

Table of Contents

Summary of problem statement, data and findings.....	2
Understanding the Business.....	2
Objective.....	2
Observations from the given Dataset.....	3
Observations from Target Class	3
Summary of the Approach to EDA and Pre-processing	4
Data Pre-processing	4
Data Pre-processing – Addressing skew of target class.....	5
Data cleaning	6
Translation	6
Word Distribution.....	6
Lemmatization & Stop words removal.....	7
Spell Check.....	7
Deciding Models and Model Building.....	8
Modelling	8
Traditional ML Models	8
Observation from the results of the traditional models.....	8
Model Clustering	8
Analysis using WordCloud.....	9
Model evaluation	10
Comparison to benchmark.....	10
How to improve your model performance?	11
Implications	11
Limitations	11
Suggestions for feature model improvement	11

CAPSTONE PROJECT- AUTOMATIC TICKET ASSIGNMENT

Summary of problem statement, data and findings

Understanding the Business

In any IT industry, Incident Management plays an important role in delivering quality support to customers. An incident ticket is created by various groups of people within the organization to resolve an issue as quickly as possible based on its severity. Whenever an incident is created, it reaches the Service desk team and then it gets assigned to the respective teams to work on the incident.

The Service Desk team (L1/L2) will perform basic analysis on the user's requirement, identify the issue based on given descriptions and assign it to the respective teams.

The manual assignment of these incidents might have below disadvantages:

- More resource usage and expenses
- Human errors - incidents get assigned to the wrong assignment groups
- Delay in assigning the tickets
- More resolution times
- If a particular ticket takes more time in analysis, other productive tasks get affected for the service desk

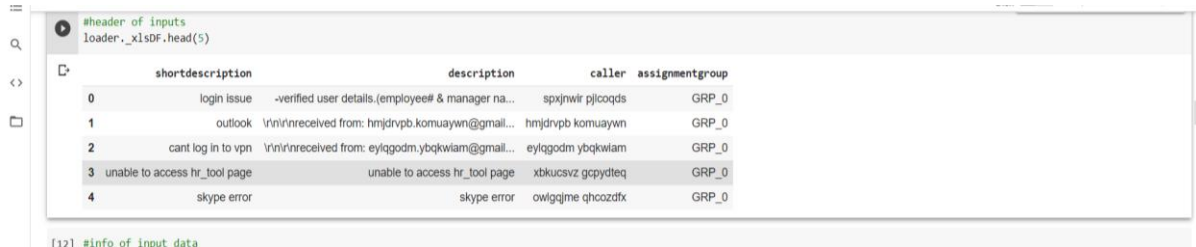
If this ticket assignment is automated, it can lower resolution time and be more cost-effective, enabling the service desk team to focus on other productive tasks.

Objective

From the given problem description, we could see that the existing system is able to assign 75% of the tickets correctly. Hence, our objective here is to build an AI-based classifier model to assign the tickets to right functional groups by analysing the given description with an accuracy of at least 85%.

Observations from the given Dataset

The data consists of four columns as given below.



```
#header of inputs
loader._xlsDF.head(5)
```

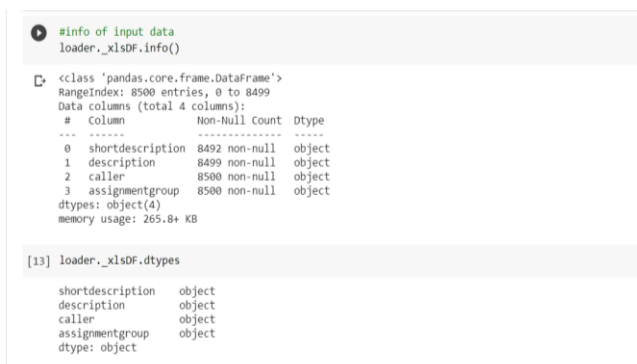
	shortdescription	description	caller	assignmentgroup
0	login issue	-verified user details.(employee# & manager na...	spxjnwir pjcoqds	GRP_0
1	outlook	\r\n\r\nreceived from: hmjdrvpb.komuaywn@gmail...	hmjdrvpb komuaywn	GRP_0
2	cant log in to vpn	\r\n\r\nreceived from: eylgodm.ybqkwiam@gmail...	eylgodm ybqkwiam	GRP_0
3	unable to access hr_tool page	unable to access hr_tool page	xbkucsvz gcpydteq	GRP_0
4	skype error	skype error	owlgqjme qhcozdfx	GRP_0

```
[12] #info of input data
```

