

|  |
| --- |
| Interim Report  2021 |
|  |
| January 9  Authored by:  Vasundhara Madan, Vimal Pandey,  Pradeep Kumar VP, Raghav Agarwal Apurba Banerjee |

Automatic Ticket Assignment System

Table of Contents

[Introduction 3](#_Toc60926155)

[Problem Statement 4](#_Toc60926156)

[Objective 4](#_Toc60926157)

[EDA (Exploratory Data Analytics) 4](#_Toc60926158)

[Data Findings 4](#_Toc60926159)

[Glimpse of the dataset 5](#_Toc60926160)

[Column attribute information 5](#_Toc60926161)

[Key Data findings 6](#_Toc60926162)

[Feature Engineering 6](#_Toc60926163)

[Final data frame: 7](#_Toc60926164)

[Visual Analysis 8](#_Toc60926165)

[Distribution of Assignment group across the dataset 8](#_Toc60926166)

[Distribution of Char\_len across the dataset 9](#_Toc60926167)

[Text Preprocessing 10](#_Toc60926168)

[Modelling 11](#_Toc60926169)

[ML Based Classification models with TF-IDF 11](#_Toc60926170)

[ML Based Classifier with Glove Embeddings 11](#_Toc60926171)

[AI Based Classification Models 12](#_Toc60926172)

[AI Based Models: Architectural Details 12](#_Toc60926173)

[Simple Dense NN 12](#_Toc60926174)

[Convolutional NN 12](#_Toc60926175)

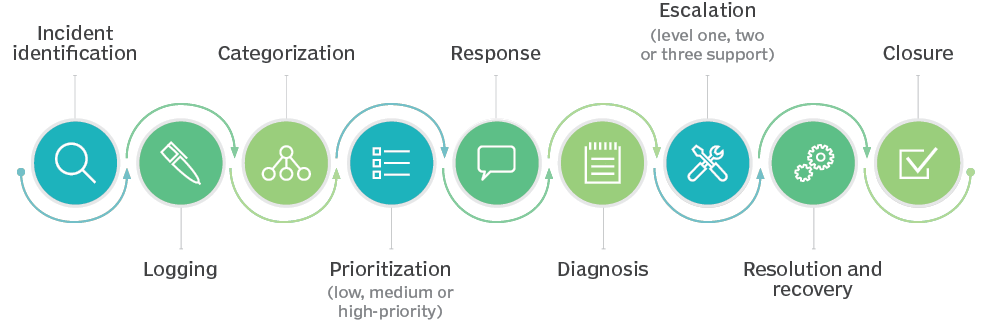
[RNN Based LSTM 13](#_Toc60926176)

[How to Improve Model Performance 13](#_Toc60926177)

# Introduction

One of the key activities of any IT function is to “Keep the lights on” to ensure there is no impact to the Business operations. IT leverages Incident Management process to achieve the above Objective. An incident is something that is unplanned interruption to an IT service or reduction in the quality of an IT service that affects the Users and the Business. The main goal of Incident Management process is to provide a quick fix / workarounds or solutions that resolves the interruption and restores the service to its full capacity to ensure no business impact. In most of the organizations, incidents are created by various Business and IT Users, End Users/ Vendors if they have access to ticketing systems, and from the integrated monitoring systems and tools. Assigning the incidents to the appropriate person or unit in the support team has critical importance to provide improved user satisfaction while ensuring better allocation of support resources. The assignment of incidents to appropriate IT groups is still a manual process in many of the IT organizations. Manual assignment of incidents is time consuming and requires human efforts. There may be mistakes due to human errors and resource consumption is carried out ineffectively because of

the misaddressing. On the other hand, manual assignment increases the response and resolution times which result in user satisfaction deterioration / poor customer service.

****

# Problem Statement

In the IT support process, incoming incidents are analysed and assessed by organization’s support teams to fulfil the request. In many organizations, better allocation and effective usage of the valuable support resources directly results in substantial cost savings.

However, successful closure is not the only thing which matters in customer satisfaction. SLA management of incident must be managed effectively. Generally, manually assigning the incident has some challenges stated below:

* More resource usage and expenses.
* Human errors - Incidents get assigned to incorrect groups
* Delay in assigning the incidents
* More resolution times
* If an incident takes more time in analysis, other productive tasks get affected for the Service Desk

# Objective

Build an AI-ML based classifier model to assign the tickets to right functional groups by analysing the given ticket/incident description with an accuracy of 60%-80%.

