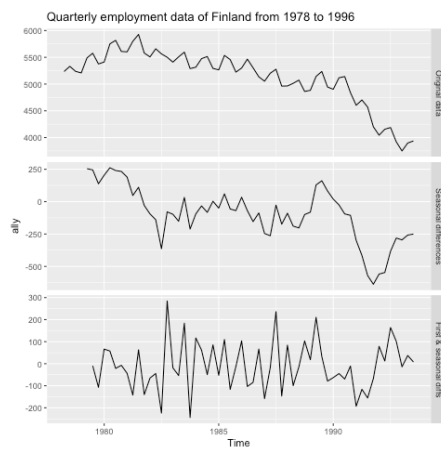


Figure 6. Differencing Process

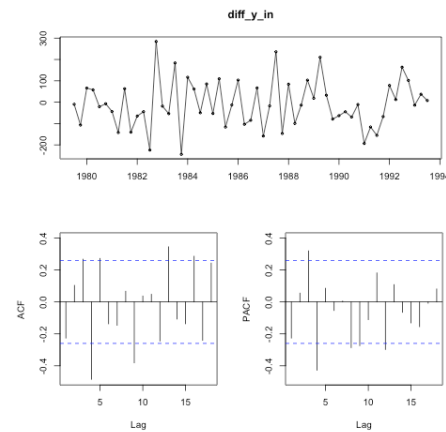


Figure 7. ACF and PACF plots of the double difference time series data

3. Aggregation Methods

Lastly, the Dynamic Optimized Theta model is selected as the two theta lines produced are suitable for long-term trend component and short-term behaviour². By computing numbers of times to grasp the picture of the forecast distribution using the power of the Theta model, the model produced a MAPE value of 4.82%.

Selected Model and prediction intervals

Models	MAPE (%)
ETS(A,Ad,M)	4.39
ARIMA(0,1,0)(0,1,0)[4]	9.55
Dynamic Optimized Theta Model	4.82

Table 3. Model Performance Comparison (h=8)

As suggested by Table 3, damped exponential smoothing seems to be the best model to produce forecasts with the best accuracy against the out-of-sample. Figure 8 references the original Time Series data of the quarterly employment mentioned in the beginning, and figure 9 contains the three chosen models' forecasts, prediction intervals for the next quarter.

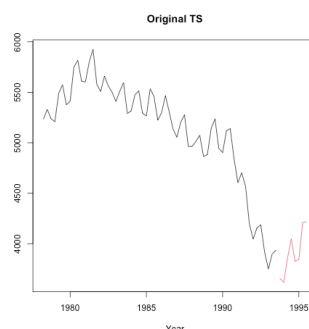


Figure 8. Original Time Series data

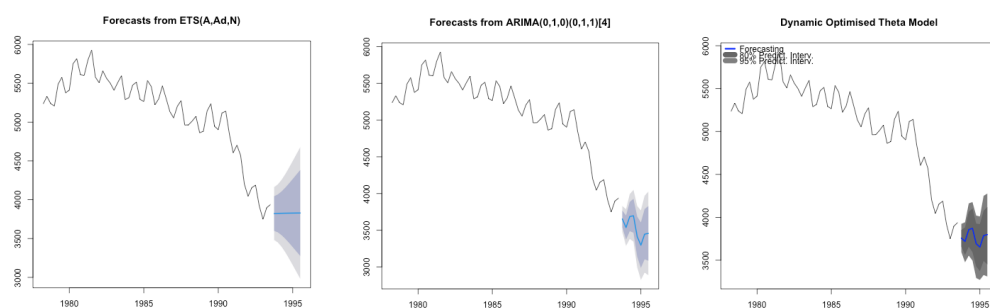


Figure 8. Model forecasts of three selected methods

IV. Batch forecasting

To briefly recap, in this section, 70 quarterly time series datasets are examined altogether using three model section methods: automatic exponential smoothing model, automatic ARIMA model, and multiple aggregation prediction algorithm model (MAPA). A model selection strategy is then implemented to select the most suitable model for each time series in order to produce the most accurate forecasts. Ultimately, a model combination strategy is incorporated, and later on measured its forecast across all 70 time series datasets against three benchmark methods to verify its performance against out-of-sample data.

