Contestants: Roshanak Khaleghi[1], Juan Xu[1]

1. Introduction

In this contest[2], we use the provided 10-year monthly demand data to find a demand forecast model and an inventory control model. We use the first 8-year data (month 1 – month 96) as the training data and the last 2-year data (month 97 – month 120) as the test data. For the inventory control part, Autoregressive Integrated Moving Average (ARIMA) has a smaller test error[3] since it is automatically updated as new demand is observed, while XGBoost has a smaller inventory cost due to the unbalanced penalty of holding cost and backlogging cost. We combine ARIMA and XGBoost as a mixed forecast model due to ARIMA's adjustment power and XGBoost's cost-robust power. For the inventory control part, we use the insights from the traditional newsvendor problem and heuristically combine the prediction errors into the model. Parameters in the inventory control model are derived by minimizing the total inventory cost when applying demand forecast and inventory control models into the test data. The workflow of the data-driven demand forecast and inventory control is illustrated as follows:

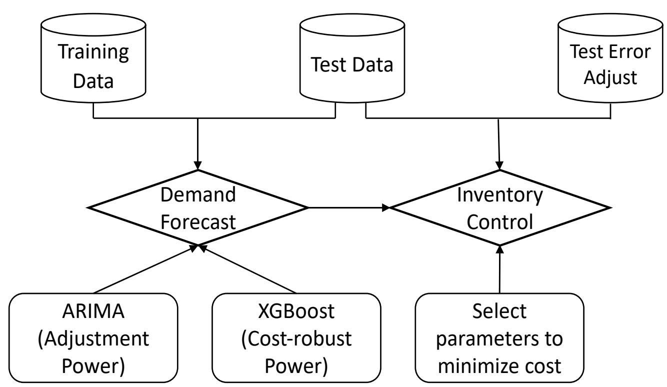

Figure 1. Data-Driven Demand Forecast and Inventory Control Workflow

Finally, we create a test data set for month 121 – month 144 (i.e., year 2006 – year 2007), called "*generated test data*" (referred to as "Two-Year-Demand.csv" in the code), by taking average of the separate predictions from XGBoost and ARIMA. We apply ARIMA model, XGBoost model, and the mixed demand forecast model to this generated test data to see the prediction errors. Meanwhile, we apply the derived inventory control model to obtain the inventory costs aligned with these three forecast models. The one we finally use for this contest is the mixed model.

2. Data Overview and Demand Forecast

2.1. Data Overview

Figure 2 shows the plot for the 10-year monthly demand data. The decomposition of the demand in terms of trend and seasonality is shown in Figure 3. From Figure 3, we can observe that the given 10-year monthly demand time series has a trend and seasonality.

---

[1] Industrial and Enterprise Systems Engineering, University of Illinois Urbana-Champaign, {khalegh2, juanxu2}@illinois.edu.

[2] Implemented in R.

[3] The test error is measured by the Residual Sum of Squares (RSS) over the 24 months in the test data.

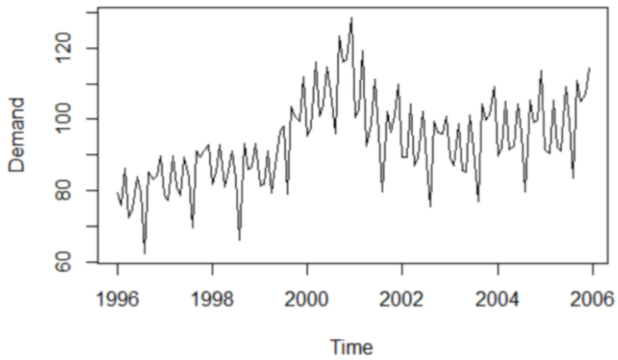

Figure 2. Demand Time Series.