- Four columns – Short Description, Description, Caller and Assignment group
- 74 Assignment groups found - Target classes
- Caller names in a random fashion (may not be useful for training data)
- European non-English language also found in the data
- Email/chat format in description
- Symbols & other characters in the description
- Hyperlinks, URLS & few image data found in the description
- Blanks found either in the short description or description field
- Few descriptions same as the short description
- Few words were combined together
- Spelling mistakes and typo errors are found

Observations from Target Class

The data contains 8500 entries, with some missing values in certain columns.



```
#info of input data
loader._xlsDF.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 8500 entries, 0 to 8499
Data columns (total 4 columns):
#   Column          Non-Null Count  Dtype
---  ---
0    shortdescription  8492 non-null   object
1    description       8499 non-null   object
2    caller            8500 non-null   object
3    assignmentgroup   8500 non-null   object
dtypes: object(4)
memory usage: 265.8+ KB
```

```
[13] loader._xlsDF.dtypes
```

```
shortdescription    object
description         object
caller              object
assignmentgroup     object
dtype: object
```

- The Target class distribution is extremely skewed

- A large no of entries for GRP_0 (amounting to 3976) which account for ~50% of the data
- There are groups with 1 entry also. We could merge all groups with small entries to a group to reduce the imbalance in the target. This may reduce the imbalance to some extent.

Summary of the Approach to EDA and Pre-processing

Data Pre-processing

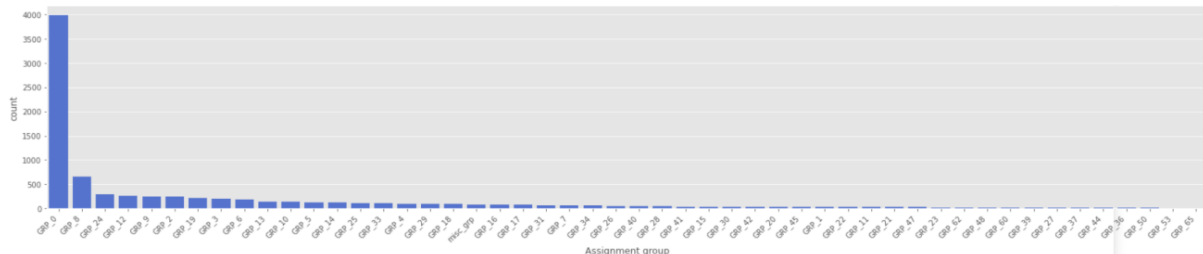
Below steps have been performed for initial pre-processing and clean-up of data:

- Dropped the caller field as the data was not found to be useful for analysis
- Replaced Null values in Short description & description with space
- Merged Short Description & Description fields for analysis
- Contraction words found in the merged Description are removed for ease of word modelling
- Changed the case sensitivity of words to the common one
- Removed Hashtags and kept the words, Hyperlinks, URLs, HTML tags & non-ASCII symbols from merged fields
- Translating all languages (German) to English
- Tokenization of merged data
- Removal of Stop words
- Lemmatization
- Word Cloud created for all available 50 groups to have more information specific to Assignment groups
- Attempted to do spell check
- Created Plot to understand the distribution of words

Data Pre-processing – Addressing skew of target class

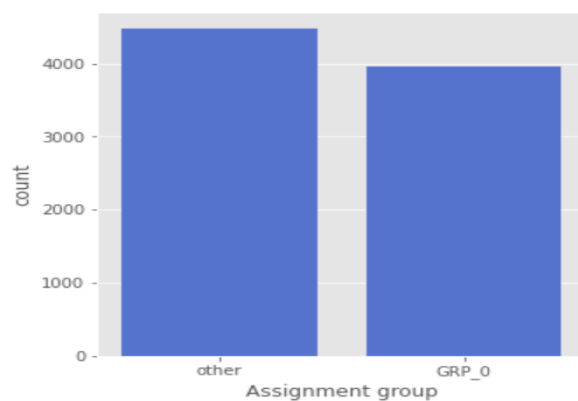
The target class is extremely skewed data. The target class were filtered for less than 10 entries and grouped together as misc_grp as there is not sufficient information with the groups individually, where there are less than 10 entries.

