

Automatic Ticket Assignment

Final Report

# Team Details

|  |  |
| --- | --- |
| Anand Sethurathinam KamalRaj M Praveen Galav Priya Rama G Vaibhav Kulkarni Vinayak Gokuldas Kudva | [anand.sethurathinam@gmail.com](mailto:anand.sethurathinam@gmail.com)  [kamalraj.m@gmail.com](mailto:kamalraj.m@gmail.com)  [pgalav@gmail.com](mailto:pgalav@gmail.com)  [ancpriya73@gmail.com](mailto:ancpriya73@gmail.com)  [ramag80@gmail.com](mailto:ramag80@gmail.com)  [vaibhavkulkarni9@gmail.com](mailto:vaibhavkulkarni9@gmail.com)  [vinayak.kudva@gmail.com](mailto:vinayak.kudva@gmail.com) |

[Team Details 2](#_heading=h.gjdgxs)

[**1.**](#_heading=h.2et92p0) **Summary of the problem statement, Data and findings** 5

[1.1 Problem](#_heading=h.tyjcwt) Statement 5

[1.2 Data & Findings 5](#_heading=h.3dy6vkm)

[**2.**](#_heading=h.1t3h5sf) **Overview of the Final Process** 7

[**3.**](#_heading=h.4d34og8) **EDA and Pre-Processing** 8

[**4.**](#_heading=h.2s8eyo1) **Visualization** 10

[4.1 GRP\_0 10](#_heading=h.17dp8vu)

[4.2 GRP\_8 10](#_heading=h.3rdcrjn)

[4.3 GRP\_24 11](#_heading=h.26in1rg)

[4.4 Plot showing Frequency Distribution of words 11](#_heading=h.lnxbz9)

[4.5 Inter-topic Distance Map 11](#_heading=h.35nkun2)

[4.6 Heat Map 14](#_heading=h.1ksv4uv)

[**5**](#_heading=h.44sinio) **Model building and evaluation** 15

[5.1 Model Approach 15](#_heading=h.2jxsxqh)

[5.2 Model creation 15](#_heading=h.z337ya)

[5.2.1 Logistic Regression Model 15](#_heading=h.3j2qqm3)

[5.2.2 Random Forest Model 16](#_heading=h.1y810tw)

[5.2.3 KNN Model 16](#_heading=h.4i7ojhp)

[5.2.4](#_heading=h.2xcytpi) LSTM Model 16

[5.2.5](#_heading=h.1ci93xb) Bidirectional LSTM with Time Distributed 17

[5.3 Model Summary 17](#_heading=h.3whwml4)

[**6**](#_heading=h.2bn6wsx) **Bench Marking - Comparison of experiments** 18

[6.1 Traditional Models 18](#_heading=h.qsh70q)

[6.1.1 Output of Logistic Regression 18](#_heading=h.3as4poj)

[6.1.2 Output of Logistic Regression with Balanced Data 18](#_heading=h.1pxezwc)

[6.1.3 Output of Random Forest 19](#_heading=h.49x2ik5)

[6.1.4 Output of KNN 19](#_heading=h.2p2csry)

[6.1.5 Summary of ML Models 19](#_heading=h.147n2zr)

[6.2 NLP Models 20](#_heading=h.3o7alnk)

[Approach to select the final model based on the outcome 20](#_heading=h.23ckvvd)

[6.2.1 LSTM 20](#_heading=h.ihv636)

[6.2.2 Bidirectional LSTM with Time distributed 22](#_heading=h.32hioqz)

[6.2.3 NLP Model Summary 22](#_heading=h.1hmsyys)

[**7**](#_heading=h.41mghml) **Implication - business value derived** 23

[**8**](#_heading=h.2grqrue) **Limitations and Scope of improvement** 23

[8.1 Limitations of data 23](#_heading=h.vx1227)

[8.2 Scope of Improvement 23](#_heading=h.3fwokq0)

[**9**](#_heading=h.1v1yuxt) **Closing Reflection** 24

[**10**](#_heading=h.4f1mdlm) **Final Note** 24

[**11**](#_heading=h.2u6wntf) **Code and References** 24

# **Summary of the problem statement, Data and findings**

## Problem Statement

In most of the organizations, incidents are created by various Business and IT Users, End Users/ Vendors in ticketing systems. Assigning the incidents to the appropriate person in the support team has critical importance to provide improved user satisfaction

Manual assignment of incidents is time consuming and requires human efforts. There may be mistakes in allocating the tickets to right group that causes delay in resolution of tickets. Manual assignment increases the response and resolution times which result in user satisfaction deterioration and poor customer service.

Applying traditional machine learning and neural network-based NLP to automatically classify tickets and assign them to the right owner in a timely manner to save effort, increase user satisfaction and improves handling of support ticketing system.

