Incident Resolution Time Prediction Modeling

MATH564 Applied Statistics

Sohaib Syed

ssyed27@hawk.iit.edu

Ivan Prskalo

iprskalo@hawk.iit.edu

12/02/22

50% Sohaib

50% Ivan

Performed pair programming for most of coding, and both worked equally on the final report.

Abstract

We chose the “incident management process enriched event log” dataset from the UCI machine learning repository to find an efficient model that can accurately predict how long different incidents from an IT company would take to resolve. This dataset had records from an audit system that gathered data from the “ServiceNowTM” platform. There were a total of 141,712 records that had 32 descriptive attributes and 2 dependent variables that we derived from a single dependent variable, time for incident resolution. While this dataset had a lot of attributes, we deemed only 10 of them as contextually relevant to our problem. After cleaning, only 59,087 records were present which meant we got rid of over half of the records. We trained 3 regression models and 3 classification models. The linear models did not perform as well as the classification models. Of the 10 relevant attributes, category, subcategory and incident state were found to be the most important. Despite having a lot of data to work with, we found a lot of this data to be unusable or low quality which led to our models having poorer performance than initially expected. The overall usage of these models in the context of our problem is still possible, but due to skewing and covariance issues, we would tell users to expect varied results.

Introduction

We decided to solve a real world problem for our report. We chose the “incident management process enriched event log” dataset from the UCI machine learning repository. This dataset includes records from an audit system that gathered data from the “ServiceNowTM” platform used by an IT company. The dependent variables provided were related to the time it took for the incidents to be resolved. The goal of this

project was to accurately predict the time it took for incidents to be resolved. We soon discovered how challenging this would be with a linear regression model. After cleaning and analyzing the data, we found our multiple regression model had a low adjusted R^2 value of ~.19. When we plotted the actual values from our test set against the predicted values the results were less than promising. We decided to instead categorize the time it took for incident resolution into intervals shifting the problem from regression to classification. We chose the intervals, less than one day, less than a week, less than two weeks, less than a month, less than 6 months, and more than 6 months. This gives users a better grasp of the situation by giving a rough estimate of how long a particular incident is expected to take.

Problem Statement/Data Sources and Data Preprocessing and Preparation

https://archive.ics.uci.edu/ml/datasets/Incident+management+process+enriched+event+log

The incident management process enriched event log dataset has 141,712 records in the dataset and 36 attributes. 32 of these attributes are descriptive, 2 dependent variables, 1 case identifier, and 1 state identifier. Of the total 141,712 records, only 59,087 were still being used post-cleaning. The two dependent variables gave the date and time for incident resolution and the date and time the incident was closed. We were only interested in when the incidents were resolved. We had to convert the dates given into POSIX datetimes and calculate the difference between when that incident record was updated and when it was resolved. This provided us our initial dependent variable, the time in days the incident record took to be resolved.

Of the 32 descriptive attributes, there were many that had a lot of NA's and many that weren't missing much data, but were not relevant enough to include in our models. For example, attributes such as *caller_id, opened_by*, and *sys_created_by* while they might improve model accuracy if they are included, the practicality is lost by including them. It would require dummy variables to be made for each individual caller, opener, and creator, which, if new individuals that weren't present in the training of the model attempted to do this, the model would not be able to function due to unrecognized values. We had a similar issue come up with subcategories, since there were so many different subcategories. It turned out that our test set had a subcategory that was not present in the training set which caused issues when predicting times in the test set.

The descriptive attributes we chose to use in one or more of our models were, *reassignment_count, reopen_count, sys_mod_count, made_sla, incident_state, category, subcategory, impact, urgency,* and *priority*:

Reassignment_count is the amount of times an incident was reassigned to another group to solve.

Reopen_count is the amount of times that an incident solution was rejected by the caller and reopened.

Sys_mod_count is the amount of times that an incident was updated. This includes the actions of the two previous attributes

Made_sla is a boolean attribute that show if the incident exceeded the target SLA

Incident_state is an eight level protocol that determines the state of an incident. Meaning a single incident could have multiple states recorded as multiple records.

Category is a first-level identifier of the affected service

Subcategory is a second-level identifier of the affected service (related to category)

Impact describes the impact caused by the incident (3-levels:Low, Medium, High)

Urgency describes the urgency given by the user (3-levels:Low, Medium, High)

Priority is calculated using *impact* and *urgency*

There were many issues with the data we had to handle in the data preparation and analysis phase. First, there was significant skewing in the distribution of data for many of the relevant descriptive attributes. For example, in *impact* and *urgency*, there were significantly more records in the “Medium” category which would also influence *priority* since this attribute was calculated using impact and urgency. Additionally, *impact* and *urgency* had considerable overlap between their most common categories. *Impact* has 56,025 records in the “Medium” category, and *urgency* has 55,848 records in its “Medium” category. The amount of records that have “Medium” for both attributes is 55,087, which means less than 1,000 of the records have a unique combination of “Medium” and one of the other categories. The issue of covariance is also present in other variables such as *reassignment_count* and *reopen_count*.