Data exploration and observations:

Upon exploration, there are 70 quarterly time series data in total to be examined. Among the time series data, categories of quarterly data are included in relation to demographic, finance, industry, macroeconomy, and microeconomy.

	Demographic	Finance	Industry	Macro	Micro
Quarterly	5	8	8	34	15

Table 4. Summary of 70 quarterly time series data

Model selection strategy:

Automatic exponential smoothing model, automatic ARIMA model, and MAPA model are able to automatically select the best forecasting parameters that produce optimal models among each method. In order to choose the best model among the above three methods, one error measure, **mean absolute error (MAE)**, is chosen as benchmark for model selection as it is commonly used to measure the accuracy of the in-sample forecast. Once the model is chosen for each time series, three other error measures, **mean percentage error (MPE)**, **mean absolute percentage error (MAPE)**, and **mean absolute scaled error (MASE)**, are measured and averaged against the out-of-sample data. MPE is chosen to help indicate if the forecasts are biased, MAPE is chosen to help determine forecast accuracy, and MASE is chosen to offers scale-independent measure in comparison to Naïve model as benchmark.

Models	Count	MPE (%)	MAPE (%)	MASE
ARIMA	42	-1.8	7.09	.887
ETS	20	1.33	11.1	1.53
MAPA	8	-3.73	20.1	2.25

Table 5. Models' error measures of 70 time series data against future data (h=8)

As shown in table 5, when measuring against the in-sample data of the 70 time series data, 42 out of 70, or 60% of the time series produce the smallest MAE values when utilizing automatic ARIMA models, 20 out of 70, or 28.6% of the time series produce the smallest MAE values when utilizing automatic exponential smoothing models, and only 8 out of 70, or 11.4% of the time series produce the smallest MAE values when utilizing MAPA models. To further examine the accuracies of the models, forecast horizon of 8 quarters/periods is arranged since it equates to the length of the out-of-sample data. And as shown on Table 5, the MPE, MAPE, and MASE error measures against the out-of-sample data are calculated for each model.

If we look at the model performances individually, we can see that the automatic ARIMA model's average MPE value is -1.8%, which indicates that the bias is uptrend, and the model tends to over-forecast by 1.8%. On top of it, the automatic ARIMA model is the

most accurate among the three as it only produces an average of 7.09% of forecast error against the out-of-sample data. Lastly, when measuring against the Naïve method where the forecasts equate to the last known value of the time series, the MASE value of 0.887 “implies that actual forecast performance is better than a Naïve method”.

With the same logic, we can draw similar insights for other models. The automatic exponential smoothing model tends to under-forecast by 1.33%, produces an average of 11.1% of forecast error against the out-of-sample data, and has worse forecast performances against Naïve method, whereas the MAPA method tends to over-forecast by 3.73%, produces 20.1%, a relatively large average forecast error value against out-of-sample data, and also has worse forecast performances against Naïve method.

Combination Strategy and benchmark performances:

One combination strategy and three benchmark models are chosen to further examine whether combining forecasting models may produce better results. Since automatic ARIMA and exponential smoothing models combine for nearly 90% of the forecast fits, not to mention the two models produce relatively lower MAPE and MASE values compare to the MAPA model, thus I will combine automatic ARIMA and exponential smoothing models to create a hybrid model that tests its combined efficacy against out-of-sample data for a forecasting horizon of 8 periods, while applying combination weights based on in-sample errors.

On the other hand, the three benchmark models chosen are the Naïve method, damped exponential model, and Theta model. As for why the three methods are chose, Naïve method is chosen to see how much better the combination strategy performs, the damped exponential model is the best performing exponential smoothing model in the previous section, and the Theta model is often used as a robust benchmark method for supply chain management and other empirical forecasting practices.

