Prediction for presentation

Zhang

2025-04-14

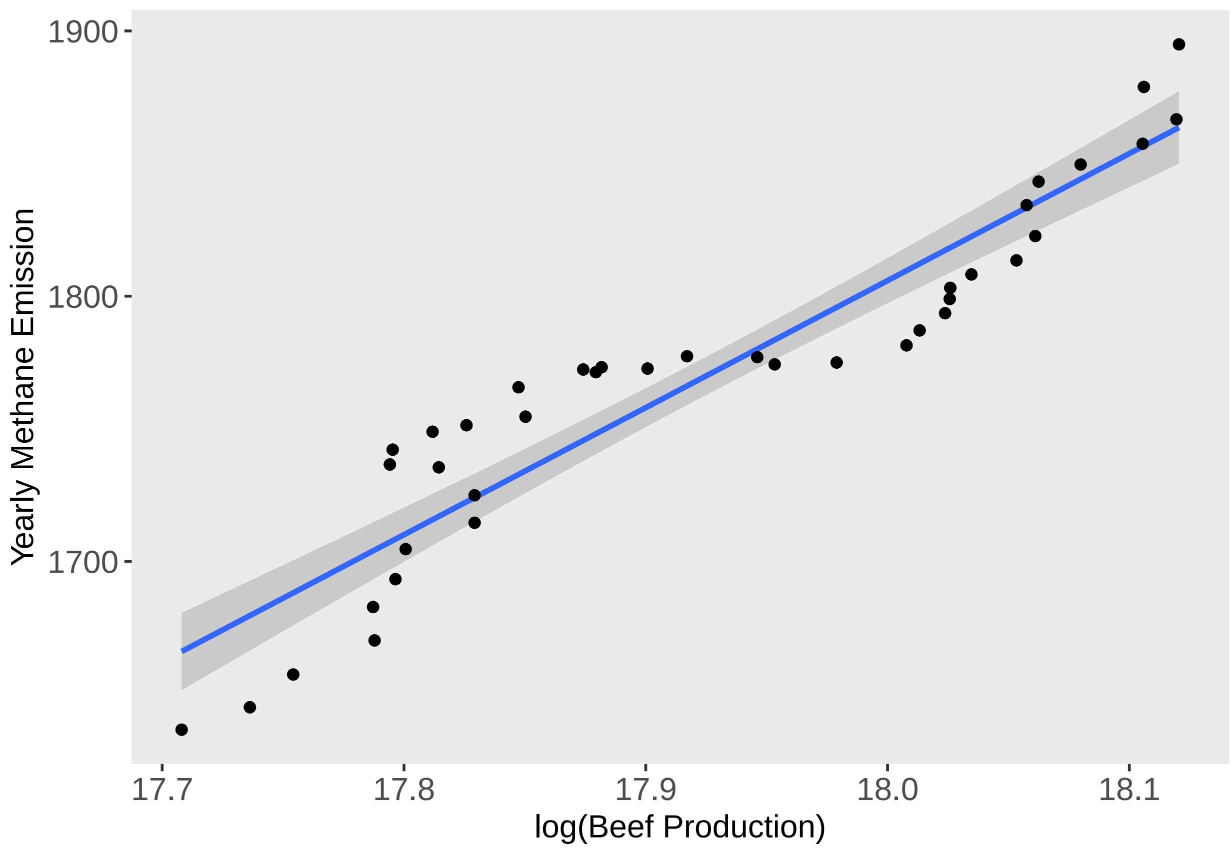
Preparation

Relationship between external variables

```
social_economic <- read.csv("../Data/Processed/social_economic_factors_monthly.csv")
beef_production <- read.csv("../Data/Processed/beef_production.csv")
```

