Ch7Report

Madi Kassymbekov

4/19/2021

Question 1

In this question, you will analyze one of the few public datasets available for uplift modeling in the package Information. The data is from a historical marketing campaign. It contains 68 predictive variables including a treatment indicator and a purchase indicator. The definition of all these variables is hard to find but the most relevant ones for your analysis are: - TREATMENT: equals 1 if the person received the marketing offer, and 0 if the person was in the control group - PURCHASE: equals 1 if the person accepted the offer, and 0 otherwise - UNIQUE_ID: unique identifier - AGE: age of the person - D_REGION_X: 1 if the person lives in region X, 0 otherwise (3 regions: A, B, C) Other variables are from credit bureau data (e.g., N_OPEN_REV_ACTS = number of open revolving accounts). The train and test datasets are in the objects uplift.train and uplift.test in the file uplift.RData.

1) Use the two-model approach using a logistic regression and all variables in the model to obtain the lift for all subjects in the test set. Print the lift for the top 10 subjects.

```r
set.seed(5984736)
# two models approach with glm
uplift.trainc=uplift.train[uplift.train$TREATMENT==0,]
glmfitc=glm(PURCHASE~. -TREATMENT,data=uplift.trainc, family = "binomial")
predc=predict(glmfitc,newdata=uplift.test, type="response")

uplift.traint=uplift.train[uplift.train$TREATMENT==1,]
glmfitt=glm(PURCHASE~. -TREATMENT,data=uplift.traint, family ="binomial")
```

```
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
```

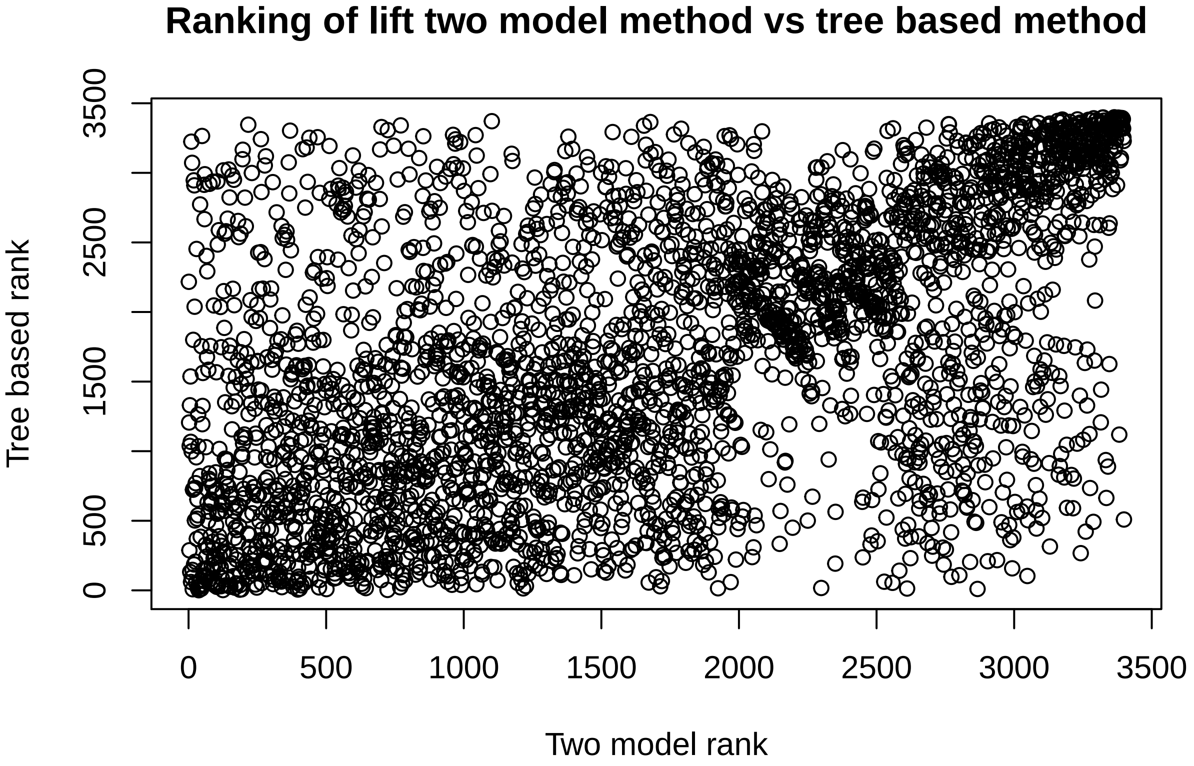
```r
predt=predict(glmfitt,newdata=uplift.test, type="response")
lift2models=predt-predc

tail(sort(lift2models) , 10)
```

```
##       915       782      2312       349      1095      2173      1630       945
## 0.3868017 0.3921861 0.4045018 0.4124169 0.4269889 0.4310817 0.4415972 0.4572648
##       203       162
## 0.7437702 0.8044193
```

