Assignment 5: Forecasting

Peter Ye

2024-04-15

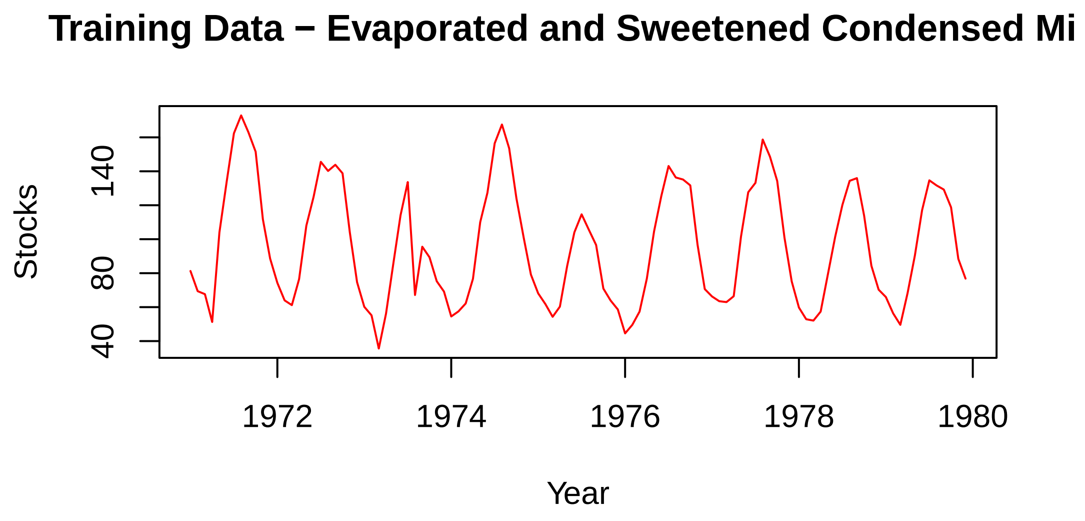
Question 1

```r
# Load conmilk.rda data
load("condmilk.rda")
```

```r
# Split the data into train and test
# Define the training dataset from January 1971 to December 1979
train_data <- window(condmilk, start=c(1971,1), end=c(1979,12))

# Define the test dataset from January 1980 to December 1980
test_data <- window(condmilk, start=c(1980,1), end=c(1980,12))
```

Question 2

```r
# Plot the training data
plot(train_data, main="Training Data - Evaporated and Sweetened Condensed Milk", xlab="Year",
     ylab="Stocks", col = "Red")
```



```r
# Find the lambda for a Box-Cox transformation
lambda <- BoxCox.lambda(train_data)
lambda
```

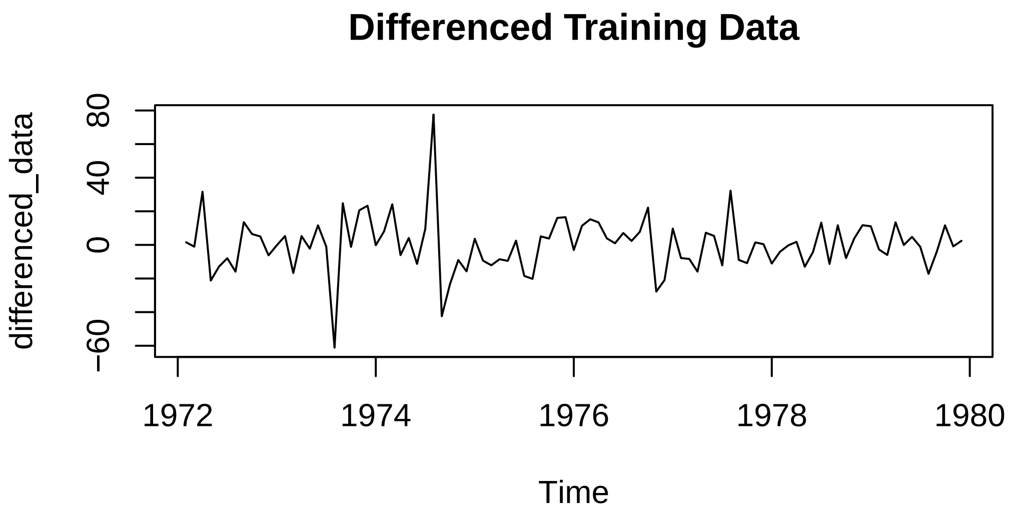
```
## [1] -0.3944237
```

The graph demonstrate there is no obvious sign that variance of the seasonal peaks and troughs changes significantly over time. Since the lambda value is negative, it implies that even more transformation is needed than a simple log transformation