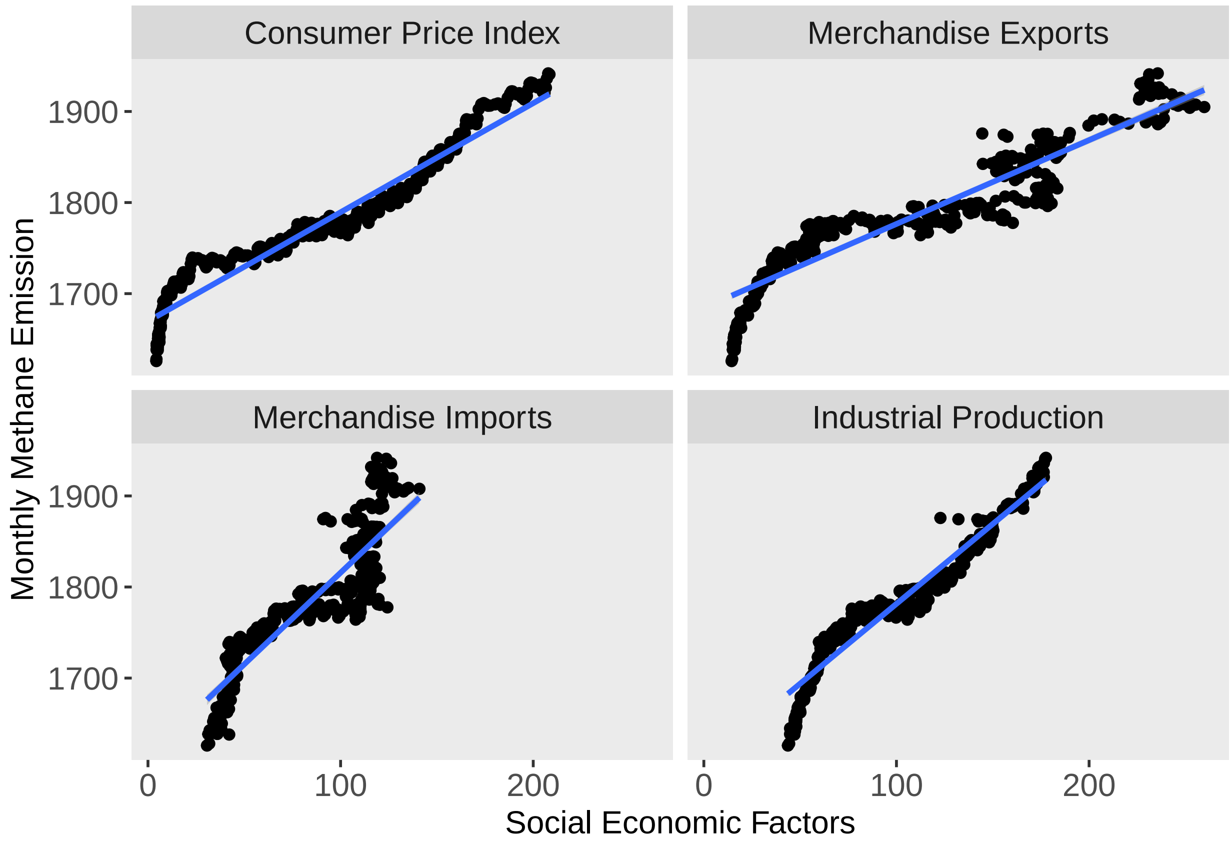
Drawing for beef production

```
methane_yearly <- read.csv("../Data/Processed/social_economic_factors_yearly.csv")

methane_yearly <- methane_all %>%
  filter(year <= 2021) %>%
  group_by(year) %>%
  summarize(yearly_average = mean(average))

methane_yearly <- cbind(methane_yearly, beef_production %>% filter(Year <= 2021))

p <- ggplot(methane_yearly, mapping = aes(x = log(Beef_Production.Tons.), y = yearly_average)) + geom_p

p + geom_point()+
  theme(panel.grid = element_blank(),
        axis.title.x = element_text(size = 12),
        axis.title.y = element_text(size = 12),
        axis.text.x = element_text(size = 12),
        axis.text.y = element_text(size = 12)
        )+
  ylab("Yearly Methane Emission")+
  xlab("log(Beef Production)")
```