Proposed Methodology

We initially planned to create a regression model using relevant descriptive attributes to predict the time it would take for the incidents to get resolved. After tweaking different multiple regression models and comparing them, we found this approach to be less than effective. With this approach in mind, we first tried to do an

exhaustive stepwise selection, however, this method proved to be ineffective and the process was forcefully terminated by the application before it was able to finish. Instead, we performed a backwards stepwise selection putting all attributes that we thought could be relevant and worked backwards.

Upon deliberation, we reconsidered what would be an effective predictive model. Predicting the exact time incidents take to be resolved would not be necessary, instead we could determine approximate time intervals for these incidents to be resolved to give an estimate to the people that it was affecting. In practical use, the exact time, especially if the accuracy was poor, wouldn't make much sense, but estimating the time it would take in intervals would give less information, but also be more practical. For this reason we chose to use a random forest classifier.

Analysis and Results

We applied a total of six predictive models to the data. Three of which were multiple linear regression models and three of which were a random forest classification model. On the training dataset, the full regression model with 10 attributes, had the lowest MSE of ~532 and the highest adjusted R squared value of 0.1888. The second regression model with 9 attributes (removed incident_state) had a MSE of 534 and adjusted R squared value of 0.1865. The third regression model with 5 attributes (category+subcategory+impact+urgency+priority) had a MSE of 534 and adjusted R squared value of 0.1861. On the test dataset, we were able to obtain that the full, second, and third regression models had MSE of 557.7, 556.6, and 556.9. Surprisingly, the second model performed the best, but only by a small margin. The third regression model performed better than the full model. This shows that even though during model

training the regression model with the most attributes performed the best, if the data lacks quality the model will not perform as well on the test set.

The random forest models had identical attributes for full, second and third models as the regression models. The random forest results for a full, 10 attribute model had an overall accuracy of ~39.7%. This is likely due to most of the data in the test set being in the ‘Within a Day’ interval when the split was done on the dataset. The second random forest model had ~38.6% overall accuracy, and the third model had ~37% overall accuracy. The results were on the test dataset which show that a trained classification model will perform better with more predictors unlike the results shown for a regression model where on the test set the performance was essentially the same.

The attributes that were most important to predicting resolution time were category, subcategory, and incident_state. The second model is the model where incident_state was removed and it caused the MSE in the training dataset to increase, and it caused accuracy of the random forest models to decrease. Since category and subcategory had numerous dummy variables, keeping those attributes in the model allowed for a larger model that helped keep performance consistent.

Conclusions

This project is a testament to the saying “more isn’t always better.” We chose a dataset that had a lot of records and a lot of attributes describing these records, but the quality of data was its main shortcoming. There was a lot of overlap present between variables and a heavy skew which made creating an accurate prediction model much more difficult. We also noticed there is considerable mislabeling of data in sections where user input goes unchecked. With better data consistency, we believe much better

models would be achievable. Andrew Ng, a prominent figure in machine learning and AI research, talked about a shift from model-centric to data-centric AI in recent years. While having a lot of data is certainly better than not, if the quality of that data is subpar, the performance of the model, no matter how optimized, will suffer.

