Take Home

Rahul Zende

April 26, 2019

Use the libraries as needed -

```
library(dplyr)
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library(ggplot2)
library(corrplot)
```

```
## Warning: package 'corrplot' was built under R version 3.5.3
```

```
## corrplot 0.84 loaded
```

```
library(leaps)
```

```
## Warning: package 'leaps' was built under R version 3.5.3
```

```
library(tree)
```

We read the file into our environment -

```
setwd("~/OneDrive - UW/Data Science/Glassdoor")
data <- read.csv('input.csv')
```

Now, we check the summary of the data being loaded -

```
summary(data)
```

```
##    ï..Start.Date       End.Date     Employer.ID           Employer.City
##  1-Jan-16 : 111   31-Dec-16: 122   Min.   :    40   na           : 834
##  1-Feb-16 :  44   31-Jan-17:  48   1st Qu.: 22670   New York     :  92
##  30-Jun-15:  42   30-Mar-17:  41   Median : 36959   San Francisco:  49
##  30-Sep-15:  42   28-Feb-17:  36   Mean   : 34800   Chicago      :  41
##  31-Mar-16:  42   29-Dec-16:  36   3rd Qu.: 48720   Los Angeles  :  37
##  1-Feb-15 :  41   29-Sep-16:  34   Max.   :108098   Atlanta      :  32
##  (Other)  :2756   (Other)  :2761                    (Other)      :1993
##         Employer.State Number.of.Slots     Price.Paid
##  na            : 834   Min.   : 10.0   $ 10,740 : 457
##  California    : 366   1st Qu.: 15.0   $ 26,850 : 251
##  New York State: 152   Median : 25.0   $ 16,110 : 238
##  Florida       : 146   Mean   : 34.7   $ 14,320 : 210
```

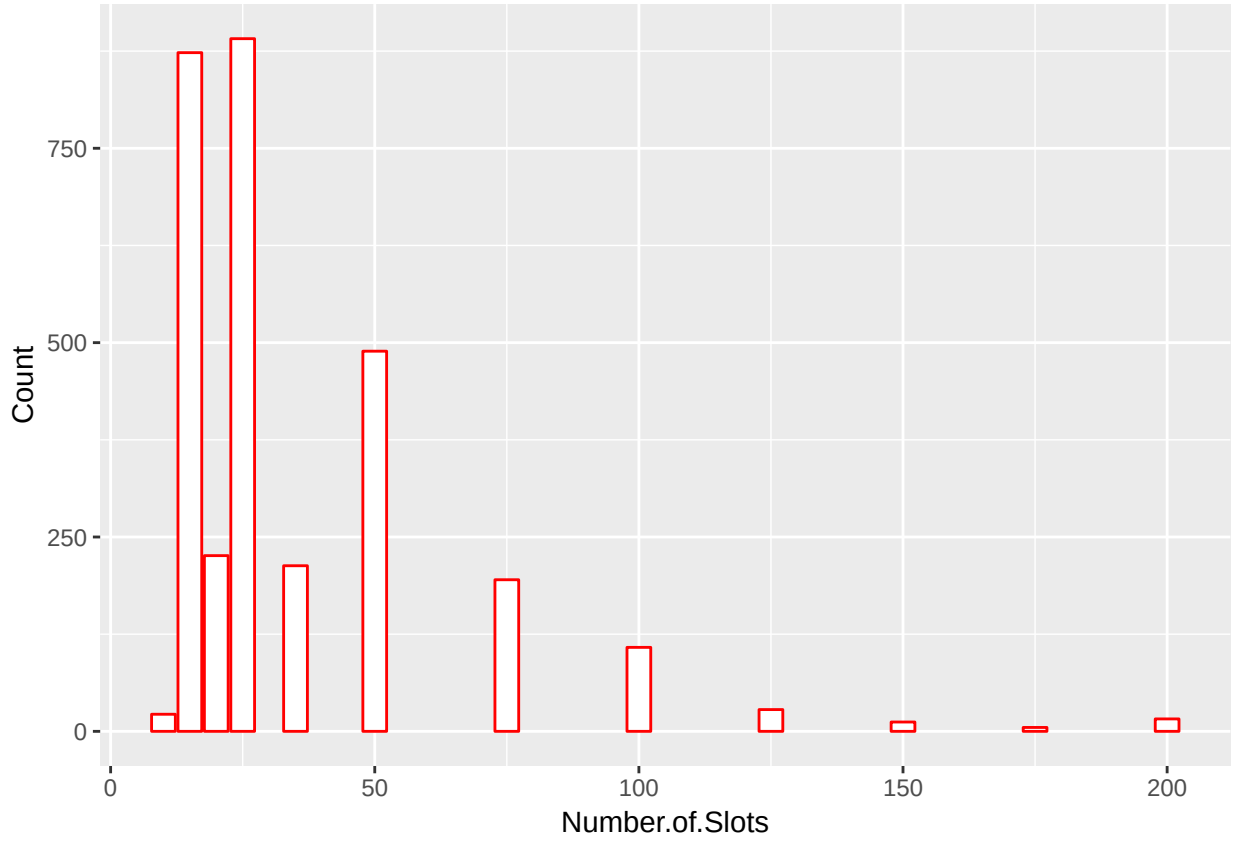
```
##  Texas          : 130   3rd Qu.: 50.0   $ 13,425 : 207
##  Illinois       :  85   Max.   :200.0   $ 21,480 : 187
##  (Other)        :1365                   (Other)  :1528
##  Marketplace.Value.Delivered  Applications     Renewed.
##  $ 10,498 :   3               1,008 :   6   Min.   :0.000
##  $ 9,177  :   3               1,119 :   6   1st Qu.:0.000
##  $ 1,390  :   2               154   :   6   Median :1.000
##  $ 1,774  :   2               648   :   6   Mean   :0.678
##  $ 1,969  :   2               1     :   5   3rd Qu.:1.000
##  $ 10,059 :   2               1,039 :   5   Max.   :1.000
##  (Other)  :3064              (Other):3044
```

PART 1: JOB SLOT SALES AND PERFORMANCE DISTRIBUTION

What is the variation across the job packages we sell? -

What packages are most popular, based on the number of slots?

```
job.slots.count.by.number.of.jobs <- aggregate(Employer.ID ~ Number.of.Slots, data = data, FUN = NROW)
job.slots.count.by.number.of.jobs <- job.slots.count.by.number.of.jobs %>% rename(Count = Employer.ID)
ggplot(job.slots.count.by.number.of.jobs, aes(x=Number.of.Slots, y=Count)) + geom_col(color="red", fill=
```




```
# write.csv(job.slots.count.by.number.of.jobs, file = 'job_slots_count_by_number_of_jobs.csv', row.name
```

The packages with 25 and 15 slots are most popular, with 891 and 873 contracts being signed respectively for those categories.

