AUTOMATIC TICKET ASSIGNMENT

Capstone Project: NLP

A Project Report submitted in partial fulfillment of the requirement for the award of the degree of PGP AI & ML program

Shwetha Rao, Lekshmi P, Kapil Soni,
Harish Hasti, Harvinder Bajaj

10/09/20

Table of Contents

Table of Figures

Table of Tables

INTRODUCTION

PROBLEM STATEMENT

Most IT organizations spend 80% of their resources and budget on maintaining existing infrastructure to ensure business continuity [1]. This is also called **"Keeping the lights on"**.

An IT ticketing system helps support teams document and solve technical problems using tickets. The main goal here is to ensure quick restoration of service operations. Thus, IT Ticketing Systems play a central role in "Keeping the lights on". However, today the following problems are seen in most incident management systems:

- The existing ticket classifying system is manual which is time consuming and is prone to human error.
- It leads to decreased user satisfaction due to increased response and resolution time.
- Dedicated resource to manually allocate tickets, leading to increased cycle time.
- Around 25% of the queries are being wrongly assigned and the functional team is spending additional effort to correct this.
- In a survey conducted among 900 CIOs, 77% of the leaders stated that the biggest roadblock to innovation and business transformation is overspending on "Keeping the lights on" [2].

OBJECTIVE

- Build an AI-based classifier that assigns tickets to right functional groups.
- Aim is to achieve > 80% accuracy i.e. higher than current manual accuracy of 75%.
- Help organizations reduce the resolving time of the issue and increase user satisfaction.
- Free up bandwidth of the teams to focus on more productive tasks.

OBSERVATIONS ABOUT DATA

1. Data is available in excel format consisting of 8500 rows and 4 columns. Following are the 4 columns:

Table 1 Data Column Details

Column Name	Details
Short Description	One liner of the issue
Description	Detailed issue description
Caller	Unique caller identifier
Assignment group	Group to which the issue is assigned

2. Column Assignment group is the target class which specifies to which class this issue belongs. Rest are input variables.
3. There are total of 74 groups (GRP_0 to GRP_73)

a. Data is highly imbalanced and skewed. Around 46% of the dataset is represented by just 1 class GRP_0
b. There are 72 groups with less than 500 records
c. There are 58 groups with less than 100 records
d. There are 25 groups which have less than 10 records

4. There are other languages present in the data. Eg:-German
5. Spellings mistakes are present
6. There are inconsistencies in data -symbols, blank spaces, image path, Urls, newline and tabs etc.
7. Format in description and short description columns are not consistent. Following are two examples out of many different format
    a. Some of the records contain complete mail including "Received From" details in the description column. There is a possibility that automation is done to convert mail on specific mail id to incident ticket directly.
    b. Some of the tickets are created from an event generated by IT systems like Oracle database.
    Example: Record 7 in the excel sheet is as follows-
    event: critical:HostName_221.company.com the value of mountpoint threshold for **/oracle/SID_37/erpdata21/sr3psa1d_7/sr3psa1d.data7,perpsr3psa1d,4524 is 98**

Current Workflow

Following image shows current workflow where ticket is created by user and it is manually assigned by a dedicated resource (ticket assigner) to respective group.

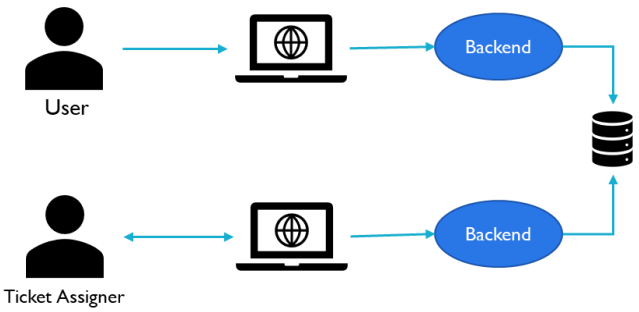


Figure 1 Current ticket assignment workflow

HIGH LEVEL DESIGN

PROPOSED WORKFLOW

In the proposed workflow Prediction pipeline code(python) can be integrated with the IT ticketing tool. This tool can trigger the prediction pipeline on receiving the new ticket and prediction pipeline and return back the group predicted. Based on this ticketing system can update the group in ticket raised. Interface between IT ticketing system and prediction pipeline can be REST or gRPC or any other mechanism and is not in scope of the current document.

In case this group is predicted as GRP_XX then it has to be manually assigned by ticket assigner. Considering GRP_XX shall have less than 10% tickets to resources required shall drastically reduce and allocation time for 90% of tickets shall be done in few seconds only.

Model can be deployed either on-prem or can be deployed on cloud. It depends on where the IT ticketing system is deployed and how it can be integrated with deployed model. To define deployment aspects is out of scope of this document.