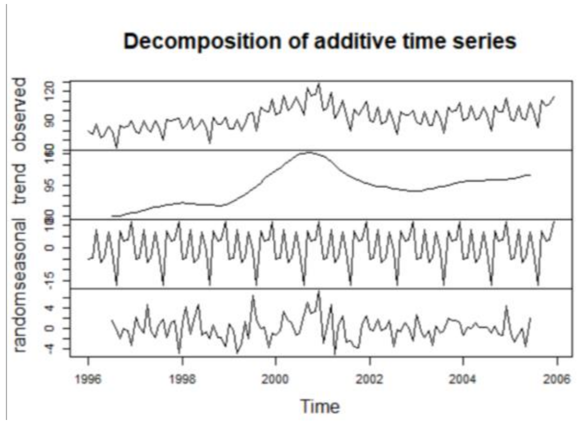

Figure 3. Decomposition of Demand Time Series

2.2. Demand Forecast – ARIMA Model

We fit the training data (month 1 – month 96) in time series type by auto.arima() function in R. The returned model is ARIMA$(1,1,5) \times (0,1,1)$, which shows there is a trend and seasonality. We use the ARIMA$(1,1,5) \times (0,1,1)$ to forecast the demand of the first month in the test data (i.e., month 97). After the demand of month 97 is realized, we use the time series from month 1 to month 97 to update the ARIMA model and forecast month 98, and we repeat the same process until we reach the end of the test period (i.e., month 120). The test error is measured by the Residual Sum of Squares (RSS) over the 24 months in the test data. The test error of the ARIMA model is 2.49. Following is the pseudocode for the ARIMA model.

Input: Demand from month 1 to month 96 (training data 1)
Initialization: $t = 97$, $ARIMA_1^{96}$ using the demand from month 1 to month 96
While $97 \leq t \leq 120$:
    Predict demand t by $ARIMA_1^{t-1}$
    Demand t is realized
    Update the ARIMA model to $ARIMA_1^{t}$
    $t = t + 1$
Output: Demand forecast from month 97 – month 120

Code 1. Pseudocode for ARIMA Model

2.3. Demand Forecast – XGBoost Model.

XGBoost is known for its good performance in prediction. For the demand in each month t ($t = 25, \dots, 96$), its features are set as the year index, the month index[4], and the demands in the past 24 months[5]. We use the demand from month 25 to month 96 along with their features to fit the XGBoost model. The parameters in the XGBoost are tuned to minimize the test error. The parameters in the XGBoost are max.depth=5, eta=0.05, subsample=0.5, nrounds=500. The test error for the XGBoost is 2.63. Compared to XGBoost, the ARIMA model has a better prediction performance. Below is the pseudocode of the XGBoost model.

---

[4] We transform the month index into the format $\{1,2,\dots,12\}$.
[5] We also tried "past 12 months" and "past 36 months". The features with the past 24 months perform the best.

Input: Demand from month 25 to month 96 with their features (training data 2)
Initialization: XGBoost using the training data with max.depth=5, eta=0.05, subsample=0.5, nrounds=500
Prediction: Predict the demand from month 97 to month 120 by XGBoost in one shot
Output: Demand forecast from month 97 – month 120

Code 2. Pseudocode for XGBoost Model

2.4. Demand Forecast – Mixed Model.

In the so-called "Mixed Model", we take advantage of the adjustment power of ARIMA and the cost-robust power[6] of XGBoost and to build a mixed model. In each step of the prediction, we compare the inventory costs induced by the XGBoost and ARIMA model[7], and choose the approach which gives the smallest cost.

3. Inventory Control

Since there is no fixed ordering cost, to minimize the total inventory cost for the 2-year period, we minimize the inventory cost for each month in 2 years. The following figure shows the order of events in each month.

