Incident Management

by
Sanchana Mohankumar

A Project submitted to a faculty of

Northeastern University

Professor: Dr. Arasu Narayan

17th December 2022

Table of Contents:

Project Overview:

IT incident management process is a complex process and companies need to put a framework for correct assignment of incident tickets to the right resolution group so that it gets resolved as soon as possible ensuring the lowest possible impact on the business and costumers.

The incident management process is used for managing the life cycle of all incidents. Companies spend lots of resources to keep their IT resources incident-free and, therefore, timely resolution of the incoming incident is required to attain that objective. Currently, in the industry, the incident management process is mostly manual rule-based and time-consuming.

There is a huge opportunity for Machine Learning use cases to improve incident management process and scope for automating multiple tasks including incident assignment.

Problem Statement:

The incident management process has been a complex process and companies need to invest money in managing incidents as it plays a major role in minimizing impact to business operations and maintaining quality. The report aims to solve 2 use cases.

Case 1: As we know resolving Incident in allotted SLA time could save millions for a company's business, so in this use case I have analyzed Total time taken by an incident to solve an incident to improve the future Incidents Time management

Data Sources:

This is an event log of an incident management process extracted from data gathered from the audit system of an instance of the ServiceNowTM platform used by an IT company. The event log is enriched with data loaded from a relational database underlying a corresponding process-aware information system. Information was anonymized for privacy.

Number of instances: 141,712 events (24,918 incidents)
Number of attributes: 36 attributes (1 case identifier, 1 state identifier, 32 descriptive attributes, 2 dependent variables)

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

Data Description:

Column	Data Description	Data Type
number	incident identifier (24,918 different values)	object
incident state	eight levels controlling the incident management process transitions from opening until closing the case	object
active	Boolean attribute that shows whether the record is active or closed/canceled	bool
reassignment count	number of times the incident has the group, or the support analysts changed	int64
reopen count	number of times the incident resolution was rejected by the caller	int64
Sys mod count	number of incident updates until that moment	int64
Made sla	Boolean attribute that shows whether the incident exceeded the target SLA	bool
Caller id	identifier of the user affected	object
opened by	identifier of the user who reported the incident	object
opened at	incident user opening date and time	object
Sys created by	identifier of the user who registered the incident	object
Sys created at	incident system creation date and time	object
Sys updated by	identifier of the user who updated the incident and generated the current log record	object
Sys updated at	incident system update date and time	object
contact type	categorical attribute that shows by what means the incident was reported	object
location	identifier of the location of the place affected	object
category	first-level description of the affected service	object
subcategory	second-level description of the affected service (related to the first level description, i.e., to category)	object
U symptom	Description of the user perception about service availability	object
CMDB ci	(Confirmation item) identifier used to report the affected item (not mandatory)	object
impact	Description of the impact caused by the incident	object
urgency	description of the urgency informed by the user for the incident resolution	object
priority	calculated by the system based on 'impact' and 'urgency'	object
assignment group	identifier of the support group in charge of the incident	object
assigned to	identifier of the user in charge of the incident	object
knowledge	Boolean attribute that shows whether a knowledge base document was used to resolve the incident	bool
U priority confirmation	Boolean attribute that shows whether the priority field has been double-checked	bool
notify	categorical attribute that shows whether notifications were generated for the incident	object

Problem id	identifier of the problem associated with the incident	object
rfc	(Request for change) identifier of the change request associated with the incident	object
vendor	identifier of the vendor in charge of the incident	object
Caused by	identifier of the RFC responsible for the incident	object
Close code	identifier of the resolution of the incident	object
Resolved by	identifier of the user who resolved the incident	object
Resolved at	incident user resolution date and time (dependent variable)	object
Closed at	incident user close date and time (dependent variable)	object

Data Statistics:

The purpose of statistics is to make an inference about a population based on a sample. Machine learning is used to make repeatable predictions by finding patterns within data.

Shape: 141712 rows, 36 columns

Duplicates: No Duplicates were found in data

Missing values: There are missing values as '?'. So, in further data cleaning process we will handle missing values

Unique Values: As we can see in Image 1 and 2, these are the number of unique values in each column

Image 1

number	24918
incident_state	9
active	2
reassignment_count	28
reopen_count	9
sys_mod_count	115
made_sla	2
caller_id	5245
opened_by	208
opened_at	19849
sys_created_by	186
sys_created_at	11553
sys_updated_by	846
sys_updated_at	50664
contact_type	5
location	225
category	59

Image 2

subcategory	255
u_symptom	526
cmdb_ci	51
impact	3
urgency	3
priority	4
assignment_group	79
assigned_to	235
knowledge	2
u_priority_confirmation	2
notify	2
problem_id	253
rfc	182
vendor	5
caused_by	4
closed_code	18
resolved_by	217
resolved_at	18506
closed_at	2707

Exploratory Data Analysis:

1.What is the split of Incident state and number of Incidents yet to be closed?
As per Incident state overall we have 38,716 active incidents followed by New Incidents which is 36,407. So, we can say that New Incident minus closed Incidents gives us No of Incidents to be closed which is 11,422