# EDA (Exploratory Data Analytics)

EDA is the Data Analysis Process where several techniques are used to better understand the dataset being used. It helps clean up a dataset. It gives you a better understanding of the variables and the relationships between them.

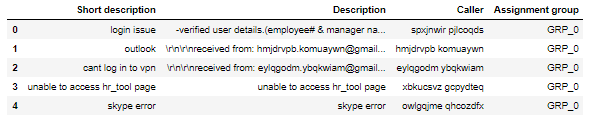
EDA was performed on the given data set. Following were the findings:

## Data Findings

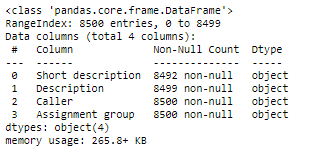
After analyzing the given data set, following observations were drawn:

* The dataset contains 8500 records with 4 attributes i.e. Short Description, Description, Caller and Assignment Group, which is the target column.
* Each column is of data type object.
* There are 7481 unique Short Description types for the given dataset. “Password Reset” is most frequently occurring Short Description.
* There are 74 different Assignment groups. Almost 47% of the tickets are assigned to GRP\_0 and 53% of the tickets are assigned to the remaining 73 assignment groups.
* Imbalanced class data
* There are 9 records with missing values- 8 records with missing/NaN Short description and 1 record with missing/NaN Description.
* There are 83 records with duplicate values.
* The data belongs to 29 different languages. Almost 84% of the records belong to English language.

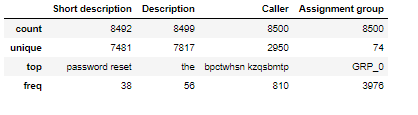
### Glimpse of the dataset



### Column attribute information



### Key Data findings



## Feature Engineering

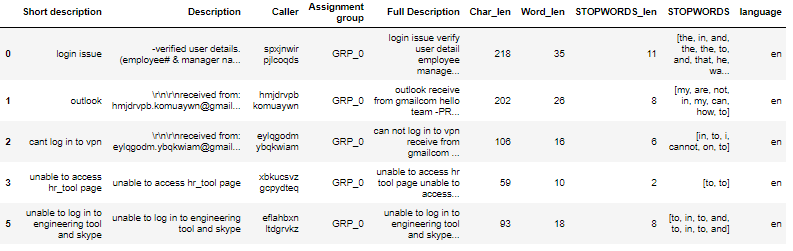
It is the process of creating features for an AI-ML based models from raw text data.

To further analyze other aspects of the data presented, we have employed different methods to analyze text and extract features that can be used to build a classification model to draw note-worthy conclusions.

We created custom columns to study and better understand the data. Following columns were created:

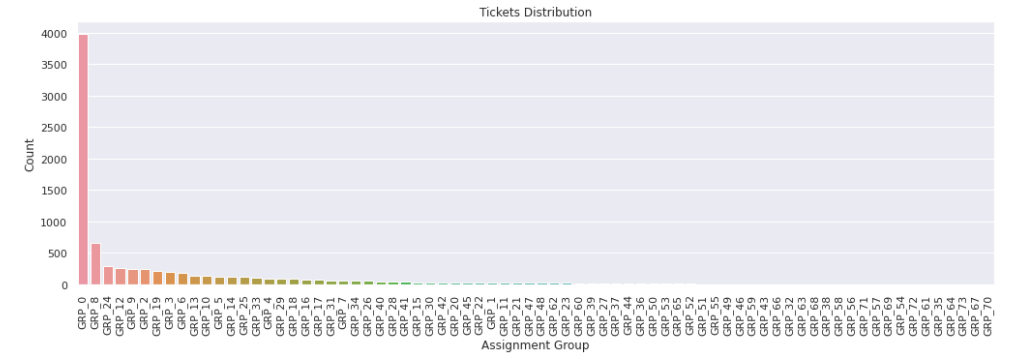
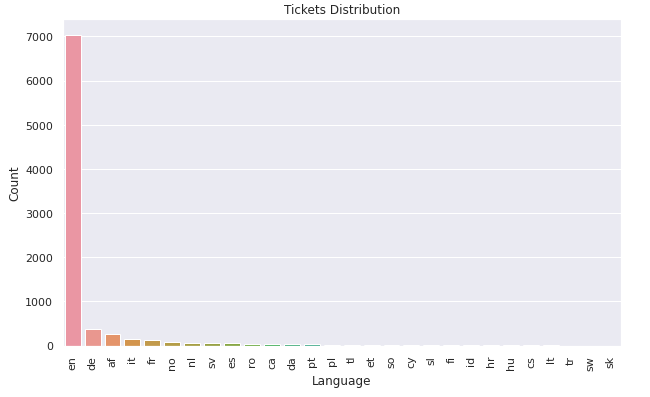
* Created a custom column “Full Description” by concatenating “Short description” and “Description” column.
* Added few custom columns like Char\_len and Word\_len to study the no. of characters and no. of words in Full Description column for every record. The min length of Full Description is 3 whereas the max length for Full Description is 13104.
* Added a ‘language’ column to study to detect the language type for every record. The data belongs to 29 different languages. Almost 84% of the records belong to English language.
* Added a column ‘STOPWORDS’ to study the stop words in Full Description column of every record.