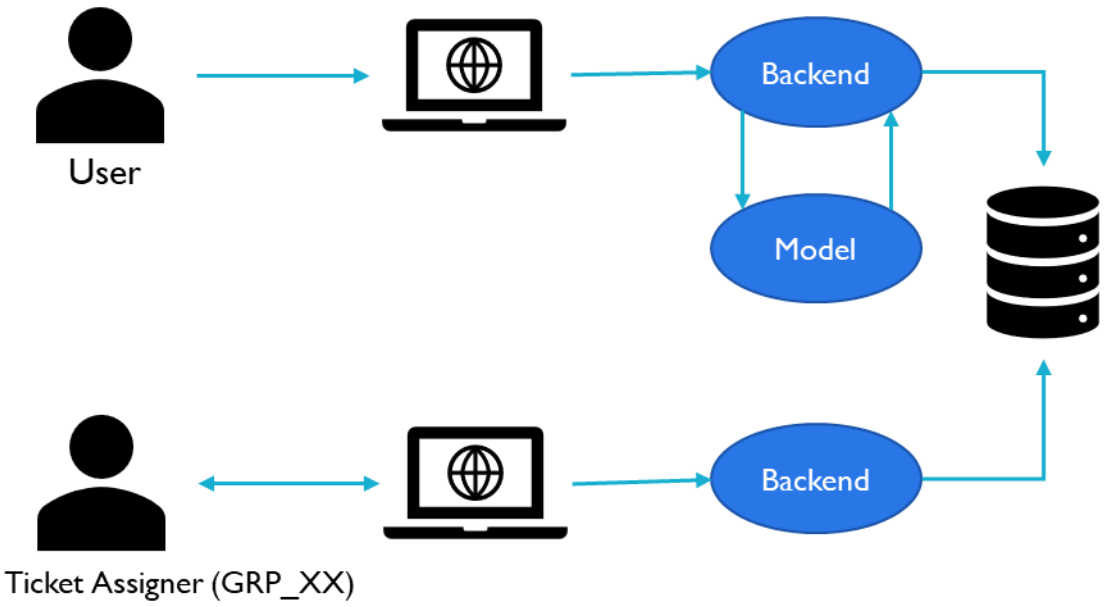


Figure 3 Proposed ticket assignment workflow

Problem involves developing a classification model to identify the group to which an incident ticket belongs. Incident ticket contain short description and description along with caller fields. Columns used for model shall be short description and description which are columns containing text. So, NLP shall be used to handle the data.

Following figure shows High level design for this problem which has 2 parts, first one for offline model training and seconds part consists of deploying the model and making online predictions by integrating with IT ticketing system. In following section details for model training are provided. Prediction pipeline has only Text Pre-processing and Feature engineering part which are same as in offline model training and then output of it is fed to model to do the prediction which is returned back to IT ticketing system for updating in the ticket.

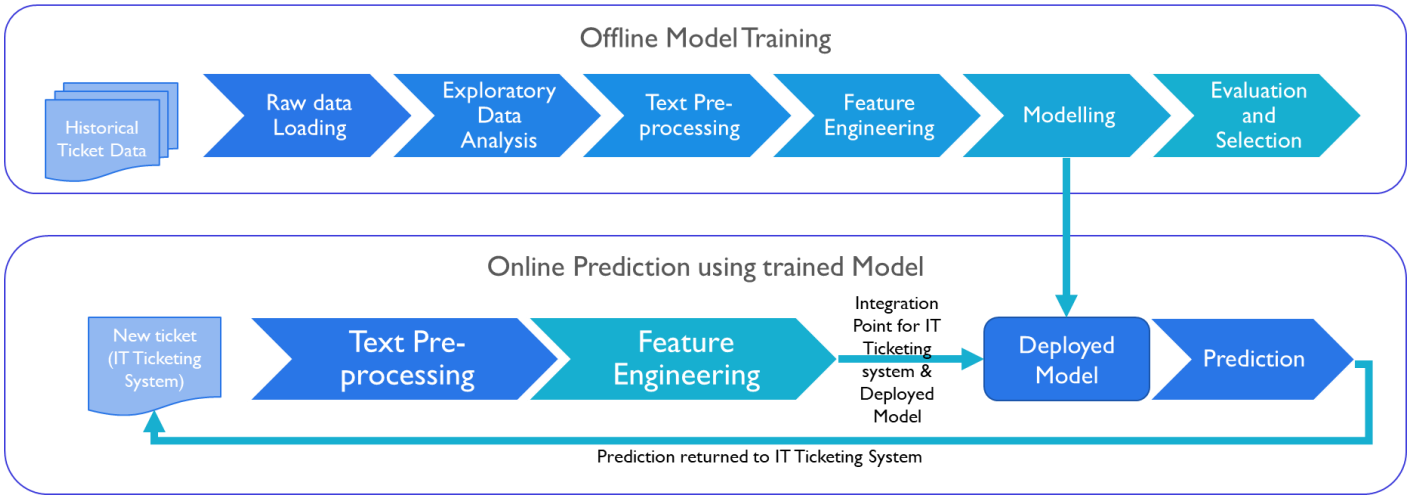


Figure 2 High Level Design

Following steps are followed to develop model using given dataset

1. **Load the data set**: Data set (xls file) is hosted on google drive. Google drive needs to be mounted and data set is be loaded using Pandas library in form of DataFrame.
2. **Exploratory data analysis** is done on loaded data. For exploring the data Pandas library is used and for visualization purposes Matplotlib and Seaborn libraries are used.
3. **Data pre-processing** is done to make data ready for modelling. For data preprocessing RegEx, Spacy library, NLTK, Gensim and FuzzyWuzzy libraries are used.
4. **Imbalanced data** may impact the model performance, so Data augmentation is done to have reasonable balanced data.
5. **Feature Selection** is done, considering which features don't add value in modelling like Called field in current dataset. So, such fields are removed.
6. **Modelling**: Now data is ready and model is trained using data. Both traditional models and deep learning models fitted to evaluate which one performs best for this problem.
7. **Model Evaluation**: All the models are evaluated on test data and based on selected evaluation criterion, models and compared and best one is selected.
8. **Model Deployment**: Selected model is deployed and used for prediction on new incoming incident ticket for automatics assignment. Web frontend can be provided to

raise the ticket and, in the backend, classification can be done and result of classification can be shown to user in real time.

LOW LEVEL DESIGN

Low level design add details to the steps defined in high level design for offline model creation.

