Time series analysis - Inventory

Problem Statement

- **Dataset**: ELECEQUIP - Demand of electrical equipment over a ten-year period
- **Model**: Rolling horizon inventory
- **Objective**: Minimize the cost by making reasonable order plan

Workflow of project

Dataset

-> 5 years data for training

-> 6 month for validation

Validation Stationary

-> Forecast demand for 5 months

-> each time move the 1 month forward

-> do it 6 times (R)

Validation Demand Error Predicted-Actual

-> DLP(AMPL)

-> SLP(Python)

-> Get the order plan for next 4 month

-> Only implement the first order plan for next month

Validation Cost Holding + lost sale

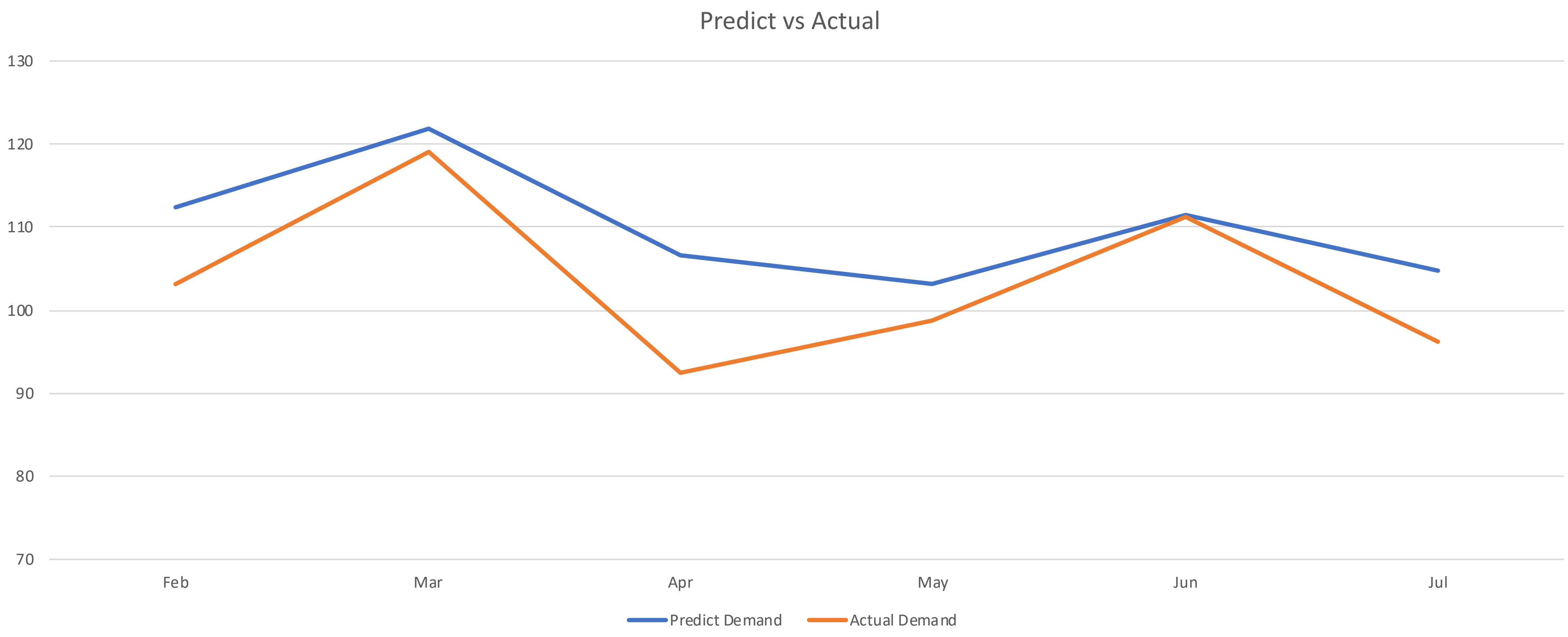
Training time series input data

```
k <- 60
for (j in 1:6) {
  k <- k+1
  y <- ts(elecequip, frequency = 12, start = c(1996,1), end = c(2001,j))
  y <- tsclean(y,replace.missing = TRUE, lambda = 'auto')
  y_de <- stl(y,"periodic")
  y1 <-y-y_de$time.series[,2]
  kp <- kpss.test(y1,null = c('Trend'))
  if (kp$p.value > 0.05){
    sprintf("have no evidence that it is not trend stationary for train data from 1996 1 to 2001 %i",j)
  }
  fit<-auto.arima(y1)
  y_pred<-forecast(fit,5,level = 90)
  y_real<-y_pred$mean+y_de$time.series[k,2]
  write(y_real, file = "out_mean2.csv", append = TRUE, sep = ",")
  write(y-y_de$time.series[,2]-y_pred$fitted, file = "out_error2.csv", append = TRUE, sep = ",")
}
```