Question 3

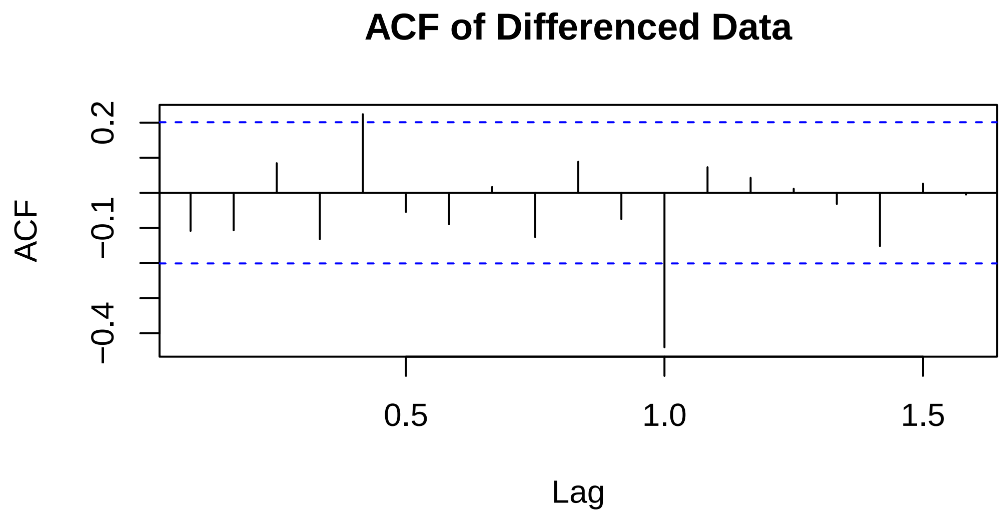
It's evident that there is a seasonal component, as well as some trend-like structures in the data. These characteristics typically suggest that the series is not stationary. Therefore, an appropriate differencing approach is necessary.

```r
# Seasonal differencing
seasonally_differenced <- diff(train_data, lag=12)

# Additional first differencing
differenced_data <- diff(seasonally_differenced, lag=1)

# Plot the differenced data
plot(differenced_data, main="Differenced Training Data")
```

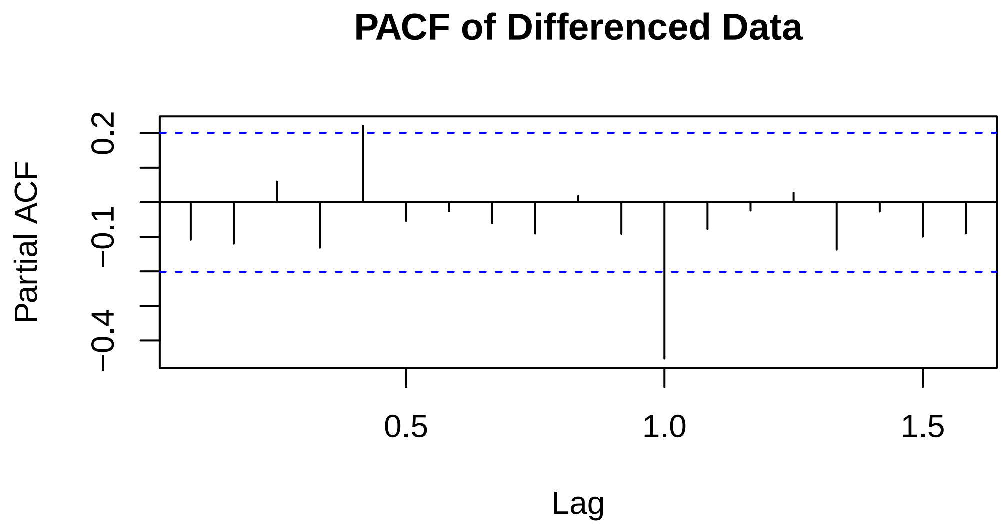


The large fluctuations have been considerably reduced, and it doesn't show an obvious trend or seasonality, which is a good indication that the differencing was effective.

```r
# Plot ACF
acf(differenced_data, main="ACF of Differenced Data")
```



```r
# Plot PACF
pacf(differenced_data, main="PACF of Differenced Data")
```