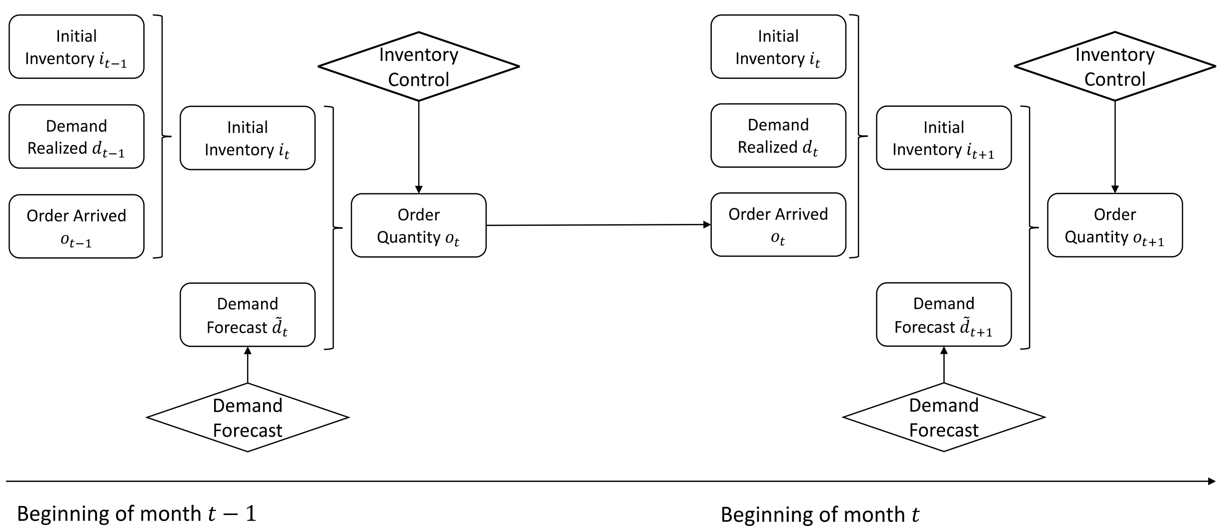

Figure 4. Order of Events in the Inventory System

3.1. Newsvendor Model for month t.

The inventory holding cost per unit per month is denoted by h. We know that when inventory is no more than 90 units, $h = 1$; when inventory exceeds 90 units, the holding cost for the exceeding part is 2 dollars. The inventory backlogging cost per unit per month is $p = 3$. At the beginning of month t, let i_t be the initial inventory, o_t be the arrived order quantity (ordered at the beginning of period $t-1$), d_t be the realized demand, and $\tilde{d}_{t+1}$ be the forecast for the demand realized at the beginning of month $t+1$. Given i_t, the newsvendor model for month t is as follows:

[6] We will see the cost-robust power in Section 3.2.
[7] If the prediction is for the first month in the test data (i.e., month 97), we would use the prediction from XGBoost.

$$\min_{o_t \geq 0}\left\{h \cdot i_{t+1} \cdot I_{\{0 \leq i_{t+1} \leq 90\}} + [(h+1) \cdot i_{t+1} - 90] \cdot I_{\{i_{t+1} > 90\}} - p \cdot i_{t+1} \cdot I_{\{i_{t+1} < 0\}}\right\}$$
$$s.t.\ \ i_{t+1} = i_t + o_t - \tilde{d}_t \tag{1}$$

3.2. Inventory Control

In the standard single-period newsvendor model, when the demand follows a Normal distribution with mean μ and standard deviation σ, the optimal order quantity is given by

$$o^{newvendor} = \mu + Z^{-1}(critical\ ratio) \cdot \sigma, \tag{2}$$

where the critical ratio is determined by the per unit ordering cost, per unit salvage value (i.e., the holding cost in this contest), and per unit backlogging cost, and $Z^{-1}(\cdot)$ returns the quantile of a standard Normal distribution.

Motivated by the standard single-period newsvendor model, we use the 1-year historic demands to estimate the standard deviation of demand at the beginning of month t. We denote the standard deviation of the 1-year historic demand by $\hat{\sigma}_t = sd(d_{t-12}, d_{t-11}, \ldots, d_{t-1})$. The order quantity at the beginning of month t in our inventory control is:

$$o_t^B = \tilde{d}_t - i_t + c_1^B \cdot \hat{\sigma}_t, \tag{3}$$

where c_1^B is a parameter to be chosen to minimize the total inventory cost on the test data. We refer to equation (3) as the "*Basic Inventory Model*".

To leverage the impact of forecast errors on the order quantity, we make improvement by adding a new term to the order quantity of each month. This new term is related to the forecast errors. We denote the prediction error of period t by $e_t = \tilde{d}_t - d_t$, where $\tilde{d}_t$ is the forecast demand and d_t is the true demand. We propose the following improved model, referred to as the "*Extended Inventory Model*":

$$o_t^E = \tilde{d}_t - i_t + c_1^E \cdot \hat{\sigma}_t + c_2^E \cdot sd(e_{t-12,}e_{t-11}, \ldots, e_{t-1}), \tag{4}$$

where c_1^E and c_2^E are parameters to be chosen to minimize the total inventory cost on the test data.

We find the values of these parameters in each inventory control model by using the ARIMA model and the XGBoost model, respectively. The results are summarized in the following next two tables.