### Final data frame:

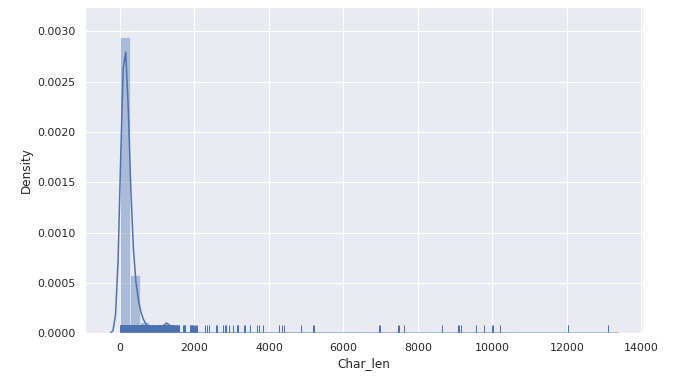


## Visual Analysis

### Distribution of Assignment group across the dataset

Distribution of Language across the dataset

### Distribution of Char\_len across the dataset

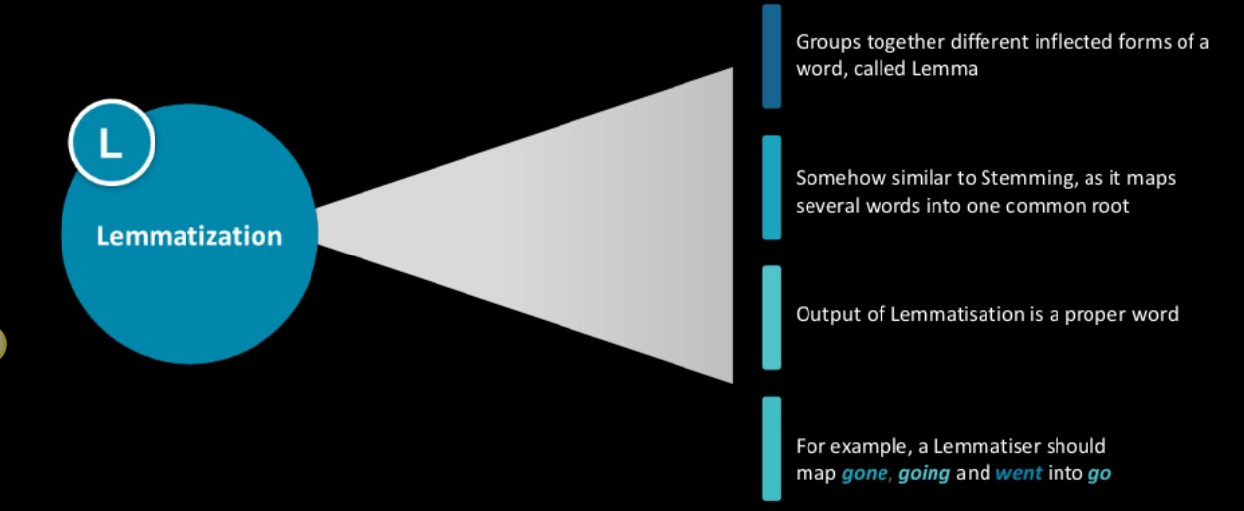
Word Cloud to analyze important and most frequently occurring words



# Text Preprocessing

Following text/data pre-processing steps were employed to clean the data before it is presented to models:

* Drop Missing Values: 9 records were found with missing or NaN values- 8 records with missing/NaN Short description and 1 record with missing/NaN Description. These records were dropped.
* Drop Duplicates: 83 records with duplicate values were found that were dropped from the dataset.
* Removed HTML Tags
* Removed Accented Text
* Removed Punctuations
* Remove Special Characters
* Performed Lemmatization



Note: Stemming and Lemmatization both generate the root form of the inflected words. The difference is that stem might not be an actual word whereas, lemma is an actual language word. Stemming follows an algorithm with steps to perform on the words which makes it faster. Whereas, lemmatization, slower than stemming. We chose Lemmatization over Stemming.