Appendix

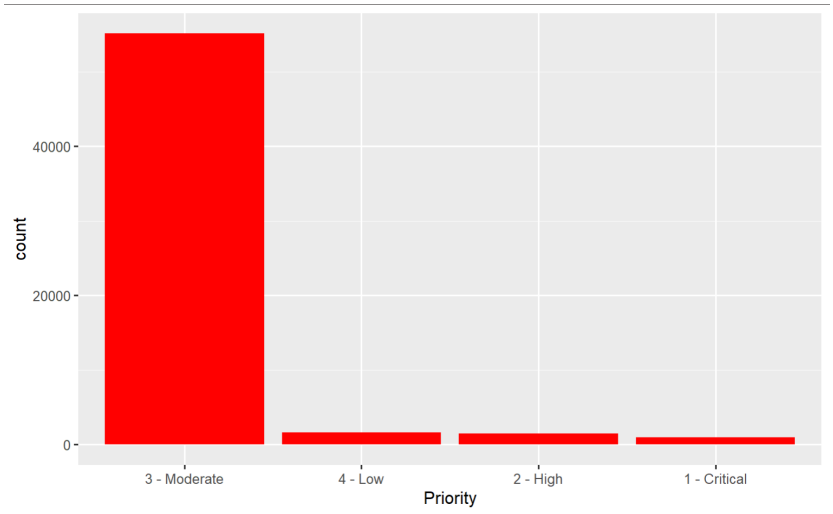


Figure 1- Distribution of Priority attribute in dataset

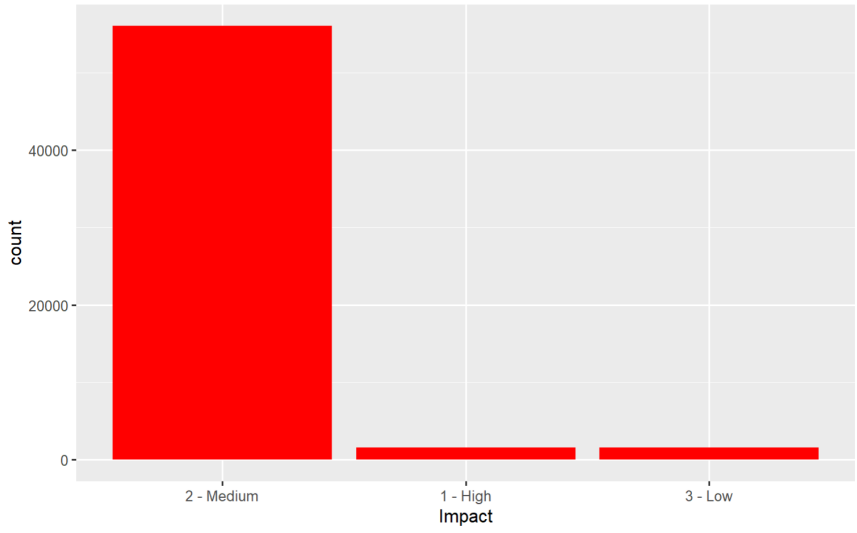


Figure 2- Distribution of Impact attribute in dataset

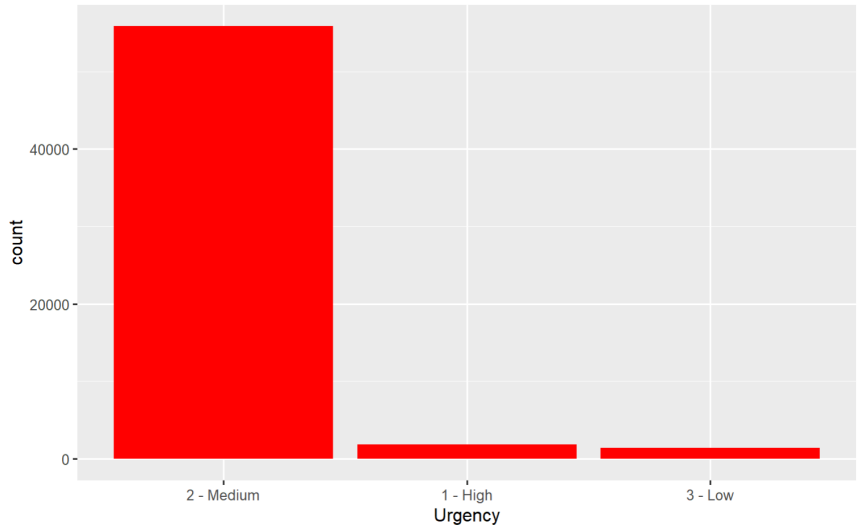


Figure 3- Distribution of Urgency attribute in dataset

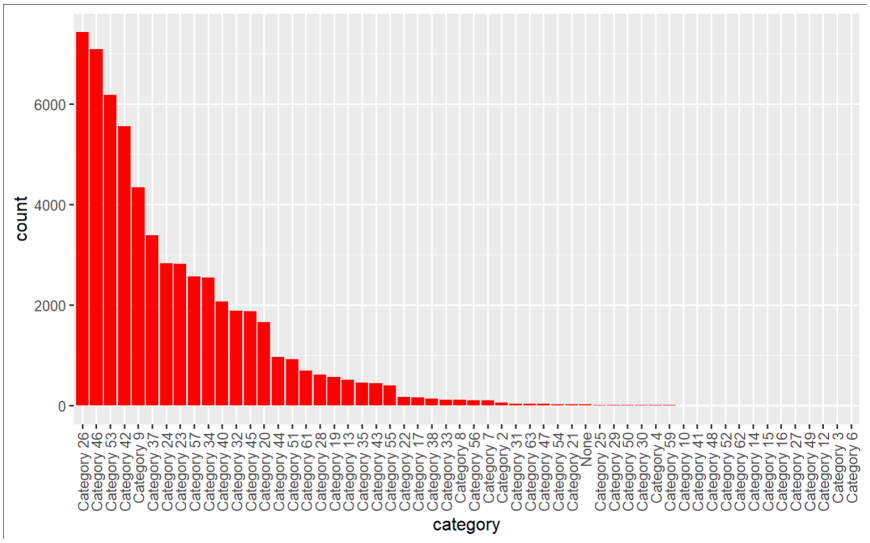


Figure 4- Distribution of Category attribute in dataset

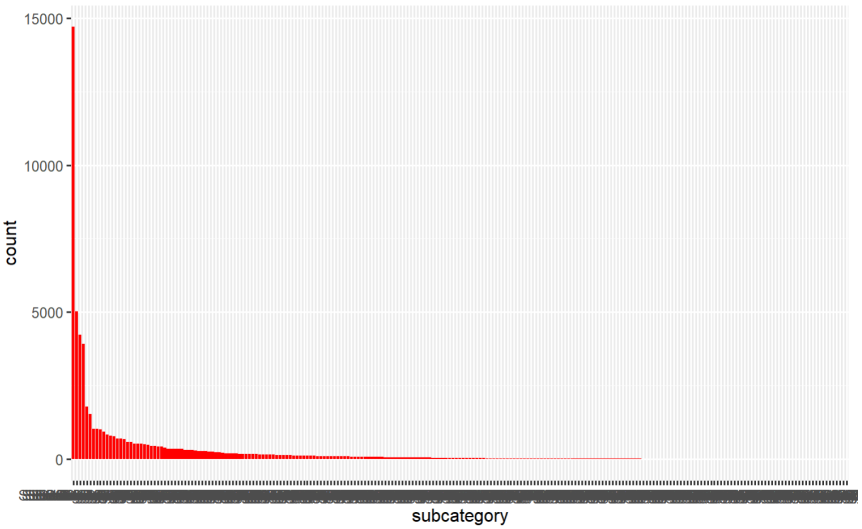


