ML1000CourseProjectRtlGrp

Ignacio Palma, Jairo Melo and Vikram Khade.

2019-03-13

Information Technology Service Management Analysis

ITSM is an area of continues improvement and for major organizations every opportunity could represent major cost savings which translate into more affortable products for patiences and parents.

The file extracted from the ITSM system contains 1.2 year worth data for two major product lines.

Loading data

You can include R code in the document as follows:

```
## [1] "/Users/jairomelo/Desktop/ML/YORK/CourseProject"
```

Cleaning the data

We will take care of duplicated, records with NA values, removing tickets that are not Resolved, as well as undertermine records, for example: Tickets with Support Level outside of the standards.

```
## [1] 21750

## [1] 21294

## [1] 21291
```

Data Understanding

- incident: Number of the ticket incident. Not a significant variable as is sequencial counter.
- application: Number of the application of the reported issue. This is a relevant variable which a certantly number of tickets are assigned to one application.
- region: Region where the user is located. Significant as a region is associated to a particular population of users reporting issues of an application.
- prod_line: Product Line is a group of related products under the same brand. For example, Web and Ecommerce, and also Internal Business process applications.
- opened: Date when the issues was opened. The ticket has 5 stages: Not Assigned, In Progress, Customer Action, Pending, Resolved, Closed. Not Assigned: The ticket was created/open, but still not been worked by the support team. In Progress: The ticket is assigned to a support group who is actively working on it. Customer Action: The ticket

goes into a stand-by because additional information is requested from the user before the current support group can continue working. Pending: The ticket goes into a stand-by because there is an activity to be performed by a third party group before the current support group can continue working. Resolved: Once the issue is fixed, the user is notified by the Support team. Closed: Each resolved ticket moves into Closed after the user confirms, or automatically, the ticket is closed after n number of days. For our analysis, we will using only tickets that are Resolved. Closed might not be relevant as there is a strong correlation between Closed and Resolved.

- app_category: Category of the Application. Relevant as this is the classification of the application.
- priority: Priority of the Issue. This is the result of Urgency and Impact.
  Low Urgency - Limited Impact = Lower Priority. -> 4 High Urgency - Limite Impact = High Priority. -> 1 The "Priority" word can be removed from the field and use the numbers 1,2,3,4. Priority 4 is low, and 1 is the highest.
- urgency: How soon the issue should be resolved. There is a strong correlation between Urgency and Priority; which might cause to ommit the field when using Priority.
- impact: What's the extension of the issue in terms of number of users. eg: Limited means small group usually 1 or 2 users, Spread-out means usually an area, department or even all organization. There is a strong correlation between Urgency and Priority; which might cause to ommit the field when using Priority.
- Closed: Date when the ticket was finally closed. Refer to the Opened field for explanation of the stages of the tickets.
- sup_grp: Support Group providing resolution to the issue. This is relevant as the support group is responsible to effectively close a ticket as soon as it's assigned.
- grp_level: Support Group Level. There are 3 different groups of support level.

Level 1: Service Desk, primary group who handles all tickets and try to troubleshot the issue. Most of the tickets should be filtered by this team. This is less specialized team, and help to keep Level 2 and 3 focus on major activities. Level 2: This is the specialist team who has greater knowledge on how the application operates. This team takes care of tickets Level 1 is not able to resolved. Level 3: This is the Developers of the applications; has complete knowledge of the application and finally able to resolve the issues scalated by L2 team.

For JnJ, the L2 and L3 are more expensive, and the interest of the company is to identify ways to reduce cost translating activities from L3 to L2 and from L2 to L1.

The "Level" word can be removed from the field and use the numbers 1,2,3. Level 1 is the less specialized, and 3 is the most specialized, usually a lot more expensive than 1.

- resolved: Date when the issue was resolved. Refer to the Opened field for explanation of the stages of the tickets.
- res_category: Category of the type of resolution support team completed.
- cust_time: Time in seconds the ticket was waiting for Customer response. Refer to the Opened field for explanation of the stages of the tickets.

- pend_time: Time ticket is on hold. Refer to the Opened field for explanation of the stages of the tickets.
- call_log: Id of the phone call When a call is involved. Not a relevant attribute as not all tickets triggers a phone call.
- chat_log: If of the chat session when user uses instance message with the support team. This new technology is not heavily used, so there are very few observations with this information.

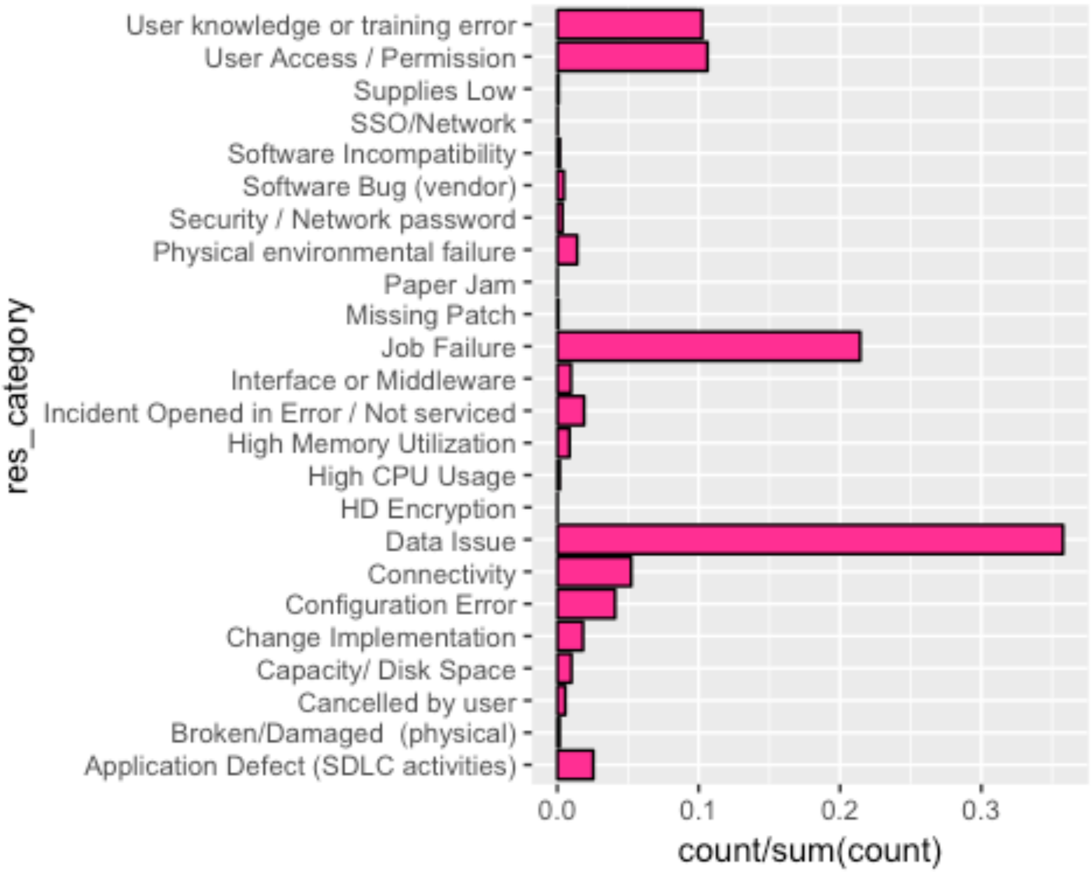
Let's chart the data to understand more about the variables associated to the support activities

Data Visualization

Let's review what the data can tell us about supporting applications for JJTS:

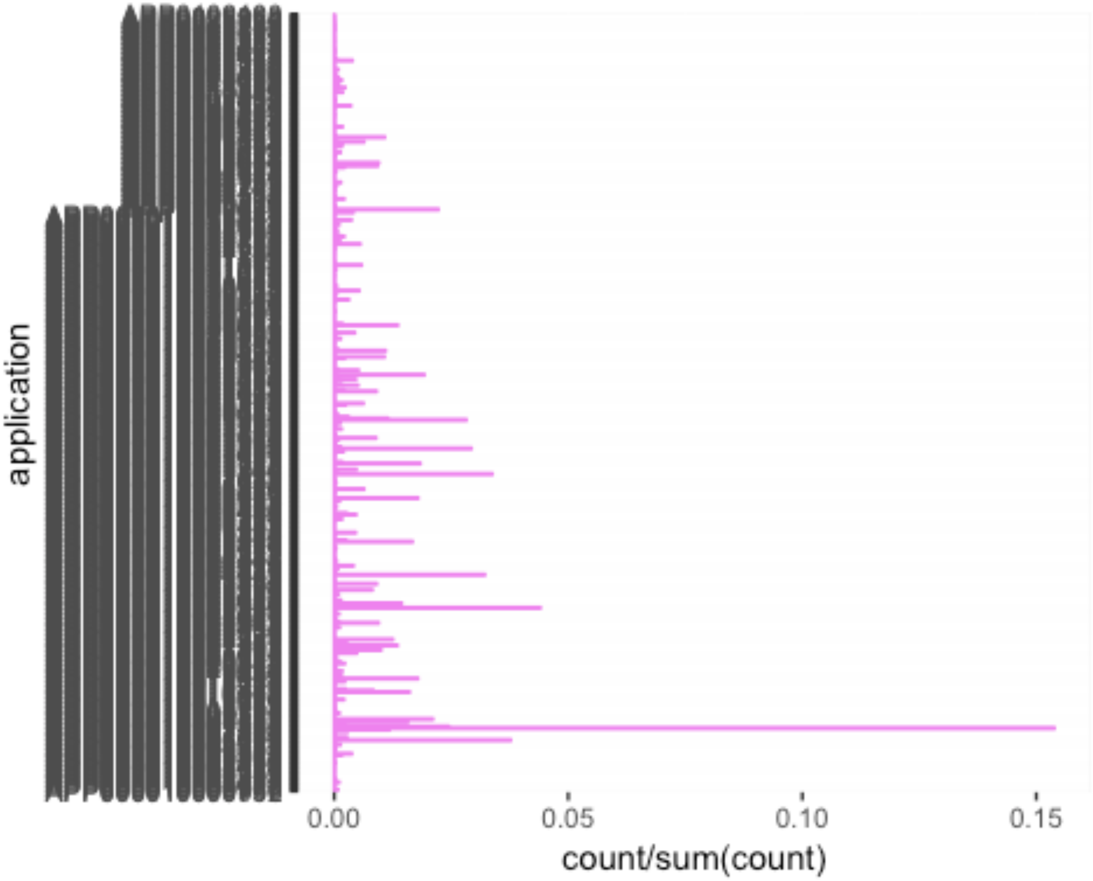
Plot by Application Category

This feature contains great information on how a particular ticket was resolved.