What does the distribution of pricing look like?

```
data$Price.Paid <- gsub("[^[:alnum:]]", "", data$Price.Paid)
data$Price.Paid <- as.numeric(data$Price.Paid)
mean(data$Price.Paid)
```

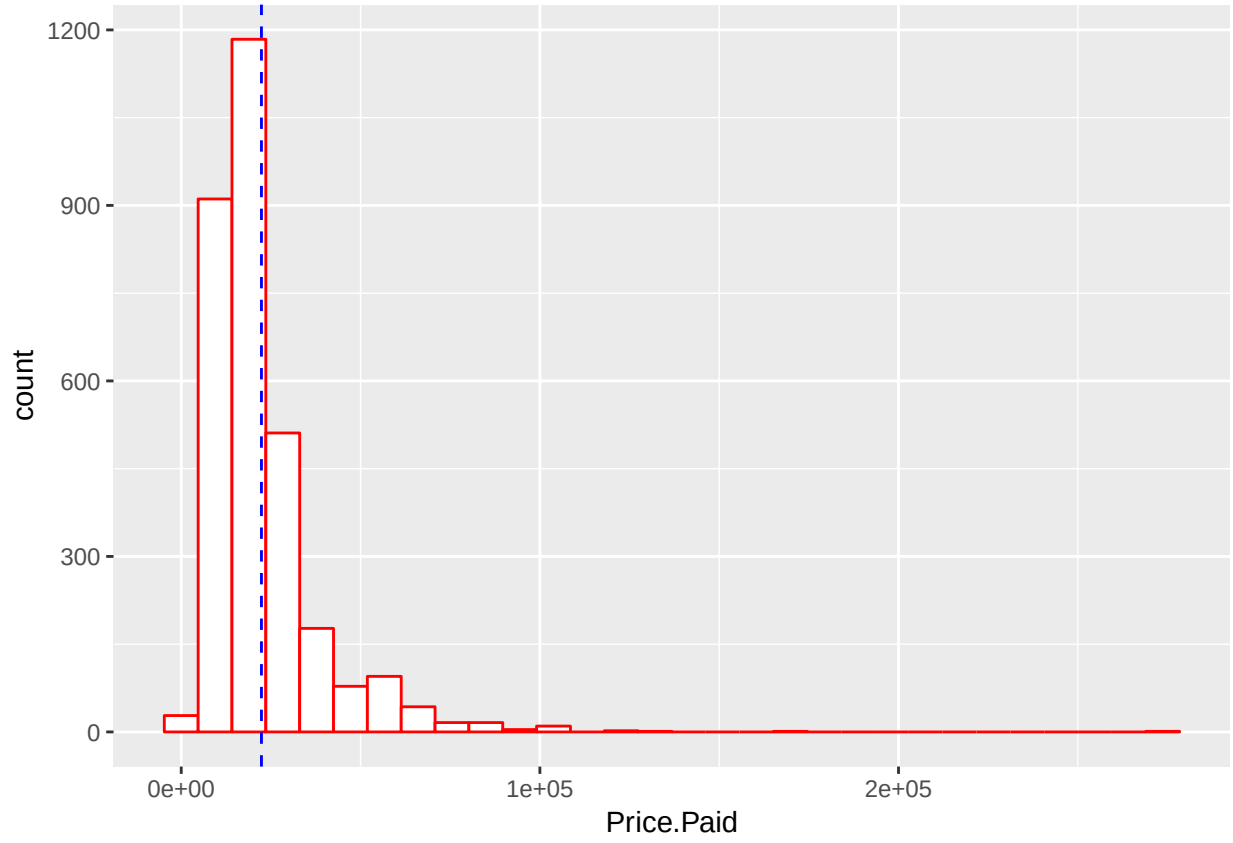
```
## [1] 22367.55
```

```
median(data$Price.Paid)
```

```
## [1] 17096.5
```

```
ggplot(data, aes(x=Price.Paid)) + geom_histogram(color='red', fill='white') + geom_vline(xintercept = m
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```




