15.780, Fall 2021

Stochastic Models in Business Analytics

Problem Set 2 - Regression & Time Series - Solutions

Due date: October 1, 2021

Instructions:

1. Submit a PDF file with your solutions to Canvas before the assigned deadline. Write your name and MIT ID on your submission.
2. All plots must have clear and easy-to-read axis labels and legends,
3. **Include relevant code in the PDF submission even if the question doesn't explicitly ask for it**. This means we can give you partial credit even if the output is wrong. if appropriate.

Problem 1. (Linear regression) (30 pts)
We want to build a forecasting model for the buyrates of Ultimate Fighting Championship (UFC) pay-per-views (ppv). Social media may be a good source of information which can predict the buyrate of the ppv's. We are going to see how good a source it is in this problem. We have a data file `UFC_data.csv` that contains rows for 10 different UFC events (they are sequentially numbered from 130 to 141). Each row has the number of `Tweets` (and some other social media stats) that contained a hashtag `#UFCXXX` in some fixed period before the event. The `buyrate` column is what we want to predict, and indicates the number of ppvs purchased for that event.

1. Let's first check if the social media features exhibit any collinearity. Calculate the pairwise correlation coefficient for each pair of features. Show the correlation matrix (do not include correlations with the dependent variable `Buyrate`). How many features are below a correlation threshold of 0.5? (10 pts)

   Hint: The `cor(df)` function in R will calculate the pairwise correlations of all column in the dataframe `df` in a matrix. You may get an error if the dataframe contains non-numeric columns – remove them before running.

```
corrs = cor(ufc[,-c(1, 8)]) # exclude ID (1st) and BuyRate (8th) column
round(corrs, 3)
```

	Tweets	Hashtags	URLs	Mentions	Unique_Users	Average_Sentiment
Tweets	1.000	0.993	0.992	0.999	0.999	-0.226
Hashtags	0.993	1.000	0.977	0.993	0.988	-0.164
URLs	0.992	0.977	1.000	0.994	0.991	-0.330
Mentions	0.999	0.993	0.994	1.000	0.996	-0.238
Unique_Users	0.999	0.988	0.991	0.996	1.000	-0.229
Average_Sentiment	-0.226	-0.164	-0.330	-0.238	-0.229	1.000

There are five correlations below the threshold, namely `Average_Sentiment` with every other social media stats. *Side note: this is interesting in and of itself: it seems that more tweets are correlated with worse sentiment about the event.* The key takeaway here is that there is reaaaaaaally high collinearity in all of the rest of the columns.

2. You probably saw that a lot of the features are very collinear. Let's ignore most of the features and only keep the `Tweets` feature. Build a linear regression model to predict `Buyrate` using `Tweets` as the independent variable. Use UFC130 to UFC136 as your training set (i.e. the first 7 rows) and the rest as a test set.

   (a) Report your model slope estimate and p-value, and the in-sample R^2. (5 pts)

   (b) Make a plot of Fitted Values vs. Residuals in your training set and add a horizontal line at residuals = 0. Do you think that a linear model is a good fit to this data? Include the plot in your submission. (5 pts)

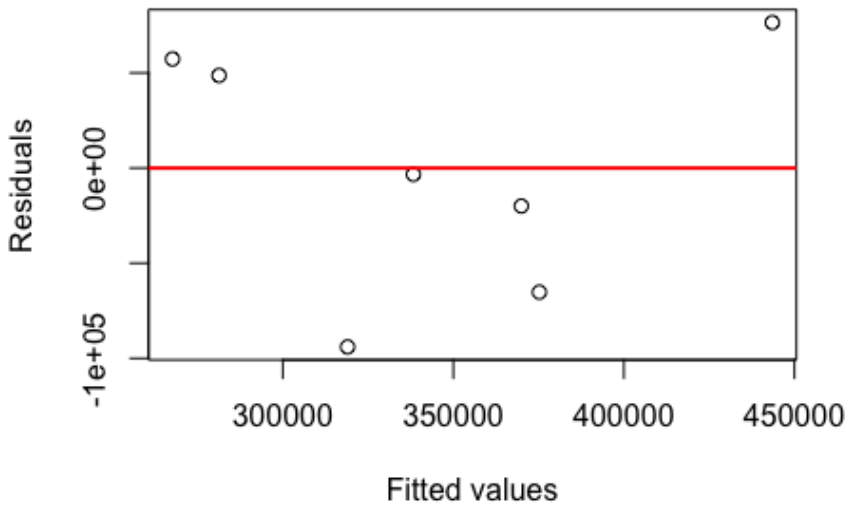
   Hint 1: the `lm` object in R contains vectors for both fitted values and residuals.

