# 

# **Automatic Ticket Assignment**

# 

# 

# 

# 

# **Team Members:**

Jayashree Nagaraja

Madhavi Evani

Yash Shah

Madhavi Evani

Madhavi Evani

Madhavi Evani

# 

# 

[**Automatic Ticket Assignment**](#_hkkdv384dxv0) **1**

[**Team Members:**](#_h4lq71lm06gj) **1**

[**Problem Statement**](#_4o72zogw26sw) **3**

[Background](#_3l1a5ooidrcz) 3

[Business Domain Value](#_xwwxd32uui2u) 3

[**Solution**](#_x5vozcqv6q39) **4**

[Overview](#_2sr3ia6vq35a) 4

[Data](#_um8hxcnul0ve) 4

[Data - Observation](#_pnvvix5lu6t8) 5

[Data - Preprocessing](#_jdzd0mc4w2dq) 8

[Target Class](#_dnqnoiapkvv0) 8

[Short Description and Description](#_e0i1lykeb6rm) 8

[Modelling](#_sugbxd149e9y) 11

[Word Embedding:](#_tpzy1riy4bum) 11

[Bi-Directional LSTM:](#_7u63u4vjumfs) 11

[BERT & DistilBERT:](#_ovb2bgd7k92m) 13

[Fasttext:](#_xj40o3ww72w8) 15

[Random Forest:](#_ciyees7kiut8) 16

[Model Evaluation](#_bleqx3kqxr5d) 16

[Implications](#_v3p17q6068wc) 18

[Limitations](#_k2vy0ts6u815) 18

[Future roadmap](#_68lgycupmctr) 18

# 

# 

# **Problem Statement**

Build an AI-Classifier which assigns the IT-Tickets to Groups using the text in the Description of the tickets.

## **Background**

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 the Incident Management process to achieve the above Objective. An incident is something that is an 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 the 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.

## **Business Domain Value**

In the Incident Management process, incoming incidents are analyzed and assessed by the organization's support teams to fulfill the request. In many organizations, better allocation and effective usage of the valuable support resources will directly result in substantial cost savings.

The incidents are initially assigned to Service Desk teams i.e L1/L2 teams. These teams review to either resolve or re-assign the tickets to correct functional teams(L3 team). It is analyzed that a Minimum of ~1 FTE effort needed only for incident assignment to correct teams.

During the process of incident assignments by L1 / L2 teams to functional groups, there were multiple instances of incidents getting assigned to wrong functional groups. Around ~25% of Incidents are wrongly assigned to functional teams. Assigning the tickets to wrong teams will lead to either of the following problems which result in additional effort to resolve a ticket:

1. Additional effort needed for Functional teams to re-assign to right functional groups.
2. Some of the incidents are in queue and not addressed timely resulting in poor customer service.

The idea is to build classifiers using AI which can analyze the incoming tickets and assign to the correct functional teams using similarities in the previous tickets and their classification.

# **Solution**

## **Overview**

We have analyzed the given data set which has 8500 entries with 4 columns. Preprocessed the data using the following steps :

1. Process the skewed target group column by reducing the number of groups to be classified.
2. Clean the Description attribute which has a lot of junk data like “to”, “from” , “http” links.
3. Translated the Description attribute to english from other non-english languages
4. Performed Lemmatization using NLTK WordNet Lemmatizer
5. Removed stop words using the stop words available in NLTK.

We then used this data with the following models:

1. Bi-Directional LSTM with Glove word embeddings and custom embedding.
2. FASTTEXT
3. BERT
4. DistilBERT
5. Random forest

## **Data**

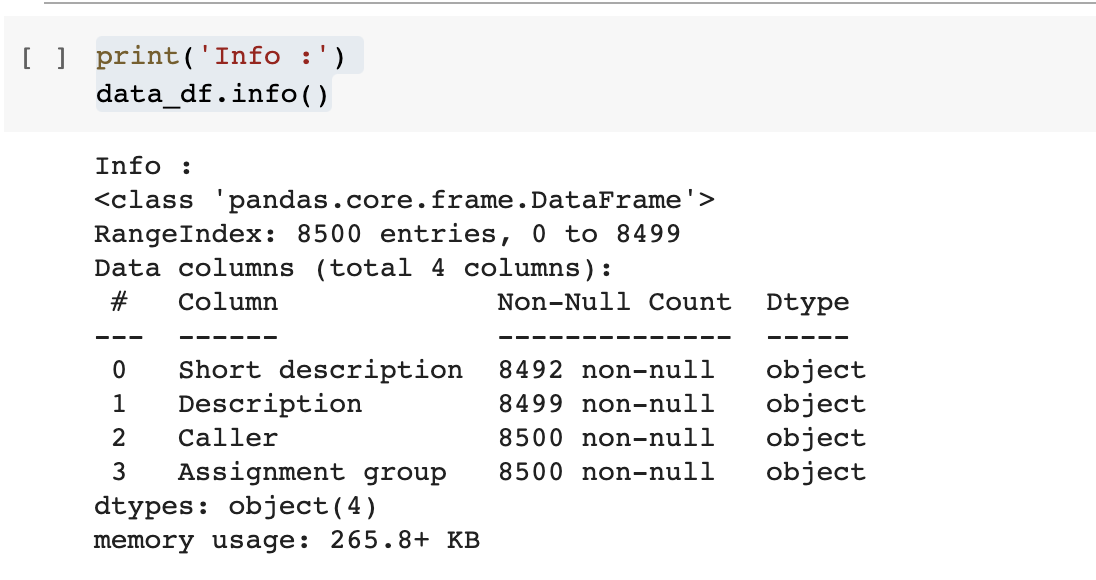
For this project we have used the IT Ticket provided to us for building a classifier model. The data consists of 4 fields