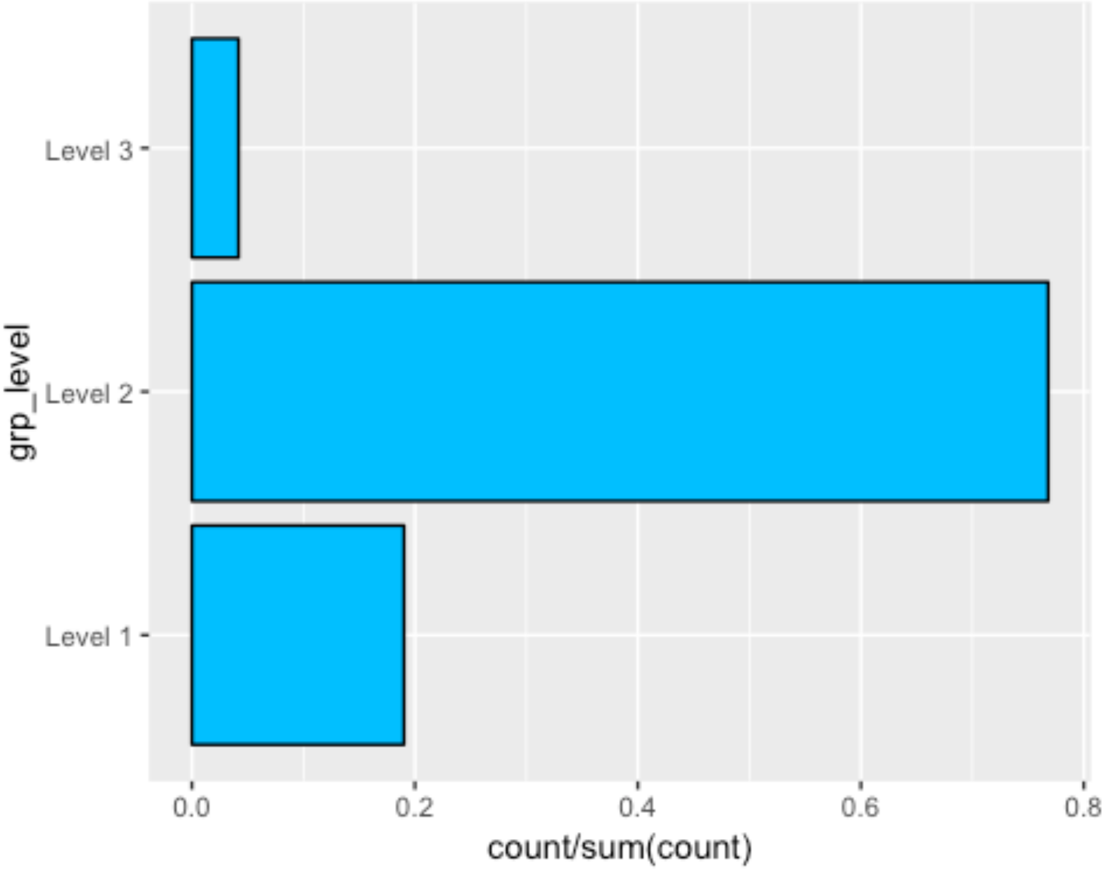
Exploring Applications

There is more than 500 applications. This feature might not be the best for Supervised Algorithms.

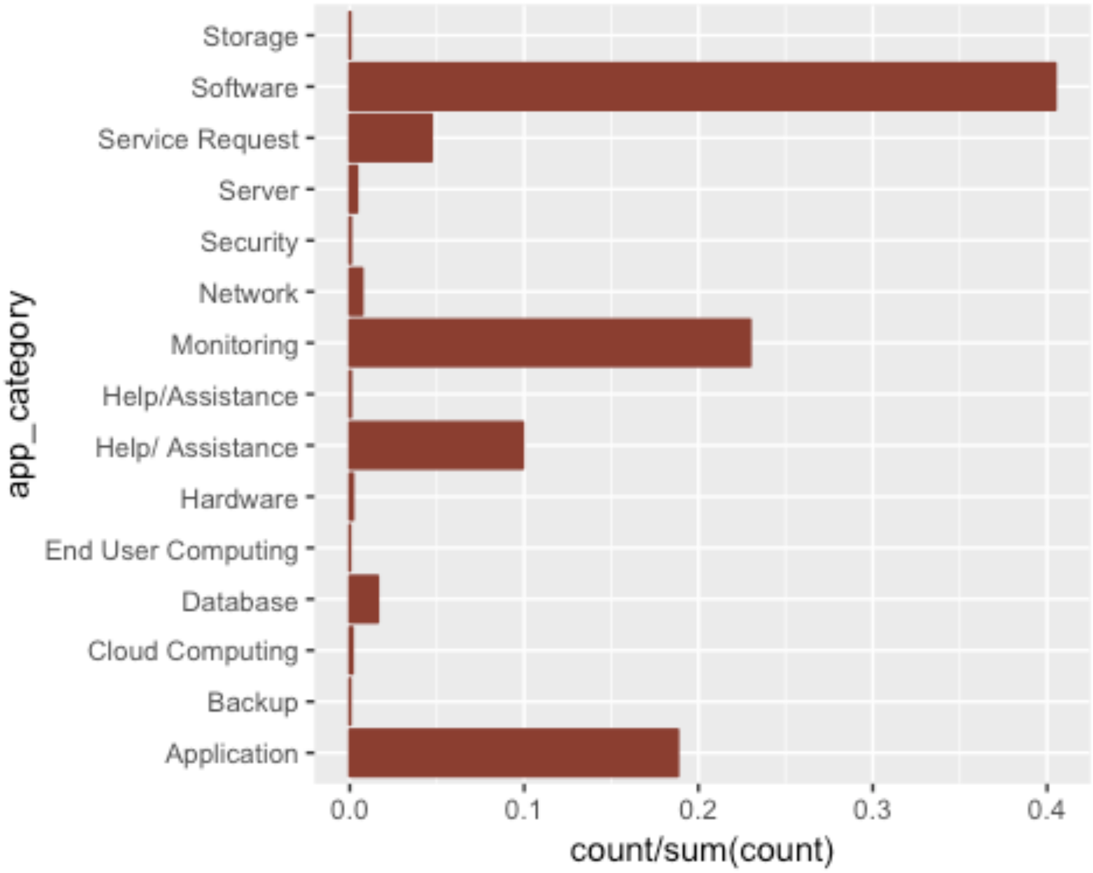


Exploring Support Group Level

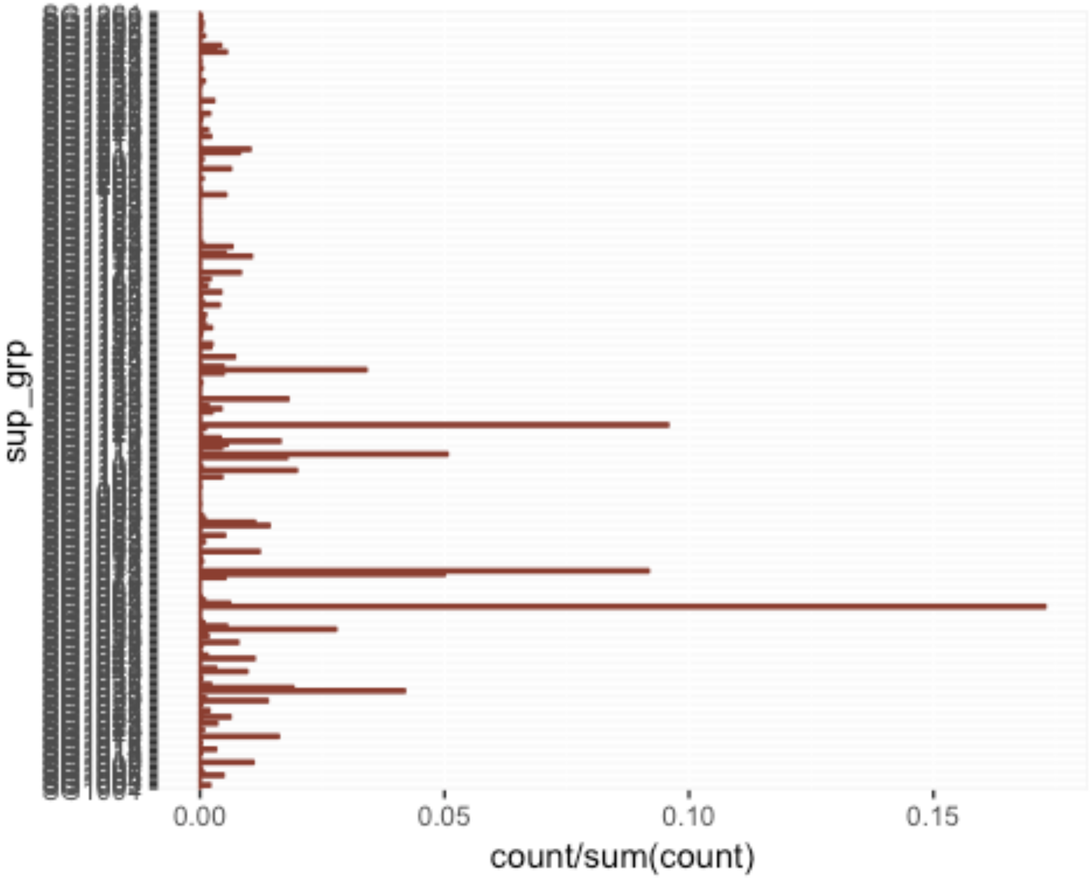
Support Group Level indicates the expertise of the support team. At the same time, the cost of the time goes up while more expert.