By looking at Table 6, the forecast of all 70 time series data using hybrid model is measured against the three mentioned benchmark methods on the three previously utilized error metrics, MPE, MAPE, and MASE. As we can see, the combination tends to over-forecast (by 1.33%), while other methods tend to under-forecast. As for the performance accuracy against the out-of-sample data, the damped exponential smoothing method generates the smallest percentage error, though the MAPE values of the four methods are in the narrow range of 8~11%. Lastly, when looking at the four MASE ratio that are larger than 1, implication is drawn that actual forecasts of all four models actually do worse than a naïve benchmark forecasting method calculated in-sample. However, combination strategy and damped ES models seem to perform the best.

Models	MPE (%)	MAPE (%)	MASE
Combination Strategy	-1.33	9.65	1.17
Naïve	1.69	10.9	1.58
Damped ES	2.39	8.64	1.16
Theta	1.51	9.24	1.22

Table 6. Combination strategy vs Naïve/Damped ES/Theta methods

V. Conclusions and Implications

Several conclusions may be drawn from this report. As shown throughout the report, every stage of time series data analysis is crucial, interconnected, and cannot be taken lightly. From singular time series data exploration, data decomposition, model consideration, analysis, forecasting, all the way to batch modelling strategy selection and performance justifications, so many components come into play in order to help stakeholders to find the optimal forecasting model. Let's first look at the analysis of singular time series data.

When first looking at a singular time series data like the quarterly employment data of Finland from 1978 to 1996, it is crucial to first examine the plot nakedly, inspect the autoregression of time series data, and the components of the series. Some chart actions may let us determine the distribution, trend, seasonality, cycle, level, and stationariness of data at first glance. By doing so, it helps us navigate the type of graphs to be produced, the statistical tests to be run, the type of data transformation to be computed in later stages. By looking at the plot of the employment data, we immediately are able to tell there's a significant downtrend, changing seasonality and level, subsequently implying this time series data is non-stationary.

Next, when conducting forecasting analysis, a time series assumes that the mean, variance, and autocorrelation of the series is constant. Otherwise, the model has no basis upon which to forecast future values. autocorrelation within time series data may be examined by looking at lagged scatter plot, ACF, and PACF plots. A randomly distributed lagged scatterplot suggests there's no strong correlation between each time series data, and a linearly correlated lagged scatterplot suggests otherwise. The phenomenon of autoregression may also be observed using ACF plot and PACF plot. Should autoregression to be found in time series data, further time series data transformation methods such as differencing or log-transformation may be applied. After autoregression is found, the noise, the trend, and the seasonal components of the time series data may be isolated by performing time series decomposition, whether the trend or seasonality to be additive or multiplicative.

Moving onto model selection and fitting, we are to fit the in-sample time series data to create different forecasting models, but how do we determine which model is optimal? One performance metric that is often utilized is the mean absolute percentage error, or MAPE, which provides decision makers an indicator of how accurate the model estimates future data against actual future data. The smaller the MAPE, the more accurate the model is. Other model quality benchmark may be utilized on other occasions, such as finding minimized information criteria (AIC, AICc, BIC). As for model selection among different forecasting methods, which are presented like the exponential smoothing families, ARIMA models, and others, MAPE can be viewed as a great indicator when comparing performances across different forecasting methods. In terms of the employment data analysis presented in the report, exponential smoothing with damped trend model is the best fit as it creates the smallest MAPE value. Forecast and its prediction intervals are then proceeded to be produce.

The second section, “batch modelling”, explore how automatic forecasting models, selection and combination strategies perform against a group of time series data, or 70 quarterly time series datasets from a variety of categories in our case. A strategy that is often used to determine the forecasting model is by simply examining the in-sample error, or the error rate you get on the same data set used to build the predictor.

