Incident management process enriched event log

Submitted towards partial fulfillment of the criteria for award of PGPDSE by Great Lakes Institute of Management

Submitted By
Group No. 8 [Batch: 2023]

Group Members

1. **Sreeja Maddika**
2. **Krutik Morker**
3. **Manikantha Brahmaji Korthurthi**
4. **Kunal Gurung**

Research Supervisor
Aishwarya Sarda

Name and signature of the team leader
MADDIKA SREEJA REDDY

Signature of the mentor

ACKNOWLEDGEMENT

<<Acknowledgements and self-attested certificate of originality and authenticity like "*We certify that the work done by us for conceptualizing and completing this project is original and authentic*".>>

Date: 18 January 2024

Place: Bengaluru

TABLE OF CONTENTS:

LIST OF TABLES AND GRAPHS:

ABBREVIATIONS:

- ITIL: Information Technology Infrastructure Library

- ITSM: IT Service Management
- SLA- Service Level Agreement
- SMOTE: Synthetic Minority Over-sampling Technique, is an algorithmic approach in machine learning that generates synthetic instances of the minority class to balance class distribution, particularly useful in addressing imbalanced datasets.
- Under-Sampling: Undersampling is a technique in machine learning where instances from the majority class are randomly removed to address the class imbalance, helping to create a more balanced dataset for training models.
- make_classification:This function is a utility provided by scikit-learn. This function facilitates the generation of synthetic datasets tailored for classification tasks. Users have the flexibility to customize various aspects of the generated dataset, such as the number of samples, features, and the distribution of classes. This functionality proves useful for creating artificial datasets to explore and evaluate the performance of classification algorithms in a controlled and customizable environment.

EXECUTIVE SUMMARY:

A comprehensive industry review highlights the significance of incident resolution for customer satisfaction. Current practices emphasize ITIL/ITSM frameworks, hybrid approaches, and collaboration for efficient service management. A literature survey explores the integration of ITSM, ITIL, and Business Continuity, while ongoing research aims to optimize ITIL implementation. The dataset, comprising 36 columns and 141,712 rows, undergoes preprocessing, EDA, and base model analysis using XGBoost. The base model reveals bias errors, necessitating further steps to address imbalances and enhance model performance.

CHAPTER 1 – OVERVIEW:

The dataset domain revolves around IT incident management, involving the identification, categorization, prioritization, response, and closure of incidents during the development, deployment, or maintenance of IT services and applications. The dataset includes both numeric and categorical columns, covering information such as incident states, user identifiers, timestamps, locations, impact, urgency, priority, and boolean indicators like SLA adherence and knowledge base usage.

Business Problem Understanding:

The primary focus of this project is to address the challenges in incident management within the IT industry, Particularly in the context of application development and service delivery. The problem revolves around efficiently resolving incidents, ensuring customer satisfaction, and optimizing the overall performance of applications. The key challenge lies in effectively managing incidents during the application development lifecycle. Swift incident resolution is crucial for customer satisfaction, and the project aims to enhance incident management practices, prioritization strategies, and response methods. Additionally, the imbalance in incident priority levels poses a classification problem, where the objective is to predict and prioritize incidents for

optimal resolution.

Business Objective:

The objective is to train a model that accurately classifies the incident prioritization.

Approach:

We use the supervised classification techniques to predict the target variable which is multi class in nature , ie. Moderate, poor, high, critical.

Conclusions :

The problem identified here is definitely one worth addressing and an accurate machine-learning algorithm. It offers valuable insights that can guide the company in making informed decisions.

CHAPTER 2 - STEP -BY-STEP WALK THROUGH OF THE SOLUTION

1. **Define the goal**:

   This is a report evaluating the priority of incident by customer services

2. **Get and understand the data:**

   - The data focuses on incident management in the IT domain; key features include incident identifiers, states, timestamps, user details, location, impact, urgency, priority, and Boolean indicators. The data reflects the process of identifying, categorizing, prioritizing, and resolving incidents in IT services. It displays diverse incident scenarios, offering insights into the effectiveness of incident management frameworks like ITIL and ITSM. Exploration reveals class imbalances, outliers, and multicollinearity challenges. The dataset's structured format supports efficient analysis for optimizing incident resolution, reducing downtime, and enhancing overall IT service performance.
   - The dataset captures the various states of incidents, shedding light on the lifecycle of IT-related issues.

3. **Pre-process the data including cleaning, transform the data:**

   Pre-Processing Data Analysis (count of missing/ null values, redundant columns, etc.)

Step 1: Remove duplicate or irrelevant observations:

There are no duplicates. We need to remove the variable, “id” which does not add any value in our analysis.

Step 2: Fix structural errors:

There are no structural errors such as dates in different formats or any word appearing with different spelling.

Step 3: Transform data:

Converted these categorical variables to numerical:

number', 'incident_state', 'caller_id', 'opened_by', 'sys_created_by'', 'sys_updated_by', 'sys_updated_at' 'contact_type', 'location', 'category', 'subcategory, 'u_symptom', 'cmdb_ci', 'impact', 'urgency', 'priority', 'assignment_group', 'assigned_to', 'notify', 'problem_id', 'rfc', 'vendor', 'caused_by', 'closed_code', 'resolve

Step 4: Handle missing data:

There are missing values; we need to impute them by using a suitable technique.

Imputation of missing values through Multiple Imputation by Chained Equation:

- Detecting and handling missing values in the correct way is important, as they can affect the results of the analysis. It cannot be imputed with general ways of using mean, mode, or median which ignores the inherent relationship among data and it can pollute the data.
- We observe that on a few occasions, data is missing in a dataset and is related to the other features and hence they can be predicted using other feature values. Imputing by prediction of missing values is superior to other techniques since the inherent relationship among data is not ignored.
- We are imputing missing numeric values using chain imputation.
- There are no missing values after imputing and we need to use this dataset for EDA and model building.

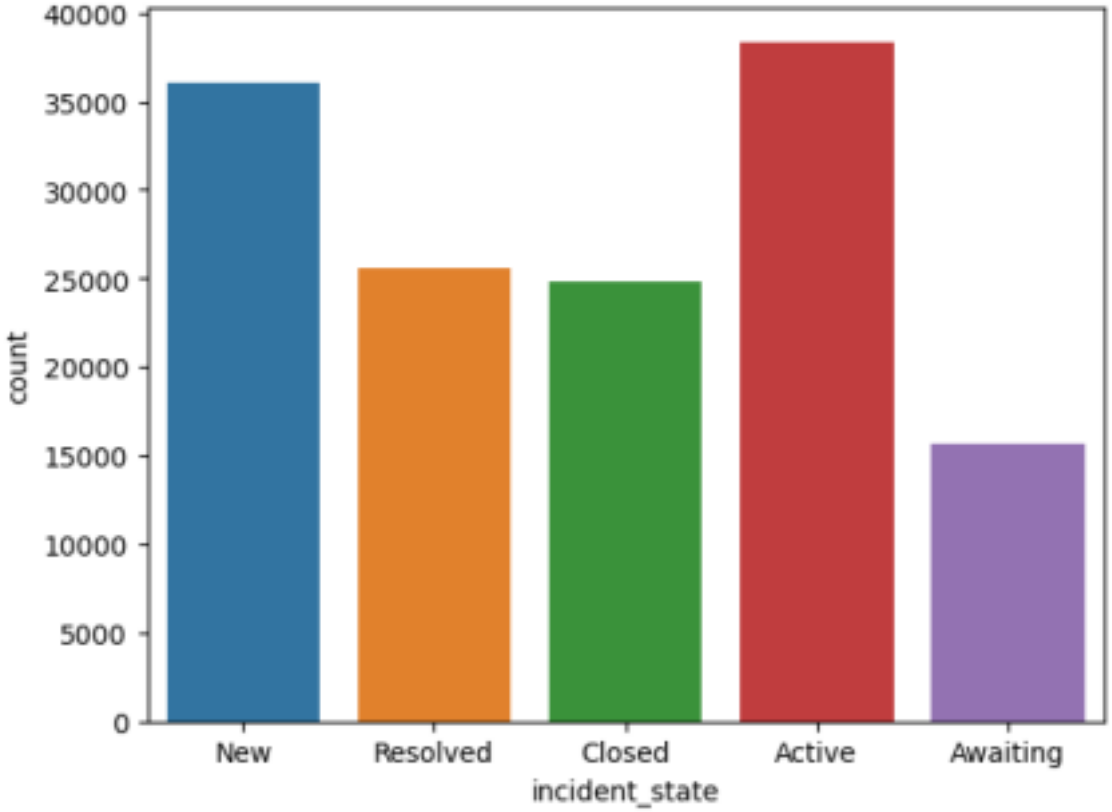
Step 5: Filter unwanted outliers:

We need to find numerical variables and check if there are any outliers. If an outlier is a natural part of the population you are studying, you should not remove or impute it.

