Oyo Case

P Chatterji

Oyo Case

Revenue Maximization

Contents

Problem Statement

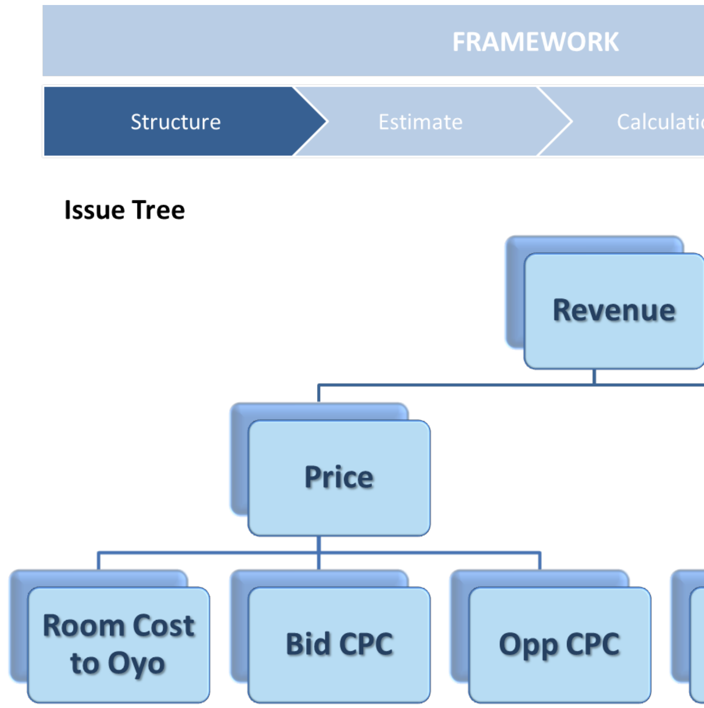
Oyo rooms advertise their facilities to a Network Agency, `Trivago`, who charge them at a Bid CPC for each ad. A combination of `Bid CPC` and Price of facility determines how highly the ad is placed. Oyo wants to improve its Bids to achieve maximum revenue with Trivago.