ARIMA Model (training data 1: month 1 – month 96; test data: month 97 – month 120)				
	c_1	c_2	Total Inventory Cost	RMSE
Basic Inventory Model	-0.14	-	138.54	2.49
Extended Inventory Model	0.3	-1.8	137.46	2.49

Table 1. Values of c_1^B, c_1^E, c_2^E with ARIMA Model

We choose the "Extended Inventory Model" with $c_1^1 = 0.3, c_2^1 = -1.8$ as the inventory control model aligned with the ARIMA model due to its smallest inventory cost on the test data.

XGBoost Model (training data 2: month 25 – month 96; test data: month 97 – month 120)				
	c_1	c_2	Total Inventory Cost	RMSE
Basic Inventory Control	0.27	-	87.40	2.63
Extended Inventory Model	-0.57	3.38	71.54	2.63

Table 2. Values of c_1^B, c_1^E, c_2^E with XGBoost Model

We choose the "Extended Inventory Model" with $c_1^2 = -0.57, c_2^2 = 3.38$, as the inventory control model aligned with XGBoost model due to its smallest inventory cost on the test data.

Note that for both of the ARIMA and the XGBoost model, the "Extended Inventory Model" outperforms the "Basic Inventory Model" in terms of the obtained total inventory cost. Also, while the ARIMA model provides slightly more accurate predictions (based on RSME metric) compared with the XGBoost model, the total inventory cost obtained by the XGBoost is smaller than the cost given by the ARIMA model. This is due to the fact that the backorder unit cost is greater than the holding unit cost, the more accurate model (ARIMA model) may not always perform well if it sometimes underestimates the demand values, which results in greater inventory costs. On the contrary, the XGboost tends to overestimate the demand values, which results in smaller inventory cost.

3.3. Inventory Control with The Mixed Model

Now, we have the inventory control models aligned with the ARIMA model and the XGBoost model. The inventory cost in each month induced by these two forecast models can be calculated. The pseudocode below shows how we apply the mixed model to forecast demand and to do inventory control.

Input: Demand from month 1 to month 96 (**training data 1**); demand from month 25 to month 96 with their features (**training data 2**); demand from month 97 to month 120 (**test data**)

Initialization: $t = 97$, $ARIMA_1^{96}$ using training data 1, XGBoost using training data 2 with max.depth=5, eta=0.05, subsample=0.5, nrounds=500; prediction.mixed=[], prediction.arima=[], prediction.xgboost=[], order.mixed=[], order.arima=[], order.xgb=[], cost.mixed=0, cost.arima=0, cost.xgboost=0

```
While 97 ≤ t ≤ 120:
    If t==1:
        prediction.mixed(t) = prediction.xgboost(t)
        order.mixed(t) = order.xgboost(t)
        demand t is realized
        cost.mixed = cost.mixed + inventory cost of mixed model at month t
        cost.xgb = cost.xgb + inventory cost of xgboost model at month t
        cost.arima = cost.arima + inventory cost of arima model at month t
        t = t + 1
    else:
        update the ARIMA model to ARIMA_1^t
        predict demand t by ARIMA_1^t
        if (cost.xgb < cost.arima):
            prediction.mixed(t) = prediction.xgboost(t)
            order.mixed(t) = order.xgboost(t)
            demand t is realized
            cost.mixed = cost.mixed + inventory cost of mixed model at month t
            cost.xgb = cost.xgb + inventory cost of xgboost model at month t
            cost.arima = cost.arima + inventory cost of arima model at month t
            t = t + 1
        else:
            prediction.mixed(t) = prediction.arima(t)
            order.mixed(t) = order.arima(t)
            demand t is realized
            cost.mixed = cost.mixed + inventory cost of mixed model at month t
            cost.xgb = cost.xgb + inventory cost of xgboost model at month t
            cost.arima = cost.arima + inventory cost of arima model at month t
            t = t + 1
Output: prediction.mixed, order.mixed, cost.mixed
```

Code 3. Pseudocode for Mixed Model and Inventory Control Model (Final Model Submitted)

Mixed Model (training data 1: month 1 – 96; training data 2: month 25 – 96; test data: month 97 – 120)				
	c_1	c_2	Total Inventory Cost	RMSE
ARIMA	0.3	-1.8	71.54	2.63
XGBoost	-0.57	3.38		

Table 3. Performance of the Mixed Model and Inventory Control Model

Note that these numbers (total inventory cost and RMSE) are exactly the same as XGBoost aligned with "Extended Inventory Model", where $c_1^2 = -0.57, c_2^2 = 3.38$. Because in each step of the mixed model, the model turns out to choose the forecast given by the XGBoost.

