Assignment 5-Decision tree, randomForest and Time Series

Andrew Abisha Hu, Elaine Su, Eric Xiong, Tianye Wang

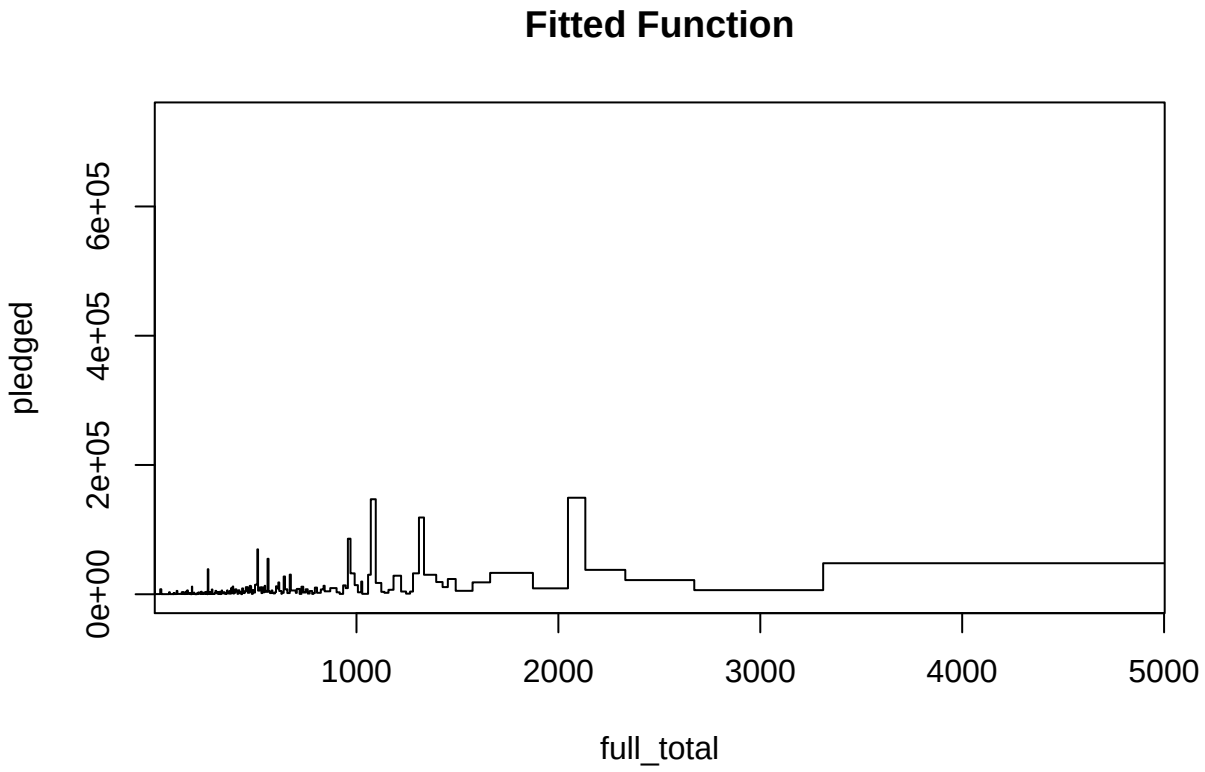
9/6/2018

Here is the description of variables:

```
category: the category to which the project belongs
goal: the project's raising goal ($)
pledged: $ pledged at the end of the fund-raising period
funded: "TRUE" if the goal is achieved (i.e., pledged >= goal); "FALSE" otherwise
blurb_adj: ratio of counts of adjectives to total word counts of the blurb
blurb_noun: ratio of counts of nouns to total word counts of the blurb
blurb_verb: ratio of counts of verbs to total word counts of the blurb
blurb_total: total word counts of the blurb on the project page
full_adj: ratio of counts of adjectives to total word counts of the main content
full_noun: ratio of counts of nouns to total word counts of the main content
full_verb: ratio of counts of verbs to total word counts of the main content
full_total: total word counts in the main content the project page
train: = 1 if the project is in the training set and 0 otherwise
```

