Forecasting Elantra Sales

Quang Duong

2017-06-26

An important application of linear regression is understanding sales. Consider a company that produces and sells a product. In a given period, if the company produces more units than how many consumers will buy, the company will not earn money on the unsold units and will incur additional costs due to having to store those units in inventory before they can be sold. If it produces fewer units than how many consumers will buy, the company will earn less than it potentially could have earned. Being able to predict consumer sales, therefore, is of first order importance to the company.

In this problem, we will try to predict monthly sales of the Hyundai Elantra in the United States. The Hyundai Motor Company is a major automobile manufacturer based in South Korea. The Elantra is a car model that has been produced by Hyundai since 1990 and is sold all over the world, including the United States. We will build a linear regression model to predict monthly sales using economic indicators of the United States as well as Google search queries.

The file elantra.csv contains data for the problem. Each observation is a month, from January 2010 to February 2014. For each month, we have the following variables:

- **Month** = the month of the year for the observation (1 = January, 2 = February, 3 = March, . . .).
- **Year** = the year of the observation.
- **ElantraSales** = the number of units of the Hyundai Elantra sold in the United States in the given month.
- **Unemployment** = the estimated unemployment percentage in the United States in the given month.
- **Queries** = a (normalized) approximation of the number of Google searches for “hyundai elantra” in the given month.
- **CPI_energy** = the monthly consumer price index (CPI) for energy for the given month.
- **CPI_all** = the consumer price index (CPI) for all products for the given month; this is a measure of the magnitude of the prices paid by consumer households for goods and services (e.g., food, clothing, electricity, etc.).

Loading the data

Load the data set. Split the data set into training and testing sets as follows: place all observations for 2012 and earlier in the training set, and all observations for 2013 and 2014 into the testing set.

```r
elantra <- read.csv('elantra.csv')
attach(elantra)
train <- subset(elantra, Year<=2012)
test <- subset(elantra, Year>2012)
str(train)
summary(train)
```

```
'data.frame':   36 obs. of  7 variables:
 $ Month       : int  1 1 1 2 2 2 3 3 3 4 ...
 $ Year        : int  2010 2011 2012 2010 2011 2012 2010 2011 2012 2010 ...
 $ ElantraSales: int  7690 9659 10900 7966 12289 13820 8225 19255 19681 9657 ...
 $ Unemployment: num  9.7 9.1 8.2 9.8 9 8.3 9.9 9 8.2 9.9 ...
 $ Queries     : int  153 259 354 130 266 296 138 281 303 132 ...
 $ CPI_energy  : num  213 229 244 210 232 ...
 $ CPI_all     : num  217 221 228 217 222 ...
     Month            Year       ElantraSales    Unemployment
 Min.   : 1.00   Min.   :2010   Min.   : 7690   Min.   :7.800
 1st Qu.: 3.75   1st Qu.:2010   1st Qu.:10690   1st Qu.:8.200
 Median : 6.50   Median :2011   Median :14449   Median :9.000
 Mean   : 6.50   Mean   :2011   Mean   :14462   Mean   :8.878
 3rd Qu.: 9.25   3rd Qu.:2012   3rd Qu.:18238   3rd Qu.:9.500
 Max.   :12.00   Max.   :2012   Max.   :22100   Max.   :9.900
    Queries        CPI_energy       CPI_all
 Min.   :130.0   Min.   :204.2   Min.   :217.3
 1st Qu.:175.2   1st Qu.:215.8   1st Qu.:218.8
 Median :270.5   Median :242.6   Median :225.3
 Mean   :264.6   Mean   :233.9   Mean   :224.2
 3rd Qu.:344.2   3rd Qu.:247.1   3rd Qu.:228.7
 Max.   :427.0   Max.   :256.4   Max.   :231.7
```

A Linear Regression Model

Build a linear regression model to predict monthly Elantra sales using Unemployment, CPI_all, CPI_energy and Queries as the independent variables. Use all of the training set data to do this.

```
saleReg <- lm(ElantraSales ~ Unemployment + CPI_all + CPI_energy + Queries, data=train)
summary(saleReg)
```

```

Call:
lm(formula = ElantraSales ~ Unemployment + CPI_all + CPI_energy +
    Queries, data = train)

Residuals:
    Min      1Q  Median      3Q     Max
-6785.2 -2101.8  -562.5  2901.7  7021.0

Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept)   95385.36  170663.81   0.559    0.580
Unemployment  -3179.90    3610.26  -0.881    0.385
CPI_all        -297.65     704.84  -0.422    0.676
CPI_energy       38.51     109.60   0.351    0.728
Queries          19.03      11.26   1.690    0.101

Residual standard error: 3295 on 31 degrees of freedom
Multiple R-squared:  0.4282,    Adjusted R-squared:  0.3544
F-statistic: 5.803 on 4 and 31 DF,  p-value: 0.00132
```

Modeling Seasonality

Our model R-Squared is relatively low, so we would now like to improve our model. In modeling demand and sales, it is often useful to model seasonality. Seasonality refers to the fact that demand is often cyclical/periodic in time. For example, in countries with different seasons, demand for warm outerwear (like jackets and coats) is higher in fall/autumn and winter (due to the colder weather) than in spring and summer. (In contrast, demand for swimsuits and sunscreen is higher in the summer than in the other seasons.) Another example is the “back to school” period in North America: demand for stationary (pencils, notebooks and so on) in late July and all of August is higher than the rest of the year due to the start of the school year in September.