4. Exploratory Data Analysis:

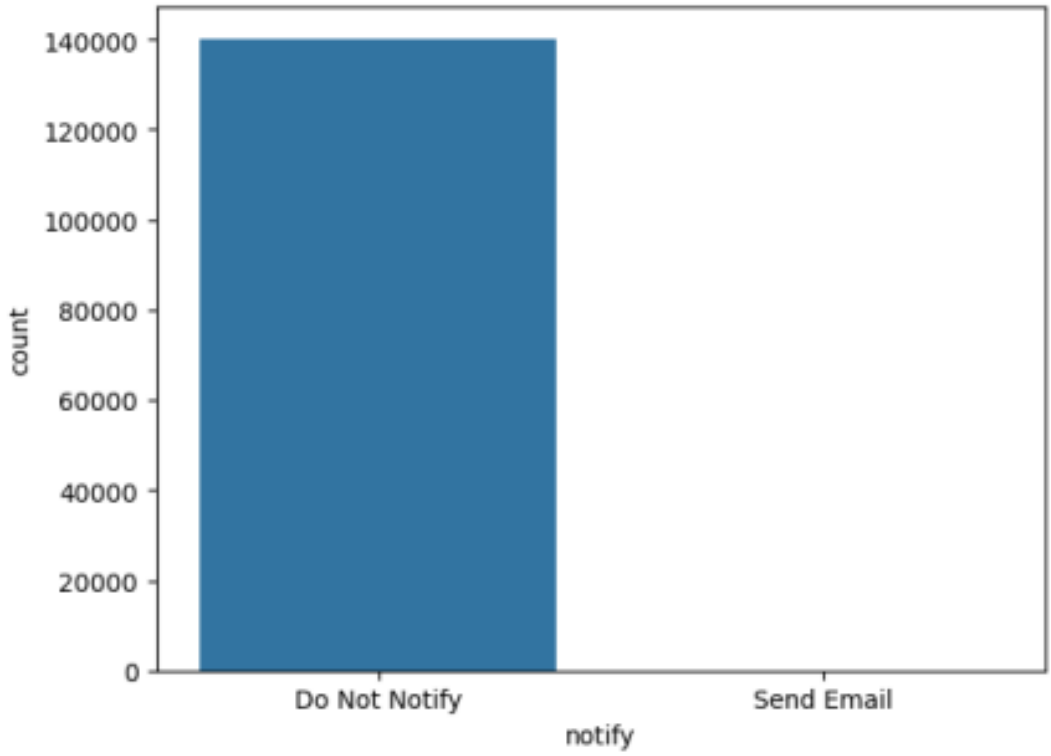
Univariate:

- **categorical column ‘incident_state’ count plot - indicating the ratio of state of incident.**



Interpretation: the above graph in incident_state shows the state of the incident in the process of solving. Most of the incidents that are coming are new or active this indicates there are incidents always coming into the service and most of them are working actively.

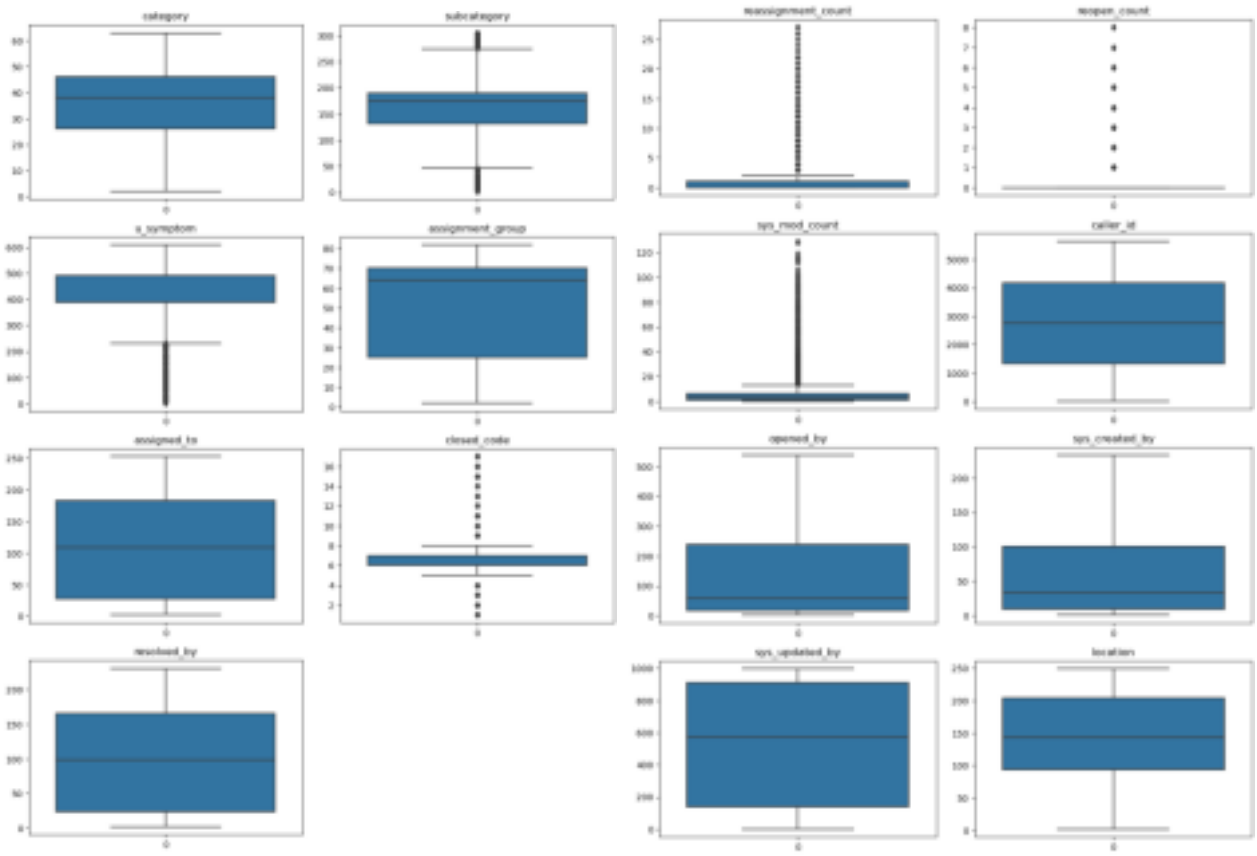
- **categorical column notify: completely biased to not notify**



Interpretation: The above variable notify indicates that the notification is generated to the incident and

or not and all of them from this dataset do not notify at all.

- **numeric :**

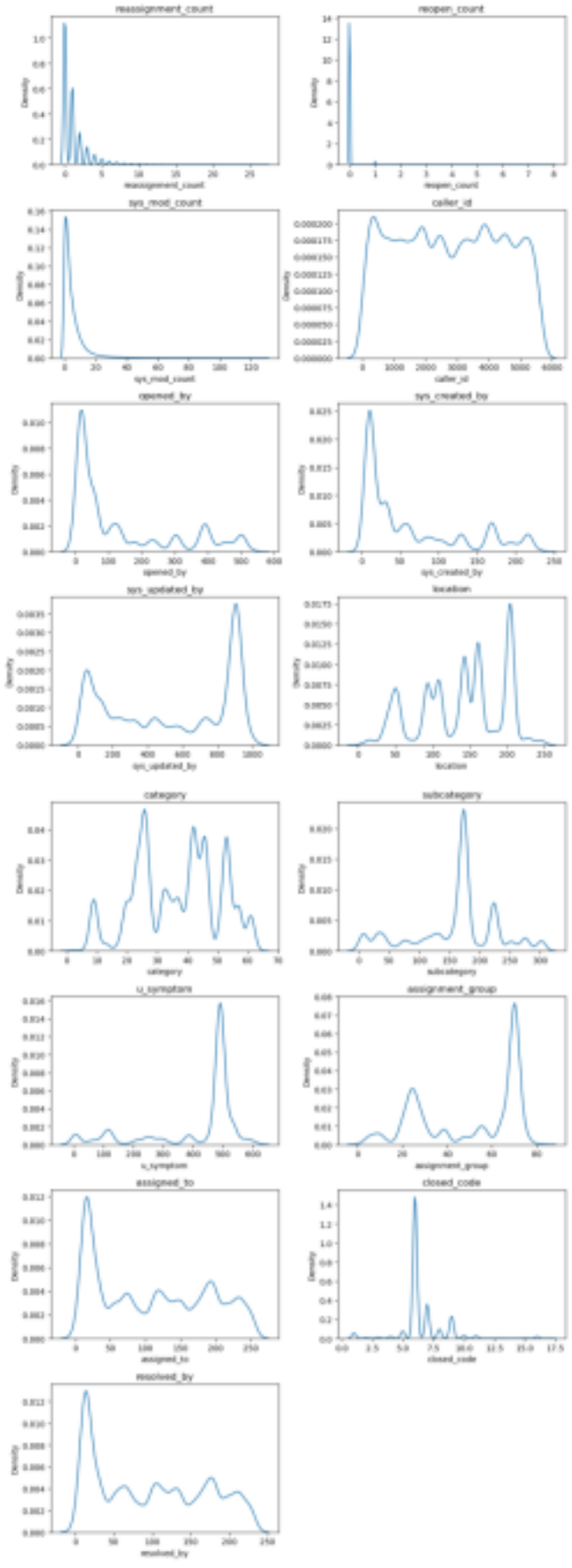


Interpretation:

There are visible outliers in closed_code, u_symptom,,subcategory, sys_mod_count, reassignment_count.