New Incident - closed Incident = 11,422

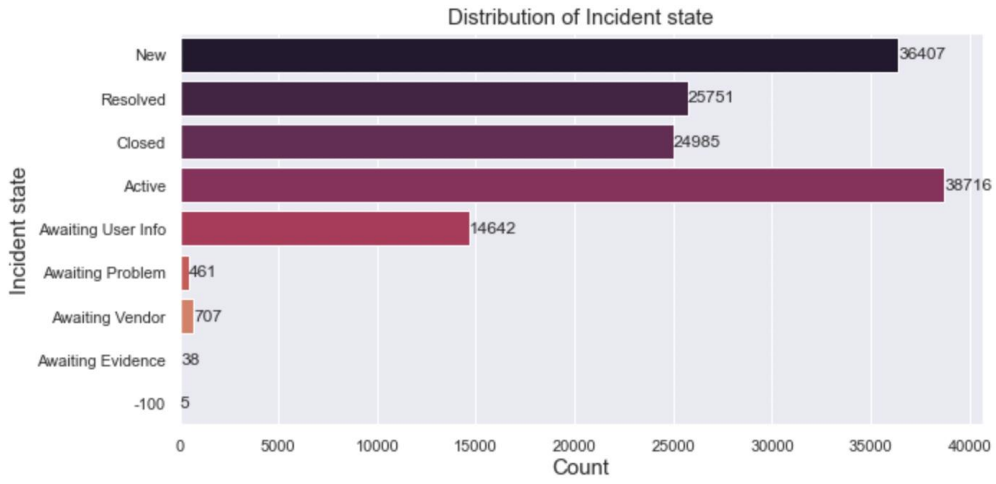


Fig 1: Distribution of Incident state

2. How many tickets have missed the SLA?
We are looking for incident ticket which was closed under scheduled SLA time.

What is SLA?

- If a schedule is not selected for an SLA, the SLA will run 24X7. Consider a scenario where you select a duration of one day, which is 24 hours, and a schedule of 9 am to 5 pm, which is 8 hours. The SLA calculation will distribute the 24 hours across three working days of 8 hours each.

- So, in our case we observe we have closed 15,831 incidents in assigned time whereas 9154 were not made within timeline

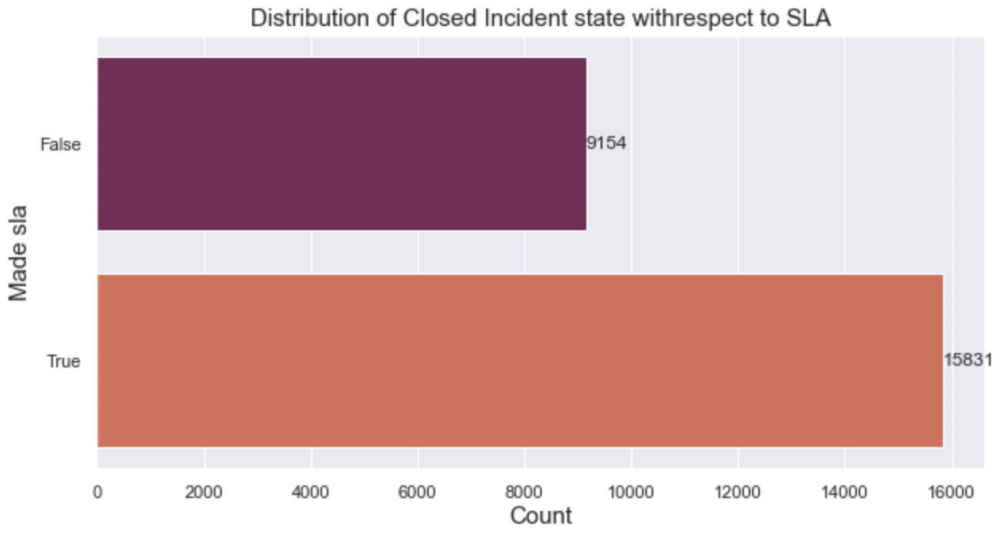


Fig 2: Distribution of I Closed Incident state with respect to SLA

3. What category code is getting closed more?

Column closed code represents each variety of closed code for example we have code 1 it may be case solved permanently, solved remotely etc.

- Over here there are around 14,923 Incidents closed with Code 6 followed by 7, 9, 8

- We don't know what code is exactly here in this case as each company can set n number of code options in our case, we have a total of 18 code verities

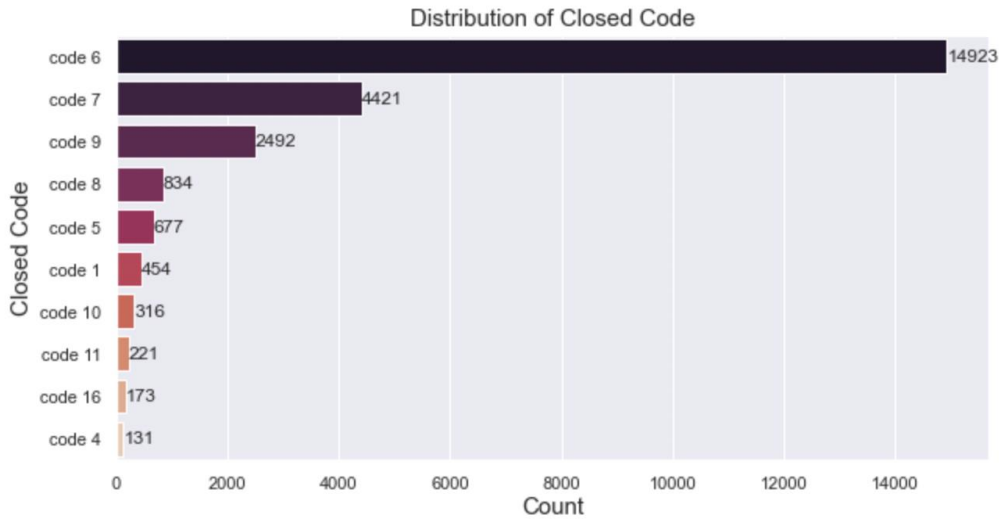


Fig 3: Distribution of Closed Code

4.When to send notification and why?