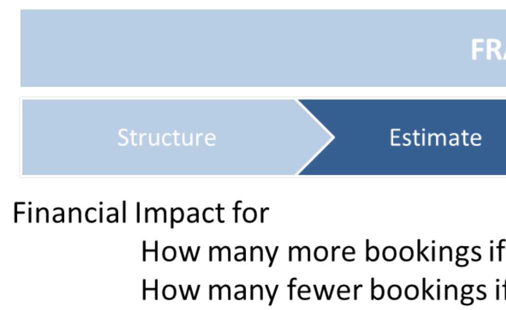
Framework

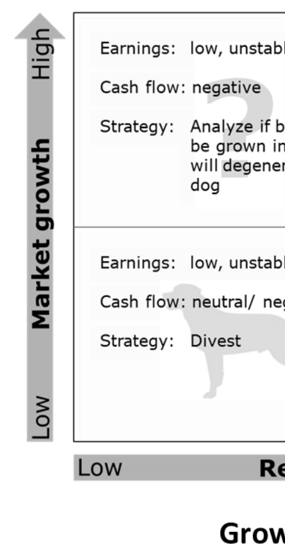
Even before the problem is considered, we must explore the boundary conditions to evaluate Revenue maximization exercise can make sense.

```
## Loading required package: readr

## Loading required package: tidyverse

## Warning: package 'tidyverse' was built under R version 3.5.1

## -- Attaching packages --------------------

## v ggplot2 2.2.1     v purrr   0.2.4
## v tibble  1.4.2     v dplyr   0.7.4
## v tidyr   0.8.0     v stringr 1.3.1
## v ggplot2 2.2.1     v forcats 0.3.0

## -- Conflicts ---- tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

We assume the following structure for the Revenue system:



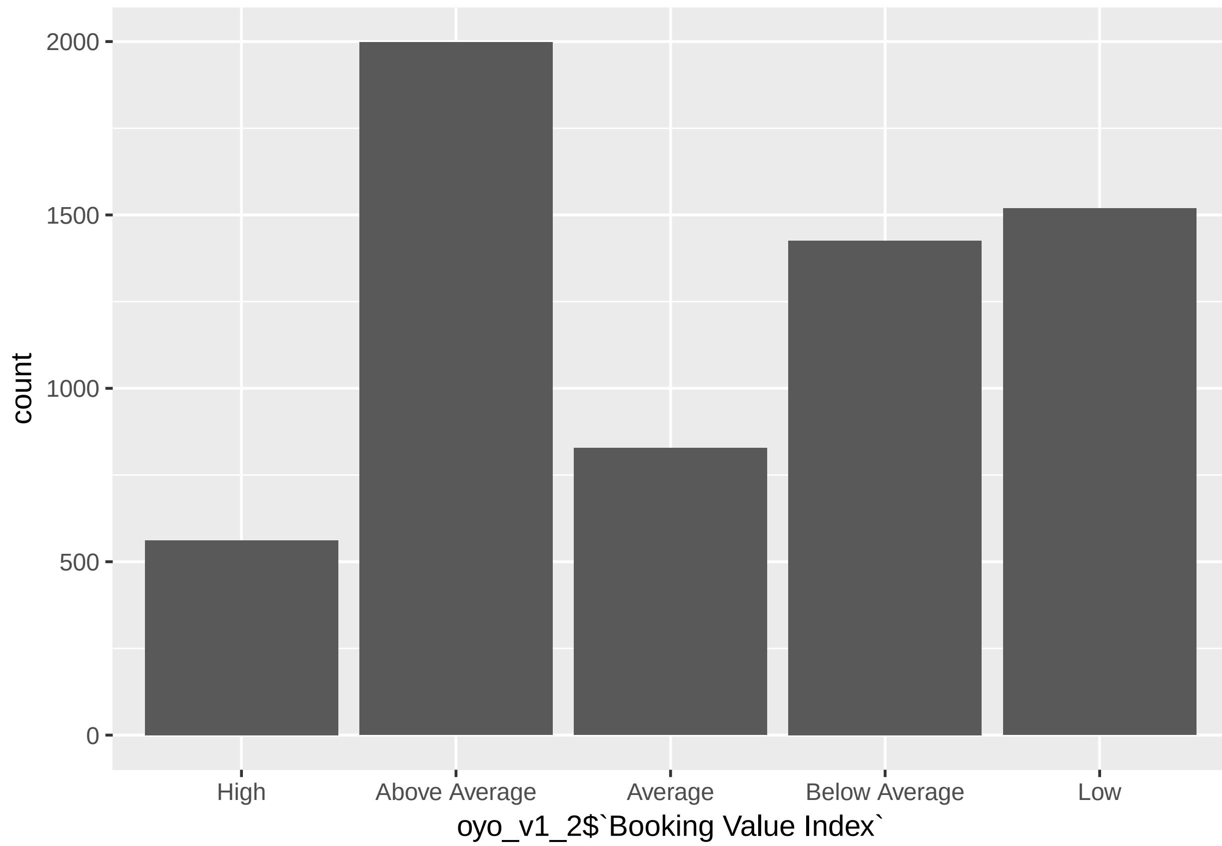




One of the ways, Oyo can calssify its properties is the GE Growth-Revenue matrix.

There is question mark on facilities that have not been evenly utilized. They should be shifted to revenue generating to the Star section.