In our problem, since our data includes the month of the year in which the units were sold, it is feasible for us to incorporate monthly seasonality. From a modeling point of view, it may be reasonable that the month plays an effect in how many Elantra units are sold.

To incorporate the seasonal effect due to the month, build a new linear regression model that predicts monthly Elantra sales using Month as well as Unemployment, CPI_all, CPI_energy and Queries. Do not modify the training and testing data frames before building the model.

```
saleReg2 <- lm(ElantraSales ~ Month + Unemployment + CPI_all + CPI_energy + Queries, data=train)
summary(saleReg2)
```

```

Call:
lm(formula = ElantraSales ~ Month + Unemployment + CPI_all + 
    CPI_energy + Queries, data = train)

Residuals:
    Min      1Q  Median      3Q     Max 
-6416.6 -2068.7  -597.1  2616.3  7183.2 

Coefficients:
              Estimate Std. Error t value Pr(>|t|)  
(Intercept)  148330.49  195373.51   0.759   0.4536  
Month           110.69     191.66   0.578   0.5679  
Unemployment  -4137.28    4008.56  -1.032   0.3103  
CPI_all        -517.99     808.26  -0.641   0.5265  
CPI_energy       54.18     114.08   0.475   0.6382  
Queries          21.19      11.98   1.769   0.0871 .
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 3331 on 30 degrees of freedom
Multiple R-squared:  0.4344,    Adjusted R-squared:  0.3402 
F-statistic: 4.609 on 5 and 30 DF,  p-value: 0.003078
```

We observe that the model is not better because the adjusted R-squared has gone down and none of the variables (including the new one) are very significant.

Understanding the Model

Let us try to understand our model.

In the new model, given two monthly periods that are otherwise identical in Unemployment, CPI_all, CPI_energy and Queries, what is the absolute difference in predicted Elantra sales given that one period is in January and one is in March?

```
110.69*2
```

```
[1] 221.38
```

In the new model, given two monthly periods that are otherwise identical in Unemployment, CPI_all, CPI_energy and Queries, what is the absolute difference in predicted Elantra sales given that one period is in January and one is in May?

```
110.69*4
```

```
[1] 442.76
```

Numeric vs. Factors

You may be experiencing an uneasy feeling that there is something not quite right in how we have modeled the effect of the calendar month on the monthly sales of Elantras. If so, you are right. In particular, we added Month as a variable, but Month is an ordinary numeric variable. In fact, we must convert Month to a factor variable before adding it to the model.

A New Model

Re-run the regression with the Month variable modeled as a factor variable. (Create a new variable that models the Month as a factor (using the as.factor function) instead of overwriting the current Month variable. We'll still use the numeric version of Month later in the problem.)

```
train$MonthF <- as.factor(train$Month)
test$MonthF <- as.factor(test$Month)
attach(train)
saleReg3 <- lm(ElantraSales ~ MonthF + Unemployment + CPI_all + CPI_energy + Queries)
summary(saleReg3)
```

```

Call:
lm(formula = ElantraSales ~ MonthF + Unemployment + CPI_all + 
    CPI_energy + Queries)

Residuals:
    Min      1Q  Median      3Q     Max 
-3865.1 -1211.7   -77.1  1207.5  3562.2 

Coefficients:
               Estimate Std. Error t value Pr(>|t|)    
(Intercept)  312509.280 144061.867   2.169 0.042288 *  
MonthF2        2254.998   1943.249   1.160 0.259540    
MonthF3        6696.557   1991.635   3.362 0.003099 ** 
MonthF4        7556.607   2038.022   3.708 0.001392 ** 
MonthF5        7420.249   1950.139   3.805 0.001110 ** 
MonthF6        9215.833   1995.230   4.619 0.000166 ***
MonthF7        9929.464   2238.800   4.435 0.000254 ***
MonthF8        7939.447   2064.629   3.845 0.001010 ** 
MonthF9        5013.287   2010.745   2.493 0.021542 *  
MonthF10       2500.184   2084.057   1.200 0.244286    
MonthF11       3238.932   2397.231   1.351 0.191747    
MonthF12       5293.911   2228.310   2.376 0.027621 *  
Unemployment  -7739.381   2968.747  -2.607 0.016871 *  
CPI_all       -1343.307    592.919  -2.266 0.034732 *  
CPI_energy      288.631     97.974   2.946 0.007988 ** 
Queries          -4.764     12.938  -0.368 0.716598    
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 2306 on 20 degrees of freedom
Multiple R-squared:  0.8193,	Adjusted R-squared:  0.6837 
F-statistic: 6.044 on 15 and 20 DF,  p-value: 0.0001469
```