Distribution of Target class for all groups (after adding misc_grp)

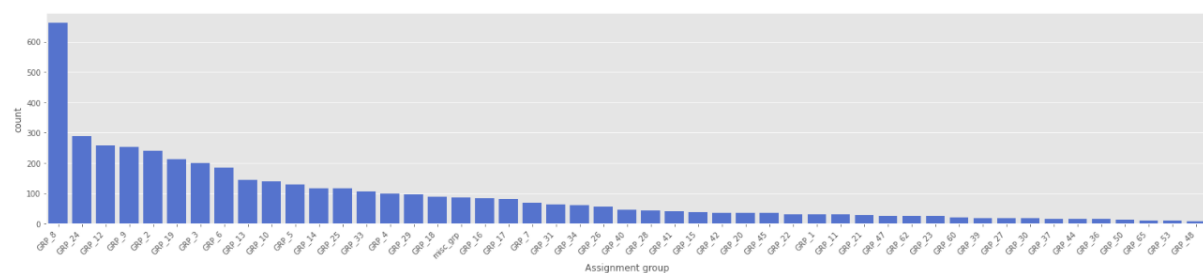


To further address the imbalance in the target class, we have split the dataset as 2 groups

- A dataset where we resample all the groups to size 660. Here note GRP0 would be down sampled & all other groups would be up sampled.
- We could use 2 separate models. Here one model would be used to classify the GRP0 & a second model would be used to classify the other groups. The dataset from the 1st model contains GRP0 data & all the remaining data combined to a single group, say, 'Others'. The dataset for the 2nd Model would contain all groups other than GRP0. The dataset here would be resampled again to address the target imbalance if any.



Distribution of Target class for Other groups



Data cleaning

Through the process of data cleaning we would want to clean up the unwanted information from initial observations. All words are converted to lowercase. The email headers and sender information is also removed. All the numbers, non-dictionary characters, newline characters, hashtag, HTML entities, hyperlinks, extra spaces and unreadable characters and any caller names included in the description column are removed. The `clean_data` function is applied to the description column and cleaned up data is generated for further analysis.