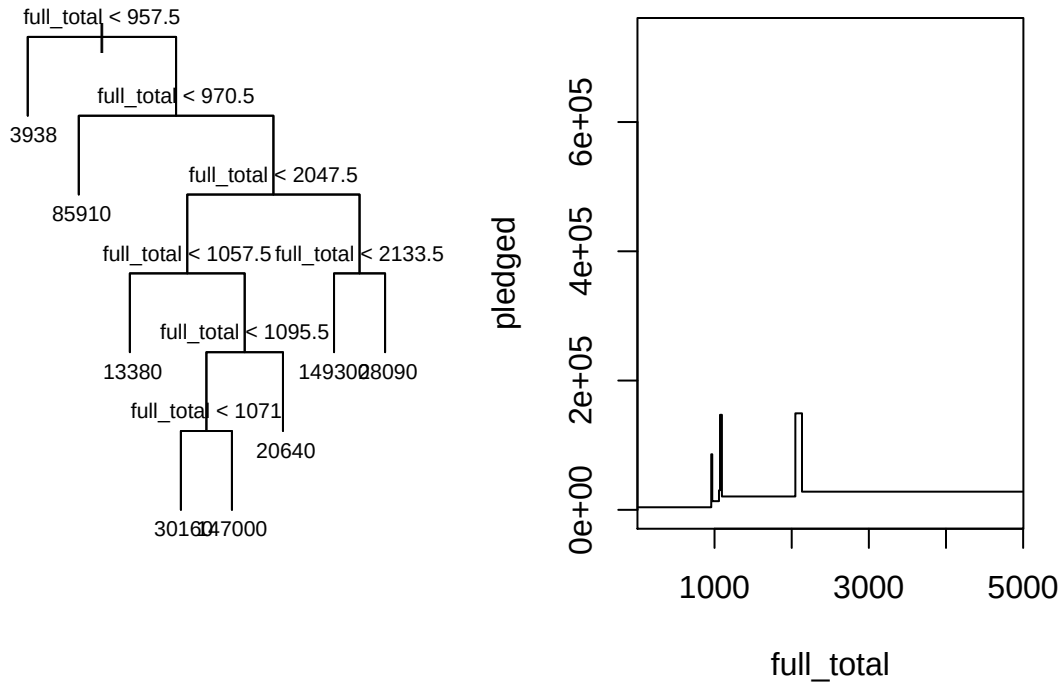
1. Use the training data set to create a regression tree and predict pledged by only full_total. Use mindev=1e-6 to overgrow the tree. Plot the partition of the tree model. (provide R codes, 0.5%)

```
setwd("/Users/andrewhu/Documents/GitHub/Machine-Learning/Assignment 5- Decision tree, randomForest and T

library(tree)
ks <- read.csv("KickStarter2.csv")
kick <- read.csv("KickStarter2.csv")


attach(ks)
fit= tree(pledged~full_total,data=kick[kick$train==1,],mindev=1e-6)

partition.tree(fit, main="Fitted Function")
```



2. Prune the tree in Question (1) by requesting the number of terminal nodes to be 8. Plot the decision tree and the partition of the pruned tree model.

```
fit2 = prune.tree(fit, best=8)
par(mfrow=c(1,2))
plot(fit2, type= "uniform")
text(fit2, cex=.7)
partition.tree(fit2, cex=.8)
```



3. According to the pruned tree, for a project with content in 1800 words, what is the predicted pledged amount? How many projects in the data also have the same predicted value? (0.5%)

```
predict(fit2, data.frame(full_total=1800))
```

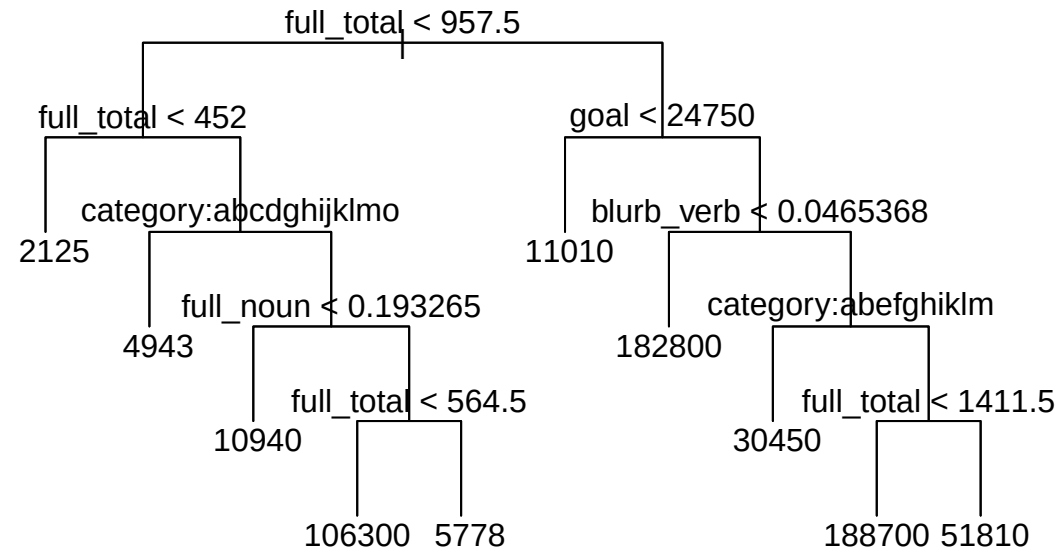
```
##        1
## 20644.63
```

```
sum(ifelse(full_total==1800,1,0))
```

```
## [1] 0
```

4. Use the training data set to create a regression tree to predict pledged by all variables except funded. Use mindev=1e-6 to overgrow the tree and then prune it by requesting the number of terminal nodes to be 10. Plot the decision tree and label the decision tree plot. What are the key predictors used to split this tree?