Most of the spikes in the ACF and PACF to fall within the confidence interval quickly, which suggests a stationary series.

```r
# Perform the Augmented Dickey-Fuller test
adf_test_result <- adf.test(differenced_data, alternative = "stationary")
adf_test_result
```

```
##
##  Augmented Dickey-Fuller Test
##
## data:  differenced_data
## Dickey-Fuller = -3.6473, Lag order = 4, p-value = 0.03307
## alternative hypothesis: stationary
```

The ADF test also provides evidence to conclude that the differenced time series is stationary with a p-value of 0.03307

Question 4 & 5

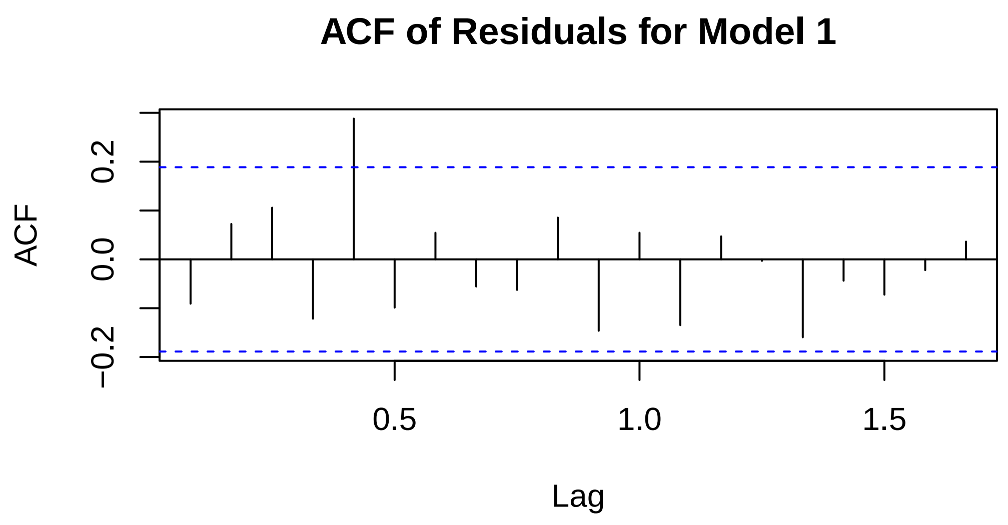
Model 1: automatic ARIMA modeling

```
# Create the model
auto_fit_1 <- auto.arima(train_data, lambda = lambda)

# Summary the model
summary(auto_fit_1)
```

```
## Series: train_data
## ARIMA(1,0,0)(2,1,0)[12]
## Box Cox transformation: lambda= -0.3944237
##
## Coefficients:
##          ar1     sar1     sar2
##        0.638  -0.7483  -0.3904
## s.e.  0.081   0.1120   0.1145
##
## sigma^2 = 0.000705:  log likelihood = 209.4
## AIC=-410.81   AICc=-410.37   BIC=-400.55
##
## Training set error measures:
##                       ME     RMSE      MAE       MPE     MAPE     MASE
## Training set -0.8181611 12.50303 8.048066 -2.059967 9.287226 0.413676
##                     ACF1
## Training set 0.007322622
```

The model seems to be performing adequately based on the coefficients and their standard errors. The non-seasonal AR1 coefficient is significant, as well as both seasonal AR coefficients (SAR1 and SAR2).

```
# Plot ACF of the first model's residuals
acf(residuals(auto_fit_1), main='ACF of Residuals for Model 1')
```



The ACF plot of the residuals for Model 1 doesn't indicate any significant autocorrelations at the lags within the confidence bounds, suggesting that the residuals are behaving like white noise.

```r
# Plot ACF of the first model
Box.test(residuals(auto_fit_1), lag=12, type="Ljung-Box")
```

```
##
##  Box-Ljung test
##
## data:  residuals(auto_fit_1)
## X-squared = 20.218, df = 12, p-value = 0.06308
```

The Ljung-Box test on the residuals, with a p-value of 0.06308, suggests that there is not enough evidence to reject the null hypothesis that the residuals are independently distributed for lag up to 12. This means that there is no significant autocorrelation in the residuals at lag 12, which generally indicates a good model fit.