```r
require("ggplot2")
# counts
ggplot(oyo_v1_2, aes(x=oyo_v1_2$`Booking Value Index`)) +
  geom_bar()
```



With the given distribution, it makes sense that some *Average* and *Below Average* bookings may be converted to more profitable tiers. This is our *Hypothesis*. In our discussion, we would either prove the null Hypothesis or otherwise.

Consider that Trivago's market is highly price sensitve. Additionally, a percentage change in Booking Value of an Oyo facility would result in larger cost than increasing the Bid value (which can be gathered from the `Opportunity CPC`). In terms of Pricing strategies, Competitive Beanchmarking is the applicable method here.

Estimation

We would need to estimate the *minimum change in bid cpc* that drives maximum revenue.

Therefore, the new bid value = `Opportunity CPC` + `Bid CPC`

Accordingly, the impressions would change from `Hotel Impressions` to `Maximum Possible Impressions`

Finally, we would need a surrogate for `Booking rate` that will associate with the trend of improved Revenue – `Response rate`. It should consider the numerical and categorical factors within the data set.

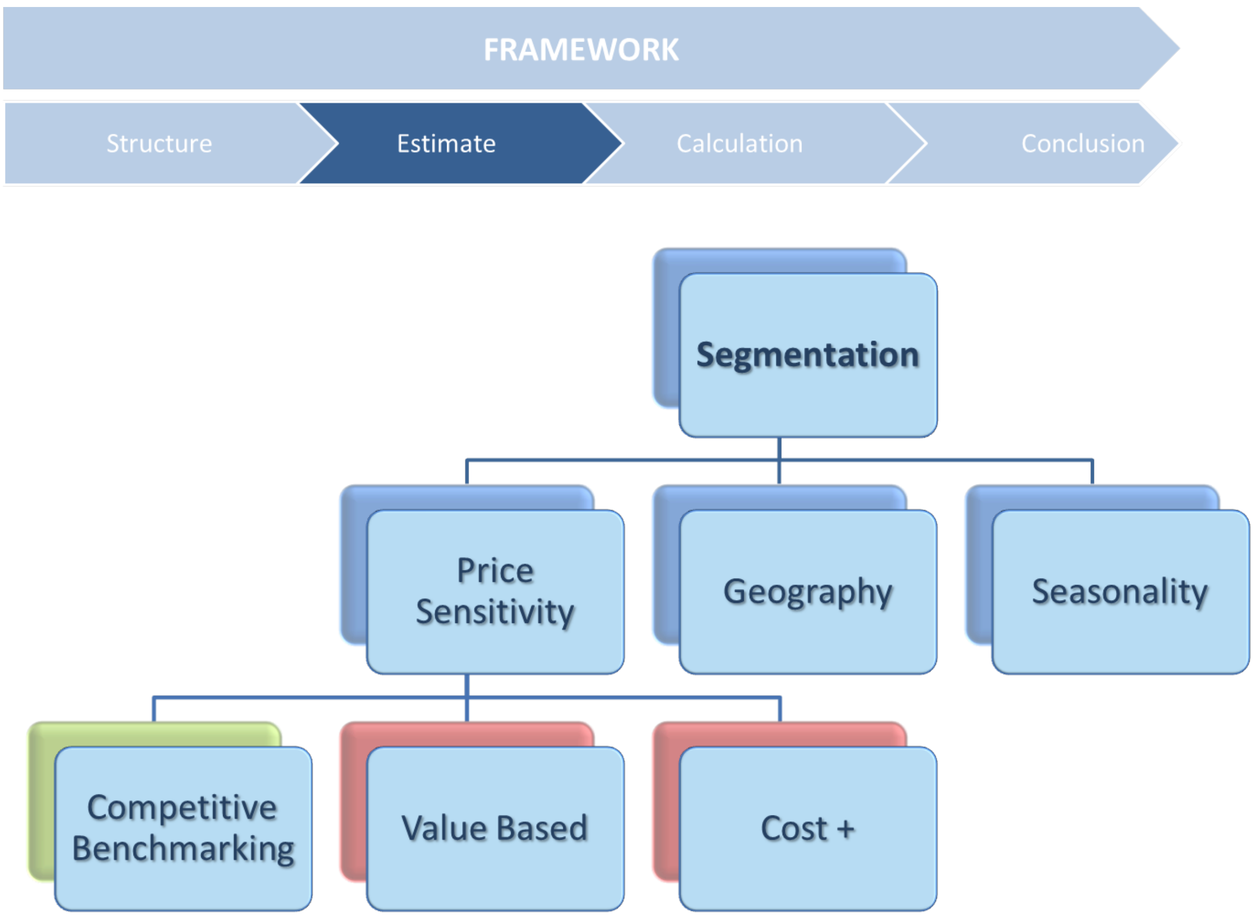


Figure 1: Issue Segmentation

Calculation

First, we calculate the Bid Price, Booking Availability, Maximum Impressions, and Maximum Booking Potential.

`Maximum Booking Potential` is the maximum number of days that can be booked at the Oyo facility. It is calculated as the minimum of the two: * Days left in the month * Maximum days that can be booked from improvement in advertisement