For a clearer picture of skewness and distribution, we look into the KDE plot.

- **Kde plot showing the distribution and skewness of numeric variables:**

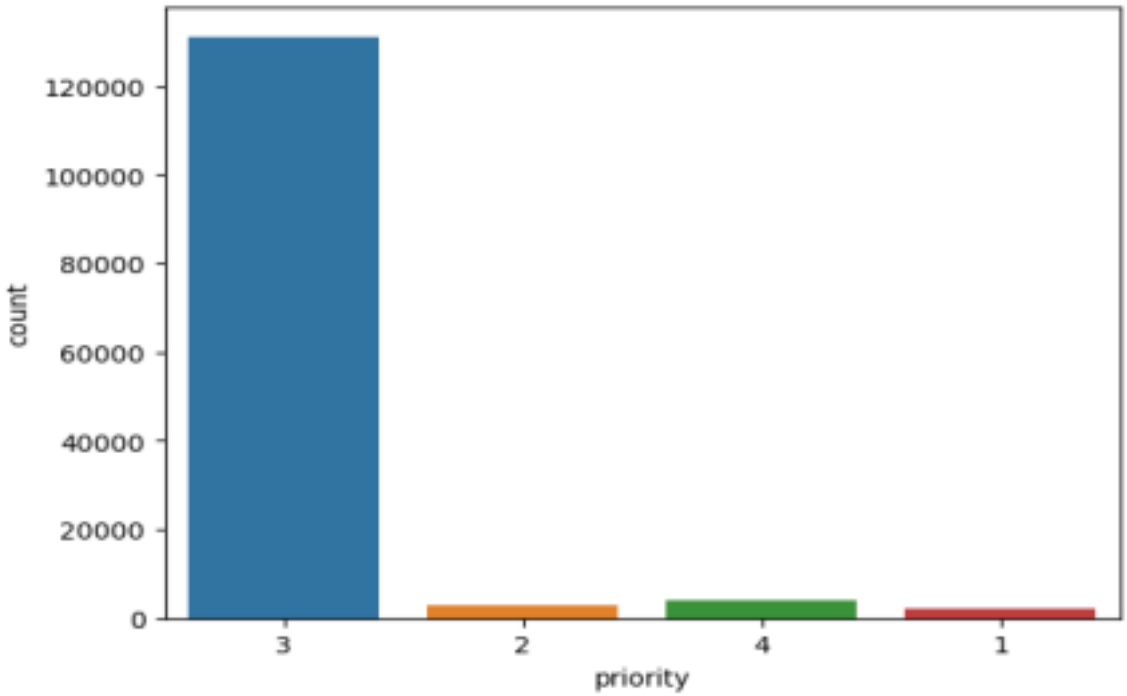


Interpretation:

• All the variables indicates the data is skewed and not normally distributed.
• In this graph except sys_mod_count, reopemed_count all other features indicates multimodal distribution.

- Even though almost all variables looked skewed specially u_symptoms and assignment group looks leftskewed.
- reassignment_count, sys_mod_count, reopened_count has peaked curve, all other variables also have peaked curve
- sys_created_by, opened_by, resolved_by and assaigned_to has graphs from increasing and then deteriorating.
- sys_mod_count has longer tail to the right when compared with other variables.

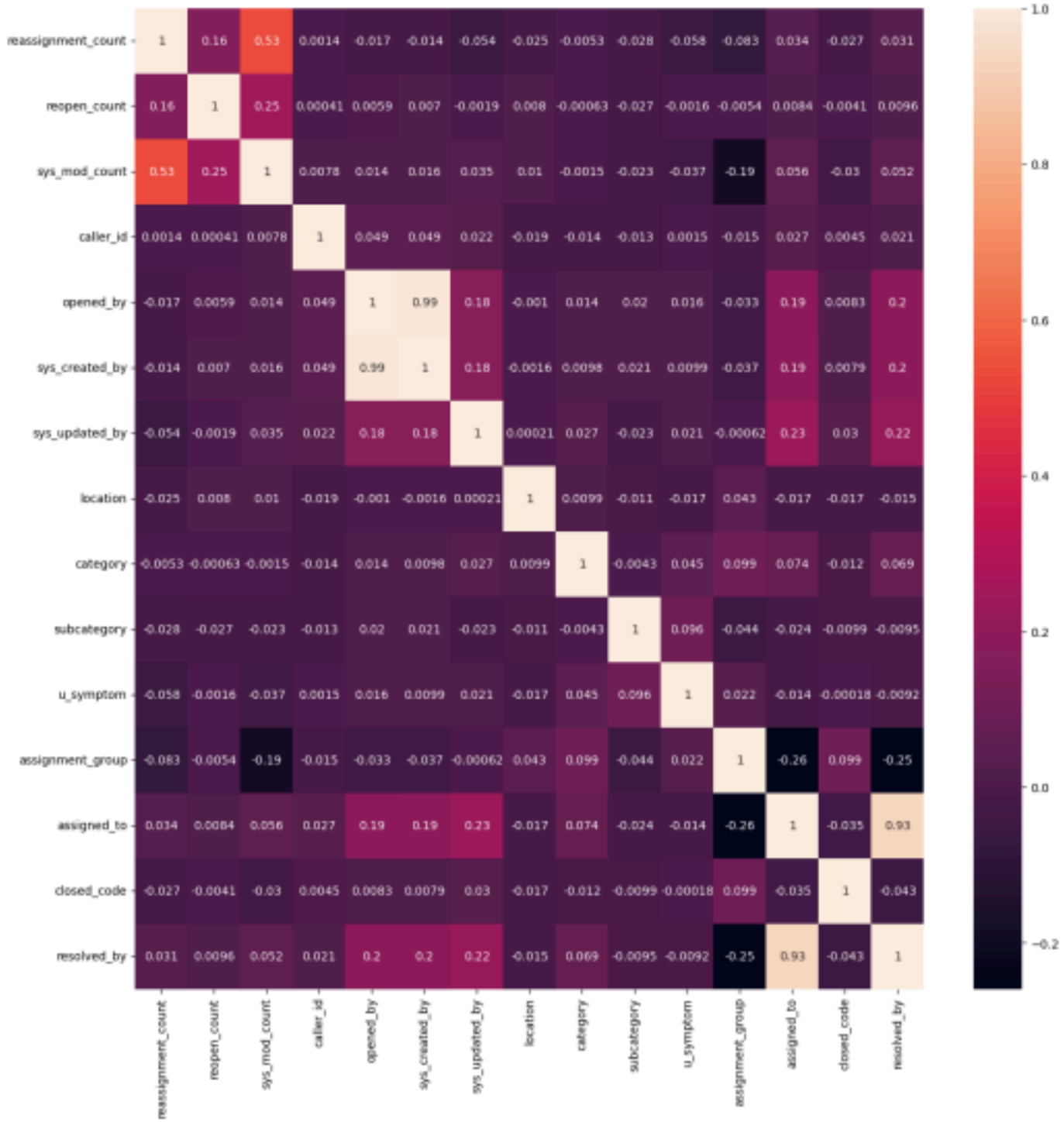
- **The Target variable is heavily imbalanced as well:**



Interpretation: The priority of the incident is to solve the incident faster, completely imbalanced and from the business perspective, it is reasonable to solve problems that make a difference and are resolved within time important
High, Critical problems need a lot of time and more labour and knowledge while poor problems are the least important considering the major problem count.