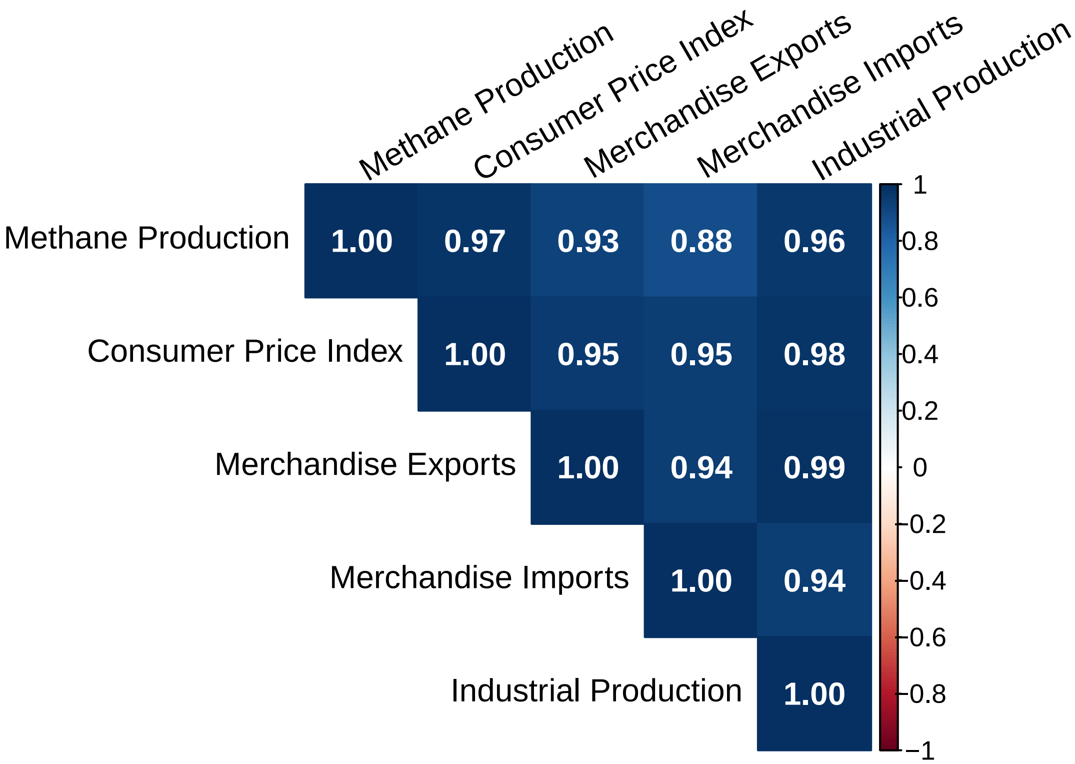


Drawing for social economic factors

```
social_economic_panel <- read.csv("../Data/Processed/social_economic_factors_monthly_paneldata.csv")

social_economic_panel$social.factors <- factor(
  social_economic_panel$social.factors,
  levels = c("cpi", "export", "import", "ip"),
  labels = c("Consumer Price Index", "Merchandise Exports", "Merchandise Imports", "Industrial Producti
)


p2 <- ggplot(social_economic_panel, mapping = aes(x = value, y = methane))

p2 + geom_point() + facet_wrap(~ social.factors, nrow = 2) + geom_smooth(method="lm") +
  theme(panel.grid = element_blank(),
        strip.text = element_text(size = 12),
        axis.title.x = element_text(size = 12),
        axis.title.y = element_text(size = 12),
        axis.text.x = element_text(size = 12),
        axis.text.y = element_text(size = 12)
        )+
  ylab("Monthly Methane Emission")+
  xlab("Social Economic Factors")
```




```
library(corrplot)


social_data_monthly <- read.csv("../Data/Processed/social_economic_factors_monthly.csv")

corr_data <- cor(social_data_monthly)

colnames(corr_data) <- c("Methane Production","Consumer Price Index", "Merchandise Exports", "Merchandis
rownames(corr_data) <- c("Methane Production","Consumer Price Index", "Merchandise Exports", "Merchandis

corrplot(corr_data,
         tl.col = "black",
         method = "color",
         addCoef.col = "white",
         type = "upper",
         tl.srt = 30)
```



