STATS 140XP: PRACTICE OF STATISTICAL CONSULTING.
The Il-Lew-minating 7: Changmin Ahn, Chuhan Guo, Minghao Li, Brenda Ngo, Thi Nguyen, Gyujin Seo, MaiThu Tran.
University of California, Los Angeles

The Art of Job Hunting

Abstract In recent years, recruiting quality individuals has been highly competitive for most companies. Assume that our client is the HR department of a company that wants to analyze what makes a job post more attractive to job seekers. Our research question is how factors, from salary offered to the length of the job posting description, are associated with the number of views per job post. In other words, we want to identify what elements should be strengthened to attract more attention. After mutating our predictors, a Random Forest algorithm was run to determine the most important variables associated with the number of clicks on a job posting. A 10-fold cross-validation boosting algorithm was also run to confirm variable importance results. Results show that the duration of job posting being online, job description length, estimated salary, month that a job is posted, and number of reviews are the five most important variables associated with the number of clicks on a job posting.

Keywords Indeed.com, job posts, more applicants, Random Forest

1 Description of the Dataset

The dataset was attained from Indeed.com and it contains various features relating to 14,586,035 job posts pulled from 2016 and 2017. Features include number of reviews of the company that posted the job, creation date of a job post, the country and state of a job post, estimated salary, length of a job description, and many more.

2 Our Research Question

Throughout the exploration of our data, we focused on identifying which statistically significant factors are the most associated with the number of clicks on a job post on Indeed.com, a major job recruiting website.

3 Exploratory Data Analysis

Our initial exploration revealed that our original dataset of 14,586,035 observations and 22 predictors contained 52,129,051 missing values spanning across 14,519,331 observations. We thus removed all missing values and continued with the remaining 66,704 observations. Variables `companyID` and `jobID` are only observation identifiers, so they were removed.

3.1 Multicollinearity

According to the colored correlation map of the original predictors (*Figure 1*), predictors `descriptionWordCount` and `descriptionCharacterLength` are highly correlated with each other. Because `descriptionWordCount` can be highly skewed with shorter or longer words and `descriptionCharacterLength` is more indiscriminate than `descriptionWordCount`, `descriptionWordCount` was removed as a predictor. Likewise, `localClicks` and Clicks are highly correlated, but because `localClicks` was not a predictor of interest, it was also removed as a predictor.

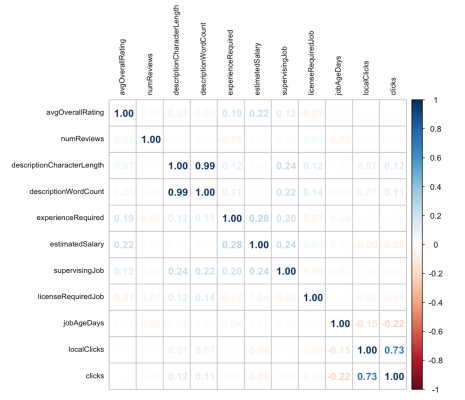

Fig.1. Colored correlation map to identify multicollinearity.

3.2 Distribution of the Response Variables

Because we were interested in how our predictors affect the number of clicks of a job post, we explored the distribution of our response variable clicks. After exploring the relationships between each predictor with our response variable, we found that there were many outliers in the distribution, which caused the distribution to be highly right-skewed (*Figure 2*). Thus, we removed all outliers defined as outside of the interquartile range (IQR) in order to balance the distribution of clicks, so it is less skewed (*Figure 3*). We continued with 60,389 observations, and all unused levels in every factor were dropped after.

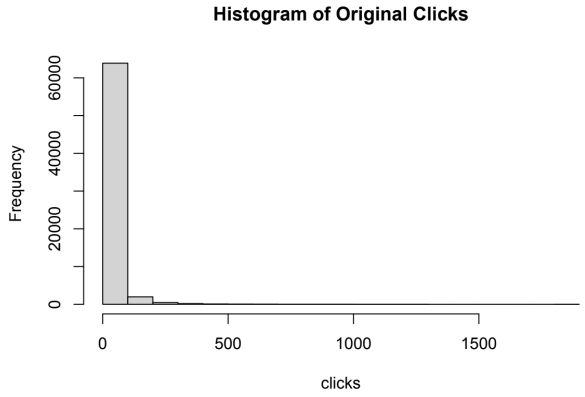

Fig.2. Histogram of `clicks` (original)