Relationship between variables

Relation between numerical variables:



Interpretation:

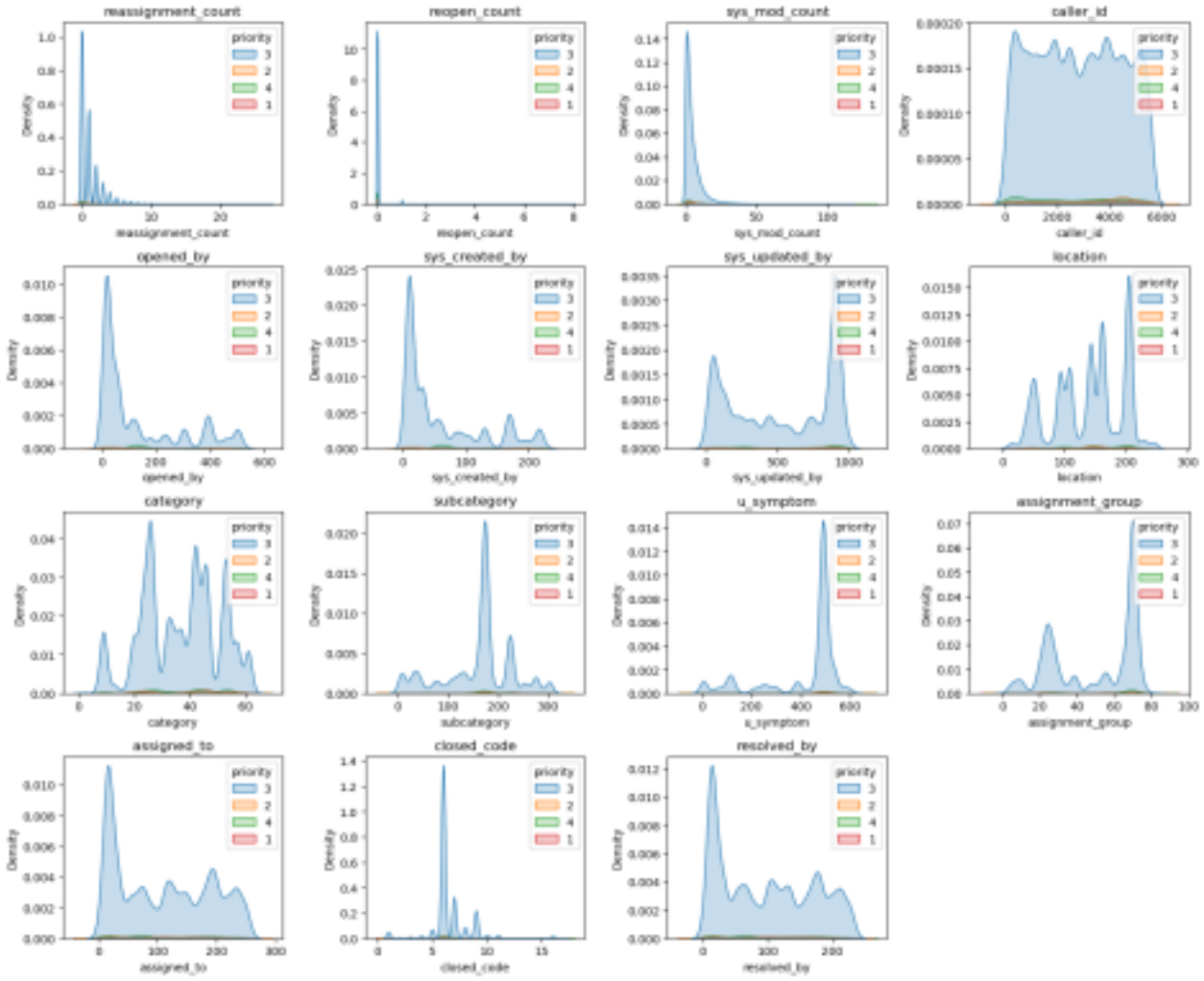
• There is a correlation between assigned_to and resolved_by as the person who resolves the incident-related issue is most of the time assigned to as well, these two columns were taken for imputation.

• Sys_created_by and opened_by are two columns that are correlating as well, indicating the one who opens the issue also creates the issue.

• There is multi-collinearity present among variables assignment_count, sys_mod_count, reopen_count, opened_by,sys_created_by, sys_updated_by. There is presence of negative correlation in assignment_group with sys_mod_count , resolved_by, assigned_to and also assigned_to to closed_code

BIVARIANT:

- **Distribution of numerical variables across target variable:**

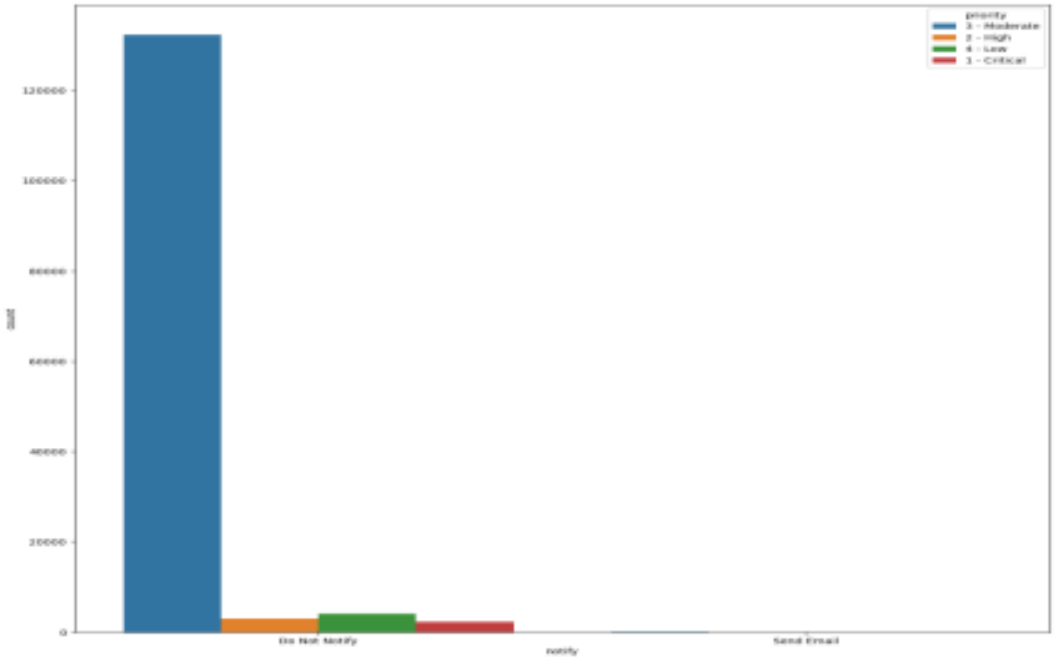


Interpretation:

▪ The distribution of data is mostly in moderate priority of the issue considering most of the issues which are given priority are moderate once and the problems the customer service received have most moderate issues than high, critical and poor.

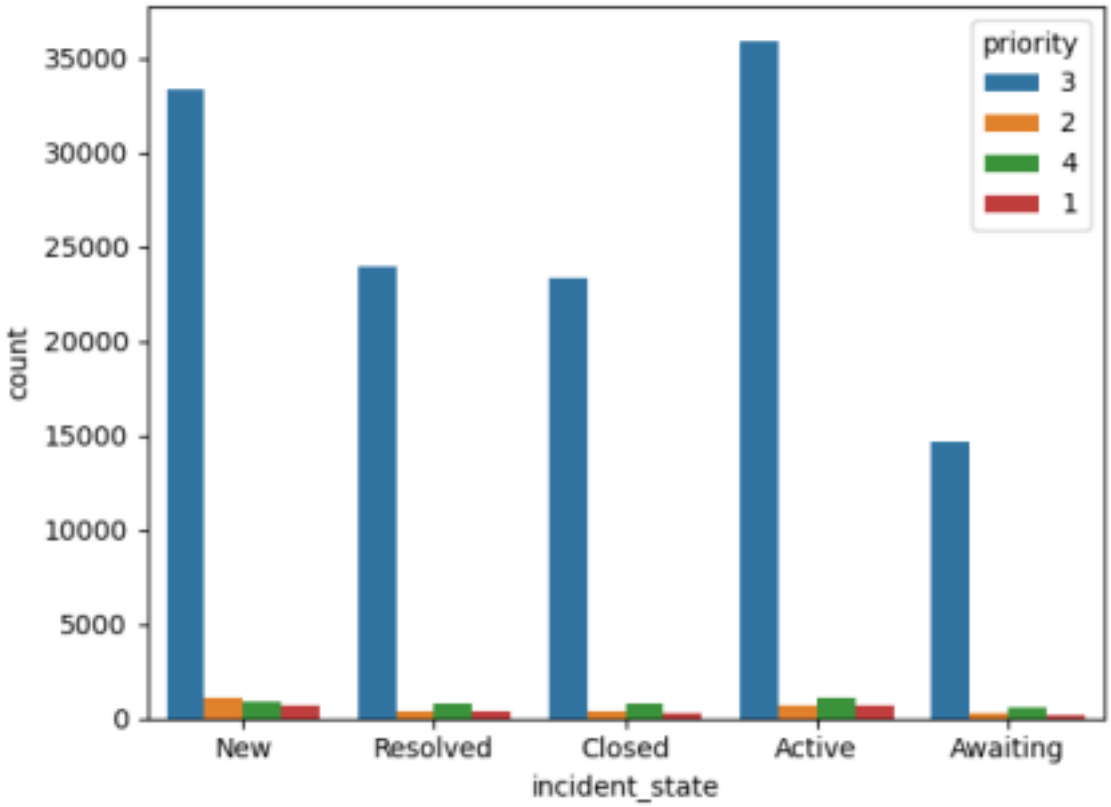
▪ The target variable is imbalanced and the above graph indicates that most of the data is biased towards one specific class.