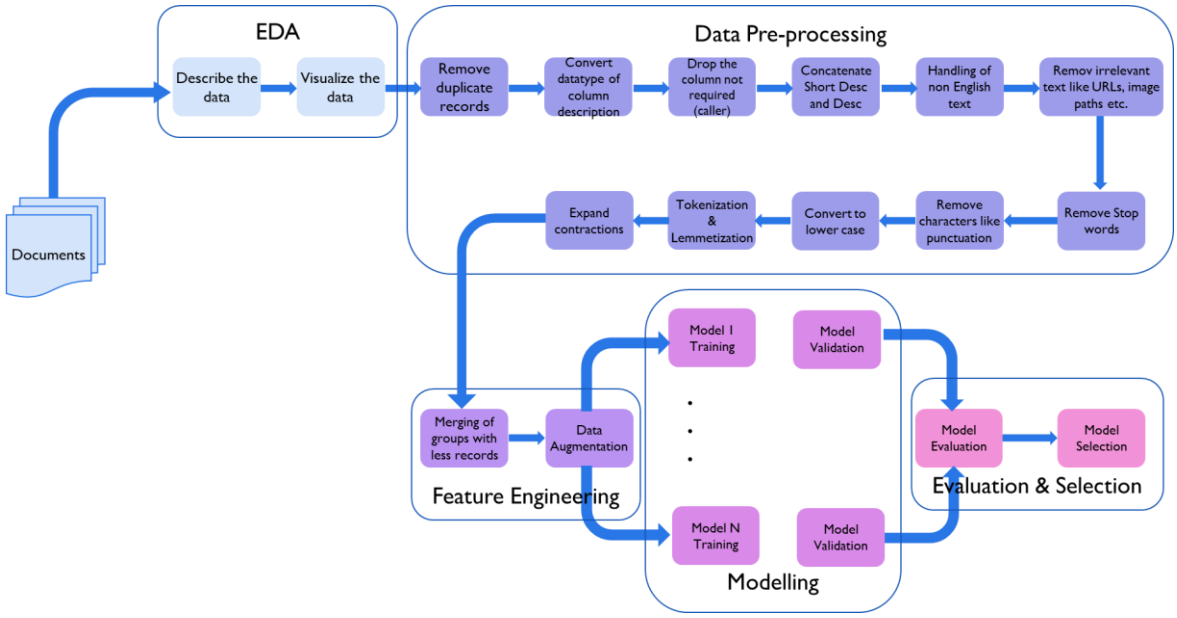


Figure 4 Low Level Design

EXPLORATORY DATA ANALYSIS

1. Describe the data, which provides details about each column like number of records in each column, unique records in each column and most frequent value for the column along with its frequency. This provides overview about the data w.r.t to records that have no value, duplicate records considering only specific column.
2. After describing data, visualization is done which can provide insight into data from different perspectives. For current data set
    a. Distribution of records across groups can be analyzed
    b. Word cloud can be visualized for different groups showing top keywords for the group
    c. Number of tokens in each record, at data set level, can be visualized and same can be visualized on group basis also.

DATA PRE-PROCESSING

1. **Remove duplicate records**: Identify the duplicate records (same value for all columns) and remove those records. These records can be generated by mistake like Clicking the submit button twice.
2. **Convert datatype of column**: As it is NLP tasks, so column should be of string type. So required columns are converted to string type using **astype** function of pandas.
3. **Drop the column:** Drop column which is not required e.g. Called field does not add any value so it should be dropped.

4. **Merging Short Description**: Short description can add value to modelling as it has unique keywords. So, concatenation of short description and description is done. Condition for concatenation is that if the short description is not part of description already then only it is concatenated.
5. **Language Translation**: Some of the records have non-English text. This text is translated using googletrans or similar library
6. **Cleaning of Data**: Handling of URLs, image paths which do not add any value to model training are removed from the text
    a. blanks and newlines are not adding any valuable information.
    b. images- there was no other information available other than the path to the image.
    c. hyperlinks and Urls were getting converted into encoded data after preprocessing and it was not adding any meaning to the analysis.
7. **Stop word removal** is done
8. **Remove Punctuation:** Punctuations do not add any value in model training, so are removed from the text
9. **Consistent case**: All words are converted to lower case
10. **Tokenization and Lemmatization** is done in the next step. Lemmatization ensures that different forms of word are converted to root form thus reducing the vocabulary size to handle.
11. **Handling Contraction:** Contractions like can't are expanded

After pre-processing, Group aggregation, for groups with few records and Data augmentation is done.

- **Model Training and Validation**: Now data is ready to be fed to different models which are selected for the problem solution. After model training, hyperparameter tuning is done and models are validated with test data. This provides us values for different parameters, mentioned in Hyper Parameter Tuning section, which helps us in selecting the final model to be recommended for this business problem.

EXPLORATORY DATA ANALYSIS

LOADING OF DATA

- Data is hosted on Google Drive.
- As data is in excel format, data can be loaded using the read_excel function of Pandas library. This shall provide the data frame with shape (8500, 4).

EXPLORATORY DATA ANALYSIS (EDA)

EDA consists of exploring the data and visualizing the data. So, as part of EDA, we explored following areas

1. Describe data frame to get information about each column like count of records in column, number of unique values, most frequently occurring value in column and its frequency.
   - Out of 8500 records, 9 records do not have a short description.
   - 1 record does not have description column
   - There are 2950 unique callers
   - There are 74 Assignment groups
   - 'Password reset' is the most frequent short description

	Short description	Description	Caller	Assignment group
count	8492	8499	8500	8500
unique	7481	7817	2950	74
top	password reset	the	bpctwhsn kzqsbmtp	GRP_0
freq	38	56	810	3976

Figure 5 Description of data column

2. Check for number of duplicate records in data frame
   - There are 140 duplicate records in data
3. Check for data types for dataframe
   - There are 4 columns of datatype object

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

Figure 6 Information about the features

4. Check for special characters or non-english characters in the description and short description columns
5. Visualize frequency plot for each target class
   - GRP_0 has ~4000 data records followed by GRP_8 with ~650 records

- There are total of 74 groups. Almost 60 groups have less than 100 data records.