   Hint 2: The command `abline(h=3)` will add a horizontal line at y = 3 to an existing plot.

```
train = ufc[1:7,]
test = ufc[-(1:7),]
linreg = lm(Buyrate ~ Tweets, data = train)
summary(linreg)
plot(linreg$fitted.values, linreg$residuals,
     xlab = "Fitted values", ylab = "Residuals")
abline(h = 0, col = "red", lwd = 2)
```

The summary of the regression and residual plot is shown below. The slope is 0.8347 with a p-value of 0.09, while the in-sample R^2 is 0.4674.

```
Call:
lm(formula = Buyrate ~ Tweets, data = train)

Residuals:
     1      2      3      4      5      6      7
 57317  48684 -19941 -65223  -3301  76464 -93999

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) 2.461e+05  5.307e+04   4.638  0.00564 **
Tweets      8.347e-01  3.984e-01   2.095  0.09034 .
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 70710 on 5 degrees of freedom
Multiple R-squared:  0.4674,    Adjusted R-squared:  0.3609
F-statistic: 4.388 on 1 and 5 DF,  p-value: 0.09034
```



Though there are few data points, there does seem to be some curvature in the residual plot which indicates non-linearity (residuals are positive on the left and right, but negative in the middle).

3. Calculate the MSE, MAD, and MAPE of the model on both the training and testing data. (10 pts)

```
train_pred = predict(linreg, train)
test_pred = predict(linreg, test)
```

```r
mean((train_pred - train$Buyrate)^2)
mean(abs(train_pred - train$Buyrate))
mean(abs(train_pred - train$Buyrate) / abs(train$Buyrate))

mean((test_pred - test$Buyrate)^2)
mean(abs(test_pred - test$Buyrate))
mean(abs(test_pred - test$Buyrate) / abs(test$Buyrate))
```

	MSE	MAE	MAPE
Train	3571502518	521232.79	16.7%
Test	59670355968	154254.6	23.5%

Problem 2. (Time Series Modeling) (50 pts)

For this question, we use the skin moisturizer sales data, located in the file `MoisturizerSalesGoogleTrend.csv`. This file has **weekly** sales data for a popular moisturizer (normalized to hide true values), and also the value of Google trends for the keyword "eczema" (also normalized). Google trends provides a measure of the search volume of a given keyword on the Google search engine. Here we think that eczema searches can help us predict sales of moisturizer. We will compare a regression model using the trends data to a time series model with a seasonal component and see which performs best.

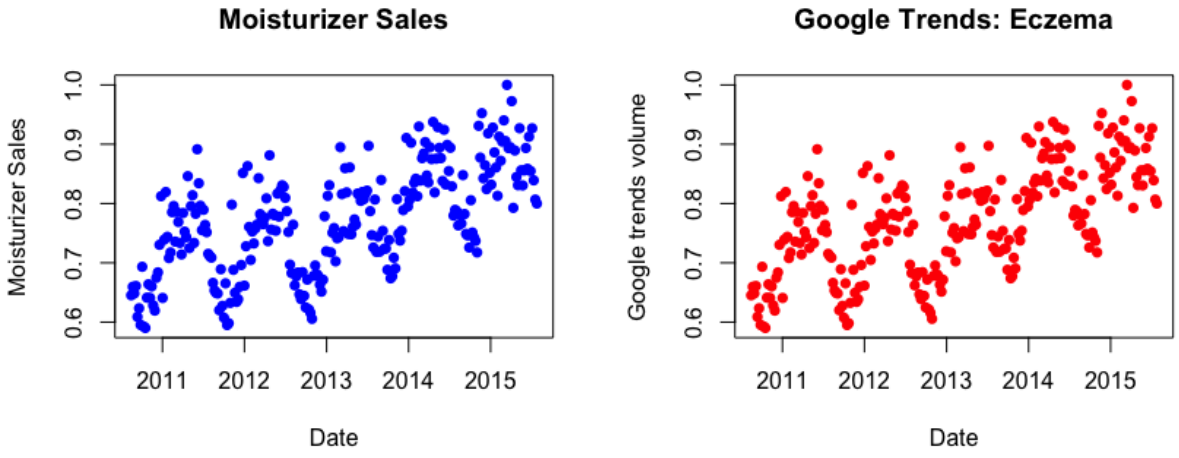
Hint: You may need to use the `anytime` R package to work with the Date column. Here is an example, if the dataframe is named `goog`.

```
library(anytime)
goog$Date = anytime(goog$Date)
```