The notification is sent in case of priority incidents in order to alert the person in charge faster as we can see there are very few Incidents which are created with send email as they were less high priority Incidents. In our case we have around 36 Incidents been notified by email

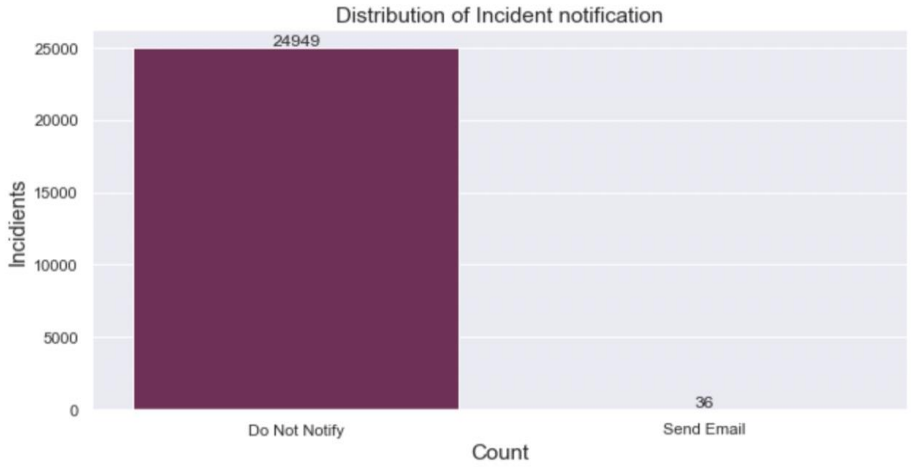


Fig 4: Distribution of Incident Notification

5.Who recorded the maximum number of Incidents?

The maximum number of incidents were opened by Person 17 in order to validate the performance of the employee company can look at the incidents created and observe the knowledge of the person in respective team

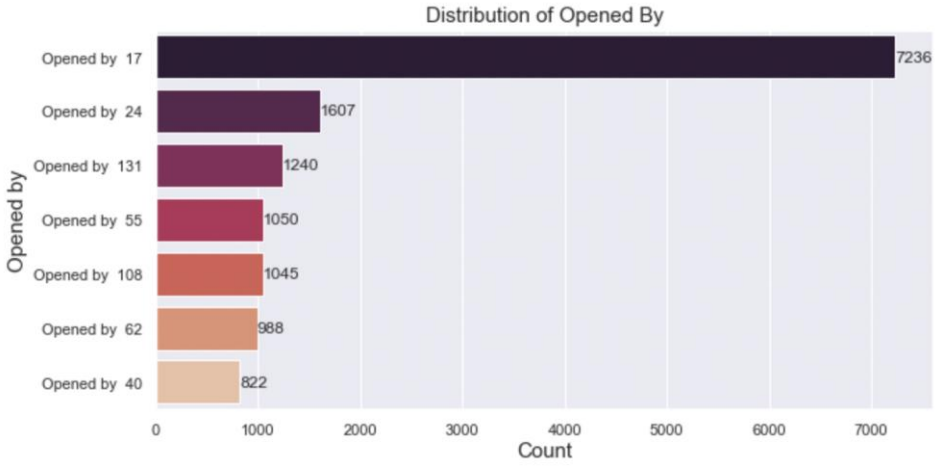


Fig 5: Distribution of Opened By

6. Who resolved the maximum number of Incidents?

The maximum number of incidents were resolved by Person 11 in order the validate the performance of the employee company can look at the incidents resolved

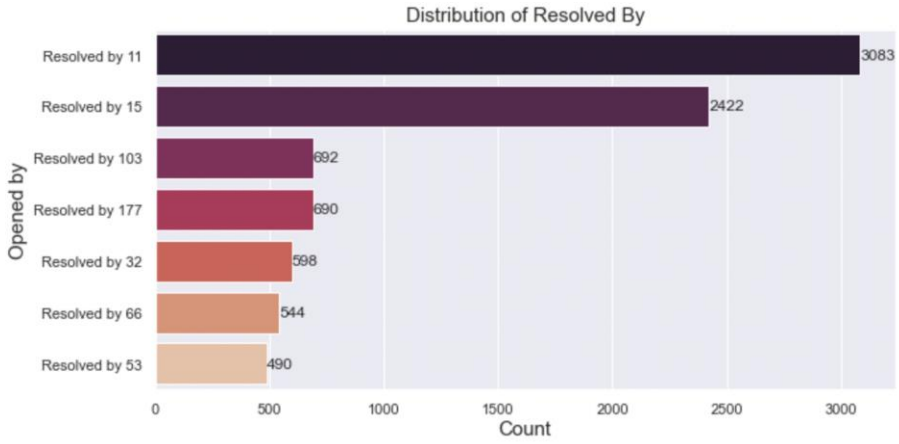


Fig 6: Distribution of Resolved by

7. What date did Incidents opened and closed, and the number of Incidents recorded?

As we can see from figure 7, we have the number of Incidents drastically decreasing after a particular period may be the date with maximum incident means they would have had testing going on in the project whereas in figure 8 we can see with respect to Figure 7 the incidents where also closed at the same time due to SLA constraints but at Date there was lot of incidents getting closed.

