Classification Kernel and Ensemble

Vladimir Sokolov

10/24/2022

This notebook explores Hotel Booking data from Kaggle.

Load the data set.

```
df <- read.csv("hotel_bookings.csv")
str(df)
```

```
## 'data.frame':    119390 obs. of  32 variables:
##  $ hotel                         : chr  "Resort Hotel" "Resort Hotel" "Resort Hotel" "Resort Hotel"
##  $ is_canceled                   : int  0 0 0 0 0 0 0 0 1 1 ...
##  $ lead_time                     : int  342 737 7 13 14 14 0 9 85 75 ...
##  $ arrival_date_year             : int  2015 2015 2015 2015 2015 2015 2015 2015 2015 2015 ...
##  $ arrival_date_month            : chr  "July" "July" "July" "July" ...
##  $ arrival_date_week_number      : int  27 27 27 27 27 27 27 27 27 27 ...
##  $ arrival_date_day_of_month     : int  1 1 1 1 1 1 1 1 1 1 ...
##  $ stays_in_weekend_nights       : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ stays_in_week_nights          : int  0 0 1 1 2 2 2 2 3 3 ...
##  $ adults                        : int  2 2 1 1 2 2 2 2 2 2 ...
##  $ children                      : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ babies                        : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ meal                          : chr  "BB" "BB" "BB" "BB" ...
##  $ country                       : chr  "PRT" "PRT" "GBR" "GBR" ...
##  $ market_segment                : chr  "Direct" "Direct" "Direct" "Corporate" ...
##  $ distribution_channel          : chr  "Direct" "Direct" "Direct" "Corporate" ...
##  $ is_repeated_guest             : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ previous_cancellations        : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ previous_bookings_not_canceled: int  0 0 0 0 0 0 0 0 0 0 ...
##  $ reserved_room_type            : chr  "C" "C" "A" "A" ...
##  $ assigned_room_type            : chr  "C" "C" "C" "A" ...
##  $ booking_changes               : int  3 4 0 0 0 0 0 0 0 0 ...
##  $ deposit_type                  : chr  "No Deposit" "No Deposit" "No Deposit" "No Deposit" ...
##  $ agent                         : chr  "NULL" "NULL" "NULL" "304" ...
##  $ company                       : chr  "NULL" "NULL" "NULL" "NULL" ...
##  $ days_in_waiting_list          : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ customer_type                 : chr  "Transient" "Transient" "Transient" "Transient" ...
##  $ adr                           : num  0 0 75 75 98 ...
##  $ required_car_parking_spaces   : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ total_of_special_requests     : int  0 0 0 0 1 1 0 1 1 0 ...
##  $ reservation_status            : chr  "Check-Out" "Check-Out" "Check-Out" "Check-Out" ...
##  $ reservation_status_date       : chr  "2015-07-01" "2015-07-01" "2015-07-02" "2015-07-02" ...
```

Factor and simplify data.

```
df$hotel <- factor(df$hotel)
df$is_canceled <- factor(df$is_canceled)
df$country <- factor(df$country)
df$market_segment <- factor(df$market_segment)
df$deposit_type <- factor(df$deposit_type)
df$customer_type <- factor(df$customer_type)
df<-df[c(1,2,3,6:10,12,14,15,23,26,27)]
str(df)
```

```
## 'data.frame':    119390 obs. of  14 variables:
##  $ hotel                      : Factor w/ 2 levels "City Hotel","Resort Hotel": 2 2 2 2 2 2 2 2 2 2 ..
##  $ is_canceled                : Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 2 2 ...
##  $ lead_time                  : int  342 737 7 13 14 14 0 9 85 75 ...
##  $ arrival_date_week_number   : int  27 27 27 27 27 27 27 27 27 27 ...
##  $ arrival_date_day_of_month: int  1 1 1 1 1 1 1 1 1 1 ...
##  $ stays_in_weekend_nights    : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ stays_in_week_nights       : int  0 0 1 1 2 2 2 2 3 3 ...
##  $ adults                     : int  2 2 1 1 2 2 2 2 2 2 ...
##  $ babies                     : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ country                    : Factor w/ 178 levels "ABW","AGO","AIA",..: 137 137 60 60 60 60 137 137
##  $ market_segment             : Factor w/ 8 levels "Aviation","Complementary",..: 4 4 4 3 7 7 4 4 7 6
##  $ deposit_type               : Factor w/ 3 levels "No Deposit","Non Refund",..: 1 1 1 1 1 1 1 1 1 .
##  $ days_in_waiting_list       : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ customer_type              : Factor w/ 4 levels "Contract","Group",..: 3 3 3 3 3 3 3 3 3 3 ...
```