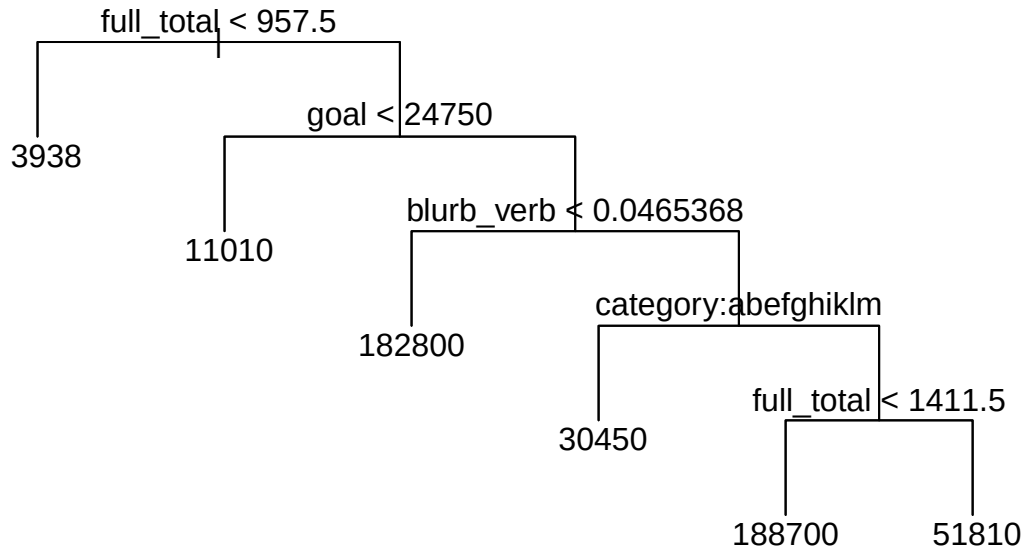
```
fit3= tree(pledged~category+goal+blurb_adj+blurb_noun+blurb_verb+blurb_total+full_adj+full_noun+full_ve


fit4 = prune.tree(fit3,best=10)

plot(fit4, type="uniform")
text(fit4)
```



full_total, goal, category, blurb_verb, full_noun

5. Further prune the tree you pruned in question (4) by requesting six terminal nodes. Which predictor is no longer a key predictor after you re-prune the tree?

```
fit5 = prune.tree(fit4,best=6)
plot(fit5, type= "uniform")
text(fit5)
```




```
full_noun
```

6. Use the testing set to compute MSE = mean((pledged – pledged_hat)2) for the two pruned tree models in Questions (4) and (5). Which one performs better in terms of test MSE? (provide R codes, 0.5%)

```
yhat1<- predict(fit4,ks[train==0,])
yhat2<- predict(fit5,ks[train==0,])

mean((ks$pledged[!train]-yhat1)^2, na.rm=T)
```

```
## [1] 4861942746
```

```
mean((ks$pledged[!train]-yhat2)^2, na.rm=T) #better
```

```
## [1] 4789667602
```

7. According to the better tree in Question (6), what is the predicted pledged amount for an art project with a goal of $10000 if the campaign runner writes 10 words in the blurb with 4 verbs and 1000 words in the main content with 124 nouns? (0.5%)

```
predict(fit5, data.frame(category="Art",goal=10000,blurb_total=10,blurb_verb=0.4,full_total=1000,full_n
```

```
##        1
## 11014.58
```

