Untitled

Irina Klein

11/26/2021

Category

Perhaps we should remove the observations for which the category of the issue is not one of the frequent ones.

```
setwd("~/CSP571ProjectGroup")
df <- read_csv("df.csv")
```

```
## Rows: 114660 Columns: 15
```

```
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (9): number, incident_state, sys_updated_by, contact_type, category, sub...
## dbl (5): reassignment_count, reopen_count, sys_mod_count, problem_id, resolv...
## lgl (1): made_sla
```

```
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
## category
#df  %>% group_by(category) %>% summarise(n = n(), share = round(n()/114660,3)) %>% arrange(n)

## subcategory
#n <- df  %>% group_by(subcategory) %>% summarise(n = n(), share = round(n()/114660,3)) %>% arrange(n)

## both
(keepnum <- df  %>% group_by(category, subcategory) %>% summarise(n = n(), share = round(n()/114660,3))
```

```
## `summarise()` has grouped output by 'category'. You can override using the `.groups` argument.
```

```
## # A tibble: 1 x 1
##       n
##   <int>
## 1 86902
```

```
 keepnum/nrow(df)
```

```
##           n
## 1 0.7579103
```

```
df_75 <- df %>%
group_by(category, subcategory) %>%
mutate(m = n()/114660) %>%
ungroup() %>%
filter(m > 0.005)
```

If we remove all the observations for which category&subcategory combinations appear in less than in 0.5% observations, we end up with 76% of observations.

Tree methods

```
# test run on the first 1000 rows
df_1000 <- df_75[1:1000,]

#as.factor
cols <- c("made_sla", "category", "incident_state", "contact_type", "subcategory", "urgency", "impact",
df_1000[cols] <- lapply(df_1000[cols], factor)

#tree
tree.incs <- tree(resolved_updated_diff ~ . -number-m-sys_updated_by, df_1000)
summary(tree.incs)
```

```
##
## Regression tree:
## tree(formula = resolved_updated_diff ~ . - number - m - sys_updated_by,
##     data = df_1000)
## Variables actually used in tree construction:
## [1] "subcategory"        "incident_state"     "category"
## [4] "priority"           "reassignment_count" "sys_mod_count"
## Number of terminal nodes:  12
## Residual mean deviance:  57310000 = 5.662e+10 / 988
## Distribution of residuals:
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##  -20480   -4270   -1545       0    1420   33560
```

```
tree.incs
```

```
## node), split, n, deviance, yval
##       * denotes terminal node
##
##   1) root 1000 1.127e+11  6744.0
##     2) subcategory: Subcategory 103,Subcategory 114,Subcategory 125,Subcategory 135,Subcategory 16,Su
##       4) incident_state: Resolved 133 1.497e+09   545.8 *
##       5) incident_state: Active,Awaiting Problem,Awaiting User Info,New 705 4.373e+10  5542.0
##        10) category: Category 20,Category 26,Category 32,Category 40,Category 53,Category 61,Categor
##        11) category: Category 23,Category 24,Category 37,Category 42 225 2.304e+10  8230.0 *
##     3) subcategory: Subcategory 123,Subcategory 154,Subcategory 200,Subcategory 231,Subcategory 28,Su
##       6) priority: 2 - High,3 - Moderate,4 - Low 157 3.205e+10 15530.0
##        12) incident_state: Awaiting User Info,Resolved 60 4.453e+09  6607.0
##          24) incident_state: Awaiting User Info 38 2.937e+09 10430.0 *
##          25) incident_state: Resolved 22 0.000e+00     0.0 *
##        13) incident_state: Active,Awaiting Problem,New 97 1.987e+10 21050.0
##          26) reassignment_count < 1.5 59 1.163e+10 17880.0
##            52) category: Category 23,Category 37,Category 45 50 9.695e+09 15480.0
##             104) incident_state: Active 17 8.214e+08  5771.0 *
##             105) incident_state: Awaiting Problem,New 33 6.446e+09 20480.0 *
##            53) category: Category 34,Category 35 9 5.105e+07 31210.0 *
##          27) reassignment_count > 1.5 38 6.718e+09 25970.0
##            54) sys_mod_count < 20.5 33 4.617e+09 28440.0
##             108) category: Category 23,Category 34,Category 37,Category 46 18 1.484e+09 21750.0 *
##             109) category: Category 45 15 1.362e+09 36470.0 *
##            55) sys_mod_count > 20.5 5 5.747e+08  9690.0 *
##       7) priority: 1 - Critical 5 1.039e+08 65110.0 *
```

The deviance is very large, meaning that the predictions would not be accurate. We will classify all the resolutions time to a specified set of buckets (<2h, <1day, ect.) and build a classification tree.