Check for null values

```
sapply(df, function(x) sum(is.na(x)))
```

```
##                      hotel                is_canceled                  lead_time
##                          0                          0                          0
##   arrival_date_week_number  arrival_date_day_of_month    stays_in_weekend_nights
##                          0                          0                          0
##       stays_in_week_nights                     adults                     babies
##                          0                          0                          0
##                    country             market_segment               deposit_type
##                          0                          0                          0
##       days_in_waiting_list              customer_type
##                          0                          0
```

#A Divide into train and test

My laptop does not like how big the data set is so the 119390 observations are reduced to 11939 before dividing into train, test, and validate.

```
set.seed(12345)
i <- sample(1:nrow(df), nrow(df)*.1, replace=FALSE)
shrink <- df[i,]
str(shrink)
```

```
## 'data.frame':    11939 obs. of  14 variables:
##  $ hotel                      : Factor w/ 2 levels "City Hotel","Resort Hotel": 1 1 1 1 1 1 1 1 1 1 ..
```

```
##  $ is_canceled              : Factor w/ 2 levels "0","1": 2 1 2 2 2 1 1 2 1 1 ...
##  $ lead_time                : int  358 41 61 355 62 51 53 552 0 6 ...
##  $ arrival_date_week_number : int  41 41 11 41 14 35 26 2 16 20 ...
##  $ arrival_date_day_of_month: int  10 5 18 7 6 22 24 12 11 14 ...
##  $ stays_in_weekend_nights  : int  1 0 0 0 0 1 1 0 1 2 ...
##  $ stays_in_week_nights     : int  1 2 1 2 3 2 2 2 0 1 ...
##  $ adults                   : int  2 1 3 2 1 2 2 2 1 1 ...
##  $ babies                   : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ country                  : Factor w/ 178 levels "ABW","AGO","AIA",..: 137 82 32 137 137 141 57 13
##  $ market_segment           : Factor w/ 8 levels "Aviation","Complementary",..: 5 7 7 5 5 7 5 5 4 5
##  $ deposit_type             : Factor w/ 3 levels "No Deposit","Non Refund",..: 2 1 1 2 2 1 1 2 1 1 .
##  $ days_in_waiting_list     : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ customer_type            : Factor w/ 4 levels "Contract","Group",..: 3 3 3 4 3 3 4 3 3 4 ...
```

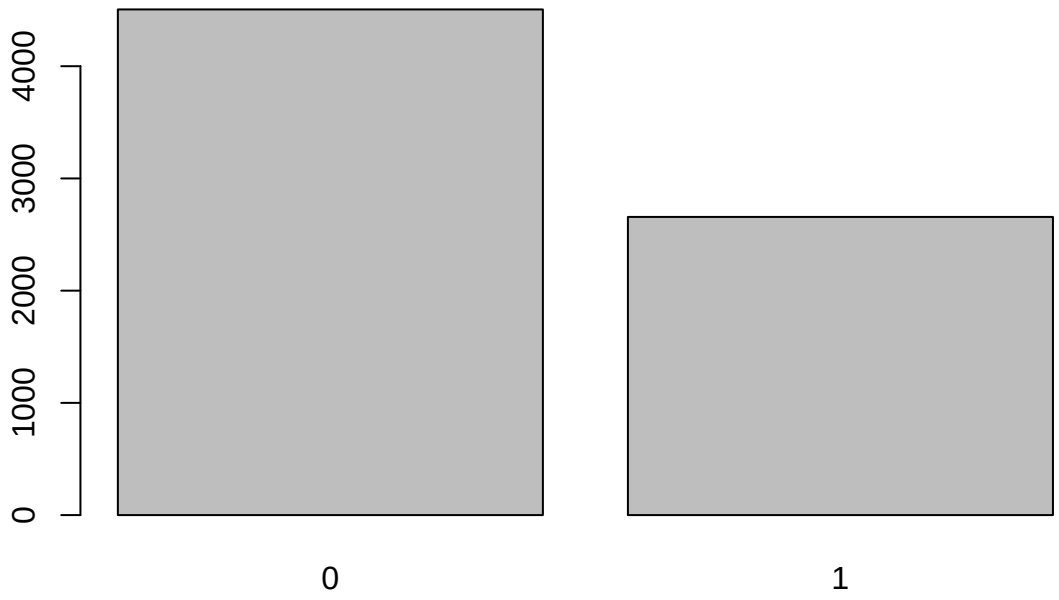
```
spec <- c(train=.6, test=.2, validate=.2)
i <- sample(cut(1:nrow(shrink), nrow(shrink)*cumsum(c(0,spec)), labels=names(spec)))
train <- shrink[i=="train",]
test <- shrink[i=="test",]
vald <- shrink[i=="validate",]
```