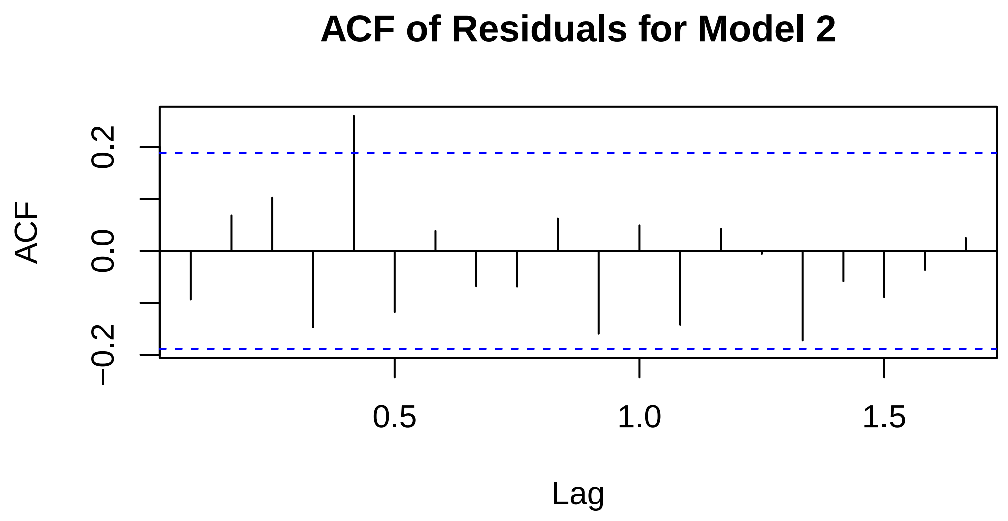
Model 2: ARIMA modeling with D=1 d=1

```r
# Create the model
auto_fit_2 <- auto.arima(train_data, D=1, d=1, lambda = lambda)

# Summary the model
summary(auto_fit_2)
```

```
## Series: train_data
## ARIMA(1,1,1)(2,1,0)[12]
## Box Cox transformation: lambda= -0.3944237
##
## Coefficients:
##          ar1      ma1     sar1     sar2
##       0.6164  -0.9566  -0.7541  -0.3823
## s.e.  0.1108   0.0638   0.1171   0.1152
##
## sigma^2 = 0.0007327:  log likelihood = 205.13
## AIC=-400.25   AICc=-399.58   BIC=-387.48
##
## Training set error measures:
##                    ME     RMSE      MAE       MPE    MAPE      MASE       ACF1
## Training set 1.409249 12.21211 7.787316 0.5090034 8.90212 0.4002733 0.00668212
```

The coefficients for the AR, MA, and seasonal AR terms are significant given their standard errors.

```r
# Plot ACF of the second model's residuals
acf(residuals(auto_fit_2), main='ACF of Residuals for Model 2')
```



The ACF plot of the residuals for Model 2 also does not indicate any significant autocorrelations at the lags within the confidence bounds. This suggests that the residuals do not have autocorrelation, and the model has captured the structure of the time series well.

```r
# Plot ACF of the second model
Box.test(residuals(auto_fit_2), lag=12, type="Ljung-Box")
```

```
##
##  Box-Ljung test
##
## data:  residuals(auto_fit_2)
## X-squared = 19.724, df = 12, p-value = 0.07249
```