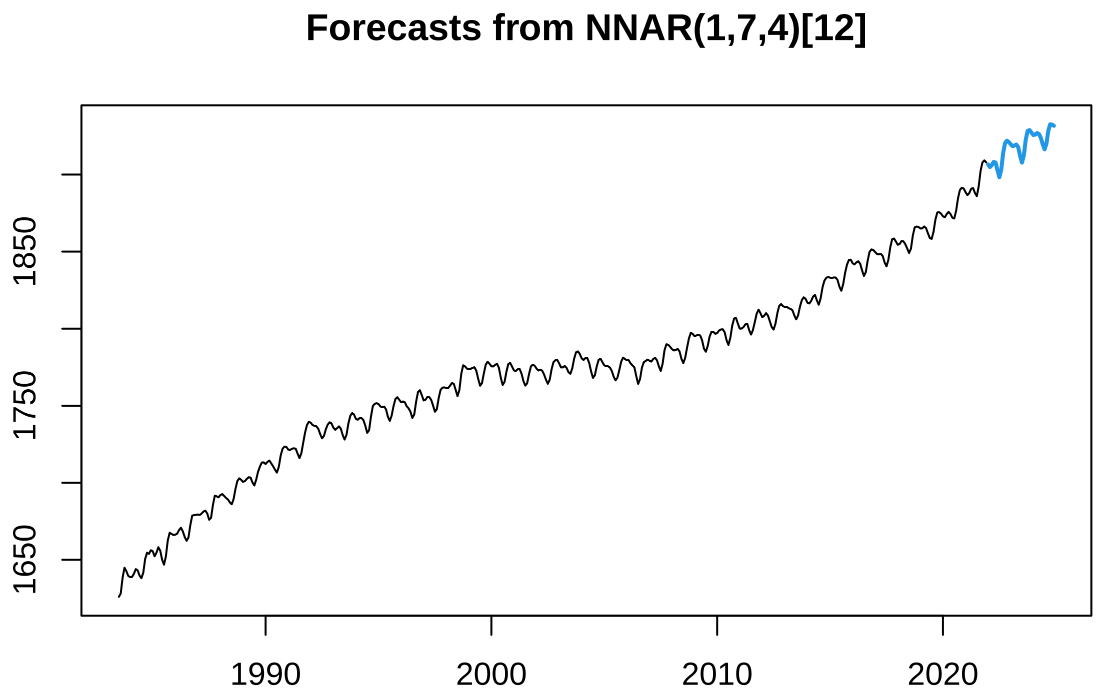
Predictions

Neuron Network

```
NN_fit <- nnetar(methane_train_ts, p = 1, P = 7)

NN_for <- forecast(NN_fit, h=36)

forecast_NN_accuracy <- accuracy(NN_for$mean, methane_test_ts)

print(forecast_NN_accuracy)
```

```
##                ME     RMSE      MAE      MPE      MAPE         ACF1 Theil's U
## Test set 1.834482 5.531544 3.331042 0.094964 0.1734473 -0.001644543 0.8194453
```

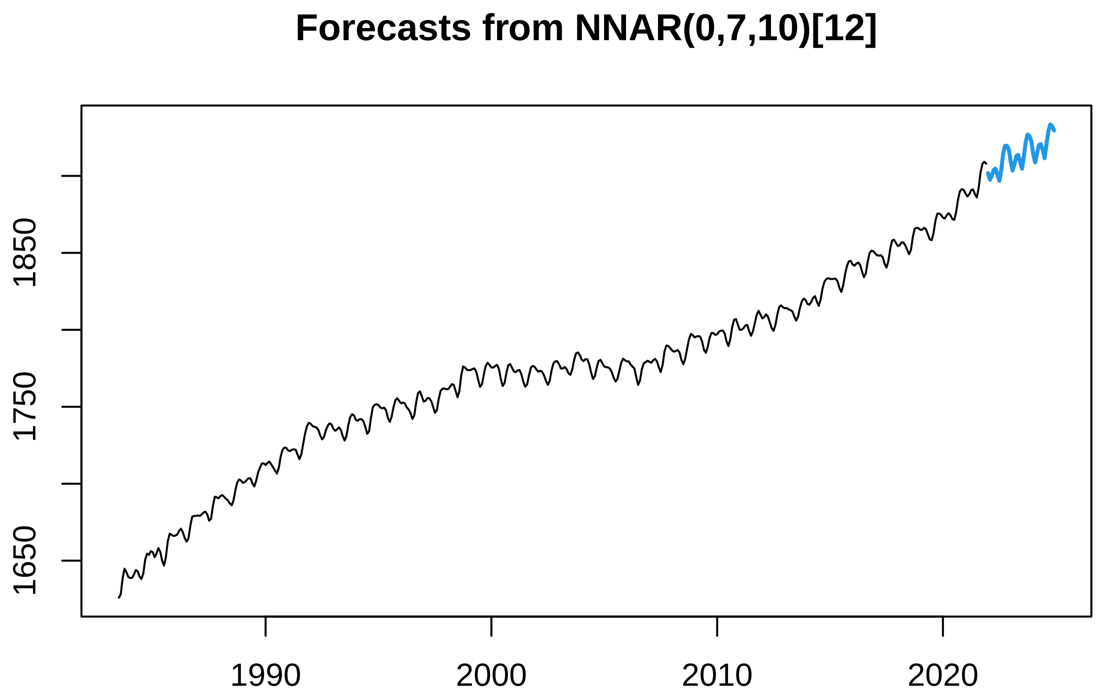
```
plot(NN_for)
```




```r
NN_fourier_fit <- nnetar(methane_train_msts, p = 0, P = 7,
                  xreg=fourier(methane_train_msts,K=c(2,6)))

NN_fourier_for <- forecast(NN_fourier_fit, h=36, xreg=fourier(methane_train_msts,K=c(2,6), h=36))

forecast_NN_fourier_accuracy <- accuracy(NN_fourier_for$mean, methane_test_msts)

print(forecast_NN_fourier_accuracy)
```