- Distribution of categorical variables across target variables:



Interpretation: There are three subclasses in the notify variable but all the data is under DoNotNotify and the distribution of that particular data with target id is again imbalanced. This particular column could be dropped as the learning will be constant for a model.

- **Incident_state column with target variable:**



Most of the data is in moderate priority irrespective of the incident state
Interpretation: The incident state column indicates the state of the incident in the process of working towards resolving. The incidents are mostly new and are active until and unless the case is closed or resolved or waiting for information to go forward.

• Check for multicollinearity :
After implementing the missing values, the imputation process involved using
Variables that have created multicollinearity. We are not Treating Multi-Collinearity in the base model, further steps are taken for better model performance

• Distribution of variables: Not Normal, skewed

• Presence of outliers and their treatment
The outliers are a natural part of the population we are studying; we should not remove or impute it. Around 20000 records are outliers in data which is needed further so we did not drop them.

Statistical significance of variables:
For Categorical Variables, We shall use the Chi-Square Test of Independence. Do we test the independence of two categorical variables?

Hypothesis:

H0: Variables are insignificant.

H1: Variables are significant.

Assumptions of Chi-Square Test:

1. Both variables are categorical.

2. All observations are independent.

3. Cells in the contingency table are mutually exclusive.

4. Expected value of cells should be 5 or greater in at least 80% of cells.

Assumption 1: Both variables are categorical.
We are selecting the variables which are of data type and our target variable, Priority, a categorical variable, object for performing chi-square test. Hence this assumption is satisfied.

Assumption 2: All observations are independent.
It's assumed that every observation in the dataset is independent. That is, the value of one observation in the dataset does not affect the value of any other observation.

Assumption 3: Cells in the contingency table are mutually exclusive.
It's assumed that individuals can only belong to one cell in the contingency table. That is, cells in the table are mutually exclusive – an individual cannot belong to more than one cell.

Assumption 4: Expected value of cells should be 5 or greater in at least 80% of cells.
Expected value of cells should be 5 or greater in more than 95% of cells which is more than the stipulated 80% of cells.

Hence, all the assumptions of the Chi-square test are satisfied.

Now, we shall apply this test for each pair of all categorical independent variables and the dependent variable.

RESULTS:

Chi-square_result number:
Chi-square_statistic: 374955.4779734049 P-value:
0.0
Degree_of_freedom: 74073
.....
Chi-square_result incident_state:
Chi-square_statistic: 423.1368364227288 P-value:
4.7360131659149216e-83
Degree_of_freedom: 12
.....
Chi-square_result active:
Chi-square_statistic: 87.03205669939486 P-value:
9.504197705084117e-19
Degree_of_freedom: 3
.....
Chi-square_result reassignment_count:
Chi-square_statistic: 479.75536870946064 P-value:
6.316912433977742e-58
Degree_of_freedom: 81

.....
Chi-square_result reopen_count:
Chi-square_statistic: 76.64393472628228 P-value:
2.0648416324042365e-07
Degree_of_freedom: 24
.....
Chi-square_result sys_mod_count:
Chi-square_statistic: 1318.080531021734 P-value:
1.8448420518212735e-114
Degree_of_freedom: 342
.....
Chi-square_result made_sla:
Chi-square_statistic: 427.5811543011791 P-value:
2.3463541369019383e-92
Degree_of_freedom: 3
.....
Chi-square_result caller_id:
Chi-square_statistic: 121875.28479263635 P-value:
0.0
Degree_of_freedom: 15567
.....
Chi-square_result opened_by:
Chi-square_statistic: 67397.85613690471 P-value:
0.0
Degree_of_freedom: 588
.....
Chi-square_result opened_at:
Chi-square_statistic: 318087.533160558
P-value: 0.0
Degree_of_freedom: 59100
.....
Chi-square_result sys_created_by:
Chi-square_statistic: 68328.29085800354 P-value:
0.0
Degree_of_freedom: 552
.....
Chi-square_result sys_created_at:
Chi-square_statistic: 211193.24884053363 P-value:
0.0
Degree_of_freedom: 34518
.....
Chi-square_result sys_updated_by:
Chi-square_statistic: 26434.300656033152 P-value:
0.0
Degree_of_freedom: 2493
.....
Chi-square_result sys_updated_at:
Chi-square_statistic: 205512.5963888588 P-value:
0.0
Degree_of_freedom: 150924

.....
Chi-square_result contact_type: Chi-square_statistic:
74.20798471941745 P-value:
2.264825303289137e-12
Degree_of_freedom: 9
.....
Chi-square_result location:
Chi-square_statistic: 11372.22380055146 P-value:
0.0
Degree_of_freedom: 663
.....
Chi-square_result category:
Chi-square_statistic: 9531.637151786717 P-value:
0.0
Degree_of_freedom: 156
.....
Chi-square_result subcategory:
Chi-square_statistic: 17565.652997875703 P-value:
0.0
Degree_of_freedom: 738
.....
Chi-square_result u_symptom:
Chi-square_statistic: 23398.09246821543 P-value:
0.0
Degree_of_freedom: 1563
.....
Chi-square_result priority:
Chi-square_statistic: 420891.0
P-value: 0.0
Degree_of_freedom: 9
.....
Chi-square_result assignment_group:
Chi-square_statistic: 13863.675441326737 P-value:
0.0
Degree_of_freedom: 234
.....
Chi-square_result assigned_to:
Chi-square_statistic: 42662.756690212045
P-value: 0.0
Degree_of_freedom: 696
.....
Chi-square_result knowledge:
Chi-square_statistic: 1556.0229955158088
P-value: 0.0
Degree_of_freedom: 3
.....
Chi-square_result u_priority_confirmation:
Chi-square_statistic: 485.0793313138398
P-value: 8.167714769285749e-105
Degree_of_freedom: 3

.....
Chi-square_result notify:
Chi-square_statistic: 5.985570830309558
P-value: 0.11231434975225893
Degree_of_freedom: 3
.....
Chi-square_result closed_code:
Chi-square_statistic: 1138.860457731044
P-value: 4.7874444405473915e-207
Degree_of_freedom: 48
.....
Chi-square_result resolved_by:
Chi-square_statistic: 44879.020861382116
P-value: 0.0
Degree_of_freedom: 636
.....
Chi-square_result resolved_at:
Chi-square_statistic: 310397.53799022216
P-value: 0.0
Degree_of_freedom: 55113
.....
Chi-square_result closed_at:
Chi-square_statistic: 73694.60682632177
P-value: 0.0
Degree_of_freedom: 7998
.....

If the p-value is less than chosen significance level (e.g., 0.05), it would reject the null hypothesis and conclude that there is a significant association between the feature and the target variable. If the p-value is greater than the significance level, it would fail to reject the null hypothesis.

The chi-square test results provide insights into the association between each feature and the target variable, helping to identify which features are potentially significant for predicting the target variable.

insignificant_features:['incident_state', 'active', 'reassignment_count', 'reopen_count', 'sys_mod_count', 'made_sla', 'contact_type', 'u_priority_confirmation', 'notify', 'closed_code']- **this columns fail to reject the null hypothesis**.

- Class imbalance and its treatment:

From the visualization, there is target variable, which is highly imbalanced, but want to assign greater contribution to classes with more examples in the dataset, then the weighted average is preferred. This is because, in weighted averaging, the contribution of each class to the F1 average is weighted by its size, precision and recall score of classes separately.

Base model is built with imbalanced data further treatments like SMOTE, Majority under sampling,

make classification are included and performed to see better results.

5. Build models:

- Building base model BASE MODEL PERFORMANCE: XGBoost Classifier

Metrics for model xgb_model

Accuracy = 0.9421952957947256

Recall = 0.9421952957947256

Precision = 0.9414227554903783

F1 Score = 0.9220218156081273

```
[[   53     0   632     0]
 [    0   122   768     0]
 [    0    26 39233    10]
 [    0     0   997   249]]
```

	precision	recall	f1-score	support
0	1.00	0.08	0.14	685
1	0.82	0.14	0.24	890
2	0.94	1.00	0.97	39269
3	0.96	0.20	0.33	1246
accuracy			0.94	42090
macro avg	0.93	0.35	0.42	42090
weighted avg	0.94	0.94	0.92	42090

Inferences:

Precision:

- Precision is the ratio of true positive predictions to the total number of positive predictions made by the model.
- Precision measures the accuracy of the positive predictions made by the model. In this case, the precision scores are:

  class 0: 100%

  class 1: 82%

  class 2: 94%

class 3: 96%

- These values indicate the percentage of instances predicted as positive that were actually true positives.

Recall**:**

- Recall, also known as sensitivity or true positive rate, is the ratio of true positive predictions to the total number of actual positives in the dataset.
- Recall measures the model's ability to capture all positive instances. In this case, the recall scores are:

class 0: 8%

class 1: 14%

class 2: 100%

class 3: 20%

- These values indicate the percentage of true positive instances that were correctly identified by the model.