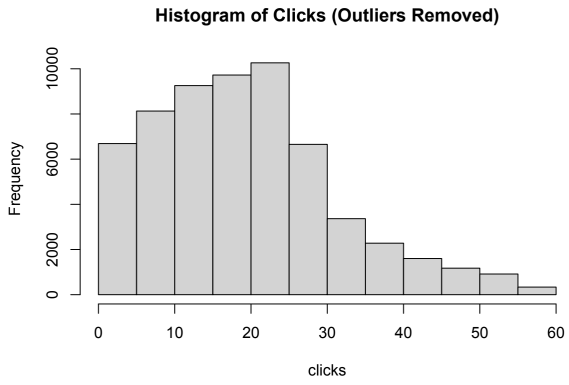

Fig.3. Histogram of `clicks` from a dataset with outliers removed.

3.3 Structure of Our Original Predictors

The table above summarizes the original set of predictors present in the dataset as well as our target variable (*Figure 4*). Included in the table is the variable type and how many levels are present (excluding unused levels) for categorical variables, and how many unique values the numeric variable contains.

variable	type	number.of.levels
date	categorical	395
country	categorical	2
stateProvince	categorical	59
city	categorical	1255
avgOverallRating	numeric	33
numReviews	numeric	288
industry	categorical	56
normTitle	categorical	1107
normTitleCategory	categorical	55
descriptionCharacterLength	numeric	2472
experienceRequired	numeric	28
estimatedSalary	numeric	820
salaryCurrency	categorical	2
jobLanguage	categorical	1
supervisingJob	numeric	2
licenseRequiredJob	numeric	2
educationRequirements	categorical	3
jobAgeDays	numeric	101
clicks	numeric	58

Fig.4. Variable Summary (Original).

Based on the table, `jobLanguage` only has one level, which has no significance in our analysis, so the predictor was removed. Moreover, categorical variables `date`, `stateProvince`, `city`, `normTitle`, `normTitleCategory`, and `industry` have too many levels for our algorithms to properly run, so we removed and mutated these variables.

3.4 Removal, Mutation, and Creation of Variables

We assumed that `normTitle`, `normTitleCategory`, and industry share the same characteristics, such as similar labor market demand, so we removed `normTitle` and industry and used

`normTitleCategory`. We reduced the number of levels in `normTitleCategory` by grouping values, based on the frequency at which they occur, into 4 levels: the minimum to the first quantile, the first quantile to the median, the median to the third quantile, and the third quantile to the maximum (*see Figure 5a for details of values*). We also removed `city` and used `stateProvince` for simplicity of variables. `stateProvince` was reduced to four levels using the same method for `normTitleCategory` (*see Figure 5b for details of values*). The original `normTitleCategory` and `stateProvince` variables were removed.

Due to the complexity of the `date` variable, we created three new variables— `month`, `year`, and `week_day` —from this variable, and the `date` variable was removed. A new summary of all variables, both predictors and response, can be seen in *Figure 6*.

variable	type	number.of.levels
country	categorical	2
avgOverallRating	numeric	33
numReviews	numeric	288
descriptionCharacterLength	numeric	2472
experienceRequired	numeric	28
estimatedSalary	numeric	820
salaryCurrency	categorical	2
supervisingJob	numeric	2
licenseRequiredJob	numeric	2
educationRequirements	categorical	3
jobAgeDays	numeric	101
clicks	numeric	58
new_normTitleCategory	categorical	4
new_state_group	categorical	4
month	categorical	12
year	categorical	2
week_day	categorical	7

Fig.6. Variable Summary (Final)

4 Description of Variables

Table 1. A List of our 16 Predictors

Predictor	Description
`Country`	The country of the job post: US or CA
`numReviews`	Total number of reviews a company received
`avgOverallRating`	Average overall rating of the company (1 - 5 stars)
`descriptionCharacterLength`	Total number of characters in description of job post
`experienceRequired`	Minimum number of years of experience required for job (in years)
`estimatedSalary`	Estimated annual salary
`salaryCurrency`	Type of salary currency: USD or CAD
`supervisingJob`	Whether this job is classified as supervising job: 0 or 1
`licenseRequiredJob`	Whether this job is classified as requiring a license: 0 or 1
`educationRequirements`	The job's level of education required: None, High School, or Higher Education
`jobAgeDays`	Age of job in days, based on job post creation date
`new_normTitleCategory`	Job title category grouped based on frequency of occurrence: 1, 2, 3, or 4
`new_state_group`	State grouped based on frequency of occurrence: 1, 2, 3, or 4
`month`	Month that job post was created
`year`	Year that job post was created
`week_day`	Week day that job post was created
Response	**Description**
`clicks`	The total number of clicks on the date