#B Explore training data

```
summary(train$is_canceled)
```

```
##    0    1
## 4506 2657
```

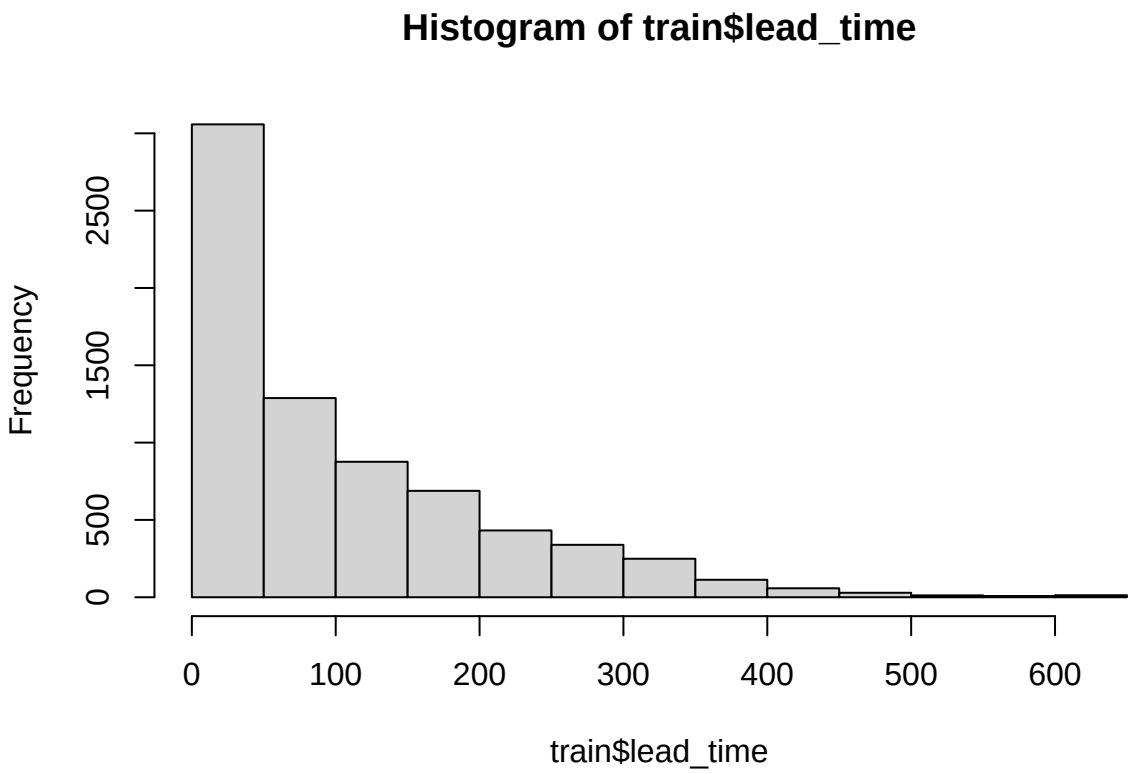
```
counts <- table(train$is_canceled)
barplot(counts)
```