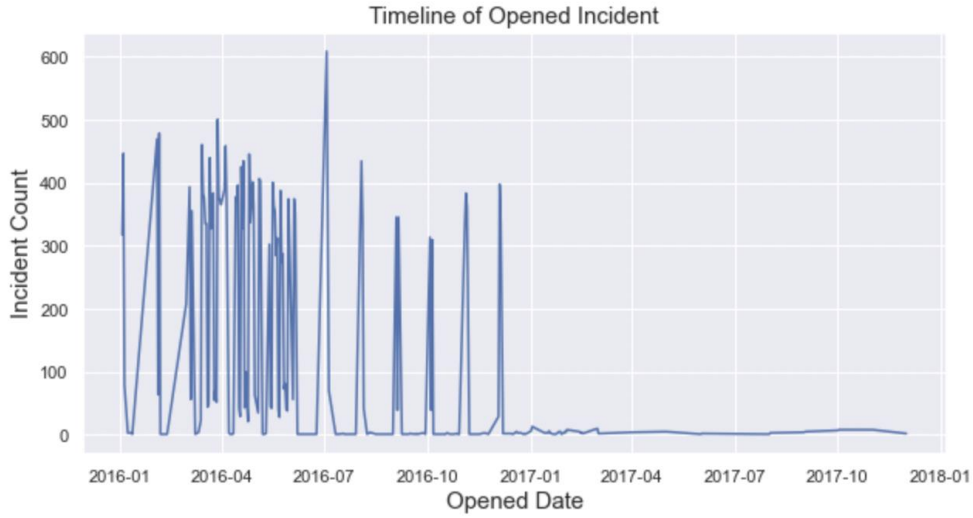


Fig 7: Timeline of Opened Incident

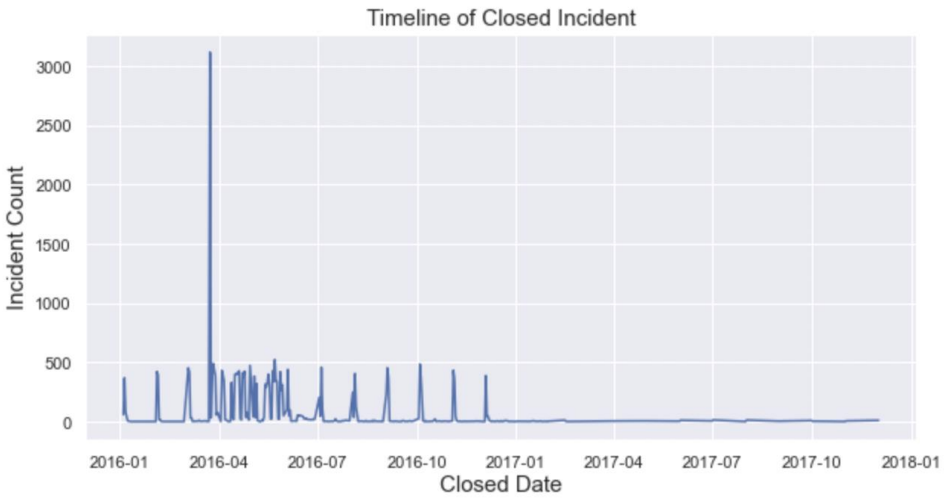


Fig 8: Timeline of closed Incident

Data Preprocessing and Cleaning:

Missing data:

In our data we have missing values as '?' so we are replacing it with nan values after which we calculated the missing values in each column resulting the below table

```
number                          0
incident_state                  0
active                          0
reassignment_count              0
reopen_count                    0
sys_mod_count                   0
made_sla                        0
caller_id                      29
opened_by                    4835
opened_at                       0
sys_created_by              53076
sys_created_at              53076
sys_updated_by                  0
sys_updated_at                  0
contact_type                    0
location                       76
category                       78
subcategory                   111
u_symptom                   32964
cmdb_ci                    141267
impact                          0
urgency                         0
priority                        0
assignment_group            14213
assigned_to                 27496
knowledge                       0
u_priority_confirmation         0
notify                          0
problem_id                 139417
rfc                        140721
vendor                     141468
caused_by                  141689
closed_code                   714
resolved_by                   226
resolved_at                  3141
closed_at                       0
```

As we can see column problem_id, rfc, vendor, caused_by, cmdb_ci have huge missing values we are dropping those columns and remaining rows of missing values columns resulting in the below table with an overall data of 53599 rows and 31 columns

```
number                    0
incident_state            0
active                    0
reassignment_count        0
reopen_count              0
sys_mod_count             0
made_sla                  0
caller_id                 0
opened_by                 0
opened_at                 0
sys_created_by            0
sys_created_at            0
sys_updated_by            0
sys_updated_at            0
contact_type              0
location                  0
category                  0
subcategory               0
u_symptom                 0
impact                    0
urgency                   0
priority                  0
assignment_group          0
assigned_to               0
knowledge                 0
u_priority_confirmation   0
notify                    0
closed_code               0
resolved_by               0
resolved_at               0
closed_at                 0
```

Duplicates data:

No Duplicates were found

Extraction of numeric values from string:

Extracted numeric values from the following columns caller_id, opened_by, sys_created_by, sys_updated_by, location, category, subcategory, u_symptom, impact, urgency, priority, assignment_group, assigned to, closed_code. An example is given below