5 Statistical Analysis

5.1 Methods

A Random Forest algorithm was used to find the most important variables associated with `clicks`. The Random Forest algorithm is a supervised regression algorithm that builds many decision trees based on different random samples of the data and takes the average for the response variable of all the decision trees produced (*see figure 7 for a simplified explanation*). A

Freq	
Min.	9
1st Qu.	160
Median	439
Mean	1098
3rd Qu.	1061
Max.	8605

(a)

Freq	
Min.	15
1st Qu.	222
Median	689
Mean	1024
3rd Qu.	1346
Max.	5815

(b)

Fig.5. 5-number summaries (a) `normTitleCategory` (b) `stateProvince`

10-fold cross-validation boosting algorithm was then used to confirm the variable importance found by our Random Forest algorithm. Lastly, an ANOVA partial F-test and an individual t-test was run on the top 10 most important predictors in order to determine their statistical significance associated with our response variable.

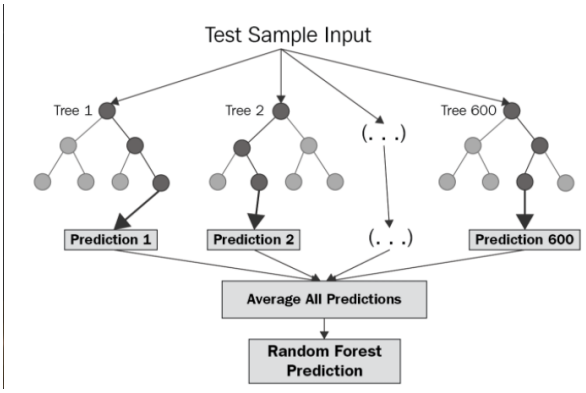

Fig.7. Random Forest algorithm (simplified) [1]

5.2 Random Forest

Our Random Forest model determined that our top five most important variables in determining clicks are `jobAgeDays`, `month`, `estimatedSalary`, `numReviews`, and `descriptionCharacterLength`, in order from highest to lowest percentage increase in mean squared error (%IncMSE). %IncMSE shows how much the model accuracy decreases if the variable is left out of the model, meaning, the higher the score, the more important the variable is (*Figure 8*).

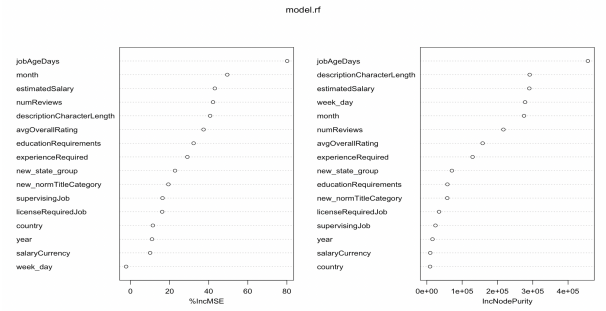

Fig.8. Random Forest variable importance plot

5.3 10-Fold Cross-Validation Boosting

Our 10-fold cross-validation performance boosting model also determined that our top five most important variables in determining clicks are `jobAgeDays`, `descriptionCharacterLength`, `estimatedSalary`, `month`, and `numReviews`, in order from highest to lowest relative influence (*Figure 9*). Respectively, relative influence values are 19.42%, 15.37%, 14.75%, 13.78%, and 10.55%. Due to seed dependence, the order of variable importance determined by the Random Forest algorithm and 10-fold cross-validation performance boosting can change. However, all top five most important variables are the same, so we can confirm that `jobAgeDays`, `descriptionCharacterLength`, `estimatedSalary`, `month`, and `numReviews` are our five most important variables associated with `clicks`,

where `jobAgeDays` is the most important variable.

	var	rel.inf
jobAgeDays	jobAgeDays	19.4165534
descriptionCharacterLength	descriptionCharacterLength	15.3658988
estimatedSalary	estimatedSalary	14.7510382
month	month	13.7767315
numReviews	numReviews	10.5502837
avgOverallRating	avgOverallRating	5.7713723
experienceRequired	experienceRequired	5.4754179
educationRequirements	educationRequirements	4.4950001
week_day	week_day	3.8796110
new_normTitleCategory	new_normTitleCategory	2.5618063
new_state_group	new_state_group	1.7055953
licenseRequiredJob	licenseRequiredJob	1.0003304
supervisingJob	supervisingJob	0.6047343
country	country	0.4018179
year	year	0.2438089
salaryCurrency	salaryCurrency	0.0000000

Fig.9. Variables' relative influence based on 10-fold CV Boosting

5.4 Statistical Significance Analysis

Comparing the full model with all predictors and the reduced model with only the top ten most important predictors, our ANOVA confirms a p-value that is less than significance level 0.001. An ANOVA on the reduced model determines that all top ten predictors are significant at a significance level of at least 0.05 (*Figure 10*).

