OPIM314 Enabling Technologies

Individual Assignment 1: Predicting Future Car Sales

James Wang

09/30/15

Annotated R Code

```
setwd("/Users/jameswang/Downloads")
data = read.csv("Assignment1Data.csv")
str(data)
summary(data)
attach(data)

### 1. In-Sample Prediction
#### a. simple OLS regression
base <- lm(log_sales~log_sales_l2+isSummer+isWinter)
coefficients(base) # B1 = 0.974195368

#### b. predicted values for auto sales in July 2015
target_row <- which(year==2015 & month==7)
log_pred_July2015 <- predict(base,data[target_row,])
pred_July2015 <- exp(log_pred_July2015) # predicted sales (07/2015) = 85845.99

#### c. accuracy: mean absolute deviation (MAE)
log_pred_values <- predict(base,data)
pred_vales <- exp(log_pred_values)
error <- (abs(pred_vales - auto_sales)) / auto_sales
MAE <- mean(error, na.rm=TRUE) # 0.03072587
```

OPIM314 Enabling Technologies

Individual Assignment 1: Predicting Future Car Sales
James Wang
09/30/15

2. Add Google Trend variables

```
google_trends <- list(GT_vehicleshopping,GT_vehiclemaintenance,GT_suvs,GT_vansminivans)
MAE_list <- list()
for (i in google_trends) {
  fit <- lm(log_sales~log_sales_l2+i+isSummer+isWinter)
  predictions <- exp(predict(fit,data))
  err <- (abs(predictions - auto_sales)) / auto_sales
  mean_sq_err <- mean(err, na.rm=TRUE)
  MAE_list <- c(MAE_list,mean_sq_err)
}
# GT_vehicleshopping     MAE: 0.03017393 --> worst-performing, better than base
# GT_vehiclemaintenance  MAE: 0.02984057
# GT_suvs                MAE: 0.03012479
# GT_vansminivans        MAE: 0.02978202 --> best-performing, better than base
```

OPIM314 Enabling Technologies

Individual Assignment 1: Predicting Future Car Sales
James Wang
09/30/15

3. moving window

```
window = 8
end = nrow(data) - window
pred_list <- list()
for (i in 2:end) {
  fit = lm(log_sales~log_sales_l2+isSummer+isWinter,data=data[i:(i+window-1),])
  pred_value <- predict(fit,data[i+window,])
  prediction <- exp(pred_value)
  pred_list <- c(pred_list,prediction)
}

error_3 <- abs((unlist(pred_list) - auto_sales[(2+window):nrow(data)]) /
auto_sales[(2+window):nrow(data)])
MAE_3 <- mean(error_3, na.rm=TRUE) # MAE = 0.03172361
```

OPIM314 Enabling Technologies

Individual Assignment 1: Predicting Future Car Sales
James Wang
09/30/15

4. moving window + Google Trend variables

```
google_trends <- list(GT_vehicleshopping,GT_vehiclemaintenance,GT_suvs,GT_vansminivans)
window = 8
end = nrow(data) - window
MAE_list_4 <- list()
for (i in google_trends) {
  pred_list4 <- list()
  gt = unlist(i)
  data$gt <- gt
  for (j in 2:end) {
    fit4 = lm(log_sales~log_sales_l2+gt+isSummer+isWinter,data=data[j:(j+window-1),])
    pred_value4 <- predict(fit4,data[j+window,])
    prediction4 <- exp(pred_value4)
    pred_list4 <- c(pred_list4,prediction4)
  }
  error_4 <- abs((unlist(pred_list4) - auto_sales[(2+window):nrow(data)]) /
auto_sales[(2+window):nrow(data)])
  MAE_4 <- mean(error_4, na.rm=TRUE)
  MAE_list_4 <- c(MAE_list_4,MAE_4)
}

# moving window + GT_vehicleshopping     MAE: 0.03759918 --> worst-performing, performs worse than base (to be expected on test data)
# moving window + GT_vehiclemaintenance  MAE: 0.03428805 --> best-performing, performs worse than base (to be expected on test data)
# moving window + GT_suvs             MAE: 0.03605072
# moving window + GT_vansminivans       MAE: 0.03464422
```

