Forecasting Bankruptcy Rates

Chris Dong

November 28, 2017

Libaries

```
library(tseries)
library(car)
library(forecast)
library(tidyverse)
library(magrittr)
library(ggcorrplot)
```

Loading Data

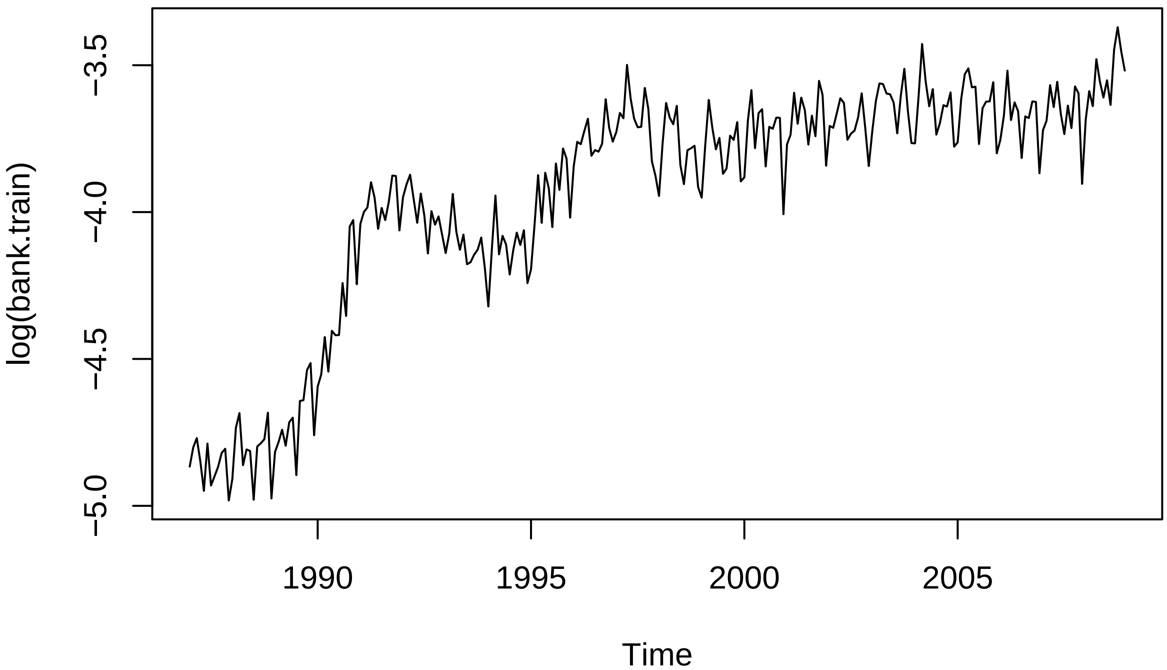
```
train <- read_csv("train.csv")
test <- read_csv("test.csv")

train %<>% na.omit()
bank <-ts(train$Bankruptcy_Rate, start = c(1987, 1), end =c(2010, 12), frequency = 12)
house <-ts(train$House_Price_Index, start = c(1987, 1),
           end =c(2010, 12), frequency = 12)
unemployment <- ts(train$Unemployment_Rate, start = c(1987, 1),
           end =c(2010, 12), frequency = 12)
population <- ts(train$Population, start = c(1987, 1),
           end =c(2010, 12), frequency = 12)
```

Create Training and Validation Set

```
bank.train <- window(bank, start = c(1987,1), end = c(2008,12))
bank.test <- window(bank, start = c(2009,1), end = c(2010,12))
house.train <- window(bank, start = c(1987,1), end = c(2008,12))
house.test <- window(bank, start = c(2009,1), end = c(2010,12))
unemployment.train <- window(unemployment, start = c(1987,1), end = c(2008,12))
unemployment.test <- window(unemployment, start = c(2009,1), end = c(2010,12))
population.train <- window(population, start = c(1987,1), end = c(2008,12))
population.test <- window(population, start = c(2009,1), end = c(2010,12))
```

```
plot(log(bank.train))
```




```r
adf.test(bank.train)
```

```
##
##  Augmented Dickey-Fuller Test
##
## data:  bank.train
## Dickey-Fuller = -2.0486, Lag order = 6, p-value = 0.5554
## alternative hypothesis: stationary
```

```r
bank.train1 <- diff(bank.train)
```

```r
adf.test(bank.train1)$p.value
```

```
## [1] 0.01
```

```r
bank.train2 <- diff(bank.train1, lag = 12)
```