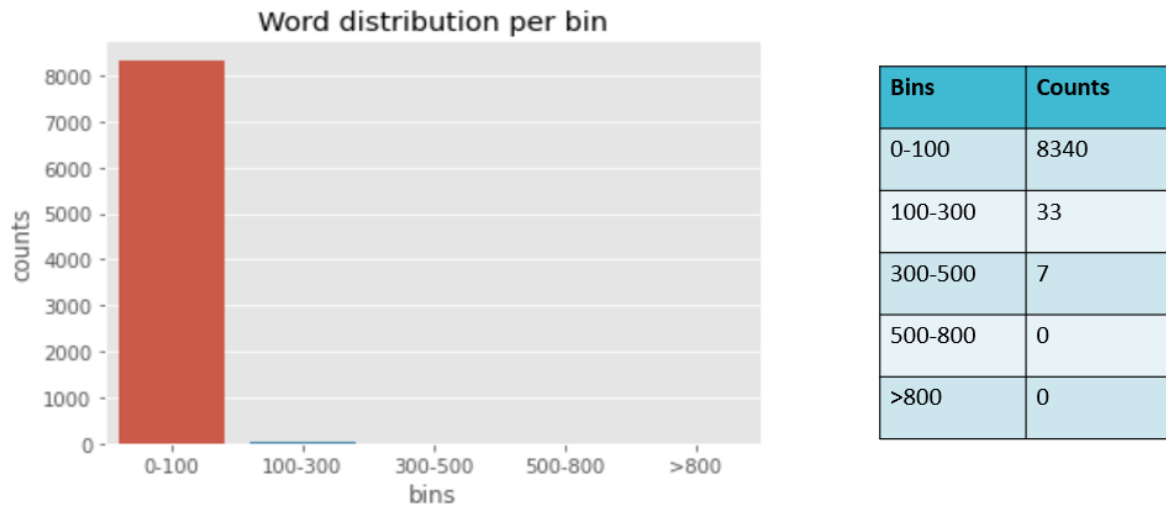
Translation

German language is found in the dataset. We attempted using many libraries such as `googletrans`, `textblob`, `goslate`, etc for translation of non-english entries to english, but found that all of them had size limitations & were able to proceed with translation. To overcome this limitation, a wordlist of non-english words was formed from the dataset. All the rows from the description column were filtered using German wordlist and have been translated to English language by passing to a Google translator.

Initial thought was to combine the 'Short Description' & 'Description' field with the assumption that the vocabulary from the 'Short Description' could help in model accuracy. But found that combining these two fields could lead to combining non-english 'short Description' to 'Description' in english or vice versa. This posed a problem of many combined entries to be not translated. To further improve the translation process, we attempted to measure the impact of model accuracy on dropping the 'Short Description'. By dropping the 'Short Description' field we only observed a minor drop in model accuracy, ~1% drop & hence concluded to proceed with dropping 'Short Description'.

Word Distribution

A plot has been created to analyse the distribution of words in each ticket. It has been found that most of the descriptions of the problems raised by callers are short within 0-100 words. Few entries are a bit descriptive.



Lemmatization & Stop words removal

Stop words have been removed using nltk corpus modules.

Lemmatization is the process of grouping together the different inflected forms of a word so they can be analysed as a single item. Lemmatization is like Stemming but it brings context to the words. So, it links words with similar meanings to one word. Here we have preferred Lemmatization over Stemming because lemmatization does morphological analysis of the words.

Spell Check

We have used pySpellchecker to perform spell check on the data. But there were few technical words which were also corrected with this function. Eg. Hostage for hostname, sky for skype, wife for wifi, etc. So, a set of exceptional words have been loaded with such IT related technical words.

We found that performing spell check was a time-consuming process. Although spell check helped in reduction of vocabulary size, it did not help in model accuracy improvement. Hence we decided against applying spell check as it did not provide any substantial improvement to the whole process.

Deciding Models and Model Building

Modelling

Traditional ML Models

Random Forest Classifier Model

Random forests is a supervised learning algorithm. It can be used both for classification and regression. It is also the most flexible and easy to use algorithm. It creates decision trees on randomly selected data samples, gets prediction from each tree and selects the best solution by means of voting. It also provides a good indicator of the feature importance.

Support Vector Machines (SVM) Model

A support vector machine (SVM) is a supervised machine learning model that uses classification algorithms for two-group classification problems. After giving an SVM model sets of labelled training data for each category, they're able to categorize new text. This is a fast and dependable classification algorithm that performs very well with a limited amount of data.