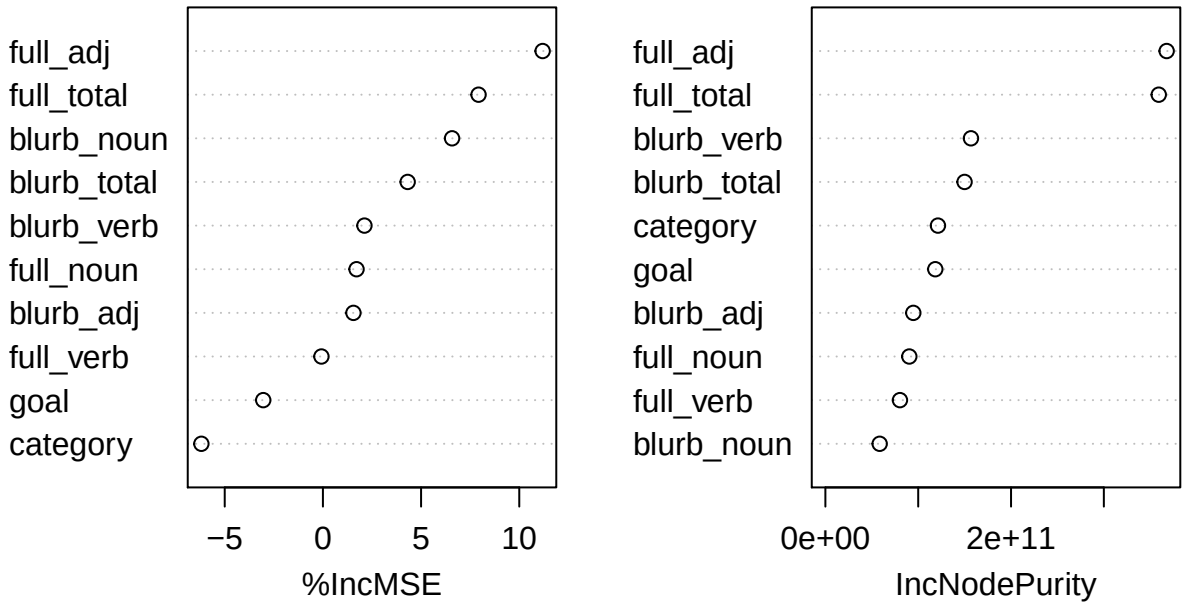
8. Use the training data set to run "bagging" by using the randomForest() function and predict pledged by all variables except funded. Create a chart to show the importance of predictors in terms of %IncMSE. What are the three most important predictors? Use set.seed(12345) to specify the random seed. (provide R codes, 1%)

```
library(randomForest)
```

```
## randomForest 4.6-14
```

```
## Type rfNews() to see new features/changes/bug fixes.
```

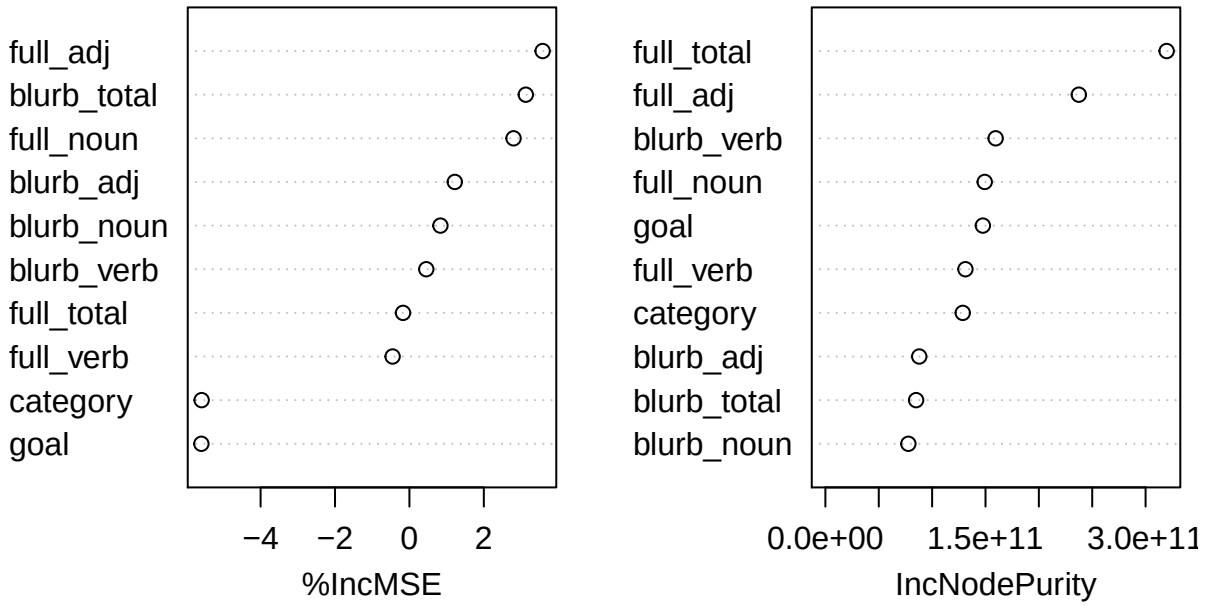
```
set.seed(12345)

fit8= randomForest(pledged~category+goal+blurb_adj+blurb_noun+blurb_verb+blurb_total+full_adj+full_noun

varImpPlot(fit8)
```