Trying `auto.arima` as baseline

```r
automl <- arima(log(bank.train), order = c(2,0,1),
      seasonal = list(order = c(0,0,2), method = "ML"))
sqrt(mean((exp(forecast(automl, level = 95, h = 24)$mean) - bank.test)^2))
```

```
## [1] 0.004791864
```

```r
result <-c()
orderlist = list()

for(i in 0:3){
  for(j in 0:3){
    for(a in 0:3){
      for(b in 0:3){

        orderlist <- c(orderlist, paste(i,j,a,b))
        bankmodel <- tryCatch({expr = arima(log(bank.train), order = c(i,1,j),
          seasonal = list(order = c(a,1,b), period = 12), method = "ML")},
```

```r
          error = function(cond) {return(NA)})

        rmse <- sqrt(mean((exp(forecast(bankmodel,
                level = 95, h = 24)$mean) - bank.test)^2))
        print(paste(i,j,a,b, ":", rmse))
        ifelse(!is.na(bankmodel), result <- c(result, rmse),
               result <- c(result, NA))
  }
  }
  }
}
save(result, orderlist, file = "bank.RData")
```

```r
load(file = "bank.RData")
```

```r
names(result) <- unlist(orderlist)
head(result[order(result)], n = 25)
```

```
##     0 3 3 3     0 3 2 3     0 2 3 3     0 2 2 3     2 0 3 3     1 1 3 2
## 0.003669244 0.003723792 0.003755809 0.003819419 0.003826357 0.003836567
##     0 0 2 3     1 0 3 3     0 1 3 3     3 1 3 3     2 1 3 3     1 2 3 3
## 0.003872421 0.003883214 0.003884174 0.003887526 0.003941891 0.003986701
##     1 1 2 2     1 0 2 3     0 1 2 2     0 1 3 2     2 3 2 2     3 2 3 3
## 0.003990225 0.003991392 0.003994789 0.003996011 0.004004227 0.004007126
##     2 0 2 2     0 2 2 2     2 2 2 2     3 0 3 3     3 0 3 2     2 0 1 3
## 0.004026622 0.004040001 0.004057212 0.004077495 0.004079149 0.004080659
##     2 1 2 2
## 0.004082576
```

```r
m1 <- arima(log(bank.train), order = c(0,1,3), seasonal = list(order = c(2,1,3), period = 12), method =
m2 <- arima(log(bank.train), order = c(0,1,3), seasonal = list(order = c(3,1,3), period = 12), method =
D <- -2*(m1$loglik - m2$loglik)
pval <- 1-pchisq(D,length(m2$coef) - length(m1$coef))
print(c("Test Statistic:",round(D, 4),"P-value:", round(pval, 4)))
```

```
## [1] "Test Statistic:" "0.0246"          "P-value:"        "0.8754"
```

SARIMA(0,1,3)(2,1,3) better than SARIMA(0,1,3)(3,1,3)

```r
m1 <- arima(log(bank.train), order = c(0,1,2), seasonal = list(order = c(2,1,3), period = 12), method =
m2 <- arima(log(bank.train), order = c(0,1,3), seasonal = list(order = c(2,1,3), period = 12), method =
D <- -2*(m1$loglik - m2$loglik)
pval <- 1-pchisq(D,length(m2$coef) - length(m1$coef))
print(c("Test Statistic:",round(D, 4),"P-value:", round(pval, 4)))
```

```
## [1] "Test Statistic:" "4.1163"          "P-value:"        "0.0425"
```

SARIMA(0,1,3)(2,1,3) better than SARIMA(0,1,2)(2,1,3)

```r
rmse <- function(logmodel) sqrt(mean((exp(forecast(logmodel, level = 95, h = 24)$mean) - bank.test)^2))
```

```r
model <-  arima(log(bank.train), order = c(0,1,3), seasonal = list(order = c(2,1,3), period = 12), meth
(score <- rmse(model))
```

```
## [1] 0.003723792
```

So far, an SARIMA(0,1,3)(2,1,3) gets a RMSE of 0.0037238 when forecasting from January 2009 to December 2010.