Observation from the results of the traditional models

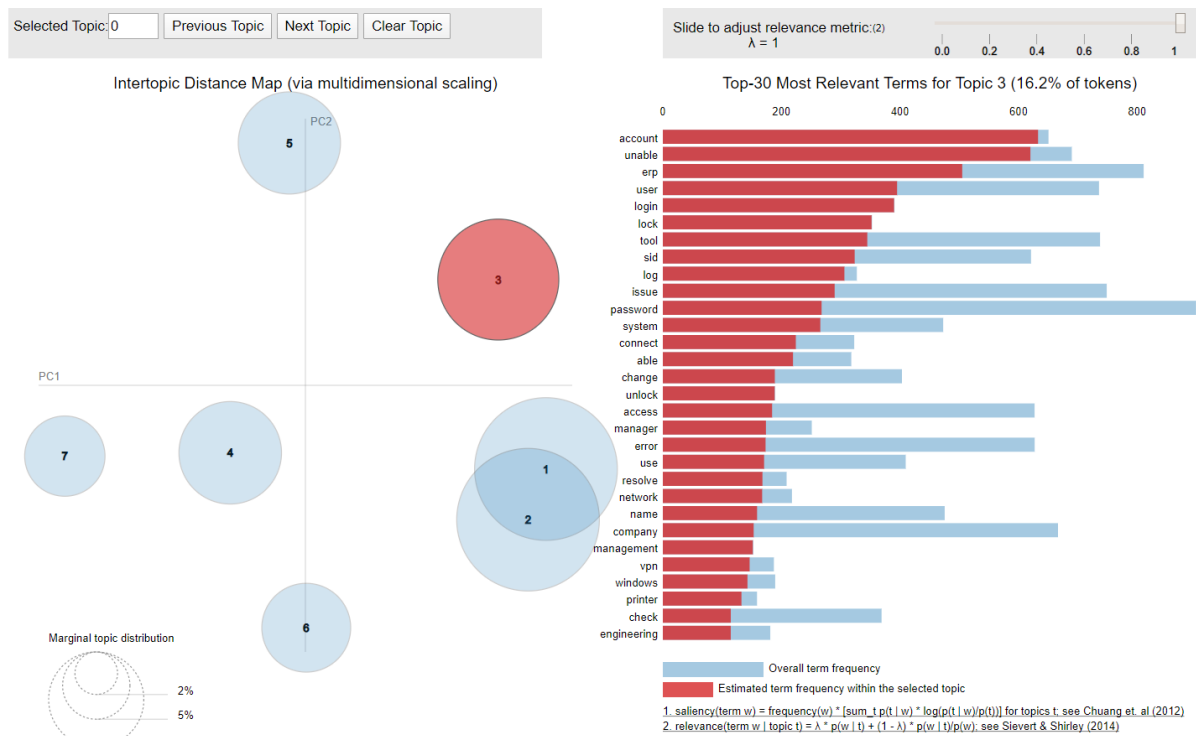
From the predicted accuracies, we could see that the traditional models with the resampled augmented dataset is performing with accuracy of 57% and 55% respectively. So, we believe the Neural network-based NLP model such as the LSTM and GRU models will give better results, as it is faster and it takes care of the many problem of the traditional models.

Model Clustering

Spacy module is used for lemmatization. PyLDAvis module is installed and applied for the plot to provide reasonable information on the data based on the topics. Bigram models are used to cluster relevant data together using GenSIM.

7 key topics have been extracted from the model. Relevancy of each topic from the provided data set is ~56% .

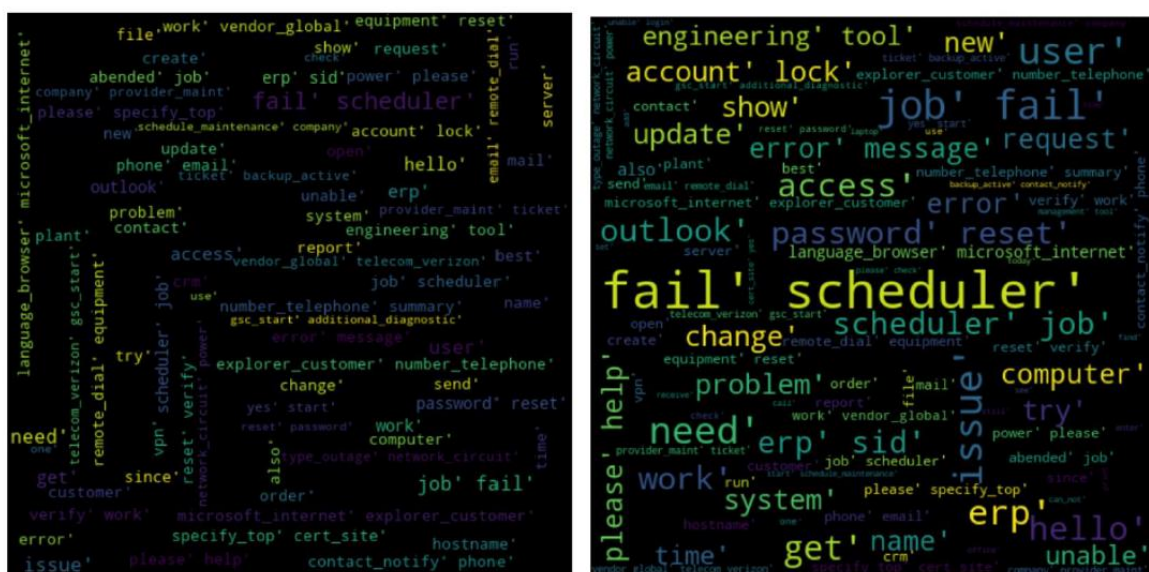
Copora dictionary has been created from the bigram words.