## Data & Findings

The input data provided as an excel sheet. It has the following details:

|  |  |
| --- | --- |
| **Short description** | A summary of the issue faced by the user |
| **Description** | Detailed description of the issue |
| **Caller** | Caller details who logs the call |
| **Assignment group** | GRP\_0 ~ GRP\_73 (total 74 classes of Assignment group) |

**Sample Data:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Short description** | **Description** | **Caller** | **Assignment group** |
| login issue | -verified user details.(employee# & manager na... | spxjnwir pjlcoqds | GRP\_0 |
| Outlook | \r\n\r\nreceived from: hmjdrvpb.komuaywn@gmail... | hmjdrvpb komuaywn | GRP\_0 |
| cant log in to vpn | \r\n\r\nreceived from: eylqgodm.ybqkwiam@gmail... | eylqgodm ybqkwiam | GRP\_0 |

**Data Findings:**

1. High imbalance seen in data with Group - GRP\_0 having 47% of representation
2. 73 Groups constitutes only 53%
3. Data has Null values:

|  |  |
| --- | --- |
| Columns | Count of NULL values |
| Short description | 8 |
| Description | 1 |
| Assignment group | 0 |

1. Caller data is available as gmail address in Description. We can generated email addresses and remove them from description. Also we will then append caller information to description after checking that its not already present.
2. Observed certain Short descriptions are same as Description. We can join both Short description and Description as a single column. It helps us to classify the tickets effectively.
3. Observed almost 9% of Non-English ticket descriptions

# **Overview of the Final Process**

The brief approach for the solution is given below

1. Solution requires model building based on the Classification model approach to predict the ticket details and assigned to expected group for quicker resolution.
2. Data cleansing and Pre-Processing are important to have a good cleaned input dataset for the model to predict the expected output. Hence the data cleansing and pre-processing steps are given in a detailed manner.
3. Visualization has been given to understand the dataset that feed into the model. This also helps to understand the structure of dataset
4. Two approaches of model creation is defined.

The first one is based on the conventional Machine learning algorithms Logical regression, Random Forest, KNN.

The second approach is based on the NLP algorithms LSTM and Bi-directional LSTM.

The evaluation approach is given as well.

1. The benchmarking of outcome has been captured. The performance of the model is tuned based on the different iterations with different parameters
2. The business derived value based on the outcome of the model is analysed.
3. Limitations of the model and scope of improvement has been covered.
4. The lessons learnt on each of the step of the project is noted down and summary is provided as Learnings

# **EDA and Pre-Processing**

Below are the Pre-Processing steps applied while performing Elaborate Data Analysis on the input data.

1. Removal of rows that Null values - Impacted 9 rows and removed
2. Uses Caller column to generate gmail addresses and clean the description that has the email addresses.
3. Observed few rows of both Short description and Description has trailing spaces which affects analysis. Hence removed the trailing spaces
4. Observed line breaks, tabs and special characters that expects to make noise. Removed the same.
5. Merged both Short description and Description columns into one which can help model to arrive at a good decision.
6. The caller column is then appended to the description
7. Most of the groups have less data but it expected to make more noise as the data is imbalanced in counts. Hence removing the rows that has lesser than threshold value of 31.

This constitutes 95% of data.

1. The augmented description column is copied into 2 new columns. ML\_Description – for the traditional ML models and NL\_Description that will be used for the Neural network models.
2. Using fast text library, the non-English descriptions are identified and removed the rows.

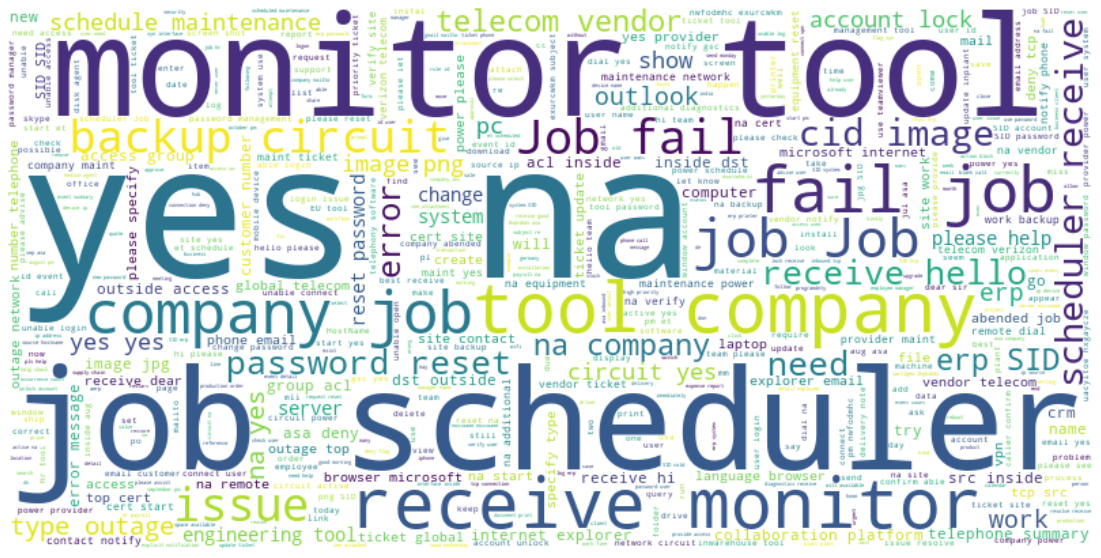
88% of data are English and others are detected as different language.