```
# boxplot(data$Price.Paid)
```

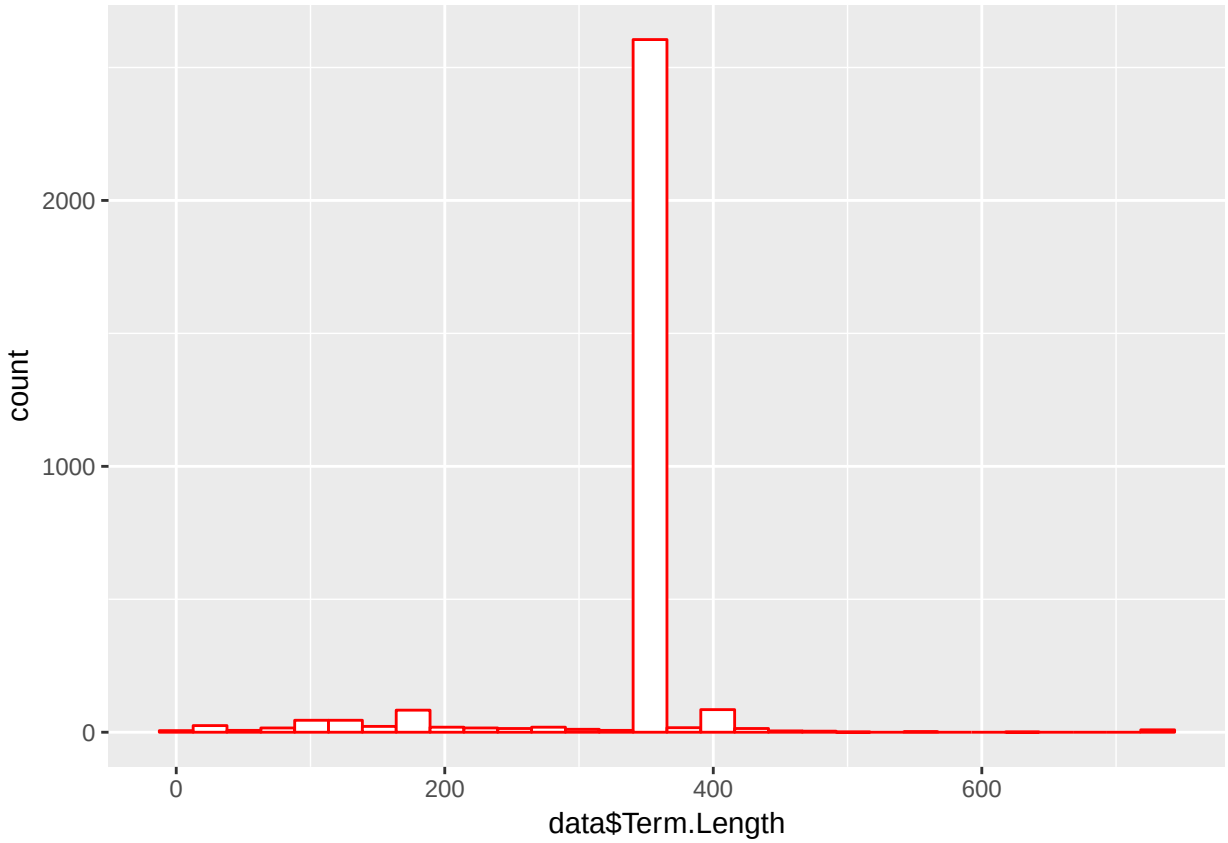
Based on the above distribution, we can see that most contracts cost around $17,096.5 (median). Also, the mean price per package is $22,367.55 - but this is higher because of outliers present in the data (for example, one package was sold for $2,73,870 : which ends up skewing the mean calculation).

How does term length vary?

```
data <- data %>% rename(Start.Date = ï..Start.Date)
data$Start.Date <- as.character(data$Start.Date)
data$End.Date <- as.character(data$End.Date)
data$Start.Date <- strptime(data$Start.Date, format = '%d-%b-%y')
data$End.Date <- strptime(data$End.Date, format = '%d-%b-%y')
data$Term.Length <- round(difftime(data$End.Date, data$Start.Date, units = 'days'), 0)
data$Term.Length <- as.numeric(data$Term.Length)

ggplot(data, aes(x=data$Term.Length)) + geom_histogram(color='red', fill='white')
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



As we can see here, the overwhelming favorite amongst contracts is the one with 1 year term (duration of 365 days). Out of the total contracts (3078) in this dataset, 2133 contracts are signed for a duration of 1 year - which is 69%.

What metrics should we use to compare delivery performance across customers? How does performance vary in terms of:

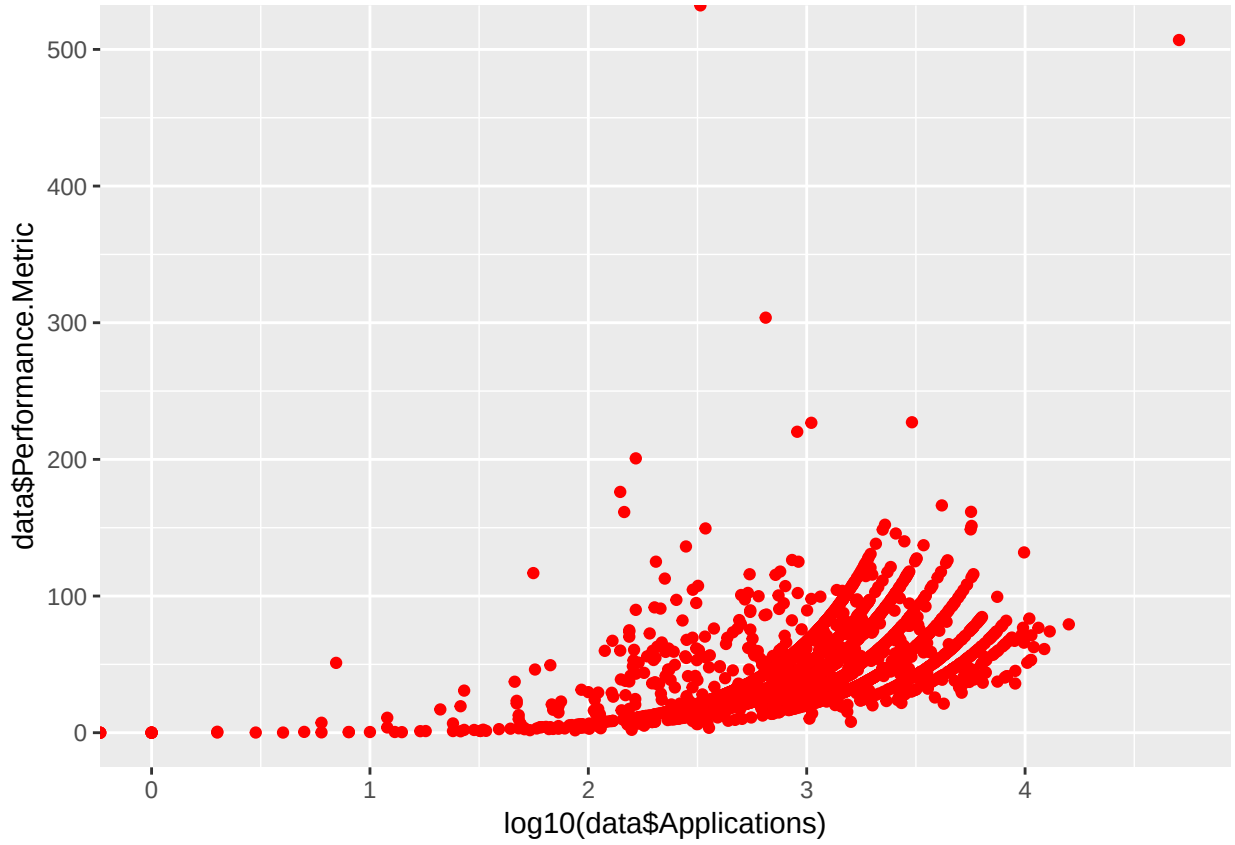
As per the information given, the performance of products is evaluated by customers based on number of applications they get vs cost-per-application. Let's create a calculated column in our dataframe to indicate that metric indicating performance.

```
data$Marketplace.Value.Delivered <- gsub("[^[:alnum:]]", "", data$Marketplace.Value.Delivered)
data$Marketplace.Value.Delivered <- as.numeric(data$Marketplace.Value.Delivered)
data$Applications <- as.character(data$Applications)
data$Applications <- gsub(",", "", data$Applications)
data$Applications <- as.numeric(data$Applications)

data$Cost.Per.Application <- data$Marketplace.Value.Delivered / data$Applications
data$Performance.Metric <- data$Applications / (data$Number.of.Slots * data$Term.Length / 365) # we use
```