1. Caller
2. Short Description
3. Description
4. Assignment Group

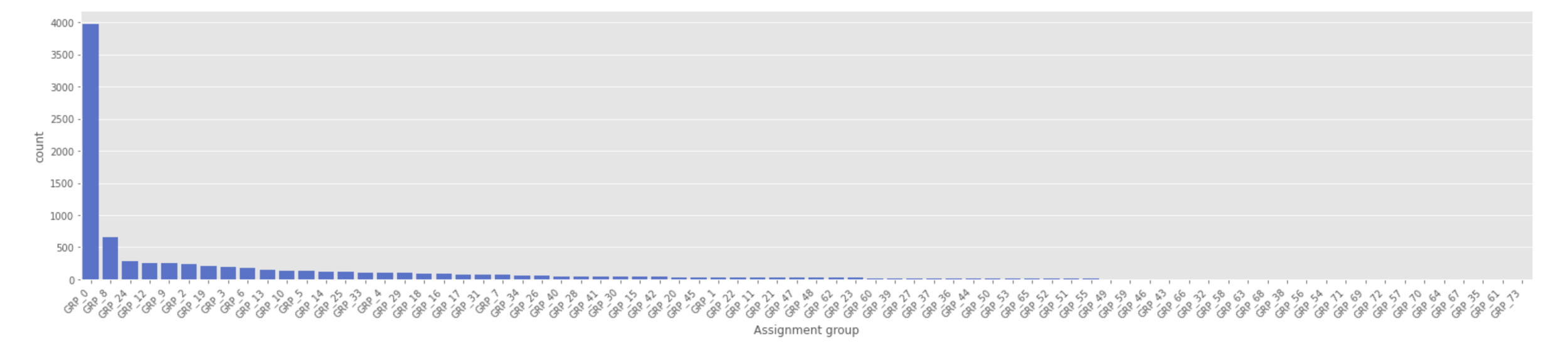
The “Assignment Group” is the functional group to which the tickets are assigned to by the L1/L2 teams.

## **Data - Observation**

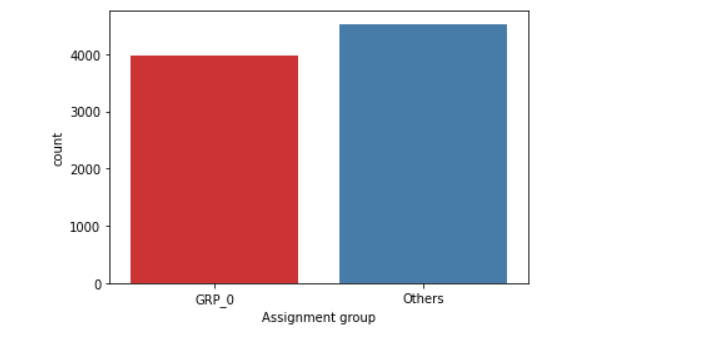
1. The data set consisted of a total of 8500 entries with 4 columns.



1. We have found a total of 74 groups with skewed distribution. Group 0 has almost 50% of the tickets in the data set.



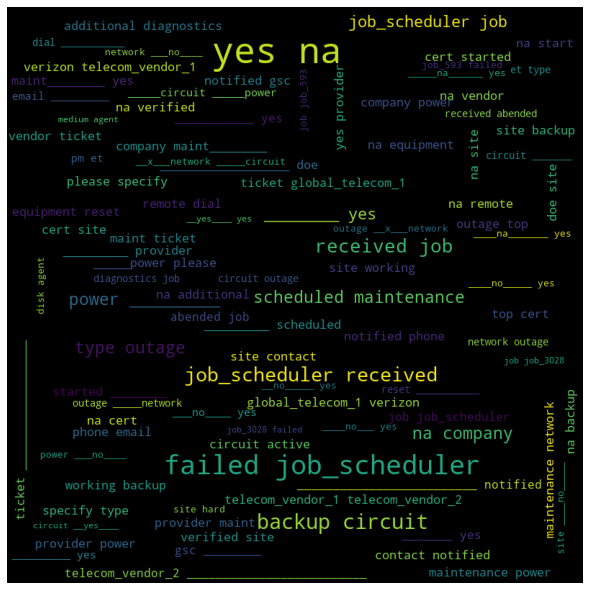
1. “Callers” attribute is the name of the person raising the incident.
2. Non-English language is also found in “Short Description” and “Description” fields
3. The “Description” field has one null value. The “Short Description” column has around 8 rows which are null.
4. The “Description” column has values in email format such as “To: “, “From:”, “CC:”, “Subject:”
5. Some of rows have same “Short Description” and “Description”