Interpretation:

- This is clear case of bias error, even though the accuracy say it is 94% accurate model, Since it is multi-class and each class performance is separately interpreted,
- The moderate class have a precision of 94% and recall of 1OO% which indicates Bias error and other classes have not been learned properly so they are not performing well.
- The recall and precision score of other classes are way low

- Depending on the business context and the cost associated with false positives and false negatives, we plan to adjust the model threshold to prioritize precision or recall.
- Considering additional techniques such as adjusting class weights, using different algorithms, or SMOTE to address imbalances and improve model performance.

6. Model evaluation:

Random Forest:

Model: Random Forest
Best Hyperparameters: {'max_depth': None, 'min_samples_split': 2, 'n_estimators': 200}
Accuracy: 0.9107
Classification Report:

	precision	recall	f1-score	support
0	0.91	0.92	0.92	4996
1	0.92	0.90	0.91	5048
2	0.91	0.91	0.91	5022
3	0.91	0.91	0.91	4934
accuracy			0.91	20000
macro avg	0.91	0.91	0.91	20000
weighted avg	0.91	0.91	0.91	20000

Precision: It measures the accuracy of positive predictions. A high precision indicates that when the model predicts a positive class, it is likely to be correct. In this case, precision values seem consistently high, ranging around 0.91, which is positive.

Recall: It measures the ability of the model to capture all the relevant instances of the positive class. A high recall indicates that the model is effective at identifying the positive instances. The recall values are also consistently high, around 0.91.

F1-score: It is the harmonic mean of precision and recall, providing a balance between the two metrics. A higher F1-score suggests a good balance between precision and recall. In this case, with precision and recall both around 0.91, the F1-score is likely to be high as well.

Gradient boost:

Model: Gradient Boosting
Best Hyperparameters: {'learning_rate': 0.2, 'max_depth': 7, 'n_estimators': 200}
Accuracy: 0.9409
Classification Report:

	precision	recall	f1-score	support
0	0.94	0.95	0.94	4996
1	0.94	0.94	0.94	5048
2	0.94	0.94	0.94	5022
3	0.94	0.94	0.94	4934
accuracy			0.94	20000
macro avg	0.94	0.94	0.94	20000
weighted avg	0.94	0.94	0.94	20000

Precision: It measures the accuracy of positive predictions. A high precision indicates that when the model predicts a positive class, it is likely to be correct. In this case, precision values seem consistently high, ranging around 0.94, which is positive.

Recall: It measures the ability of the model to capture all the relevant instances of the positive class. A high recall indicates that the model is effective at identifying the positive instances. The recall values are also consistently high, around 0.94.

F1-score: It is the harmonic mean of precision and recall, providing a balance between the two metrics. A higher F1-score suggests a good balance between precision and recall. In this case, with precision and recall both around 0.94, the F1-score is likely to be high as well.

Xgboost:

Model: XGBoost
Best Hyperparameters: {'learning_rate': 0.2, 'max_depth': 7, 'n_estimators': 200}
Accuracy: 0.9437
Classification Report:

	precision	recall	f1-score	support
0	0.94	0.95	0.95	4996
1	0.94	0.94	0.94	5048
2	0.95	0.94	0.94	5022
3	0.94	0.94	0.94	4934
accuracy			0.94	20000
macro avg	0.94	0.94	0.94	20000
weighted avg	0.94	0.94	0.94	20000

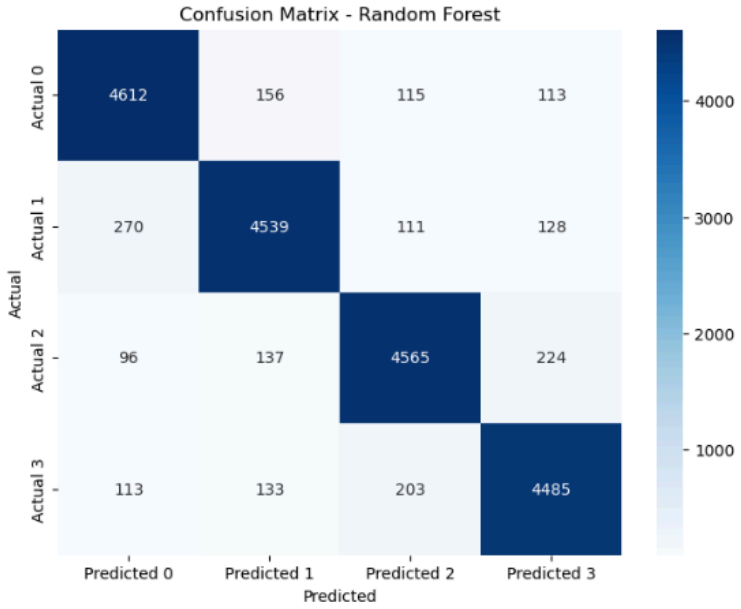
Precision: It measures the accuracy of positive predictions. A high precision indicates that when the model predicts a positive class, it is likely to be correct. In this case, precision values seem consistently high, ranging around 0.94, which is positive.

Recall: It measures the ability of the model to capture all the relevant instances of the positive class. A high recall indicates that the model is effective at identifying the positive instances. The recall values are also consistently high, around 0.94.

F1-score: It is the harmonic mean of precision and recall, providing a balance between the two metrics. A higher F1-score suggests a good balance between precision and recall. In this case, with precision and recall both around 0.94, the F1-score is likely to be high as well.

- There is a certain level of balance In data distribution among classes which has been achieved by make_classification.

7. **Insights from the best model and EDA:**
   - **Random Forest:**



- Diagonal Elements (True Positives):

  Class 0: 4612 correct predictions.
  Class 1: 4539 correct predictions.
  Class 2: 4565 correct predictions.
  Class 3: 4485 correct predictions.

- Off-Diagonal Elements:

  Elements outside the diagonal represent misclassifications.
  For example, in the row for Class 0, there are 270 instances predicted as Class 1, 96 instances predicted as Class 2, and 113 instances predicted as Class 3.

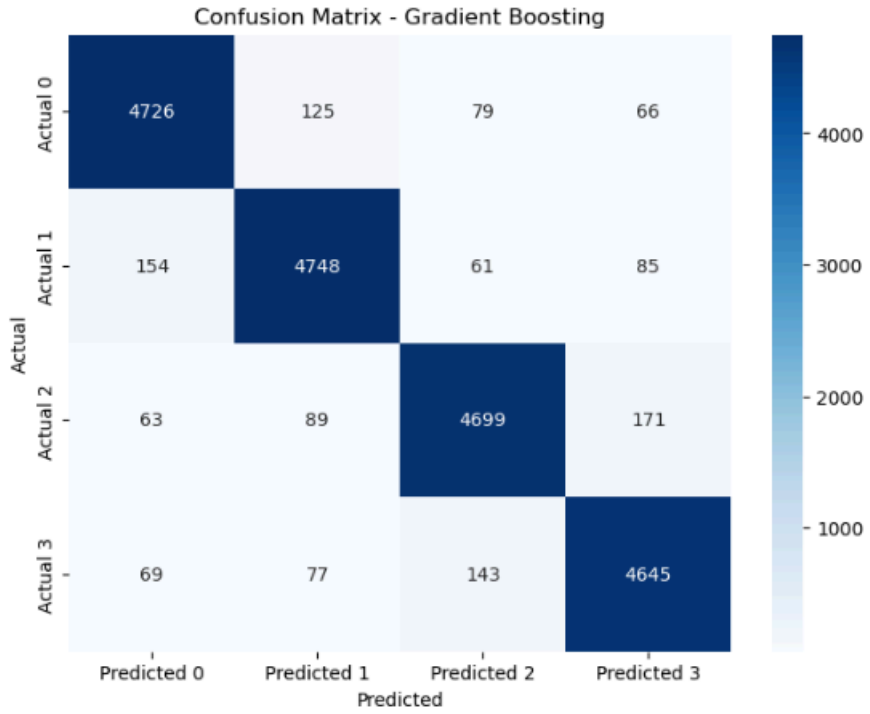
- Accuracy Calculation:

  The accuracy of 91.07% is calculated as (Sum of Diagonal) / (Total Instances).

- Precision, Recall, and F1 Score:

  Precision measures the accuracy of positive predictions, while recall measures the ability to capture all positives.
  The F1 score combines precision and recall into a single metric (F1 = 2 * (Precision * Recall) / (Precision + Recall)).

- **Gradient Boost:**



- Diagonal Elements (True Positives):

  Class 0: 4726 correct predictions.
  Class 1: 4748 correct predictions.
  Class 2: 4699 correct predictions.
  Class 3: 4645 correct predictions.

- Off-Diagonal Elements:

  Elements outside the diagonal represent misclassifications.

For example, in the row for Class 0, there are 154 instances predicted as Class 1, 63 instances predicted as Class 2, and 69 instances predicted as Class 3.