Analysis of Variance Table

Response: clicks

	Df	Sum Sq	Mean Sq	F value	Pr(>F)	
avgOverallRating	1	602	602	4.5182	0.0335394	*
numReviews	1	651	651	4.8883	0.0270429	*
descriptionCharacterLength	1	65919	65919	494.7920	< 2.2e-16	***
experienceRequired	1	1498	1498	11.2448	0.0007989	***
estimatedSalary	1	79418	79418	596.1173	< 2.2e-16	***
licenseRequiredJob	1	26996	26996	202.6362	< 2.2e-16	***
educationRequirements	2	55254	27627	207.3712	< 2.2e-16	***
jobAgeDays	1	383383	383383	2877.7063	< 2.2e-16	***
new_normTitleCategory	3	9243	3081	23.1259	5.915e-15	***
month	11	12652	1150	8.6336	1.814e-15	***
Residuals	60365	8042132	133			

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Fig.10. ANOVA Partial sifnificance F-test on top of 10 most important predictors associated with `clicks`

5.5 Plots of Top Five Predictors

5.5.1 `JobAgeDays` *vs.* `Clicks`

Based on the plot of the relationship between `jobAgeDays` and clicks, the less time a job post is active, the higher the number of clicks the job post receives (*Figure 11*).

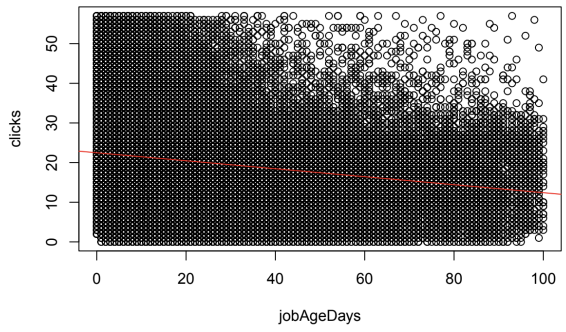

Fig.11. Relationship between `jobAgeDays` and `clicks`

5.5.2 `DecriptionCharacterLength` *versus* `Clicks`

Our plot on the relationship between `descriptionCharacterLength` and `clicks` revealed that the longer the description of a job post is, the higher the number of clicks a job post receives (*Figure 12*).

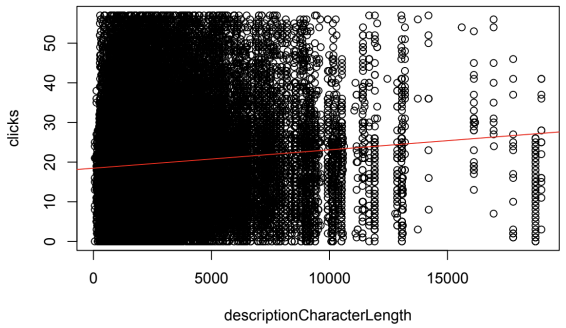

Fig.12. Relationship between `descriptionCharacterLength` and `clicks`

5.5.3 `Month` *versus* `Clicks`

Based on the plot of the relationship between `month` and `clicks`, it seems that there is no statistically significant difference between each month. However, the individual t-test of the top ten predictors revealed that month03, or March, is the most statistically significant of all months with a p-value less than a significance level of 0.001, followed by February, July, and September, in no particular order, with p-values less than a significance level of 0.01, and then followed by April and August, in no particular order, with p-values

less than a significance level of 0.05. The boxplot revealed that March has the lowest median of clicks and September has the highest median of clicks (*Figure 13 & 14*). Despite removing outliers based on the distribution of clicks, outliers are still present when observing the relationship between `months` and `clicks`.