```
require(dplyr)

oyo_v1_2 <- tbl_df(oyo_v1_2)
# glimpse(oyo_v1_2)




o1 <- mutate(oyo_v1_2,
             Always_Win_Bid_CPC = oyo_v1_2$`Bid CPC` + oyo_v1_2$`Opportunity CPC`,
             Booking_Availability_in_the_month = 30 - oyo_v1_2$Bookings,
             Max_Possible_Impressions = oyo_v1_2$`Hotel Impr` + oyo_v1_2$`Max Potential`,
             Max_Booking_Potential = floor(oyo_v1_2$`Max Potential` * (oyo_v1_2$Bookings / oyo_v1_2$`Ho
             )
# glimpse(o1)
```

First, we need a cleaner, relevant dataset. The twin objecctives here are: * Create a dataset that can be used to train a model * Create a Look-up table to identify the relevant sections

```
#Creating a relevant data set
o1_test <- select(o1, PartnerRef, `Bid CPC`, `Opportunity CPC`, `Hotel Impr`, Clicks, `Avg. CPC`, `Top P
# glimpse(o1_test)

# Selecting rows without any missing values
o1_test <- o1_test[complete.cases(o1_test), ]

# Extract Partner Reference COlumn

# Copying Reference COlumn
o1_test_partnerRef <- o1_test$PartnerRef

# Selecting tbl minus Partner Column
o1_test <- select(o1_test, `Bid CPC`, `Opportunity CPC`, `Hotel Impr`, Clicks, `Avg. CPC`, `Top Pos Shar
# glimpse(o1_test)
```

`Response Rate` would be our closest surrogate to the `Booking rate`. It would signify how well the customers might react to the new ad campgain by Oyo.

In this exercise, we will use 10-fold CrossValidation method as used in `caret` package. Model fitment is tested by in-sample RMSE in this case.

```
require("caret")
```

```
## Loading required package: caret

## Warning: package 'caret' was built under R version 3.5.1

## Loading required package: lattice

##
## Attaching package: 'caret'
```

```
## The following object is masked from 'package:purrr':
##
##     lift
```

```r
#set seed for reproducability
set.seed(42)

# Fit lm model using 10-fold CV: model
foomodel <- train(
  `Booking Rate` ~ ., o1_test,
  method = "lm",
  trControl = trainControl(
    method = "cv", number = 10,
    verboseIter = TRUE
  )
)
```

```
## + Fold01: intercept=TRUE
## - Fold01: intercept=TRUE
## + Fold02: intercept=TRUE
## - Fold02: intercept=TRUE
## + Fold03: intercept=TRUE
## - Fold03: intercept=TRUE
## + Fold04: intercept=TRUE
## - Fold04: intercept=TRUE
## + Fold05: intercept=TRUE
## - Fold05: intercept=TRUE
## + Fold06: intercept=TRUE
## - Fold06: intercept=TRUE
## + Fold07: intercept=TRUE
## - Fold07: intercept=TRUE
## + Fold08: intercept=TRUE
## - Fold08: intercept=TRUE
## + Fold09: intercept=TRUE
## - Fold09: intercept=TRUE
## + Fold10: intercept=TRUE
## - Fold10: intercept=TRUE
## Aggregating results
## Fitting final model on full training set
```

```r
predicted_response_rate <- predict(foomodel, o1_test, type = "raw")
actual <-  o1_test$`Booking Rate`
rmse_in_sample <- sqrt(mean((actual-predicted_response_rate)^2))
rmse_in_sample
```

```
## [1] 0.06683033
```

```r
o1_test$Predicted_Response_Rate <- abs(predicted_response_rate)
```

Considering that we have received a good, low Root Mean Square Errror of 0.067, we combine the dataset again.