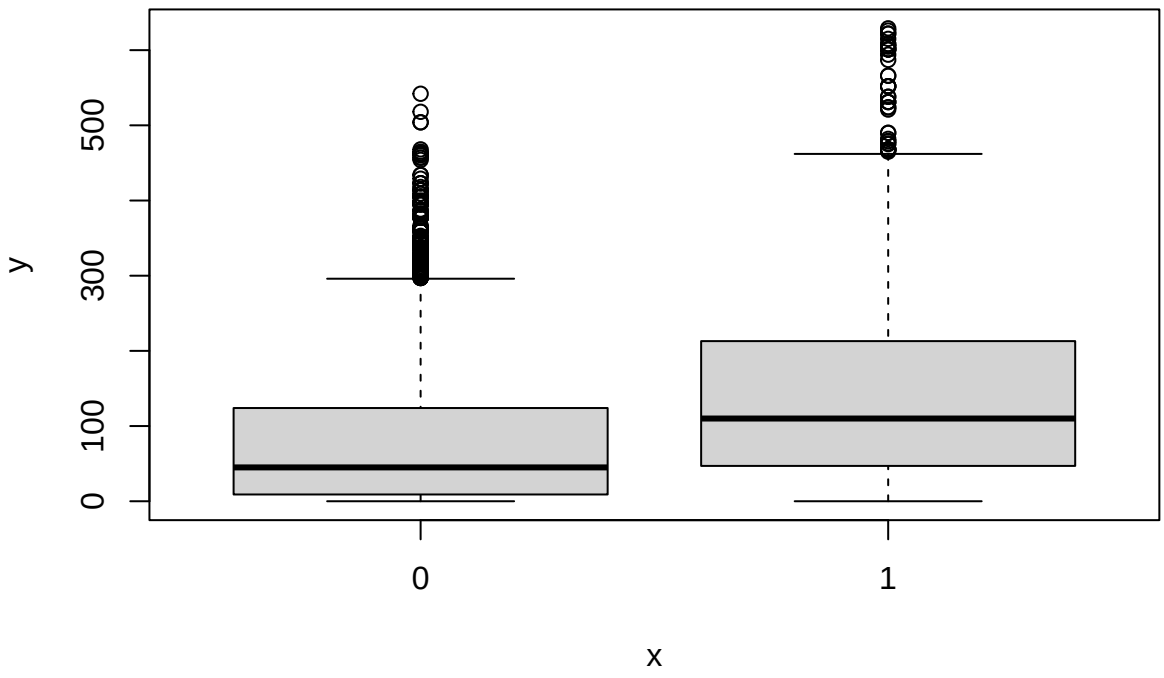

```
summary(train$lead_time)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##     0.0    18.0    70.0   104.1   160.0   629.0
```

```
hist(train$lead_time)
```



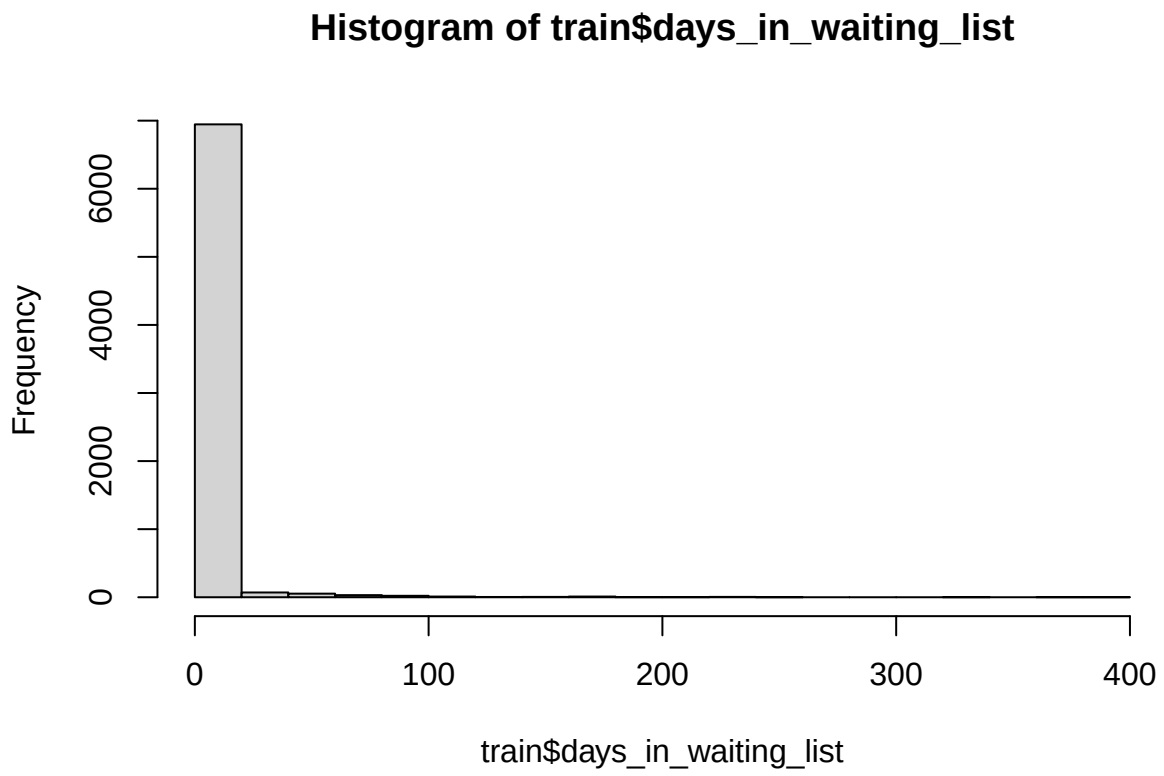

```
plot(train$is_canceled, train$lead_time)
```