OPIM314 Enabling Technologies

Individual Assignment 1: Predicting Future Car Sales
James Wang
09/30/15

5. find a model that beats the baseline (3)

```
google_trends <- list(GT_vehicleshopping,GT_vehiclemaintenance,GT_suvs,GT_vansminivans)
window_5 <- 6:24
best_window_list <- list()
for (k in window_5) {
  end = nrow(data) - k
  MAE_list_5 <- list()
  for (i in google_trends) {
    pred_list5 <- list()
    gt = unlist(i)
    data$gt <- gt
    for (j in 2:end) {
      fit5 = lm(log_sales~log_sales_l2+gt+isSummer+isWinter,data=data[j:(j+k-1),])
      pred_value5 <- predict(fit5,data[j+k,])
      prediction5 <- exp(pred_value5)
      pred_list5 <- c(pred_list5,prediction5)
    }
    error_5 <- abs((unlist(pred_list5) - auto_sales[(2+k):nrow(data)]) / auto_sales[(2+k):nrow(data)])
    MAE_5 <- mean(error_5, na.rm=TRUE)
    MAE_list_5 <- c(MAE_list_5,MAE_5)
  }
  best_window_list <- c(best_window_list,MAE_list_5)
}
best_window_mat <-
matrix(unlist(best_window_list),ncol=length(google_trends),nrow=length(window_5),byrow=TRUE)
min_list <- list()
for (i in 1:(ncol(best_window_mat))) {
  min <- min(best_window_mat[,i])
  index <- which(best_window_mat[,i]==min)
  min_list <- c(min_list,index,min)
}
min_mat <- matrix(unlist(min_list),ncol=4,nrow=2)

# GT_vehicleshopping + best window size (22)      MAE: 0.03234851
# GT_vehiclemaintenance + best window size (9)    MAE: 0.03119291
# GT_suvs + best window size (21)                 MAE: 0.03126661
# GT_vansminivans + best window size (23)         MAE: 0.03058064 --> best, % absolute improvement =
0.036029
```

OPIM314 Enabling Technologies

Individual Assignment 1: Predicting Future Car Sales

James Wang

09/30/15

6. optimal window for housing data?

I do not believe so. Housing sales data will be much different than auto sales data. We cannot extrapolate assume that because a particular window size performed well on one set of data, it will perform well another set of data. It could very well be that 23 is the optimal size for housing data, but we need to check be sure.

7. Google Trends variable do not improve baseline model for housing sales?

Perhaps the Google Trends variables do not add any new information to the baseline model. In this case, adding Google Trends variables would introduce multicollinearity in the model, as the predictors as a whole would be highly correlated to each other because they are explaining the same variance in the response. Thus, the overall model performance will go down. In addition to the multicollinearity effect, the parsimony principle may also have an effect. The parsimony principle implies that simpler models perform better on data because the lower number of predictors used means the model is less likely to overfit the training data.

OPIM314 Enabling Technologies

Individual Assignment 1: Predicting Future Car Sales
James Wang
09/30/15

8. Error in prediction plot by year

```
window8 = 23
end8 = nrow(data) - window8
pred_list <- list()
for (i in 2:end8) {
  fit = lm(log_sales~log_sales_l2+GT_vansminivans+isSummer+isWinter,data=data[i:(i+window8-1),])
  pred_value <- predict(fit,data[i+window8,])
  prediction <- exp(pred_value)
  pred_list <- c(pred_list,prediction)
}
error_in_pred <- unlist(pred_list) - auto_sales[(2+window8):nrow(data)]
years = unique(data$year[(window8 + 2):nrow(data)])
count = 1
year_avg_list <- list()
for (i in 1:10) {
  year_avg = mean(error_in_pred[(count):(count+11)],na.rm=TRUE)
  year_avg_list <- c(year_avg_list,year_avg)
  count = count+12
}
plot(years, unlist(year_avg_list), xlab="Year", ylab="Error in Prediction (Avg from
months)",main="Error in Prediction by Year")
abline(a = 0, b = 0, col = "red")
```

Individual Assignment 1: Predicting Future Car Sales

James Wang

09/30/15

```
#### 9. Error in prediction plot by month
num_of_months =  1:length(error_in_pred)
labels = list()
for (i in 1:120) {
  labels = c(labels,NA)
}
labels_vec = unlist(labels)
counter = 1
counter2 = 1
for (i in 1:10) {
  labels_vec[counter2] = years[counter]
  counter = counter+1
  counter2 = counter2+12
}
plot(num_of_months, error_in_pred, xlab="Year", ylab="Error in Prediction",main="Error in Prediction by Year",xaxt="n")
axis(side=1,at=num_of_months,labels=labels_vec)
abline(a = 0, b = 0, col = "blue")
```

OPIM314 Enabling Technologies

Individual Assignment 1: Predicting Future Car Sales

James Wang

09/30/15

10. Use health and beauty variables?
No. Correlation does not imply causation. Using variables that are simply correlated with the response often introduces over-fitting, because the model is capturing more noise than it should be. It is critical to use common sense when choosing predictors in models.