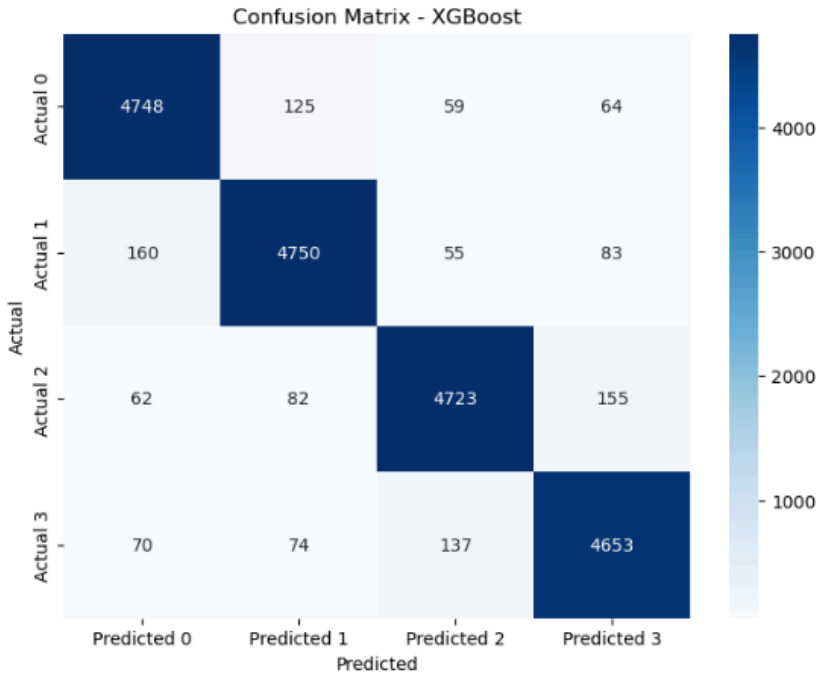
- Accuracy Calculation:

  The accuracy of 94.09% is calculated as (Sum of Diagonal) / (Total Instances).

- Precision, Recall, and F1 Score:

  Precision measures the accuracy of positive predictions, while recall measures the ability to capture all positives.
  The F1 score combines precision and recall into a single metric (F1 = 2 * (Precision * Recall) / (Precision + Recall)).

- **XGBoost:**



- Diagonal Elements (True Positives):

  Class 0: 4748 correct predictions.
  Class 1: 4750 correct predictions.
  Class 2: 4723 correct predictions.
  Class 3: 4653 correct predictions.

- Off-Diagonal Elements:

  Elements outside the diagonal represent misclassifications.

For example, in the row for Class 0, there are 160 instances predicted as Class 1, 62 instances predicted as Class 2, and 70 instances predicted as Class 3.

- Accuracy Calculation:

  The accuracy of 94.37% is calculated as (Sum of Diagonal) / (Total Instances).

- Precision, Recall, and F1 Score:

  Precision measures the accuracy of positive predictions, while recall measures the ability to capture all positives.
  The F1 score combines precision and recall into a single metric (F1 = 2 * (Precision * Recall) / (Precision + Recall)).

Model name	Class	Model				Model using SMOTE				Model using Under-Sampling			
		accuracy	Precision	Recall	F1 Score	accuracy	Precision	Recall	F1 Score	accuracy	Precision	Recall	F1 Score
Random Forest	0	0.99	0.92	0.84	0.88	0.91	0.92	0.84	0.88	0.99	0.92	0.84	0.88
	1		0.93	0.84	0.88		0.93	0.82	0.87		0.93	0.82	0.87
	2		0.99	1.00	0.99		0.99	1.00	0.99		0.99	1.00	0.99
	3		0.98	0.93	0.95		0.98	0.93	0.95		0.98	0.93	0.95
Gradient Boost	0	0.95	0.91	0.13	0.23	0.94	0.94	0.12	0.21	0.95	0.94	0.12	0.21
	1		0.85	0.21	0.33		0.79	0.21	0.33		0.79	0.21	0.33
	2		0.95	1.00	0.97		0.95	1.00	0.97		0.95	1.00	0.97
	3		0.87	0.33	0.47		0.83	0.32	0.52		0.83	0.38	0.52
XGBoost	0	0.94	0.94	0.03	0.06	0.98	0.90	0.58	0.70	0.98	0.90	58	0.70
	1		0.83	0.13	0.22		0.92	0.63	0.75		0.92	0.63	0.75
	2		0.94	1.00	0.97		0.98	1.00	0.99		0.98	1.00	0.99
	3		0.98	0.20	0.33		0.97	0.81	0.88		0.97	0.81	0.88

Model name	Class	Make Classification				Best Hyper-parameter	Make Classification with hyperparameter			
		Precision	Recall	F1 Score	Accuracy		Precision	Recall	F1 Score	Accuracy
Random Forest	0	0.91	0.92	0.91	0.91	{'max_depth': None, 'min_samples_split': 2, 'n_estimators': 200}	0.91	0.92	0.92	0.91
	1	0.91	0.90	0.91			0.92	0.90	0.91	
	2	0.91	0.91	0.91			0.91	0.91	0.91	
	3	0.91	0.91	0.91			0.91	0.91	0.91	
Gradient Boost	0	0.80	0.83	0.82	0.81	{'learning_rate': 0.2, 'max_depth': 7, 'n_estimators': 200}	0.94	0.95	0.94	0.94
	1	0.81	0.79	0.80			094	0.94	0.94	
	2	0.82	0.81	0.82			0.94	0.94	0.94	
	3	0.79	0.79	0.79			0.94	0.94	0.94	
XGBoost	0	.93	0.94	0.93	0.93	{'learning_rate': 0.2, 'max_depth': 7, 'n_estimators': 200}	0.94	0.95	0.95	0.94
	1	.93	0.93	0.93			0.94	0.94	0.94	
	2	0.94	0.92	0.93			0.95	0.94	0.94	
	3	0.92	0.93	0.92			0.94	0.94	0.94	

Choosing the best model:

1) XGBoost Outperforms:
    a) XGBoost consistently outperforms Random Forest in terms of precision and recall for all classes.
    b) XGBoost is comparable to Gradient Boost in terms of precision and recall.
    c) The results with SMOTE,Majority undersampling has given good performance but there is good growth in performance of XGBOOST with make_classification without being overfitting.
2) Decision to Choose XGBoost:

a) The decision to choose XGBoost can be justified by its strong performance across precision and recall metrics. It shows a balanced capability in making accurate positive predictions and capturing relevant positive instances.

3) Consideration for F1-Score:
   a) It's important to consider F1-score as well for a balanced evaluation. If F1-scores are similar between XGBoost and Gradient Boost, it reinforces the suitability of XGBoost.
4) Continuous Monitoring:
   a) Despite the high performance, continuous monitoring and updates to the model are recommended to adapt to changing data distributions and business requirements.

CHAPTERS 3 – Suggestions, recommendations to the stakeholder(s)

The solution presented in the project significantly impacts the incident management process in the domain or business. Here are the key ways in which the solution affects the problem:

1. Improved Incident Priority Prediction:

The developed model provides a reliable way to predict the priority of incidents. By accurately classifying incidents based on their severity, the incident management team can prioritize their response and allocate resources more efficiently. This leads to a more streamlined and effective incident resolution process.

2. Enhanced Resource Allocation:

With a precise incident priority prediction, the business can optimize resource allocation. Critical incidents can be addressed promptly, preventing potential disruptions and minimizing downtime. This contributes to increased operational efficiency and customer satisfaction.

3. Informed Decision-Making:

The insights gained from exploratory data analysis (EDA) and the model evaluation phase equip decision-makers with a better understanding of the incident management landscape. This knowledge enables informed decision-making, such as identifying patterns, trends, and areas for improvement in the incident resolution process.

Recommendations:

1. Address Class Imbalances:

- Given the imbalanced nature of the dataset, further efforts should be made to balance the classes, particularly those with fewer instances. Techniques such as oversampling, under sampling, or the use of

synthetic data generation can be explored.

2. Fine-Tune Model Parameters:

- Conduct a more exhaustive hyper parameter tuning process for the model, aiming to enhance its overall performance. This includes optimizing parameters specific to the chosen algorithm (e.g., XGBoost).

3. Explore Ensemble Methods:

- Investigate the potential benefits of ensemble methods, such as stacking or bagging, to further boost the predictive power of the model.

4. Continuous Monitoring and Updating:

- Establish a system for continuous monitoring of model performance in a real-world setting. Regularly update the model as new data becomes available to ensure its relevance and accuracy over time.

Confidence Level:

The confidence level in the recommendations is moderate to high. This is based on the thoroughness of the analysis conducted, the understanding of the dataset and domain, and the iterative nature of the model development process. However, the actual implementation and impact of these recommendations should be validated through ongoing monitoring and collaboration with domain experts. Regular feedback loops and updates to the model will contribute to an increasingly effective incident management solution.

CHAPTER 4 – LIMITATIONS

Imbalanced Data: The dataset exhibits imbalance in the distribution of incident priorities, which may lead to Biased Model performance. Stakeholders should be aware that the model may struggle with minority classes.