Table 1: Before extracting

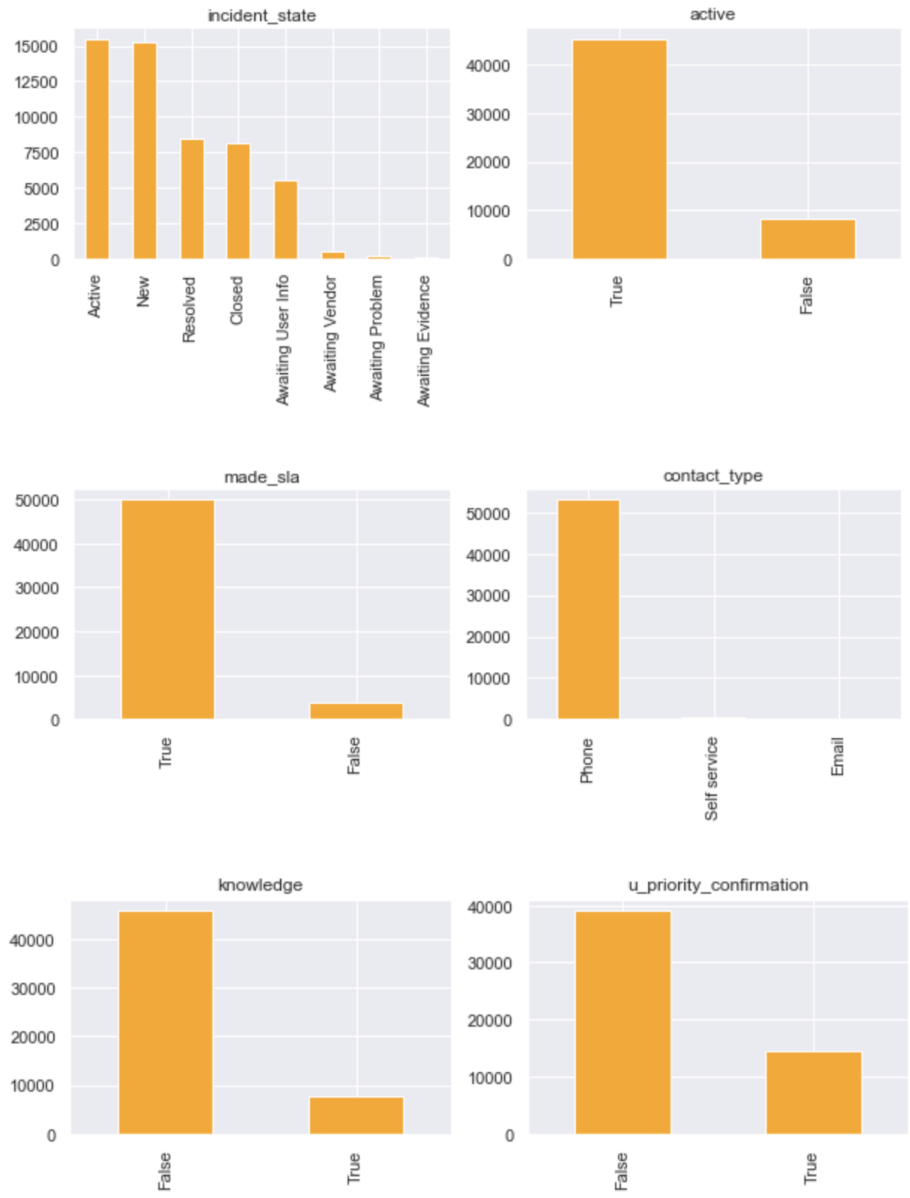
	number	incident_state	active	made_sla	caller_id	opened_by	opened_at	sys_created_by	sys_created_at	sys_updated_by	...	priority
4	INC0000047	New	True	True	Caller 2403	Opened by 397	29/2/2016 04:40	Created by 171	29/2/2016 04:57	Updated by 746	...	3 - Moderate
5	INC0000047	Active	True	True	Caller 2403	Opened by 397	29/2/2016 04:40	Created by 171	29/2/2016 04:57	Updated by 21	...	3 - Moderate

Table 2: After extracting

	number	incident_state	active	reassignment_count	reopen_count	sys_mod_count	made_sla	caller_id	opened_by	opened_at	...	priority	assignment_grou
4	0000047	New	True	0	0	0	True	2403	397	29/2/2016 04:40	...	3	7(
5	0000047	Active	True	1	0	1	True	2403	397	29/2/2016 04:40	...	3	2

Columns to Encode:

Below are the columns with categorical features. Furthermore, we are going to encode the below predictor variables using Label Encoding



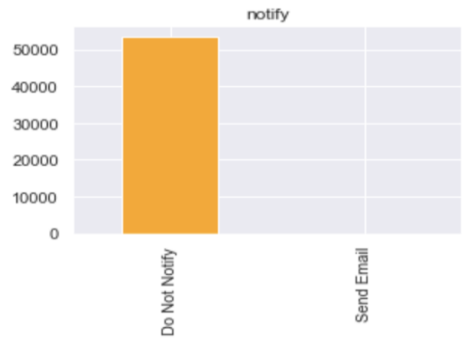


Fig 9: Distribution of Categorical column data

The below table displayed data is after Label Encoding the data

	number	incident_state	active	reassignment_count	reopen_count	sys_mod_count	made_sla	caller_id	opened_by	opened_at	...	priority
4	0000047	6	1	0	0	0	1	2403	397	1.456721e+09	...	3
5	0000047	0	1	1	0	1	1	2403	397	1.456721e+09	...	3
6	0000047	0	1	1	0	2	1	2403	397	1.456721e+09	...	3
7	0000047	0	1	1	0	3	1	2403	397	1.456721e+09	...	3
8	0000047	0	1	1	0	4	1	2403	397	1.456721e+09	...	3

Column Drop

1. Column - Problem_id, rfc, vendor, caused by, cmdb_ci removed following column as had large amount of missing values
2. Column - impact, notify, urgency removed as it does not contribute much to our use case
3. Column - resolved at as per given data resolved at is highly correlated with closed at so due to collinearity issue, we removed the column