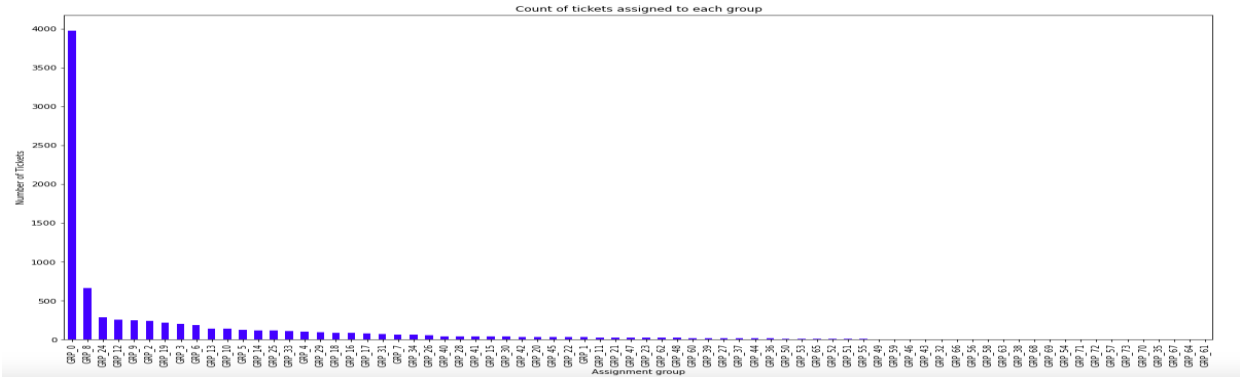


Figure 7 frequency plot for each target class

6. Visualize frequency plot for each target class after regrouping.
   - Target classes were regrouped as there was a high imbalance in data.
   - 22 groups with the maximum number of data records were left intact.
   - All the other records were merged, and a new group named GRP_XX was created.
   - The new group has ~850 data after the re-grouping.
   - Since GRP_XX contains less than 10% of total data any tickets assigned to GRP_XX by the model can be inspected manually to be assigned to the appropriate class. This will only take 1/10th of resources and bandwidth when compared with manual assignment of all the tickets done before automation.

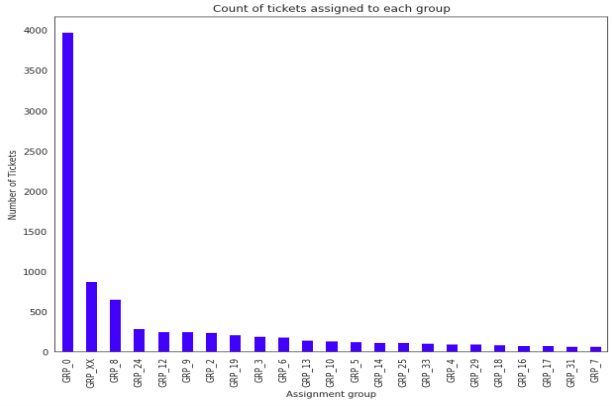


Figure 8 Frequency plot for target classes (after re-grouping)

7. Visualize Word cloud for each target class
   - For GRP_0, password, gmail, reset, company, outlook, account lock etc are the frequent words
   - For GRP_8, yes, na, monitor, job, scheduler, tool, company etc are the most frequent words.
   - From the word cloud it is clear that GRP_0 receives IT infrastructure related issues, which is used by every employee of company, and that is why this group has the maximum number of issues.

- GRP_8 has keywords related to job scheduler and monitoring application which may be specific to a team, so number of records are less as compared to GRP_0. Also, keywords in both groups are quite distinct which helps in distinguishing these 2 groups easily.

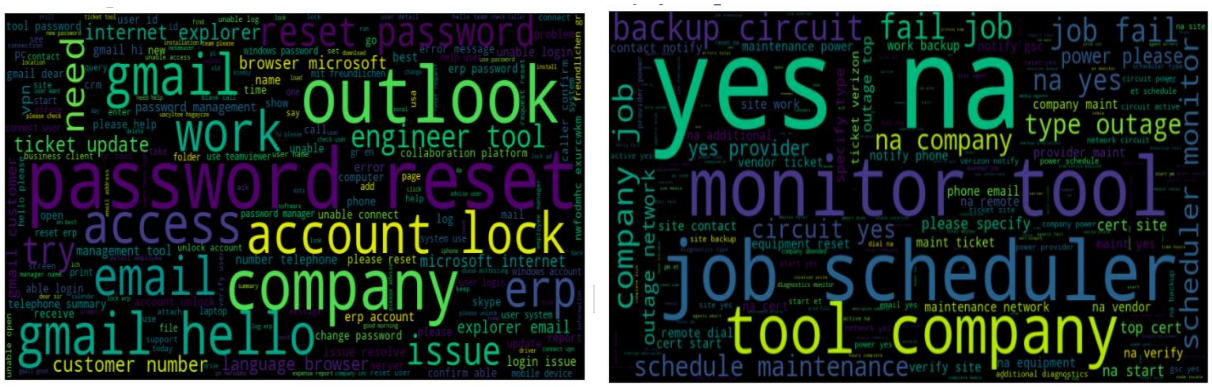


Figure 9 Word cloud for group_0 & Group_8

8. Visualize the number of words in the description column of each record. This can be used to identify the size of the description column to use while doing padding.