Identify And Replace Outliers & Missing Values In A Time Series

```
6: In kpss.test(y1, null = c("Trend")) :
  p-value greater than printed p-value
> kp$p.value
[1] 0.1
> sprintf("have no evidence that it is not trend stationary for train data from 1996 1 to 2001 %i",j)
[1] "have no evidence that it is not trend stationary for train data from 1996 1 to 2001 6"
```

Output data – predicted demand



Decision Models

predict demand, start inventory, (error term)

INPUT

- DLP(AMPL)

Min E$[\sum_{t=0}^{T-1} h_t x_t + b_t z_t]$

s.t. $y_{t+1} - x_t - \Delta_t = 0$

$y_{t+1} \leq R_{t+1}$

$\Delta_t \leq U_t$

Capacity

$-y_t + x_t \geq -D_t$

$y_t + z_t \geq D_t$

$x_t, y_t, z_t \geq 0$

End Inventory

$x_t = Max(0, y_t - D_t)$

Lost Sales

$z_t = Max(0, D_t - y_t)$

- SLP(Python)

Min E$[\sum_{t=0}^{T-1} h_t x_t(w) + b_t z_t(w)]$

s.t. $y_{t+1}(w) - x_t(w) - \Delta_t(w) = 0$

$y_{t+1}(w) \leq R_{t+1}(w)$

$\Delta_t \leq U_t$

$-y_t(w) + x_t(w) \geq -D_t(w)$

$y_t(w) + z_t(w) \geq D_t(w)$

$x_t(w), y_t(w), z_t(w) \geq 0$

OUTPUT

order plan for next 4 month
Implement first order plan for next month

Run it 6 times

```
# parameters
var x1, >= 0;
var x2, >= 0;
var x3, >= 0;
var x4, >= 0;
var x5, >= 0;
var y1, >= 0;
var y2, >= 0;
var y3, >= 0;
var y4, >= 0;
var y5, >= 0;
var z1, >= 0;
var z2, >= 0;
var z3, >= 0;
var z4, >= 0;
var z5, >= 0;
var d2, >= 0;
var d3, >= 0;
var d4, >= 0;
var d5, >= 0;


minimize object: x1+x2+x3+x4+x5+3*(z1+z2+z3+z4+z5);

s.t. c1: y1 = 106.0461;
s.t. c2: -y1 + x1 >= -104.6734;
s.t. c3: y1 + z1 >= 104.6734;

s.t. c4: y2 = x1 + d2;
s.t. c5: -y2 + x2 >= -89.2025;
s.t. c6: y2 + z2 >= 89.2025;

s.t. c7: y3 = x2 + d3;
s.t. c8: -y3 + x3 >= -116.6163;
s.t. c9: y3 + z3 >= 116.6163;

s.t. c10: y4 = x3 + d4;
s.t. c11: - y4 + x4 >= -110.6412;
s.t. c12: y4 + z4 >= 110.6412;

s.t. c13: y5 = x4 + d5;
s.t. c14: - y5 + x5 >= -110.7723;
s.t. c15: y5 + z5 >= 110.7723;
```