When subsetting, you can also use this package:

```
subset(goog, goog$Date <= anytime("2013-12-29")).
```

1. Plot the sales data versus time, and the trend data versus time in separate plots. What do you observe about both plots? Include the plots in your Canvas submission. (10 pts)



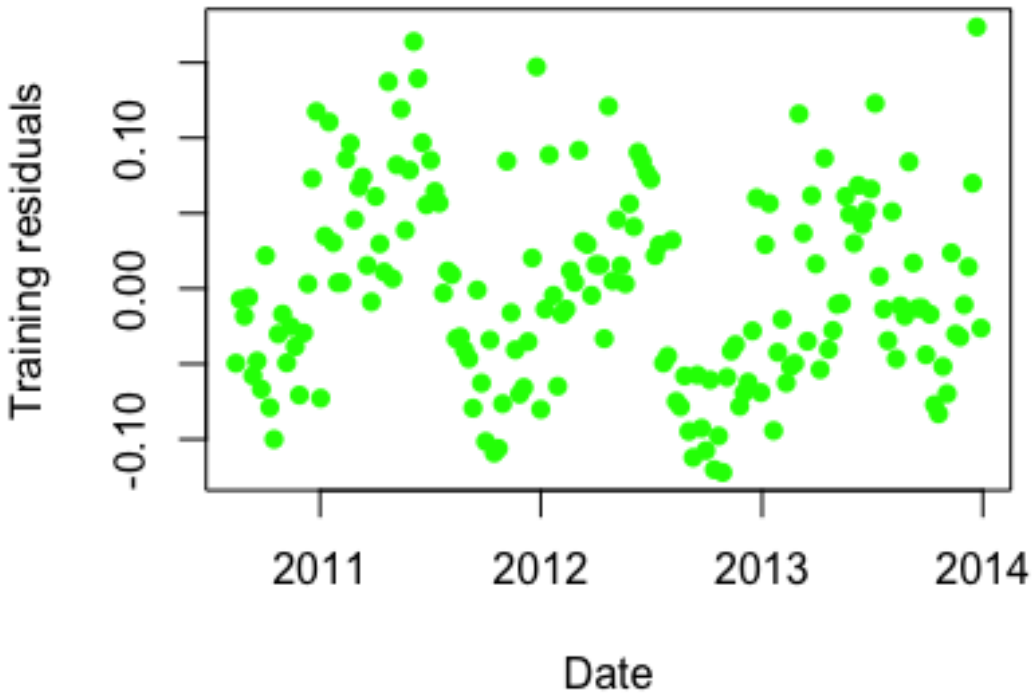
Both plots show similar seasonality and trend patterns, suggesting that trends may be able to predict the other.