Exploring Application Category

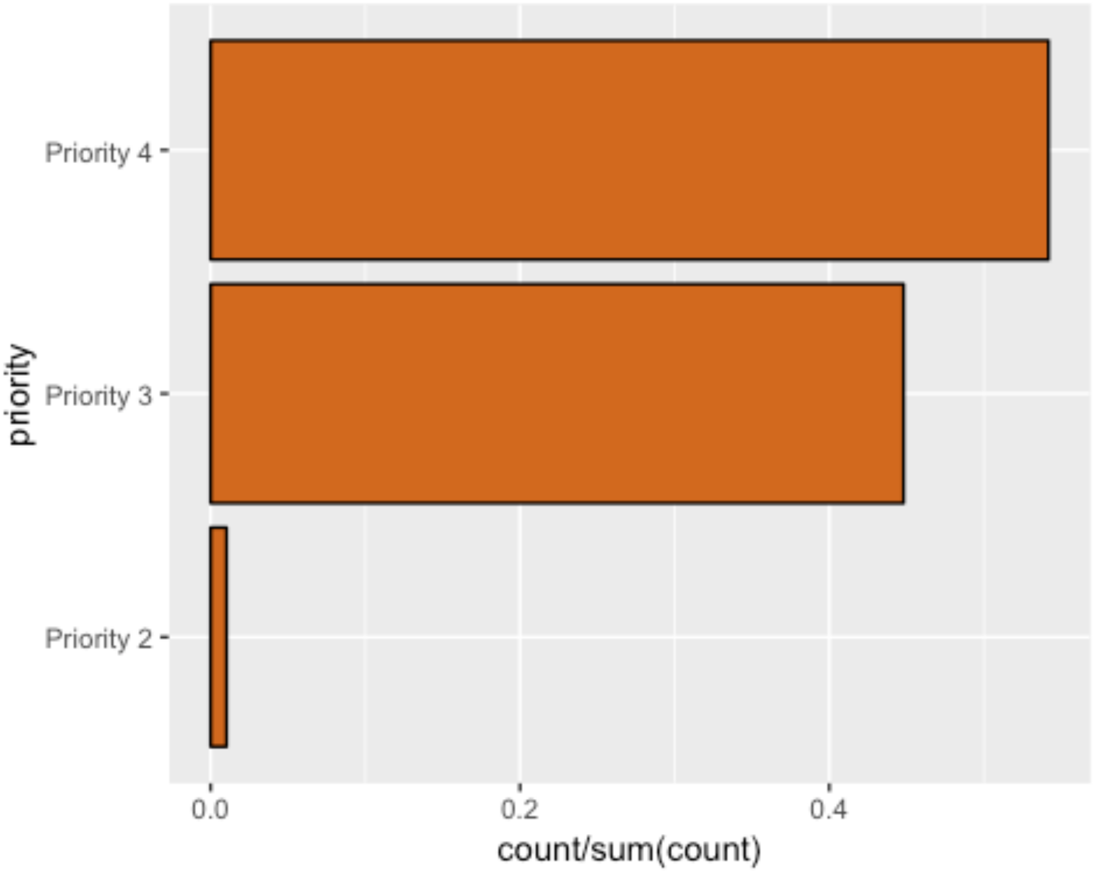


Exploring Support Group



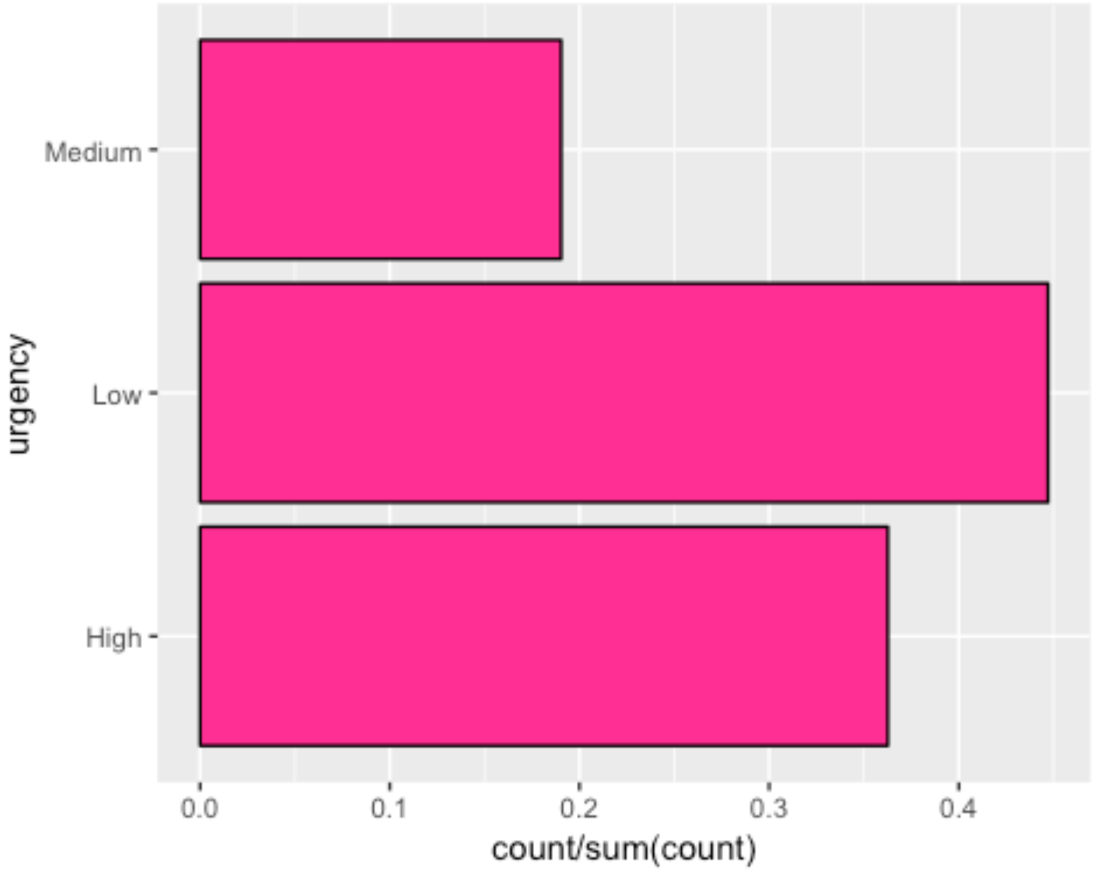
Exploring Priority

Priority is one of the most important features, and it comes from the combination or Urgency and Impact.



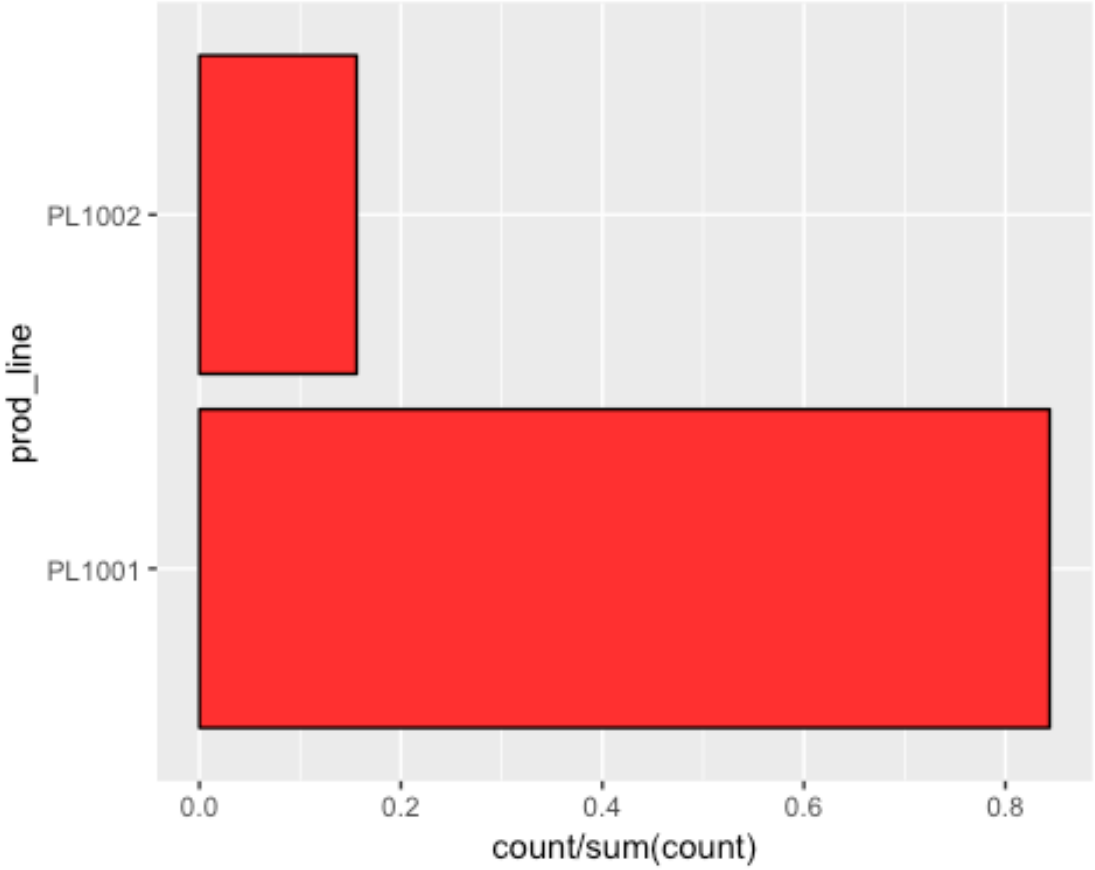
Exploring Urgency

How soon the issue needs to be resolved. There is a strong correlation with Priority.