1. Word cloud is an image composed of words, in which the size of each word indicates its frequency or importance. Word cloud of top 5 Assignment group and important words used in the incidents



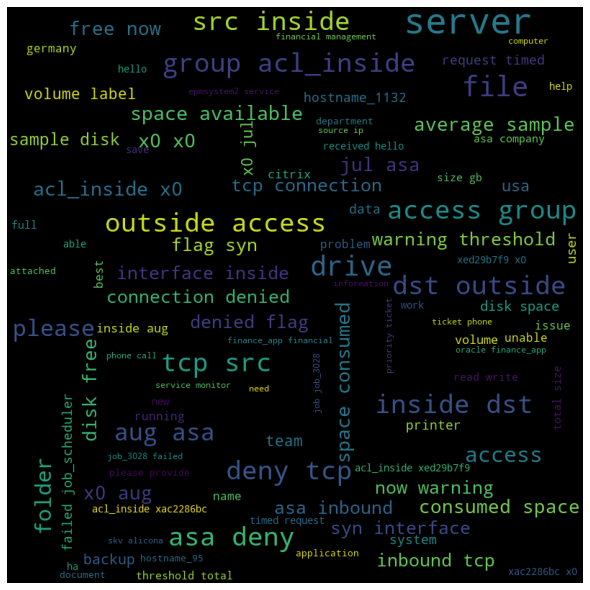
Group 0



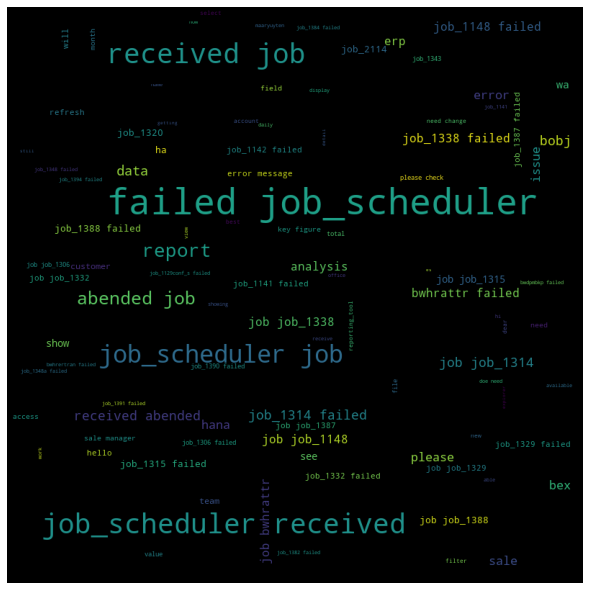
Group 8



Group 24



Group 12



Group 9

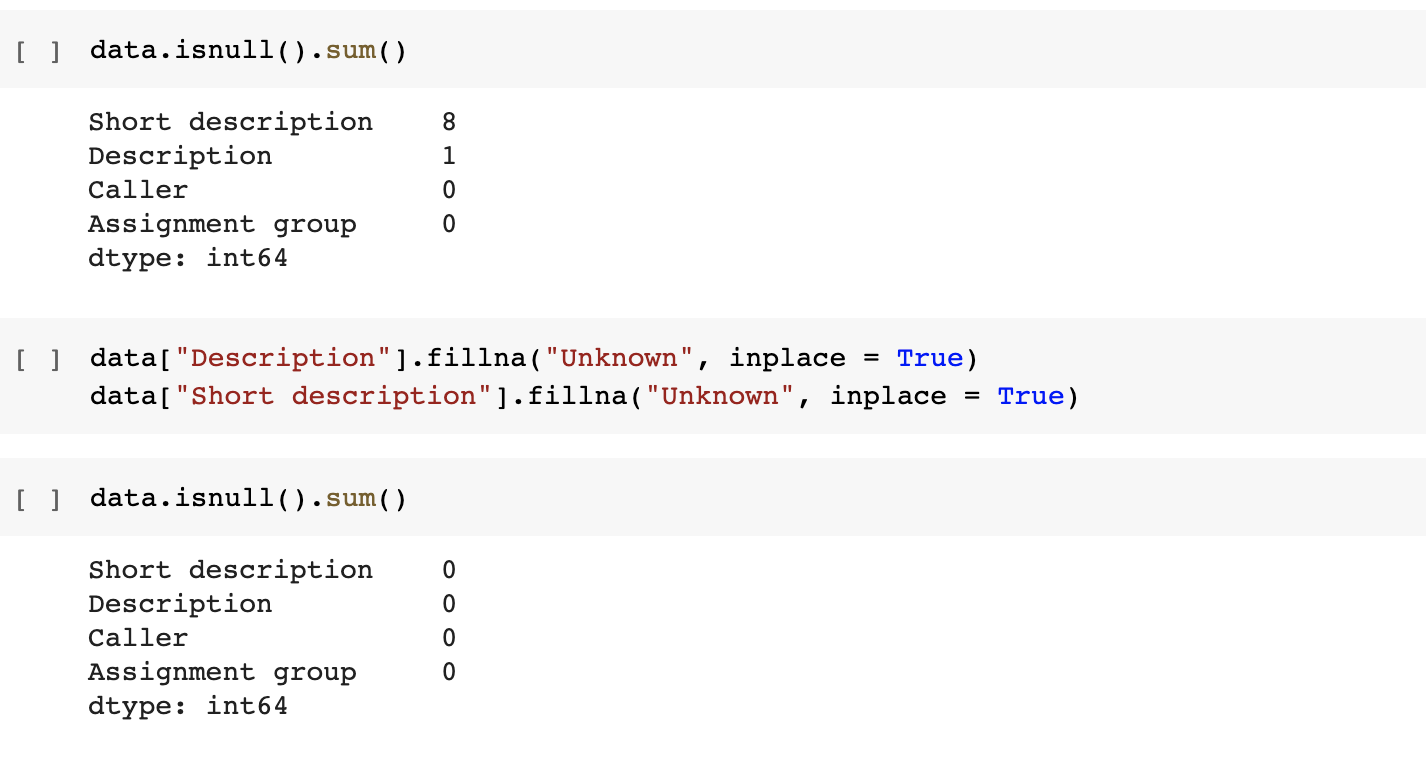