- Application Metrics?

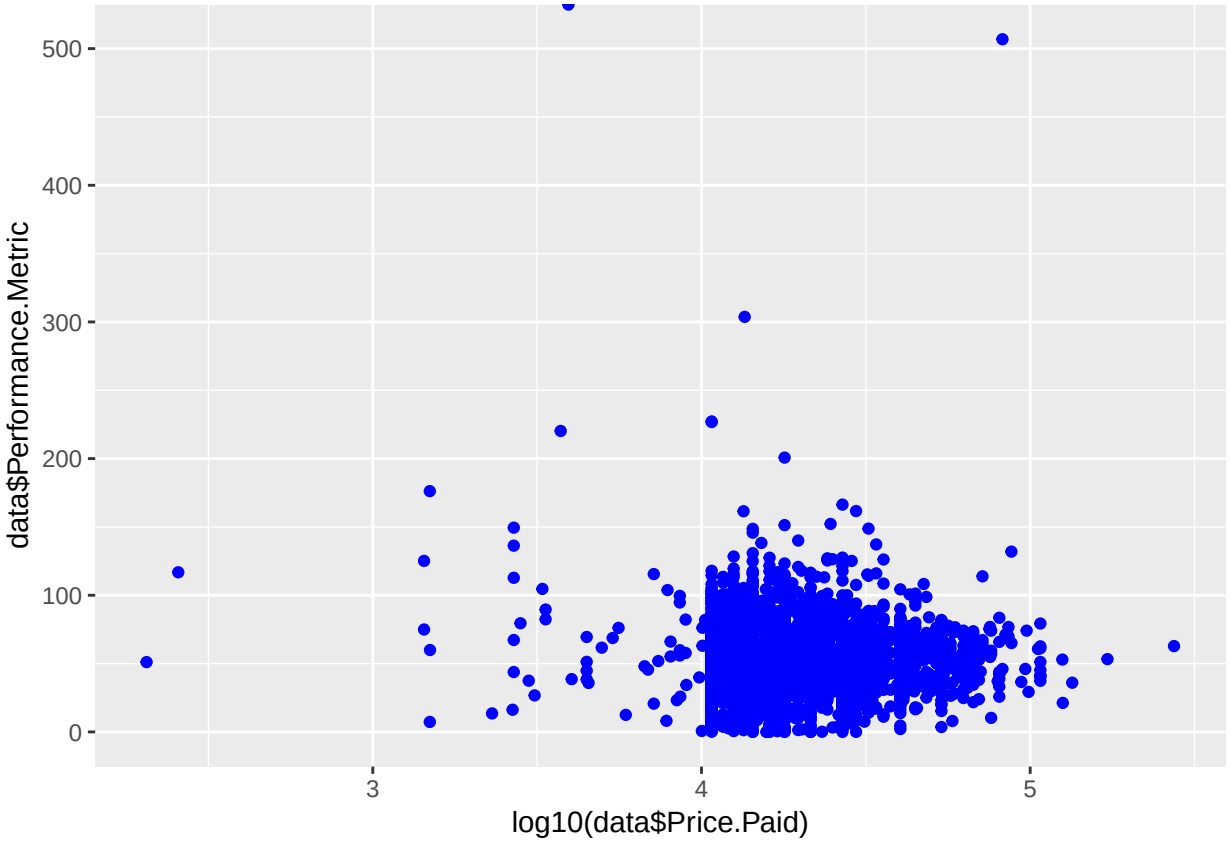
```
# So, we plot the number of applications against the performance metric
ggplot(data, aes(x=log10(data$Applications))) + geom_point(aes(y=data$Performance.Metric), color='red')
```



The relationship between the performance metric and number of appears to be decidely non-linear - i.e. exponential in nature.

- Cost?