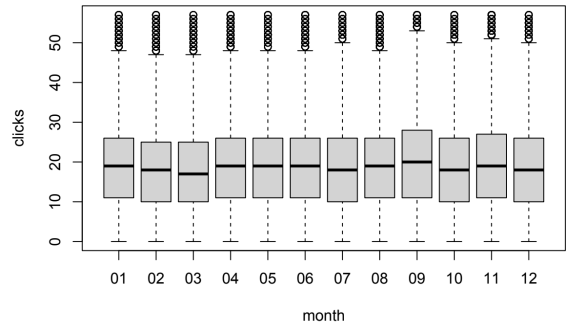

Fig.13. Relationship between `months` and `clicks`

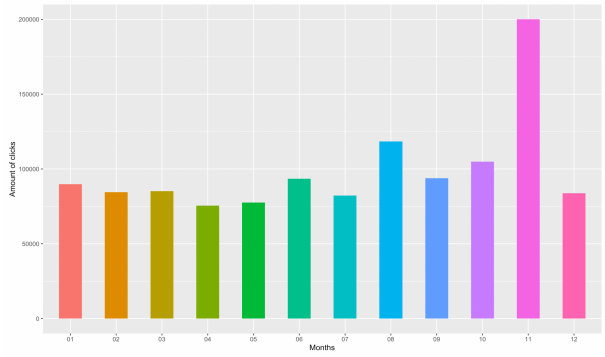

Fig.14. Relationship between `month` and median number of clicks

5.5.4 *`EstimatedSalary` versus `Clicks`*

The plot on the relationship between `estimatedSalary` and `clicks` revealed that lower estimated salary is associated with higher number of clicks (*Figure 15*). However, because outliers seem to be present, this relationship may be skewed.

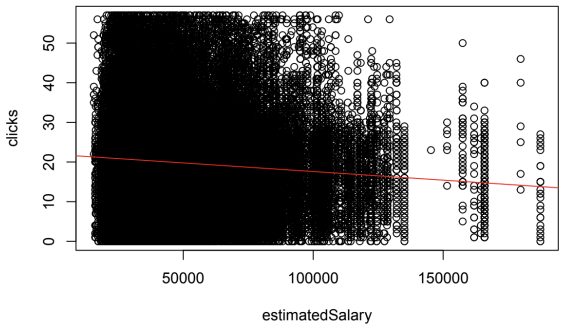

Fig.15. Relationship between `estimatedSalary` and `clicks`

5.5.5 *`NumReviews` versus `Clicks`*

The plot of the relationship between `numReviews` and `clicks` revealed there is a slight association between less number of reviews and higher number of clicks (*Figure 16*). However, outliers are present, so the relationship may be highly skewed.

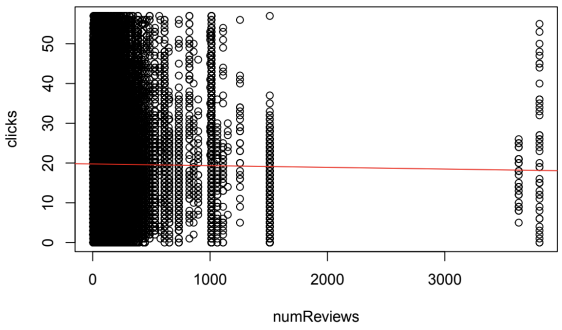

Fig.16. Relationship between `numReviews` and `clicks`

6 Summary of Results

Our data analysis demonstrated that `jobAgeDays`, `month`, `estimatedSalary`, `numReviews`, and `descriptionCharacterLength`, in no particular order except `jobAgeDays` being the most important, are the top five most important variables associated with the number of clicks a job post received on Indeed.com between 2016 and 2017. More specifically, the less time a job post is active online, the higher the number of clicks a job post received. The longer the job description was, the higher the number of clicks a

job post received. Jobs posted in March were revealed to have the least average number of clicks whereas jobs posted in September received the most average number of clicks. Moreover, our analysis revealed that lower estimated salary was associated with higher number of clicks, with possible skewness. Lastly, the less reviews the company of a job post receives, the higher the number of clicks the job post receives, with possible skewness.

7 Overall Conclusions

From our study, we determined the top ten most important predictors associated with the number of clicks that a job post from between 2016 to 2017 received with focus on the exploration of the top five. All top ten predictors were found to be statistically significant. We also confirmed that the most important predictor of all our predictors associated with number of clicks on a job post was `jobAgeDays`, or the duration (in days) that a job post is active on Indeed.com.

Based on our analysis, several recommendations for job recruiters are as follows:

(1) Jobs posted in September received the most average number of clicks. Fall tends to be recruiting season for many of fields, so other companies can also take advantage of this time period to post jobs positions and receive a higher number of applicants.