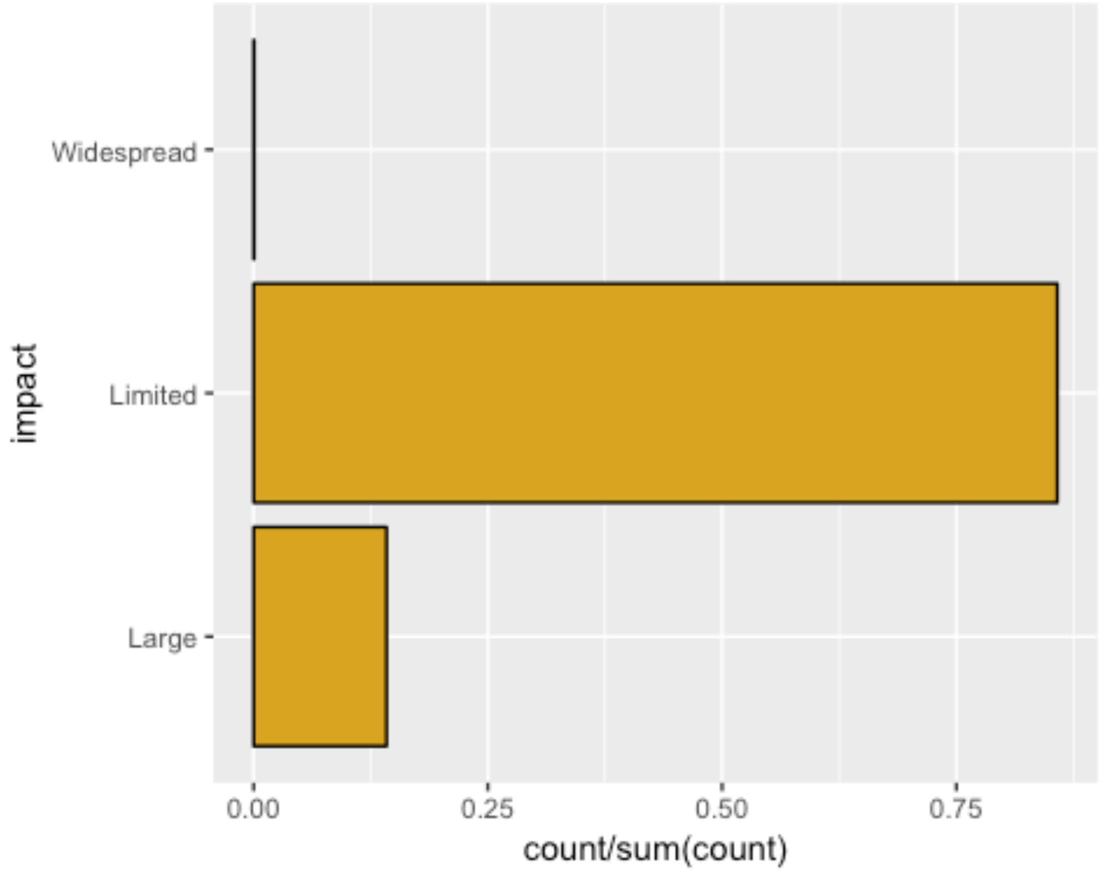
Exploring Product Line

There are only two product lines in this data set. Its relevance might not be the highest but we'll keep it while our analysis.



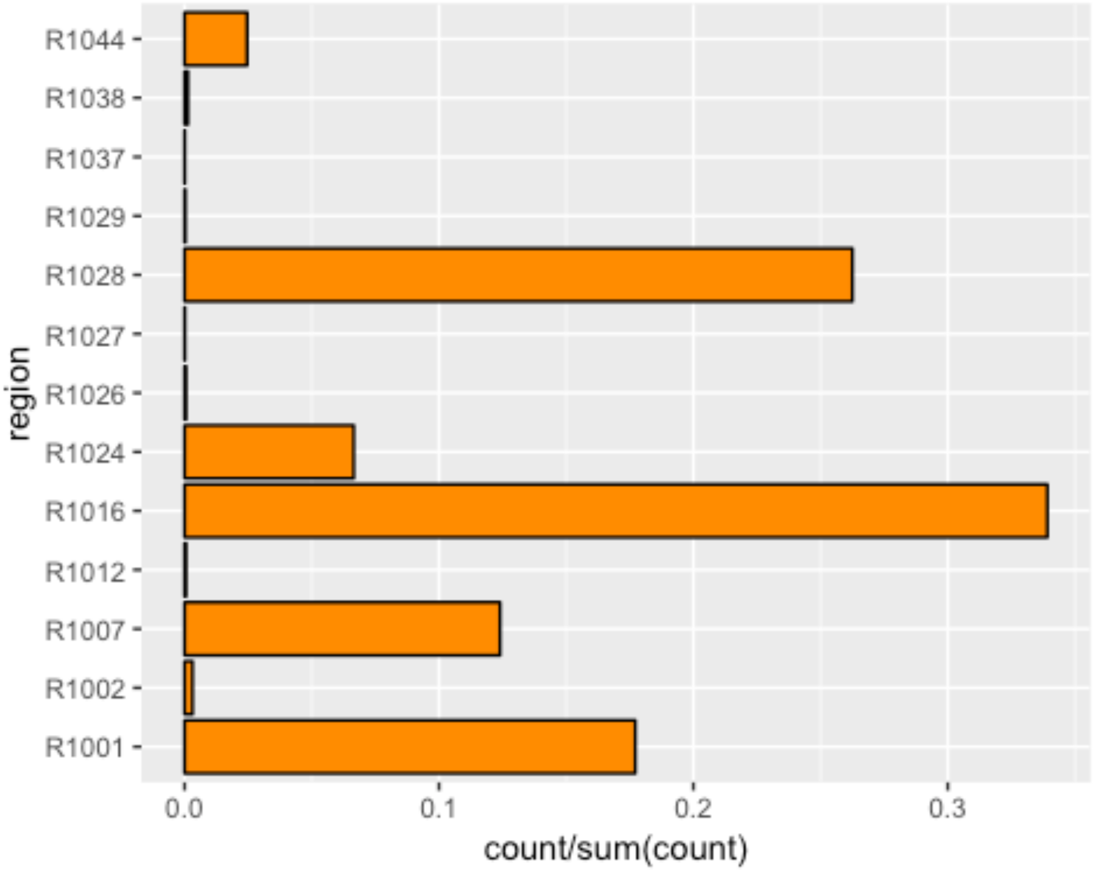
Exploring the Impact

This is associated to the number of users affected by the issue.



Exploring Region of where the Users are located

When a ticket is created, the location of the user is recorded as well.



Data Preparation

Below attributes will be removed from the Dataset due to the low analytical value: Incident: This is the ID of the ticket. We only use it to ensure there are not duplicates. cust_time: We will focus on the time that the ticket is resolved, customer time with other teams is not relevant for our analysis Pend_time: We will focus on the time that the ticket is resolved, pending time with other teams is not relevant for our analysis call_log: This feature is not used mainly; while Support team uses Skype IM chat_log: Less than 1% of the tickets are manages through chat from ServiceNow; support team usually use Skype IM, which is not recorded in the dataset. Closed: We will focus our analysis on Resolved tickets, close is an automatic process happening 12 days after the ticket was resolved.

Here is the dataset:

```
## 'data.frame':    21291 obs. of  12 variables:
##  $ application : Factor w/ 535 levels "APP000010000791",..: 375 523 286
375 188 69 148 49 126 226 ...
##  $ region      : Factor w/ 13 levels "R1001","R1002",..: 9 9 5 9 3 9 3 9 1
3 ...
```