Computed Column Analysis

Incident opened at time

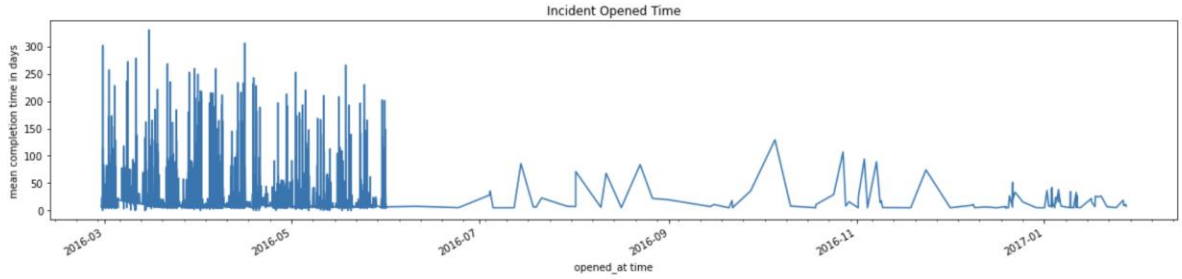


Fig 10: Incident opened time with respect to mean completion days

Incident closed at time

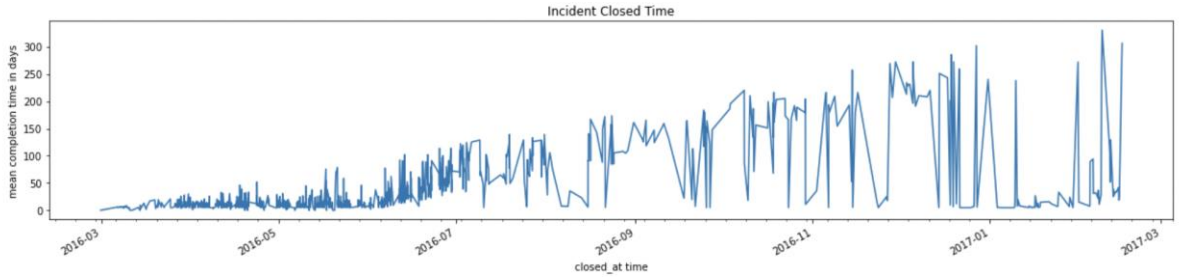


Fig 11: Incident closed time with respect to mean completion days

Figures 11 and 10 in the above plots display the incident's open and closed times relative to mean completion days. In the year 2016, we can observe a surge in the number of open incidents around the months of March and May, and a significant amount of incident closures around the months of January and March in the year 2017

Mean completion time in Day, Month and Year grouping by Open Year

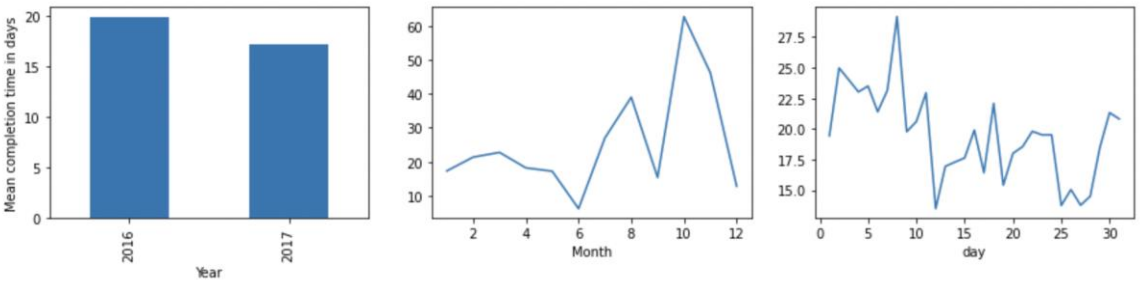


Fig 12: Incident Mean Completion time with respect to Day, Month and Year of Open Year

Mean completion time in Day, Month and Year grouping by closed Year

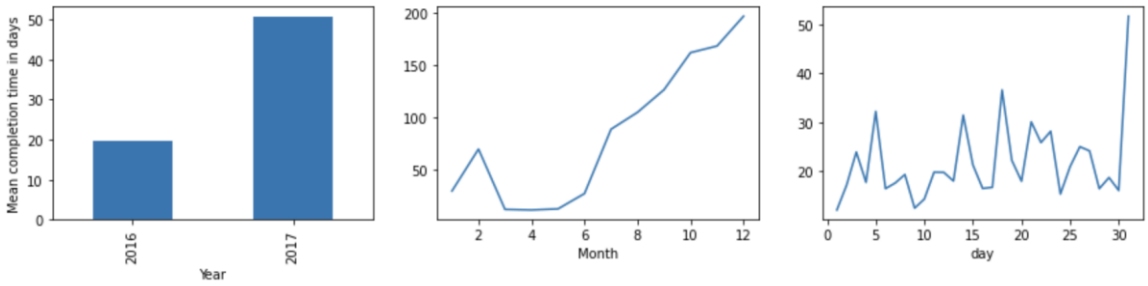


Fig 13: Incident Mean Completion time with respect to Day, Month and Year of Close Year

Figures 12 and 13 in the above plots display the incident's open and closed times relative to mean completion days. In the year 2016, we can observe more open incidents around the months of October to December, In the year 2017, we can observe more closed incidents around the months of June to December

completion_time_days	open_month	open_year	open_day	open_week_day	closed_month	closed_year	closed_day	closed_week_day
6.222222	2	2016	29	0	3	2016	6	6
6.222222	2	2016	29	0	3	2016	6	6
6.222222	2	2016	29	0	3	2016	6	6

Table 3: Completion time in days – Computed column display

In the above table we can observe that Completion_time_days column is computed from Open and Closed at Date and Time

Methodology:

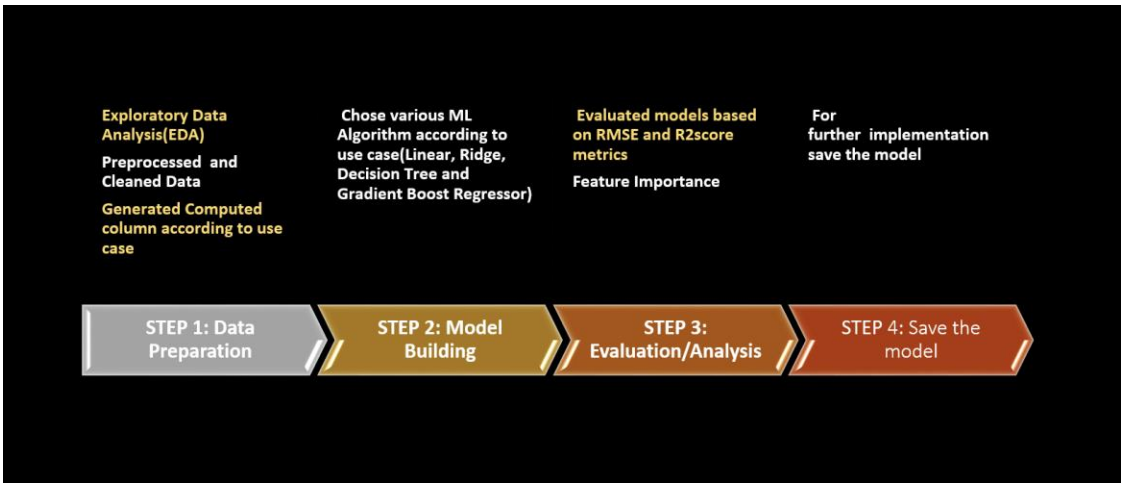


Image 1: Methodology of incident Management Regression Model Implementation

Methodology of our Project

STEP 1:

In my project after collecting data, I analyzed the data from all perspectives with respect to target variable. While analyzing the data I came across various use cases among which I opted for analyzing the completion time of the incidents in order to improve the business with respect to profits.

EDA: For analyzing data I plotted various graphs to derive insights with respect to target column and observed the distribution of categorical and numerical columns and its importance.

Data Cleaning and Data Preprocessing: In this step of the project, I removed few columns due to the large number of null values and removed a few rows too as they didn't contribute much to target column. There was various column with string and numeric attached, so I removed the numeric value for computation

Computed Column: As our target column was in Date and Time which we know we can't implement models; I derived a computed column names Completion time in days from Opened and closed Date which I have used in this project to implement various ML models

STEP 2:

In my project I have implemented Linear Regression, Ridge Regression, Lasso Regression, Decision Tree Regression, Gradient Boost Regression

Model: We first implemented a linear regression model, followed by a ridge regression, to improve model performance in case our data had any multicollinearity issues, as seen from the data by deriving different information and insights. We have used Lasso regression to assess the model's feature importance.

We constructed a decision tree to further improve the model, and to strengthen the decision tree we used a gradient boost regressor because it is effective with linear data.

Feature Importance: For our data we have analyzed Feature Importance through Lasso regression as Lasso regression specializes in feature selection

STEP 3:

Model Evaluation: In my project Iam analyzing Model performance using RMSE (Root Mean square Deviation) and R2score.

The RMSE is the square root of the variance of the residuals. It indicates the absolute fit of the model to the data–how close the observed data points are to the model's predicted values. Whereas R-squared is a relative measure of fit, RMSE is an absolute measure of fit.

STEP 4:

Save the Model: After our model is executed, we can save our model for further implementation in various platform

Model Training and Evaluation:

Model Output:

Model	RMSE	R2 Score
Linear Regression	19.33	0.67
Ridge Regression	19.28	0.67
Lasso Regression	0.61	0.61
Decision Tree	13.98	0.82
Gradient Boost Regressor	11.20	0.89

For our project dataset we have chosen Linear, Ridge, Decision Tree and Ada boost regression methods to predict the closed date and time of our incidents our data

Linear Regression:

As we know the key assumptions for Linear regression model is

1. Linear Relationship
2. Independence
3. Normally distributed

We proceeded with this model because these three key presumptions were met, and consequently, we obtained a model R2score of 0.67 and an RMSE error value of 19.33. This results in poor model performance; therefore, we may utilize regularization techniques like Ridge Regression and Lasso Regression to improve it. Let's think about Ridge Regression in our situation

Ridge Regression:

As we know Ridge regression is a model tuning method used in case of multicollinearity. From the table below we can observe that there are a few columns with high VIF issue has multicollinearity issue. A VIF less than 5 indicates a low correlation of that predictor with other predictors. A value between 5 and 10 indicates a moderate correlation, while VIF values larger than 10 are a sign for high, not tolerable correlation of model predictors

But from the result of R2score we can observe that there isn't much change in model performance even though there is light decrease in error. So further Iam checking the Feature importance using Lasso Regression model

	features	vif_Factor
0	category	7.322765
1	subcategory	7.678307
2	priority	29.696127
3	caller_id	3.728979
4	made_sla	13.505739
5	open_month	21.592281
6	open_week_day	2.549125
7	closed_month	15.303791
8	closed_week_day	2.657782

Table 4: VIF Output

Lasso Regression:

In the previous model we observe that there is not much improvement in model performance. As we Lasso regression is used for eliminating automated variables and the selection of features, we have used it to identify important feature for the improving model performance