- The number of words in the entire data set and in each group is analyzed.
- 3 groups are taken to check whether the distribution of words in each group is different from the distribution of the entire data.
- No. of records with less than 10 words is max for both the entire data set and for each group. As the number of words are limited it will be easier to identify the key words that represent a particular group.

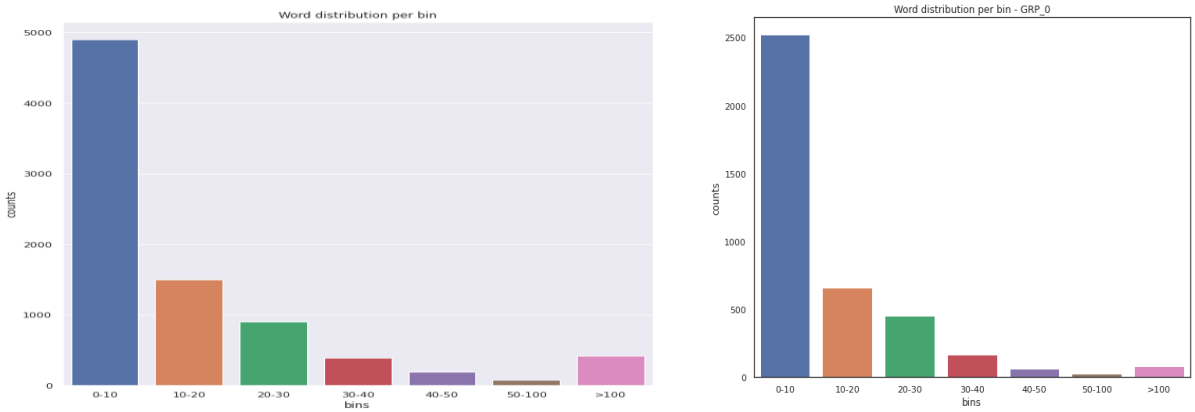


Figure 10 Word distribution per bin(over the entire data) and GRP_0

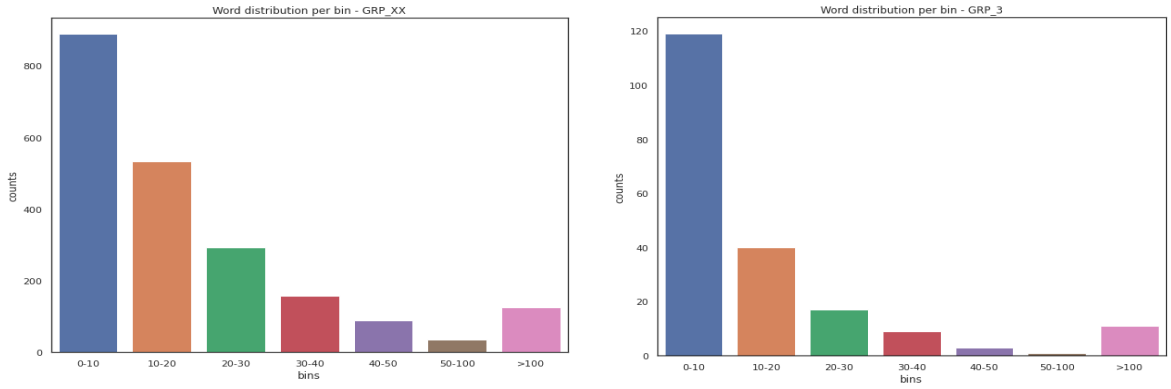


Figure 11 Word distribution per bin(GRP_XX & GRP_3)

DATA PRE-PROCESSING

Data preprocessing involves preparing the data for model training. Data that we collect will be unclean to be used directly to the model(fig.9).

	Short description	Description	Caller	Assignment group
0	login issue	-verified user details.(employee# & manager na...	spxjnwir pjlcoqds	GRP_0
1	outlook	received from: hmjdrvpb.komuaywn@gmail.com...	hmjdrvpb komuaywn	GRP_0
2	cant log in to vpn	received from: eylqgodm.ybqkwiam@gmail.com...	eylqgodm 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

Figure 12 Raw data as collected from various sources

For preparing the data all the preprocessing steps as detailed in Data pre-processing . Before data preprocessing text of one of the record is like as shown in Figure 13

```
data['Description'][21]
```

```
'    received from: ugephfta.hrbqkvij@gmail.com    hello h
elpdesk    i am not able to connect vpn from home office.
couple f hours ago i was connected, now it is not working
anymore. getting a message that my session expired but if
i click on the link, nothing happens.    [cid:image001.jpg
@01d233aa.3f618be0]    **********    need help
with your dynamics crm?  click here<    chat with a live a
gent regarding your dynamics crm questions now! click here
<    best '
```

Figure 13 Description column (before data pre-processing)

After all the data preprocessing steps, the description column (Figure 14) will be ready to be fed to the model for training along with the label column.

```
data['Cleaned Description'][21]
```

```
'hello helpdesk    i am not able to connect vpn from home
office  couple f hours ago i was connected  now it is not
working anymore  getting a message that my session expired
but if i click on the link  nothing happens      cid
need help with your dynamics crm   click here     chat wit
h a live agent regarding your dynamics crm questions now
click here     best'
```

Figure 14 Description column (after data pre-processing)

MODELLING

IMBALANCED DATA

Once data is preprocessed then it is ready to train the model. Considering this is imbalanced data , we have used following methods to handle imbalance in data

1. **Re-group data**: Top 29 groups were left intact and all the other records were merged, to create new group named GRP_XX.
    a. Since GRP_XX contains less than 10% of total data any tickets assigned to GRP_XX by the model can be inspected manually to be assigned to the appropriate class. This will only take 1/10th of resources and bandwidth when compared with manual assignment of all the tickets done before automation.
2. **Data Augmentation**: Use NLPAug library to create additional records by generating new records, for groups with less records, by using SynonymAug. This function replaces some of the words in sentence with its synonym and create a new sentence.