Welcome | ts1.py

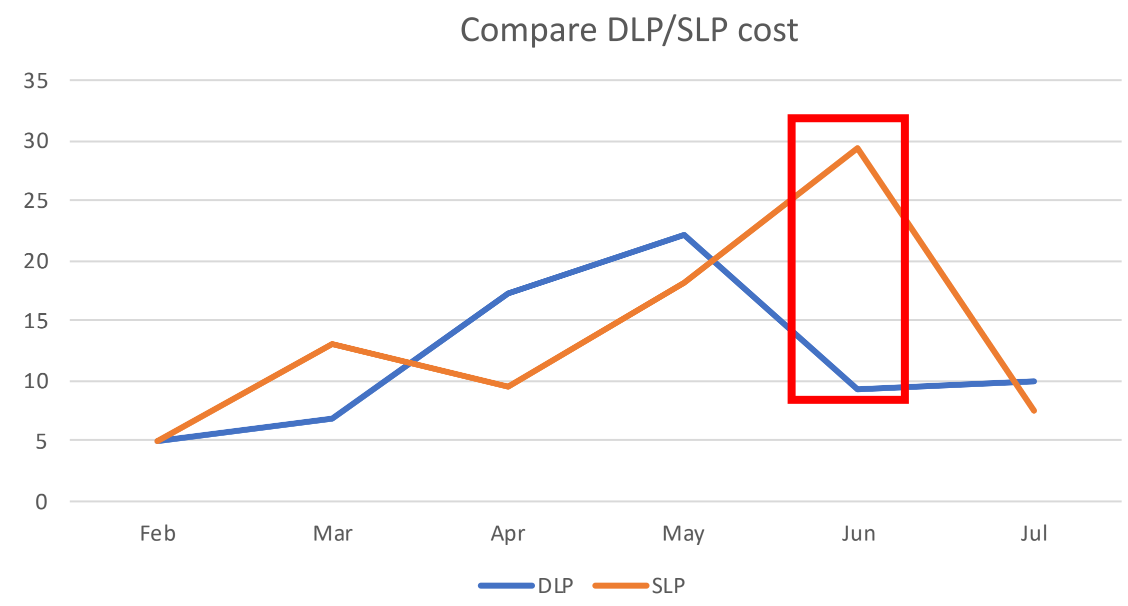
```python
# ____________________________________________________________
#
# ad: Annotated with location of stochastic rhs entries
#       for use with pysp2smps conversion tool.

import itertools
import random

from pyomo.core import *
from pyomo.pysp.annotations import (PySP_ConstraintStageAnnotation,
                                    PySP_StochasticRHSAnnotation)

#
# Define the probability table for the stochastic parameters
#
demand=[0, 110.2848,123.4493,108.7793,111.7802,120.6422]
y_start=101.38

d1_rhs_table=\
[-0.000857745,-0.004360359,0.006247019,-0.007466512,-0.005190053,
0.003739198,-0.000281543,-0.01775575,0.00521002,0.002557547,
0.003315072,0.008304421,-2.209539,-0.5987525,0.6191887,
4.866312,-0.01349693,1.377007,-0.6311139,2.350632,
0.5359557,1.159893,1.646791,-1.402757,-2.018934,
3.587489,0.04279932,-0.8683697,4.851779,0.7572866,
-1.262768,-2.717473,2.686227,-2.43172,-2.64504,
0.406923,-0.9558075,-2.79891,-3.523102,-5.197554,
-1.106004,-0.3417804,6.252632,2.106569,0.1870523,
1.089157,-2.945359,3.39181,-2.86756,-0.9906556,
6.519179,2.175354,-0.5513542,1.097396,-4.053061,
2.083399,4.809785,1.185102,2.337311,5.029688,
-8.733815]
```

DLP Result

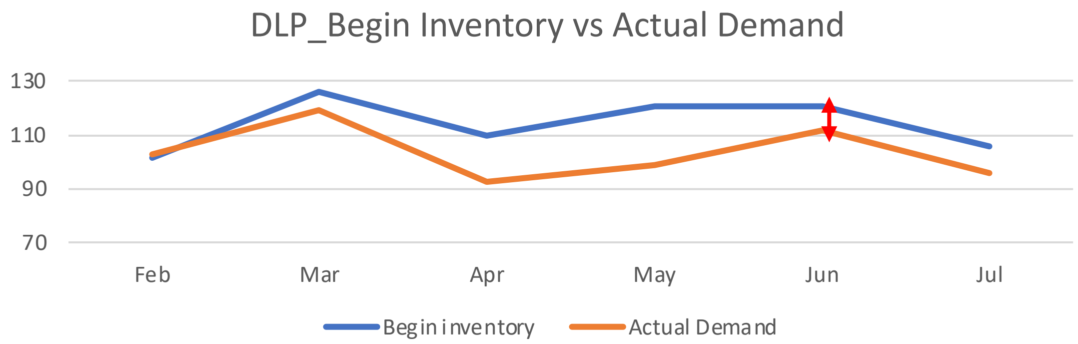
Deterministic model	Feb	Mar	Apr	May	Jun	Jul
Begin inventory	101.38	125.858	109.718	120.891	120.5172	106.0461
Actual demand	103.05	119.06	92.46	98.75	111.14	96.13
End Inventory	0	6.798	17.258	22.141	9.3772	9.9161
Holding cost	0	6.798	17.258	22.141	9.3772	9.9161
Lost sales	1.67	0	0	0	0	0
Lost sales cost	5.01	0	0	0	0	0
Total cost	5.01	6.798	17.258	22.141	9.3772	9.9161
Order	125.858	102.92	103.633	98.3762	96.6689	87.8298