OPIM314 Enabling Technologies

Individual Assignment 1: Predicting Future Car Sales
James Wang
09/30/15

11. 5 years from now?
I do not think that the model will perform well 5 years from now. The model will perform increasingly worse as time goes on, simply because the model cannot predict data that it has not seen before. For example, certain events like a recession or new innovation in the car industry can drastically effect sales, and the model cannot account for that since it happens in the future.
A model that would be stable in 5 years would be a model that predicts the exact position of the Earth around the Sun. Our models for tracking Earth's position have accounted for many factors such as gravitational forces of the Sun and other planets.
A model that would not be stable is one that tries to predict the S&P 500 in 5 years. There are simply too many unpredictable events that will happen in the future and our model cannot possibly account for them.

OPIM314 Enabling Technologies

Individual Assignment 1: Predicting Future Car Sales
James Wang
09/30/15

```
#### 12. Predict October 2015 ####

setwd("/Users/jameswang/Downloads")
data1 = read.csv("Assignment1Data.csv")
data2 = read.csv("gt_variables.csv")

#### Prepare the data
years = 2011:2015
months = 1:12

var_2010 <- c(2010)
list_2010 <- list()
for (i in var_2010) {
  for (j in 10:12) {
    targets <- data2[which(data2$Year==i & data2$Month==j),]
    month_avg <- colMeans(targets,na.rm=TRUE)
    list_2010 <- c(list_2010,month_avg)
  }
}
mat <- unlist(list_2010)
data_2010 <- matrix(mat,ncol=6,byrow=TRUE)

mat_list <- list()
for (i in years) {
  for (j in months) {
    targets <- data2[which(data2$Year==i & data2$Month==j),]
    month_avg <- colMeans(targets,na.rm=TRUE)
    mat_list <- c(mat_list,month_avg)
    if (j==9 & i==2015) break
  }
}
mat <- unlist(mat_list)
data_rest <- matrix(mat,ncol=6,byrow=TRUE)

data <- rbind(data_2010,data_rest)
data <- as.data.frame(data)
colnames(data) <- c("year","month","brands","trucks_suvs","hybrids","specs")

data1 <- data1[3:62,]
data_full <- cbind(data1,data)
dataset <- data_full[,-c(5,6)]
```

OPIM314 Enabling Technologies

Individual Assignment 1: Predicting Future Car Sales
James Wang
09/30/15

```
row.names(dataset) <- 1:nrow(dataset)

#install.packages("DataCombine")
#library(DataCombine)
dataset_lags1 <- slide(dataset,Var="auto_sales",slideBy=-1)
dataset_lags2 <- slide(dataset_lags1,Var="auto_sales",slideBy=-2)
dataset_lags3 <- slide(dataset_lags2,Var="auto_sales",slideBy=-3)
colnames(dataset_lags3) <-
c("year","month","auto_sales","local.dealerships","brands","trucks_suvs","hybrids","specs","auto_sales_1","auto_sales_2","auto_sales_3")

#### Data Exploration
summary(dataset_lags3)
pairs(dataset_lags3)
head(dataset_lags3)
str(dataset_lags3)
len <- 1:length(dataset_lags3$month)
plot(len,log(dataset_lags3$auto_sales)) # auto_sales exhibits a positive, linear trend over time
abline(lm(log(dataset_lags3$auto_sales)~len))

plot(t,log(dataset_lags3$auto_sales))
abline(lm(log(dataset_lags3$auto_sales)~t))

#### Model Building
### Step 1. Fit models
### Step 2. Perform model selection (use BIC)
### Step 3. Test model assumptions (linearity, normality, homoscedastic residuals, no autocorrelated residuals)
### Step 4. Fix model accordingly if assumptions are violated
### Step 5. Tune model on test data
### Step 6. Average the model's predictions

## Attempt 1: Linear Regression
fit <- lm(auto_sales~local.dealerships+brands+trucks_suvs+hybrids+specs,data=dataset)
summary(fit)
BIC(fit) # 1109.958

fit1 <- lm(log(auto_sales)~local.dealerships+brands+trucks_suvs+specs,data=dataset)
summary(fit1)
BIC(fit1) # -208.3966
```