2) Repeat the same question by using a forest with 100 trees. Compare the ranks of the lift for both models.

```r
uprf=upliftRF(PURCHASE~.+trt(TREATMENT),data=uplift.train,split_method ="ED",ntree=100)
preduprf=predict(uprf,newdata=uplift.test)
liftuprf=preduprf[,1]-preduprf[,2]
#Counter of rows for id and rank
rank<-seq(1 , length(liftuprf) , 1)
uplift_rank<-data.frame(cbind(rank,liftuprf))
```

```
twomodels_rank<-data.frame(cbind(rank,lift2models))
#Sorting by lift in decreasing order
twomodels_rank <- twomodels_rank[order(twomodels_rank$lift2models , decreasing = TRUE),]
uplift_rank <- uplift_rank[order(uplift_rank$liftuprf , decreasing = TRUE),]
# assigning rank for sorted data frames
twomodels_rank$twomodelrank <- rank
uplift_rank$upliftrank <- rank
#Aggregating data
rankings <- merge(twomodels_rank , uplift_rank , by="rank" )
# Plot to compare the ranking
plot(rankings$twomodelrank , rankings$upliftrank,
main= "Ranking of lift two model method vs tree based method",
xlab= "Two model rank",
ylab= "Tree based rank")
```



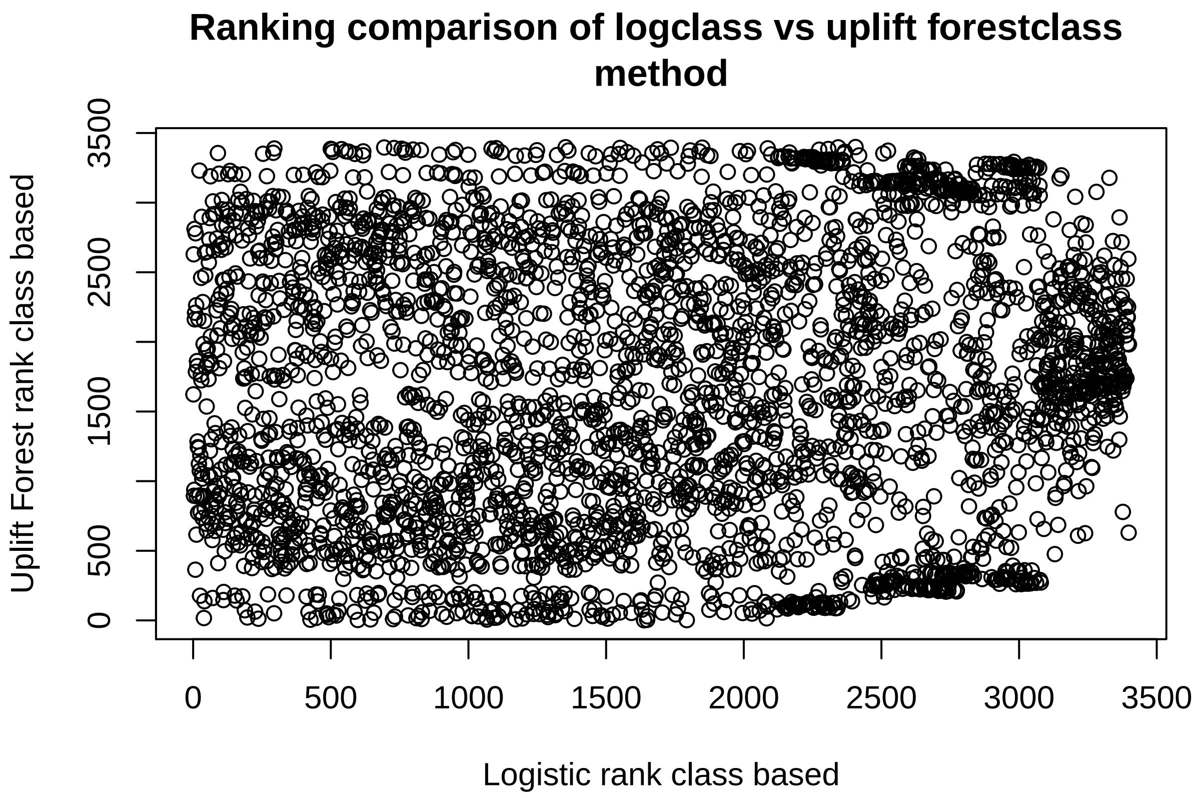
Answer: There is a noticeable agreement among two methods to rank top lift and even more on bottom lift subjects. However, there is a disagreement in the middle of the rankings as there is a bigger variance in rankings.

3) Repeat the same question by using a class transformation using

i) a logistic model and ii) a forest with 100 trees. You can assume a propensity score of 0.5.