Missing Information: Despite attempts to impute missing values, the dataset contains instances with incomplete Information, potentially affecting the model's predictions. Stakeholders should consider investing in data collection and quality improvement.

Multicollinearity: The presence of multicollinearity among certain features may impact the model's interpretability. Further investigation or advanced feature engineering techniques may be necessary.

Response Time: Incident response time is critical, and delays in identification and resolution can have severe consequences. However, achieving a rapid response time can be challenging, especially for large organizations with complex systems.

CHAPTER 5 – CLOSING REFLECTIONS

In closing, the analysis of the incident management dataset within the IT domain has provided valuable insights and raised considerations for stakeholders. The exploration revealed challenges such as imbalanced data, missing information, and multicollinearity, emphasizing the need for continuous improvement in data quality and model robustness.

The integration of ITIL and ITSM frameworks, along with collaboration in business continuity, emerged as crucial for a holistic incident management approach. Stakeholders are encouraged to invest in optimizing ITIL implementation based on ongoing research findings.

Despite limitations, the model's base performance showed promise, but further steps, including addressing imbalances and refining features, are essential for enhanced accuracy and reliability. Continuous collaboration, user feedback, and training initiatives will contribute to an agile and resilient incident management system.

As stakeholders navigate these insights, a proactive and adaptive approach will be key to fostering effective incident resolution, improving customer satisfaction, and fortifying organizational resilience in the dynamic IT landscape.

BIBLIOGRAPHY:

References:

- https://www.atlassian.com/itsm/incident-management/itscm#:~:text=In%20ITIL%204%2C%20service%20continuity,timelines%20after%20major%20service%20disruptions.
-
- https://phoenixnap.com/blog/what-is-business-continuity-management
- https://core.ac.uk/download/pdf/301359912.pdf
- https://www.mdpi.com/2076-3417/10/1/34
- http://scikit-learn.org/stable/modules/generated/sklearn.datasets.make_classification.html

ANNEXURE

- **Annexure 1: Detailed Model Performance Metrics**

  Xgboost:

```
Model: XGBoost
Best Hyperparameters: {'learning_rate': 0.2, 'max_depth': 7, 'n_estimators': 200}
Accuracy: 0.9437
Classification Report:
              precision    recall  f1-score   support

           0       0.94      0.95      0.95      4996
           1       0.94      0.94      0.94      5048
           2       0.95      0.94      0.94      5022
           3       0.94      0.94      0.94      4934

    accuracy                           0.94     20000
   macro avg       0.94      0.94      0.94     20000
weighted avg       0.94      0.94      0.94     20000

==================================================
```

Precision: It measures the accuracy of positive predictions. A high precision indicates that when the model predicts a positive class, it is likely to be correct. In this case, precision values seem consistently high, ranging around 0.94, which is positive.

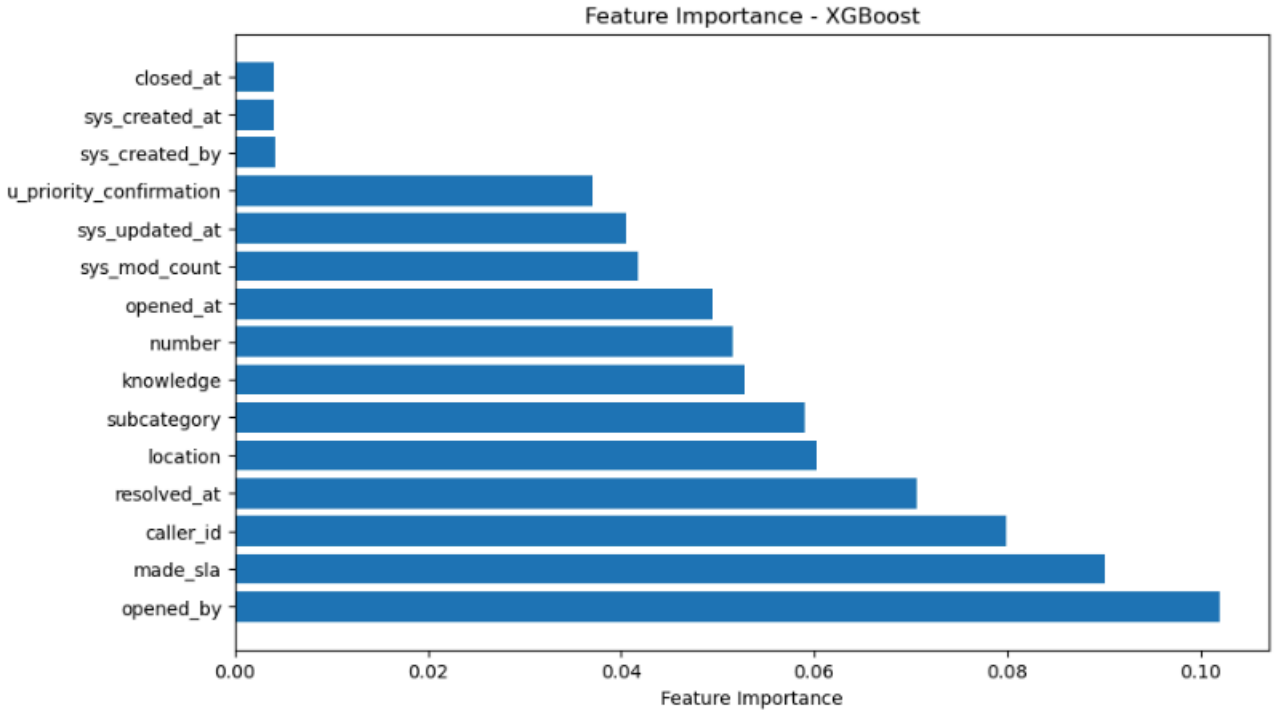
Recall: It measures the ability of the model to capture all the relevant instances of the positive class. A high recall indicates that the model is effective at identifying the positive instances. The recall values are also consistently high, around 0.94.

F1-score: It is the harmonic mean of precision and recall, providing a balance between the two metrics. A higher F1-score suggests a good balance between precision and recall. In this case, with precision and recall both around 0.94, the F1-score is likely to be high as well.

There is a certain level of balance In data distribution among classes which has been achieved by make_ classification.

- **Annexure 2: Detailed Feature Importance**

  Table showing the detailed feature importance scores for the selected features:



A feature importance graph indicates the contribution of each feature to a model's predictions. High bars for certain features suggest they have a more significant impact, while low bars indicate less influence.

Features: [‘opened_by’, ‘made_sla’, ‘caller_id’, ‘resolved_at’, ‘location’, ‘subcategory’, ‘knowledge’, ‘number’, ‘opened_at’] – this feature have high impact on the model.

1. DATA DICTIONARY

0 number: incident identifier (24,918 different values);

1 incident state: eight levels controlling the incident management process transitions from opening until closing the case;

2 active: a boolean attribute that shows whether the record is active or closed/canceled;

3 reassignment_count: number of times the incident has the group or the support analysts changed;

4 reopen_count: number of times the caller rejected the incident resolution;

5 sys_mod_count: number of incident updates until that moment;

6 made_sla: a boolean attribute that shows whether the incident exceeded the target SLA;

7 caller_id: identifier of the user affected;

8 opened_by: identifier of the user who reported the incident;

9 opened_at: incident user opening date and time;

10 sys_created_by: identifier of the user who registered the incident;

11 sys_created_at: incident system creation date and time;

12 sys_updated_by: identifier of the user who updated the incident and generated the current log record;

13 sys_updated_at: incident system update date and time;

14 contact_type: a categorical attribute that shows by what means the incident was reported;

15 location: identifier of the location of the place affected;

16 category: first-level description of the affected service;

17 subcategory: second-level description of the affected service (related to the first level description, i.e., to category);

18 u_symptom: description of the user perception of service availability;

19 cmdb_ci: (confirmation item) identifier used to report the affected item (not mandatory);

20 impact: description of the impact caused by the incident (values: 1â€“High; 2â€“Medium; 3â€–Low);

21 urgency: description of the urgency informed by the user for the incident resolution (values: 1â€–High; 2â€–Medium; 3â€–Low);

22 priority: calculated by the system based on 'impact' and 'urgency';

23 assignment_group: identifier of the support group in charge of the incident;

24 assigned_to: identifier of the user in charge of the incident;

25 knowledge: a boolean attribute that shows whether a knowledge base document was used to resolve the incident;

26 u_priority_confirmation: a boolean attribute that shows whether the priority field has been double-checked;

27 notify a categorical attribute that shows whether notifications were generated for the incident;

28 problem_id: identifier of the problem associated with the incident;

29 rfc: (request for change) identifier of the change request associated with the incident;

30 vendor: identifier of the vendor in charge of the incident;

31 caused_by: identifier of the RFC responsible for the incident;

32 close_code: identifier of the resolution of the incident;

33 resolved_by: identifier of the user who resolved the incident;

34 resolved_at: incident user resolution date and time;

35 closed_at: incident user close date and time .