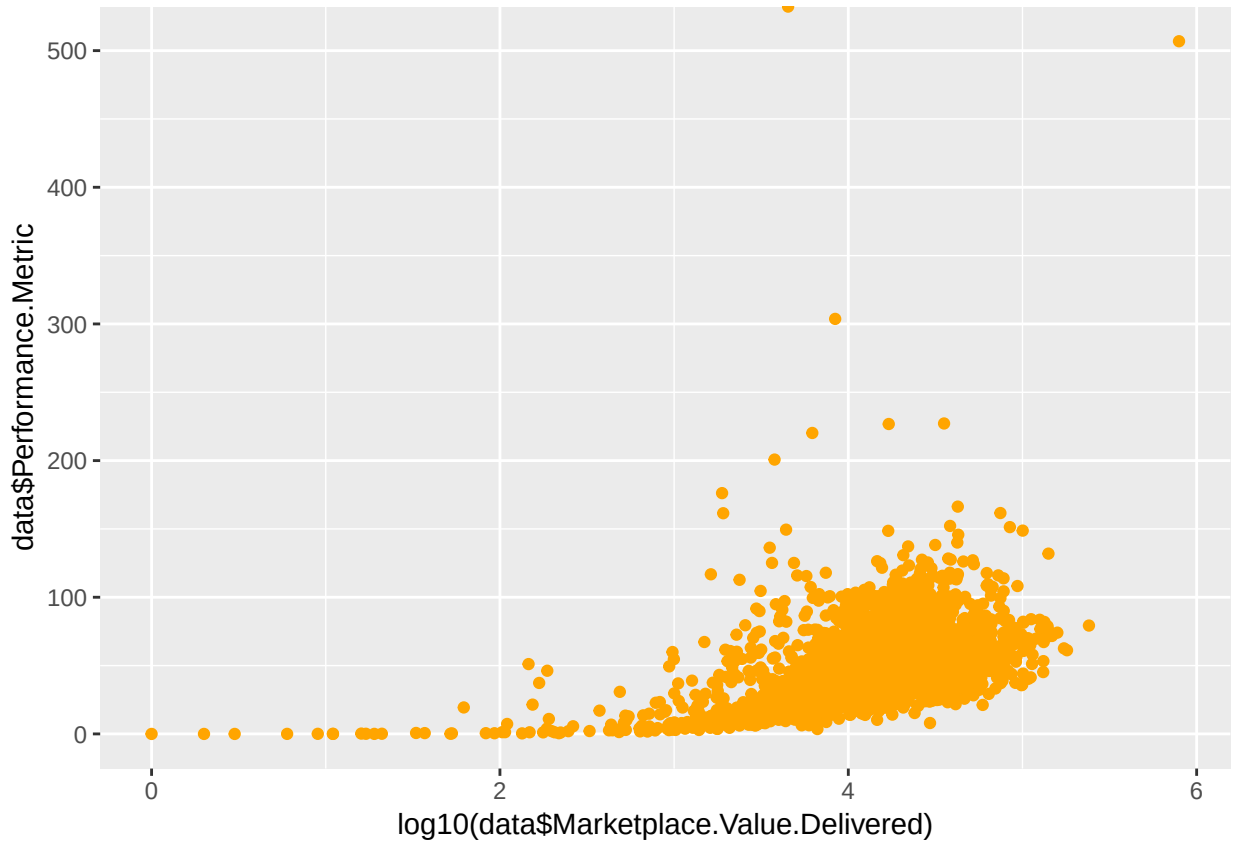
```
# So, we plot the cost per package/contract against the performance metric
ggplot(data, aes(x=log10(data$Price.Paid))) + geom_point(aes(y=data$Performance.Metric), color='blue')
```



Here, the relationship between the performance metric and Price Paid is not to clear. It is hard to spot a trend here.

- Marketplace value?

```
# So, we plot the marketplace value against the performance metric
ggplot(data, aes(x=log10(data$Marketplace.Value.Delivered))) + geom_point(aes(y=data$Performance.Metric
```



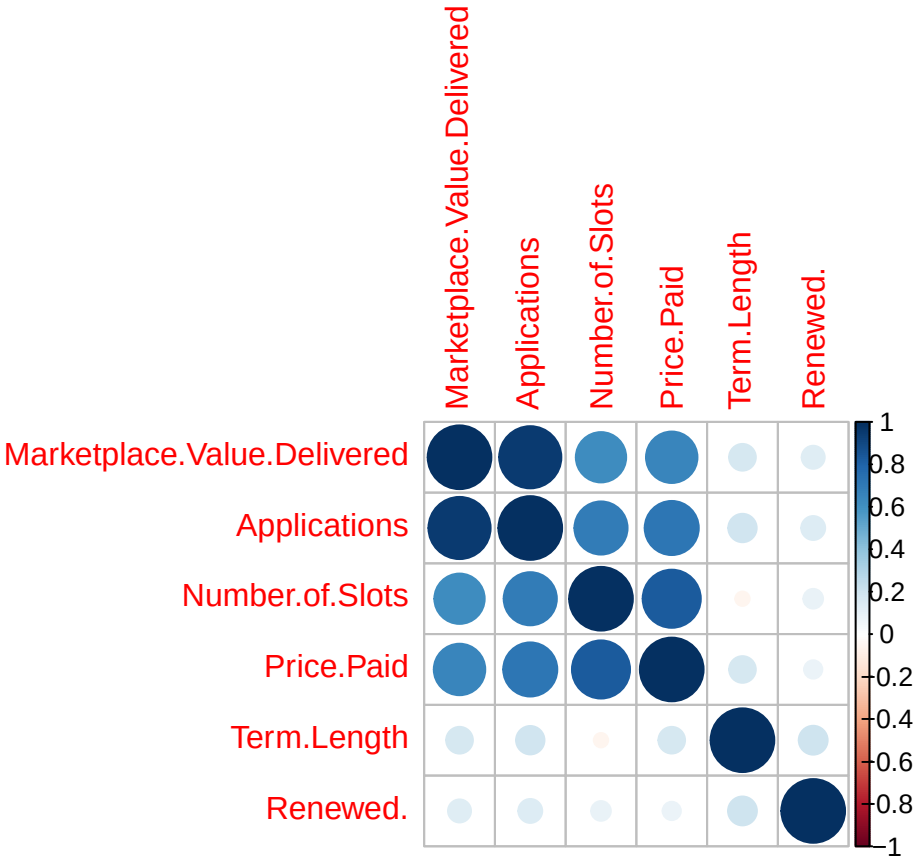
The relationship here, appears to be non-linear again - and seems to be exponential.

Based on the above plots, we can see that the logarithm (base 10) all of 2 metrics (Applications & Marketplace Value Delivered) correlate well with our calculated performance metric.
Comparing all plots above, the performance metric seems to be something feasible that can be used to evaluate performance across customers!