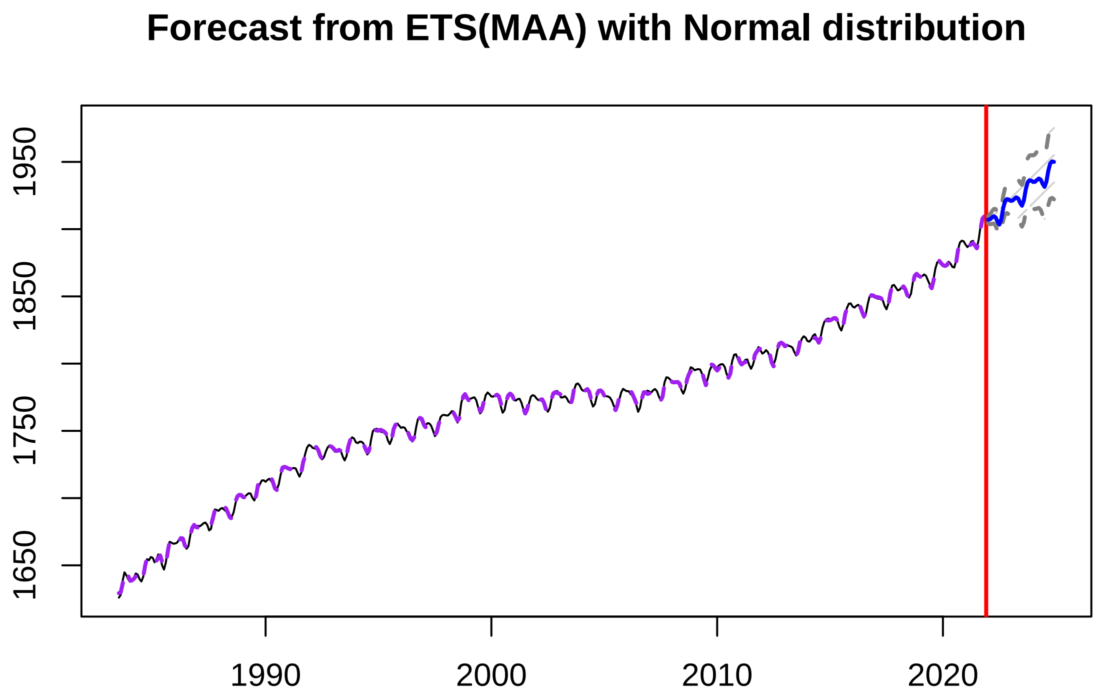
```
##                ME     RMSE      MAE       MPE      MAPE      ACF1 Theil's U
## Test set 7.041148 8.063206 7.046965 0.3663598 0.3666629 0.6777405  2.221493
```

```r
plot(NN_fourier_for)
```



State Space - Smooth

```
SSES <- es(methane_train_ts,model="ZZZ",h=36,holdout=FALSE)

SSES_for <-forecast(SSES,h=36, interval="prediction")

plot(SSES_for)
```