1. Punctuation marks part of English grammar are retained for the NL Model
2. All special characters are removed for the traditional models
3. Using nltk - POS and Lemmatizer, we have further enhanced the ML\_Desription data to help the model to predict effectively.
4. Still we observed non-ascii characters using word cloud that makes noise, hence removed them.
5. In Description, English STOP words, all special characters are removed for the traditional models

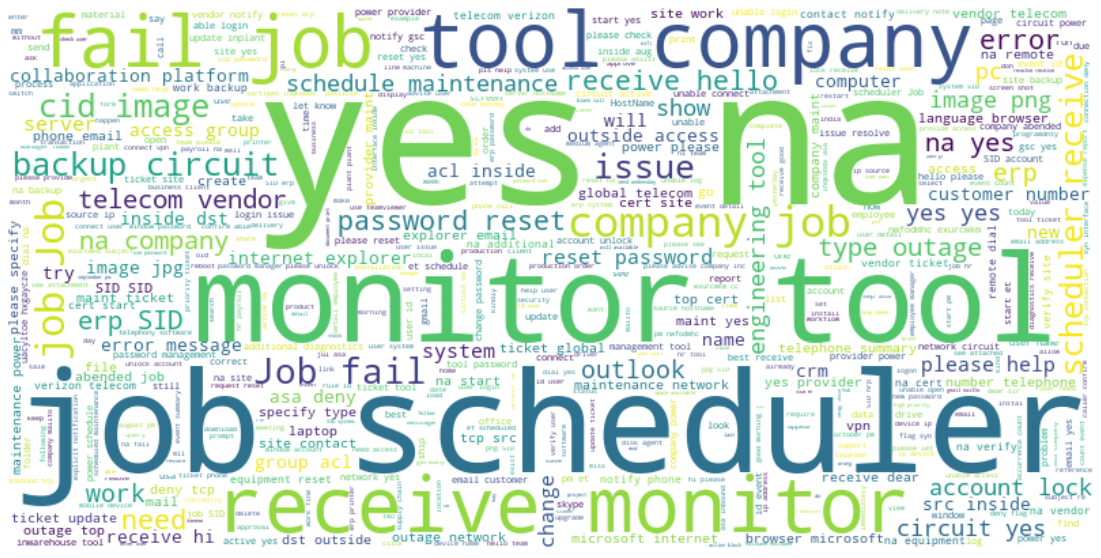
# **Visualization**

Top 3 Groups – in Word Cloud

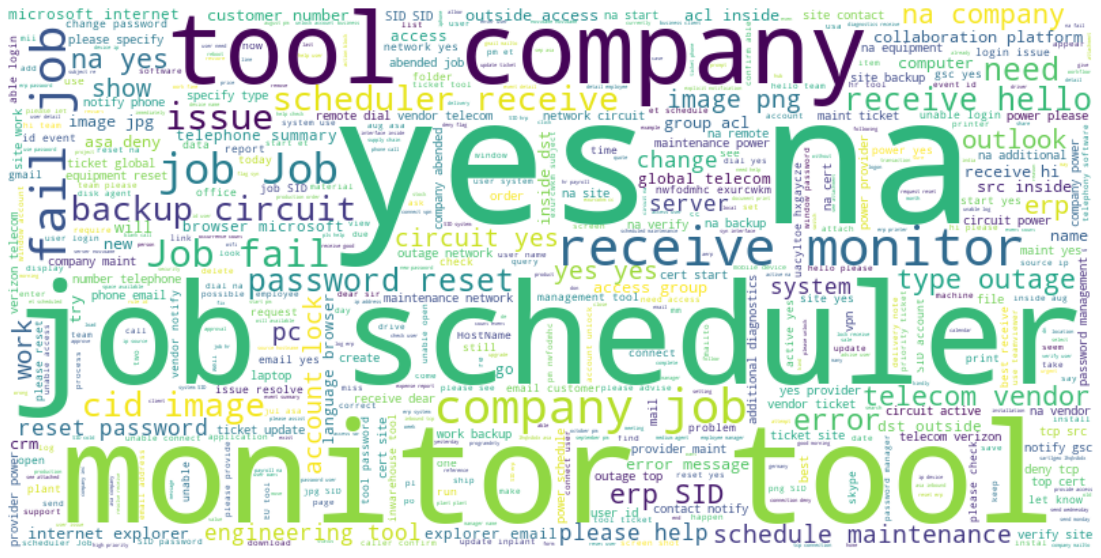
## 4.1 GRP\_0



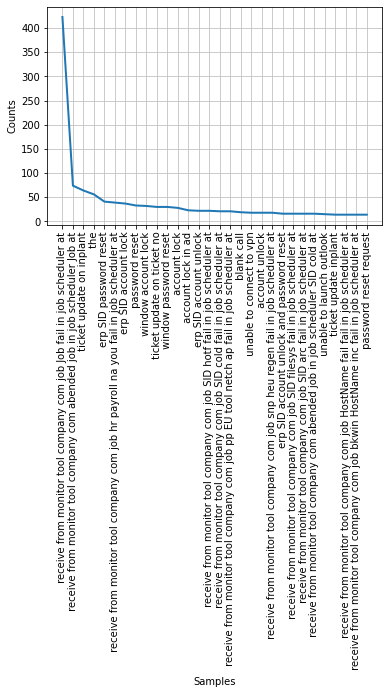
## 4.2 GRP\_8



## GRP\_24

****

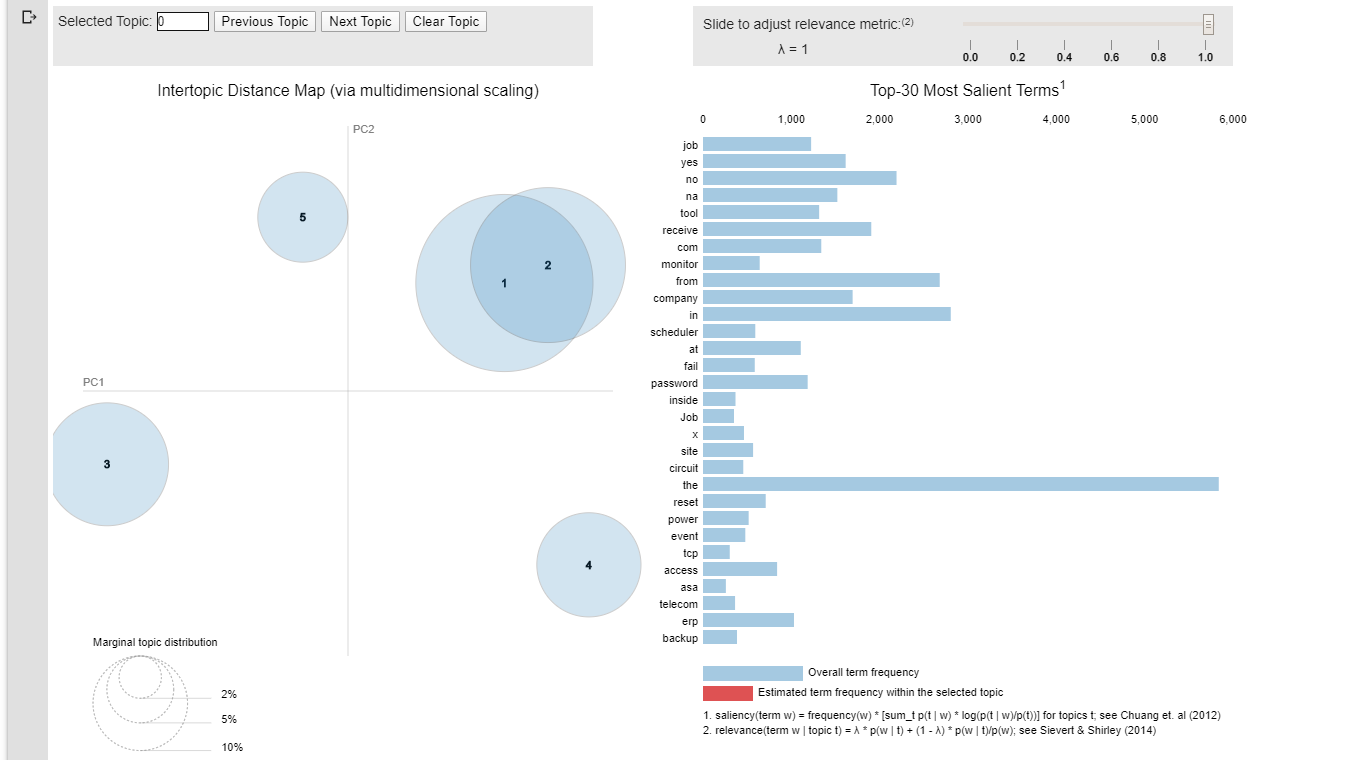
## Plot showing Frequency Distribution of words