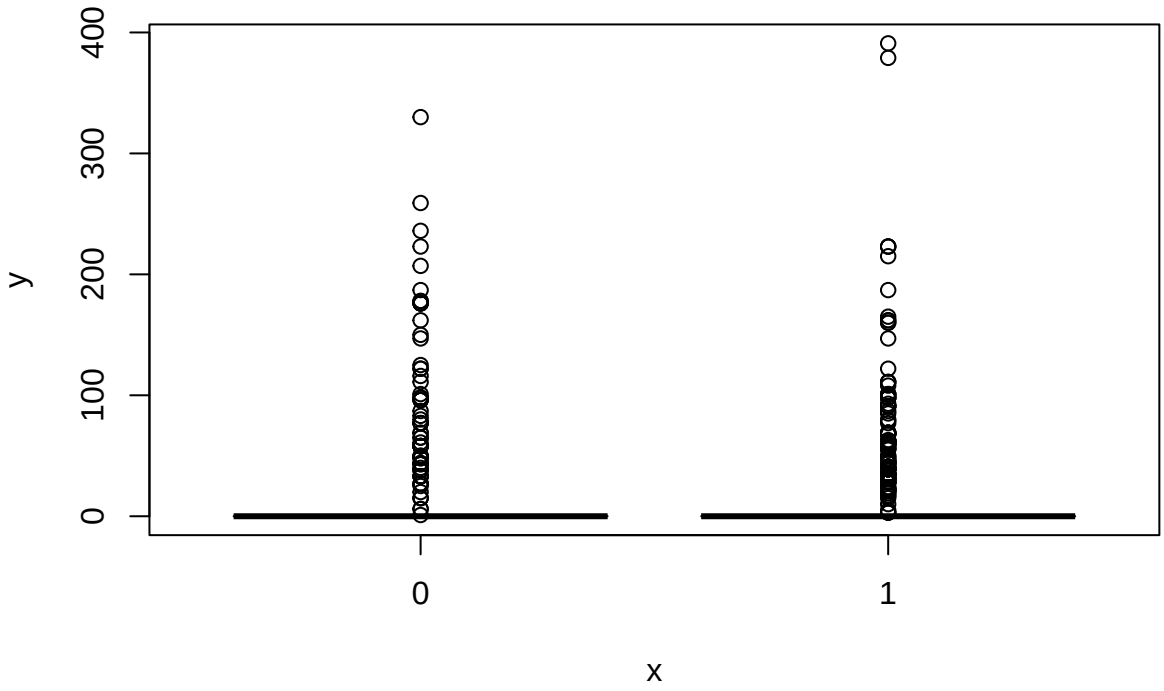

```
summary(train$days_in_waiting_list)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   0.000   0.000   0.000   2.319   0.000 391.000
```

```
hist(train$days_in_waiting_list)
```




```
plot(train$is_canceled, train$days_in_waiting_list)
```



#C SVM Classification

Linear Kernel with c of 10.

```
library(e1071)
svm1 <- svm(is_canceled~., data=train, kernel="linear", cost=10, scale=TRUE)
summary(svm1)
```

```
##
## Call:
## svm(formula = is_canceled ~ ., data = train, kernel = "linear", cost = 10,
##     scale = TRUE)
##
##
## Parameters:
##    SVM-Type:  C-classification
##  SVM-Kernel:  linear
##        cost:  10
##
## Number of Support Vectors:  3603
##
##  ( 1776 1827 )
##
##
## Number of Classes:  2
##
```

```
## Levels:
##  0 1
```