OPIM314 Enabling Technologies

Individual Assignment 1: Predicting Future Car Sales
James Wang
09/30/15

```
fit2 <-
lm(log(auto_sales)~local.dealerships+brands+trucks_suvs+specs+I(auto_sales_1),data=dataset_lags3)
summary(fit2)
BIC(fit2) # -281.4357

fit2a <- lm(log(auto_sales)~specs+I(auto_sales_1),data=dataset_lags3)
summary(fit2a)
BIC(fit2a) # -289.9299 <- pretty much the same as fit3

fit3 <- lm(log(auto_sales)~I(auto_sales_1),data=dataset_lags3)
summary(fit3)
BIC(fit3) # -290.0594 <- lower BIC/SIC means better model (more parsimonious)

# Residual Plot for fit1
fit.res3 <- resid(fit3)
num_of_months =  1:length(fit.res3)
plot(num_of_months, fit.res3, xlab="Time", ylab="Error",main="Residual Plot for fit1")
abline(a = 0, b = 0, col = "blue") # residuals look good

# Durbin-Watson test for autocorrelation
#install.packages("car")
#library(car)
dwt(fit3,simulate=TRUE) # 2.052544 <- ~2 ... no autocorrelation

# plot normal qq plots ... test for normality
qqPlot(fit3,main="Normal QQ Plot for FIT3 Residuals") # residuals look normally distributed
hist(resid(fit3)) # slightly left-skewed

# conduct RESET test ... test for linearity
#install.packages("lmtest")
#library(lmtest)
reset(fit3,power=2:3,type="regressor",data=dataset_lags3)  # p-value = 0.01823 is significant at the
0.05 level, strong evidence that function form is not linear!

# conduct Breusch-Pagan Test ... test for heteroskedasticity
bptest(fit3) # p-value = 0.524

# plot leverage plot ... identify influential points
plot(fit3) # see the 4th plot ... no influential points

# conduct Bonferroni-adjusted outlier test ... test for outliers ...
```

outlierTest(fit3) #p-value = 0.013242 ... outliers are present, but they are not influential points (as shown by the leverage plot & Cook's distance)

Attempt 2: Time Series Regression ... given that the data seems to exhibit a trend over time and the RESET test showed strong evidence for non-linear form

```
# Create quarterly/seasonal dummies
q1.body = matrix(data = rep(c(rep(1,3),rep(0,9)),4),nrow=48,ncol=1)
q2.body = matrix(data = rep(c(rep(0,3),rep(1,3),rep(0,6)),4),nrow=48,ncol=1)
q3.body = matrix(data = rep(c(rep(0,6),rep(1,3),rep(0,3)),4),nrow=48,ncol=1)
q4.body = matrix(data = rep(c(rep(0,9),rep(1,3)),4),nrow=48,ncol=1)

q1.front = rbind(c(0),rbind(c(0),rbind(c(0),q1.body)))
q2.front = rbind(c(0),rbind(c(0),rbind(c(0),q2.body)))
q3.front = rbind(c(0),rbind(c(0),rbind(c(0),q3.body)))
q4.front = rbind(c(1),rbind(c(1),rbind(c(1),q4.body)))

q1.back = matrix(data = rep(c(rep(1,3),rep(0,6)),1),nrow=9,ncol=1)
q2.back = matrix(data = rep(c(rep(0,3),rep(1,3),rep(0,3)),1),nrow=9,ncol=1)
q3.back = matrix(data = rep(c(rep(0,6),rep(1,3)),1),nrow=9,ncol=1)
q4.back = matrix(data = rep(c(rep(0,9)),1),nrow=9,ncol=1)

q1 <- rbind(q1.front,q1.back)
q2 <- rbind(q2.front,q2.back)
q3 <- rbind(q3.front,q3.back)
q4 <- rbind(q4.front,q4.back)

# Create monthly dummyies
oct <- matrix(data = rep(c(rep(1,1),rep(0,11)),5),nrow=60,ncol=1)
nov <- matrix(data = rep(c(rep(0,1),rep(1,1),rep(0,10)),5),nrow=60,ncol=1)
dec <- matrix(data = rep(c(rep(0,2),rep(1,1),rep(0,9)),5),nrow=60,ncol=1)
jan <- matrix(data = rep(c(rep(0,3),rep(1,1),rep(0,8)),5),nrow=60,ncol=1)
feb <- matrix(data = rep(c(rep(0,4),rep(1,1),rep(0,7)),5),nrow=60,ncol=1)
mar <- matrix(data = rep(c(rep(0,5),rep(1,1),rep(0,6)),5),nrow=60,ncol=1)
apr <- matrix(data = rep(c(rep(0,6),rep(1,1),rep(0,5)),5),nrow=60,ncol=1)
may <- matrix(data = rep(c(rep(0,7),rep(1,1),rep(0,4)),5),nrow=60,ncol=1)
jun <- matrix(data = rep(c(rep(0,8),rep(1,1),rep(0,3)),5),nrow=60,ncol=1)
jul <- matrix(data = rep(c(rep(0,9),rep(1,1),rep(0,2)),5),nrow=60,ncol=1)
aug <- matrix(data = rep(c(rep(0,10),rep(1,1),rep(0,1)),5),nrow=60,ncol=1)
sep <- matrix(data = rep(c(rep(0,11),rep(1,1)),5),nrow=60,ncol=1)

# Resume model building
```