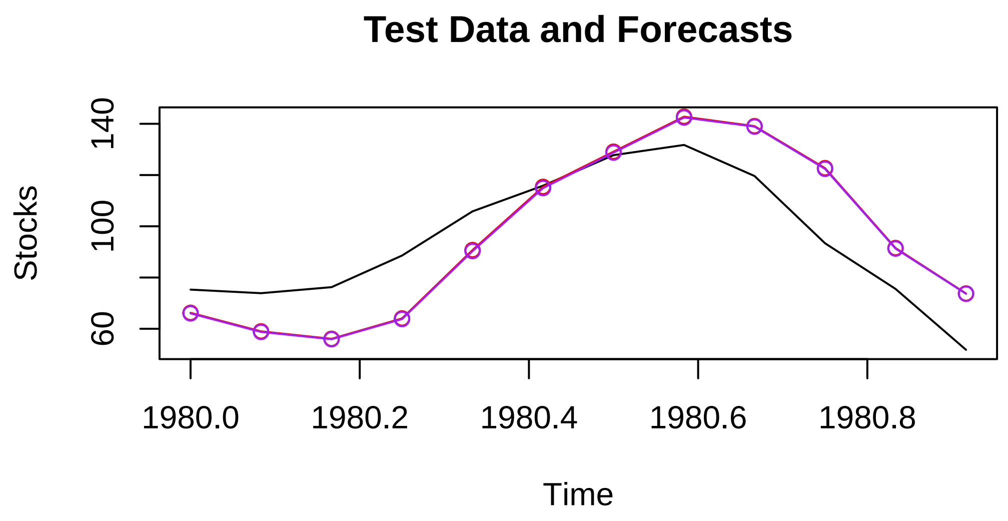
The Ljung-Box test result for Model 2, with a p-value of 0.07249, also indicates that there is not enough statistical evidence to reject the null hypothesis of no autocorrelation among residuals up to lag 12.

Overall, Model 1 has a lower AICc and BIC value, suggesting that it may be the better model in terms of information criteria, which penalize for model complexity. Both models have residuals that appear to be white noise, as neither ACF plot showed significant autocorrelations, and the Ljung-Box test p-values are above the conventional threshold of 0.05.

Question 6

```r
# Forecasting using Model 1
forecast_model_1 <- forecast(auto_fit_1, h=12)

# Forecasting using Model 2
forecast_model_2 <- forecast(auto_fit_2, h=12)
```

```r
# Determine the range for the y-axis
y_range <- range(c(test_data, forecast_model_1$mean, forecast_model_2$mean), na.rm = TRUE)

# Plot the actual test data and forecasted values for Model 1
plot(test_data, main="Test Data and Forecasts", xlab="Time", ylab="Stocks", ylim=y_range)

# Adding the forecasted values from Model 1 to the plot
lines(forecast_model_1$mean, col="red", type="o")

# Adding the forecasted values from Model 2 to the plot
lines(forecast_model_2$mean, col="purple", type="o")
```



The graph demonstrates that predicted values from both models are overlapped. It suggests that the two models are providing very similar forecasts for the test data period in 1980. This overlapping could also imply that the seasonal and trend components captured by both models are closely aligned, leading to similar predicted paths

Question 7

```r
# Calculate MAPE and MSE for Model 1
mape_model_1 <- mean(abs((test_data - forecast_model_1$mean) / test_data), na.rm = TRUE) * 100
mse_model_1 <- mean((test_data - forecast_model_1$mean)^2, na.rm = TRUE)
```

```r
# Calculate MAPE and MSE for Model 2
mape_model_2 <- mean(abs((test_data - forecast_model_2$mean) / test_data), na.rm = TRUE) * 100
mse_model_2 <- mean((test_data - forecast_model_2$mean)^2, na.rm = TRUE)
```

Table 1: Error Metrics for ARIMA Models

Model	MAPE	MSE
Model 1	18.47494	303.4648
Model 2	18.51091	303.4596

Model 1 has a slightly lower MAPE than Model 2, suggesting it was marginally more accurate in terms of percentage error. Conversely, Model 2 has a slightly lower MSE, indicating it had a slightly smaller average for the squared errors. However, the differences between the two models are minimal. Such a small discrepancy may not be practically significant