(2) Job posts with longer descriptions receive more clicks. We recommend employers ensure that their job posts are detailed and well-written in order to attract more applicants.

(3) On Indeed.com, newer job posts tend to receive more clicks, therefore if a job listing has been up for a long time, companies should re-post that job to attract more applicants.

By focusing on optimizing these various features when creating job posts on Indeed.com, job recruiters can save time and money while attracting more potential job candidates.

8 Shortcomings and Challenges

Some challenges we faced when conducting our study was that our original data set consisted of millions of missing values, so we were forced to reduce our data set from millions of observations to only several ten thousands. Furthermore, the distribution of our response variable was highly right-skewed, which would cause issues with our data analysis and finding statistical significance. Likewise, our original categorical variables contained too many levels for our algorithms to properly run, so we had to implement different methods to reduce the number of levels as well as remove many original predictors. Lastly, we also faced a large number of outliers throughout our data analysis, which has led to skewness in our relationships of predictors with our response variable. As mentioned above, we faced outliers throughout our data analysis, up to plotting the relationships between our top five predictors with our response variable, so we possibly face a skewed association between some of our predictors and response variable, which may not reflect their true relationships. As time did not permit, we were not able to process the remaining outliers, and removing the remaining outliers may reduce our data greatly and affect our analysis.

9 Recommendations for the Future

We recommend that imputations of the missing values are made in order to have a larger sample of the population and hopefully run a data analysis that reflects the true population and relationships between the most important predictors and the response variable. We recommend that with our model found using the most important predictors associated with our response vari-

able, we try to predict a new set of data to determine the accuracy of our model.

10 Appendix

Our team's R script containing the code for this project: The Il-Lew-minating 7 Google Drive

For the appendix, see the following pages below.

References

[1] Chaya. (2020). *Random Forest Regression Diagram.* velup.gitconnected.com. Retrieved 2022, from https://levelup.gitconnected.com/random-forest-regression-209c0f354c84.