## **Data - Preprocessing**

### Target Class

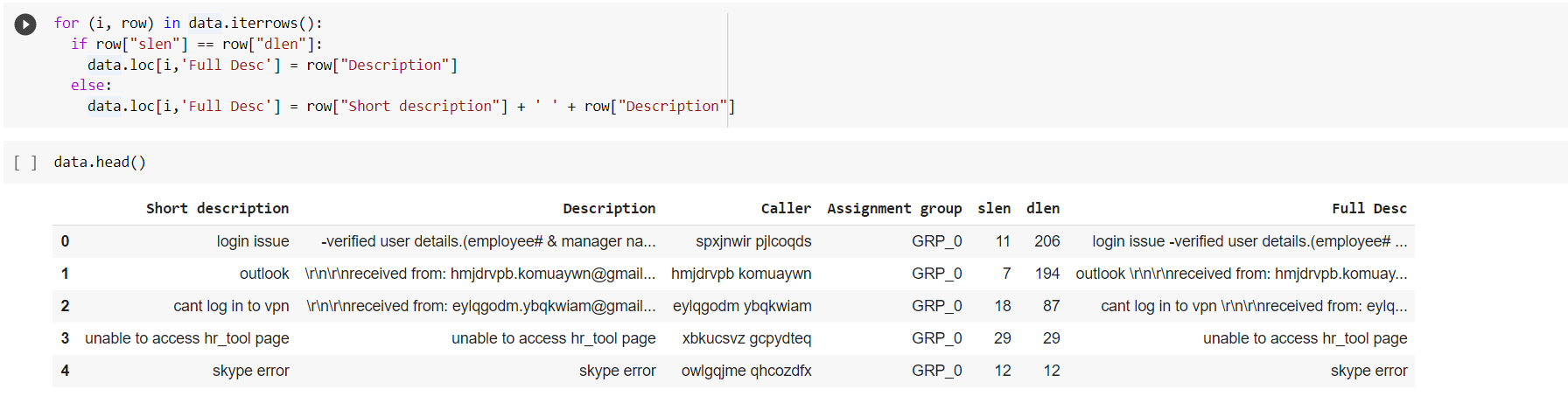
We have observed that the target class “Assignment Group” is extremely skewed. There are some Groups which have just 1 entry making it difficult to test/train the data. To address this issue, we have merged all the target classes other than the first five into a group called “GRP\_D”.