4. Generated Test Data and Simulated Results

We create a test data set[8] for month 121 – month 144 (i.e., year 2006 – year 2007), called "*generated test data*", by taking average of the separate predictions from XGBoost and ARIMA. We apply the mixed demand forecast model to this generated test data and apply the derived inventory control model to compare the inventory cost.

ARIMA Model (training data 1: month 1 – month 120; test data: month 121 – month 144)				
	c_1	c_2	Total Inventory Cost	RMSE
Extended Inventory Model	0.3	-1.8	86.07	1.77

Table 4. Performance of the ARIMA Model and Inventory Control Model on the Generated Test Data

XGBoost Model (training data 2: month 25 – month 120; test data: month 121 – month 144)				
	c_1	c_2	Total Inventory Cost	RMSE
Extended Inventory Model	-0.57	3.38	124.77	1.62

Table 5. Performance of the XGBoost Model and Inventory Control Model on the Generated Test Data

Mixed Model (training data 1: month 1 – 120; training data 2: month 25 – 120; test data: month 121 – 144)				
	c_1	c_2	Total Inventory Cost	RMSE
ARIMA	0.3	-1.8	95.45[9]	1.77[10]
XGBoost	-0.57	3.38		

Table 6. Performance of the Mixed Model and Inventory Control Model on the Generated Test Data

By including the ARIMA model, the mixed model can reduce the risk of overfitting which might result from the XGBoost. That is why we eventually decided to use the mixed model for predicting the demand values and calculating the order quantities for the inventory model.

[8] This generated test data set has been uploaded to the Github Repository as "Two-Year-Demand.csv".
[9] For this instance, the mixed model always chooses ARIMA in each step expect the first one. In the first step, we force the mixed model to use the forecast by XGBoost.
[10] The total inventory cost and RMSE could be slightly different due to the randomness of xgboost, different versions of R, and different computers.

5. Readme File and GitHub Repository Link

- **The program, named as "RJ_fORged.R", is coded in R.**
- The running time is approximately 20-30 minutes.

- **Required input files:**
  - A csv file containing the 10-year demand from Jan. 1996 to Dec. 2005 (i.e., "Ten-Year-Demand.csv" file found at "https://sites.google.com/usc.edu/gomez/data");
  - A csv file containing the 2-year demand from Jan. 2006 to Dec. 2007 (the format of the data in the csv file should be the same as the "Ten-Year-Demand.csv"). The default name is "Two-Year-Demand.csv". **The month column should start from 121 to 144.**

- **Make the following changes in the R script as required:**
  1. Set the working directory of the R project in line1: setwd('...');
  2. Make sure to input "Ten-Year-Demand.csv" as data<-read.csv("Ten-Year-Demand.csv") in line 17;
  3. Make sure to input "Ten-Year-Demand.csv" as dataTrainOriginal<-read.csv("Ten-Year-Demand.csv") in line 103;
  4. Make sure to input "Two-Year-Demand.csv" as test <- read.csv("Two-Year-Demand.csv") in line 112.

- **The output can be found in 2 separate csv files:**
  1. A csv file named "inventoryOutputMonthly.csv" saved into the working directory of the R project; the file includes the following data:
     - Monthly beginning inventory (Initial.Inventory);
     - Monthly order quantity (Order.Quantity);
     - Monthly ending inventory (Ending.Inventory);
     - Monthly holding cost (Holding.Cost);
     - Monthly backorder cost (Backorder.Cost);
  2. A csv file named "inventoryOutputSummary.csv" saved into the working directory of the R project; the file includes the following data:
     - Total inventory cost from Jan. 2006 to Dec. 2007 (Total.Inventory.Cost);
     - Total holding costs from Jan. 2006 to Dec. 2007 (Total.Holding.Cost);
     - Average holding costs from Jan. 2006 to Dec. 2007 (Avg.Holding.Cost);
     - Total backorder costs from Jan. 2006 to Dec. 2007 (Total.Backorder.Cost);
     - Average backorder costs from Jan. 2006 to Dec. 2007 (Avg.Backorder.Cost).

- **GitHub Link:**

https://github.com/RoshanakKhaleghi/fORged-Contest