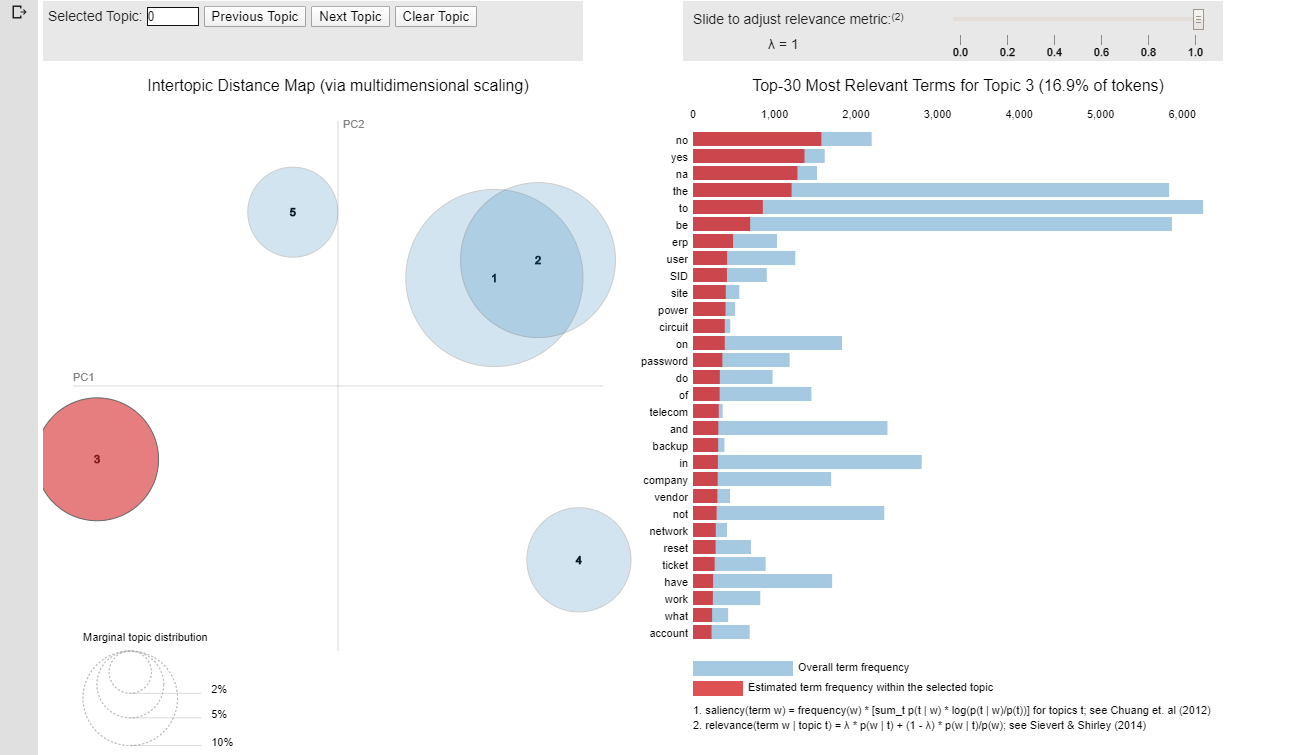
****

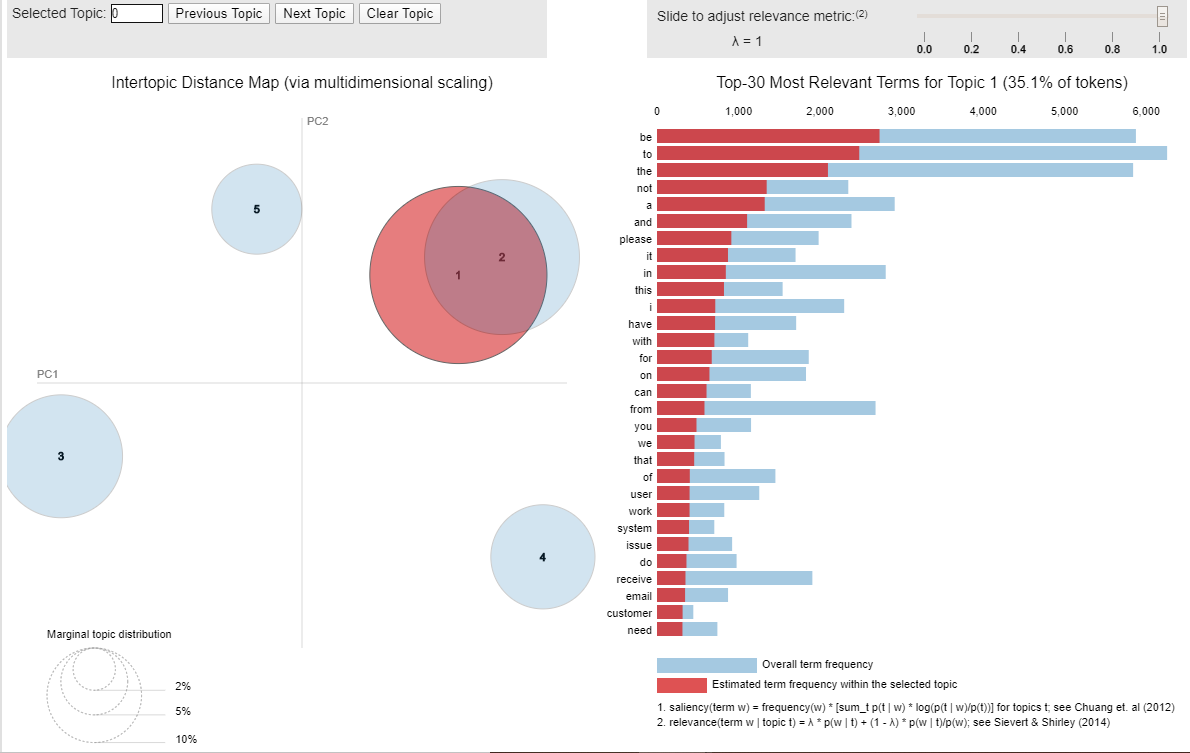
## Inter-topic Distance Map

LDA’s approach to topic modelling is it considers each document as a collection of topics in a certain proportion. And each topic as a collection of keywords, again, in a certain proportion.