```r
# Reattaching the Partner Reference Column
o1_test$PartnerRef <-  o1_test_partnerRef
```

Now, we calculate the revenue, cost and Profit from the exercise.

```r
#glimpse(o1_test)
o1_test <- mutate(o1_test,
             Increased_Cost = (o1_test$`Max Potential`) * (o1_test$`Bid CPC`+ o1_test$`Opportunity CPC`
             Possible_Bookings = pmin(o1_test$Max_Booking_Potential, o1_test$Booking_Availability_in_the
             Increased_Revenue =  Possible_Bookings * o1_test$`Gross Rev`,
             Increased_Profit = Increased_Revenue - Increased_Cost
             )
# head(o1_test$`Gross Rev`)
# newData_view <- select(o1_test,Increased_Profit, Increased_Cost, Increased_Revenue)
# head(newData_view)
```

The entire data set is now sorted by Profits. After that, we have distributed the facilitiesin 10 groups. The final result is stoed in `Uplift table`

```r
# ntile

o1_test %>% mutate(decile = ntile(desc(Increased_Profit), 10)) -> oyo_deciles
head(oyo_deciles)
```

```
## # A tibble: 6 x 20
##   `Bid CPC` `Opportunity CPC` `Hotel Impr` Clicks `Avg. CPC`
##       <dbl>             <dbl>        <int>  <int>      <dbl>
## 1      0.18              0.23        26255   1100      0.18
## 2      0.27              0.34        10902   1017      0.290
## 3      0.14              0.19        13083    995      0.23
## 4      0.21              0.26        19971    837      0.18
## 5      0.13              0.2         20950    701      0.24
## 6      0.21              0.26        12603    701      0.19
## # ... with 15 more variables: `Top Pos Share` <dbl>, `Impr Share` <dbl>,
## #   `Max Potential` <int>, Max_Booking_Potential <dbl>, `Gross Rev` <dbl>,
## #   `Booking Rate` <dbl>, `Booking Value Index` <fct>,
## #   Booking_Availability_in_the_month <dbl>,
## #   Predicted_Response_Rate <dbl>, PartnerRef <chr>, Increased_Cost <dbl>,
## #   Possible_Bookings <dbl>, Increased_Revenue <dbl>,
## #   Increased_Profit <dbl>, decile <int>
```

```r
oyo_deciles %>% group_by(decile) %>% # Group by decile
  summarize(response_rate_percent = round((mean(Predicted_Response_Rate)*100),2),
            Profit = sum(Increased_Profit),
            Cost = sum(Increased_Cost),
            Return_On_Ad_Spend = round(Profit/Cost,2),
            count = n()) -> Uplift_table

Uplift_table
```

```
## # A tibble: 10 x 6
##    decile response_rate_percent    Profit     Cost Return_On_Ad_Spe~ count
##     <int>                 <dbl>     <dbl>    <dbl>             <dbl> <int>
##  1      1                  6.02    97754.  112664.              0.87   509
##  2      2                  3.8    -39570.   47562.             -0.83   508
##  3      3                  3.62   -69570.   75097.             -0.93   508
##  4      4                  3.55  -108465.  118147.             -0.92   509
##  5      5                  3.2   -160517.  171737.             -0.93   508
##  6      6                  3.26  -237501.  256348.             -0.93   508
##  7      7                  3.22  -369175.  395094.             -0.93   509
```

```
## 8       8                    3.11  -616632.  657094.             -0.94   508
## 9       9                    3.41 -1203771. 1276943.             -0.94   508
## 10     10                    3.38 -4464942. 4555621.             -0.98   508
```

Conclusion

- Oyo should retarget the deciles with positive porfit – 1.
- Response rate is an indicator of better returns. Anything below 4 is not profitable.
- Require Ensamble Algorithms for better RMSE or R2
- Geography based analysis may shift pricing from competitive (low Revenue ? Group) to Stars or Dogs group
- A robust Pricing system (requires more data on Domain Expretise, Seasonality and Time Series Projection)