### Short Description and Description

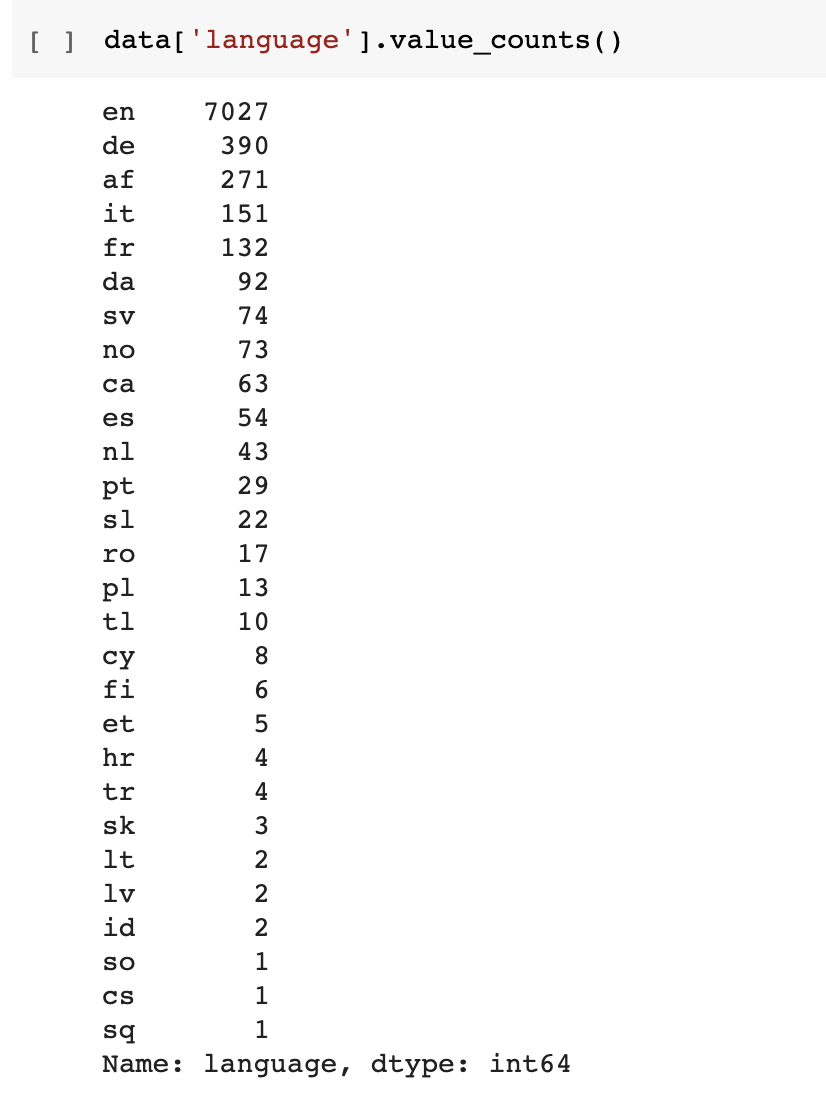
1. We are handling null values by replacing them with values as “Unknown” in both Description and Short Description columns.



1. Concatenating Short Description and Description attribute to obtain “Full Description”, in case both attributes consist of the same information then only “Short Description” is being considered for training.



1. Cleaned the “Full Description” Column to remove lines like “From”, “To” e.t.c using the clean\_data function.
2. We process data before and after translation. Some are listed below:
   1. Convert to lower case.
   2. Remove Email and hyperlinks
   3. Remove caller names from the text
   4. Remove digits except the ones followed by an alphabet or underscore to retain JOB\_2123 format to help in classification
   5. Remove white space and multispace
   6. After Translation, removed the unicode characters which were not translated to english
   7. Remove all special characters.
3. We have **translated all the non-english values** in “Full Description” attribute to english to ensure common language for the data used for training. The preprocessed data is then saved in the file for later use.



1. Stop words have been removed using NLTK corpus module.
2. We have used WordNetLemmatization to Lemmatize the dataset which is part of the NLTK module.

## **Modelling**

Dataset is extremely skewed, so less occuraning groups have been remapped to a dummy group before training a few models.

### Word Embedding:

Word embedding is a learned representation for text where words that have the same meaning have a similar representation.

We have experimented with below mentioned types

1. GloVe (Global Vectors) Embedding with dimension 50 and 100: It is an unsupervised learning algorithm for obtaining vector representations for words. Training is performed on aggregated global word-word co-occurrence statistics from a corpus, and the resulting representations showcase interesting linear substructures of word vector space. We have tried both Glove 50d and 100d embeddings, 100d embeddings works best for this dataset.
2. Text processing for BERT model: It is a combination of 3 types of embeddings.

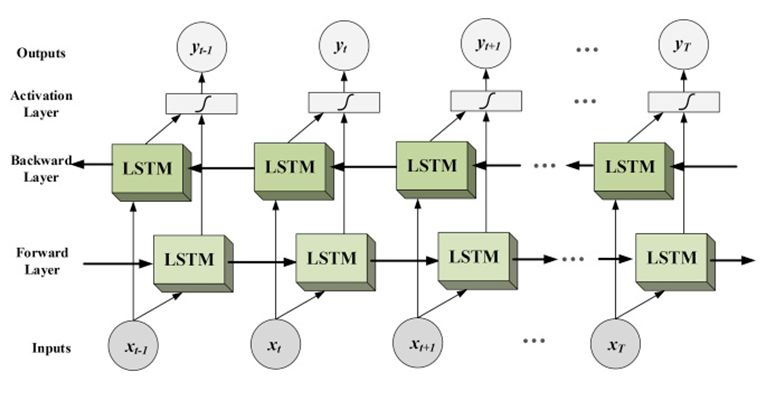
Position Embeddings: BERT learns and uses positional embeddings to express the position of words in a sentence.