Inter-topic Distance Map (via multidimensional scaling) showing Marginal topic distribution

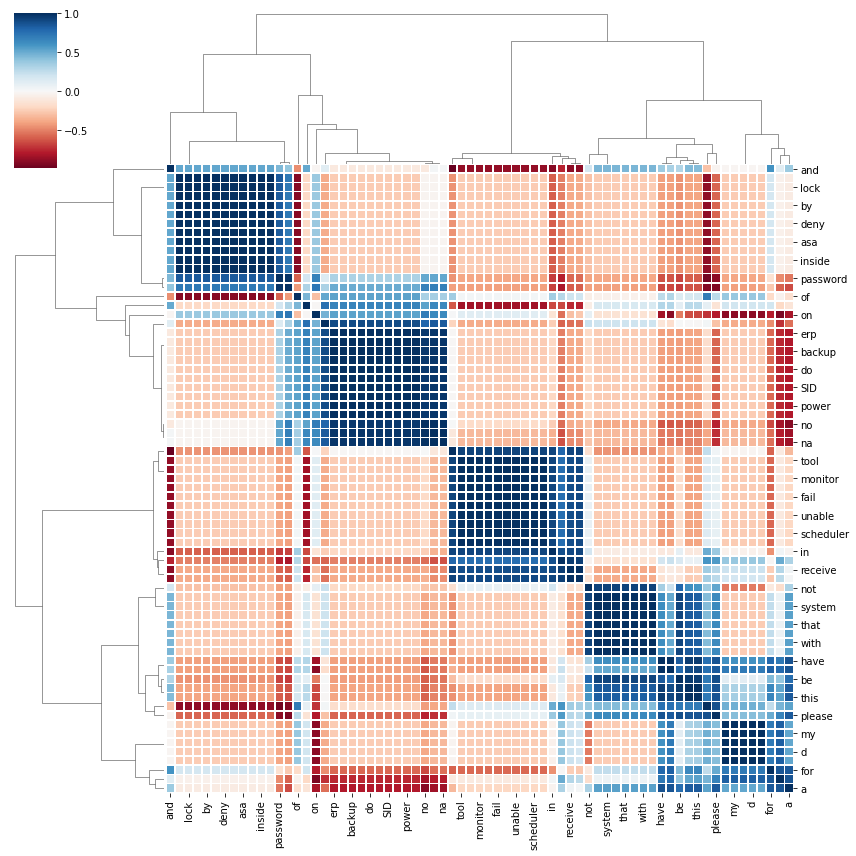
****

****

****

## Heat Map

LDA’s approach to topic modelling considers each document as a collection of topics in a certain proportion. And each topic as a collection of keywords, again, in a certain proportion.

****

# **Model building and evaluation**

## Model Approach

Solution requires model building based on the Classification model approach to predict the ticket details and assigned to expected group for quicker resolution.

We can approach solution with both Conventional model and using NLP.

In Conventional Model, we are using Logistic regression, Random Forest and KNN models to predict the ticket group

The second approach is using LSTM and Bi-directional LSTM

## Model creation

Following Model and accuracy scores are given as per the initial interim stage.

Further Model tuning and performance has been given in the next section

### Logistic Regression Model

Using TfidfTransformer library in sklearn, bag of words is created to get the vocabulary (ngram 1,3)

Using Vectorizer transformation, features are mapped to training.

Logistic regression model is created and trained with 75-25 train test split.

**Logistic Regression Summary**

**Train Accuracy: 94%**

**Test Accuracy: 74%**

Another Logistic model is also built after balancing the input data using imblearn library

**Logistic Regression with Balanced Training data Summary**

**Train Accuracy: 94%**

**Test Accuracy: 70%**

### Random Forest Model

Random Forest Model created and trained with 75 – 25 Train Test Split. The parameters for the Random forest model are

max\_depth=15

class\_weight="balanced"

**Random Forest Summary**

**Train Accuracy: 93.80%**

**Test Accuracy: 59.88%**

### KNN Model

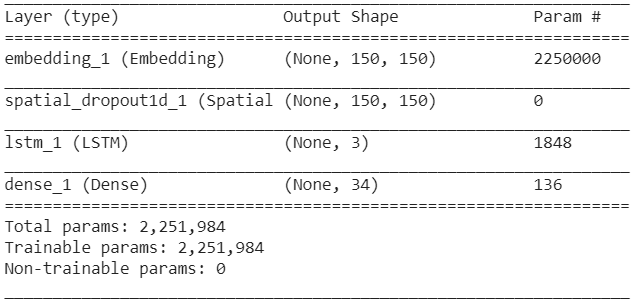
Model created and trained using KneighboursClassifier using neighbours = 5 and weights as distance