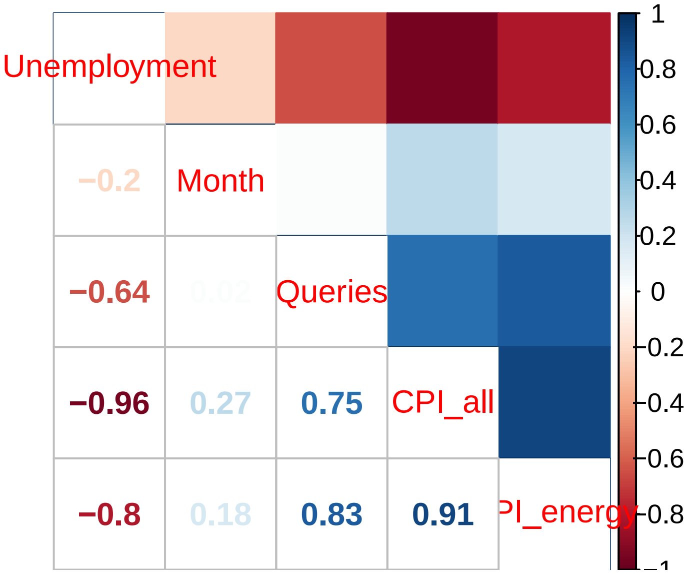
Multicolinearity

Another peculiar observation about the regression is that the sign of the Queries variable has changed. In particular, when we naively modeled Month as a numeric variable, Queries had a positive coefficient. Now, Queries has a negative coefficient. Furthermore, CPI_energy has a positive coefficient – as the overall price of energy increases, we expect Elantra sales to increase, which seems counter-intuitive (if the price of energy increases, we'd expect consumers to have less funds to purchase automobiles, leading to lower Elantra sales).

As we have seen before, changes in coefficient signs and signs that are counter to our intuition may be due to a multicolinearity problem. To check, compute the correlations of the variables in the training set.

Which of the following variables is CPI_energy highly correlated with?

```r
library(corrplot)
M <- cor(train[,c('Month','Unemployment','CPI_all','CPI_energy','Queries')])
corrplot.mixed(M, upper = 'color', lower = 'number',order="hclust", addrect=2)
```



A Reduced Model

Let us now simplify our model (the model using the factor version of the Month variable). We will do this by iteratively removing variables, one at a time. Remove the variable with the highest p-value (i.e., the least statistically significant variable) from the model. Repeat this until there are no variables that are insignificant or variables for which all of the factor levels are insignificant. Use a threshold of 0.10 to determine whether a variable is significant.

```r
saleReg4 <- lm(ElantraSales ~ MonthF + Unemployment + CPI_all + CPI_energy)
summary(saleReg4)
```

```
Call:
lm(formula = ElantraSales ~ MonthF + Unemployment + CPI_all +
    CPI_energy)

Residuals:
    Min      1Q  Median      3Q     Max
-3866.0 -1283.3  -107.2  1098.3  3650.1
```

```
Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept)  325709.15  136627.85   2.384 0.026644 *
MonthF2        2410.91    1857.10   1.298 0.208292
MonthF3        6880.09    1888.15   3.644 0.001517 **
MonthF4        7697.36    1960.21   3.927 0.000774 ***
MonthF5        7444.64    1908.48   3.901 0.000823 ***
MonthF6        9223.13    1953.64   4.721 0.000116 ***
MonthF7        9602.72    2012.66   4.771 0.000103 ***
MonthF8        7919.50    2020.99   3.919 0.000789 ***
MonthF9        5074.29    1962.23   2.586 0.017237 *
MonthF10       2724.24    1951.78   1.396 0.177366
MonthF11       3665.08    2055.66   1.783 0.089062 .
MonthF12       5643.19    1974.36   2.858 0.009413 **
Unemployment  -7971.34    2840.79  -2.806 0.010586 *
CPI_all       -1377.58     573.39  -2.403 0.025610 *
CPI_energy      268.03      78.75   3.403 0.002676 **
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 2258 on 21 degrees of freedom
Multiple R-squared:  0.818, Adjusted R-squared:  0.6967
F-statistic: 6.744 on 14 and 21 DF,  p-value: 5.73e-05
```

Test Set Predictions

Using the model from Problem 6.1, make predictions on the test set. What is the sum of squared errors of the model on the test set?

```
SalePredict <- predict(saleReg4, test)
SSE <- sum((SalePredict-test$ElantraSales)^2)
SSE
```

```
[1] 190757747
```

Comparing to a Baseline

What would the baseline method predict for all observations in the test set? Remember that the baseline method we use predicts the average outcome of all observations in the training set.

```
basePredict <- mean(train$ElantraSales)
basePredict
```

```
[1] 14462.25
```

Test Set R-Squared

```
SST <- sum((basePredict-test$ElantraSales)^2)
1-SSE/SST
```

```
[1] 0.7280232
```

Absolute Errors

What is the largest absolute error that we make in our test set predictions?

```
max(abs(SalePredict - test$ElantraSales))
```

```
[1] 7491.488
```

Month of Largest Error

```
test[which.max(abs(SalePredict - test$ElantraSales)),c('Month','Year')]
```

```
   Month Year
14     3 2013
```