In terms of the selection strategy, in this particular section, MAE is used as benchmark measure for the automatic exponential smoothing, automatic ARIMA, and MAPA models fitting. Each of the 70 time series datasets is fitted to the model that produced the smallest MAE, and at the end, 60% of the time series data are fitted using automatic ARIMA models, 28% of the time series data are fitted using automatic exponential smoothing data, and the remaining time series data are fitted using the MAPA models. On top of it, forecasts are generated using the 70 associated out-of-sample datasets, and later cross-validated using the three performance measures, mean percentage error, mean absolute percentage error, and mean absolute scaled error. These three metrics can help management to navigate models’ bias, accuracy, and benchmark performances against the Naïve method.

Regarding the combination strategy, several methods are available for the management based on specific criteria, such as cross-sectional aggregation, temporal aggregation, and many more. Since automatic ARIMA and exponential smoothing models are selected for nearly 90% of the time, on top of producing relatively smaller MASE, the two models are combined and measured against three benchmark models, Naïve, Damped ES, and Theta models. The best models can again be determined by examining the three mentioned error measures.

Practical applications of time series forecasting and model fitting varies depends on countless numbers of elements, the characteristics of the time series data, the duration of observation, the presence of missing data and outliers, not to mention the external influences such as geopolitical impacts, regulation changes, and many more.

Though this report covers only a small portions of time series data exploration exercises, forecasting methodologies, and performance measurement practices, these techniques serve as the fundamentals and crucial tools to help individuals and enterprises to make informed and data-drive business decisions.

VI. References

1. E. KOSKELA, R. UUSITALO, 2003. *THE UN-INTENDED CONVERGENCE: HOW THE FINNISH UNEMPLOYMENT REACHED THE EUROPEAN LEVEL* [Online]. CESIFO Working Paper No. 878. Available from: https://www.econstor.eu/bitstream/10419/76301/1/cesifo_wp878.pdf [Accessed Aug, 23rd, 2021]
2. J. A. Fioruccia, T. R. Pellegrini, F. Louzada, F. Petropoulos, A. B.Koehler, 2016. *Models for optimizing the theta method and their relationship to state space models* [Online]. Scient Direct. Available from : <https://towardsdatascience.com/time-series-forecasting-with-deep-learning-and-attention-mechanism-2d001fc871fc> [Accessed Aug, 23rd, 2021]
3. Mohamed, 2017. *MASE – Mean Absolute Scaled Error* [Online]. Available from: <https://support.numxl.com/hc/en-us/articles/115001223523-MASE-Mean-Absolute-Scaled-Error> [Accessed Aug, 23rd, 2021]

VII. Appendices

Part 1 – Manual Modelling

```
library(Mcomp)
library(ggplot2)
library(fpp)
library(forecast)
library(smooth)
library(stats)
library(tibble)
library(dplyr)
library(nlme)
library(forecTheta)
```

```
M3[[1357]]$x
```

```

# Part 1
# My student ID is 209572823

y_raw <- M3[[1357]]

autoplot(y_raw)
str(y_in)
summary(y_in)

# Create in-sample / out-sample subset
y_in <- y_raw$x
y_out <- y_raw$xx
y_out
autoplot(y_in,
  xlab = "Time", ylab = "Employment",
  main="Quarterly employment data of Finland from 1978 to 1996"
)

summary(y_in)

# autoregression
gglagplot(y_in)

ggtsdisplay(y_in, main="Time Series Data, ACF, & PACF")

# Classical decomposition

# MA approach
length(y_in)
k4 <- 4
k8 <- 8
m4 <- k4 %/% 2
m8 <- k8 %/% 2

MA4 <- array(NA, length(y_in))
MA8 <- array(NA, length(y_in))

# MA4
for (i in (m4 + 1):(length(y_in) - m4)){

  MA4[i] = mean(c(y_in[(i-m4):(i+m4-1)], y_in[(i-m4+1):(i+m4)]))

}

# MA8
for (i in (m8 + 1):(length(y_in) - m8)){