Analysis using WordCloud

WordCloud is a data visualization technique used for representing text data in which the size of each word indicates its frequency or importance. Significant textual data points can be highlighted using a word cloud. WordClouds have been generated with all available words & top 100 words. We have also inferred few observations over the target class – Assignment groups with word clouds for top 50 words from each group.

Word clouds for all words and top 100 words are as shown below.



Observations from WordClouds for Top 10 Assignment Groups

Assignment Group	Most Common Words	Observations
Grp_0	password, reset, unable, account, email, explorer_customer, engineering, number_telephone, microsoft_internet, management, collaboration_platform, outlook, crm, erp, lock, phone, skype, computer, error, pron, change, vpn	User account, browser related issues
Grp_8	vendor, equipment, power_provider, outage, job, scheduler, gsc_start, top_cert, contact_notify, site, specify_outage, schedule_maintenance, additional_diagnostic, telecom_Verizon, remote_dial	Network communications related
Grp_24	Problem, defective, tool, printer, computer, setup, install, new_ws, work	System related issues (most words were in German before translations)
Grp_12	server, hostname, drive, folder, access, file, inside_outside	Server related issues
Grp_9	job, fail, scheduler, abended, report	Job Scheduler related issues
Grp_2	sid, event, transaction_code, http, com, pron, message_recreate, authorize, condition_nsu, sle_inspector, access, attached, result, content, hrp, erp	Message/ web related
Grp_19	computer, laptop, printer, monitor, network, connect, software, office, system, pron, error, inside_outside, install, contact, detail, dell	System & Hardware related issues
Grp_3	boot, file, window, location, drive, run, document, computer, printer, update, dell, client, tool, replace, outlook, pc, erp, email, inside_outside, user, new, phone, print	System/OS related
Grp_6	job, fail, scheduler, abended	Job Scheduler related issues (similar to grp_48)
Grp_13	billing, sale, price, inwarehouse, delivery, material, customer, tool, block, can_not, quote, item, fix, time, unable, receive, erp, correct, note, print, send, system, check	Sales related issues

Model evaluation

Using the NLP models, from the predicted accuracies, we could see that the LSTM and GRU models with the resampled augmented dataset is performing well with more than 90% accuracy. Although the differences in the accuracy are marginal, we have decided to go with the GRU model as it is faster than LSTM and it takes care of the vanishing gradient problem.

Comparison to benchmark

From the given problem description, we could see that the existing system is able to assign 75% of the tickets correctly.

So, our objective here is to build an AI-based classifier model to assign the tickets to right functional groups by analysing the given description with an accuracy of at least 85%.

From the prediction results we see that the GRU model based on the resampled data is able to achieve an accuracy of 91.24% which is above our benchmark.

How to improve your model performance?

Prior to improving the model performance, we better understand the improvement areas of the current model. We can assess the best suited parameters for our model by using Grid Search CV and Random Search CV techniques. We intend to use these techniques more comprehensively in the subsequent part of the project. Further as detailed below if some of the limitations of the data can be addressed, we can obtain a model with a superior accuracy.

Implications

Although this model can classify the IT tickets with 91.24% accuracy, to achieve better accuracy in the real world it would be good if the business can collect additional data around 300 records for each group.

Limitations

As part of Data pre-processing, we had grouped all assignment groups with less than 10 entries as one group (misc_grp) which had reduced the Target class from 74 to 50 groups. While applying this model in the real world there could be additional intervention required to classify the tickets if it has been classified as misc_grp by our model. Since the number of elements reported under misc_grp is less, we expect this intervention to be done less often.

Suggestions for feature model improvement

We found the data was present in multiple languages and in various formats such as emails, chat, etc bringing in a lot of variability in the data to be analysed. The Business can improve the process of raising tickets via a common unified IT Ticket Service Portal which reduces the above-mentioned variability. By doing this, the model can perform better which can help businesses to identify the problem area for relevant clusters of topics.