Figure 5- Distribution of Subcategory attribute in dataset

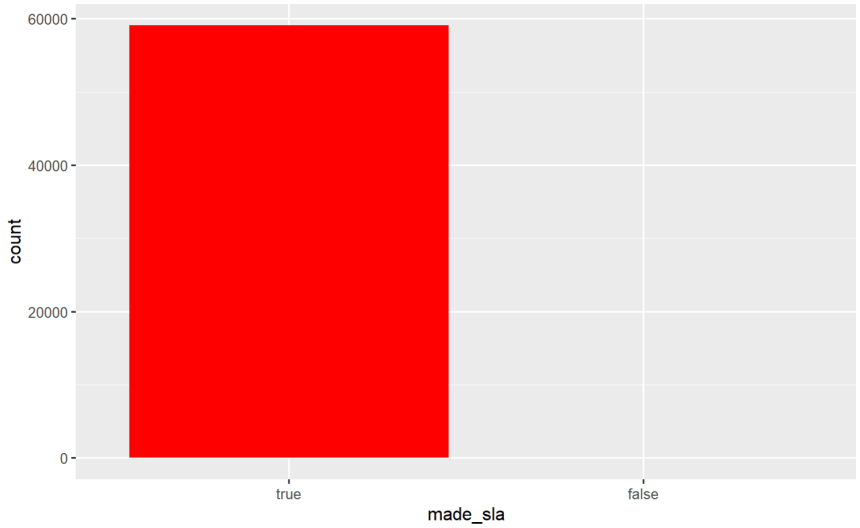


Figure 6- Distribution of made_sla attribute in dataset

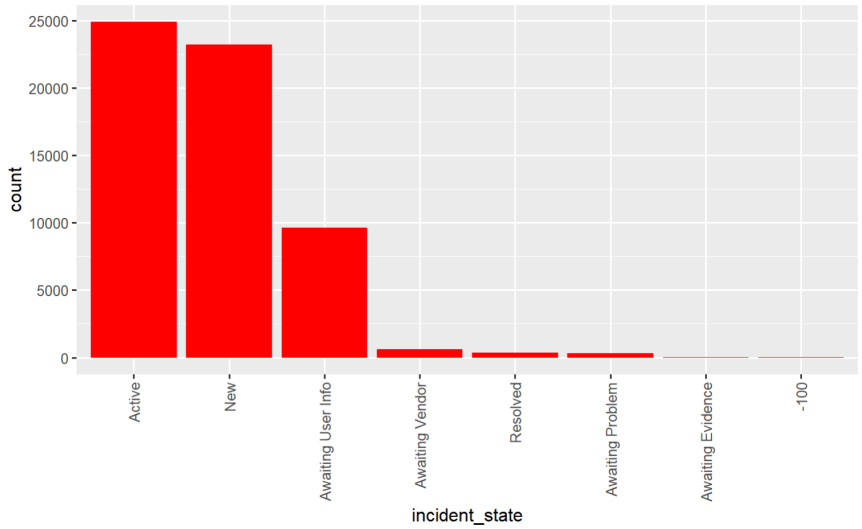


Figure 7- Distribution of incident_state attribute in dataset

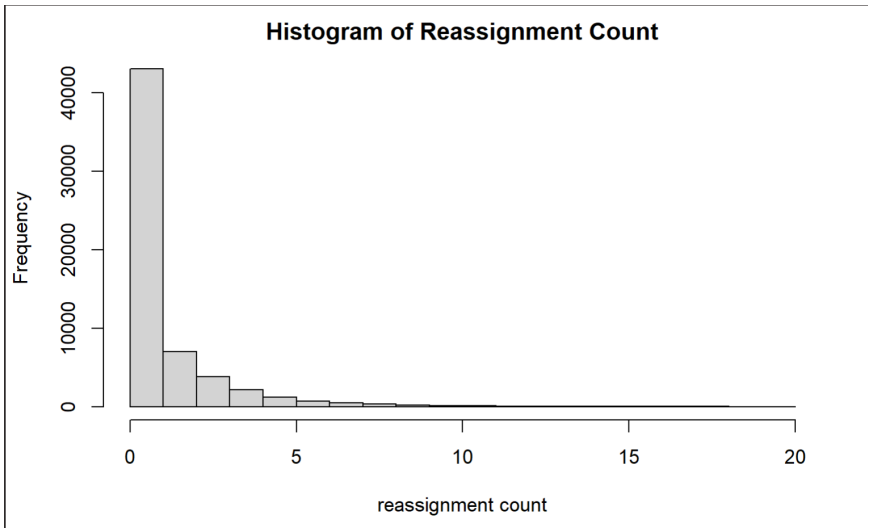


Figure 8- Distribution of reassignment_count attribute in dataset

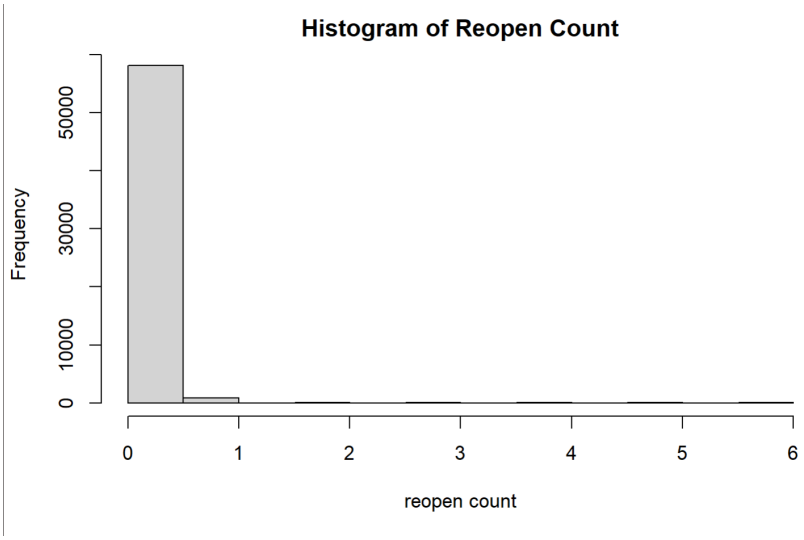