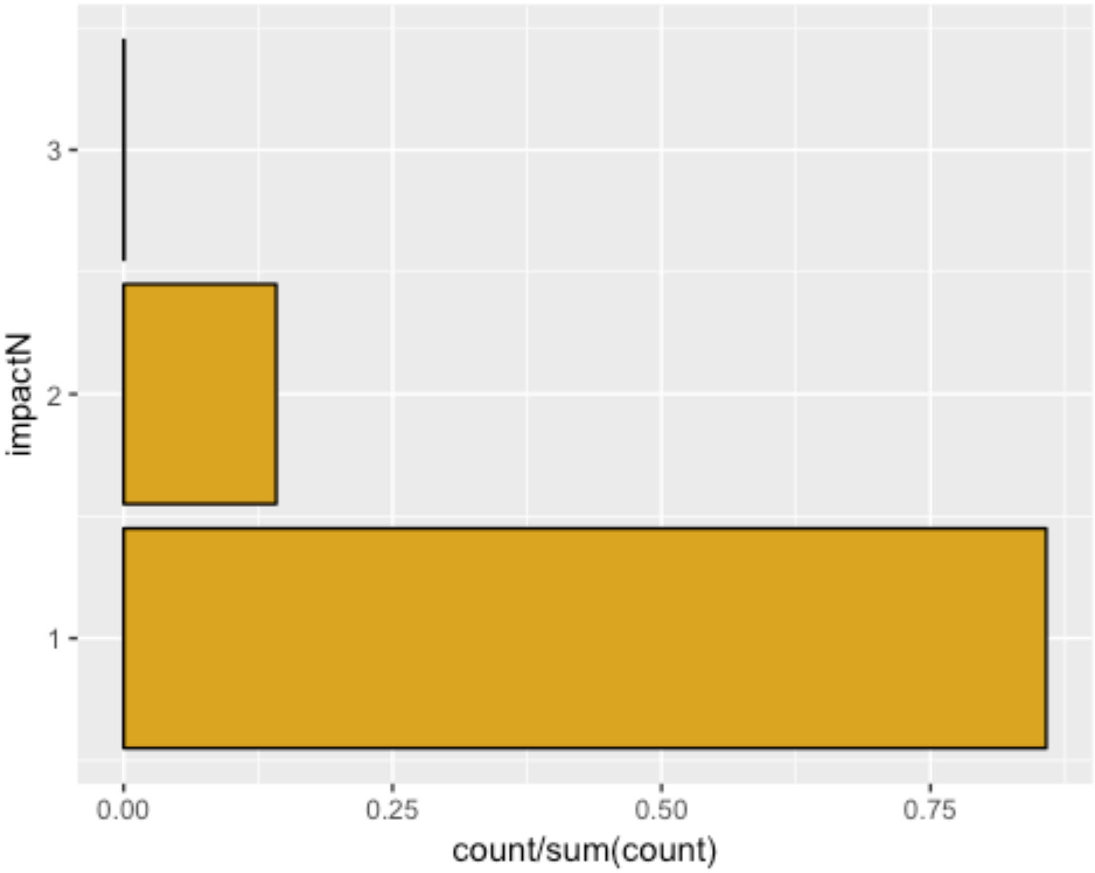
```
##  $ prod_line   : Factor w/ 2 levels "PL1001","PL1002": 2 2 1 2 2 1 1 1 1 2
...
##  $ opened      : Factor w/ 20047 levels "2018-01-01 20:03",..: 9697 6935
12718 9721 13647 2021 3796 6688 3270 16404 ...
##  $ app_category: Factor w/ 15 levels "Application",..: 15 15 15 15 14 14
14 14 14 14 ...
##  $ priority    : Factor w/ 3 levels "Priority 2","Priority 3",..: 3 3 2 2
3 2 3 3 3 3 ...
##  $ urgency     : Factor w/ 3 levels "High","Low","Medium": 2 2 1 1 2 1 2 3
2 3 ...
##  $ impact      : Factor w/ 3 levels "Large","Limited",..: 2 2 2 2 2 2 2 2
2 2 ...
##  $ sup_grp     : Factor w/ 261 levels "SG1001","SG1002",..: 230 39 62 123
136 248 18 114 33 87 ...
##  $ grp_level   : Factor w/ 5 levels "","3rd Party",..: 3 3 4 3 3 4 4 5 4 4
...
##  $ resolved    : Factor w/ 20314 levels "","2018-01-02 0:45",..: 9414 6806
13406 9321 13459 2002 3986 6544 3133 16987 ...
##  $ res_category: Factor w/ 25 levels "","Application Defect (SDLC
activities)",..: 25 15 9 13 8 9 9 9 9 8 ...
```

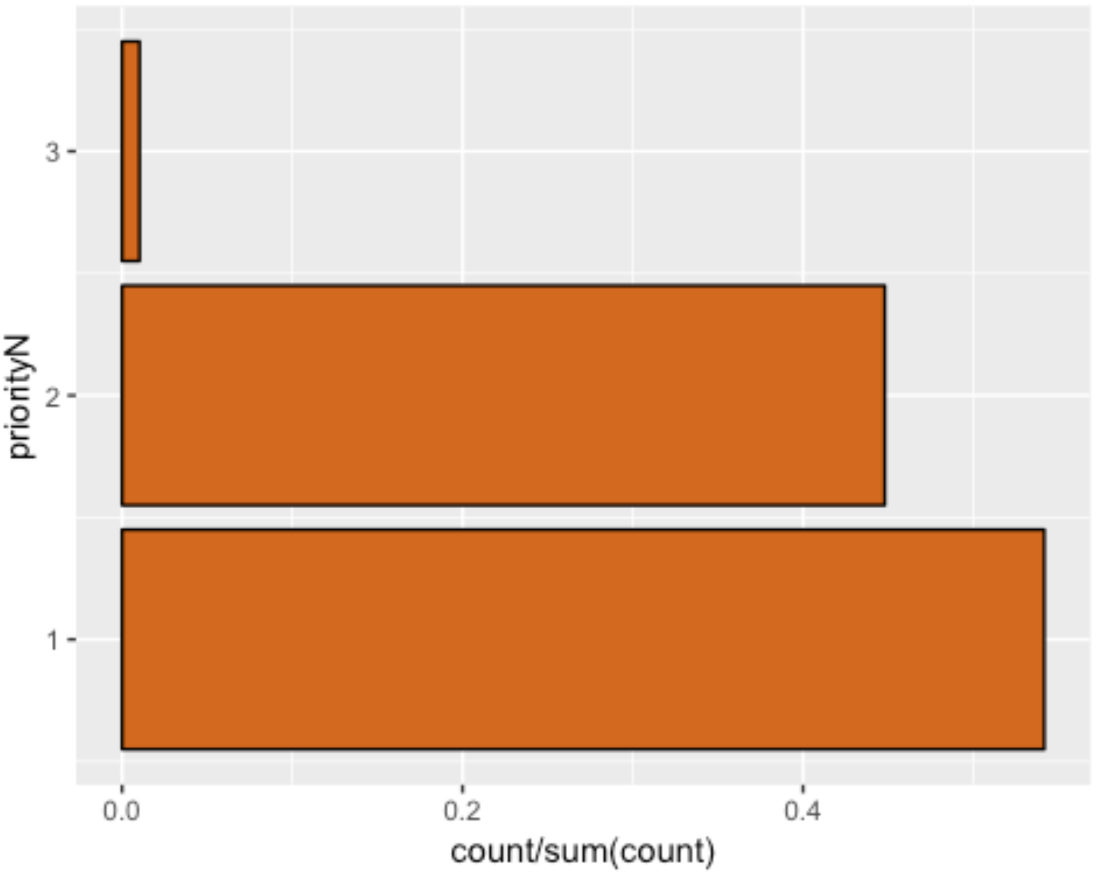
New Numeric Variables

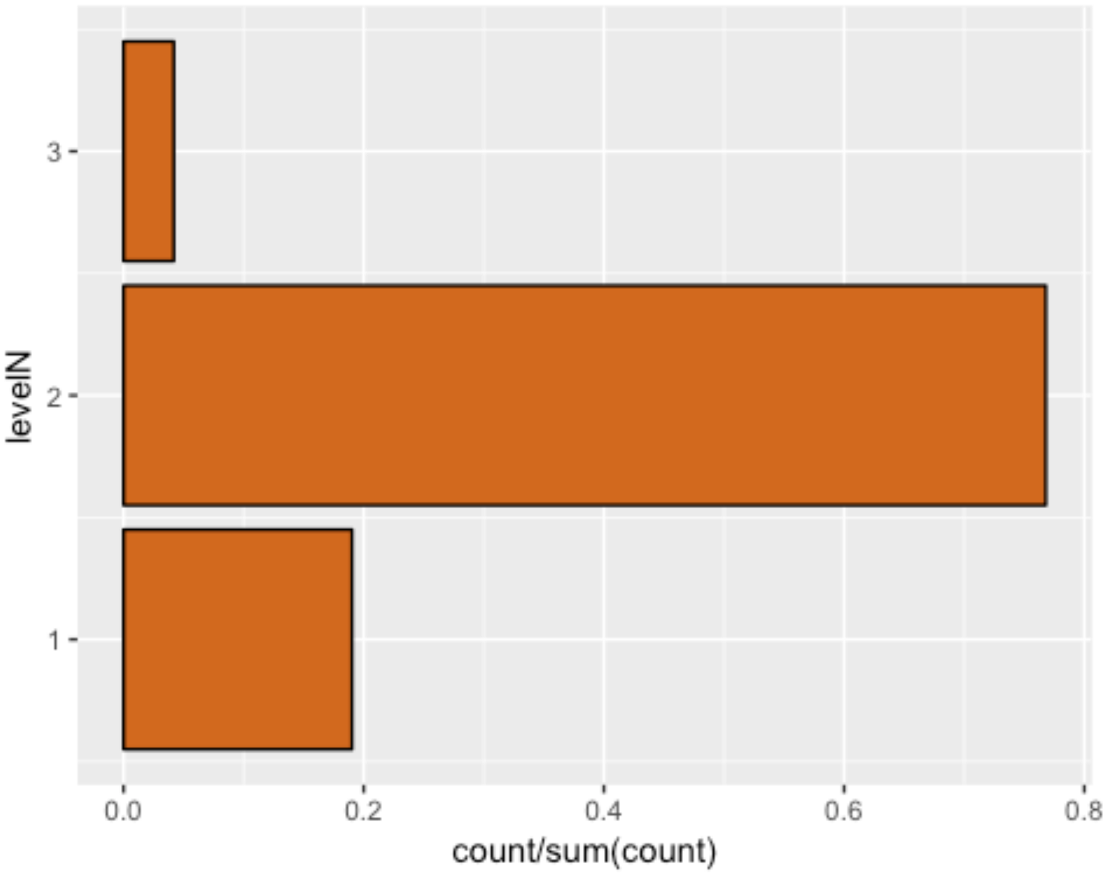
We are now creating Numeric representation of Impact -> impactN Urgency -> urgencyN Priority -> priorityN Group Level -> LevelN