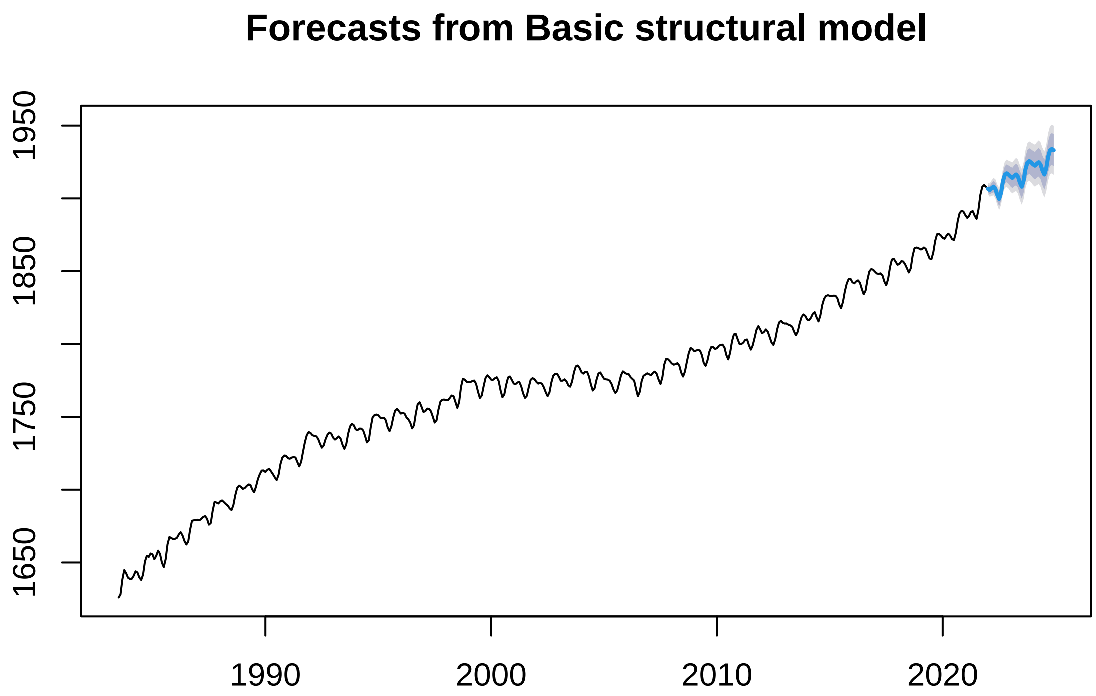



```
forecast_SSES_accuracy <- accuracy(SSES$forecast, methane_test_ts)

print(forecast_SSES_accuracy)
```

```
##                  ME     RMSE      MAE        MPE      MAPE     ACF1 Theil's U
## Test set -5.370893 9.168926 5.462944 -0.2794874 0.2843147 0.322089  1.357204
```

State Space - BSM

```
SSBSM <- StructTS(methane_train_ts,
                    type="BSM",fixed=c(NA,NA,NA,NA))

SSBSM_for <- forecast(SSBSM,h=36)

plot(SSBSM_for)
```




```r
forecast_SSBSM_accuracy <- accuracy(SSBSM_for$mean,methane_test_ts)

print(forecast_SSBSM_accuracy)
```

```
##                ME     RMSE      MAE       MPE      MAPE       ACF1 Theil's U
## Test set 3.319761 6.380075 4.822869 0.1721673 0.2510164 0.02674152 0.9460437
```

Performance Comparison

```r
kable(forecast_performance,
      #format = "latex",  # Works for PDF output
      caption = "Forecast Accuracy",
      digits = 3,
      booktabs = TRUE)
```

Table 1: Forecast Accuracy

	ME	RMSE	MAE	MPE	MAPE	ACF1	Theil.s.U
Neural Network	1.834	5.532	3.331	0.095	0.173	-0.002	0.819
Neural Network w/fourier	7.041	8.063	7.047	0.366	0.367	0.678	2.221

	ME	RMSE	MAE	MPE	MAPE	ACF1	Theil.s.U
State Space w/Exponential smoothing	-5.371	9.169	5.463	-0.279	0.284	0.322	1.357
State Space w/BSM	3.320	6.380	4.823	0.172	0.251	0.027	0.946

A tool to find the optimal parameter

Find the optimal parameter for NN

```
# i and j in the loop are the parameter for p and P. Basically it could output the model performance it

for (i in 0:3){
    for (j in 0:8){

      if (i == 0 & j == 0){
        next
      }

      NN_fit <- nnetar(methane_train_ts, p = i, P = j)

      NN_for <- forecast(NN_fit, h=36)

      forecast_NN_accuracy <- accuracy(NN_for$mean, methane_test_ts)


      cat("p=",i,"P=",j,"\n")
      print(forecast_NN_accuracy)


    }
}
```

Find the optimal parameter for NN w/fourier

```
for (i in 0:2){
    for (j in 0:8){

      if (i == 0 & j == 0){
        next
      }

      NN_fourier_fit <- nnetar(methane_train_msts, p = i, P = j,
                   xreg=fourier(methane_train_msts,K=c(2,6)))

      NN_fourier_for <- forecast(NN_fourier_fit, h=36, xreg=fourier(methane_train_msts,K=c(2,6), h=36))

      forecast_NN_fourier_accuracy <- accuracy(NN_fourier_for$mean, methane_test_msts)
```

```
      cat("p=",i,"P=",j,"\n")
      print(forecast_NN_fourier_accuracy)

    }
}
```