9. Now use the training data set to run "random forest" by using the randomForest() function and predict pledged by all variables except funded. Create a chart to show the importance of predictors in terms of %IncMSE. What are the three most important predictors? Use set.seed(12345) to specify the random seed.

```
set.seed(12345)

fit9= randomForest(pledged~category+goal+blurb_adj+blurb_noun+blurb_verb+blurb_total+full_adj+full_noun

varImpPlot(fit9)
```

fit9



10. Compute test MSE for the bagging and random forest models in Questions (8) and (9). Which one performs better in terms of test MSE?

```
yhat3<- predict(fit8,ks[train==0,])
yhat4<- predict(fit9,ks[train==0,])

mean((kick$pledged[train==0]-yhat3)^2, na.rm=T)
```

```
## [1] 4681213364
```

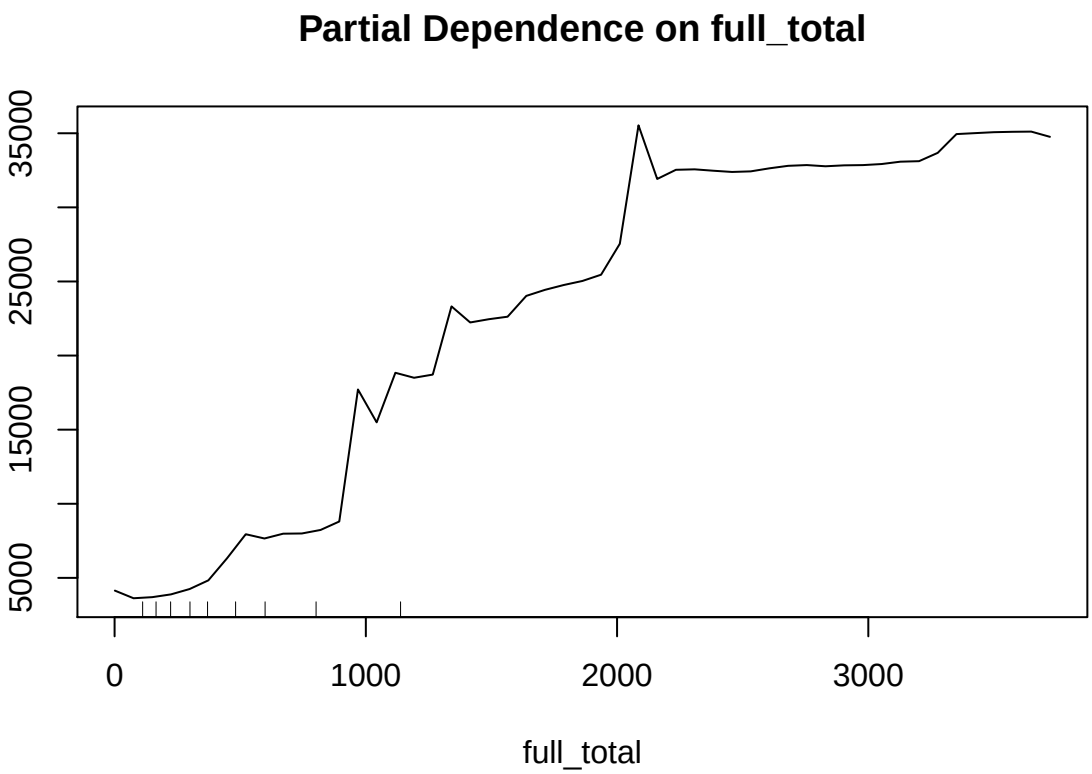
```
mean((kick$pledged[train==0]-yhat4)^2, na.rm=T)
```

```
## [1] 4583550580
```

```
#randomForest is better
```