DATA MODELS

After this step, model needs to be identified which shall be trained with data
Following models are being considered for modeling this task

DEEP LEARNING MODELS

Artificial Neural Network	• ANN was selected because it is one of the deep learning models which can learn itself using data by minimizing the error. • Our problem which is text classification will benefit from its potential to reach high accuracy with less need of engineered features
Bi directional LSTM with GloVe	• The LSTM cell performs astonishingly well in NLP tasks.. • As our problem requires context based classification Bi-directional LSTM is selected. • GloVe is more accurate than Continuous Bag of Words or Word2Vec. So GloVe is selected as the embedding method in the project.

TRADITIONAL MODELS

SVM	• The complexity of the classifier depends only on the count of support vectors, not dataset size. • So it is one of the most suitable classification methods for our problem as it has a large number of class labels (74 groups which are re-grouped into 23 groups).
Naïve Bayes	• We are trying to achieve text classification and Naive Bayes works well on text data. • It also provides good performance when the training data is less and does not contain all possibilities.Since the data set which we are trying

	to model is highly imbalanced with less training data,Naive Bayes could be a good fit.
Random Forest	• The dataset which we are using is prone to overfit. So Random Forest which inherently avoids overfitting of data will be a good choice to be used as a model. • Our data is highly imbalanced, as Random forest classifiers handle the missing values and maintain the accuracy of a large proportion of data.

MODEL EVALUATION

Currently the results have been obtained on base models, in the next phase we are planning to optimize our models through hyperparameter tuning through Grid Search, Random Search and evaluate the models on various performance metrics.

The following parameters and metrics will be used to assess and further improve the performance of models:

1. **Accuracy**: It would give an insight into how accurate models are at predicting the correct group of a ticket given previously unseen data.
2. **Sensitivity:** It is known as the True Positive Rate. It is also known as Recall. Essentially, it informs us about the proportion of correctly classified groups.
3. **Specificity**: Specificity is known as the True Negative Rate. It informs us about the proportion of actual negative cases that have gotten predicted as negative by our model.
4. **F1-Score**: It is the harmonic mean of precision and recall. In the multi-class case, this is the average of the F1 score of each class. Taking the harmonic mean helps to penalize extreme values, thus making it a better measure than accuracy for our dataset which is highly imbalanced.
5. **Latency (Training and Prediction Time)**: Training time is the time taken by a model to train given the input data and Prediction time specifies the time taken for the model to provide output given a new data record for prediction. We will take into consideration these two parameters for selection of models as they have a direct impact on catering to the time criticality of the use case as well as the project budget. This task is not real time so we can use model which is taking more time but is giving better accuracy
6. **Robustness**: It is a measure of how good a model is at handling noisy data (e.g. raw data without any pre-processing). A robust model should provide reasonable accuracy even when raw data is used for training the model. So model performance on raw data will also be considered to measure the robustness accuracy of the model.

HYPER PARAMETER TUNING

Hyperparameter tuning was done on the models to find out the optimal set of parameters. These parameters help us to get an optimal model which minimizes a predefined loss function on given independent data. Such parameters express important properties of the model such as its complexity or how fast it should learn. Following table has the details of the parameters that were tuned in this project and their corresponding values.

Model	Function	hyperparameters	Value
Naive Bayes	MultinomialNB()	alpha	0.01

	TfidfVectorizer	ngram_range	(1,2)
		norm	l2
SVM	Glove	embedding size	300
	SVC()	c	10
		gamma	100
ANN	Glove	embedding size	300
	Sequential	activation	softmax
		batch_size	(128,512,64) -- best : 128
		beta1	(.7,1,.1) -- best .8
		dense units param 1	(600,1000,30) -- best: 870
		dense units param 2	(200,500,30) -- best 470
		learning rate	(.001,.01,.05) -- best .001
		spatial dropout	(0,.2,.1) -- best 0
		decay rate	0.02
		beta 2	0.999
Random Forest	Glove	Embedding Size	300
	RandomForestClassifier()	n_estimators	1,005,001,000
		max_features	sqrt,log2
		max_depth	100,None
		criterion	gini,entropy
Bidirectional LSTM	Glove	embedding size	200
	Sequential	batch_size	128
		LSTM units	200
		Dense layer neurons	100
		LSTM dropout	0.25
		LSTM recurrent dropout	0.25