Figure 8- Distribution of reopen_count attribute in dataset

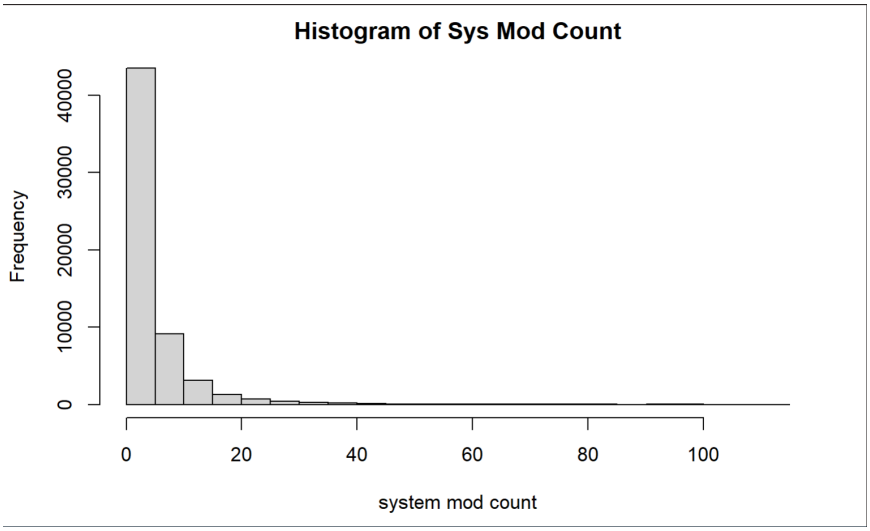


Figure 9- Distribution of sys_mod_count attribute in dataset

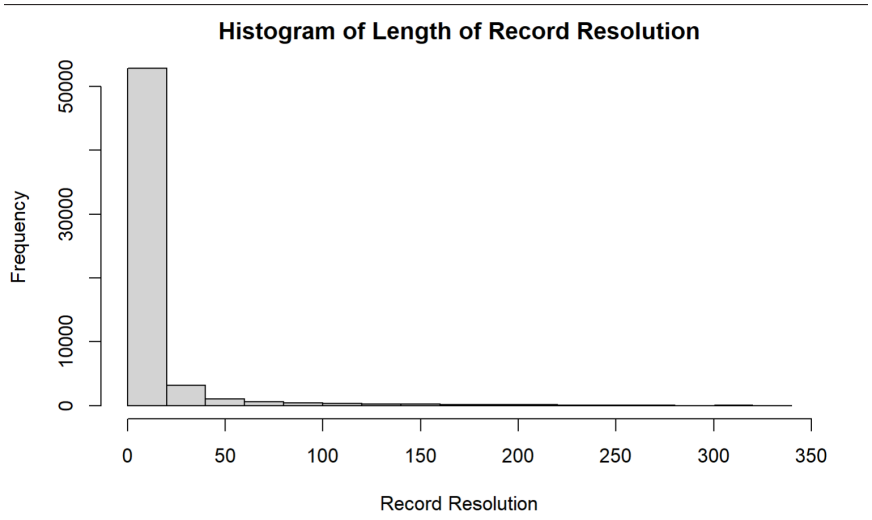


Figure 10- Distribution of Length of Record Resolution attribute in dataset

```
[1] "total number of records where reassignment_count and reopen_count is 0"
[1] 25284
[1] "total number of records where reassignment_count is 0"
[1] 25524