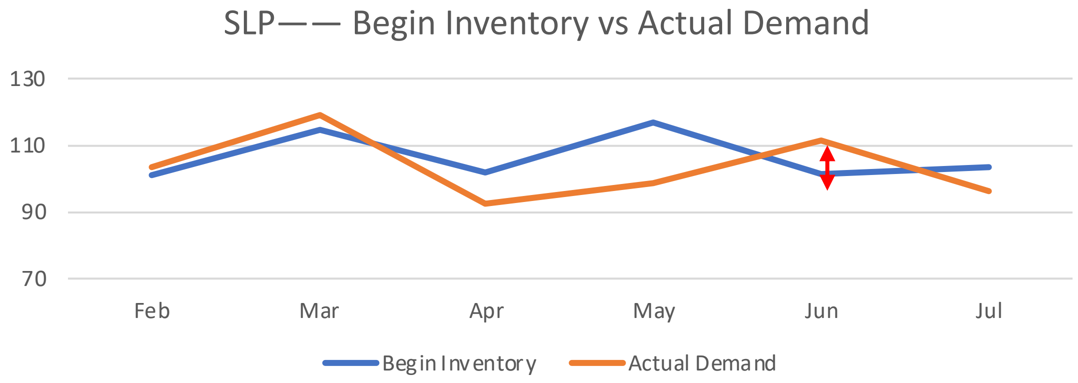
SLP Result

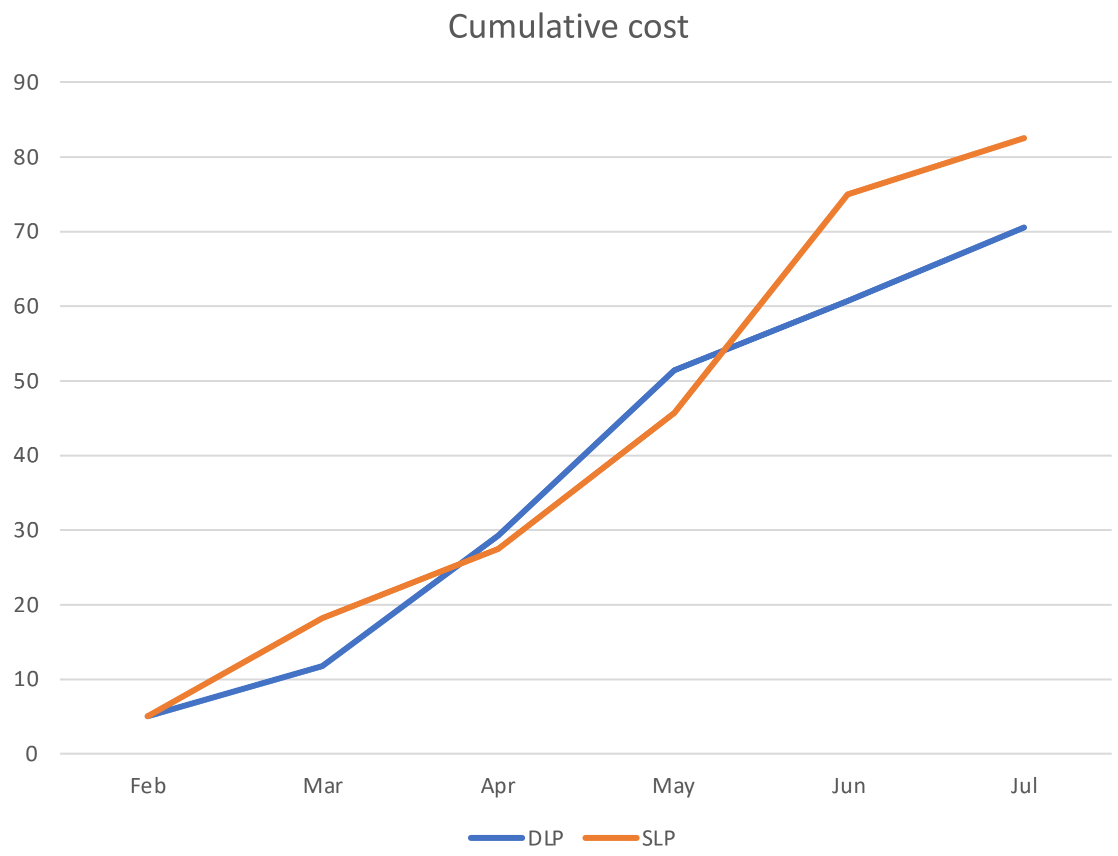
	Feb	Mar	Apr	May	Jun	Jul
Begin inventory	101.38	114.7155	101.9015	116.8272	101.35796	103.6961
Actual demand	103.05	119.06	92.46	98.75	111.14	96.13
End Inventory	0	0	9.4415	18.0772	0	7.5661
Holding cost	0	0	9.4415	18.0772	0	7.5661
Lost sales	1.67	4.3445	0	0	9.78204	0
Lost sales cost	5.01	13.0335	0	0	29.34612	0
Total cost	5.01	13.0335	9.4415	18.0772	29.34612	7.5661
Order	114.7155	101.9015	107.3857	83.28076	103.6961	