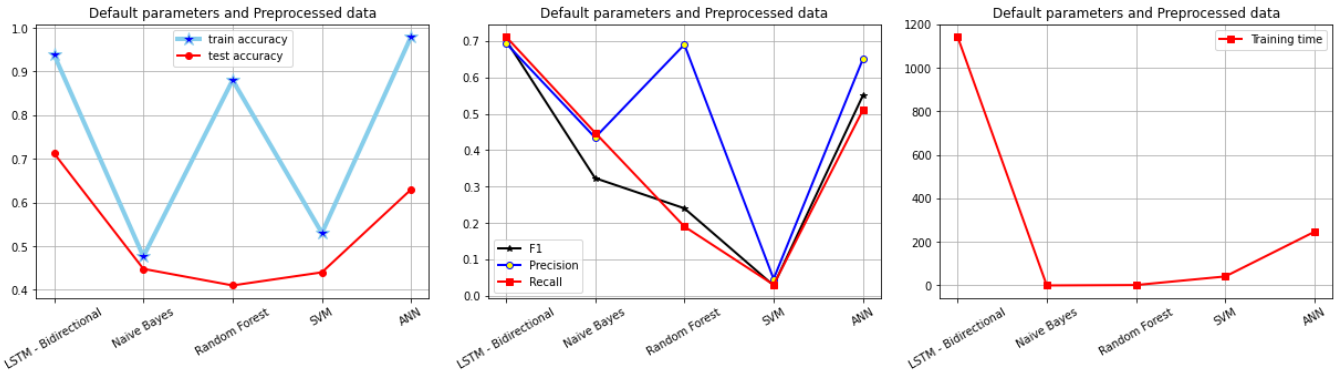
		spatial dropout	0.3
		Embedding trainable	TRUE
		learning rate	0.001

Observing the Hyperparameters for Bidirectional LSTM model from above table, We see values of some of them can be altered to get better prediction accuracy e.g. changing glove embedding size to 300 from 200 (more robust vector representation for words), changing LSTM units per layer from 200 to some higher multiple of output of word embeddings and including some more dense layers to get better intermediate representation of features. But this will lead to significant increase in training time and hardware infrastructure required to train the model.

Therefore, hyperparameters values are selected to get the best trade of between training time and model performance.

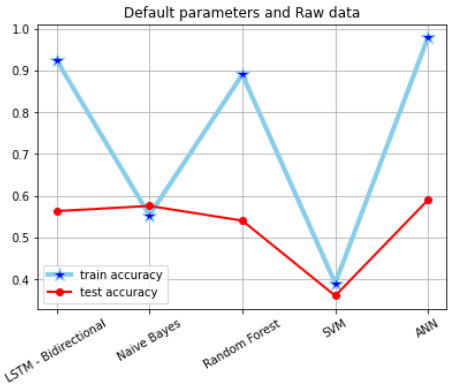
Model Comparison

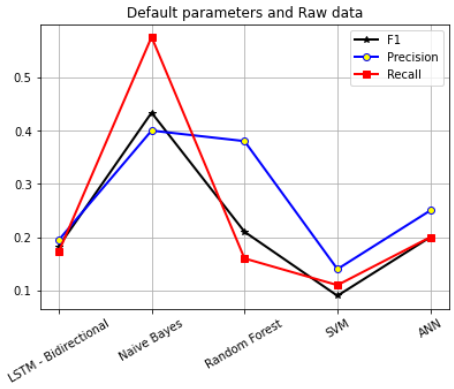
Model shall be evaluated on each of the above-mentioned parameter and a model shall be selected based on weightage of each of the parameter. Weightage can be defined based on the business case as defined by customer.

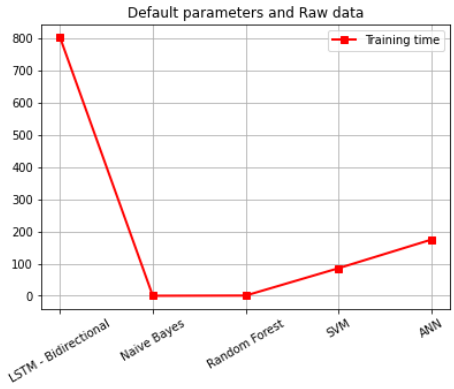


- Except for Naïve Bayes and SVM other models are overfitting to the training data. This is evident from big difference between training and test accuracy.
- Based on test accuracy, recall and precision value, Bidirectional LSTM is performing best. Its accuracy is closer to the manual accuracy of 75% in current system.
- Based on latency for training Bidirectional LSTM is slowest of all taking ~1150 seconds to train on set of 11148 records. Prediction time for each model is very low and all are in range of few seconds only.

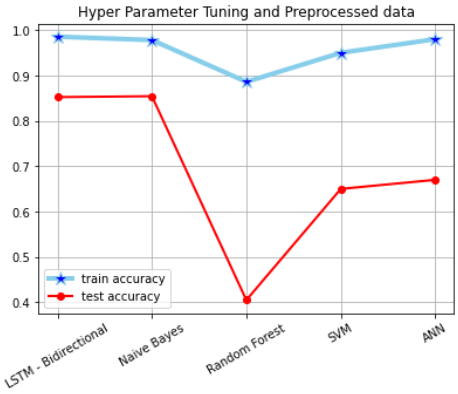
- Traditional models are not performing well for the current problem, but deep learning models are performing reasonably with default parameters.