[1] "total number of records where reopen_count is 0"
[1] 58111
[1] "total number of records where urgency and impact are both 2 - Medium"
[1] 55087
[1] "total number of records where urgency is 2 - Medium"
[1] 55848
[1] "total number of records where impact is 2 - Medium"
[1] 56025
```

Figure 11- Displaying values to demonstrate skewed distribution of data

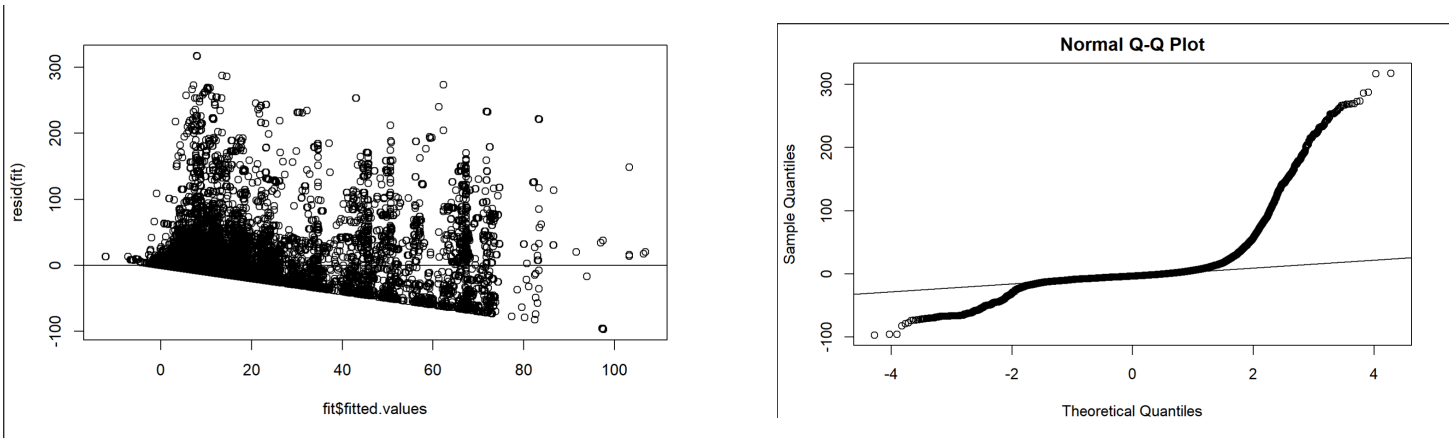
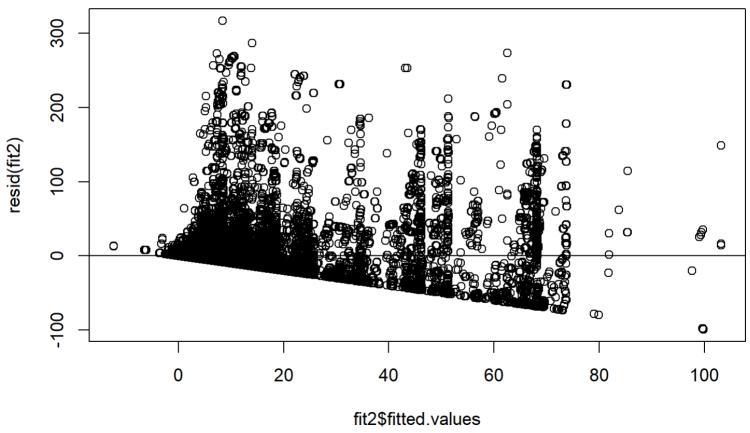