STATS140XP Final Project Report Appendix

The Il-Lew-minating Seven

2022-11-29

```
# exploratory data analysis
datafest <- read.csv("datafest2018-Updated-April12.csv", stringsAsFactors = TRUE,
                     na.strings=c("","NA")
)

## missing values ##
sum(is.na(datafest))

# remove missing values
datafest.na <- na.omit(datafest)

# check size and predictors
dim(datafest) # pre-removal - 14586035 obs, 23 variables
dim(datafest.na) # post-removal - 66704 obs, 23 variables
str(datafest.na)

head(datafest.na) # first 6 obs of data

# remove companyId, jobId
datafest.na <- datafest.na[, -which(colnames(datafest.na) %in%
                                      c("companyId", "jobId"))]
```

```
# correlation plot
library(corrplot)

X <- model.matrix(~ avgOverallRating + numReviews + descriptionCharacterLength
                  + descriptionWordCount + experienceRequired+estimatedSalary
                  + supervisingJob + licenseRequiredJob + jobAgeDays
                  + localClicks + clicks - 1,
                  data = datafest.na)
contrast_corr <- cor(X)
corrplot(contrast_corr,type = "full", addgrid.col = "gray", tl.col = "black",
         tl.srt = 90, method = "number", tl.cex = 0.733)

# remove descriptionWordCount and localClicks based on high multicollinearity
datafest.na <- datafest.na[, - which(colnames(datafest.na) %in%
                                       c("descriptionWordCount", "localClicks"))]
```

```
## relationship between predictors and clicks
plot(clicks ~ date, data = datafest.na)
plot(clicks ~ country, data = datafest.na)
plot(clicks ~ stateProvince, data = datafest.na)
plot(clicks ~ city, data = datafest.na)
```

```
plot(clicks ~ avgOverallRating, data = datafest.na)
plot(clicks ~ numReviews, data = datafest.na)
plot(clicks ~ industry, data = datafest.na)
plot(clicks ~ normTitle, data = datafest.na)
plot(clicks ~ normTitleCategory, data = datafest.na)
plot(clicks ~ descriptionCharacterLength, data = datafest.na)
plot(clicks ~ experienceRequired, data = datafest.na)
plot(clicks ~ estimatedSalary, data = datafest.na)
plot(clicks ~ salaryCurrency, data = datafest.na)
plot(clicks ~ jobLanguage, data = datafest.na)
plot(clicks ~ supervisingJob, data = datafest.na)
plot(clicks ~ licenseRequiredJob, data = datafest.na)
plot(clicks ~ educationRequirements, data = datafest.na)
plot(clicks ~ jobAgeDays, data = datafest.na)
```

```
## investigate distribution of clicks
hist(datafest.na$clicks,
     main = "Histogram of Original Clicks",
     xlab = "clicks")

library(dplyr)
summary(datafest.na$clicks)
upper <- 12-1.5*(30-12)
lower <- 30+1.5*(30-12)
datafest.new <- datafest.na %>%
  filter(clicks<=57) %>%
  filter(clicks>=0)

hist(datafest.new$clicks,
     main = "Histogram of Clicks (Outliers Removed)",
     xlab = "clicks")

# drop unused levels
for (i in seq_len(ncol(datafest.new))){
  if (is.factor(datafest.new[, i])) {
    datafest.new[, i] <- droplevels(datafest.new[, i])
  }
}
```

```
# variable summary - name, type, number of levels for categorical
# (without removed variables)
variable_names <- colnames(datafest.new)
variable_type <- rep(NA, ncol(datafest.new))
num_levels <- rep(NA, ncol(datafest.new))
for (i in seq_len(ncol(datafest.new))){
  if (is.numeric(datafest.new[, i]) | is.integer(datafest.new[, i])){
    variable_type[i] <- "numeric"
    num_levels[i] <- length(unique(datafest.new[, i]))
  } else {
    variable_type[i] <- "categorical"
    num_levels[i] <- length(levels(datafest.new[, i]))
  }
}
variable_summary <- data.frame(
```

```
  variable = variable_names,
  type = variable_type,
  "number of levels" = num_levels
)
knitr::kable(variable_summary, "simple")

# remove jobLanguage (only 1 factor - models won't work with predictors with
# only 1 level)
datafest.new <- datafest.new[, - which(colnames(datafest.new) == "jobLanguage")]
```

```
library(tidyverse)
library(stringr)
# remove city, normTitle, industry
datafest.new <- datafest.new[, - which(colnames(datafest.new) %in%
                                         c("city", "normTitle", "industry"))]

## Group normTitleCategory based on the frequencies variables to
## create new normTitle variable
normTitle_table <- as.data.frame(table(datafest.new$normTitleCategory))
summary(normTitle_table)
normTitle_table$Var1 <- as.character(normTitle_table$Var1)

indice1 <- which(normTitle_table$Freq<=160)
normtitle_level1 <- c()
for(i in 1:length(indice1)){
  normtitle_level1[i] <- normTitle_table[indice1[i],1]
}

indice2 <- which(normTitle_table$Freq>=160 & normTitle_table$Freq <= 439)
normtitle_level2 <- c()
for(i in 1:length(indice2)){
  normtitle_level2[i] <- normTitle_table[indice2[i],1]
}
indice3 <- which(normTitle_table$Freq>=439 & normTitle_table$Freq <= 1061)
normtitle_level3 <- c()
for(i in 1:length(indice3)){
  normtitle_level3[i] <- normTitle_table[indice3[i],1]
}
indice4 <- which(normTitle_table$Freq>=1061)
normtitle_level4 <- c()
for(i in 1:length(indice4)){
  normtitle_level4[i] <- normTitle_table[indice4[i],1]
}

datafest.new <- datafest.new%>%
  mutate(new_normTitleCategory = as.factor(case_when(normTitleCategory %in% normtitle_level1 ~ 1,