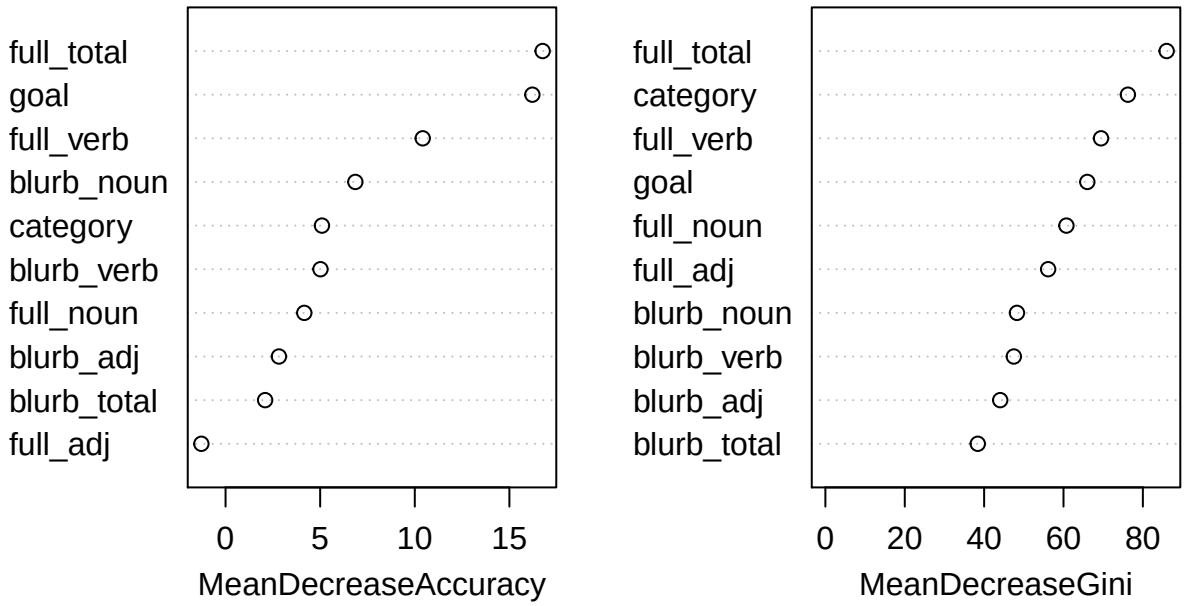
11. Create a partial dependence plot to depict the marginal effect of full_total using the random forest model with the test data.

```
fit= randomForest(pledged~category+goal+blurb_adj+blurb_noun+blurb_verb+blurb_total+full_adj+full_noun+

partialPlot(fit, kick[kick$train==0,], full_total)
```



12. Use the training data and run a "random forest" to predict funded by all variables except pledged. Create a chart to show the importance of predictors in terms of MeanDecreaseAccuracy. What are the three most important predictors? Use set.seed(12345) to specify the random seed.

```
fit= randomForest(as.factor(funded)~category+goal+blurb_adj+blurb_noun+blurb_verb+blurb_total+full_adj+

varImpPlot(fit)
```



13. Based on your findings, would you conclude that full_total has a causal effect on funded? Why?

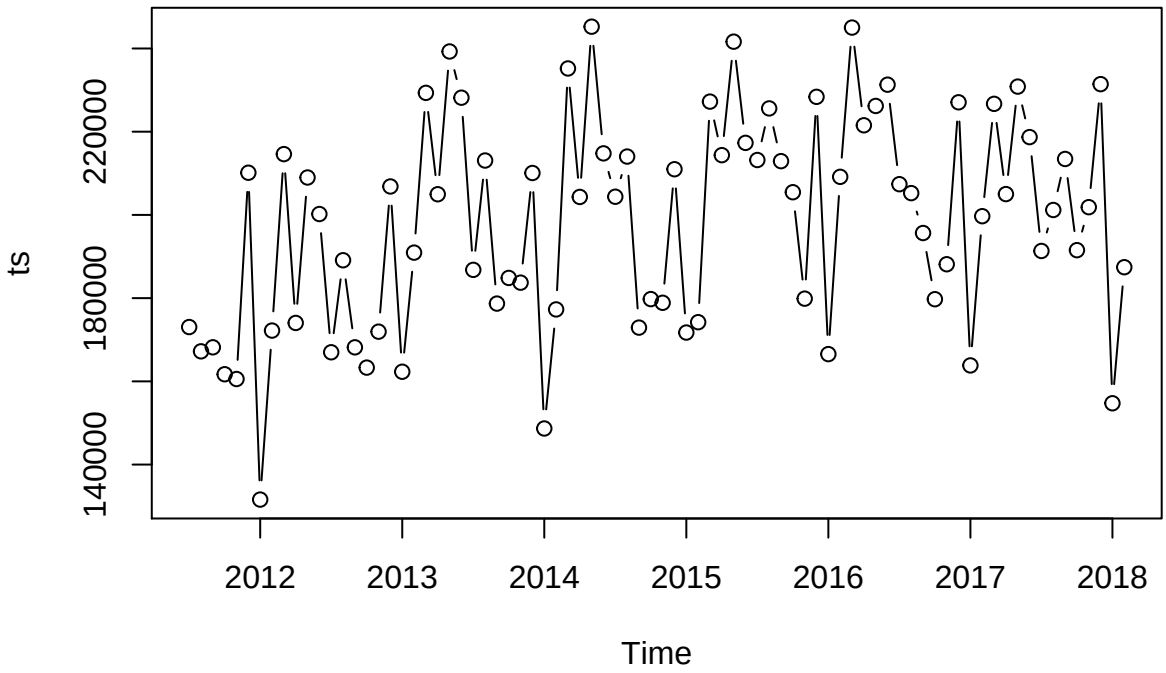
```
Yes, because it is the most important predictor of funded according to MeanDecreaseAccuracy
in the random forest model.
```