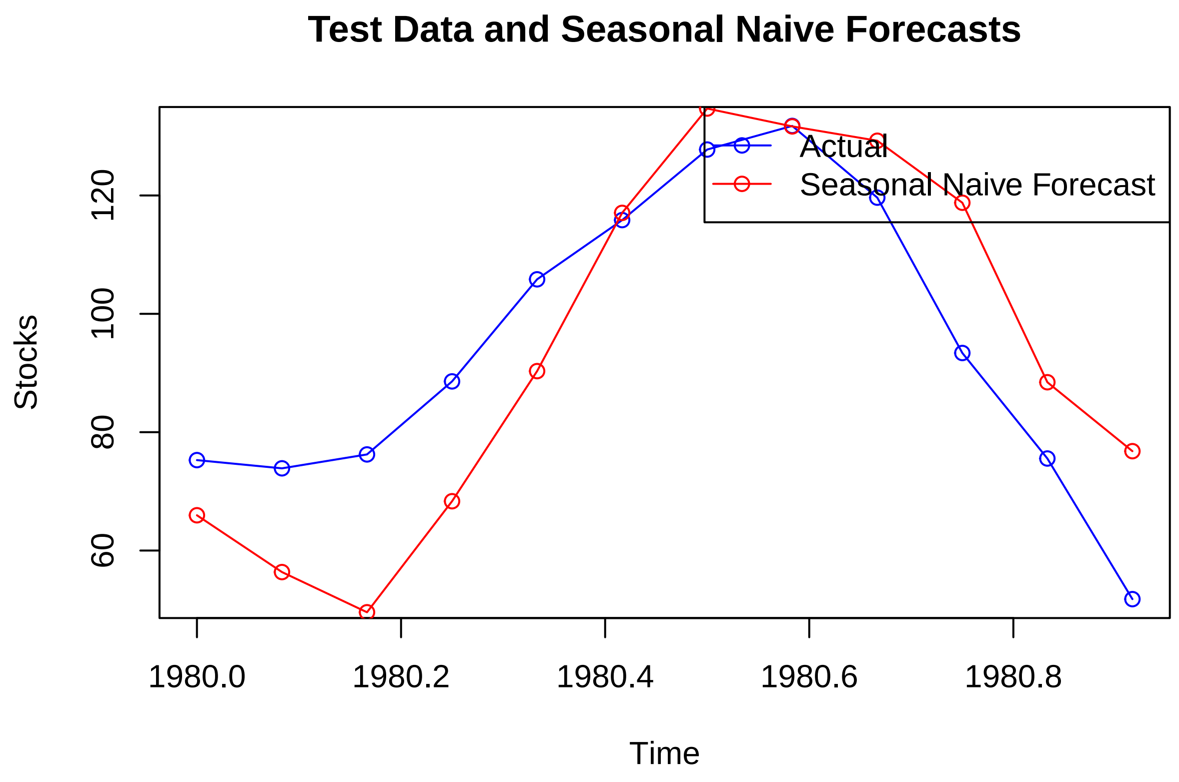
Question 8

```r
# Create seasonal naive model
Model_SNaive <- snaive(train_data, 12, lambda = lambda)

# Forecast using the seasonal naive model
forecast_SNaive <- forecast(Model_SNaive, h=12)

# Calculate MAPE and MSE for the seasonal naive forecast
mape_SNaive <- mean(abs((test_data - forecast_SNaive$mean) / test_data), na.rm = TRUE) * 100
mse_SNaive <- mean((test_data - forecast_SNaive$mean)^2, na.rm = TRUE)
```

Table 2: Error Metrics for Model SNaive

Model	MAPE	MSE
Model SNaive	17.97548	277.8286

```r
# Plot the test data and the seasonal naive forecasted values
plot(test_data, main="Test Data and Seasonal Naive Forecasts",
     xlab="Time", ylab="Stocks", type='o', col='blue')
lines(forecast_SNaive$mean, type='o', col='red')

# Add a legend to the plot
legend("topright", legend=c("Actual", "Seasonal Naive Forecast"), col=c("blue", "red"), lty=1, pch=1)
```



Question 9

Based on the table for MAPE and RMSE, the Model SNaive outperforms with 17.97548 and 277.8286 respectively over the best model. In that case, the best model we have probably lack of complexity.

Question 10

```r
# Define a function to compute the forecast based on an AR(4) model
calculate_forecast <- function(series) {
  series_length = length(series)
  recent_values = series[(series_length-3):series_length]
  forecast_value = 30993 + 0.82 * recent_values[4] - 0.29 * recent_values[3] -
    0.01 * recent_values[2] - 0.22 * recent_values[1]
  return(forecast_value)
}

# Obtain the first forecast using the training data
first_forecast <- calculate_forecast(train_data)

# Get the second forecast by adding the first forecasted point to the training data
second_forecast <- calculate_forecast(c(train_data, first_forecast))
```

```r
# Calculate the third forecast by including the first two forecasted points
third_forecast <- calculate_forecast(c(train_data, first_forecast, second_forecast))
```

Table 3: Forecast without forecast() function

order	Forecasted_Value
first_forecast	31000.70
second_forecast	56364.29
third_forecast	68201.29