Segment Embeddings: BERT can also take sentence pairs as inputs for tasks (Question-Answering). That’s why it learns a unique embedding for the first and the second sentences to help the model distinguish between them.

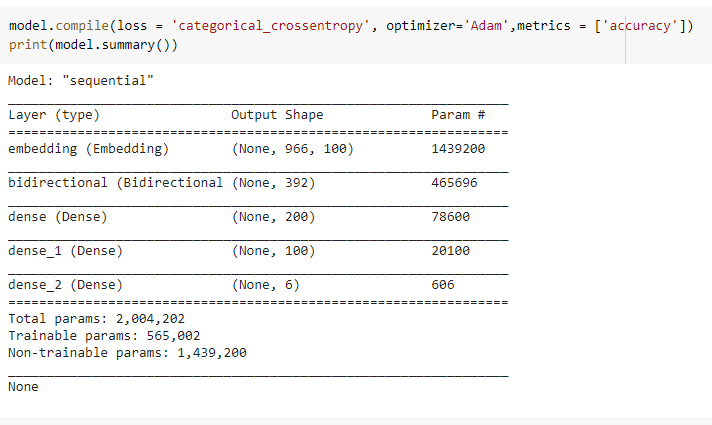
Token Embeddings: These are the embeddings learned for the specific token from the WordPiece token vocabulary.

### Bi-Directional LSTM:

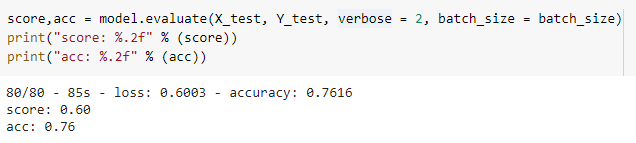
Bidirectional LSTMs is an extension of traditional LSTMs that can improve model performance on sequence classification problems. In problems where all timesteps of the input sequence are available, Bidirectional LSTMs train two instead of one LSTMs on the input sequence. The first on the input sequence as-is and the second on a reversed copy of the input sequence. This can provide additional context to the network and result in faster and even fuller learning on the problem.

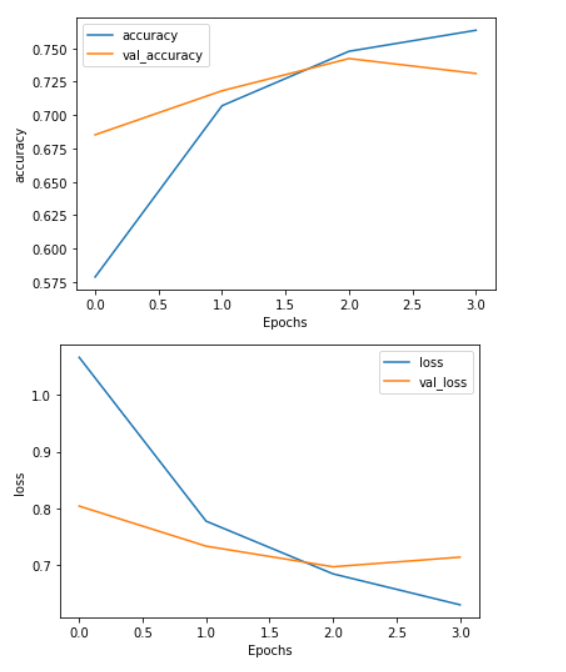


Summary of Bi-Directional LSTM model used is as shown below.