2. Build a linear regression model for moisturizer sales with the Google trends data, using training data up to and including 2013-12-29, and the remaining being testing data. This should give you 177 training data points and 82 testing data points. Your linear regression equation should be `MoisturizerSales ~GoogleTrendVolumeEczema`. Report the model's slope and p-value, as well as the MAPE for both the training and the testing data. (10 pts)

```
train = subset(goog, goog$Date <= anytime("2013-12-29"))
test = subset(goog, goog$Date > anytime("2013-12-29"))

linreg = lm(MoisturizerSales ~ GoogleTrendVolumeEczema, data=train)
summary(linreg)

Call:
lm(formula = MoisturizerSales ~ GoogleTrendVolumeEczema, data = training)

Residuals:
     Min       1Q   Median       3Q      Max
-0.12205 -0.04929 -0.00881  0.04578  0.17364

Coefficients:
                         Estimate Std. Error t value Pr(>t)
(Intercept)                0.3859     0.0464    8.32  2.4e-14 ***
GoogleTrendVolumeEczema    0.4680     0.0622    7.52  2.7e-12 ***
---

Residual standard error: 0.0642 on 175 degrees of freedom
Multiple R-squared:  0.244,     Adjusted R-squared:  0.24
F-statistic: 56.6 on 1 and 175 DF,  p-value: 2.7e-12

train_pred = linreg$fitted.values
test_pred = predict(linreg, test)
mean(abs(train_pred - train$MoisturizerSales) / abs(train$MoisturizerSales))
mean(abs(test_pred - test$MoisturizerSales) / abs(test$MoisturizerSales))
```

The MAPE for the training set is about 7.2% and for the testing set 6.6%.

3. Plot your residuals for the above linear regression. Your x-axis should be the time index (or observation index). Do you observe a pattern? (5 pts)



There still seem to be strong seasonal component in the residuals which suggests the need for a time series model that can capture seasonality.

Now let's try to build an ARIMA model. **From this point on, use only the training data.**

4. Use the `auto.arima` function from the `forecast` package to build a time series model. (10pts)

   Hint: You will need to convert the column into an R time series `ts` type. What frequency should you specify for the seasonal component given that this is weekly data?

```
salests = ts(train$MoisturizerSales, frequency=52)
arima = auto.arima(salests)
summary(arima)
```

   (a) What are the parameters (p, d, q) of the fitted ARIMA model? (5pts) From the summary output, we see that $p = 0, d = 1, q = 1$ (and though the question doesn't ask for it, $P = 0, D = 1, Q = 1$).

(b) Find the **training** set MAPE of your ARIMA model? (5pts) The summary also gives us the training MAPE! It's a much better 3.2%.

As we saw, time series models are not *explanatory*, since they just use previous values of the time series to predict the future (rather than relating them to other variables). Let's look at a simple way that you might incorporate external data into a time series model.

5. Create a new dataframe called `resid` that has the same number of rows as the training set, and contains three columns: `Date` and `GoogleTrendVolumeEczema` (copied directly from the training set) and `Residuals` (the prediction error of each training point from the model you just built). Then build a regression model to predict `Residuals` based on `GoogleTrendVolumeEczema` and print the summary. (10pts)

   Hint: `as.vector(yourarimamodel$residuals)`

```
resid = data.frame(Date = train$Date,
                   GoogleTrendVolumeEczema = train$GoogleTrendVolumeEczema,
                   Residuals = arima$residuals)

lmresid = lm(Residuals ~ GoogleTrendVolumeEczema, data = resid)
summary(lmresid)
```

```
Call:
lm(formula = Residuals ~ GoogleTrendVolumeEczema, data = resid)

Residuals:
      Min        1Q    Median        3Q       Max
-0.161095 -0.012607  0.001574  0.009572  0.148464

Coefficients:
                        Estimate Std. Error t value Pr(>|t|)
(Intercept)             -0.04755    0.02765  -1.720   0.0872 .
GoogleTrendVolumeEczema  0.06882    0.03709   1.856   0.0652 .
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.03828 on 175 degrees of freedom
Multiple R-squared:  0.0193,    Adjusted R-squared:  0.01369
F-statistic: 3.444 on 1 and 175 DF,  p-value: 0.06518
```

6. Create a new fitted model for the training points by composing (adding together) the predictions of the ARIMA model and the linear regression. What is the MAPE of this prediction? (5pts)

```
hybrid_pred = as.vector(arima$fitted) + modresid$fitted.values
mean(abs(hybrid_pred - train$MoisturizerSales) / abs(train$MoisturizerSales))
```

The training MAPE of the hybrid model is 3.2%. So in this case the regression didn't make a big difference.