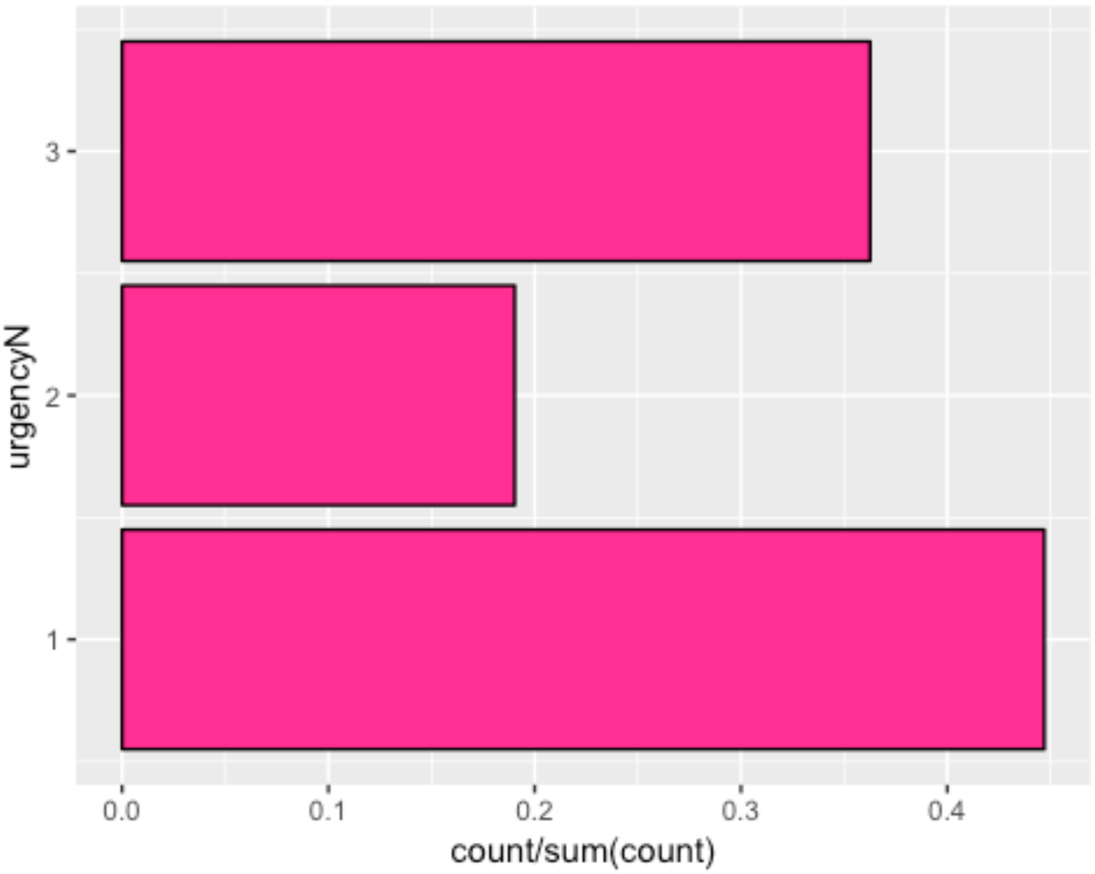
```
## 'data.frame':    21291 obs. of  12 variables:
##  $ application : Factor w/ 535 levels "APP000010000791",..: 375 523 286
375 188 69 148 49 126 226 ...
##  $ region      : Factor w/ 13 levels "R1001","R1002",..: 9 9 5 9 3 9 3 9 1
3 ...
##  $ prod_line   : Factor w/ 2 levels "PL1001","PL1002": 2 2 1 2 2 1 1 1 1 2
...
##  $ opened      : Factor w/ 20047 levels "2018-01-01 20:03",..: 9697 6935
12718 9721 13647 2021 3796 6688 3270 16404 ...
##  $ app_category: Factor w/ 15 levels "Application",..: 15 15 15 15 14 14
14 14 14 14 ...
##  $ sup_grp     : Factor w/ 261 levels "SG1001","SG1002",..: 230 39 62 123
136 248 18 114 33 87 ...
##  $ resolved    : Factor w/ 20314 levels "","2018-01-02 0:45",..: 9414 6806
13406 9321 13459 2002 3986 6544 3133 16987 ...
##  $ res_category: Factor w/ 25 levels "","Application Defect (SDLC
activities)",..: 25 15 9 13 8 9 9 9 9 8 ...
##  $ impactN     : chr  "1" "1" "1" "1" ...
##  $ urgencyN    : chr  "1" "1" "3" "3" ...
##  $ priorityN   : chr  "1" "1" "2" "2" ...
##  $ levelN      : num  1 1 2 1 1 2 2 3 2 2 ...
```

Let’s see visually the new Numeric Features




priorityN
3
2
1
0.0
0.2
0.4
count/sum(count)


levelN
3
2
1
0.0
0.2
0.4
0.6
0.8
count/sum(count)



Dates to chart and Duration of a Ticket

Now we will create the numeric representation of the Date variables and calculate the number of days that support team took to resolve an issue.

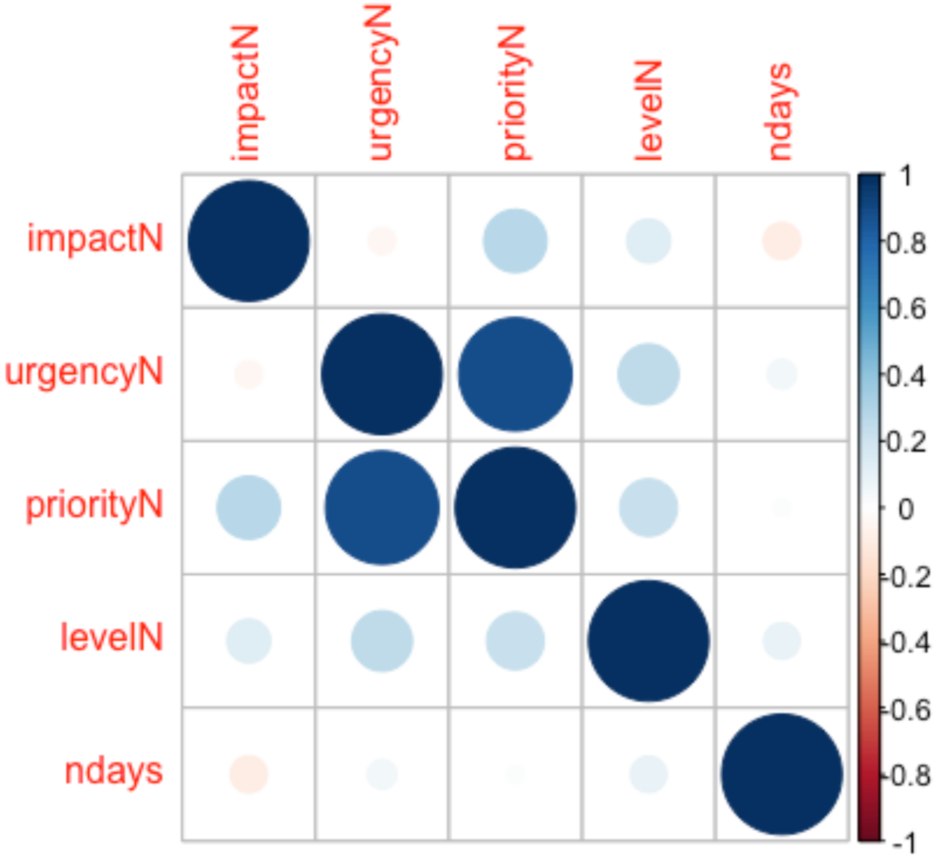
ndays is the time support team took to resolved the issue, in this case is calculated as Resolved - Opened

```
## 'data.frame':    21291 obs. of  13 variables:
##  $ application : Factor w/ 535 levels "APP000010000791",..: 375 523 286
375 188 69 148 49 126 226 ...
##  $ region      : Factor w/ 13 levels "R1001","R1002",..: 9 9 5 9 3 9 3 9 1
3 ...
##  $ prod_line   : Factor w/ 2 levels "PL1001","PL1002": 2 2 1 2 2 1 1 1 1 2
...
##  $ app_category: Factor w/ 15 levels "Application",..: 15 15 15 15 14 14
14 14 14 14 ...
##  $ sup_grp     : Factor w/ 261 levels "SG1001","SG1002",..: 230 39 62 123
136 248 18 114 33 87 ...
##  $ res_category: Factor w/ 25 levels "","Application Defect (SDLC
activities)",..: 25 15 9 13 8 9 9 9 9 8 ...
##  $ impactN     : num  1 1 1 1 1 1 1 1 1 1 ...
##  $ urgencyN    : num  1 1 3 3 1 3 1 2 1 2 ...
##  $ priorityN   : num  1 1 2 2 1 2 1 1 1 1 ...
##  $ levelN      : num  1 1 2 1 1 2 2 3 2 2 ...
##  $ open_date   : Date, format: "2018-10-03" "2018-07-30" ...
##  $ resolve_date: Date, format: "2018-10-05" "2018-08-02" ...
##  $ ndays       : num  2 3 11 0 0 3 14 3 1 3 ...
```

Correlation Matrix

Based on our previous charts, we are now curious to see if there is any feature which its correlation might cause to have it dropped.

```
##               impactN    urgencyN  priorityN     levelN       ndays
## impactN    1.00000000 -0.04820566 0.27259729 0.13022254 -0.09302363
## urgencyN  -0.04820566  1.00000000 0.88568657 0.25126735  0.05838574
## priorityN  0.27259729  0.88568657 1.00000000 0.22596517  0.01865288
## levelN     0.13022254  0.25126735 0.22596517 1.00000000  0.09020365
## ndays     -0.09302363  0.05838574 0.01865288 0.09020365  1.00000000
```



From the figure we identify that urgency and priority are strongly correlated Priority and impact are weakly correlated (0.27) this could be because it is defined by the user during the ticket triaging. Similarly priority and level are weakly correlated. ndays has a very low correlation.