Figure 12 and 13- Residuals vs Fitted values and QQ plot of full regression model

```
Analysis of Variance Table

Response: time_diff_res_update
                     Df   Sum Sq Mean Sq  F value    Pr(>F)
category             55  4554148   82803 155.5818 < 2.2e-16 ***
subcategory         218  2024477    9287  17.4490 < 2.2e-16 ***
incident_state        7    70578   10083  18.9447 < 2.2e-16 ***
reassignment_count    1        0       0   0.0006   0.97969
reopen_count          1       24      24   0.0447   0.83256
sys_mod_count         1    27535   27535  51.7364 6.432e-13 ***
made_sla              1     2869    2869   5.3912   0.02024 *
impact                2    18928    9464  17.7822 1.905e-08 ***
urgency               2    17448    8724  16.3917 7.645e-08 ***
priority              3    26434    8811  16.5558 9.510e-11 ***
Residuals         52888 28147688     532
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

Table 1- Anova table of full model



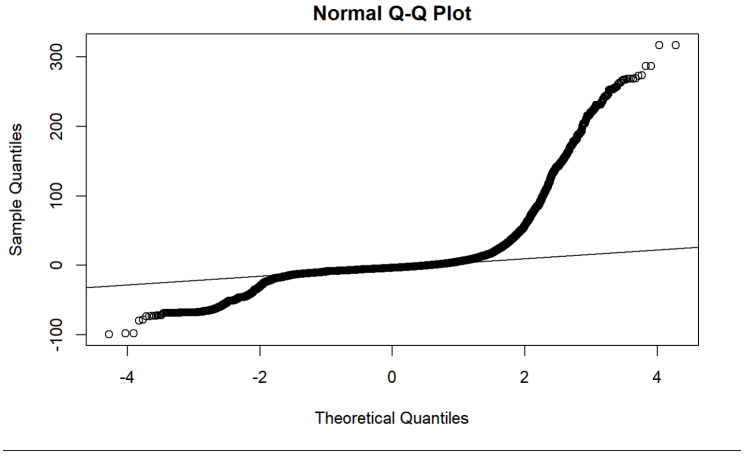
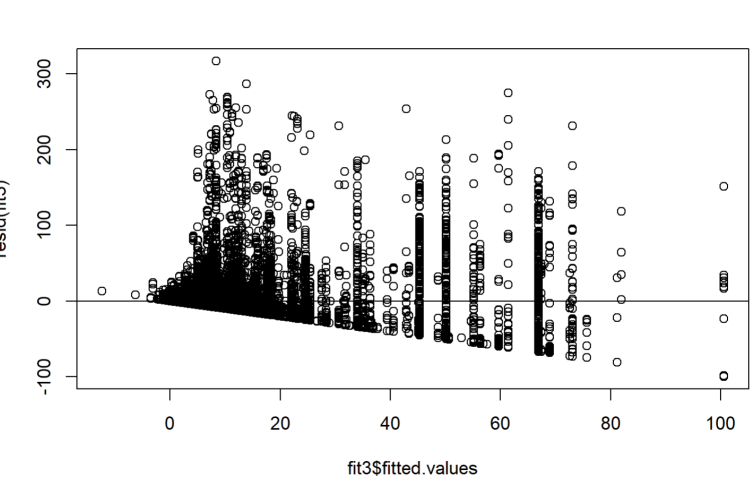


Figure 14 and 15- Residuals vs Fitted values and QQ plot of second regression model