Holt-Winters

```r
hw <- HoltWinters(x = log(bank.train),seasonal = "add",
        alpha = 0.2, beta = 0.2, gamma = 0.4)
rmse(hw)
```

```
## [1] 0.0135522
```

Additive Triple Exponential Smoothing

```r
hwresult <-c()
hworderlist = list()

for(i in seq(0.05,1, by = 0.05)){
  for(j in seq(0.05,1, by = 0.05)){
    for(a in seq(0.05,1, by = 0.05)){

        hworderlist <- c(hworderlist, paste(i,j,a))
        bankmodel <- HoltWinters(x = log(bank.train),seasonal = "add",
                     alpha = i, beta = j, gamma = a)
        measure <- rmse(bankmodel)
        print(paste(i,j,a, ":", measure))
        hwresult <- c(hwresult, measure)

  }
  }
}
save(hwresult, hworderlist, file = "holt.RData")
```

```r
head(holt_add[order(holt_add)], n = 10)
```

```
## 0.25 0.65 0.35   0.7 0.95 0.55   0.75 0.9 0.7   0.4 0.7 0.35  0.75 0.9 0.75
##    0.004401478    0.004427086    0.004645165    0.004703887    0.004714159
##   0.2 0.05 0.2  0.2 0.05 0.25  0.2 0.05 0.15   0.2 0.05 0.3  0.2 0.05 0.35
##    0.004744472    0.004748176    0.004751193    0.004757214    0.004769387
```

Multiplicative Triple Exponential Smoothing

```r
hwresult <-c()
hworderlist = list()

for(i in seq(0.05,1, by = 0.05)){
  for(j in seq(0.05,1, by = 0.05)){
    for(a in seq(0.05,1, by = 0.05)){

        hworderlist <- c(hworderlist, paste(i,j,a))
        bankmodel <- HoltWinters(x = log(bank.train),seasonal = "mult",
                     alpha = i, beta = j, gamma = a)
        measure <- rmse(bankmodel)
        print(paste(i,j,a, ":", measure))
        hwresult <- c(hwresult, measure)
```

```
  }
  }
}
save(hwresult, hworderlist, file = "holt2.RData")
```

```
head(holt_mult[order(holt_mult)], n = 10)
```

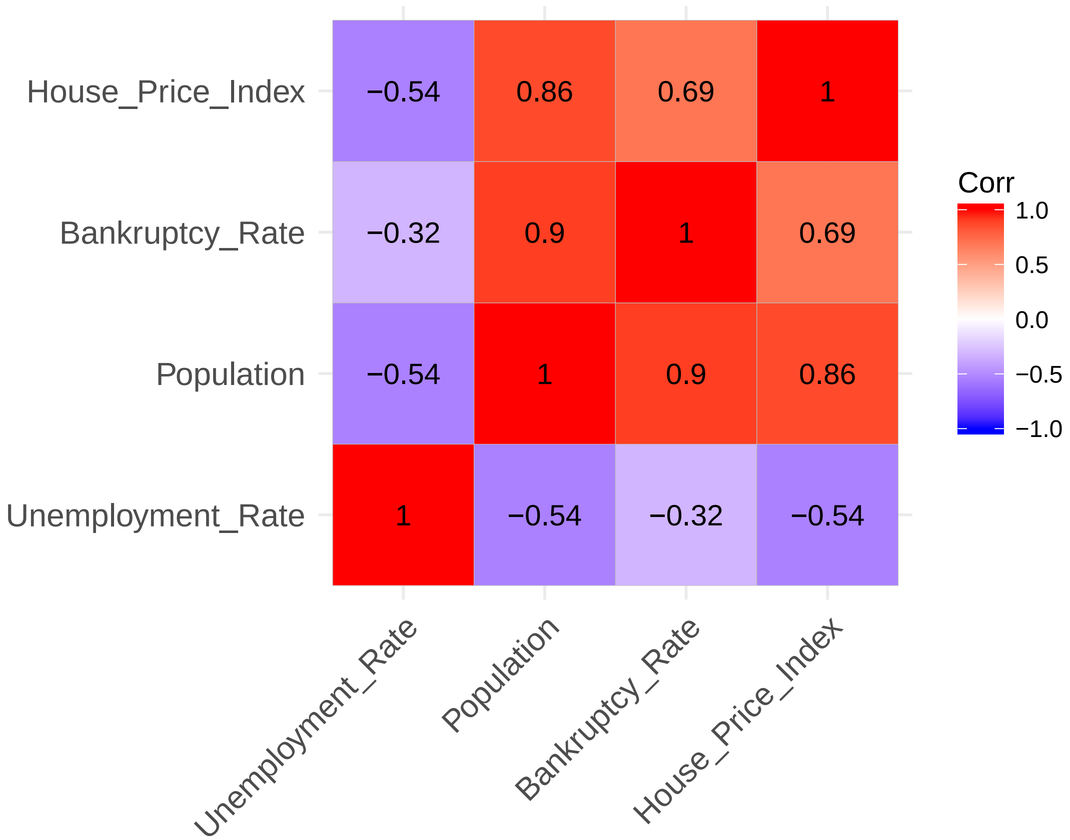
```
##    0.6 0.9 0.75       0.3 1 0.2    0.3 0.95 0.2     0.6 0.9 0.7  0.65 0.8 0.95
##     0.003863030     0.003945563     0.004082934     0.004086046     0.004130179
##    0.65 0.8 0.9     0.55 1 0.55   0.6 0.95 0.65  0.75 0.95 0.9 0.35 0.85 0.25
##     0.004135408     0.004438784     0.004497856     0.004507986     0.004634386
```

```
#save(result, holt_add, holt_mult, file = "bank.RData")
```