```

```

    MA8[i] = mean(c(y_in[(i-m8):(i+m8-1)], y_in[(i-m8+1):(i+m8)]))
  }

MA4 <- ts(MA4, start=c(1978,2), frequency=4)
MA8 <- ts(MA8, start=c(1978,2), frequency=4)

autoplot(y_in, series = "Data", main="Time Series vs MA(4) vs MA(8)") +
  autolayer(MA4, series="MA(4)") +
  autolayer(MA8, series="MA(8)")

# Decomposition
dym <- decompose(y_in, type = "multiplicative")

autoplot(dym)

plot(dym)

checkresiduals(y_in)

nsdiffs(dy$seasonal)

adf.test(y_in)
kpss.test(y_in)

time(y_in)

# ETS

# Compare ES models
# 1. Holt's Exponential Smoothing with linear trend
# 2. Damped Exponential Smoothing
# 3. Holt-Winters with additive seasonality
# 4. Holt-Winter's Exponential Smoothing with Damped Trend

no_ETS_models <- 4

models <- c("AAN", "AAN", "AAA", "MAM")
damping <- c(FALSE, TRUE, FALSE, TRUE)

```

```

ets(y_in, model="MAM", damped = TRUE)$par['phi']

MAPEs_ETS <- array(NA, no_ETS_models)
phi_ETS <- array(NA, no_ETS_models)
aic_ETS <- array(NA, no_ETS_models)

for (m in 1:no_ETS_models){

  fit <- ets(y_in, model=models[m], damped=damping[m])
  aic_ETS[m] <- fit$aic
  fcs <- forecast(fit, h=8)$mean
  MAPEs_ETS[m] <- accuracy(fcs, y_out)[5]
  phi_ETS[m] <- fit$par['phi']
  print(m)
}

ETS_Comparison <- tibble(model = models, damping = damping,
  phi = phi_ETS, AIC =aic_ETS, MAPE = MAPEs_ETS)

ETS_Comparison

forecast(ets(y_in, model="AAN", damped=TRUE),h=8)
autoplot(forecast(ets(y_in, model="AAN", damped=TRUE),h=8))

# ARIMA

tsdisplay(y_in)
# No variation, no need for box-cox

Box.test(y_in, lag=1, type="Ljung")

tsdisplay(diff(y_in, 4))
tsdisplay(diff(diff(y_in, 4), 1))

diff_y_in <- diff(diff(y_in,4),1)

adf.test(diff_y_in)
tsdisplay(diff_y_in)

# OR ndiffs / nsdiffs
nsdiffs(y_in)
ndiffs(diff(y_in,4))

ndiffs(diff(diff(y_in,4),1))

```



```

ally <- cbind("Original data" = y_in,

             "Seasonal differences" = diff(y_in, 4),

             "First & seasonal diffs" = diff(diff(y_in, 4),1))

autoplot(ally, facets=TRUE) +

  xlab("Time") +

  ggtitle("Quarterly employment data of Finland from 1978 to 1996")

fit_arma <- Arima(y_in, order = c(0,1,0), seasonal = c(0,1,1))

accuracy(forecast::forecast(fit_arma,h=8), y_out)

checkresiduals(fit_arma)

plot(forecast(test_fit_ACF,h=8))

# Theta Method

fit_theta <- dotm(y_in,h=8,level=c(80,95))

fcs_theta <- accuracy(fit_theta$mean, y_out)

plot(fit_theta)
accuracy(fcs_nn, y_out)

# Show Down of the three model

fcs_ETS_final <- forecast(ets(y_in, model="AAN", damped=TRUE),h=8)
fcs_ARIMA_final <- forecast::forecast(fit_arma, h=8)
fcs_Theta_final <- fcs_theta

final_model <- c("ETS","ARIMA","Theta")
final_MAPE <- c(accuracy(fcs_ETS_final, y_out)[2,5],
               accuracy(fcs_ARIMA_final, y_out)[2,5],