PART 2: RETENTION ANALYSIS

What factor or combination of factors best predict likelihood to retain (i.e., Renewed = 1)? Which factors appear to have the greatest impact on retention?

```
data.cor <- cor(select(data, Term.Length, Number.of.Slots, Price.Paid, Marketplace.Value.Delivered, Appl
corrplot(data.cor, order = "hclust")
```




```
reg.fss <- regsubsets(Renewed. ~ ., data = data[, -c(1,2,3,4,5,12,13)], nvmax = 10)
reg.fss.summary <- summary(reg.fss)
reg.fss.summary
```

```
## Subset selection object
## Call: regsubsets.formula(Renewed. ~ ., data = data[, -c(1, 2, 3, 4,
##     5, 12, 13)], nvmax = 10)
## 5 Variables  (and intercept)
##                              Forced in Forced out
## Number.of.Slots                  FALSE      FALSE
## Price.Paid                       FALSE      FALSE
## Marketplace.Value.Delivered      FALSE      FALSE
## Applications                     FALSE      FALSE
## Term.Length                      FALSE      FALSE
## 1 subsets of each size up to 5
## Selection Algorithm: exhaustive
##          Number.of.Slots Price.Paid Marketplace.Value.Delivered
## 1  ( 1 ) " "             " "        " "
## 2  ( 1 ) "*"             " "        " "
## 3  ( 1 ) "*"             "*"        " "
## 4  ( 1 ) "*"             "*"        " "
## 5  ( 1 ) "*"             "*"        "*"
##          Applications Term.Length
## 1  ( 1 ) " "          "*"
## 2  ( 1 ) " "          "*"
## 3  ( 1 ) " "          "*"
```

```
## 4  ( 1 ) "*"          "*"
## 5  ( 1 ) "*"          "*"
```

```
reg.fss.summary$rsq
```

```
## [1] 0.04211637 0.05331556 0.06034504 0.06411965 0.06412359
```

Based on the above correlation plot and the forward subset selection calculation results, we can conclude that the predictor variables being checked here are correlating well with the response variable (albeit to varying degrees).

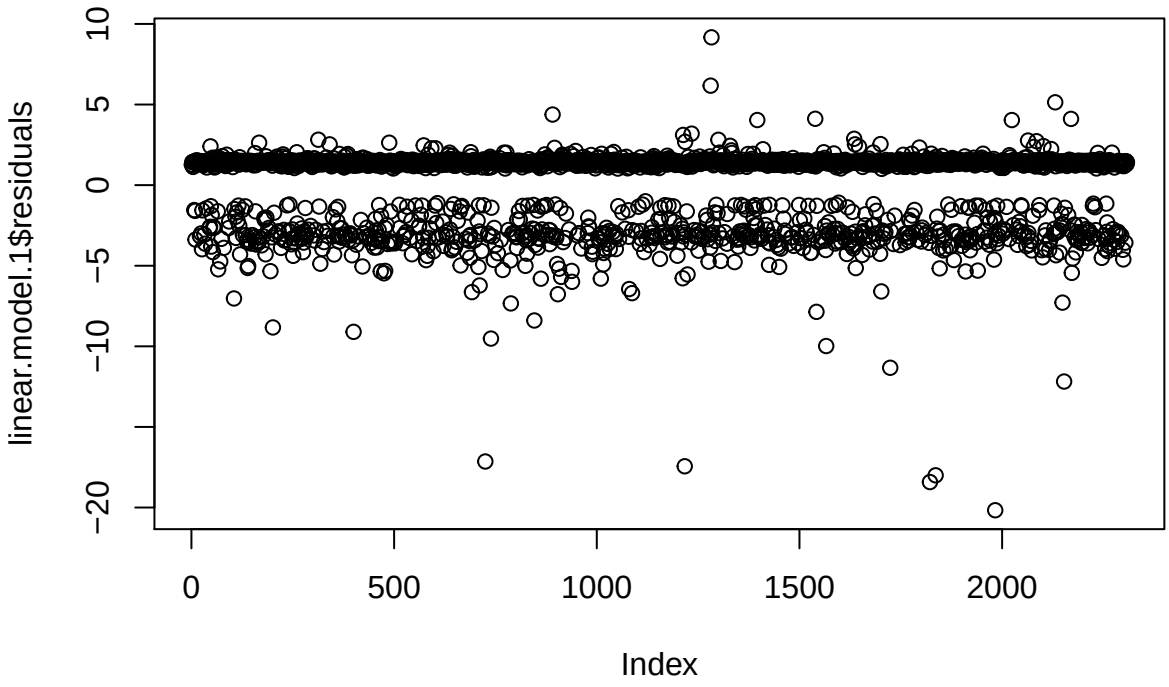
```
sample_size <- floor(NROW(data) * 0.75) # choosing 75% data for training, rest 25% for testing
set.seed(1)                             # setting a seed value so that the split is reproducible
training.indexes <- sample(seq_len(NROW(data)), size = sample_size)

training.data <- data[training.indexes, ] # training dataset
testing.data <- data[-training.indexes, ] # testing dataset, the two are mutually exclusive


linear.model.1 <- glm(Renewed. ~ Number.of.Slots + Price.Paid + Marketplace.Value.Delivered + Applicati
summary(linear.model.1)
```

```
##
## Call:
## glm(formula = Renewed. ~ Number.of.Slots + Price.Paid + Marketplace.Value.Delivered +
##     Applications + Term.Length, family = binomial, data = training.data)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.4512  -1.3418   0.7685   0.8651   2.1051
##
## Coefficients:
##                               Estimate Std. Error z value Pr(>|z|)
## (Intercept)                 -1.808e+00  2.754e-01  -6.565 5.22e-11 ***
## Number.of.Slots              2.127e-02  4.261e-03   4.992 5.97e-07 ***
## Price.Paid                  -4.562e-05  7.869e-06  -5.797 6.76e-09 ***
## Marketplace.Value.Delivered  6.257e-06  7.990e-06   0.783   0.4336
## Applications                 1.923e-04  1.099e-04   1.750   0.0802 .
## Term.Length                  7.059e-03  7.947e-04   8.883  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 2891.3  on 2307  degrees of freedom
## Residual deviance: 2723.2  on 2302  degrees of freedom
## AIC: 2735.2
##
## Number of Fisher Scoring iterations: 4
```

```
plot(linear.model.1$residuals)
```