Multivariate

Correlation Matrix

```
train[,-1] %>% na.omit() %>% cor() %>% ggcorrplot(lab = T)
```



SARIMAX

```
model.population <-  arima(log(bank.train), order = c(0,1,3),
                 seasonal = list(order = c(2,1,3), period = 12),
```

```r
                method = "ML",
                xreg = data.frame(population.train))
```

```r
(score2 <- sqrt(mean((exp(forecast(model.population, level = 95, h = 24,
                xreg = data.frame(population.test))$mean) - bank.test)^2)))
```

```
## [1] 0.003216034
```

Population improved RSME from 0.0037238 to 0.003216.

Trying Population + Housing Price Index

```r
model.unemploy.pop <-  arima(log(bank.train), order = c(0,1,3),
                seasonal = list(order = c(2,1,3), period = 12),
                method = "ML",
                xreg = data.frame(population.train, house.train))
```

```r
(score3 <- sqrt(mean((exp(forecast(model.unemploy.pop, level = 95, h = 24,
                xreg = data.frame(population.test, house.test))$mean) - bank.test)^2)))
```

```
## [1] 0.003194314
```

Comparing Population and (Housing + Population) with Log-Likelihood Test

```r
D <- -2*(model.population$loglik - model.unemploy.pop$loglik)
pval <- 1-pchisq(D,length(model.unemploy.pop$coef) - length(model.population$coef))
print(c("Test Statistic:",round(D, 4),"P-value:", round(pval, 4)))
```

```
## [1] "Test Statistic:" "506.086"         "P-value:"        "0"
```

Having both variables is indeed better.

Trying three variables

```r
model.allthree <-  arima(log(bank.train), order = c(0,1,3),
                seasonal = list(order = c(2,1,3), period = 12),
                method = "ML",
                xreg = data.frame(population.train, house.train, unemployment.train))
```

```r
(score4 <- sqrt(mean((exp(forecast(model.allthree, level = 95, h = 24,
                xreg = data.frame(population.test, house.test,
                                  unemployment.test))$mean) - bank.test)^2)))
```

```
## [1] 0.003267467
```

Doesn't seem better, let's try running a log-likelihood test

```r
D <- -2*(model.unemploy.pop$loglik - model.allthree$loglik)
pval <- 1-pchisq(D,length(model.allthree$coef) - length(model.unemploy.pop$coef))
print(c("Test Statistic:",round(D, 4),"P-value:", round(pval, 4)))
```

```
## [1] "Test Statistic:" "0.2221"          "P-value:"        "0.6374"
```

Taking the log of Population

```
model.unemploy.pop.log <-  arima(log(bank.train), order = c(0,1,3),
                seasonal = list(order = c(2,1,3), period = 12),
                method = "ML",
                xreg = data.frame(log(population.train), house.train))
(score3.log <- sqrt(mean((exp(forecast(model.unemploy.pop.log, level = 95, h = 24,
                xreg = data.frame(log(population.test), house.test))$mean) - bank.test)^2)))
```

```
## [1] 0.003186518
```

```
model.unemploy.pop.log2 <-  arima(log(bank.train), order = c(0,1,3),
                seasonal = list(order = c(2,1,3), period = 12),
                method = "ML",
                xreg = data.frame(log(population.train), log(house.train)))
(score3.log2 <- sqrt(mean((exp(forecast(model.unemploy.pop.log2, level = 95, h = 24,
                xreg = data.frame(log(population.test), log(house.test)))$mean) - bank.test)^2)))
```

```
## [1] 3.397136e-17
```

Why is this so low..? Any Ideas?

Current Best Model

The best model so far is a SARIMAX (0,1,3)(2,1,3) along with the explanatory variables Population and Housing Price Index. It has a RMSE of 0.0031865.