Output table and accuracy.

```
pred <- predict(svm1, newdata=test)
table(pred, test$is_canceled)
```

```
##
## pred    0    1
##    0 1453  449
##    1   57  429
```

```
mean(pred==test$is_canceled)
```

```
## [1] 0.7881072
```

Polynomial kernel with c of 10.

```
svm2 <-svm(is_canceled~., data=train, kernel="polynomial", cost=10, scale=TRUE)
summary(svm2)
```

```
##
## Call:
## svm(formula = is_canceled ~ ., data = train, kernel = "polynomial",
##     cost = 10, scale = TRUE)
##
##
## Parameters:
##    SVM-Type:  C-classification
##  SVM-Kernel:  polynomial
##        cost:  10
##      degree:  3
##      coef.0:  0
##
## Number of Support Vectors:  5343
##
##  ( 2646 2697 )
##
##
## Number of Classes:  2
##
## Levels:
##  0 1
```

Output table and accuracy.

```
pred2 <- predict(svm2, newdata=test)
table(pred2, test$is_canceled)
```

```
##
## pred2    0    1
##     0 1508  850
##     1    2   28
```

```
mean(pred2==test$is_canceled)
```

```
## [1] 0.6432161
```

Radial kernel with cost of 10 and gamma of 1.

```
svm3 <- svm(is_canceled~., data=train, kernel="radial", cost=10, gamma=1, scale=TRUE)
summary(svm3)
```

```
##
## Call:
## svm(formula = is_canceled ~ ., data = train, kernel = "radial", cost = 10,
##     gamma = 1, scale = TRUE)
##
##
## Parameters:
##    SVM-Type:  C-classification
##  SVM-Kernel:  radial
##        cost:  10
##
## Number of Support Vectors:  5594
##
##  ( 1927 3667 )
##
##
## Number of Classes:  2
##
## Levels:
##  0 1
```

Output table and accuracy.

```
pred3 <- predict(svm3, newdata=test)
table(pred3, test$is_canceled)
```

```
##
## pred3    0    1
##     0 1324  383
##     1  186  495
```

```
mean(pred3==test$is_canceled)
```

```
## [1] 0.7617253
```

Try tuning radial.

```
set.seed(12345)
tune.out <-tune(svm, is_canceled~., data=vald, kernel="radial", ranges=list(cost=c(0.1,1,10,100), gamma=
summary(tune.out)
```

```
##
## Parameter tuning of 'svm':
##
## - sampling method: 10-fold cross validation
##
## - best parameters:
##  cost gamma
##     1   0.5
##
## - best performance: 0.2328259
##
## - Detailed performance results:
##     cost gamma     error dispersion
## 1    0.1   0.5 0.3169878 0.02288478
## 2    1.0   0.5 0.2328259 0.02092895
## 3   10.0   0.5 0.2525017 0.02054745
## 4  100.0   0.5 0.2571147 0.02420143
## 5    0.1   1.0 0.3588587 0.02589165
## 6    1.0   1.0 0.2491474 0.02603433
## 7   10.0   1.0 0.2533385 0.02569840
## 8  100.0   1.0 0.2529183 0.02560815
## 9    0.1   2.0 0.3601174 0.02476900
## 10   1.0   2.0 0.2809711 0.03003777
## 11  10.0   2.0 0.2696776 0.02730548
## 12 100.0   2.0 0.2696776 0.02701903
## 13   0.1   3.0 0.3601174 0.02476900
## 14   1.0   3.0 0.2897595 0.02489752
## 15  10.0   3.0 0.2839123 0.02553586
## 16 100.0   3.0 0.2839123 0.02553586
```

Next trying the tuned radial kernel with c of 1 and gamma of .5

```
svm4 <- svm(is_canceled~., data=train, kernel="radial", cost=1, gamma=0.5, scale=TRUE)
summary(svm4)
```

```
##
## Call:
## svm(formula = is_canceled ~ ., data = train, kernel = "radial", cost = 1,
##     gamma = 0.5, scale = TRUE)
##
##
## Parameters:
##    SVM-Type:  C-classification
##  SVM-Kernel:  radial
##        cost:  1
##
## Number of Support Vectors:  4872
##
```

```
##  ( 1906 2966 )
##
##
## Number of Classes:  2
##
## Levels:
##  0 1
```

Output table and accuracy.

```
pred4 <- predict(svm4, newdata=test)
table(pred4, test$is_canceled)
```

```
##
## pred4    0    1
##     0 1382  379
##     1  128  499
```

```
mean(pred4==test$is_canceled)
```

```
## [1] 0.7876884
```