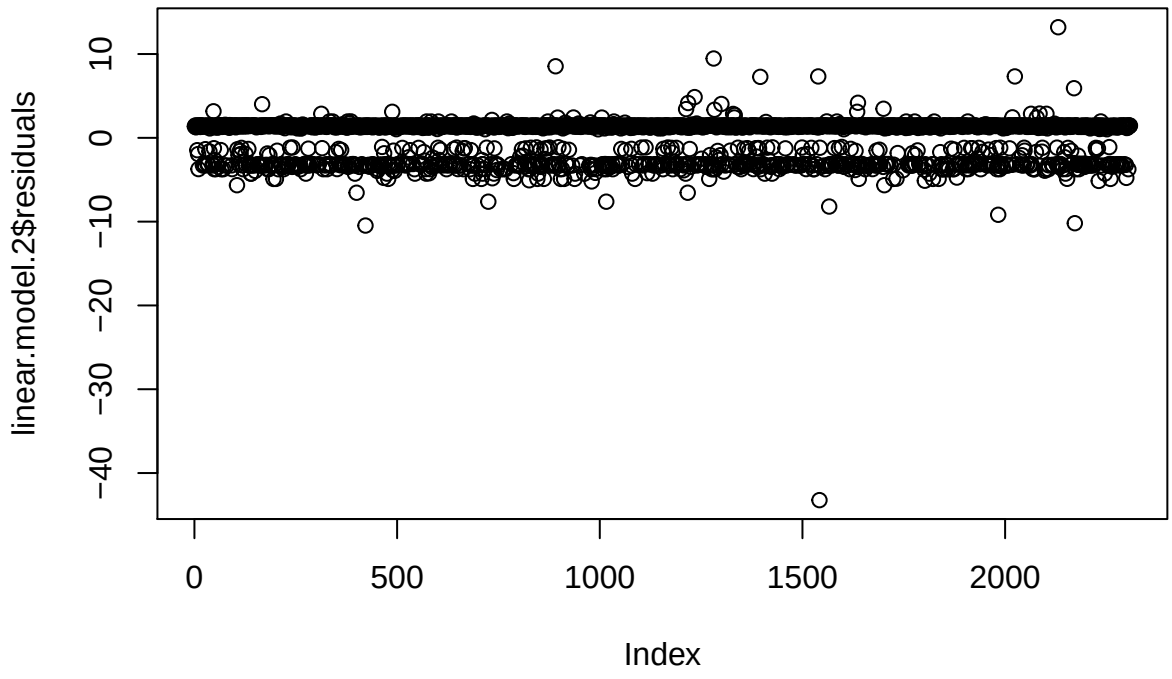
We choose the features with low p-values (***) for our subsequent model, designed to only include the ones where the null hypothesis can be rejected for an alpha value of 0.001!

```
linear.model.2 <- glm(Renewed. ~ Number.of.Slots * Term.Length, data = training.data, family = binomial)
summary(linear.model.2)
```

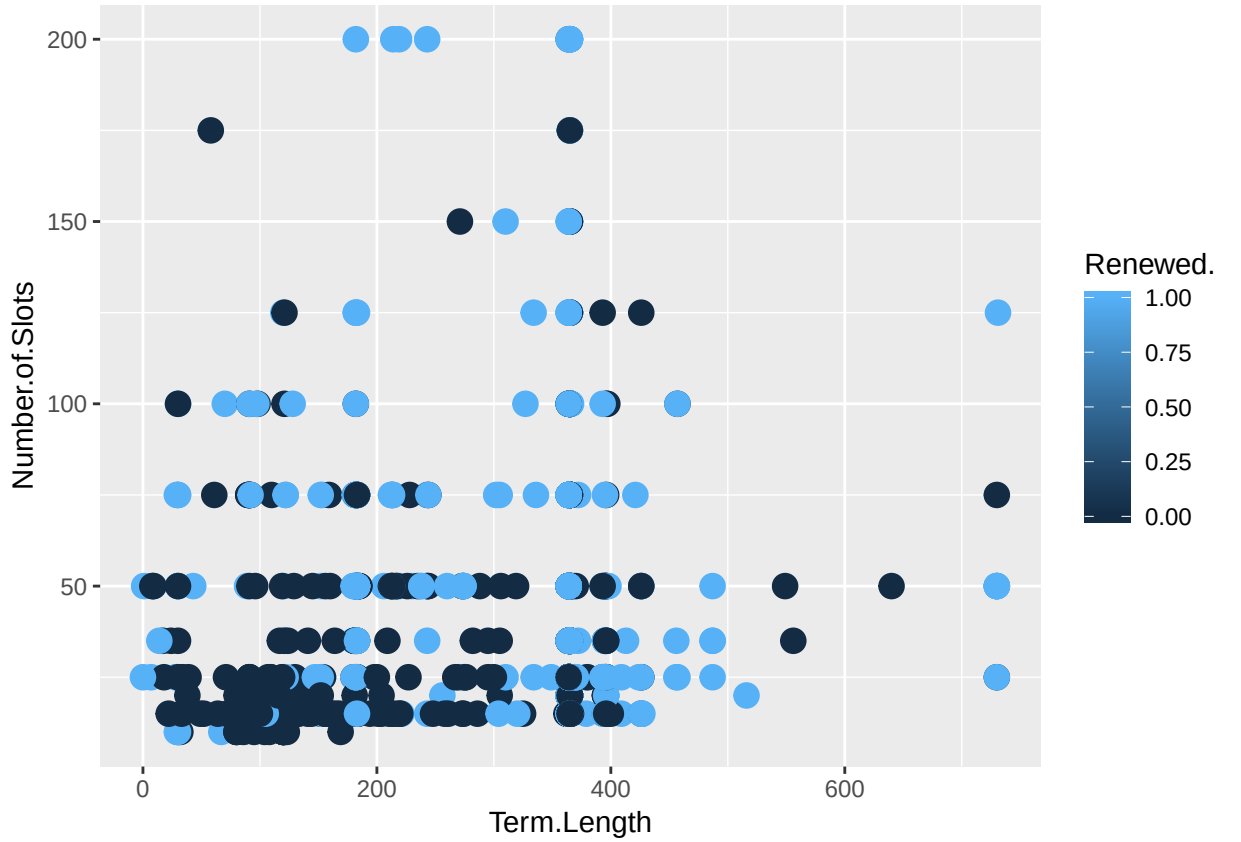
```
##
## Call:
## glm(formula = Renewed. ~ Number.of.Slots * Term.Length, family = binomial,
##     data = training.data)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.7447  -1.5097   0.7883   0.8524   2.2716
##
## Coefficients:
##                                Estimate Std. Error z value Pr(>|z|)
## (Intercept)                  -3.249e+00  4.402e-01  -7.381 1.57e-13 ***
## Number.of.Slots               4.691e-02  8.447e-03   5.554 2.79e-08 ***
## Term.Length                   1.070e-02  1.238e-03   8.644  < 2e-16 ***
## Number.of.Slots:Term.Length -1.092e-04  2.396e-05  -4.559 5.13e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 2891.3  on 2307  degrees of freedom
```

```
## Residual deviance: 2741.4  on 2304  degrees of freedom
## AIC: 2749.4
##
## Number of Fisher Scoring iterations: 4
```

```
plot(linear.model.2$residuals)
```




```
qplot(Term.Length, Number.of.Slots, data=data, colour=Renewed., size=I(4))
```