```
#Class transformation
uplift.trainclass <- uplift.train
uplift.trainclass$z <- factor(((uplift.trainclass$PURCHASE==uplift.trainclass$TREATMENT)))
#Logistic regression
logclass <- glm(z~ . -TREATMENT - PURCHASE, data=uplift.trainclass , family = "binomial")
```

```
logclass_pred <- 2*predict(logclass , newdata = uplift.test , type = "response")-1
#Tree based method
upliftrf_tree <- rfsrc(z~. -TREATMENT -PURCHASE, data = uplift.trainclass , ntree = 100)
upliftrf_tree_pred <- 2*predict(upliftrf_tree, newdata = uplift.test)$predicted[,2]-1

#Ranking
logranklift <- data.frame(cbind(rank,logclass_pred ))
upliftranklift <- data.frame(cbind(rank, upliftrf_tree_pred))
logranklift <- logranklift[order(logranklift$logclass_pred , decreasing = TRUE),]
upliftranklift <- upliftranklift[order(upliftranklift$upliftrf_tree_pred ,
                                       decreasing=TRUE),]
logranklift$logrank <- rank
upliftranklift$upliftrank <- rank

classranking <- merge(logranklift, upliftranklift, by="rank" )
# Plot to compare the ranking
plot(classranking$logrank , classranking$upliftrank,
main= "Ranking comparison of logclass vs uplift forestclass
method",
xlab= "Logistic rank class based",
ylab= "Uplift Forest rank class based")
```



Answer: There is no agreement most of the time across the rankings. There are some little groups in dark showing agreement in range of 2200-3400 ranking range. Otherwise, everything seems random with no linear pattern.