```
Analysis of Variance Table

Response: time_diff_res_update
                     Df   Sum Sq Mean Sq  F value    Pr(>F)
category             55  4554148   82803 155.1350 < 2.2e-16 ***
subcategory         218  2024477    9287  17.3989 < 2.2e-16 ***
reassignment_count    1      388     388   0.7265   0.39402
reopen_count          1       38      38   0.0708   0.79011
sys_mod_count         1    12207   12207  22.8711 1.737e-06 ***
made_sla              1     2595    2595   4.8622   0.02746 *
impact                2    19129    9564  17.9192 1.661e-08 ***
urgency               2    18199    9100  17.0485 3.966e-08 ***
priority              3    26456    8819  16.5223 9.990e-11 ***
Residuals         52895 28232492     534
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

Table 2- Anova table of second model



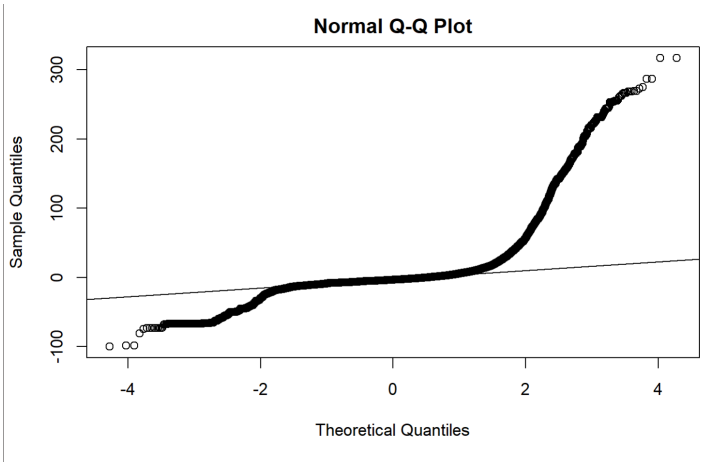


Figure 16 and 17- Residuals vs Fitted values and QQ plot of third regression model

```
Analysis of Variance Table

Response: time_diff_res_update
               Df   Sum Sq Mean Sq F value    Pr(>F)
category       55  4554148   82803 155.065 < 2.2e-16 ***
subcategory   218  2024477    9287  17.391 < 2.2e-16 ***
impact          2    19796    9898  18.536 8.965e-09 ***
urgency         2    18743    9371  17.550 2.403e-08 ***
priority        3    25679    8560  16.030 2.059e-10 ***
Residuals   52899 28247286     534
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

Table 3 - Anova table of third regression model

```
Overall Statistics

               Accuracy : 0.3973
                 95% CI : (0.3848, 0.4099)
    No Information Rate : 0.3599
    P-Value [Acc > NIR] : 1.441e-09

                  Kappa : 0.079

 Mcnemar's Test P-Value : NA
```

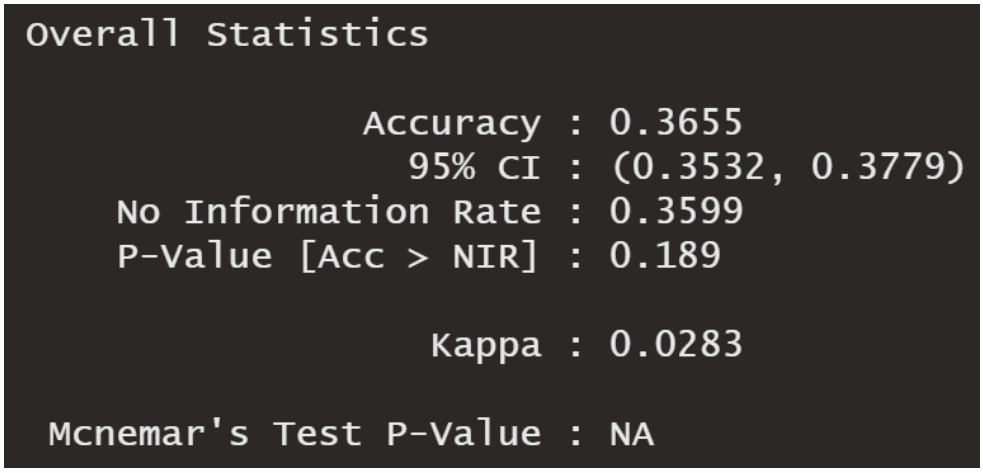
```
Overall Statistics

               Accuracy : 0.3859
                 95% CI : (0.3735, 0.3985)
    No Information Rate : 0.3599
    P-Value [Acc > NIR] : 1.718e-05

                  Kappa : 0.0665

 Mcnemar's Test P-Value : NA
```

Confusion matrix statistics: Left is full and right is second random forest model



Confusion matrix statistics: Third random forest model