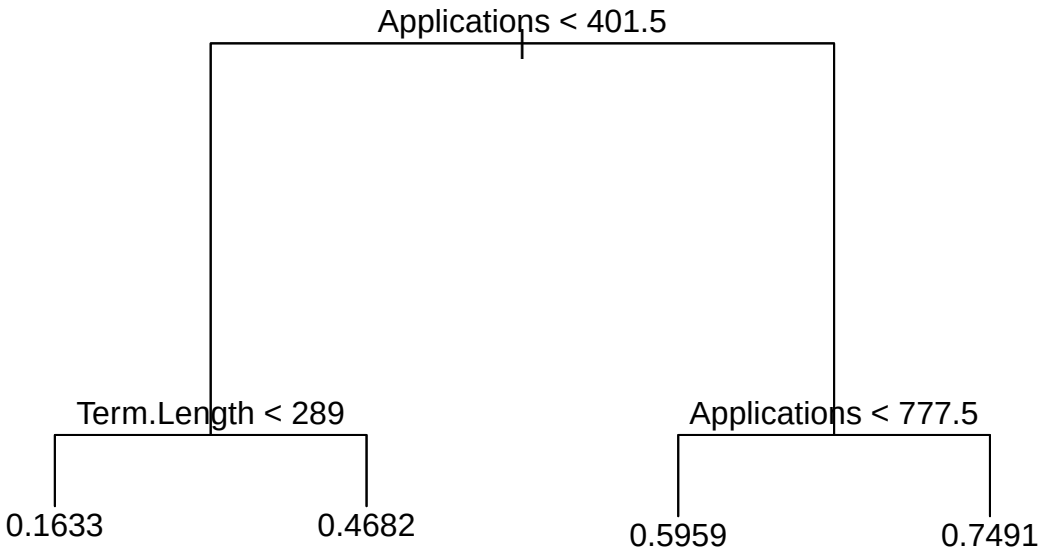


The plot does not display any evident pattern in the data based on our features used.

```
data$Renewed. <- as.factor(data$Renewed.)
tree.model <- tree(Renewed. ~ Number.of.Slots + Price.Paid + Marketplace.Value.Delivered + Applications
summary(tree.model)
```

```
##
## Regression tree:
## tree(formula = Renewed. ~ Number.of.Slots + Price.Paid + Marketplace.Value.Delivered +
##     Applications + Term.Length, data = training.data)
## Variables actually used in tree construction:
## [1] "Applications" "Term.Length"
## Number of terminal nodes:  4
## Residual mean deviance:  0.1984 = 457.2 / 2304
## Distribution of residuals:
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
## -0.7491 -0.4682  0.2509  0.0000  0.2509  0.8367
```

```
plot(tree.model)
text(tree.model)
```




```
predictions.lm.1 <- predict(linear.model.1, testing.data)
predictions.lm.2 <- predict(linear.model.2, testing.data)
predictions.tree <- predict(tree.model, testing.data)


# assessing raw classification accuracy

accuracy.lm.1 <- mean(testing.data$Renewed. == round(predictions.lm.1, 0))
accuracy.lm.1
```

```
## [1] 0.6038961
```

```
accuracy.lm.2 <- mean(testing.data$Renewed. == round(predictions.lm.2, 0))
accuracy.lm.2
```

```
## [1] 0.6493506
```

```
accuracy.tree <- mean(testing.data$Renewed. == round(predictions.tree, 0))
accuracy.tree
```

```
## [1] 0.6935065
```

```
# creating a confusion matrix for the tree model
table(testing.data$Renewed., round(predictions.tree, 0))
```

```
##
##       0   1
##   0  57 197
##   1  39 477
```

How well does your analysis in #3 predict retention? What other factors might you want to investigate to see if they could improve your analysis?

Based on the confusion matrix for the tree model (which is the better amongst the three we fitted) we developed, we can see that it has -

Raw accuracy is 69.3%
Sensitivity = TP/(TP + FN) = 57/(57 + 39) = 57/96 = 0.594 i.e. 59.4%
Specificity = TN/(TN + FP) = 477/(477 + 197) = 477/674 = 70.8%

So, we can conclude that our classification tree model does a better job of predicting whether a contract will be renewed, compared to whether the contract will not be renewed.
Also, we might want to figure out what impact factors like job industry, job roles being posted, the employer's attractiveness to the job applicants etc. have on the renewal of contracts that are being posted - see if there is some correlation between those variables!

Based on your analysis, what modifications would you recommend we make to our ad platform algorithm to improve retention?

Based on my analysis, I can say that the Term length of the contract and the Number of slots being offered play a huge role in determining whether a contract is renewed or not.
Glassdoor should try to sell more contracts that are one year (365 days in terms of duration) and focus on assessing which packages are renewed more (compare the number of slots being offered per package)!