4) Compare all models using the Qini coefficient.

```
twomodel_perf <- performance(predt , predc , uplift.test$PURCHASE ,
                             uplift.test$TREATMENT)
twomodel_qini <- qini(twomodel_perf , plotit = FALSE)

uplift_perf <- performance(preduprf[,1],preduprf[,2],
                           uplift.test$PURCHASE, uplift.test$TREATMENT)
uplift_qini <- qini(uplift_perf , plotit=FALSE)

classlog_perf <- performance(rep(1 , length(logclass_pred)),
              rep(1 , length(logclass_pred)) - logclass_pred,
              uplift.test$PURCHASE, uplift.test$TREATMENT)
classlog_qini <- qini(classlog_perf, plotit = FALSE)

classuplift_perf <- performance(rep(1 , length(upliftrf_tree_pred)),
           rep(1 , length(upliftrf_tree_pred)) - upliftrf_tree_pred,
           uplift.test$PURCHASE , uplift.test$TREATMENT)
classuplift_qini <- qini(classuplift_perf, plotit = FALSE)

Qini_results <- data.frame("Two Model", twomodel_qini$Qini)
names(Qini_results) <- c("Model", "Qini")
Qini_results[nrow(Qini_results) + 1,] = list(Model="Tree Based Method",
                                   Qini=uplift_qini$Qini)
Qini_results[nrow(Qini_results) + 1,] = list(
                       Model="Class transformation logistic regression",
                       Qini=classlog_qini$Qini)
Qini_results[nrow(Qini_results) + 1,] = list(
                       Model="Class transformation based tree model",
                       Qini=classuplift_qini$Qini)
knitr::kable(Qini_results, caption="Qini values of 4 different models")
```

Table 1: Qini values of 4 different models

Model	Qini
Two Model	0.0111299
Tree Based Method	0.0126451
Class transformation logistic regression	0.0088995
Class transformation based tree model	0.0360753

5) Comments on the results and give your thought about this analysis. Answer: Qini coefficient indicates effect of treatment comparison of various models against the random treatment curve. The higher the value, the better model is assuming all models are valid. The best model based on the Qini coefficient is class transformation tree based method with Qini of 0.0360753. Overall, all the Qini coefficients are pretty small and are close to each other. In my opinion, it's not the best measurement to comprehend the best performing model. Classical errors like MSE, MAE are better for understanding the models performance.

Question 2 Multiple-Choice Questions

1. Which of the following is true about prediction intervals?

A. The prediction interval is another name for confidence interval

Answer; False. Prediction interval provides an interval for predicted unobserved response, while confidence interval provides a range interval for observed mean value of the variable.

B. The width of prediction and confidence intervals are on average the same.

Answer; False. Due to uncertainty of predicting unobserved instances,width of prediction intervals will be generally wider than for confidence intervals which are based on observed instances.

C. Due to the presence of MSE in prediction interval formula, PI will be wider than confidence interval

Answer; True. In PI, additional MSE term is added compared to the confidence interval formula which makes the overall interval wider due to bigger bounds thanks to the additional MSE term.

D. Marginal interpretation's validity means that the predictive performance is good across all covariates' space.

Answer; False. Marginal interpretation states that on average (1-alpha)% time true Y value is contained in the prediction interval so that model is good enough but not perfect across whole space.

2.Which of the following is true about prediction intervals?

A. Coverage shows on how much time predicted values matched the true values of the random variable.

Answer; False. Coverage shows the percentage fore prediction interval including true value of the random variable within it's range.

B. Generally, point predictions as well as key performance metric measure such as MSE, MAPE, etc. are enough to conclude whether the model is good for predicting the target variable.

Answer; False. Even though performance metrics can outline performance compared to other models' metrics, it doesn't mean a lot without knowing the coverage rate of true values of target variable within the prediction interval to make conclusion on the model's performance.

C. Prediction interval's reliability is not dependent on the model's variable selection.

Answer; False. Variable selection plays an important role as bad variable selection will the affect the coverage rate which will make prediction intervals non-reliable and meaningless for generalized use.

D. The main advantage of conformal prediction interval is that it can be used with almost any prediction model type

Answer; True. It can be used with almost any predictive model with one requirements that all covariates should be numeric.