FordUS.csv records the number of vehicles sold under the brand "Ford" in the US from July 2011 to February 2018. (data from http://shareholder.ford.com/financials/monthly-sales-reports).

```
Month: the calendar month
US_sales: the number of vehicles sold under the brand "Ford" in the US in the
month
```

14. Read the data into R and use ts() to make US_sales a time-series object in R. Create a plot of US_sales over time.

```r
ford<- read.csv("FordUS.csv")
attach(ford)
ts= ts(US_sales, start=c(2011,7), frequency = 12)
plot(ts, type="b")
```



15. Based on the trajectory of US_sales over time, comment on the trend and seasonality of the sales. Which model may better describe the data, an additive model or a multiplicative model? Why?

Since the amplitude of seasonality seems constant, an additive model may better describe the data.

16. Use the data from July 2011 to June 2017 as your training data and the rest as your test data. Fit the training data with an additive Halt-Winters model. What are the values of the smoothing parameters: alpha, beta, and gamma? (provide R codes, 0.5%) Hint: use window() to obtain a subset of a time-series object

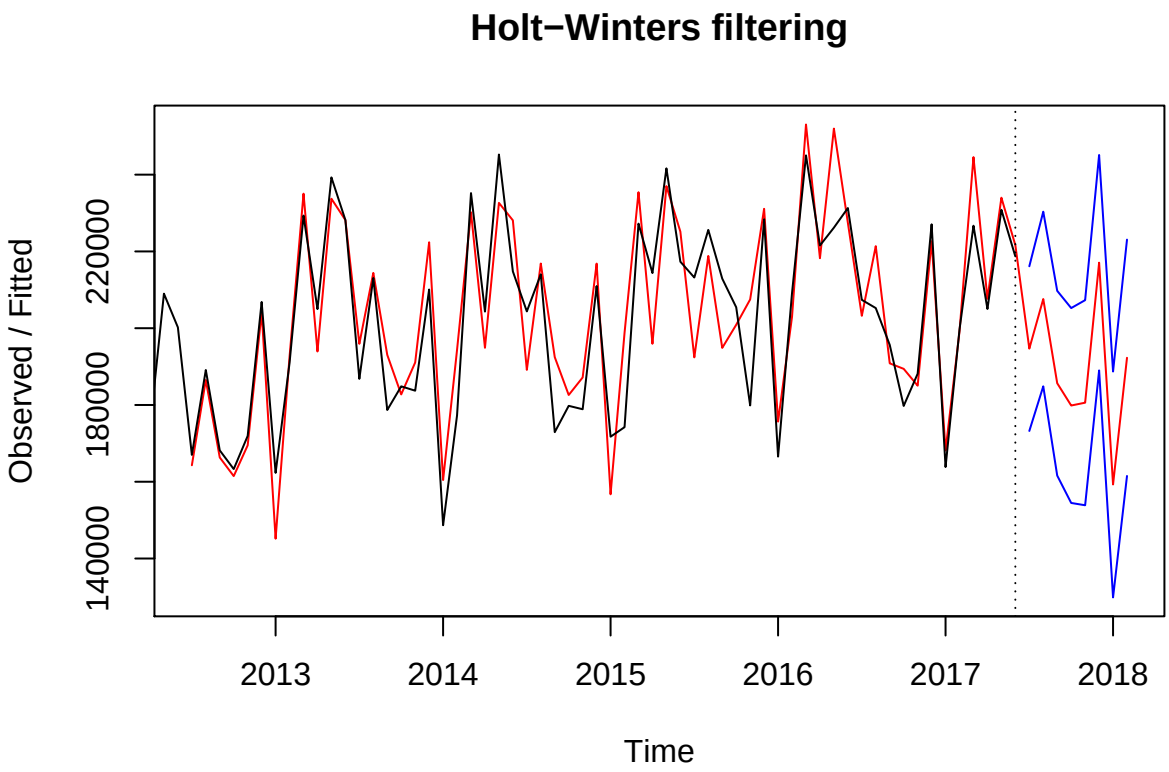
```
train= window(ts, start=c(2011,7), end=c(2017,6))
fit9 =HoltWinters(train)
fit9
```

```
## Holt-Winters exponential smoothing with trend and additive seasonal component.
##
## Call:
## HoltWinters(x = train)
##
## Smoothing parameters:
##  alpha: 0.3421359
##  beta : 0.03396044
##  gamma: 0.2469973
##
## Coefficients:
##            [,1]
## a   201440.8584
## b      118.7739
## s1   -6880.2864
## s2    5920.3180
## s3  -16131.2601
## s4  -22057.7516
## s5  -21443.4037
```

```
## s6   14910.1273
## s7  -42984.7477
## s8  -10100.8965
## s9   31686.8356
## s10   3132.0107
## s11  30110.3516
## s12  18714.4465
```