Trying to tune polynomial to see if there is any hope for it.

```
set.seed(12345)
tune.out <-tune(svm, is_canceled~., data=vald, kernel="polynomial", ranges=list(cost=c(0.1,1,10,100)))
summary(tune.out)
```

```
##
## Parameter tuning of 'svm':
##
## - sampling method: 10-fold cross validation
##
## - best parameters:
##  cost
##   100
##
## - best performance: 0.3241008
##
## - Detailed performance results:
##    cost     error dispersion
## 1   0.1 0.3596990 0.02512132
## 2   1.0 0.3592806 0.02515367
## 3  10.0 0.3592789 0.02473841
## 4 100.0 0.3241008 0.02532296
```

Next trying the tuned polynomial kernel with c of 100

```
svm5 <- svm(is_canceled~., data=train, kernel="polynomial", cost=100, scale=TRUE)
summary(svm5)
```

```
##
## Call:
## svm(formula = is_canceled ~ ., data = train, kernel = "polynomial",
##     cost = 100, scale = TRUE)
##
##
## Parameters:
##    SVM-Type:  C-classification
##  SVM-Kernel:  polynomial
##        cost:  100
##      degree:  3
##      coef.0:  0
##
## Number of Support Vectors:  4979
##
##  ( 2455 2524 )
##
##
## Number of Classes:  2
##
## Levels:
##  0 1
```

Output table and accuracy.

```
pred5 <- predict(svm5, newdata=test)
table(pred5, test$is_canceled)
```

```
##
## pred5    0    1
##     0 1505  690
##     1    5  188
```

```
mean(pred5==test$is_canceled)
```

```
## [1] 0.7089615
```

Trying to tune linear to see if it can improve.

```
set.seed(12345)
tune.out <-tune(svm, is_canceled~., data=vald, kernel="linear", ranges=list(cost=c(0.1,1,10,100)))
summary(tune.out)
```

```
##
## Parameter tuning of 'svm':
##
## - sampling method: 10-fold cross validation
##
## - best parameters:
##  cost
##   100
```

```
##
## - best performance: 0.2236156
##
## - Detailed performance results:
##    cost     error dispersion
## 1   0.1 0.2340740 0.02644917
## 2   1.0 0.2236173 0.03567245
## 3  10.0 0.2236173 0.03631955
## 4 100.0 0.2236156 0.03510901
```

Next trying the tuned linear kernel with c of 1

```
svm6 <- svm(is_canceled~., data=train, kernel="linear", cost=1, scale=TRUE)
summary(svm6)
```

```
##
## Call:
## svm(formula = is_canceled ~ ., data = train, kernel = "linear", cost = 1,
##     scale = TRUE)
##
##
## Parameters:
##    SVM-Type:  C-classification
##  SVM-Kernel:  linear
##        cost:  1
##
## Number of Support Vectors:  3614
##
##  ( 1787 1827 )
##
##
## Number of Classes:  2
##
## Levels:
##  0 1
```

Output table and accuracy.

```
pred6 <- predict(svm6, newdata=test)
table(pred6, test$is_canceled)
```

```
##
## pred6    0    1
##     0 1458  453
##     1   52  425
```

```
mean(pred6==test$is_canceled)
```

```
## [1] 0.788526
```

#D Analysis

```
mean(pred==test$is_canceled)
```

```
## [1] 0.7881072
```

```
mean(pred2==test$is_canceled)
```

```
## [1] 0.6432161
```

```
mean(pred3==test$is_canceled)
```

```
## [1] 0.7617253
```

```
mean(pred4==test$is_canceled)
```

```
## [1] 0.7876884
```

```
mean(pred5==test$is_canceled)
```

```
## [1] 0.7089615
```

```
mean(pred6==test$is_canceled)
```

```
## [1] 0.788526
```

The order was linear, polynomial, radial, tuned radial, tuned polynomial, tuned linear. Polynomial performed the worst even after turning against both linear and radial. I am not sure why it performed worse because intuitively polynomial should perform at least as well as linear. Linear had minor gains from its tuning but that might be the result of not exploring more c values or that the starting value was close to the final. The difference between linear and radial were small after tuning which could be down to random chance and other factors. Overall I think that the data is probably linearly separable which lead to small gains over using linear kernel. Some issues might have happened because the data had to be shrunk in order to run on my computer.