```

```

    fcs_Theta_final[5])

final_performance <- tibble(Model = final_model, MAPE= final_MAPE)

final_performance

plot(y_raw, main = "Original TS")
plot(fcs_ETS_final)
plot(fcs_ARIMA_final)
plot(fit_theta)

```

Part 2 – Batch Forecasting

```

library(Mcomp)
library(ggplot2)
library(fpp)
library(forecast)
library(base)
library(MAPA)
library(forecastHybrid)
library(Metrics)
library(stats)
library(dplyr)
library(tibble)
library(gtools)

origin <- 701:1400

# Abstract ID that ends with 3
array <- NA

for(i in origin){
  if(i%10 == 3){
    array[which(i==origin)] <- i
  }
}

series_index <- data.frame(array)[!is.na(data.frame(array)$array),]

# Examine 70 series

M3[series_index]

```

```

# Create empty array including selected model and three error measures

final_array <- array(NA, c(length(series_index),4))

final_array

# Across all 70 time series
for (i in 1:length(series_index)){
  y_in <- M3[[series_index[i]]]$x
  y_out <- M3[[series_index[i]]]$xx

  fit_ets <- ets(y_in)
  fit_arima <- auto.arima(y_in)
  fit_mapa <- mapa(y_in, fh = 8, conf.lvl=c(0.8, 0.95))

  ets_in_MAE <- forecast::accuracy(fit_ets)[3]
  arima_in_MAE <- forecast::accuracy(fit_arima)[3]
  mapa_in_MAE <- fit_mapa$MAE

  temp_MAE <- c(ets_in_MAE, arima_in_MAE, mapa_in_MAE)

  if(temp_MAE[which.min(temp_MAE)]==ets_in_MAE){

    final_array[i, 1] <- "ETS"
    final_array[i, 2] <- forecast::accuracy(forecast::forecast(fit_ets, h =
8), y_out)[2,4]
    final_array[i, 3] <- forecast::accuracy(forecast::forecast(fit_ets, h =
8), y_out)[2,5]
    final_array[i, 4] <- forecast::accuracy(forecast::forecast(fit_ets, h =
8), y_out)[2,6]

  } else if(temp_MAE[which.min(temp_MAE)]==arima_in_MAE){

    final_array[i, 1] <- "ARIMA"
    final_array[i, 2] <- forecast::accuracy(forecast::forecast(fit_arima, h
= 8), y_out)[2,4]
    final_array[i, 3] <- forecast::accuracy(forecast::forecast(fit_arima, h
= 8), y_out)[2,5]
    final_array[i, 4] <- forecast::accuracy(forecast::forecast(fit_arima, h
= 8), y_out)[2,6]

  } else if(temp_MAE[which.min(temp_MAE)]==mapa_in_MAE){

    final_array[i, 1] <- "MAPA"
    final_array[i, 2] <- forecast::accuracy(fit_mapa$outfor, y_out)[4]
    final_array[i, 3] <- forecast::accuracy(fit_mapa$outfor, y_out)[5]
    final_array[i, 4] <- mase(y_out, fit_mapa$outfor)
  }
}

```

```

    }

}

# Checkpoint
final_array

# Create tibble form, rename column, and convert error measures
# into numeric type, then calculate average error measures

temp_tibble <- final_array %>% as_tibble()
names(temp_tibble) <- c("Model", "MPE", "MAPE", "MASE")

temp_tibble$MPE <- as.numeric(temp_tibble$MPE)
temp_tibble$MAPE <- as.numeric(temp_tibble$MAPE)
temp_tibble$MASE <- as.numeric(temp_tibble$MASE)
temp_tibble

temp_group <- group_by(temp_tibble, Model)

error_table <- temp_group %>% summarise(
  MPE = mean(MPE),
  MAPE = mean(MAPE),
  MASE = mean(MASE)
)

# Keep count of the number of model chosen

cnt <- count(temp_tibble, Model)
error_table <- error_table %>% add_column(d=cnt['n'], .after = "Model")

names(error_table) <- c("Model", "Count", "MPE", "MAPE", "MASE")

error_table

# Combination Strategy and benchmark performance comparison

hybrid_errors <- array(NA, c(length(series_index),3))
naive_errors <- array(NA, c(length(series_index),3))
HES_errors <- array(NA, c(length(series_index),3))
Theta_errors <- array(NA, c(length(series_index),3))

for(i in 1:length(series_index)){