17. Compute the predicted sales from July 2017 to February 2018 with the 95% prediction interval. Create a plot comprising of the training data, the fitted values, the mean predicted value, and the prediction interval. (provide R codes, 0.5%)

```
yhat= predict(fit9, n.ahead=8, prediction.interval=T)
plot(fit9, yhat,type= "l")
```



18. Use your test data to compute the prediction error for each month from July 2017 to February 2018. Report the mean square deviation (MSD) and mean absolute deviation (MAD). (provide R codes, 0.5%)

```
test= window(ts, start=c(2017,7), end=c(2018,2))
mean((test-yhat[,1])^2)
```

```
## [1] 207916147
```

```
mean(abs(test-yhat[,1]))
```

```
## [1] 11781.33
```

19. Fit the training data with a multiplicative Halt-Winters model. What are the values of the smoothing parameters: alpha, beta, and gamma?

```
fit10 = HoltWinters(train, seasonal= "multiplicative")
fit10
```

```
## Holt-Winters exponential smoothing with trend and multiplicative seasonal component.
```

```
##
## Call:
## HoltWinters(x = train, seasonal = "multiplicative")
##
## Smoothing parameters:
##  alpha: 0.3039754
##  beta : 0.04317377
##  gamma: 0.3999511
##
## Coefficients:
##              [,1]
## a    2.033418e+05
## b   -2.648996e+01
## s1   9.787966e-01
## s2   1.025615e+00
## s3   9.229359e-01
## s4   8.866755e-01
## s5   8.814004e-01
## s6   1.064688e+00
## s7   7.789854e-01
## s8   9.316931e-01
## s9   1.128857e+00
## s10  1.011839e+00
## s11  1.137035e+00
## s12  1.085936e+00
```

20. Compute the prediction error for each month from July 2017 to February 2018 with the multiplicative model. Report the MSD and MAD.

```
yhat2 = predict(fit10, n.ahead=8,prediction.interval=T)
mean((test-yhat2[,1])^2)
```

```
## [1] 208757927
```

```
mean(abs(test-yhat2[,1]))
```

```
## [1] 11918.44
```

21. Based on the MSD and MAD you computed, which model predicts the test data better? Why? (0.5%)

Since the MSD and MAD of the additive model is lower than the multiplicative model, we can conclude that the additive model better predicts the test data. This may be due to the constant amplitude of seasonality of this data.

22. Fit the training data with an additive Halt-Winters model with beta = FALSE (i.e., the trend is a constant value). Compute the prediction error for each month from July 2017 to February 2018 with the new additive model. Report the MSD and MAD.

```
fit11 = HoltWinters(train, beta=F)
yhat3 = predict(fit11, n.ahead=8, prediction.interval=T)
test-yhat3[,1]
```

```
##            Jan       Feb Mar Apr May Jun       Jul       Aug       Sep
## 2017                                    -2947.246 -5741.574 28392.771
## 2018 -3546.347 -3579.194
##            Oct       Nov       Dec
## 2017 12375.813 22188.579 15273.869
## 2018
```

```
mean((test-yhat3[,1])^2)
```

```
## [1] 218996680
```

```
mean(abs(test-yhat3[,1]))
```

```
## [1] 11755.67
```

23. Based on the MSD and MAD, is the additive model that assumes beta = 0 better than the additive model that estimates beta in predicting the test data set? Why?

As we can see, the MSD for the additive model that estimates beta is significantly lower than the model assuming beta=0, while the MAD is just a bit higher. The additive model that estimates beta in predicting the test data set is better.

24. Use the overall data and decompose US_sales into the trend, seasonality, and random. Select the model type (i.e., additive or multiplicative) that you think better fit the data. Plot the decomposition results over time. Comment on which months you expect to see the highest and lowest sales respectively.

```
fit12 = decompose(ts, type="additive")
plot(fit12, type="b")
```