Since priority and urgency are highly correlated (0.89) urgency is dropped from further analysis.

```
## 'data.frame':    21291 obs. of  12 variables:
##  $ application : Factor w/ 535 levels "APP000010000791",..: 375 523 286
375 188 69 148 49 126 226 ...
##  $ region      : Factor w/ 13 levels "R1001","R1002",..: 9 9 5 9 3 9 3 9 1
3 ...
##  $ prod_line   : Factor w/ 2 levels "PL1001","PL1002": 2 2 1 2 2 1 1 1 1 2
...
##  $ app_category: Factor w/ 15 levels "Application",..: 15 15 15 15 14 14
14 14 14 14 ...
##  $ sup_grp     : Factor w/ 261 levels "SG1001","SG1002",..: 230 39 62 123
136 248 18 114 33 87 ...
##  $ res_category: Factor w/ 25 levels "","Application Defect (SDLC
activities)",..: 25 15 9 13 8 9 9 9 9 8 ...
##  $ impactN     : num  1 1 1 1 1 1 1 1 1 1 ...
##  $ priorityN   : num  1 1 2 2 1 2 1 1 1 1 ...
##  $ levelN      : num  1 1 2 1 1 2 2 3 2 2 ...
##  $ open_date   : Date, format: "2018-10-03" "2018-07-30" ...
##  $ resolve_date: Date, format: "2018-10-05" "2018-08-02" ...
##  $ ndays       : num  2 3 11 0 0 3 14 3 1 3 ...
```

Low Frequency Cleaning

In particular, App Category, Resolution category and Region contains very low frequency levels which could reduce accuracy for our predictions or computing time during our cluster analys. We choose the Threshold = 2.5% to remove observations.

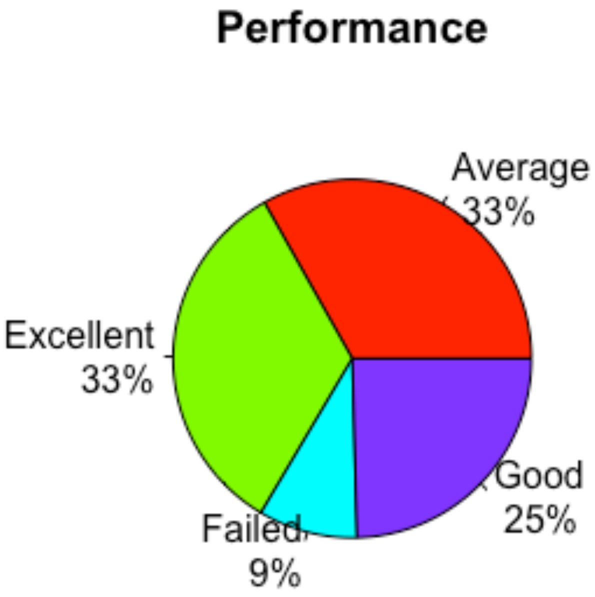
```
##           application      region      prod_line
##  APP000010007299: 3199   R1001:3237   PL1001:14920
##  APP000010022869:  892   R1007:2013   PL1002: 2503
##  APP000010006077:  651   R1016:6372
##  APP000010027175:  608   R1024:1028
##  APP000010027900:  558   R1028:4773
##  APP000010027488:  498
##  (Other)        :11017
##            app_category     sup_grp
##  Application     :3144   SG1062 :3266
##  Help/ Assistance:1735   SG1123 :1837
##  Monitoring      :4374   SG1073 :1482
##  Service Request : 937   SG1112 : 990
##  Software        :7233   SG1033 : 847
##                          SG1072 : 775
##                          (Other):8226
##                             res_category     impactN        priorityN
##  Configuration Error             : 840   Min.   :1.000   Min.   :1.00
##  Connectivity                    :1049   1st Qu.:1.000   1st Qu.:1.00
##  Data Issue                      :7069   Median :1.000   Median :1.00
##  Job Failure                     :4484   Mean   :1.148   Mean   :1.45
##  User Access / Permission        :2074   3rd Qu.:1.000   3rd Qu.:2.00
##  User knowledge or training error:1907   Max.   :3.000   Max.   :3.00
##
##      levelN        open_date            resolve_date
##  Min.   :1.000   Min.   :2018-01-01   Min.   :2018-01-02
##  1st Qu.:2.000   1st Qu.:2018-06-04   1st Qu.:2018-06-08
##  Median :2.000   Median :2018-10-21   Median :2018-10-24
##  Mean   :1.846   Mean   :2018-09-13   Mean   :2018-09-17
##  3rd Qu.:2.000   3rd Qu.:2018-12-28   3rd Qu.:2018-12-31
##  Max.   :3.000   Max.   :2019-02-26   Max.   :2019-02-26
##
##      ndays
##  Min.   :  0.000
##  1st Qu.:  0.000
##  Median :  1.000
##  Mean   :  3.969
##  3rd Qu.:  4.000
##  Max.   :265.000
##

## [1] 17423
```

Calculating Performance

We are rating the servcies provided by the vendor the following way: Excellent: All P1, P2 or P3 tickets resolved within 1 day Good: For P2 tickets resolved within 5 Days Average: For P2 or P1 Tickets resolved within 10 days Failed: For any other ticket is considered Bad performance is rated as failed service. - P3 resolved in more than 1 day - P2 resolved in more than 5 days - P1 resolved in more than 10 days

```
##   Average Excellent    Failed      Good
##      5778      5807      1556      4282
```



Supervised Learning

Adding the reason of using the below predictors priority + grp_level + app_category + res_category + region + prod_line

Feature selection

Let's run a random forest to quantify the relative importance of these features. We will use features with less than 50 Levels, so Application and Support Group will go away. Also Dates should not be considered. Ndays is not a predictable variable because we can't actually trying to Predict the Performance of resolving a ticket when Duration is provided since we won't know how long the ticket will last unresolved, but we will know what Priority the ticket is raised.

Finally, Resolution category is unknown as we don't know what the issue is. We can predict based on the resolution.

```
##              MeanDecreaseGini
## priorityN           1971.9048
## levelN               676.6039
## app_category         380.5669
## region               277.6348
## impactN              130.9250
## prod_line            119.8023
```

As shown in the table, Product Line has the lowest predictable power. Understandable because there is only two Product Lines; and the decision is either one of the other. Very limited predictibility power.

Suppervised Modeling

Classification and Regression Trees

For our Analysis, we will select 5 of the most predictable features:

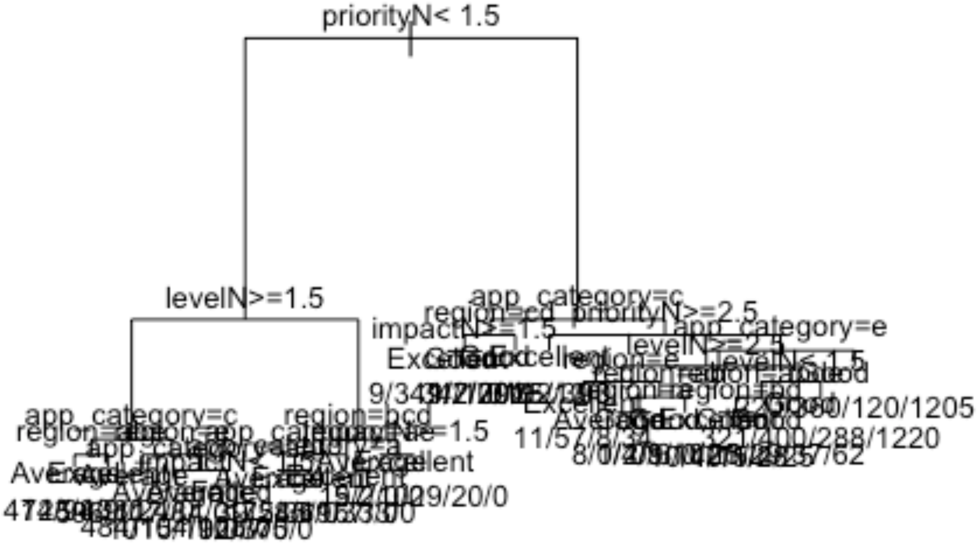
Priority, Support Level, App Category, Resolution Category, Region

```
##
## Classification tree:
## rpart(formula = performance ~ ., data = trainDF, control =
rpart.control(cp = 1e-04))
##
## Variables actually used in tree construction:
## [1] app_category impactN      levelN       priorityN    region
##
## Root node error: 8713/13069 = 0.66669
##
## n= 13069
##
##            CP nsplit rel error  xerror      xstd
## 1  0.29255136      0   1.00000 1.00964 0.0061545
## 2  0.12441180      1   0.70745 0.70745 0.0065497
## 3  0.00832090      2   0.58304 0.58304 0.0063957
## 4  0.00659933      4   0.56640 0.56789 0.0063640
## 5  0.00286928      6   0.55320 0.55320 0.0063305
## 6  0.00087991      7   0.55033 0.55079 0.0063247
## 7  0.00045908     10   0.54769 0.55136 0.0063261
## 8  0.00042083     13   0.54631 0.54918 0.0063208
## 9  0.00034431     16   0.54505 0.54872 0.0063197
## 10 0.00028693     18   0.54436 0.54872 0.0063197
## 11 0.00026780     20   0.54379 0.54757 0.0063169
## 12 0.00010000     23   0.54298 0.54642 0.0063141
```

The tree has a misclassification rate of 0.66669 * 0.52531 * 100% = 35% in cross-validation (65% of prediction accuracy).