                                                     normTitleCategory %in% normtitle_level2 ~ 2,
                                                     normTitleCategory %in% normtitle_level3 ~ 3,
                                                     normTitleCategory%in% normtitle_level4 ~ 4)))

## Group stateProvince based on the frequencies variables to create new
## new_state_group variable
state_table <- as.data.frame(table(datafest.new$stateProvince))
summary(state_table)
```

```
state_table$Var1 <- as.character(state_table$Var1)

indice1 <- which(state_table$Freq <= 222)
state_level1 <- c()
for(i in 1:length(indice1)){
  state_level1[i] <- state_table[indice1[i],1]
}

indice2 <- which(state_table$Freq>=222 & state_table$Freq <= 689)
state_level2 <- c()
for(i in 1:length(indice2)){
  state_level2[i] <- state_table[indice2[i],1]
}
indice3 <- which(state_table$Freq>=689 & state_table$Freq <= 1346)
state_level3 <- c()
for(i in 1:length(indice3)){
  state_level3[i] <- state_table[indice3[i],1]
}
indice4 <- which(state_table$Freq>=1346)
state_level4 <- c()
for(i in 1:length(indice4)){
  state_level4[i] <- state_table[indice4[i],1]
}

datafest.new <- datafest.new %>%
  mutate(new_state_group = as.factor(case_when(stateProvince %in% state_level1 ~ 1,
                                               stateProvince%in% state_level2 ~ 2,
                                               stateProvince%in% state_level3 ~ 3,
                                               stateProvince  %in% state_level4 ~ 4)))

#remove the old normTitleCategory and stateProvince
datafest.new <- datafest.new[,-which(colnames(datafest.new) %in%
                                       c("normTitleCategory", "stateProvince"))]
```

```
## new variables
# Create a new variable month based on variable date
patter.month <- "\\b\\d{2}\\b"
datafest.new$month <- as.factor(str_match(datafest.new$date, patter.month))

# Create a new variable year based on variable date
patter.year <- "\\d{4}"
datafest.new$year <- as.factor(str_match(datafest.new$date, patter.year))

# Create a new variable week days based on variable date
datafest.new$week_day <- as.factor(weekdays(as.Date(datafest.new$date)))

# drop the old date variable
datafest.new <- datafest.new[,-which(colnames(datafest.new) == "date")]
```

```
# variable summary - name, type, number of levels for categorical
# without removed variables)
variable_names <- colnames(datafest.new)
variable_type <- rep(NA, ncol(datafest.new))
num_levels <- rep(NA, ncol(datafest.new))
```

```r
for (i in seq_len(ncol(datafest.new))){
  if (is.numeric(datafest.new[, i]) | is.integer(datafest.new[, i])){
    variable_type[i] <- "numeric"
    num_levels[i] <- length(unique(datafest.new[, i]))
  } else {
    variable_type[i] <- "categorical"
    num_levels[i] <- length(levels(datafest.new[, i]))
  }
}
variable_summary <- data.frame(
  variable = variable_names,
  type = variable_type,
  "number of levels" = num_levels
)
knitr::kable(variable_summary, "simple")
```

```r
#### Random forest ####
set.seed(1)
library(randomForest)

test.i <- sample(1:66704,46693, replace = F)
train.df <- datafest.new[-test.i,]
test.df <- datafest.new[test.i,]

model.rf <-randomForest(clicks~.,data = train.df, ntree = 100,importance = T,
                        na.action = na.omit, mtry = 6) # 100 trees and split by 6
model.rf
varImpPlot(model.rf)
round(importance(model.rf), 2)
```

```r
#### 10-fold Cross-validation boosting ####
library(gbm)
set.seed (1)
boost.datafest <-gbm(clicks~.,data = train.df , distribution = "gaussian",
                     n.trees = 500, interaction.depth = 4, cv.folds = 10)
summary(boost.datafest)
```

```r
### Using full dataset for the all steps below

lm.model.full <- lm(clicks~., data = datafest.new)# Fit the full model

## Top 10 important variables from randomforest model + clicks
inp_var_i <- names(datafest.new)%in% c("clicks","descriptionCharacterLength",
                                       "jobAgeDays", "estimatedSalary", "month",
                                       "educationRequirements","new_normTitleCategory",
                                       "avgOverallRating", "licenseRequiredJob",
                                       "experienceRequired", "numReviews")

datafest.final <- datafest.new[, inp_var_i]

## Fit top 10 important variables using lm()
lm.model.reduced <- lm(clicks ~ ., data = datafest.final)
summary(lm.model.reduced)
anova(lm.model.reduced)
```

```r
# using ANOVA to test for 2 models (partial f-test)
ano.model <- anova(lm.model.reduced, lm.model.full)
ano.model ## since p_value is small, we favor the reduced model.
```

```r
## plots for top five predictors##
plot(clicks ~ jobAgeDays, data = datafest.new)
abline(lm(clicks ~ jobAgeDays, data = datafest.new), col = "red")

plot(clicks ~ descriptionCharacterLength, data = datafest.new)
abline(lm(clicks ~ descriptionCharacterLength, data = datafest.new), col = "red")

plot(clicks ~ month, data = datafest.new)

plot(clicks ~ estimatedSalary, data = datafest.new)
abline(lm(clicks ~ estimatedSalary, data = datafest.new), col = "red")

plot(clicks ~ numReviews, data = datafest.new)
abline(lm(clicks ~ numReviews, data = datafest.new), col = "red")
```