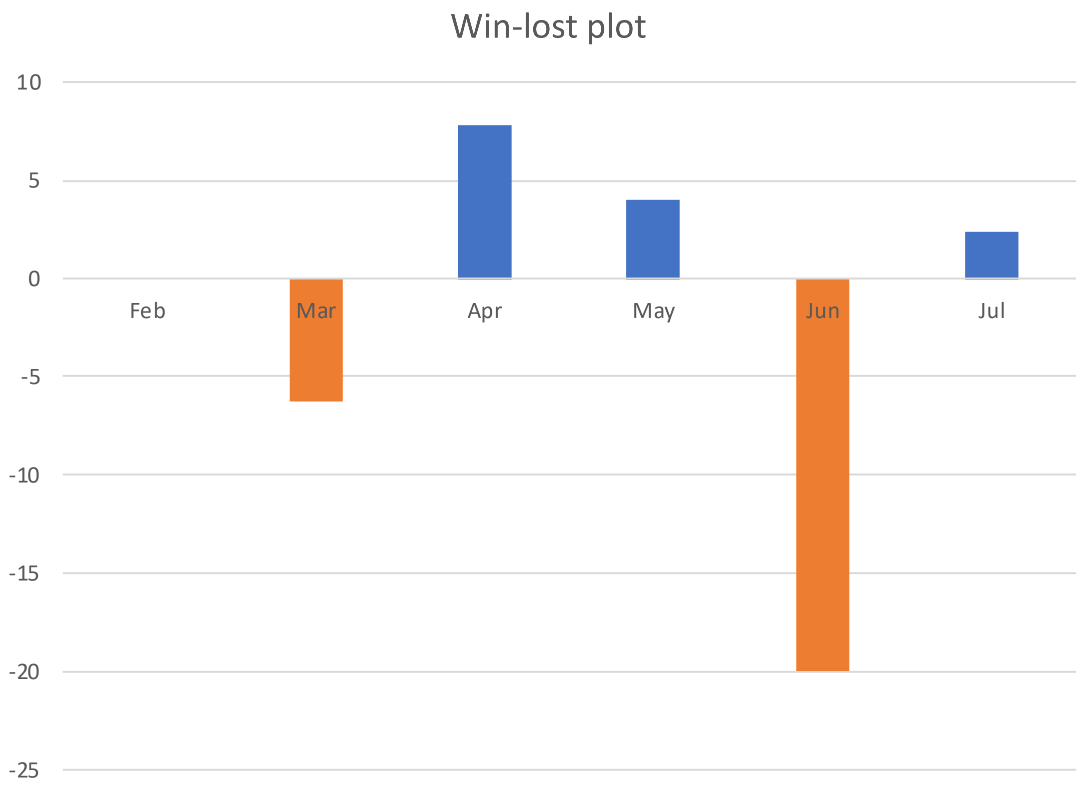


Stochastics - much higher cost in June.
The amount of difference is similar.
But cost of lost sales is 3 times of cost of holding.









Cost of stochastics model is a little higher than the deterministic model

Only for March and June DLP win, but win a lot in June

Conclusion: Still hard to tell which model is perform better, need to validate the cost in the longer period.