Training accuracy



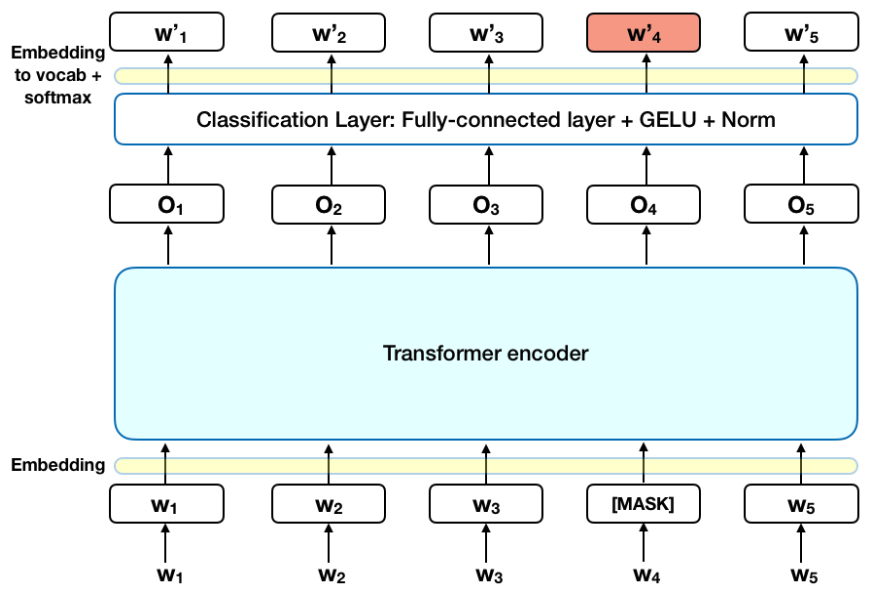


In Dense layers Leaky ReLU activation function has been used. Performance of the model was better with Leaky ReLU compared to ReLU activation function. We have tried Multilayer Bi-directional LSTM also, accuracy was almost the same as one layer Bi-directional LSTM.There was an increase in execution timings so we did not go for it.

### BERT & DistilBERT:

BERT is designed to pretrain deep bidirectional representations from unlabeled text by jointly conditioning on both left and right context in all layers. BERT stands for **“Bi-Directional Representations from Transformers”**. It is a deeply bidirectional model. Bidirectional means that BERT learns information from both the left and the right side of a token’s context during the training phase. **DistilBERT** is a distilled version of the BERT model, it is smaller,faster, cheaper and lighter.

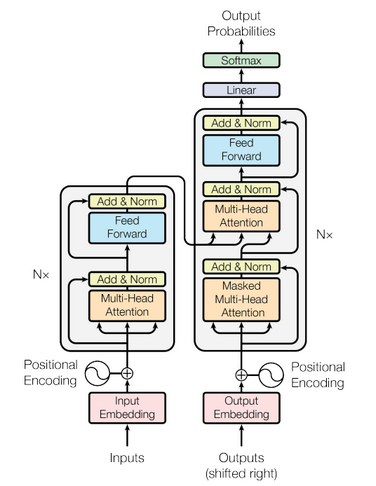
Architecture of BERT model



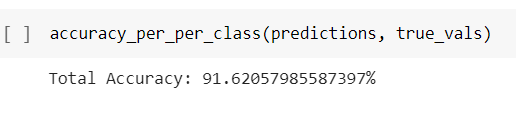
The BERT architecture builds on top of the Transformer.

* BERT Base: 12 layers (transformer blocks), 12 attention heads, and 110 million parameters
* BERT Large: 24 layers (transformer blocks), 16 attention heads and, 340 million parameters

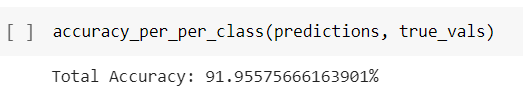
Architecture of each Transformer block



Accuracy for DistilBERT



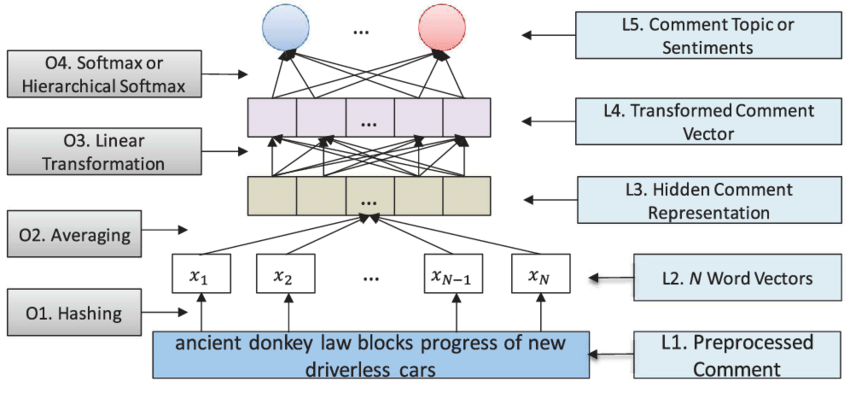
Accuracy for BERT



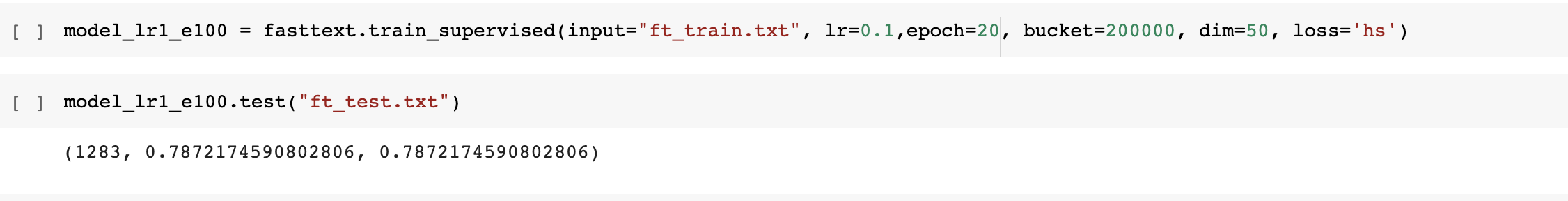
### Fasttext:

FastText uses a simple and efficient baseline for sentence classification. It uses negative sampling, hierarchical softmax and N-gram features to reduce computational cost and improve efficiency.