**KNN algorithm**

**Training Accuracy: 93.26%**

**Testing Accuracy: 54.84%**

### LSTM Model

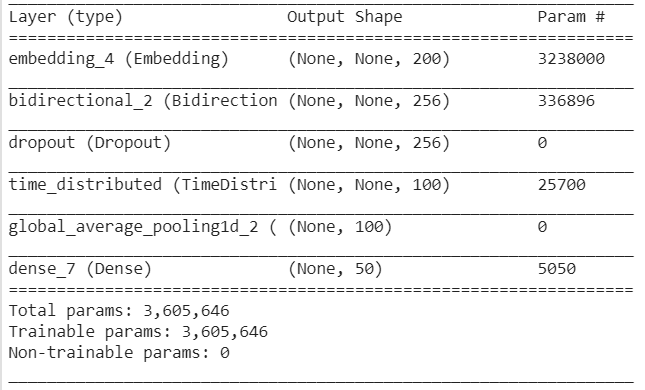
****

**Training Accuracy 58.9%**

**Testing Accuracy 57.0%**

### Bidirectional LSTM with Time Distributed

Embedding model is created and trained using glove 6B 200 Dimensions

****

**Train Accuracy is 90%**

**Test Accuracy is 54.9%**

## Model Summary

Summary of Model outputs

Below are the accuracy based on the Classes as 34

|  |  |  |  |
| --- | --- | --- | --- |
| **Srl**  **No** | **Model** | **Train**  **Accuracy %** | **Test Accuracy %** |
| 1 | Logistic Regression | 94 | 74 |
| 2 | Logistic Regression with Balanced data | 94 | 70 |
| 3 | Random Forest | 93.80 | 59.88 |
| 5 | KNN | 93.26 | 54.84 |
| 6 | LSTM | 58.9 | 57.0 |
| 7 | Bidirectional LSTM with Time Distributed | 90 | 54.9 |

# **Bench Marking - Comparison of experiments**

## Traditional Models

Model has been tried to have Classes as 5 groups

### Output of Logistic Regression

Train Accuracy – 96; Test Accuracy - 94

**Confusion Matrix**

array([[912, 2, 0, 0, 2],

[ 14, 37, 0, 4, 0],

[ 5, 0, 18, 0, 0],

[ 4, 7, 0, 163, 9],

[ 11, 0, 0, 22, 24]])

**Classification report**

Precision recall f1-score support

GRP\_0 0.96 1.00 0.98 916

GRP\_8 0.80 0.67 0.73 55

GRP\_24 1.00 0.78 0.88 23

GRP\_12 0.86 0.89 0.88 183

GRP\_9 0.69 0.42 0.52 57

accuracy 0.94 1234

macro avg 0.86 0.75 0.80 1234

weighted avg 0.93 0.94 0.93 1234

### Output of Logistic Regression with Balanced Data

Train Accuracy – 96 ; Test Accuracy : 91

**Classification report**

Precision recall f1-score support

GRP\_0 0.98 0.98 0.98 916

GRP\_8 0.68 0.75 0.71 55

GRP\_24 0.95 0.91 0.93 23

GRP\_12 0.97 0.62 0.76 183

GRP\_9 0.44 0.86 0.58 57

accuracy 0.91 1234

macro avg 0.80 0.82 0.79 1234

weighted avg 0.94 0.91 0.92 1234

### Output of Random Forest

Train Accuracy : 95.4; Test Accuracy : 87.5

**Classification report**

Precision recall f1-score support

GRP\_0 0.91 1.00 0.95 915

GRP\_8 0.73 0.18 0.29 62

GRP\_24 0.94 0.77 0.85 22

GRP\_12 0.97 0.58 0.73 180

GRP\_9 0.38 0.62 0.47 55

accuracy 0.88 1234

macro avg 0.79 0.63 0.66 1234

weighted avg 0.89 0.88 0.86 1234

### Output of KNN

Training Accuracy : 95.30 ; Testing Accuracy : 85

**Confusion Matrix**

[[900 1 4 10 0]

[ 49 9 0 0 4]

[ 17 0 5 0 0]

[ 23 5 0 105 47]

[ 20 0 0 5 30]]

**Classification report**

precision recall f1-score support

0 0.89 0.98 0.94 915

1 0.60 0.15 0.23 62

2 0.56 0.23 0.32 22

3 0.88 0.58 0.70 180

4 0.37 0.55 0.44 55

accuracy 0.85 1234

macro avg 0.66 0.50 0.53 1234

weighted avg 0.85 0.85 0.83 1234

### Summary of ML Models

Although the outcome of Logistic regression accuracy is good, but the model is not expected to perform well for text processing of high volume of data.

Random Forest and KNN are not performed well.

## NLP Models

## Approach to select the final model based on the outcome

### LSTM

Initial LSTM model was created using

Embedding with SpatialDropout1D(0.05)

LSTM(3, dropout=0.05, recurrent\_dropout=0.05)

Dense(34, activation='softmax')

34 Assignment Group classes have been taken.

The model is little modified with

LSTM(128, dropout=0.2, recurrent\_dropout=0.2)(embedded\_sequences)

Dense(34, activation='softmax')

As a result, we got good training accuracy but no improvement in testing accuracy.