OPIM314 Enabling Technologies

Individual Assignment 1: Predicting Future Car Sales
James Wang
09/30/15

```
log_auto_sales_ts = ts(log(dataset_lags3$auto_sales))
t = seq(1,length(log_auto_sales_ts))

fit_ts = lm(log_auto_sales_ts~t+auto_sales_1,data=dataset_lags3)
summary(fit_ts)
BIC(fit_ts) # -306.2341 <- lowest BIC/SIC

fit_ts1 = lm(log_auto_sales_ts~t+I(q1)+I(q2)+I(q3)+I(q4)+I(auto_sales_1),data=dataset_lags3)
summary(fit_ts1)
BIC(fit_ts1) # -294.9015

fit_ts2 =
lm(log_auto_sales_ts~t+oct+nov+dec+jan+feb+mar+apr+may+jun+jul+aug+sep+I(auto_sales_1),data=d
ataset_lags3)
summary(fit_ts2)
BIC(fit_ts2) # -266.5085

# Plot time series
plot.ts(t,log_auto_sales_ts,main="Log of Auto Sales over Time")
par(cex=0.60)
legend("bottomright",c("fit_ts","fit_ts1","fit_ts2","fit3","fit2"),lty=1,col=c("red","green","blue","purple
","yellow"))
lines(fitted(fit_ts),col="red")
lines(fitted(fit_ts1),col="green")
lines(fitted(fit_ts2),col="blue")
lines(fitted(fit3),col="purple")
lines(fitted(fit2),col="yellow")

plot(t,log(dataset_lags3$auto_sales))
abline(lm(log(dataset_lags3$auto_sales)~t))

## Predictions & Forecasts
# Predict September
coefficients(fit_ts)
pred_1 <- exp((10.64773 + (0.004233639  * 60) + ((5.541855 * (10^(-6))) * 85827)))
# fit_ts:  prediction(Sep): 87324.74

coefficients(fit_ts1)
pred_2 <- exp((1.065546 * (10^1)) + ((4.301679 * (10^-3)) * 60) + ((5.431493 * (10^-6)) * 85827))
# fit_ts1: prediction(Sep): 87529.35
```

OPIM314 Enabling Technologies

Individual Assignment 1: Predicting Future Car Sales
James Wang
09/30/15

```
coefficients(fit_ts2)
pred_3 <- exp((1.063763 * (10^1)) + ((4.133104 * (10^-3)) * 60) + ((5.791333 * (10^-6)) * 85827))
# fit_ts2: prediction(Sep): 87787.02

values <- c(pred_1,pred_2,pred_3)
auto_sales_sep <- mean(values) # 87547.04

# Predict October
coefficients(fit_ts)
pred_4 <- exp((10.64773 + (0.004233639  * 60) + ((5.541855 * (10^(-6))) * 87547.04)))
# fit_ts:  prediction(Sep): 87324.74

coefficients(fit_ts1)
pred_5 <- exp((1.065546 * (10^1)) + ((4.301679 * (10^-3)) * 60) + ((5.431493 * (10^-6)) * 87547.04))
# fit_ts1: prediction(Sep): 87529.35

coefficients(fit_ts2)
pred_6 <- exp((1.063763 * (10^1)) + ((4.133104 * (10^-3)) * 60) + ((-4.321430 * (10^-3)) * 1) +
((5.791333 * (10^-6)) * 87547.04))        # fit_ts2: prediction(Sep): 87787.02

values2 <- c(pred_4,pred_5,pred_6)
auto_sales_oct <- mean(values2) # 88265.19
```

Final prediction for October 2015 car sales: $88,265.19 (in millions)