Architecture of FastText Model



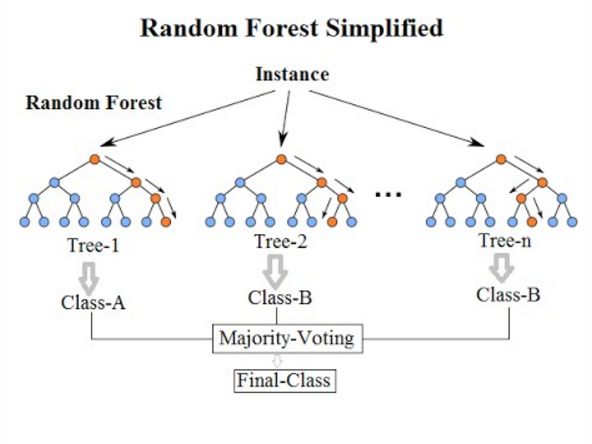
Fasttext Model Training and Testing

****

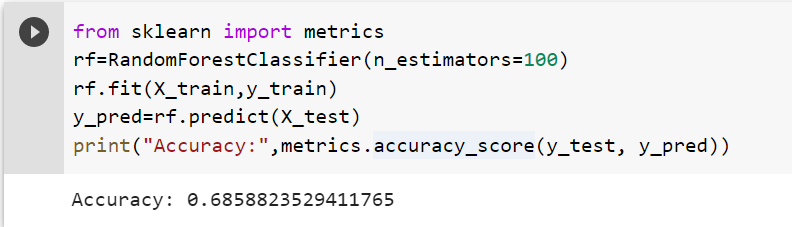
### Random Forest:

Random forest is a supervised learning algorithm which is used for both classification as well as regression. It creates decision trees on data samples and then gets the prediction from each of them and finally selects the best solution by means of voting. It is an ensemble technique.

Architecture of Random Forest



Training accuracy

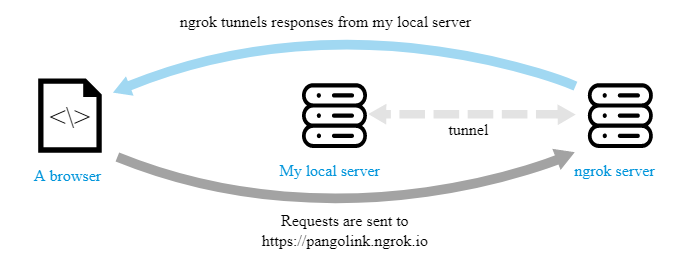


## **Model Evaluation**

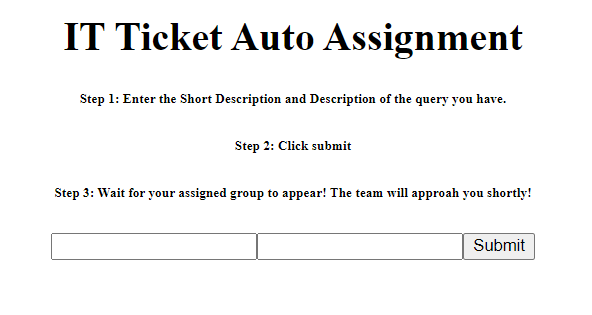
|  |  |
| --- | --- |
| **Model** | **Accuracy** |
| RandomForest | 68.5% |
| Bi-directional LSTM | 76% |
| **Fasttext** | **91.16%** |
| **Bert** | **91.95%** |
| **DistilBert** | **91.62%** |

**Model Deployment**

1. Deployment of the model means making it available for the end user is done using Dash app as frontend and Flask web application as backend.
2. Steps involved are:
3. Write a Dash App and Flask Application which has below parts
4. app.py — This contains Flask APIs that receives ticket details through GUI or API calls, performs preprocessing steps and then computes the predicted value based on our fasttext model.
5. Then it uses the trained model to make a prediction, and returns that prediction, which can be accessed through the API endpoint.
6. Dash App is built on top of Flask to get details through GUI and respond back to the end user with the assigned group.



**WebApp UI Demo**



## **Implications**

In the Incident Management process, incoming incidents will be assigned to correct functional teams(L3 team) thereby reducing the effort of Service Desk teams ie:L1/L2 teams. Model uses similarities in the previous tickets and their classification to predict upcoming tickets Assignment group.

## **Limitations**

Assignment groups with very few incidents won’t be trained properly, so manual assignment will be required for these groups by the Service Desk team.

## **Future roadmap**

1. New Datasample for groups with less number of tickets will be required to predict efficiently for those groups.
2. Training the model in different languages instead of converting the dataset to English.