```

```

y_in <- M3[[series_index[i]]]$x
y_out <- M3[[series_index[i]]]$xx

# Hybrid + Naive + Holts ES(Linear Trend)
hybrid_fit <- hybridModel(y_in, models="ae", weights = "insample.errors")
naive_fit <- naive(y_in, h=8, level=c(80,85,90,95,99))
HES_fit <- ets(y_in, model="MAM", damped=TRUE)
Theta_fit <- thief(y_in, h = 8, comb="struc", usemodel="theta")

# Generate Forecast
fcs_hybrid <- forecast::forecast(hybrid_fit, h=8)
fcs_naive <- forecast::forecast(naive_fit, h=8)
fcs_HES <- forecast::forecast(HES_fit, h=8)
fcs_Theta <- forecast::forecast(Theta_fit, h=8)

# Calculate out-sample model accuracy (MPE, MAPE, MASE)
hybrid_errors[i, 1] <- forecast::accuracy(fcs_hybrid, y_out)[2,4]
hybrid_errors[i, 2] <- forecast::accuracy(fcs_hybrid, y_out)[2,5]
hybrid_errors[i, 3] <- forecast::accuracy(fcs_hybrid, y_out)[2,6]

naive_errors[i, 1] <- forecast::accuracy(fcs_naive, y_out)[2,4]
naive_errors[i, 2] <- forecast::accuracy(fcs_naive, y_out)[2,5]
naive_errors[i, 3] <- forecast::accuracy(fcs_naive, y_out)[2,6]

HES_errors[i, 1] <- forecast::accuracy(fcs_HES, y_out)[2,4]
HES_errors[i, 2] <- forecast::accuracy(fcs_HES, y_out)[2,5]
HES_errors[i, 3] <- forecast::accuracy(fcs_HES, y_out)[2,6]

Theta_errors[i, 1] <- forecast::accuracy(fcs_Theta, y_out)[2,4]
Theta_errors[i, 2] <- forecast::accuracy(fcs_Theta, y_out)[2,5]
Theta_errors[i, 3] <- forecast::accuracy(fcs_Theta, y_out)[2,6]
}

# Change to tibble and rename column name
hybrid_errors <- hybrid_errors %>% as_tibble()
naive_errors <- naive_errors %>% as_tibble()
HES_errors <- HES_errors %>% as_tibble()
Theta_errors <- Theta_errors %>% as_tibble()

names(hybrid_errors) <- c("MPE", "MAPE", "MASE")
names(naive_errors) <- c("MPE", "MAPE", "MASE")
names(HES_errors) <- c("MPE", "MAPE", "MASE")
names(Theta_errors) <- c("MPE", "MAPE", "MASE")

# Calculate average error

```

```
HB_error_avg <- colMeans(hybrid_errors)
naive_error_avg <- colMeans(naive_errors)
HES_error_avg <- colMeans(HES_errors)
Theta_error_avg <- colMeans(Theta_errors)

Benchmark_Performance <- bind_rows(HB_error_avg, naive_error_avg,
HES_error_avg, Theta_error_avg)

method <- c("Combination Strategy", "Naive", "Damped ES", "Theta")

Benchmark_Performance$Model <- method
Benchmark_Performance <- Benchmark_Performance[, c(4,1,2,3)]
Benchmark_Performance
```