Pruning and ploting model

We now pick the tree size that minimizes prediction error which is a misclassification rate Prediction error rate in training data = Root node error * rel error * 100% Prediction error rate in cross-validation = Root node error * xerror * 100% We want the cp value that minimizes the xerror

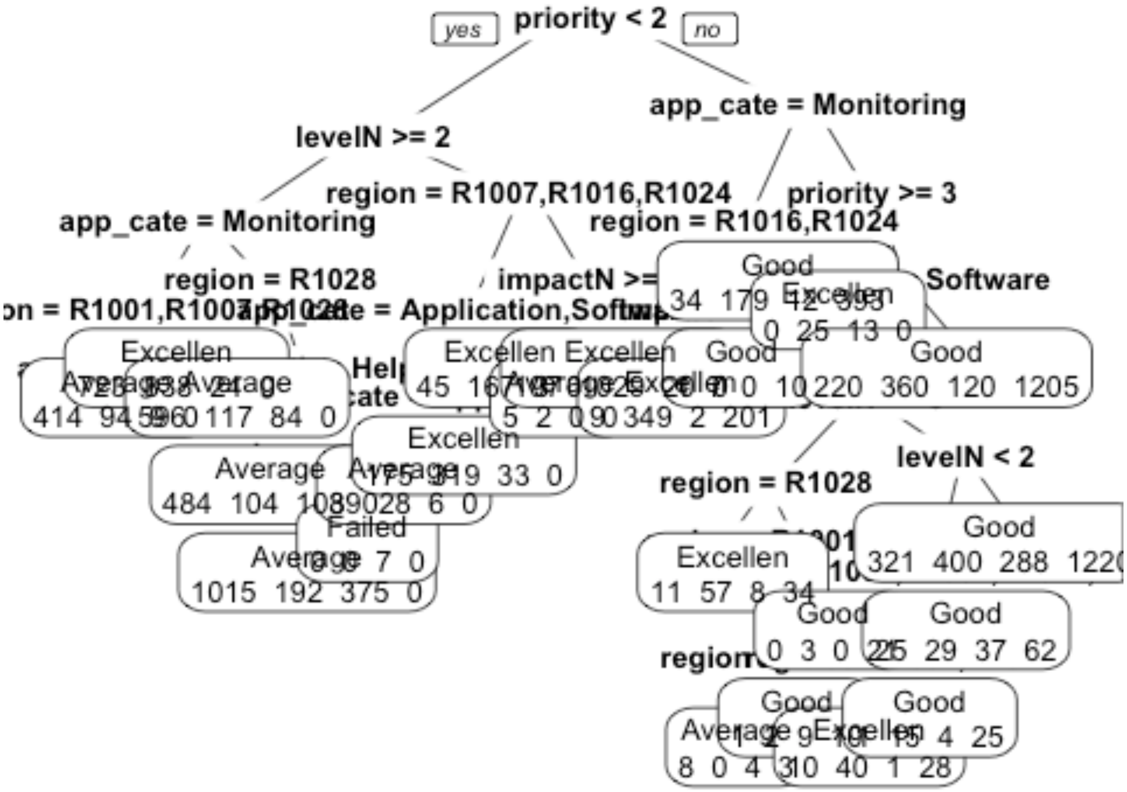


##

Confusion Matrix for Training data

```
##
##                     Pred:Average Pred:Excellent Pred:Failed Pred:Good
##   Actual:Average            2561           1170           0       603
##   Actual:Excellent           537           2824           0       995
##   Actual:Failed              586            104           7       470
##   Actual:Good                  3            263           0      2946
```

Ploting the Tree Pruned



Let’s test the model (accuracy of testing dataset)

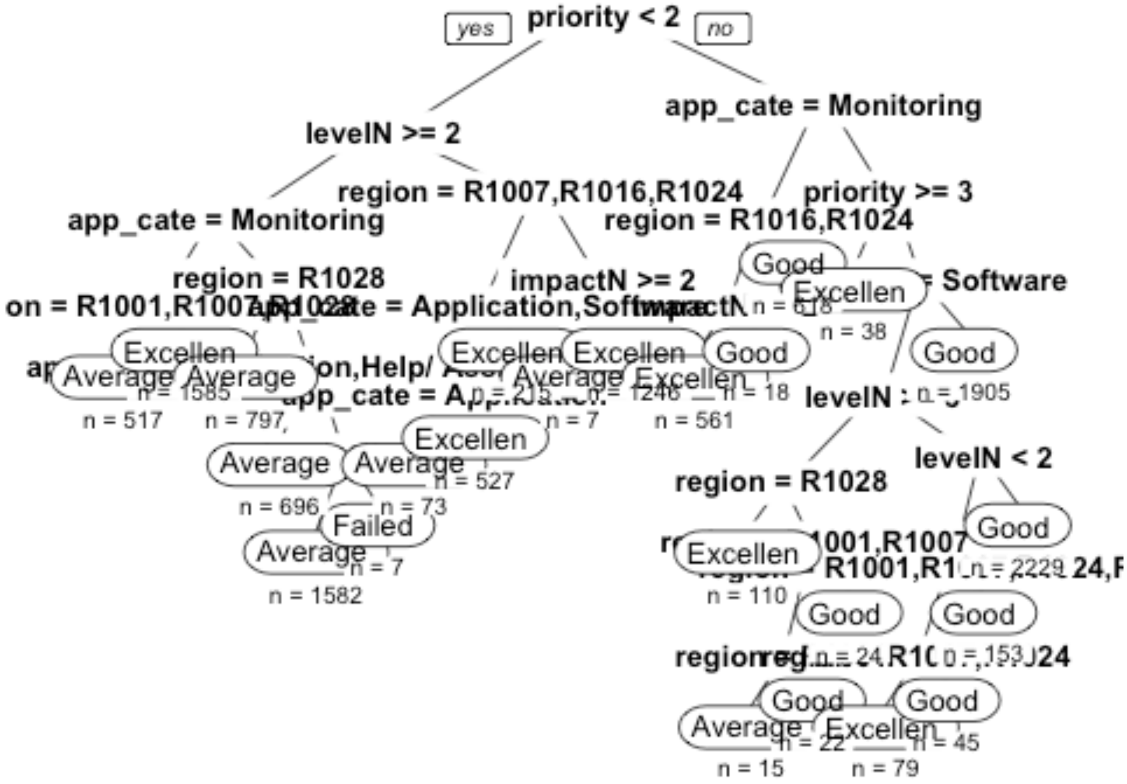
a) Classification Tree / Recursive Partitioning and Confusion Matrix

```
##   Average Excellent    Failed      Good
##      1444      1451       389      1070
```

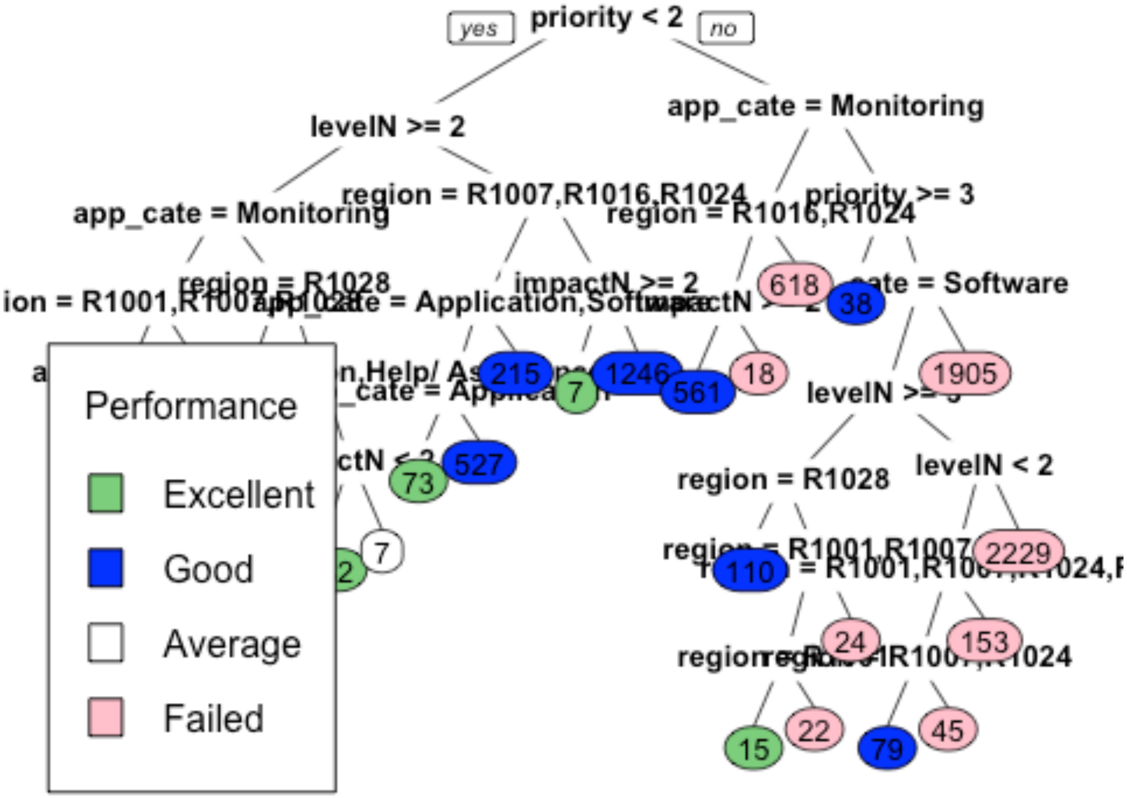
```
##     Average         Excellent          Failed             Good
##  Min.   :0.0000   Min.   :0.0000   Min.   :0.00000   Min.   :0.0000
##  1st Qu.:0.1155   1st Qu.:0.1795   1st Qu.:0.01605   1st Qu.:0.0000
##  Median :0.1581   Median :0.1890   Median :0.06299   Median :0.0000
##  Mean   :0.3295   Mean   :0.3328   Mean   :0.08913   Mean   :0.2486
##  3rd Qu.:0.6416   3rd Qu.:0.5287   3rd Qu.:0.12921   3rd Qu.:0.5473
##  Max.   :0.8008   Max.   :0.8258   Max.   :1.00000   Max.   :0.8750
```

```
##      Average Excellent    Failed      Good
## 6  0.1440108 0.1794527 0.1292059 0.5473306
## 13 0.1440108 0.1794527 0.1292059 0.5473306
## 15 0.6415929 0.1213654 0.2370417 0.0000000
## 16 0.3320683 0.6053131 0.0626186 0.0000000
## 17 0.3320683 0.6053131 0.0626186 0.0000000
## 20 0.6415929 0.1213654 0.2370417 0.0000000
```

Ploting the Tree in black in White in case of not having a Color printer



Adding color to the Tree for each Performance



Deployment

Predicting outcome of an unseen data point

```
##      Average Excellent    Failed Good
## NA 0.6415929 0.1213654 0.2370417    0
```