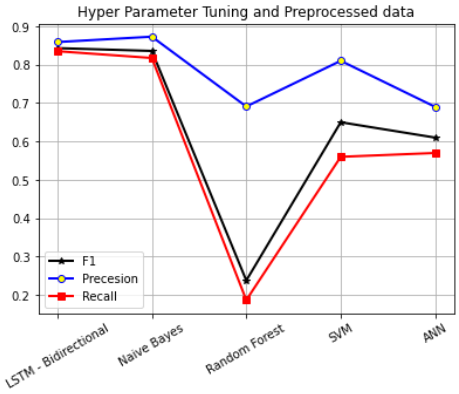


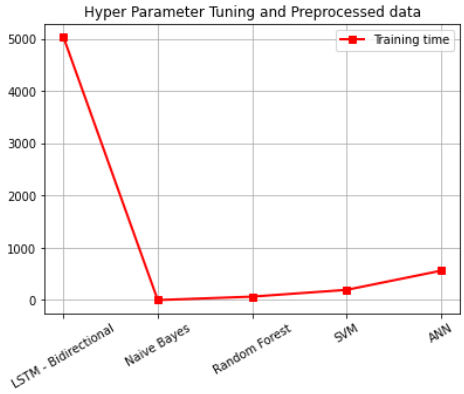




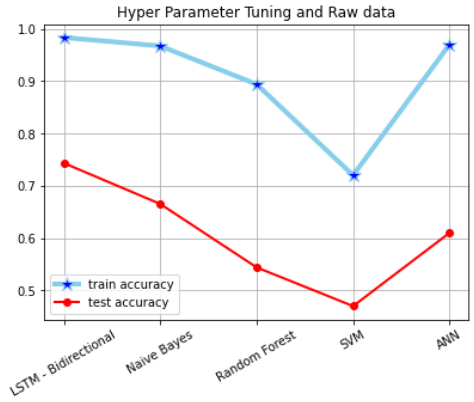
- Based on current Accuracy, Recall and Precision value, ANN is performing best. Its accuracy (~60%) is much lower than the manual accuracy of 75% in current system.
- Based on latency for training Bidirectional LSTM is slowest of all taking ~800 seconds to train on set of 7650 records and Prediction time for each model is very low and all are in range of few seconds only for 850 records.

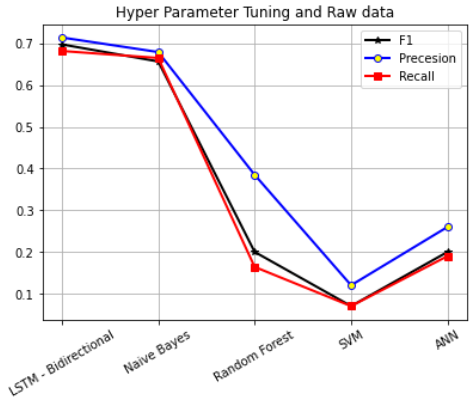


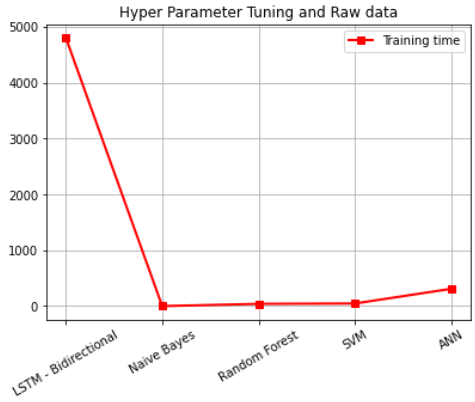




- Test accuracy for Bidirectional LSTM is ~85% which is much better than manual accuracy of 75% and above the target set(>80%) for this project. Test accuracy for Naïve Bayes is also very close to Bidirectional LSTM.
- F1 and Recall value for Naïve Bayes is slightly lower than Bidirectional LSTM
- Training time and prediction time is much better for Naïve Bayes as compared to Bidirectional LSTM.







- Test accuracy with raw data is highest for Bidirectional LSTM model. Other parameters i.e. F1, Recall and Precision are also highest for this model. So most robust model is Bidirectional LSTM.
- Training time is highest for Bidirectional LSTM and prediction time is almost similar for all , in range for few seconds only.

Final Recommendation

Based on the above model evaluation, it is clear that from Test Accuracy and F1 score perspective, Bidirectional LSTM is the best model. But training time for this model is highest. But considering the training time does not impact the run time performance of the model and is mostly done offline and then model is deployed so this is not critical factor for not selecting a model. Runtime performance is similar for all models (in range of few seconds) so recommendation is to use **Bidirectional LSTM** model for this task.

Table 2 Tools Used

S.No	Tool Name	Purpose
1	Pandas	DataFrame handling. Data is loaded from CSV using pandas and dataFrame is created. All the processing on the dataframe.
2	numpy	Numerical computing library to be used for any numerical operation on data
3	Tensorflow/Keras	Deep Learning models like ANN and Bidirectional LSTM are used from Tensorflow/Keras
4	Spacy	This library is used NLP preprocessing like tokenization, lemmatization etc.
5	Nltk	NLTK is used to get stop words list for removal of stop words
6	sklearn	SKLearn is used for Traditional ML model like SVM,RandomForest and Naïve Bayes
7	gensim	Gensim is used for loading of Glove data to be used for creating embeddings
8	BeautifulSoup	HTML text processing
9	matplotlib	It is used to various visualization done on data
10	seaborn	It is used to various visualization done on data
11	FuzzyWuzzy	It is required to find if a substring exists in another string. It provides percent

		match where 100% match mean complete substring exist in the string.
12	Googletrans	This library is used for translating languages other than English to English text
13	NLPAug	This library is used for data augmentation by generating new sentences by replaces words with its synonym. This helps in balancing the input data.

REFERENCES

1.https://www.servicenow.com/content/dam/servicenow/other-documents/ebook/downloads/How_todays_CIOs_deliver_strategic_IT.pdf.

2.https://www.businesswire.com/news/home/20180613005299/en/Global-IT-and-Finance-Leaders-Survey-Finds-Biggest-Blocker-to-Innovation-is-Overspending-on-%E2%80%9CKeeping-the-Lights-On%E2%80%9D.

3.https://medium.com/miq-tech-and-analytics/how-to-detect-non-english-language-words-and-remove-them-from-your-keyword-insights-599b91916071

4.https://www.revuze.it/blog/bert-nlp/#:~:text=Because%20BERT%20practices%20to%20predict,methodologies%2C%20such%20as%20embedding%20methods.

APPENDIX A

Code

capstone_sol_v2.py