# Modelling

## ML Based Classification models with TF-IDF

We have used ML based classification models as well as AI based classification models to assign tickets to functional groups.

* Full Description column i.e. Short description + Description has been vectorized to TF-IDF using TfidfVectorizer using unigrams
* Select top 5000 features using Chi-squared test
* Different ML based classifications models were used, and following accuracies were found

|  |  |  |
| --- | --- | --- |
| Model Type | Training Accuracy | Test Accuracy |
| Naïve Bayes Classifier | 57.05 | 57.53 |
| SVM Classifier | 66.48 | 64.26 |
| Decision Tree Classifier | 63.07 | 60.95 |
| Random Forest Classifier | 95.06 | 64.1 |

Note: Random Forest Classifier model is overfitting

## ML Based Classifier with Glove Embeddings

* Vectorized Full Description column using Glove embedding with embedding size as 300
* Label encoding of the target variable i.e. Assignment group
* Different ML based classification models were used, and following accuracies were found

|  |  |  |
| --- | --- | --- |
| Model Type | Training Accuracy | Test Accuracy |
| SVM Classifier | 89.25 | 66.59 |
| Decision Tree Classifier | 62.27 | 55.87 |
| Random Forest Classifier | 95.16 | 61.43 |

Note: Using glove embeddings, all the models are overfitting

## AI Based Classification Models

* Vectorised Full Description column using Glove embedding with embedding size as 300
* Label encoding of the target variable i.e. Assignment group
* Different AI based classification models were used and following accuracies were found

|  |  |  |
| --- | --- | --- |
| Model Type | Training Accuracy | Test Accuracy |
| Simple Dense NN | 53.46 | 49.21 |
| Convolutional NN | 63.57 | 58.32 |
| RNN based LSTM | 60.79 | 55.94 |

Observations:

* All the AI based model are overfitting.
* Hyper tuning of the models in required to further improve the performance all the while ensuring that the model doesn’t overfit

## AI Based Models: Architectural Details

### Simple Dense NN

* Input Dense Layer wit 512 neurons/nodes and ‘he\_normal’ initializer
* 3 hidden layers with 512 neurons in each layer, relu activation
* Every hidden layer is followed by Batch Normalization layer and Dropout layer with dropout ratio of 0.3
* Output layer with activation function softmax

### Convolutional NN

* Input layer with max length as 200
* Embedding layer with embedding size as 300
* 5 layers of (Conv1D+MaxPool1D) with kernel size in each Conv1D layer as 2,3,4,5,6 and nodes=128, activation: relu
* 2 layers of (Conv1D+dropout+Batch Norm+MaxPool1D) with 128 nodes in each Conv1D, kernel size 5, activation: relu, dropout ratio 0.5,
* Next flatten followed by dense layer of 1024 and 512 resp.
* Output layer with 74 nodes and softmax activation

### RNN Based LSTM

* Embedding layer with max length: 200, dense embedding size 300
* Bidirectional LSTM Layer with 256 units
* 3 units of (Dense + Activation + Dropout) layers
* Every dense layer has 512 nodes
* Activation function used is ReLu
* Dropout layer with a dropout ratio of 0.5 is used
* Output dense layer with 74 units along with softmax activation is used

# How to Improve Model Performance

To improve model performance, we will try out following:

* Try glove embeddings with size 100, 200 and 300
* Try to learn embeddings instead of using glove embeddings
* For LSTM Model, try unidirectional as well as bidirectional LSTM
* For LSTM Model, try different no. of LSTM layers. Try with different no. of LSTM units like 64, 128, 256 etc. in each layer of the LSTM architecture
* Try different optimizers like Adam, RMS Prop by varying the learning rate
* Try different activation functions like ReLu, P-ReLu, Leaky ReLu etc
* Try working with data belonging to top 5 and top 10 groups to address the class imbalance issue.
* Try including Caller information in the input dataset.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* THANK YOU \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*