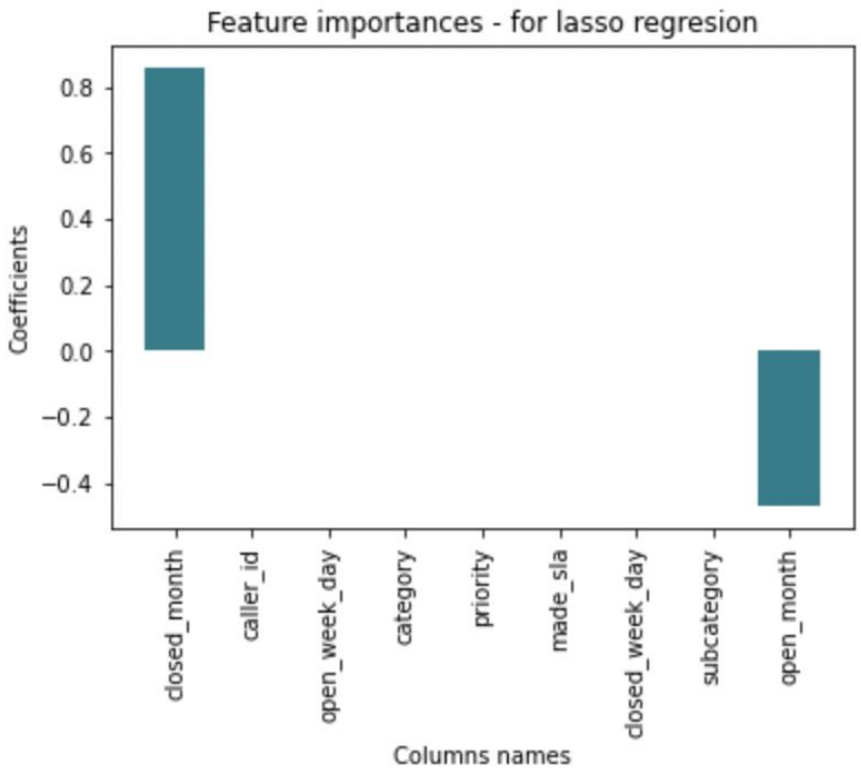


Fig 14: Lasso regression Feature Importance

Since open month is negative and closed month contributes more to the model than all other columns, as seen in the above plot, we will further exclude open month from the models' consideration.

Decision Tree:

As we saw in Ridge regression, we concluded our data doesn't have more multicollinearity issue so as we know one of the assumptions of decision tree is data shouldn't have multicollinearity issue and few other assumptions are listed below

1. Linear relationship
2. No Autocorrelation
3. No Multicollinearity
4. Error term normal distribution

After implementing the above model, we can see a significant increase in model performance. Our R2score has increased from 0.67 to 0.82 even our errors have decreased. As we observe even decision tree does not perform well with Linear data there are exception cases and also,

they do perform well with linear data also we can improve model performance by boosting Technique further we will implement Gradient Boost Regressor for our data

Gradient Boost Regressor

As we know Gradient Boost regressor is used as an estimator that builds an additive model in a forward stage-wise fashion; it allows for the optimization of arbitrary differentiable loss functions. In each stage a regression tree is fit on the negative gradient of the given loss function.

For our project we have used Gradient Boost Regressor as a boosting Technique to improvise Decision Tree performance. As we know we can improve model performance by tuning various hyper parameter like max_depth below graph depicts the improvement in model performance as the increase in error rate is drastic

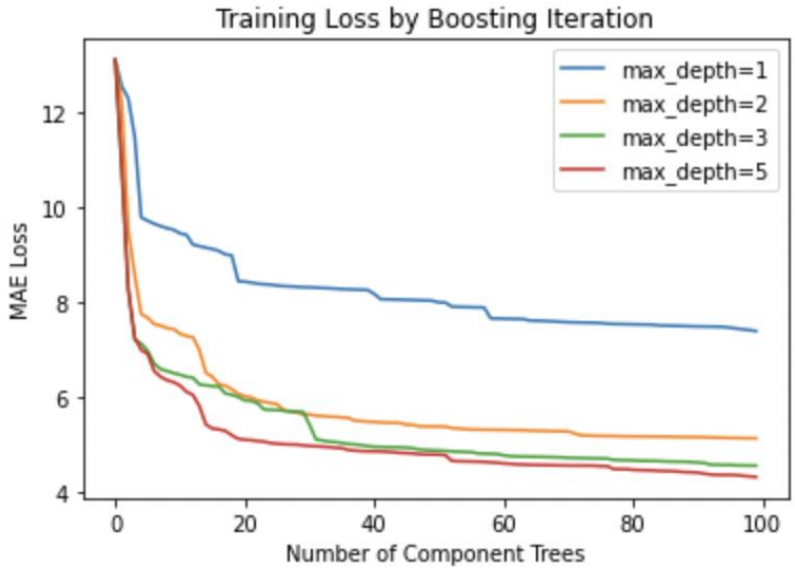


Fig 15: Training Loss by Tuning max depth hyperparameter

For our model we have considered max_depth 5 as it showed significant decrease in RMSE value and improvement in R2 score which is 11.20 and 0.89

Conclusion:

We can conclude from the above models Gradient Boost Regressor is giving us good performance compared to other models. So, we will implement Gradient Boost Regressor to predict the future incidents time taken to resolve the Incident. Further, we will be able to implement the model to reduce the time taken in order to improve Profit for the Business.

Further this will be useful as currently all companies use different kinds of applications like service now as it works with many legacy systems for smooth integration.

Future Work:

Further in future we can improvise the model by adding more relevant features containing text data and analyze using NLP Models. For example, in the Incidents the reason for this incident raised column is available in actual service now so we can incorporate those features.

GitHub Link:

https://github.com/Sanchana1997/Incident_Management_Machine_Learning

Reference:

Referred Professor Modules for code