Various combination of parameters changed to check the accuracy

|  |  |  |
| --- | --- | --- |
| Parameters | Training Accuracy | Testing Accuracy |
| epochs = 10;batch\_size = 100  DIM EMBEDDINGS – 50, MAX SEQ LENGTH = 170 | 70 | 57 |
| epochs = 10;batch\_size = 100  DIM EMBEDDINGS – 50, MAX SEQ LENGTH = 200 | 74 | 61 |
| epochs = 15;batch\_size = 100  DIM EMBEDDINGS – 60, MAX SEQ LENGTH = 250 | 89.9 | 61.23 |
| epochs = 15;batch\_size = 100  DIM EMBEDDINGS – 80, MAX SEQ LENGTH = 250 | 90.78 | 63.89 |
| epochs = 15;batch\_size = 100  DIM EMBEDDINGS – 100, MAX SEQ LENGTH = 250 | 93.83 | 62.97 |

Training Accuracy is increased beyond 90, but no significant difference in Testing Accuracy

Tried in changing N\_Classes Parameter.

|  |  |  |
| --- | --- | --- |
| N\_Classes | Training Accuracy | Testing Accuracy |
| 74 | 92 | 68 |
| 10 | 95.47 | 76.45 |
| 8 | 96.49 | 78.02 |
| 5 | 94.10 | 90.76 |

Since Assignment Group are having imbalanced data, testing accuracy is not being improved with more classes.

Top 5 groups has been given as input for classes, it has given good results.

Training Accuracy - 94.10, Testing Accuracy – 90.76

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Model: "model\_5"

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Layer (type) Output Shape Param #

=================================================================

input\_6 (InputLayer) [(None, 250)] 0

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

embedding\_5 (Embedding) (None, 250, 100) 1500000

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

lstm\_5 (LSTM) (None, 128) 117248

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

dense\_5 (Dense) (None, 5) 645

=================================================================

Total params: 1,617,893

Trainable params: 1,617,893

Non-trainable params: 0

### Bidirectional LSTM with Time distributed

Similar pattern is observed in Bidirectional LSTM with Time distributed as well.

|  |  |  |
| --- | --- | --- |
| N\_Classes | Training Accuracy | Testing Accuracy |
| 34 | 84.4 | 57.5 |
| 8 | 93.6 | 81 |
| 5 | 91.38 | 88.96 |

Model: "model\_7"

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Layer (type) Output Shape Param #

=================================================================

input\_8 (InputLayer) [(None, 100)] 0

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

embedding\_7 (Embedding) (None, 100, 200) 2129400

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

bidirectional\_1 (Bidirection (None, 100, 200) 240800

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

batch\_normalization\_1 (Batch (None, 100, 200) 800

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

time\_distributed\_1 (TimeDist (None, 100, 100) 20100

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

flatten\_1 (Flatten) (None, 10000) 0

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

dense\_10 (Dense) (None, 100) 1000100

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

dense\_11 (Dense) (None, 34) 3434

=================================================================

Total params: 3,394,634

Trainable params: 3,394,234

Non-trainable params: 400

### NLP Model Summary

LSTM model has performed well when compared to Bidirectional LSTM.

It shows bidirectional wont perform well for text processing as there would be no additional significant difference when performing bidirectional

# **Implication - business value derived**

Due to Manual assignment of tickets to the group causes delay in resolution of tickets. Manual assignment increases the response and resolution times which result in user satisfaction deterioration and poor customer service.

Applying NLP to automatically classify tickets and assign them to the right owner in a timely manner to save effort, increase user satisfaction and improves handling of support ticketing system.

Nevertheless, Business need to give the input balanced data to derive this benefit

# **Limitations and Scope of improvement**

## Limitations of data

Input data plays a vital role in creation of model and deriving the output from the model to get the expected business benefit.

The input data that was provided have multiple combination of languages.

Due to language translation limitations, some of the data has not taken into account for processing.

There is an imbalanced data within the Assignment groups that caused lot of noises

As a result, we have taken only top 5 classes to the model to arrive good training and testing accuracy.

The team had a short of time to change the way of pre-processing and to try in a different approach. This is one of the reason as well.

## Scope of Improvement

As mentioned in the limitation,

1. The pre-processing can be improved
2. Requesting business to give proper balanced data between the groups
3. Collecting more data for other Classes and reduce class difference within 15%
4. Equal and Standard unique words 200 or 500 on all the groups
5. Smote synthetic data.
6. Combine the similar classes with business decisions

The model is open for further improvement.

# **Closing Reflection**

1. With 34 classes that comprised 94% imbalanced data. The model accuracy was less and lead to over fit
2. Top 5 classes taken for evaluation comprises 64% data. It has given a generalized model that has good model accuracy scores.

**Learnings**

* Had a good opportunity to learn preprocessing of text using various modules
* Learnt and used Language detection modules and translation
* Visualization using Word cloud to visualize the different combination of words in a group
* Visualizing Frequency distribution of words, Heat Map and Inter-topic Distance Map (via multidimensional scaling)
* Handling multi-classes distribution
* Handling imbalanced data and its implication

# **Final Note**

Thanks to Great Learning team for the help to learn AIML and to do this Capstone Project

Thanks to Abhijeet who has helped in many ways to complete this course

Many thanks to our Mentor Abhishek, his experience in the field of AIML has guided us in learning throughout this course. The team appreciates his patience. His practical knowledge gives us a lot of insights to tackle the issues.

# **Code and References**

<https://colab.research.google.com/drive/1jhpJnW5QXVkxM3UmriEiRzEJYHkKKEZ4>